Improvements in track to train communications for railway jointless track circuits

Collins, M. J.

Award date:
1997

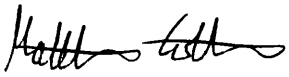
Awarding institution:
University of Bath

IMPROVEMENTS IN TRACK TO TRAIN COMMUNICATIONS FOR RAILWAY JOINTLESS TRACK CIRCUITS

Submitted by
M. J. Collins

for the degree of Ph.D.
of the University of Bath
1997

COPYRIGHT



M. J. Collins

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Summary

The track circuit is the main element of fixed-block railway train detection. Audio frequency track circuits extend this function to allow the transmission of data to the train. There is a requirement to greatly increase the amount of track circuit data and since the existing track circuit equipment has been designed for a very low data capacity, new forms of modulation and coding must be considered. In section I of this thesis, a new track circuit design is described. The main feature of this design, in terms of track circuit data coding, is that the phase and amplitude distortion and restriction on bandwidth are removed. The potential improvement in train detection and track circuit data communications offered by this design are explored in this thesis.

A fully synchronisable serial binary coding scheme for track circuits is presented in section II. The synchronisation performance is assessed using Gaussian white noise as well as an impulsive interference model based on typical railway interference. It is shown that this coding scheme is efficient and robust synchronisation can be achieved with a low amount of redundancy compared to other synchronisable coding methods.

Section III is concerned with new modulation schemes that can be applied to the track circuit, with the realistic requirement that signals will be fed through the existing tuned track circuit terminations in order to achieve a gradual upgrade path. This condition introduces additional phase and amplitude distortion and places an initial restriction on the signal bandwidth (≈200Hz), which makes a serial binary data stream unsuitable. Two modulation and coding schemes are proposed: a non-coherent MFSK scheme, and a coded multi-tone scheme. The later is chosen for further study of its demodulation and coding techniques in non-Gaussian noise.

Equipment cost and simplicity are factors that are considered throughout this thesis, since they can affect the safety and reliability and, importantly, railway authorities' acceptance of a proposed scheme. The algorithms used in the modulation/demodulation of the multi-tone scheme presented in section III could be considered as extensions of the algorithms of existing rate-coded track circuits. Error correction is also discussed in section III, a concept normally rejected by railway authorities. It is argued that error correction is appropriate, if safety is also taken into account. It is also demonstrated that the error correction algorithms used in this chapter are extensions of the techniques used in existing coded track circuits.

Table of Contents

Acknowledgements

The author would like to thank Mr J. D. Martin for his help and guidance throughout this project, and in particular his detailed reviewing at the final stages. I would also like to thank Dr. P. Sapiano (University of Bath) and Mr R. J. Roberts, Mr M. R. Reeves and Mr I. J. Lewis (Westinghouse Signals Ltd.) for their technical advice, and Mr D. W. Bethune for his invaluable help with the final presentation of this thesis. Finally, I would like to thank my parents for their unending support and encouragement.

This work was sponsored by Westinghouse Signals Ltd and the University of Bath.

Glossary of Acronyms

A/D	Analogue to Digital
AF	Audio Frequency
ARQ	Automatic Repeat Request
ASIC	Application Specific Integrated Circuit
ASK	Amplitude Shift Keying
ATC	Automatic Train Control
ATO	Automatic Train Operation
AWGN	Additive White Gaussian Noise
BCH	Bose Chaudhuri Hocquenghem
BEC	Binary Erasure Channel
BER	Bit Error Rate
BPF	Band Pass Filter
CO	Correct Operation
CWR	Continuously Welded Rail
DPSK	Differential Phase Shift Keying
DSP	Digital Signal Processing
FEC	Forward Error Correction
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FSK	Frequency Shift Keying
GF	Galois Field
I&D	Integrate and Dump
IBJ	Insulated Block Joints
IIR	Infinite Impulse Response
ISI	Inter-symbol Interference
JTC	Jointless Track Circuit
LPF	Low Pass Filter
LSB	Lower Side Band
LUL	London underground Limited
MDS	Modulation Derived Synchronisation
MFSK	Multiple Frequency Shift Keying
MPSK	Multiple Phase Shift Keying
MSDD	Minimum Soft Distance Decoding
MSK	Minimum Shift Keying

MTBF	Mean Time Between Failure
MTBSF	Mean Time Between Safe Failure
MTBUF	Mean Time Between Unsafe Failure
OQPSK	Offset Quadrature Phase Shift Keying
PSD	Power Spectrum Density
PSK	Phase Shift Keying
QPSK	Quadrature Phase Shift Keying
RS	Reed Solomon
SEMDD	Successive Erasure Minimum Distance Decoding
SF	Safe Failure
SNR	Signal to Noise Ratio
UF	Unsafe Failure
USB	Upper Side Band
WED	Weighted Erasure Decoding

For Kimberley

frequency track circuit. Also, the potential use of automatic train control (ATC) techniques such as brake assurance [8][9] to improve the train performance and minimise headway, will require more data to be transmitted continuously to the train (i.e. track circuit length, track gradient, distance to the next train, track circuit position).

The potential of the track circuit to meet these requirements has not been explored, and discouraging factors such as the passive tuning of the transmitter/receiver to a fixed frequency, the track distortion and the lower data capacity of audio frequencies [10][11] has led some designers to explore the use of transmission media other than the track circuit. These include inductive loops, passive transponders, radio waves and satellite communications [8][12][13][14][15]. However, the initial cost of these systems is high and as the existing train detection track circuit provides a convenient and cheap transmission medium, many railway authorities will continue to use track circuits to transmit automatic train control data for years to come.

This thesis explores the use of digital codes to increase the track circuit data capacity. A message will be composed of a number of discrete symbols. If the message is k symbols long and each symbol can be one of M values, there are M^k possible messages. To provide immunity to interference and cross talk, the different messages must be separated distinctly, or be orthogonal to each other. With the rate coded signals described above, this distinction is frequency. With digital messages, the data is encoded with redundancy and the messages are separated by the *minimum distance* between the code words. The encoder will transform the data block, u, of k symbols, to a coded data block, v, of n symbols, where $k<n$. Redundancy is therefore introduced and the total data rate has been reduced by a factor of k/n. To maintain the same data rate with the coded system, an expansion in the signal bandwidth is required. Since the audio frequency track circuit bandwidth is limited, the efficient use of redundancy is important and forms a major part of this thesis.

1.1 Error Control, Safety and Reliability

Signalling interference from electronic traction control and equipment failures are among the many possible causes of error in the track circuit coding. When these effects are present, the track circuit signalling will be affected in one of three ways:

1. The interference is sufficiently low so as to not disturb the signal, or alternatively the redundancy inherent in the signal will tolerate the interference.
2. The signal is corrupted such that the receiver decides that a valid signal is not present.

3. The interference is such that the signal data is changed into a different code word, or a valid signal is received while the track circuit is occupied.

Correct operation of the track circuit is described by (1) above. Case (2) corresponds to a safe failure, since the track circuit will falsely indicate 'track occupied', while case (3) corresponds to an unsafe failure or false 'track clear' condition. To meet the fail safe requirement the probability of (2) occurring must be sufficiently larger than (3). Safety and reliability measures in railway signalling are usually given in terms of mean time between unsafe failures (MTBUF) and mean time between safe failures (MTBSF).

There are few or no specific target figures on MTBUF and MTBSF as they relate to track circuit coding failures. D. Poole [16] states that, on British Rail's 68,000 track circuits there may be several transient wrong side failures per year. A mean time between failure of 5 years is quoted in [17]. A ratio of safe failures to unsafe failures is quoted as 4600 in [18] along with a rough estimate of MTBUF of 9 years. These figures however, relate to all track circuit failures and the majority are caused by factors such as faulty track circuit bonds and leads, connections to the rail and damage to cables and insulation. The number of transient failures causing detectable delays or undetected trains due to the track circuit signalling and coding is unknown.

In this thesis, the performance of a particular coding scheme is often represented by plots of correct operation, safe failure and unsafe failure. The plot shown below is for a particular code whose details will be discussed in chapter 8.

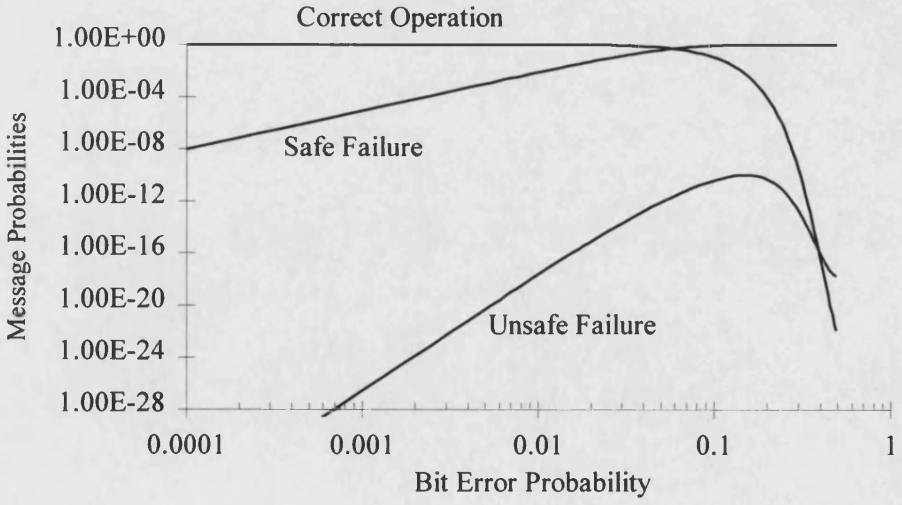


Figure 1.2. Sample Plot of Correct Operation, Safe Failure and Unsafe Failure for a Track Circuit Code against Bit Error Probability.

the modulation scheme, the SNR and the detection time (or track circuit code length) need to be known. For example, if the interference is Gaussian noise and non-coherent FSK is used, the relationship between bit error probability and SNR (dB) is given by:

$$P_e = \frac{1}{2}\exp\left(-\frac{10^{\frac{SNR(dB)}{10}}}{2}\right) \tag{1.1}$$

If the response time is 1s, then the MTBSF and MTBUF for the failure probabilities of figure 1.2 are shown in figure 1.3. The ratio of MTBUF/MTBSF is shown in figure 1.4.

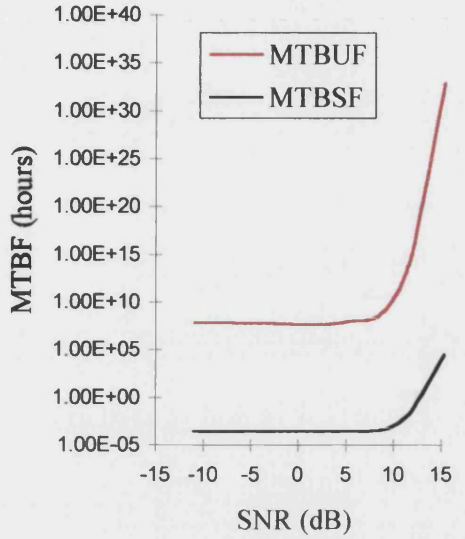


Figure 1.3. MTBSF and MTBUF against SNR for non-coherent FSK Modulation.

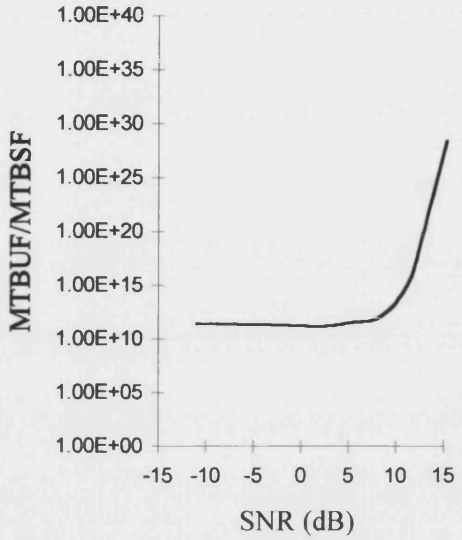


Figure 1.4. MTBUF/MTBSF against SNR for non-coherent FSK Modulation.

Hill [2] quotes track circuit signal to noise ratios of 10-13 dB. We can see that for the example above that the code is biased towards safety and that the ratio of MTBUF to MTBSF is considerably higher than the figure quoted in [18] at all SNR values.

Along with the difficulty in defining target MTBFs, the definition of a suitable SNR for which a track circuit coding scheme is to meet these targets can be equally difficult. The interference encountered in railway systems is likely to be impulsive/bursty noise of high amplitude for short periods of time. The amplitudes and time/frequency characteristics of this noise can depend on many location specific factors such as the number of trains, the location of the trains, the traction equipment, the type of electrification and the cross bonding. Unlike traditional telecommunication applications, track circuit data modulation techniques have not been optimised for a minimum power for a specific error rate. The transmitter power is primarily defined by the train detection function. Early track circuits were designed using ad hoc techniques, and the gradual upgrading from proven systems to present day systems has resulted in few specific figures for SNR or MTBF.

In the evaluation of coding schemes in this thesis, worst case interference or fault conditions causing high symbol error rates are considered. Under these conditions the coding scheme must be shown to have a significantly low unsafe failure rate compared to safe failure rate. For lower symbol error rates, the code and code parameters are chosen to optimise safety and reliability. The performance is assessed with Gaussian and non-Gaussian/impulsive noise models.

Interpretation of code performance curves

Although direct translation of the code failure probabilities to measurable values of MTBF is difficult if not impossible, the performance of a coded track circuit can still be optimised given the modulation scheme and data requirements. The code failure probability curves such as those of figure 1.2 should be realistically interpreted. For example, there may be little advantage in decreasing the theoretical probability of unsafe failure from, say 10^{-12} to 10^{-13}, since these two figures are equally unmeasurable. Similarly the reliability curves must be realistically interpreted. Reliable operation in 'railway' terms requires consistent correct decoding. If we say that reliability becomes affected when the safe failure rate rises above, say 10^{-2}, then there is little real advantage is halving the safe failure rate from say, 0.8 to 0.4.

1.2 Thesis Structure

This thesis has been split into three sections. Figure 1.5 shows how the thesis is developed and the subject area of each section. Section I includes chapters 2 to 6. It starts with an introduction to railway track circuits and existing track circuit systems. The design considerations and trade offs are discussed, with respect to train detection and data communications. Chapter 3 discusses the electrical modelling of railway track, for the purposes of simulating track circuit operation. Standard transmission line theory is applied to a two-wire model of railway track and transmission line parameter data is gathered from many sources.

Chapter 4 briefly describes the proposal for a new track circuit design which, together with chapter 5, sets the backdrop for the main thesis topic of track circuit coding and modulation schemes. The track circuit terminations are active and the potential signal bandwidth is increased as there is no passive tuning to one frequency. Also described in chapter 4 are the many potential ATO/ATP features of this track circuit, many of which require a greatly increased data capacity than is offered by any existing design. A simulation of the track

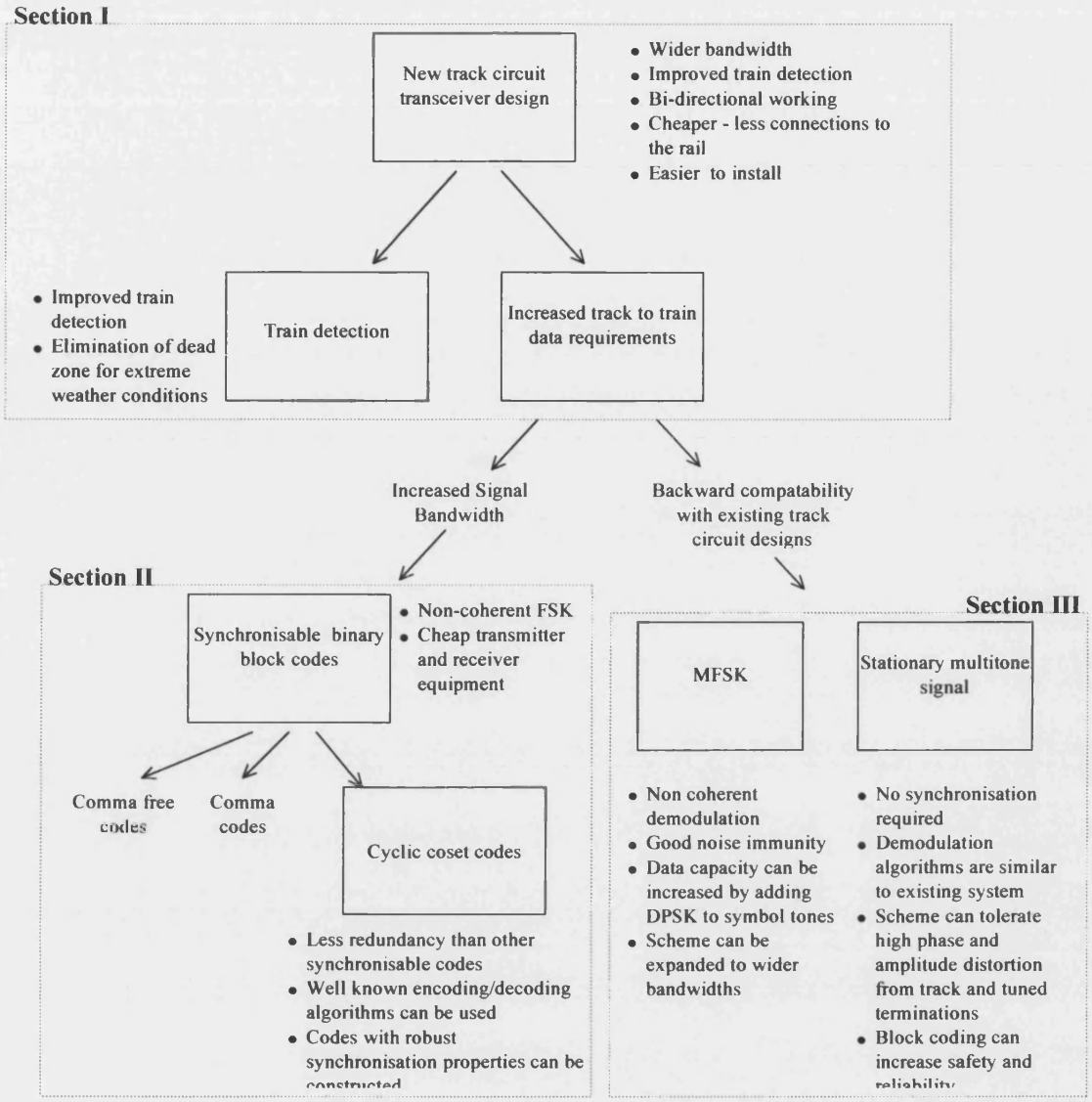


Figure 1.5. Development of Thesis.

circuit performance in terms of train detection is presented in chapter 5. This demonstrates that the design, as well increasing the potential bandwidth and data capacity, offers greatly improved train detection. The simulation uses the track circuit parameters presented in chapter 3, and the track circuit is modelled for all conditions. Chapter 6 gathers information from many sources, on typical power and traction interference which affect the track circuit. From this study, a general impulsive interference model is defined, which is not specific to a particular location or equipment, but will model the expected noise more accurately than the Gaussian model.

Section II includes chapters 7 to 9. It is concerned with reliably increasing the track circuit data capacity with serial binary coding schemes. Chapter 7 describes the binary coding schemes that previously have been applied to track circuits. Presented in chapter 8 is a fully synchronisable binary coding scheme that can be applied to track circuit modulation schemes

with a higher data rate and code size than previous designs. The encoding algorithm is simple and is not restricted to a specific block code size. The synchronisation performance is studied in Gaussian noise and the codes are shown to be extremely robust in terms of safe synchronisation in the presence of errors. However, the analytical study of a coding scheme's performance in Gaussian noise causing independent errors is convenient, but unrealistic, since the interference found in railway environments is typically non-Gaussian. The interference model defined in chapter 6 is used in chapter 9 to study the performance of the proposed synchronisable codes with impulsive interference. Results of a full simulation of the coding scheme with non-coherent FSK demodulation are presented, and compared to those of chapter 8, with Gaussian noise. The coding scheme is proved robust with typical railway interference.

Section III includes chapters 10 to 13. It is concerned with new modulation schemes suitable for a track circuit signal fed through the existing tuned track circuit terminations. This is required to achieve a gradual upgrade path to the termination design described in chapter 4. This may introduce extra phase and amplitude distortion and places an initial restriction on the signal bandwidth (≈200Hz), which makes a serial binary data stream unsuitable. Chapter 10 presents the data, bandwidth and response time requirements for the new scheme and also includes a description and comparison of traditional modulation schemes, with these requirements in mind. When sufficient synchronisation and error control redundancy are taken into account it is shown that it is not possible to meet the desired data capacity within the bandwidth constraints.

Chapters 11 and 12 describe two separate modulation and coding schemes which aim to use the available bandwidth more efficiently and fulfil the data requirements within the limitations described in chapter 10. In chapter 11, binary FSK is extended to *M*-ary FSK with simple non-coherent detection. Algorithms are proposed for simple demodulation and symbol synchronisation using correlation processing and the FFT. To increase the data capacity, differential PSK can be encoded onto the symbol phase.

In chapter 12 a modulation scheme is proposed which consists of parallel tones spaced at regular intervals. The signal requires no carrier, symbol or coding synchronisation and so simple demodulation can be achieved through the FFT. Unlike the MFSK scheme discussed in the previous chapter, the hardware and detection philosophies are similar to those already used for the Westinghouse FS2000 series of track circuits. The signal can be viewed as an extension of the rate coded scheme, where information is encoded onto the successive phases of adjacent tones, rather than the distance of the sidebands to the carrier. Cheap equipment and simple demodulation algorithms can be used. The scheme can tolerate the high phase and amplitude distortion caused by the rail and the tuned track circuit terminations.

Coding schemes for this signal are discussed in chapter 13. The expected error patterns caused by typical railway interference are discussed and appropriate coding schemes are proposed. A simulation of the modulation and coding scheme is described, which uses the impulsive interference model discussed in chapter 6. For the coding scheme simulated there are many different combinations of error detection/correction that can be applied. There are also many ways in which information from the demodulation process can be used in the decoding process to improve both the safety and reliability of the overall system. These methods (including sub-optimal soft decision decoding and minimum distance decoding) are discussed and simulation results are presented. These techniques applied to a simpler, interleaved coding scheme are found to produce superior results than those of the more complicated RS codes. Error correction is discussed, with respect to track circuit safety along with further potential demodulation and decoding techniques.

Each chapter ends with a discussion/conclusions section. Overall conclusions are drawn in chapter 14.

1.3 Summary of Conclusions

1.3.1 Section I

When railway track is used as a transmission line, its attenuation and propagation characteristics are shown to vary greatly with environmental conditions. A mathematical method of modelling the track circuit is presented and, taking these worst case variations into account, the train detection function of the track circuit is studied in chapter 5. It is shown that with the existing track circuit design, there may exist a combination of track parameter conditions where a short circuit across the rails will not be detected. The new track circuit design [19] is studied in a similar manner and it is shown that this 'dead zone' is eliminated, and the overall train detection sensitivity is greatly improved.

1.3.2 Section II

A fully synchronisable binary coding scheme is presented in section II suitable for a track circuit signal modulated by a serial binary data stream. It is shown that the coding scheme offers many advantages over previously proposed schemes. The encoding and decoding algorithms are based on well known cyclic codes and as the same redundancy is used for both error control and synchronisation there is no need for additional synchronisation sequences to be added to the code. The effect of the 'decoding philosophy' is studied as well as the code performance itself. For the coding scheme presented, a decoding philosophy dictating that two consecutive code words must be received before a decision is made provides simple and

effective error detection. This allows a small amount of error correction to be used on each individual code word to optimise the decoding reliability.

A synchronisable code using (31,11) BCH codes is presented, which has excellent synchronisation properties and will greatly increase the data capacity of the track circuit. As the receiver requires the reception of two valid code words, the data capacity could be further increased with a track circuit message split into two code words. It is shown that these codes have lower safe and unsafe failure rates than other synchronisable coding methods.

1.3.3 Section III

A novel form of multi-tone modulation is presented in section III which fulfils the conditions of increased data requirements and backwards compatibility with existing track circuit equipment. Data symbols are separated in frequency rather than time and a high data capacity is achievable within a small bandwidth and detection time. The scheme can tolerate severe rail and track circuit equipment distortion. An interleaved Hamming code is proposed, providing immunity to periodic or narrowband interference. It is shown how soft decision techniques can provide efficient decoding and the greatest control over the safety and reliability trade off.

In section II and III, algebraically efficient burst error control codes are considered as well as the simpler codes. Due to the synchronisation and safety requirement it is generally concluded that the increase in performance with the use of these codes is negligible or non-existent and that the use of codes designed for efficient error correction is inappropriate for track circuit coding. It is also desirable to use less complicated algorithms with lower hardware and development costs.

It is shown in this thesis that the use of error correction, although generally discouraged in railway safety applications, is appropriate for higher data capacity track circuits, providing the overall safety of the scheme is carefully considered. A limited amount of error correction is needed to equip the scheme with the interference tolerance required to meet reliability standards. It is shown in section III that the soft decision error correction algorithm is directly equivalent to the noise tolerance threshold settings of current track circuit equipment.

SECTION I

Chapter 2: Audio Frequency Track Circuit Design

Chapter 3: Railway Track Modelling

Chapter 4: A New Track Circuit Transceiver Termination

Chapter 5: Track Circuit Performance Simulation

Chapter 6: Modelling Traction and Power Interference

The section is concerned with the railway track circuit and environment as, primarily, a train detection circuit. It provides the track circuit background information which is essential when considering the application of coding and modulation techniques. Chapter 2 provides an introduction to track circuit design. The design considerations and trade offs are discussed, with respect to train detection. Chapter 3 discusses the electrical modelling of railway track as a transmission line. Chapter 4 describes a new track circuit design and describes the potential improvements in train detection and data communication. Using the techniques described in chapter 3, the train detection sensitivity of this track circuit design is modelled and compared to existing designs. Chapter 6 discusses typical interference encountered in railway environments and proposes a simple interference model. This model will be used in sections II and III, to evaluate the performance of coding and modulation schemes.

2. Audio Frequency Track Circuit Design.

2.1 Track Circuit Designs.

A track circuit can be described as an electrical circuit which uses the rails as conductors, for the purpose of detecting the presence of a train. They are the main component of fixed block signalling. The first track circuits, early this century, used a d.c. power source at the transmitter end and a relay at the receiver end. The presence of a train axle will short out the current and cause the relay to de-energise. The system is known as 'fail-safe' since any mechanical failure (e.g. broken wires, battery failure) will result in the *track occupied* condition. This is still the principle of all track circuit designs today.

Each d.c. track circuit is isolated by the use of insulated block joints (IBJs) which electrically separates two sections of rail. AC power frequency track circuits were introduced to be compatible with d.c. electrified railways which use the running rails as a return path for the traction current. The diagram below shows a single line vane relay track circuit operating at 50Hz. The a.c signal is phase shifted by 90 degrees and then sent down the track. If the received signal is of the correct phase and amplitude, the track is assumed unoccupied. The bottom rail can be used for the traction return current.

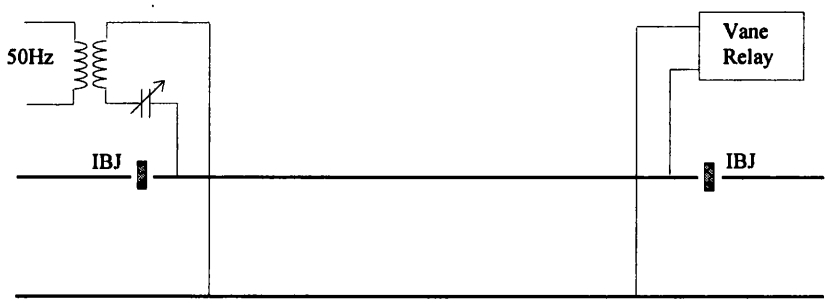


Figure 2.1. Single Line a.c. Track Circuit.

The invention of the *impedance bond* allowed both rails to be used for traction return. An impedance bond consists of a centre tapped inductance connected across the rails which provides a low impedance for the traction current to flow past the IBJs. The impedance of the bond to the signalling frequency is high. Extra bonds are sometimes used for long track circuits to equalise the traction return current in each rail.

Insulated block joints are expensive to maintain and are weak points in the track. They have been replaced by the use of continuously welded rail (CWR), which has made it necessary to move the track circuit frequencies to the audio frequency range. These track circuits are called *jointless track circuits.* The change in track parameters with frequency means that the

attenuation is higher at audio frequencies but the characteristic impedance of the track is high compared to the train shunt impedance and so adequate train detection is still possible. The termination at the receive end of the jointless track circuit is usually a low impedance created by a tuned area, compared to the high impedance of the IBJs. It can be said that the design of jointless track circuits using tuned area termination is more complicated than d.c. or power frequency a.c. track circuits and introduces more design problems and trade offs. It is possible to use a shorting bar to create a low impedance and couple the transmitter to the track somewhere near this shorting bar. This however a very wasteful in power and creates a large *dead zone* where the presence of an axle cannot be detected. Several designs of tuned termination are shown below.

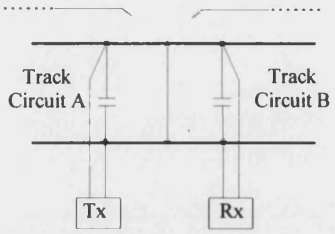


Figure 2.2. Tuned Termination with Simple Shorting Bond

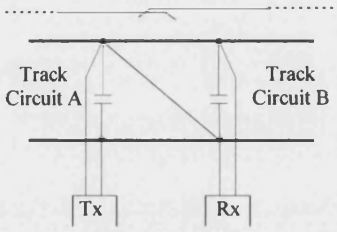


Figure 2.3. Tuned Termination with D bond

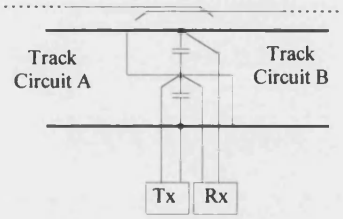


Figure 2.4. Tuned Termination with Z Bond

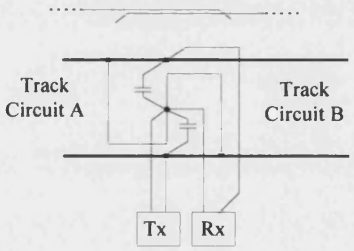


Figure 2.5. Tuned Termination with S bond

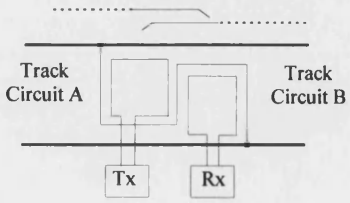


Figure 2.6. Inductive Coupling with S Bond

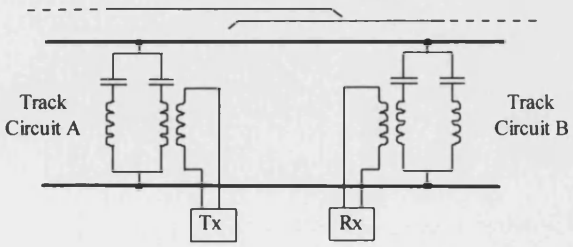


Figure 2.7. Tuned Termination with Series/Parallel Resonant Path

The termination shown in figure 2.2 is used in the Westinghouse FS series of jointless track circuits. A second shorting bar can be added at a distance of 0.75m from the first, as it was found that the mutual coupling between like frequency track circuits could be as high as 15% [20]. The transmitter/receiver is usually placed 6m from the shorting bar. This length, and the value of the capacitor are chosen to resonate with the inductance of the rail at the track circuit frequency. The current is fed to the track via a series resonant circuit. Due to the resistance of the rail and ballast leakage the circuit has a low Q (≈5). This is required so that the circuit will accept both sidebands of the FSK signal (only the carrier and first side bands are used for detection). The high circulating current helps keep the rail to rail voltage constant with changes in track impedance. Also, any equipment failure will have the effect of detuning the circuit and reducing the level at the receiver, resulting in a track circuit safe failure. The receiver is part of a parallel resonant circuit which represents a high termination impedance (≈1Ω) even though the extreme ends are defined by a short circuit. As we will see, this higher termination impedance provides greater sensitivity for train shunt detection. This design has a *dead zone* near the shorting strap, of about 3m where a train shunt will not be detected.

Figures 2.3 and 2.4 are known as a D bond and Z bond respectively. They create overlapping track circuits where a single shunt can be detected by two adjacent track circuits.

Figure 2.5 shows the S bond. This type of termination is used in the Siemens S-Bond [20] and Aster 1 Watt [22] track circuits. This is similar to the previous designs but a higher Q is possible because of inductive coupling of the bond to the rail. This means that the termination is more sensitive to the presence of a train shunt which will disturb the resonant circuit.

Figure 2.6 shows an S bond with inductive coupling to the track. This method gives a reduced variation of load impedance at the transmitter, but is not as efficient as the direct coupling or transformer coupling method.

Figure 2.7 shows the termination used in the ML T121 track circuit. Each tuning unit (transmitter or receiver) contains a series resonant path for one frequency and a parallel resonant path for the other. The series resonant path takes the place of the shorting bar. The tuning of the parallel resonant path takes into account the tuning of the series path and the inductance if the rail.

High Voltage/Impulse Track Circuits.

These track circuits have been designed for use on lightly used lines where contamination and rust can build up, reducing the shunting capability of the circuit. The current-voltage characteristic for the wheel-rail contact under contamination is shown below [21]. An equivalent circuit for the contact is also shown to be two back to back zener diodes.

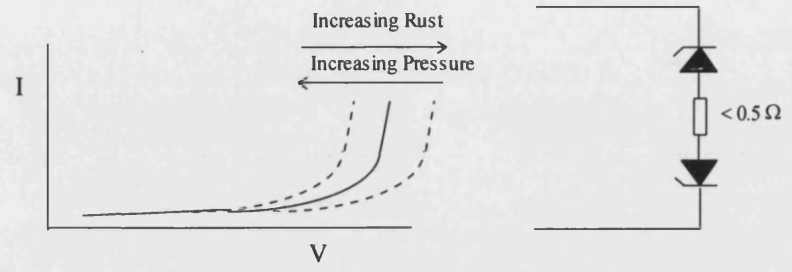


Figure 2.8. Voltage-Current Characteristic of Wheel Rail Contact

For these track circuits, high voltage impulses are transmitted, to break down the contamination layer. It is not possible to use continuous high voltages because of the increased power consumption, and the electrical danger to personnel. The shape of the impulse (peak ≈ 80v) is such that the positive and negative parts have the same energy (i.e. same area). Track circuit lengths can be up to 2km.

Pre-shunt/Extended Shunt Distance.

Instead of a dead zone, it is possible with some termination designs to have a distance before and after the terminations where a train is detected even though it is some distance outside the track circuit. These distances are known as the pre-shunt distance and the extended shunt distance. They are caused by feed through voltage past the termination and the length depends on the track parameters and the track circuit frequency. If the terminations are symmetrical, both distances will be the same. Pre-shunt and extended shunt distance cause an increase in headway and should be kept as small as possible.

2.2 Track Circuit Length

The maximum track circuit length is limited by the train detection function. There must be sufficient sensitivity to the presence a of train shunt, in all weather conditions. The train shunt sensitivity is usually given by the ratio :

$$S = \frac{\text{Received signal amplitude under wet conditions with no train shunt}}{\text{Received signal under dry conditions with } 0.5\Omega \text{ train shunt}}$$

which must always be greater than one. The worst case train shunt resistance is considered to be 0.5Ω in Britain, and a lot less (0.06Ω) in USA, partly due to increased loading. The theoretical maximum length of a track circuit design can be determined as follows: The

track and terminations are modelled and two curves are obtained. One shows the receiver signal level for dry conditions with a 0.5Ω train shunt, the other shows the receiver signal level for wet conditions with no train shunt. They are both plotted against track circuit length. The graph will have the shape shown below:

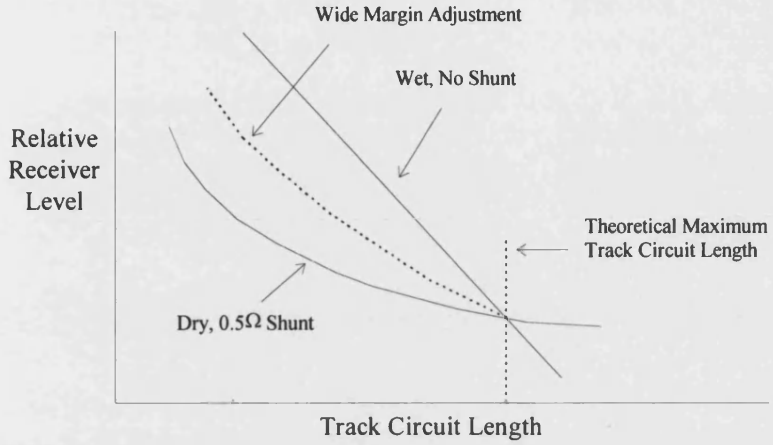


Figure 2.9 Calculation of Maximum Track Circuit Length

Where the two curves meet shows the theoretical maximum track circuit length, above which the presence of a train shunt will not be detected under these conditions. A safety margin should be included which will lower the maximum length. Increasing the transmitter power will not increase the maximum length since both curves will rise simultaneously with no change in the crossover point. The only situation where increasing the transmitter power will increase the maximum length is where the crossover point corresponds to a receiver level too small to detect.

2.3 Broken Rail Detection

The use of IBJs helped to dissipate the stresses caused by rail expansion. With CWR the stresses in the ties, fasteners and rails are greater and there is an increased risk of a broken rail occurring. The track circuit should be able to detect this condition by indicating 'track occupied', but this is complicated by the use of cross bonding. Cross bonding is used to connect a network of railway track to provide protection against potential build ups (e.g. lightning strikes) and to equalise the traction return current. For some track circuits the transmitter is coupled to an impedance bond, which equalises the current in both rails. This impedance bond may be connected to adjacent or parallel track circuits, or structures such as bridges/tunnels etc. which can cause run-around paths, and some of the signal will be present at the receiver even if one of the rails is broken. A simple example is shown in figure 2.10. It must be shown that the signal level at the receiver in this case is below the 'track clear' threshold.

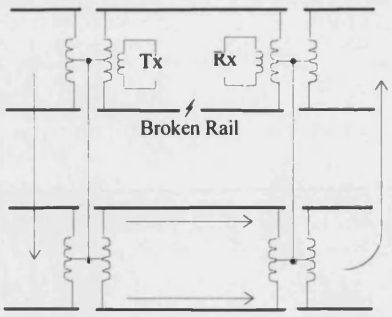


Figure 2.10. Broken Rail with Cross Bonding.

2.4 Receiver Threshold Level

The obvious setting for the receiver threshold is midway between the two curves of figure 2.9 . This is sometimes called 'Wide-Margin Adjustment'. For track circuit terminations with a feed through voltage the wide margin adjustment setting may result in a large pre/extended shunt distance in dry conditions (the operating frequency also has an effect on the pre/extended shunt distance. Higher frequencies are attenuated more and so the distance is smaller). If this is the case, the threshold can be lowered, at the expense of the safety margin for an undetected shunt. The diagram below shows some of the trade offs involved in the choice of the receiver threshold level .

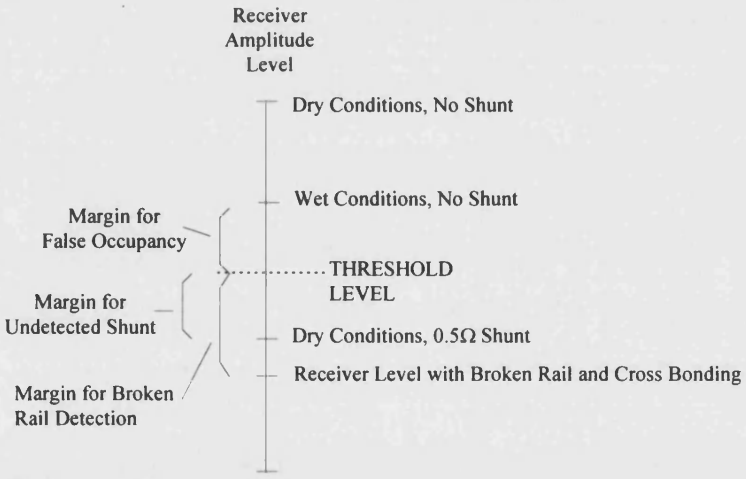


Figure 2.11. Receiver Levels

The receiver level changes with the position of the train shunt. In general, the worst position for detection is near the track circuit terminations. Assuming the terminations and track are symmetrical, the best place for train detection is in the centre of the track circuit. If the termination has a dead zone, it should be designed so that the shortest length train will be detected on either side of the termination.

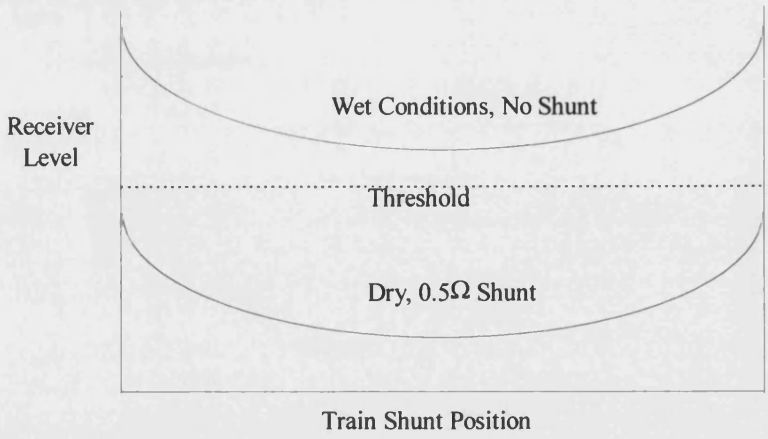


Figure 2.12. Receiver Levels with Moving Train Shunt

As can be seen, the margins described in figure 2.11 and the sensitivity ratio *S*, will change as the train shunt moves through the track circuit.

2.5 Termination Impedance

For long track circuits the termination impedance should ideally be matched to the characteristic impedance of the track. This would result in the maximum power being dissipated in the receiver, and the maximum sensitivity to train shunts. This also maximises the rail to rail voltage. It can be generally said that for short track circuits a higher termination impedance, in relation to the impedance of the train shunt, will improve the train shunt sensitivity. Depending on the termination design, the impedance could affect the feed through voltage, and so a lower impedance will reduce the pre/extended shunt distance.

Figure 2.13 shows the characteristic impedance (magnitude) of the track for extreme wet and dry track conditions (wet = *R* typical, *L* typical, *C* high, *G* high, dry = *R* typical, *L* typical, *C* low, *G* low) It can be seen that there is large variation in the characteristic impedance with environmental conditions. This suggests that the termination impedance could be chosen for reasons other than maximum power transfer. For the termination shown in figure 2.2, the impedance is usually ≈1Ω. This is produced by the parallel resonance of the track inductance and the capacitor. It is chosen to give a reasonable tuned area length (≈6m) and capacitance value (≈50-100μF). Another consideration is the signal bandwidth. For a rate coded signal which consists of a carrier and two sidebands the termination impedance must be fairly constant for the whole bandwidth. This limits the selectivity of a tuned area (Q factor) and hence the overall impedance.

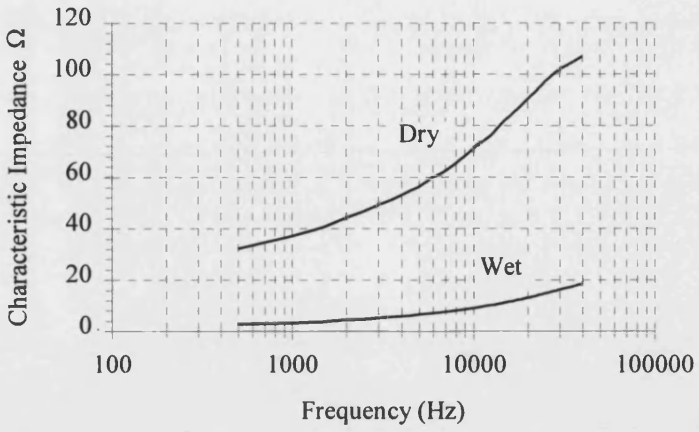


Figure 2.13: Characteristic Impedance of Track with Frequency

2.6 Choice of Operating Frequency

A higher frequency will keep the length and capacitance of the tuned area small, which is desirable. It does, however, decrease the maximum track circuit length since the attenuation is higher, and increase the transmitter power required. A higher operating frequency will provide greater immunity to traction control interference and harmonics which are stronger at lower frequencies. In general, audio frequency track circuits operate in the range of 500Hz - 2kHz, although they can operate up to 20kHz. With termination designs that allow a certain amount of feed through signal, or where cross bonding is present, there is a risk of wrong side failure where the signal leaks to the nearest like frequency track circuit. As higher frequencies are attenuated more, there is less risk of wrong side failure in this way.

2.7 Summary

In has be seen that track circuit design involves many trade offs and compromises. In many previous track circuit designs the parameters have been optimised for simplicity and cost and well as train shunt detection and data transfer. In fact, data transfer is a secondary function of the track circuit and in many cases has been added to the existing design. It has been shown that for traditional jointless track circuit designs, the train shunt sensitivity is lowest at the track circuit extremes. The table below lists some of the considerations for track circuit design. The positive and negative points are marked with ticks and crosses respectively.

3. Railway Track Modelling

3.1 Introduction

When modelling a track circuit, we are interested in the voltage and current levels at the receiver and transmitter, and the distributions along the track. Standard transmission line theory is normally used for communication lines, where we are only interested in the voltage and current at the send and receive ends. However, the theory is still applicable to track circuits and was first used by D. C. Gall in 1933 [25].

Transmission line theory must be used with caution when applied to track circuits, for several reasons.

1. The equations describe a two wire model, with assumed ideal insulation and no leakage to earth. In railway systems, earth bonding is used to provide a return path for lightning strikes or voltage build ups, as shown below

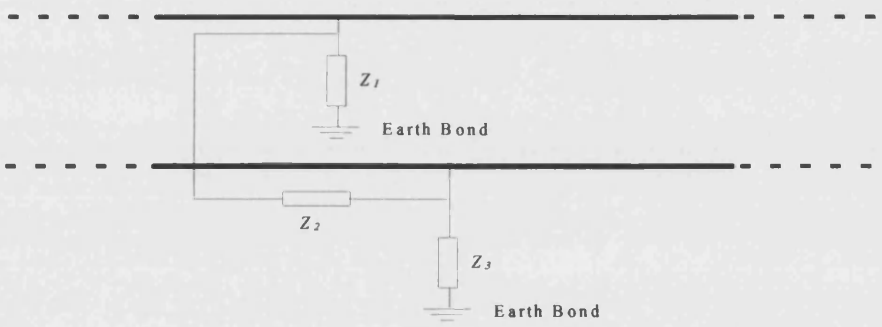


Figure 3.1. Earth Bonding.

The bonding could be to structures such as bridges or tunnels. It can be seen that the conduction across the ballast is increased by Z_2, which could be uniformly distributed or concentrated (lumped) at one point. Also, adjacent track circuits can be bonded via impedance bonds to provide a low impedance return path for the traction current (p113, [26]) which can cause 'run-around' paths.

2. The presence of overhead lines, power rails and parallel track circuits can cause electromagnetic coupling. Multi-wire models are needed to simulate these effects [27][28].

3. The track parameters are frequency dependent and so different frequencies will have different attenuation and propagation characteristics. This must be taken into account when a wide band signal is being considered.

4. The transmission line equations assume that the track parameters are uniformly distributed throughout the length of the line. This is not always the case with railway track.

3.2 The lumped Element Model.

The transmission line parameters are :

Series Resistance (R) Ω/metre

Series Inductance (L) H/metre

Shunt (parallel) Conductance (G) S/metre

Shunt (parallel) Capacitance (C) F/m

It is possible to create a 'lumped element' model of a length of track by multiplying the track parameters by the length of track being considered (figure 3.2).

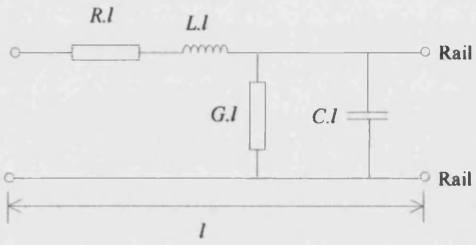


Figure 3.2. Lumped Element Section

There is a limit to the length of track that can be modelled in this way since the lumped element model does not take into account the fact that all the parameters are distributed within the materials that make up the railway track, and not 'lumped' into discrete elements. If the length is too high, the model acts as a resonant circuit, with a low pass filter characteristic. The choice of signal frequency affects the length of section that can be accurately modelled as a lumped element. For d.c. or power frequency signals, the smaller track circuits could be modelled as a single lumped element [25], but for audio frequency signals the length is usually in the range 20-50m, or approx. $\lambda/3$, where λ is the wavelength of the signal. For long track circuits there will be a high number of elements and circuit components, resulting in a high computer simulation time.

The lumped element approximation to the transmission line equations has several advantages:

1. Each section can contain an impedance to earth, and so uniformly or non-uniformly distributed earth leakage can be simulated in a three wire lumped element section as shown in figure 3.3.

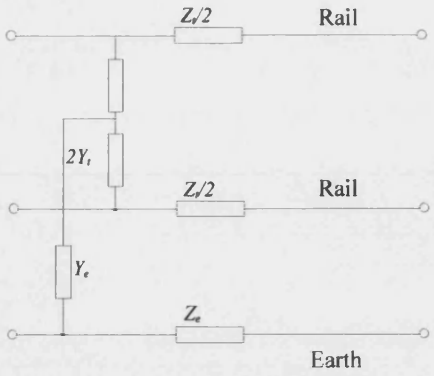


Figure 3.3. 3-wire Lumped Element Section

The elements can be cascaded and source and load terminations added to each end. A moving train shunt can be simulated by placing a shunt resistance between the rails at different points in the cascaded sections.

2. Each section can incorporate variations in the rail parameters, to simulate non-even distribution.

3. The effects of third rail and overhead cables can be simulated [28].

It is, however, hard to simulate these effects due to the lack of information on their magnitudes and variances. For general track circuit operation, for a non-specific site, the two wire model is considered satisfactory for initial modelling of voltage/current distributions, shunt performance, and broken rail detection [25][30][31].

3.3 Transmission Line Theory.

For sinusoidal signals it is convenient to think of voltage and current represented by the complex exponential, $e^{j\omega t}$. The instantaneous voltage or current can be found by taking the real part of the complex term. Transmission line theory is presented in Appendix A, and the following equation is derived for the voltage at position x along a transmission line:

$$v(x,t) = v_1 e^{j(\omega t+\beta x)} e^{\alpha x} + v_2 e^{j(\omega t-\beta x)} e^{-\alpha x} \qquad (3.1)$$

where α is the attenuation constant (nepers/m) and describes the losses in the line, β is the phase constant (radians/m) and $\gamma = \alpha + j\beta = \sqrt{(R + j\omega L)(G + j\omega C)}$ [29]. Equation 3.1 represents two travelling waves of voltage. The first term represents a wave travelling in the negative x direction, the second, in the positive x direction.

3.3.1 Reflections and Standing Waves.

Two waves travelling in opposite directions can occur when the source voltage wave is reflected back due to a mismatch in the termination impedance and the characteristic

impedance. The characteristic impedance is the ratio of voltage to current of a wave travelling along a semi-infinite length of line. It is defined by :

$$Z_o = \sqrt{\frac{(R + jwC)}{(G + jwC)}} \tag{3.2}$$

When a transmission line of characteristic impedance Z_o is terminated by a load impedance $Z_L \neq Z_o$ some of the incident power will be reflected.

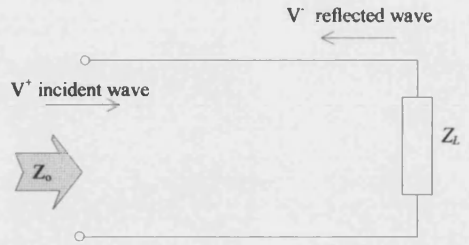


Figure 3.4. Mismatched Termination Impedance.

The incident wave, V^+ will be attenuated and delayed by the line as it propagates forward. When it reaches the load impedance, Z_L, the proportion of voltage reflected is given by:-

$$\rho_v = \text{voltage reflection ratio} = \frac{\frac{Z_L}{Z_o} - 1}{\frac{Z_L}{Z_o} + 1} \tag{3.3}$$

The proportion of current reflected is given by:

$$\rho_i = \text{current reflection ratio} = \frac{1 - \frac{Z_L}{Z_o}}{1 + \frac{Z_L}{Z_o}} \tag{3.4}$$

For a transmission line of length greater than $\approx \lambda/3$, where λ= wavelength = $2\pi/\beta$, the incident wave combines with the reflected wave. At some points along the line, the waves are in phase and the resultant amplitude will be high. At other points the waves will be out of phase and the resultant amplitude will be low. Standing waves will occur and the amplitude of the sinusoid will have peaks and troughs, separated by a distance of $\lambda/2$ metres.

For long track circuits at high frequencies it must be shown that the standing wave does not produce troughs that are lower than the train shunt ionisation voltage required. Figure 3.5 shows the rms voltage distribution along a 1.5km line, with typical rail parameters and open circuit termination. The signal frequency is 10kHz.

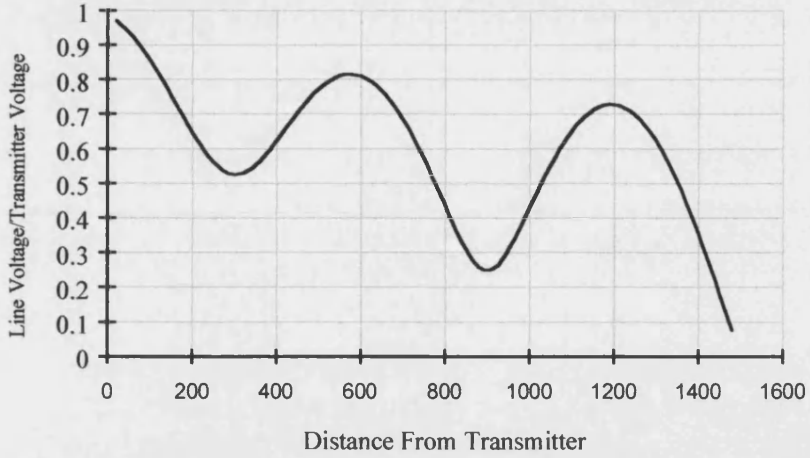


Figure 3.5. Voltage Standing Waves.

When the source current and voltage are known, the values at distance x from the source can be calculated:

$$V(x) = V_s \cosh \gamma x - I_s Z_o \sinh \gamma x$$
$$I(x) = I_s \cosh \gamma x - \frac{V_s}{Z_o} \sinh \gamma x \qquad (3.5)$$

3.3.2 Effective Line Impedance.

The effective impedance seen from the input to a line of length l and characteristic impedance Z_o terminated by impedance Z_L is given by:

$$Z = \frac{Z_L + Z_o \tanh \gamma l}{1 + \frac{Z_L}{Z_o} \tanh \gamma l} \qquad (3.6)$$

For the open circuit line, Z_L=∞ and equation 3.6 becomes:

$$Z_{oc} = \frac{Z_o}{\tanh \gamma l} \qquad (3.7)$$

For the short circuit line, Z_L = 0 and equation 3.6 becomes:

$$Z_{sc} = Z_o \tanh \gamma l \qquad (3.8)$$

Combining these equations:

$$Z_o = \sqrt{Z_{oc} Z_{sc}} \qquad (3.9)$$

and

$$\tan \gamma l = \sqrt{\frac{Z_{sc}}{Z_{oc}}} \qquad (3.10)$$

3.3.3 Line Distortion.

The phase velocity describes the speed at which a point of constant phase propagates along the line. It is defined by:

$$v_p = \frac{\omega}{\beta} \text{ m/s} \tag{3.11}$$

If the signal is thought of as a series of discrete sinusoids of different frequencies, then all of these frequencies must propagate with the same velocity for no phase distortion to occur. Since $v_p = \omega/\beta$, this implies that the phase constant must be proportional to frequency. The attenuation constant, α, must be a constant value over the signal bandwidth for no amplitude distortion to occur.

For band-pass signals the group velocity is often a more important parameter than the individual phase velocity of each sinusoid. This describes the effective velocity of the combined signal or envelope. The group velocity, v_g is given by the slope of ω with β:

$$v_g = \frac{\partial\omega}{\partial\beta} \text{ m/s} \tag{3.12}$$

Note that ideally the group velocity needs to be constant over the signal bandwidth. Also note that when the phase velocity is constant for all frequencies ($\delta v_p/\delta\omega = 0$) then the group and phase velocities are the same.

3.4 Exact Equivalent T Model.

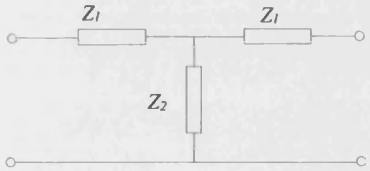


Figure 3.6. Equivalent T Model

It is possible to form an exact equivalent of the transmission line model, for a specific length, in the form of a T model, as shown in figure 3.6. Since the transmission line model is symmetrical, both series elements (Z_1) are the same. Equation 3.7 and 3.8 show the input impedance of a transmission line with short circuit and open circuit terminations. These can be equated to the T model to find expressions for Z_1 and Z_2. For the T model, the open circuit input impedance is:

$$Z_{oc} = Z_1 + Z_2 \tag{3.13}$$

Equating this to equation 3.7:

$$\begin{aligned} Z_1 + Z_2 &= \frac{Z_o}{\tanh \gamma l} \\ Z_2 &= \frac{Z_o}{\tanh \gamma l} - Z_2 \end{aligned} \tag{3.14}$$

The short circuit input impedance is:

$$Z_{sc} = Z_1 + \frac{Z_1 . Z_2}{Z_1 + Z_2} \tag{3.15}$$

Equating this to equation 3.8:

$$Z_1 + \frac{Z_1 . Z_2}{Z_1 + Z_2} = Z_o \tanh \gamma l \tag{3.16}$$

Substituting equation 3.14 into equation 3.16:

$$\frac{Z_o}{\tanh \gamma l} - \frac{Z_2^{\,2}}{Z_o} \tanh \gamma l = Z_o \tanh \gamma l \tag{3.17}$$

which leads to:

$$\begin{aligned} Z_2 &= Z_o \sqrt{\frac{1}{\tanh \gamma l} - 1} \\ &= Z_o \sqrt{\frac{\cosh^2 \gamma l}{\sinh^2 \gamma l} - 1} \\ &= \frac{Z_o}{\sinh \gamma l} \end{aligned} \tag{3.18}$$

since $\cosh^2 \gamma l - \sinh^2 \gamma l = 1$. This expression for Z_2 can be substituted into equation 3.14:

$$\begin{aligned} Z_1 &= \frac{Z_o}{\tanh \gamma l} - \frac{Z_o}{\sinh \gamma l} \\ &= Z_o \left[\frac{1}{\tanh \gamma l} - \frac{1}{\sinh \gamma l} \right] \\ &= Z_o \left[\frac{\cosh \gamma l - 1}{\sinh \gamma l} \right] \\ &= Z_o \tanh\left(\frac{\gamma l}{2} \right) \end{aligned} \tag{3.19}$$

since $\tanh(x) = \dfrac{\cosh(2x) - 1}{\sinh(2x)}$. So, equations 3.18 and 3.19 give us expressions for the T model parameters in terms of the line parameters R, L, G and C. The characteristic impedance can be expressed in terms of the T model values:

$$Z_o = \sqrt{2 Z_1 Z_2 + Z_1^{\,2}} \tag{3.20}$$

The exact equivalent Π model is shown in figure 3.7:

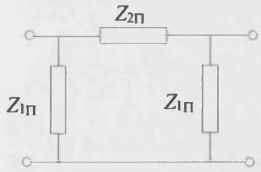


Figure 3.7.
Equivalent Π Model

The impedances for the Π parameters in terms of the impedances for the T parameters are shown below:

$$Z_{1\Pi} = \frac{Z_{1T}^{\,2} + 2(Z_{2T} Z_{1T})}{Z_{1T}} \tag{3.21}$$

$$Z_{2\Pi} = \frac{Z_{1T}^{\,2} + 2(Z_{2T} Z_{1T})}{Z_{2T}} \tag{3.22}$$

3.5 Transmission Line Parameters of Railway Track.

From the last few decades there are many sources of data concerning the transmission line parameters of railway track. This section will gather this information in order to obtain the extreme values of *R, L, G* and *C*. Railway track is not designed to be an data efficient transmission medium and so, for track circuit design, it is essential that the safety and reliability are proved under the most extreme conditions found.

3.5.1 Measurement of Track Parameters.

The standard method of measuring the railway track parameters is by the use of short and open circuit tests [30].

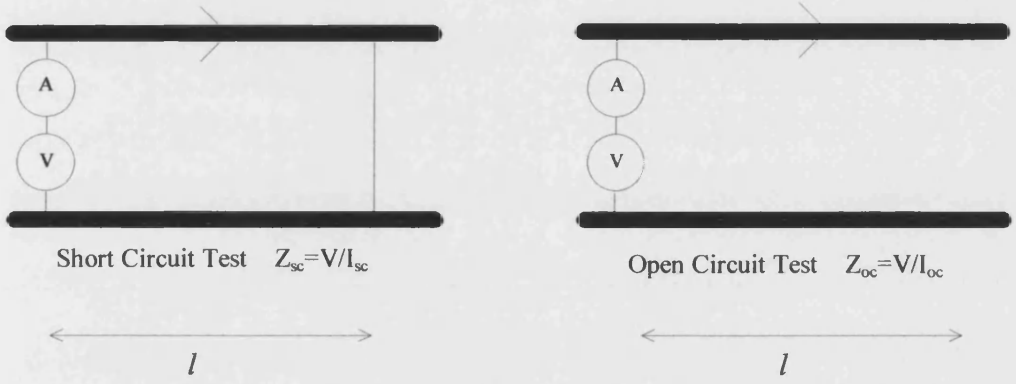


Figure 3.8. Short and Open Circuit Tests.

The characteristic impedance and propagation constant can be calculated as:

$$Z_o = \sqrt{Z_{sc} Z_{oc}} \qquad \tanh \gamma l = \sqrt{\frac{Z_{sc}}{Z_{oc}}} \tag{3.23}$$

The parameters *R, L, G* and *C* can be found by equating the real and imaginary parts of:

$$Z_o = \sqrt{\frac{(R + j\omega L)}{(G + j\omega C)}} \qquad \gamma = \sqrt{(R + j\omega L)(G + j\omega C)} \tag{3.24}$$

With continuously welded rail (CWR) open circuit test cannot be performed and so a method has been devised that will determine the rail parameters from short circuit tests only. Such a test is described by MacEntee [30] where the transfer impedance is measured for two lengths of short circuited rail, but with the constraint of constant short circuit current for both measurements.

Series Rail Resistance

The atmospheric conditions have little effect on the rail resistance. The value will vary with shape, material and cross sectional area of the rail. For example, flat bottomed rail has a smaller resistance than bull-head rail. The resistance of the rail increases with frequency. This phenomenon is known as the 'skin effect'.

Series Rail Inductance

The internal inductance of any pair of conductors, including rails, is a function of the individual inductance of a rail and the mutual inductance between the rails. As the frequency increases, the skin effect reduces the effective rail radius and hence the contribution to the total rail inductance made by the individual inductance, will go down. This explains a slight decrease in the series inductance with increasing frequency.

Shunt Rail Conductance

Inter-rail conductivity is directly affected by environmental conditions. It is increased in salt water areas and with higher frequencies. For flat bottomed rail, the conductivity is reduced by the use of insulating pads between the rails and the ballast.

Shunt Rail Capacitance

The shunt capacitance is mainly determined by the ballast material which, with the separating air, acts as a dielectric. The ballast material can vary from dry granite stones, to wet soil, sand or cinders. The relative permeability of the ballast material is higher than the relative permeability of the air and the sleeper (wood or concrete) and decreases with frequency.

3.5.2 Track Parameter Sources.

B. Mellitt : 'Data Transmission Characteristics of Railway Track' [31]

The results quoted are used in [30]. The tests were conducted on bull head rail sidings in London using conventional open/short circuit tests and double short circuit tests.

Freq. (kHz)	R (mΩ/m)	L(μH/m)	C(μF/m)wet	C(μF/m)dry	G(mS/m)wet	G(mS/m)dry
0.1	0.43	2.1	0.28	0.009	0.67	0.15
0.2			0.22	0.007		
0.3			0.19	0.0056		
1	1.4	1.7	0.09	0.003	0.69	0.193
2	2.3	1.6	0.053	0.0021	0.72	0.21
3	2.7	1.57	0.034	0.0016	0.728	0.219
4	3.2	1.56	0.0235	0.00132	0.739	0.227
5	3.55	1.56	0.0175	0.00115	0.75	0.232
6	4	1.55	0.0134	0.001	0.76	0.24
7	4.34	1.54	0.0106	0.00087	0.77	0.245
8	4.6	1.53	0.0088	0.00079	0.78	0.248
9	5	1.52	0.0073	0.0007	0.785	0.252
10	5.2	1.5	0.0057	0.00062	0.79	0.255
12			0.0036	0.00052		
15	6.4	1.44	0.0018	0.0004	0.82	0.27
20	7.3	1.4	0.004	0.00029	0.85	0.28
40	10	1.37		0.00013	0.92	0.31
70	13	1.35		0.0006	1	0.34

Table 3.1. Track Transmission Line Parameters from [31]

Frielinghaus, K. H. : 'Contingencies in the Design of the Audio Track-Circuit' [23]

Freq.(kHz)	R(mΩ/m)	L(μH/m)
0.1		1.57
0.3		1.5
0.5	0.95	1.42
1		1.34
3	1.97	1.22
5	1.2	1.2
10	1.15	1.15

Table 3.2. Track Transmission Line Parameters from [23]

Iancu, O.D.: 'Computer Aided Design of Non Insulated Track Circuits' [32] , and 'The Influence of Track Circuit Parameter Variation on Maximum Permissible Length' [33]

These papers quote mean values of R and L for type 49 rails, at 2kHz :

R = 3.122 mΩ/m L = 1.5 mH/m G=1mS/m(max)

Westinghouse Signals Ltd Document [34]

This document quotes worst case conditions (highest losses) for 113A flat bottomed rail, at 5kHz.

R=2mΩ/m L=1μH/m G=0.5mS/m C=1nF/m

Westinghouse Signals Ltd Document [35]

This document quotes typical values of R, L and G, for 113A flat bottomed rail at 5kHz,

R=6.6mΩ/m L=1.25μ/m G=0.1mS/m

Fisher, A. N. : 'Track, Track Circuits and Traction' [36]

This paper quotes parameters for standard gauge bull headed rail, laid on wooden sleepers, at 2kHz.

R=1.1mΩ/m L=0.7μH/m G=0.005mS/m(dry) G=0.5mS/m(wet) C=2nF/m(wet)

Hill, R. J. and Weedon, D. N. : 'Computer Aided Design of Audio Frequency Track Circuits' [27].

The following track parameter information is also found in references [10] and [37].

Series Resistance and Inductance:

Freq.(kHz)	R(mΩ/m) high	R(mΩ/m) low	L(μH/m) high	L(μH/m) low
0.1	3.1	0.3	2	1.4
1	8	0.9	1.7	1.2
2	11.04	1.4		
3	13.4	1.85	1.6	1.15
4	15.8	2.23		
5	17.6	2.6	1.56	1.12
10	26.5	4.1	1.53	1.11
20	39	7	1.5	1.11
40			1.49	1.11
70			1.48	1.11
100	90	21		

Table 3.3 Track Transmission Line Parameters [10] and [37].

Shunt Conductance and Capacitance.

Freq.(kHz)	G(mS/m)high	G(mS/m)low	C(μF/m)high	C(μF/m)low
0.1	0.53	0.0034	0.3	0.0017
0.2			0.25	0.0015
0.3				0.00133
0.5			0.165	
1	0.61	0.005	0.1	0.00085
2	0.65	0.006	0.055	0.0006
3	0.67	0.0068	0.036	0.00047
4	0.69	0.0076	0.0276	0.0004
5	0.7	0.0083		
6			0.0175	0.0003
8			0.013	
10	0.73	0.011	0.01	0.0002
15			0.0058	
20	0.8	0.016	0.0036	0.0001
40			0.0009	0.00004
100	0.92		0.0001	

Table 3.4 Track Transmission Line Parameters from [10] and [37].

All these rail parameter sources are plotted in figures 3.9, 3.10, 3.11 and 3.12.

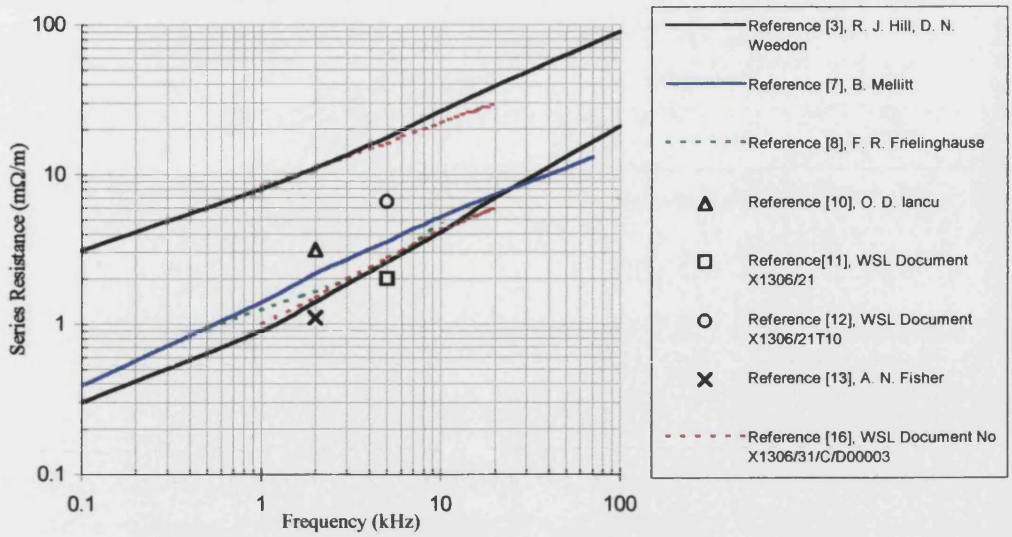


Figure 3.9. Series Resistance

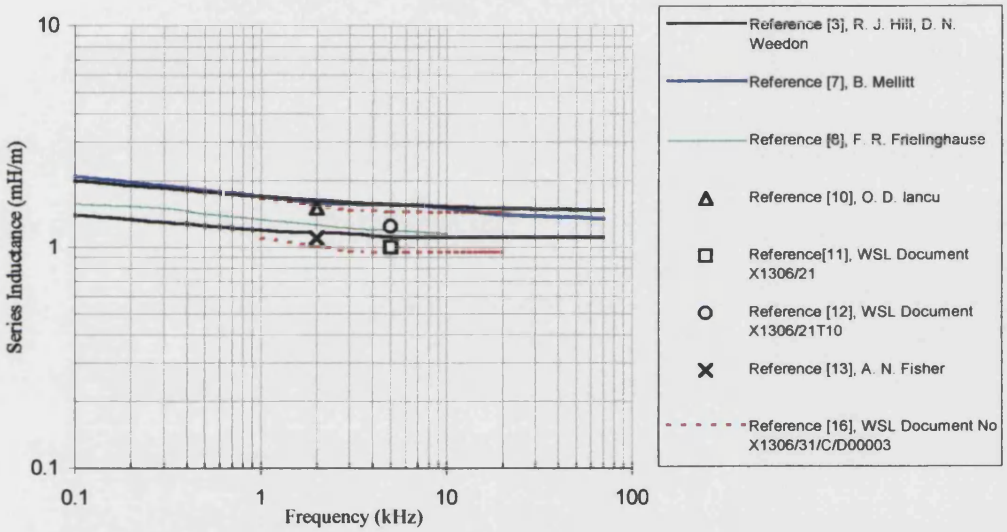


Figure 3.10. Series Inductance

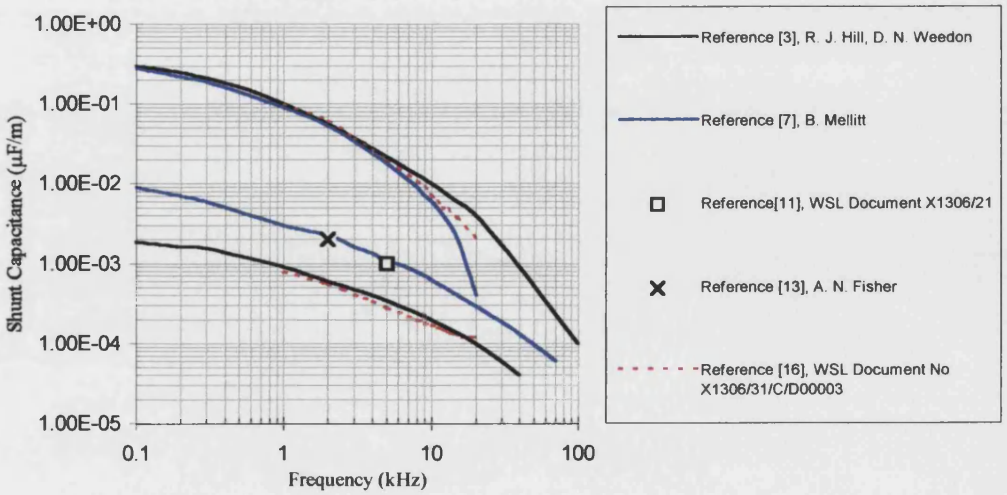


Figure 3.11. Shunt Capacitance

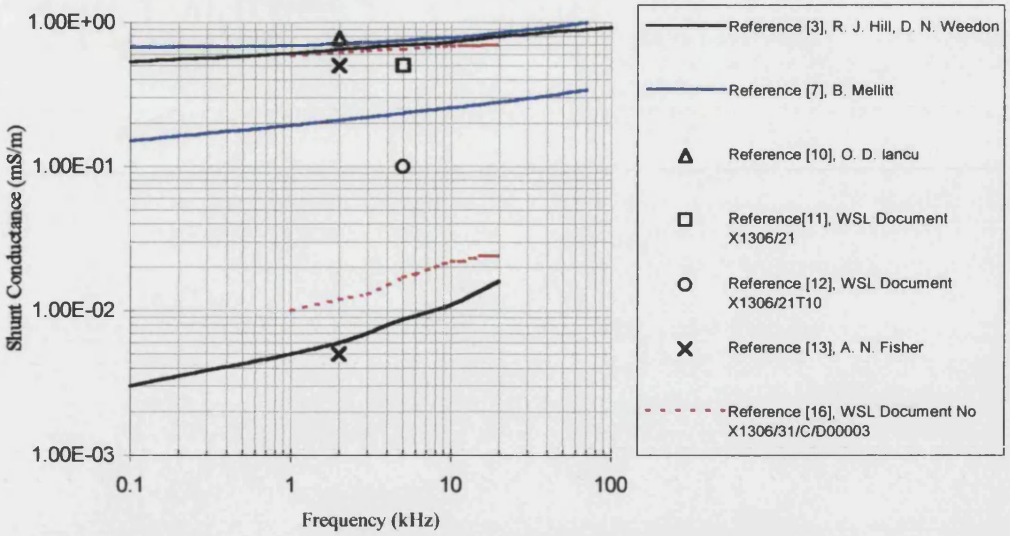


Figure 3.12. Shunt Conductance

The extreme values of all the track parameters (shown below) will be used for track circuit modelling simulations in the next chapter.

R (high/low) - Reference [27]

L (high/low) - Reference [27]

C (high/low) - Reference [27]

G (high) - Reference [31]

G (low) - Reference [27]

3.6 Transmission Line Properties of Railway Track.

We can use railway track transmission line parameters to model the track circuit as a communications channel. Shown below are graphs showing the characteristic impedance of the track, for various combinations of track parameters. Typical values are taken to be midway between the two extreme values.

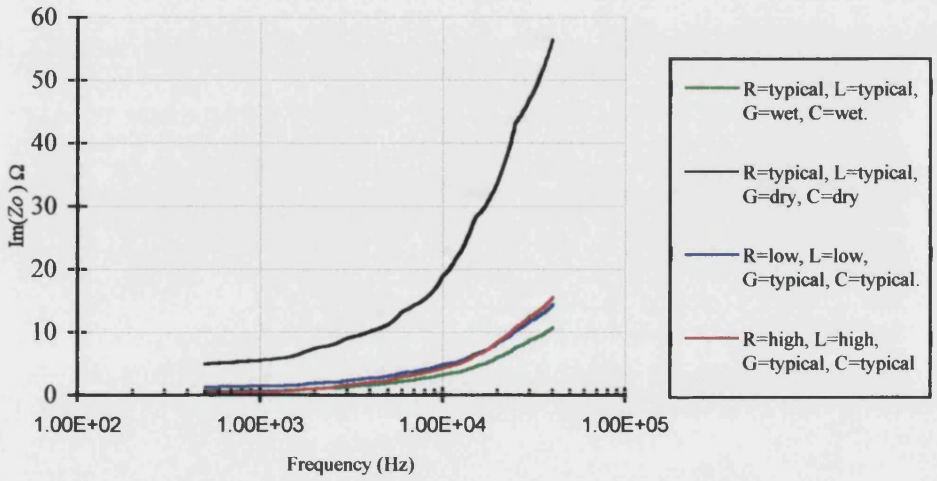


Figure 3.13. Imaginary Component of Railway Track Characteristic Impedance

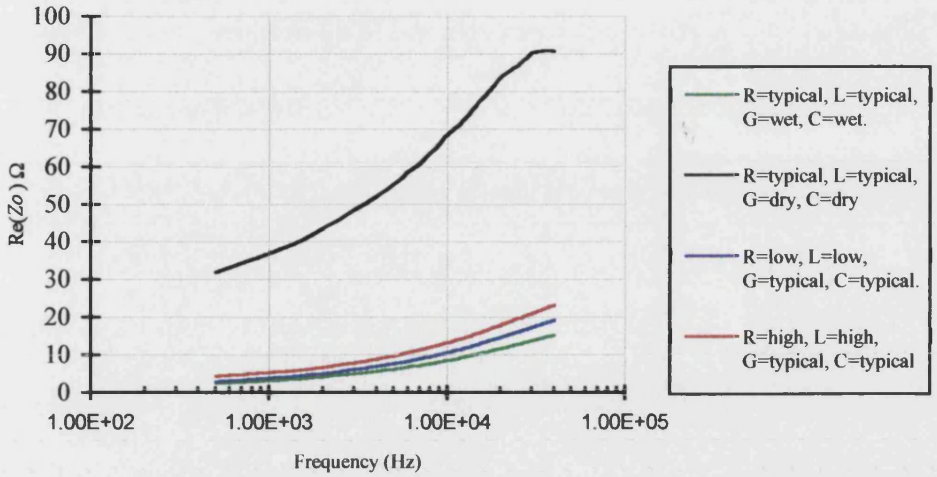


Figure 3.14. Real Component of Railway Track Characteristic Impedance

It can be seen that the variation of the series components has little overall effect, compared to the shunt components. The graphs below show the attenuation constant and phase constant, for extreme values of G and C.

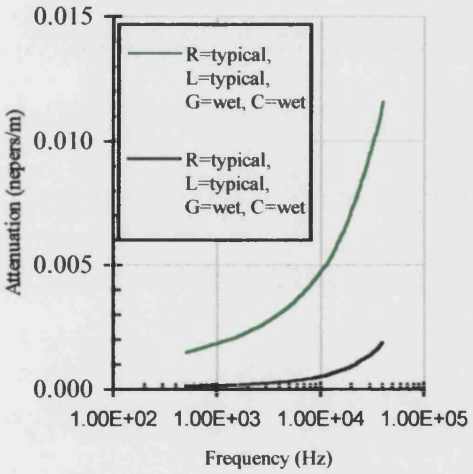


Figure 3.15. Attenuation Constant.

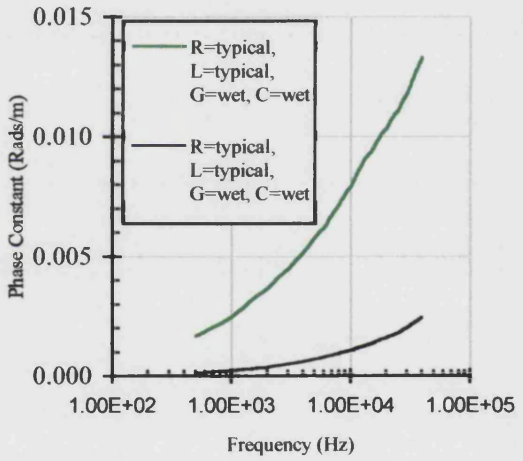


Figure 3.16. Phase Constant.

This demonstrates that both signal attenuation and phase distortion increase with frequency. Using equations 3.11 and 3.12, the phase and group velocities can be found :

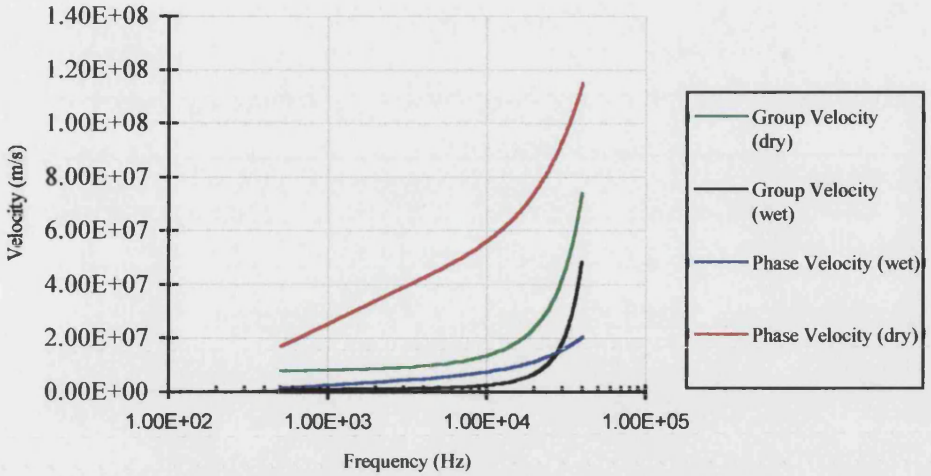


Figure 3.17. Phase Velocity and Group Velocity

It can be seen that for frequencies up to approximately 5kHz, the phase velocity is roughly proportional to frequency and so the group velocity will be constant. As the frequency increases the 'channel' becomes more dispersive.

The wavelengths for extreme wet and dry conditions are shown below.

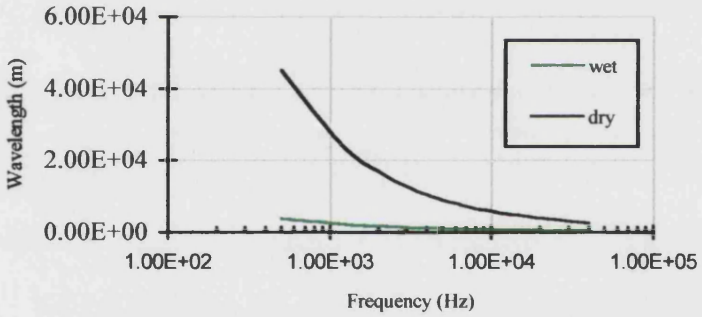


Figure 3.18. Signal Wavelength for Railway Track

For long track circuits, especially at higher frequencies, it can be seen that the wavelength can approach the track circuit length and transmission line effects such as standing waves can occur.

3.7 Track Circuit Transfer Function.

It may be of interest to develop a track circuit transfer function which corresponds to the voltages and currents at the transmitter and receiver, with no train shunt present. This model could then be used, for instance, to predict the worst case quiescent amplitude and phase distortion that may occur.

It has been shown [24][39] that the dynamic range at the receiver input (as it varies with environmental conditions) is lowest when a voltage source is used as the transmitter and a current detector is used as the receiver. We will assume that continuous rail is used and the

track circuit boundaries are defined by two connections to the rail. A track circuit design of this type is described in the following chapter. The transmitter is represented by a constant voltage supply, with a series impedance, *Zs*. The receiver is represented by an impedance, *Zr*, across the rails. The received signal will be proportional to the current through this impedance. The transmission line model will be used to represent the track, as shown below. We will assume that the characteristic impedance is seen from either end of the track circuit.

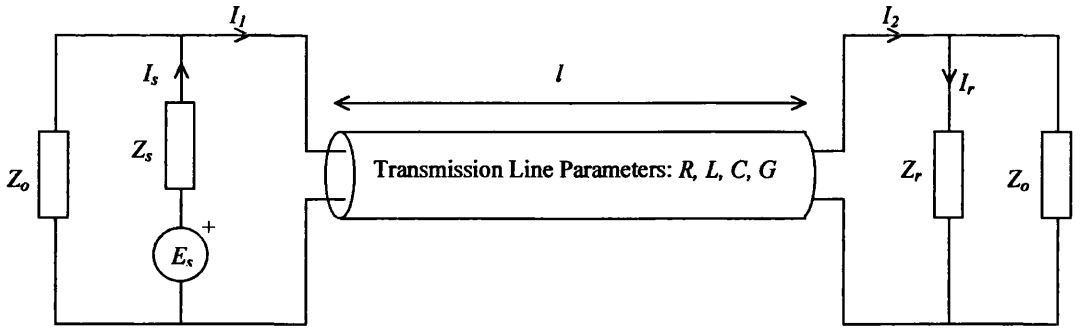


Figure 3.19. Track Circuit Voltages and Currents.

We are interested in finding an expression for the *transfer admittance*, $Y_t = I_r/E_s$. If the applied signal is sinusoidal, the following equations apply:

$$E_s = V_1 + Z_s(I_s) = V_1 + Z_s\left(I_1 + \frac{V_1}{Z_o}\right) \tag{3.25}$$

$$V_1 = V_2\cosh(\gamma l) + I_2 Z_o \sinh(\gamma l) \tag{3.26}$$

$$I_1 = I_2\cosh(\gamma l) + \frac{V_1}{Z_o}\sinh(\gamma l) \tag{3.27}$$

$$I_2 = I_r + \frac{V_2}{Z_o} \tag{3.28}$$

Substituting equation 3.26 into equation 3.25:

$$E_s = \left(V_2 + \frac{V_2 Z_s}{Z_o}\right)\cosh(\gamma l) + \left(Z_o I_2 + Z_s I_2\right)\sinh(\gamma l) + Z_s I_1 \tag{3.29}$$

Substituting equation 3.26 into equation 3.27:

$$I_1 = I_2\cosh(\gamma l) + \frac{V_2}{Z_o}\cosh(\gamma l)\sinh(\gamma l) + I_2\sinh^2(\gamma l) \tag{3.30}$$

Substituting equation 3.30 into equation 3.29

$$\begin{aligned} E_s = &\left(V_2 + Z_s I_2 + \frac{V_2 Z_s}{Z_o}\right)\cosh(\gamma l) + \left(Z_o I_2 + Z_s I_2\right)\sinh(\gamma l) \\ &+ \frac{V_2 Z_s}{Z_o}\cosh(\gamma l)\sinh(\gamma l) + Z_s I_2\sinh^2(\gamma l) \end{aligned} \tag{3.31}$$

Substituting equation 3.28into equation 3.31 and using $V_2 = I_r Z_r$:

$$\frac{I_r}{E_s} = \begin{bmatrix} Z_s\left(1+\frac{Z_r}{Z_o}\right)\left(\cosh(\gamma l)+\sinh(\gamma l)+\sinh^2(\gamma l)\right) \\ +\frac{Z_r Z_s}{Z_o}\left(\cosh(\gamma l)+\cosh(\gamma l)\sinh(\gamma l)\right) \\ +Z_r\cosh(\gamma l)+Z_o\left(1+\frac{Z_r}{Z_o}\right)\sinh(\gamma l) \end{bmatrix}^{-1} \qquad (3.32)$$

Using the relationships: $\sinh(x)=\frac{1}{2}(e^x - e^{-x})$, and $\cosh(x)=\frac{1}{2}(e^x + e^{-x})$, equation 3.32 becomes:

$$Y_t = \frac{I_r}{E_s} = \left[Z_o \sinh(\gamma l) + \left(Z_s + Z_r + \frac{2Z_s Z_r}{Z_o} \right) e^{(\gamma l)} \right]^{-1} \qquad (3.33)$$

The plots below show the transfer function for a 1000m length track circuit, with termination impedances of 10mΩ. The track parameters are the extreme values taken from section 3.5. Two plots are shown, one with dry ballast values and the other with wet ballast values. The values of the series elements are taken as midway between the two extremes.

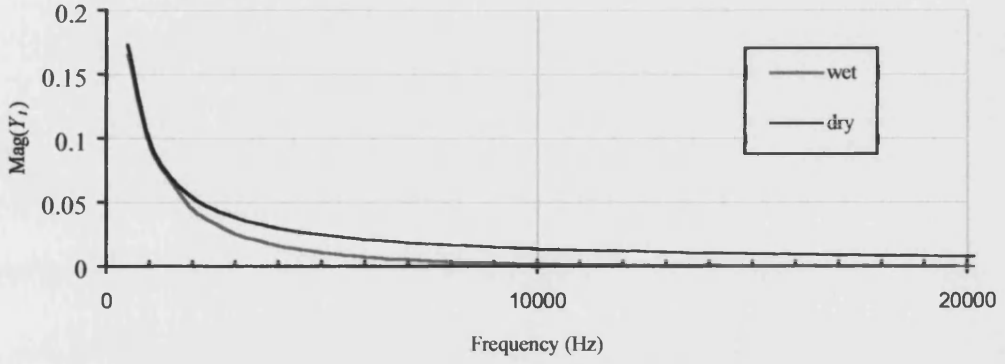


Figure 3.20. Track Circuit Admittance Transfer Function Magnitude

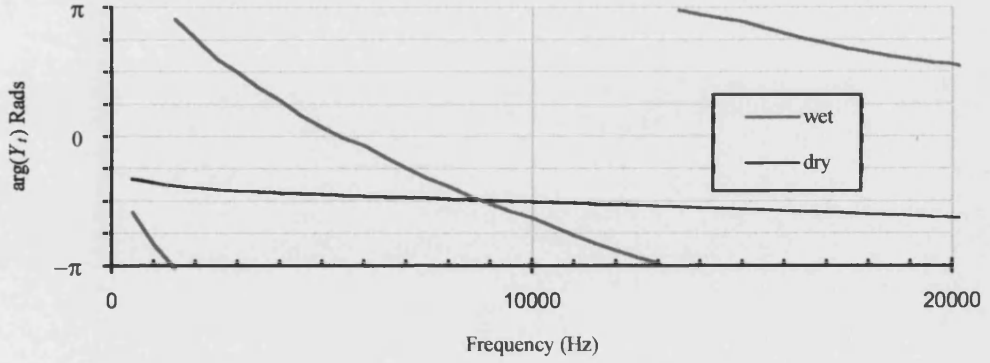


Figure 3.21. Track Circuit Admittance Transfer Function Argument

We can see how higher frequencies suffer greater attenuation, and how the environmental conditions increase the phase dispersion. Note that for signal bandwidths under approximately 1kHz, the phase response is approximately linear.

3.8 Comments on Track Parameters.

It has been seen that the series elements (R and L) have the least variation, as they are not greatly affected by environmental conditions. In fact, in early track circuit analysis, R and L were usually taken as constant for a particular frequency. The shunt parameters are highly influenced by environmental conditions as well as the physical properties of the material used. They are important as they have a large influence on the dynamic range of the track circuit receiver current. The condition of the rail ties/insulating pads, sleepers and the ballast affect the shunt impedance and the operation of the track circuit. In wet conditions the shunt impedance and the current in the track circuit receiver are reduced. In dry conditions, and where the ballast is not in direct contact with the rails, the ballast impedance can sometimes be considered to be infinite [24]. Hence the range of shunt conductance and capacitance must be accurately specified so that the safety of the track circuit can be proved.

For audio frequency track circuits, the series rail inductive impedance predominates over the resistance at frequencies above approximately 500Hz [23]. Also, at audio frequencies, the shunt capacitive impedance becomes low, causing high attenuation and transmission line effects, such as standing waves, for long track circuit lengths. However, it can generally be stated that train detection is generally improved at higher frequencies [24].

4. A New Track Circuit Transceiver Termination

4.1 Introduction

This chapter will outline the proposal for a different type of track circuit which will achieve the following primary aims:

- Improved sensitivity to the presence of a train shunt
- Increased rail to rail voltage throughout the track circuit to increase the shunting capability with contaminated rails
- Low cost
- Fewer connections to the rails
- No tuned area
- Bi-directional working
- Wideband signal with immunity to traction interference
- Improved broken rail detection
- Increased data capacity

This track circuit design has been proposed by Westinghouse Signals Ltd [19][40] and it is not the purpose of this thesis to study the hardware in detail. However, simulations have been performed, demonstrating the improvements to train detection offered by this design, and the results are presented in the following chapter.

This chapter (and the following chapter) will form a background to the main thesis topic of coding and modulation schemes which will achieve the primary aims above. The most important of these aims is to increase the data capacity, which will offer the potential of increased redundancy, increased safety, increased tolerance to interference, improved track to train communications as well as the features presented in section 4.4.

4.2 Track Termination

The main difference between this track circuit design and previous designs is that active terminations will be used instead of passive tuning to one frequency. The receiver, transmitter and termination are combined in a low impedance (ideally short circuit) connection to the track. One such unit will be called a *transceiver*. The currents flowing through the transceiver can be measured to form the receiver and a signal can be injected through the termination to form the transmitter. Since the transceiver is active it has the ability to limit its current when a train shunt is near. This means that trains can be detected by the low track impedance at the transceiver as well as the reduced signal level from other

transceivers. This property is needed because a low impedance termination, although good for minimising feed through voltage to other track circuits, is not ideal for train detection (for typical track circuits the termination impedance must be higher than the train shunt impedance for detection near the receiver). The design of the transceiver is outlined below :

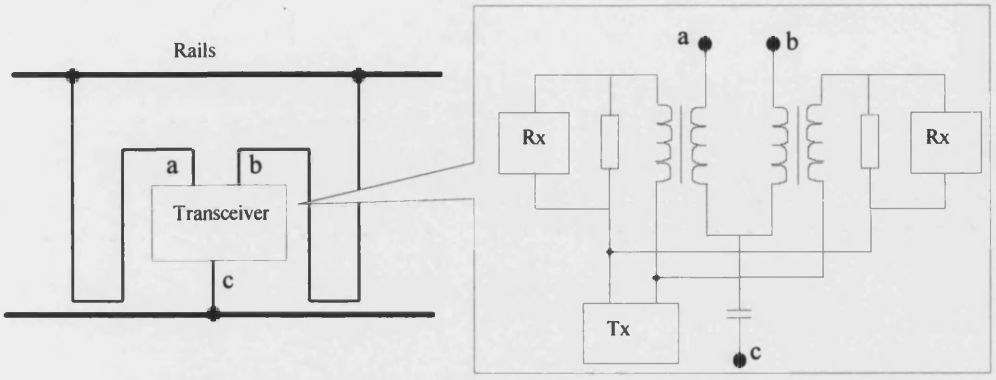


Figure 4.1 Outline of Transceiver Design.

The transceivers will form overlapping centre-fed track circuits. Each track circuit will have signals from both transceivers present. We shall call these signals 'channels' and assume that they occupy separate frequency bands. The minimum number of channels required is three.

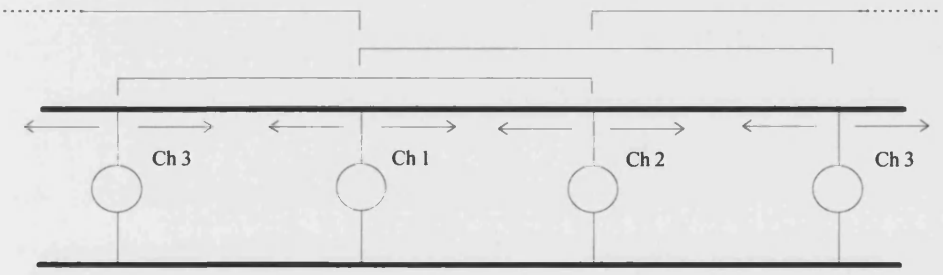


Figure 4.2 Overlapping Track Circuits

Figure 4.1 shows that the transceiver has three connections to the rail, compared to 8 connections for a tuned termination with a double shorting strap. The transceiver is able to distinguish between trains from the left and the right because of the inductive coupling to the rails and the longer impedance path. The transceiver can be modelled as an ideal voltage source, with ideally zero impedance at the rails to the signalling currents. For long track circuits, standing waves may occur, caused by the mismatch of characteristic and termination impedance. It must be shown that these standing waves will not cause a voltage null below the train receiver threshold. Also, the frequency separation of two channels must be large enough to ensure that the beat frequency does not cause a long null, otherwise the

voltage could be too low to break down the contamination layer over the sampling time of the receiver.

4.2.1 Improved Rail to Rail Voltage

A high (unoccupied) rail to rail voltage is good for train detection because it is more likely to break down the contamination layer between the rail and wheel. For track circuit designs with separate transmitter and receiver, the rail to rail voltage will decrease towards the receiver and is kept higher by the termination impedance. For the transceiver termination the *received* voltage will reduce to zero, where the *transmitted* voltage will be maximum. With no transmission line effects the total rail to rail voltage from both channels will be constant along the length the track circuit, as shown below.

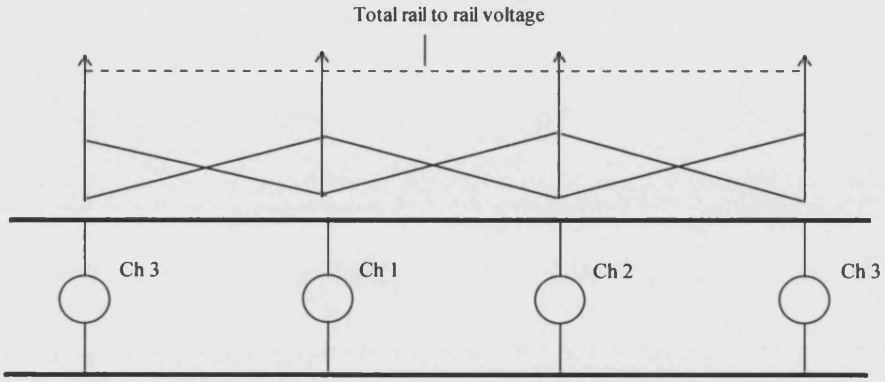


Figure 4.3 Track Circuit Voltages

4.2.2 Improvements to Train Detection

Train detection will be improved since, on a single track circuit there will be four ways that a train can be detected, instead of one. The two transceivers at the track circuit ends can detect a train by a reduced signal from adjacent transceivers or by seeing a low impedance caused by a train shunt. When a train is close to a transceiver it may not be able to detect it from the reduced signal levels because the termination impedance is low. However the current drawn form the transceiver will have increased and the train will be detected. Similarly, when the train is further away, there will be a point where the loop impedance will be similar to the impedance of the unoccupied track circuit. At this point the received levels will be lower and the train will be detected. The improvements to train detection are demonstrated by the simulation presented in the following chapter.

4.2.3 Upgrading Existing Equipment/Fault tolerance

Since the terminations are active, they should present a high enough impedance to the track when powered down, to not affect the track circuit operation. When upgrading existing track

circuits, the transceiver could be installed and powered up when the old equipment is disconnected. It is also possible that for fault tolerance, two transceivers could be connected in parallel, with one as a standby unit.

4.3 Track Circuit Signals.

An advantage of using the transceiver termination is that there is no tuning unit to limit the bandwidth of the signal. A wider bandwidth signal can be used, increasing the amount of track circuit and ATP data. The bandwidth may be limited by the channel distortion and the available transmitter power depending on the modulation scheme. Traditional serial binary modulation techniques could be applied. In section II, a synchronisable block coding scheme is presented, suitable for a serial binary data stream.

These techniques, however, may not be optimum in terms of data capacity within a time and bandwidth constraint. Also, the envelope of the signal must be controlled, since the amplitude conveys track circuit occupancy information and provides a breakdown of the contamination layer. Section III proposes a parallel tone modulation technique which is suitable for the electrical environment and meets the required data capacity within a signal bandwidth limitation, needed to ensure backwards compatibility with existing tuned terminations.

4.4 Transceiver Potential Features.

The following are potential features of the track circuit design which could be investigated:

Measurement of Train Position Within the Track Circuit

The change in impedance of the track as a train shunt approaches could be calibrated against distance (within the tolerance of the track parameters). This distance information could be used for decreasing headway or decreasing the number of track circuits needed in a section. If the trackside equipment knew the position of the train within the track circuit, it could alter the speed commands to the train as it moves through the block. Here the situation would be more like a moving block system, where the need for an overlap track circuit between the train and an obstacle is eliminated and headway in reduced. Another possible method of achieving this reduced headway is to have the train calculate its own braking profile from a knowledge of its position within the track circuit. The train would need to know the number of track circuits from the obstruction, the track circuit length, the gradient of the track circuit etc. To achieve this, the track circuit signal would need to have a large data capacity.

The transceiver current could also be used to zmeasure the track parameters, and detect a broken or damaged rail.

High Speed Intermittent Data to Train

When the train is close to the transceiver it could transmit a burst of non-safety critical data at high speed (i.e. 40-100kHz) to the train.

Self Initialisation

With tuned terminations, each track circuit has to be manually tuned and the receiver threshold configured, which takes time and is expensive to carry out. It would be an advantage if the transceiver was to automatically initialise after installation.

5.2.1 Wavelength

The wavelengths for extreme track parameters have been calculated and are shown in figure 5.6.

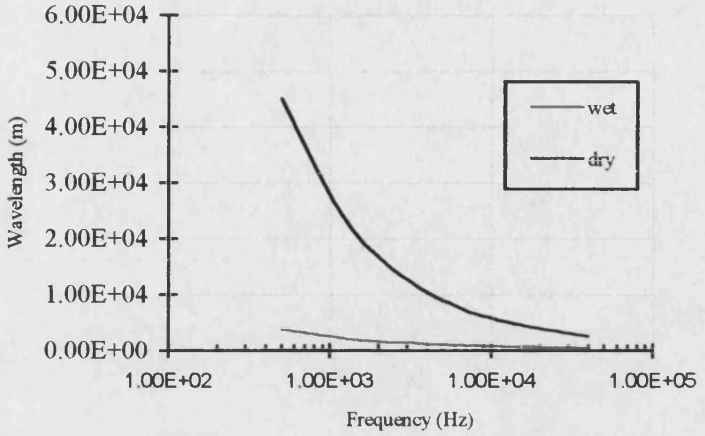


Figure 5.6. Signal Wavelength for Railway Track

It can be seen that for long track circuits, especially at higher frequencies, the wavelength can approach the track circuit length and standing waves may occur. This is demonstrated in figure 5.7 which shows the voltage distribution along a 1.5km track circuit at 10kHz. The termination is set to open circuit. *R, L* and *C* are set to typical values, and *G* is set to its lowest value (for dry track). This combination produces the highest standing wave variation.

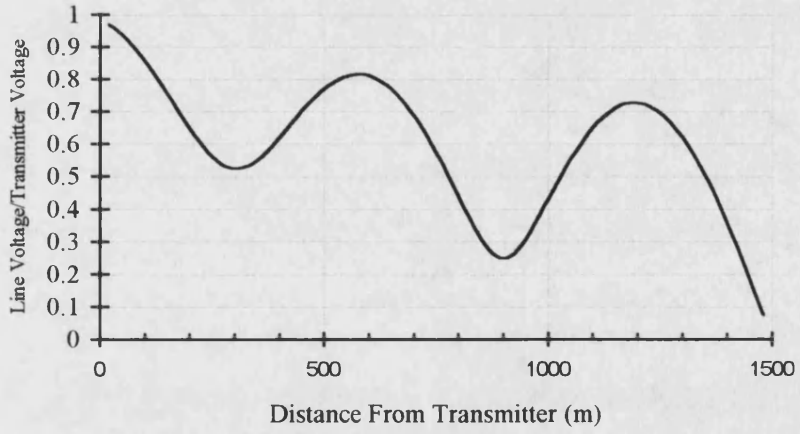


Figure 5.7. Voltage Standing Waves.

The distance between the two troughs represent the half wavelength, which is measured to be approximately 615m, giving a wavelength of 1230m. The calculated wavelength is 1237m.

5.3 Results for Existing Track Circuit Terminations

With the track circuit model, the voltage and current distributions of existing systems can be investigated. With reference to Figure 5.1, *Es* is set to 60v, *Er* is set to zero (passive receiver), *Zs* and *Zr* are set to 1Ω. This is the signal frequency impedance of the tuned termination, of

the type used for the Westinghouse FS series of track circuits [19]. This is shown in figure 2.2. *Ztr* is set to 0.5Ω (worst case maximum for British Isles). The track circuit length is 1km and the frequency is 2kHz. Typical values of *R, L, G* and *C* were used.

Figures 5.8 and 5.9 below show the transmitter current and the receiver rail to rail voltage. The *x* axis is the train shunt position, from the transmitter. The dashed lines are the values with no train shunt present.

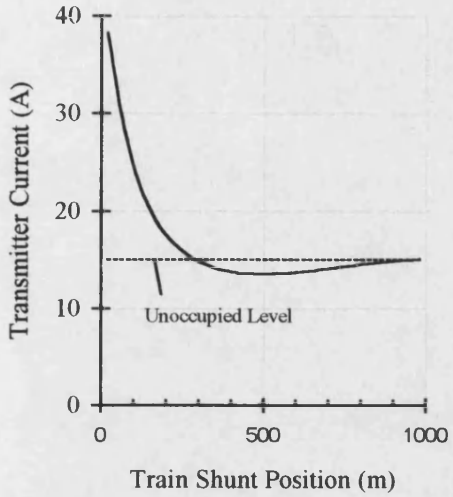


Figure 5.8. Transmitter Current

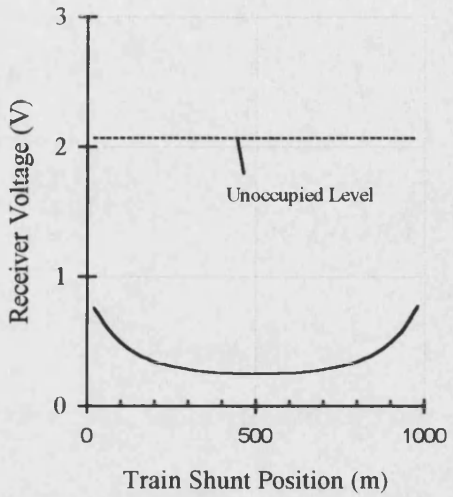


Figure 5.9. Receiver Voltage

A sharp rise in transmitter current can be seen as the train approaches the transmitter, and there is a position where the current drawn is the same, with or without a train shunt. Figure 5.9 shows that the worst position for train shunt detection is near the transmitter or receiver. The best position is in the middle of the track circuit. Note that with tuned terminations, there will actually be less variation in the total transmitter current, since the current is fed through a low impedance series resonant path.

Figure 5.10 and 5.11 below show the current through the train shunt and the unoccupied track circuit voltage.

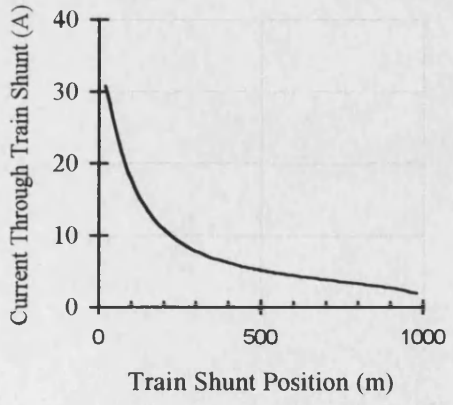


Figure 5.10. Current Through 0.5Ω Train Shunt

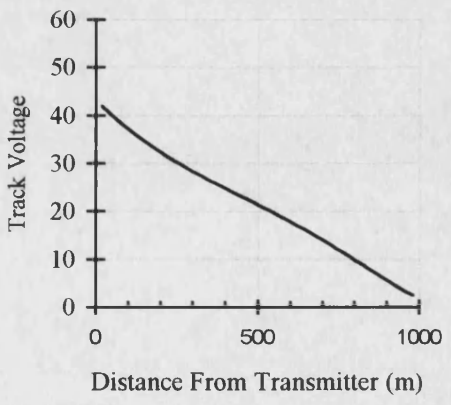


Figure 5.11. Unoccupied Track Circuit Voltage

The track circuit voltage distribution is not quite a straight line due to transmission line effects caused by the mismatch of characteristic impedance to termination impedance. We can see that towards the receiver the rail to rail voltage falls below the 6v recommended breakdown voltage.

Reducing Termination Impedance.

Figures 5.12 and 5.13 shows the transmitter current and receiver voltage when the termination impedance is reduced to 0.5Ω.

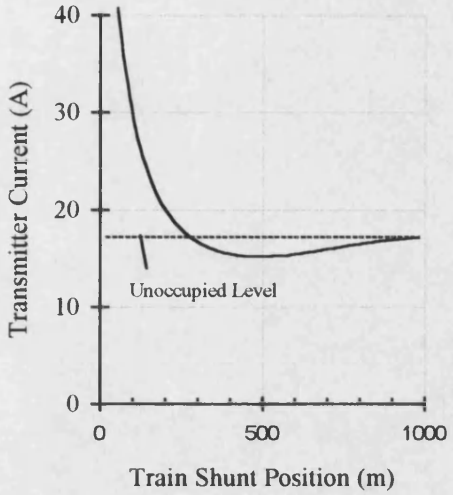


Figure 5.12. Transmitter Current with termination Impedance reduced to 0.5Ω

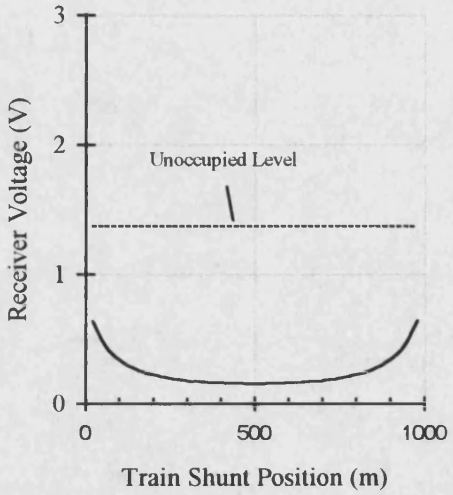


Figure 5.13. Receiver Voltage with termination Impedance reduced to 0.5Ω

With a reduced termination impedance, the current drawn from the transmitter has risen and the receiver voltage is less sensitive to the presence of a train shunt. The maximum reduction in receiver voltage with a train shunt present is only 8.5dB compared to 18dB for figure 5.9. So, for existing systems, train detection is increased with increased termination impedance. This has been limited by several factors:

1. Feed through voltage.
   Increasing the termination impedance will increase the leakage into the next track circuit (although with tuned terminations, e.g. FS2000, the track circuits can be isolated by shorting straps).

2. Signal bandwidth.
   With tuned terminations, the resonant Q factor is kept low so that the receiver will detect both sidebands of an FSK signal.

3. Circuit sensitivity.

   A low Q factor has the effect of lowering the termination impedance, but the resonant peak is less sensitive to circuit changes.

So, the choice of termination impedance has been a compromise between train shunt sensitivity and the above factors.

5.3.1 Receiver Threshold Level and Dead Zone

The threshold level must ideally be set so that the receiver can detect the difference between an occupied track and an unoccupied track under all conditions. The track series impedance is usually taken as being constant for all conditions. The relationship between the variations of C and G is not know and often C is also held constant. However, the graphs below show some of the effects of varying the track parameters. Figures 5.14 and 5.15 show the same graphs as figures 5.8 and 5.9, but with two extra traces. R, L and C are kept at their typical values and G is taken as the extremes values (wet and dry). Note that the unoccupied levels (no train shunt present) are shown by the straight lines.

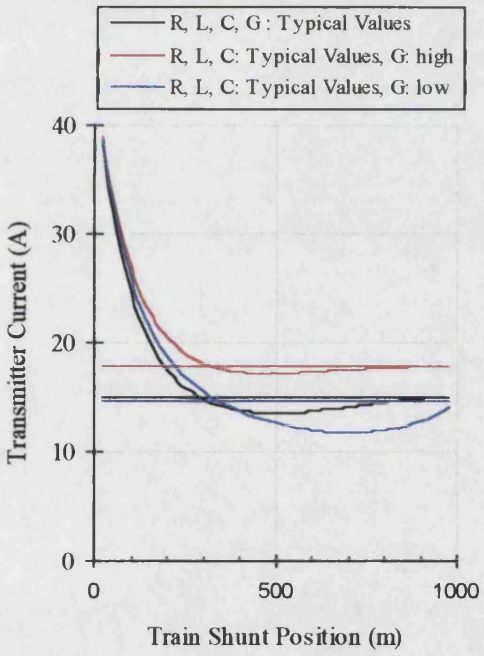


Figure 5.14. Transmitter Current.

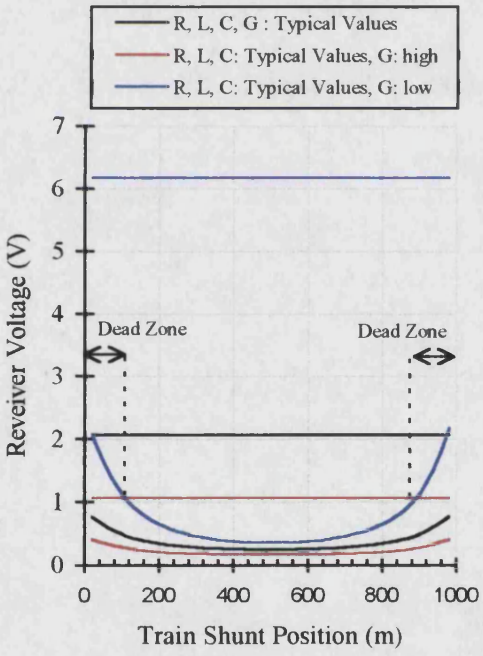


Figure 5.15. Receiver Voltage

With these results, it can be seen that any receiver threshold setting will result in a considerable dead zone (≈210m) at each end of the track circuit (note that this dead zone is different from the dead zone produced by a shorting strap). Only a large increase in termination impedance would reduce or eliminate this dead zone. Figures 5.16 and 5.17 show the same curves, except that both C and G are taken as their extreme wet and dry values.

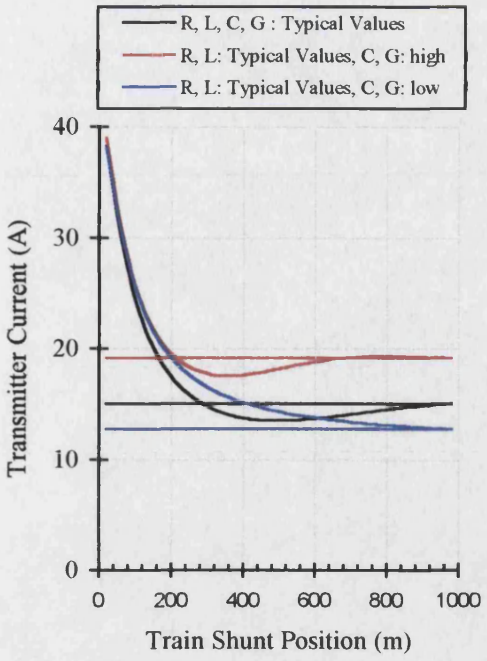


Figure 5.16. Transmitter Current.

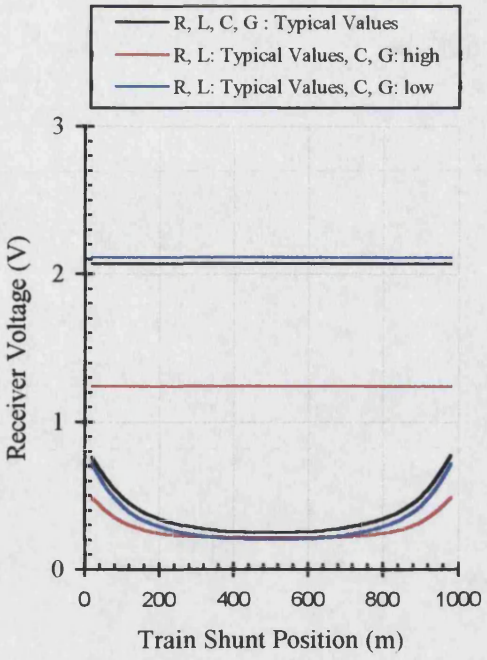


Figure 5.17. Receiver Voltage

Here, the optimum receiver threshold would be approximately 1 volt, and there is no dead zone.

For the majority of the time, the track parameters will be centred around the typical values and it not likely that extreme values of G will occur at the same time as typical values of C. However, as the relationship between G and C is not known, a large increase in G accompanied by a small increase in C could produce a situation approaching figure 5.15.

Another point of note, from figures 5.14 and 5.16, is that the sharp increase in current drawn as the shunt approaches the transmitter is relatively independent of variations in environmental conditions up to 300m from the transmitter. This increase in current occurs where the receiver is least sensitive to the presence of a train shunt (and where a dead zone of figure 5.15 is most likely to occur).

5.3.2 Frequency Variations

Railway track represents a dispersive, time varying transmission medium, with a slow propagation velocity, due to high losses and a non-linear phase response due to the variation of track parameters with frequency. Figure 5.18 shows the variation of characteristic impedance with frequency.

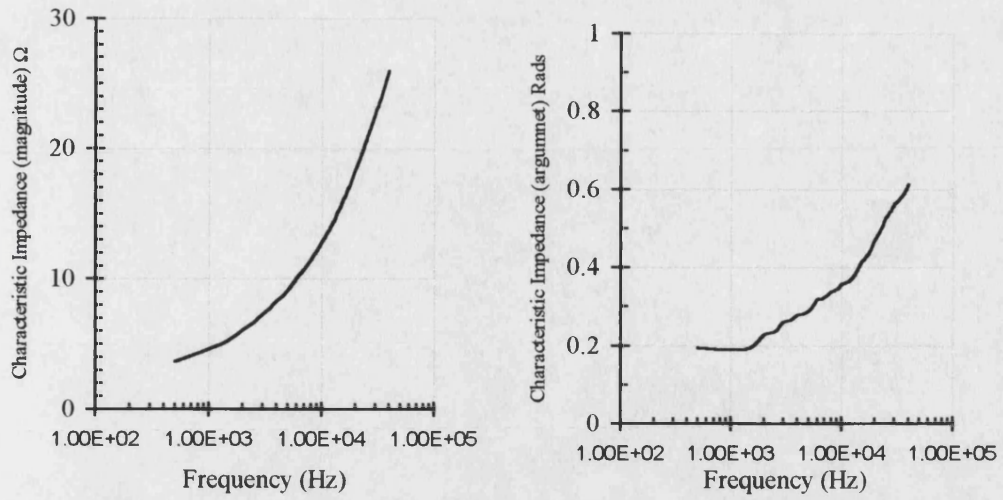


Figure 5.18. Railway Track Characteristic Impedance (Magnitude and Argument)

Figures 5.19 and 5.20 show the receiver voltage and transmitter current for various frequencies (the same parameters as in figure 5.9 are used).

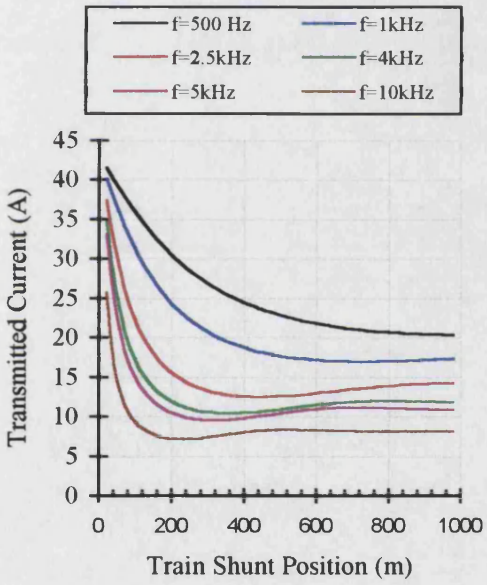


Figure 5.19. Transmitter Current with a Moving Train Shunt, for Different Frequencies.

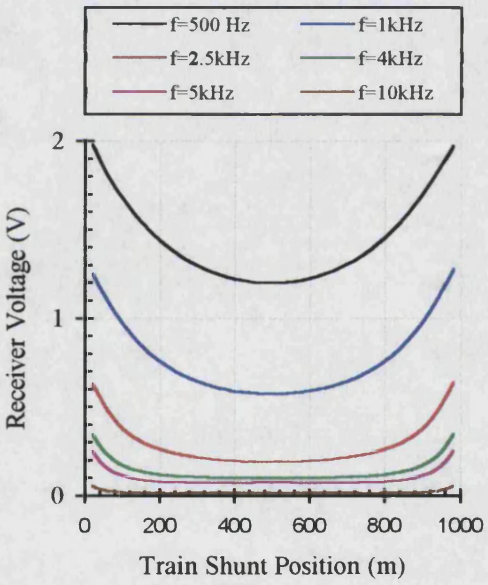


Figure 5.20. Receiver Voltage With a Moving Train Shunt, for Different Frequencies.

The large variation in receiver voltage with frequency shows that the threshold for detecting track occupancy needs to be set for a specific frequency (or narrow band FSK signal). This also demonstrates how higher frequencies suffer higher attenuation and that using a lower signal frequency will result in less power and a longer possible track circuit length.

5.4 Results for Transceiver Termination

With reference to figure 5.1, the two terminations can both be made voltage sources, with *Er*=*Es*=60V rms. Each voltage source will inject current to the left and the right to produce

overlapping track circuits. Figure 5.21 shows the total rms rail to rail voltage along a 1km track circuit for various frequencies. The source impedances, *Zs* and *Zr*, are set to 10mΩ and the impedance *Zo* is set to the characteristic impedance at the frequency being simulated.

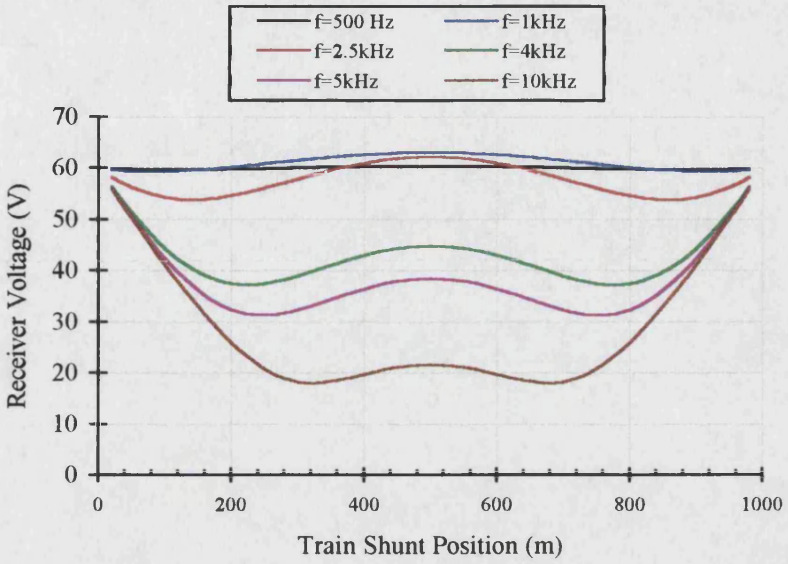


Figure 5.21 Track Circuit Rail to Rail Voltage

For each trace both *Er* and *Es* are at the same frequency which causes symmetrical standing waves for the higher frequencies. For train detection, the currents from each source need to be separated by the transceiver, which is usually achieved by using separate frequency channels which will cause non-symmetrical standing waves.

So, the track circuit can now be considered as two superimposed track circuits, with transmitter and receiver at opposite ends to each other, and a low termination impedance. This is shown in figure 5.22. Also shown are the transmitted and received currents for each 'channel' of a 1km track circuits at 2kHz. The graphs assume that the track parameters are the same for both channels (taken as typical values), and that both channels can be isolated perfectly at the transceivers. The dotted horizontal lines represent the unoccupied track currents. It can be seen that when the train comes within 260m of the termination its current starts to rise sharply and the 'transmitter' will detect the train shunt. Above the detection threshold, the current will have to be limited. Each track circuit section has 2 channels and 2 transceivers and so there are four ways that the train can be detected.

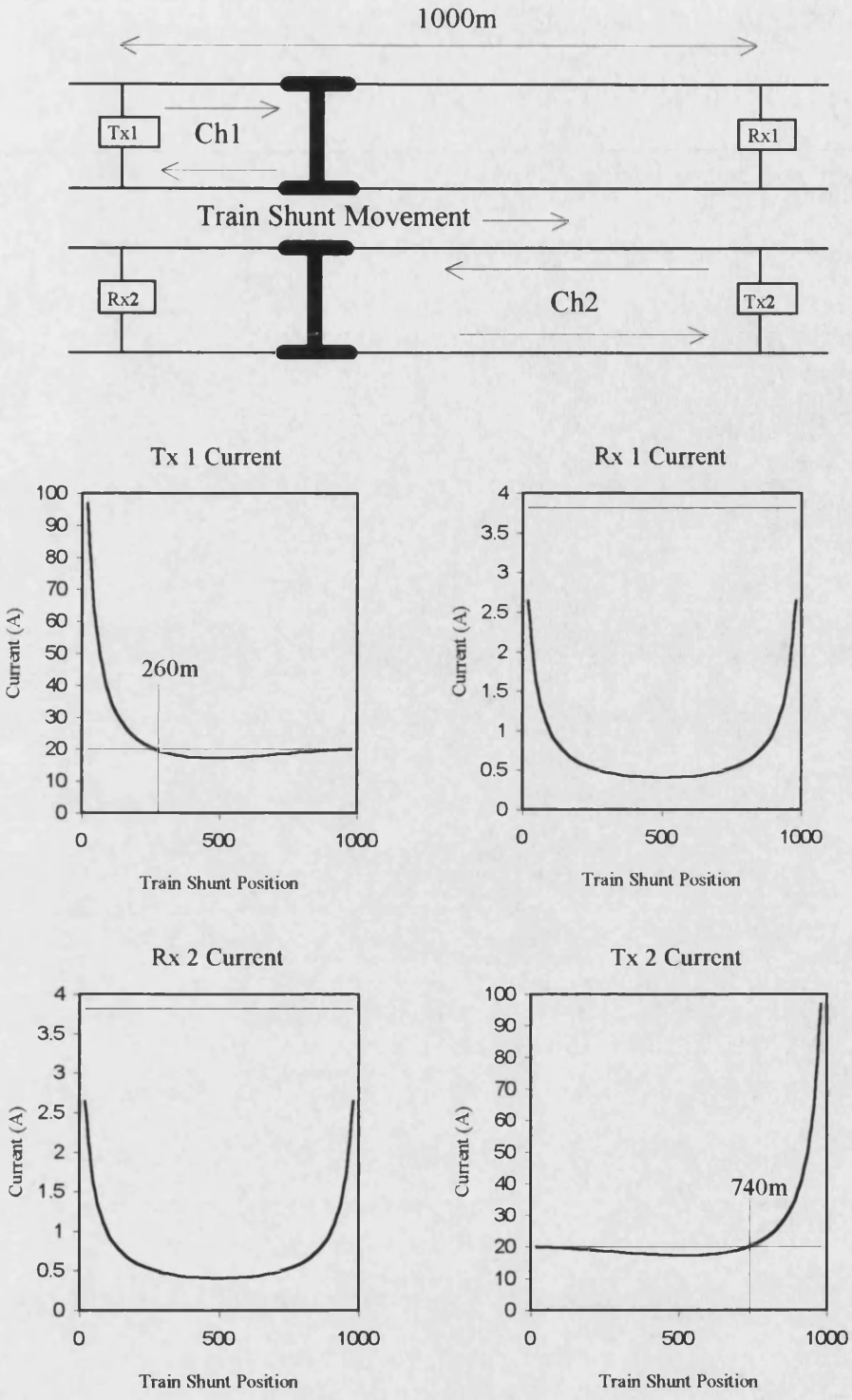


Figure 5.22. Transceiver Currents For Moving 0.5Ω Train Shunt

5.5 Improvements to Train Detection.

5.5.1 Transceiver Detection of Low Impedance.

The low impedance caused by a train shunt near the transceiver will cause the output current to rise. In this way the presence of a train shunt is detected. Because adjacent track circuits

have overlapping channels, the transceivers must also determine on which side this train shunt is. This could be achieved by the transceiver detecting a low impedance and then deciding which of the adjacent channels is below the threshold for occupancy. However, because of the low termination impedance of the transceiver, the reduction of received signal (with a train shunt present) is less than that of the tuned terminations. For example, figure 5.9 shows a 1km track circuit with a termination impedance of 1Ω, using typical rail parameters. The worst case reduction in received signal (at the ends of the track circuit) is approximately 8.5dB. The received current levels of figure 5.22 (using the same rail parameters) show a worst case reduction of approximately 3.5dB.

The termination is achieved with three connections to the rail, as shown in figure 4.1. The transceiver can determine which direction has the lowest impedance and accurately detect on which side the low impedance is, and when a train has passed into the next track circuit.

5.5.2 Transceiver Threshold Levels.

Each transceiver will have two threshold levels to determine track circuit occupancy: one for its transmitted signal and one for its received signals. If the received current is below the threshold and/or the transmitted current is above the threshold then the track is assumed occupied.

The position of these threshold levels can be chosen to optimise the train shunt sensitivity under all conditions.

5.5.3 Train Shunt Sensitivity.

5.5.3.1 Existing Track Circuit Termination.

For comparison, here is a brief look at train shunt sensitivity of existing track circuits of the type used for the Westinghouse FS series. For reliability, the receiver level must fall above the threshold when no train is present, in extremely wet conditions and for the longest track circuit. For safety, the receiver level must fall below the threshold when a 0.5Ω train shunt is present in the driest conditions and for the shortest track circuit.

A measure of this sensitivity could be:

$$S = \frac{\text{Maximum received signal under wet conditions with no train shunt}}{\text{Maximum received signal under dry condition with a 0.5}\Omega\text{ train shunt}} \qquad (5.6)$$

Ideally this ratio should be greater than one for all possible conditions. Figure 5.15 showed the receiver voltage levels for a 1km track circuit, at 2kHz with a 1Ω termination impedance. Extreme wet and dry values of G and typical values of R, L and C were used. Figure 5.23 shows the sensitivity of the receiver for these conditions.

Figure 5.17 showed the receiver voltage level for the same track circuit using extreme wet and dry values of G and C, and typical values of R and L. Figure 5.24 shows the sensitivity of the receiver for these conditions.

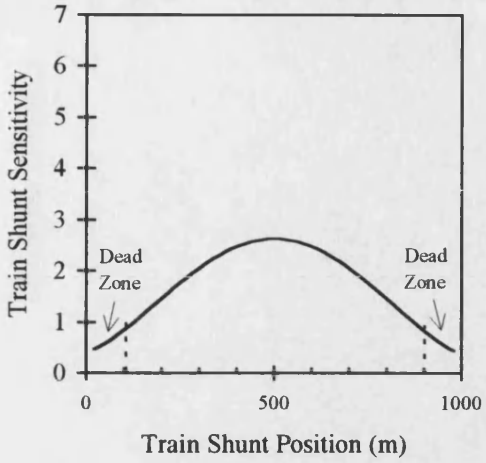


Figure 5.23. Train Shunt Sensitivity for Extreme values of C, G and Typical Values of R, L.

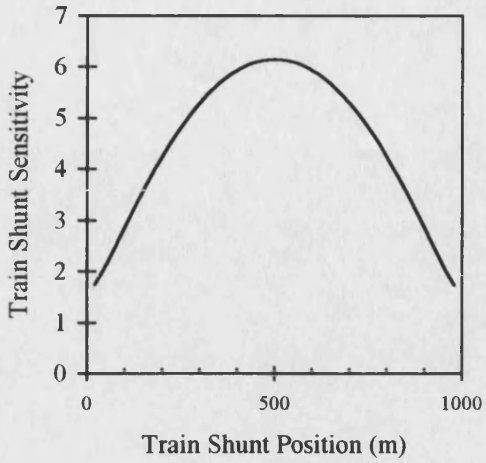


Figure 5.24. Train Shunt Sensitivity for Extreme values of G and Typical Values of R, L and C.

It can be seen that for figure 5.23, the sensitivity falls below 1, and a train shunt cannot be detected at the ends of the track circuit for these conditions.

5.5.3.2 Transceiver Termination.

For track circuits terminated by transceivers, there will be four selectivity ratios. Referring to figure 5.22, these are:

1. Sensitivity for Rx1, receiving Channel 1.
2. Sensitivity for Tx1, transmitting Channel 1.
3. Sensitivity for Rx2, receiving Channel 2.
4. Sensitivity for Tx2, transmitting Channel 2.

Assuming that both channels are attenuated by the same amount, 1 and 2 will be the same as 3 and 4. Figures 5.25 and 5.26 shows the transmitted and received currents of the transceiver under the same conditions used for figure 5.23 (R, L, and C, typical values, G wet/dry values, f = 2kHz, track length = 1km).

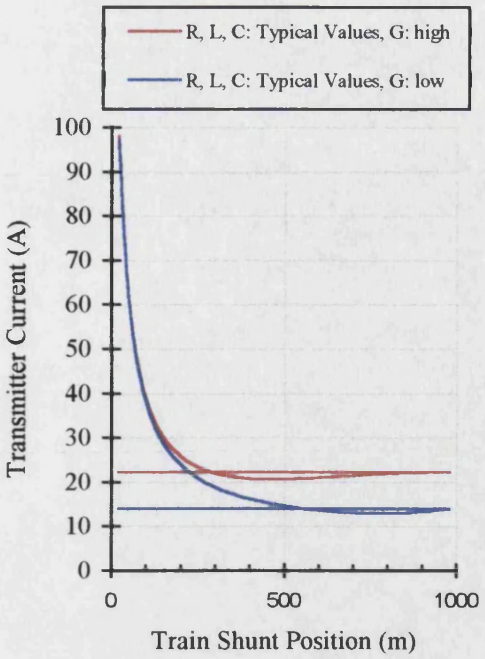


Figure 5.25. Transceiver Transmitted Current, for Extreme values of G and Typical Values of R,L,C.

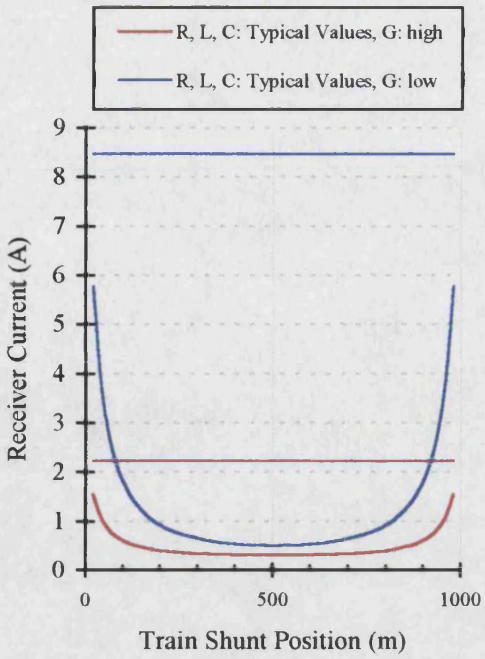


Figure 5.26. Transceiver Received Current, for Extreme values of G and Typical Values of R,L,C.

Figure 5.27. shows the received and transmitted current sensitivity. (Note: the sensitivity for the transmitted current is taken as the reciprocal of *S*)

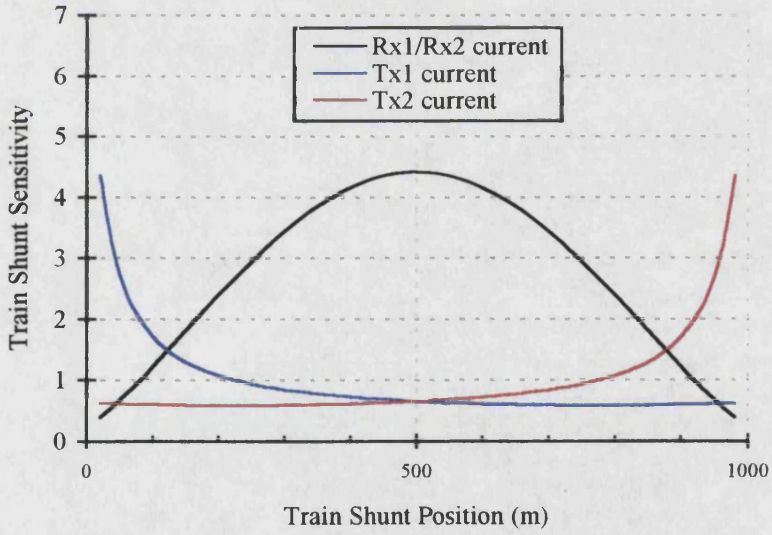


Figure 5.27. Sensitivity of Transceiver Currents for Extreme Values of G and Typical Values of R,L,C.

It can be seen that at no point do all four sensitivity ratios fall below 1, and a train shunt will always be detected under these conditions. The lowest sensitivity is 1.5 and occurs 110m from the transceiver. This represents a 50% change in the received and transmitted signal levels. Figure 5.28 shows the transceiver sensitivity ratios, with typical values of *R* and *L*, and extreme values of *G* and *C*.

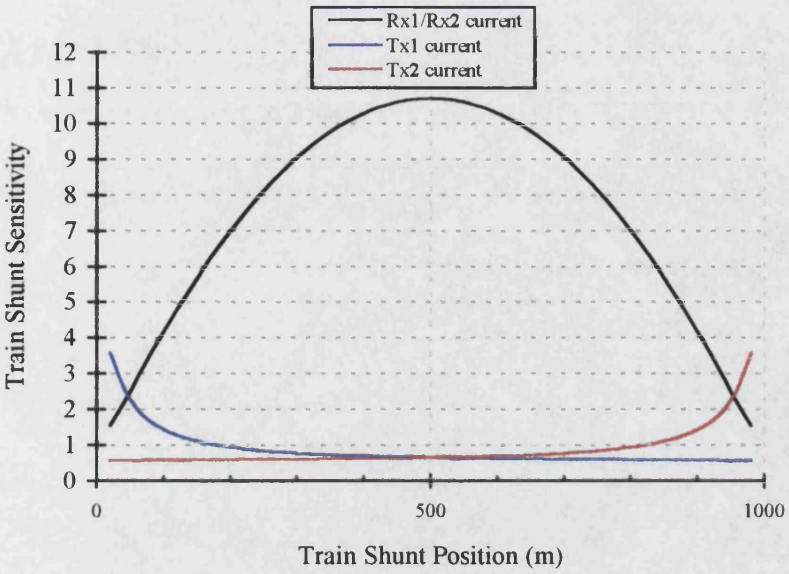


Figure 5.28. Sensitivity of Transceiver Currents for Extreme Values of C, G and Typical Values of R, L.

Under these conditions the lowest sensitivity is 2.6 and occurs 50m from the transceiver. This represents a 160% change in the received and transmitted signal levels.

5.6 Comments

This chapter has presented a method of mathematically modelling the track circuit with a moving train shunt, using the transmission line theory of the previous chapter. The track circuit model simulates the general, ideal case, ignoring the effects of earth leakage, structure/impedance bonding and parallel rail mutual inductance. For specific sites where these effects occur, a three or greater wire model would have to be used to simulate the track circuit performance [42][43], assuming accurate parameters are available. Also, it is assumed that the distributions of *R, L, G* and *C* are uniform.

In these simulations the transceiver has an impedance of 10mΩ. This value is not critical to the sensitivity calculations shown in section 5.5.3, but the impedance will contribute to the amount of feed through voltage that occurs and so ideally is kept as low as possible. Simulation results have been presented for a 1km track circuit at a signal frequency of 2kHz.

This thesis is concerned with improving the track circuit data communications, assuming a potentially wider signal bandwidth than is offered by passive tuned termination. Track circuits have been tuned to one particular frequency to increase the termination impedance and improve the train shunt sensitivity, especially near the terminations. This chapter has shown that improved train shunt detection is possible, even with a very low transceiver termination impedance, if the transmitter can also detect a train shunt by monitoring its output current.

Since each track circuit is overlapping, and supports two channels, there are four ways that a shunt can be detected, instead of just one.

6. Modelling Power and Traction Interference

This chapter considers the sources of interference that will affect the track circuit signal. The assessment of any track circuit coding or modulation performance in AWGN would be simple and convenient. For this reason, all previous simulation and analyses of proposed schemes have used the Gaussian noise model [2][16][44][45]. However, the interference encountered in railway systems is unlikely to be AWGN. Any track circuit signalling scheme must be immune to the interference caused by power and traction schemes. This section will look at the possible sources of interference with the aim of defining a more general noise model to use in signalling scheme simulations.

6.1 Introduction

The primary source of interference to track circuit equipment comes from power substations, braking systems and traction systems. Electrified railways use one or both of the running rails as the return path. With d.c. traction the higher current requires that both rails are bonded, usually via impedance bonds, so that equal current flows through both rails. There will also be bonding to large metal structures to avoid voltage build ups.

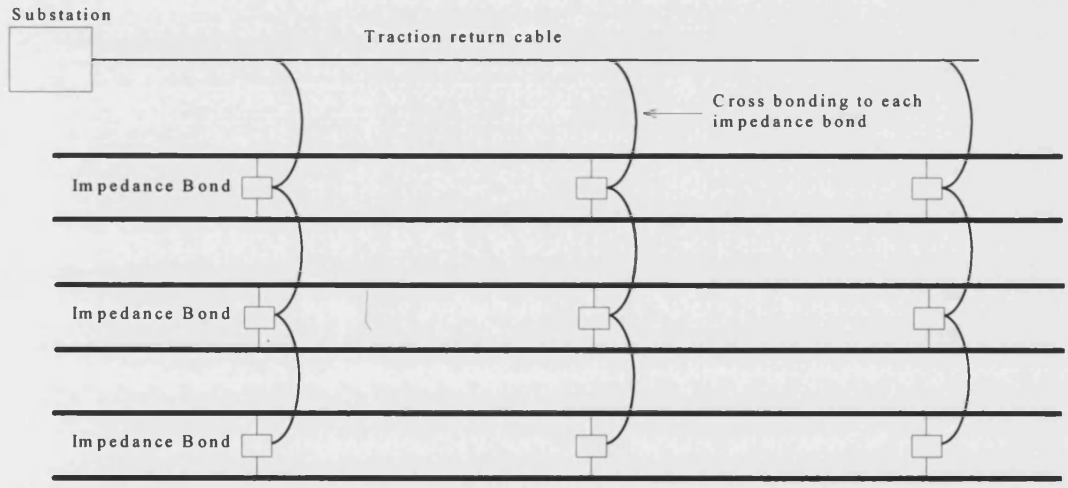


Figure 6.1. Cross Bonding of Parallel Tracks.

The traction return current will flow in the same direction through both rails, towards the substation. The signal current will flow in different directions through the rails and so two antennas above each rail are connected in antiphase, to increase the received signal current and cancel the traction current (figure 6.2). An imbalance in traction return current may be caused by faulty bonding/impedance bonds or a broken rail. In this case, the traction return current will interfere with the signal.

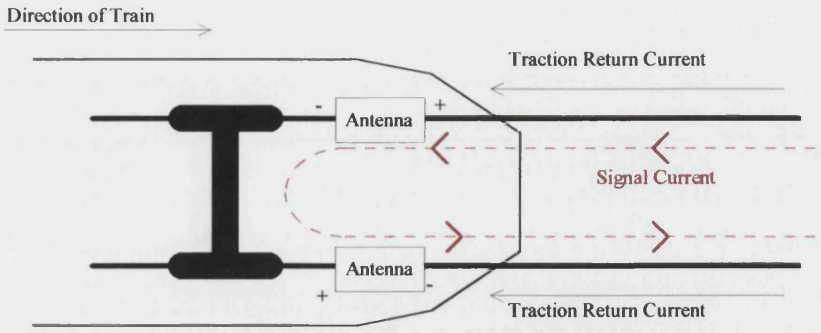


Figure 6.2. Train Mounted Antenna.

With a.c. traction, lower currents may mean that only single rail return is required and single rail track circuits may be used.

6.2 Sources of Interference.

The use of power electronic traction control is a major source of interference. In the early seventies, thyristors operating at audio frequencies allowed trains to use three phase motors [47]. The line voltage (15-25kV) is converted to d.c. and from this voltage, three phase invertors generate variable frequency and voltage for the motors. In d.c. supplied traction systems, rapid advances in gate turn off (GTO) technology enable chopper and invertor equipment to replace the less efficient rheostatically controlled equipment [47][48] . This equipment operates at frequencies well below a.f. track circuit frequencies, but produces high harmonic levels within the signal band, through conductive or inductive coupling. Formulae for the magnitude of the chopper harmonics have been proposed. From the total d.c. current, $I_{d.c.}$, the fundamental chopper frequency, f_c, and the resonant frequency of the input filter, f_o, the equation below gives us the magnitude of the integer multiples of the frequency $u.f_c$ for a u-phase chopper circuit.

$$I_I(nuf_c) = \frac{\sqrt{2} \cdot I_{dc}}{nu\pi} \left(\frac{f_o}{nuf_c} \right)^2 \qquad (6.1)$$

This equation alone cannot be used as a model of chopper interference since the parameters $I_{d.c.}$, f_c and f_o will vary with different equipment and tolerances. Also, due to chopper circuit imbalance, other components at multiples of f_c will occur. In the time domain, the interference appears as a series of impulses of varying period depending on the load on the train. Figure 6.3, taken from reference [51], shows an oscilloscope waveform of the interference from a chopper controlled train fed from a 600V d.c. overhead catenary system with traction return through the running rails.

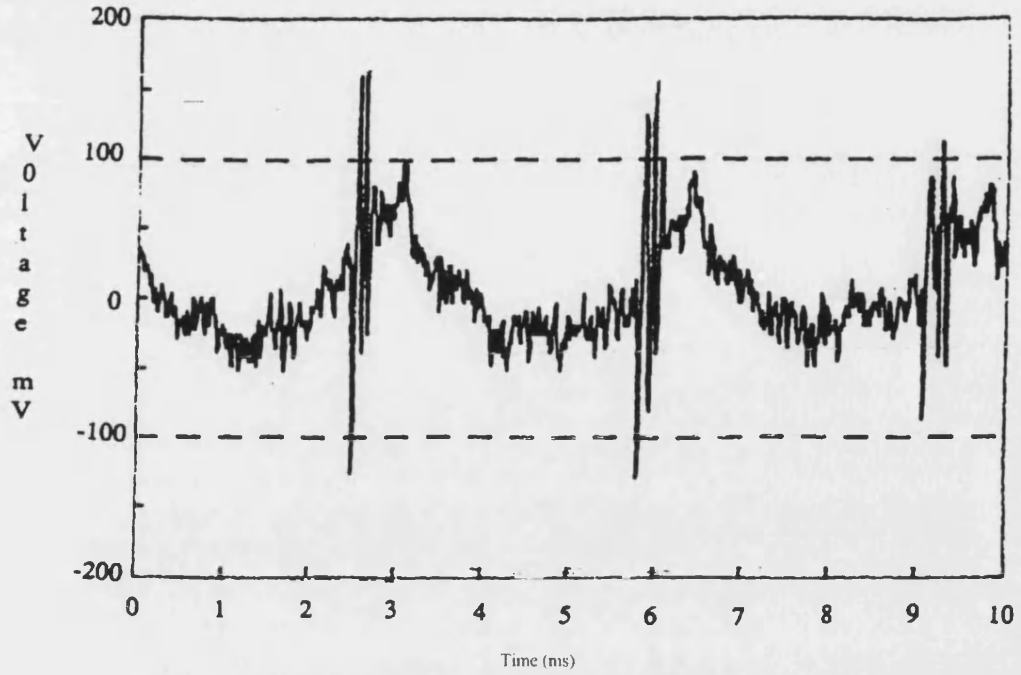


Figure 6.3. Interference from a 600V d.c. Chopper Controlled Train.

These impulses were seen to vary from 3ms to 10ms at a rate of 1ms/s. Other, irregular impulses were also observed.

Invertor controlled induction motors generate less predictable noise and harmonics [1]. Practical invertor equipment will normally be connected to the d.c. supply via a d.c. chopper. The input spectrum will normally consist of harmonics of the chopper frequency f_c, with sidebands at $(f_c \pm n6f_m)$ where n=1, 2, 3 etc. The output fundamental will typically vary from 4Hz to 100Hz.

Mains frequency harmonics will also be present in the traction supply and return current. Reference [1] states that for a 'p' pulse rectifier, the magnitude of the no-load voltage harmonics, in terms of the d.c. traction voltage, V_{dc}, are given by:

$$V(n,p,f_s) = \frac{\sqrt{2} \cdot V_{d.c.}}{(np)^2 - 1} \qquad (6.2)$$

where n=0,1,2,3 etc. In practice, other harmonics are present due to unbalance in the rectifier circuit. Also, rectifier fault conditions will cause higher harmonics in the audio frequency range. The harmonics produced from power substations are generally accepted to be the odd harmonics of 50Hz (i.e.50Hz, 150Hz, 250Hz etc.) [44][45], with the amplitude decaying exponentially, as shown in figure 6.4 below.

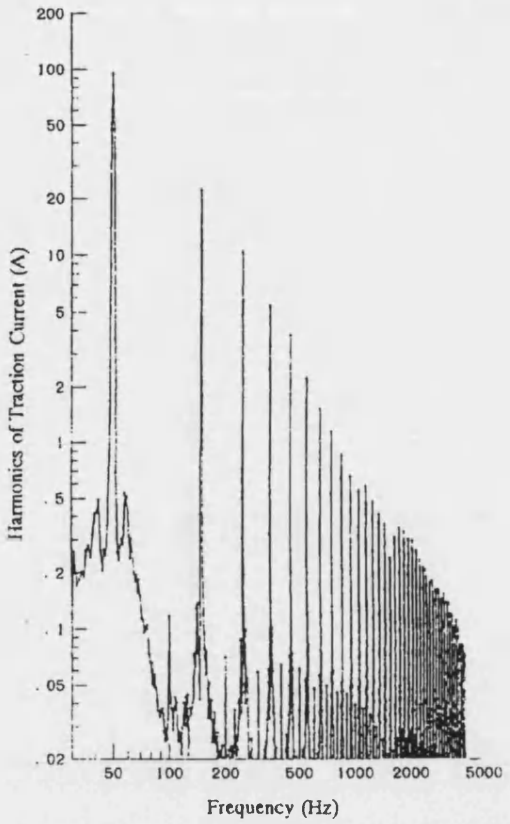


Figure 6.4. Harmonics of Traction Current for a.c. Electrification.

In the USA, the harmonics produced from power substations are generally accepted to be the harmonics of 360Hz (i.e. 360Hz, 720Hz, 1080Hz etc.) [46][48].

Mains frequency harmonics generated by trains depend on the type of train. Four quadrant choppers such as class 323 or the Eurostar trains generate mostly 50Hz, with harmonics mainly between 1500Hz and 2000Hz. Trains with phase angle control, two sequential bridges or rheostat control generate harmonics modelled by the following formula [49].

$$I_n = \frac{2I_{50}}{n^{1.6}} \tag{6.3}$$

where n=1,2,3 etc. and I_{50} is the 50Hz current.

'Gapping' can occur when there is ice or contamination on the track, causing breaks in the traction return path. A spike is formed which resonates the impedance bonds and track circuit receiver filter and this is known as 'bobbing'. With simple a.c. track circuits this phenomenon can falsely energise the track circuit relay even when a train shunt is present. These spikes can also be caused by collector shoe bounce at gaps in the third rail.

6.3 Using Power Harmonics for Signal Synchronisation

Several proposed track circuit signalling schemes [44][45][46] use a power interference harmonic for carrier synchronisation of the track circuit signal. It is intended that the transmitter's carrier is synchronised to a harmonic of the power frequency. This harmonic is also available to synchronise the train and track receiver to the signal carrier. Such a technique would solve the problem of receiver clock synchronisation for a discontinuous signal[1], because in theory, the harmonics will always be available at the receiver, and would be of the same phase and frequency as the harmonic at the transmitter.

For this method to succeed, the noise would have to be a constant harmonic of the fundamental (usually 50Hz). This may not always be the case [51][54], and the use of pulse width modulation, for example, will produce noise that is not 50Hz related. The catenaries could be fed from different phases of a three phase supply, and as the rail return is continuous, there could be multiphase 50Hz harmonics in certain areas. Also, the phase delays along the track for the synchronising signal must be addressed. If a low frequency harmonic is used then the phase delay (on a wet 1500m track, say) will be different from the delay of the a.f. signal carrier. For example, on a 1500m wet track, the difference in phase delays between a 150Hz tone and a 2kHz tone is approx. 0.33ms, which would produce a phase difference of approximately 240 degrees and so a coherent reference could not be extracted under these conditions.

It must be guaranteed that the power harmonics are always available, and that the rails are used for traction return current. This would exclude non-electrified and LUL lines, where the rails are not part of the traction return. Since the signalling scheme proposed in this thesis must be suitable for a.c., d.c., dual, and non electrified lines, the use of power harmonics for synchronisation will not be discussed further.

6.4 A General Interference Model

Very little specific interference data exists. Information that is specific to a particular location, equipment type and measurement method is of limited value since any new track circuit scheme must be designed to function with all classes of traction equipment (e.g. class 92, 323, 373 and 465) and electrification (e.g. 25kV a.c., 750V d.c., 600V d.c., 25kV a.c and 750V d.c. dual electrification) [52][53]. All the interference modes described above must be

[1] For the train born receiver, fast synchronisation must be aquired each time the train enters a new track circuit.

considered. There are also many other interference sources and fault conditions which are unpredictable.

The difficulty in describing a suitable noise model has led all previous researchers to use the Gaussian noise model for initial evaluation of communication systems in a railway environment. Because of the relatively high signal levels and the electrical environment, Gaussian noise is not a significant interferer and the model is unrealistic. The aim of this section is to describe a possible interference model which will apply to all the environments previously described and will be more accurate than the Gaussian model.

The common denominator of all the sources described in section 6.2 seems to be interference which is wideband and impulsive in nature. Chopper controlled equipment produces impulsive interference at a period not less that 3ms. This period varies slowly depending on the state of the train. Other interference sources may produce single isolated impulses. The model used in this thesis will assume that the noise consists of an impulse train. The period between the impulses will vary at a rate of 1ms/s, from 3ms to a period that will produce a single period in the observation interval, as shown below.

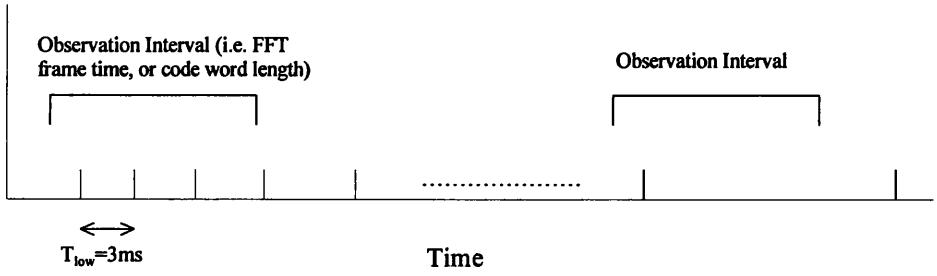


Figure 6.5. General Impulsive Interference Model.

These impulses will be shaped by the receiver filter into a series of overlapping or non-overlapping impulse responses, depending on the period between the impulses. This will be added to the signal.

This interference model will be used in the following two sections of this thesis.

SECTION II

Chapter 7: Serial Binary Coding Schemes Applied to Track Circuits.

Chapter 8: Synchronisable Cyclic Coset Codes for Railway Track Circuits

Chapter 9: Cyclic Coset Code Performance with non-coherent FSK and non-Gaussian Noise.

Section II is concerned with increasing the data capacity of the track circuit and the tolerance to interference and crosstalk, by the use of binary block coding schemes. Chapter 7 provides an introduction to some of the factors involved when applying a serial binary modulation and coding scheme to track circuits, to increase the data capacity. Previous binary coding schemes which have been applied to track circuits are discussed. Chapter 8 presents a new track circuit binary coding scheme. The code is fully synchronisable and the performance in terms of safe failure and unsafe failure with error correction is analysed. The interference noise model presented in chapter 6 is used in a simulation of the performance of the coding scheme, with FSK modulation and non-coherent demodulation. The results are presented in chapter 9.

7. Serial Binary Coding Schemes Applied to Track Circuits.

7.1 Introduction

The evolution of track circuits, from the single supply frequency or d.c. track circuit, to the modulated jointless track circuit has lent itself to the addition of data coding to provide the transfer of a limited amount of information from the track to the train. The track circuit frequency was originally moved to the audio frequency range because the greater attenuation of the transmission medium provided better sensitivity in the presence of a train shunt (although the maximum track circuit length decreases) and allowed Insulated Block Joints (IBJ) to be replaced by tuned terminations [23][24]. Also, the amplitude of power and chopper frequency harmonics decreases in the higher frequency range. Such track circuits are known as 'jointless' and the track circuit extremes are defined by short circuits rather than open circuits. Modulation (typically FSK) was added to the track circuit signal in the early 1980s to provide immunity from traction interference. With this increase in carrier frequency comes an increase in the potential channel capacity. Carrier modulation has been used to represent speed restriction commands to the train (rate coding), but the design of the train detection circuit has restricted the bandwidth of the signal and hence the number of commands that are available. With tuned track circuit terminations, the data rate is typically limited to approximately 100 baud. With the transceiver termination described in chapter 4, the potential data rate will be higher. The track circuit response time is typically limited to approximately 0.5 seconds.

The unusual nature of the track as a transmission medium greatly restricts the information capacity but a greater restriction on track circuit coding has been the downward compatibility requirement from proven track circuit designs [2]. The track circuit is, after all, a train detection circuit. The information requirement for track circuits now exceeds the capacity of rate coded signals [52] and data coding provides a means of increasing the amount of information that can be safely transmitted to the train.

The application of standard coding techniques [55][56][57] to track circuit applications is restricted because of the way that a track circuit operates and the harsh electrical environment [28][58]. For instance, consider the case when a train is entering a new block. The track-side transmitter must continually send information to the track-side receiver, which will indicate 'track unoccupied' if the signal level is above the threshold and a valid signal is received. If this signal consists of a binary code, it must continually be repeated by

the transmitter. As a train enters as new track circuit (or at switch on), the train borne receiver has no prior knowledge of block code synchronisation and may start receiving bits from any point in the code word. Since the communication from the track side transmitter to the train is one way, the train cannot indicate a loss of synchronisation to the transmitter. This also applies to the previous track side receiver after a train has left the track circuit, which must identify its code to prove that the track circuit is unoccupied. The receiver must be able to correctly decode the first complete code word received, which implies that it must distinguish between correctly framed words and misframed words for all code words and for all instances of code synchronisation loss. Codes of this sort are described as *comma free*.

The hostile electrical environment suggests that error correction or detection techniques should be applied. Error detection is essential and redundancy needs to be applied to the code to prevent a small number of errors changing, for example, a speed restriction code to a different value. Error correction has previously been applied with caution or not at all, in railway safety applications. In low data capacity binary data track circuit systems, error correction may seem unsuitable, since the code length is very short. However, with longer code lengths, a limited amount of error correction may greatly increase the reliability, without realistically compromising safety. The safety and reliability trade-off for the overall code performance corresponds to a trade-off between error correction/detection. These methods apply to correctly framed (synchronised) words. For the overall code performance to be assessed, the synchronisation capabilities of the code must also be considered.

The track circuit response time is limited and so a binary decoding scheme must be able to receive and decode the data within this time. Error control techniques increase the code redundancy and present a trade off between safety and reliability with data rate and response time.

There follows a brief description of the data coding schemes that have been previously applied to track circuits.

7.2 Track Circuit Coding

7.2.1 Gold Codes

A track circuit which uses pseudo random binary codes and correlation processing has been proposed [16]. These *Gold codes*, originally designed for spread spectrum communication, have sharp autocorrelation and low cross correlation properties [59]. The track circuit receiver does not need to determine block synchronisation (i.e. it doesn't need to know where the start and end of each code is) and determines the presence of a particular Gold

code by correlating the received code with a replica of the code assigned to the individual track circuit. The correlation processing in the receiver is simple compared to block code decoding. The error correction/error detection trade off is achieved by varying the value of the digital correlation threshold. There is however a penalty in redundancy. There are only 65 different codes, of length 126 bits, in this example. This corresponds to a code rate (k/n) of 0.05, where n is the total number of bits in the block and k is the number of information bits. As a train enters a track circuit, the maximum number of bits that must be received before a complete correctly framed code is 'seen' is $2n-1 = 251$. The modulation scheme used is non-coherent FSK at 126 bits/s, the signal being fed through jointless track circuit tuned terminations.

For the message decoding, the correlation algorithm must be performed separately for each Gold code. This increases the processor power and memory requirements. The number of messages is limited to 65 by the number of different Gold codes and cannot easily be expanded. Thus, this particular coding scheme is limited to only approximately 6 information bits.

The decoding scheme meets a target theoretical probability of false decoding (unsafe failure) of 4.10^{-15} when random data (Pe=0.5) is presented at the input to the decoder. No analysis is given for the performance of the code when the bit errors are non-randomly distributed.

Although this coding scheme can be shown to have theoretically low failure probabilities, the lack of an algebraic method of encoding or decoding and the limited number of Gold codes means that this technique would be unsuitable for higher data capacity coding.

7.2.2 6 Bit Comma Free Code

The San Francisco Bay Area Rapid Transit (BART) uses a 6-bit binary comma free code to transmit speed commands to the train [60][61][62]. The term *comma free* is a general description for codes in which no overlap or misframed code is a valid code in the dictionary. Hence, when the receiver sees a valid code in the data stream it assumes this is a correctly framed word and inserts a comma after every subsequent n bits in the data stream. The codes used in the BART system are shown below. The receiver required 2 consecutive codes to be received and so the effective code rate of the scheme (k/n) is 0.25.

n-bit comma free code dictionaries are constructed by simply removing all n-tuples that are cyclic shifts of overlapped code words. It can be seen that no overlap of two (not necessarily different) code words is another code word, and so, assuming bit synchronisation is achieved and there are no bit errors, each code can be uniquely identified. There is no

algebraic method of encoding or decoding and so error correction/detection cannot easily be applied [63][64] and the code length and data capacity of these codes is severely limited.

Speed (mph)	Code
80	101111
70	100111
50	101011
36	100011
27	100101
18	101001
6	100001
0	100000

Table 7.1

The speed decoding system suffered reliability problems [65] and less than a month after the first passenger service commenced, an accident involving a train overshooting a terminal led to an enquiry into the automatic train control (ATC) system. The cause of the accident was claimed to be a faulty crystal in a non-fail-safe circuit, but suggestions for improving the train control systems included the use of higher signalling power and error correcting codes [62]. It can be seen from table 7.1, that with the reception of 2 consecutive codes, a single bit error will cause a safe failure and two bit errors can cause an unsafe failure. This also applies to overlapped code words.

7.2.3 Cyclic BCH Codes

BCH codes have been proposed for track circuit applications [66]. BCH codes are a subset of cyclic codes and possess a highly algebraic structure and powerful error correcting properties [55]. Cyclic codes have no 'comma freedom' and hence the problem of block synchronisation still remains. A method of constructing and analysing comma free BCH codes is presented in the following chapter.

7.2.4 Specially Constructed Synchronisable Track Circuit Codes

The problem of achieving safe block synchronisation and increased data transmission has been studied in [2][67]. The proposed binary data message, modulated by non-coherent FSK, is split into three separate blocks: synchronisation, data and parity. The data coding is achieved using an extended (16,11) Hamming code. The overall message can correct one error or detect three errors. It has been shown [68] that no Hamming codes or cosets of Hamming codes can be made comma-free and so the synchronisation sequence must be

carefully chosen to have a maximum distance to all possible data sequences. This type of code is known as a comma code.

The synchronisation sequence is recognised by its peaky autocorrelation function. Two types of sequence commonly used are pseudo random sequences and Barker sequences [69]. This method of synchronisation can become unreliable when surrounded by random data, and so each combination of synchronisation sequence and data has been exhaustively analysed and code words that could cause unsafe failures in the presence of errors have been removed. This reduces the number of possible codes available. For instance, in the format proposed, there are 600 separate messages contained in a block length of 24 to 32 bits, which represents a coding efficiency of roughly 0.38 to 0.28. The code performance in terms of safe failure and unsafe failure in Gaussian noise has been studied [67] and the code is shown to have a peak unsafe synchronisation failure of approximately 10^{-2}, although at this point the safe failure probability is ten times greater.

Although the data capacity of this code (9.2 bits) is greater than that of the Gold code and comma free techniques, the decoding requires a dictionary check of all the possible code words, which increases the receiver hardware and complexity. Also, since the synchronisation and data code word sequences are separate, once the code is correctly synchronised, three bit errors in the data section of the message could could cause an unsafe failure.

7.3 Discussion

The evolution of track circuits to include data coding is a natural progression. Track to train communication is needed for advancing ATP/ATO functions and the existing track circuit and rail infrastructure provides a cheap and convenient transmission medium. For the purposes of train detection in a noisy environment, fail safe principles suggest that a complicated, rather than a simple signal should be used, with the receiver checking that specific properties of the signal are present as well as the overall signal level. Rate coding uses this principle where the receiver compares the FSK modulation phase and amplitude parameters against the valid signal thresholds. The data decoding simply consists of observing the modulation frequency. As a means of increasing the data capacity, a serial binary data stream seems an obvious method. Non coherent FSK modulation allows simple receiver and transmitter equipment to be used. Coding schemes have been proposed, and some implemented with varying success.

Assuming that hard symbol decisions are made at the receiver, the demodulation process involves no redundancy or error checking. Random interference of sufficient amplitude will

produce random binary data. This is not quite the case with rate coding, where the demodulation process determines whether a signal is rejected. If a serial binary modulation scheme is used, the binary coding/decoding schemes assume a lot of responsibility. They must provide all the redundancy and error detection to ensure that random or corrupted data cannot be falsely decoded, as well as maximise the data capacity. Since an increased data capacity implies a lower redundancy within a fixed response time and bandwidth the choice of an efficient coding scheme is important. Fulfilling these requirements and addressing the problem of safe and reliable code synchronisation is not a simple task.

The Gold code and comma free code schemes described in section 7.2.1 and 7.2.2 both lack a systematic encoding/decoding scheme, which increases the decoding time and receiver complexity. There is little or no improvement in data capacity over the existing rate coded signals, since the number of unique messages is limited by the nature of the code construction. The comma code described in section 7.2.4 is an attempt to increase the data capacity and uses systematic error control algorithms. However, since the synchronisation sequence has a low Hamming distance from certain overlapped code words, 'manual' dictionary checking must be incorporated into the receiver, reducing the data capacity to 600 messages. The message contains separate redundancy for synchronisation and error control and the data section of the message is incorporated into a single error correcting Hamming code. Only two bit errors in this section may cause an unsafe failure.

In the following chapters a method of synchronisable code construction is presented where robust synchronisation and error control is achievable without the need for special synchronisation sequences. A new method of analysing the synchronisation performance of these codes is presented and it is shown that codes with a high data capacity and high distances from overlapped words to valid code words can be constructed. The encoding/decoding algorithms are based on the well known cyclic/BCH codes which offer flexibility in terms of redundancy, code length and error correction/detection. A theoretical analysis and simulation of the code performance is presented in terms of safe and reliable code synchronisation with error correction, in a noisy environment.

8. Synchronisable Cyclic Coset Codes for Railway Track Circuits.

This chapter presents a new method for the construction and analysis of fully synchronisable binary codes suitable for track circuit data. The dual problems of error detection/correction and synchronisation are simultaneously addressed by cyclic coset codes. This method enables codes with a high index of comma freedom to be constructed without the use of special synchronisation sequences or 'start of message' indicators, and has been used in military and space systems such as the NASA Gemini and Apollo projects [70][71]. The technique requires fewer bits than is required for standard methods of message framing to achieve the same level of code performance, since the same redundancy is used for both error control and synchronisation. The method of analysis presented here introduces a new measure of the comma free properties of these codes, the *overlap weight distribution,* which provides a better criterion for the choice of synchronisable code than is given by the index of comma freedom alone.

Various codes and cosets are chosen for analysis of their synchronisation performance in Gaussian noise. This analysis allows the choice of code and amount of error correction to be optimised.

8.1 Synchronisable Codes for Track Circuits

In normal communication systems, frame synchronisation is usually concerned with correcting synchronisation slips within a certain distance of true synchronisation. A trial and error approach could be adopted, using the fact that sync slip is recognised by an unusually large number of errors. Track circuits are restricted by the fixed response time and synchronisation must be achieved on the reception of the first valid code word. Also, the code must be fully synchronisable since a train entering a new block has no prior knowledge of synchronisation. This implies that the codes must be comma free[1]. Let us define what we might want to look for in a synchronisable code:

The *synchronisation detection* capability of a code is related to the degree of comma freedom of the code. If a code has a degree of comma freedom $r < n/2$ (n = code length) then any synchronisation slip of up to r symbols will not be a code word and will therefore

[1] Comma freedom dictates that no overlap between any two code words (including the same code word) will contain a bit sequence that is the same as any code word.

be detected. If $r \geq n/2$ then any synchronisation slip can be detected and the code is said to be comma free.

The *synchronisation recovery* capability of a code is the ability for a synchronisation loss to be detected and correct synchronisation achieved without losing any code words. If a code has a degree of comma freedom $r < n/2$, then any sync loss of $r/2$ symbols can be corrected. It is necessary to try all alignments within $r/2$ symbols of the current (mis-framed) alignment. Only one of these alignments will be a valid code word, and hence synchronisation will be recovered. If the code is comma free, synchronisation can be recovered with any slip.

Synchronisation recovery applies to a semi-infinite sequence of data where, if sync is lost, the previous $r/2$ bits are available to the receiver. This would not be the case when, for instance, a train has entered a track circuit, and so *synchronisation detection* will be the important parameter.

The use of error detection/correction further complicates matters. These methods assume correct synchronisation at the receiver. For a comma free code, any overlapped/mis-framed word will generate a non zero-syndrome pattern. We must make sure that the syndromes of all possible overlaps are not the syndromes of our correctable error patterns. This means that we are able to distinguish between error patterns caused by additive errors and error patterns caused by mis-framing. We could also consider the situation of a mis-framed word with additive errors. Comma freedom states that no mis-framed word is also a code word, but does not say anything about the distance between a mis-framed word and a valid code word. For example, a comma free code with a minimum distance of 6 could have an overlapping minimum distance of 1. So, we could impose a condition stronger than comma freedom, in which any overlap differs from a code word by a certain number of positions. This is known as the *index of comma freedom.* Ideally the index of comma freedom (synchronisation detection) should be comparable with the minimum distance of the code itself.

8.1.1 Comma Free Codes

A finite code is called *synchronisable* if there exists an integer M, such that the knowledge of the last M letters of any message is enough to determine the separation of code words [73]. Comma-free codes form a subclass of mis-framed block codes for which M is $2n$-2, where n is the block length. Golomb, Gordon and Welch [74] have shown that the number of code words M, in a q-ary comma-free code cannot exceed

$$M \leq \frac{1}{n}\sum_{d} u(d) q^{n/d} \tag{8.1}$$

where the sum is over all divisors d of the code length n, and u is the Möbius function

$u(d) = 1 \quad : \quad d = 1$

$= 0 \quad : \quad d$ has any square factor

$= (-1)^r \quad : \quad d = p_1 p_2 \ldots\ldots p_r$ where the p_i are distinct primes.

Eastman [63] in 1965 proved that this bound is attainable for all odd values of n. Encoding and decoding of these codes is complex, compared to cyclic codes. An iterative algorithm for the construction of these codes has been developed by Sholtz [64]. The codes are constructed by removing all periodic words from the n^q possible code words, and then dividing the remaining words into equivalence classes (Non Periodic Cyclic Equivalence Classes) by cyclic shifting. Only one word for each equivalence class can be used as a code word.

These codes have a comma-free index of one, which makes them sensitive to additive errors. Comma-free codes of index greater than one have been investigated [74] but the results are of little practical use.

A subclass of comma-free codes are Path-Invariant Comma-Free codes. For word lengths of n symbols, these classes of codes can be synchronised after observing a maximum of L symbols [75].

Comma-free codes have no error control properties [76] and lack the algebraic properties which would enable efficient decoding. There is only one decoding algorithm, developed by Scholtz [64]. Comma-free codes are generally not employed today because of their complexity. Because of their sensitivity to errors and the absence of error correction, multiple transmissions would be necessary before a decision is made at the receiver [60].

8.1.2 Comma Free Cyclic Codes

Cyclic codes form an important subclass of linear block codes. They have considerable inherent algebraic structure and form efficient burst and random error correcting codes. To understand the construction and properties of cyclic codes, a knowledge of Galois field arithmetic is needed. Standard text books on Galois field algebra can be rather involved, and so a brief description of the algebra in relation to cyclic codes and construction techniques (in particular, BCH codes) is presented in Appendix B.

By their definition, cyclic codes possess no comma freedom, but their powerful error detection/correction capabilities make them more attractive than comma-free codes. Cyclic codes can be equipped with comma freedom by forming cosets. A fixed polynomial, $p(X)$ is added to each code vector before transmission:

$$t(X) = i(X)g(X) + p(X) \qquad (8.2)$$

where $t(X)$ is the transmitted code polynomial of degree less than n, $g(X)$ is the generator polynomial of degree n-k, and $i(X)$ has degree less than k. The receiver will subtract $p(X)$ from the received polynomial, and the additive error detection/correction capabilities are the same.

> **Theorem 8.1** For any coset code with $n \leq 2k$, the degree of comma freedom is bounded by $r < n - k - 1$

This implies that a coset code can only be comma free if $k \leq (n-1)/2$ (See chapter 12, [57]). It can be seen that this bound requires more redundancy than is necessary to assure comma freedom, as given by the bound in equation 8.1, so there is a price to be paid in redundancy for the advantages of using a cyclic coset code.

8.1.2.1 Comma Free Random Error Correcting Codes

These coset codes were proposed by Tong in 1968 [73]. Assume that $g(X)=g_1(X)p(X)$ generates an (n,k) cyclic code of minimum distance at least $2t+1$. $p(X)$ generates a cyclic code of minimum distance d_1, and $p(X)$ has at least one root of order n, i.e. $p(X)$ is a primitive polynomial of GF(2^n) where $n=2^m-1$. Now consider a coset code formed by $X^{-1}p(X)$. At the receiver, $X^{-1}p(X)$ is first subtracted from the received polynomial and the syndrome is calculated. This syndrome is associated with a particular error pattern, or *coset*. It can be shown that no coset corresponding to a sync gain or loss of r symbols is the same as the coset corresponding to t or fewer errors, where $|r| < d_1-t$, and $|r| < s/2$ (s = degree of $g_1(X)$).

So, under these limits, it is possible to distinguish uniquely between the non-zero syndrome caused by mis-framing and by additive errors. Now, for this code to be comma free we need $|r| \geq n/2$. Therefore $n/2 < s/2$. This states that n is smaller than the degree of $g_1(X)$, which is impossible since the degree of $g(X)$ is $(n-k)$ and must be greater than the degree of $g_1(X)$.

This shows that synchronisation *recovery* is not possible with comma free cyclic coset codes.

8.1.2.2 Comma Free Burst Error Correcting Codes

These codes, also proposed by Tong in 1968 [73], are constructed from cosets of cyclic codes capable of correcting all bursts of length b. This work is an extension of comma free cyclic codes proposed by Stiffler, 1965 [68], who states that any cyclic code can be made comma free if $k \leq (n-1)/2$, by complementing the first bit of each word, or the first and last bit of each word. This is the principle of the code: The coset is specified by $p(X)$. If $p(X) = X^{n-1}+1$ then the syndrome produced by a synchronisation slip of r bits is the syndrome produced by end around bursts of length $r+2$. It is assumed that, in a bursty channel, these

end around bursts are less likely than normal bursts and so their cosets (in the standard array/decoding table) can be used for sync *recovery.* Let us look at it in more detail:

The transmitted code word is: $t(X) = i(X)g(X) + p(X) = t_{n-1}X^{n-1} + t_{n-2}X^{n-2} + \cdots\cdots + t_0$

Now suppose that an r bit synchronisation loss occurs, with no additive errors. The received code word will be :

$$r(X) = X^r t(X) + d_1(X) - X^n d_2(X) \tag{8.3}$$

where :

$$X^r t(X) = t_{n-1}X^{n+r-1} + \cdots\cdots t_{n-r+1}X^n + t_{n-r}X^{n-1}\cdots\cdots + t_0 X^r$$

$$\longleftarrow \text{These terms subtracted by } d_2(X) \longrightarrow$$

$$d_1(X) = d_{r-1}X^{r-1} + d_{r-2}X^{r-2} + \cdots\cdots + d_0 \tag{8.4}$$

$$d_2(X) = t_{r-1}X^{r-1} + t_{r-2}X^{r-2} + \cdots\cdots + t_{n-r+1} \tag{8.5}$$

$$X^n d_2(X) = t_{r-1}X^{n+r-1} + t_{r-2}X^{n+r-2} + \cdots\cdots + t_{n-r+1}X^n \tag{8.6}$$

(Note that when cyclic codes are expressed in polynomial form, the algebra is modulo $X^n + 1$. Therefore, $X^n \equiv$ the remainder from dividing X^n by $X^n + 1 \equiv 1$). Now, since the coefficients of $d_1(X)$ and $d_2(X)$ are not known, equation 8.3 can be re-written as:

$$r(X) = X^r t(X) + d(X) \tag{8.7}$$

where $d(X)$ has degree r-1 or less and has unknown coefficients. The equations are similar for sync gain.

Now the first step at the receiver is to subtract $p(X)$. The resulting polynomial is :

$$r(X) - p(X) = (X^r - 1)p(X) + X^r i(X)g(X) + d(X) \tag{8.8}$$

The syndrome of this is calculated by dividing by $g(X)$ and taking the remainder (or in other words, the syndrome is the receiver polynomial, modulo $g(X)$). Since the middle term is a multiple of $g(X)$ the syndrome is (substituting $p(X) = X^n - 1 + 1$):

$$s(X) = \left[X^{r-1} - 1 + X^r - X^{n-1} + d(X)\right] \bmod g(X) \tag{8.9}$$

Remember, if the polynomial ground field is GF(2), then -1=+1. The syndrome is actually the result from dividing the error pattern (expressed as a polynomial) by $g(X)$ (this is because the received vector is the sum of the valid code word and the error pattern. The syndrome of the received vector will be the sum of the individual syndromes). Hence if the error pattern is a code word the syndrome will be zero. The term in square brackets in equation 8.9 is the error pattern. Noting that $d(X)$ is a polynomial of degree r-1 or less, we can see that this is the syndrome of an end-around-burst of length r+2. For example, say n=15, and r=2 (i.e., a sync loss of 2 symbols). The syndrome is the same as that which

would be caused by an error pattern of: [1000000000001xx], (x = don't know). It can be proved that the error pattern caused by a sync gain of two symbols is : [xx1000000000001].

Now if $b \geq r+2$ then this error pattern is a coset leader. Note that there are 4 possible combinations of error patterns that can be caused by a sync slip of two symbols. In fact there are 2^r possible error patterns that can be produced by a slip of r symbols.

8.1.2.3 Cyclic Comma-Free Coset Code Synchronisation Correction

Again we will show that, for comma free coset codes, it is impossible to uniquely identify all syndromes caused by synchronisation loss. The burst error correction capability of the cyclic code needs to be greater than or equal to $(r+2)$, where r is the synchronisation slip that is to be *corrected.* For the code to be comma free, $r \geq n/2$. Now the Reiger bound (theorem 4.15, [57]) states that the burst error correcting capability of a cyclic code is bounded by $b \leq (n-k)/2$. For the comma free case, $(n-k)/2 \geq n/2 + 2$, which is impossible for positive values of k.

8.1.2.4 Cyclic Comma-Free Coset Code Synchronisation Detection/Error Correction

As stated before, synchronisation correction is inappropriate for track circuits and comma freedom with a high minimum distance is more important. The burst error detection capability of an (n,k) cyclic code is $(n-k)$. For the coset code to *detect* all burst error patterns caused by mis-framing, $(n-k) \geq n/2 + 2$. If any error correction is to be used it is important to achieve as great a distance as possible between the correctable error patterns and the error patterns produced by mis-framing. The following example will look at the error patterns caused by mis-framing. Figure 8.1 below shows two consecutive words with all possible mis-framed words:

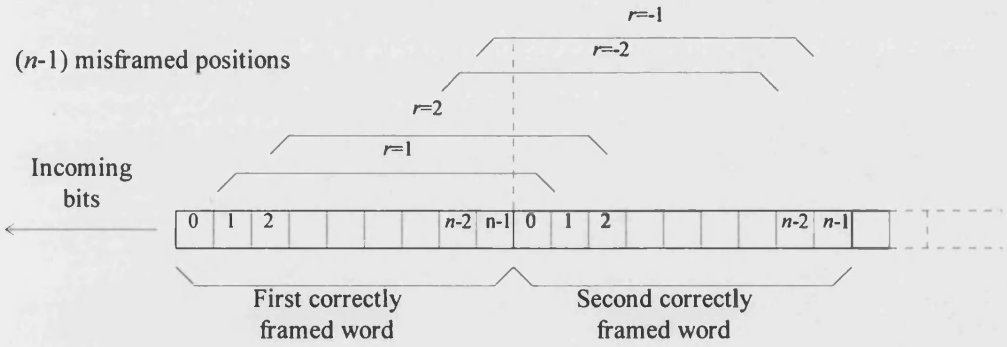


Figure 8.1. Mis-framing Between Correctly Framed Code Words

For example, say n=15 and $p(X)=X^{n-1}+1$. Using equation 8.9, the error patterns caused by these mis-framed words (with no additive errors) are :

[x10000000000001] $r = +1$

[xx1000000000001] $r = +2$

[xxx100000000001] $r = +3$

: : : : : :

[xxxxxx100000001] $r = +(n - 1)/2 = +7$

[10000001xxxxxxx] $r = -(n - 1)/2 = -7$

[100000001xxxxxx] $r = -6$

: : : : : :

[10000000000001x] $r = -1$

[000000000000000] $r = 0$

It can be seen that, for the coset code defined by $p(X)=X^{n-1}+1$, the minimum possible weight of error pattern is 2 (it will later be seen that a minimum weight of 3 is achievable for the (15,5) BCH code, by using different values of $p(X)$). This means that all error patterns with weight greater than one must be *detected* as either mis-framing or additive errors. All error patterns of weight one can be corrected. So, if a syndrome that corresponds to a single weight error pattern occurs, it is assumed that synchronisation has been achieved and a single error has occurred. This increase in reliability is accompanied by a decrease in safety, as it can be seen that a single additive error with a sync slip of one, could result in a decoding error. The probability of this decoding error occurring could be greatly reduced by, for instance, the receiver decoding two consecutive codes before making a decision. Also, sequencing in the train borne receiver could greatly reduce the probability of false decoding.

The comma free codes studied in this thesis will be cyclic coset codes. The principle of comma-free sync detection/error correction will be used. Cosets of any cyclic code can be investigated for their comma free properties.

Example:

Choose block length = $n = 15$. For comma freedom, with a coset code specified by $p(X) = X^{n-1} + 1$, we require that $n - k \geq n/2 + 2$. Lets say $(n - k) = 10$. Now all the end-around-burst error patterns that are caused by mis-framing are *detectable*.

We could choose the (15,5) BCH code, which has a minimum distance of 7. This is normally a triple error correcting code but in the comma free case, we are restricted to only correcting single errors.

So, the parameters for this code are: $n = 2^m - 1 = 2^4 - 1 = 15$; $m = 4$; number of parity digits = $(n-k) = 10$ and $t = 3$.

The generator polynomial is the LCM of the minimal polynomials of the elements α, α^3, and α^5, which are elements of the field GF(2^4). These minimal polynomials can be found in Appendix B of [55].

$$g(X) = LCM[(X^4 + X + 1)(X^4 + X^3 + X^2 + X + 1)(X^2 + X + 1)]$$

It can be seen that $g(X)$ has degree $(n-k) = 10$. Also see Appendix C of [55] for tables of BCH generator polynomials.

Also, different cosets (i.e. different values of $p(X)$) for each code will have different synchronisation properties. The aim of choosing a code and coset is to achieve error patterns with the greatest weight for overlapping words.

The restriction on the amount of error correction suggests that, within the system response time, a message split into smaller individual code blocks will have greater error correcting capabilities, and will achieve block synchronisation faster. For instance, consider a message length of up to 31 bits. A (31,11) BCH coset code could be used, capable of correcting a single error every 31 bits, and needing to receive a maximum of $2n - 1 = 61$ bits before synchronisation is achieved. There are 2048 possible code words. Alternatively, a message consisting of two code words from a (15,5) BCH coset code could be used. This would be capable of correcting a single error every 15 bits, and needing to receive a maximum of $2n - 1 = 29$ bits to achieve synchronisation. There are $2^5.2^5 = 1024$ possible combinations of two blocks, but this would be reduced because the *message* would also have to be self synchronising. Thus there is a trade off between error correction/sync speed and the number of serial code words used.

8.1.3 Higher Comma-Free Index Coset Codes.

8.1.3.1 Repetition Cyclic Codes

For continuous systems where synchronisation loss is only expected within a few positions of correct synchronisation, full comma freedom may not be required. The sensitivity of the code to loss of synchronisation can be characterised by a pair of numbers $[s, \delta]$ such that a synchronisation loss of s symbols or less will result in a bit pattern that differs from a valid code word by at least δ bits. For comma freedom, the aim is to specify a code with $s \leq \lfloor n/2 \rfloor$, where n is the code word length, and $\delta > 1$. We can represent each code word in polynomial form as:

$$t_i(X) = a_i(X)g(X) + p(X) = w_i(X) + p(X) \qquad (8.10)$$

where $a_i(X)$ = one of 2^k possible data polynomials and $t_i(X)$ is the corresponding code word. Again considering a slip of r symbols, the received code word is:

$$r(X) = X^r t_i(X) + d(X) = X^r w_i(X) + X^r p(X) + d(X) \qquad (8.11)$$

$$r = 1,2,3 \ldots . s$$

where $d(X)$ has degree $(r-1)$ or less. We will use $\alpha_k(X)$ to represent any polynomial of weight k and degree $< n$. This polynomial will represent the additive errors. For a code with

characteristic $[s,\delta]$ the distance between the mis-framed word of equation 8.11 and any valid code word must be at least δ for all values of r from 1 to s. So:

$$X^r t_i(X) + d(X) + t_j(X) + a_{(k-1)}(X) \neq 0 \qquad (8.12)$$

$$r = 1,2,3 \ldots s \quad k = 1,2,3 \ldots \delta$$

or

$$[X^r w_i(X) + w_j(X)] + p(X)[X^r + 1] + d(X) + a_{(k-1)}(X) \neq 0 \qquad (8.13)$$

$$r = 1,2,3 \ldots s \quad \delta = 1,2,3 \ldots \delta$$

where $w_j(X)$ represents any valid code word in the dictionary. Because $X^r w_i(X)$ is a multiple of a code word it is therefore a code word itself and because the sum of two code words is another code word:

$$[X^r w_i(X) + w_j(X)] = w_l(X) \qquad (8.14)$$

So we may write equation 8.14 as:

$$p(X)[X^r + 1] + d(X) + a_{(k-1)}(X) \neq w_l(X) \qquad (8.15)$$

$$r = 1,2,3 \ldots s \quad k = 1,2,3 \ldots \delta$$

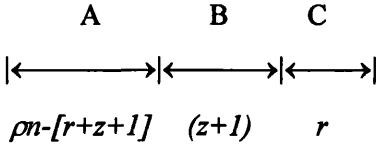
If this condition is satisfied, then the code will have characteristic $[s,\delta]$.

A ρ-repetition $(\rho n,k)$ cyclic code can be generated from an (n,k) cyclic code by repeating the generator polynomial ρ times, so it becomes:

$$g(X)[1 + X^n + X^{2n} + \cdots\cdots + X^{n(\rho-1)}] \qquad (8.16)$$

This will generate a $(\rho n,k)$ cyclic code. For each code word, any n consecutive digits are identical to the next n consecutive digits. If the original code has a minimum distance of d, then the repetition code will have a minimum distance of ρd.

For this class of code, let us picture the vector $p(X)[X^r + 1] + d(X)$ appearing in equation 8.15 as:



where z = degree of $p(X)$. Section A contains $\rho n-[r+z+1]$ zeros. Section B contains the $(z+1)$ places of $p(X)[X^r + 1]$ that are of degree $\geq r$. Section C represents the unknown polynomial $d(X)$. Now for the code to have a characteristic $[s,\delta]$, this vector must always differ from a valid code word by at least δ places. Due to the periodicity of a valid repetition

code word, section A must contain within itself a replica of the word fragment represented by section B, if there is any possibility of the vector forming a code word. If we specify that the weight of that part of $p(X)[X^r + 1]$ of degree $\geq t$ is at least δ, then this could be brought about, for instance, by the 'wiping out' (complementing) of δ 1's in section B. In fact, no error pattern of weight less than δ will change the vector $p(X)[X^r + 1]$ into a ρ - repetition code word. So, if the following conditions are satisfied:

$$\rho n - [s + z + 1] \geq n$$

$$\rho n - [s + z + 1] \geq z + 1$$

$$z + 1 \leq n$$

then the coset of the ρ - repetition cyclic code will have the characteristic $[s, \delta]$.

For the comma free case, we require that $s=<n/2>$. These codes have been investigated by Levy [77] and several comma free ρ - repetition codes are shown below (note : $N = \rho n$).

ρ	(N,k)	$[s,\delta]$	$r(X)$
3	(21,7)	[10,2]	$X^2 + X$
3	(45,15)	[22,4]	$X^6 + X^5 + X^4 + X$
4	(28,7)	[14,4]	$X^6 + X^5 + X^4 + X$
4	(60,15)	[30,8]	$X^{14}+X^{13}+X^{12}+X^{11}+X^{10}+X^7+X^6+X^4+X^2$

Table 8.1

8.2 Cyclic Coset Code Error Performance.

8.2.1 Codes for Track Circuits.

When cyclic codes are applied to track circuit coding there are many factors to consider. The code must be comma free and the index of comma freedom will affect the code performance when the receiver is searching for correct synchronisation. Track circuit performance is usually expressed is terms of MTBSF (Mean Time Between Safe Failure) and MTBUF (Mean Time between Unsafe Failure). The two parameters are a measure of safety and reliability. For the cyclic coset code, the safety and reliability trade off represents a trade off between error detection and error correction. Also, the decoding philosophy will provide a further trade off between safety and reliability. The number of bit errors corrected, t, must be less than the index of comma freedom, c. The difference between these two parameters must ideally be as high as possible to prevent additive errors causing false synchronisation for mis-framed/overlapped words. However, to create codes with higher indices of comma freedom a large amount of redundancy is needed, creating longer codes

for the same number of information bits. The limited response time of track, the particular decoding philosophy being used and the bit rate, set a limit on the code word length.

8.2.2 Error Probability Calculations

8.2.2.1 Synchronised Words

For a synchronised cyclic code, the weight distribution has to be known before accurate expressions for unsafe failure (UF) and safe failure (SF) can be derived. The weight distribution takes the form of a histogram, A, where A_i is the number of code words (not coset code words) with weight i. The weight distributions of dual and triple error correcting BCH codes have been determined [78]. For other codes, the weight distributions can be determined directly by computer program. Lets consider the case of no error correction. For a decoding error to be made, the error pattern must be the same as a valid code word. The probability of this occurring is:

$$P_{d.e.} = \sum_{i=1}^{n} A_i p^i (1-p)^{n-i} \tag{8.17}$$

where n = code word length, A_i = weight distribution, and p = bit error probability. For a t error correcting code the condition $d_{min} \geq 2t + 1$ must apply, where d_{min} is the minimum distance of the code. There are many error patterns of weight $(t+1)$ that are correctable, so an upper bound for decoding error is :

$$P_{d.e.} \leq \sum_{i=t+1}^{n} \binom{n}{i} p^i (1-p)^{n-i} \tag{8.18}$$

where $\binom{n}{i} = \frac{n!}{i!(n-i)!}$.

For track circuit applications, the safety priority, the comma free and code length restriction dictate that simultaneous error detection and correction must be employed. For example, consider the (15,5) BCH code, with d_{min}=7. If this code is used for single error correction, one random error will be corrected, 2 - 5 errors will be detected and 6 or more errors may or may not cause a decoding error. In this case the upper bound for the decoding error is :

$$P_{d.e.} \leq \sum_{i=6}^{n} \binom{n}{i} p^i (1-p)^{n-i} \tag{8.19}$$

To obtain accurate expressions for decoding errors with t error correction we need to modify equation 8.17 to include all code words that are a distance of t or less from valid code words. Let us first consider single error correction. Each code word of weight i is also an error pattern that will change a code word into a different code word. For this code word, there are $(n-i)$ error patterns that are weight $(i+1)$ and i error patterns that are weight $(i-1)$.

These error patterns need to be included in the expression for $P_{d.e.}$. Generalising for t error correction, for each code word of weight i, there are $\binom{n-i}{t}$ error patterns of weight $i+t$, and $\binom{i}{t}$ error patterns of weight $(i-t)$. The expression for $P_{d.e.}$ must be modified to include these terms:

$$P_{d.e.} = \sum_{i=2t+1}^{n} A_i p^i (1-p)^{n-i} + \sum_{j=1}^{t} \left[\sum_{i=2t+1}^{n} A_i \binom{n-i}{j} p^{i+j} (1-p)^{n-i-j} + \sum_{i=2t+1}^{n} A_i \binom{i}{j} p^{i-j} (1-p)^{n-i+j} \right] \tag{8.20}$$

($P_{u.f.} \equiv P_{d.e.}$ represents unsafe failure probability). Note that the bottom limit on the summation for the i index is set to $2t+1$. This is because, for a t error correction code, A_0 to A_{2t} are zero. Correct decoding occurs when t or less errors occur :

$$P_{c.d.} = \sum_{i=0}^{t} \binom{n}{i} p^i (1-p)^{n-i} \tag{8.21}$$

Error detection corresponds to safe failure (SF). For a code with minimum distance d_{min}, used to correct t random errors :

$$\begin{aligned} P_{s.f.} &\leq \sum_{i=1+t}^{(d_{min}-1-t)} \binom{n}{i} p^i (1-p)^{n-i} \\ &= 1 - (P_{c.d} + P_{u.f}) \end{aligned} \tag{8.22}$$

since $P_{s.f} + P_{u.f} + P_{c.d} = 1$.

8.2.2.2 Mis-framed Words

There is no known method for predicting the distribution of Hamming distances between valid code words and mis-framed words. The index of comma freedom, c, is the lowest possible distance of a mis-framed word to a valid code word, but there are only a few instances where this distance occurs. The majority of mis-framed words have a higher distance from valid code words.

A computer program has been developed to investigate the properties of specific coset codes [82][83]. The coset code is created, and from this all possible mis-framed words are generated. The difference vectors between these mis-framed words and the valid coset code words are calculated. From these vectors, an *overlap weight distribution* O_i can be formed. This is similar to the weight distribution of the correctly framed words. It contains the number of unique distance vectors from the valid coset code words to the overlapped coset code words, of weight i. The *index of comma freedom, c,* is the lowest weight. These distance vectors are actually the error patterns that will change an overlapped coset code word into a valid coset code, and cause a synchronisation error. If we divide the *overlap*

weight distribution by the total number of weights considered, we will get a probability distribution, which we will call the *overlap probability distribution,* p_{o_i}. This distribution gives the overall probability, for all values of mis-framing, of a mis-framed word being a certain distance from a valid code word. Since the *overlap probability distribution* is normalised, it can be used to compare the synchronisation properties of different cosets and codes (see section 8.4). Note that for track circuit coding, the receiver will have no prior knowledge of synchronisation when the train has left a track section, and so no distinction is made between different values of synchronisation slip. The overall probability for all values of synchronisation slip is calculated.

For mis-framed words, correct operation will occur when an error is detected. Failure will occur when bit errors change the mis-framed word into a valid code word or correctable code word. We will call the probability of decoding error for mis-framed words $P_{m.f.}$. For a particular code with no error correction :

$$P_{m.f.} = \sum_{i=1}^{n} O_i p^i (1-p)^{n-i} \tag{8.23}$$

where O is the overlap weight distribution. For t error correction the index of comma freedom, c, must be at least $t+1$, to ensure that the overlapped words are different from the correctable code words. The expression for $P_{m.f.}$ is extended to include all the error patterns that are a weight t or less from the patterns included in the overlap weight distribution.

$$P_{m.f.} = \sum_{i=c}^{n-c} O_i p^i (1-p)^{n-i} + \sum_{j=1}^{t} \left[\sum_{i=c}^{n-c} O_i \binom{n-i}{j} p^{i+j} (1-p)^{n-i-j} + \sum_{i=c}^{n-c} O_i \binom{i}{j} p^{i-j} (1-p)^{n-i+j} \right] \tag{8.24}$$

The probability of error detection for a mis-framed word (i.e. correct operation) will be called $P_{m.c.}$:

$$P_{m.c} = 1 - P_{m.f.} \tag{8.25}$$

8.3 Synchronisation Decoding Philosophy.

8.3.1 Choice of Philosophy

Before the synchronisation performance of the code can be assessed, the decoding philosophy must be specified. The decoder must identify valid blocks of n bit words. At switch on, or when a train is leaving a track circuit section, the receiver has no prior knowledge of synchronisation. It will receive the first n bits and store them in a buffer or *window*. These n bits may or may not represent a valid or correctable code word. The window will then slide as the next bit is received and the first bit is discarded. Once the receiver decides that synchronisation has been achieved it need only look at consecutive blocks of n bits. This is because, for cyclic coset codes the index of comma freedom is

typically less than the minimum distance of the code itself and so a decoding error is more likely when the receiver is making a decision on a mis-framed word than a correctly framed word[2]. The error correction may also be increased once synchronisation has been achieved. Possible decoding philosophies are :

1. On reception of the first valid code word the receiver assumes that synchronisation has been achieved.
2. The receiver must decode 2 consecutive valid code words, without recognising a valid code word in the (n-1) overlap positions in between.
3. The receiver must decode 2 consecutive valid code words, whilst recognising up to one valid code word in between. This code word is assumed to be an error.
4. Same as (2) but the two consecutive code words must be the same code word.

Method 1 is only suitable for low noise environments. Methods 2 and 3 demonstrate a trade off between safety and reliability. The following analysis applies to method 4 and is only appropriate to track circuit applications, due to the restriction that consecutive code words must be the same. The decoding philosophy can be combined with the error probabilities for correctly framed and mis-framed words to obtain expressions for the overall code performance. The following assumptions will simplify the expressions :

1. All bits are subject to independent errors.
2. It is assumed that each successive frame is subject to independent errors.

In the 'real world' the receiver stores n bits, with possible errors and then slides the window to receive the next bit. Each random error will therefore be stored in the receiver for n successive frames. Exhaustive simulations have been performed to check that the second assumption is valid. Error probabilities were calculated for mis-framed words subject to both independent errors for each frame, and to 'sliding window' errors for each frame. The overall probabilities were found to be the same, when the average for all overlap positions is taken, and the bit errors are independent. The overall performance of the code will be expressed in terms of correct operation (CO), safe failure (SF) and unsafe failure (UF). They are described below.

Correct Operation :- Decoding of a correctly framed word, no decoding of the subsequent (n - 1) overlapped words, and decoding of the next word.

---

[2] We will see later, however, that if the code is carefully chosen, the probability of a decoding error on a mis-framed word can be almost as low as the decoding errors for correctly framed word.

Unsafe Failure :- The corruption of a correctly framed word into another code word, followed by (n-1) non valid code words, and the corruption of the next correctly framed word into the same valid code word.

OR :- The decoding of a valid code word at an overlap position, followed by (n-1) non valid code words, then the decoding of the same valid code word at an overlap position.

Safe Failure :- Decoding error for a correctly framed code word, followed by anything other than (n-1) undecoded overlapped words and a decoding error for the next correctly framed word, into the same code word.

OR :- The reception of a valid code word at an overlap position followed by anything other than (n-1) undecoded words, then the same valid code word at an overlap position.

8.3.2 Decoding Probability Expressions

Mutually exclusive events can be identified as either correct operation (CO), safe failure (SF) or unsafe failure (UF). These events are listed on the following page. Note that the sum of the probabilities of each subsection will add to one. For example, P(1.1) +P(1.2) +P(1.3) = 1.

The following are used in the calculation of the exclusive event probabilities:

$(1 - P_{m.u.f.})^{n-1} = (P_{m.s.f.})^{n-1}$ = The prob. of no decoding errors for the (n-1) overlapped words between correctly framed words.

P_a Prob. of one or more decoding errors in (n-1) overlapped words
$= 1 - (P_{m.s.f.})^{n-1}$

O_i Overlap weight distribution of original code ($0 < i < n$)

$O1_i$ Overlap weight distribution with all non zero components replaced by 1.

P_b Prob. of an overlapped word being decoded into a particular code word (t = number of bits corrected)

$$= \sum_{i=c}^{n-c} O1_i p^i (1-p)^{n-i} + \sum_{j=1}^{t} \left[\sum_{i=c}^{n-c} O1_i \binom{n-i}{j} p^{i+j} (1-p)^{n-i-j} + \sum_{i=c}^{n-c} O1_i \binom{i}{j} p^{i-j} (1-p)^{n-i+j} \right]$$

A_i Weight distribution of original code ($0 < i < n$)

$A1_i$ Weight distribution with all non zero components replaced by 1.

P_c Prob.of framed word being wrongly decoded into a specific code word

$$= \sum_{i=2t+1}^{n} A1_i p^i (1-p)^{n-i} + \sum_{j=1}^{t} \left[\sum_{i=2t+1}^{n} A1_i \binom{n-i}{j} p^{i+j} (1-p)^{n-i-j} + \sum_{i=2t+1}^{n} A1_i \binom{i}{j} p^{i-j} (1-p)^{n-i+j} \right]$$

Exclusive Events

1. Correct decoding of 1st correctly framed word
    1.1 No decoding errors in overlap positions
        1.1.1 Correct decoding of 2nd correctly framed word ..... *CO*
        1.1.2 Error detected in 2nd correctly framed word ..... *SF*
        1.1.3 Decoding Error in 2nd correctly framed word..... *SF*
    1.2 One decoding error in overlap positions
        1.2.1 No decoding errors in (n-2) overlap positions
            1.2.1.1 Error detected in 2nd correctly framed word
                1.2.1.1.1 Decoding error in overlap position into same word as 1.2 ..... *UF*
                1.2.1.1.2 Any event other than (1.2.1.1.1) ..... *SF*
            1.2.1.2 Decoding error in 2nd correctly framed word..... *SF*
            1.2.1.3 Correct decoding in 2nd correctly framed word..... *SF*
        1.2.2 One or more decoding errors for the next (n-2) overlap positions ..... *SF*
    1.3 More than one decoding error in overlap positions..... *SF*
2. Error detected in 1st correctly framed word
    2.1 No decoding errors in overlap positions ..... *SF*
    2.2 One decoding error in overlap positions
        2.2.1 No decoding errors in (n-2) overlaps
            2.2.1.1 Error detected in 2nd correctly framed word
                2.2.1.1.1 Decoding error in overlap position into same word as 2.2..... *UF*
                2.2.1.1.2 Any event other than (1.2.1.1.2) ..... *SF*
            2.2.1.2 Decoding error in 2nd correctly framed word..... *SF*
            2.2.1.3 Correct decoding in 2nd correctly framed word..... *SF*
        2.2.2 One or more decoding errors for the next (n-2) overlap positions ..... *SF*
    2.3 More than one decoding error in overlap positions..... *SF*
3. Decoding error in first correctly framed word
    3.1 No decoding errors in overlap positions
        3.1.1 Decoding error in 2nd correctly framed into the same word as (3) ..... *UF*
        3.1.2 Any event other than (3.1.1) ..... *SF*
    3.2 One decoding error in overlap positions
        3.2.1 No decoding errors in (n-2) overlaps
            3.2.1.1 Error detected in 2nd correctly framed word
                3.2.1.1.1 Decoding error in overlap position into same word as 3.2..... *UF*
                3.2.1.1.2 Any event other than (3.2.1.1.1) ..... *SF*
            3.2.2 One or more decoding errors for the next (n-2) overlap positions ..... *SF*
    3.3 More than one decoding error in overlap positions..... *SF*

Note: P_b and P_c represent upper bounds because they take into account all code words of different weights, not just the particular code word that formed the original decoding error. This upper bound is valid, because they are used in the calculations of unsafe decoding error.

Exclusive Event Probabilities.

1.1.1 $P_{c.d.}(1-P_{m.u.f})^{n-1}(P_{c.d})$ ..... CO

1.1.2 $P_{c.d.}(1-P_{m.u.f})^{n-1}(P_{s.f})$ ..... SF

1.1.3 $P_{c.d.}(1-P_{m.u.f})^{n-1}(P_{u.f})$ ..... SF

1.2.1.1.1 $P_{c.d.}P_{m.u.f.}(1-P_{m.u.f})^{n-2}P_{s.f}P_b$ ..... UF

1.2.1.1.2 $P_{c.d.}P_{m.u.f.}(1-P_{m.u.f})^{n-2}P_{s.f}(1-P_b)$ ..... SF

(1.2.1.2)+(1.2.1.3) $P_{c.d.}P_{m.u.f.}(1-P_{m.u.f})^{n-2}(1-P_{s.f})$ .... SF

1.2.2 $P_{c.d.}P_{m.u.f.}[1-(P_{m.s.f})^{n-2}]$ ..... SF

1.3 $P_{c.d.}P_d$ ..... SF

2.1 $P_{s.f}(1-P_{m.u.f})^{n-1}$ ..... SF

2.2.1.1.1 $P_{s.f}P_{m.u.f.}(1-P_{m.u.f})^{n-2}P_{s.f}P_b$ ..... UF

2.2.1.1.2 $P_{s.f}P_{m.u.f.}(1-P_{m.u.f})^{n-2}P_{s.f}(1-P_b)$ ..... SF

(2.2.1.2)+(2.2.1.3) $P_{s.f}P_{m.u.f.}(1-P_{m.u.f})^{n-2}(1-P_{s.f})$ .... SF

2.2.2 $P_{s.f}P_{m.u.f.}[1-(P_{m.s.f})^{n-2}]$ ..... SF

2.3 $P_{s.f}P_d$ ..... SF

3.1.1 $P_{u.f.}(1-P_{m.u.f})^{n-1}(P_o)$ ..... UF

3.1.2 $P_{u.f.}(1-P_{m.u.f})^{n-1}(1-P_o)$ ..... SF

3.2.1.1.1 $P_{u.f}P_{m.u.f.}(1-P_{m.u.f})^{n-2}P_{s.f}P_b$ ..... UF

3.2.1.1.2 $P_{u.f}P_{m.u.f.}(1-P_{m.u.f})^{n-2}P_{s.f}(1-P_b)$ ..... SF

3.2.2 $P_{u.f.}P_{m.u.f.}[1-(P_{m.s.f})^{n-2}]$ ..... SF

3.3 $P_{u.f.}P_d$ ..... SF

8.4 Overlap Probability Distributions.

The previous section shows that, for a coset code specified by $p(X)=X^{n-1}+1$, a mis-framed word could be a Hamming distance of 2 from a valid code word, thus restricting the comma free code to single error correction. However, the probability of an overlap occurring which is distance 2 from a valid code word is low, because there are only a small number of mis-framed words that are actually this distance from a valid code word. There is no known method of predicting the distribution of weights from all code overlaps to the valid code words. The index of comma freedom is the lowest possible distance of a mis-framed word to a valid code word, but there are only a few instances where this distance occurs. The

majority of mis-framed words have a higher distance from valid code words. The *overlap weight distributions, O_i,* and *overlap probability distributions, p_{o_i}* that follow have been generated by computer program. The overlap probability distribution gives the overall probability, for all values of mis-framing, of a mis-framed word being a certain distance from a valid code word. Note that for track circuit coding the receiver will have no prior knowledge of synchronisation when the train has left a track section, and so no distinction is made between different values of synchronisation slip. The overall probability for all values of synchronisation slip is calculated. For cyclic codes the obvious choice is the BCH class of code, for their powerful error detection/correction capabilities. The computer search time is great and so the calculation of p_{o_i} has been restricted to smaller BCH codes (up to 31 bits).

8.4.1 BCH Coset Codes

Shown below are the overlap probability distributions for different cosets of the (15,5) BCH code, which satisfies the condition $(n\text{-}k) \geq n/2 + 2$.

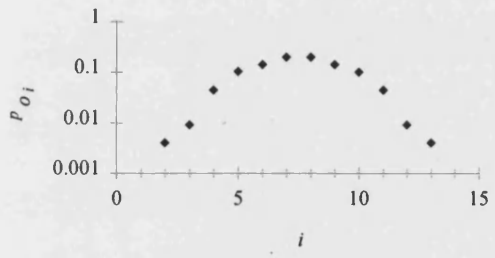


Figure 8.2. (15,5) BCH Code, $p(X)=X^{14}+1$

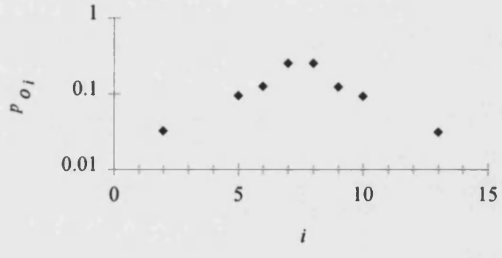


Figure 8.3. (15,5) BCH Code, $p(X)=X^{14}$

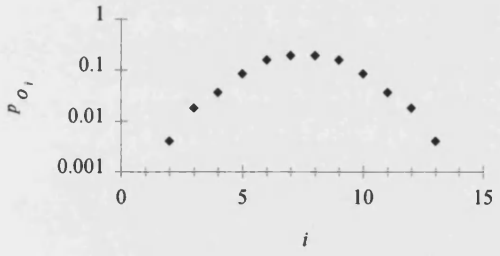


Figure 8.4. (15,5) BCH Code, $p(X)=X^{14}+X^{13}+1$

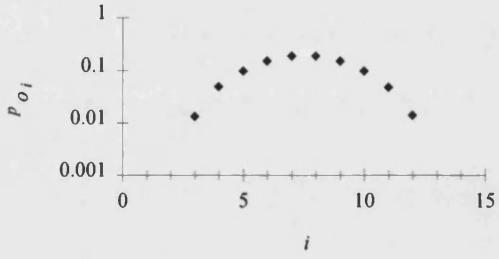


Figure 8.5. (15,5) BCH Code, $p(X)=X^{14}+X^{2}+1$

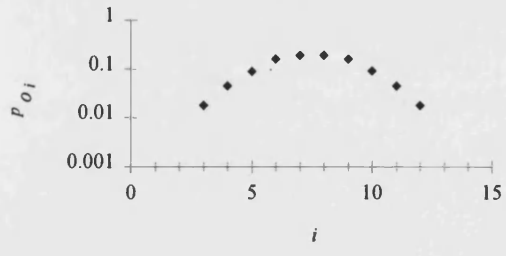


Figure 8.6. (15,5) BCH Code, $p(X)=X^{14}+X^{12}+X^{2}+1$

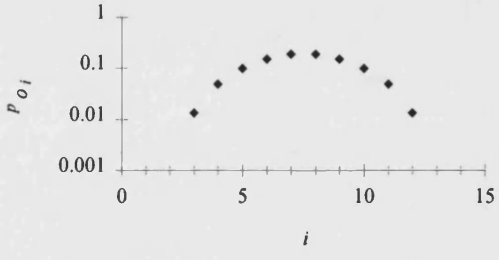


Figure 8.7.(15,5) BCH Code, $p(X)=X^{4}+X^{3}+X^{2}+1$

A steep curve is desirable, giving a low probability that a small number of bit errors in a mis-framed word will cause a decoding error. Note that figures 8.5, 8.6 and 8.7 show a comma free index of 3 enabling double error correction. *p(X)* in figure 8.7 is a Barker sequence [69] and this coset has the best comma free properties out of those shown above.

Cosets of the (15,7) BCH code were also studied. The best coset code is given by $p(X)=X^{14}+1$ and has a comma free index of 2, as shown in figure 8.8.

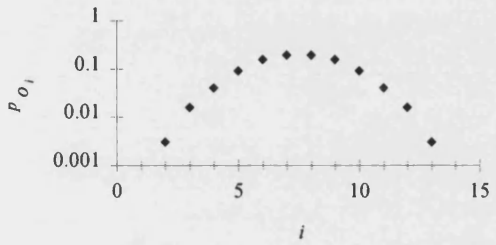


Figure 8.8: (15,7) BCH Code, $p(X)=X^{14}+1$

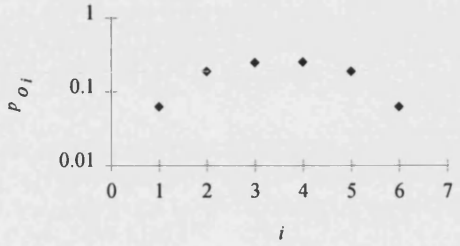


Figure 8.9: (7,4) BCH Code, $p(X)=X^6+1$

This demonstrates that $n-k \geq n/2+2$ is a sufficient but not necessary condition for a comma free index of 2. The (7,4) code has a minimum distance of 3 and is single error correcting. No coset was found to have an index of comma freedom greater than 1, and so only error detection can be used. The overlap probability for the (7,4) BCH code with coset $p(X)=X^6+1$ is shown in figure 8.9.

The largest BCH code studied was the (31,11) code which has 2047 separate code words. When correctly synchronised this code has a minimum distance of 11 and can correct up to 5 random errors. Many cosets were investigated. The coset defined by the Barker sequence $p(X)=X^{10}+X^9+X^8+X^4+X$ was found to give to best comma free properties. It has a comma free index of 5 and very low probabilities at the extremes of the overlap probability distribution.

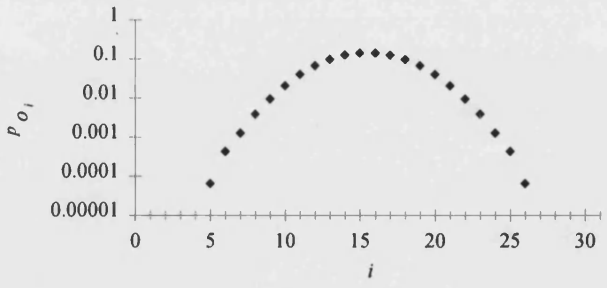


Figure 8.10. (31,11) BCH Code, $p(X)=X^{10}+X^9+X^8+X^4+X$

8.4.2 Shortened Cyclic Codes

Some very efficient cyclic codes and shortened cyclic codes for correcting short single bursts have been found either analytically of with the aid of a computer search (p269 [55],

[79][80]). The (21,9) code is capable of correcting single burst errors up to length 6 when correctly synchronised. The overlap probability distribution when the coset is defined by the Barker sequence $p(X)=X^6+X^5+X^4+X$ is shown below. Many different cosets were investigated and this was found to produce the steepest distribution. The index of comma freedom is 3, allowing up to double error correction..

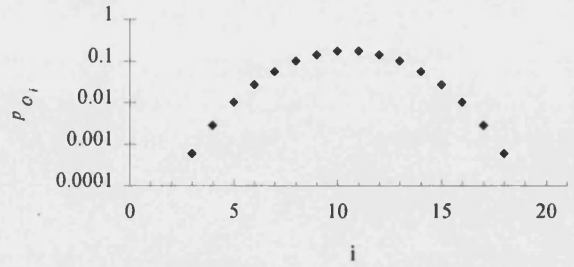


Figure 8.11. (21,9) Cyclic Code, $p(x)= X^6+X^5+X^4+X$.

8.4.3 Repetition Codes

Repetition codes were introduced by Levy [77] for generating cyclic codes with higher indices of comma freedom. A ρ-repetition $(\rho n,k)$ cyclic code can be generated from an (n,k) cyclic code by repeating the generator polynomial ρ times. The (28,7) code is a four fold repetition of the trivial (7,7) code. The overlap probability for the (28,7) coset code defined by $p(X)=X^6+X^5+X^4+X$ is given below.

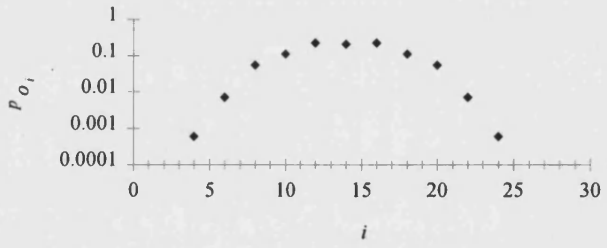


Figure 8.12. (28,7) Repetition Code, $p(X)=X^6+X^5+X^4+X$

8.5 Overall Code Performance.

The simplest decoding philosophy would be for the receiver to make a decision on the first valid code received. This however would result in unsafe failure at high bit error rates. For example, the failure probabilities for the (15,5) BCH coset code with $p(X)=X^4+X^3+X^2+1$ and single error correction are shown in figures 8.13 and 8.14. Note that, for correctly framed words, *unsafe failure* represents a decoding error, and *safe failure* represents an error detection.

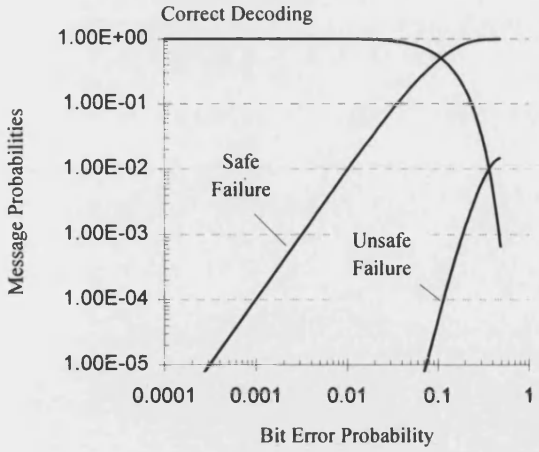


Figure 8.13. Failure Probabilities for Correctly Framed (15,5) BCH Code

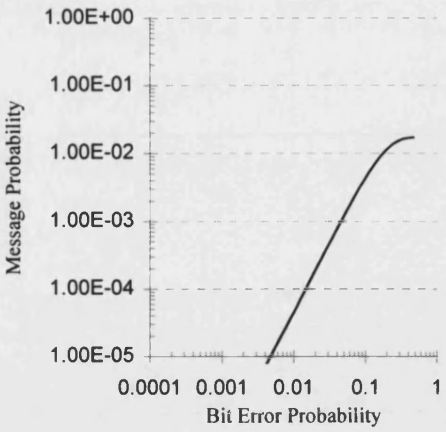


Figure 8.14. Failure Probabilities for Mis-framed (15,5) BCH Code.

For correctly synchronised words, safe failure is always more probable than unsafe failure. For mis-framed words, the failure probability is an unsafe failure probability, since it represents a mis-framed word being changed into a valid or correctable code word. For random data ($P_e = 0.5$) there would be unsafe errors approximately once every 100 frames, for both correctly framed and mis-framed words. To reduce this unsafe failure probability, the decoding philosophy from section 8.3.1 is used for the results presented in this section. Here, the receiver must decode two consecutive *same* code words before a decision is made. The synchronisation performance of the coset codes described in section 8.4 has been evaluated and the results are presented below.

8.5.1 (15,5) BCH Code

The coset is defined by $p(X)=X^4+X^3+X^2+1$. The Index of Comma Freedom is 3 which enables up to double error correction. Figures 8.15, 8.16 and 8.17 demonstrate the effects of error correction on the synchronisation performance of this code. The graphs correspond to no, single and double error correction, respectively.

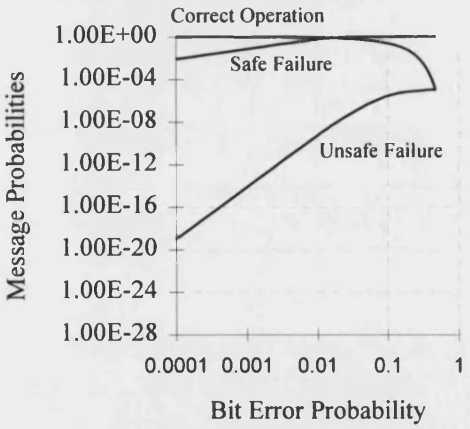


Figure 8.15. (15,5) BCH Coset Code with No Error Correction

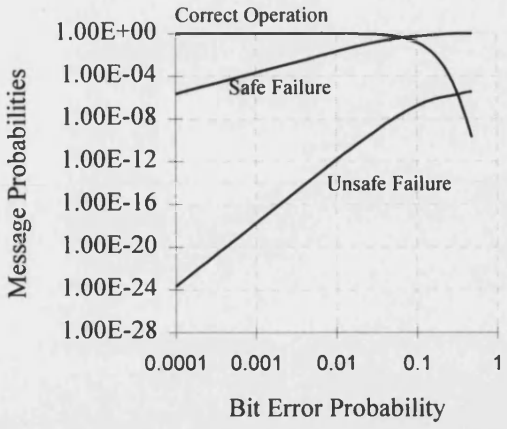


Figure 8.16. (15,5) BCH Code with Single Error Correction

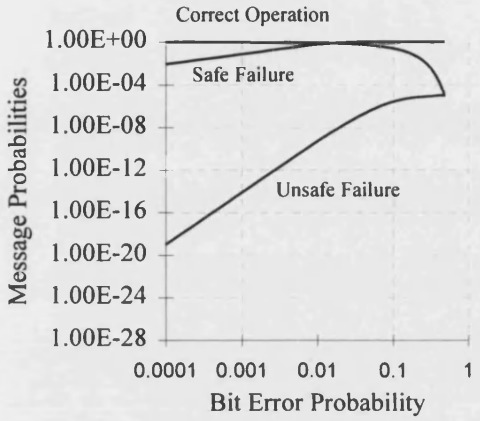


Figure 8.17. (15,5) BCH Code with Double Error Correction

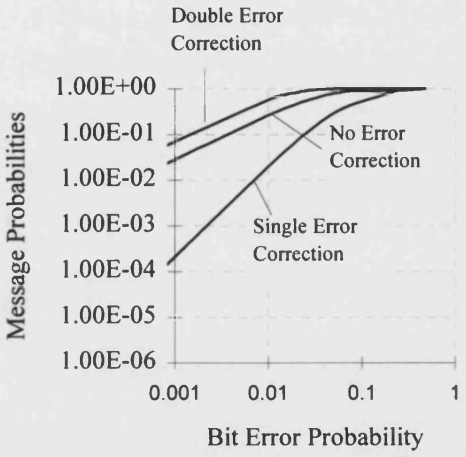


Figure 8.18. Safe Failure Curves for (15,5) BCH Code.

Thesafe failure curves of figures 8.15, 8.16 and 8.17 are plotted in figure 8.18. This clearly shows that the optimum reliability is obtained with single error correction. The unsafe failure curves are shown below. It can be seen that there is a safety penalty for increasing the reliability, although the peak unsafe failure rates are still extremely low when compared to the safe failure rate.

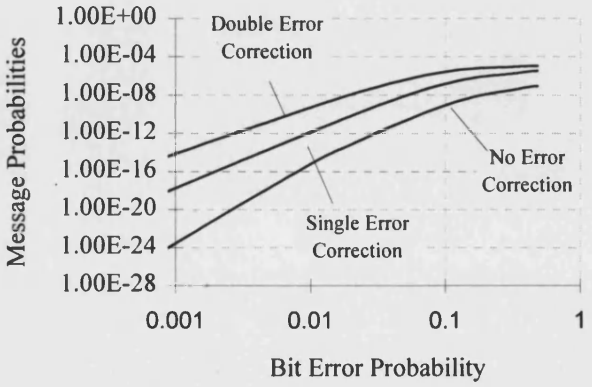


Figure 8.19. Unsafe Failure Curves for (15,5) BCH Code.

8.5.2 (15,7) BCH Code

The coset is defined by $p(X)=X^{14}+1$. The index of comma freedom is 2, so this code is synchronisable with single error correction. Figures 8.20 and 8.21 show the synchronisation performance for no, and single error correction, respectively.

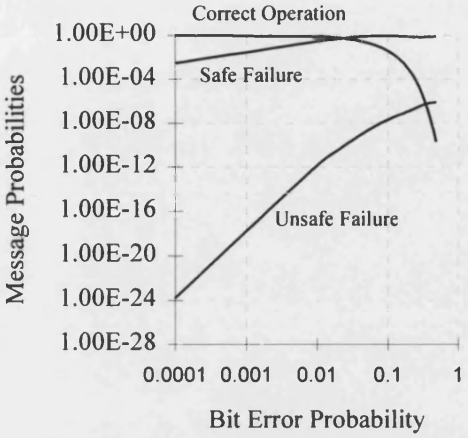


Figure 8.20. (15,7) BCH Code with No Error Correction

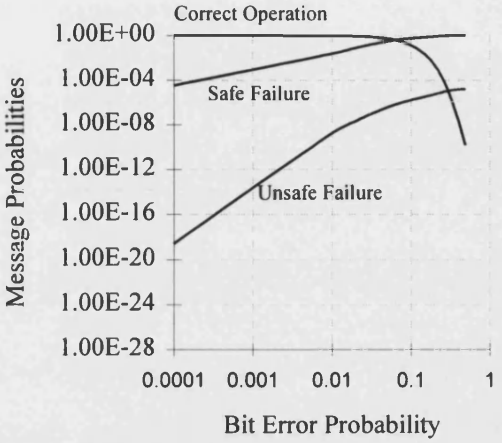


Figure 8.21 (15,7) BCH Code with Single Error Correction

It can be seen that the peak unsafe failure rates for single and no error correction are comparable. Single error correction has improved reliability since the safe failure curve falls more rapidly with decreasing bit error probability.

8.5.3 (7,4) BCH Code

The coset is defined by $p(X)=X^{6}+1$. The index of comma freedom is 1, and so no error correction can be used, whilst searching for synchronisation.

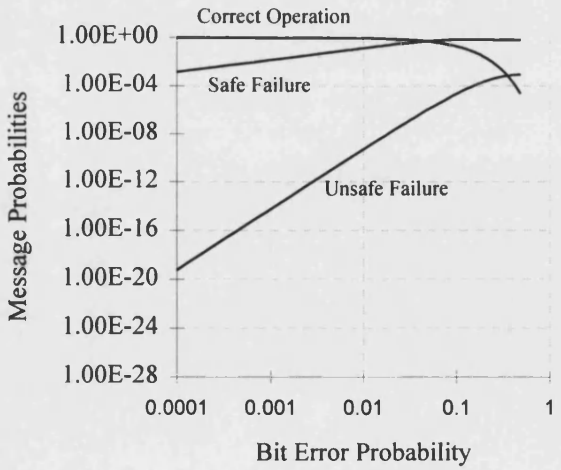


Figure 8.22: (7,4) BCH Code with No Error Correction

8.5.4 (28,7) Repetition Code

The coset is defined by $p(X)=X^6+X^5+X^4+X$ and the index of comma freedom is 4. The minimum distance of the correctly framed words is also 4, so the code can only correct a single error.

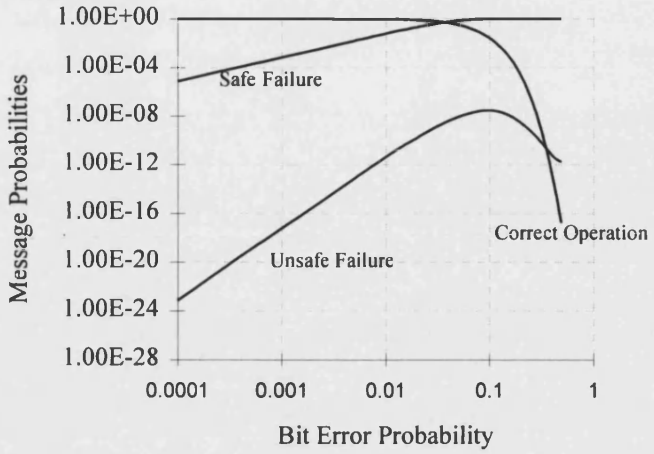


Figure 8.23: (28,7) Repetition Code with Single Error Correction

8.5.5 (21,9) Shortened Cyclic Code

The coset is defined by $p(X)=X^6+X^5+X^4+X$. The index of comma freedom is 3, and the minimum distance of the correctly framed words is 6. The optimum performance is obtained with single error correction. Figures 8.24 and 8.25 show the synchronisation performance with no error correction and single error correction.

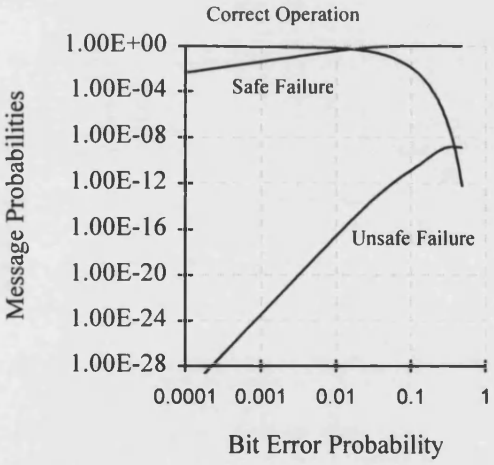


Figure 8.24: (21,9) Shortened Cyclic Code with No Error Correction

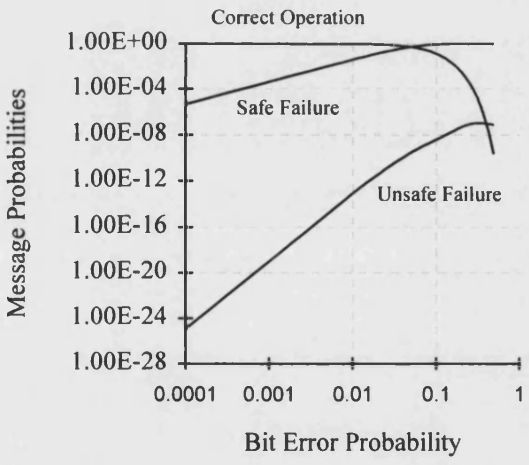


Figure 8.25. (21,9) Shortened Cyclic Code with Single Error Correction

8.5.6 (31,11) BCH Code

The (31,11) BCH code is the largest code studied, with the highest index of comma freedom. The coset is defined by $p(X)=X^{10}+X^9+X^8+X^4+X$. The index of comma freedom is 5 so in theory, up to 4 random errors can be corrected whilst searching for synchronisation. The optimum performance, shown below, is obtained with double error correction.

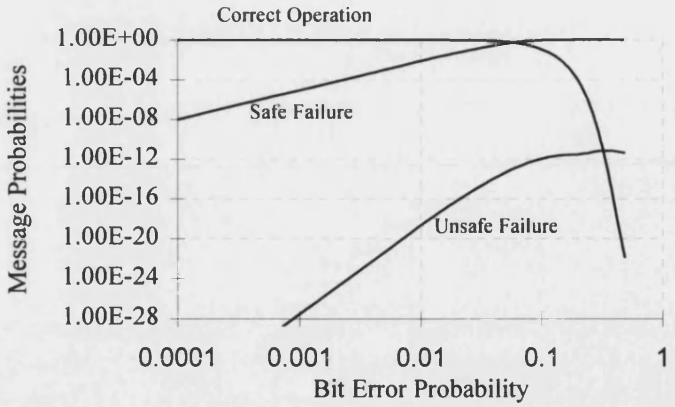


Figure 8.26. (31,11) BCH Code with double error correction.

The safe failure and unsafe failure rates for different values of error correction are shown in figures 8.27 and 8.28.

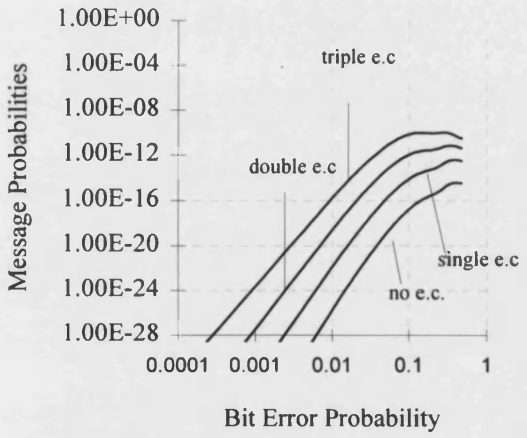


Figure 8.27. Unsafe Failure Probabilities for the (31,11) BCH Code

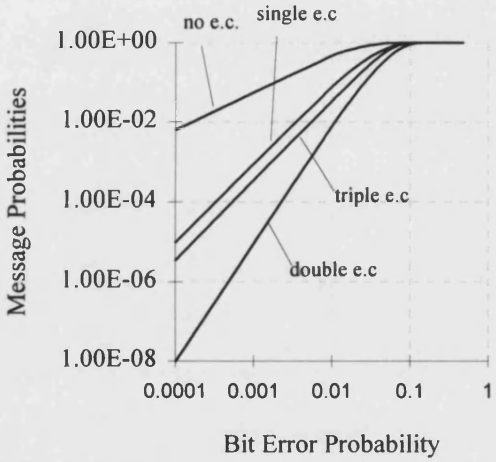


Figure 8.28. Safe Failure Probabilities for the (31,11) BCH Code

8.6 Discussion

For track circuit applications, the maximum response time is very important and is related to the code length. With a receiver decoding philosophy that requires the reception of two valid code words, the maximum response time (in the absence of errors) is $(3n - 1)T_b$, where n is the code word length and T_b is the bit length is seconds. To achieve higher indices of comma freedom than that of the codes presented, the redundancy needed may make the code length impractical for track circuit applications. For the (31,11) code the maximum number of bits needed for the reception of two valid code words is 92. For the (15,7) code it is 44 bits.

The (15,5) code from section 8.5.1 has an index of comma freedom of 3 which allows correct operation with double error correction, in the absence of errors. However, figure 8.18 shows that the use of double error correction has severely affected the reliability of the code. This is because, for a mis-framed word, a single bit error can produce a correctable code word. This usually will result in a safe failure. Notice also the decrease in reliability of the (15,5) code with no error correction (figures 8.15 and 8.18). The peak unsafe failure probability with no error correction is slightly lower than that obtained with single error correction.

The (15,7) code has the advantage of 127 code words, compared to only 32 codes for the (15,5) code. The (15,7) code has a lower index of comma freedom. and so the safe failure rate of the (15,7) code, shown in figure 8.21, is higher than the (15,5) code while the unsafe failure rates are comparable at high bit error rates. This is because the *overlap probability* is very low for low distances. The peak unsafe failure probability is approximately 10^{-6} for the (15,5) code and 10^{-5} for the (15,7) code, with single error correction. At low bit error rates the unsafe failure probabilities of the (15,7) code are higher than the (15,5) code, due to the lower distance between valid code words. Figure 8.20 shows the (15,7) code with no error correction. With random data (P_e=0.5) the unsafe failure rate has decreased by a factor of ten, compared to single error correction, although there is a significant decrease in reliability.

For the (31,11) BCH code, the optimum performance is obtained with double error correction. The coset used for figure 8.26 has a high index of comma freedom (5) and a steep overlap probability distribution. The data rate (k/n) is approximately the same as the (15,5) code but the performance is considerably better, and the number of code words is high (2047). Note that there is a slight kink in the unsafe failure curve at high bit error rates. A possible explanation of this is found by considering the minimum distances between mis-framed words and correctly framed words. The minimum distance between mis-framed words and correctly framed words is 5. The minimum distance between correctly framed words is 11. The probability of receiving i errors in n bits, with a bit error probability of p, is $\binom{n}{i} p^i (1-p)^{n-1}$. The plot below gives the probability of receiving 5 and 11 bit errors in 31 bits, for different bit error probabilities.

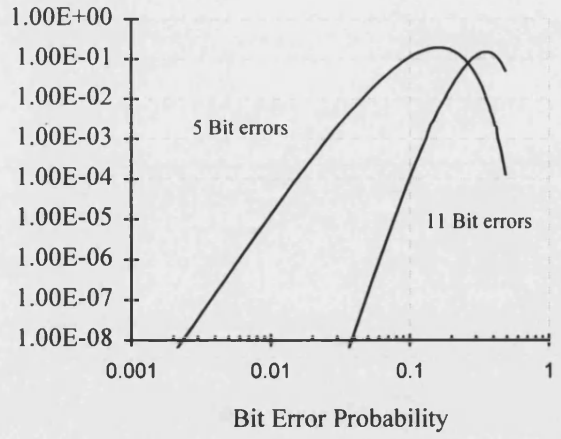


Figure 8.29. Probability of 5 and 11 bit errors in 31 bit word

The two peaks occur at the same bit error probability as the peaks in the unsafe failure curve of figure 8.26. The curve rises at $P_e \approx 0.3$, since this is where a decoding error for correctly framed words is most likely.

The performance of the (28,7) repetition code is almost as good as the (15,5) code. This code has an extra index of comma freedom, but high redundancy is needed to achieve this. The minimum distance is 4 which restricts it to single error correction. For just correctly synchronised words at low bit error probabilities this code will not be as reliable as the (15,7) code, because multiple errors are more likely to occur in 28 bits than in 15 bits. Also, the minimum distance of the (15,7) code is one greater than the (28,7) code, and so when correctly synchronised the (15,7) will have a lower probability of decoding error. Note that the unsafe failure curve peaks at $Pe \approx 0.1$, then decreases. This is due to the shape of the overlap probability distribution, which is zero for odd values of i.

The (7,4) BCH coset code has an index of comma freedom of one and so no error correction can be used. This code will have a faster response time as a maximum of only 20 bits are needed to receive two correctly framed words. However, the safety, reliability and data capacity of this code is much less than the other codes considered.

The (21,9) shortened cyclic coset code was constructed for single burst error correction and is capable of correcting bursts of length 6, when correctly synchronised. The data rate of this code is comparable to the (15,7) BCH code, but the performance is considerably better since this code has an index of comma freedom of 3 and a minimum distance of 6. This code also has 511 code words, compared to only 127 for the (15,7) code.

The advantages of using coset codes for track circuit coding can be seen when these results are compared with those presented in [2], where a comma code was constructed and certain

code words removed from the dictionary in order to improve the performance. The code length is 31 bits and there are 600 messages, corresponding to 9.22 information bits. For coded track circuits, the information rate can be expressed as the (number of information bits)/(maximum number of bits to be received before decoding can commence). For the code presented in [2], the synchronisation sequence is 13 bits long. Correct decoding requires detection of this sequence and no synchronisation occurring in the 12 subsequent overlapped sync/data sequences. Hence the information rate for this code is 0.167. For the (15,7) coset code, this figure is 0.16, and so these two codes have comparable information rates. Table 8.2 below shows the synchronisation failure probabilities for the comma code and the (15,7) coset code with single error correction. At high bit error rates, the coset code has a considerably lower unsafe failure probability. Also, the (15,7) code, when correctly synchronised, has a minimum distance of 5, compared to a minimum distance of 3 for the comma code.

	Comma code (see[2])		(15,7) coset code, single error correction.	
P_e	P_{SF}	P_{UF}	P_{SF}	P_{UF}
0.5	0.9	0.1	≈ 1	10^{-5}
0.1	0.4	0.05	0.9	10^{-6}
0.01	0.01	10^{-3}	0.02	10^{-9}

Table 8.2. Failure probabilities of Comma Code and Coset Code.

8.7 Conclusions.

This chapter has presented a method of construction and analysis of synchronisable binary codes suitable for track circuit data. The codes are easy to construct since the encoding algorithm of the coset code is the same as that of the original cyclic code. Many cosets of BCH codes are studied since these are among the most powerful codes known and have well know encoding/decoding algorithms. A high index of comma freedom is achievable without the use of special synchronisation sequences. Since the original codes have powerful error control properties, the synchronisation performance is superior to that of comma or comma free codes.

The *overlap probability distribution* has been introduced to provide a better measure of the self synchronising capabilities of cyclic coset codes than is given by the index of comma freedom alone. It has been seen that a narrow distribution is desirable, giving a low probability that a small amount of bit errors will cause a decoding error in the overlap region between two correctly framed words. However, for the codes considered, the

probability of decoding error is high at high bit error probabilities. A receiver decision based on the reception of two valid code words provides the extra error detection required to produce safe failures at low bit error probabilities. A further restriction, suitable for track circuit coding, requires that consecutive code words be the same. This will further reduce the unsafe error probability. After the receiver decides that synchronisation has been achieved, it need only test consecutive (non-overlapping) words. Because the minimum distance of the correctly framed words is less than the index of comma freedom, the performance of the code will be better than shown in section 8.5.

The (31,11) code has the same data rate as the (15,5) code, but the performance is considerably better. This would suggest that, as long as a good coset is found, longer codes of the same data rate will have a better synchronisation performance. However, the code length is limited because of the limited response time and bandwidth of the track circuit, and so this code is the longest code studied. Also, the computer response time in generating the overlap probability distribution and the synchronisation performance would be prohibitive for longer codes. Of the codes studied in this chapter, the (31,11) BCH coset code offers the best performance in terms of failure rates and data capacity.

For each code, many cosets were investigated for their synchronisation properties. It is interesting to note that in many cases the best cosets were defined by Barker sequences [69]. These were originally designed as synchronisation sequences to prefix data blocks.

The analysis of the synchronisation performance of these codes has been achieved through analytical probability expressions. This allows very low failure rates to be compared and the codes to be optimised. Independent bit errors occur when the signal is disturbed by additive white Gaussian noise (AWGN). For a particular modulation scheme, the bit error rate (P_e) can be related to the SNR at the receiver [81]. However, the noise encountered in railway systems is likely to be non-Gaussian, and so the assumption of independent bit errors may be invalid. Using the interference model described in chapter 6, the following chapter will present results of statistical simulations of the (31,11) BCH coset code and the (21,9) shortened cyclic coset code performance with non-coherent FSK disturbed by this interference. These results will be compared to those obtained in this chapter, to analyse how the performance is affected by non-Gaussian noise.

9. Cyclic Coset Code Performance with Non-Coherent FSK and non-Gaussian Noise.

9.1 Introduction

The synchronisation performance of comma free cyclic coset codes has been studied in chapter 8. A decoding philosophy has been proposed which results in safe operation when the signal is disturbed by additive white Gaussian noise (AWGN). Other proposed schemes [2][16][66] have also been studied in AWGN, which is useful, but not realistic, since the noise encountered in railway systems is likely to be impulsive and bursty.

This is an attempt to evaluate the performance of the cyclic coset codes, with interference other than AWGN. If the noise bursts were long (i.e. they would always cover several code words), then we could say that the burst state is equivalent to an independent bit error rate of 0.5 (random data) and a study of the codes with AWGN at low SNR may be sufficient. However, since the noise is impulsive, each 'burst' may affect only a few bits, depending on the channel and receiver structure, and so a more detailed study is required.

With AWGN the coding performance deteriorates smoothly as the SNR reduces. With burst or impulsive noise, this may not be the case, depending on the coding scheme. If the correct scheme is chosen to suit the expected noise, then little performance degradation can be achieved over a wide SNR range. This chapter presents the performance of the (31,11) BCH coset code and the (21,9) shortened cyclic coset code, based on the impulsive interference model of chapter 6. The decoding philosophy that requires two consecutive code words to be received before a decision is made, produces extremely low unsafe failure probabilities. Since is it not possible to obtain statistical figures for these low probabilities, the simulation is performed separately for correctly framed words and for mis-framed words.

The simulation and interference model is described in section 9.4. The results are presented in section 9.5. To verify the theoretical equations for code performance with independent errors (see section 8.2), results are first presented for Gaussian noise in section 9.5.1. These results are then compared to those obtained with the impulsive interference model in section 9.5.2. Separate results are shown for correctly framed words and mis-framed words, for both the (21,9) shortened cyclic coset code and the (31,11) BCH coset code. The comments and conclusions are presented in sections 9.6 and 9.8.

9.2 Objectives of Simulation

The performance of several cyclic coset codes has been evaluated in AWGN and a decoding philosophy has been defined which gives very low failure rates. It is known that BCH codes provide good immunity to burst noise (an (n,k) code is capable of detecting all bursts of up to length $(n-k)$ ([89] p53)). It would be useful to see how the synchronisation performance of the proposed codes is affected by the bit error patterns caused by the type of interference encountered in railway systems. This involves simulating an FSK signal, defining a suitable noise model and demodulator design, and applying these to the coding scheme. The results will be compared to those obtained in chapter 8.

9.3 Binary Frequency Shift Keying

In previous track circuit coding systems using an FSK binary stream, the equipment and signal parameters are chosen for convenience and cost, rather than theoretical performance. For example, an MFSK signal, which minimises the signal bandwidth (d=0.5) and uses coherent detection, has the same performance in AWGN as coherent PSK (p1753, [91]), but there are restrictions on the signal parameters and the equipment is conceptually complex. Non-coherent detection has been employed successfully in the past [92].

The binary FSK signal can be represented mathematically as:

$$v(t) = A\sin\left(2\pi f_c t + 2\pi f_d \int_{-\infty}^{t} D(t)dt + \theta\right) \qquad (9.1)$$

where $D(t)$ represents the random binary signal with levels ± 1. θ is the phase of the carrier at t=0. The correlation distance of the 0 and 1 symbols depends on the normalised frequency separation, $d=2f_dT_b$, where T_b is the bit duration. The signal has continuous phase, which leads to a narrower spectrum with lower side-lobes (p226 [90]) than that of a signal with abrupt phase changes.

9.3.1 Demodulation

9.3.1.1 Coherent Demodulation with Orthogonal Signals

If the receiver has perfect knowledge of the carrier phase, the signals are orthogonal whenever:

$$2f_dT_b = \frac{m}{2} \qquad \text{where } m \text{ is any integer.} \qquad (9.2)$$

So, the minimum normalised frequency separation for coherent demodulation with orthogonal signals is $2f_dT_b=0.5$. The bit error probability with Gaussian noise is shown below (p227 [90]):

$$P_e = \frac{1}{2} Q\left(\sqrt{\frac{E_b}{2N_o}}\right) \tag{9.3}$$

where E_b is the energy per bit ($= \frac{A^2 T_b}{2}$ for FSK), N_o is the noise power and

$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy$. The structure of the demodulator is complex due to the need for coherent oscillators and so this design is rarely used.

9.3.1.2 Non-Coherent Demodulation with Orthogonal Signals

Since the information is contained in frequency and not phase, a phase discontinuity at the symbol transitions will not affect the performance of a non-coherent demodulator. Non-coherent demodulation leads to a much simpler receiver design. The signals are orthogonal if:

$$2 f_d T_b = m \qquad \text{where } m \text{ is any integer} \tag{9.4}$$

The minimum frequency separation is $2f_dT_b=1$, which is twice the separation needed for coherent demodulation. The demodulator is implemented with matched filters tuned to each frequency, followed by envelope detection. A bit decision is made on the output of the envelope detectors. The bit error probability with Gaussian noise is shown below :

$$P_e = \frac{1}{2} \exp\left(-\frac{E_b}{2N_o}\right) \tag{9.5}$$

The noise input to the demodulator is assumed to be white, with a single sided psd of η watts/Hz. The filter bandwidth is usually in the order of $2r_b = 2T_b$, and so N_o is approximately equal to $2\eta/T_b$. [81].

An alternative implementation of the receiver would be through the use of a limiter and discriminator (p228 [90]). If this is assumed to be a perfect instantaneous frequency discriminator then the performance of this demodulation is only 0.5dB below the equation above, if $2f_dT_b \approx 1$.

9.4 Define the Simulation

9.4.1 Overall Structure

Figure 9.1 below shows a block diagram of the complete simulation structure [83]. The simulation is performed in C. The decoding philosophy that dictates that 2 consecutive code words must be received before a decision is made, produces very low unsafe failure probabilities. For example, figure 8.26 shows the synchronisation performance of the (31,11) BCH code with double error correction, in AWGN. The peak unsafe failure probability is

approximately 10^{-12}. To obtain statistical figures for this failure rate, the simulation time would be prohibitive. We can simulate the failure rates for correctly framed words and separately for mis-framed words. This would give us a means of assessing and comparing the code performance when synchronised, and the synchronisation performance. The simulation time is, however, extremely long, and a low number of failure events (5-10) has been accepted for the lowest failure probability points in the following results.

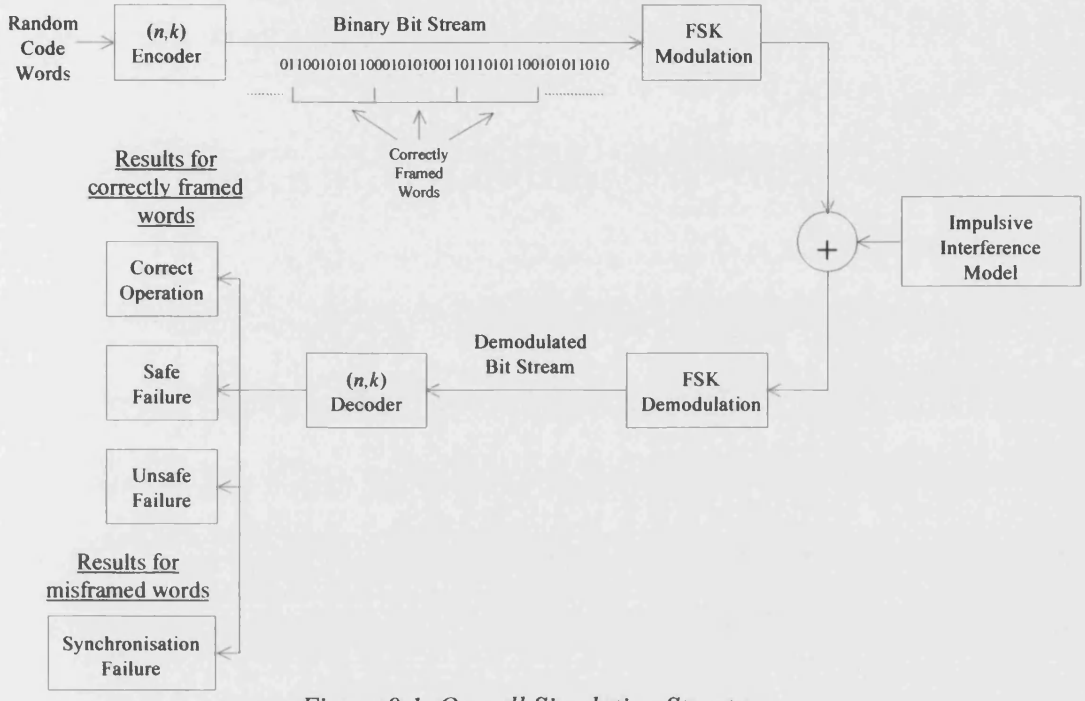


Figure 9.1. Overall Simulation Structure

The binary coding scheme is FSK modulated, the noise is added and the signal is demodulated. The decoding algorithm determines whether a safe or unsafe failure has occurred. The FSK parameters are f_1 = 1600Hz, f_2 = 1700Hz, f_d = 50Hz, r_b=50 b/s. The normalised frequency deviation (modulation index), $d=2f_d T_b=2$ (note: these FSK parameters have been chosen arbitrarily. In an actual system, the signal frequencies would be chosen to avoid odd harmonics of 50Hz. This simulation however, will not be modelling the power harmonics). The sampling frequency is chosen to be 8.4kHz, giving 5 samples/cycle, and 168 samples per symbol.

9.4.2 Non-Coherent FSK Demodulator Architecture

The simulation will not implement envelope detection, since this is a non-linear analogue operation. It will instead implement digital correlation detectors (integrate and dump) for both symbols, the bit decision being made on the symbol frequency containing the highest power (figure 9.2). Band-pass filters can be added to the design, so the effect of band-limiting the impulsive interference can be studied. Band-pass filters are used as frequency discriminators

for non-coherent demodulation with envelope detection (p413, [81]). For this simulation, they have been implemented with 6th order Chebychev IIR filters, centred at the signal frequencies. This may not be the optimum receiver filter, and designs that produce a raised cosine frequency characteristic for the symbols may reduce ISI. However, since the bit rate is low, by data communication standards, this is not considered necessary. The digital receiver design with these filters is shown in figure 9.3 (See also [92]). The BPF response is shown in figure 9.4.

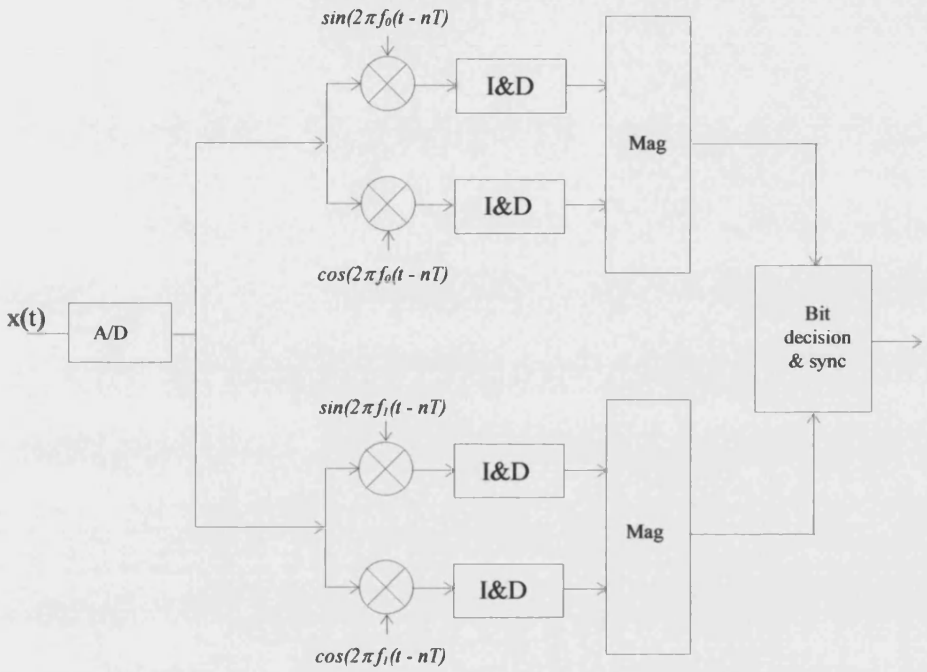


Figure 9.2. Digital Non-Coherent FSK Detector

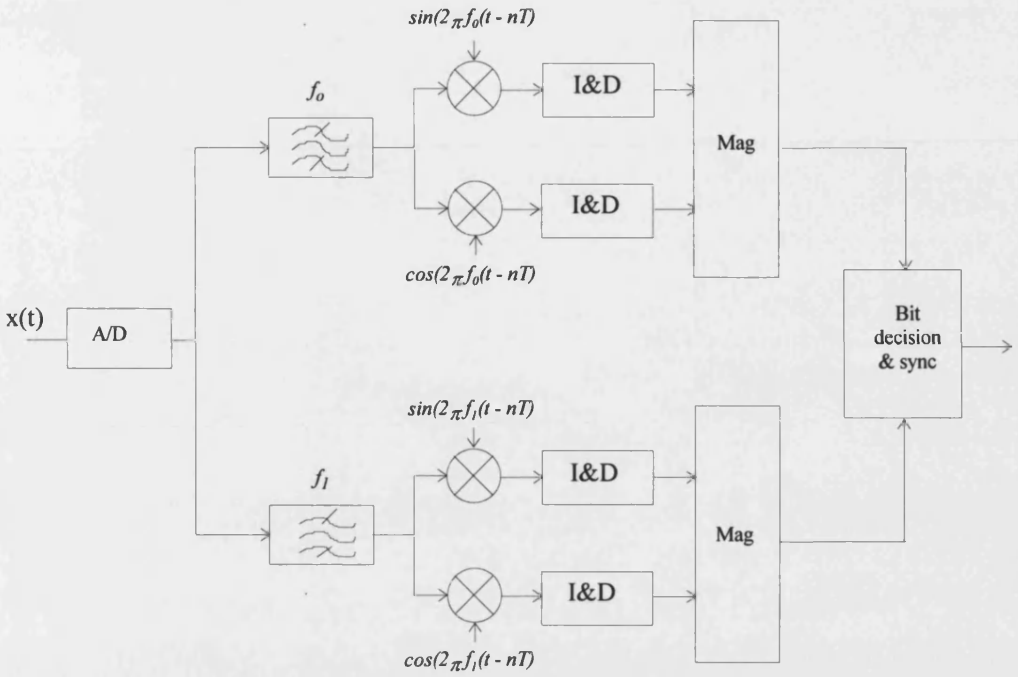


Figure 9.3. Receiver Design with Signal Frequency Filters.

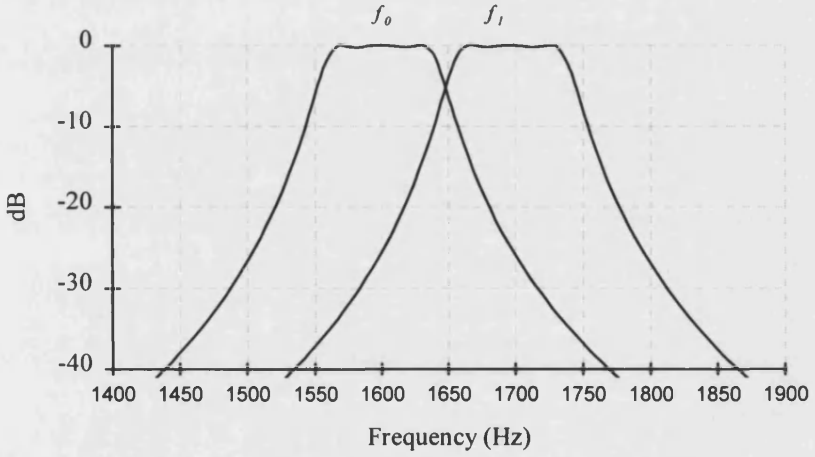


Figure 9.4. FSK Filter response

The filter impulse responses are shown below:

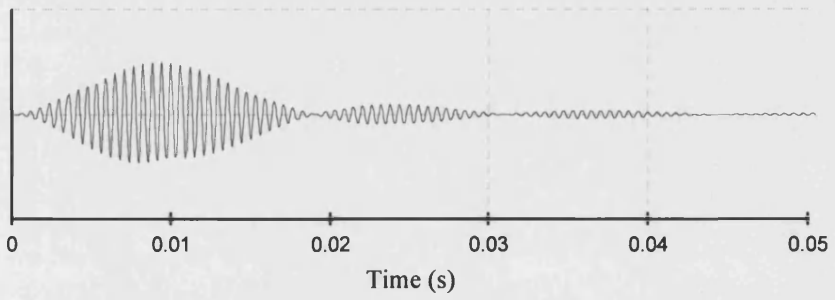


Figure 9.5. IIR Filter Impulse Response for f_1

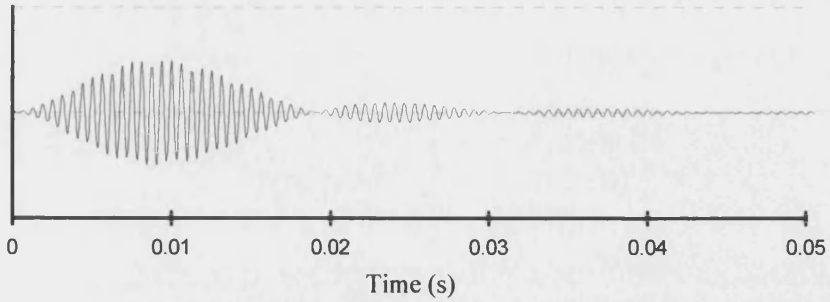


Figure 9.6. IIR Filter Impulse Response for f_2

Figure 9.7 shows the FSK signal with alternating symbols after filtering by one BPF (note that since there are only 5 samples per carrier period there appears to be some envelope fluctuation, which does not really exist).

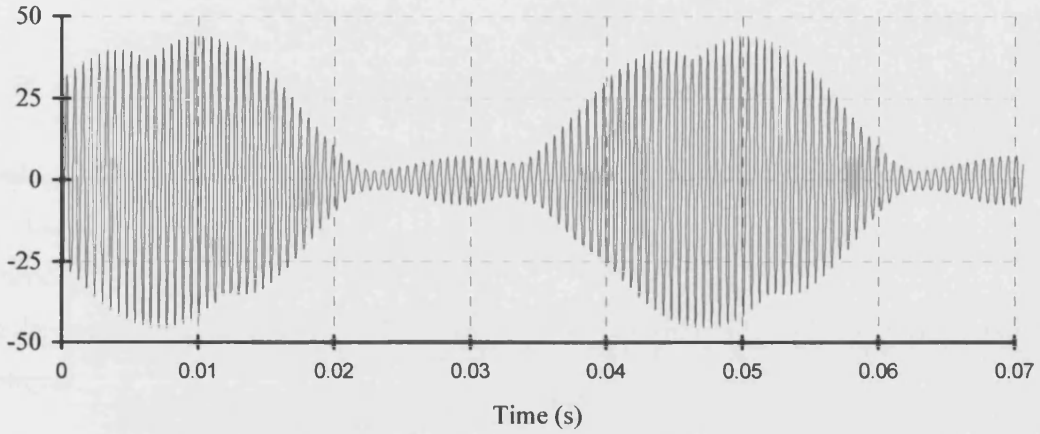


Figure 9.7. Sampled FSK Signal after filtering by one FSK BPF.

9.4.3 Effect of Receiver Filters

To investigate to effect of the receiver filters on the demodulation, a simulation was performed with random data in AWGN with (a) an ideal demodulation process (figure 9.2) and (b) with the receiver filters as figure 9.3 (signal and noise through band pass filters). The SNR is taken as the SNR at the receiver input, $\text{SNR} = 10\log_{10}\left(\frac{E_b}{\eta}\right)$ where E_b is the energy per bit $\left(\frac{A^2 T_b}{2}\right)$ and η is the noise power per unit bandwidth (single sided). The results are shown in figure 9.8.

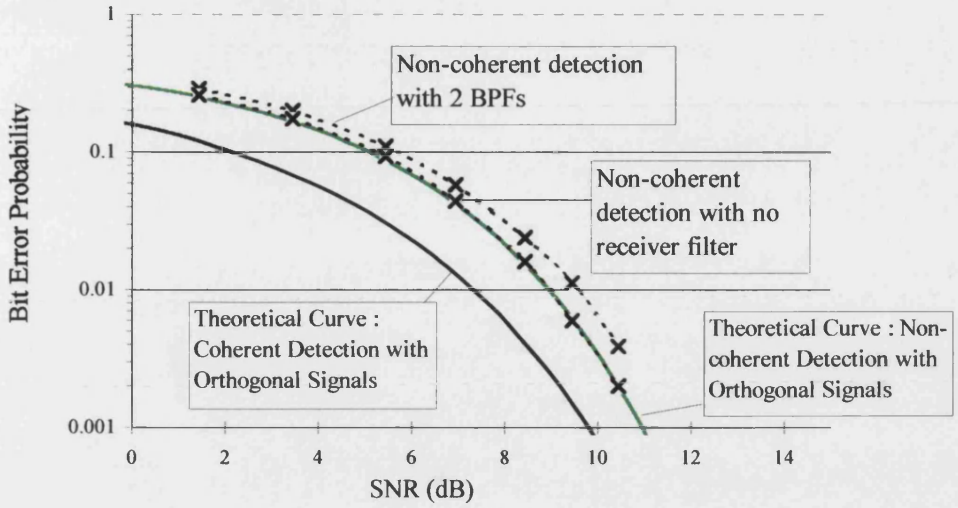


Figure 9.8. Probability of bit error curves for FSK

It can be seen that the curve for non-coherent detection with non-band-limited Gaussian noise closely fits the theoretical curve. The bandpass filters will correlate the Gaussian noise samples and introduce ISI at the symbol boundaries. The SNR penalty (with Gaussian noise) for using the two BPFs centred on the signal frequencies is about 0.5dB. Note that the filtering process introduces a delay and affects the optimum bit sampling time. This has been taken into account in the simulation by 'manually' delaying the bit sampling to the point where the output of the integrate and dump process is at a maximum.

9.4.4 Defining the Noise Model

The impulsive noise model described in chapter 6 will be used to asses the coding performance. The model consists of an impulse train, whose period is varied at a rate of 1ms/s. The period is swept up and down from the lowest period, 3ms, to the highest period, 600ms (arbitrarily chosen to produce one impulse per code word). The simulation will assume that each impulse will result in the BPF impulse response being added to the sampled signal. This implies that the impulse train is filtered with a perfect anti-aliasing filter and the sampling instant coincides exactly with the peak of the sinc pulse after this filter (figure 9.9). This simplifying assumption has been made since the simulation and interference model is designed to study the effects of impulsive/bursty errors on a particular coding scheme, rather than assess the performance of specific receiver equipment and interference characteristics. The effects of A/D clipping are ignored.

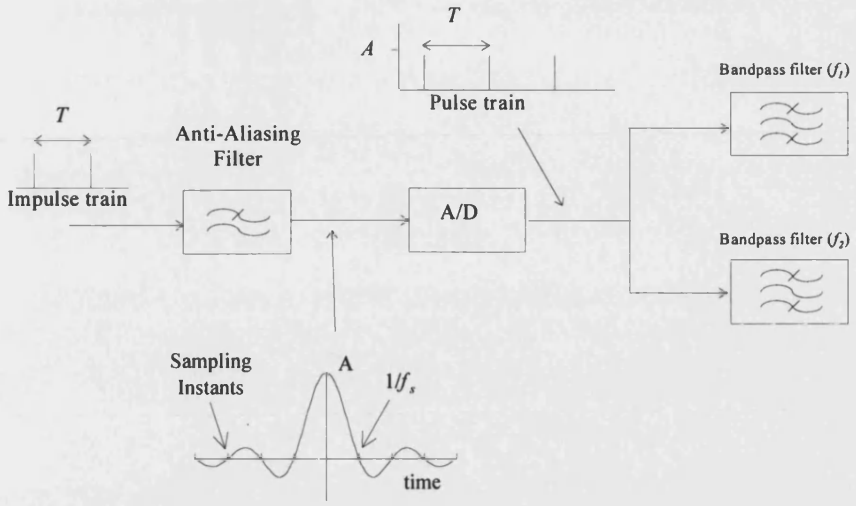


Figure 9.9

In the simulation, the impulse responses are truncated at a point where the amplitude is negligible (260 samples). If the original pulse spacing (T) is less than the impulse response length, then the interference will cause overlapping impulse responses. These overlapping responses may add or cancel each other, depending on the distance between successive impulses.

Figure 9.10 below shows the FSK bit error rate with this impulsive interference. These results are obtained from statistical simulations. Each bit error probability point is calculated from a trial of 16.10^6 bits.

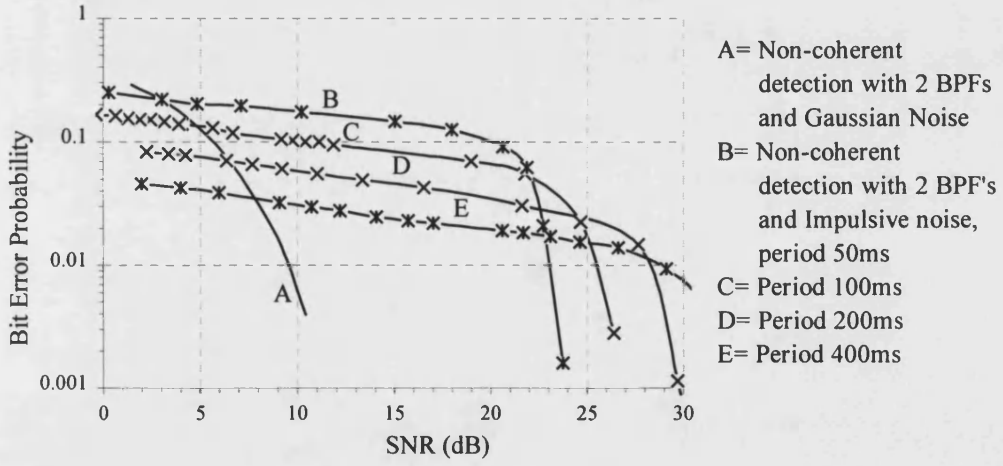


Figure 9.10. Bit Error Rates for Impulsive Interference

The impulses are of varying periods (bit rate = 50b/s). Also shown, for comparison, is the bit error rate with Gaussian noise and 2 BPFs centred on the signal frequencies. For this curve, the SNR is calculated as the (energy per bit)/(Gaussian noise variance) or 'average noise power'. The Gaussian noise is added after the BPFs, in order to cause independent bit errors.

For the impulsive noise, the SNR is calculated as the (energy per bit)/(average impulsive noise power *after* the BPFs).

Note that the impulsive noise curves are flatter. Since the interference is concentrated in bursts, once these bursts are strong enough to corrupt the symbols, increasing the power will have little effect on the adjacent symbols. When the SNR is high enough, the bursts will not cause symbol errors and the bit error rate falls rapidly. A higher bandwidth receiver filter will produce a flatter curve in the low SNR values, since the bursts will decay rapidly, and not affect adjacent symbols.

For higher signal to noise ratios, the impulsive noise generally produces more errors, since the energy is concentrated in short bursts. Note that the points where curves B to E fall rapidly are separated by approximately 3dB. This is because the period between pulses is successively doubled and so the overall power will be halved. Figure 9.11 below shows the same curve with the *x* axis as (Energy per bit *(Eb)* / Energy per burst). A burst is one impulse response from the receiver filter, which is actually truncated to 260 samples (where the amplitude is negligible). This measure is independent of the period between the impulses.

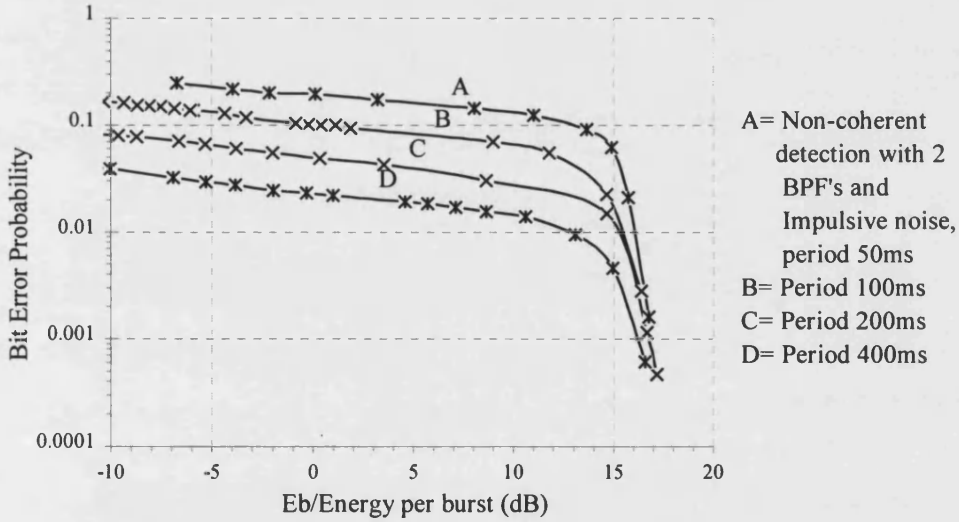


Figure 9.11. Bit Error Rates for Impulsive Interference

We can see that the curves fall rapidly beyond a certain threshold (≈ 15dB), where the impulse response is not strong enough to cause a bit error.

9.5 Results for Coding

9.5.1 Comparisons with Theoretical Results: Gaussian Noise

The aim of the simulation is to compare the performance of the coset codes with AWGN and impulsive noise. The simulations were first performed with AWGN in order to verify the simulation model and the theoretical equations for code performance presented in section 8.2.

9.5.1.1 (21,9) Shortened Cyclic Code

The following results apply to the (21,9) shortened cyclic code. This code has been designed for efficient single burst error correction (p269 [55]). When correctly synchronised, the code is capable of correcting single bursts of up to length 6. It has a minimum distance of 6 and an index of comma freedom of 3. The coset is defined by $p(X)=X^6+X^5+X^4+X$. Its has been shown in section 8.5.5 that with Gaussian noise the optimum performance is obtained with single error correction. Results for no error correction and single error correction will be compared.

(21,9) Shortened Cyclic Coset Code with AWGN : Correctly Framed Words

Figure 9.12 below shows the simulation results for correctly framed words with no error correction, in AWGN. The failure probabilities are plotted against bit error probability instead of SNR, so that the theoretical curves (continuous line) can also be shown. The theoretical curves are taken from equations 8.20 to 8.22.

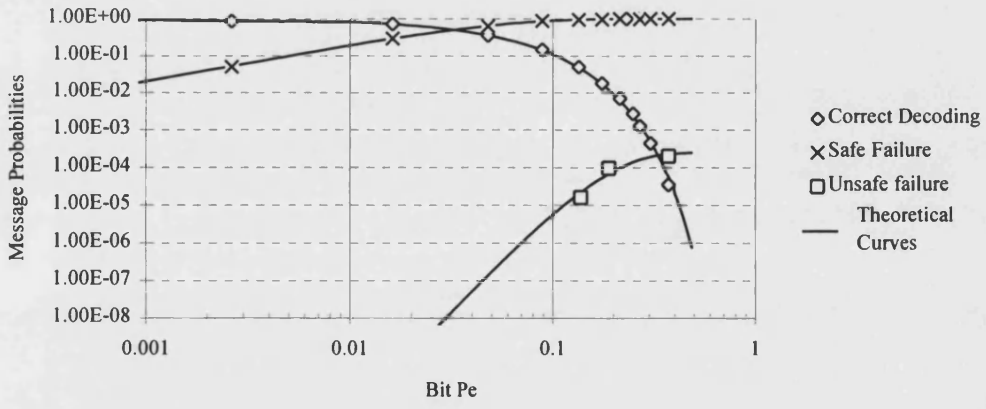


Figure 9.12. Message Error Probabilities for Correctly Framed (21,9) Shortened Cyclic Code Words with no Error Correction (Gaussian Noise).

Figure 9.13 shows the same plot, for single error correction. It can be seen that the simulated results with AWGN agree with the theoretical equations. The discrepancies in the low unsafe failure points are due to the fact that it takes several days for one unsafe failure to occur in the simulation and so the measurement variance for these results is high.

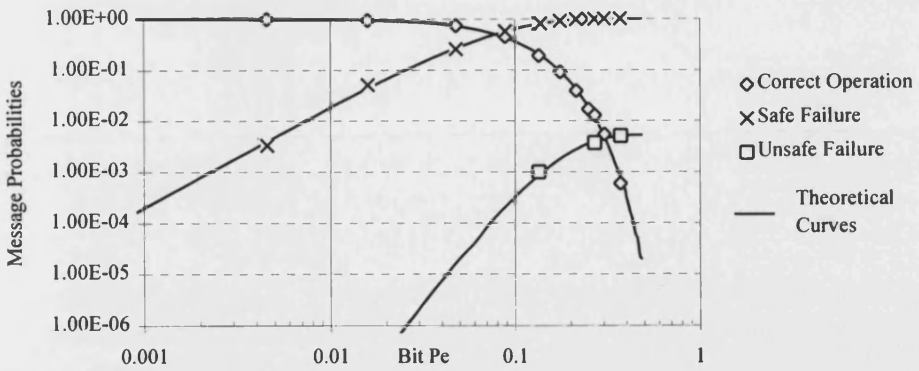


Figure 9.13. Message Error Probabilities for Correctly Framed (21,9) Shortened Cyclic Code Words with Single Error Correction (Gaussian Noise).

(21,9) Shortened Cyclic Code with AWGN: Mis-framed Words

Shown below are the theoretical curves and measured results for the failure rate of mis-framed words, with AWGN (in this context, a *failure* occurs when bit errors change a mis-framed word into a valid or correctable code word). The theoretical curve is taken from equation 8.24. Results for no error correction and single error correction are shown. The simulated results take an extremely long time, since all values of mis-framing (synchronisation slip) and all code words have to be considered. For this reason, the simulation has been restricted to just one code word from the dictionary. This, and a high measurement variance could explain the discrepancy in the measured results to the theoretical results at high message error rates.

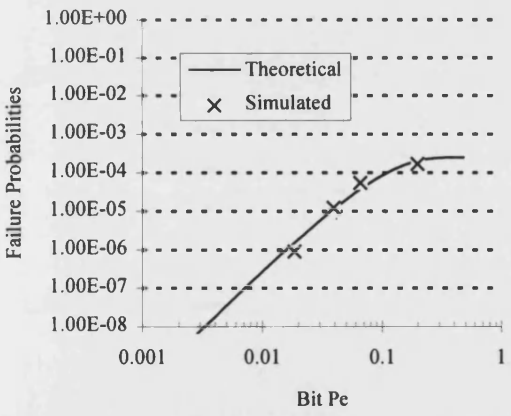


Figure 9.14. Message Error Probabilities for Mis-framed (21,9) Shortened Cyclic Code Words with no Error Correction (Gaussian Noise).

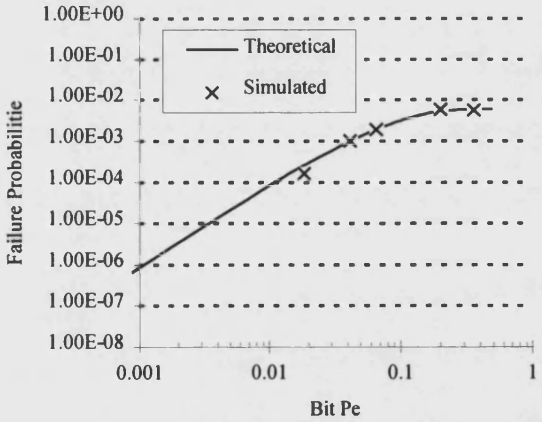


Figure 9.15. Message Error Probabilities for Mis-framed (21,9) Shortened Cyclic Code Words with Single Error Correction (Gaussian Noise).

9.5.1.2 (31,11) BCH Coset Code.

The following results apply to the (31,11) BCH coset code, which has an index of comma freedom of 5, and a minimum distance of 11. The coset is defined by $p(X)=X^{10}+X^{9}+X^{8}+X^{4}+X$. Its has been previously shown that with Gaussian noise the optimum performance is obtained with double error correction (see figure 8.26). Results for no error correction and double error correction will be compared.

(31,11) BCH Coset Code with AWGN : Correctly Framed Words

Figure 9.16 below shows the simulation results for correctly framed words with no error correction, in AWGN. The failure probabilities are plotted against bit error probability instead of SNR, so that the theoretical curves can also be shown. Figure 9.17 shows the same plot, for double error correction. It can be seen that the simulated results with AWGN agree with the theoretical equations.

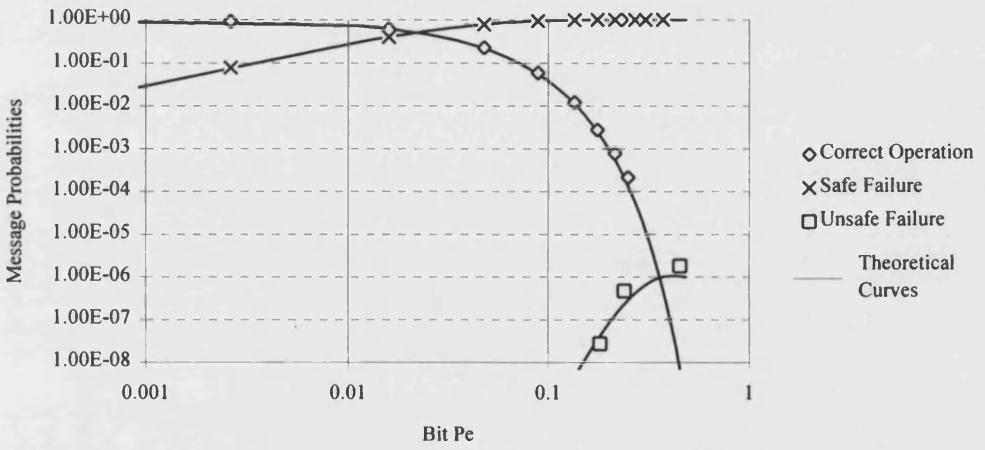


Figure 9.16. Message Error Probabilities for Correctly Framed (31,11) BCH Code Words with no Error Correction (Gaussian Noise)

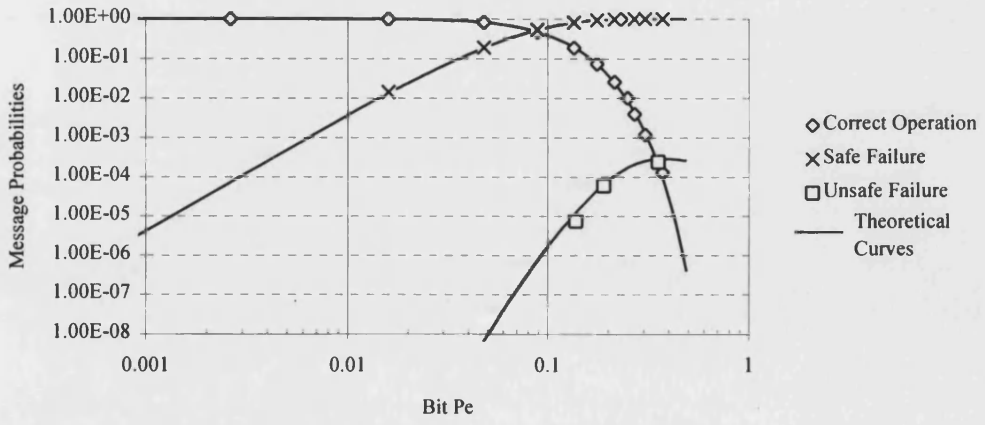


Figure 9.17. Message Error Probabilities for Correctly Framed (31,11) BCH Code Words with Double Error Correction.(Gaussian Noise)

(31,11) BCH Coset Code with AWGN : Mis-framed Words

Shown in figure 9.18 is the theoretical curve and measured results for the failure rate of mis-framed (31,11) BCH code words, with AWGN (in this context, a *failure* occurs when bit errors change a mis-framed word into a valid or correctable code word). No error correction is used. The simulated results take an extremely long time, since all values of mis-framing (synchronisation slip) and all code words have to be considered. For this reason, the simulation has been restricted to just one code word from the dictionary. This, and a high measurement variance could explain the discrepancy in the measured results to the theoretical results at high message error rates.

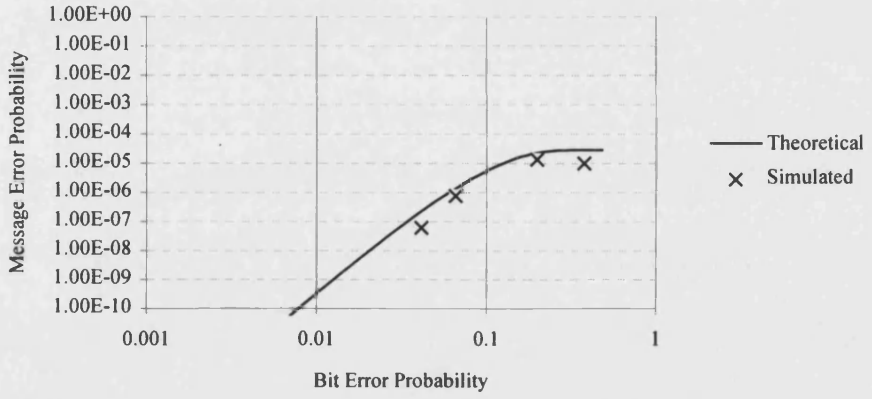


Figure 9.18. Message Error Probabilities for Mis-framed (31,11) BCH Code Words with no Error Correction (Gaussian Noise).

Figure 9.19 shows the failure probability for mis-framed (31,11) BCH words, with double error correction.

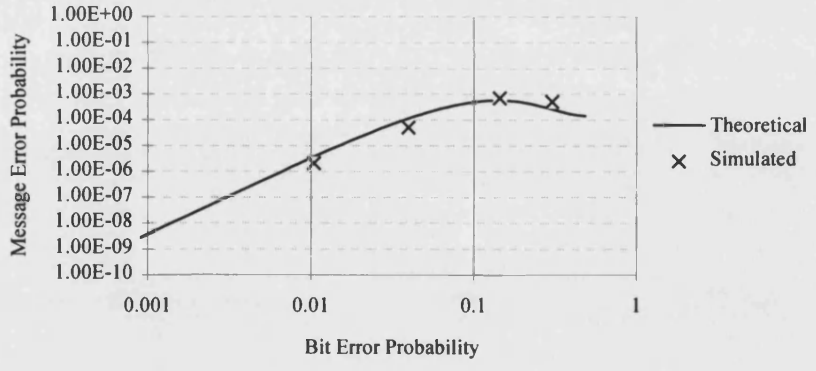


Figure 9.19. Message Error Probabilities for Mis-framed (31,11) BCH Code Words with Double Error Correction (Gaussian Noise).

9.5.2 Coding Results with Impulsive noise model

9.5.2.1 (21,9) Shortened Cyclic Code

9.5.2.1.1 Correctly Framed Words

Figure 9.20 below shows the simulation results for the failure probabilities of correctly framed (21,9) code words, with impulsive noise and no error correction. Also shown in figure 9.21 for comparison are the failure probabilities with Gaussian noise. Figures 9.22 and 9.23 show the same results for single error correction.

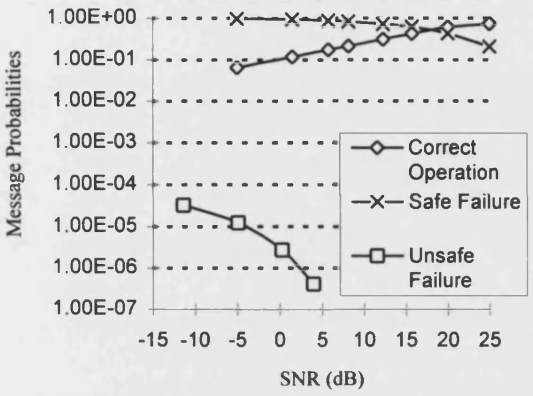


Figure 9.20. Correctly Framed (21,9) Code Words with No Error Correction. Impulsive Noise.

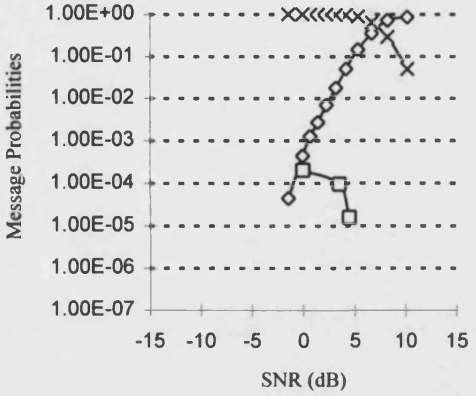


Figure 9.21. Correctly Framed (21,9) Code Words with No Error Correction. Gaussian Noise.

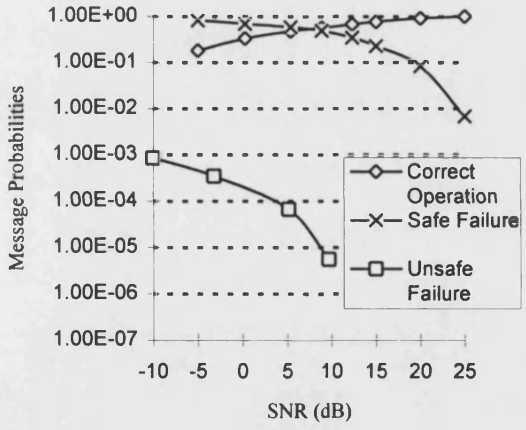


Figure 9.22. Correctly Framed (21,9) Code Words with Single Error Correction. Impulsive Noise.

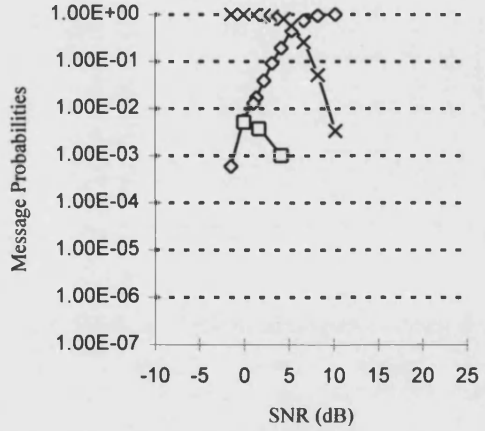


Figure 9.23. Correctly Framed (21,9) Code Words with Single Error Correction. Gaussian Noise.

It will be useful to view the same results against the average bit error rate. For both the impulse interference and Gaussian noise results, the average bit error rate is taken over the

complete trial, for each point. Figure 9.24 shows the results for no error correction. Figure 9.25 shows the results for single error correction.

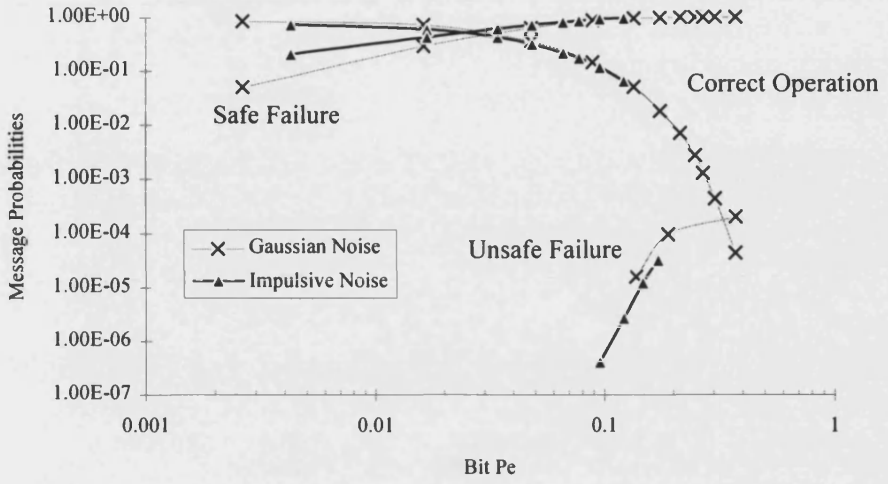


Figure 9.24. Correctly Framed (21,9) Shortened Cyclic Code Words with No Error Correction

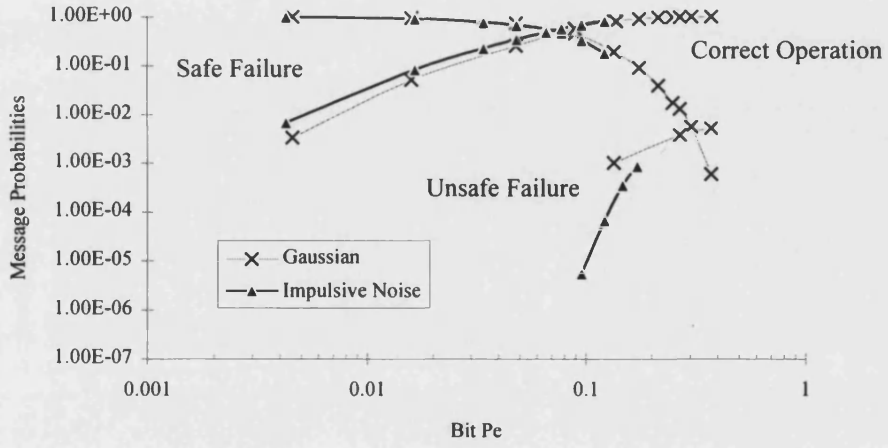


Figure 9.25. Correctly Framed (21,9) Shortened Cyclic Code Words with Single Error Correction

9.5.2.1.2 Mis-framed Words

Shown below are the failure probabilities for mis-framed (21,9) shortened cyclic code words. The curves for both impulsive and Gaussian noise are shown on the same graph. Figure 9.26 shows the results for no error correction and figure 9.27 shows the results for single error correction.

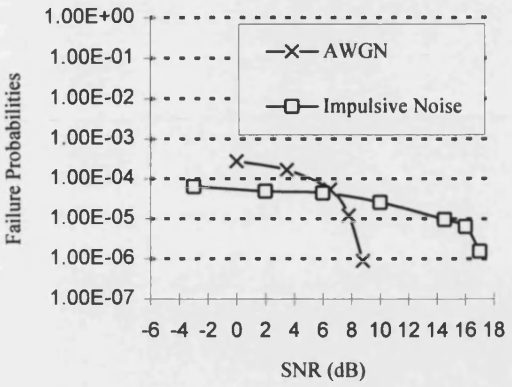


Figure 9.26. Failure probabilities for mis-framed (21,9) words with no error correction

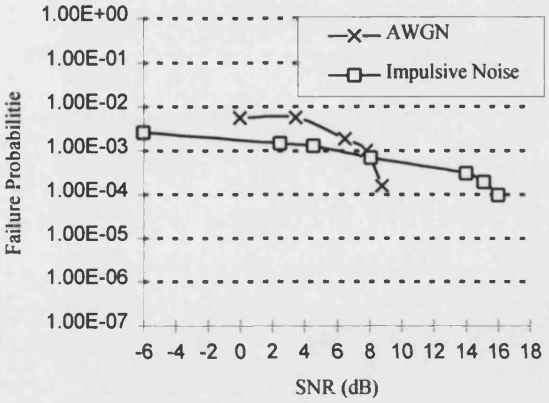


Figure 9.27. Failure probabilities for mis-framed (21,9) words with single error correction

The same results are plotted in figures 9.28 and 9.29 against the average bit error rate.

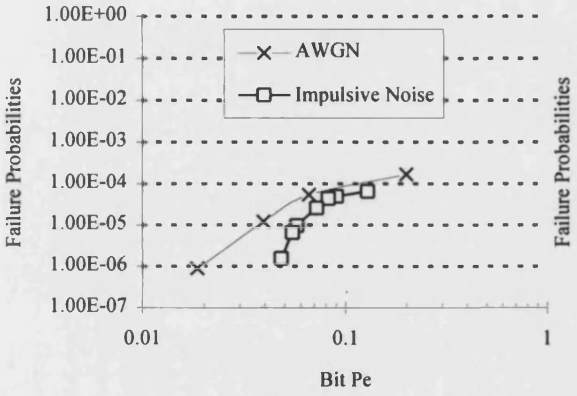


Figure 9.28. Failure probability for mis-framed (21,9) words, with no error correction

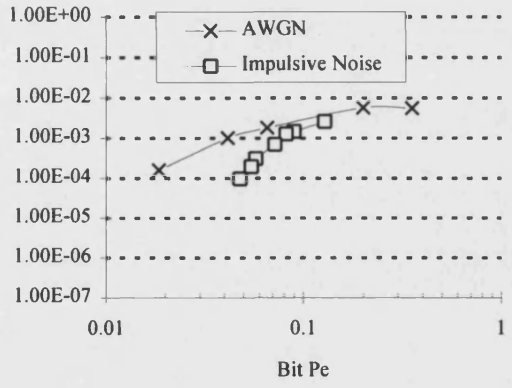


Figure 9.29. Failure probability for mis-framed (21,9) words, with single error correction

9.5.2.2 (31,11) BCH Coset Code.

9.5.2.2.1 Correctly Framed Words

Figure 9.30 below shows the simulation results for the failure probabilities of correctly framed (31,11) BCH code words, with impulsive noise and no error correction. Also shown in figure 9.31 for comparison are the failure probabilities with Gaussian noise. Figures 9.32 and 9.33 show the same results for double error correction.

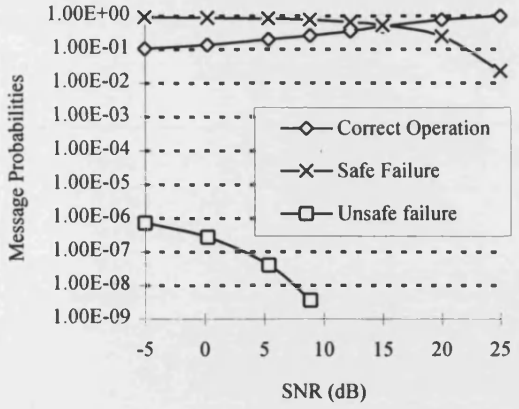


Figure 9.30. Correctly Framed (31,11) BCH Code Words with No Error Correction. Impulsive Noise

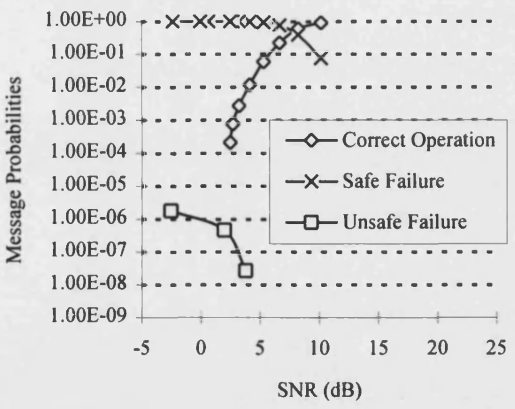


Figure 9.31. Correctly Framed (31,11) BCH Code Words with No Error Correction. Gaussian Noise

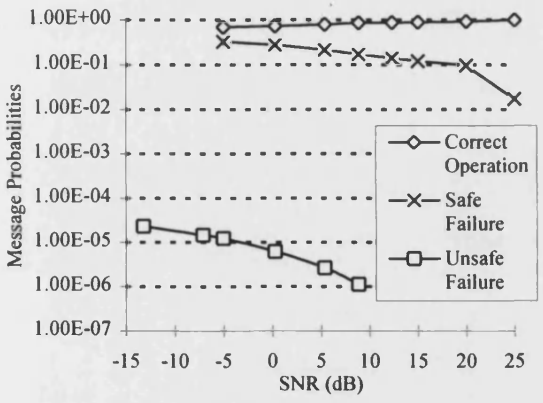


Figure 9.32. Correctly Framed (31,11) BCH Code Words with Double Error Correction. Impulsive Noise

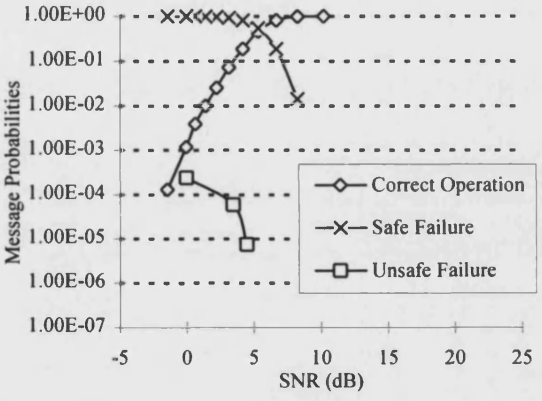


Figure 9.33. Correctly Framed (31,11) BCH Code Words with Double Error Correction. Gaussian Noise

Figure 9.34 and 9.35 show the results for no error correction and double error correction when plotted against average bit error rate.

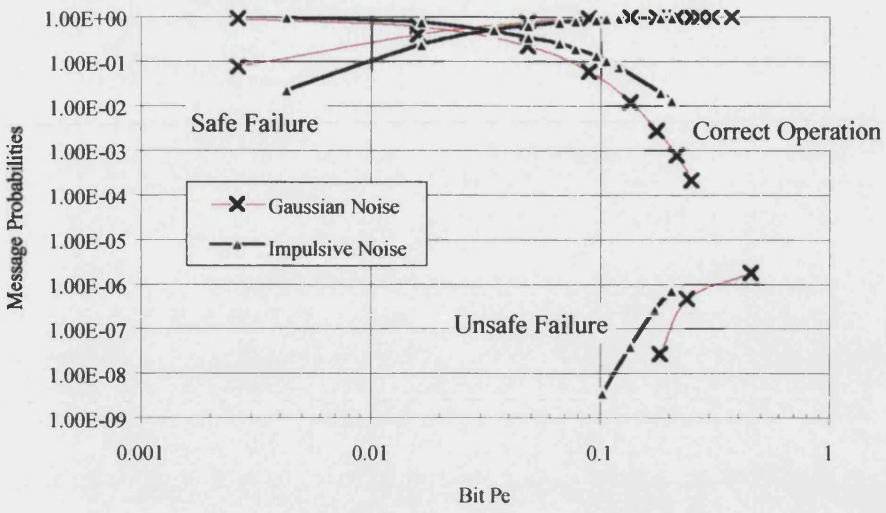


Figure 9.34. Correctly framed (31,11) BCH Code Words with No Error Correction.

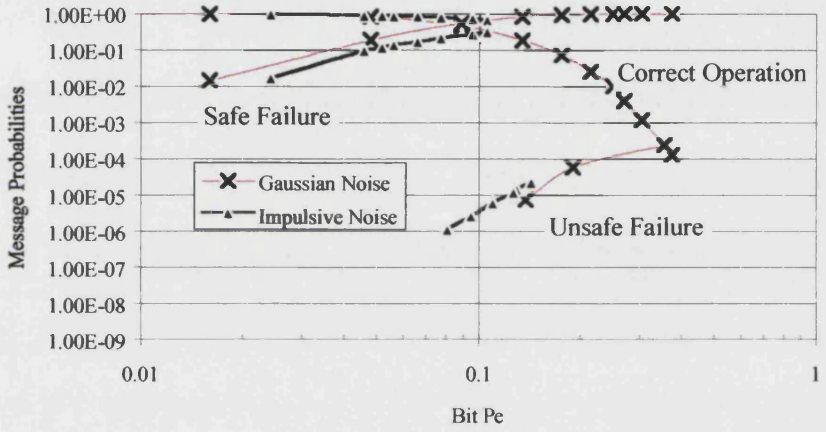


Figure 9.35. Correctly framed (31,11) BCH Code Words with Double Error Correction

9.5.2.2.2 Mis-framed Words

Shown below are the failure probabilities for mis-framed (31,11) BCH coset code words. The curves for both impulsive noise and for Gaussian noise are shown on the same graph. Figure 9.36 shows the results for no error correction and figure 9.37 shows the results for double error correction.

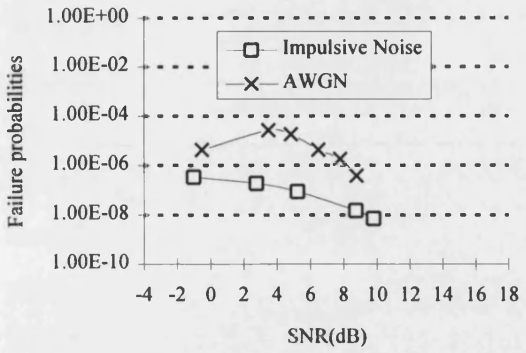


Figure 9.36. Failure Probabilities for Mis-framed (31,11) Words with no Error Correction

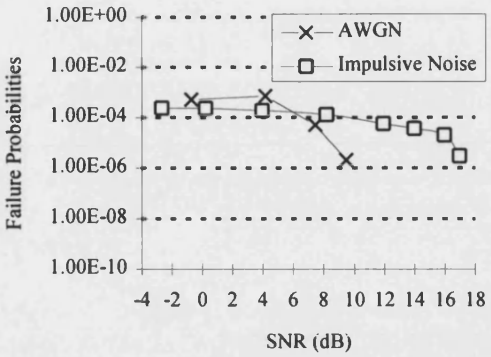


Figure 9.37. Failure Probabilities for Mis-framed (31,11) Words with Double Error Correction

The same results are plotted below against the average bit error rate.

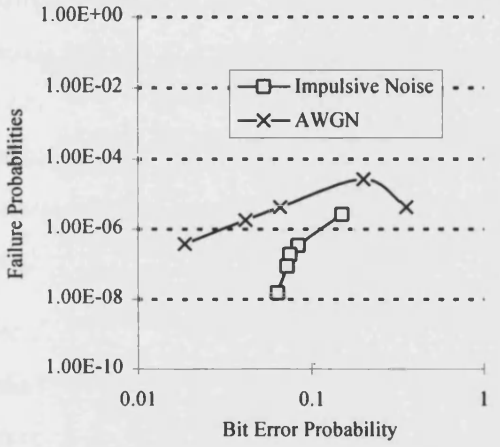


Figure 9.38. Failure Probability for Mis-framed Words, with no Error Correction

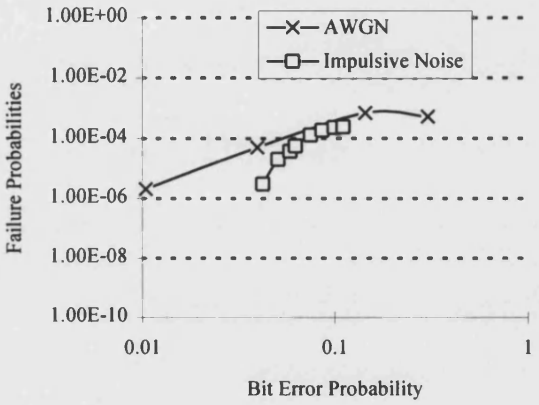


Figure 9.39. Failure Probability for Mis-framed Words, with Double Error Correction

9.6 Comments

Correctly framed words

When the receiver has decided that synchronisation has been achieved, it need only look at successive frames of n bits. An unsafe failure occurs when the wrong code word is decoded at the receiver. A safe failure occurs when an error is detected. The correct decoding and safe failure curves give us a measure of the reliability of the code.

When the code failure probabilities are plotted against the SNR (figures 9.20 to 9.23 and figures 9.30 to 9.33) it can be seen that the correct operation and safe failure curves for impulsive noise are generally flatter than those for Gaussian noise. We can see that at low SNR, the impulsive noise model causes lower unsafe failures than Gaussian noise. However,

the unsafe failure curve for Gaussian noise decreases sharply with increasing SNR. With Gaussian noise, the safe failure curves fall sharply above 5 dB, but the impulsive noise curves only start falling at approximately 20dB. This is because all the noise power is concentrated in bursts and so will cause bit errors even at high SNR. Comparison of figures 9.20 and 9.22 for the (21,9) code, and figures 9.30 and 9.32 for the (31,11) code, suggest that, for impulsive noise, error correction is a more effective method of increasing the reliability than increasing the SNR. These curves also demonstrate that the trade off between error correction and error detection corresponds to a trade off between safety and reliability. This applies to both Gaussian and impulsive noise.

When the failure probabilities of correctly framed words are viewed against the same overall bit error probability we can see that for the (21,9) cyclic code (figs 9.24 and 9.25) the impulsive noise (bursty errors) produces slightly lower reliability and higher safety that Gaussian noise. This applies to both no error correction and single error correction. For an unsafe error to occur, the bit error pattern must be the same as a valid or correctable (21,9) cyclic code word. Since the code was designed for single burst error correction (of length 6), the code vectors will not contain single bursts of length 12 or less (p53 [89]). When the period between the impulses is large (>400ms) the impulsive interference is likely to cause these single bursts, and a decoding error will not occur.

For correctly framed (31,11) BCH code words (figures 9.34 and 9.35), it can be seen that the reliability is higher for impulsive noise (bursty errors), for both zero and double error correction. This is accompanied by an increase in the unsafe failure curve, although, for high bit error rates with double error correction, this increase is marginal. For an unsafe failure to occur, the bit error pattern must be the same as a valid (or correctable) (31,11) BCH code word. Figures 9.34 and 9.35 suggest that random error patterns are marginally less likely to be the same as valid code words as the error patterns caused by impulsive noise, when the same overall bit error rate is compared.

Mis-framed words

When searching for synchronisation, the receiver will look at the first n bits to see if they represent a valid or correctable code word. If no decoding occurs, then the receiver will slide along by one bit, and repeat the process. The failure probabilities for mis-framed words represent the probability of bit errors changing these words into valid or correctable code words.

We can see that for both the (21,9) cyclic code and the (31.11) BCH code (figures 9.26, 9.27, 9.36 and 9.37) that at lower SNRs, the impulsive noise produces fewer unsafe failures, but the curves do not decay as rapidly as the Gaussian noise curves. If the simulation was extended to higher SNRs the failure rates would fall rapidly, since the impulse power would be too low to cause any bit errors. This is because the noise power is concentrated in bursts. Once the bursts are strong enough to corrupt the symbols, increasing the power will have little effect on the adjacent symbols.

If we plot the failure probabilities against the overall bit error probability (figures 9.28, 9.29, 9.38 and 9.39), we see that the peak of the plot for impulsive noise is lower that that of Gaussian noise, and falls off rapidly for lower overall bit error rates. At these lower rates, the SNR is higher and the noise will only cause errors at the peak of the impulse response and so the errors will be concentrated in bursts. The lower failure rate could be explained by thinking about the distance vectors between the mis-framed words and the valid code words. If these distance vectors have random rather than bursty patterns, the failure probability for mis-framed words will be less with bursty errors, when the same overall bit error probability is compared. It can also be seen that the decrease in failure rate for mis-framed (31,11) BCH code words, when no error correction is used, is higher for impulsive noise.

9.7 Modelling the Effects of Impulsive Interference on Symbols

If further simulation is required it will be easier to model the interference effects on the symbols rather than the modulated signal. The simple model described below applies to the type of interference and receiver filters described in this chapter. The bandwidth of the receiving filters is approximately r_b Hz and we can see from figure 9.4 that the impulse response will not affect more than 2 adjacent symbols. When the period between the interfering impulses is larger than the symbol length, the probability of an impulse causing a symbol error is given by the curve below.

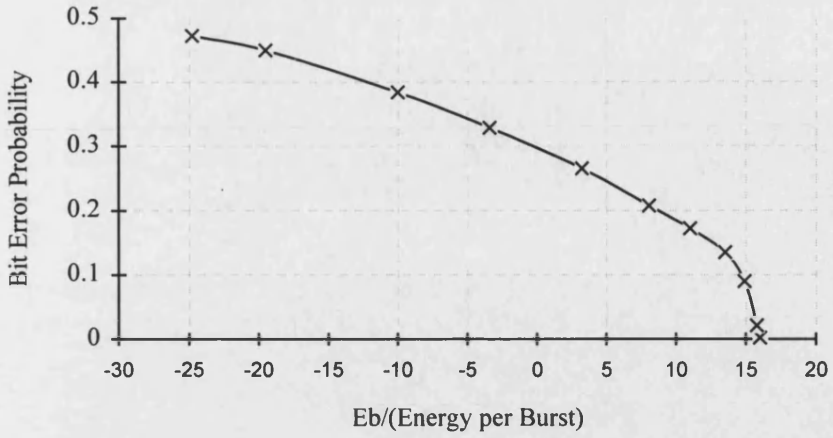


Figure 9.40. Probability of an impulse causing a bit error.

This can be approximated by:

$$Pe = \begin{cases} 0.5 & \text{if } Eb / (\text{Energy per Burst}) < -30 \\ -8.22 \,.\, 10^{-3}\left(Eb / \text{Energy per Burst}\right) + 0.26 & \text{if } -30 < Eb / (\text{Energy per Burst}) < 16 \\ 0 & \text{if } Eb / (\text{Energy per burst}) > 16 \end{cases} \qquad (9.6)$$

Given that an impulse has caused a single bit error, the probability of the succeeding bit being in error was observed to be approximately 0.5. If the period between the impulses is less that the bit length, then the bit error probability of each bit is also approximated by equation 9.6.

9.8 Conclusions

Since the simulations take an extremely long time to run (in the order of days, for an unsafe failure to occur, at the low failure rates) the number of results is restricted. Also for this reason, a high measurement variance has been accepted at high message error rates.

The results are specific to the particular noise model being used. The period between the impulses is ramped up and down so that at low periods, multiple impulses occur within one symbol and at high periods, single impulses (bursts) will occur in the code word. The results have been restricted to two codes: the (21,9) shortened cyclic coset code and the (31,11) BCH coset code. Some of the parameters for these codes are shown below.

	(21,9) Shortened Cyclic Coset Code	(31,11) BCH Coset Code
Code length (bits)	21	31
Information rate (k/n)	0.43	0.36
Minimum distance	6	11
Index of comma freedom	3	5

Table 9.1. Parameters of (21,9) Shortened Cyclic Code and (31,11) BCH Code.

For a given average bit error probability it can be concluded that these codes perform as successfully with this impulsive noise model as they do with Gaussian noise. For both codes studied, the synchronisation failure rate is actually lower for impulsive noise (figures 9.28, 9.29, 9.38 and 9.39), which implies that the distance vectors from the mis-framed words to the valid code words contain randomly distributed patterns.

9.8.1 (21,9) Shortened Cyclic Code

The correct decoding and safe failure curves of figures 9.20 to 9.23 give us a measure of the reliability of the code. Comparison of figures 9.20 and 9.22 suggest that, for impulsive noise, error correction is a more effective method of increasing the reliability than increasing the SNR. However, there is an increase in the unsafe failure rate by a factor of over 10. When the same curves are viewed against the overall bit error probability (figures 9.24 and 9.25) the impulsive noise model will bias the results toward safety and against reliability. This is due to the structure of the code, which is designed for single burst error correction.

The synchronisation performance of this code plotted against the average bit error rate is better than the performance with Gaussian noise (figures 9.28 and 9.29), especially at lower bit error probabilities.

The (21,9) code is capable of correcting all bursts of length 6, when correctly synchronised. For this to be achieved, the design of the decoder would have to assume that all the errors are confined to bursts of length 6 or less. With the Gaussian and impulsive noise models, this would not be the case, and the performance would be worse with this type of decoder. Burst error *correction* is not feasible, due to the comma free requirement and so the use of specially constructed burst error control codes, such as the (21,9) code, may provide little advantage.

9.8.2 (31,11) BCH Coset Code

For the (31,11) BCH code, a few points were taken for single error correction. As expected, these points lay approximately midway between zero and double error correction, in terms of

safety and reliability. In general it can be concluded that with impulsive noise as well as Gaussian noise, double error correction provides the best trade off between safety and reliability. However, figures 9.34 and 9.35 show that for correctly framed words, the impulsive noise will bias the results against safety (towards reliability). Although this effect is marginal, conservatism may require that the error correction be reduced, to single error correction.

When similar bit error rates are compared, the error patterns caused by the impulsive noise model are less likely to cause synchronisation errors than random error patterns, especially at low bit error rates.

When synchronisation has been achieved the receiver need only look at successive blocks of n bits. The coding results for correctly framed words will show us the performance under these conditions. Comparison of figures 9.24, 9.25 and figures 9.34, 9.35 shows us that the (31,11) BCH code has a lower unsafe failure rate, by a factor of between 10 and 50, than the (21,9) code. With impulsive noise, the (31,11) code also has a much greater reliability. There is a penalty in information rate however, as the (31,11) has 20% more redundancy.

It can generally be concluded that, if a good coset is found, BCH codes give better performance that the other codes considered in this thesis. BCH codes also have good burst error detection capability, and good distance properties for correctly framed words. The (31,11) BCH coset code is the largest code studied and is recommended for track circuit use, since it has the highest data capacity and best performance with both Gaussian noise and the impulsive interference model.

A further increase in the track circuit data capacity would be possible if the message is split into several code words. For example, a message consisting of 3 (31,11) BCH coset code words would have a data capacity of 33 bits. The receiver would need to 'see' a maximum of ($4n$-1) code words within the track circuit response time. If the response time is 0.5 seconds, a signalling rate of 246 bit/s is required, occupying a bandwidth of approximately 500Hz for FSK modulation. The extra message synchronisation requirements would have to be further studied and an appropriate decoding philosophy would need to be applied. For example, the restriction that 2 consecutive code words must be the same may not be applied, but decoding would required that 3 valid consecutive code words must be decoded with no decoding in the overlap positions. The performance of this decoding philosophy would have to be analysed.

SECTION III

Chapter 10: Modulation Schemes for Track Circuits

Chapter 11: MFSK for Track Circuit Modulation

Chapter 12: A Stationary Multi-tone Modulation Scheme Requiring no Synchronisation

Chapter 13: Block Coding Techniques with Hard and Soft Decisions for Stationary Multi-tone Signal

Section III is concerned with increasing the track circuit data capacity using modulation and coding schemes suitable for a signal which may be required to be fed through existing track circuit tuned terminations. This is necessary in order to achieve a gradual upgrade path but places a restriction on the usable signal bandwidth. The termination tuning will also introduce phase and amplitude distortion.

Chapter 10 sets out the data, bandwidth and response time requirements for the track circuit signal. Traditional modulation schemes are described, and the synchronisation/error control redundancy is discussed. In the following two chapters (chapter 11 and 12), two separate modulation and coding schemes are proposed which utilise the available bandwidth efficiently and aim to fulfil the data requirements set out in chapter 10. An MFSK modulation scheme is proposed in chapter 11, with DPSK information encoded onto each symbol. Non-coherent demodulation is described and methods of error detection/correction are proposed. It is shown, however, that the required signal bandwidth exceeds the maximum allowed bandwidth quoted in chapter 10, and so the performance and error control strategies of this scheme have not been studied further.

A novel parallel tone modulation scheme is described in chapter 12, which requires no symbol or code synchronisation. This scheme is chosen for further study and simulation of the coding and demodulation in chapter 13.

10. Modulation Schemes for Track Circuits

This chapter will first state the data, time and bandwidth requirements for the modulation scheme. The factors affecting the maximum possible data capacity (such as track circuit response time, code synchronisation and error control redundancy) and the implications of non Gaussian noise are discussed. Section 10.2 describes some of the traditional modulation schemes which can be considered for use in track circuits and compares their performance with respect to these requirements.

10.1 Data Requirements

The table below shows the present view of the requirements of the modulation/coding scheme [52]. Note that the detection times are higher than the rejection times. For the track circuit receiver, the rejection time is traditionally called the 'drop' time, and the detection time is called the 'pick' time. Safety dictates that it should take longer to switch from 'track occupied' to 'track clear' than from 'track clear' to 'track occupied' and so the pick time is always greater than the drop time. The different types of data have different detection and rejection times. The track to train data, and local track circuit identity are safety critical data and must be decoded within the detection time of existing systems (≈1.2s for the Westinghouse FS series of track circuits). The full track circuit identity will be unique to each individual track circuit on a particular network. It may also be used to transmit ATO data such as track gradient, and other location specific information.

Data type	Number of states		Number of bits required		Detection time (s)		Rejection times (s)	
	Minimum	Desirable	Minimum	Desirable	Maximum	Desirable	Maximum	Desirable
Local track circuit identity	30	100	5	7	0.5	0.25	0.3	0.1
Full track circuit Identity	10,000	1,000,000	14	20	20	5	5	1
Track to train data	60	100	6	7	1	0.25	0.3	0.1
Total number of bits			25	34				

Table 10.1. Track Circuit Data and Response Time Requirements

The tuned area of the existing termination (figure 2.2) limits the bandwidth to no more that 200Hz.

The data will be encoded into a block code and a full code word must be received and decoded within the track circuit response time. When a train enters a track circuit, the train receiver has no knowledge of coding phase (i.e. the first bit/symbol received could be anywhere in the code word). Considering the worst case, if the first received bit is the second bit of the code word, and n is the code word length, then the receiver must receive $(2n\text{-}1)$ bits before it sees the first valid code word. This figure increases to $(3n\text{-}1)$ if two code words must be received before a decoding decision is made.

The track circuit code must be *comma free*[1]. Comma freedom requires at least 50% redundancy (i.e. in the code word, only half of the bits can be used for information [77]). For reliable code synchronisation and error control the redundancy needs to be increased to approximately 67% (i.e. a code rate of 0.33) [2][82][83].

10.2 Description of Digital Modulation Schemes

The demodulation of digital carrier schemes can generally be divided into two categories: coherent and non-coherent. The optimum detector assumes coherent detection, where the receiver has a local carrier wave-form in exact phase with the received carrier. In many systems, hardware implementation of coherent detection involves complex receiver design, and so non-coherent detection systems can be of importance.

For multilevel systems (M>2) the information rate is $\log_2 M$ times greater than for the binary system and so the required bandwidth can be scaled down by the same ratio.

Bit error probabilities are traditionally quoted for AWGN and are plotted against E_b/N_o (energy per bit/noise power per Hz) (Note $E_b = A^2T_b/2 = A^2/r_b 2$, where A is the peak carrier voltage, T_b is the information bit duration and r_b is the information rate. For M>2 the SNR is scaled by $\log_2 M$).

More detailed descriptions (i.e. signal construction, derivation of P_e etc.) of the schemes considered are found in [81][90][94].

It is known that Gaussian noise is not the limiting factor for low to medium data rate systems, and this is true for the track circuit environment. The comparison with AWGN is convenient for analytical expressions but impulsive, bursty or single tone interference will change the performance greatly, when SNR is compared, and so the P_e figures quoted may have limited

[1] Comma freedom dictates than no overlap between any two code words (including the same code word) will contain a bit sequence that is the same as any code word.

meaning. Impulsive (or bursty) noise will tend to block out the signal completely for the duration of the impulse ($P_e \approx 0.5$) and have low P_e for other times.

10.2.1 Binary PSK

10.2.1.1 Coherent Detection

Coherent detection requires carrier recovery, which increases the receiver complexity and the time to acquire a usable signal. The bit error probability is given by the equation:

$$P_e = Q\left(\sqrt{\frac{A^2 T_b}{\eta}}\right) \tag{10.1}$$

where $\eta/2$ is the 2 sided noise psd and $Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy$. The signal to noise ratio for $P_e = 10^{-4}$ is 8.45dB. The bandwidth required is approximately $2r_b = 2/T_b$.

10.2.1.2 Differentially Coherent Detection (DPSK).

Differentially coherent detection uses the phase of the previous symbol as a reference for demodulating the current symbol. The information must be differentially encoded at the transmitter. For the binary case (M=2) the DPSK scheme requires only about 1dB more power than the coherent scheme. It is worth noting that successive symbol errors are no longer independent and bit errors tend to occur in pairs.

The bit error probability is given by the equation :

$$P_e = \exp\left(\frac{-A^2 T_b}{2\eta}\right) \tag{10.2}$$

where $\eta/2$ is the 2 sided noise psd. The signal to noise ratio for $P_e = 10^{-4}$ is 9.3dB.
The bandwidth required is approximately $2r_b$.

10.2.2 QPSK

10.2.2.1 Coherent Orthogonal QPSK

The in-phase and quadrature bit streams that make up the QPSK signal are orthogonal, and so they can be demodulated independently. Hence the bit error probabilities of coherent BPSK and QPSK are the same [94].

The bandwidth required is approximately r_b (half that of BPSK).

10.2.2.2 Offset QPSK

In the OQPSK scheme, the 2 independent bit streams that make up the quadrature signal are staggered so that both bit streams cannot change state simultaneously. Both QPSK and OQPSK have the same spectral shape associated with the rectangular pulse.

OQPSK performs better than some other modulation schemes when undergoing bandlimiting and hard limiting operations since phase transitions of 180° are avoided.

10.2.3 Non-Coherent Orthogonal QPSK

This is known as differentially coherent QPSK (DQPSK). The bit error probability is given by the equation :

$$P_e = 2Q\left(\sqrt{\frac{A^2 T_s}{\eta} 2\sin^2\left(\frac{\pi}{2M}\right)}\right) \qquad (10.3)$$

where M=4 for DQPSK. The increase in power requirement compared to coherent QPSK is about 2 dB (to achieve the same P_e)(p425 [81]).

The bandwidth required is approximately r_b (half that of BPSK). The signal to noise ratio for P_e=10^{-4} is (approximately) 10.5dB.

10.2.4 Binary FSK

10.2.4.1 Non-Coherent Orthogonal FSK

Since the information is contained in frequency and not phase, a phase discontinuity at the symbol transitions will not affect the performance of a non-coherent demodulator. Non-coherent demodulation leads to a much simpler receiver design than coherent demodulation. The signals are orthogonal if:

$$2f_d T_b = m \quad \text{where } m \text{ is any integer} \qquad (10.4)$$

The minimum frequency separation is $2f_dT_b$=1, which is twice the separation needed for coherent demodulation (e.g. MSK). The demodulator is implemented with matched filters tuned to each frequency followed by envelope detection. A bit decision is made on the output of the envelope detectors. The bit error probability with Gaussian noise is shown below :

$$P_e = \frac{1}{2}\exp\left(-\frac{E_b}{2N_o}\right) \qquad (10.5)$$

This scheme has the advantage of very simple receiver and transmitter design, and has been previously applied to track circuits [2][92].

An alternative implementation of the receiver would be through the use of a limiter and discriminator (p228 [90]). If this is assumed to be a perfect instantaneous frequency discriminator then the performance of this demodulation is only 0.5dB below the equation above, if $2f_dT_b \approx 1$.

The bandwidth required is $>2r_b$.The signal to noise ratio for $P_e=10^{-4}$ is 15.33dB.

10.2.5 Minimum Shift Keying (MSK) (Coherent Orthogonal FSK with signal frequency separation of $1/2r_b$)

MSK can be represented as a special case of OQPSK, with sinusoidal pulse shapes of length $2T_b$. The pulse shaping achieves a constant envelope (i.e. no sudden phase transitions). It is however more commonly thought of as a continuous phase binary FSK signal with a signal frequency separation of $1/2r_b$ (the minimum separation required for orthogonal coherent detection).

The bandwidth required to transmit 99% of the total signal power is approx. 1.2 r_b (compared to 8 r_b for QPSK and OQPSK[2]) although the bandwidth required for no inter-symbol interference (ISI) is often quoted as 1.5 r_b. The spectral sidelobes are about 10-20 dB lower than QPSK or OQPSK and so the scheme performs better when bandlimited.

Since MSK is a type of FSK, it can be demodulated non-coherently (e.g. a frequency discriminator). The SNR penalty however is severe.

With coherent detection, the bit error rate is the same as coherent BPSK (and QPSK, OQPSK) (equation 10.1). The signal to noise ratio for $P_e=10^{-4}$ is 8.45dB.

10.2.6 MFSK

This scheme is an extension of binary FSK, to M different symbols or frequencies. The performance of an MFSK system with AWGN improves as the number of symbols is increased. MFSK schemes have been used to conserve power at the expense of bandwidth (p427, [81]). This is the opposite of MPSK schemes which can be used to conserve bandwidth at the expense of power (for the same error probability). The probability of error is given by:

$$P_e = \sum_{r=1}^{M-1} \frac{(-1)^{(r+1)}(M-1)!}{(r+1)!(M-r-1)!} \exp\left(\frac{-r}{r+1}\frac{A^2 T_b}{2\eta}\right) \qquad (10.6)$$

The normalised SNR required of $P_e=10^{-4}$ is 12.3, 9.8, 9, 8.4, 7.5, and 6.7dB, for M=2, 4, 6, 8, 16, 32 respectively.

The normalised bandwidth (see section 11.3) for different values of M is shown below[3]

---

[2] T is the information bit rate, not the symbol rate

[3] The occupied bandwidth is assumed to extend 1/(one symbol interval) Hz beyond the extreme tone frequencies. In this case, the bandwidth contains 99% of the signal power (p36, [95]).

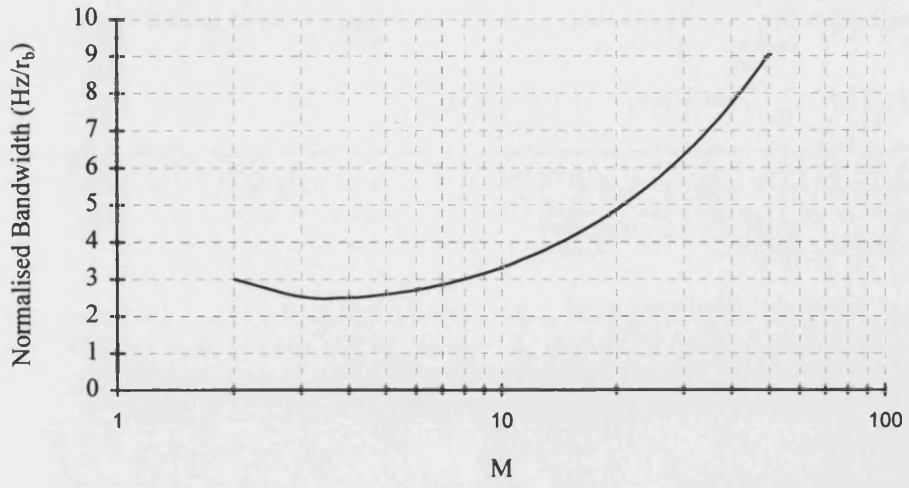


Figure 10.1. Normalised Bandwidth for MFSK

It can be seen that the bandwidth is minimised to $2.5r_b$-$2.8r_b$ for $3<M<6$. Beyond this, the bandwidth increases with M.

10.3 Comparison of Schemes

Table 10.2 shows the bandwidth and power requirements for constant bit P_e for the schemes considered. MSK, DQPSK and OQPSK seem strong contenders for a modulation scheme. MSK has good error performance and bandwidth efficiency, but a more complex receiver structure. DQPSK has a slightly inferior performance but less complex equipment can be used. Figure 10.2 shows the normalised frequency spectrum for MSK, BPSK and QPSK. It can be seen that MSK has a wider main lobe but the side lobes fall off much faster. For this reason, MSK is much more spectrally efficient than QPSK (or OQPSK).

10.4 Conclusions

Taking into account the code synchronisation factors mentioned in section 10.1, a signal bandwidth of 200Hz and a response time of, say, 0.3 seconds, let's assume the modulation scheme requires a bandwidth of r_b Hz (the best case, from the schemes considered). At 200 bits/second the receiver can 'see' 66 bits in 0.3 seconds. The maximum code word length is (66-1)/2 = 32 bits (for the receiver to decode a single code word, with no initial knowledge of coding phase). If the code word contains 33% redundancy to achieve reliable synchronisation, the maximum number of information bits will be 10 (this is based on the reception of a single code word and ignoring any delay in establishing a carrier phase reference and processing time after the code has been received).

Modulation	Bandwidth Required (Hz)	SNR (dB) for $P_e=10^{-4}$	Rx/Tx Complexity	Comments
BPSK (coherent)	$2r_b$	8.45	Large	Signal has 180° phase transitions
DBPSK (non-coherent)	$2r_b$	9.3	Medium	Bit errors may occur in pairs
QPSK (coherent)	r_b	8.45	Large	
OQPSK	r_b	8.45	Large	Performs better than QPSK with bandlimiting
DQPSK	r_b	10.5	Medium	Bit errors may occur in pairs
BFSK (non-coherent)	$>2r_b$	15.33	Small	
MSK (coherent)	$1.2\ r_b$ - $1.5\ r_b$	8.45	Large	
MFSK (non-coherent)	M=2 : $3\ r_b$ M=3 : $2.5\ r_b$ M=4 : $2.58\ r_b$ M=8 : $3.1\ r_b$ M=16 : $4.6\ r_b$	M=2 : 12.3 M=3 : 10.2 M=4 : 9.8 M=8 : 8.4 M=16 : 7.5	Small	FFT techniques can be used for receiver

Table 10.2. Comparison of Modulation Schemes

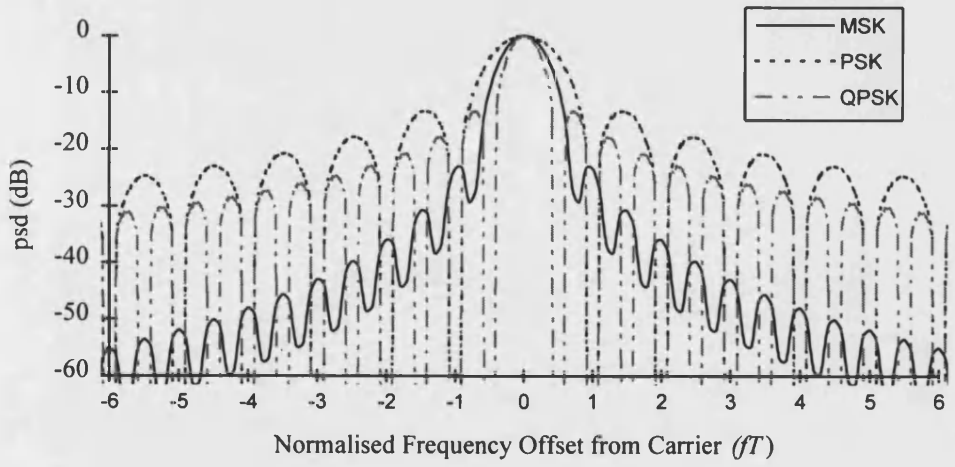


Figure 10.2. Normalised Frequency Spectrum of MSK, PSK and QPSK

Table 10.1 states a minimum data requirement of 25 bits and a maximum of 34 bits [52]. With a bandwidth of 200Hz and a signal occupancy of 1-$2r_b$ this may initially seem

achievable. However, when the comments above are considered the effective data rate is lowered and these requirements are not achievable.

With unusual interference (i.e. impulsive or single tone) the error performance figures of table 10.2 may well be meaningless and noise immunity will be provided by methods other than the modulation scheme alone (i.e. coding, redundancy processing). Factors such as equipment complexity and cost may be more important. For example, non-coherent FSK and MFSK can be implemented with fairly simple DSP and FFT receiver structures while MSK requires a complex fully coherent receiver.

If the data requirements are to be met within the 200Hz restriction, a more bandwidth efficient modulation scheme is needed than is offered by serial binary modulation.

In chapters 11 and 12, two separate modulation schemes are proposed, which aim to fulfil the requirements within the time/bandwidth restrictions. Table 10.1 contains different detection times for different types of data (e.g. 0.25s for train data and 5s for 'full track circuit identity'). The data capacity of a scheme could be increased by taking advantage of these different times and using a different/separate encoding and detection philosophy for the different types of data. This is discussed for the proposed modulation and coding schemes of chapters 11 and 12. The scheme proposed in chapter 12 has been chosen for a thorough study of the demodulation and decoding in chapter 13.

11. MFSK for Track Circuit Modulation

This chapter presents an initial investigation into the parameters and potential demodulation techniques of an MFSK modulation scheme for track circuits. To increase the data capacity of the scheme, within a fixed system response time and bandwidth, it is proposed that DPSK (or DQPSK) be applied to each MFSK symbol. Since the MFSK tone frequencies are detected non-coherently, DPSK encoding will not affect the MFSK signal, and the two schemes can be considered independently. Different encoding/decoding philosophies can then be applied to the two schemes which can carry different types of data (table 11.1).

It is proposed that the MFSK signal shall be encoded with the safety critical data (the 'track to train data' and the local 'track circuit identity' - see table 10.1), and the DPSK encoding (discussed in section 11.7) shall be used to represent the 'full track circuit identity', which has a longer detection time.

When combining the analysis with the data requirements (section 11.5), it will be shown that the required bandwidth for the MFSK signal will exceed the 200Hz restriction quoted in [52], for the existing tuned terminations. For this reason, the scheme has not been chosen for further investigation. This chapter is presented, however, since an increase in the maximum signal bandwidth, or a decrease in the required data capacity may occur in future track circuit developments. Also, the methods presented in this chapter, for error control and code synchronisation suitable for a *M*-ary signalling scheme, are conceptually simple and flexible with varying data requirements.

11.1 Introduction

This scheme is an extension of binary FSK, to *M* different frequencies or symbols. It has been proved that the performance of MFSK systems improves as the number of symbols is increased, but that the improvement is negligible for large values of *M* [96]. Despite this, binary systems are still the most popular choice, for reasons of familiarity and availability. Also, most work on error control coding and synchronisation has been aimed toward the binary symmetric channel (BSC). In the design of an MFSK scheme the data source itself must be taken into account, especially for higher values of *M*.

MFSK schemes have been used to conserve power at the expense of bandwidth (p427, [81]). This is the opposite of MPSK schemes which can be used to conserve bandwidth at the expense of power (for the same error probability). Optimum detection of MFSK is achieved when the *M* different tone frequencies are orthogonal to each other, with respect to the observation interval. This means that the symbol length must be an integer multiple of the

reciprocal of the tone spacing. The amplitude and length of each symbol is kept the same (p14, [95]) and if the phase is continuous at symbol transitions, the signal will have a constant envelope. This is desirable in this application for keeping the rail to rail voltage high at all times.

11.2 Error probability

In the case of white Gaussian noise, the probability of symbol error for an MFSK system has been shown to be ([95] p31):

$$p = \sum_{r=1}^{M-1} \frac{(-1)^{(r+1)}(M-1)!}{(r+1)!(M-r-1)!} \exp\left(\frac{-r}{r+1}\frac{A^2T}{2\eta}\right) \qquad (11.1)$$

where M is the number of tones, A is the peak signal level, T in the symbol length and η is the 2 sided noise psd. The graph below shows the error probabilities for $M = 2, 4, 6, 8, 16, 32$.

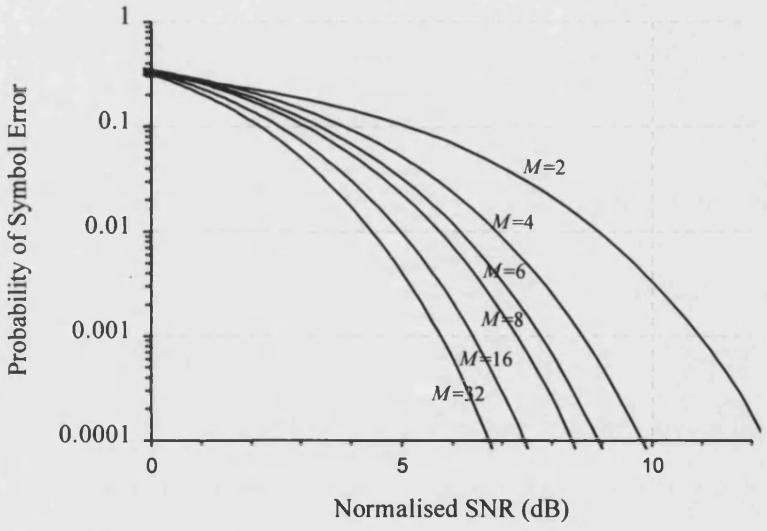


Figure 11.1. Probability of Error for MFSK

Note that for each curve the x axis is normalised to be: signal energy per bit/noise power per Hz bandwidth = SNR / $\log_2 M$.

These curves show that, when systems of equal data rate are compared, a higher value of M provides better protection against noise. This is because the symbol length is increased and so the receiver can average out the effects of interference and ISI.

11.3 Occupied bandwidth

The normalised bandwidth is defined as

$$B_n = \frac{\text{Occupied Bandwidth}}{\text{Data rate}} \text{ Hz / bits / s} \qquad (11.2)$$

If the signal bandwidth is assumed to occupy G/T Hz beyond the extreme tone frequencies, where T is the symbol length, then the normalised bandwidth is given by

$$B_n = \frac{(M-1+G)T}{T\log_2 M} \text{ Hz / bit / s} \qquad (11.3)$$

For the worst case, lets consider an binary FSK signal switching between the two extreme tone frequencies. Since the ratio of f_d/r_b is high (where $2f_d$ is the distance between the tones, and r_b is the bit rate), the signal can be considered as two interleaved ASK waveforms (p410, [81]). The bandwidth requirements for an ASK signal are approximately $2r_b$. This means that the required bandwidth will extend r_b Hz beyond both the highest and lowest tone frequencies. If the signal is bandlimited to $1/T$ Hz beyond the extreme tone frequencies (i.e. G=1), then the resulting signal spectrum will contain 99% of the original signal power. In this case, the normalised bandwidth (Hz/bits/s) for different values of M is shown below.

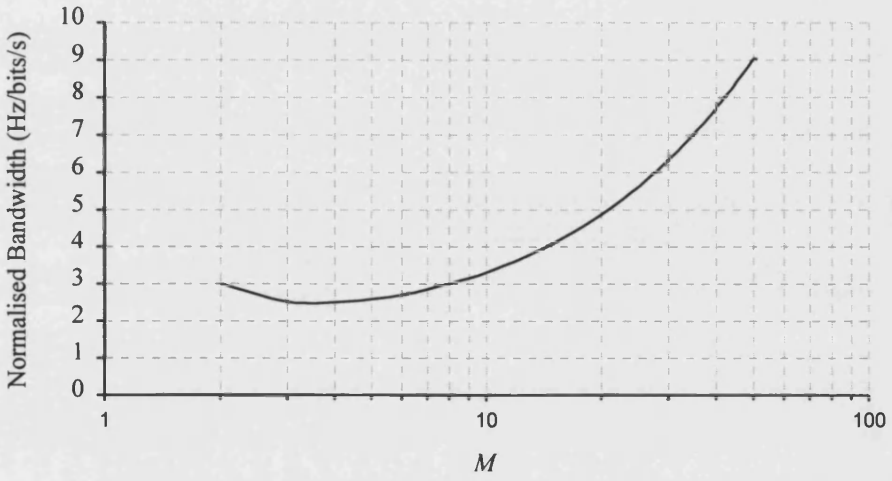


Figure 11.2. Normalised Bandwidth for MFSK Signal

It can be seen that the bandwidth is minimised to $2.5r_b$-$2.8r_b$ for $3<M<6$. Beyond this, the bandwidth increases with M.

11.4 Error Control

For a block code, some redundancy will be required for error detection. Also, to improve the reliability of the decoding, some error correction may be used. The use of error correction however, will be restricted by the safety requirement of track circuits. As with synchronisation, the majority of the useful work on error control codes and commercial error coding equipment has been aimed at the binary channel [95]. It can easily be demonstrated that binary coding techniques applied to an M-ary channel are largely ineffective, especially as M is increased. For example, consider an M=32 scheme. Each

symbol could be used to represent 5 bits of information, encoded and decoded with a binary coding scheme. A symbol error occurs when the receiver chooses any of the (M-1) symbols which do not correspond to the transmitted symbol. If a single bit error correcting code is used, then for correct decoding to occur, the demodulated M-ary symbol must represent a 5-bit sequence that is a distance of 1 from the correct sequence. This occurs in only 15.6% of the symbol errors.

11.4.1 M-ary Parity Check - Error detection

This technique is similar to the binary parity check, where one bit is added to a word. In the multi-level case, each symbol is allocated a weight, from 0 to (M-1), and the weights of each symbol in a sequential block are added, modulo M. The resulting symbol is added to the block. The same process is performed at the receiving end and the calculated parity symbol is compared to the received parity symbol. The expression for the probability of a data word of n (M-ary) symbols containing undetected errors is given by P_{uf} below [97]:

$$P_{uf} = \frac{M-1}{M}\left(1-\frac{Mp}{M-1}\right)^n + \frac{1}{M} - (1-p)^n \qquad (11.4)$$

where p is the symbol error probability. Figure 11.3 shows this error probability for different values of M, with blocks of comparable *information* content.

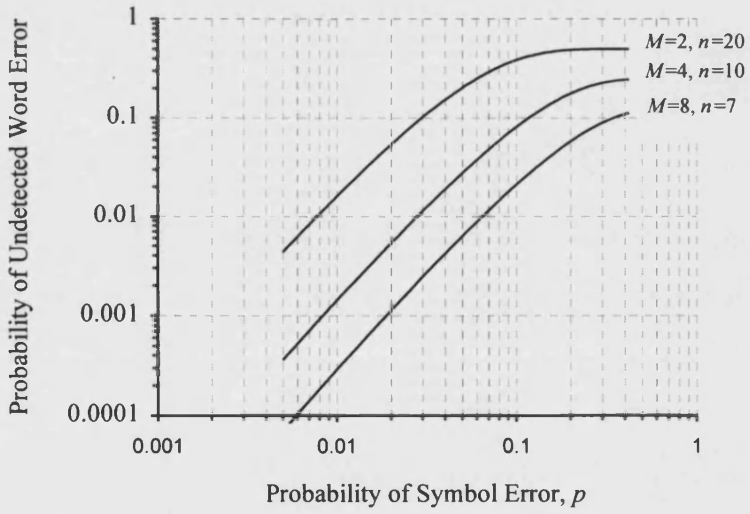


Figure 11.3. Probability of Undetected M-ary Parity Error

It can be seen that the error checking is more effective for higher values of M. If the SNR is used as the x axis, the improvement of the error detection with M would be even more marked, since the symbol error probability decreases as M is increased. If the decoding process is fed with random data, that is, overwhelming noise, the check symbol has a probability of error of $p = (M-1)/M$ and

$$P_{uf} = \frac{1}{M} - \frac{1}{M^n} \qquad (11.5)$$

It can easily be seen that a higher M produces a higher detection probability with random input data, since the parity symbol has only a $1/M$ chance of producing the correct parity result.

For higher SNR, an M-ary parity check on a smaller block length will produce a lower error probability, since fewer errors are expected. The number or position of the parity bits is not restricted by the data word size. For example, in a data block of 10 symbols, one parity symbol can be inserted after the first five symbols, and another after the last five.

11.4.2 Error Correction

11.4.2.1 M-ary Parity Check Error Correction Algorithm

A relatively simple method of error correction using the M-ary parity check is described as follows: one complete block of a code with a parity check symbol is repeated, giving two separate versions. If only one version satisfies the parity check, then this is output. If neither version satisfies the check then the two blocks are compared and where they differ, the symbols are swapped, and the parity check is reapplied [95]. This is an obvious application for soft decision decoding [98], since it would make sense to swap the least reliable symbol first (i.e. the symbol that is most corrupted by noise). A method of obtaining this reliability data is prescribed in [104]. We will assume for now that we can rank the symbols in order of their reliability and that the least reliable symbol is swapped. The process is repeated, swapping each symbol in order of increasing reliability until one or both of the parity checks are satisfied. A limit on the number of symbols swapped will provide a compromise between undetected error and error detection (unsafe failure and safe failure). For example, the number could be limited to 2, so that if, after the second symbol is swapped, both parity checks do not agree, the receiver will indicate a safe failure.

There are many possible coding philosophies that can be applied to this error correction scheme to optimise both safety and reliability. Some possibilities are shown in figure 11.4. For track circuits, the track code will remain the same for the majority of the time, since traditional fixed block signalling will transmit the same track circuit identity and speed code while a train is occupying a track section. However, for advanced ATP/ATO functions it may be required to change the code while a train is within a track section (see section 4.4). In the example in figure 11.4 the first three code words are the same, after which the code word changes. Undetectable errors occur in the fifth code word.

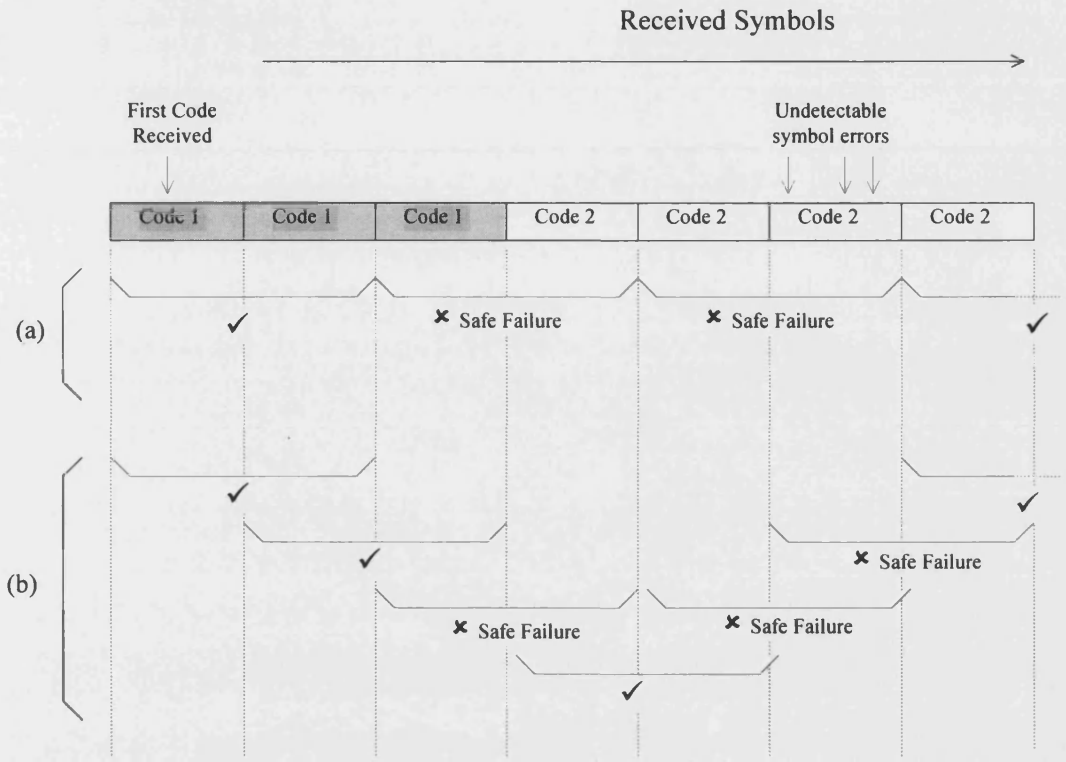


Figure 11.4. Decoding Philosophies for MFSK Code with M-ary Parity Check Error Correction

Decoding philosophy in (a) decodes separate *blocks* of two consecutive code words, as described above. The second block contains two different code words and a safe failure will occur, since the decoder requires that the code words be the same. This increases the receiver response time to a changing code word. A safe failure occurs in the third block, since the two code words in this block are different. In (b), each consecutive *block* overlaps by one code word which results in a smaller coding delay with a code word change, for the example shown.

A derivation of the theoretical performance of this error correction scheme would be complicated, especially when the decoding philosophy is considered. The performance would be best evaluated by simulation. The error correction scheme will work well with both random and bursty interference. If an interference burst were to corrupt most of one code, but leave the next code error free, then the errors will be corrected. There are also many correctable random error patterns, since an uncorrectable error requires symbol errors in the same symbol of both code words.

The error correction method has the advantage that it can be applied to any block size and value of *M*, and so the MFSK parameters can be optimised for others factors.

11.4.2.2 Reed Solomon Codes.

Reed Solomon (RS) codes can be described as BCH codes extended to non-binary symbols [55]. A t error correcting RS code with symbols from GF(2^m) is described as an (n,k) code, where $n=2^m-1$, k = $n-2t$ and $d_{min}= 2t + 1$. If the symbols are transmitted in their binary equivalent form the RS codes give efficient burst error control. In the MFSK case however, we are *transmitting* M-ary symbols, where $M = 2^m$.

As an example, consider the RS code defined in GF(2^4). If we reserve one symbol for frame synchronisation our modulation scheme requires M=17. If we require the code to correct one symbol error and detect up to three symbol errors, then $d_{min} = 5$, $n = 15$ and $k = 11$. A single code word represents 44 bits of information.

The choice of M is no longer independent of the coding scheme. For smaller values of M the code length and data capacity is limited. A larger value of M provides more flexibility since a long RS code can easily be shortened by not transmitting some of the symbols and assuming that they are zero at the receiver (Appendix F). Increasing M, however, increases the required bandwidth.

Decoding philosophies similar to those described in section 11.4.2.1 can be used to increase the safety of the overall scheme. RS codes are efficient in terms of the minimum redundancy required for error correction. However, a decoding philosophy requiring the reception of two valid code words will increase the effective redundancy to a level higher than that of the M-ary parity check error correction described above. No attempt is made to evaluate or compare the performance of the two coding schemes described in this chapter. A simulation would require careful construction, since the different modulation, coding, data rates and response times would be compared.

It is likely that, since the overall coding scheme must be biased towards error detection that the use of powerful error correction codes will provide little advantage over the simpler scheme described in section 11.4.2.1. The extra complexity and memory/equipment cost of the RS decoder must also be considered. The following evaluation of the MFSK system will assume that M-ary parity check error correction has been used.

11.5 Initial Analysis of MFSK parameters.

11.5.1 Data Requirements

We start by restating the initial data requirements and response times for the different types of data, in table 11.1 [52]:

Data type	Number of bits required		Detection time (s)		Rejection times (s)	
	Minimum	Desirable	Maximum	Desirable	Maximum	Desirable
Local track circuit identity	5	7	0.5	0.25	0.3	0.1
Full track circuit Identity	14	20	20	5	5	1
Track to train data	6	7	1	0.25	0.3	0.1

Table 11.1. Data Capacities and Detection Times.

If the signal is to be fed through existing tuned terminations the bandwidth is restricted to ≈ 200Hz, depending on the type of termination used.

Initially, consider only the local track circuit identity and the track to train data (the full track circuit identity will be considered in section 11.7). If binary encoding is used, they both require 7 bits to achieve the desired number of states. If we assume they both require a detection time of 0.3 seconds then for 14 bits to be received, an information rate of 47 b/s is needed.

11.5.2 Frame Synchronisation

The nature of track circuits dictates that each frame should be self synchronising. This requires extra information in the available time, which cannot be used for data. Practically all work on self synchronising codes has been aimed at the binary channel. For example:

- Reference [2] describes a comma code specially constructed for track circuits. The code consists of a 13 bit synchronisation sequence prefixed to a (15,11) single error correcting Hamming code. The block length is 28 bits, of which 11 are used for data (there are in fact fewer data bits since certain code words are not used as they are too close to the synchronisation sequence.

- Cyclic coset codes are an alternative to comma codes [77][82]. It has been proved that a cyclic code can be made comma free if $n > k/2$, where n is the block length and k is the number of information bits.

We can generally say that, for block synchronisation requirements, at least 50% redundancy is needed (or code rate = $k/n = 0.5$), halving the time available for usable data. This figure depends on how robust the synchronisation is. A possibility for an MFSK signal is to reserve one symbol for block synchronisation, a principle which has been used in binary

digital telemetry systems, where a third symbol is reserved for synchronisation. The loss of data capacity can be justified because each block must be reliably self synchronising and so the block synchronisation is just as important as the data content. Also, 50% redundancy for synchronisation represent a greater loss of capacity than reserving one symbol. The reliability of this method would have to be studied.

11.5.3 Bandwidth Required

For the methods of synchronisation and error control described in sections 11.5.2 and 11.4.2.1, and the initial requirement (section 11.5.1) of 14 bits of information in 0.3 seconds, the bandwidth required for various values of M is shown below:

M	(M-1). Reserve one tone for sync.	Information content of one symbol (bits)	Number of symbols for 14 bits	+2 symbols (for parity and sync)	(x 2). Repeat block for f.e.c.	Symbol length for detection time of 0.3 s	min signalling rate for orthogonal spacing	Bandwidth = (M-1) . baud rate
32	31	4.95	3	5	10	0.03s	33.33 bd	1033.3 Hz
16	15	3.9	4	6	12	0.025s	40 bd	600 Hz
8	7	2.8	5	7	14	0.021s	46.67 bd	326.67 Hz
4	3	1.58	9	11	22	0.0136s	73.33 bd	220 Hz

Table 11.2. MFSK Tone Spacing and Bandwidth Requirements for Different Values of M.

Note that it is not necessary to have M as a power of 2. The bandwidth of the last column is the distance between the highest and lowest tone frequency. The bandwidth required for reliable transmission (99%) of the total signal power is shown below.

M	Bandwidth
4	366 Hz
8	419 Hz
16	680 Hz
32	1100 Hz

It can be seen that the signal bandwidths will be larger than the 200Hz restriction quoted in [52]. However, since the data requirements and bandwidth restriction may be relaxed in future track circuit developments, the following potential methods of demodulation, symbol synchronisation and DPSK encoding are briefly discussed.

11.6 Demodulation and Symbol Synchronisation.

Since the orthogonal MFSK signal tone frequencies are spaced at regular frequency intervals and the symbol envelope is rectangular, demodulation is most conveniently performed by the DFT, which is equivalent to a bank of matched filters. Assuming the start

of the symbol is known, a DFT is applied to the signal, whose duration is equal to the symbol length. The DFT bin with the highest magnitude squared corresponds to the symbol tone which is assumed to be the transmitted symbol.

For an audio frequency track circuit signal it is possible for one large DFT to be taken, extending from d.c. to the tone frequencies. Alternatively, to save memory and decoding time, the signal can be sampled, heterodyned to baseband and decimated. In this way a smaller DFT can be taken, concentrating on the frequencies of interest. These techniques are discussed in [103]. Since the symbol lengths discussed in the previous section are of the order of ms, processing power should not be a major constraint [101].

A method is required by which the receiver can separate the incoming waveform into symbols. This can be achieved continuously by including synchronisation information with the transmitted data, or by inserting synchronisation information between data blocks. These methods have previously been employed in MFSK systems but they demand extra complexity and overhead penalties in terms of power and bandwidth [95]. A method of obtaining the MFSK synchronisation information from the data signal itself, using a technique known as Modulation Derived Synchronisation (MDS) is proposed in [104]. DFT results for the frequency bins corresponding to the symbol frequencies are required for every sample, to enable the point of synchronisation to be identified. Appendix C describes a DFT update routine which can be used for only the frequency bins of interest. This avoids calculating redundant information. Three complex multiplications and three additions are required to update each bin. The routine is applied sample by sample and preserves the phase information of each symbol.

11.7 Applying DPSK to the Symbol Tone.

The 'full track circuit identity' data requirement of section 11.5.1 requires up to 20 bits of information, but the allowed detection time is greater. If each tone was also modulated with a binary DPSK signal then the full track circuit identity could be encoded onto the signal, independently of the other information. Since the 'full track circuit identity' requires more information bits than the 'local track circuit identity' and the 'track to train data', the block code length of the DPSK data could be made larger than the MFSK block length. The MFSK symbol synchronisation can also be used for the PSK synchronisation. Once symbol synchronisation has been achieved, it can be taken as the reference phase for demodulating the PSK information.

Differential PSK can be thought of as non-coherent PSK, since a phase coherent carrier is not needed at the receiver [81]. The phase of each symbol is encoded with respect to the

previous symbol and so a small amount of error between the transmitter and receiver sampling clocks can be tolerated.

A possible disadvantage of this scheme is that the envelope will pass through zero at 180 degree phase shifts. However, if these envelope transitions are fast at the wheel-rail contact (i.e. if the bandwidth is sufficient) then the axle should still provide a short circuit at the symbol transitions.

11.7.1 DPSK Parameters.

The DPSK symbol length will be the same as the MFSK symbol length. The data rate increases as the symbol length decreases. From table 11.2, this corresponds to a smaller value of M.

From table 11.2, the desirable number of full track circuit identities is 1,000,000, which corresponds to 20 information bits. The method of code synchronisation will not be prescribed now. The MFSK code synchronisation could be used, but if the PSK block length is longer than the MFSK length, extra information will be required. We will assume that the synchronisation requirements and the error correction will double the number of bits[1]. Since the coding phase is unknown as a train enters a track circuit the worst case total number of bits to be received within the detection time is again doubled. This would occur if the first bit received is the second bit of the DPSK block code. These factors raise the total number of bits to be received to 80 bits. Using table 11.2, the time for the reception of 80 symbols (MFSK symbol ≡ PSK bit) for different values of M is shown in table 11.3.

M	Symbol Length	Reception Time for 80 Symbols
32	0.03s	2.4 s
16	0.025s	2 s
8	0.021s	1.68 s
4	0.0136s	1.09 s

Table 11.3. Reception Time for DPSK Data Block

For all values, the detection time is within the desirable figure of 5 seconds.

[1] It has been shown that, with comma free serial binary coding, a redundancy of 0.5 can provide error control and comma free synchronisation [77][82]. Less redundancy may actually be needed, since the MFSK block code synchronisation is available.

11.8 Conclusions

This chapter has proposed an MFSK scheme for track circuit data. MFSK is initially attractive because simple non-coherent demodulation can be employed and the signal has a constant envelope, which is desirable for a track circuit signal. The symbol tones are orthogonal to each other and a higher value of M requires a higher bandwidth (for the same data rate). It has been demonstrated that the error performance improves with higher values of M, in the presence of AWGN. Since MFSK modulation is generally known as a wide-band scheme, and exchanges bandwidth for error performance in AWGN, the encoding of the symbol phase with DPSK information will make more efficient use of the available bandwidth. The ultimate limitation, however, is the track circuit response time. It has been seen that the MFSK encoding of the 'track to train data' and 'local track circuit identity' (table 11.1) requires a signal bandwidth higher than the 200Hz restriction of the existing tuned terminations [52]. For this reason, the proposed synchronisation and error correction schemes have not been investigated further. It is, however, possible that the bandwidth restriction may be relaxed in the future, since the track circuit designs can differ, and the bandwidth of the termination tuning circuit is controlled by a simple damping resistor. It is also possible that, in future developments of the track circuit, the required data capacity may decrease, and since the proposed methods of synchronisation and error control are conceptually simple and flexible, this chapter is included as a proposition for future simulation or development.

The decoding procedures for the MFSK and DPSK information are independent of each other. Since the 'full track circuit identity' is not safety critical, the detection time is much greater and it is proposed that this information is DPSK encoded. It could also be possible to use DQPSK to increase the data capacity.

Listed below are some of the factors affected by the choice of M:

Increase M

Better noise immunity ✓

Wider Bandwidth ×

More effective error detection/correction for MFSK ✓

Decrease PSK data rate (since the symbol length is longer) ×

Since there has been virtually no previous work on code synchronisation for $M>2$, the proposed method of reserving one symbol for *code* synchronisation will demand less overhead than including a synchronisation sequence in the data, and will not limit the choice of M. Simulations will have to determine the safety and reliability of this method. It is

possible to extend this method to a synchronisation *sequence* which includes the synchronisation symbol. If the DPSK information block is longer than the MFSK block, extra synchronisation information will have to be included in the DPSK block code. This could be achieved using more traditional binary code synchronisation techniques, such as those presented in chapter 8.

The M-ary parity check error control method described in section 11.4.1 is simple, is applicable to all values of M and code length and is more effective with higher values of M. The use of RS code for MFSK code is also discussed, since the code is defined in a non-binary field. With this code, the choice of M is no longer independent, due to the restrictions that $n=2^m-1$ and $M=2^m$. No attempt is made in this chapter to compare the performances of these two schemes. However, it is likely that, since the error control must be biased towards safety (error detection), the performance advantage from using the RS code is negligible or non-existent, and that the use of more complicated codes designed for efficient error *correction* are inappropriate for track circuits. This is also suggested by the coding performance results in chapter 13 (see also p155 [95]). It must also be noted that less complex algorithms and equipment will affect the overall system safety as well as the cost.

12. A Multi-tone Modulation Scheme Requiring no Synchronisation.

This chapter proposes a track circuit modulation scheme that consists of tones spaced at regular intervals. The signal requires no carrier, symbol or coding synchronisation and so simple demodulation can be achieved through the FFT. The signal can be viewed as an extension of the track circuit rate coded signalling scheme, but information is encoded onto the relative phases of adjacent tones, rather than the distance of the sidebands to the carrier. As the code symbols are spread across the signal bandwidth, a large amount of information can be decoded within the track circuit response time and block coding techniques can provide immunity to single tone or narrow-band interference.

The signal has no phase discontinuities. Since the information is spread across the signal bandwidth and each symbol is demodulated using the relative phases of adjacent tones, the scheme can tolerate a high phase distortion introduced by the channel and by the existing tuned track circuit terminations.

The hardware and detection philosophies are similar to those already used for the Westinghouse FS series of track circuits. The demodulation algorithm is similar to the algorithm used to check the phase relationship of the carrier and two sidebands in the FS track circuit signal. The use of proven techniques and hardware will keep the development/production costs low, simplify the safety case and will be a favourable factor in obtaining the railway authorities' acceptance. It is for these reasons that demodulation and coding techniques for this scheme are proposed and simulated in the following chapter.

12.1 Description of Signal

The existing rate coded track circuit signal consists of a carrier and two sideband frequencies. The range of carrier frequencies is approximately 500Hz to 2kHz, although much higher frequencies are possible [9]. The signal is detected and demodulated at the receiver with the use of an FFT. The FFT parameters are chosen so that all the signal carriers and sidebands correspond exactly to the FFT bin frequencies. Since the sidebands are the result of sinusoidal frequency modulation of the carrier, they are of a smaller amplitude and have a constant phase relationship with the carrier, irrespective of the starting time of the FFT [103].

The following section will describe how the phase relationship of the sidebands to the carrier can be used to convey information, and how the information capacity can be expanded by adding extra tones to the signal.

12.1.1 Encoding Phase Information

We will initially consider a signal consisting of three spectral lines: a carrier at frequency ω_c, and a lower sideband (LSB) and upper sideband (USB) of equal amplitude at frequency ω_m from the carrier. The carrier and sideband frequencies sit exactly on an FFT frequency bin and the FFT frame time is unsynchronised with the signal (i.e. it could start at any time). The following analysis will demonstrate how this signal can be used to convey information. Let *v(t)* be the signal, and ϕ be the phase of the sidebands at t=0:

$$v(t) = A\left\{\cos\omega_c t + \cos\left[(\omega_c + \omega_m)t + \phi\right] + \cos\left[(\omega_c - \omega_m)t + \phi\right]\right\} \qquad (12.1)$$

This could be written in complex notation:

$$v(t) = \mathrm{Re}\left[A\left\{e^{-j\omega_c t} + e^{-j\left[(\omega_c+\omega_m)t+\phi\right]} + e^{-j\left[(\omega_c-\omega_m)t+\phi\right]}\right\}\right] \qquad (12.2)$$

Lets say this signal is complex heterodyned by multiplying by the phasor $e^{+j\omega_c t}$. Assuming that all the sum frequency components are filtered out, the resulting complex signal is:

$$\begin{aligned} r(t) &= A\left[e^{+j(\omega_m t-\phi)} + e^{-j(\omega_m t+\phi)} + 1\right] \\ &= A\left[e^{-j\phi}.2\cos(\omega_m t) + 1\right] \end{aligned} \qquad (12.3)$$

Ignoring the last term which can easily be eliminated, equation 12.3 describes a vector at angle ϕ whose amplitude is varying from +2A to -2A at a frequency of ω_m.. Alternatively, in equation 12.3, *r(t)* (ignoring the last term) can be described graphically as 2 phasors, $Ae^{-j\omega_m t}$ and $Ae^{+j\omega_m t}$ rotating in opposite directions. The sum of these two phasors produces a resultant vector at angle ϕ or (ϕ+π) to the positive real axis, as shown in figure 12.1.

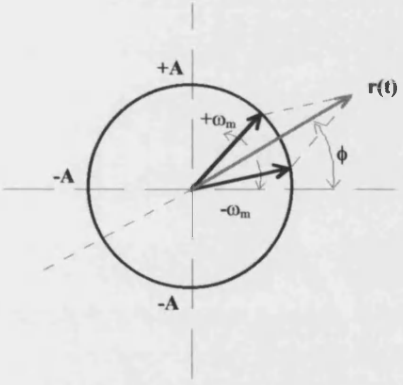


Figure 12.1. Vector Addition of USB and LSB.

Note that it was assumed that the signal was heterodyned by a complex phasor that was in phase with the carrier, ω_c. If we do not perform the heterodyning process, and, at t=0, the carrier is at arbitrary phase, ϕ_2, and the sidebands are at phase ϕ_1+ϕ_2, then the waveform will be:

$$v(t) = \mathrm{Re}\left[A\left\{e^{-(j\omega_c t+\phi_2)} + e^{-j\left[(\omega_c+\omega_m)t+\phi_1+\phi_2\right]} + e^{-j\left[(\omega_c-\omega_m)t+\phi_1+\phi_2\right]}\right\}\right] \qquad (12.4)$$

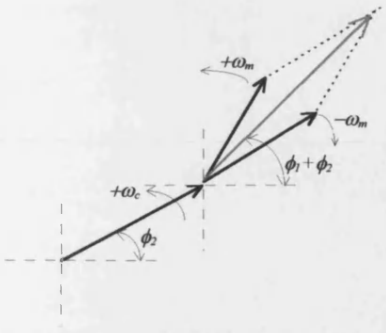


Figure 12.2. Calculation of Resultant Phase, Relative to Carrier Phase.

The phase of the carrier is taken as the datum, and the resultant phasor of the two side bands will be at an angle ϕ_1 to this datum. This is shown in figure 12.2.

The phase of the upper side band will lead the phase of the carrier by ω_m rad/s, and the phase of the lower side band will lag the phase of the carrier by ω_m rad/s. The resultant phasor of the two side bands will lead the phase of the carrier by ϕ_1 rad, irrespective of the initial phase. i.e., no FFT frame synchronisation is required.

12.1.2 Expanding the signal

In this group of three tones, the information is contained in the phase of the resultant 'side-band' vector relative to the phase of the 'carrier'. We shall call these three spectral lines, a *sub-channel.* Another sub-channel can be added at a higher frequency if one extra tone is added and the first carrier is used for the LSB of the 2nd sub-channel. This would require one extra spectral line, to effectively double the information capacity. More sub-channels could be added to this, by adding one extra sinusoid per sub-channel.

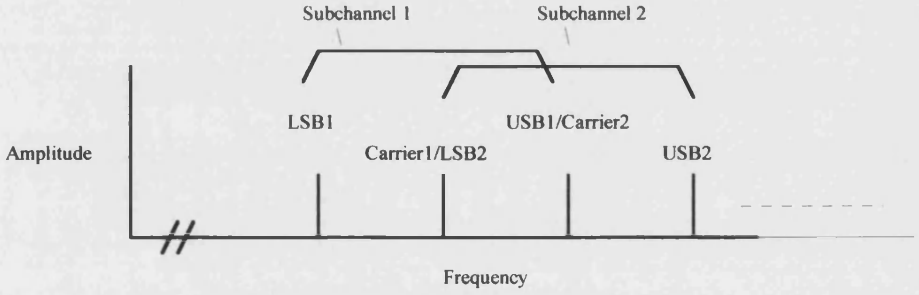


Figure 12.3. Adding More Sub-channels to the Signal.

In the general case, consider two sinusoids, ω_{LSB} and ω_{CAR}, at arbitrary phase, ϕ_{LSB} and ϕ_{CAR}. These represent the LSB and carrier of a subchannel. We wish to add one sinusoid, ω_{USB} at phase ϕ_{USB} to achieve the desired angle, ϕ_{RES}, which is the angle of the resultant of the sidebands to the carrier. The signal is:

$$v(t) = A\left[e^{-j(\omega_{LSB}t+\phi_{LSB})} + e^{-j(\omega_{CAR}t+\phi_{CAR})} + e^{-j(\omega_{USB}t+\phi_{USB})}\right] \qquad (12.5)$$

The phasors, at t=0, are drawn below:

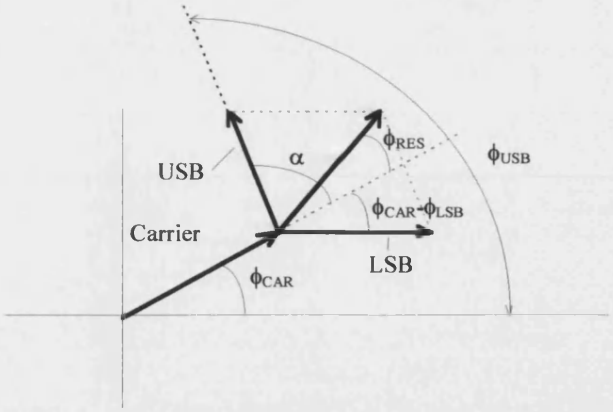


Figure 12.4. Phase of Carrier, Sidebands and Resultant.

$$\phi_{USB} = \phi_{CAR} + \alpha \tag{12.6}$$

since the LSB and USB have the same magnitude:

$$\phi_{RES} = \frac{\alpha - (\phi_{CAR} - \phi_{LSB})}{2} \tag{12.7}$$

therefore:

$$\phi_{USB} = 2\phi_{RES} + 2\phi_{CAR} - \phi_{LSB} \tag{12.8}$$

So, the phase of the USB at t=0, depends on the required resultant phase, and the phases of the carrier and LSB, at t=0. ω_l will be used to represent the lowest frequency in the signal, and ω_m will represent the spacing between the frequencies. The multi-subchannel signal will have the form:

$$\begin{aligned} v(t) &= A\{e^{-j(\omega_l t+\phi_1)} + e^{-j((\omega_l+\omega_m)t+\phi_2)} + e^{-j((\omega_l+2\omega_m)t+\phi_3)} + e^{-j((\omega_l+3\omega_m)t+\phi_4)} + e^{-j((\omega_l+4\omega_m)t+\phi_5)} + \ldots \\ &\quad \ldots + e^{-j\left((\omega_l+(N+1)\omega_m)t+\phi_{(N+2)}\right)}\} \\ &= \sum_{i=1}^{N+2} \exp -j\left((\omega_l+(i-1)\omega_m)t+\phi_i\right) \end{aligned} \tag{12.9}$$

where N is the number of subchannels. The choice of ϕ_1 and ϕ_2 can be arbitrary. If ϕ_{R_k} is used to represent the resultant phase of the kth subchannel, the phases of the remaining sinusoids are given below:

$$\begin{aligned} \phi_3 &= 2\phi_{R_1} + 2\phi_2 - \phi_1 \qquad \text{(if we make } \phi_2 = 0 \text{, then } \phi_1 = \phi_3 = \phi_{R_1}\text{)} \\ \phi_4 &= 2\phi_{R_2} + 2\phi_3 - \phi_2 \\ \phi_5 &= 2\phi_{R_3} + 2\phi_4 - \phi_3 \\ &\vdots \quad \vdots \quad \vdots \quad \vdots \\ \phi_k &= 2\phi_{R_{k-2}} + 2\phi_{k-1} - \phi_{k-2} \\ &\vdots \quad \vdots \quad \vdots \quad \vdots \\ \phi_{N+2} &= 2\phi_{R_N} + 2\phi_{N+1} - \phi_N \end{aligned} \tag{12.10}$$

If the signal is to be encoded with a binary code word $W = (w_0, w_1, w_2 \ldots w_{n-1})$ of length n bits, we make

$$\phi_{R_k} = (\pi . w_k)/2 \qquad \text{where } k=0,1....(n-1) \tag{12.11}$$

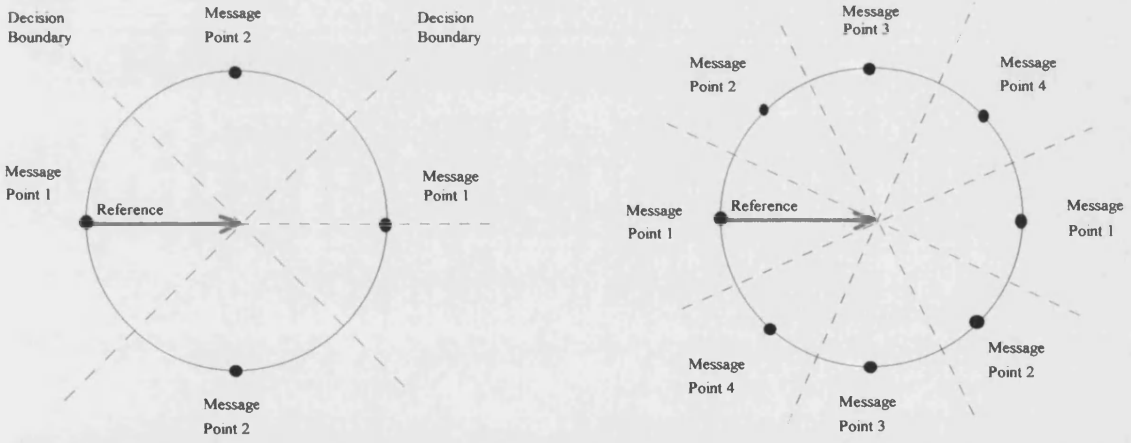
The signal can then be encoded using equations 12.11, 12.10 and 12.9.

If ω_m is the spacing between sinusoids, then for N subchannels, a bandwidth of $(N+2)\omega_m /2\pi$ Hz is required. Each subchannel has an independent reference carrier midway between the two sidebands. This means that a high phase distortion over the signal bandwidth could be tolerated, as long as the phase distortion between any two sidebands is acceptably low.

12.1.3 Signal Space.

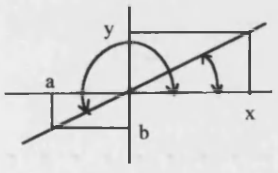
The encoder will assign a value of ϕ_i for each subchannel. As the receiver cannot differentiate between ϕ_i and $\phi_i+\pi$, the encoded value of phase may take on discrete values from 0 to π. These values will be evenly spaced. For each subchannel there will be a value of β_i decoded at the receiver. A decision will be made at the receiver that the value of ϕ_i nearest β_i or $\beta_i+\pi$ is the correct encoded information.

The diagrams below show the signal space diagrams for one subchannel with (a) M=2 and (b) M=4.



(a) M=2 Signal Space (b) M=4 Signal Space

Figure 12.5. Signal Constellation Diagram.



$\tan^{-1}(y/x)=\tan^{-1}(b/a)$

Figure 12.6. Demodulation with the $\tan^{-1}$ function

The choice of which system to use depends on many factors, e.g. receiver signal to noise ratio, redundancy in the message, the use of a guard interval between the decision boundaries where a signal will be rejected etc. Note that the 180° phase ambiguity enables the $\tan^{-1}$ function to be used for demodulation, which also has this ambiguity, as shown in figure 12.6.

12.2 Demodulation of Noisy Signal

Appendix D studies the demodulation of the signal plus Gaussian noise (AWGN). Appendix E shows the details of a simulation used to determine to signal error probability.

Figure 12.7 compares the simulation results for a single subchannel of a multi-tone signal for values of M=2 and M=4. Since the total number of tones required is (N+2), where N is the number of subchannels, the error probability *per subchannel* will approach figure 12.7 as the number of subchannels increases. Also shown, for comparison, is the bit error probability for non-coherently detected FSK [81], which has previously been applied to track circuit signalling [27].

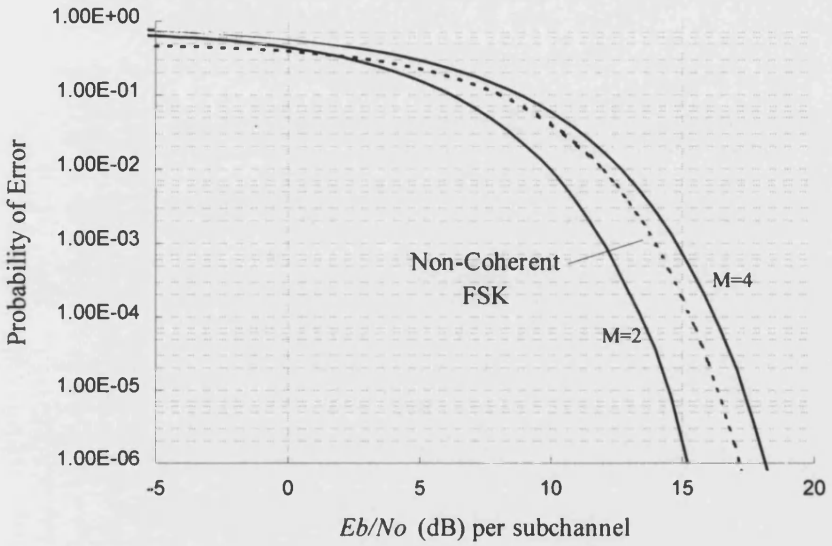


Figure 12.7. Probability of Error for Multi-tone signal

12.3 Rail to Rail Contamination Breakdown Voltage

A desirable property for a new track circuit signal is a high rail to rail voltage to break down any contamination or rust on the rail surface which may cause a shunt to be undetected. With long transmitter/receiver track circuits, the signal is lowest at the receiver and must not be attenuated below the contamination breakdown voltage (3-6v). With a signal that consists of sinusoids spaced at regular intervals, care must be taken that the envelope of the signal does not fall below the breakdown voltage for a high proportion of the detection window time. For example, consider 2 sinusoids at 1600Hz and 1605Hz. The sampling time to separate these at the receiver is 0.2 seconds. The envelope of this signal is a sine wave of 2.5Hz. If both sinusoids have an amplitude of one volt then the signal over the detection window is shown in figure 12.8.

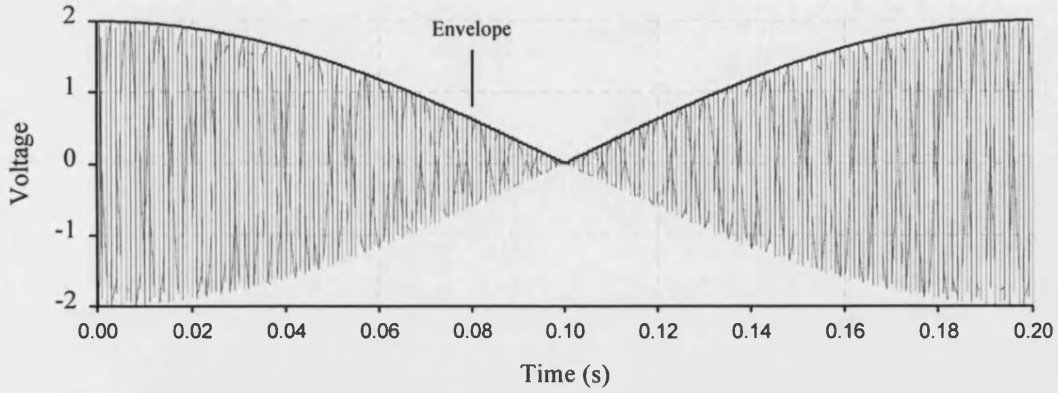


Figure 12.8. Envelope of 2 Sinusoids

The envelope is low for a large proportion of the time due to the low beat frequency of the two sine waves. Not enough is known about the nature of the contamination layer to say how high the beat frequency should be. Carrier frequencies as low as 500Hz have been used successfully[1] and so a beat frequency as low as 500Hz could be acceptable.

12.4 Envelope of Signal

A signal with N+2 sinusoids has the form :

$$v(t) = \mathrm{Re}\left[\sum_{i=1}^{N+2} \exp\left(-j\left(\omega_1 + (i-1)\omega_m\right)t + \phi_i\right)\right] \qquad (12.12)$$

The envelope of this signal is given by :

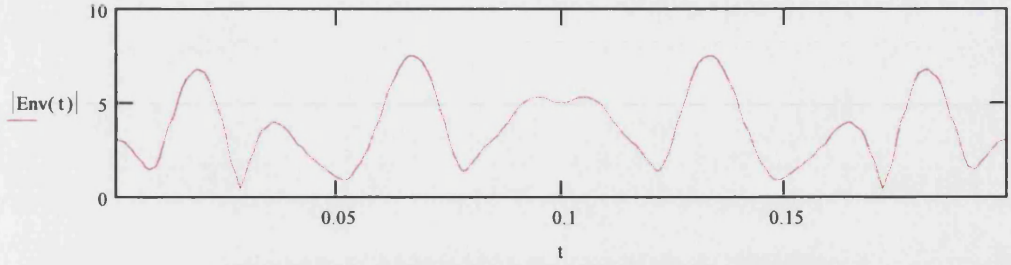
$$e(t) = Mag\left[\sum_{i=1}^{N+2} \exp\left[-j(\omega_k + (i-1)\omega_m)t + \phi_i\right]\right] \qquad (12.13)$$

where $\omega_k = -\left(\frac{N+1}{2}\right)\omega_m$.

Note: the envelope is independent of the frequency of the signal. If the contamination layer can be modelled as zener diode with a specific breakdown voltage [21] then a figure of merit for a signal (over one observation interval) would be the percentage of time that the signal is above the breakdown voltage. As an example, the graphs below show the envelope for a signal coded with several (15,5) BCH code words. The signal consists of 17 sine waves (3 sinusoids to encode the first bit, and 1 sinusoid for each subsequent bit) each with a peak amplitude of V_{peak}=1V.

[1] i.e. the zero crossings of this carrier wave seem to be fast enough to maintain a short circuit between the wheels and rail.

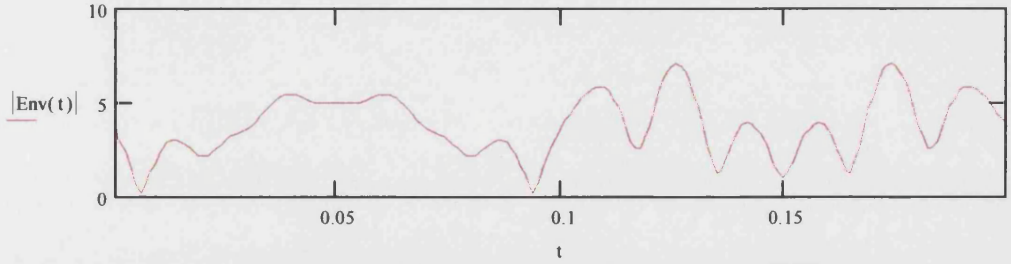
Code Word 4



Percentage of time envelope is above V_{peak} = 96.5%

Percentage of time envelope is above $2V_{peak}$ = 77.1%

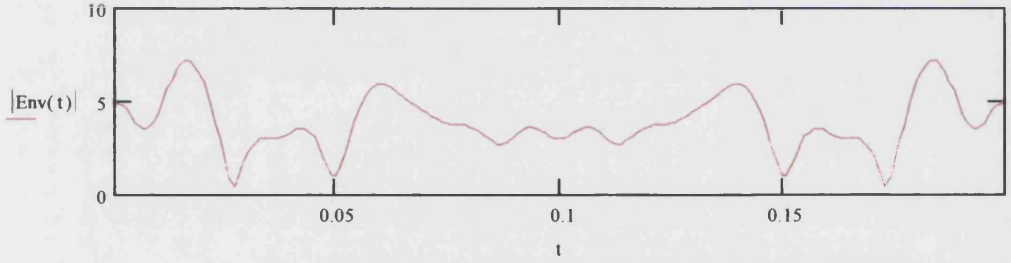
Code Word 7



Percentage of time envelope is above V_{peak} = 96.5%

Percentage of time envelope is above $2V_{peak}$ = 87.6%

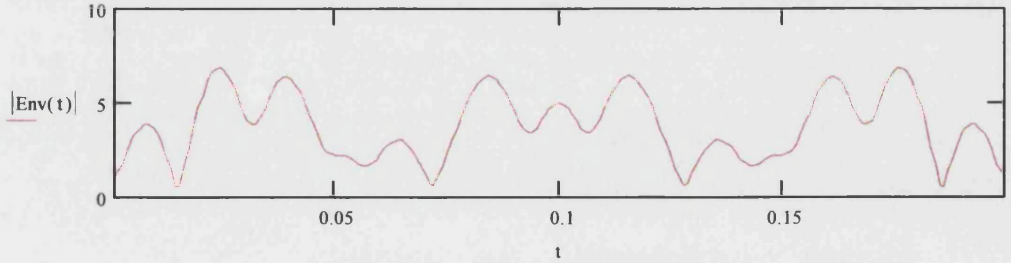
Code Word 16



Percentage of time envelope is above V_{peak} = 96.5%

Percentage of time envelope is above $2V_{peak}$ = 91%

Code Word 22



Percentage of time envelope is above V_{peak} = 95.5%

Percentage of time envelope is above $2V_{peak}$ = 81%

Figure 12.9. Envelopes of Multi-tone Signal Representing (15,5) BCH code words.

Less variation in the envelope is desirable, which will give less dynamic range at the receiver. We can see that the envelope may approach zero at a rate no lower that the tone spacing.

12.4.1 Controlling The Envelope.

If the tone spacing is too low, and the envelope variations too slow, then one method of obtaining a more constant envelope would be to have one sinusoid of larger amplitude than the others, as shown below:

Figure 12.10. Multi-tone Signal with One Higher Amplitude Tone to Control Envelope.

This large sinusoid could be in the middle or at the edge of the bandwidth. If the large tone is the carrier of a subchannel, the calculation of the resultant is the same, as shown in figure 12.11. If the large frequency is one of the sidebands of a subchannel, the resultant can still be calculated by reducing the magnitude of the large frequency until it is the same as the other sideband, then taking the resultant. This is shown in figure 12.12.

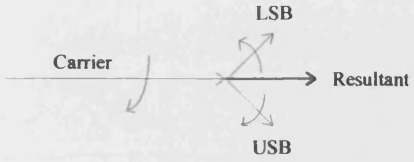


Figure 12.11. Calculation of Resultant with High Amplitude Carrier.

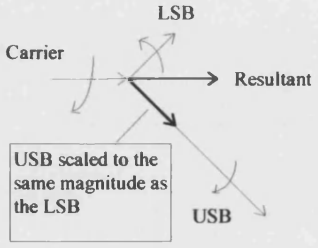


Figure 12.12. Calculation of Resultant with High Amplitude Sideband.

If the total power is to be kept the same, the amplitude of all but one of the sinusoids can be reduced and that power can be transferred to one sinusoid. This has the disadvantage of reducing the signal to noise ratio for most of the signal. If the dominating interference is high amplitude narrow band impulsive noise then this reduction of SNR may not affect the performance as much as a signal with AWGN. This is because the portion of signal affected by noise will be lost, whatever the amplitude, and the only way of combating this interference is by the use of redundancy and error correction.

Shown below is the envelope of a signal consisting of 17 sinusoids, encoded with a (15,5) BCH code word. The power of each signal is the same. The black line shows the 3rd code word, with all sinusoids the same amplitude (1V peak). The red line shows the envelope when the first sinusoid is 5 times greater than the others (peak voltage). The blue line shows the

envelope when the first sinusoid is 10 times greater than the others. The overall power for all three signals is the same.

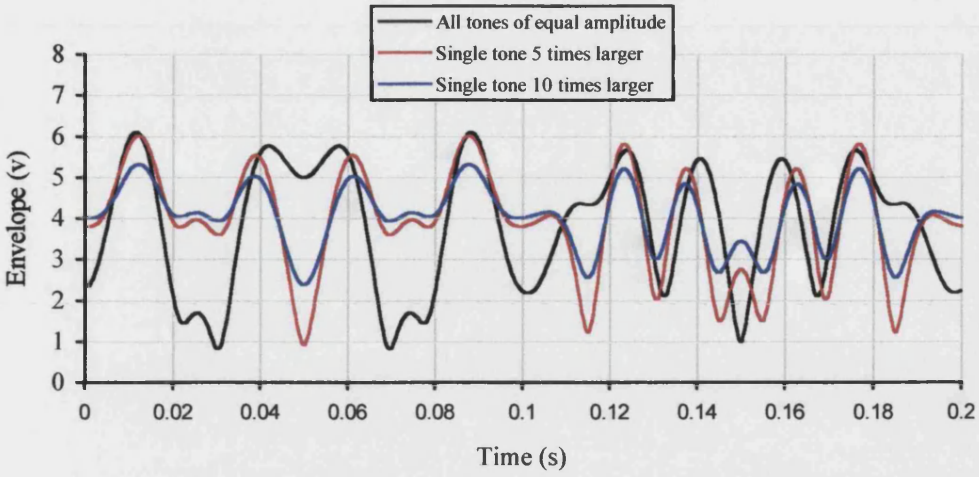


Figure 12.13. Multi-tone Signal Envelopes for the (15,5) BCH Code Word with One Higher Amplitude Tone.

12.5 Comparison with Serial Data Modulation

In this section, we will compare the performance of the multi-tone signal with non-coherent FSK and QPSK. We will compare the power and bandwidth requirements of the systems with equal response time, data capacity and bit error rate. We shall ignore coding redundancy for error control.

Multi-tone signal with M=2, and Non-coherent FSK.

Non-coherent FSK had been used successfully in the past as a form of modulation for track circuit signals. Its performance in AWGN is inferior to some other coherent modulation forms (PSK, MSK) but the equipment is simple and reliable.

Assume we have an available bandwidth of 200Hz, a response time of 0.3 seconds and we require a bit error probability of 10^{-4}:

- For the multi-tone signal, the tone spacing required is 1/0.3 = 3.33Hz. This gives us 60 sinusoids in a 200Hz bandwidth, which can convey 58 bits of information. The SNR required for one subchannel to have a bit error probability of 10^{-4} is 14 dB (figure 12.7). This corresponds to a voltage ratio of 5.01. For 58 bits, the voltage ratio will be (58 . 5.01)= 290.58 = 24.6dB.

- For non-coherent FSK to convey 58 bits of information we must have at least 50% coding redundancy for synchronisation. Reliable synchronisation requires a figure

closer to 67% redundancy [77][82] (i.e. one third of the bits can be used for information).

This means the code word length must be at least 175 bits. Since the coding phase is unknown as a train enters a new track circuit, the receiver must receive $(2n\text{-}1)$ bits in 0.3 seconds. This raises the number of bits to 349, in 0.3 seconds. The bit rate required is therefore 1163.3 bits/sec. Assuming the bandwidth required to reliably transmit at a bit rate of r_b is $\approx 2.5 r_b$ Hz [81], the signal bandwidth required will be 2.9 kHz. The SNR required for a bit error rate of 10^{-4} is 15.33dB.

Multi-tone signal with M=4, and QPSK.

- For the multi-tone signal, again assuming a bandwidth of 200Hz and a response time of 0.3 seconds, the tone spacing required is 1/0.3 = 3.33Hz. This gives us 60 sinusoids in a 200Hz bandwidth, which can convey 58 symbols and 116 bits of information. The SNR required for one subchannel to have a symbol error probability of 10^{-4} is 16.2 dB (figure 12.7). This corresponds to a voltage ratio of 6.45. For 58 bits, the voltage ratio will be (58 . 6.45) = 374.5 = 25.7dB.

- For a QPSK signal to convey 116 bits of information, again assuming 67% redundancy for reliable code synchronisation (i.e. one third of the bits can be used for information) the required data capacity is 348 bits, or 174 symbols, in 0.3 seconds, ignoring any carrier synchronisation delay. Since the coding phase is unknown as a train enters a new track circuit, the receiver must receive $(2n\text{-}1)$ symbols in 0.3 seconds, raising the symbol rate to 1156.67 bd. The bandwidth required to reliably transmit a QPSK signal at symbol rate r_s is $\approx 2r_s$ Hz. So the signal bandwidth required will be 2.31kHz. The SNR required for a symbol error rate of 10^{-4} is 8.8dB.

These comparisons are summarised below.

Data Capacity = 58 bits, Response time = 0.3s Bit Error Rate = 10^{-4}		
	SNR (dB)	Bandwidth (Hz)
Multi-tone Signal, M=2	24.6	200
Non-Coherent FSK	15.33	2900

Table 12.1. Bandwidth and Power Requirements for Multi-tone and FSK Signals with Equal Data Capacities

Data Capacity = 116 bits, Response time = 0.3s Symbol Error Rate = 10^{-4}		
	SNR (dB)	Bandwidth (Hz)
Multi-tone Signal, M=4	25.7	200
QPSK	8.8	2310

Table 12.2. Bandwidth and Power Requirements for Multi-tone and QPSK Signals with Equal Data Capacities

This comparison shows a dramatic trade off between bandwidth and power. The bandwidths shown for the serial data modulation schemes are not possible for AF track circuits and so under these conditions, the serial data modulation schemes are restricted to very low data capacities.

Since the track circuit is primarily a train detection circuit, the transmitter power of track circuits is very high compared to traditional communication systems. Also, the interference is impulsive and bursty rather than Gaussian, and so the power advantage of the FSK and QPSK schemes in the comparison above is unrealistic.

12.6 Coding and redundancy techniques.

In data transmission systems, data coding error control can be divided into two categories: Forward Error Control (FEC) and Automatic Repeat Request (ARQ). If the communication is one way, then the FEC must be accomplished using block coding techniques. Here, the coding has a constant level of redundancy. For two way systems an adaptive amount of redundancy can be achieved by employing ARQ. Error detection algorithms are used at the receiver, which can request that the information be re-transmitted. These systems may perform better in bursty channels, where the channel can be error free for the majority of the time, but have strong bursts where FEC would be ineffective. If these bursts occur, the data will be re-transmitted.

Track circuit data coding is unusual in that the data message can be constant for a long period of time. In this respect we could employ error detection only, and liken the system to an ARQ strategy. The Westinghouse FS series of track circuit equipment employs a sinusoidally modulated FSK signal, detected by the FFT. Error detection is achieved by requiring that three successive FFT results must agree, before a decision is made [6]. If they do not agree, the system response time and reliability are affected since a safe failure will occur. For the more complicated signal proposed, the reliability will need to be improved by employing some form of FEC.

There is an infinite combination of these FEC/ARQ (or error correction/detection) redundancy techniques that could be employed in order to increase both safety and reliability. Here are some examples :

1. Encode the symbols with a linear block code. This could be used for error detection and/or error correction, depending on the decoding algorithm.
2. Require multiple receptions of a block code before a decision is made. Error correction can be employed on the code.

3. Employ a 2 out of 3 voting system for successive FFT results. Error detection could also be applied to the data. This error detection must agree with the voting result.

It is also possible to average the results of several successive or overlapped FFTs. If the FFTs are overlapped, windowing can be applied to each FFT to reduce any spectral leakage produced by an interference tone between two FFT bins (p393, [99]). In the presence of interference, this averaging reduces the variance of the FFT results. If the track circuit signal can be assumed to be *ergodic* (that is, time averages equal ensemble averages), then the more FFTs that are averaged, the lower the variance of the final result. This assumes the interference is random in nature. If the interference is a stationary tone or periodic interference pattern, then averaging the FFT results may not reduce its effect. In this case, block coding would have to be used to improve immunity to interference.

Combinations of time averaging and coding could be used. For example, the portion of the signal corresponding to the 'track to train data' could have a large amplitude and use a block code for error control. The portion of the signal corresponding to 'Full Track Circuit Identity' could have a lower amplitude (to save power and control the envelope) and use the results from successive time averaged FFTs.

Block coding techniques are investigated in the next chapter. The analysis is restricted to the use of a single FFT, with error detecting/correcting block codes. Also discussed are methods of using more demodulation information in the decoding process to increase the safety and reliability of the scheme.

12.7 Conclusions

The receiver structure and algorithms for the proposed signal can be seen as an extension of the existing Westinghouse FS track circuit receiver. The FFT is a proven algorithm and the scheme requires no synchronisation.

A binary word can be modulated using equations 12.11, 12.10 and 12.9. The demodulation involves the calculation of the angle of the resultant of the two sidebands, with respect to the angle of the carrier, as shown in figure 12.4. Since the demodulation of each subchannel is equivalent to the demodulation of the existing rate coded signal and the information is encoded into the phases of successive tones, the scheme is immune to phase and amplitude distortion introduced by the track and the tuned terminations. Unlike serial data modulation schemes, there are no spectral sidebands, and so the scheme is well suited to the requirement of a high

data capacity with a restricted bandwidth and detection time. There is also no block code synchronisation overhead, which reduces the data capacity within a fixed detection time.

The scheme trades off information rate per unit bandwidth, with signal power, with a comparison based on Gaussian noise. The power required (for same BER) is much higher that that required for other digital modulation schemes. However, this may not be a drawback since the interference will not be Gaussian, and the power output must be high enough to break down rail contamination and overcome track attenuation. With this scheme, there is a lot of scope for block coding and redundancy techniques to increase the reliability of the signal. A method of controlling the envelope has also been suggested.

These advantages of the scheme promoted a further study of the demodulation and coding/decoding which is presented in the following chapter. A coding scheme is chosen which fulfils the requirements of table 10.1, and the performance is simulated with typical railway interference.

13. Block Coding Techniques with Hard and Soft Decisions for Multi-tone Signal.

This chapter discusses possible coding techniques suitable for the multi-tone signal described in the previous chapter. A coding scheme should be matched to the expected interference and so an impulsive interference model is defined, based on section 6.4. Several binary coding schemes are chosen with this model in mind and their performance has been assessed by simulation. The codes are compared in terms of data capacity, safety and receiver complexity. Several methods of using demodulation data in the decoding process are also investigated.

13.1 Non-Gaussian Noise Model

An assessment of the coding performance in AWGN would be simple and convenient. However, the interference encountered in railway systems is more likely to be narrow-band or impulsive noise that will cause burst interference to the coding scheme used. Bursty channels are notoriously difficult to accurately characterise. The *classic bursty channel* [79] is described as being a channel with two states: *burst* and *guard*. The burst state lasts for no more than B symbols when the channel outputs carry no information about the channel inputs. The guard state is error free and lasts for at least G symbols. An alternative model, proposed by Gilbert [105] uses a two-state Markov chain and includes a probability of random errors in the guard state.

Most bursty channels, however, do not have two separate states but meander erratically in quality. The classic bursty channel limits the length of a burst to no longer than B symbols. An error correcting code could be optimal for a particular channel model but may not perform well in the real channel. Consequently, the following analysis uses an impulsive noise model based on interference encountered in railway environments.

13.1.1 Impulsive Noise Model

The impulsive noise model described in chapter 6 will be used to assess the coding performance. The model consists of an impulse train, whose period is varied at a rate of 1ms/s. The receiver band-pass filter will shape the impulse train into a series of overlapping or no-overlapping impulse responses, depending on the period between the impulses. The period is swept up and down from the lowest period, 3ms, to the highest period, 200ms (arbitrarily chosen to produce one impulse per FFT block).

13.1.2 Receiver Design

A signal bandwidth of 200Hz is assumed (see section 10.1). The receiver digital filter used for the simulation is a 6th order Chebychev filter whose the transfer function is shown below. The receiver filter is used to reduce the total noise and adjacent track circuit signal power at frequencies outside the 200Hz signal bandwidth.

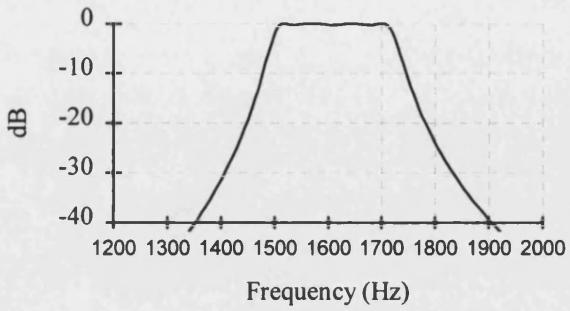


Figure 13.1. Frequency Response of Multi-tone Signal BPF

The impulse response is shown in figure 13.2

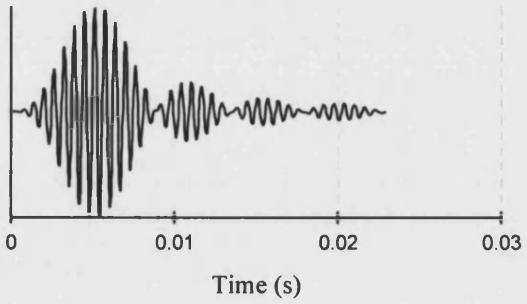


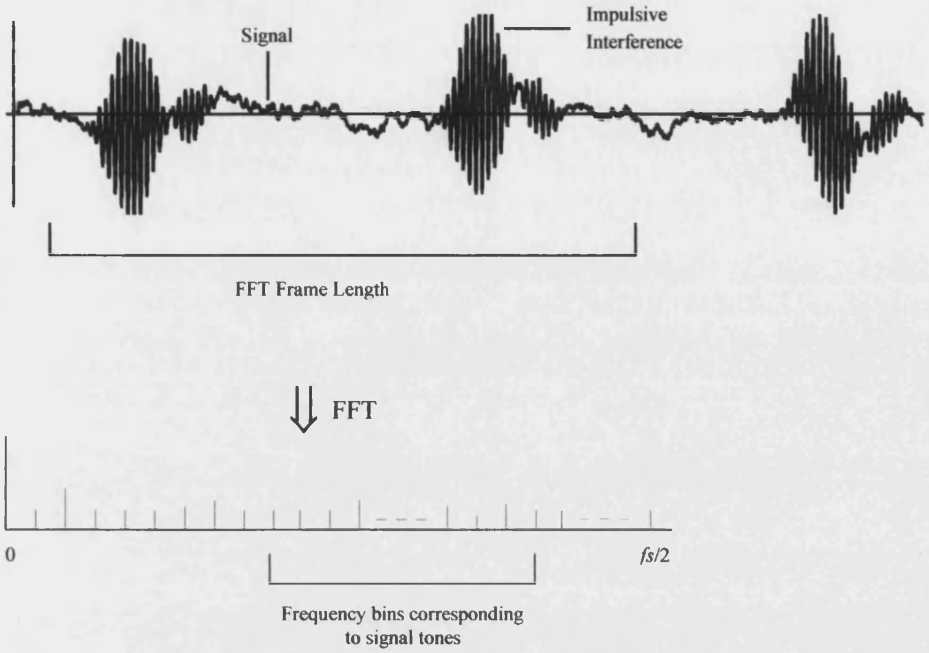
Figure 13.2. Impulse Response of Multi-Tone Signal BPF.

After the filter, the simulation performs an N=4096 point FFT. In the actual real time implementation, the FFT size could be reduced by hetrodyning and decimation techniques [103]. The sampling frequency, f_s, is 13.6533kHz, which gives an FFT bin width of 3.33Hz, and a total sampling time of 0.3 seconds. 60 frequency bins are therefore available in a 200Hz bandwidth, from 1500Hz to 1700Hz and so the block code can be up to 58 bits long.

13.1.3 Received Error Patterns

The aim of the simulation is to choose a coding scheme which will provide maximum immunity to the expected interference. The first step in such a choice would be to observe the error patterns produced by the interference model (for example, a good random error correcting code will be ineffective with single burst errors, and vice versa).

The filter impulse responses are added to the signal, the FFT is then taken and the binary code word is extracted through the demodulation algorithm, as shown in figure 13.3



Code Word : 00001100111001010111000000O

Figure 13.3. Signal Demodulation Example

Let $i(t)$ denote the interference impulse train of period τ samples (τ/f_s seconds), at the input to the filter:

$$i(t) = \sum_{k=-\infty}^{k=+\infty} \delta(t - kT_s\tau) \tag{13.1}$$

If this sequence is passed through a filter with impulse response $h(t)$, the resulting sequence is:

$$\begin{aligned} x(t) &= \sum_{k=-\infty}^{k=+\infty} h(t - kT_s\tau) \\ &= i(t) * h(t) \end{aligned} \tag{13.2}$$

The resulting sampled sequence is:

$$\begin{aligned} x(n) &= \sum_{k=-\infty}^{k=+\infty} h(n - k\tau) \\ &= i(n) * h(n) \end{aligned} \tag{13.3}$$

where $i(n) = \sum_{k=-\infty}^{k=+\infty} \delta(n - k\tau)$ and $\delta(j) = \begin{cases} 1 & j = 0 \\ 0 & j \neq 0 \end{cases}$. If we take an N point DFT,

$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N}$ and assuming that N is a multiple of τ (i.e. there is an integer number of impulse periods within the FFT frame time), the resulting discrete spectrum is:

$$X(k) = I(k)H(k) \tag{13.4}$$

where $H(k)$ is the filter frequency response, and $I(k)$ is the DFT of the sequence $i(n)$:

$$\begin{aligned} I(k) &= \frac{1}{N}\sum_{n=0}^{N-1} i(n)e^{-j2\pi nk/N} \\ &= \frac{1}{N}\sum_{n=0}^{\frac{N}{\tau}-1} e^{-j2\pi nk\tau/N} \\ &= \frac{1}{\tau}\sum_{n=0}^{\tau-1} \delta(k - \frac{nN}{\tau}) \end{aligned} \tag{13.5}$$

So, the DFT $X(k)$ will consist of periodic impulses scaled by the filter frequency response, of amplitude inversely proportional to τ and period equal to N/τ DFT bins. If the period between the impulses is made larger, then there is less overall interference energy and the affected bins will have a lower magnitude, reducing the bit error probability.

This is demonstrated in the experimental results shown below. Figure 13.4 shows the results of a 4096-point FFT performed on an impulse train passed through the filter of figure 13.1, of period 25ms. The sampling frequency is 20.48kHz (making the bin-width 5Hz) which means that τ=512 and N/τ=8. We see from figure 13.4 that one in every eight FFT bins is affected.

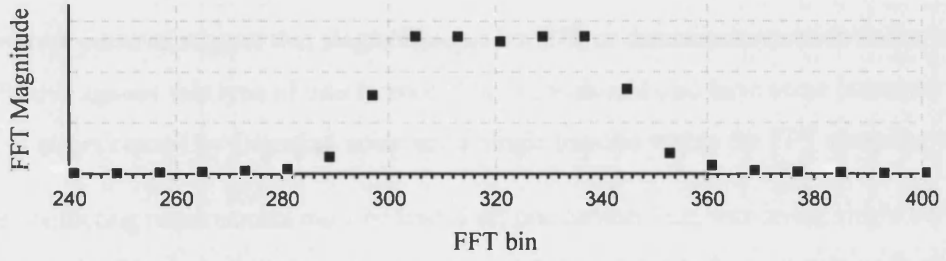


Figure 13.4. FFT of Filtered Impulse Train

The FFT bins affected (300-340) correspond to the pass band frequencies of the filter (1500-1700Hz), and will generate interference at these frequencies.

The FFT is a time 'snapshot' of the actual signal (unlike the Fourier transform) and assumes that the signal is a periodic extension of the FFT frame. If the FFT frame time is not a multiple of the period between the impulses, then the power in the affected bins will be smeared into adjacent bins and the bit error pattern will appear to be extended. Shown below are the FFT results corresponding to an impulse response train of period 12ms.

$$z = \frac{2l}{n-k} \tag{13.8}$$

In general, efficient burst error correcting codes are not efficient random error correcting codes and vice versa. Burst and random error correcting codes often require a compromise of the two construction methods.

13.2.1 Fire Codes

Fire codes were the first class of cyclic codes to be constructed systematically for correcting burst errors [106]. For large codes the efficiency (z) is only approximately 2/3 as the redundancy bits are spread thinly across the code word, but the implementation is fairly simple when feedback shift registers are used.

13.2.2 Cyclic and Shortened Cyclic Single Burst Correcting Codes.

Many of these codes have been discovered by computer search [79]. They are among the most efficient short single burst error correcting codes known, with the shorter lengths ($n \leq 22$) having an efficiency of one. A list of these codes is found in [55] p269.

13.2.3 Compound Fire/BCH Codes for Burst and Random Error Correction.

These codes are described in [107]. They are sub-codes of a Fire code and a BCH code and form a compromise between random and burst error correction. This means that each code word can be independently decoded as a Fire code and a BCH code. The codes are not as efficient as, for example, pure BCH codes but the encoding and decoding processes are simpler.

13.2.4 Interleaving

Interleaving is a method of breaking up a burst into shorter bursts, or single random errors. Given an (n,k) code, it is possible to construct an $(\lambda n, \lambda k)$ code. λ code words of the (n,k) code are arranged as rows of an array. Each column of this array is transmitted column by column. At the receiving end the array is reconstructed and the separate code words are decoded. The parameter λ is known as the *interleaving degree*. It can be seen that a burst of length λ will cause a single error in each code word. If the (n,k) code is a single error correcting code word, the interleaved code will be a λ burst error correcting code word. This has the advantage of simple single error correction. If the original code is cyclic, the interleaved code is also cyclic.

With this technique it is possible to construct long optimal single burst-error-correcting codes from short optimal burst error correcting codes. For example, the (21,9) cyclic code shown on p269 [55] has a burst error correcting efficiency of one (i.e. is capable of correcting all bursts of length 6). If this code is interleaved to degree λ, then the resulting code is capable of correcting all bursts of length 6λ, and also has an efficiency of one.

13.2.5 Product Codes

These codes were proposed by Burton and Weldon [108]. They are capable of simultaneously correcting random errors and single/multiple bursts.

A two-dimensional product code will have sub-codes C_1 and C_2 which are (n_1,k_1) and (n_2,k_2) linear codes. The product, C, will be a (n_1n_2,k_1k_2) linear code. The code is formed such that each word is a rectangular array with n_1 columns and n_2 rows. The k_1k_2 digits in the upper right corner are the information digits. To the left, are the check sum digits according to C_1. Below are the check sum digits according to C_2. In the lower left hand corner are check sum digits using either the C_1 or the C_2 check digits as the information digits. If the codes C_1 and C_2 have minimum distances d_1 and d_2 respectively then the product code will have minimum distance d_1d_2. It can be seen that if C_1 is a trivial (k,k) code then the product code is equivalent to an interleaved code.

If the component codes C_1 and C_2 are cyclic and have lengths which are relatively prime then the product code is cyclic if the bits are transmitted in the right order. For n_1 and n_2 to be relatively prime there must exist a pair of integers a, b, such that $an_1 + bn_2 = 1$.

The order in which the bits from the array are to be transmitted is described in [108] as a mapping of each element in the array to a term in the product polynomial. The mapping is termed $m(i,j)$ where i is the array row and j is the array column, and is defined as :

$$m(i,j) \equiv \left[(j-i)bn_2 + i,\right], \qquad m(i,j) = 0,1,.....n_1n_2 - 1 \tag{13.9}$$

This mapping is shown graphically in [55] p277. Since the product code is cyclic it has a generator polynomial which can be expressed in terms of the component generator polynomials.

$$g(X) = GCD\left[X^{n_1n_2} - 1,\ g_1(X^{bn_2})g_2(X^{an_1})\right] \tag{13.10}$$

where GCD is the greatest common divisor and $g_1(X)$, $g_2(X)$ are the component code generator polynomials.

As the minimum distance between the product code words is d_1d_2, the code is capable of correcting t errors, where

$$t = \left\lfloor \frac{d_1d_2 - 1}{2} \right\rfloor \tag{13.11}$$

but whether this is achieved depends on the decoding procedure [108].

Suppose that C_1 has burst error correcting capability l_1 and random error correcting capability t_1 (and similarly for C_2). The burst error correcting capability of the cyclic product code is:

$$l \geq \max(n_1t_2 + l_1, n_2t_1 + l_2) \tag{13.12}$$

At the receiver the code can be treated as a cyclic code or the received vector can be rearranged as an array. The decoder can decode each of the rows and columns separately according to $g_1(X)$ and $g_2(X)$. In this way the total complexity of the decoder is comparable to the complexity of the individual codes. These codes are more effective for larger code lengths. An example product code follows.

> Consider the component codes : $C_1 = (15,10)$ extended distance 4 cyclic Hamming code, $C_2 = (7,3)$ extended distance 4 cyclic Hamming code (see p113 [55] for a description of these codes). Since (1)(15) + (-2)(7) = 1 the two lengths are relatively prime. The product code is a (105,30) code which is capable of correcting all bursts of length 30 or less, or 7 random errors. As the component codes are single error correcting the receiver complexity is low compared to similar size cyclic codes.

13.2.6 Reed Solomon (RS) Codes

These are BCH codes defined over the ground field GF(q), where q is the power of a prime (usually 2). The codes are defined as follows:

A q-ary (n,k) cyclic code is generated by a polynomial of degree $n-k$ with coefficients from GF(q), which is a factor of X^n-1. The RS codes have the following parameters :

Block length	$n = q - 1$
Number of parity-check symbols	$n-k = 2t$
Minimum Distance	$d_{min} = 2t + 1$

If the symbols are taken from the field GF(2^m) then the generator polynomial is defined as:

$$\begin{aligned} g(X) &= (X+\alpha)(X+\alpha^2)....(X+\alpha^{2t}) \\ &= g_0 + g_1 X + g_2 X^2 + ... + g_{2t-1} X^{2t} \end{aligned} \qquad (13.13)$$

where α is the primitive element of GF(2^m) (see Appendix B). Non primitive RS codes have the same properties. Encoding can be performed in systematic form. Let the message polynomial be

$$a(X) = a_0 + a_1 X + ... + a_{k-1} X^{k-1} \qquad (13.14)$$

We state that $X^{2t}a(X)$ represents the last k terms of the code word polynomial. The first $n-k$ terms are the coefficients of the remainder resulting from dividing the message polynomial $X^{2t}a(X)$ by the generator polynomial $g(X)$. This is generally accomplished using a feedback shift register division circuit. Alternatively a systematic generator matrix can be constructed. This is more suitable for microprocessor encoding [102].

The decoding procedure is similar to the binary case. The received polynomial is evaluated at $X=\alpha^i$, $i = 1,2....2t$ (for the primitive code) to produce $S_1, S_2, ...S_{2t}$. Each value of S will be an element from GF(2^m). Alternatively, $S_1, S_2, ...S_{2t}$ can be evaluated by taking the remainder

obtained from dividing the received polynomial $r(X)$ by the minimal polynomials (over GF(2^m)) of the first $2t$ elements of GF(2^m). These minimal polynomials are $(X + \alpha^i)$, $i = 1,2....2t$.

Error Correction algorithms for multiple error correcting RS codes appear in [109] (algorithm 10.15) or p173 [55]. These are iterative algorithms designed to find the error polynomial of the smallest weight which solves the syndrome equations. It is fairly complex and is not necessary for single, double or triple error correcting RS codes [102].

The RS code word can be converted to a binary code word by representing each element of GF(2^m) by a binary m-tuple. These m-tuples are sometimes known as *bytes*. A single byte error correcting RS code will correct one byte, no matter how many of the m-tuple bits are corrupted. These codes are good at correcting single or multiple burst errors. In general, a t-byte-correcting RS code can correct any single burst of length $(t\text{-}1)m + 1$ or less. For multiple bursts, it can correct any combination of

$$\frac{t}{1+\lfloor (l+m-2)/m \rfloor} \qquad (13.15)$$

or fewer bursts of length l or less. RS codes are not so effective against random errors since a single error will affect the whole byte. Concatenation is a method of equipping RS codes with some immunity to random errors.

13.2.7 Concatenated Codes.

This is another method of constructing longer codes from shorter ones. The code is formed from two codes: a binary inner code C_1, (n_1,k_1) and an outer code, C_2, (n_2,k_2) which is non binary (usually RS). C_2 has symbols from GF(2^{k_1}). The non binary code is represented in binary form and each symbol is taken to be the k_1 information bits for a code word from C_1. The resulting code is an (n_1n_2,k_1k_2) binary code. If the minimum distance of the inner code is d_1 and the minimum distance of the outer code is d_2 (in non binary terms) then the minimum distance of the concatenated binary code will be at least d_1d_2.

Decoding is the reverse procedure of encoding. First each C_1 code vector is decoded and the check digits removed, leaving a sequence of n_2 k_1-digit bytes. These bytes are then decoded according to C_2. The complexity of the decoder is roughly equal to the complexity of the individual codes.

Example :

The (10,6) shortened RS code can be constructed from the (15,11) RS code. Code symbols are taken from GF(2^5) and the encoding/decoding procedures are the same as for the full length code, except that 4 information symbols (the higher order symbols) are

assumed to be zero and therefore not transmitted (p116 [55]). In this way RS codes of any length can be constructed. The binary Hamming (7,4) code is chosen as the inner code. The concatenated code is a (70,24) binary code. The minimum distance of the Hamming code is 3. The (15,11) RS code is a double symbol correcting and so has a minimum distance of 5; the shortened code has the same minimum distance. The concatenated code has a minimum distance of 15 and is capable of correcting any error pattern such that the number of inner code vectors with more than one bit error is no more than two. All bursts of length 9 will be corrected, and phased bursts of length 14 will be corrected. A phased burst is a burst of bit errors which covers the minimum possible number of non-binary symbols (p273[55]).

13.2.8 Chosen Codes

The following codes have been chosen for simulation. A major criterion in the choice is receiver complexity. A simpler demodulation algorithm is easier to prove safe and will save hardware costs.

1. 8 Interleaved Hamming (7,4) codes

The (7,4) Hamming code is a single error correcting code with a minimum distance of 3. The decoding algorithm is simple. The interleaved code will be a (56,32) code, with an interleaving depth of 8. The code rate (k/n) is 0.57. Two decoding techniques will be considered:

1.1 Single error correction for each code

All error patterns that produce a single error in each code will be corrected. This includes all bursts of length 8 or less. This decoding technique is clearly unsafe since two bit errors in one word will produce a decoding error. It is included for comparison.

1.2 Error detection for each code

If no error correction is used then each individual code can detect up to 2 bit errors. The interleaved code is capable of detecting all error patterns that produce up to 2 errors in each code. This includes all bursts of length 16 or less.

2. 7 Interleaved shortened Hamming (8,4) codes

This code is a shortened version of the (15,11) Hamming code ([55] p80). It has a minimum distance of 4 and so can simultaneously correct a single error and detect a double error. The interleaved code will be a (56,28) code. It is capable of detecting all error patterns that cause a double error in each word and correcting all error patterns that cause a single error in each word. This includes detecting all bursts of length 16 and correcting all bursts of length 8. The code rate (k/n) is 0.5.

3. Shortened double error correcting Reed-Solomon (11,7) code

The symbols are defined in the ground field GF(2^5) and are represented by *bytes* of 5 bits. The binary code is a (55,35) code whose code rate (*k/n*) is 0.64. The code is a shortened version of the double error correcting (31,27) RS code, defined in GF(2^5). Appendix F shows an algebraic method of implementing double error correction which is suitable for relatively simple micro-processor decoding. For triple error correction or higher, the decoding algorithm becomes more complicated and is achieved using shift register circuits or ASICs. Double error correcting Reed Solomon codes have been previously applied to train control systems via a radio link. [102].

Three decoding techniques will be considered :

3.1 Double error correction

Any 2 5-bit bytes will be corrected. This includes all bursts of length 5 and *phased bursts* of length 10.

3.2 Single error correction

A single 5-bit byte will be corrected. Two byte errors will be detected. This includes error correction for all phased bursts of length 5, and error detection for phased burst of length 10.

3.3 No error correction

Up to two symbol errors will be detected. This includes all phased bursts of length 10.

4. Shortened double error correcting Reed-Solomon (10,6) code concatenated with single error detecting parity bit

The symbols are defined in the ground field GF(2^5) and are represented by bytes of 5 bits. The code is a shortened version of the double error correcting (31,27) RS code, defined in GF(2^5). The decoding algorithm is similar to that given in appendix F. A single error detecting parity bit is added to each symbol, or 5-bit byte. The binary code is a (60,30) code, with a code rate (*k/n*) of 0.5.

Three decoding techniques will be considered :

4.1 Double error detection with parity checking

The first stage of decoding is to observe how many symbols have parity errors. If there are more than 2, an code error is detected. If there are 2 parity errors and the syndrome is that of a double error, then the error correction algorithm is executed. The code is only accepted if the corrected symbols correspond to the symbols with parity errors. Similarly, if there is 1 parity error, the code is only accepted if there is a

non-zero syndrome corresponding to the symbol with the parity error. If there are no parity errors, the code is only accepted if there is a non-zero syndrome.

The corrected/detected error patterns of this code are more complex. Although the maximum error correction is used for the RS code, the concatenated code is biased toward error detection since the error correction process must agree with the parity checking. For example, two bit errors in two symbols will be detected instead of corrected, since the symbol parities will not agree with the syndrome.

4.2 Single error detection with parity checking

If there is more than one parity error, a code error is detected. If there is one parity error and the syndrome is that of a single error, then the error correction algorithm is executed. The code is only accepted if the corrected symbol correspond to the symbols with a parity error. If there are no parity errors, the code is only accepted if there is a non-zero syndrome.

4.3 No error detection with parity checking

If there is one or more parity errors, a code error is detected. Also, if there is a non zero syndrome, a code error is detected. This is a powerful error detection code.

The data capacity of all these codes exceed the minimum requirement stated in section 10.1.

13.3 Coding Performance

The software simulation is written in C. It is performed in two separate stages. Firstly, the signal is constructed using pseudorandom data, and the impulsive noise added (section 13.1). The period between impulses is continually swept up to the maximum and down again at a rate of 1ms/s. The starting phase of the period for each cycle is a random factor. Successive FFTs are taken (sampling time = 0.3s) and the demodulation algorithm is performed. The difference between the transmitted bit pattern and the received bit pattern is used to create a bit error file for a particular SNR. The SNR is taken as $10\log_{10}$(Energy per *info* bit/Energy per single burst) i.e. it is scaled by the code rate to take into account the different redundancies. A burst is defined as one impulse response from the receiver filter (figure 13.2).

Due to memory and simulation time limitations, the total number of impulsive interference cycles was limited to approximately 2400, giving a total of approximately 4.10^6 FFTs for each SNR.

The second stage is to perform the decoding and evaluate the performance of each code. Since the simulation has prior knowledge of the received error pattern, and the decoding consists of linear block code syndrome calculation, it is not necessary to implement the encoding process.

Instead, the bit error patterns from the first stage can be used directly as inputs to the decoding algorithms (see Appendix B, equation B.15).

The process is shown in figure 13.6.

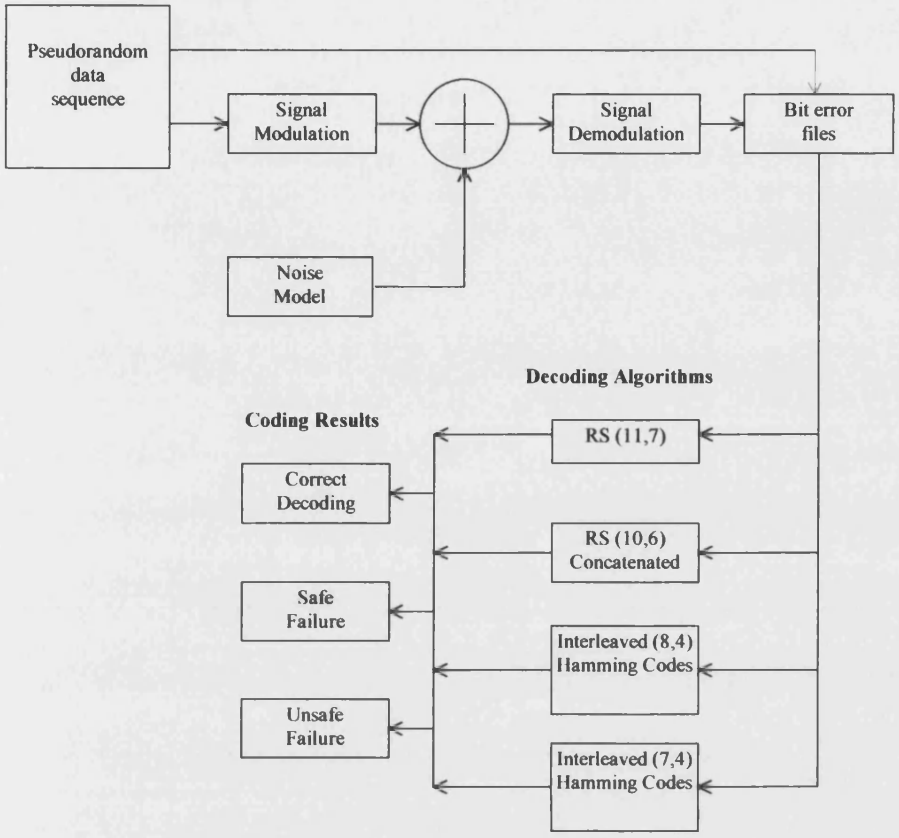


Figure 13.6. Simulation Process.

13.3.1 Code Performance with Impulsive Noise Model

The performance of the codes described above is shown in figures 13.7, 13.8 and 13.9. Figure 13.7 shows the probability of correct decoding. Figure 13.8 shows the probability of safe failure (an error being detected in the decoding process) and figure 13.9. shows the probability of unsafe failure (a decoding error). Figures 13.7 and 13.8 give us an indication of the reliability of the code and figure 13.9 gives us an indication of the safety of the code. Note that a high 'correct decoding' curve is generally accompanied by a low 'safe failure' curve, and vice versa.

Comments

Shown in figures 13.7 and 13.9 are the curves for uncoded 'raw' data. A data block length of 35 bits was used in the simulation, corresponding to the desired data capacity, shown in table 10.1. Since there in no redundancy, any bit error will cause an unsafe failure. The correct decoding curve for raw data (fig. 13.7) is actually higher than the coding schemes at some points. This is because the uncoded data has more energy per information bit. However, the

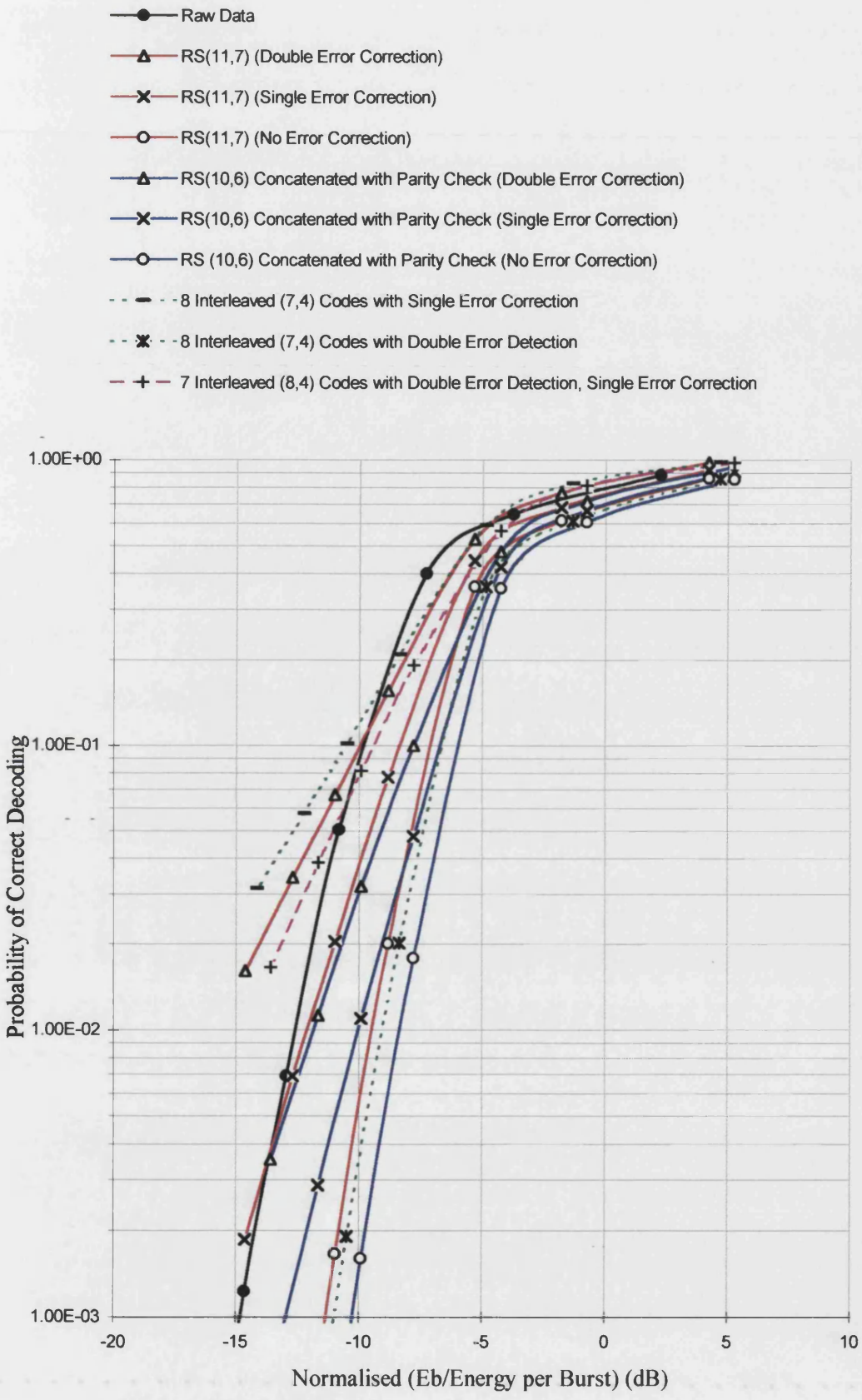


Figure 13.7. Probability of Correct Decoding

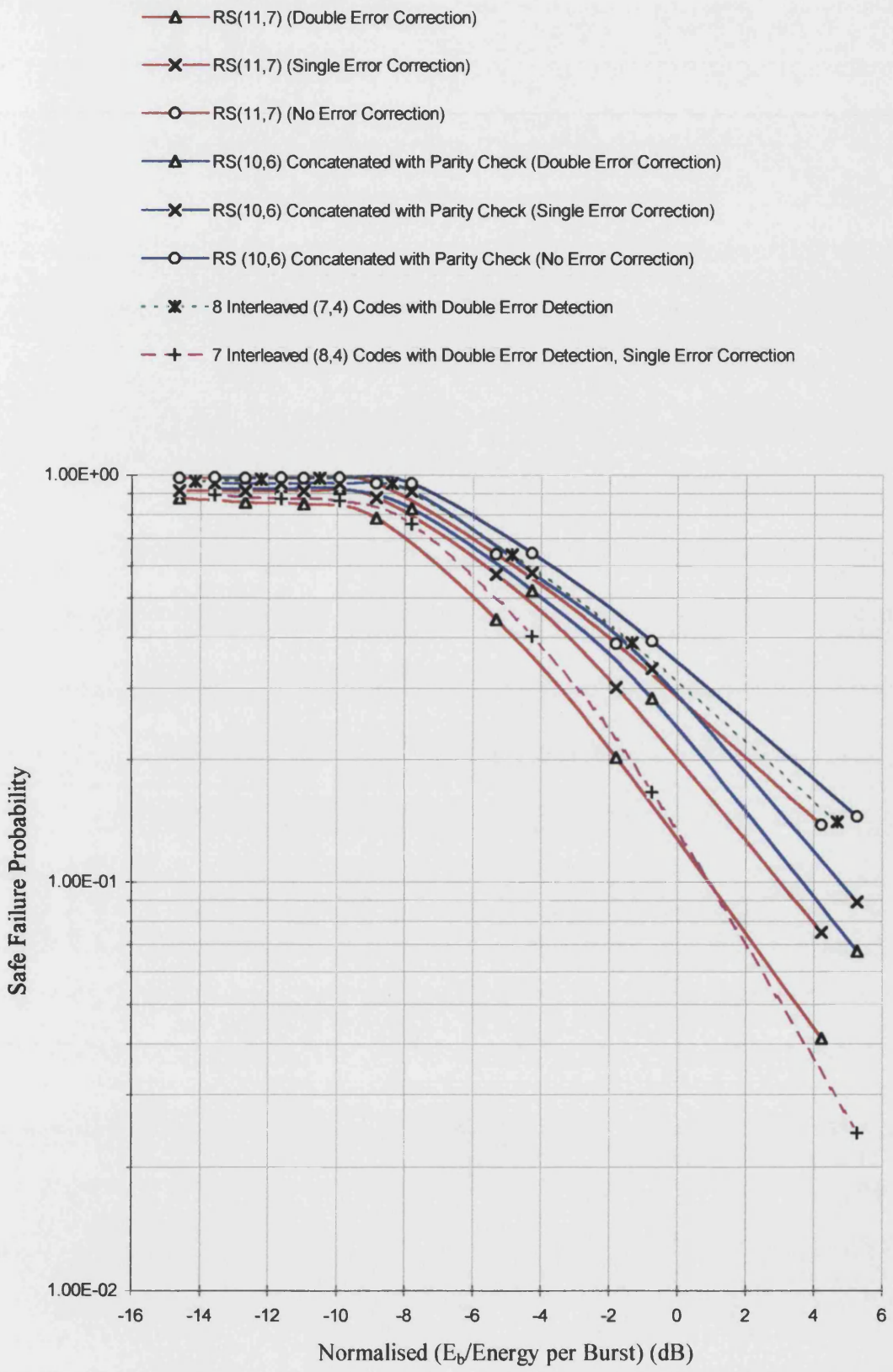


Figure 13.8. Probability of Safe Failure

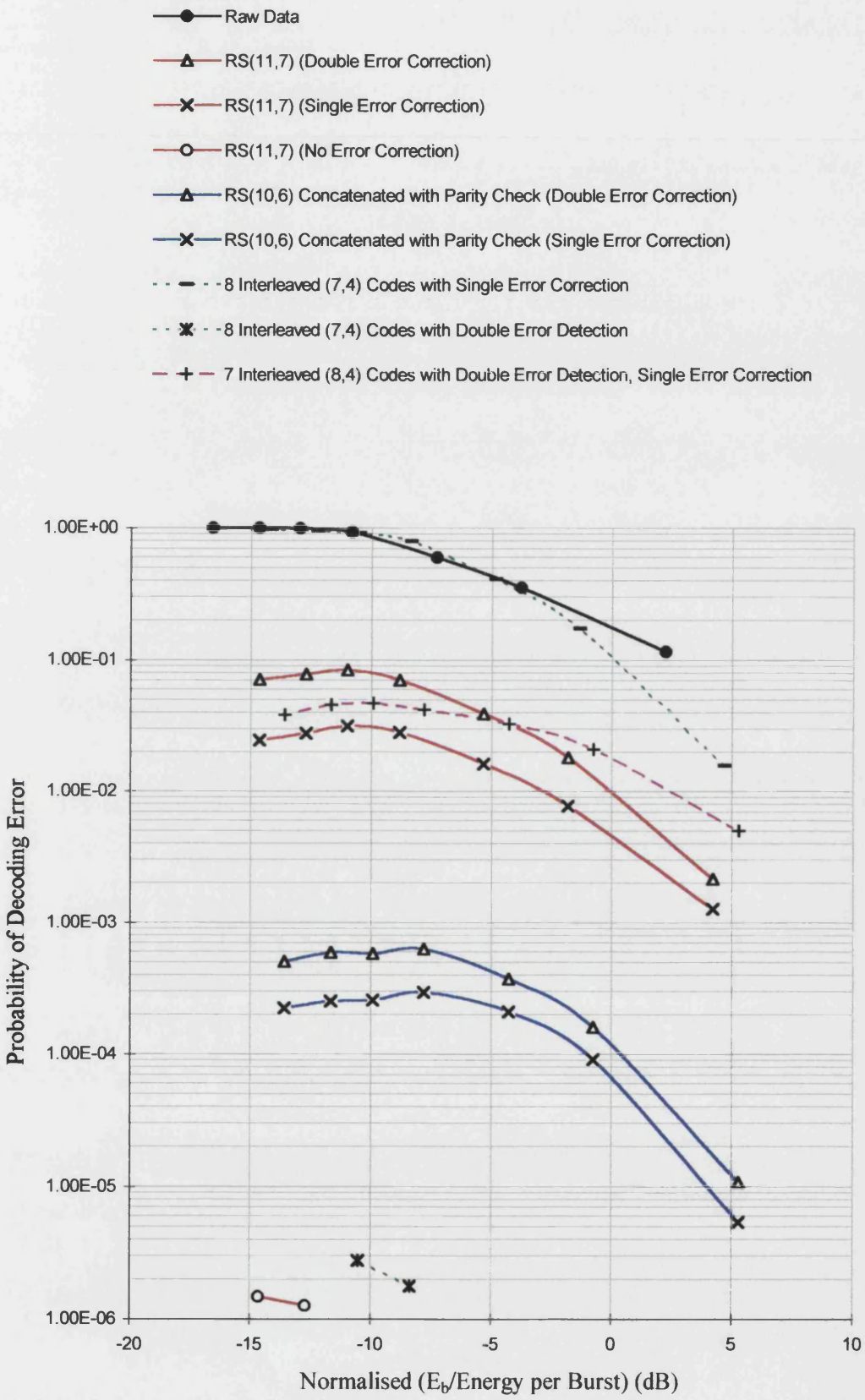


Figure 13.9. Probability of Unsafe Failure

most important graph for track circuit applications is the probability of unsafe failure (figure 13.9). We can see that all the coding schemes offer a substantial coding gain over raw data[1].

Note how the unsafe failure curves decrease slightly at the low SNRs, which is to do with the distance between code words. The unsafe failure curve will peak at an SNR that is most likely to produce the number of bit errors equal to the distances between code words. Hence, with a bit error probability of 0.5 (i.e. random data, see section 13.3.3) the probability of unsafe failure for each code is always lower than the curves on figure 13.9.

No instances of unsafe failure occurred in 46.10^6 trials of the (10,6) shortened RS code concatenated with parity check bits, with no error correction. With random data (section 13.3.3) this code has an extremely low unsafe failure probability ($0.931.10^{-9}$). The trade off in reliability can be seen in figure 13.7, where this code has the lowest probability of correct operation. At the other extreme, the 7 interleaved (8,4) Hamming codes with single error correction have a high reliability but a high unsafe failure probability, with a maximum of ≈ 0.07.

In general, the safety and reliability of the interleaved Hamming codes are at the extreme values of all the codes considered, while the greater error correction capability of the RS codes allows a more gradual trade-off in the code performance. The RS (10,6) concatenated code with single error correction achieves a good reliability (approaching a similar value to raw data at low SNRs) and an increase in safety by a factor of over 10^4 compared to raw data.

13.3.2 Code Performance with Gaussian Noise

Figures 13.10, 13.11, and 13.12 show the code probability of correct decoding, probability of safe failure, and probability of unsafe failure, when the simulation is performed with Gaussian noise. We can see that the code ranking for the probability of correct decoding is approximately the same as for impulsive noise, and that the correct decoding curve for raw data with Gaussian Noise is higher than all the other curves. For the RS codes, the peak unsafe failure is marginally higher with Gaussian noise that with the impulsive noise model, which suggests that these codes are better suited to impulsive noise than the interleaved Hamming codes.

---

[1] Except for the Interleaved (7,4) Hamming code with single error correction. This is to be expected since all error patterns will be decoded.

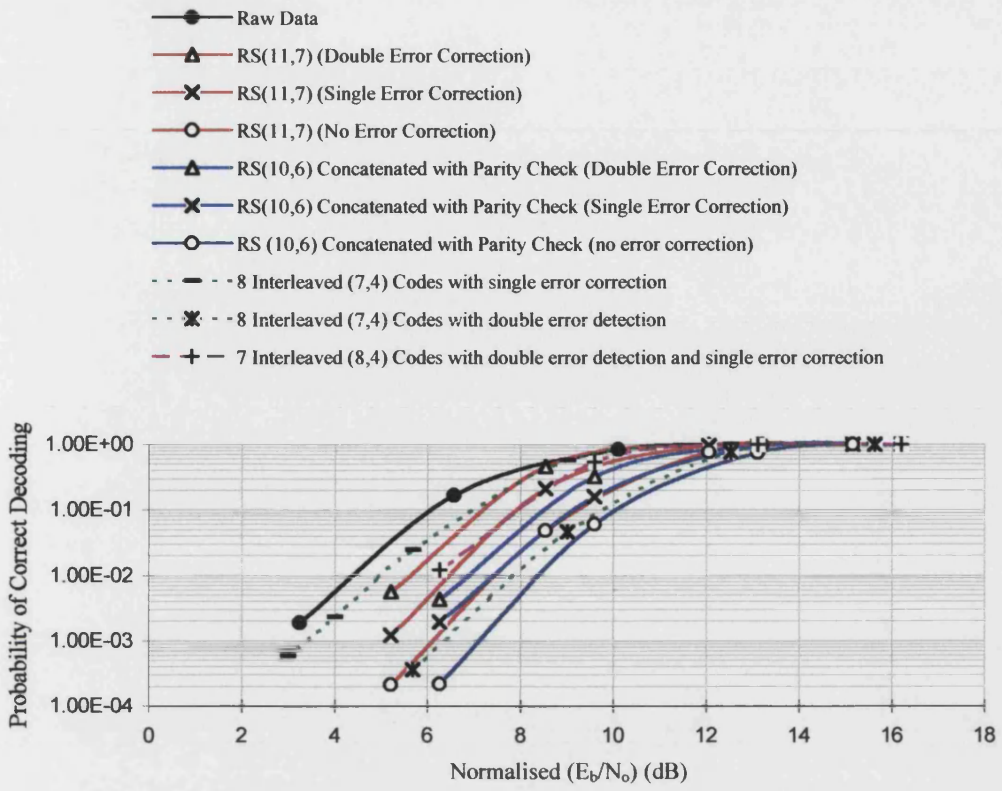


Figure 13.10. Probability of Correct Decoding, with Gaussian Noise

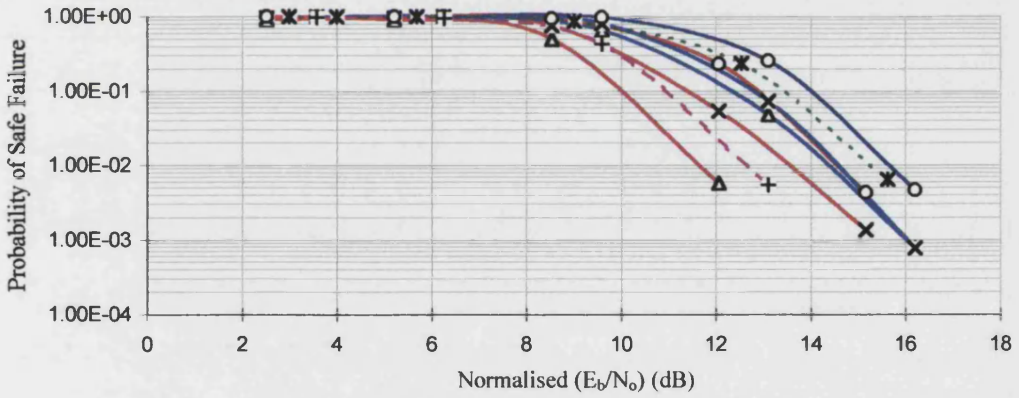


Figure 13.11. Probability of Safe Failure, with Gaussian Noise

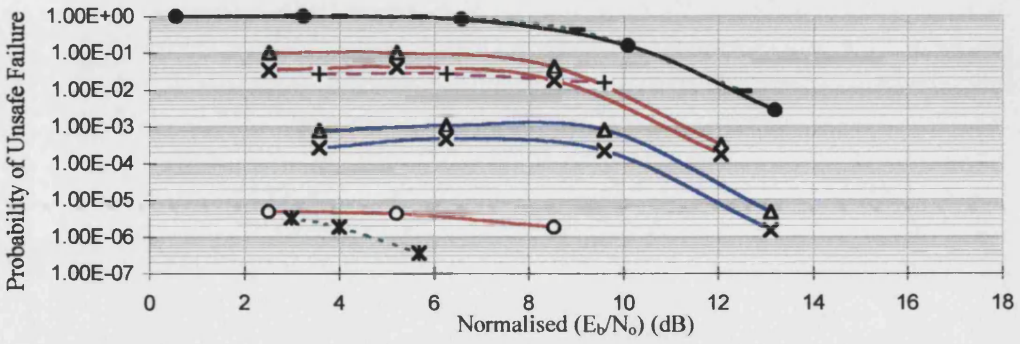


Figure 13.12. Probability of Unsafe Failure, with Gaussian Noise

13.3.3 Code Performance with Random Data

The safety of the coding scheme when the demodulated data contains no valid information (bit error probability = 0.5) is important. In this case, the input to the decoding process is random binary data. This may occur when the SNR is extremely low, or if there is an equipment malfunction. The safety is measured in terms of the probability of false decoding, which can be found directly from equation 8.20. Substituting p=0.5, this equation reduces to:

$$P_{d.e.} = \sum_{i=2t+1}^{n} A_i (0.5)^n + \sum_{j=1}^{t} \left[\sum_{i=2t+1}^{n} A_i \binom{n-i}{j} (0.5)^n + \sum_{i=2t+1}^{n} A_i \binom{i}{j} (0.5)^n \right] = \frac{\text{number of code words} + \text{number of correctable words}}{2^n} \quad (13.16)$$

The weight distribution, A_i, of the code is not needed for the calculation of $P_{d.e.}$. For the Interleaved Hamming (7,4) code with single error correction, each binary word is a valid or correctable code word and so the probability of false decoding will be 1, and this code is clearly unsuitable. Evaluating equation 13.16 for the RS codes with error correction is more complicated and the probability of decoding error has been determined by computer simulation.

Code	Unsafe Failure Probability
Interleaved Hamming (7,4) Code with single error correction	1
Interleaved Hamming (7,4) Code with double error detection	$35.57 \cdot 10^{-8}$
Interleaved Hamming (8,4) Code with double error detection, single error correction	$11 \cdot 10^{-3}$
RS (11,7) Code with double error correction	$85.15 \cdot 10^{-3}$
RS (11,7) Code with single error correction	$33.29 \cdot 10^{-3}$
RS (11,7) Code with no error correction	$0.954 \cdot 10^{-6}$
RS (10,6) Code concatenated with parity check bit. Double error correction	$0.18 \cdot 10^{-3}$
RS (10,6) Code concatenated with parity check bit. Single error correction	$42.6 \cdot 10^{-6}$
RS (10,6) Code concatenated with parity check bit. No error correction	$0.931 \cdot 10^{-9}$

Table 13.1. Unsafe Decoding Failure Probability with Pe=0.5.

13.3.4 Single Tone Interference

Since the interference found in railway systems is often harmonic in nature, the performance of the scheme with a single tone interferer may be of interest. If the tone frequency is on or near an FFT bin frequency, then all the schemes with error correction considered above will result in correct decoding, since only 3 adjacent bits will be affected. Worst case conditions occur when the tone is between FFT frequency bins. With no windowing, the interference tone will leak into all the frequency bins and so if the amplitude is sufficiently large, the decoding will be affected. However, the interference sidebands decrease rapidly. For example, the

amplitude of the 4th bin away from the interference tone is already 20db below that of the 1st bin. Error correction/detection is most likely to be effective since the bit errors will be confined to a single burst.

13.3.5 Out of Band Power Monitoring

The coding performance can be improved if the decoding algorithm has more information on the demodulated data. Since the signal consists of orthogonal tones, each corresponding to an FFT bin, the power in the FFT bins above and below the signal should ideally be zero. Effects such as component tolerance, quantisation and equipment failure, as well as received interference will introduce power into these bins. We shall call this power, the 'out of band' power. The ratio of signal power to 'out of band' power, determined by the demodulation process, will give the decoding process an indication of the integrity of the data. If the integrity is low, then the amount of error correction could be reduced, or the decoding process could simply indicate a safe failure. This is illustrated in figure 13.13 below.

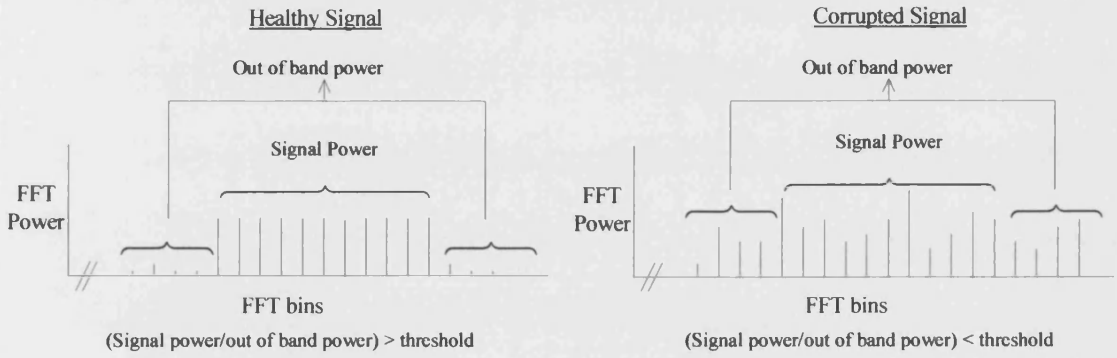


Figure 13.13. Using 'Out of Band Power' in Decoding Process

The number of bins that can be used for determining the out of band power is limited by the bandwidth of the receiver filter.

This process was incorporated into the simulation of section 13.3.1, for the 7 interleaved Hamming (8,4) codes with single error correction and double error detection and the 8 interleaved Hamming (7,4) codes with single error correction. The out of band power was estimated by the sum of the power in the five bins either side of the data bins. The interleaved code is capable of correcting all error patterns that produce no more that one error per individual code. From observation of the ratio of signal power to out of band power, the threshold was set at a value where an uncorrectable number of errors is likely to occur. Also two values above and below this were taken. These ratios were:

$$\frac{\sum_{\text{All signal tones}} (\text{Magnitude})^2}{\sum_{\substack{\text{5 bins either side} \\ \text{of signal tones}}} (\text{Magnitude})^2} = \begin{matrix} \text{(a) } 600 \\ \text{(b) } 300 \\ \text{(c) } 180 \end{matrix} \qquad (13.17)$$

These codes were used in the simulation since their performance in section 13.3.1 is biased towards reliability compared to the other codes. Since the 'out of band power' check will result in a safe failure, the aim of the process is to exchange some of the 'unsafe failures' for 'safe failures' without seriously affecting the reliability (correct operation curve).

Figures 13.14 and 13.15 show the correct decoding curves and unsafe failure curves for the interleaved (7,4) Hamming codes with single error correction. Figures 13.16 and 13.17 and show the correct decoding curves and unsafe failure curves for the interleaved (8,4) Hamming codes with single error correction and double error detection. Also shown for comparison are the curves from section 13.3.1, with normal decoding.

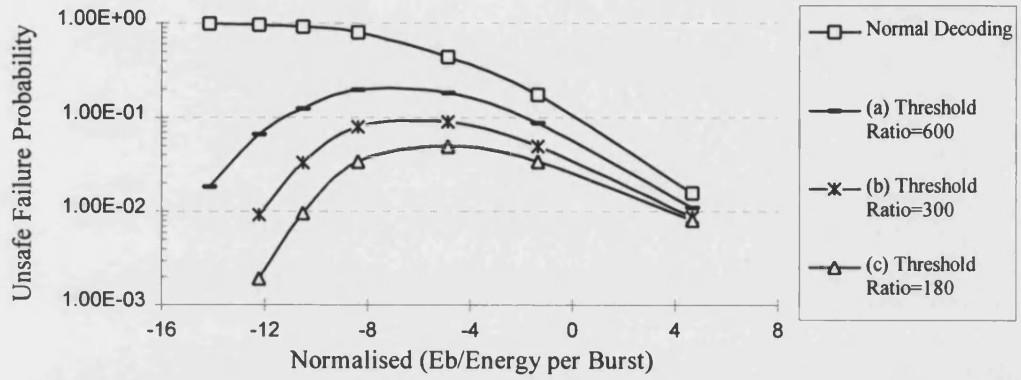


Figure 13.13. Unsafe Failure Curves for Interleaved Hamming (7,4) Code with Single Error Correction and 'Out of Band' Power Checking.

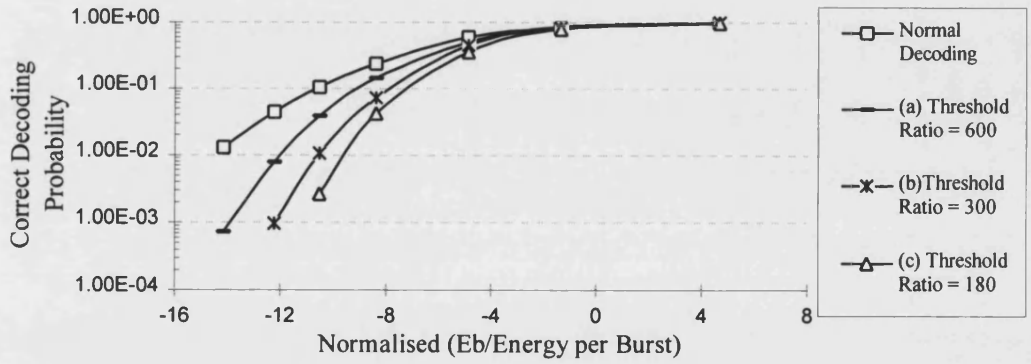


Figure 13.14. Correct Decoding Curves for Interleaved Hamming (7,4) Code with Single Error Correction and 'Out of Band' Power Checking.

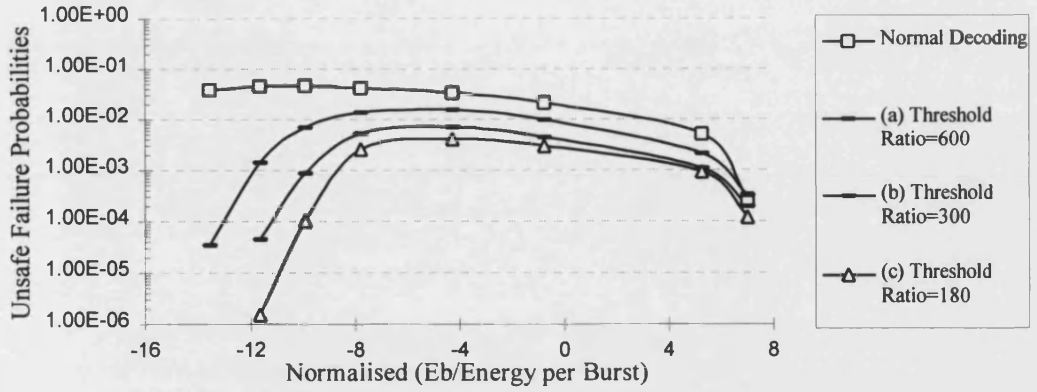


Figure 13.15. Unsafe Failure Curves for Interleaved Hamming (8,4) Code with Single Error Correction/Double Error Detection and 'Out of Band' Power Checking.

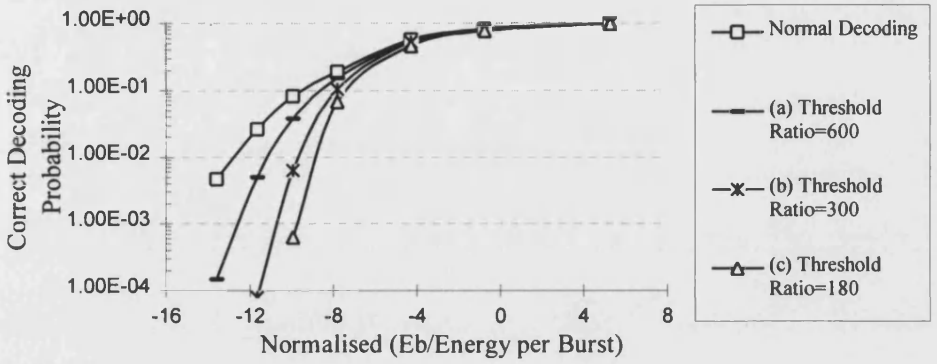


Figure 13.16. Correct Operation Curves for Interleaved Hamming (8,4) Code with Single Error Correction/Double Error Detection and 'Out of Band' Power Checking

In figure 13.13, the unsafe failure curve reaches a peak and then falls sharply as the SNR deteriorates. A lower threshold results in a lower peak unsafe failure rate. However, some reliability is sacrificed as shown by figure 13.14. A similar trend is shown for the interleaved Hamming (8,4) code in figures 13.15 and 13.16. The peak unsafe failure rate has been reduced by a factor of 10.

13.3.5 Sub-optimal Soft Decision Decoding

All the previous work has assumed that the receiver demodulation process performs *hard decisions* on the analogue data, quantising the input to the decoding process as either 1's or 0's. Since no account is taken of the position of each bit in the signal space, much information which could be of value to the decoder is lost. Quantising the demodulator output to Q>2 levels allows the confidence of each bit to be assessed by the decoder. Soft decision techniques are known to offer a 1-2dB advantage for the Gaussian channel, and an even greater improvement in burst channels [98][110][111].

Soft decision decoding is described in appendix G. Two sub-optimal decoding techniques are described: forced erasure decoding and weighted erasure decoding. Weighted erasure

decoding (WED) has been implemented in the demodulation/decoding algorithms and the results are presented in this section. WED is chosen since it does not have a variable decoding delay depending on the error patterns, and it makes use of more demodulation information than the binary erasure channel (BEC).

The performance of the interleaved (7,4) Hamming code has been simulated, with the impulsive noise model. Bit demodulation into Q=5 and Q=8 regions has been implemented, as well as Q=2, for comparison. This is illustrated in figures 13.17, 13.18 and 13.19 below, where one quadrant of the demodulated symbol resultant angle is shown (see figure 12.5).

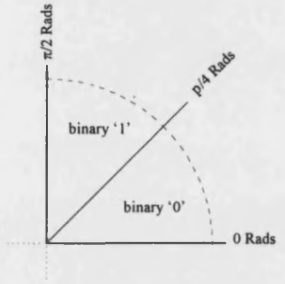


Figure 13.17. Hard Decision Demodulation

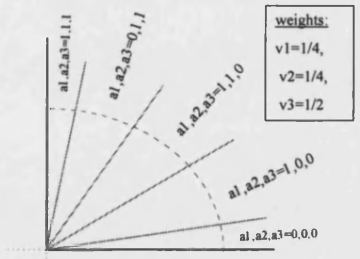


Figure 13.18 Weighted Erasure Demodulation into Q=5 regions

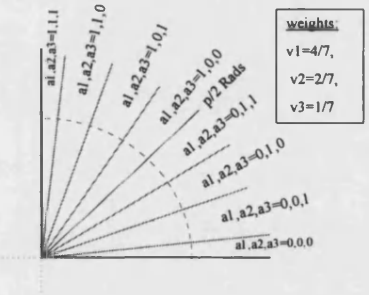


Figure 13.19 Weighted Erasure Demodulation into Q=8 regions

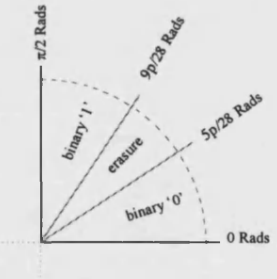


Figure 13.20. Binary Demodulation with Erasure

Figures 13.21 and 13.22 show the probability of correct decoding and unsafe failure for 8 interleaved (7,4) Hamming codes, with binary hard decision decoding, Q=5 WED and Q=8 WED. Also presented are the results for the interleaved (7,4) code with no error correction on the binary erasure channel (BEC). A symbol in the erasure region (figure 13.20) will result in a safe failure.

We can see that the Q=8 WED scheme offers an approximately 1.1dB advantage over binary hard decision decoding (figure 13.21). Q=5 WED offers a fractionally smaller improvement. There were no instances of unsafe failure for error detection on the BEC, which implies that the advantage in using the BEC for error detection, in terms of safety is

considerable. This is however at the expense of reliability, which can be seen from the lowest curve of figure 13.22.

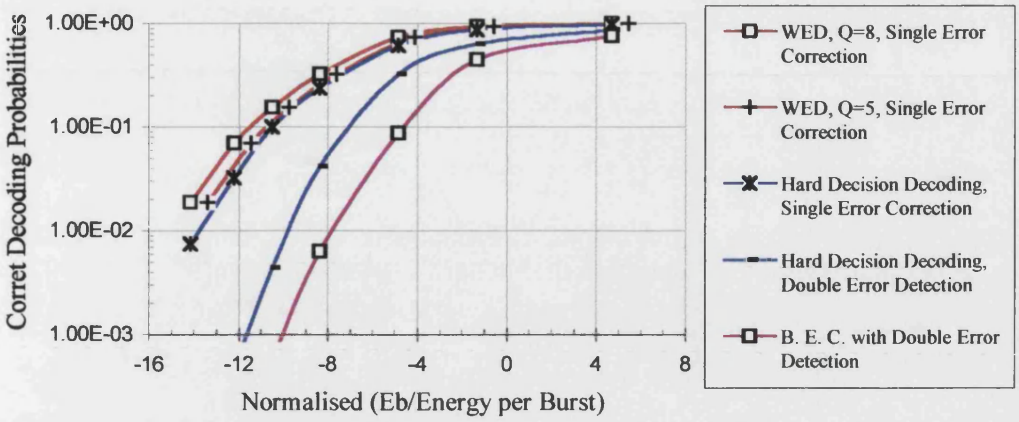


Figure 13.22.Correct Operation Curves for Interleaved (7,4) Hamming Code with Hard Decision and WED Soft Decision Decoding.

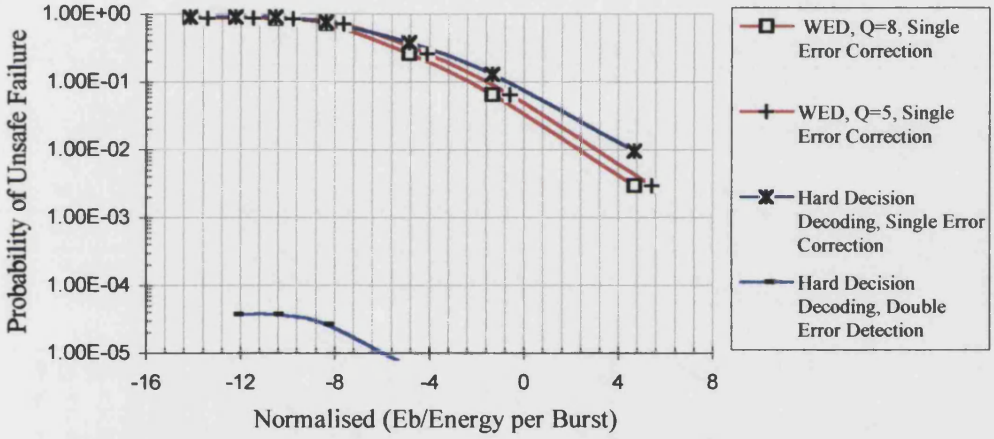


Figure 13.23.Unsafe Failure Curves for Interleaved (7,4) Hamming Code with Hard Decision and WED Soft Decision Decoding.

Soft decoding techniques are aimed at increasing the reliability of a decoding scheme. If they are to be applied to a track circuit coding scheme, some method of error detection must also be employed to keep the unsafe failure rate acceptable low. The following results (figures 13.25 and 13.26) apply to the Q=8 channel with the two regions of least confidence declared an erasure (figure 13.24). If no symbols are received in the erasure region, then the interleaved code is decoded using WED. If any symbols are in the erasure region, then a safe failure is indicated. The unsafe failure rate has improved by a factor of over 10^2, at the expense however, of the reliability.

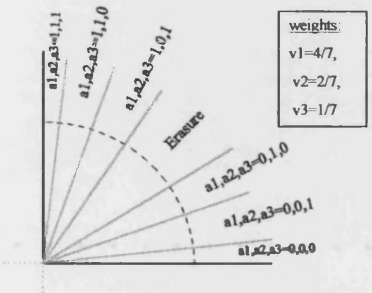


Figure 13.24. Q=8 Demodulation with Erasure

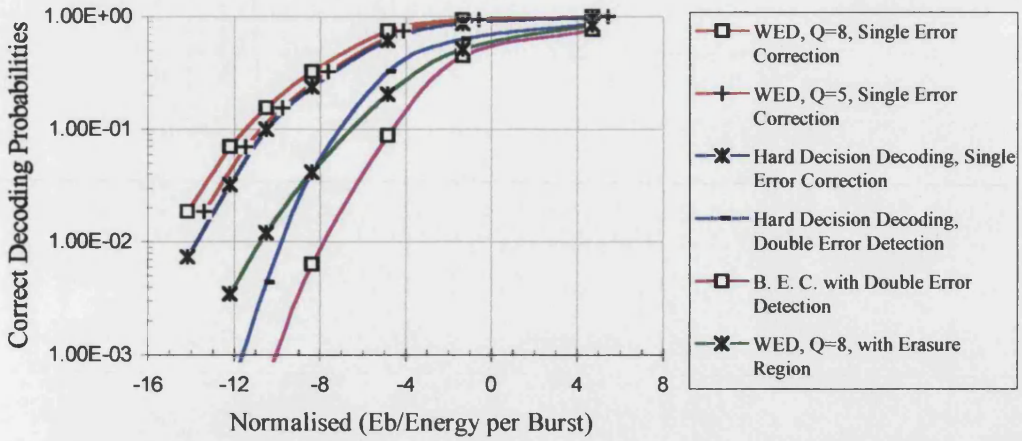


Figure 13.25.Correct Operation Curves for Interleaved (7,4) Hamming Code with Hard Decision and WED Soft Decision Decoding, with Erasure Region.

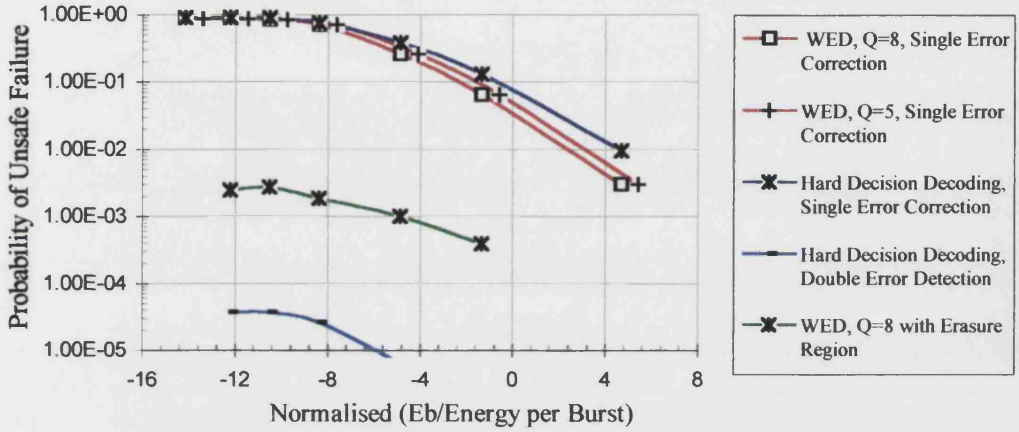


Figure 13.26.Unsafe Failure Curves for interleaved (7,4) Hamming Code with Hard Decision and WED Soft Decision Decoding, with Erasure Region.

Note that the performance of the interleaved (7,4) code, with Q=8, WED and an erasure region is comparable to that of the RS (10,6) concatenated code with double error correction (figures 13.7, 13.8 and 13.9). If the width of the erasure region is increased, the performance of this code can be biased even more toward safety.

13.3.7 Minimum Soft Distance Decoding

Binary decoding involves the quantising of the demodulator output into binary symbols, and assumes the transmitted code word is the code word with the lowest *Hamming distance* to the received code word (see section G.2). Decoding algorithms (e.g. BCH or parity check matrix) eliminate the need for comparing the Hamming distance of all the valid code words, and allow fast decoding of long codes. The optimum minimum soft distance decoding method (MSDD) utilises unquantised reliability data for each symbol to calculate soft distances. The valid code word with the lowest soft distance from the received signal is then chosen as the transmitted code word. This procedure is only practical for short codes since it involves comparing the

soft distance between each code word. This section presents a minimum distance soft decision method which is applicable to the interleaved (7,4) Hamming code described in section 13.2.8.

13.3.7.1 Description of Method

As a simple example, consider an n=2 code, with code words (0,0) and (1,1). The 2 dimensional code vector space is shown below. If the received code lies on the point shown as '*' on figure 13.27, binary demodulation will result in the uncorrectable code word (0,1). However, comparison of the soft distances a and b, to the valid code word positions will result in correct decoding. We could liken this process to the maximum likelihood demodulation of Q-ary symbols in a 2 dimensional signal space.

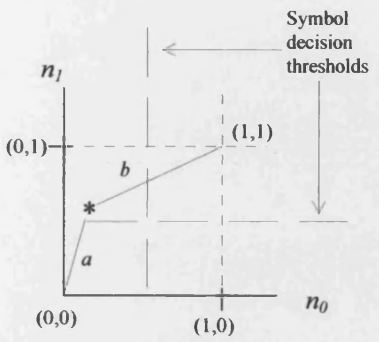


Figure 13.27: Vector Space for n=2 code word

The trade off between error correction and error detection is no longer limited to integer values of Hamming distance up to the largest integer less than $d_{min}/2$. We can set a radius, r, around each point, where $0 \leq r \leq d_{min}/2$. The code word radius containing the received code word point is assumed to be the transmitted code word. If the received code word point lies outside all the code word radii then an undetectable error has occurred. This situation is shown in figure 13.28. Increasing the value of r will increase the likelihood of error correction and decrease the likelihood of error detection, and vice versa. This vector space is extended to n>2 dimensions for practical codes.

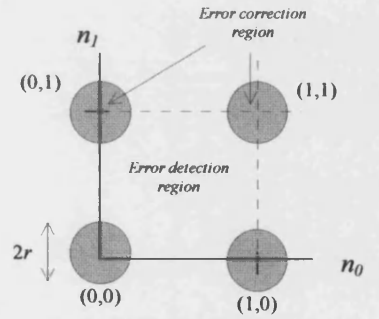


Figure 13.28: Vector Space for n=2 code word

For maximum likelihood minimum soft distance decoding (the best theoretical performance) to be achieved each received symbol needs to be assigned a reliability number [112]. If the channel statistics are known and stationary, the reliability of each symbol is given by the log likelihood ratio (see section G.1.1) [113]. As the channel statistics are not known, the reliability figure of each symbol is made proportional to the distance of the received symbol point to the nearest binary data point. The performance penalty for using this simple reliability measure is likely to be negligible [114].

13.3.7.2 Decoding Algorithm

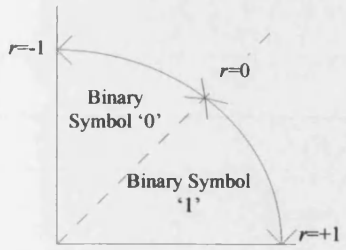


Figure 13.29: Demodulated Angle and Reliability Number

The demodulated data for each bit consists of an angle. This is the angle of the resultant of the sidebands to the carrier. For each code bit, we will assign a number, r_i, $0 \leq i \leq (n-1)$, ranging from -1 (corresponding to the '0' binary symbol) to +1 (corresponding to the '1' binary symbol), as shown in figure 13.29. For the whole code word, this will result in the received reliability vector $R=\{r_0, r_1, r_2, \ldots r_{n-1}\}$. For an (n,k) binary code, the receiver must store 2^k vectors, C_i, $0 \leq i \leq (2^k-1)$, corresponding to the valid code words. For each code word, the soft distance, D_i, from the received code is calculated as:

$$D_i = \frac{n-(C_i \bullet R)}{2} \qquad (13.18)$$

where $0 \leq i \leq (2^k-1)$, and $\bullet$ represents the matrix dot product, or *scalar* product. The code word, C, with the minimum soft distance, D_{min} (also known as the 'minimum squared Euclidean distance' [115]) to the received reliability vector, R, is chosen as the transmitted code word, only if D_{min} is smaller than the error correction radius of figure 13.28. We can see that if the received vector, R, lies exactly on a code vector, C, then the soft distance is the same as the Hamming distance.

13.3.7.3 Code Performance

The performance of the interleaved Hamming (7,4) code has been assessed with minimum distance decoding. Each single code has only 16 code words, which limits the minimum distance search time. Figures 13.30 and 13.31 show the curves for correct operation and unsafe failure, with r = 1, 1.1, 1.2 and 1.5. Also shown, for comparison, are the curves for hard decision decoding (single error correction and double error detection) and the RS (10,6) concatenated code with single error correction.

These curves show that a larger error correction radius, r, increases the reliability (correct operation curve) at the expense of the unsafe failure rate. Note that the correct operation curve for r=1.1 shows a similar reliability to the RS (10,6) concatenated code with single error correction, while the unsafe failure curve is actually lower. This decoding method is efficient, and any value of r can easily be chosen to give the desired error correction/detection trade off.

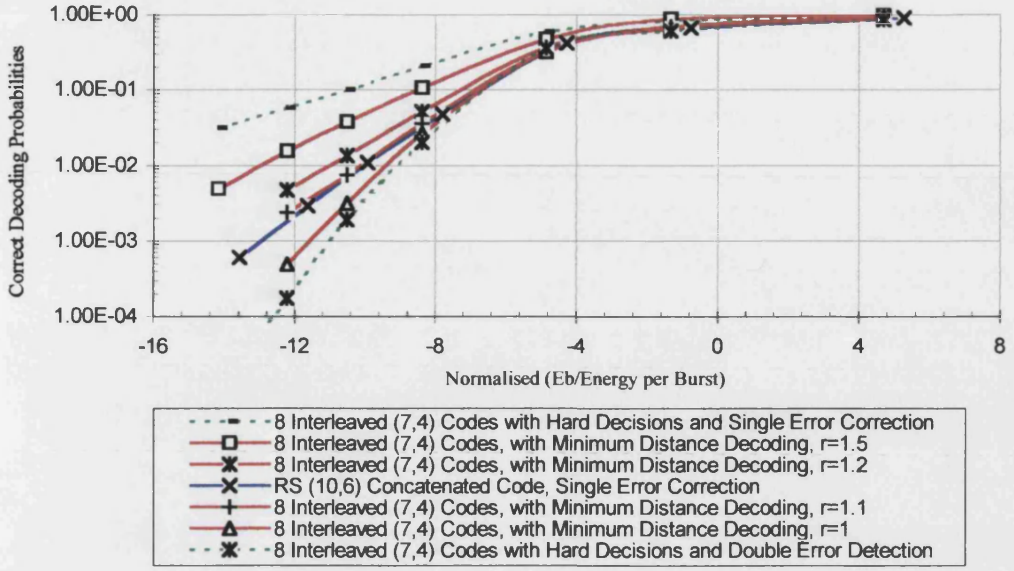


Figure 13.30. Correct Decoding Curves for the Interleaved Hamming (7,4) Code with Minimum Soft Distance Decoding.

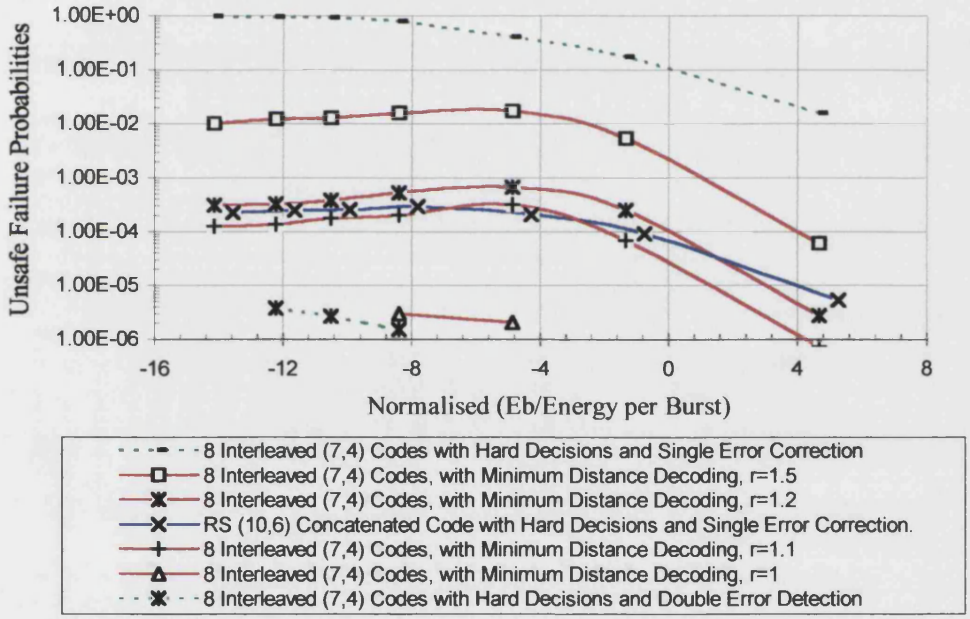


Figure 13.31.Unsafe Failure Curves for the Interleaved Hamming (7,4) Code with Minimum Soft Distance Decoding.

13.4 Conclusions

Code Performance with Hard Decisions

There is no straight forward figure of merit when the results in figures 13.7, 13.8 and 13.9 are considered. A high tolerance/immunity to interference is attractive to rail authorities since this translates to less safe failures and fewer train delays. However, safety must be considered the most important factor. The relation between the unsafe failure probability of the codes presented in figure 13.9 and the railway MTBUF is impossible to state since it is also

dependent on many unpredictable factors, i.e. position of trains, equipment malfunction, duration of interference. Whatever the encoding/decoding philosophy, the trade off between safety and reliability is inevitable and so the final choice of coding scheme will depend as much on experience, and the results of an overall fault tree analysis as it will on rigorous coding theory.

This section has considered some codes with no error correction. The use of error correction in railway environments is generally discouraged or not accepted. This philosophy applies well to small data capacity systems but with longer codes, the increase in the time/bandwidth requirement means that a single data block is more likely to be affected by errors, and reliability is seriously affected. If a scheme is chosen carefully, error correction used simultaneously with error detection can improve system reliability whilst still meeting the safety requirement. For instance, consider the RS (10,6) concatenated code with single error correction and no error correction (i.e. only error detection). From figure 13.7 we see that use of single error correction has increased the reliability by a factor of 10. The peak decoding error probability is approximately 2.10^{-4}. With random data the decoding error probability is $(42.6).10^{-6}$. This could be lowered further by using some of the techniques described in section 13.5.

For the interleaved codes, the decoding of the total data block is dependent on all the individual codes. So, a detected error in one word will result in a safe failure for the whole data block. Also, for the results presented, a decoding error in one word will result in an unsafe failure for the whole word. It is possible to separate the decoding of different types of data, so that, for instance, and error could be detected in non-vital data, while speed restriction commands can be decoded independently. There is an infinite number of possible combinations.

The interleaved Hamming codes have simple algorithms and provide an easy method of achieving efficient burst error control. Note that the interleaved Hamming (8,4) code with single error correction achieves a similar performance to the RS(11,7) code with double error correction, with much simpler encoding/decoding algorithms. However, for the different configurations of interleaved Hamming codes, each individual code is capable of only single error correction, and the results have shown that they are either heavily biased toward safety or reliability (i.e. the use of no error correction provides very low tolerance to interference, and single error correction provides low safety against a decoding error). The unsafe failure curve of the interleaved Hamming (8,4) code quoted above is unacceptably high. The Reed Solomon codes have more powerful error control properties and so more error detection/correction combinations are possible allowing the desired trade off between safety

and reliability. Of the hard decision codes considered, the (10,6) RS concatenated code with single error correction provides the best performance for track circuit applications.

Out of Band Power Monitoring

This technique can reduce the unsafe failure rate of a code by estimating the integrity of the data from calculation of the power above and below the signal. If the estimate is accurate, the unsafe failure rate will decrease and the safe failure rate will increase. If the calculated 'out of band' power is above the threshold, and a correctable error pattern has occurred, then the safe failure will occur at the expense of correct decoding. We can see from figures 13.13 and 13.15 that the technique is successful in increasing the safety of the coding schemes. From figures 13.14 and 13.16 it may first appear that the reduction in reliability is too high for this technique to be efficient. However, we notice from figure 13.31 that at E_b/(Energy per burst) lower than -8dB, the bit error probability is such that more that 8 bit errors are likely to occur in the block of 56 bits, causing an uncorrectable error pattern. Since reliable operation in 'railway' terms requires consistent correct decoding, there is little real advantage in increasing the probability of correct decoding from, say, 10^{-4} to 10^{-3}. If we say that reliability becomes affected when the correct decoding probability falls to 0.8, we see from figures 13.14 and 13.16 that the use of 'out of band power' monitoring provides a negligible reduction in reliability for a considerable increase in safety.

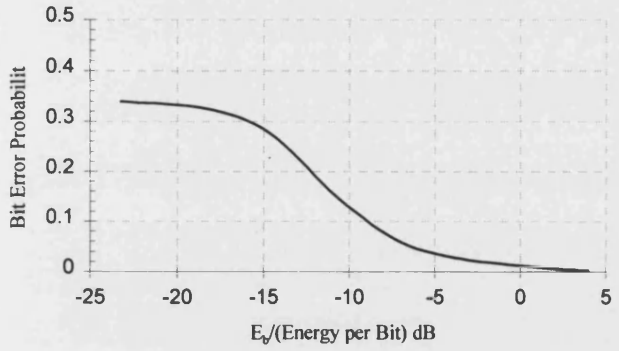


Figure 13.31 Bit Error Probability against SNR for Multi-tone signal with Impulsive Interference

This scheme will provide a greater increase in performance if the interference source is Gaussian, since the variance of each bin is the same, and the 'out of band' power will be a more accurate estimation of 'in band' noise.

Comparison of figures 13.9 and 13.15 shows us that this technique does not increase the safety of the interleaved Hamming codes to a level comparable to the RS (10,6) codes. We can conclude that, with impulsive interference, this technique is valuable for preventing erroneous decoding errors at low SNRs and decreasing the unsafe failure rate, but cannot compete with a more powerful error detecting code.

Weighted Erasure Decoding (WED)

The use of Q=8 'soft' demodulation with WED has been shown to offer an approximately 1.1dB performance gain in terms of reliability (figure 13.22). However, error detection must be incorporated into the overall decoding philosophy, and a method is presented (figure 13.24) which combines an erasure region with Q=8 WED. This method effectively reduces the unsafe failure rate (figure 13.26) at the expense of a 3-4dB loss in reliability (figure 13.25). This performance is only marginally worse than the RS (10,6) code with double error correction. It is likely, however, that the use of WED with an erasure region for error detection of short codes, is not the most efficient use of soft decision methods. This is because the WED algorithm is designed to optimise reliability by making use of the regions of least confidence to determine which errors are corrected.

The amount of computation involved in this technique is comparable to the RS code, although there is more repetition and therefore less complexity. For the Q=5 and Q=8 schemes, each received (7,4) code word is represented by the (3 x 7) 'a' matrix. The Hamming (7,4) decoding algorithm must be performed on each row of this matrix, and then each bit decision will be performed by equation G.20. Since this technique is based on the original hard decision decoding algorithm, it is applicable to larger codes.

Minimum Distance Decoding

Minimum distance decoding makes full use of the demodulated symbol data in the decoding process, and so no information is lost through symbol quantisation. The method is limited to shorter block codes since the algorithm must compare the distances of each valid code word. The use of minimum distance decoding provides efficient decoding without the use of complex algorithms (such as the RS decoding algorithm). The performance of the interleaved (7,4) codes is no longer limited by the minimum Hamming distance to extreme safety (low reliability) or reliability (low safety). Any value of error correction radius can be chosen, where $0 \leq r \leq d_{min}/2$. We can see from figures 13.30 and 13.31 that the code performance with r=1.1 rivals that of the RS (10,6) concatenated code with single error correction and hard bit decisions.

We could liken this technique to the FSK phase check algorithm used in the Westinghouse FS series of track circuits [103]. The signal here consists of three tones. The resultant phasor of the two sidebands should ideally be at an angle of 90° to the carrier. The algorithm accepts the signal if the resultant angle is within a tolerance of ±45° of the ideal. It is obvious that this tolerance must be greater than zero. It is also obvious that the value of r in figure 13.28 must be greater than zero, as the demodulated data will not correspond exactly with the ideal code word points. There is a direct relationship between the setting of the error correction radius of

figure 13.28 and the use of error correction with hard decisions. The use of unquantised bit information allows the amount of error correction to be set to non integer values. This shows the railway industry's rejection of 'error correction' to be a purely conceptual dislike. The use of 'error correction' is simply saying: 'given component tolerances, a noisy environment and channel distortion, by how much is a valid received signal likely to differ from the ideal', and setting the receiver tolerances accordingly. This occurs in all existing track circuit schemes.

This is the simplest algorithm considered. The complete set of code words must be stored in the receiver and Equation 13.18 must be performed a maximum of 2^k times, for each interleaved code. The (7,4) Hamming code has only 16 code words, which makes this algorithm easily practical.

Summary

Various coding schemes have been chosen to provide error control when the impulsive noise interference model is used with the multi-tone signalling scheme of chapter 12. The results are presented in terms of safety (decoding error probability) and reliability (correct decoding probability). The shortened RS code has the most complex decoding algorithm. It has powerful error correction capabilities, and of the hard decision decoding schemes considered, the RS(10,6) concatenated code with single error correction provides good performance with an adequately low unsafe failure rate. The interleaved Hamming codes are efficient, (note that the interleaved (7,4) code with double error detection provides the same performance as the RS (11,7) code with only double symbol error detection) but the limited amount of error correction in each code gives us little control over the error detection/correction trade off. Traditionally this would mean that a longer, more complicated code must be used, but the techniques described in sections 13.3.5, 13.3.6 and 13.3.7 will transfer information from the demodulation process to the decoding process. This allows a more subtle trade off between error correction/detection than is provided by binary error control techniques alone. These techniques can improve the code performance, whilst still using simple decoding algorithms.

A method of monitoring the out of band power is described in section 13.3.5. This technique will indicate a safe failure if the out of band power is above a certain threshold. Correct interpretation of the results shows a considerable increase the code performance. At the lowest SNR simulated, the signal is 'swamped' by interference and correct decoding is not possible. In this case we are concerned only with reducing the unsafe failure rate. At these SNRs, this simple technique offers a substantial increase in code safety.

Weighted erasure decoding (WED) [114] has been shown to increase code performance in terms of reliability by approximately 1.1dB. To increase the safety of the scheme, a method of

combining an erasure region with WED has been presented. The width of the erasure region will affect the error detection/correction trade off. A code performance similar to that of the RS (10,6) concatenated code with single error correction is achieved. However, incorporating an erasure region with WED is not the most efficient used of 'soft' demodulation, since the WED algorithm is designed to optimise reliability by making use of the regions of least confidence to determine which errors are corrected.

The minimum distance decoding scheme, presented in section 13.3.7 is the most efficient and also the simplest decoding scheme simulated. Decoding requires searching through the code word set, and so the technique is not practical for long codes. Full use is made of unquantised symbol data[2] and the desired safety/reliability trade off is easily set by the value of r. This means that short interleaved codes can be used, since the amount of error correction is no longer limited to integer values of Hamming distance. The performance of the interleaved Hamming (7,4) code with minimum distance decoding has been show to be superior to the RS (10,6) concatenated code with single error correction. Applying the out of band power monitoring technique will further improve performance.

13.5 Further Techniques

Successive FFT results

The probability of decoding error for the codes could be reduced further by taking successive FFTs and requiring that the results agree. This technique is used in the Westinghouse FS2000-FS2700 track circuits. It has the disadvantage of increasing the response time.

FFT redundancy processing techniques.

Redundancy can also be used in the demodulation of the code. For example, redundancy processing for the FFT can take say, k FFTs and average the results. If the FFTs are overlapped, windowing can be applied to each FFT without smearing produced by the widening of the main lobe (p393, [99]). If an interference tone existed between two FFT bins, the windowing will reduce the spectral leakage of this tone into all the other bins. In the presence of interference, this averaging reduces the variance of the FFT results. If the FFTs are not overlapped, then the variance is reduced by an amount equal to the reciprocal of the number of FFTs. If the FFTs are overlapped, the variance is reduced by a smaller amount (p397, [99]).

If the interference is ergodic in nature (that is, time averages equal ensemble averages (p379, [99]), then the more FFTs that are averaged, the lower the variance on the final result. If the

---

[2] The symbol is represented by binary word (e.g. 16 bits) rather than a single bit.

interference is stationary (i.e. a single tone), then averaging the FFT results will not reduce its effect (here, data coding would have to be used). If the interference is a tone whose frequency is sweeping through the band of interest, then a time average of many FFTs will improve the demodulation.

Combinations of time averaging and coding could be used. For example, the portion of the signal corresponding to the 'track to train data' could have a large amplitude and use a block code for error control. The portion of the signal corresponding to 'Full Track Circuit Identity' could have a lower amplitude (to save power and control the envelope) and use the results from successive time averaged FFTs.

Sequencing of track circuit identities

If the train stores a 'map' of the sequences of the local and full track circuit identities (table 10.1) it could check the decoding at each track circuit boundary. This would greatly increase the safety of any track circuit coding scheme, since a decoding error would be detected if the received track circuit identity does not correspond to stored sequence. The memory requirements of the on board train equipment are greatly increased with this technique.

14. Conclusions

A proposal for a new track circuit transceiver termination has inspired this study into improvements in track to train communications. The transceiver design offers many potential ATO/ATP features, described in chapter 4, which would require a much greater signal data capacity to be realised. Figure 1.5 shows how the thesis has developed and the subject area of each section. The work on track circuit data modulation and coding is split into two sections. Section II (chapters 7 to 9) is aimed at increasing the data capacity of the track circuit signal by using a serial binary data stream, taking advantage of the potentially wider bandwidth offered by the new transceiver termination design. Section III proposes a multi-tone modulation and coding scheme which fulfils the maximum data requirements, with the restriction that the signal may have to be fed through the existing tuned terminations. This is necessary to achieve a gradual upgrade path, and places an initial bandwidth restriction of 200Hz on the signal, but these schemes can easily be extended to a wider bandwidth.

14.1 Error Control

The coding scheme results in sections II and III suggest that the performance improvements when using more complicated and algebraically efficient codes are negligible or non-existent, and that the use of codes designed for efficient error *correction* is inappropriate for track circuits. This is because the amount of error correction is limited by (a) the comma free requirement for serial binary codes and (b) the requirement for error detection. When there is a large bias toward error detection, simpler coding schemes can be designed to perform equally as well as the more algebraically perfect codes.

A simpler scheme is also preferred for track circuits, since this will lower the equipment complexity and cost and facilitate the safety case.

It has been demonstrated that, for a higher data capacity track circuit signal, a small amount of error correction is appropriate when the overall system safety is considered. For example, the results of section II have shown that error correction can greatly reduce the safe failure probability of the decoding scheme while a decoding philosophy requiring the multiple reception of data blocks can produce very low unsafe failure rates.

14.2 Section I

The transceiver termination design and its potential advantages are discussed in chapter 4. Chapter 5 introduces a new method of mathematically modelling a track circuit with a

moving train shunt. This 2-wire model is based on the transmission line theory of the chapter 3. The simulation results demonstrate the improvement to train detection offered by the transceiver track circuit. An analysis of the train detection performance of existing track circuit designs is first presented. It is shown how the receiver is least sensitive at the track circuit ends, and how the detection is improved with a higher termination impedance. When extreme weather conditions are considered, it is shown that any receiver threshold setting may result in a considerable detection dead zone near the track circuit extremes.

A measure of train shunt sensitivity is defined and it is shown that, when the same conditions are applied to the track circuit with transceiver terminations, this detection dead zone is eliminated. Since the transceiver termination forms overlapping track circuits and can detect the presence of a train shunt by monitoring its own output current, there are four ways a train can be detected, rather than just one. Even though the termination impedance is low, train detection is greatly improved since the transceiver's output current rises sharply as the train approaches. It is also shown how there is little variation in the output current with environmental conditions as the train approaches. This is encouraging, since further research is proposed into the measurement of the train position within the track circuit.

In chapter 6 an impulsive interference model is presented, which will more accurately model the expected railway interference than AWGN.

14.3 Section II

Section II is concerned with applying serial binary data to track circuits in order to increase the data capacity. The comma free cyclic coding scheme described in chapter 7 has certain obvious advantages over previously used schemes. The same redundancy is used for synchronisation and error control, so there is no need to design special synchronisation sequences. Well known cyclic encoding/decoding algorithms can be used such as the BCH code. These codes have good algebraic properties which make them efficient codes for error control. Coset codes with high indices of comma freedom have been found and a method of assessing their synchronisation performance with random errors using the *overlap weight distribution* is presented. Expressions are derived for the synchronisation error performance of the coset code with t error correction, with a receiver decision based on the reception of 2 consecutive valid code words. It is shown how error correction can improve system reliability, especially for long codes, without realistically affecting safety. For example, consider the (31,11) BCH code (see section 8.6.3). At a bit error probability of 0.001, the use of double error correction has decreased the safe failure rate from 0.08 to 10^{-5}, and increased the unsafe failure rate from approximately 10^{-34} to 10^{-28}. An increase in such

immeasurably small unsafe failures is not likely to have much real significance. The advantages in reliability however, are immediately apparent.

Many different cosets were studied in order to find the best overlap weight distribution. It is interesting to note that in many cases the best performance is achieved when $p(X)$ is defined by a Barker sequence.

The data allocation within the code has not been prescribed in section II. The (31,11) BCH code has 11 data bits. Data could typically be spilt into track circuit identity (e.g. 6 bits) and track to train data (5 bits). The data capacity could be increased further by splitting the message into two or more codes. The increase in required bandwidth and signal distortion would have to be further studied.

Chapter 9 presents results of a simulation of the coset code performance, with non-coherent FSK demodulation. The two primary aims of this simulation are (a) to validate the theoretical equations for code performance in Gaussian noise derived in chapter 8, and (b) to assess the performance of these codes in interference typically encountered in railway environments. The code performance simulation is performed separately for correctly framed words and misframed words, and the results for Gaussian noise agree well with the predicted curves. When the simulation is performed with the non-Gaussian impulsive noise we see that the reliability curves for correctly framed words and the synchronisation failure curves are relatively flat, when plotted against SNR. This is due to the nature of the impulsive noise. Increasing the SNR will produce little change in the error patterns until a threshold is reached where the noise is not strong enough to produce bit errors. The results for correctly framed (21,9) and (31,11) BCH code words suggest that the use of error correction is a more efficient method of increasing the decoding reliability than increasing the SNR.

When the code failure probabilities are plotted against overall bit error probability we see that the peak synchronisation failure probabilities are comparable, although the curve decays more rapidly for impulsive noise. Similarly, the peak unsafe failure curves for the (21,9) and (31,11) correctly framed words are comparable. For the (21,9) code, the curve for impulsive noise decays with a slight bias toward safety when compared to the Gaussian noise case. For the (31,11) code the correctly framed performance curves decay with a slight bias towards reliability when compared to the Gaussian noise case.

It can generally be seen that, for determining code safety under bad SNR conditions, an assessment of cyclic coset codes in Gaussian noise is valid. For lower bit error probabilities it can be seen that the error patterns caused by impulsive noise will result in fewer instances

of false decoding on misframed words, which will improve the synchronisation performance.

It can be concluded that the BCH coset codes perform better than the other codes considered. Of the these codes, the (31,11) BCH coset code gives the best performance and the highest data capacity. The repetition codes achieve a higher index of comma freedom at the expense of the correctly framed minimum distance and data rate, and so there is no overall advantage in performance. Codes specially constructed for burst error correction such as the (21,9) cyclic code, provide little advantage. Burst error correction is not feasible due to the comma free and the safety requirement.

14.4 Section III

Chapter 10 introduces the initial data, time response and bandwidth constraints that apply to the transceiver terminated track circuit signal, in order to achieve a gradual upgrade with existing equipment. The most restricting of these requirements is the signal bandwidth limitation to 200Hz. This is the maximum signal bandwidth that can be fed through existing tuned track circuit terminations. Chapter 10 describes some traditional binary and M-ary modulation schemes and compares their performance in terms of required bandwidth and error rate in Gaussian noise. This traditional rating of modulation schemes, however, may not be appropriate for track circuits. The signal is primarily a train detection signal, and must be of sufficient power to overcome the impedance of the track under all conditions and contamination on the wheel-rail interface. Hence, rating a potential modulation scheme in terms of minimum power for a fixed error rate may not be relevant. Also, the interference is likely to be impulsive or bursty, which will tend to block out the signal completely for the duration of the impulse ($P_e \approx 0.5$) and have low P_e for other times. Hence, the redundancy and coding may be more significant in increasing noise immunity than the modulation scheme alone.

Equipment complexity and cost are important factors. For example, non-coherent FSK and MFSK can be implemented with fairly simple DSP and DFT receiver structures while MSK requires a complex fully coherent receiver.

It is shown in chapter 10 that, once synchronisation and error control redundancy are taken into account it is not possible to meet the data requirement within the time/bandwidth restrictions, even for the most bandwidth efficient of the modulation schemes considered. In the following 2 chapters, two modulation and coding schemes are proposed, which aim to fulfil the data requirements.

For the novel multi-tone modulation scheme discussed in chapter 12, the receiver design can be seen as an extension of the existing FS series FFT demodulation, which will greatly aid the scheme acceptance. The modulation scheme has no phase transitions. The signal could be spread across a high bandwidth and can tolerate a high phase and amplitude distortion, because the information is contained in the relative phases of adjacent tones. Each subchannel is demodulated independently, assuming orthogonality is preserved. This is an important advantage, especially since the signal may have to be fed through existing track circuit terminations. The tuning of these terminations to one frequency will introduce phase and amplitude distortion additional to that produced by the railway track. Whereas serial binary/*M*-ary modulation schemes (such as the MFSK scheme of the previous chapter) may suffer group delay distortion and ISI from the restricted bandwidth, the multi-tone signal can be considered immune to such effects.

The performance of block coding techniques for the multi-tone signal are discussed and assessed in chapter 13, using the impulsive noise model described in chapter 6. Several codes are chosen for simulation.

The coding results of these schemes with hard bit decisions indicate that the interleaved Hamming codes offer good performance in comparison to the RS codes. However, since the error control capability of the individual codes is small, there is little control over the safety/reliability trade off. Longer, more complicated codes have the advantage that a small amount of error correction can be implemented, with the majority of the redundancy fulfilling the track circuit safety requirement. Since the complexity of the decoding algorithm has an effect on the equipment cost and the overall safety case, the simplicity of the interleaved Hamming code decoder is desirable. Hence, techniques are explored in chapter 13 in which information is passed from the demodulation process to the decoding process. Out of band power monitoring is a simple and effective method of assessing the integrity of the demodulated data by measuring the power in the frequency bins surrounding the data. When the signal is corrupted by noise, or the orthogonality of the tones is lost, these bins will contain interference power. If this power is above a certain threshold, then the receiver can indicate a safe failure. This technique dramatically reduces the probability of unsafe failure of the coding schemes at low SNRs.

A minimum distance soft decision decoding algorithm for the multi-tone signal is presented. With this algorithm, the error correction is no longer restricted to integer values of Hamming distance and any value of error correction radius can be set, allowing any combination of error detection/correction. The results show that the safety and reliability of the interleaved (7,4) Hamming code with minimum soft distance decoding can be

comparable to that of the RS concatenated codes, without the considerable complexity of an RS decoder. The unsafe failure rate is also considerably lower than the RS concatenated codes, and so the interleaved Hamming (7,4) code with minimum distance soft decision decoding is recommended for this multi-tone modulation scheme.

The multi-tone signal with minimum distance decoding is flexible, since the code length, redundancy and error correction/detection can easily be set according to specific requirements. The results presented apply to decoding with a single FFT. However, a simple method of increasing the error detection and further lowering the unsafe failure would be to base the receiver decision on two or more consecutive FFT results. These FFTs could be overlapping. Non safety critical data, with a longer decoding time could be averaged from successive FFTs, reducing the noise variance.

14.5 Serial Binary Modulation vs. Multi-tone Modulation

This thesis has presented a study and simulation of two separate coding and modulation schemes: the serial binary coding scheme (section II) is suitable for a higher bandwidth track circuit signal, and the multi-tone modulation scheme is suitable for a track circuit with bandwidth restrictions and signal distortion due to the termination design.

The traditional comparison of signal to Gaussian noise power for a constant *Pe*, for the multi-tone technique and non-coherent FSK modulation, presented in section 12.5, suggests that the choice of the scheme involves a trade off between power and bandwidth. However, it is also shown that, since the transmitter output power is high, as defined by the train detection function, and the noise is likely to be high amplitude, impulsive and bursty, this traditional comparison is likely to be misleading. The restriction on bandwidth, backwards compatibility with existing equipment, transmitter/receiver complexity, cost and the industry's acceptance will be major factors affecting in the choice of scheme.

The data requirements (table 10.1) were produced [52] to promote research into modulation and coding techniques which will greatly increase the data capacity of existing track circuits, within the bandwidth constraints. It is likely that, during the development, these data requirements may change. An increase in required data capacity, within a constant response time, implies a higher signal bandwidth for the serial binary coding scheme, and this will be possible for a higher frequency transceiver track circuit. A data message could be split into several cyclic coset code words. Although the transmitter power is high, the extra power required to increase the number of tones for the multi-tone modulation scheme may be prohibitive, and so this scheme would be recommended for the data capacity quoted in table 10.1 or below.

14.6 Future Work

The track circuit transceiver termination has many potential features which would require further research:

- If each track circuit could self initialise, this would eliminate the time consuming task of manually setting each receiver threshold.
- The change in impedance of the track as a train shunt approaches could be calibrated against distance (within the tolerance of the track parameters). This distance information could be used for decreasing headway or decreasing the number of track circuits needed in a section. If the trackside equipment located the position of the train within the track circuit, the speed commands to the train could be altered as it moves through the block. Here the situation would be more like a moving block system, where the need for an overlap track circuit between the train and an obstacle is eliminated and headway in reduced. Another possible method of achieving this reduced headway is to have the train calculate its own braking profile from a knowledge of its position within the track circuit. The train would need to know the number of track circuits from the obstruction, the track circuit length, the gradient of the track circuit etc.
- When the train is close to the transceiver, a burst of non-safety critical data could be transmitted at high speed (i.e. 40-100kHz) to the train.
- Since the transceiver can be designed to present a high impedance across the rails when powered down, fault tolerance could be achieved in the event of a transceiver failure, since the two adjacent transceivers could form one, larger, track circuit. The handshaking requirement would need to be studied.

The next stage for the serial binary cyclic coset code decoder is to build a prototype and assess its performance on an actual track circuit, or a channel model of the rail track. A higher data capacity requires a higher bit rate and a wider bandwidth. The track distortion needs to be assessed for a wider bandwidth signal.

The block coding analysis of the multi-tone signal (chapter 13) has been based on the reception of a single code block. It has been seen in chapter 8 that a receiver decision based on the reception of two consecutive code blocks can dramatically improve the safety of the signal. This same technique could be used to increase the noise tolerance by averaging out the FFT results, either over consecutive or overlapped FFT results. Also, non safety critical data, such as the 'track circuit location', or geographical data, could be averaged over many FFT results, since the detection time available is longer.

15. References

[1] Mellitt, B.: 'Electromagnetic Compatibility in d.c. Railways using Power Electronic Traction Control', IRSE International Conference Proceeding, 1984, pp616-621.

[2] Hill, R. J.: 'Optimal Construction of Synchronizable Coding for Railway Track Circuit Data Transmission', IEEE Trans. on Vehicular Technology, Vol.39, No.4. Nov 1990, pp390-399.

[3] Hill, R. J.: 'The Automation of Railways', Physics in Technology, Vol. 14, No.1, 1983, pp37-47.

[4] Bradley, D. N.: 'Track Circuiting for Electrified Lines - The Specification of an Unfriendly Environment', Proc Aspect 1991, pp470-484.

[5] Norton D. J.: 'Blockjointless Track Circuit - London Transport', Proc. IRSE, 5 March 1975, pp111-128.

[6] Hollands, R. D., Barton, D.: 'Continuous Automatic Train Control with Microprocessors', Proc IRSE, 13 Jan 1987.

[7] 'Cab Signalling for Italy', Modern Railways, Vol.28, No.279, December 1971, p241.

[8] Weir, R. S.: 'Moving Block Signalling offers Cost Savings', Railway Gazette International, Sept 1994, pp19-20.

[9] Clifton, R. M., Haywood, C., March, A. P.: 'Existing Westinghouse FS2000 ATC System', Westinghouse Signals Document No TBS10-20-1, 12 Feb, 1991.

[10] Hill, R. J.: 'Train Position Detection and Track-Train Data Transmission Using Audio Frequency Track Circuits', Journal of Electrical and Electronics Engineering, Austrailia, Vol.14, No.1 Jan 1983, pp37-47.

[11] 'Track Circuit Overviews', Westinghouse Signals Document, No. D00009DB, 22 March 1993.

[12] Nishinaga, E., Evans, J. A.: 'Wireless Automatic Train Control', Proc. IEEE/ASME Joint Railroad Conference, 1994, pp31-46.

[13] Sjöberg, A.: 'Automatic Train Control', Ericsson Review, Vol.58 No.1, 1981, pp22-29.

[14] Isihara, T.: 'Railroads: Green Lights for the Bullets', IEEE Spectrum, Vol.14, No.9, Sept 1977, pp71-74.

[15] Hope, R.: 'Satellites Track the Trains', Railway Gazette International, March 1993, pp171-172.

[16] Poole, D., Barker, D.; 'Digitally Coded Track Circuit', IRSE International Conference on Railway Safety, Control and Automation towards the 21st Century, London, 25-27 Sept 1984, pp333-336.

[17] Stanley, P. W.: 'Design for Signalling System Performance', Railway Engineering International, 1980, Vol.5, pp28-30.

[18] Hadaway H. W.: 'Fail Safe' Proc. IRSE, 1966/1967, pp160-196.

[19] 'Advanced JTC Study - Track Connections', Westinghouse Signals Ltd Document.

[20] Brown, C. R.: 'A Review of Jointless Track Circuits', Proc. IRSE, 7 Feb, 1985.

[21] Short, R. C.: 'Irregular Operation of Track Circuits by Modern Diesel Multiple Units'

[22] Nock, O. S.: 'Railway Signalling', A & C Black, London, 1980.

[23] Frielinghaus, K. H. ; 'Contingencies in the design of the audio Track-Circuit', Rail Engineering International, Vol. 4 No 4 May 1974, pp182-188.

[24] Kalra, P. ; 'Track Circuits for Modern Rapid Transit Systems', Proc. ASME Intersociety Conf. on Transportation, 23-27 Sept 1973.

[25] Gall, D. C. : 'Railway Track Circuits', Pitman, London 1933.

[26] Grose, B. H. : 'Jointless Track Circuits and Electrified Railways', Tech. Meeting of the Intist. of Electrical Engineers, Nov 15th 1972.

[27] Hill, R. J. , Weedon, D. N. : 'Computer Aided Design of Audio Frequency Track Circuits', IEE International Conference on Electric Railway Systems for a New Century, London, 22-25, Sept 1987.

[28] Hill, R. J., Yu, S. L. : 'Computer Modelling Techniques for Railway Signalling Interference', 15th IASTED International Conference on Modelling and Simulation, Santa Barbara Ca., 27-29 May 1987.

[29] Davidson, C. W. : 'Transmission Lines for Communications', The Macmillan Press Ltd.

[30] MacEntee, J. A. : 'Railway Track as a Short Transmission Line', M.Sc. Thesis, Birmingham University, Sept 1972.

[31] Mellit, B. 'Data Transmission Characteristics of Railway Track', Electronics Letters, Vol.9, No.23, Nov. 1973, pp550-551.

[32] Iancu, O. D.:'Computer Aided Design of Non-Insulated Track Circuits', Rail International, Vol.5, No.6, June 1974, pp423-436.

[33] Iancu, O. D.: 'The influence of track circuit parameter variation on maximum permisible length', Rail International, Vol.7, No.1, Jan 1976, pp39-43.

[34] 'Preliminary Design Investigation Report : JTC System Cable/Track Analysis', WSL Document X1306/21 April 1991.

[35] 'Engineering Test Report : Waterloo and City JTC Track Parameter Measurment Tests', WSL Document X1306/21T10, 26/01/92.

[36] Fisher, A. N.: 'Track, Track Circuits and Traction', IEE International Conference on Electric Railway Systems for a New Century, London 22-25 Sept 1987 pp184-188.

[37] Electrical Properties of Track Laid on Wooden Sleepers and Practical Determination of its Essential Characteristic Data. UIC-ORE Quention A25 Report 6, April 1963. Utrecht: UIC-ORE.

[38] 'CAE and Track Modelling Investigation', WSL Document Number X1306/31/C/D00003.

[39] Iancu, O. D., Giuhat, I. R.: 'Non-insulated track circuits with direct-coupled receiver: analysis relations, method of calculation' Rail International, Vol.7, No.11, Jan 1976, pp638-645.

[40] 'VT1 Replacement Technical Feasibility Study', Westinghouse Signals Ltd, Document No. X1306/34/D00003.

[41] Watanabe I., Takashige T.: 'Advanced Automatic Train Protection System'. Proc IEEE VTC conf. 1994, pp1126-1129.

[42] Hill, R. J., Carpenter, D. C.: 'Rail Track Modelling for Signalling and Electrification System Simulation', Rail Engineering International, Vol.19, No.4, December 1990, pp16-20.

[43] Hill, R. J., Carpenter, D. C., Tasar T.: 'Railway Track Admittance, Earth-Leakage Effects and Track Circuit operation', Proc. ASME/IEEE Railway Conference, 1989, pp55-62.

[44] Watanabe I., Takashige T.: 'Advanced Automatic Train Protection' IEEE 44th Veh. Tech. Conf., Stockholm, June 8-10, 1994 pp1126-1129.

[45] Takashige T.: 'Digital ATP Revolutionizes Train Control System' Quarterly Report of RTRI, November 1992 33(4) pp285-290.

[46] Thakore P., Kanefsky M.: 'On Multicarrier Modulation for Train-to-Wayside Communication' IEEE 44th Veh. Tech. Conf., Stockholm, June 8-10, 1994 pp1117-1122.

[47] Uebel, H.: 'New Developments in the field of Train Detection systems used in Germany', IEE Conference on Railways in the Electronic Age, London, Nov. 1981, pp136-145.

[48] Mokkapati C., Stark D. E.: 'Traction Power and Chopper Harmonics in a Rapid Transit System', 34th IEEE Vehicular Technology Conference, May 1984, pp213-219.

[49] 'AC Traction and Bonding', Technical Note X1306/37/7-8, Westinghouse Signals Ltd.

[50] Holstrom F. R.: 'Assuring Compatability of Rapid Transit Propulsion and Signalling Systems', 34th IEEE Vehicular Technology Conference, May 1984, pp445-453.

[51] Powell K. C.: 'Interference and Propagation Measurements on Madrid Metro' Westinghouse Signals Ltd. Document Number EIR/92/009 (X1358/1/7/6).

[52] 'Communications Techniques Investigation Work Specification', Westinghouse Signals Ltd Document Number X1306/34/D00009.

[53] 'FS2700 Customer Requirement Specification', Westinghouse Signals Ltd Document Number X1306/37/D00004.

[54] Timson W. J.: Visit Report, Brush Traction Ltd., Jan 18th 1992, Westinghouse Signals Ltd, Document No. X1306/31/C/1. Company Confidential.

[55] Lin, S., Costello, D. J.: 'Error Control Coding', Prentice Hall, Englewood Cliffs, New Jersey, 1983.

[56] Lucky, R. W., Saltz, J., Weldon, E.J.: 'Principles of Data Communication', McGraw-Hill, 1968.

[57] Petersen, W. W.: 'Error Correcting Codes', The Technology Press and Wiley, New York 1961.

[58] British Railways Board : 'General Specification for Permitted Electrical Interference Produced by Traction and Rolling Stock', BR1914, March 1991.

[59] Gold, R.: 'Optimal Binary Sequences for Spread Spectrum Multiplexing', IEEE Transactions of Information Theory, Vol. 12, 1967, pp619-621.

[60] Friedlander, G. D.: 'The BART Chronicle', IEEE Spectrum, Vol.9, No.9 Sept 1972, pp34-36.

[61] Friedlander, G. D. ; 'Bigger Bugs in BART', IEEE Spectrum, Vol.10, No.3 March 1973, pp32-37.

[62] Friedlander, G. D. ; 'A Prescription for BART', IEEE Spectrum, Vol.10, No.4 April 1973, pp40-44.

[63] Eastman W. L.: 'On the Construction of Comma-Free Codes', IEEE Trans. on Information Theory, Vol.IT-11, April 1965, pp263-266.

[64] Scholtz, R. A.: 'Maximal and Variable Word-Length Comma-Free Codes', IEEE Trans. on Information Theory, Vol. IT-15, No. 2, pp 135-142, March 1969.

[65] Turner, D. B.: 'Reliability Improvement of BART Vehicle Train Control', IEEE Veh. Tech. Conf., 1979, pp279-288.

[66] Chapront, P.: 'Transmission and Decoding of Vital Data', IRSE International Conference on Railway Safety, Control and Automation towards the 21st Century, London, 25-27 Sept 1984, pp161-165.

[67] Hill, R.J., Weedon, D. N.: 'Safety and Reliability of Synchronizable Digital Coding in Railway Track-Circuits', IEEE Trans. on Reliability, Vol.39, No.5, Dec 1990, pp581-591.

[68] Stiffler, J. J.: 'Comma-Free Error-Correcting Codes', IEEE Trans. on Information Theory, Vol.IT-11, Jan 1965, pp107-112.

[69] Barker R. H.: 'Group Synchronisation of Binary Digital Systems', Communications Theory, W. Jackson (Ed.), Butterworth Scientific Publications, London 1953, pp273-287.

[70] Maxwell M. S. and Kutz R. L.: 'An efficient PCM error correction and synchronisation code', NASA Tech. note TND-2317, National Aeronautics and Space Administration, Washington D.C., June 1964.

[71] Rudin M. B. and Fiorino F.: 'Advancements in the design and evaluation of group synchronisers for PCM telemetry', Record 1962 National Symposium on Space Electronics and Telemetry.

[72] Eastman W. L.: 'On the Construction of Comma-Free Codes', IEEE Trans. on Information Theory, Vol.IT-11, April 1965, pp263-266.

[73] Tong, S. Y.: 'Synchronisation Recovery Techniques for Binary Cyclic Codes', Bell System Tech. J., 45, 1968, pp561-596.

[74] Golomb, S. W., Gordon, B., Welch, L. R.: 'Comma-free Codes', Can. J. Math., vol.10 1958 pp 202-209.

[75] Kendall, W. B., Reed, I. S.: 'Path-invariant Comma-free Codes', IRE Trans. on Information Theory, Vol.IT-8, Oct. 1962, pp 1365-1367.

[76] Frey A. H.: 'Message framing and error control', IEEE Trans. on Military Electronics, April 1965, pp143-147.

[77] Levy, J. E.: 'Self-Synchronising codes derived from Binary Cyclic Codes', IEEE Trans. on Information Theory, Vol.IT-12, No 3, July 1966, pp286-290.

[78] Kasami, T.: 'Weight Distributions of Bose-Chauduri-Hocquenghem Codes', Proc. Conf. Combinatorial Mathematics and its applications, University of North Carolina Press, Chapel Hill, N. C., 1968.

[79] Kasami T.: 'Optimum Shortened Cyclic Codes for Burst Error Correction', IEEE Trans. on Information Theory, Vol. IT-9,pp 105-109, April 1963.

[80] Kasami T., Matoba, S.: 'Some Efficient Shortened Cyclic Codes for Burst Error Correction', IEEE Trans. on Information Theory, Vol. IT-10, pp 252-253, July 1964.

[81] Shanmugam K. Sam.: 'Digital and Analogue Communication Systems', 1985, John Wiley and Sons.

[82] Collins M. J., Martin J. D.: 'Synchronisation of Cyclic Coset Codes for Railway Track Circuit Data', Proc. 1996 ASME/IEEE Joint Railroad Conference.

[83] Collins M. J., Martin, J. D.: 'Performance of Railway Track Circuit Codes with Burst Interference' Proc. IEEE Vehicular Technology Conference 1997.

[84] Gilbert E. N.: 'Synchronisation of Binary Messages', IRE Trans. Information Theory, pp 470-477, Sept 1960.

[85] Campbell, L. L.: 'Two Properties of Pseudo Random Sequences', IRE Trans. March 1959, p32.

[86] Sholtz R. A.: 'Codes with Synchronisation Capability', IEEE Trans. on Information Theory, Vol.IT-12, No 2, April 1966, pp135-142.

[87] Jiggs B. H.: 'Recent Results in Comma-Free Codes', Canadian Journal of Mathematics, Vol.15, 1963, pp178-187.

[88] Gallager, R. G.: 'Information Theory and Reliable Communication.', John Wiley and Sons, 1968.

[89] Hideki Imai: 'Essentials of Error-Control' Academic Press, Inc. 1990.

[90] Benedetto S., Biglieri E., Castellani V.: 'Digital Transmission theory' Prentice Hall 1987.

[91] Oetting, J. D.: 'Comparison of Modulation Techniques for Digital Radio', IEEE Transactions on Communications, Vol. COM-27, No-12. December 1979.

[92] Snorre Kjesbu, 'Digital Communications in Advanced Safety Systems for Trains', Proc. 1996 ASME/IEEE Joint Railroad Conference.

[93] Simon Haykin, 'Digital Communications', John Wiley and Sons, 1988.

[94] Subbarayan Pasupathy: 'Minimum Shift Keying: A Spectrally Efficient Modulation'. IEEE Communication Magazine, July 1979, pp14-22.

[95] Ralphs, J. D.: 'Princlipes and Practice of Multi-Frequency Telegraphy', ISBN 0-86341-022-7, 1985.

[96] Slepian, D.: 'Bounds on Communication', Bell Syst. Tech. J., May 1963, 42, No 3, pp681-708.

[97] Vincent, C. H.: 'Modulo-M parity check in MFSK techniques', The Radio and Electronic Engineer, 1978, 48, No. 5, p248.

[98] Farrell, P, 'Soft Decision Decoding Techniques', Algebraic Coding Theory and Applications, CISM Cources and Lectures No.258, Ed. Longo G., Springer-Verlag Wein, New York 1979.

[99] DeFatta D. J., Lucas J. G., Hodgekiss W. S.: 'Digital Signal Processing' John Wiley and Sons, 1988.

[100] Dr. Kamilo Feher,: 'Digital Communications', Prentice Hall, 1981.

[101] Ferland G., Chan Y. T., Couture F.: 'On the Implementation of a 32-Channel FFT Based MFSK Demodulator', Canadian Conference on Electrical and Computer Engineering, 1994, Vol 1 pp186-189.

[102] Shayan Y. R., Tho Le-ngoc, Vijay K. B.: 'Design of Reed-Solomon (16,12) Codec for North American Advanced Train Control System'. IEEE Trans. Inf. Th., Vol. IT-39, No.4, Nov 1990, pp400-409.

[103] Cook, A.: 'The Detection of Narrowband ASK and FSK Signalling Tones using FFT Techniques' M.Sc. Thesis, The University of Bath, 1984.

[104] Shaw M., Honary, B.: 'New Synchronisation Technique for MFSK Demodulator', Electronics Letters, Vol.25, No.11, pp750- 751, 25th May,1989.

[105] Forney G. D.: 'Burst-Correcting Codes for the Classic Bursty Channel' . IEEE Trans. Com. Tech., Vol. COM-19, No.5, Oct 1971, pp772-781.

[106] Fire P.: 'A Class of Multiple Error Correcting Binary Codes for Non-independent errors', Sylvania Report No. RSL-E-2, Sylvania Electronics Defence Laboratory, Reconnaissance Systems Division, Mountain View, California, March 1959.

[107] Hsu T. H., Kasami T.: 'Error Correcting Codes for a Compound Channel' IEEE Trans. Inf. Th., Vol. IT-14, No.1, Jan 1968, pp135-139.

[108] Burton H. O., Weldon E. J.: 'Cyclic Product Codes' IEEE Trans. Inf. Th., Vol. IT-11, July 1965, pp433-439.

[109] Berlekamp E. R.: 'Algebraic Coding Theory', New York: Mc-Graw-Hill, 1968.

[110] Bate S. D., Honary B. K., Farrel P. G.: 'Soft and Hard Decision Decoding of Product Codes for Communication Systems', Systems Science, Vol.12, No.4 1986 pp79-89.

[111] White H. E.: 'Failure-Correction Decoding', IEEE Trans. Com. Tech.., Vol. COM-15, No.1, Feb 1967, pp23-31.

[112] Forney G. D.: 'Generalised Minimum Distance Decoding' IEEE Trans. Inf. Th., Vol. IT-12 April 1966, pp125-131.

[113] Cahn, C. R.: 'Binary Decoding Extended to Nonbinary Demodulation of Phase Shift Keying', IEEE Trans. Com. Tech., Vol. COM-17, No.5, Oct 1969, pp583-589.

[114] Weldon, E. J.: 'Decoding Binary Block Codes on Q-ary Output Channels' . IEEE Trans. Inf. Th., Vol. IT-17, Nov 1971, pp713-718.

[115] Ungerboeck, G.: 'Trellis-Coded Modulation with Redundant Signal Sets - Part 1: Introduction', Vol.25, No.2, Feb. 1987.

[116] Lindsey W. C., Simon M. K. : 'Telecommunication Systems Engineering', Prentice Hall 1973.

[117] Papoulis, A.: 'Probability, Random Variables, and Stochastic Processes', 1984.

[118] Chien, R. T.: 'Cyclic Decoding Procedures for Bose-Chaudhuri-Hocquenghem codes' IEEE Tranactions on Information Theory, vol. IT10, pp 357-363, Oct. 1964.

[119] Bloom, F. J., Chang, S. S. L., Harris, B., Hauptschein, A., Morgan, K. C.: 'Improvement of Binary Transmission by Null Zone Reception', Proc. IRE, Vol.45, pp963-975, July 1957.

[120] Goran E., Sundberg C.: 'A Note on Soft Decision Decoding with Successive Erasures', IEEE Trans. Inf. Th., Vol. IT-21, Jan 1976, pp88-96.

[121] Nobukazu Doi, Morishi Izumita, Seiichi Mita: 'Soft Decision Decoding for Reed-Solomon Codes', Electronics and Communications in Japan' Part 3, Vol. 72, No.4, 1989, pp72-77.

[122] Heller R. M.: 'Forced Erasure Decoding and the Erasure Reconstruction Spectra of Group Codes'. IEEE Trans. Com. Tech.., Vol. COM-15, No.3, Oct 1967, pp390-397.

[123] Marquart R. G.: 'The Perfomance of Forced-Erasure Decoding'. IEEE Trans. Com. Tech., Vol. COM-15, No.3, Oct 1969, pp397-404.

[124] Wainberg S., Wolf J. K.: 'Burst Decoding of Binary Block Codes on Q-ary Output Channels', IEEE Trans. Inf. Th., Vol. IT-1, Sept 1972, pp684-687.

Appendix A. Transmission Line Theory

Consider a small line section, of length Δx:

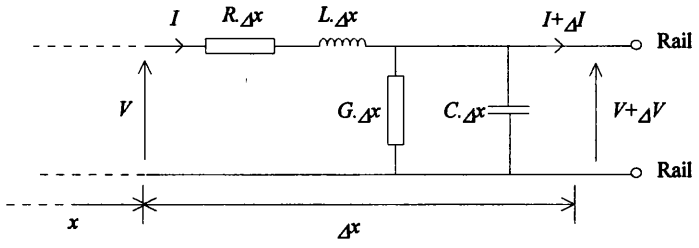


Figure A.1. Voltage and Current Drop across a small lumped section.

The change in voltage across the section is :

$$\Delta V = -I(R + j\omega L)\,\Delta x \qquad \text{(A.1)}$$

The change in current is:

$$\Delta I = -V(G + j\omega C)\,\Delta x \qquad \text{(A.2)}$$

Re-arranging:

$$\frac{\mathrm{dV}}{\mathrm{dx}} = -I(R + j\omega L) = IZ \qquad \text{(A.3)}$$

and

$$\frac{dI}{dx} = -V(G + j\omega L) = -VY \qquad \text{(A.4)}$$

differentiating with respect to x:

$$\frac{d^2V}{dx^2} = -\left[I\frac{dZ}{dx} + Z\frac{dI}{dx}\right] = -I\frac{dZ}{dx} + ZYV \qquad \text{(A.5)}$$

and

$$\frac{d^2I}{dx^2} = -\left[V\frac{dY}{dx} + Y\frac{dV}{dx}\right] = -V\frac{dY}{dx} + ZYI \qquad \text{(A.6)}$$

where $Z = R + j\omega L$ and $Y = G + j\omega C$. These equations lead to :

$$\frac{d^2V}{dx^2} = \gamma^2 V \qquad \text{(A.7)}$$

$$\frac{d^2I}{dx^2} = \gamma^2 I \qquad \text{(A.8)}$$

where $\gamma = \sqrt{(R + j\omega L)(G + j\omega C)}$. A solution to equation A.7 is $V = e^{\pm\gamma x}$, or more generally:

$$V = v_1 e^{\gamma x} + v_2 e^{-\gamma x} \qquad \text{(A.9)}$$

Including the time variation:

$$v(x,t) = v_1 e^{j\omega t} e^{\gamma x} + v_2 e^{j\omega t} e^{-\gamma x} \qquad \text{(A.10)}$$

We split γ into its real and imaginary terms, $\alpha + j\beta$:

$$v(x,t) = v_1 e^{j(\omega t+\beta x)} e^{\alpha x} + v_2 e^{j(\omega t-\beta x)} e^{-\alpha x} \qquad \text{(A.11)}$$

γ is known as the propagation constant. α is the attenuation constant (nepers/m) and describes the losses in the line. β is the phase constant (radians/m). Equation A.11 represents two travelling waves of voltage. The first term represents a wave travelling in the negative x direction, the second, in the positive x direction.

Appendix B. Cyclic Code Algebra

Appendix B.1 Introduction

Cyclic codes have a considerable inherent algebraic structure. These properties have been used to provide simple and fast encoding, parity checking and decoding circuits. A powerful subclass of cyclic codes are BCH codes. The construction of these codes requires an understanding of the algebra of finite fields and Galois theory.

Appendix B.2 Description of Algebra

Appendix B.2.1 Groups

A group is a set of elements over which one operation is defined, and has the following properties and elements:

Closure : When the operation is applied to any two group elements, the result is also a group element.

Associative : $(a + b) + c = a + (b + c)$ etc.

Identity Element : If the group operation is addition, the identity element is zero, and $0 + a = a + 0 = a$. For multiplication, the identity element is 1.

Inverse Element : Addition : $a + (-a) = 0$. Multiplication : $a.a^{-1} = 1$

Appendix B.2.2 Fields

A field is a set of elements in which we can perform addition, multiplication, subtraction and division without leaving the set. The conditions above apply. We can see that the field must at least contain the additive and multiplicative identity element (0 and 1). The number of elements in the field is called the *order* of the field.

The set of real numbers is a field. It has an infinite number of elements. For finite fields the number of elements must be a prime number, or the power of a prime number.

If p is a prime number, then the field with order p is called GF(p) (e.g. the binary field, GF(2)). This is sometimes called the *ground field.* The field containing p^m elements, denoted GF(p^m) is called the *extension field.* Extensions of the binary field are the most commonly used.

Appendix B.2.3 Characteristic of a Field, GF(q).

q is a prime number or the power of a prime number. Consider the sequence:

$\sum_{i=1}^{1} 1 = 1$, $\sum_{i=1}^{2} 1 = 1 + 1$,..... in a finite field. Since the field is closed the sequence must

repeat at some point, i.e. $\sum_{i=1}^{\lambda} 1 = 0$. λ is called the *characteristic* of the field. The characteristic of a field with a prime number of elements, GF(p), is p. Extension fields, GF(p^m), have characteristic p and these fields can be expressed as a field of polynomials over GF(p), with m elements. (This means that we have polynomials of degree (m-1) with coefficients taken from GF(p)) Note: GF(p) is usually taken as the binary field, GF(2).

Example : Consider the binary field GF(2). This is the *ground* field. Consider an extension field $GF(2^4) = GF(16)$, with 15 non zero components. Now this extension field can be expressed as binary polynomials of degree 3, i.e. $(a_3X^3 + a_2X^2 + a_1X + a_0)$ where a_n are binary integers. All possible combinations of the coefficients a_n are equivalent to all the elements of GF(16). So there are two representations of the same field, GF(16). It could be represented by 15 distinct non zero symbols, or it could be represented by polynomials of degree 3, over the binary field. The two representations are said to be *isomorphic.*

Appendix B.2.4 Primitive Elements

Let a be a non zero element of GF(q). Consider the sequence of powers of a: $a^1 = a, a^2 = a.a, a^3 = a.a.a$. As the field is closed under multiplication, this sequence will be cyclic and there will be a power of a where a^n=1. It can be shown than for fields with q elements, the value of n will always divide (q-1). n is called the *order* of the element. An element where n=(q-1) is called the *primitive element* (i.e., the order is equal to the number of elements in the field - 1). Therefore the powers of a primitive element generate all the non zero elements of GF(q).

Appendix B.2.5 Primitive Polynomials

As mentioned before, Galois fields can be expressed as polynomial (or n-tuples, or vectors) with coefficients taken from the ground field (usually GF(2)). There are many parallels between fields of integers and fields represented by polynomials. An *irreducible* polynomial is equivalent to a *prime* integer. A polynomial of degree m over GF(2) is said to be irreducible if it is not divisible by any other polynomial of degree less than m and greater than zero.

For example, p(X)=X^2+X is not irreducible since $X^2 + X = (X)(X + 1)$. But, p(X)= $X^2 + X + 1$ is irreducible. Alternatively, we may say that any polynomial over GF(2) that has either 0 or 1 as a root is not irreducible since it must have (X+1) or X as a factor (any polynomial with an even number of terms is divisible by X+1). These reducible polynomials over GF(2) *do* have roots, but they must come from an extension field GF(p^m).

Theorem B.1. Any irreducible polynomial over GF(2) of degree *m*, divides $X^{2^m-1}+1$. (p28 [55])

An irreducible polynomial *p(X)* of degree *m* is said to be primitive if the smallest integer *n* for which *p(X)* divides X^n+1 is $n=2^m-1$. So, p(*X*) will not divide X^n+1 for $0 \le n \le 2^m-2$.

So, not all irreducible polynomials are primitive. Tables of irreducible polynomials of different degrees are given in [55]

Appendix B.2.6 Minimal Polynomials.

Theorem B.2. The non zero elements of GF(2^m) are all roots of $X^{2^m-1}+1$. (p35 [55])

Since this polynomial is of degree 2^m-1 and we have just found 2^m-1 roots, the following equation must be true :

$$X^{2^m-1}+1=\prod_{i=1}^{2^m-1}(X-a_i) \qquad \text{(B.1)}$$

where α_i represents the non-zero elements of GF(2^m). The term on the left hand side of the equation can be considered to be a polynomial over GF(2). The term on the right hand side is a polynomial over GF(2^m) (Remember, arithmetic over GF(2) is modulo 2, where -1≡1). Now, all these non zero elements of GF(2^m) may be roots of polynomials over GF(2) that have degree less than 2^m. So, each element has a *minimal polynomial*, of the smallest degree, over GF(2), for which it is the root. For example, the field GF(2^4) contains the element 1, which is a root of the equation in theorem 1. It is also a root of the polynomial *X*+1, which is its minimal polynomial. Therefore, the element 1 cannot be the primitive element. So, the minimal polynomial of the primitive element is the equation $X^{2^m-1}+1$. Different elements of an extension field may have the same minimal polynomial. These elements are known as *conjugates* of each other.

Lets say that $\phi_i(X)$ is the minimal polynomial of α_i. We can relate these minimal polynomials to equation B.1 by:

$$X^{2^m-1}+1=\prod_{i=1}^{L} f_i(X) \qquad \text{(B.2)}$$

(Note that all the 2^m-1 non-zero elements each have a minimal polynomial, but some of them are repeated, so they are removed in the above equation, and $L<2^m-1$). Both sides of the equation are polynomials over the ground field GF(2).

Appendix B.2.7 Construction of a Galois Field GF(2^m)

Section 2.4, reference [55] contains an excellent explanation of how a Galois field can be constructed. We start with a primitive polynomial over GF(2), of degree m. From this we can construct the field GF(2^m), with each element represented as an m-tuple, or a polynomial of degree m-1, with coefficients over GF(2). Let $p(X)$ be the primitive polynomial and let α be the primitive element. For $0 \le i < 2^m - 1$, we divide the polynomial X^i by $p(X)$ and obtain the following:

$$X^i = q_i(X)p(X) + a_i(X) \qquad \text{(B.3)}$$

where $a_i(X)$ is the remainder, which has a degree less than m. If we replace X with α, the primitive element, we get (noting that $p(\alpha)=0$):

$$a_i(a) = a^i = a_0 + a_1 a + a_2 a^2 + \cdots\cdots + a_{m-1} a^{m-1} \qquad \text{(B.4)}$$

For all values of $0 \le i < 2^m - 1$, $a_i(X)$ is non-zero and unique. So we see that the 2^m-1 non-zero components of GF(2^m) are represented by 2^m-1 distinct polynomials of α, over GF(2). The zero element may be represented by the zero polynomial (p33 [55]). Addition of field elements can be done by modulo 2 addition of the polynomial representations. Multiplication can be done by representing each elements as the power of the primitive element and summing the powers, modulo (m-1).

Appendix B.2.8 Vector Spaces

The set of all possible n-tuples is called a *vector space* (p40 [55] and p219 [57]). They follow similar rules to that of fields i.e. they are commutative, associative, distributive, have a zero element and an identity element. The vector space is defined over a *field*, from which the coefficients are taken. The vector space, V may have a subset, or a *subspace, S*. The vectors in this subspace follow the same rules: the addition of any two vectors in S also results in a vector in S, and any field element multiplied by a vector in S, results in a vector in S.

A vector subspace can be formed from *linear combinations* of *linearly independent* vectors. (See p 42,43, [55] for definitions). Consider a set of k linearly independent 5-tuples, over the field GF(2). There are $2^5 = 32$ possible linear combinations, which form a subspace. The k linearly independent vectors are said to *span* the vector subspace. They form the *basis* of the subspace. The subspace is said to be k dimensional.

Dot Product or Inner Product

Let $\mathbf{u} = (u_0, u_1, \ldots\ldots u_{n-1})$ and $\mathbf{v} = (v_0, v_1, \ldots\ldots v_{n-1})$. The *inner product* is defined as

$\mathbf{u}\cdot\mathbf{v} = (u_0v_0, u_1v_1, ...u_{n-1}v_{n-1})$. If $\mathbf{u}\cdot\mathbf{v}=0$ then the two vectors are said to be *orthogonal* to each other. If all vectors in a subspace, S_d, are orthogonal to the subspace S, then S_d is said to be the *null space* of S.

If S is a k dimensional subspace of n-tuples, then its null space has the dimension n-k.

Appendix B.2.9 Matrix Description of Linear Block Codes.

Linear block codes are defined in terms of a *generator matrix* and a *parity check matrix.* The generator matrix has k rows of n-tuples. Also linear combinations of these rows form the code words. Therefore, there are 2^k possible code words, and each n-tuple carries k information bits, and n-k *redundant* or *parity check* bits. The code is described as a (n,k) code. The generator matrix for a (6,3) code is shown below:

$$\mathbf{G} = \begin{bmatrix} 1\,0\,0 \mid 0\,1\,1 \\ 0\,1\,0 \mid 1\,0\,1 \\ 0\,0\,1 \mid 1\,1\,0 \end{bmatrix} \qquad \text{(B.5)}$$

It can be seen that the first k columns of the generator matrix form an identity matrix. This is the generator of a *systematic linear block code.* The first k bits of the code word are the message bits, arranged in normal binary order. The 2^k linear combinations or rows can be formed by multiplying the generator matrix by a binary k-tuple. Lets call the code word **C**, and the k-tuple (information digits), **D**. An example of a code word could be:

$$\mathbf{C} = \mathbf{DG} = [111].\begin{bmatrix} 1\,0\,0 \mid 0\,1\,1 \\ 0\,1\,0 \mid 1\,0\,1 \\ 0\,0\,1 \mid 1\,1\,0 \end{bmatrix} = [1\,1\,1\,0\,0\,0] \qquad \text{(B.6)}$$

The parity check matrix has n-k rows and each row is a separate check on the redundant digits of the code word.

Hamming Weight

The Hamming weight of a particular code word, is the number of non-zero elements. The *weight distribution* of a code determines the probability of false detection in the presence of errors (p65 [55]).

Minimum Distance

This parameter determines the random error correcting and random error detecting capabilities of a code. It is defined as the minimum number of bits that would have to be in error to change one code word into another. Obviously a larger minimum distance would provide greater error detecting/correcting capabilities at the expense of greater redundancy. It can be shown that the minimum distance of a code is equal to the minimum weight of the code.

Appendix B.3 Cyclic Codes

Appendix B.3.1 Code Construction

Cyclic codes are a subclass of linear codes. They are described by using their polynomial representation.

> **Definition:** An (n,k) linear block code is cyclic if every cyclic shift of a code vector is also a code vector.

Reference [57] gives a thorough but heavy going development of the polynomial algebra involved with cyclic codes. A condensed version follows. See also p85 [55].

> **Theorem B.3.** In the algebra of polynomials modulo X^n+1 (or X^n-1), a subspace is a cyclic subspace only if it is an *ideal.*

This is more clearly explained by drawing a parallel with the algebra of polynomial mod $f(X)$, and the algebra of integers mod m, say mod 16. In this algebra, no integer can be above 15 and the integers [0,1,2.....15] form a ring (because the number of elements is the power of a prime). Say if we wanted to multiply 5 by 10, we would divide the result by m-1 and the remainder is the answer, i.e.: $5.10 = q\,.\,(m\text{-}1) + r = 3.15 + 5$. Therefore 50 mod 16 is 5.

With polynomials the degree is equivalent to the magnitude of the integer. So, in the algebra of polynomials modulo $f(X)$, where $f(X)$ has degree n, there do not exist any polynomials of degree greater than n-1. Multiplication in this algebra is performed by taking the remainder from division by $f(X)$. Lets take two polynomials is this algebra, $a(X)$ and $b(X)$. Suppose that (degree $a(X)$ + degree $b(X) > n$). Then the multiplication of these two polynomials in the algebra of polynomials mod $f(X)$ is done by taking the remainder $r(X)$ as follows:

$$a(X).b(X) = q(X).f(X) + r(X) \qquad \text{(B.7)}$$

An *ideal* is a subset of polynomials and consists of all multiples of some polynomial. Lets say that $g(X)$ is the lowest degree polynomial in the ideal (degree $< n$). Then the ideal is described by $a(X).g(X)$, where $a(X)$ could be any possible polynomial. Remember that the result is modulo $f(X)$ (degree n), and will therefore be a polynomial of degree (n-1) or less. A cyclic code is an ideal.

With this definition of a cyclic code (p85 [55]) lets look at the algebra of polynomials mod X^n-1. Consider a code polynomial which has degree n-1, $v(X)$:

$$v(X) = v_0 + v_1X + v_2X^2 \cdots\cdots + v_{n-1}X^{n-1} \qquad \text{(B.8)}$$

Now, by the definition of cyclic code, any cyclic shift of the coefficients of $v(X)$, is also a code vector. Let say that $v^{(i)}(X)$ represents a cyclic shift of the coefficients of $v(X)$ by i places. For the case where i=1, lets multiply $v(X)$ by X^i:

$$\begin{aligned} X.v(X) &= v_0X + v_1X^2 + v_2X^3 \cdots\cdots + v_{n-1}X^n \\ &= v_{n-1}(X^n + 1) + (v_{n-1} + v_0X + v_1X^2 + \cdots\cdots + v_{n-2}X^{n-1}) \\ &= v_{n-1}(X^n + 1) + v^{(1)}(X) \end{aligned} \qquad \text{(B.9)}$$

All cyclic shifts of $v(X)$ can be achieved by multiplying by any power of X. So, $v^{(i)}(X)$ is a multiple of $v(X)$ in the algebra of polynomials mod X^n-1, and hence is in the ideal.

Now lets see how cyclic codes are constructed. We start off with the minimum degree code vector $g(X)$, in a cyclic code (this can be shown to be unique). This has a degree n-k. (i.e. the top k-1 coefficients are zero). $g(X)$ must divide $X^n + 1$, but it is not necessarily irreducible. Now lets consider the k polynomials $g(X)$, $Xg(X)$, $X^2g(X)$, ...$X^{k-1}g(X)$. We have already seen that these vectors consist of cyclic shifts of $g(X)$. These vectors can be arranged as rows of the generator matrix, as shown below :

$$G = \begin{bmatrix} g_0, g_1, \cdots\cdots g_{n-k}, 0, & 0, & 0 \\ 0, & g_0, g_1 \cdots\cdots & g_{n-k}, 0, & 0 \\ 0, & 0, & g_0, g_1 \cdots\cdots\cdots & g_{n-k}, & 0 \\ 0, & 0, & 0, & g_0, g_1 \cdots\cdots\cdots & g_{n-k} \end{bmatrix} \qquad \text{(B.10)}$$

As with linear block codes, all linear combinations of these rows form the cyclic code. In terms of the definition of an ideal, the linear combinations can also be described as follows :

$g(X)$ is called the *generator polynomial* of the cyclic code. All linear combination of the rows of G are described by $a(X)g(X)$ where $a(X) = (a_0 + a_1X + \ldots\ldots + a_{k-1}X^{k-1})$. There are 2^k possible combinations of $a(X)$ corresponding to the 2^k code words. The code is not in systematic form.

Appendix B.3.2 Parity Check Matrix.

The parity check matrix uses the property that every code vector is divisible by $g(X)$. Recall that the generator matrix is a factor of $X^n + 1$. Say $X^n + 1 = g(X)h(X)$, where $g(X)$ has order n-k and $h(X)$ has order k. The matrix **H**, consisting of n-k rows can be constructed in the same way as the matrix **G**, except that the non zero coefficients are written in reverse order. From equation B.5, if a particular polynomial is equal to $X^n + 1$ or a multiple of $X^n + 1$, then its value is 0 (polynomials mod $X^n + 1$) Therefore $\mathbf{GH}^T$=0. Since all valid code words

are multiples of $g(X)$, then $a(X)g(X).h(X) = a(X)(X^n + 1) = 0$ (where $h(X)$ is the first row of **H**).

This is also true for all the rows of **H**:

$$a(X)g(X).\{X.h(X)\} = a(X).X.(X^n + 1) = 0$$

$$\vdots \qquad \vdots$$

$$\vdots \qquad \vdots$$

$$a(X)g(X).\{X^2.h(X)\} = a(X).X^2.(X^n + 1) = 0$$

In matrix form, this is written as :

$r_{(1,n)}$ = code vector = $\mathbf{D}_{(1,k)}\mathbf{G}_{(k,n)}$ (**D** is a k-tuple, equivalent to $a(X)$)

$$r_{(1,n)}\mathbf{H}^T{}_{(n,n-k)} = 0_{(1,n-k)}$$

This last result is called the *syndrome* calculation. If the received vector is a valid code word (i.e. its polynomial form is divisible by $g(X)$) then the syndrome is the zero vector, with n-k elements. Going back to the polynomial representation, the syndrome is calculated by dividing the received polynomial by $g(X)$ and taking the remainder. The result will be a polynomial of degree (n-k-1) or less, called $s(X)$:

$$r(X) = a(X)g(X) + s(X) \qquad \text{(B.11)}$$

Encoding and syndrome calculation can be achieved by using feedback shift register circuits with modulo two adders. For both, the feedback connections are determined by the coefficients of the generator polynomial, $g(X)$.

Appendix B.3.3 Alternative Description of Cyclic Codes, and BCH Codes.

We have seen from Theorem B.2 that the roots of the polynomial $X^{2^m-1}+1$ are all the non-zero values of GF(2^m). Now each element of GF(2^m) has a *minimal polynomial*, of which it is the root. The primitives elements, which have an order of m-1, have the primitive polynomial as the minimal polynomial, which has degree m and is used to construct the field GF(2^m).

Now, equation B.2 shows us that the polynomial $X^{2^m-1}+1$ can be represented as the least common multiple (LCM) of all the minimal polynomials of the elements of GF(2^m). i.e., if we group all the conjugate elements together, and multiply the separate minimal polynomials, the result is $X^{2^m-1}+1$.(See p38, [55]).

$$X^{2^m-1}+1 = LCM\{f_1(X), f_2(X)\cdots f_{2^m-1}(X)\} \qquad \text{(B.12)}$$
$$(Note: f_{2^m-1} \equiv 1)$$

Now lets say that *g(X)* is some factor of $X^{2^m-1}+1$. We could describe *g(X)* as being the LCM of a few of the minimal polynomials of GF(2^m), not all of them i.e.,

$$g(X)=\frac{(X^{2^m-1}+1)}{h(X)}=LCM\{f_1(X),f_2(X)\cdots f_r(X)\} \qquad \text{(B.13)}$$
$$where\ r<2^m-1$$

We can now be see a link between this, and cyclic codes. If the code length, *n*, is 2^m - 1, then the generator polynomial can be described in this way. These codes are BCH codes. For a value, *t* (will be defined later) we can define the generator polynomial as:

$$g(X)=LCM\{f_1(X),f_2(X),f_3(X)\cdots f_{2t}(X)\}$$
$$\text{removing repeated polynomials due to conjugate roots;} \qquad \text{(B.14)}$$
$$g(X)=LCM\{f_1(X),f_3(X),f_5(X)\cdots f_{2t-1}(X)\}$$

So, we can see that *g(X)* has $\alpha,\alpha^2,\alpha^3,\cdots,\alpha^{2t}$ (and their conjugates) as roots. Therefore $\alpha,\alpha^2,\alpha^3,\cdots,\alpha^{2t}$ are also roots of every code polynomial (since each code polynomial is a multiple of *g(X)*). This fact is used to construct the parity check polynomial.

The parameters of BCH codes are as follows:

Codes length :	$n = 2^m - 1$
Number of parity-check digits :	$n-k \le mt$
Minimum Distance	$d_{min} \ge 2t + 1$

This code is capable of correcting *t* or fewer random errors.

Parity Check Matrix : BCH Codes

To recap, the parity check matrix for a general cyclic code is described above using the fact that multiplying the received polynomial by any row of **H** (expressed as a polynomial) will produce a polynomial that is a multiple of X^n - 1. This is equal to zero, in the field of polynomials mod X^n - 1. Using the vector and matrix description: the code vectors are orthogonal to every row of **H**.

Now, for BCH codes, we construct the parity check matrix by using the fact that the elements $\alpha,\alpha^2,\alpha^3,\cdots,\alpha^{2t}$ of the field GF(2^m) are *roots* of each code polynomial which has coefficients from the field GF(2). Now, what does this really mean? :

> Consider the polynomial $5x^2+2x+$ over the field of real numbers. Now over this field there are no roots. But the field of complex numbers could be considered an extension field, and there are two roots belonging to this field : (-0.2 + j0.4) and its

conjugate (-0.2 - j0.4). So we have described a polynomial whose roots belong to an extension field.

So, let $v(X) = v_0 + v_1X + \cdots\cdots + v_{n-1}X^{n-1}$ be a code polynomial in a t error-correcting BCH code of length $n = 2^m - 1$. Since α^i is a root of $v(X)$ for $1 \le i \le 2t$, then:

$$v(a^i) = v_0 + v_1a^i + v_2a^{2i} + \cdots\cdots + v_{n-1}a^{(n-1)i} = 0 \qquad \text{(B.15)}$$

Note that this is a polynomial over GF(2^m), with the coefficients v_i taken from the field GF(2), which is a subgroup of GF(2^m). This can be written in matrix form as :

$$(v_0, v_1 \cdots, v_{n-1}) \cdot \begin{bmatrix} 1, (a), (a)^2 \cdots\cdots\cdots\cdots\cdots a^{n-1} \\ 1, (a^3), (a^3)^2, \cdots\cdots\cdots\cdots (a^3)^{n-1} \\ \vdots \\ 1, (a^{2t-1}), (a^{2t-1})^2, \cdots\cdots (a^{2t-1})^{n-1} \end{bmatrix}^T = v \cdot \mathbf{H}^T = [00\cdots\cdots 0]$$

Note that the rows that correspond to conjugates of the roots have been removed. Now the result (syndrome) shown on the right hand side is a t dimensional vector of zero elements from GF(2^m). Elements from this field are not much use to us and we want to be able to convert this to a binary representation. We have already seen that the elements of the extension field can be represented as m-tuples from the ground field. This means that the syndrome, in its binary representation, would be a zero vector of dimension mt. To obtain the binary representation of the parity check matrix, we simply replace each element with its binary m-tuple equivalent, arranged in column form. This results in changing a (t,n) matrix, over GF(2^m) into a $(mt,n) = (n\text{-}k,n)$ matrix over GF(2).

Appendix B.3.4 Error Detection/Correction

Linear block codes can correct additive errors by using the standard array decoding table. However, the complexity and memory requirements tend to grow exponentially with code size. Hence, the algebraic properties of cyclic codes have been used to reduce the complexity of the decoder (Section 4.5 [55]). As cyclic codes are a subset of linear block codes they can still be decoded with the standard array.

The Standard Array

The Standard array is constructed as a matrix. The first element of the first row is the zero vector and the 2^{k-1} valid code vectors complete the row. Now, the syndrome is $n\text{-}k$ bits long and this corresponds to 2^{n-k} *correctable error patterns* that are added to the received code vector. These error patterns are chosen as the *minimum weight* patterns, as these are most likely to occur. They are placed under the zero vector and are called *coset leaders*. The rest

of the rows are formed by adding (modulo 2) the coset leaders to the code vectors in the first row. No n-tuple should appear twice in the standard array.

Lets say that code vector **v** is transmitted, and the error pattern **e** is added to it. The received vector is:

$$\mathbf{r} = \mathbf{v} + \mathbf{e}$$

At the receiver, this is multiplied by the parity matrix, $\mathbf{H}^T$:

$$\mathbf{s} = (\mathbf{v} + \mathbf{e}).\ \mathbf{H}^T = \mathbf{v}\mathbf{H}^T + \mathbf{e}\mathbf{H}^T = \mathbf{e}\mathbf{H}^T \qquad \text{(B.16)}$$

(In polynomial terms, the syndrome is the remainder from dividing the error pattern by the generator polynomial, $g(X)$.)

Now, the standard array has the correctable error patterns as coset leaders. If we now calculate the syndrome that would occur if these error patterns were decoded we can construct a decoding table that associates each syndrome with each error pattern. Finally, this error pattern is added to the received code vector to produce a valid code vector.

If an error pattern that is a code vector occurs, the syndrome will be zero and a decoding error will occur. This is because the sum of two code vectors is also a code vector. Also if an error pattern occurs that is not a coset leader, a decoding error will occur.

A particular syndrome will correspond to many error patterns. It is assumed, in the decoding table, that the error pattern with the minimum weight has occurred.

In summary: an (n,k) linear code is capable of detecting $2^n - 2^k$ error patterns, and is capable of correcting 2^{n-k} error patterns.

Burst Error Detection/Correction

Suppose that the error pattern is confined to n-k consecutive positions. For example, a burst of length 4 could be **e** = (00011010). An *end around Burst* of length 4 could be **e** = (110000010).

An (n,k) cyclic code is capable of detecting all busts of length n-k. (p102 [55]). Also, a large number of bursts of length n-k+1 are capable of being detected. The *Reiger bound* (p110 [57]) states that in order to correct all bursts of length b, at least $2b$ parity bits are needed. i.e. $b \leq (n\text{-}k)/2$.

Appendix C. DFT Update Routine.

The DFT update routine described here provides a method of 'sliding' the DFT along by one sample and makes use of the fact that each successive N point DFT frame has (N-2) points in common with the previous frame. Lets consider 2 successive N point DFT frames, as shown below:

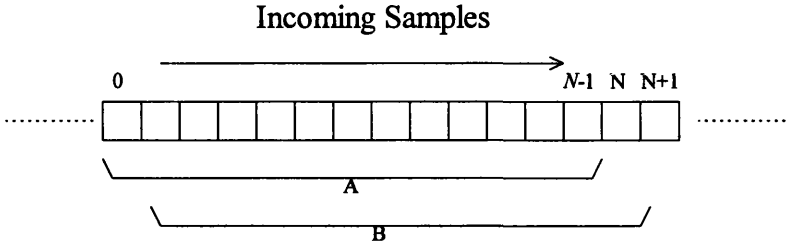


Figure C.1. Successive DFT Frames.

We will assume that DFT results exist for frame A according to

$$X_A(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N} \qquad \text{(C.1)}$$

where $X_A(k)$ represents the value of the kth bin. If no previous DFT results exist, the full DFT/FFT can be calculated, or the DFT update can be performed N times using initial dummy values. Now we want to update $X_A(k)$, corresponding to point B of Figure C.1.

$$X_B(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-j2\pi k(n-1)/N} - x(0)e^{j2\pi k/N} + x(N)e^{j2\pi k/N} \qquad \text{(C.2)}$$

From the first term on the right, we can factor out $X_A(k)$.

$$X_B(k) = \frac{1}{N}\left[X_A(k) - x(0) + x(N)\right]e^{j2\pi k/N} \qquad \text{(C.3)}$$

Generalising to the nth sample, where n is the first sample on the nth DFT frame :

$$X_{n+1}(k) = \frac{1}{N}\left[X_n(k) - x(n) + x(n+N)\right]e^{j2\pi k/N} \qquad \text{(C.4)}$$

This equation requires three complex multiplications (and three complex additions) for each DFT bin. This can be compared to the DFT which requires $(N \log_2 N)$ complex multiplications and $\frac{N}{2}\log_2 N$ complex additions. If the update routine is performed for all the DFT bins, the DFT update routine is less computationally intensive for $N > 64$.

Appendix D. Multi-Tone Signal Demodulation with Noisy Input.

Appendix D.1 FFT Response to Noisy Input

Each FFT bin corresponds to a 2 dimensional signal space. The *basis functions* for the FFT are:

$$\phi_R(t) = \sqrt{\frac{2}{T}}\cos(2\pi f_c t)$$
$$\phi_I(t) = \sqrt{\frac{2}{T}}\sin(2\pi f_c t) \quad \text{(D.1)}$$

where T is the FFT time. These are described in [93] as being orthogonal functions of unit energy which are individually correlated with the received signal to produce a vector of coefficients which describe a point in the signal space. The coefficients corresponding to each axis of the two dimensional signal space are calculated by:

$$s_R = \int_0^T s(t)\phi_R(t)dt$$
$$s_I = \int_0^T s(t)\phi_I(t)dt \quad \text{(D.2)}$$

Each output can be described by a two dimensional vector : $S = \begin{bmatrix} s_R \\ s_I \end{bmatrix}$ where s_R represents the real signal value, and s_I represents the imaginary signal value. Note that the FFT actually correlates the incoming signal with $e^{-j(2\pi f_c t)} = \cos(2\pi f_c t) - j\sin(2\pi f_c t)$ and so the actual FFT real and imaginary value will differ from the values in the vector S, by a factor of $\sqrt{2/T}$ i.e. : $s_{R/I}(FFT) = s_{R/I}\sqrt{\frac{T}{2}}$.

Lets say that the signal at the receiver input, with no noise is *s(t)*. The signal with AWGN is :

$$X(t) = s(t) + W(t) \quad \text{(D.3)}$$

where $W(t)$ is a Gaussian white noise process of zero mean and psd $N_o/2$. We will assume that the signal $s(t)$ only contains spectral components that correspond to FFT bins, and so can be completely separated at the receiver. We will only consider one spectral component. The vector X is calculated: (p69 [93])

$$X_R = \int_0^T X(t)\phi_R(t)dt \qquad X_I = \int_0^T X(t)\phi_I(t)dt \quad \text{and} \quad X = \begin{bmatrix} X_R \\ X_I \end{bmatrix} \tag{D.4}$$
$$= s_R + W_R \qquad = s_I + W_I$$

where X_R and X_I are random variable (hence capitals) with mean values of s_R and s_I respectively and variance equal to the psd of the noise ($N_o/2$). Also, since X_R and X_I form an orthogonal set, they are statistically independent. The probability density function of a Gaussian variable of mean μ and variance σ^2 is given by the standard equation :

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{(x-\mu)^2}{2\sigma^2}\right] \tag{D.5}$$

hence, the conditional probability density function for the random variables X_R and X_I are:

$$f_{X_R}(x_R|s_R) = \frac{1}{\sqrt{\pi . N_o}} \exp\left[-\frac{1}{N_o}(x_R - s_R)^2\right] \tag{D.6}$$

$$f_{X_I}(x_I|s_I) = \frac{1}{\sqrt{\pi . N_o}} \exp\left[-\frac{1}{N_o}(x_I - s_I)^2\right] \tag{D.7}$$

The total conditional probability of the vector X, may be expressed as the product of the two independent functions :

$$\begin{aligned} f_X(X|S) &= f_{X_R}(x_R|s_R).f_{X_I}(x_I|s_I) \\ &= \frac{1}{\pi N_o} \exp\left[-\frac{1}{N_o}\left[(x_R - s_R)^2 + (x_I - s_I)^2\right]\right] \end{aligned} \tag{D.8}$$

Appendix D.2 Vector Addition with noisy input

Calculating the resultant vector requires the vector addition of the two sidebands. Firstly lets ignore the effect of error in the carrier phase. Referring back to the signal constellation diagram we assume that all message points are equally as likely, so the error probability on one specific message point will be the same as the rest.

Lets consider a resultant vector which, in the absence of noise, lies along the real axis (i.e. in phase with the carrier). At t=0, the USB and LSB will be in phase with the carrier and the resultant vector with has a magnitude of $2\sqrt{E}$,where E is the energy per sinusoid. This is shown in figure D.1. At t $= \pi / 4\omega_m$, the USB and LSB will be at 90° to each other and the resultant vector will have a magnitude of $\sqrt{2E}$. This is shown in figure D.2

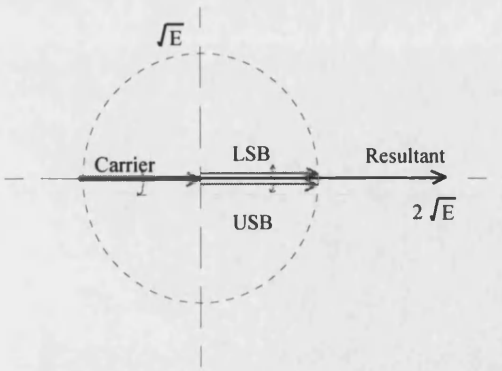


Figure D.1

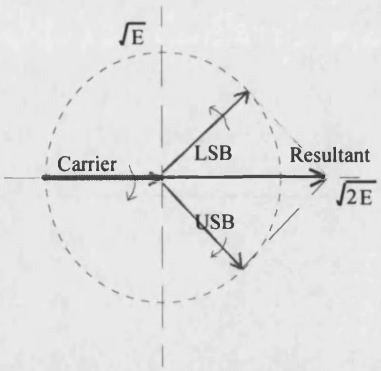


Figure D.2

Between $t = \pi / 4\omega_m$ and $t = 3\pi / 4\omega_m$ the resultant vector magnitude passes through zero, from $\sqrt{2E}$ to $-\sqrt{2E}$. However, if, in this region, the difference vector is taken then the magnitude of the resultant vector will range from $2\sqrt{E}$ to $\sqrt{2E}$. This is shown in figure D.3.

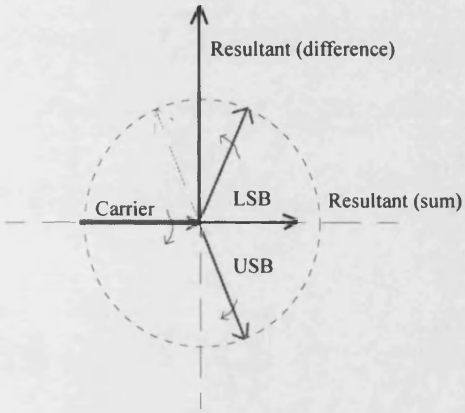


Figure D.3

The difference vector will always be at right angles to the sum vector. The technique is used in the existing Westinghouse FS systems to ensure that an ASK interference signal is not mistaken for a valid FSK signal.

The FFT frame time can occur at any time, so we can say that the mean value of the magnitude of the resultant vector (with no noise) will be :

$$\frac{2\sqrt{E} + \sqrt{2E}}{2} = \sqrt{E}\left(1 + 1/\sqrt{2}\right) \qquad \text{(D.9)}$$

The resultant vector is formed by adding the real and imaginary components of the USB and LSB :

$$X(\text{resultant})_R = X(\text{USB})_R + X(\text{LSB})_R \qquad \text{(D.10)}$$

$$X(\text{resultant})_I = X(\text{USB})_I + X(\text{LSB})_I \qquad \text{(D.11)}$$

(where capital X denotes a random variable, whose sample value, x_R/x_I consists of the signal plus noise). Because all the terms on the right hand side are independent random variables with variance $N_o/2$, the resultant real and imaginary components will have variance N_o. Lets say that the two sidebands are sampled so that the signal vectors are :

$$S(USB) = \begin{bmatrix} S(USB)_R \\ S(USB)_L \end{bmatrix}, \quad S(LSB) = \begin{bmatrix} S(LSB)_R \\ S(LSB)_L \end{bmatrix} \qquad \text{(D.12)}$$

The conditional probability density functions for the real and imaginary components of the resultant vector $R = \begin{bmatrix} R_R \\ R_I \end{bmatrix}$ now become :

$$f_{R_R}(r_R|s(USB)_R, s(LSB)_R) = \frac{1}{\sqrt{2\pi.N_o}} \exp\left[-\frac{1}{2N_o}\left[r_R - \left(s(USB)_R + s(LSB)_R\right)\right]^2\right] \quad \text{(D.13)}$$

$$f_{R_I}(r_I|s(USB)_I, s(LSB)_I) = \frac{1}{\sqrt{2\pi.N_o}} \exp\left[-\frac{1}{2N_o}\left[r_I - \left(s(USB)_I + s(LSB)_I\right)\right]^2\right] \quad \text{(D.14)}$$

The total conditional probability density function for the vector *R*, is :

$$f_R(R|S(USB), S(LSB)) = f_{R_R}\left(r_R|s(USB)_R, s(LSB)_R\right).f_{R_I}\left(r_I|s(USB)_I, s(LSB)_I\right)$$

$$= \frac{1}{2\pi N_o} \exp\left[\frac{-1}{2N_o}\left[\left(r_R - \left(s(USB)_R + s(LSB)_R\right)\right) + \left(r_I - \left(s(USB)_I + s(LSB)_I\right)\right)\right]^2\right]$$

(D.15)

Lets consider a specific message point, where the resultant vector is in phase with the carrier. (it could also be in antiphase for the same message point). It will have mean values of

$$\mathrm{E}[R_R] = \sqrt{E}\,(1+1/\sqrt{2}\,)$$

$$\mathrm{E}(R_I) = 0 \quad \text{(D.16)}$$

(where E on the left denotes statistical average and *E* on the right is the signal energy for the FFT frame.) and variances of

$$\mathrm{var}(R_R) = N_o\,, \quad \mathrm{var}(R_I) = N_o \quad \text{(D.17)}$$

The probability of decoding error is the probability that the resultant vector falls outside the decision boundary for that message point. Lets say that the number of message points is *M* (remember that each message point is duplicated every 180°). For the message point described by D.16 above, the decision region is bounded by lines drawn at angles of $+\pi/2M$ and $-\pi/2M$ to the real axis, as shown below.

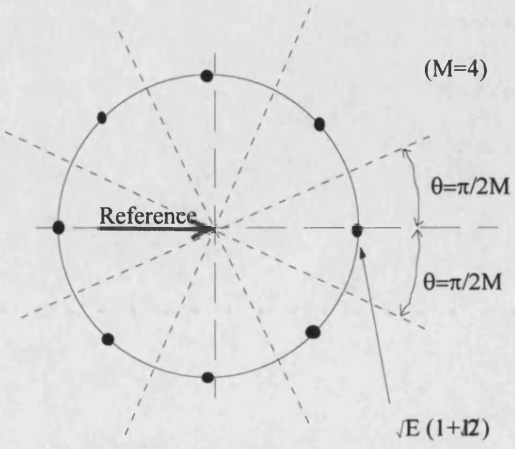


Figure D.4

The probability of decoding error will be :

$$P_e = 1 - \int_{-\pi/2M}^{+\pi/2M} f_{\theta R}(\theta_R) d\theta_R \qquad \text{(D.18)}$$

where $f_\theta(\theta)$ is the p.d.f. of the random variable θ, which represents the angle of the resultant vector to the positive real axis. This function has been shown to be (p316, [93]):

$$f_{\theta_R}(\theta_R) = \frac{1}{2\pi}\exp\left(\frac{-E\left(1+1/\sqrt{2}\right)^2}{2N_o}\right) + \sqrt{\frac{E\left(1+1/\sqrt{2}\right)^2}{2\pi N_o}}\cos(\theta_R)\exp\left(\frac{-E\left(1+1/\sqrt{2}\right)^2}{2N_0}\sin^2\theta_R\right).$$
$$\left[1-\frac{1}{2}erfc\left(\frac{E\left(1+1/\sqrt{2}\right)^2}{2N_o}\cos\theta_R\right)\right] \qquad \text{(D.19)}$$

Shown below is a plot of the phase p.d.f of the resultant vector for values of E/N_o of 10, 5 and 2.

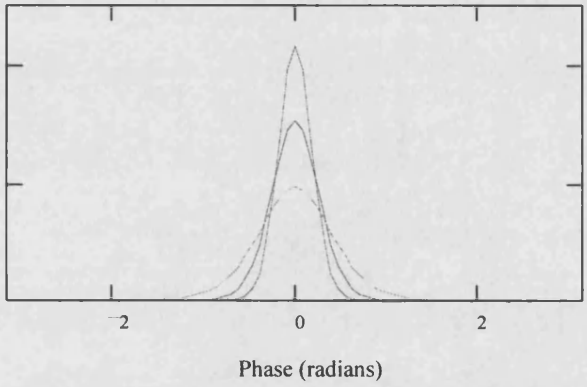


Figure D.5. P.D.F. of Phase for Resultant Vector.

The integral of this function does not reduce to a simple form and must be evaluated using numerical integration. Tables of P_e for varying E/N_o are found in (pp232-233, [116]). Using simplifications (for large values of E/N_o) the probability of error can be shown to be: (p317, [93])

$$P_e = erfc\left(\sqrt{\frac{E\left(1+1/\sqrt{2}\right)^2}{2N_o}}\sin\left(\frac{\pi}{2M}\right)\right) \qquad \text{(D.20)}$$

where M is the number of unique message points.

Appendix D.3 Error in Phase of Carrier

The previous section has assumed a perfect carrier phase reference. The phase p.d.f. of the carrier has the form :

$$f_{\theta_C}(\theta_C) = \frac{1}{2\pi}\exp\left(-\frac{E}{N_o}\right) + \sqrt{\frac{E}{\pi N_o}}\cos(\theta_C)\exp\left(\frac{-E}{N_0}\sin^2\theta_C\right).\left[1-\frac{1}{2}erfc\left(\frac{E}{N_o}\cos\theta_C\right)\right] \quad \text{(D.21)}$$

Shown below is a plot of the phase p.d.f of the carrier vector for values of E/N_o of 10, 5 and 2.

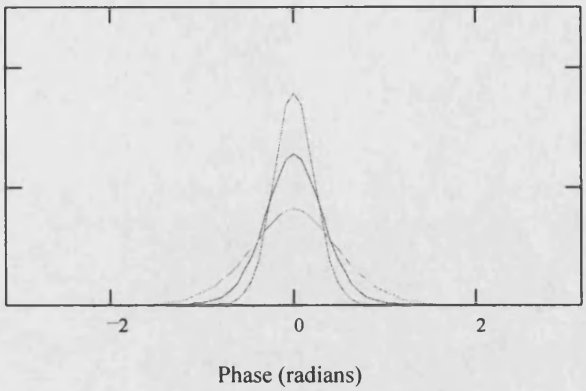


Figure D.6. p.d.f. of Phase for a Single sinusoid

Note that the pdf of the resultant vector is sharper than the pdf of the carrier. This is because the vector addition has the effect of averaging out some of the noise.

The message phase is the difference between the resultant phase and the carrier phase :

$$\theta_{(output)} = \theta_R - \theta_C \quad \text{(D.22)}$$

The pdf of the output ph$P_e = 1 - \int_{}^{+\pi/2M} f_{\theta(output)}(\theta_{(output)})d\theta$he resultant phase p.d.f and the carrier phase p.d.f. [117]. The total error probability will be :

$$P_e = 1 - \int_{-\pi/2M}^{+\pi/2M} f_{\theta(output)}(\theta_{(output)})d\theta_{(output)} \quad \text{(D.23)}$$

This will not reduce to a simple form and so is best evaluated by simulation, to find the probability of error, for various values of E/N_o and various values of M.

Appendix E. Error Probability Simulation

The simulation program (written in C) performs the following steps (the simulation will assume that the resultant phase in the absence of noise will be in phase with the carrier, and that the carrier phase in the absence of noise is arbitrarily chosen to be 0):

1. Set a certain value of M (i.e. M=2, M=4 etc.), and a ratio of E/N_o
2. Take a sample value of the carrier phase using

   $\theta_c = \tan^{-1}\left(\frac{W_{Q(Carrier)}}{\sqrt{E} + W_{I(Carrier)}}\right)$ where $W_{Q(\text{Carrier})}$ and $W_{I(\text{Carrier})}$ are independent Gaussian noise samples, with a variance of $N_o/2$.
3. Take a sample value of the phase of the resultant vector using

   $\theta_R = \tan^{-1}\left(\frac{W_{Q(Resultant)}}{\sqrt{E}(1+1/\sqrt{2}) + W_{I(Resultant)}}\right)$ where $W_{Q(\text{Resultant})}$ and $W_{I(\text{Resultant})}$ are independent quadrature Gaussian noise samples, with a variance of N_o.
4. Evaluate $|\theta_R - \theta_C|$
5. If $|\theta_R - \theta_C| > \pi/2M$, then error in decoding. If $|\theta_R - \theta_C| < \pi/2M$ then correct decoding.
6. Repeat, to step (2)
7. Vary E/N_o and repeat.

The results are plotted against $10\log_{10}(E_b/N_o)$, where E_b is the energy per bit, for one subchannel. One subchannel consists of three sinusoids (compared to one sinusoid for conventional PSK).

So, $E_b = \frac{E_s}{\log_2(M)}$ and $E_s = 3E$, where E is the energy for one sinusoid, as used above.

The C function for Gaussian noise has a variance of 1. The corresponds to $N_o = 2$. The value of E is varied to produce the different ratios of E_b/N_o.

Appendix F. Encoding/Decoding of Shortened Reed-Solomon (11,7) Code

This section will describe a systematic encoding and decoding algorithm for the shortened RS (11,7) code. In general, the decoding of an RS code consists of five basic steps:

1. Compute the syndrome values from the received vector.
2. Determine the error locator polynomial $\sigma(x)$ from the syndrome values.
3. Solve the roots of $\sigma(x)$, which are the error locators.
4. Calculate the error values.
5. Correct the located errors.

The most difficult part of the procedure is step 2, since this involves the solutions of non-linear equations. The method that follows uses the Peterson's direct solution method [57], which is applicable to the correction of up to 3 symbol errors. Here, the coefficients of $\sigma(x)$ are solved from a set of linear equations involving the syndrome values. The algorithms also apply to the shortened RS(10,6) code and are suitable for simple microprocessor implementation. The RS (11,7) code is a shortened version of the (31,27) code. The code is capable of correcting $t=(n-k)/2$ symbols. The error correcting properties of the shortened code are exactly the same as the original code.

Appendix F.1 Encoding

The code symbols are defined in GF(2^m) where m=5. If α is the primitive element[1] of GF(2^5) then the generator polynomial is defined as the product of the minimal polynomials of the first $2t$ elements of GF(2^5):

$$\begin{aligned} g(X) &= (X+\alpha)(X+\alpha^2)....(X+\alpha^{2t}) \\ &= g_0 + g_1X + g_2X^2 + ... + g_{2t-1}X^{2t} \end{aligned} \tag{F.1}$$

For the (11,7) (and the (31,27)) code:

$$\begin{aligned} g(X) &= (X+\alpha)(X+\alpha^2)(X+\alpha^3)(X+\alpha^4) \\ &= X^4 + \alpha^{24}X^3 + \alpha^{12}X^2 + \alpha^{29}X + \alpha^7 \end{aligned} \tag{F.2}$$

The data polynomial is defined as

$$a(X) = a_0 + a_1X + ... + a_{k-1}X^{k-1} \tag{F.3}$$

where k=27 for the original (31,27) code. For the shortened code, we ignore the last 20 symbols, effectively making k=7. These symbols are assumed to be zero at the receiver. The term $X^{2t}a(X)$ makes up the k higher order symbols of the (11,7) code word. The term $p(X)$

[1] i.e. the powers of α generate all the non-zero elements of GF(2^5)

will be used to represent the (n-k) lower order symbols (parity check symbols) of the code word. $p(X)$ is the result of dividing $X^{2t}a(X)$ by $g(X)$ and taking the remainder. In other words:

$$p(X) = X^{2t} \bmod g(X) = p_0 + p_1 X + p_2 X^2 \cdots\cdots + p_{2t-1} X^{2t-1} \qquad \text{(F.4)}$$

The final code word is given by:

$$c(X) = p(X) + X^{2t} a(X) \qquad \text{(F.5)}$$

Each coefficient from GF(2^5) can be represented by a 5 bit byte (5-tuple). A list of the binary representation of the galois field elements is given in p562 of [55] (see also p33).

Appendix F.2 Decoding

The syndrome will be $2t$ symbols long. All code polynomials have the elements α, α^2..... α^{2t} as their roots, since all code polynomials are multiples of the generator polynomial $g(X)$.

The $2t$ syndrome components are the results of evaluating the received polynomial at $X=\alpha$, α^2..... α^{2t}:

$$\begin{aligned}
&S_1 = c(\alpha) = c_0 + c_1\alpha + c_2\alpha^2 + \ldots\ldots c_{n-1}\alpha^{n-1} \\
&S_2 = c(\alpha^2) = c_0 + c_1\alpha^2 + c_2\alpha^4 + \ldots\ldots c_{n-1}\alpha^{2(n-1)} \\
&\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \qquad \text{(F.6)} \\
&S_{2t} = c(\alpha^{2t}) = c_0 + c_1\alpha^{2t} + c_2\alpha^{4t} + \ldots\ldots c_{n-1}\alpha^{2t(n-1)}
\end{aligned}$$

Lets say that 2 symbol errors have occurred. The received polynomial can be represented as:

$$r(X) = c(X) + e(x) \qquad \text{(F.7)}$$

where $e(X)$ is the error polynomial:

$$e(x) = E_{i_1} X^{i_1} + E_{i_2} X^{i_2} \qquad \text{(F.8)}$$

E_{i_1} and E_{i_2} represent the error values (an element of GF(2^5)) and i_1, i_2 represent the error locations. Since the syndrome of a valid code word is zero, the syndrome values are given by $e(X)$ evaluated at α, α^2..... α^{2t}:

$$S_1 = E_{i_1}\alpha^{i_1} + E_{i_2}\alpha^{i_2} \qquad \text{(F.9a)}$$

$$S_2 = E_{i_1}\alpha^{2i_1} + E_{i_2}\alpha^{2i_2} \qquad \text{(F.9b)}$$

$$S_3 = E_{i_1}\alpha^{3i_1} + E_{i_2}\alpha^{3i_2} \qquad \text{(F.9c)}$$

$$S_4 = E_{i_1}\alpha^{4i_1} + E_{i_2}\alpha^{4i_2} \qquad \text{(F.9d)}$$

If all the syndrome values are zero, then no errors have occurred and the code word is accepted. If any of the values are non zero, then the next step is to determine whether 1 or 2 errors have occurred. This information can be found directly from equation F.9, as follows.

Appendix F.2.1 Single Error Correction.

Assuming a non-zero syndrome, if the following equation is satisfied, then a single byte error has occurred:

$$S_2^{\,2} = S_1 S_3 \tag{F.10}$$

The error location α^{i_1} can be evaluated from:

$$\alpha^{i_1} = \frac{S_2}{S_1} \tag{F.11}$$

The error value, E_{i_1}, can be evaluated from:

$$E_{i_1} = \frac{S_1}{\alpha^{i_1}} \tag{F.12}$$

Appendix F.2.2 Double Error Correction.

Assuming a non-zero syndrome, if the following equations are satisfied, then a double error has occurred:

$$\begin{aligned} S_2^{\,2} &\neq S_1 S_3 \\ S_1 S_4 &\neq S_2 S_3 \end{aligned} \tag{F.13}$$

We now define an error location polynomial, $\sigma(X)$, whose roots are the reciprocals of α^{i_1} and α^{i_2}:

$$\begin{aligned} \sigma(X) &= (1+\alpha^{i_1}X)(1+\alpha^{i_2}) \\ &= 1+\sigma_1 X+\sigma_2 X^2 \end{aligned} \tag{F.14}$$

where

$$\sigma_1 = \alpha^{i_1} + \alpha^{i_2} \tag{F.15a}$$

$$\sigma_2 = \alpha^{(i_1+i_2)}. \tag{F.15b}$$

Note that once we know $\alpha^{i_2}, \alpha^{i_2}$ $0 \le i_1, i_2 \le n-1$, then we also know the location of the error, since there are $(2m-1)$ individual elements in the field GF(2^m), each represented by a power of α. ([55] p562). Once the power is known, it corresponds to a symbol position in the code word. Substituting equation F.9 into equation F.15 gives us:

$$\sigma_1 = \frac{S_1 S_4 + S_2 S_3}{S_3^{\,2} + S_2 S_4} \tag{F.16}$$

$$\sigma_2 = \frac{S_2^{\,2} + S_1 S_3}{S_3^{\,2} + S_2 S_4} \tag{F.17}$$

Once these coefficients are calculated the roots of equation F.14 need to be found. This can be accomplished by a look up table approach, or by Chien search [118]. The Chien search

method finds the first root, R_1, by simply evaluating $\sigma(X)$ for $X=\alpha^i$, $i=0, 1, 2.....31$, until a zero value is found. The second root is found by:

$$R_2 = \sigma_1 + R_1 \tag{F.18}$$

The inverse of the roots[2] are the actual error locations α^{i_1} and α^{i_2}. The error values E_{i_1} and E_{i_2}, can be found by re-arranging equation F.9 :

$$E_{i_1} = \frac{S_2 + S_1\alpha^{i_2}}{\left(\alpha^{i_1} + \alpha^{i_2}\right)\alpha^{i_1}} \tag{F.19}$$

$$E_{i_2} = \frac{S_1 + E_{i_1}\alpha^{i_1}}{\alpha^{i_2}} \tag{F.20}$$

So, we have the error locations, α^{i_1} and α^{i_2}, where i_1, i_2 tell us the positions of the errors, and the error values, E_{i_1} and E_{i_2}. The errors are corrected by adding the error values (in their binary representation) to the affected symbols.

Appendix F.3 Arithmetic in GF(2^5)

Reed-Solomon codes are constructed and decoded using arithmetic in the field GF(2^m). These fields are constructed from the *primitive polynomial, p(X)*, of which the *primitive element, α,* is the root. From this polynomial, all the elements of GF(2^m) can be represented by the polynomial $a_0 + a_1\alpha + a_2\alpha^2 + \cdots a_{m-1}\alpha^{m-1}$. The coefficients $(a_0, a_1, a_2, \ldots a_{m-1})$ form a binary m-tuple corresponding to each element, α^0 to α^{2^m-2}. The zero element is represented by the zero m-tuple.

Addition of two elements of GF(2^m) is performed by adding each element of the binary m-tuple, modulo 2. Multiplication of two elements, α^n and α^m is performed by adding the exponents, modulo (2^m-1). For example, in the field GF(32), $\alpha^{16}.\alpha^{20}=\alpha^{36}=\alpha^5$.

---

[2] The roots will be exressed as a power of α. In the field GF(2^5), the inverse of α^i is $\alpha^{(31-i)}$, since $\alpha^i.\alpha^{(31-i)} = \alpha^{31} = \alpha^0 = 1$

Appendix G. Soft Decision Decoding Techniques

An optimum method of detection for a block coded data transmission system is the matched filtering of the sequence of signal elements corresponding to the block length [98]. For practical block code lengths, the ideal detector is too complex to realise because of the large amount of analogue signals required to be correlated with the input signal. So the code is split into smaller 'symbols' which are processed by the analogue correlator. A hard decision is made on these symbols which results in a loss of channel capacity since information about the reliability of each symbol decision is not used in the decoding process. This section describes some practical soft decision methods, where the output of the demodulator is quantised to Q>2 levels, allowing the confidence of each bit to be assessed by the decoder.

Appendix G.1 Defining Symbol Reliability

Appendix G.1.1 Binary Data

Null zone or *erasure detection* are methods of transferring information from the demodulator to the decoder. The channel can then be modelled as a Binary Erasure Channel (BEC) as shown below

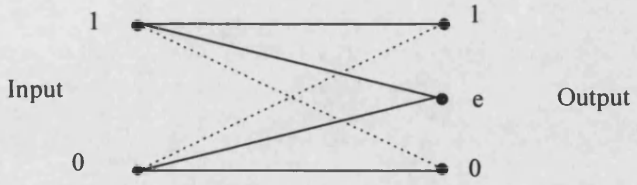


Figure G.1. Binary Erasure Channel.

The decoder has some knowledge of where the errors are likely to be. To obtain the best performance, the null zone threshold needs to be adjusted as a function of the signal to noise ratio (p23,[111]) [119]. This however assumes that the signal to noise ratio can be measured at the receiver and that the channel statistics are known.

This null zone detection can be extended to double null zone detection with an improvement in performance [113]. The general case is a binary input, Q-ary output channel, where the demodulator quantises the received signal space into Q regions.

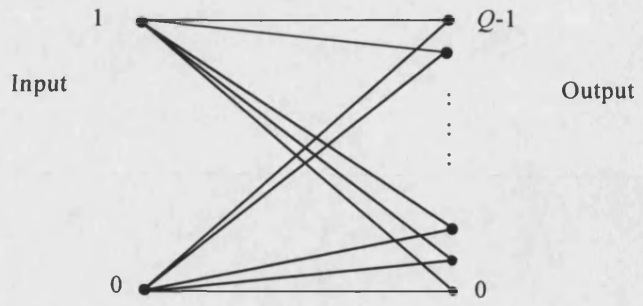


Figure G.2. Binary Input, Q-ary Output Channel.

The improvement of performance diminishes as the number of regions is increased [114]. This technique can potentially double the error correcting capabilities of a block code, so a code with a Hamming distance d can correct d-1 errors. A simple example is a code with a single parity bit (single error detection): If an error is detected, the bit most likely to be in error (on a Q-ary output channel) is corrected.

The Q-ary output channel can be interpreted as a hard decision, y, with a reliability measure α. The transmitted code word can be represented by the vector $\boldsymbol{x} = (x_1, x_2, \ldots x_n)$. A received code word of n symbols can be represented by a hard decision vector, $\boldsymbol{y} = (y_1, y_2, \ldots y_n)$ and a reliability vector, $\alpha = (\alpha_1, \alpha_2, \ldots \alpha_n)$. The effectiveness of the soft decision decoding will depend, to a greater or lesser extent, on the choice of reliability measure, α. For the AWGN (memoryless) channel, maximum likelihood decoding is permitted if the reliability of each of the Q outputs is chosen according to [120, p90] :

$$\alpha_i = \boldsymbol{ln}\frac{P(L_i|x_i = y_i)}{P(L_i|x_i = y_i')} \qquad \text{(G.1)}$$

where L_i is one of Q outputs (w_j, $0 \le j \le Q\text{-}1$) for the ith symbol, x_i is the transmitted bit, y_i is the hard decision for that bit and y_i' is the compliment of that hard decision (see section G.1.2 for non binary data). The (Gaussian) probability function is shown in figure G.3 below for a binary input to a 4 output channel:

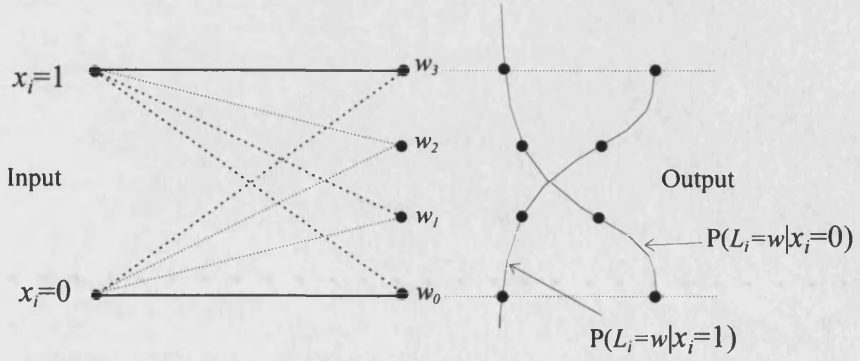


Figure G.3. Binary Input, 4-ary Output Channel, with Gaussian Transition Probabilities.

A practical implementation of this was proposed by Forney [112] which is equivalent to the maximum likelihood reliability assignment, except at the extremes, where α is hard limited to 0 and 1, representing *no confidence (or erasure)* and *full confidence* in the received symbol, respectively.

$$\alpha_i = \begin{cases} 1\ , & A_i > 2s \\ A_i\ , & 0 \le A_i \le 2s \\ 0\ , & A_i < 0 \end{cases} \qquad A_i = \ln \frac{P(L_i | x_i = y_i)}{P(L_i | x_i = y_i')} \tag{G.2}$$

where s is a constant determined by the decoding method [112](p90,[120]).

This method can have two disadvantages. It assumes the statistics of the channel are known and if the noise is not AWGN, then this method is no longer optimum. Also, the assignment of α can be complex, and simpler schemes can be used, with a small decrease in performance [114].

Appendix G.1.2 Non Binary Data

In the soft decision decoding of a multilevel code, such as an RS code, the generation of the reliability information of each symbol is not as straight forward. Lets assume that there are 2^n symbols, and each symbol is represented by n bits. The result of hard decisions on the received symbol is denoted by $H_i = (h_{i1}, h_{i2}, \ldots h_{in})$. The actual received symbol can be denoted by $R_i = (r_{i1}, r_{i2}, \ldots r_{in})$. Note that the r values could represent, for instance, the analogue output of a matched filter (they do not represent the *reliability* of each bit). The optimum reliability of each *symbol* is given by [121]:

$$\alpha_i = \begin{cases} 1\ , & A_i > 2s \\ A_i\ , & 0 \le A_i \le 2s \\ 0\ , & A_i < 0 \end{cases} \qquad A_i = \ln \frac{P(R_i | H_i)}{\sum_{H_j \neq H_i} P(R_i | H_j)} \tag{G.3}$$

where the expression for A_i can be expressed in terms of the individual symbol bits :

$$P(R_i | H_i) = P_0(R_i, H_i) \qquad \sum_{H_j \neq H_i} P(R_i | H_j) = \sum_{j=1}^{2^n - 1} P_j(R_i, H_i) \tag{G.4}$$

where

$$P_j(R_i, H_i) = \prod_{l=0}^{n} P(r_{il} | h_{il}^{\delta_{jl}}) \tag{G.5}$$

and δ_{jl} is the l-th bit (l= *1,2 . . n*) from the lowest in the binary expansion of integer j; h_{ij} is described as follows :

$$h_{ij}^{\partial_{ij}} = \begin{cases} h_{ij} & \partial_{ij} = 0 \\ h_{ij}'(inverse\ of\ h_{ij}) & \partial_{ij} = 1 \end{cases} \quad \text{(G.6)}$$

In practice, this analogue weight for non binary symbol expressed as binary n-tuples, involves a lot of calculations, and it invariably impractical. A sub-optimal method of weight generation, described in [121], uses a method is called most suspicious bit (MSB). The value of L_i is simply taken to be proportional to the least reliable bit of that symbol :

$$\begin{aligned} L_i &= 4r_{min}\sqrt{E}/N_o \\ r_{min} &= min(|r_{i1}|,|r_{i2}|\ldots|r_{in}|) \end{aligned} \quad \text{(G.7)}$$

where $\sqrt{E}/N_o$ represents the signal to noise ratio. The difference in performance of the optimal weight generation and MSB weight generation methods is negligible, in the AWGN channel.

Appendix G.2 Sub-Optimal Soft Decision Decoding

Once the vectors y and α are known, the next stage is to find the most probable transmitted vector. For normal hard decision decoding, the Hamming distance between the received code word and the valid code words is used. The code word with the lowest hamming distance is chosen as the correct code word. The Hamming distance is defined as below :

$$d_H(a,b) = \begin{cases} 0, & a \neq b \\ 1, & a = b \end{cases} \quad \text{(G.8)}$$

so, the *hard* distance between the received code word, y and a valid code word, c, is

$$D_H(y,c) = \sum_{i=1}^{n-1} d_H(y_i,c_i) \quad \text{(G.9)}$$

The *soft* distance between the received code word, y and a valid code word, c, is

$$D_\alpha(y,c) = \sum_{i=0}^{n-1} \alpha_i d_H(y_i,c_i) \quad \text{(G.10)}$$

The optimum method of soft decision decoding is full minimum soft distance decoding (MSDD) [110]. This is done by comparing the soft distances between every possible code word and the received code word. There is generally no simple way of doing this, short of searching the whole set of code words, which can become impractical for large codes. The following sub-optimal procedures minimise the decoder complexity and achieve a large part of the maximum improvement theoretically possible through analogue (unquantised) correlation.

Appendix G.2.1 Forced Erasure Decoding

Forced erasure decoding [122][123] is a sub-optimum method, closely related to minimum soft distance decoding. It is also known as successive erasure minimum distance decoding

(SEMDD) [112]. In this method the reliability information for each symbol is utilised in the decoding process. Let e be the error correction capability of the code with hard decision decoding (i.e. $e = \lfloor (d-1)/2 \rfloor$ where d is the minimum distance). Forced erasure decoding is capable of decoding $2e$ errors providing that these errors are identified as being the symbols with the lowest reliability. In fact, a high percentage of patterns containing $(2e + 1)$ errors are correctable, and a smaller percentage of $(2e + 2)$ and so on, up to a small percentage of $(n\text{-}k)$ errors. For cyclic codes, all erasure bursts of length $\leq (n\text{-}k)$ are correctable, since bursts of length $\leq (n\text{-}k)$ are detectable.

The procedure is outlined as follows: For each received n bit word (on which initial 'hard' decisions are made, see p316, [98]) there is a set of n positive numbers, $w_1, w_2, \ldots w_n$ which correspond to the posterior confidence levels of each bit, obtained from the Q-ary output channel. Using the notation $\min_j(w_i)$ to mean the jth lowest value among w_i we first determine $\min_1(A_i)$. We then erase this digit and attempt to reconstruct it whilst simultaneously satisfying the parity equations. For the binary case this simply means inverting the initial hard decision and testing for a zero syndrome. If this is unsuccessful (due to errors in the non erased digits) we then determine $\min_2(A_i)$. These two digits are then erased and reconstruction of the word is attempted. For a non binary input channel, erasures should be replaced by the symbol closest to the original values of the erased bits. If this is unsuccessful we continue with $\min_3(A_i)$. We could repeat this until $(n\text{-}k)$ digits have been erased and no solution has been found, in which case an uncorrectable error pattern has occurred.

It can be seen that the overall decoding time depends on the particular error pattern. Analytic expressions for the performance of forced-erasure decoding are extremely complicated, and the improvement of this decoding system over hard decision decoding is often evaluated by simulation [123].

Appendix G.2.2 Weighted Erasure Decoding

This was proposed by Weldon in 1971 [114]. The algorithm is closely related to Forney's Generalised minimum distance decoding [112] and is applicable to all block codes and Q-ary output channels where Q>2. The work was extended for burst errors by Wainberg and Wolf in 1972 [124]. Unlike forced erasure decoding this procedure does not have a variable decoding delay according to the erasure pattern. A detailed description of the algorithm is given in [114]. A brief description follows :

The channel has Q possible output values. Each value is given a particular weight, w_i, subject to the following restrictions:

$$0 = w_0 \le w_1 \le w_2 \le \ldots\ldots \le w_{Q-1} = 1 \tag{G.11}$$

and

$$w_i + w_{Q-1-i} = 1 \tag{G.12}$$

The w distance between two levels is defined as :

$$d_w(L_i, L_0) = d_w(L_0, L_i) = w_i \tag{G.13}$$

$$d_w(L_i, L_{Q-1}) = d_w(L_{Q-1}, L_i) = w_{Q-1-i} \tag{G.14}$$

It has been shown (p716, [114]) that almost optimum performance can be obtained by setting the weights w_i at equal intervals, i.e. $w_i = i/(Q-1)$. The following theorem is well known for hard decision decoding:

> **Theorem G.1** *:* A binary (n,k) code with a minimum distance d can correct any error pattern such that
>
> $$\sum_{v=0}^{n-1} e_v = E < d/2 \tag{G.15}$$

The equivalent theorem for soft decision encoding is as follows :

> **Theorem G.2***:* Let N_i denote the number of error digits of weight w_i. Any error pattern is correctable provided
>
> $$\sum_{i=0}^{Q-1} N_i w_i < d/2 \tag{G.16}$$

So, for theorem G.1 we are concerned with the *Hamming* weight of the error vector, i.e. a single error vector of weight one would be [0000001000]. For theorem G.2, the actual weight of each error is used (i.e. the w distance from either w_0 or w_{Q-1}).

Each value of w_i can be expressed as a binary vector, with each bit a_i corresponding to a weight, v_i, such that :

$$w_i = a_{ri} v_r + a_{(r-1)i} v_{r-1} + \ldots\ldots + a_{1i} v_1 \tag{G.17}$$

and:

$$\sum_{\sigma=1}^{r} v_\sigma = w_{Q-1} = 1 \tag{G.18}$$

and $r \le (Q+1)/2$. For example, for Q=7, we could choose $v_1 = 1/Q$, $v_2 = 2/Q$, $v_3 = 4/Q$. For Q=9 we could choose: $v_1 = 1/Q$, $v_2 = 2/Q$, $v_3 = 2/Q$, $v_4 = 4/Q$ (Note that the all one vector must represent a weight of 1).

The received vector can now be represented as an $r \times n$ array of binary digits. Each row of this vector is decoded in the normal way. For the σth row the number F_σ represents the number of changes made in this row (i.e. how many bits are changed). Now we assign each row a reliability indicator :

$$R_\sigma = \max(0, d - 2F_\sigma) \tag{G.19}$$

The decoding of each bit is as follows: Consider the first column of the $r \times n$ array. Certain of these digits are 1, are others are zero. Let S_0 and S_1 denote the set of rows corresponding to 0 and 1 in the first column. Choose the first information digit to be 0 if

$$\sum_{S_0} R_\sigma v_\sigma > \sum_{S_1} R_\sigma v_\sigma \tag{G.20}$$

otherwise choose this digit to be a 1.

A similar algorithm can be applied to the decoding of product codes although the weighting of each bit in a column vector comes from the number of bits corrected in the row vector, not from a Q-ary output channel (See Appendix 1, [114]).

Appendix H. Publications Arising from this Work

1. Collins M. J., Martin J. D. : 'Synchronisation of Cyclic Coset Codes for Railway Track Circuit Data', Proc. 1996 ASME/IEEE Joint Railroad Conference, pp72-76.

2. Collins M. J., Martin, J. D.: 'Performance of Railway Track Circuit Codes with Burst Interference' Proc. IEEE Vehicular Technology Conference 1997, vol.3, pp1927-1931

Synchronisation of Cyclic Coset Codes for Railway Track Circuit Data.

M. J. Collins, J. D. Martin
School of Electronic and Electrical Engineering
University of Bath
BA2 7AY, UK
e-mail : eepmjac@bath.ac.uk

Abstract : **The data capacity of railway track circuits can be increased by the use of binary block codes. The application of standard block coding techniques is limited because of the fail safe requirement and because each code must be independently synchronisable. Also, the hostile electrical environment suggests that error detection/correction should be employed. The two problems of error detection/correction and synchronisation are simultaneously addressed by cyclic coset codes, without incorporating any special signals or bit patterns for 'start of message' or 'end of message' indicators. Codes suitable for track circuit data are analysed in terms of their synchronisation error performance and a new measure of the comma free properties of these codes, the *overlap probability distribution*, is introduced. This provides a better criterion for the choice of coset codes than is given by the index of comma freedom alone.**

1. Introduction

In railway signalling, the purpose of the track circuit is to prove that a section of track is clear of all vehicles. The track circuit is also often used to communicate a limited amount of information to the train via antennas inductively coupled to the rails. This information would typically include the identity of the track circuit and speed commands to the train. The coding scheme should include redundancy to combat interference, since accurate data transfer is vital. Provision of a large number of track circuit identities would protect against wrong side failure in the presence of longitudinal and lateral leakage (Fig. 1).

FSK modulation has been used in the past, where the information is contained in the modulation frequency, and this is known as rate coding. Generally the signal bandwidth and hence the information capacity is limited by the tuned impedances defining the track circuit boundaries and by channel distortion. To increase the information capacity many methods of serial data coding have been proposed [1][2][3]. The signalling rate and track circuit response time

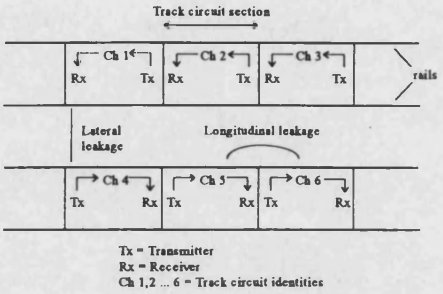


Figure 1. Track circuit leakage

are typically limited to 100 baud and 0.5 seconds respectively.

The application of standard coding techniques to track circuits is restricted because of the way the track circuit operates and because of the harsh electrical environment. A train entering a new block has no prior knowledge of synchronisation. The track circuit code will be repeated cyclically and synchronisation must be achieved on the reception of the first valid code word. This implies that the codes must be comma free[1].

The use of error detection/correction techniques further complicates matters. These methods assume correct synchronisation at the receiver and so it must be shown that the syndromes of all possible overlapped words are not the syndromes of correctable error patterns. This means that the receiver is able to distinguish between error patterns caused by additive errors and error patterns caused by misframing. The two problems of error detection/correction and synchronisation are addressed simultaneously by cyclic coset codes, without incorporating any special signals or bit patterns for 'start of message' or 'end of message' indicators.

[1] Comma freedom dictates than no overlap between any two code words (including the same code word) will contain a bit sequence that is the same as any code word.

This technique has been used in military and space systems such as the NASA Gemini and Apollo projects [4][5]. The technique requires fewer bits than required for standard methods of message framing, to achieve the same level of code performance.

2. Comma Free Cyclic Codes

Comma-free cyclic codes were first suggested by Stiffler in 1965 [6] and by Frey in the same year [7]. The work was extended by Tong in 1968 [8]. By their definition, cyclic codes are not comma-free, but they can be equipped with comma freedom by forming cosets. Stiffler states that if $k < n/2$, then any cyclic (n,k) code can be made comma-free by, for example, complementing the first and last bit of every code word. More generally, a fixed polynomial, p(X) is added to each code vector before transmission:

$$t(X) = a(X)g(X) + p(X) \qquad \text{Equation 1}$$

where t(X) is the transmitted code polynomial, of degree less than n, g(X) is the generator polynomial of degree n-k, and a(X) has degree less than k. The receiver will subtract p(X) from the received polynomial, leaving the additive error detection/correction capabilities unchanged.

Let's suppose that an r bit synchronisation loss occurs, with no additive errors. The received code word will be :

$$r(X) = X^r t(X) + d(X) \qquad \text{Equation 2}$$

where d(X) has degree r-1 or less and has unknown coefficients. The first step at the receiver is to subtract p(X). The resulting polynomial is :

$$r(X) - p(X) = (X^r - 1)p(X) + X^r a(X)g(X) + d(X) \qquad \text{Equation 3}$$

The syndrome of this is calculated by dividing by g(X) and taking the remainder (or in other words, the syndrome is the received polynomial, modulo g(X)). Since the middle term is a multiple of g(X) the syndrome for $p(X) = X^{n-1} + 1$ is :

$$s(X) = \left[X^{r-1} - 1 + X^r - X^{n-1} + d(X)\right] \bmod g(X) \qquad \text{Equation 4}$$

Similar patterns occur for sync gain. The syndrome is actually the result of dividing the error pattern expressed as a polynomial, by g(X) and by taking the remainder. The term in square brackets in equation 4 is the error pattern caused by misframing for the coset code defined by $p(X)=X^{n-1}+1$. It can be seen that the minimum possible weight of error pattern is two, and so this code is restricted to single error correction.

3. Overlap Probability Distributions.

The *index of comma freedom, s,* is the minimum distance between a valid code word and an overlapped code word, and should be as high as possible. The choice of p(X) will affect the index of comma freedom which in turn will affect the amount of error correction possible. However, if we take into account all possible instances of misframing and assume that they are all equally likely, then there is a low probability of an overlap occurring which is distance *s* from a valid code word. The majority of misframed words have a larger distance from the nearest valid code word.

There is no known method of predicting the distribution of distances from all code overlaps to the valid code words. A computer program has therefore been developed to investigate the properties of specific coset codes. A probability distribution can be formed, which gives the overall probability, for all values of misframing, of a misframed word being a certain distance from a valid code word, *given that misframing has occurred.* We will call this distribution the *overlap probability distribution,* $p_o(i)$, where i represents the distance from a valid code word. Figure 2 shows $p_o(i)$ for the (15,5) BCH coset code defined by the Barker sequence $p(X)= X^4+X^3+X^2+1$.

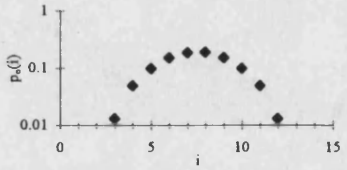


Fig 2 : (15,5) BCH Code, $p(X)=X^4+X^3+X^2+1$

A steep curve is desirable, giving a low probability that a small number of bit errors in a misframed word will cause a decoding error. Note that this code has a comma free index

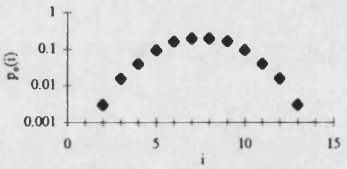


Fig 3 : (15,7) BCH Code, $p(X)=X^{14}+1$

of 3 enabling double error correction.

Cosets of the (15,7) BCH code have been studied. The coset given by $p(X)=X^{14}+1$ has a comma free index of 2, as shown in figure 3.

Figure 4 shows the overlap probability distribution for the (21,9) shortened cyclic coset code defined by the Barker sequence $p(X)=X^6+X^5+X^4+X$. This code had a comma free index of 3 and is capable of correcting all single bursts of length up to 6 when correctly synchronised.

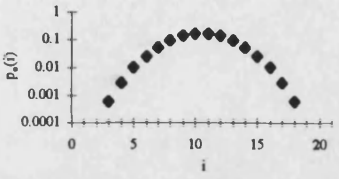


Fig 4 : (21,9) Shortened Cyclic Code, $p(X)=X^6+X^5+X^4+X$

The largest code studied is the (31,11) BCH code. This code has a minimum distance of 11. The steepest overlap probability distribution (figure 5) was found with the coset defined by $p(X)= X^{10}+X^9+X^8+ X^4+X$. Again, this is a Barker sequence.

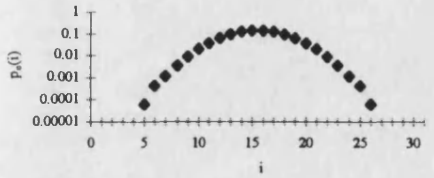


Fig 5 : (31,11) BCH Code, $p(X)= X^{10}+X^9+X^8+ X^4+X$

Repetition codes were introduced by Levy [9.] for generating cyclic codes with higher indices of comma freedom. A ρ-repetition (ρn,k) cyclic code can be generated from an (n,k) cyclic code by repeating the generator polynomial ρ times, thus the (28,7) code is a four fold repetition of the trivial (7,7) code. The overlap probability for the (28,7) coset code defined by $p(X)=X^6+X^5+X^4+X$ is given in figure 6. It has an index of comma freedom of 4.

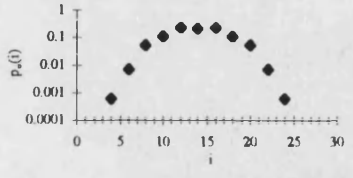


Fig 6 : (28,7) Repetition Code, $p(X)= X^6+X^5+X^4+X$

4. Code Performance

Notation

The following probabilities are defined :

$P_{u.f.}$ Prob of decoding error for a framed word (unsafe failure)

$P_{s.f.}$ Prob of detected error for a framed word (safe failure)

$P_{c.d.}$ Prob of correct decoding for a framed word

$P_{m.u.f.}$ Prob of decoding error for misframed word (unsafe failure)

$P_{m.s.f.}$ Prob of error detection for misframed word (safe failure)

4.1 Synchronised Words

The probability of decoding error (unsafe failure) with no error correction is show below.

$$P_{u.f.} = \sum_{i=1}^{n} A_i p^i (1-p)^{n-i} \qquad \text{Equation 5}$$

where A is the weight distribution of the code. For a t error correcting code the condition $d_{min} \geq 2t + 1$ must apply, where d_{min} is the minimum distance of the code. Equation 5 can be modified to include all error patterns that are distance t and below, from each valid code word.

$$P_{u.f.} = \sum_{j=-t}^{t} \sum_{i=1}^{n} A_i p^{i+j} (1-p)^{n-i-j} \qquad \text{Equation 6}$$

Correct decoding occurs when t or less errors occur.

$$P_{c.d.} = \sum_{i=0}^{t} \binom{n}{i} p^i (1-p)^{n-i} \qquad \text{Equation 7}$$

The safe failure probability is shown below.

$$P_{s.f.} = 1 - (P_{c.d.} + P_{u.f.}) \qquad \text{Equation 8}$$

4.2 Misframed Words.

For misframed words, correct operation will occur when an error is detected, since this will cause the receiver to ignore this frame and look at the next frame. Failure will occur when bit errors change the misframed word into a valid code word or correctable code word. For no error correction, this is shown below :

$$P_{m.u.f.} = \sum_{i=1}^{n} \binom{n}{i} p_e(i) p^i (1-p)^{n-i} \qquad \text{Equation 9}$$

For t error correction, equation 9 becomes

$$P_{m.u.f.} = \sum_{j=t-1}^{n-1} \sum_{i=1}^{n} \binom{n}{i} p_e(i) p^{i+j} (1-p)^{n-i-j} \qquad \text{Equation 10}$$

The probability of error detection for a misframed word is:

$$P_{m.s.f.} = 1 - P_{m.u.f.} \qquad \text{Equation 11}$$

4.3 Overall Synchronisation Performance.

Before the synchronisation performance of the code can be assessed, the decoding philosophy must be specified. The decoder must identify valid blocks of n-bit words. At switch on, or when a train is leaving a track circuit section, the receiver has no prior knowledge of synchronisation. It will receive the first n bits and store them in a buffer or *window*. These n bits may or may not represent a valid code word. The window will then slide as the next bit is received and the first bit is discarded. The receiver must decode two consecutive code words before a decision is made and these code words must also be the same. Once the receiver decides that synchronisation has been achieved it need look only at consecutive blocks of n bits. This is because, for cyclic coset codes the index of comma freedom is typically less than the minimum distance of the code itself. The overall performance of the code will be expressed in terms of correct operation, unsafe failure and safe failure.

Correct Operation :- Decoding of a correctly framed word, no decoding of the subsequent (n-1) overlapped words, and decoding of the next word.

Unsafe Failure :- The corruption of a correctly framed word into another code word, followed by (n-1) non valid code words, and the corruption of the next correctly framed word into the same valid code word.
OR :- The decoding of a valid code word at an overlap position, followed by (n-1) non valid code words, then the decoding of the same valid code word at an overlap position.

Safe Failure :- Decoding error for a correctly framed code word, followed by anything other than (n-1) undecoded overlapped words and a decoding error for the next correctly framed word, into the same code word.
OR :- The reception of a valid code word at an overlap position followed by anything other than (n-1) undecoded words, then the same valid code word at an overlap position.

Mutually exclusive events can be constructed from equations 5-11, and identified as either correct operation, safe failure or unsafe failure.

5. Synchronisation Failure Probabilities

5.1 (15,5) BCH Code

The (15,5) BCH coset code defined by $p(X)=X^4+X^3+X^2+1$ has an index of comma freedom of 3. The synchronisation failure probabilities are shown in figure 7.

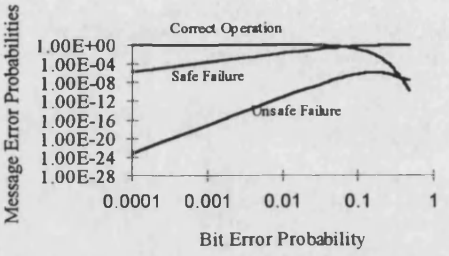


Fig 7 : (15,5) BCH Coset Code with Single Error Correction.

Figure 8 shows the (15,7) BCH coset code, which has an index of comma freedom of 2.

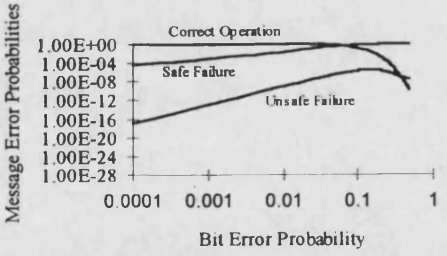


Fig 8 : (15,7) BCH Coset Code with Single Error Correction.

Figure 9 shows the (21,9) shortened cyclic code with single error correction. Figure 10 shows the (31,11) BCH coset code with double error correction.

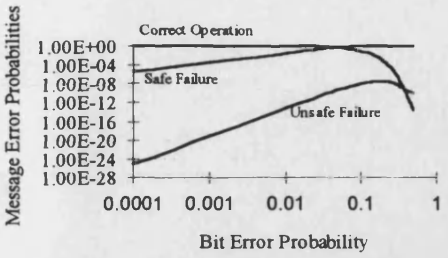


Fig 9 : (21,9) Shortened Cyclic Code with Single Error Correction

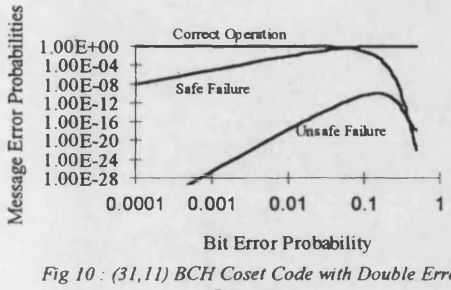


Fig 10 : (31,11) BCH Coset Code with Double Error Correction.

6. Conclusions

Many cosets of the codes shown were investigated. It is interesting to note that in many cases the best overlap probability distribution is achieved when the coset is defined by a Barker sequence. For the (15,7) and (21,9) codes the optimum trade off between safety (error detection) and reliability (error correction) is achieved with single error correction. For the (31,11) BCH coset code the optimum performance is achieved with double error correction, because this code has a steep overlap probability distribution.

Codes with a higher data capacity are preferable, but the code length is limited because of the response time and the data rate of the code. The (31,11) BCH code has a better synchronisation performance than the other codes presented, but the data rate is lower.

The method of constructing codes with a high index of comma freedom [9] is at the expense of the data rate and minimum distance of the code. The index of comma freedom alone is not a suitable criteria for the synchronisation properties of a comma free code. The coset codes in this paper have been chosen for their overlap probability distribution. The unsafe failure probabilities of the codes presented are significantly lower than those obtained by comma codes [1]. There is, however, a penalty in detection time since the receiver must decode two consecutive codes words before a decision is made. These codes have the advantage of easy construction and decoding using well known algorithms, since the same redundancy is used for the error correction/detection and synchronisation.

The data allocation within the code has not been addressed in this paper. The data is typically split into track to train data and track circuit identity. For example, the (31,11) BCH coset code could have 5 bits for train data (32 messages) and 6 bits for track circuit identity (64 messages).

7. References

[1.] Hill, R. J. ; 'Optimal Construction of Synchronizable Coding for Railway Track Circuit Data Transmission', IEEE Trans. on Vehicular Technology, Vol.39, No.4. Nov 1990, pp390-399.

[2.] Barker, D., Poole, D. ; 'Digitally Coded Track Circuit', IRSE International Conference on Railway Safety, Control and Automation towards the 21st Century, London, 25-27 Sept 1984, pp333-336

[3.] Chapront, P. ; 'Transmission and Decoding of Vital Data', IRSE International Conference on Railway Safetey, Control and Automation towards the 21st Century, London, 25-27 Sept 1984, pp161-165.

[4.] Maxwell M. S. and Kutz R. L., 'An efficient PCM error correction and synchronisation code', NASA Tech. note TND-2317, National Aeronautics and Space Administration, Washington D.C., June 1964.

[5.] Rudin M. B. and Fiorino F., 'Advancements in the design and evaluation of group synchronisers for PCM telemetry', Record 1962 National Symposium on Space Electronics and Telemetry.

[6.] Stiffler, J. J., 'Comma-Free Error-Correcting Codes', IEEE Trans. on Information Theory, Vol.IT-11, Jan 1965, pp107-112.

[7.] Frey A. H., 'Message framing and error control', IEEE Trans. on Military Electronics, April 1965, pp143-147.

[8.] Tong, S. Y. ; 'Synchronization Recovery Techniques for Binary Cyclic Codes', Bell System Tech. J., 45, 1968, pp561-596.

[9.] Levy, J. E., 'Self-Synchronising codes derived from Binary Cyclic Codes', IEEE Trans. on Information Theory, Vol.IT-12, No 3, July 1966, pp286-290.

PERFORMANCE OF RAILWAY TRACK CIRCUIT CODES WITH BURST INTERFERENCE

M. J. Collins, J. D. Martin.

The authors are with the School of Electronic and Electrical Engineering, University of Bath, England, BA2 7AY, UK.
e-mail: j.d.martin@bath.ac.uk

Abstract - The data capacity of railway track circuits can be increased by the use of binary block codes. A comma-free coding scheme suitable for railway track circuit data is presented which out performs the other comma-free code construction methods in terms of synchronisation in Gaussian noise. However, the interference encountered in a railway environment is likely to be impulsive and bursty rather than Gaussian. Simulation results of the (31,11) BCH coset code modulated by non-coherent FSK are presented, using an impulsive interference model which more accurately represents typical railway noise.

I. INTRODUCTION

In railway signalling, the purpose of the track circuit is to prove that a section of track is clear of all vehicles. The track circuit is also often used to communicate a limited amount of information to the train via antennas inductively coupled to the rails. This information would typically include the identity of the track circuit and speed commands to the train. The coding scheme should include redundancy to combat interference, since accurate data transfer is vital.

The application of standard coding techniques to track circuits is restricted because of the way the track circuit operates and because of the harsh electrical environment. A train entering a new block has no prior knowledge of synchronisation. The track circuit code will be repeated cyclically and synchronisation must be achieved on the reception of the first valid code word. This implies that the codes must be comma-free[1]. The two problems of error correction/detection and synchronisation are simultaneously addressed by cyclic coset codes. This paper will present a method of construction and analysis of these codes with AWGN. Also presented are simulation results that support the theory and can be extended to non-Gaussian noise which models the expected interference more accurately.

[1] Comma freedom dictates than no overlap between any two code words (including the same code word) will contain a bit sequence that is the same as any code word.

II. COMMA-FREE CYCLIC COSET CODES

By their definition, cyclic codes are not comma-free, but they can be equipped with comma freedom by forming cosets [1][2]. A fixed polynomial $P(X)$ is added to each code word (expressed as a polynomial in X) before transmission.

$$t(X) = a(X)g(X) + p(X) \tag{1}$$

where $t(X)$ is the transmitted code polynomial, of degree less than n (code length), $g(X)$ is the generator polynomial of degree n-k, and $a(X)$ has degree $\leq k$-1. The receiver will subtract $p(X)$ from the received polynomial, leaving the additive error detection/correction capabilities unchanged. For example, $P(X)=X^{n-1}+1$ will have the effect of inverting the first and last bit of every code word.

A. Code Performance - Correctly Synchronised Words

For correctly synchronised cyclic code words, with no error correction and independent bit errors, the probability of decoding error is given by

$$P_{d.e.} = \sum_{i=1}^{n} A_i p^i (1-p)^{n-i} \tag{2}$$

where A_i is the weight distribution, and p is the bit error probability. For a t error correcting code the condition $d_{min} \geq 2t + 1$ must apply, where d_{min} is the minimum distance of the code. In this case, (2) must be modified to include all code words which are a distance of t or less from the valid code words.

$$P_{d.e.} = \sum_{i=2t+1}^{n} A_i p^i (1-p)^{n-i} + \sum_{j=1}^{t} \sum_{i=2t+1}^{n} A_i \binom{n-i}{j} p^{i+j} (1-p)^{n-i-j} + \sum_{j=1}^{t} \sum_{i=2t+1}^{n} A_i \binom{i}{j} p^{i-j} (1-p)^{n-i+j} \tag{3}$$

Correct decoding occurs when there are t or less bit errors:

$$P_{c.d.} = \sum_{i=0}^{t} \binom{n}{i} p^i (1-p)^{n-i} \tag{4}$$

B. Code Performance - Misframed Words

Different values of $P(X)$ will produce cosets with different synchronisation properties. The *overlap weight distribution*, O_i, can be used to assess the comma-free properties of a particular coset [3]. This is similar to the weight distribution of the correctly framed words. It contains the total number of unique distance vectors from the valid code words to the misframed words, of weight i. The lowest weight is the *index of comma freedom*. If we normalise the overlap weight distribution, dividing by the total number of weights considered, we will get the *overlap probability distribution*, p_{o_i}. This can be used to compare the synchronisation properties of different codes and cosets. For example, the overlap probability distribution for the (31,11) BCH coset code is shown below.

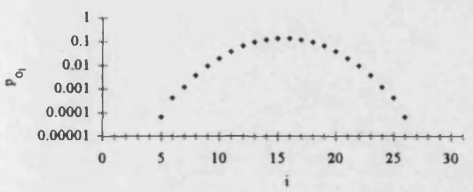


Fig. 1 : (31,11) BCH Code, $p(X) = X^{10}+X^9+X^8+X^4+X$

We can see that the index of comma freedom, c, in this case is 5 which means we can correct up to 4 random errors whilst searching for synchronisation. There is a very low probability of 5 bit errors causing a synchronisation failure, since the majority of misframed words are a greater distance from the valid code words.

When the receiver is searching for code synchronisation (i.e. at switch on, or when the train has just entered a track circuit), it will receive the first n bits. If these do not represent a valid or correctable code word then the receiver window slides on to the next bit. A synchronisation error occurs when bit errors change a misframed word into a valid or correctable code word. The probability of this occurring is given by :

$$P_{m.f.} = \sum_{i=c}^{n-c} O_i p^i (1-p)^{n-i} + \sum_{j=1}^{t}\sum_{i=c}^{n-c} O_i \binom{n-i}{j} p^{i+j}(1-p)^{n-i-j} + \sum_{j=1}^{t}\sum_{i=c}^{n-c} O_i \binom{i}{j} p^{i-j}(1-p)^{n-i+j} \quad (5)$$

The probability of an error detection on a misframed word (correct operation) is given by

$$P_{m.c} = 1 - P_{m.f} \quad (6)$$

III. CODE SYNCHRONISATION PERFORMANCE

The expressions above can be combined with an overall receiver decoding philosophy to obtain the synchronisation performance. For example, extra error detection is obtained by requiring the receiver to decode two consecutive code words before a decision is made. Fig. 2 shows the synchronisation performance of the (31,11) BCH coset code with this decoding philosophy. The optimum reliability was obtained with double error correction, while the decoding philosophy keeps the unsafe failure rate very low.

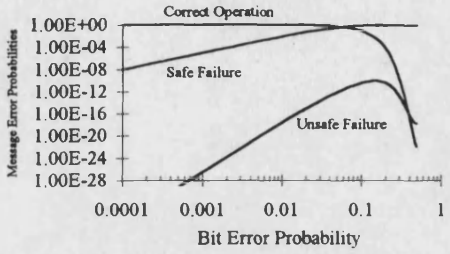


Fig. 2 : Synchronisation Performance of the (31,11) BCH Coset Code with Double Error Correction.

Many smaller codes have also been investigated [3]. Fig. 3 shows the synchronisation performance of the (15,7) code with single error correction. Since this code has less redundancy than the (31,11) BCH code, the index of comma freedom is only 3, and the peak unsafe failure rate is higher.

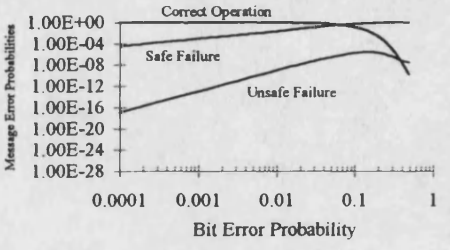


Fig. 3 : Synchronisation Performance of the (15,7) BCH Coset Code with Single Error Correction.

IV. NON-COHERENT FSK SIMULATION

A software simulation of the binary coding scheme, modulated by FSK and non-coherently demodulated has been performed. Non-coherent FSK has been used successfully in the past for track circuit modulation [4][5]. The aim of the simulation is to verify the theory of equations (1) to (4) and to assess the code performance with typical non-Gaussian interference. The simulation process is shown in Fig. 4. The demodulator contains 2 integrate and dump filters centred on the two symbol frequencies. A bit decision is made on the frequency with the highest power. The simulation is performed separately on correctly synchronised words and misframed words.

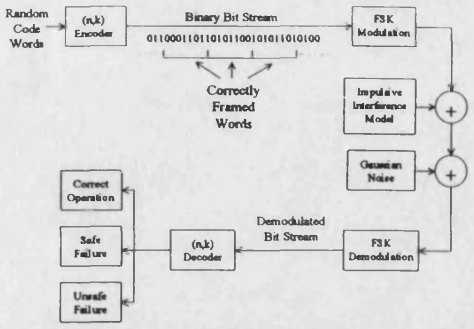


Fig. 4 : Block Diagram of Simulation

A. Results for Gaussian Noise

Fig. 5 shows the simulation results for correctly framed (21,6) shortened cyclic coset code words [6] with single error correction. When correctly synchronised, this code is capable of correcting all single bursts of length 6. However, the best coset investigated was found to give an index of comma freedom of 3, restricting the error correction to 2 random errors. The optimum performance is obtained with single error correction.

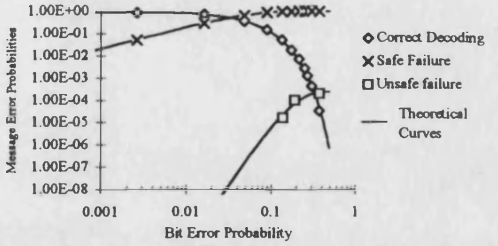


Fig. 5 : Performance of Correctly Framed (21,9) Shortened Cyclic Code Words with Gaussian Noise

Note that a decoding error corresponds to an unsafe failure, and an error detection corresponds to a safe failure. The simulation results agree with the theoretical equations. Fig. 6 shows the simulation results for misframed (21,9) shortened cyclic code words with single error correction. These results show the probability of bit errors changing the misframed word into a valid or correctable code word and give us an indication of the synchronisation properties of the code. Figs. 7 and 8 show the results for the (31,11) BCH coset code for correctly synchronised words and misframed words, with double error correction.

B. Results for Impulsive Interference

A suitable impulsive interference model has been defined which is a more accurate representation of typical railway interference. The track circuit coding scheme must operate

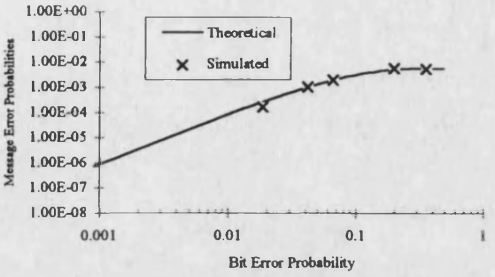


Fig. 6 : Performance of Misframed (21,9) Shortened Cyclic Code Words with Gaussian Noise

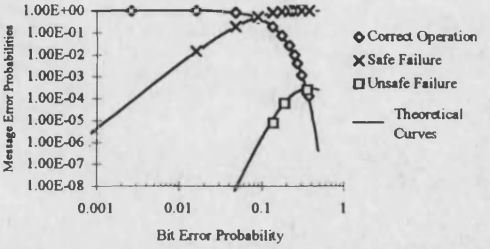


Fig. 7 : Performance of Correctly Framed (31,11) BCH Cyclic Code Words with Gaussian Noise.

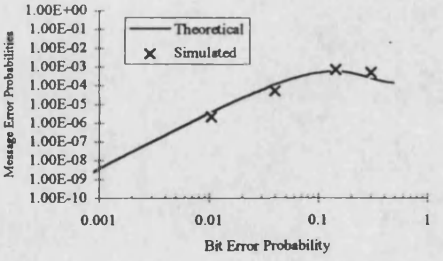


Fig. 8 : Performance of Misframed (31,11) BCH Cyclic Code Words with Gaussian Noise

with all classes of traction equipment and electrification, and so site measurements specific to a particular location and traction system are of little use. A common noise characteristic of electrical traction systems seems to be a periodic impulse train, with a slowly varying period not less than 3ms, depending on the state of the train. There are also many sources of isolated impulses in a railway environment.

The simple model used for these simulations consists of an impulse train whose period varies at a rate of 1ms/s. The lowest period is 3ms, and the highest period is chosen such that one impulse occurs within the observation interval (one code word).

Fig. 9 shows the results for correctly framed (31,11) BCH coset code words with no error correction. The failure probabilities are plotted against the overall SNR, defined as $10\log_{10}$(average signal power/average noise power, after receiver BPF). Fig. 10 shows the same results with double error correction.

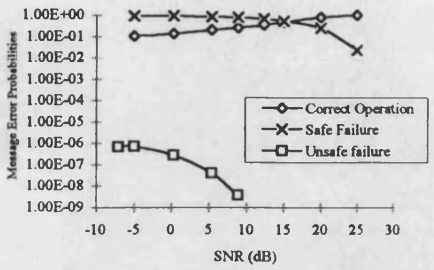


Fig. 9 : Performance of Correctly Framed (31,11) BCH Coset Code Words with Impulsive Interference and No Error Correction

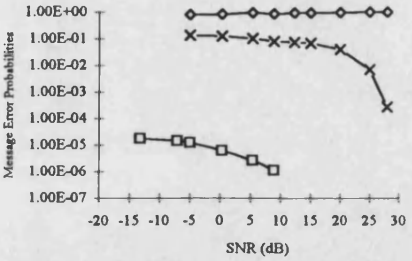


Fig. 10 : Performance of Correctly Framed (31,11) BCH Coset Code Words with Impulsive Interference and Double Error Correction

It can be seen that the curves for correct decoding and safe failure (error detection) are flat for a wide SNR range and fall sharply above 20dB. This is because all the noise power is concentrated in bursts and so will cause bit errors even at high SNRs. Comparison of these curves in Figs. 9 and 10 suggests that for correctly synchronised words, error correction is a more effective method of increasing the reliability of the code than increasing the signal power.

Fig. 11 shows the results for misframed words with no error correction and double error correction. Again the curves are flat for a wide range of SNR. We see that the use of error correction whilst searching for synchronisation increases the false synchronisation probability. An unsafe failure is avoided however, by requiring that the receiver decodes 2 consecutive code words before a decision is made that synchronisation has been achieved.

Optimum synchronisation performance of the (31,11) BCH code in Gaussian noise was obtained with double error correction (Fig. 2). Shown in Fig. 12 are the failure probabilities of correctly framed (31,11) BCH coset code words with double error correction, plotted against overall bit error probability. Results for both Gaussian and impulsive noise are shown for comparison.

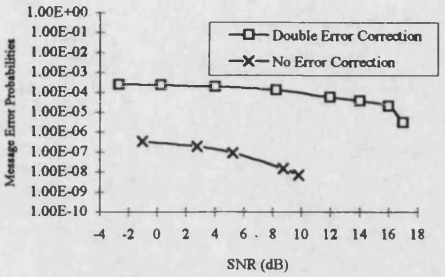


Fig. 11: Performance of Misframed (31,11) BCH Coset Code Words with Impulsive Interference and No/Double Error Correction

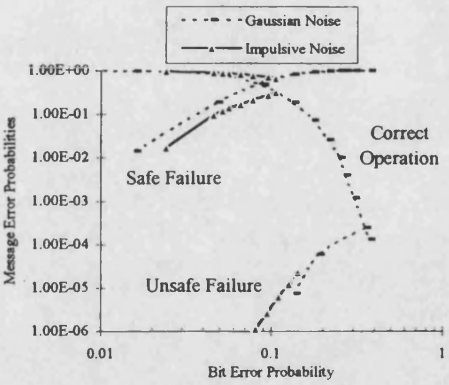


Fig. 12: Performance of Correctly Framed (31,11) BCH Coset Code Words with Gaussian and Impulsive Interference and Double Error Correction.

We can see that the reliability of the code is slightly greater with impulsive noise, when the same overall bit error probability is compared. With this increase in reliability comes an increase in decoding error probability, although the difference is marginal.

Fig. 13 shows the failure probabilities of misframed (31,11) BCH coset code words, with both Gaussian and Impulsive interference. We can see that for impulsive noise, the peak failure probability is marginally lower, and falls off more rapidly for lower bit error rates. This may be explained by thinking about the error patterns caused by the interference and the distance vectors from the misframed words to the valid codes words. These distance vectors have random rather than bursty patterns, and so the failure probability for

misframed words is less with bursty errors, when the same overall bit error probability is compared.

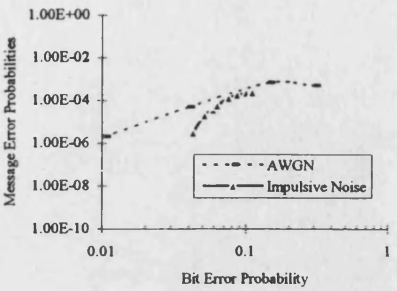


Fig. 13: Performance of Misframed (31,11) BCH Coset Code Words with Gaussian and Impulsive Interference and Double Error Correction

V. CONCLUSIONS

A method of construction and analysis of the synchronisation properties of comma-free cyclic coset codes suitable for railway track circuits has been presented. A new measure, the overlap weight distribution, is introduced, which provides a better measure of the synchronisation performance of the code than is given by the index of comma freedom alone. Analytical analysis of the performance, combining the code failure probability equations with an overall decoding philosophy allows the low unsafe failure rates of different codes to be compared, and the amount of error correction to be optimised.

Many different codes have been studied. The BCH codes are attractive for their powerful error correction properties when correctly synchronised, and well known encoding/decoding algorithms. Comma-free cosets are easily formed with the use of special synchronisation sequences. The best code studied is the (31,11) BCH coset code, which has 2047 code words and an index of comma freedom of 5. The overlap weight distribution is formed by a computer search, but the run time is prohibitive for longer codes.

The use of error correction increases the reliability of the code when correctly synchronised, but also increases the probability of false synchronisation, since fewer bit errors can change a misframed word into a valid or correctable code word. The decoding philosophy dictating that two consecutive code words be received before a decision is made, provides the extra error detection needed to give an extremely low false synchronisation probability (Fig. 2).

A simulation of the code performance has been performed, to verify the theoretical equations and to assess the performance with a simple non-Gaussian noise model. This impulsive noise model will produce bursty rather than random errors. Due to the comma-free requirement of the code, burst error correction is not possible and so there is little advantage in using specially constructed single burst error correcting codes.

The simulation was performed separately on correctly framed words and misframed words. For the (31,11) BCH code, the impulsive interference model results in a slightly higher correct decoding curve and unsafe failure curve when the results are plotted against overall bit error probability (Fig. 12) and compared with those for Gaussian noise. This difference, however, is marginal.

The results for misframed (31,11) coset code words with impulsive noise give us an indication of the synchronisation performance of the code. The failure probabilities for misframed (31,11) coset code words are actually lower with impulsive noise (Fig. 13) and we can conclude that this comma-free code is suitable for use in such interference as well as in Gaussian noise.

For both the impulsive and Gaussian noise models it was seen that the use of double error correction on a misframed word increases the error probability by a factor of approximately 10^2. The total synchronisation error probability would be reduced if error correction were only used when the receiver decides that synchronisation has been achieved. The possible increase in safe synchronisation failure (decrease in reliability) would have to be investigated.

ACKNOWLEDGEMENT

This work has been funded by Westinghouse Signals Ltd. and The University of Bath, UK.

REFERENCES

[1] Stiffler, J. J., 'Comma-Free Error-Correcting Codes', IEEE Trans. on Information Theory, Vol.IT-11, Jan 1965, pp107-112.

[2] Frey A. H., 'Message framing and error control', IEEE Trans. on Military Electronics, April 1965, pp143-147.

[3] Collins, M. J. Martin, J. D. ; 'Synchronisation of Cyclic Coset Codes for Railway Track Circuit Codes', Proceedings of the 1996 ASME/IEEE Joint Railroad Conference, pp137-142.

[4] Hill, R. J. ; 'Optimal Construction of Synchronizable Coding for Railway Track Circuit Data Transmission', IEEE Trans. on Vehicular Technology, Vol.39, No.4. Nov. 1990, pp390-399.

[5] Barker, D., Poole, D. ; 'Digitally Coded Track Circuit', IRSE International Conference on Railway Safety, Control and Automation towards the 21st Century, London, 25-27 Sept. 1984, pp333-336

[6] Kasami T., Matoba, S. ; 'Some Efficient Shortened Cyclic Codes for Burst Error Correction', IEEE Trans. on Information Theory, Vol. IT-10, pp 252-253, July 1964.