1999

RAPID CLOCK RECOVERY ALGORITHMS FOR DIGITAL MAGNETIC RECORDING AND DATA COMMUNICATIONS

SMITHSON, PAUL MICHAEL

Volume 1

RAPID CLOCK RECOVERY ALGORITHMS FOR DIGITAL MAGNETIC RECORDING AND DATA COMMUNICATIONS

by

PAUL MICHAEL SMITHSON

A thesis submitted to the University of Plymouth
in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

School of Electronic, Communication and Electrical Engineering
Faculty of Technology

February 1999

Author's declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award.

Several IEE and IEEE colloquia's and international conferences have been attended.

Seven refereed papers have been published and presented at international conferences, relevant publications of which are reproduced in Appendix 11.

In addition this research has been included, and the author referenced, within the text of two published technical textbooks [1] and [2].

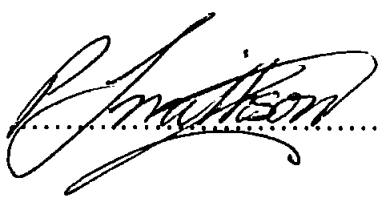
Signed  Date 28-4-99

Table of Contents

Volume 1

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Volume 2

Appendices

List of Figures

LIST OF FIGURES FOR VOLUME 1

LIST OF FIGURES FOR VOLUME 1

List of Tables

Glossary of Abbreviations

ADC	Analogue-to-Digital Converter
AGC	Automatic Gain Control
AM	Amplitude Modulation
APLL	Analogue Phase Locked Loop
ASCII	American Standard Code for Information Interchange
ASIC	Application Specific Integrated Circuit
BER	Bit Error Rate
BIOS	Basic Input and Output System
BNSC	British National Space Centre
BPF	Band Pass Filter
BPSK	Binary Phase Shift Keying
CAD	Computer Aided Design
CCIR-601	Recommendations for digitising video signals
CCIR-656	Recommendations for interfacing CCIR 601 signals
CCITT	Consultative Committee for International Telegraph & Telecommunication
CD	Compact Disk
CLB	Configurable Logic Block
CMOS	Complementary Metal Oxide Semiconductor
COFF	Common Object File Format
CR	Clock Recovery
CRC	Cyclic Redundancy Check
CRSYNC	Clock Recovery Synchroniser
D1	Professional digital video recording format in composite form. (CCIR-656)
D2	Professional digital video recording format in composite form. (NTSC, PAL, SECAM)
DAB	Digital Audio Broadcasting
DAC	Digital-to-Analogue Converter
DASH	Digital Audio Stationary Head
DBPSK	Differential Binary Phase Shift Keying
DBS	Direct Broadcast Satellite
DC	Direct Current
DCRSi	Digital Cassette Recorder System Incremental
DDS	Direct Digital Synthesiser
DFT	Discrete Fourier Transform
DIL	Dual in Line
DIT	Decimation in Time

DLL	Delay Locked Loop
DM	Delay Modulation
DMA	Direct Memory Access
DOS	Disk Operating System
DPLL	Digital Phase Locked Loop
DSP	Digital Signal Processing or Digital Signal Processor
DSV	Digital Sum Variation
DTH	Direct to Home
DTTL	Digital Data Transition Tracking Loop
DVB	Digital Video Broadcasting
DVC	Digital Video Cassette
EBN	European Business News
EBU	European Broadcasting Union
ECPD	Edge Controlled Phase Detector
EPROM	Erasable Programmable Read Only Memory
ESA	European Space Centre
FDD	Frequency Difference Detector
FDS	Filter Design System
FEC	Forward Error Correction
FFT	Fast Fourier Transform
FIFO	First-In-First-Out memory
FIR	Finite Impulse Response (filter)
FM	Frequency Modulation
FPD	Frequency Phase Detector
FPGA	Field Programmable Gate Array
FSK	Frequency Shift Keying
GPIB	General Purpose Interface Bus (IEEE488 standard)
GOPS	Giga Operations per Second
HC	High speed CMOS
HCT	High speed CMOS (TTL compatible)
HD	Hard (rigid disk) Drive
HEX	Hexadecimal
HPF	High Pass Filter
IEE	Institute of Electrical Engineers
IEEE	Institute of Electrical and Electronic Engineers
IF	Intermediate Frequency
IIR	Infinite Impulse Response (filter)

IO	Input and Output
ISA	Industry Standard Architecture
ISI	Inter-Symbol Interference
ISO	International Standards Organisation
ISP	Internet Service Provider
ITU	International Telecommunications Union
JTAG	Joint Test Action Group
LAN	Local Area Network
LC	Inductor and Capacitor
LED	Light Emitting Diode
LMS	Least Mean Squared
LPF	Low Pass Filter
LSB	Least Significant Bit
MAP	Maximum *a posterori*
MFM	Modified Frequency Modulation
MIPS	Million Instructions per Second
ML	Maximum Likelihood
MPEG	Motion Pictures Expert Group
MSB	Most Significant Bit
MSK	Multiple Shift Keying
MUX	Multiplexer
NRZ	Non Return to Zero
NRZI	Non Return to Zero Inverse
NRZ-L	Non Return to Zero Level
NRZ-M	Non Return to Zero Mark
NTSC	National Television Standard Committee
OFFT	Offset Fast Fourier Transform
PAL	Phase Alternate Line
PC	Personal Computer
PD	Phase Detector
PDF	Probability Density Function
PFD	Phase Frequency Detector
PISO	Parallel-In-Serial-Out
PLD	Programmable Logic Device
PLL	Phase Locked Loop
PRBS	Pseudo Random Binary Sequence

PSD	Power Spectra Density
PSK	Phase Shift Keying
PSU	Power Supply Unit
QPSK	Quadrature Phase Shift Keying
RAM	Random Access Memory
RDAT	Rotary-head Digital Audio Tape recorder
RDS	Radio Data System
RLL	Run Length Limited
RLL3	Maximum run length limited to 3 (PRBS)
RLL7	Maximum run length limited to 7 (PRBS)
ROM	Read Only Memory
RS232	Standard asynchronous serial communication interface
RS422	Standard differential interface (twisted pair)
RZ	Return to Zero
SAW	Surface Acoustic Wave (filter)
SCPC	Single Carrier per Channel
SISP	Satellite Internet Service Provision
SMPTE	Society of Motion Pictures and TV Engineers
SNG	Satellite News Gathering
SNR	Signal to Noise Ratio
SPW	Signal Processing Worksystem
SWDS	SoftWare Development System
TDM	Time Division Multiplexing
TDMA	Time Division Multiple Access
TTL	Transistor-Transistor-Logic
TV	TeleVision
TVRO	TeleVision Receive Only
UART	Universal Asynchronous Receiver Transmitter
UW	Unique Word
VCO	Voltage Controlled Oscillator
VSAT	Very Small Aperture Terminal
WAN	Wide Area Network
WWW	World-Wide-Web

Introduction

The 1990's may well be considered as the era of information technology, with the prevalence of computers, digital communications and hence the need for data storage. Diverse data storage media have been introduced, such as silicon (flash memory) [3] and the CD (Compact Disk) [4], however improvements in magnetic recording has resulted in magnetic media as the prominent leader in mass storage technology. With improvements to the storage medium, the read/write heads and the channel electronics, the capacity of magnetic rigid disks has approximately doubled every 3 years (period 1960-90) [5]. These improvements to the channel, coupled with fast data-access times, has resulted in the magnetic disk drive remaining the dominant data storage device in computer technology.

As storage densities increased, digital audio became viable and, driven by the enormous consumer market, products such as the audio CD [6] and RDAT (Rotary Head Digital Audio Tape) [7] have taken over from their analogue counterparts, as state of the art digital recording.

Data communications is another huge growth area, with satellite data communications providing exciting new applications such as MPEG-2 (Motion Pictures Expert Group) compressed digital TV [8], Internet access [9] and personal communications [10].

These new applications have been made possible largely due to the advances in digital electronics. DSP (Digital Signal Processing) processors now offer more than 2 GOPS (Giga 10^9 Operations per Second) [11] while semi-custom devices such as FPGA's (Field Programmable Gate Array), provide more than 130,000 gates with toggle rates in excess of 150MHz [12]. The cost of the more modest DSP devices have reduced considerably and such devices with on-chip memory can provide a dual role of signal processing and control, replacing the more common microprocessor.

Digital solutions are often preferred over their analogue equivalents since designs are usually more repeatable, reliable and robust. As DSP devices become faster and more powerful, the work that the processor can perform may be increased to encompass such areas as clock recovery, channel filtering, demodulation, error correction and control. DSP allows the use of re-configurable hardware, resulting in a flexible and cost-effective product. Low data rate systems are already emerging [13], and it is likely that the DSP-based algorithms developed today will be adopted by higher speed systems, as faster and more powerful devices become available.

This thesis restricts attention to the topic of clock recovery, which is a vital processing part of digital magnetic recording and communications. Despite much work in this field, few novel advances have been made in this important area of data recovery (see Section 1.1 Clock recovery architectures), which is surprising when

one considers that potential improvements can be achieved, both in terms of storage capacity, bandwidth and BER (Bit Error Rate).

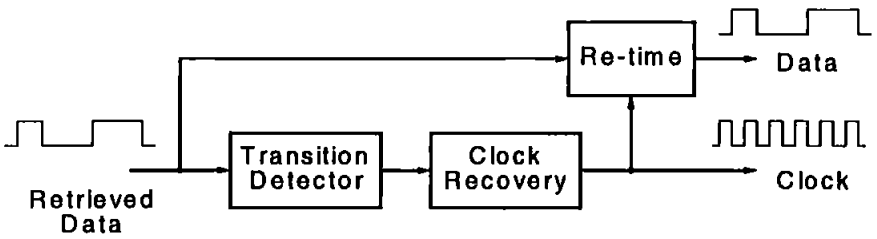


Figure 1 Clock recovery used for synchronous re-timing of data

In data recording systems and data modem receivers, it is necessary to generate a time-base, which is synchronised to the retrieved data. The recovered clock is normally extracted from timing information contained within the data-bearing signal. Figure 1 illustrates a clock synchroniser, where the recovered clock is used to re-time the retrieved data, and for subsequent synchronous post-processing. The channel-coding scheme used to store or transmit the data is in part chosen to provide a strong spectral component at the clock rate and such codes are termed "self clocking" [14].

The principle of locking a local oscillator within the receiver, to the frequency and phase of the retrieved data, is typically used to derive a synchronised time-base. The most common technique for this purpose is the basic PLL (Phased-Locked Loop) [15]. In order to ensure that the PLL is in a locked condition before data are retrieved, a preamble containing a synchronising sequence must be applied before data can be accurately clocked.

In magnetic recording, the synchronising preamble must be stored on the recording medium. This occupies capacity that could otherwise be used for data. Each data block recorded is preceded by a preamble of transitions whose main purpose is to provide a timing reference for synchronising the PLL. Often, in tape recording systems, a post-amble is also applied to allow a block to be read in the reverse direction and to prevent a transient, when the heads are turned off, from corrupting the last bits written [16]. Latest hard-disk drives [17] [18] devote approximately 6% of the total storage capacity to the purpose of clock synchronisation.

A clock recovery system with improved acquisition would release some of this capacity for the storage of useful data as well as increasing data-access time.

Also, loss of lock and synchronisation can occur if the data synchronising the PLL "drops out" for a period of time, due to defects on the magnetic medium, and problems can occur in regaining lock if the data signal is poor. The "dropout", together with prolonged re-acquisition time, results in corrupted data.

Similarly, in burst-mode communication systems, a preamble is transmitted to allow the receiver to achieve synchronisation before message data can be decoded for subsequent processing. For certain applications the message data can be very short, for example a network "ping" packet [19]. In such cases the synchronising preamble may be longer than the message itself, clearly an undesirable arrangement. Furthermore, once synchronisation has been achieved and the packet successfully received, a guard band must be applied before the next packet can be transmitted.

This is to allow the synchroniser to drop out of lock, thereby allowing a new acquisition cycle to begin [20]. This is in direct conflict to the essential flywheel feature necessary to maintain synchronisation over periods of lost data. Terrestrial and satellite communication systems also suffer from temporary loss of data when the signal fades. This is typically due to adverse weather conditions, such as heavy rain or multi-path reflections producing signal cancellation.

While magnetic recording and communication systems appear to be widely different technologies, they do in fact share a common objective; namely the retrieval of burst data. The investigation of clock synchronisation described in this thesis is therefore relevant to both technologies and the algorithms investigated will be helpful in solving many of the problems identified.

Clock recovery systems derive their timing information from the retrieved data transitions. Hence, the channel-coding scheme employed influences the performance of the clock recovery system. One area that is distinctly different between magnetic recording and communication systems, is the method of increasing the channel efficiency. Broadband communication systems use multilevel signals and hence the channel capacity can be improved by employing complex modulation schemes. These include amplitude, frequency and phase modulation or a combination of all three [21].

In general, digital magnetic recording operates with just two levels ±1. This is because the magnetic particles of the medium are saturated in one of two directions. The channel capacity may, however, be increased by the use of RLL (Run Length Limited) coding [22]. If data are encoded to contain fewer transitions, then these transitions can be recorded closer together for a given packing density, resulting in increased storage capacity. However this can affect the self-clocking properties of the coded data, which could adversely affect clock synchronisation.

Throughout this research, computer simulations have been performed to simulate and analyse system performance [23] [24]. SPW (Signal Processing Worksystem) is a powerful integrated software suite, which has been extensively employed for system design and analysis [25]. Full simulation models and results are located in the appendices, key extracts of which are included in the relevant chapters. Computer simulations have been standardised with the use of normalised frequencies. The sample rate has been normalised to 1Hz and the nominal symbol rate normalised to 16 samples/symbol = 0.0625Hz. The advantage of using normalised frequencies is that the system can easily be scaled to any absolute frequency. For example, at 16 samples/symbol a sample rate of 1MHz results in a symbol rate of 1MHz/16 = 62.5KHz.

Chapter 1 provides a background to the investigation. Current clock recovery algorithms are reviewed together with a summary of their limitations for burst data

applications. The channel-coding scheme employed is fundamental to clock recovery performance. Hence, channel coding is reviewed and five codes are selected as bench marks for subsequent analysis of clock recovery performance. These codes are described and have been selected to represent a typical cross-section of available codes. A summary of popular recording codes is provided to demonstrate the various trade-offs.

Three new DSP-based algorithms have been investigated:

- A fixed frequency clock recovery system.
- An adaptive (tracking) clock recovery system.
- An OFFT (Offset Fast Fourier Transform) based clock recovery system.

These algorithms together with results, are presented in Chapters 2, 3 and 4, respectively.

Chapter 2 describes a novel DSP-based clock recovery scheme utilising a high-Q IIR filter that demonstrates superior performance over that of a PLL, particularly in terms of clock acquisition. Long flywheel periods can be tolerated, maintaining clock synchronisation over severe data dropouts or fades. Simulation results are analysed for the new clock recovery scheme to determine clock acquisition, flywheel performance and clock jitter in the presence of noise. Performance using several magnetic recording codes is investigated. Hardware and software implementation of the new algorithm is described. The simulation results are

verified in real-time using a fixed-point digital signal processor [26]. The IIR filter algorithm has also been implemented in an FPGA, using fast coefficient multipliers [Appendix 11{3}]. This device has been successfully incorporated into a commercial satellite modem for reception of computer data, utilising a sub-carrier on an analogue FM satellite TV broadcast [Appendix 11{4}].

Chapter 3 describes the investigation of an adaptive tracking system, based upon two detection filters. These detection filters have a wider bandwidth than the tracking filter and are placed symmetrically around the nominal input (symbol rate) frequency. The output of the detection filters are envelope-detected and subtracted to generate a dc error-control signal which is used to move the tracking filter to the correct symbol rate.

Chapter 4 describes an OFFT [27] that is used to estimate the frequency of the retrieved symbols. The frequency estimate is used to determine the correct coefficients necessary to move the centre frequency of the clock recovery filter to the correct symbol rate. The OFFT has been implemented, and results verified in real-time, using a specially designed dual DSP board based around two fixed-point DSP processors [28].

Finally, Chapter 5 summarises the investigation with conclusions, a discussion of the results, applications and recommendations.

Chapter 1

1. Background to the investigation

Current clock recovery algorithms, which are represented by basic architectures, are reviewed. These architectures are described and their individual performance noted. A summary of their relative performance is provided, with particular reference to acquisition of burst data.

The channel-coding scheme employed is fundamental to clock recovery performance. Hence, the principles of channel coding are reviewed and five codes are selected as bench marks for subsequent analysis of clock recovery performance. These codes are Scrambled NRZ (Non Return to Zero) [29], Manchester [30], Miller [31], Miller^2 (Miller Squared) [32] and RLL2,7 [33]. These codes are described and have been selected to represent a typical cross-section of available codes, from high clock-content codes through to the more efficient block codes. In practice recording codes are pre-coded with NRZI (Non Return to Zero Inverse) [34] encoding to eliminate phase ambiguity and error propagation. The NRZI code is also described. A summary of popular recording codes is provided to demonstrate the various trade-offs.

1.1. Clock recovery architectures

Figure 1-1 illustrates the concept of a clock recovery system, where a local oscillator is synchronised to the input transitions, which oscillates freely in their absence. The oscillator must be capable of fly wheeling over periods of missing data transitions, with negligible drift.

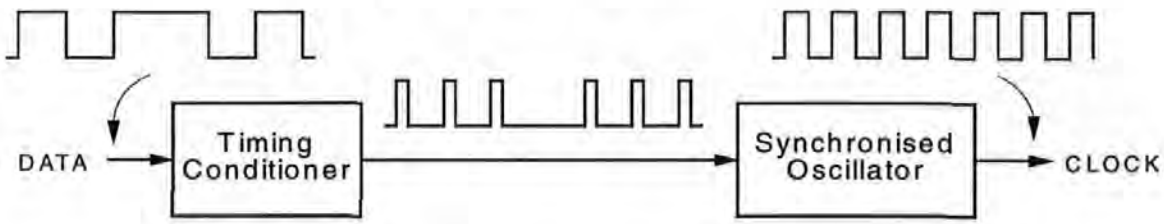


Figure 1-1 Clock recovery system

A clock recovery system in its simplest form can be represented by two basic building blocks; a timing conditioner and an oscillator that generates a periodic output synchronised to the input symbol rate.

The timing conditioner is required to extract, from the data-bearing signal, a strong spectral component at the symbol rate. This can be achieved by means of auto-correlation of the retrieved data, and this technique is described in Section 2.1.1. In its simplest form the timing conditioner can be implemented by means of a transition detector, where the data are delayed by the propagation delay of a few logic gates and then "Exclusive-Or" gated with the non-delayed version. Alternatively, a differentiator can be used, followed by a squaring circuit to invert the negative impulses.

Synchronisation of the oscillator to the input transitions is typically achieved by means of a feedback loop, employing phase error detection. The method of phase detection often governs the type of timing conditioner employed. Many detectors are level sensitive and require a rectangular pulse shape [35]. Whereas the use of a transition detector necessitates the use of an ECPD (Edge Controlled Phase Detector) [36], where the phase error is derived from the time difference between the input signal and the local clock, setting and resetting a flip-flop.

The retrieved symbols usually contain timing jitter due to channel noise [37]. It is necessary for the clock recovery system to reject jitter in the recovered clock and re-sample the retrieved symbols at the centre of the symbol period, thereby avoiding data errors, as illustrated in Figure 1-2.

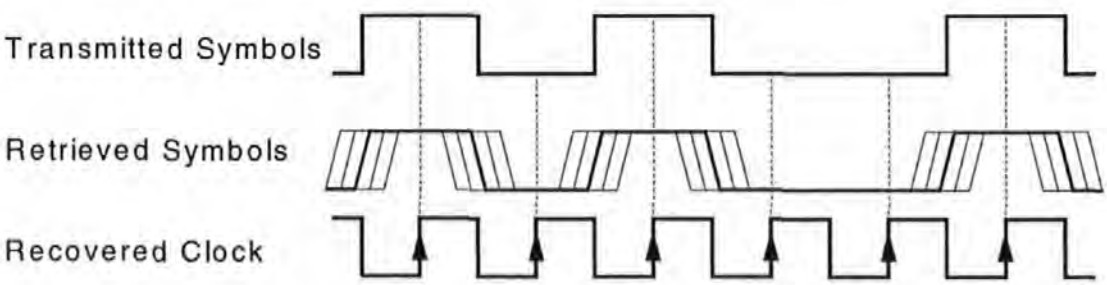


Figure 1-2 Optimum re-sampling

The frequency of the local oscillator is usually controlled by means of an error signal in a closed loop configuration. It is also possible to implement a burst oscillator in an open loop configuration by means of a high-Q self-synchronising ringing filter, and it is this concept that forms the basis of a DSP approach discussed in later chapters.

1.1.1. High-Q passive resonator clock recovery

In concept, the simplest clock recovery system, is one employing a high-Q passive resonating filter [38], as illustrated in Figure 1-3.

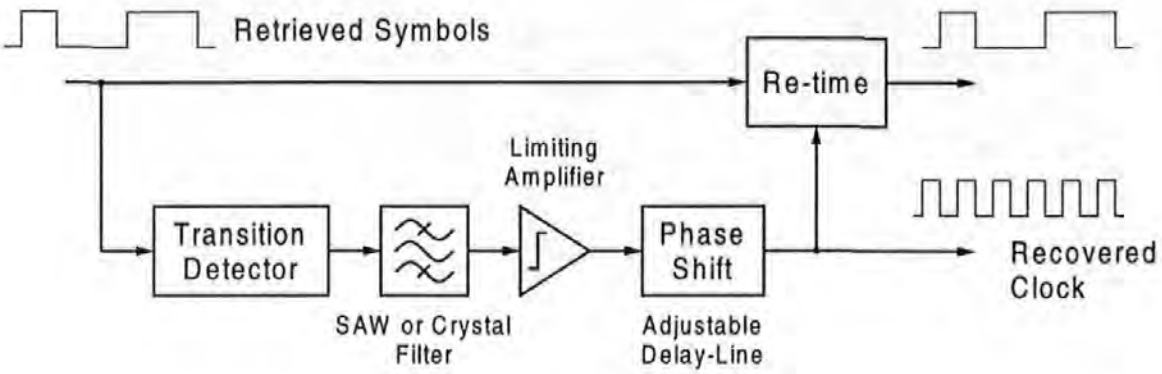


Figure 1-3 High-Q passive resonator clock recovery

A transition detector provides pulses at the symbol rate, which excites a high-Q bandpass filter, pre-tuned to the symbol frequency. The filter "rings" at the symbol rate and continues to oscillate for a period of time, even in the absence of input transitions. The filter output is provided with gain and hard limited to produce a recovered clock, which is then delayed to align the clock to the centre of the symbol period, for optimum re-sampling.

This simple approach, in several respects, provides an ideal clock recovery system. The recovered clock is rapidly acquired and flywheels over long symbol run-lengths. Also, the filter's high-Q, and hence, narrow bandwidth provides high noise immunity, virtually eliminating jitter in the recovered clock.

Hence, this system is often used as a benchmark, against which, clock recovery performance of other systems is compared. However, certain restrictive assumptions are made. Firstly, the symbol rate must be accurate and stable. This is usually the case in communications systems, where the symbol rate is derived from a crystal-controlled oscillator in the transmitter [39]. Unfortunately, this is rarely the case for tape recording systems, which suffer from mechanical limitations and tape stretch [40]. A second assumption is that the high-Q filter, is and remains, accurately aligned to the frequency and phase of the input symbols. Hence, the design of such circuits requires careful alignment of the clock, relative to the data, with consideration given to temperature, power supply, component ageing and process variations.

A passive tuned LC (inductors and capacitors) "tank" filter may be used to implement the high-Q filter, but the parameters of these components suffer from variations due to temperature change and ageing. To combat these variations, the high-Q filter is often implemented by means of a SAW (Surface Acoustic Wave) [41] or crystal filter [42]. These expensive devices are manufactured to a fixed frequency, centred at the symbol rate.

Clock recovery employing SAW and crystal filters are generally used at data rates above 200Mb/s [43], and are therefore unsuitable for use at low data rates. These devices exhibit a large insertion loss (-20dB is typical for a SAW filter) and hence, a high-gain limiting amplifier is required to generate a suitable logic-level clock, for

re-timing the data. A problem with these non-linear devices is that they exhibit an unknown group delay, due to manufacturing variations. The resulting phase variation must be compensated, with a manually adjusted delay-line, to achieve optimum timing for re-sampling the symbols.

It is not practical to integrate a SAW or crystal resonator into an integrated circuit and therefore, clock recovery schemes of this type are implemented using discrete components. While a clock recovery system employing a high-Q filter is attractive in concept, it is these practical considerations, which make a realisable system expensive and unpopular.

It is however, possible to implement a high-Q filter using DSP, thereby eliminating undesirable variations and the need for manual tuning. Hence, a repeatable, reliable and robust clock recovery system can be produced, which may be re-configured "on-the-fly" to accommodate a wide range of different symbol rates. It is this new approach to an old problem that forms the basis of the novel DSP algorithms, discussed in later chapters.

Example publications of clock recovery systems employing SAW or crystal filters are referenced in [44] and [45] respectively.

1.1.2. PLL clock recovery

The PLL is popular for clock recovery and forms the basis of most other types of closed-loop clock recovery systems. Appendices 1 and 2 supplement this section, analysing the performance of the basic PLL. Appendix 1 provides a glossary of commonly used PLL terminology, together with detailed analysis and loop equations. Appendix 2 contains PLL simulation models and results, which are used for comparison with the new DSP approach, discussed in later chapters.

A survey of APLLs (Analogue PLL) by Gupta [46] provides over 40 publications, while Lindsey and Chie [47] have carried out analysis and a survey of DPLLs (Digital PLL). More recently, monolithic PLLs have become popular, and Razavi [48] provides a collection of 47 publications, 15 of which are specific to clock recovery.

Figure 1-4 shows how a hybrid (digital and analogue) PLL can be used with a transition detector to perform clock recovery, from Manchester encoded data.

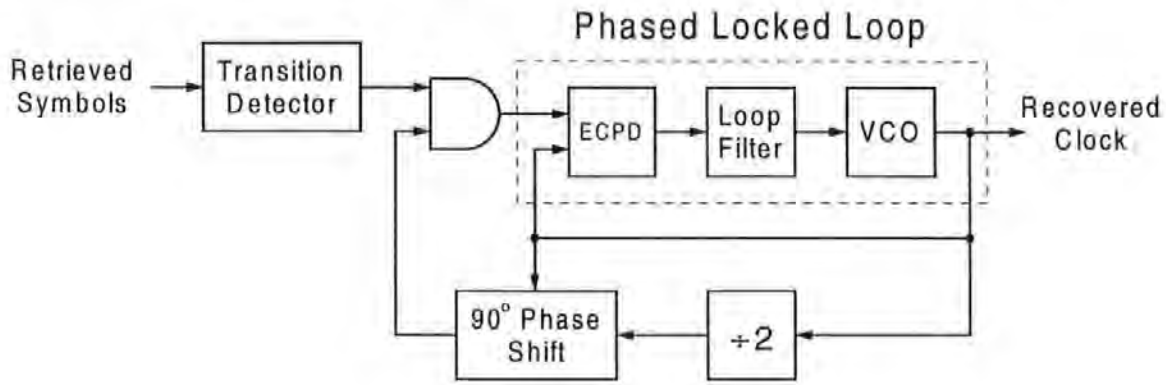


Figure 1-4 PLL clock recovery circuit

Retrieved symbols are applied to a transition detector producing pulses, which are gated with a 90^{o} phase-shifted version of the divided output clock. This provides periodic pulses at half the symbol rate for input to the PLL. These pulses are applied to an ECPD, producing a phase error signal proportional to the phase difference between the input and feedback transitions. The output from the ECPD is low-pass filtered, producing a dc control voltage to the VCO (Voltage Controlled Oscillator), so as to sustain or correct the output frequency. The negative edges of the output are used to clock the phase shifter, thereby providing the 90^{o} phase-shift.

Many PLL integrated circuits [49] contain a choice of PDs (Phase Detector). A multiplier or "Exclusive-Or" PD is level sensitive and requires rectangular pulses confined to the symbol period. A PFD (Phase Frequency Detector) can be edge controlled and is often employed to aid clock acquisition.

The simplest form of PFD has a 3-state output [50] for driving the loop filter, the average of which controls the VCO. If a phase error exist, then the PFD output attempts to drive the VCO control voltage to either its positive or negative extreme (depending upon the direction of the phase error) for the duration of the phase difference, and at a high impedance state for the remainder of the time. This produces a "sample-and-hold" effect providing a "charge-pump" to the capacitor at the loop (low-pass) filter. Hence the LPF (Low Pass Filter) achieves its final control voltage more rapidly than that of a simple PD, thereby aiding acquisition.

Simple flip-flop PFDs of this type rely on a periodic input, but unfortunately provide an incorrect output in the absence of input transitions. During missing data transitions, these simple detectors interpret the VCO output to be higher than the input frequency, driving the control voltage in such a direction so as to correct the apparent difference. The "Exclusive Or" PD on the other hand, drives the VCO to its nominal frequency in the absence of transitions. Therefore the choice of phase and frequency detectors, for random or burst data, requires careful consideration in the absence of missing data transitions.

Figure 1-5 illustrates an example of a DPLL, first described by Levy and Cessna [51], consisting of a sampler, a divide by N counter and a fixed rate clock oscillator. The operation is similar to an earlier design by Natali [52] employing a DCO (Digitally Controlled Oscillator).

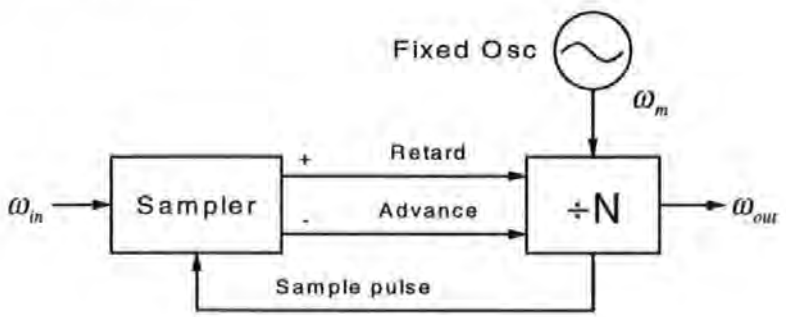


Figure 1-5 Lead-lag digital PLL

In the absence of feedback, the loop is designed for $\omega_m / N \cong \omega_{in}$. The $\div N$ counter counts down the oscillator frequency and generates a sample pulse each time it passes through a zero count. The sampler detects changes of input polarity and when the loop is phase-locked, the sampler is clocked synchronous to the input.

The counter is arranged so that its count is advanced by one increment if the sample is positive and decremented if it is negative. If the output is identical to the input frequency then the sampler detects an equal number of positive and negative polarities, resulting in cancellation of the advance and retard signals.
If however, the input frequency is increased, then the advance signal reduces the division by one count and the phase at the counter output increases by *1/N* cycles. Over successive cycles the change in phase at the counter output moves, so as to reduce the output frequency to that of the input.

This simplified DPLL is similar to a non-linear first order APLL, that is a loop with no filter. Unlike its analogue equivalent, the output phase has now been quantized creating a source of phase jitter in the recovered clock. This can be minimised with a high-frequency fixed oscillator and hence, a large $\div N$ counter.

With this type of DPLL, there is no VCO and this makes them far less sensitive to voltage or temperature variations. A useful feature of a DPLL is that the centre frequency and control bandwidth can be made programmable and adjusted by means of a microprocessor, hence these parameters can be dynamically adjusted.

The DLL (Delay Locked Loop) [53] is a close relative of the PLL and provides a means of replacing the VCO of an APLL with a voltage controlled delay-line. When in lock, the DLL delays its periodic input by an integer number of input periods (usually one), such that the relative phase shift at the PD can be considered

zero. For clock synchronisation a tapped delay-line can be employed, and the optimum timing selected for re-sampling the retrieved symbols. However, unlike the PLL the DLL does not contain an oscillator and therefore does not generate its own clock. Hence, the DLL is only suitable for generating an output at the same frequency as its periodic input.

Sonntag and Leonowich [54] have presented a DPLL for burst-mode re-timing of Manchester data, incorporating a DLL, as shown in Figure 1-6.

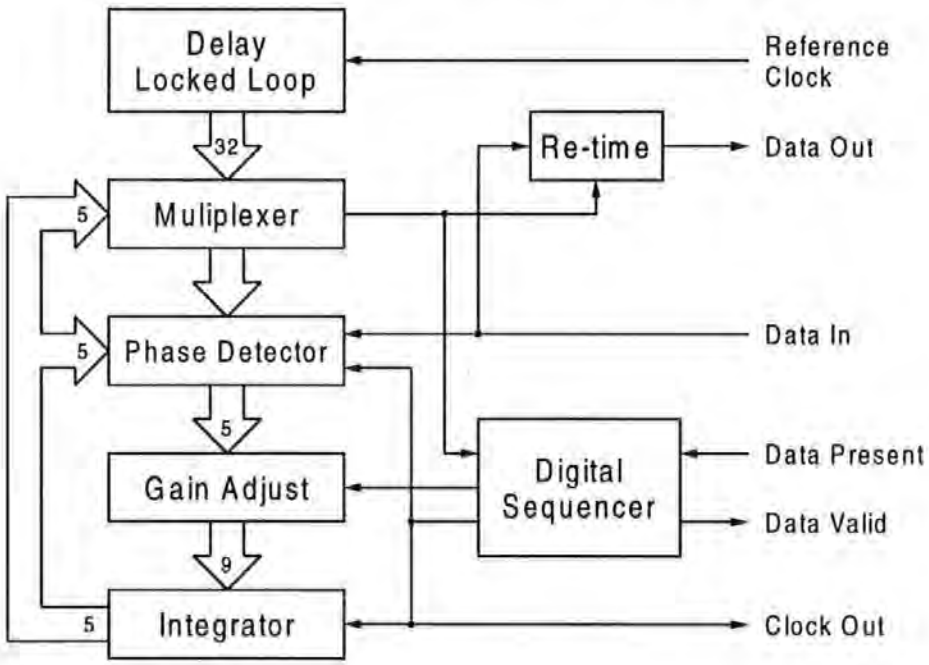


Figure 1-6 DPLL with DLL phase selection

A stable fixed-frequency reference clock is input to a DLL containing a digital delay-line, providing 32 equally spaced clock phases. The "Phase Detector", "Gain Adjust" and "Integrator" blocks, form a DPLL. When the DPLL is locked, a multiplexer selects 1 of 32 phases for optimum re-timing of the data. The selected

phase is also input to a PD, which produces a 5-bit data word, representing the phase difference between the selected local clock and the input data. The resulting error signal is provided with gain and integrated, producing an address to the multiplexer to select the appropriate clock phase.

Upon detection of burst data, an acquisition cycle is initiated and a digital sequencer applies a pseudo-clock, based upon the retrieved data transitions. The pseudo-clock drives the DPLL until a suitable DLL clock phase is found. When the loop is locked, the pseudo-clock is replaced with the DLL clock for the duration of the burst. The referenced publication [54] quotes a preamble of just 3 transitions, required to align the phase of the local DLL clock to the optimum timing.

A problem with this algorithm is that when acquisition is complete, the pseudo clock is replaced with the locally generated DLL clock. The DLL clock is derived from a reference frequency, which is not locked to the retrieved data. Hence, this algorithm assumes that the frequency of the locally generated DLL clock matches closely the symbol rate frequency, and that the data burst is short. Otherwise, drift between the locally generated clock with respect to the retrieved data symbols, would eventually produce cycle slips, resulting in data errors. The output clock also suffers from quantization jitter, resulting from the discrete DLL phases.

Examples of clock recovery algorithms used on magnetic recording systems, employing combinations of PLLs and DLLs are referenced in [55] [56].

1.1.3. Quadricorrelator

A clock recovery architecture that has been implemented in both analogue and digital domains is the "Quadricorrelator". The Quadricorrelator was first presented by Shaeffer [57], given its name by Richman [58] and modified by Belliso [59]. Shown in Figure 1-7 is a block diagram of the Quadricorrelator, which contains a combination of three loops sharing the same VCO.

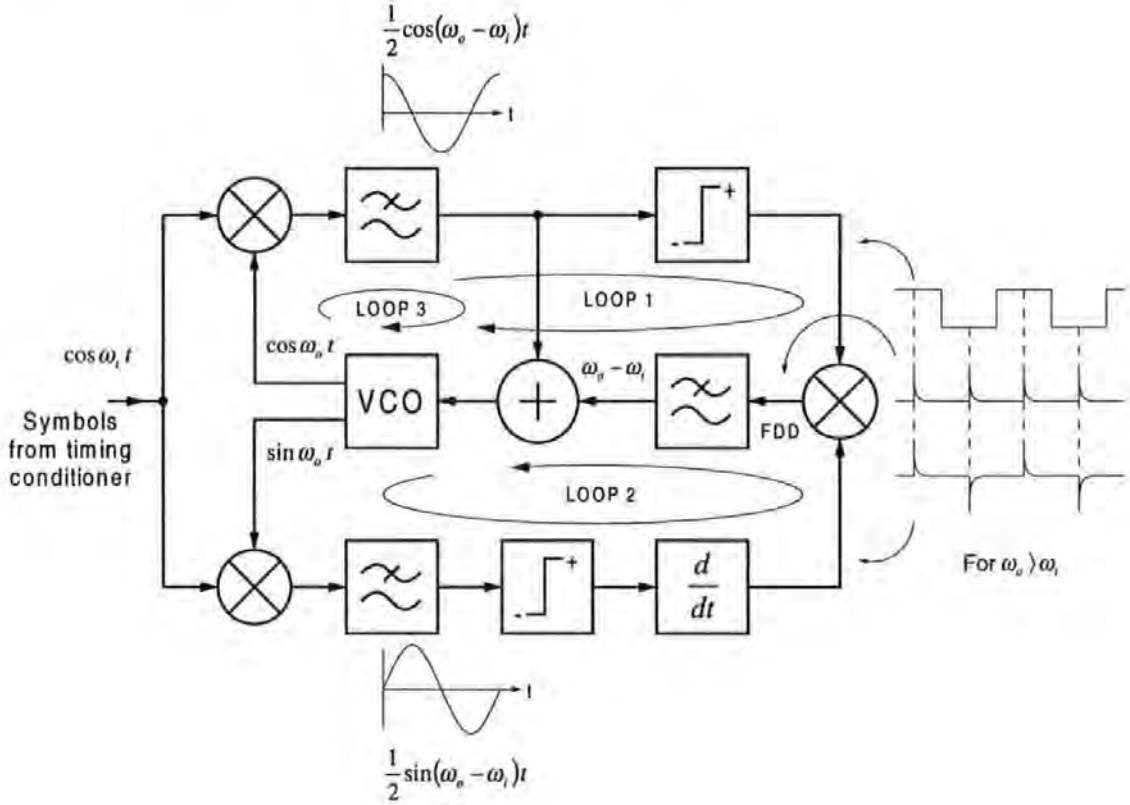


Figure 1-7 Quadricorrelator

Loops 1 and 2 perform frequency detection and form a FDD (Frequency Difference Detector), while Loop 3 provides a simple PLL. The input from a timing conditioner contains a strong symbol timing frequency component ω_i, while the VCO oscillates at ω_o.

When the loop is out of lock, the quadrature phase detectors (multipliers) generate beat signals with a 90° phase difference, one proportional to the cosine of the frequency difference, and the other the sine. One will lead the other by 90° if the VCO frequency is too high. The reverse is the case, if the VCO frequency is too low. This reversal in the phase relationship is detected by a third detector (FDD), and is used to produce either a positive or negative signal, depending upon the direction of the phase error. This signal is then low-pass filtered to produce a dc control voltage to adjust the VCO, so as to reduce the frequency error.

The differentiator in Loop 2 produces a positive and negative pulse each time a cycle is slipped (i.e. one cycle of the beat signal). These pulses are then multiplied by the quantized output of Loop 1, producing unipolar pulses, whose polarity is indicative of the sign of the frequency difference. Two pulses are produced for each cycle slip, and the dc output from the LPF is proportional to the frequency difference, ω_o-ω_i.

As the magnitude of ω_o-ω_i drops, Loop 3 (a simple PLL) begins to assist the lock process. When the VCO frequency approaches the output frequency, the dc feedback from the FDD approaches zero relinquishing control, and Loop 3 dominates locking the VCO output to the input signal.

When lock is achieved, the beat cycles disappear (due to no cycle slips), and the output from the differentiator is zero. Hence, the output from the FDD multiplier is also zero, and therefore does not detract from the PLL performance.

The use of frequency detection in the Quadricorrelator makes the capture range independent of the locked loop bandwidth, allowing a narrow PLL bandwidth in the LPF of Loop 3. In such applications it is more desirable to replace the filter in Loop 3 with two LPFs, one with a wide bandwidth for Loop 1 (frequency detection), and another with a narrow bandwidth in Loop 3 (phase detection and tracking). This will then provide good jitter rejection and memory (to flywheel over symbol run lengths or dropouts), once phase lock has been achieved. Because the frequency detection circuits can respond to noise and spurious components due to their wide bandwidths, it is also preferable to disable Loops 1 and 2, once phase lock has been attained.

A drawback of the Quadricorrelator in discrete form is the requirement for a VCO with quadrature outputs. This can be achieved by passing the VCO output through a delay line to shift the phase by 90°. However, dependence of the delay upon frequency, temperature and component tolerance make a stable design difficult.

Theoretical analysis of the Quadricorrelator is referenced in [59] and [60]. Publications of clock recovery systems employing the Quadricorrelator are referenced in [61]-[63].

1.1.4. Gated oscillator clock recovery

Banu and Dunlop [64] have presented an open-loop burst-mode clock recovery system, with instantaneous locking to the first retrieved transition. The block diagram is shown in Figure 1-8.

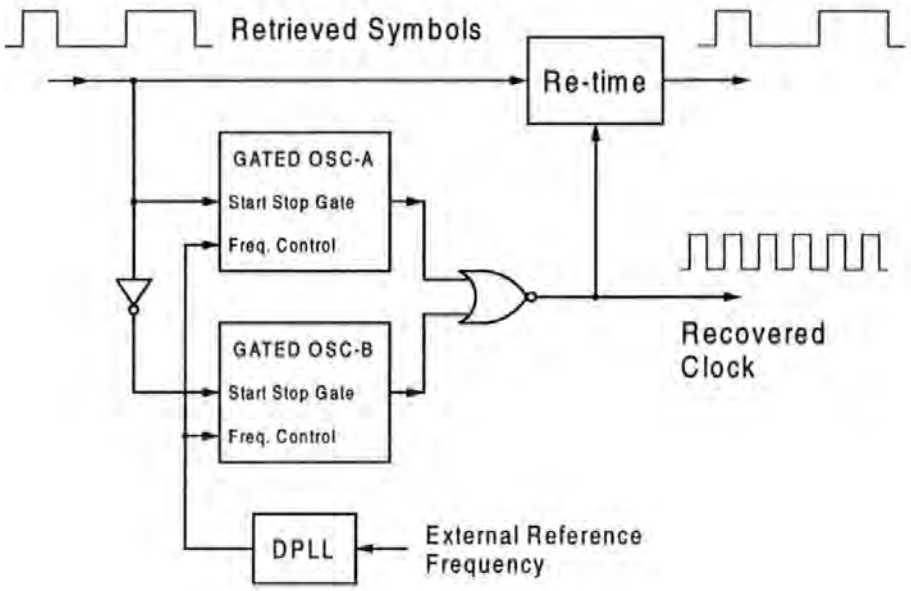


Figure 1-8 Gated oscillator clock recovery

A DPLL is locked to an external local reference oscillator, providing a stable frequency to the gated oscillators. Two matched, gated square-wave oscillators are started and stopped alternately, by the symbol transitions. The recovered clock is obtained by combining their outputs, by means of a logic gate. In a closed-loop system, accumulated phase errors between the retrieved symbols and the local oscillator are re-circulated around the loop. Whereas, this open-loop implementation discards timing errors every time the oscillators are stopped. The phase of the recovered clock is therefore reset to that of the retrieved symbols, upon every symbol transition.

The problem with this algorithm is that there is no noise immunity, whatsoever. The action of resetting the phase of the gated oscillators, to each symbol transition, causes the recovered clock to inherit all of the symbol jitter.

This renders the recovered clock unsuitable for synchronous post processing, necessitating the use of a flexible memory interface, such as a FIFO (First-In-First-Out) buffer [65]. Also, during the data burst, it is necessary to ensure that the FIFO buffer neither becomes empty nor full, otherwise data errors will occur. This may be achieved by the use of asynchronous handshakes, normally made available within FIFO memory devices.

Although this gated oscillator algorithm provides instantaneous clock acquisition, it is only suitable for asynchronous communications (unless the retrieved symbols are free from timing jitter, which is rarely the case).

1.1.5. Maximum Likelihood

The theory of estimating epoch, of a received signal plus noise, can be approached using the theory of maximum *a posteriori* estimation of an unknown signal in Gaussian noise [66] [67]. In the context of clock recovery, there is optimum clock timing, known as the ML (Maximum Likelihood) estimate or MAP (maximum *a posteriori*) estimate [68].

The theory assumes that the baseband pulse shape is known and is confined to the bit period *(T)*, thereby allowing a stored replica to be generated at the receiver. The history of the received signal is observed over a finite number of symbols (K), during which, the timing misalignment between the retrieved symbols and the locally generated clock is assumed constant, but unknown. Based upon this information, an estimate of the unknown optimum timing can be found.

Theoretical analysis of MAP-estimation is referenced in [68].

The MAP estimate, $\hat{\epsilon}$, of the random epoch ϵ, is the value that maximises:

$$\Lambda(y,\epsilon) = \ln f(y,\epsilon) = \sum_{k=0}^{K} \ln\cosh\left[\frac{2}{N_o}\int_{T_k(\hat{\epsilon})} y(t)p_s[t-(k-1)T-\epsilon]dt\right] \quad \ldots\ldots\ldots \quad (1\text{-}1)$$

An interpretation of equation (1-1) in the time domain, is to cross-correlate the received signal ($y(t)$), with a stored replica ($p_s(t)$), in each sub-interval (k), take the log hyperbolic cosine of this result, and accumulate these values over all sub-intervals (K). The epoch value (ϵ) that yields the largest accumulated value, is then

declared the best timing estimate ($\hat{\epsilon}$). A parallel approach to the MAP-estimator is not practical, due to the large number of "integrate-and-dump" circuits, required for each iteration of k (where k is large to minimise phase quantization). Figure 1-9 shows a serial realisation, where a single "integrate–and-dump" circuit is employed.

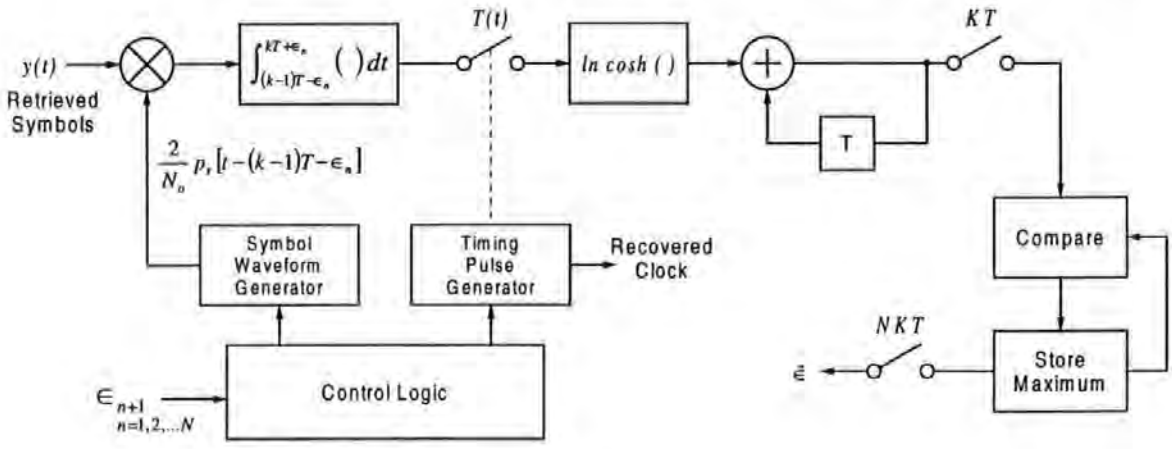


Figure 1-9 Open-loop MAP-Estimate clock synchroniser

During each successive interval, a different local symbol-timing phase is selected, and the corresponding epoch estimated. The current sub-interval epoch is compared with the largest past epoch (within the interval *NKT*), and the greater of the two is stored, together with its timing phase information. After all sub-intervals have been processed, the currently stored epoch is declared the MAP-estimate, and the corresponding timing phase is used as the reference for the recovered clock. Compared to a parallel approach, this serial implementation increases clock acquisition time by a factor of N, since all sub-intervals must be processed to arrive at the MAP-estimate. Also, the block processing involved renders this open loop approach unsuitable for tracking short-term symbol rate drift.

Stiffler [69] and Mengali [70] have devised a closed loop tracking synchroniser, that converges towards the MAP-estimate. Megali's tracker is shown in Figure 1-10

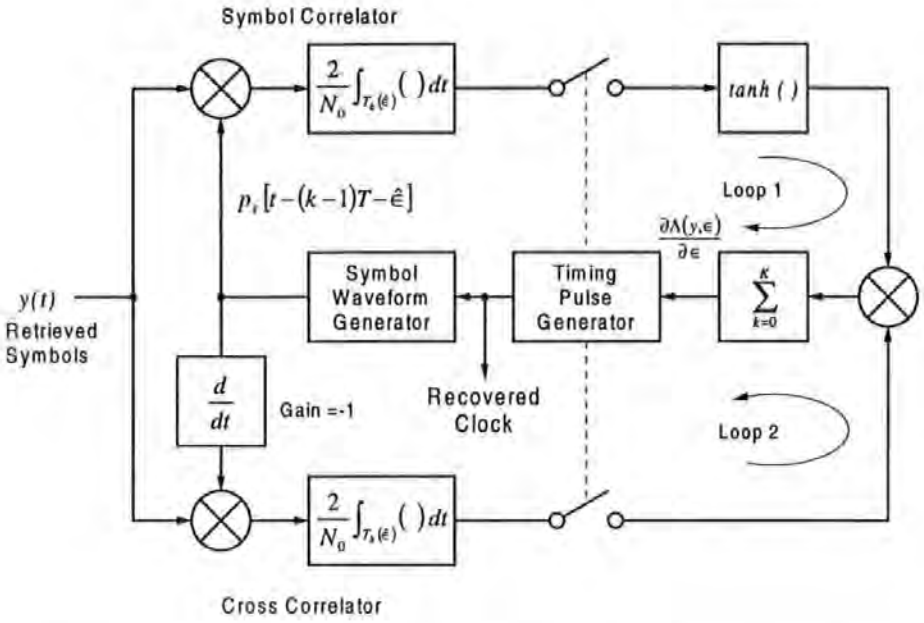


Figure 1-10 Closed-loop MAP-estimate clock synchroniser

An alternative interpretation of equation (1-1), is achieved by differentiating with respect to ϵ and equating to zero, yielding:

$$\left.\frac{\partial\Lambda(y,\epsilon)}{\partial\epsilon}\right|_{\epsilon=\hat{\epsilon}} = \sum_{k=0}^{K}\left[\frac{2}{N_o}\left\{\int_{T_k(\hat{\epsilon})} y(t)\frac{\partial\, p_s[t-(k-1)T-\hat{\epsilon}]}{\partial\hat{\epsilon}}dt\right.\right.$$

$$+ y(kT+\hat{\epsilon})p_s(T) - y[(k-1)T+\hat{\epsilon}]p_s(0)\}]$$

$$\times\tanh\left\{\frac{2}{N_o}\int_{T_k(\hat{\epsilon})} y(t)p_s[t-(k-1)T-\hat{\epsilon}]dt\right\} \qquad \ldots\ldots \quad (1\text{-}2)$$

$$= 0$$

For an estimate of ϵ other than the MAP-estimate, the function $\partial\Lambda(y,\epsilon)/\partial\epsilon$ will be either positive or negative, depending on whether ($\epsilon < \hat{\epsilon}$) or ($\epsilon > \hat{\epsilon}$). Hence, $\partial\Lambda(y,\epsilon)/\partial\epsilon$ can be used to indicate the search direction.

The phase of the timing pulse generator, which controls the "integrate-and-dump" circuits, is incremented every symbol period, by an amount proportional to the magnitude of $\partial\Lambda(y,\epsilon)/\partial\epsilon$, and in a direction based upon the sign of $\partial\Lambda(y,\epsilon)/\partial\epsilon$, as computed from the previous KT interval.

The closed loop dynamics produce an estimation of the frequency error, which is used to control the VCO. This is different from the open loop approach shown in Figure 1-9, which estimates the optimum timing (not the error).

By definition, MAP-estimators operate upon historical information, with little regard given to the problem of rapid acquisition. Synchronisers based upon MAP-estimation operate at low signal to noise ratios and are often used in coherent modems for deep space exploration [71].

1.1.6. Early-late synchroniser

An Early-Late gate synchroniser, as shown in Figure 1-11, is popular for rectangular pulses. The index k is a sequence of symbol pulses, of period T, while τ represents the timing error.

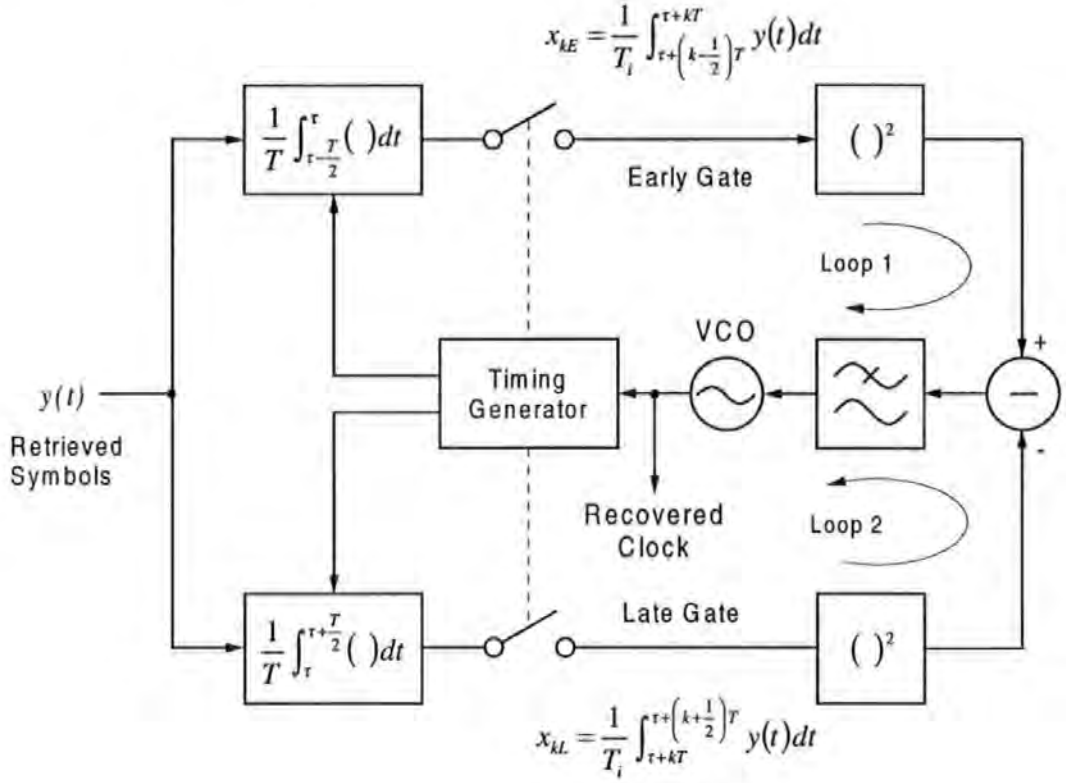


Figure 1-11 Early-late gate synchroniser

The circuit consists of a pair of gated integrators, each performing its integration over a time interval of *T/2*. Integration by the early gate occurs in the *T/2* interval preceding the nominal location of data transitions, while the late gate integrates during the *T/2* interval following the transition. Gate intervals adjoin but do not overlap. If the timing error is zero, then the data transitions fall exactly on the boundary between early and late gates, as shown in Figure 1-12.

If the timing error is not zero, a transition falls not on the boundary but within one or the other of the gates. Since signal polarity changes within the gate containing a transition, the associated integration reaches a lesser magnitude than when the transition is external to the gate. Comparison of the magnitudes of the two integrators therefore gives an indication of the timing error.

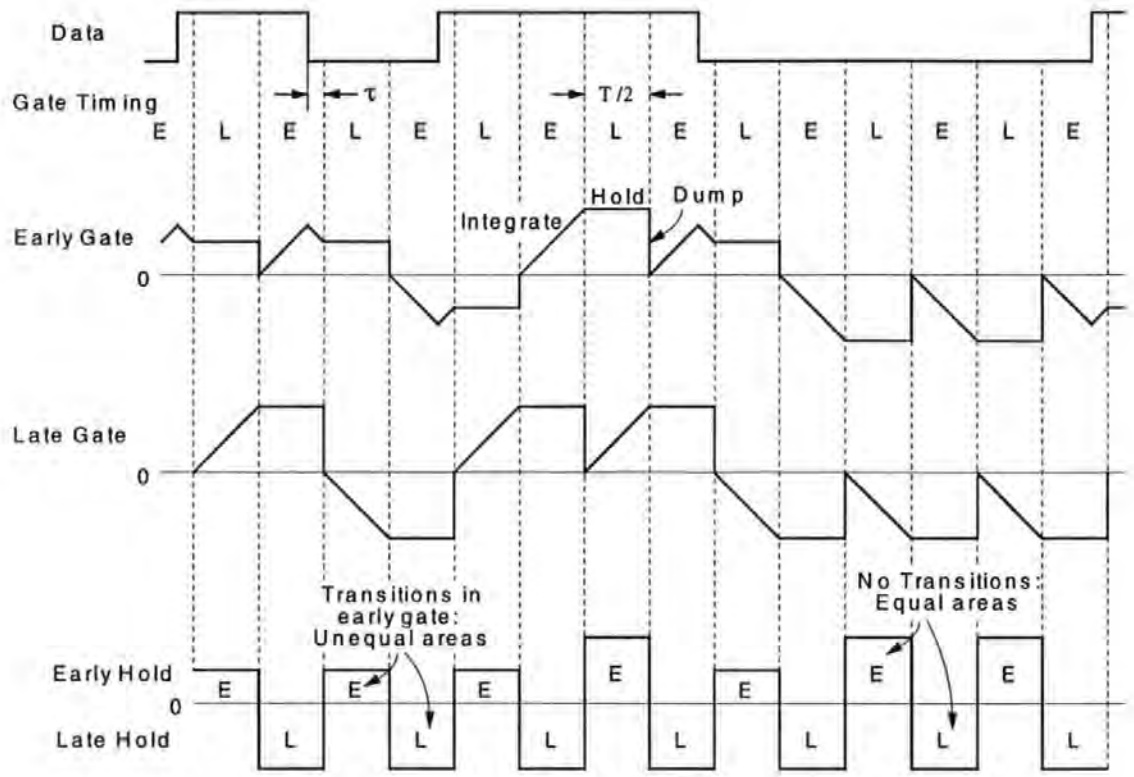


Figure 1-12 Timing diagram for early-late synchroniser

Useful gate output is the integration value that has been accumulated after *T/2* sec; so at this point the magnitudes are compared and dumped and the cycle repeats. The resulting error voltage controls the frequency of the VCO, which in turn adjust the sampling point at the gates, so as to minimise the phase error.

In the absence of transitions, no timing information is present and the voltage from both loops are zero. If there is a long run of data without transitions, no error information is supplied to the loop and the phase of the VCO slowly drifts, eventually resulting in cycle slips of the recovered clock.

In a practical system, the held samples are compared sequentially (as seen from the waveforms in Figure 1-12), not simultaneously, which results in simplified circuitry. A more detailed analysis of the Early-Late gate synchroniser can be found in [72] [73].

The Early-Late gates acts to place the boundary between the two gates exactly at the transition instant and departure from this timing generates the loop error. It is also possible to produce an error signal by using a single gate system that straddles the transition. Lindsey and Tausworthe [74] have invented such a scheme, known as the "Digital Data-Transition Tracking Loop" (DTTL). If the transition is exactly centred within this mid-phase gate, integration over the gate interval is zero. If however the transition is not centred, then the integration produces either a positive or negative error output. Sense of the error is determined according to the direction of the transition. In the absence of transitions, no information is presented and the integrator output is ignored.

As for the MAP-Estimator, the early-late synchroniser operates at low signal-to-noise ratios and has been used for deep space exploration [75].

1.1.7. Summary of clock recovery architectures

Clock recovery algorithms can be classified into open or closed loop and wide or narrow bandwidth. Table 1-1 summarises the described algorithms and compares their relative performance.

System	Class		Performance				Imple-mentation
	Open/ Closed Loop	Band-width	Tracking Range	Jitter Reject	Clock Acqis-ition	Clock Fly-Wheel	
High-Q Resonator	Open	Narrow	Poor	Good	Good	Good	Simple
PLL (wide loop filter)	Closed	Wide	Good	Poor	Moderate	Poor	Moderate
PLL (narrow loop filter)	Closed	Narrow	Poor	Good	Poor	Good	Complex
PLL (frequency detection)	Closed	Narrow	Moderate	Good	Moderate	Good	Complex
PLL (DLL phase selection)	Open & Closed	Narrow	Poor	Good	Good	Good	Complex
Quadricorrelator	Closed	Wide & Narrow	Good	Good	Moderate	Good	Complex
Gated Oscillator	Open	Wide	Good	None	Good	Good	Simple
MAP-Estimator	Open	Wide	Moderate	Good	Poor	Poor	Complex
MAP-Estimator Tracking Synchroniser	Closed	Wide	Moderate	Good	Poor	Poor	Complex
Early-Late Gate Synchroniser	Closed	Wide	Moderate	Good	Poor	Poor	Complex

Table 1-1 Summary of clock recovery algorithms

In general, open-loop algorithms demonstrate rapid clock acquisition and long clock flywheel periods. The exception to this case is the MAP-estimator, which by definition, operates upon historical information (i.e. previously received symbols). Open-loop algorithms also have the advantage that jitters due to noise, are flushed-out of the system and are not regenerated around a closed loop.

For systems that exhibit an accurate and stable symbol frequency, the high-Q resonator provides good overall performance, producing rapid clock acquisition, long flywheel periods and good jitter rejection. Unfortunately, practical considerations make these systems expensive to implement.

The open-loop gated oscillator is simple to implement, provides instantaneous acquisition and exhibits good flywheel performance. However, this algorithm must be rejected for synchronous operation, because the recovered clock inherits all of the symbol jitter. This is unacceptable for a synchronous system, since clock jitter causes the symbols to be incorrectly sampled, resulting in data errors.

A closed-loop algorithm is usually employed, for systems that exhibit symbol rate drift. In order that phase and frequency variations may be tracked, many closed-loop algorithms are based upon the basic PLL.

Appendix 2 concludes with a summary of PLL characteristics, and points out some of the difficulties and undesirable features. These include: instability of a feedback system, conflicting loop parameters, phase variations of the recovered clock relative to the symbols, variable start-up conditions and the need for a long synchronising pre-amble. Techniques such as zero phase start and variable loop time constants may be employed to aid acquisition [55], but these parameters render the loop complex to design and implement.

A PLL with a narrow-loop filter seriously limits the capture range, which is usually less than 10 times its closed-loop bandwidth [62]. For similar noise rejection performance to that of a high-Q resonator, a loop bandwidth of 0.1% is required, which provides a capture range of less than ±1%. Also, a training sequence of several hundred symbols are required, in order to achieve lock.

Hence, the basic PLL is often modified with frequency detection to extend the capture range and improve acquisition. The most successful of these algorithms is the Quadricorrelator, which employs a wide bandwidth FDD to aid acquisition and a narrow bandwidth PLL once phase-lock is achieved. This improves the capture range to ±5% and reduces the training sequence to approximately 30 phase-reversal symbols. However, the Quadricorrelator requires the use of an expensive quadrature VCO, and stable designs are difficult to implement.

Other PLL clock synchronisers use a combination of open and closed loops. A DLL is used to provide a selection of clock phases, the optimum of which is chosen to re-time the symbols. The DLL can only accept a periodic input and therefore cannot be locked to the input transitions. Hence, the DLL is locked to a local reference frequency. A wide-band PLL takes a snapshot of the symbol timing, which is used to select the optimum clock phase from the DLL. The recovered clock is then taken from the DLL, which operates in an open-loop configuration, relative to the retrieved symbols. The wide-band PLL achieves rapid clock acquisition and the

changeover to an open-loop configuration provides good noise immunity and clock flywheel performance. However, if the data burst is long, drift between the local reference clock and the retrieved symbols, eventually results in cycle slips.

The closed-loop tracking MAP-estimator and the Early-Late gate synchroniser, operate at low signal to noise ratios, providing optimum clock timing for re-sampling the data. However, these synchronisers are intended for use in coherent communications modems and do not address the problem of rapid acquisition. Hence, clock acquisition is poor, requiring several hundred symbols in order to achieve lock. Also, the MAP-estimator is constrained to a strictly confined pulse shape and is therefore unsuitable for rectangular pulses.

The summary of clock recovery architectures, tabulated in Table 1-1, demonstrates the various performance trade-offs. There is no single algorithm, which satisfies all the requirements of an ideal clock recovery system. The high-Q passive resonator provides good overall performance, with the exception of tracking symbol rate drift. It is possible to implement a high-Q "ringing" filter using DSP techniques, which eliminates many of the practical problems associated with the implementation of a passive resonator. The use of DSP also allows system parameters to be re-configured "on-the-fly", and this may be exploited to provide an adaptive high-Q tracking filter. Hence, it is the use of DSP to implement a high-Q "ringing" filter, which forms the basis of the new algorithms described in subsequent chapters.

1.2. Channel coding

In communications systems the channel code is usually called the transmission or line code, while in magnetic recording it is more commonly known as the modulation or recording code. The time interval between signal transitions significantly affects the ability of a clock recovery system to achieve and maintain synchronisation. Therefore, a channel-coding scheme is in part chosen to provide frequent transitions, so as to aid clock recovery. However, this is often achieved at the expense of channel bandwidth efficiency [76].

In communication systems, binary data are commonly transmitted in the NRZ_L (Non Return to Zero Level) format. In this format, each bit is of duration T_b and has equal probability of being a one or a zero. The bit rate is $1/T_b$ and is measured in bits/second. Binary data can also be transmitted in the RZ (Return to Zero) format, in which case the signal returns to a zero between consecutive bits. Manchester data are encoded and the corresponding symbol rate is $2/T_b$, measured in symbols/second. The Manchester code is described more fully in Sub-Section 1.2.2.2. The symbols of the Manchester code have guaranteed transitions at every bit period and is often referred to as a "self-clocking" code, since it has a high spectral component at the clock rate. However, this is achieved at the expense of poor channel efficiency and a reduced jitter margin or timing window. Therefore, RZ data are only suitable for low noise systems where bandwidth is plentiful.

NRZ data on the other hand has optimum efficiency and jitter margin, but has two properties that make the task of clock recovery difficult.

The first problem is that the spectrum of NRZ data has nulls at multiples of the bit rate and therefore has no spectral component for direct input to the clock recovery system. The PSD (Power Spectra Density) for NRZ data is shown in Figure 1-13, where $f = 1$ corresponds to the bit rate.

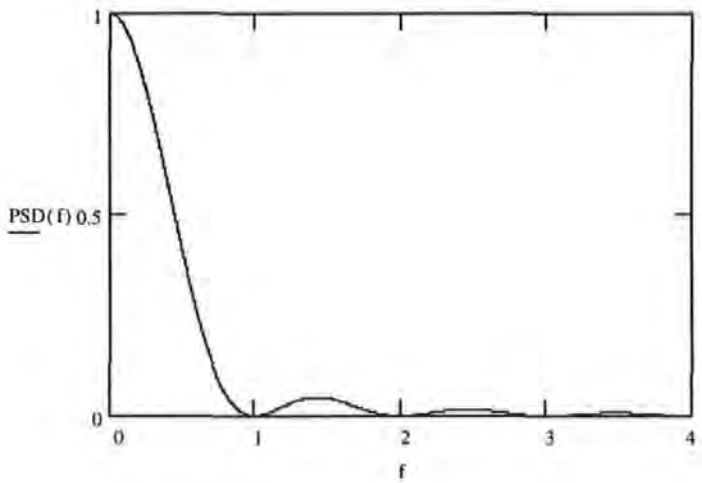


Figure 1-13 PSD of NRZ data

The PSD for NRZ data is given by:

$$PSD(f) = T\left[\left|\frac{\sin\left(\frac{\omega T}{2}\right)}{\frac{\omega T}{2}}\right|\right]^2 \qquad \ldots\ldots\ldots\ldots\ldots\ldots \quad (1\text{-}3)$$

Due to the lack of spectral component at the bit rate in the NRZ format, a clock recovery circuit will fail to lock to the required frequency. NRZ data must therefore undergo a non-linear process to extract, from the retrieved transitions, a frequency component at the bit rate.

As already described in Section 1.1 (Clock recovery architectures), a strong spectral component at the symbol rate may be achieved by means of edge detection. This topic is discussed further in Chapter 2, using auto-correlation to derive a strong clock frequency component. The second problem is that the data may contain long sequences of ones and zeros and hence, in the absence of transitions, the clock recovery system may lose synchronisation. This can be solved with the use of a data scrambler to break up long sequences of ones and zeros.

1.2.1. Scrambled NRZ

A data scrambler can be used to convert NRZ data into a communication channel code, in a non-redundant fashion, by scrambling the data with a PRBS (Pseudo Random Binary Sequence) utilising an *M*-sequence polynomial. This process results in 100% efficiency with a jitter margin equal to the bit period. For data retrieval, the same PRBS must be used for de-scrambling, to restore the original data [29].

For *M*-sequences there are always 2^{n-1} ones and 2^{n-1}-1 zeros, implying nearly equal probability for large *n*. When *n* is large ($n>10$), the auto-correlation function of *M*-sequences closely approximates to that of white noise, and it is the highly random nature of these sequences which makes them ideal as data scramblers.
Shown in Figure 1-14 is a standard "CCITT V22.bis" data scrambler and de-scrambler arrangement [77], which can be easily implemented in logic hardware.

The modulo-2 adders are "Exclusive-Or" gates and the delay-line is implemented as a SIPO (Serial-In Parallel-Out) shift-register.

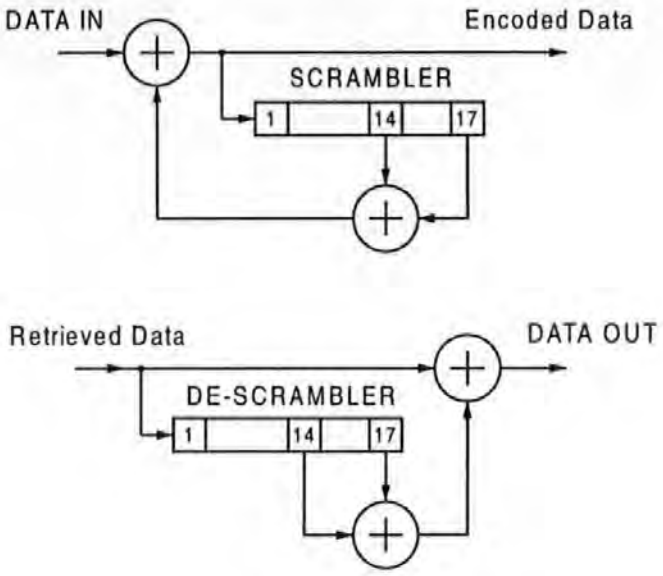


Figure 1-14 Data scrambler/de-scrambler

The shift-register taps in Figure 1-14 corresponds to a polynomial, defined as $x^{17}+x^{14}+1$. A single channel error will generate an immediate error and two subsequent errors, as the error propagates through the shift-register. This corresponds to an error extension factor of 3, and in general the error extension factor is the number of shift-register taps.

In order to implement a particular scrambler system, it is necessary to select a shift-register of length n and a suitable primitive polynomial (has no factors except 1 and itself, i.e. no sub-sequences repeat within the main sequence). A list of irreducible polynomials is provided in Table 1-2. The optimum polynomial for a specified value of n is often that with the minimum number of feedback taps, since any

channel error will then cause minimum error extension in the de-scrambled sequence.

n	Polynomial
1	$1+x$
2	$1+x+x^2$
3	$1+x+x^3$
4	$1+x+x^4$
5	$1+x^2+x^5$
6	$1+x+x^6$
7	$1+x+x^7$
8	$1+x+x^2+x^7+x^8$
9	$1+x^4+x^9$
10	$1+x^3+x^{10}$
11	$1+x^2+x^{11}$
12	$1+x+x^4+x^6+x^{12}$
13	$1+x+x^3+x^4+x^{13}$
14	$1+x+x^6+x^{10}+x^{14}$
15	$1+x+x^{15}$
16	$1+x+x^3+x^{12}+x^{16}$
17	$1+x^3+x^{17}$

Table 1-2 Sequence of irreducible polynomials

If by chance the input data are identical to the PRBS at the encoder feedback, then the output from the "Exclusive-Or" gate will remain at a logic zero and the system breaks down. However, a matching sequence is likely to be only short-term and the probability of such an occurrence is low [1]. Nevertheless, even a short-term loss of symbol transitions could adversely affect clock recovery performance.

1.2.2. Recording codes

Communication systems usually employ multilevel signals and hence channel efficiency can be improved by the use of complex modulation schemes. These include amplitude, frequency and phase modulation or a combination of all three [78]. In general, digital magnetic recording operates with just two levels ±1. This is because the magnetic particles of the medium are saturated in one of two directions [79] [80].

Channel capacity may however, be increased by the use of RLL (Run Length Limited) coding [81] [82]. If data are encoded to contain fewer transitions, then these transitions can be recorded closer together for a given packing density, resulting in increased storage capacity. However, this can affect the self-clocking properties of the coded data, which could adversely affect clock synchronisation.

Transmission of dc should be avoided, since dc wander at playback can render peak (threshold) detection unreliable. Also, if the dc-content of a code is allowed to grow indefinitely, then eventually the magnetic record heads becomes saturated and no magnetic flux is induced at playback [83].

To minimise signal distortion, the recording code should ideally match the spectrum of the recording channel [84]. Shown in Figure 1-15 is the response of a playback head, which increases from dc at 6dB per octave, until tape thickness and spacing

losses causes a fall. The construction of the playback head consists of two poles and a small delay produced by a finite head gap. This mechanism gives the action of a two-pole transversal filter, producing a comb filter response, with nulls where flux cancellation takes place across the gap [85]. Other factors affecting the channel response are the pulse shape [86] and the playback equalisation [87]. Hence, the selected recording code should ideally match the resulting channel response.

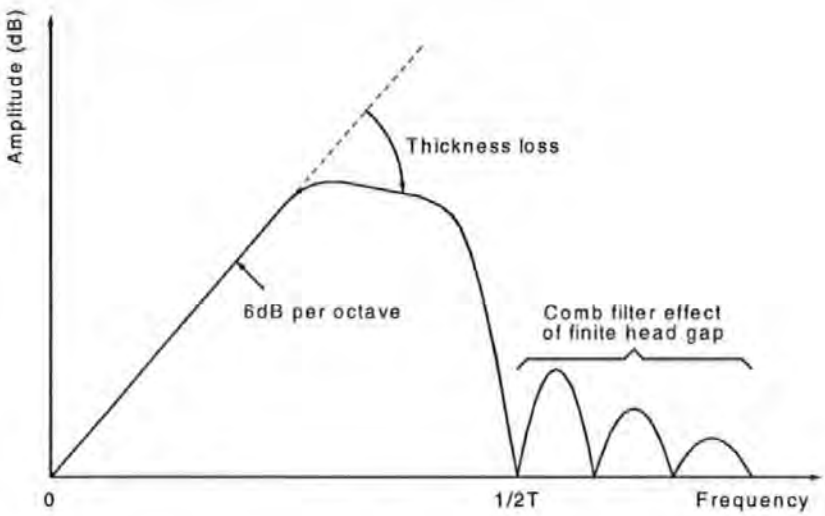


Figure 1-15 Frequency response of a recording channel

NRZ data contains a large dc frequency component, due to long sequences of ones and zeros. Scrambled NRZ data reduces the dc-content, but does not ideally match the band-pass frequency response of a magnetic recording channel, resulting in playback pulse distortion and ISI (Inter Symbol Interference) [88]. Following convention, ones and zeros in the coded sequence are recorded on the magnetic medium as the presence and absence of magnetic polarity transitions. The magnetic recording equivalent of Scrambled NRZ then becomes "Randomised NRZI", which is described in Section 1.2.2.1.

A suitable RLL recording code combines the clock with the data prior to recording in such a way so as to reduce the dc-content and provide optimum separation of adjacent transitions. The number of symbol periods between transitions is known as the "run-length". An increase in the run-length enables channel data to be recorded at a higher rate onto the magnetic medium, thereby increasing the packing density. However, this improvement in storage efficiency is achieved at the expense of reduced clock content.

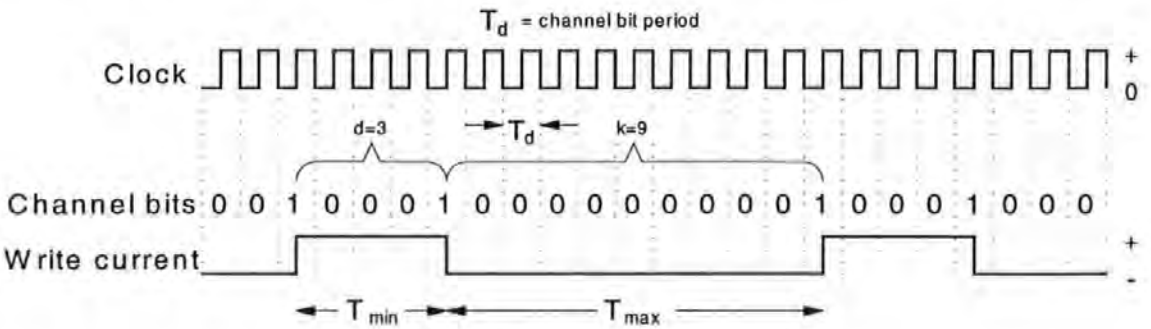


Figure 1-16 Channel bit convention

Figure 1-16 illustrates the channel bit convention. The shortest run-length of zeros between consecutive ones is known as the "*d*-constraint" and determines the highest transition density. The longest run of zeros between consecutive ones is known as the "*k*-constraint" and determines the clocking properties of the code. In other words, the *d*-constraint defines the high frequency content of the code, while the *k*-constraint defines the low frequency content. The *d*-constraint is chosen to maximise the recording density, whilst minimising ISI, and the *k*-constraint is chosen to provide adequate transitions for clock recovery purposes.

The DSV (Digital Sum Variation) is the running integral of the area beneath the coded sequence. If the DSV of the code is allowed to grow indefinitely, then the code has dc-content, which could ultimately result in saturation of the magnetic record heads.

When m data symbols are encoded into n code symbols the code rate is:

$$Code\ Rate = \frac{m}{n} \qquad \text{..... (1-4)}$$

A convenient measure of code efficiency is the Density Ratio (DR) [16]:

$$T_{\min} = DR = \frac{m}{n} \cdot (d+1) \qquad \text{..... (1-5)}$$

And the capacity of a code is given by:

$$C = \log_2 x_{\max} \qquad \text{..... (1-6)}$$

where x_{max} is the largest root of the equation:

$$x^{k+2} - x^{k+1} - x^{k+1-d} + 1 = 0$$

Choosing a high d-constraint could increase the density ratio but will impair the jitter margin or timing window (T_w), making the system more sensitive to noise.

More commonly, the efficiency of a code is given by:

$$Efficiency = \frac{Code\ Rate}{Capacity} \qquad \text{..... (1-7)}$$

Figure 1-16 also shows that the maximum distance between transitions (T_{max}) influences the lower spectral limit.

$$T_{\max} = \frac{(k+1)}{n} \cdot m \qquad \text{..... (1-8)}$$

1.2.2.1. NRZI coding

Many coding schemes result in the propagation of a single bit error into several errors. For example, consider the basic longitudinal recording system, shown in Figure 1-17.

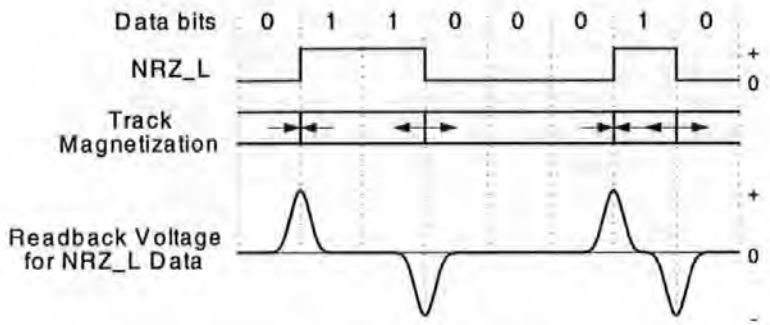


Figure 1-17 Saturated magnetic recording

A polarity reversal in the write current only occurs if the present bit is not the same as the previous bit. A reversal generates a change in the direction of saturation and upon playback, this transition generates a voltage pulse in the read head. Since a transition represents a switch from a run of ones to zeros (or vice versa), then failure to detect the playback pulse will give indefinite error propagation.

Fortunately, this problem is easily resolved by means of NRZI encoding, where the difference between adjacent bits is detected and a transition is recorded only for a binary one, but not for zeros. Hence failure to detect a transition gives only a single bit error, which can be later corrected. However, it should be noted that adjacent transitions could result in ISI (Inter Symbol Interference), which causes peak shift degrading performance, particularly at high-density ratios.

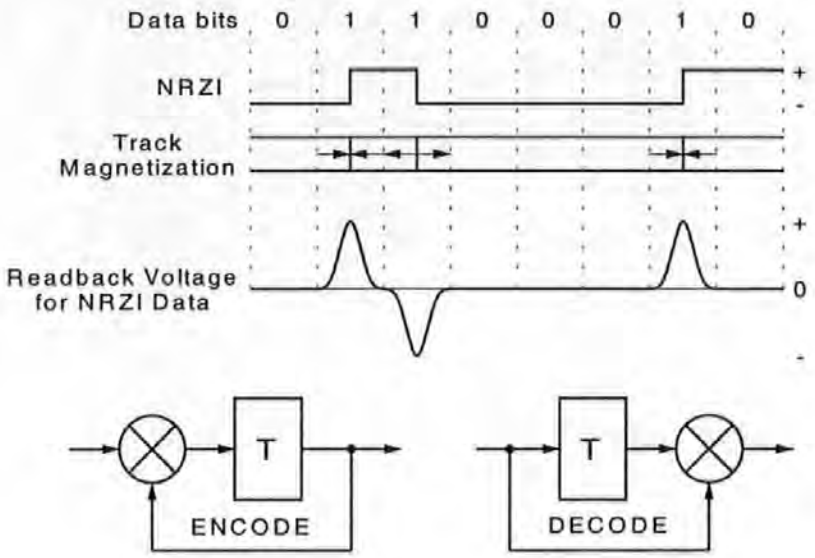


Figure 1-18 NRZI or differential coding

NRZI decoding shown in Figure 1-18, is also useful for eliminating phase ambiguity of the recovered data, which is desirable in recording systems, because the data at playback have a 50% chance of being inverted. In practice NRZI coding is pre-coded with a RLL code.

Scrambled NRZI coding has been used in the D-1 digital video recorder [89] and the Ampex DCRSi instrumentation recorder [90]. The PRBS reduces the dc-content of the code, which is essential in these recorders, since rotary transformers are used.

NRZI is also used in communications systems, but is known as "Differential Coding". Satellite communication systems often employ DBPSK (Differential Binary Phase Shift Keying) modulation [23]. Differential decoding provides a simple and robust method of demodulating a DBPSK signal, as well as eliminating phase ambiguity of the demodulated data.

1.2.2.2. Manchester code

The Manchester code (also known as BiPhase_L) was the first practical self-clocking binary code and is dc free. The Manchester code is very easy to encode since the NRZ data are simply gated with the record clock by means of an "Exclusive-Or" gate.

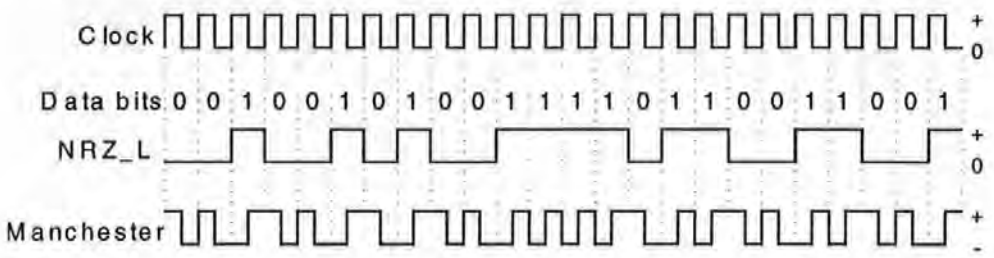


Figure 1-19 Manchester coding

The timing diagram in Figure 1-19 shows that each data bit is represented by two channel symbols. The code always changes state in the centre of the bit period. The first half of the bit period is encoded as the complement of the data, whilst the second half is the actual data. Since there can be two transitions over one bit period, the jitter margin can only be ½ a bit. The high clock content of the Manchester code is achieved at the expense of bandwidth and is therefore only used where recording density is not critical.

The Manchester code remains in use today, for example single-density floppy disk [91], SMPTE /EBU time code [92] [93] used on professional audio and video recorders and the reference track in the DASH format [94]. It is also commonly used for communications systems such as the European RDS [95], low-bit rate weather satellites [96] and local area networks [97].

1.2.2.3. Miller code

The Miller Code (also known as Modified Frequency Modulation, MFM and Delay Modulation, DM) has a reduced clock content compared to that of the Manchester code, hence the need for a clock flywheel effect over missing transitions in the clock recovery system. Transitions in the centre of the bit period are retained, but transitions at the end of the bit period are only required between successive zeros. There are still two channel bits for every data bit providing a code rate of ½, but adjacent channel bits will never be a one, doubling the minimum time between transitions and hence doubling the density ratio. The proceeding bit influences the coding of the current bit, hence, memory must be provided to store the previous bit. Figure 1-20 shows the DSV of the Miller code which suffers from baseline (or dc) wander. This can lead to peak shift at threshold detection resulting in timing errors.

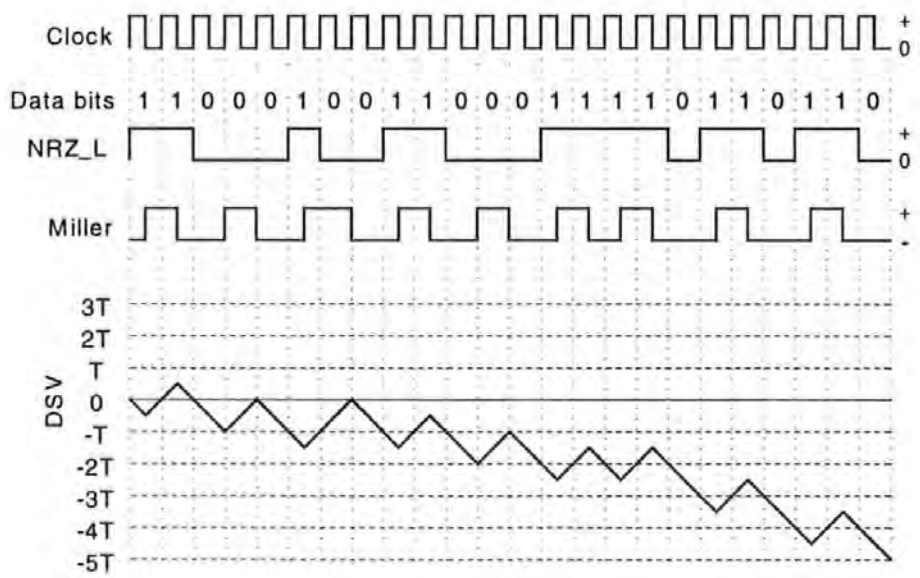


Figure 1-20 Miller coding

Many rigid disks adopted the Miller code at the time of their development and it remains in use on double density floppy disk [91].

Shown in Appendix 3 is a circuit diagram of the Miller encoder, developed for real-time testing.

1.2.2.4. Miller^2 code

The Miller^2 code is a direct modification of the Miller code, to eliminate the dc-content. The timing diagram of Figure 1-21 shows that the DSV of the Miller^2 code is bounded, preventing the dc level from growing.

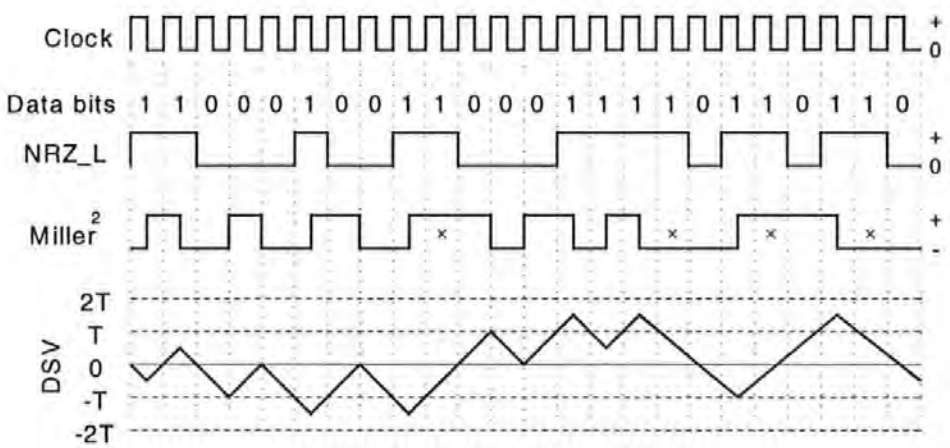


Figure 1-21 Miller^2 coding

The coding involves a new generalised rule: whenever an even number of ones occurs between zeros, the last one transition is suppressed. Simulation results have revealed that this generalised rule is open to misinterpretation, particularly if a random snap shot of a coded sequence is analysed. The coded sequence is bounded into three sequence bands, where one complete band ends, before the next band begins. Reference to the Miller^2 code in Figure 1-23 reveals two sequence bands of

type-C, followed by a sequence band of type-B and A. The following two data one bits are of a new sequence band of type-A, therefore, the last one is not suppressed, even though a snap shot of the data bits suggest that the new coding rule should be applied. If this principle of sequence bands is not applied, then the code will not be dc free. Many books overlook this important fact and care should be taken when designing a suitable encoder.

The sequence coding rules as specified in the original patent [34] are as follows:

Type-A	Any number of ones but no zeros = Change at the centre of the bit period.
Type-B	An odd number of ones = Change at the centre of the bit period. Two zeros = Change at the end of the bit period.
Type-C	A zero followed by an even number of ones = Don't encode the last bit.

Simulation results have revealed an error in the original patent and this is described in Appendix 3. The Miller^2 code was used in some early stationary-head digital audio recorders and is currently in use in high-bit rate, instrumentation recording and the professional D-2 composite digital videocassette recorder [98].
Shown in Appendix 3 is a circuit diagram of the Miller^2 encoder, developed for real-time testing. For simulation of the DSP-based clock recovery schemes, a new custom Miller^2 encoder has been developed, and this is shown in Appendix 3.

1.2.2.5. RLL 2,7 code

The RLL 2,7 code is a member of the more advanced group codes, where input data are converted to patterns of channel bits by means of a lookup table. The lookup code avoids undesirable patterns, thereby improving overall efficiency. The main advantage of the 2,7 code is that the *d*-constraint is increased by one, compared to the Miller and Miller[2]. Hence, there is an increase in the density ratio for the same jitter margin. There are two encoded bits corresponding to each data bit. The mapping of the variable-length code words is given in Table 1-3.

Data	Code Word
10	0100
11	1000
000	000100
010	100100
011	001000
0010	00100100
0011	00001000

Table 1-3 Code table for RLL 2,7

The encoding of the data can be achieved by partitioning the data sequences into two, three or four-bit groups to match the entries in the code table, and then mapping them into the corresponding code words, as indicated in Figure 1-22.

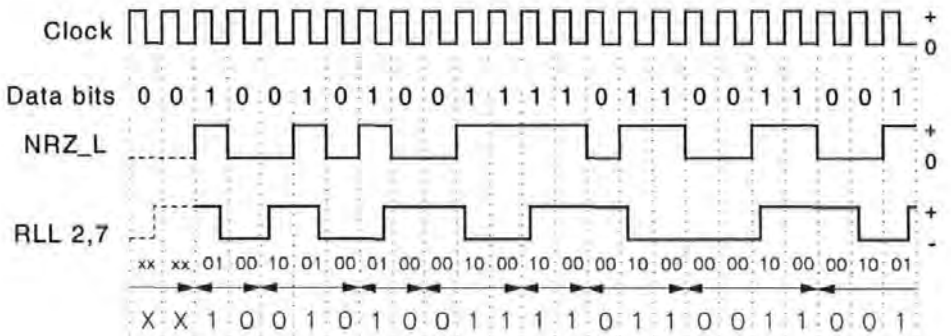


Figure 1-22 RLL 2,7 coding

The RLL 2,7 code is used in high performance rigid disks such as the IBM 3370, 3375 and 3380 [99].

Shown in Appendix 3 is a circuit diagram of the RLL2,7 encoder, developed for real-time testing. For simulation of the DSP-based clock recovery schemes, a new custom RLL2,7 encoder has been developed, and this is shown in Appendix 3.

1.2.2.6. Summary of recording codes

Kobayashi and Mackintosh have provided surveys of recording codes and these publications are referenced in [100] and [101].

Research into recording codes is active and new codes are invented regularly. The DSP-based clock recovery algorithms, discussed in later chapters, may be applied to any of the described recording codes. A clock recovery scheme, which is capable of fly wheeling over long encoded run lengths, could reduce the constraints currently imposed on recording codes, and this could ultimately lead to new more efficient codes, providing increased storage capacity.

Hence, rapid clock acquisition and long flywheel periods are primary objectives of the new DSP-based algorithms.

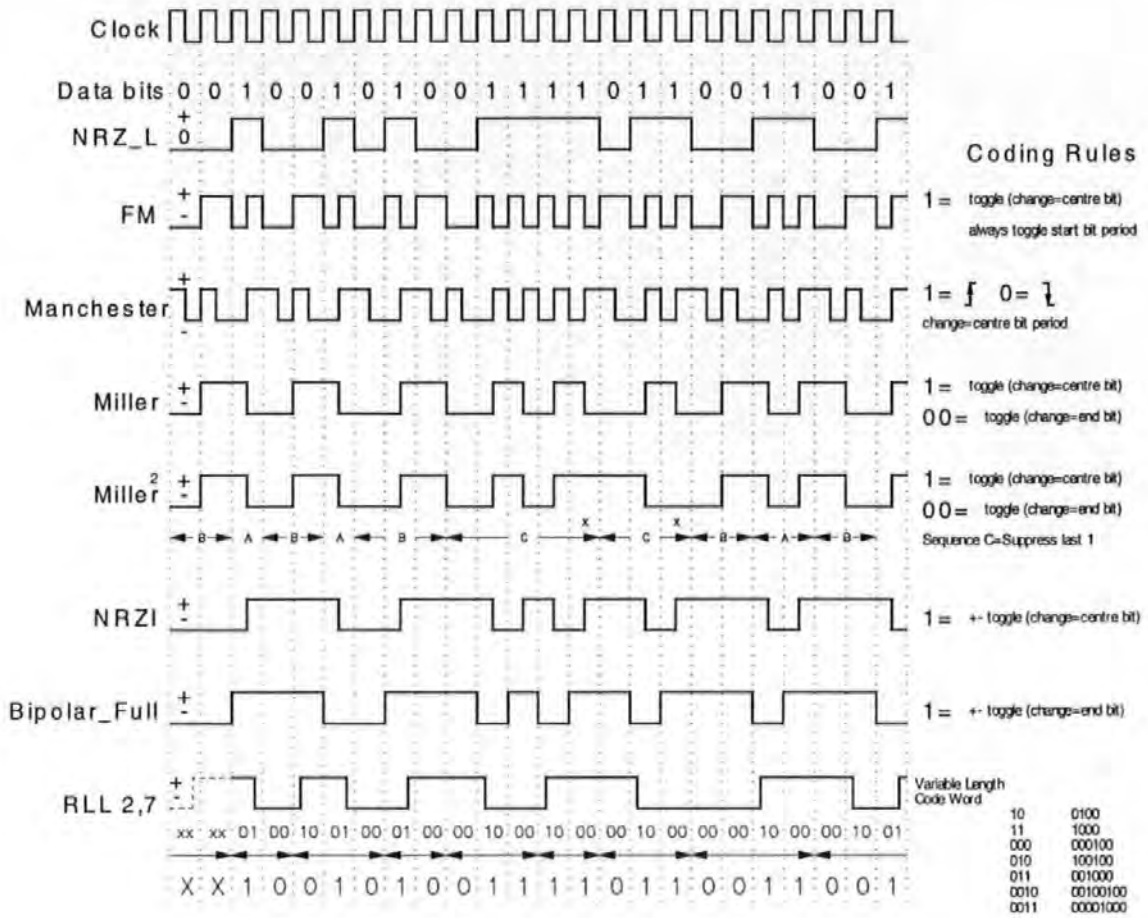


Figure 1-23 Common recording codes

Figure 1-23 shows the timing diagrams of common recording codes, together with the applied coding rules. Viewing the waveforms from top to bottom, one can observe an increasing minimum run length (*d*-constraint) and hence the potential for recording transitions closer together, thereby increasing the density ratio. Comparing the Manchester code with the RLL 2,7 code, one can clearly observe a difference in the coded run length. The Manchester code has a high number of transitions, providing a good clock component for clock recovery. While the RLL 2,7 code has a reduced clock component, but improved efficiency.

Tabulated in Table 1-4 is a collection of popular recording codes, summarising the relative performance.

Code Alternative Names	Rate	Capacity	Efficiency	d,k	Density Ratio	Detection Window	DC Free
NRZ-L	1	1	100%	0,∞	1	T	No
NRZ-M NRZI BiPolar_Half Randomized NRZ	1	1	100%	0,∞	1	T	No
BiPolar_Full	1	1	100%	0,∞	1	T	No
FM Biphase_M Double_Freq Phase_Encoding FSK	1/2	0.6942	72%	0,1	0.5	T/2	Yes
Manchester BiPhase_L	1/2	0.6942	72%	0,1	0.5	T/2	Yes
E-NRZ	7/8	1	87.5%	0,13	0.875	T	No
Miller MFM DM	1/2	0.5515	90.6%	1,3	1	T/2	No
Miller^2	1/2	0.6509	76.8%	1,5	1	T/2	Yes
GCR 4,5	4/5	0.8792	79.8%	0,2	0.8	0.8T	No
ZM	1/2	0.5515	90.6%	1,3	1	T/2	Yes
RLL 2,7	1/2	0.5174	96.6%	2,7	1.5	T/2	No

Table 1-4 Summary of popular recording codes

The Manchester, Miller, Miller^2 and RLL2,7 recording codes have been selected as test codes, for subsequent evaluation of the DSP-based clock recovery algorithms.

These recording codes have been implemented in a specially designed FPGA encoder, for real-time testing of clock recovery performance. The test generator hardware is described in Appendix 3, together with encoder simulation models and results.

Chapter 2

2. DSP-based clock recovery

This chapter describes a novel DSP-based, fixed frequency clock recovery algorithm, which exploits the "ringing" feature of a high-Q IIR (Infinite Impulse Response) filter [1] [2]. The scheme demonstrates superior performance over that of a PLL, particularly in terms of clock acquisition. A recovered clock is achieved at the first transition of the retrieved symbol. Hence, the synchronising preamble can be significantly reduced, resulting in faster access times and releasing synchronising space for data. Also, potentially long flywheel periods can be achieved, maintaining clock synchronisation over severe data dropouts. This can relax the constraints imposed on channel codes, which could lead to new, more efficient RLL recording codes, further improving storage capacity.

Simulation results are analysed for the new clock recovery scheme, to determine clock acquisition, flywheel performance and clock jitter in the presence of applied gaussian noise. These results are determined for each of the test recording codes (Manchester, Miller, Miller^2 and the 2,7 code).

The simulation results have been verified in real-time using a TMS320C25 fixed point digital signal processor [102].

The conceptional realisation of a clock recovery system is shown in Figure 2-1. This illustrates a high-Q local oscillator, which is normally synchronised to the input transitions by means of phase locking.

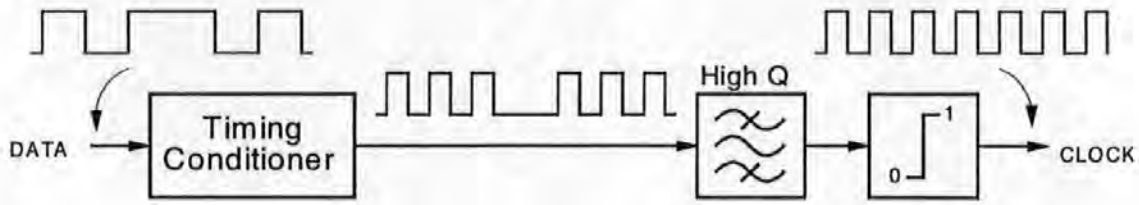


Figure 2-1 Feed-forward clock recovery system.

The review of clock recovery architectures summarised in Section 1.1.7, demonstrated that in general, feedback systems exhibit relatively poor acquisition performance. On the other hand, an advantage of a PLL is its ability to track symbol rate drift. However, the simulation results presented in Appendix 2, demonstrate that a narrow loop bandwidth to improve flywheel performance, for use with RLL data, restricts the tracking range to less than ±1%.

Many applications have a stable symbol rate, which is often derived from a crystal-controlled clock in the transmitter or record electronics. Also, the velocity of recording media is often crystal controlled, as in helical scan tape recorders. In satellite communications it is usual for the data rate to be stable to within 0.001% (10-parts/million) [39], while for a hard disk the speed change is typically less than 0.1% [18]. In such applications, once acquisition has been achieved, there is no requirement to track large variations in symbol-rate drift and this provides the potential for a feed forward clock recovery system.

The DSP-based clock recovery scheme exploits the "ringing" feature of a high-Q IIR filter, when the poles are close to the unit circle (i.e. high gain and narrow bandwidth). This feature of a high-Q IIR filter is demonstrated by the impulse response, shown in the simulation of Figure 2-2.

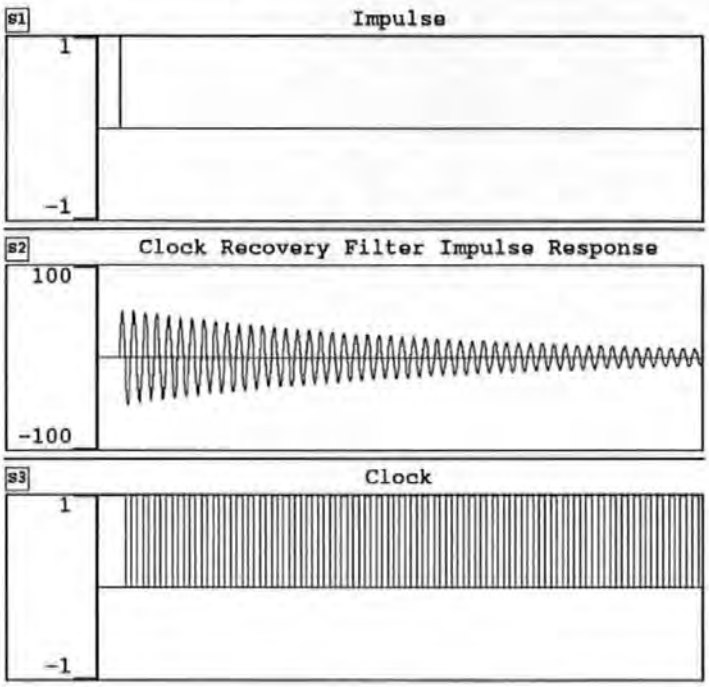


Figure 2-2 IIR filter impulse response

The filter "rings" at the resonant frequency and decays exponentially to zero. The filter output can be hard limited, with a threshold at zero to produce a logic level clock, which flywheels for the duration of the filter oscillations.

In the context of clock recovery this feature is desirable, since it can be exploited to provide rapid clock acquisition and a good clock flywheel effect over periods of lost input.

2.1. System description

The clock recovery system and associated waveforms are shown Figure 2-3.

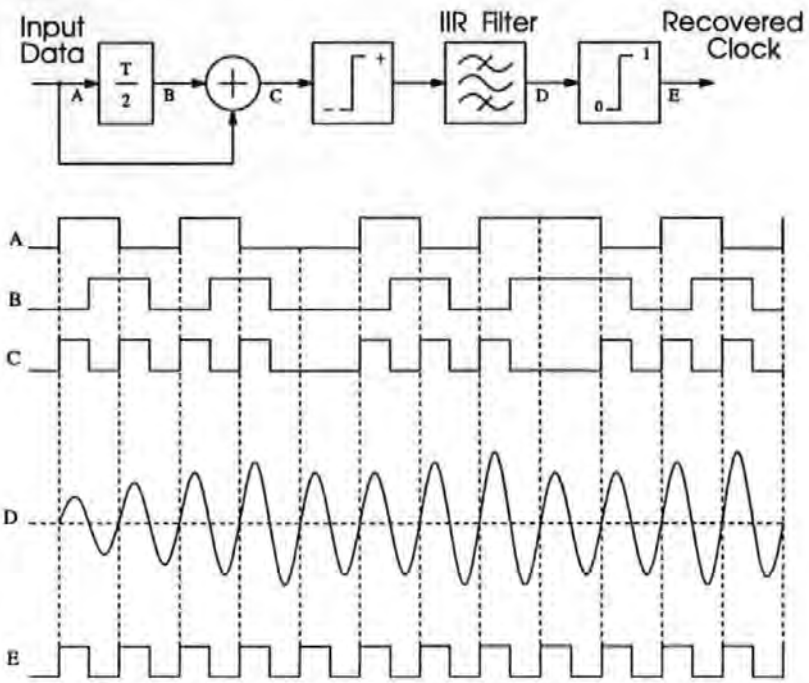


Figure 2-3 Clock recovery system with waveforms

Coded data symbols, *A*, are applied to a timing conditioner, where they are delayed by a half symbol period and modulo-2 added to the non-delayed symbol to give signal *C*, which has a strong frequency component at the desired clock rate. This signal resembles the desired clock, but at this stage is unusable due to gaps resulting from run lengths in the coded symbols. Before application to the IIR filter, the waveform *C* is converted from logic levels to positive and negative numeric values to stimulate the filter. These numeric values must be carefully chosen, to provide the highest possible stimulation (thereby maximising the flywheel period), whilst ensuring that the filter does not overflow. As a result of the applied stimulus, the filter rings at its tuned frequency, which has been set to the symbol rate, to produce

waveform *D*. Subsequent stimulus pulses reinforce the filter oscillations, causing its output to grow in amplitude. This waveform is now hard limited back to logic levels to give waveform *E*, the recovered clock output. If input pulses should fail to occur (for example, due to run lengths in the code or a data dropout), then the filter output will begin to decay towards zero. The recovered clock will flywheel over these periods, and as long as the stimulus to the filter is reinstated before the filter output reaches zero, then clock synchronisation is maintained.

2.1.1. Timing conditioner

The timing conditioners previously described in Section 1.1 consists of edge detectors, which utilise the properties of an ECPD [103]. For the DSP-based clock recovery scheme a strong clock component with a 50% duty cycle is preferable, for stimulating the IIR filter. The input levels then combine, in phase within the internal feedback of the filter structure to achieve maximum amplitude at the filter output. The auto-correlation function of NRZ data provides a correlation peak at the bit period. When the retrieved data are delayed by half a bit period (T/2) and multiplied by the non-delayed data, as illustrated in Figure 2-4, a strong spectral component at the desired clock rate is achieved.

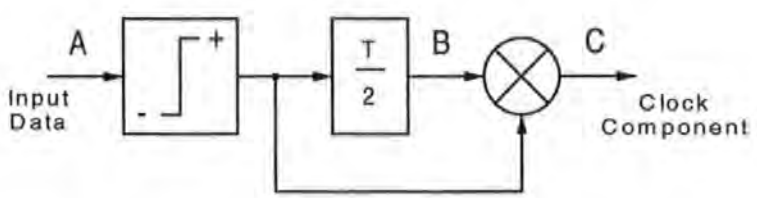


Figure 2-4 Delay and Multiply conditioner

Since the input symbols are logic levels, the "Delay and Multiply" can be modified to a modulo-2 adder by means of an "Exclusive-Or" gate, followed by a bipolar scalar, as illustrated in Figure 2-5.

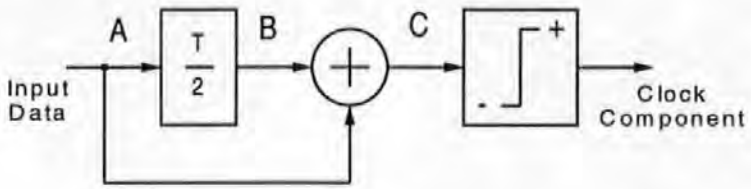


Figure 2-5 Delay and Add conditioner

DSP processors contain a multiply register and the "Delay and Multiply" function is easy to execute. Whereas, multiplication is difficult to perform using logic hardware, hence the "Delay and Add" conditioner is preferred for logic implementation, such as an FPGA. The timing of the "Delay and Add" conditioner is shown in Figure 2-6, together with the resulting response of the IIR filter.

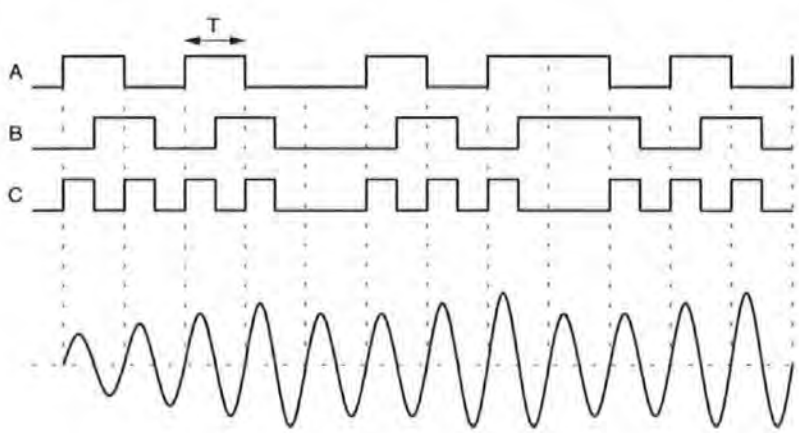


Figure 2-6 Delay and Add Timing

Half rate recording codes have transitions at the centre and the end of the bit period. The auto-correlation function of these codes provides two correlation peaks, one at the bit period and another at half the bit period (the symbol rate).

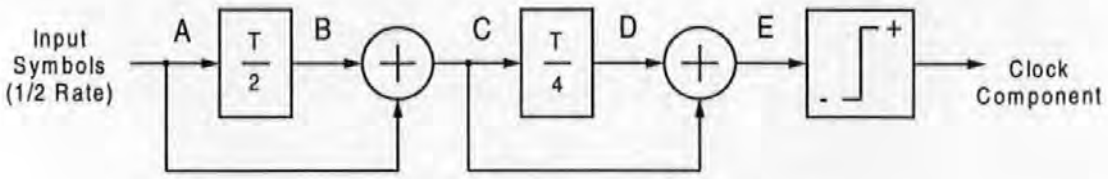


Figure 2-7 Two stage Delay and Add conditioner

A clock must therefore be recovered at the symbol rate, to successfully decode the retrieved symbols. For ½ rate codes a two-stage signal conditioner is applied; the first T/2 stage generates a clock component at the bit rate and the second T/4 stage doubles the frequency, generating a clock component at the symbol rate for stimulating the IIR filter.

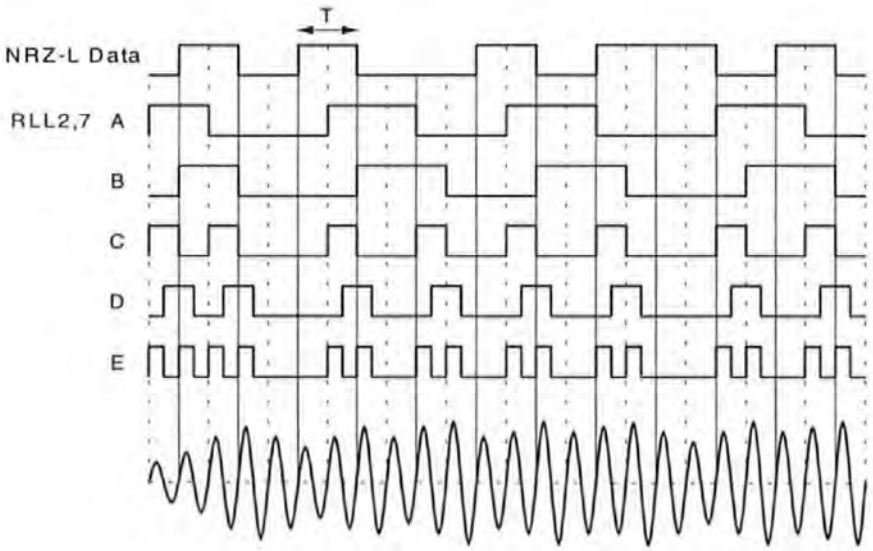


Figure 2-8 Timing diagram for 1/2 rate codes

The IIR filter is tuned to the symbol rate frequency, the output of which is hard limited to provide the recovered clock, which can then be used to re-sample the ½ rate encoded symbols. In practice, the hard limiter is achieved by taking the most significant bit, the sign bit in 2's complement arithmetic.

2.2. Clock recovery filter theory

An IIR filter is normally formed as a second order section containing both poles and zeros, providing a centre frequency and stop-band nulls. In this application, the stop-band null is of no consequence and the design can be simplified to have feedback paths only.

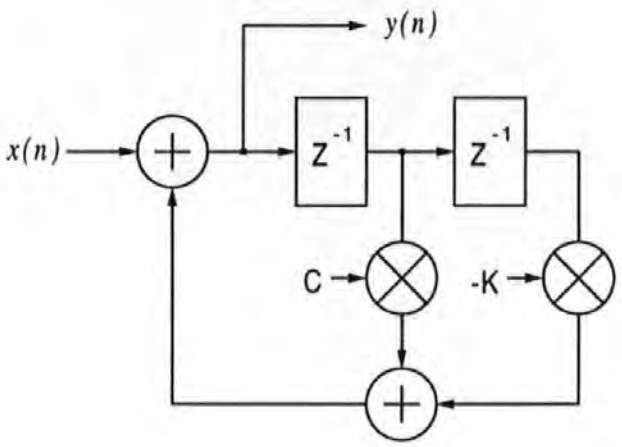


Figure 2-9 Architecture of clock recovery IIR filter

The filter output can be expressed in the time domain as:

$$y(n) = x(n) + Cy(n-1) - Ky(n-2) \quad \text{..... (2-1)}$$

Applying the time-shift property of the z-transform [104].

i.e. $x(n) \Leftrightarrow X(z), \quad y(n) \Leftrightarrow Y(z)$

$$x(n-n_0) \Leftrightarrow Z^{-n_0}X(z), \quad y(n-n_0) \Leftrightarrow Z^{-n_0}Y(z)$$

Taking the Z-transform of equation (2-1)

$$Y(z) = X(z) + Cz^{-1}Y(z) - Kz^{-2}Y(z)$$

$$X(z) = Y(z) - Cz^{-1}Y(z) + Kz^{-2}Y(z)$$

$$X(z) = Y(z)[1 - Cz^{-1} + Kz^{-2}]$$

$$H(z)=\frac{Y(z)}{X(z)}=\frac{1}{1-C\,z^{-1}+K\,z^{-2}} \qquad \text{..... (2-2)}$$

$$H(z)=\frac{z^2}{z^2-C\,z+K}=\frac{zeros}{poles} \qquad \text{..... (2-3)}$$

Equation (2-3) is the z-domain transfer function of the filter, which contains poles and a double zero at the z-plane origin.

Substituting $e^{j\omega T}$ for z into equation (2-2) yields the frequency response:

$$H\left(e^{j\omega T}\right)=\left(1-C\,e^{-j\omega T}+K\,e^{-j2\omega T}\right)^{-1} \qquad \text{..... (2-4)}$$

Expressed as a function of frequency *(f)*:

$$H(f)=\left(1-C\,e^{-j2\pi\frac{f}{f_s}}+K\,e^{-j4\pi\frac{f}{f_s}}\right)^{-1} \qquad \text{..... (2-5)}$$

For normalised frequencies, the sample rate, $f_s = 1$ and the centre frequency, $f_c = {}^{1}/_{16} = 0.0625$, the coefficients are then given as $C = 1.84414$ and $K = {}^{255}/_{256}$.

The corresponding frequency response is plotted in Figure 2-10.

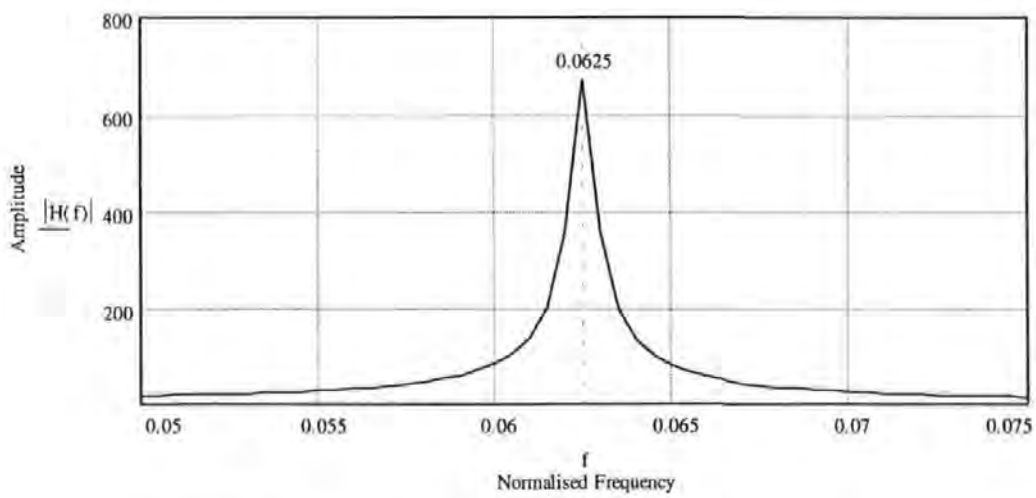


Figure 2-10 Frequency response of the IIR filter

The phase response is plotted in Figure 2-11.

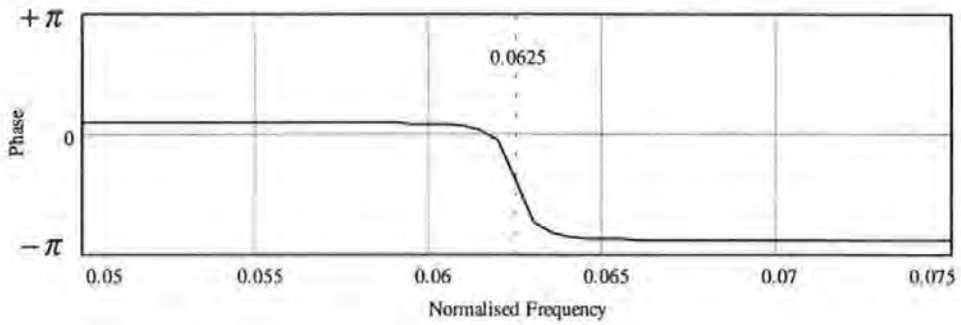


Figure 2-11 Phase response of the IIR filter

The poles of equation (2-3) describe the resonance and stability of the filter. The denominator (poles) is in the form of a second order polynomial (i.e. $ax^2 + bx + c$) and the roots can be determined by $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$.

Hence from equation (2-3) $z^2 + Cz + K = 0$

$$z_{pole} = \frac{C}{2} \pm \frac{1}{2}\sqrt{4K - C^2} \qquad \text{..... (2-6)}$$

The poles must lie within the unit circle to maintain stability. From equation (2-6), and analysis of the Z-plane diagram in Figure 2-12, r can be determined.

$$r^2 = \left(\frac{1}{2}\sqrt{4K - C^2}\right)^2 + \left(\frac{C}{2}\right)^2$$

$$r = \sqrt{K} \qquad \text{..... (2-7)}$$

The bandwidth of the filter is given by: $BW = \frac{(1-r)f_s}{\pi}$

And the filter's Q becomes: $$Q = \frac{f_c}{BW} = \frac{f_c}{\frac{(1-r)f_s}{\pi}} \quad \text{.....} \quad (2\text{-}8)$$

If the sample rate is normalised, $f_s = 1$ then:

$$Q = \frac{\pi f_c}{1-r} \quad \text{.....} \quad (2\text{-}9)$$

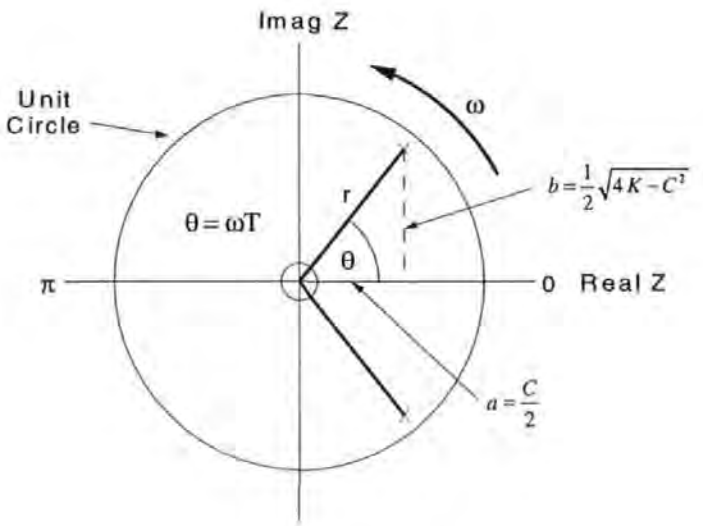


Figure 2-12 Z-Plane diagram

Assuming complex-poles z can be expressed in polar form.

$$z_{pole} = a \pm jb = r\cos\theta \pm jr\sin\theta$$

When $$a = \frac{C}{2} = r\cos\theta \qquad \text{and} \qquad b = \frac{1}{2}\sqrt{4K - C^2} = r\sin\theta$$

Then equation (2-6) is expressed using complex notation.

$$z_{pole} = \frac{C}{2} \pm j\frac{1}{2}\sqrt{4K - C^2} \quad \text{.....} \quad (2\text{-}10)$$

For a high-Q filter, the poles are very close to the unit circle ($r = 1$), then from the Z-plane diagram:

$$\tan\theta = \frac{\frac{1}{2}\sqrt{4K - C^2}}{\frac{C}{2}}$$

$$\theta = \tan^{-1}\left(\sqrt{\frac{4K}{C^2} - 1}\right) \qquad \text{..... (2-11)}$$

Rearranging equation (2-11) the second coefficient can be determined:

$$C = \pm\sqrt{\frac{4K}{1+\tan^2\theta}}$$

Given the identities $\sec^2\theta = 1+\tan^2\theta$ and $\sec^2\theta = \frac{1}{\cos^2\theta}$

and with $\theta = \frac{2\pi f_c}{f_s}$

Then: $$C = \pm\sqrt{\frac{4K}{\sec^2\frac{2\pi f_c}{f_s}}} = \pm\sqrt{\frac{4K}{\frac{1}{\cos^2\frac{2\pi f_c}{f_s}}}} = 2\sqrt{K} \pm \sqrt{\cos^2\frac{2\pi f_c}{f_s}}$$

$$C = \sqrt{K}\cdot 2\cos\left(\frac{2\pi f_c}{f_s}\right) \qquad \text{..... (2-12)}$$

Rearranging equation (2-12), the corresponding sample rate can be determined:

$$f_s = \frac{2\pi f_c}{\cos^{-1}\left(\frac{C}{2\sqrt{K}}\right)} \qquad \text{..... (2-13)}$$

Coefficient K is chosen to be very close to unity to provide a high-Q filter. Good clock recovery performance has been achieved with coefficient $K = {}^{255}/_{256} = 0.99609$, coefficient $C = 1.84414$, a sample rate normalised to $f_s = 1$ and a symbol rate centre frequency $f_c = {}^{1}/_{16} = 0.0625$, providing a filter $Q = 100$.

These theoretical parameters have been applied to SPW FDS (Filter Design System) [25] and the resulting frequency response, impulse response and pole-zero diagram are shown in Figure 2-13.

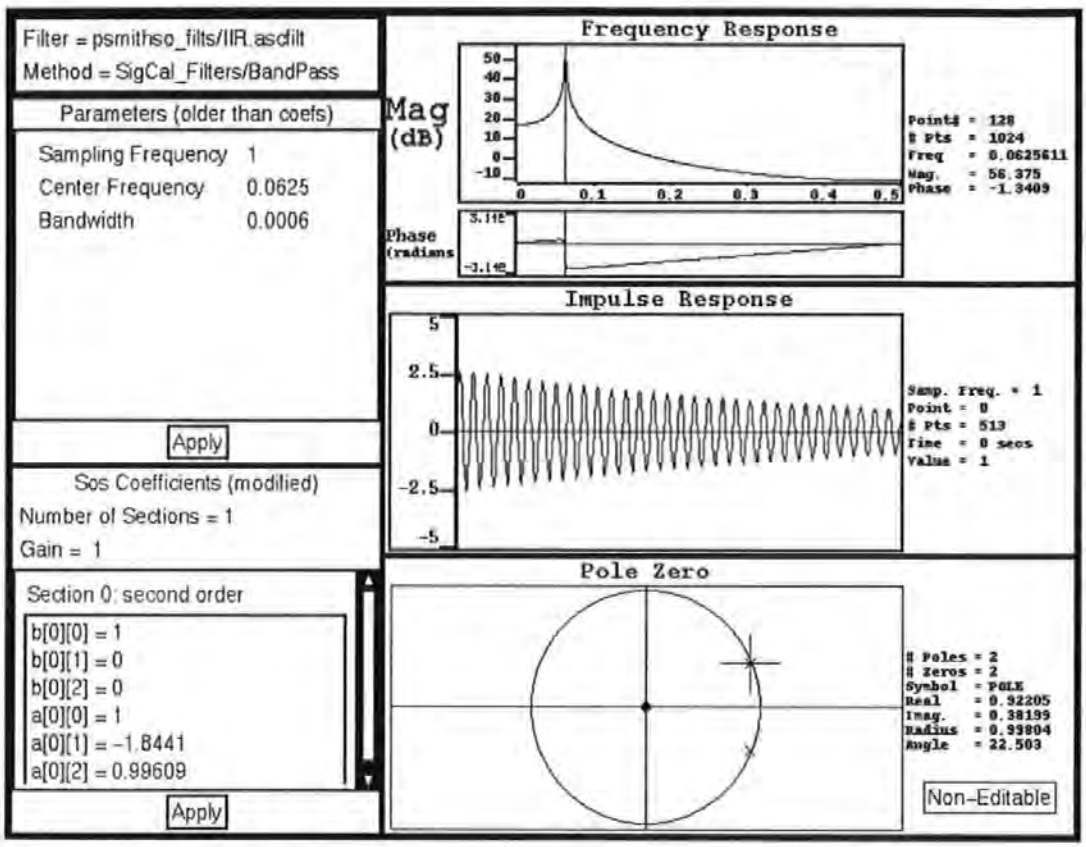


Figure 2-13 Filter design analysis

2.3. Simulations

Full simulation models and results of the DSP-based, fixed frequency clock recovery algorithm are presented in Appendix 4, key extracts of which are included in this chapter. The scheme has been simulated to evaluate:

- **Acquisition performance.**

  For each of the test codes, clock acquisition is achieved within a symbol period, from the first transition of the received symbols.

- **Flywheel performance.**

  For a filter Q of 100, the system can withstand a data dropout of 80 symbol periods without losing clock synchronisation. Cascading filter sections can double this flywheel period. The graph in Figure 2-17 summarises a collection of simulation data, representing the clock flywheel performance versus filter Q, together with the corresponding acceptable symbol rate drift. A filter Q of 100 has an acceptable symbol rate drift of ±1%.

- **Noise rejection performance.**

  A precisely controlled amount of gaussian jitter [105] representing noise has been applied to the input symbols. The histograms of Figure 2-25 represent the PDF (Probability Density Function) of jitter in the recovered clock. For a filter Q of 100, the PDF with zero jitter is almost 98% for Manchester data and 90% for RLL2,7 encoded data, indicating good jitter rejection properties. The graph of Figure 2-26 summarises the PDF of jitter in the recovered clock relative to the filter Q.

2.3.1. Simulation model

For simulation of the clock recovery algorithm, several custom hierarchy blocks have been created [25]. Simulation models for the RLL encoders, as described in Section 1.2.2, can be seen in Appendix 3, while a description of the developed "Jitter Generator Block" and the "PDF Block" are provided in Appendix 4.

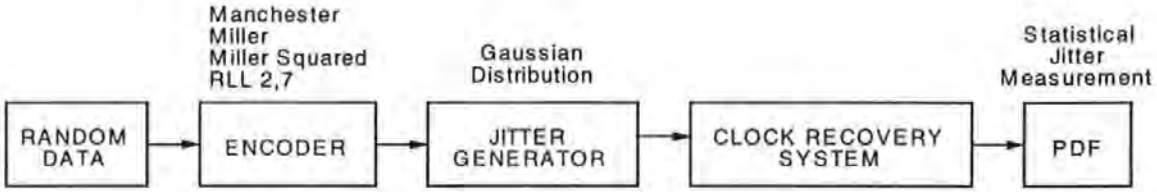


Figure 2-14 Overview of simulation model

Figure 2-14 provides an overview of the simulation model. Random NRZ data are encoded into one of the test recording codes. The encoded symbols are then passed through a specially designed jitter generator, where a controlled amount of gaussian jitter is applied to the encoded symbols to simulate noise. The DSP-based clock recovery system recovers the symbol clock, which is analysed for clock acquisition, flywheel and jitter rejection performance.

2.3.2. Simulation results

Clock recovery performance has been evaluated using the four test recording codes: Manchester, Miller, Miller^2 and RLL2,7; and also scrambled NRZ data (maximum run length = 15). Full simulation results are presented in Appendix 4.

2.3.2.1. Acquisition performance

The simulation results of Figure 2-15 show the acquisition performance for Manchester encoded data. After the application of data there is a 2-sample delay before the output of the filter is asserted. At 16 samples per symbol period this acquisition time represents an interval of $^2/_{16}$ of the first symbol period, where the clock frequency is twice the bit rate (½ Rate code). This short acquisition time is the same for each of the test recording codes.

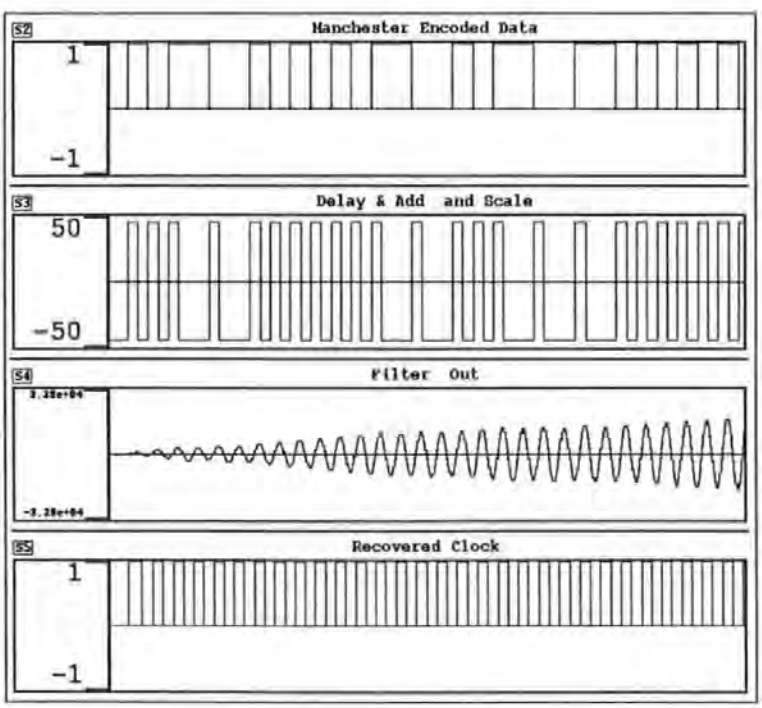


Figure 2-15 Acquisition performance - Manchester encoded data

It can be seen that as soon as the first data symbol is applied, the filter starts to ring at the fundamental frequency, which has been set to the symbol rate. Subsequent edges then reinforce the ringing and the filter grows in amplitude. The hard limited filter output provides the recovered clock.

Similarly, Figure 2-16 shows the acquisition performance for the RLL 2,7 code.

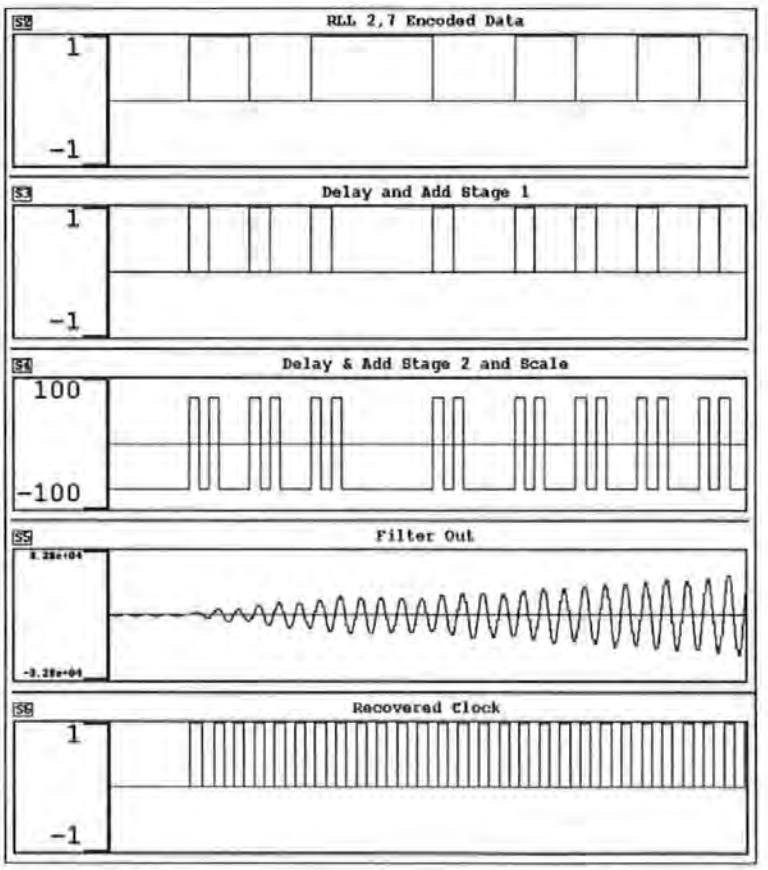


Figure 2-16 Acquisition performance - RLL 2,7 encoded data

The *d* constraint of the RLL2,7 code (*d* =2) increases the period between adjacent transitions, hence the stimulus to the IIR filter is increased with a two stage "Delay and Add" timing conditioner, as previously illustrated in Figure 2-7. The first T/2 stage generates a clock component at the bit rate and the second T/4 stage doubles the frequency, generating a clock component at the symbol rate for stimulating the

IIR filter. The simulation results in Figure 2-15 and Figure 2-16 show that the recovered clock is acquired within the first symbol period of the retrieved data. This demonstrates a major advantage over that of a PLL, which requires several hundred input cycles to achieve lock (see PLL results in Appendix 2).

Similar performance is obtained for the Miller and Miller^2 codes and full simulation results are shown in Appendix 4.

2.3.2.2. Flywheel performance

The impulse response shown in Figure 2-2 demonstrates the ringing properties of the IIR filter when the poles are close to the unit circle. This feature is exploited to provide a good flywheel effect over periods of lost input, thereby maintaining clock synchronisation. The closer the poles are placed to the unit circle (higher filter Q), then the greater the flywheel period over lost input. A higher-Q reduces the bandwidth of the bandpass filter; hence the trade-off to this increased flywheel time is a reduction in data rate drift.

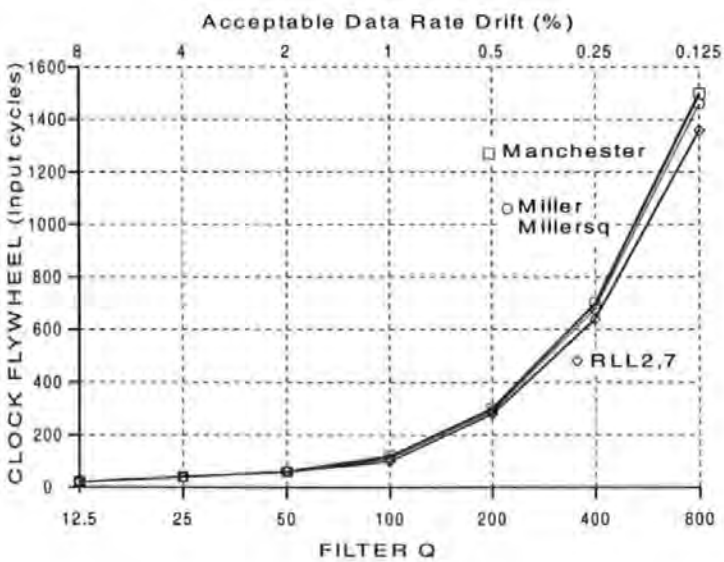


Figure 2-17 Filter-Q versus flywheel

The graph of Figure 2-17 shows that the number of clock periods for which the clock flywheels, increases exponentially with the filter Q. If the acceptable data-rate drift of a system is known, then the graph can be used to determine the highest permissible filter Q, and hence maximise the clock flywheel period.

The simulation results presented, used a filter Q of 100, which represents tolerance to a 1% data rate drift.

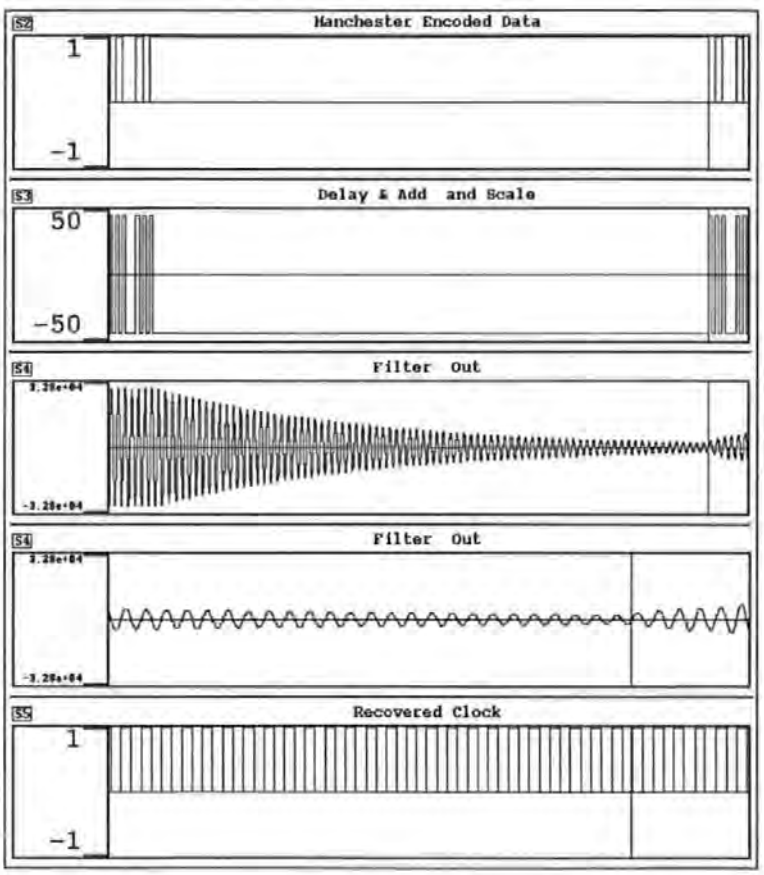


Figure 2-18 Flywheel performance - Manchester encoded data

Figure 2-18 shows the flywheel performance for the Manchester code when subjected to a 40-bit (80-symbols) data dropout. The IIR filter decays exponentially and the hard limited filter output provides an uninterrupted recovered clock.

Zooming-in at the end of the dropout shows the recovered clock remaining in-phase when data are re-applied.

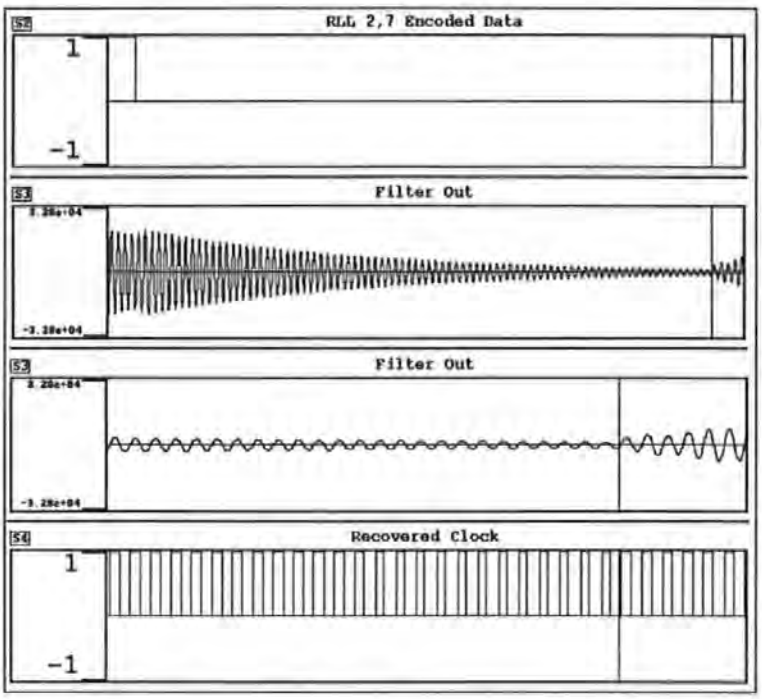


Figure 2-19 Flywheel performance - RLL2,7 encoded data

Figure 2-19 shows the flywheel performance for the RLL 2,7 code over a 40-bit data dropout. The RLL2,7 code has larger run length (k =7 constraint) than that of the Manchester code and hence a reduction in the flywheel performance is to be expected. Zooming-in at the end of the dropout shows a reduction in the recovered clock mark-space-ratio as the phase begins to slip, however new symbols arrive before total loss of lock, and synchronisation is maintained.

For codes with even longer run lengths, a two-stage filter can be applied.

Scrambled NRZ data has a run length of 2^{n-1}, where n is the length of the shift-register of the PRBS generator, as described in Section 1.2.1. For example, a 64K-length PRBS sequence is typical, where the maximum run length is 15.

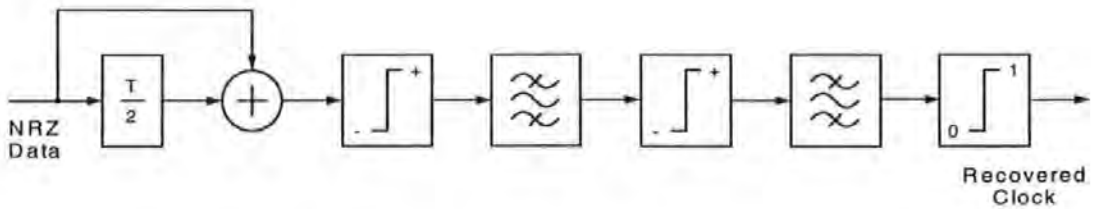


Figure 2-20 Two-stage filter section

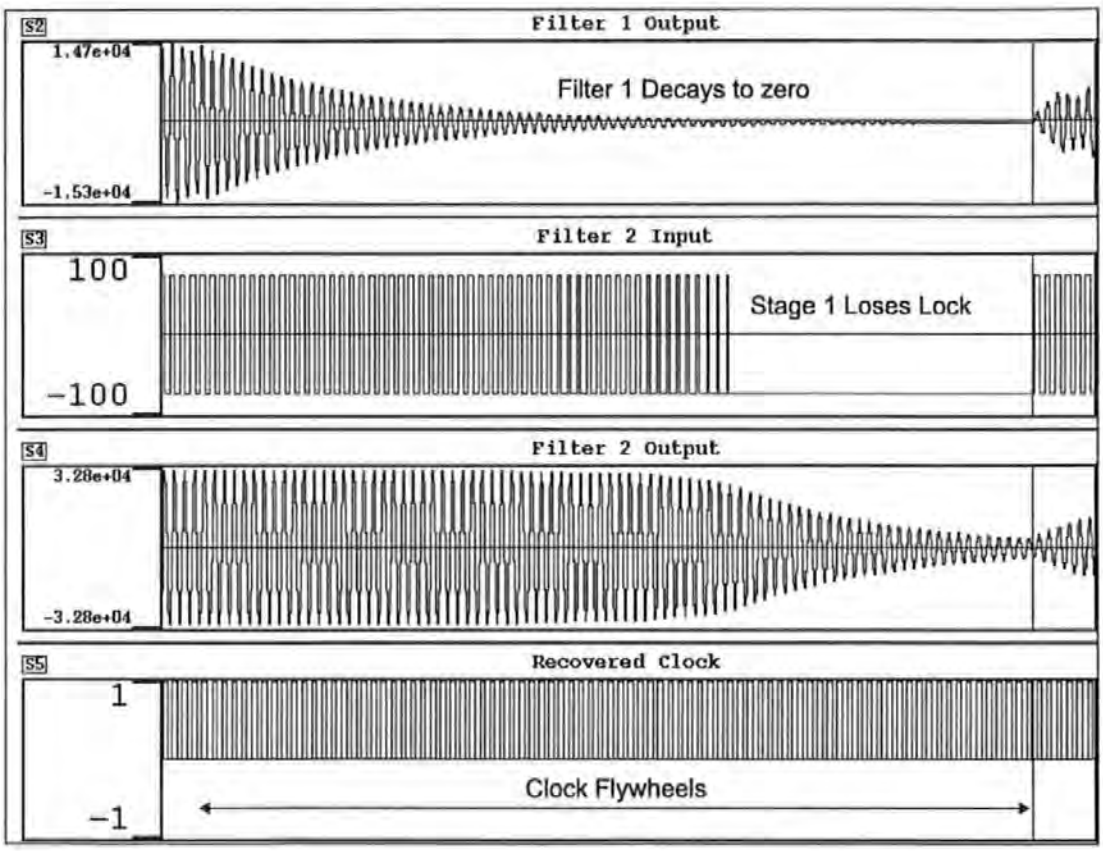


Figure 2-21 Flywheel performance - Two stage filter (NRZ data)

Cascading filter sections can increase the flywheel period by a factor of 2, as shown in Figure 2-20 and Figure 2-21. Clock acquisition is delayed by only 2 samples, corresponding to the 2-tap delay within the IIR filter architecture.

This simulation demonstrates how the clock recovery system can be made to tolerate long encoded run lengths and severe data dropouts.

2.3.2.3. Noise rejection performance

A practical system contains timing errors due to system noise. The simulation of Figure 2-22 shows the effect of additive noise on retrieved data. While Figure 2-23 shows the corresponding eye diagram, with the resulting phase jitter indicated. A good clock recovery system should have filtering to minimise jitter in the recovered clock and hence, reduce the probability of miss-timing the data when re-sampled.

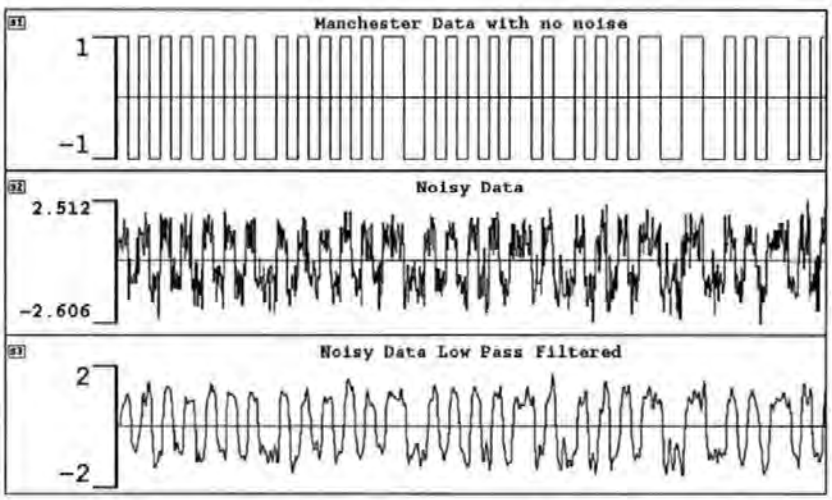


Figure 2-22 Data with additive noise

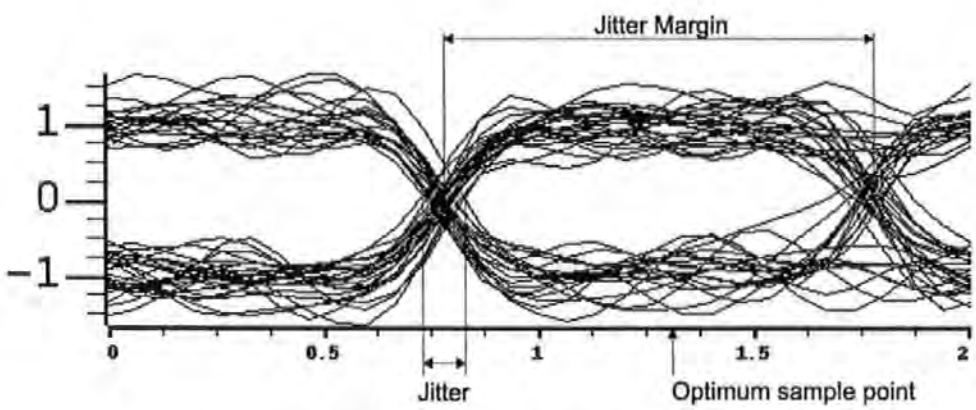


Figure 2-23 Eye diagram of noisy data

In the following simulations, ±3 samples jitter with a gaussian distribution (corresponding to ±67.5 degrees phase jitter) have been applied to the encoded symbols and the jitter rejection performance, statistically analysed over half a million samples.

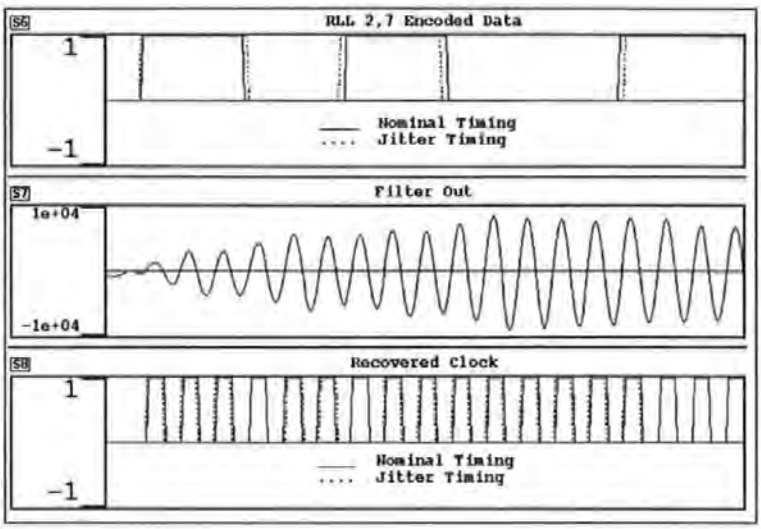


Figure 2-24 Acquisition with applied jitter

Figure 2-24 shows clock acquisition performance when jitter is applied to the RLL 2,7 code. The narrow bandwidth IIR filter averages phase errors in the retrieved data and as the output of the filter grows in amplitude, the clock recovery system becomes less sensitive to input phase variations, minimising jitter in the recovered clock.

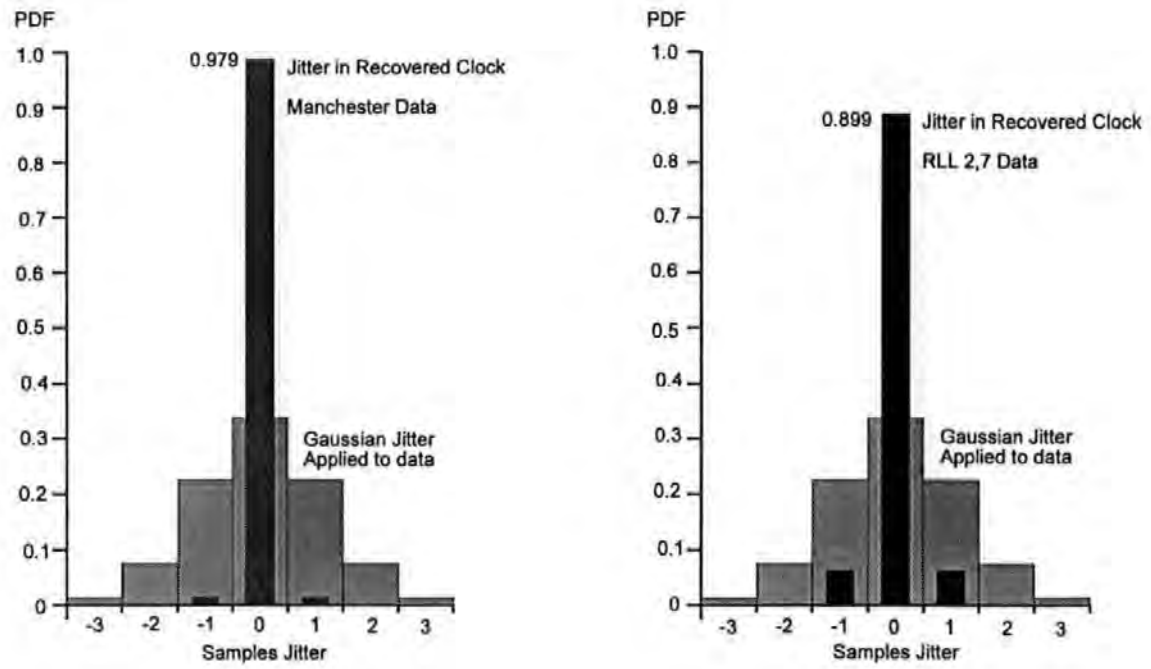


Figure 2-25 PDF of jitter in the recovered clock

When gaussian jitter is applied to Manchester encoded data the clock recovery filter virtually eliminates all jitter in the recovered clock. The standard deviation of the recovered clock is only 3.3 degrees. While for the RLL2,7 code, nearly 90% of the recovered clock contains zero phase error, and the corresponding standard deviation is 7.1 degrees. These results are quantified in the histograms of Figure 2-25, which were obtained from simulation results over 500,000 samples with a filter Q of 100.

To reduce sampling errors, the clock is delayed such that the sampling edge re-times the data at the centre of the symbol period, thereby maximising the jitter detection window. Jitter of ±½ a symbol period can then be tolerated before data errors occur, as illustrated in Figure 2-23.

As previously described, the higher the filter Q, the narrower the filter bandwidth and hence the greater the rejection to jitter in the recovered clock.

The graph of Figure 2-26 demonstrates jitter rejection for Manchester encoded data for different values of filter Q.

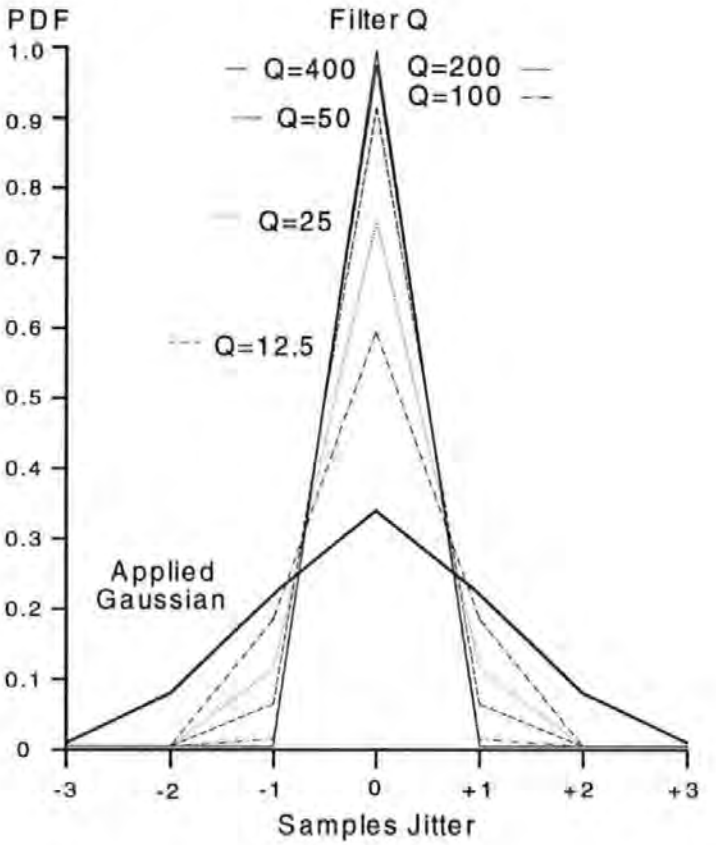


Figure 2-26 PDF of jitter versus Filter Q - Manchester data

It can be seen that a filter Q of 100 provides good jitter rejection, minimising jitter in the recovered clock.

2.4. Test apparatus

The simulation results have been verified in real-time, using specially developed hardware, based around a TMS320C25 DSP processor. A description of the DSP software is included in Appendix 6, together with the assembler source code and experimental results.

Clock recovery at nine different data rates ranging from 50 to 4800 bits/sec have been programmed into common hardware, with selectable sample rates of 6.4KHz, 38.4KHz and 76.8KHz. A two-stage IIR filter is used, as shown Figure 2-29, operating from DSP software within the processor's on-chip memory. In all cases, the flywheel time of the recovered clock is greater than 46 NRZ data periods. Figure 2-27 shows a block diagram of the DSP-based clock recovery hardware.

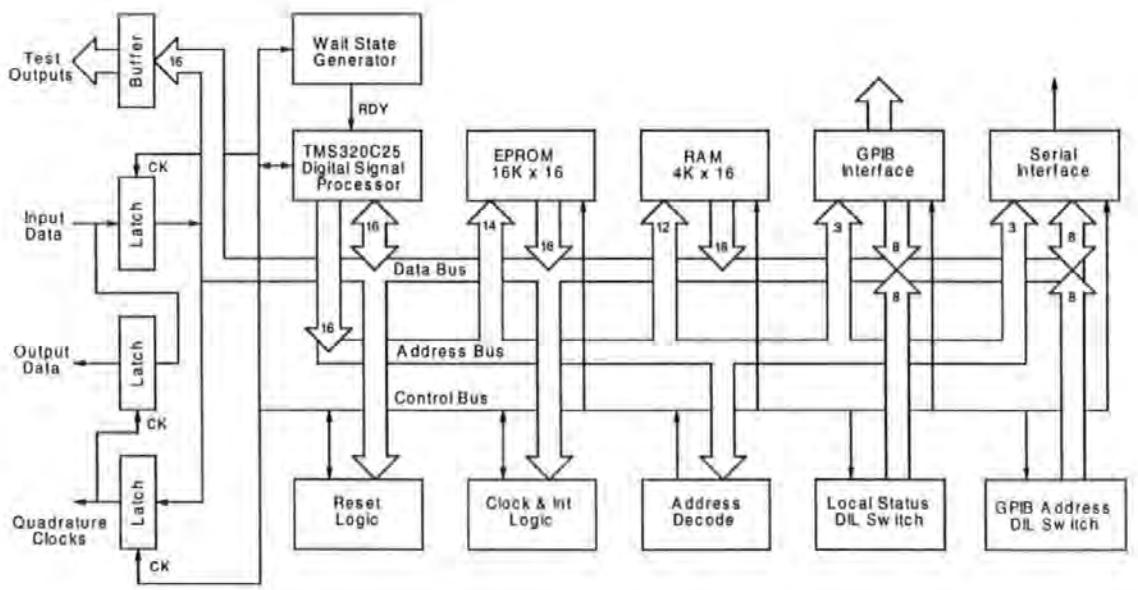


Figure 2-27 DSP-based clock recovery hardware - Block diagram

Figure 2-28 shows the corresponding prototype board, which has been constructed using a wire-wrap technique. Schematic diagrams are shown in Appendix 5.

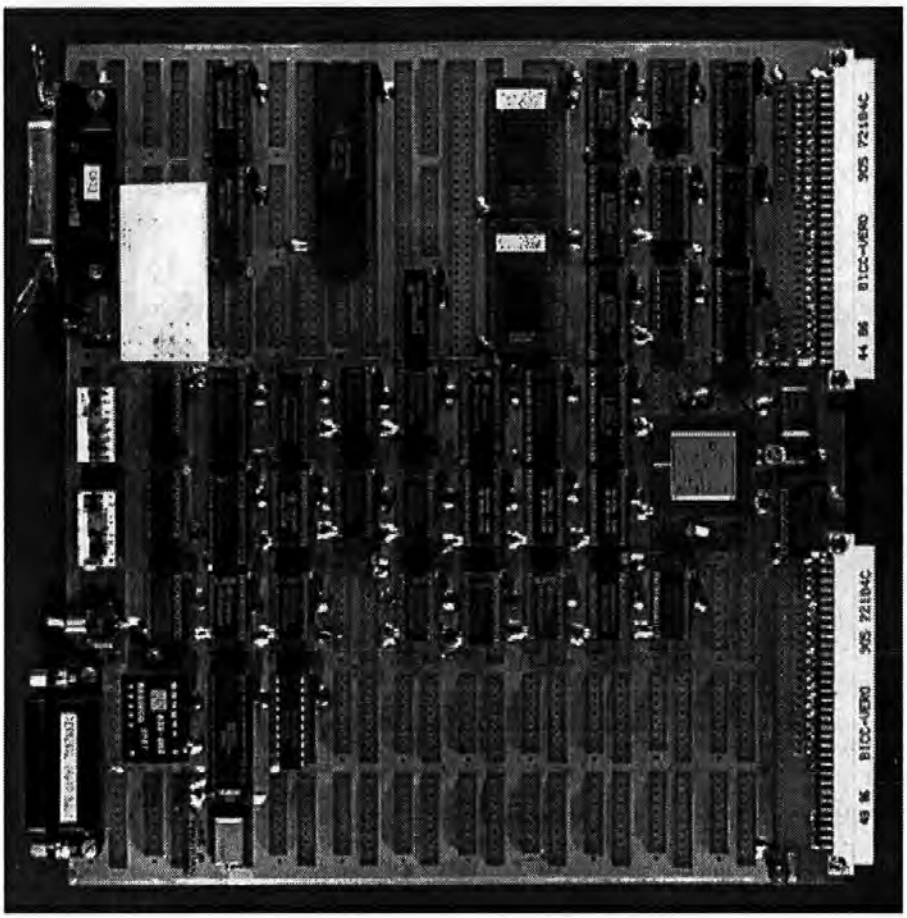


Figure 2-28 DSP-based clock recovery board

Test outputs are provided, such that various points within the clock recovery algorithm can be written, under software control, to DAC's (Digital to Analogue Converters), enabling digital signals to be viewed in analogue form on an oscilloscope. This is particularly useful for viewing the output of the IIR filters. A separate DAC board, with a bank of six DACs has also been developed to allow multiple viewing of signals. The schematic diagram for this DAC board is also included in Appendix 5.

The desired default data rate is pre-selected with a Local Status DIL (Dual In Line) switch. At power-up, the binary code of the switch is read, resulting in a software branch to initialise one of the nine pre-programmed data rates. The appropriate sample rate is selected, and the software delays and filter coefficients are loaded into the clock recovery algorithm for automatic operation, with no user interaction.

In addition to the turnkey operation, clock recovery at different data rates can be remotely selected with a "dumb-terminal" or alternatively via a standard GPIB instrumentation interface (General Purpose Interface Bus, also known as IEEE488 Bus) [106]. To facilitate this remote operation, a serial-interface (comprising a UART and an RS232 transceiver) and a GPIB interface are provided in hardware and software.

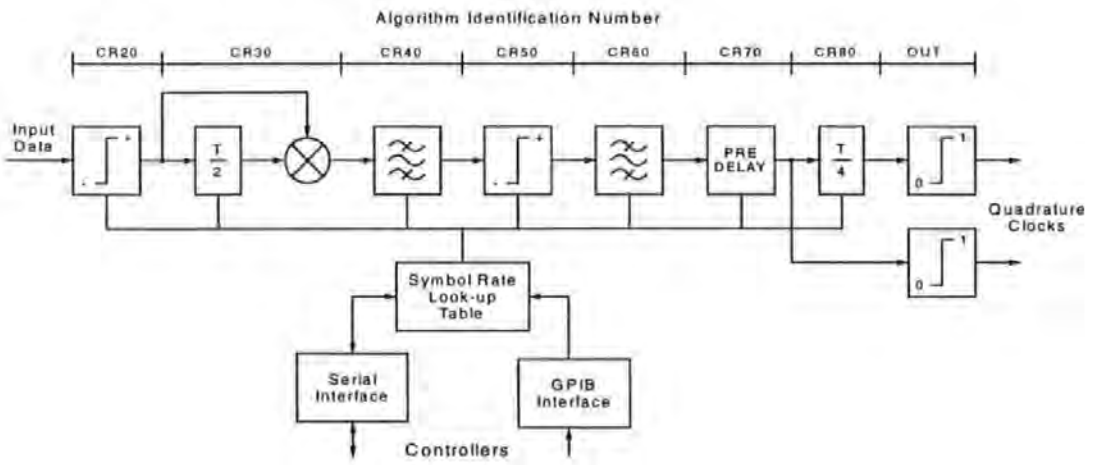


Figure 2-29 DSP Software algorithm overview

Figure 2-29 provides an overview of the clock recovery algorithm implemented in DSP software. Each algorithm block is assigned an identification number, which can be conveniently cross-referenced to the assembler source code.

As well as the clock recovery algorithm, the software includes a BIOS (Basic Input Output System) to handle remote terminal commands and display messages. The BIOS includes menu options, to aid user interaction for selecting the required data rate and for displaying the currently selected operation. A second menu, displayed on the remote terminal, allows memory to be viewed and manipulated, providing a means of fine-tuning "on-the-fly" without the need to reprogram EPROM's.

To avoid phase noise, the sample rate is derived in hardware, where the 40MHz-system clock is divided with binary counters, to provide three sample rate clocks 6.4KHz, 38.4KHz and 76.8KHz. A data selector is controlled by software and selects one of the three sample rates, depending on the required clock recovery data rate. The selected sample rate clock is converted to an interrupt pulse, which causes the software to execute the clock recovery interrupt service routine.

Three interrupt service routines are provided, INT0 has the highest priority and is used to execute the clock recovery algorithm. INT2 handles messages from the terminal, while INT1 actions commands derived from the GPIB interface. All interrupt service routines include a software stack for saving and restoring the contents of the processors internal registers.

The software has been developed into three separate assembly language modules, each of which are compiled and then linked to form an executable programme in Intel Hex format, for programming the onboard EPROM's.

A description of the software together with source code listings is included in Appendix 6.

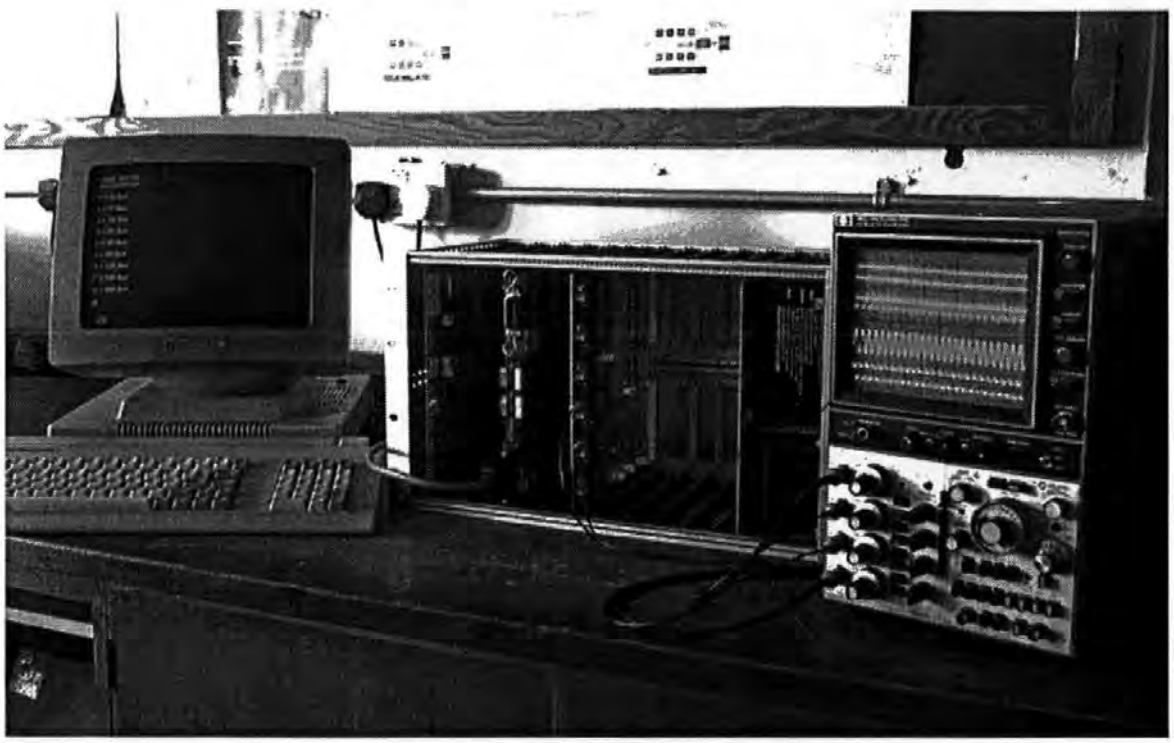


Figure 2-30 Clock recovery test system

A photograph of the prototype test system, complete with a terminal and oscilloscope, is shown in Figure 2-30. A standard 6U rack contains the DSP-based clock recovery board, a DAC board and also a test generator board for generating internal NRZ scrambled data at selectable bit rates.

Detailed analysis using a logic analyser, on the real-time clock recovery system, has verified the theoretical and simulation results. Measurements were carried out on clock acquisition, flywheel performance, coefficient quantization and data rate drift. These experimental results are included in Appendix 6.

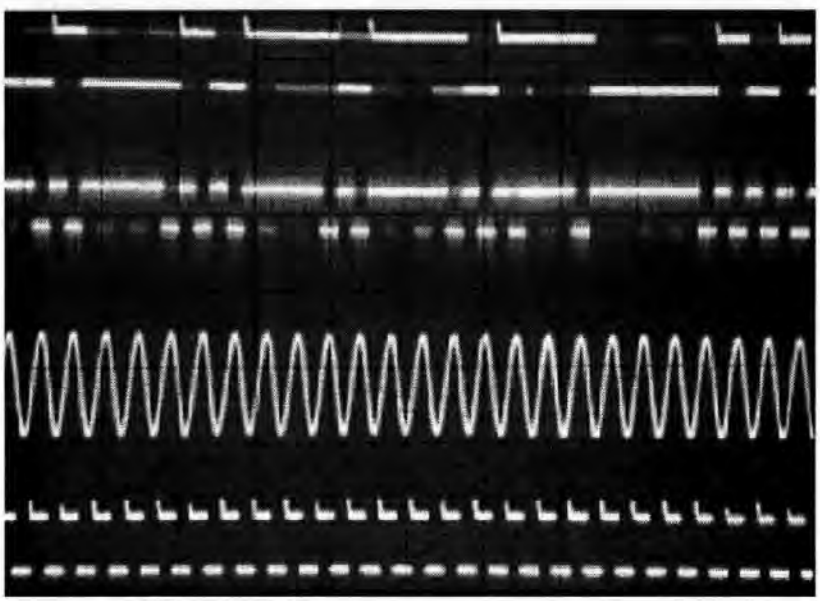


Figure 2-31 Oscilloscope display of clock recovery signals

Figure 2-31 shows a photograph of the oscilloscope screen, displaying the real-time clock recovery signals. The top trace shows the random input data. While the second trace shows the output of the "Delay & Multiply" timing conditioner. The slow camera shutter speed allows faint overlaid pulses to be seen, demonstrating the high clock content used to stimulate the IIR filter. The third trace is the output of the IIR filter section, which is sliced to produce the bottom trace, the recovered clock.

2.5. FPGA Implementation

A processor-based implementation may be compromised by the speed of software operation, which is executed sequentially. While DSP processors are becoming more powerful, both in terms of speed and architecture, these advancements in technology are at present subject to an increased cost penalty. To accommodate high sample rates, a parallel logic approach must be applied, either in discrete logic or an ASIC (Application Specific Integrated Circuit).

To achieve this high-speed solution, whilst still maintaining a degree of programmability, the clock recovery IIR filter has been implemented in a Xilinx 3020 (50MHz) FPGA [107]. The filter characteristics are programmed into the gate array at power-up or "on-the-fly". This FPGA implementation is referenced in the author's published paper, [Appendix 11{3}].

To avoid using conventional binary multipliers, which are slow and occupy a lot of silicon, the filter coefficients have been implemented using a novel technique of binary-weighted coefficients. This is achieved by bit-shifting the data words and adding or subtracting to obtain the required coefficient value.

Figure 2-32 shows the architecture of the FPGA filter, where the filter coefficients are implemented automatically at the sample rate, without the need for sub-multiplication processing.

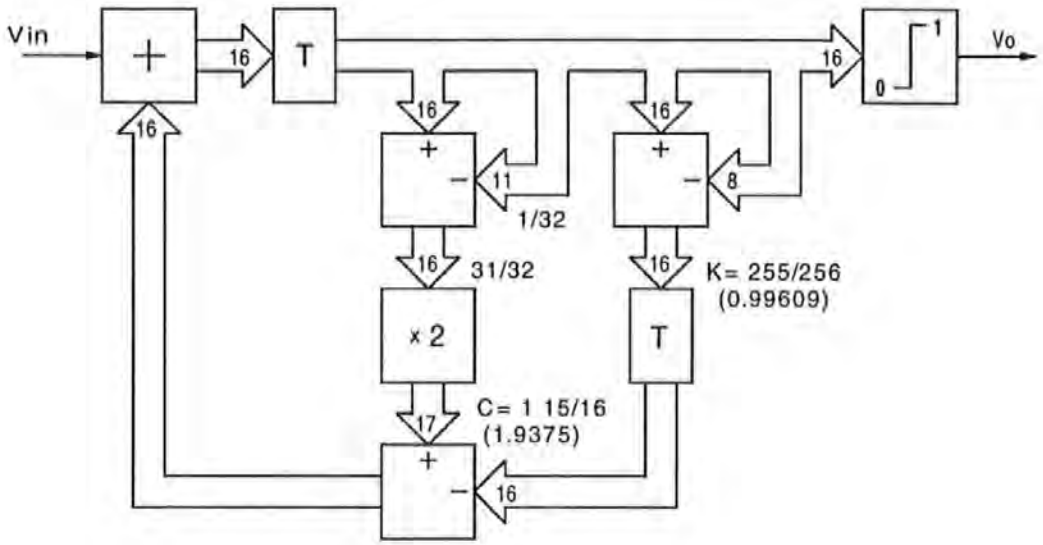


Figure 2-32 FPGA Filter architecture

Coefficient *K* is chosen to be close to unity producing poles close to the unit circle. Hence, coefficient *K* is set to 0.99609, which has a binary weighting of $^{255}/_{256}$. This coefficient is achieved by bit-shifting the 16 bit data word, 8 bits to the right, providing a binary divided value of $^{1}/_{256}$. This bit-shifted value is then subtracted from the non-shifted data word to provide the required coefficient of $^{255}/_{256}$ = 0.99609. Similarly, Coefficient *C* is chosen to be 1.9375, which has binary weighting of $1^{15}/_{16}$. The 16 bit data word is shifted 5 bits to the right to provide $^{1}/_{32}$, which is subtracted from the non-shifted data word to provide $^{31}/_{32}$. This intermediate value is then shifted to the left 1 bit, providing x2 multiplication as a new 17 bit data word, producing the required *C* coefficient $^{62}/_{32} = 1^{15}/_{16} = 1.9375$.

The binary weightings do impose quantization restrictions in the selection of the filter coefficients, however this restriction is offset by the choice of sample rate, which effectively fine-tunes the filter to the required centre frequency.

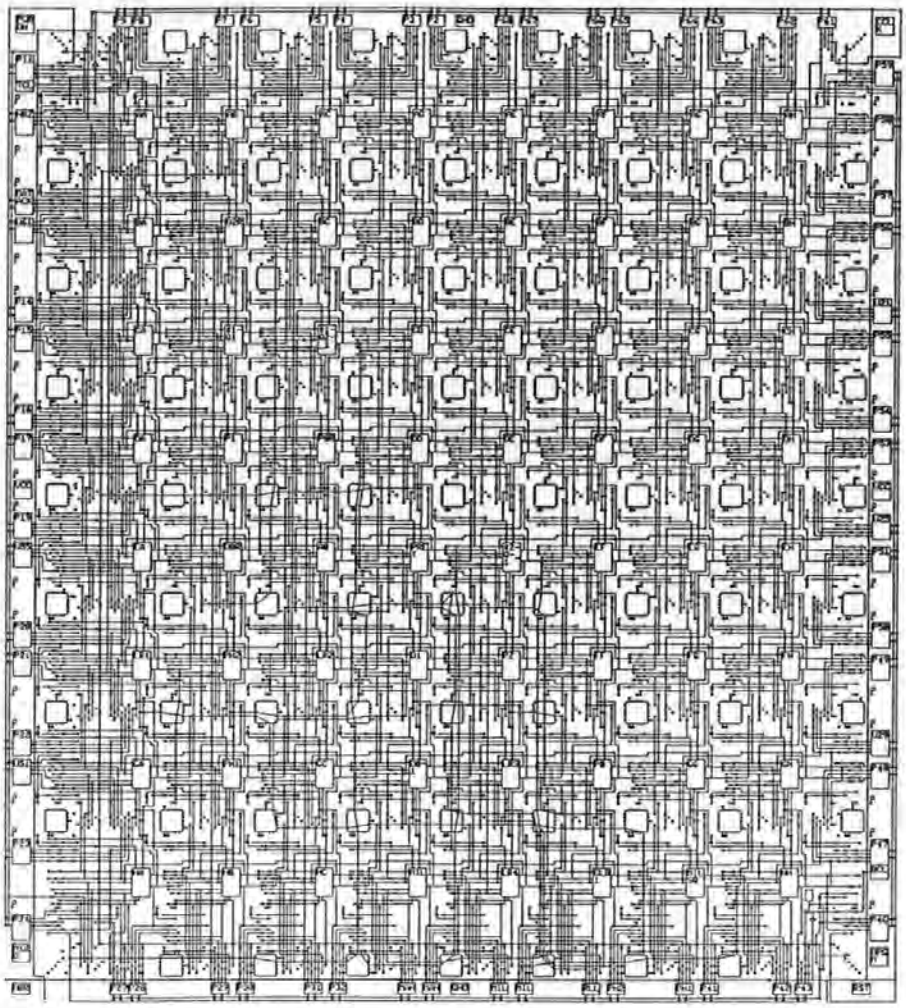


Figure 2-33 Routed FPGA die

Figure 2-33 shows the routed FPGA design, which utilises 61 of 64 CLBs (Configurable Logic Blocks). Appendix 5 contains the corresponding schematic diagrams, which have been developed using a hierarchy structure. The above FPGA design has been implemented for clock recovery on a satellite data modem, operating at a sample rate of 2.59MHz and a data rate of 100Kbits/sec. This application is briefly described in Section 2.5.2.

With the use of larger, faster FPGA devices, and employing adder look-ahead carry techniques, much higher sample rates are achievable. The author has also successfully extended the technique to FIR (Finite Impulse Response) filter architecture's for channel filtering applications. The technique of using binary weighted coefficients, results in an approximate reduction of 50% in silicon utilisation and a 50% increase in speed performance [108]-[110]. As a result, GEC-Plessey Semiconductors (at Plymouth) are investigating the use of this technique for dedicated filter integrated circuits.

2.5.1. Quantization

Finite word length can reduce the dynamic range of the filter, while quantized coefficients can result in de-tuning the filter's centre frequency.

All modern DSP processors contain at least a 16-bit data bus and usually contain an internal 32-bit multiply and accumulate register [111]. The FPGA filter has been implemented with a 16-bit word length, and quantization simulations using FDS have confirmed that quantization errors are negligible, as shown in Figure 2-34. However, the larger the word length, then the greater the capacity occupied within the FPGA device. Some applications may require the FPGA to incorporate other processes, in which case a device with increased capacity must be used, or alternatively the word length reduced. As described in Section 2.1, numeric values are chosen to provide the highest possible stimulation (thereby maximising the flywheel period), whilst ensuring that the filter does not overflow.

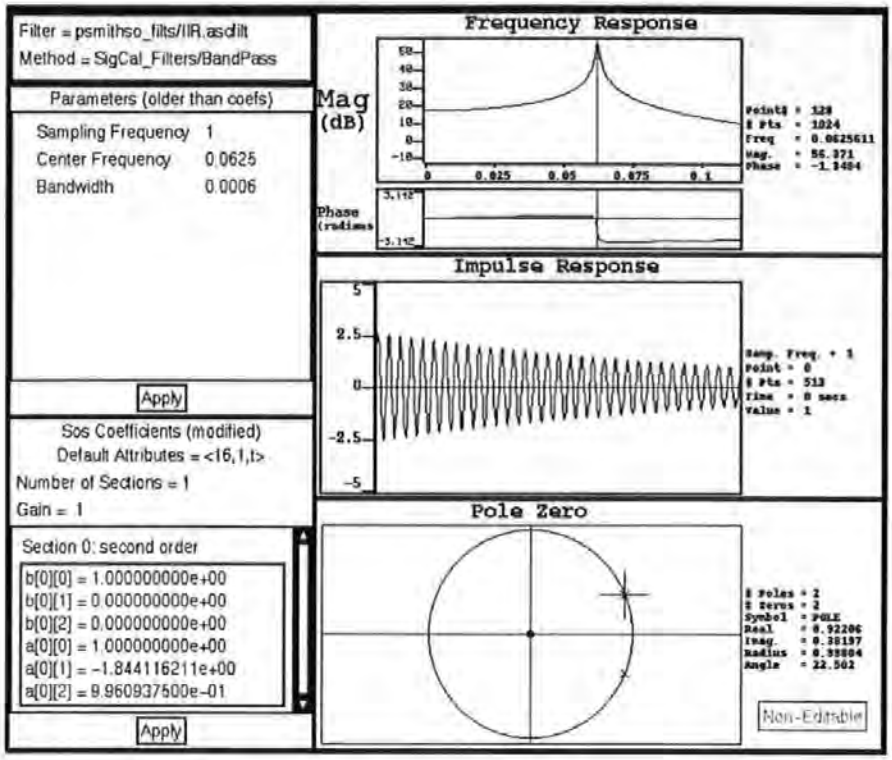


Figure 2-34 IIR Filter quantized to 16 bits

The best-case input signal is phase-reversal symbols, which cause maximum stimulus to the IIR filter. The numeric values chosen to stimulate the filter are ±45 (logic 1 = +45 and logic 0 = -45), resulting in the amplitude growing to just under $\pm 3.27E^4$, for a phase-reversal input. For a 16 bit word length, the dynamic range is $\pm 2^{15} = 3.2768E^4$, where the most significant bit is the sign bit, in 2's complement arithmetic. SPW simulations operate with double precision arithmetic; however results show that the filter amplitude has been restricted to operate within the dynamic range of $\pm 3.2768E^4$.

FDS on the other hand, has provision for quantizing coefficients and this has been used to determine the filter's frequency response and stability.

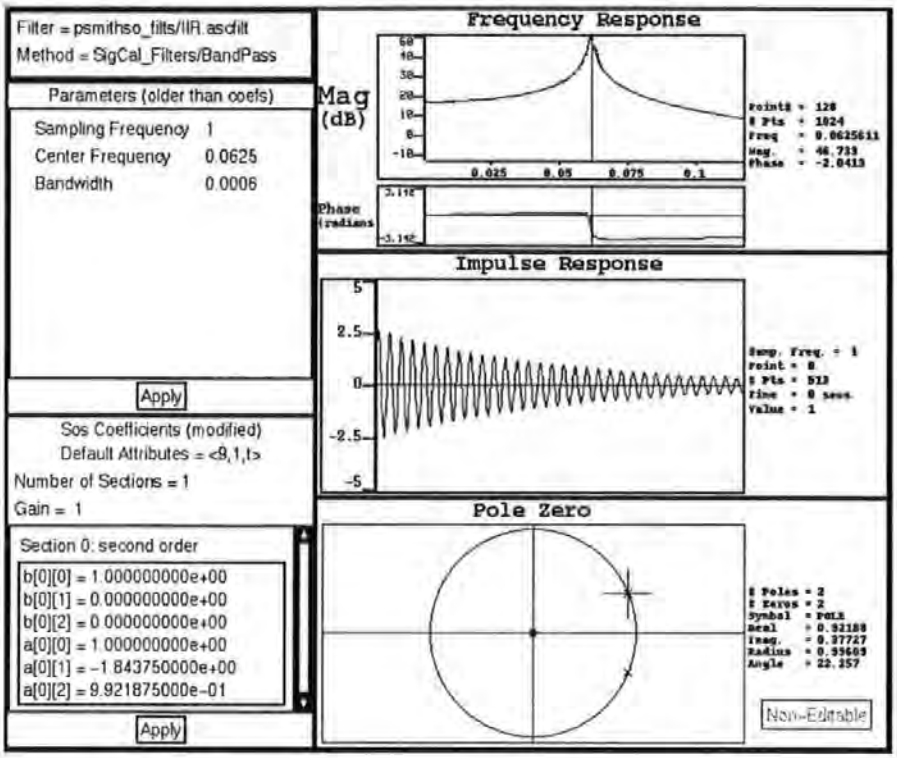


Figure 2-35 IIR Filter quantized to 9 bits

The FDS simulation of Figure 2-35 shows the filter response, when the filter coefficients are quantized to 9 bits. It can be seen from the nominal overlay (f_{nom} = 0.0625Hz), that the filter centre frequency has de-tuned (f_c = 0.06125Hz) by approximately 2%. This error is unacceptable, since the filter's amplitude would fail to grow to a sufficient level, so as to flywheel over data dropouts. This frequency error is corrected by appropriate selection of the sample rate, to re-tune the filter back to the required centre frequency. Quantization to 12 bits limits the frequency error to less than 1%, which is within the filter's bandwidth.

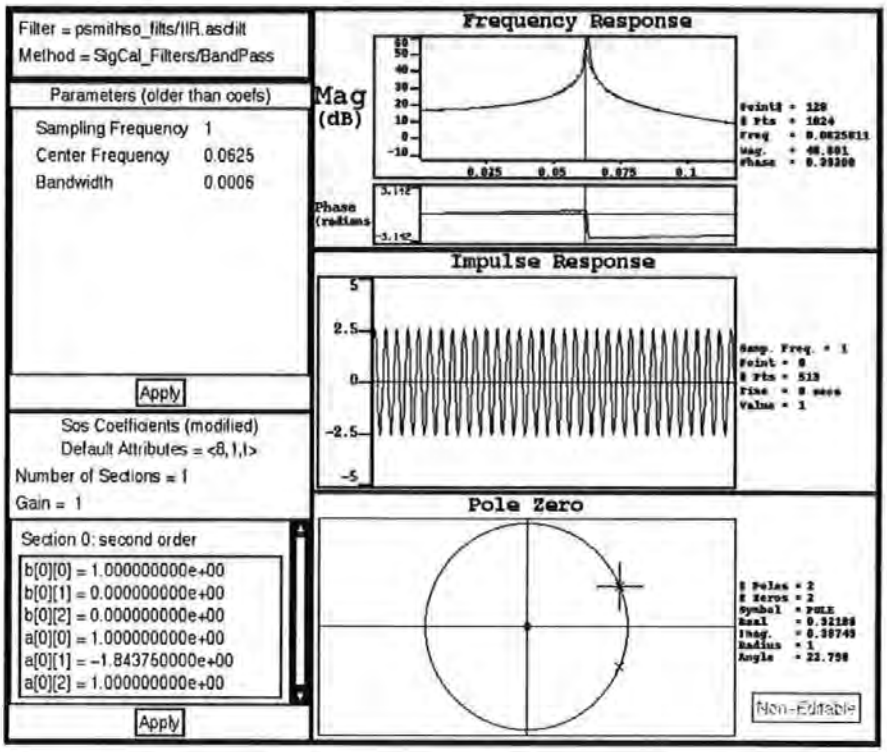


Figure 2-36 IIR Filter quantized to 8 bits

If the coefficients are quantized even further to just 8 bits, the poles lie on the unit circle and the filter becomes unstable, as demonstrated in the FDS simulation of Figure 2-36.

2.5.2. Application

The University of Plymouth operates live interactive distance learning TV programmes, from their TV studio to remote students within the U.K. and Europe. These tutorials are delivered via satellite using FM (Frequency Modulated) television. There was a requirement to deliver to these students, computer disk files, such that the presenter could provide course notes or interactively discuss computer-related problems (such as spreadsheets, programming, etc.). The author has developed a satellite data broadcasting modem, where computer binary files are transmitted utilising a sub-carrier on an analogue FM TV signal.

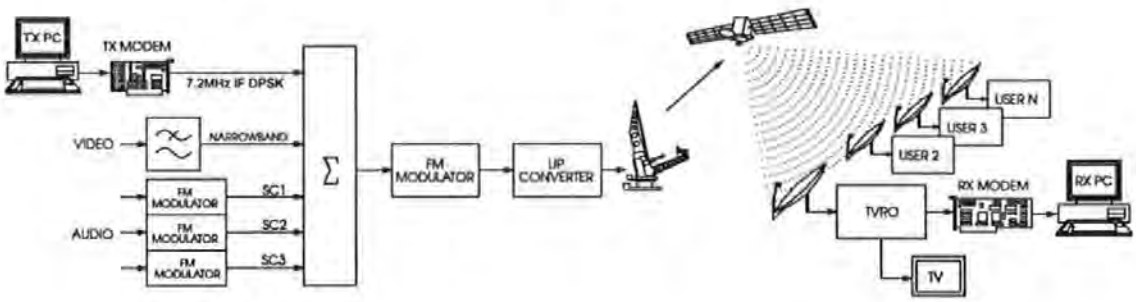


Figure 2-37 Satellite data broadcasting system

The data sub-carrier is modulated using DBPSK, onto a 7.2MHz I.F. (Intermediate Frequency) and transmitted as part of an FM TV broadcast. The remote students receive the satellite TV broadcast, using a domestic satellite TVRO (TeleVision Receive Only) system. The video and audio are connected to a TV, and tutorial presentations are viewed in the normal way. In addition, the standard decoder output from the TVRO receiver is connected to a specially designed data receive modem board, as shown in Figure 2-38, which is located within the student's PC (Personal Computer).

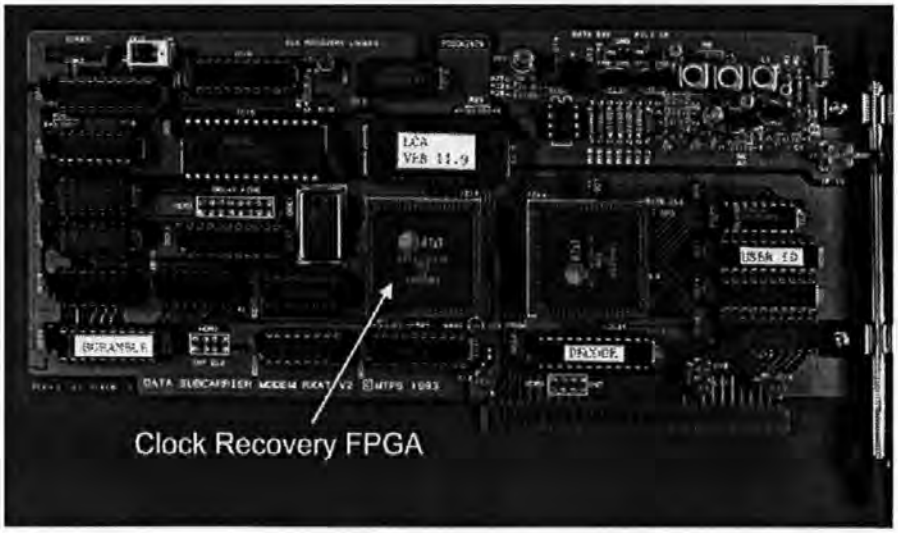


Figure 2-38 Sub-carrier receive modem

Data at 100Kbits/sec are digitally demodulated and clock recovery is performed, using the IIR filter algorithm implemented in an FPGA, as described in Section 2.5. Given the fixed binary weighted coefficients, a sample-rate of 2.59MHz fine-tunes the IIR filter to the desired centre frequency of 100KHz (100Kbits/sec). Since the FPGA is capable of operating at high sample rates, it is desirable to operate with a large number of samples/symbol, thereby minimising residual clock jitter. In this design, 26-samples/bit period is used.

A second FPGA provides word synchronisation and a buffered interface to the PC. The recovered binary words are read into the PC's memory at high speed using DMA (Direct Memory Access). A Windows (3.1 or 95) software application provides a user interface. The PC software logs received files, whereupon they can then be viewed or launched. For further information on this application, refer to the author's published paper, [Appendix 11{4}].

Chapter 3

3. Adaptive clock recovery

The results in Chapter 2 show that a clock recovery algorithm, with a fixed frequency IIR filter, provides rapid acquisition and good flywheel performance over long symbol run lengths and data dropouts. This algorithm has proven successful for clock recovery in burst mode applications, where rapid acquisition is essential.

This fixed frequency system does however rely on an accurate and stable symbol rate, at which the centre frequency of the IIR filter is placed. With a filter Q = 100, the IIR filter can accommodate ±1% drift of the input frequency, and for many applications this is adequate.

However, other data systems may experience mechanical limitations, which can cause much larger symbol rate frequency variations. A magnetic disk drive may experience drive-motor servo errors and the disk may spin at a slightly slower speed, resulting in a constant but reduced symbol rate. An even harsher environment is a longitudinal magnetic tape drive, where the tape can suffer from stretch as well as mechanical fluctuations. In such cases the retrieved symbol rate frequency can drift outside the bandwidth of the fixed IIR filter, resulting in loss of clock recovery. In such applications, a clock recovery system must be capable of tracking symbol rate frequency variations.

Ideally, the advantages of rapid clock acquisition and good flywheel performance, demonstrated by the fixed frequency filter should be maintained. If the precise symbol rate frequency can be sensed, then the filter coefficients can be updated, to reposition the centre frequency of the clock recovery filter, to track these input frequency variations. This concept is illustrated in Figure 3-1.

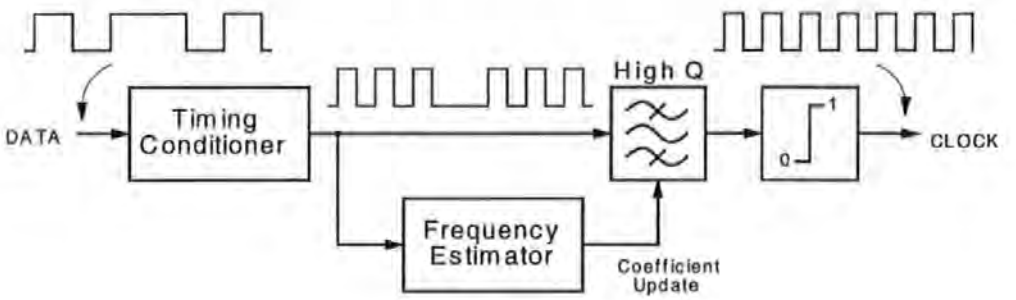


Figure 3-1 Concept of adaptive clock recovery

Conventional adaptive IIR filters, using for example the LMS algorithm [112], are unsuitable because the error surface is usually non-quadratic and multi-nodal, with respect to the filter coefficients. Hence, the coefficients can get stuck in local minima and fail to converge to the global minima. Genetic search algorithms [113] can be used to reduce this affect, but the wasted time searching along incorrect routes result in slow convergence speed and high computational complexity.

A novel adaptive system has been investigated, based upon two IIR detection filters, placed symmetrically around the nominal input frequency. The output of the detection algorithm generates an error control signal, which is used to apply new coefficients, to adjust the centre frequency and gain of an IIR clock-tracking filter.

Theoretical analysis has been applied, for an acquisition and tracking range of up to ±10% from the nominal centre frequency. Analysis has resulted in the development of a symmetrical tracking scheme, for both positive and negative frequency offsets. Mathematical expressions have been derived for the frequency estimator, to represent the coefficients as a function of the error control signal.

Simulation results are analysed for the adaptive clock recovery algorithm, to determine clock acquisition, the number of symbols needed to adapt and tracking performance. Simulations have been performed, initially for a sinusoidal input, resulting in further development for use with rectangular pulses.

Simulation results have demonstrated improvements over that of a basic PLL, both in terms of acquisition time and range, relative to flywheel performance. A PLL with a narrow loop bandwidth restricts the acquisition range to less than ±1%, and requires a training sequence of several hundred symbols to achieve lock. Whereas the DSP-based adaptive system provides an acquisition range of up to ±10%, and recovers a clock upon receipt of the first symbol transition. Also, the adaptive system demonstrates good clock flywheel performance, maintaining clock synchronisation over a dropout of 80 symbol periods.

3.1. System description

The adaptive clock recovery system is based upon two high-Q detection filters operating at ¼ of the system sample rate. These detection filters have a wider bandwidth than the tracking filter and are placed symmetrically around the nominal input (symbol rate) frequency. One is placed at a low frequency (f_L) and the other placed at a high frequency (f_H) with equal spacing (f_Δ) from the nominal (f_c).

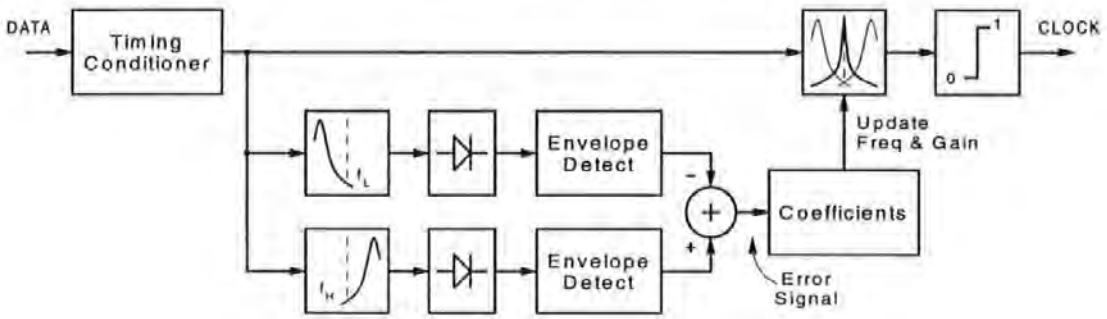


Figure 3-2 Adaptive clock recovery system

The outputs of the detection filters are full-wave rectified and averaged, to detect the filter's positive amplitude envelopes. These dc levels are then subtracted to produce an error signal, which is proportional to the frequency offset of the input signal (the symbol rate). If the input signal is at its nominal frequency, then both detection filters produce the same output amplitude and the error signal is zero. If however, the input signal has moved to a frequency above the nominal, then the signal will have moved further into the band of the higher detection filter and further out of the band of the lower detection filter. Hence, the oscillations of the higher detection filter will be of greater amplitude than that of the lower detection filter, and the resulting imbalance provides a corresponding error signal.

Similarly, if the input signal has moved to a frequency lower than the nominal, then a negative error signal will reflect the change in direction. The error signal is then used to calculate new coefficients, to move the centre frequency of the high-Q tracking filter to the new frequency of the input signal. Changing the centre frequency of the tracking filter results in a change in its amplitude. Hence, the error signal is also used to control gain compensation, maintaining a constant amplitude across the entire tracking range.

The high-Q tracking filter is exploited, to provide rapid clock acquisition and a good flywheel effect over data dropouts. Initially, the tracking filter may be offset (off-tune) from the input frequency, however, as long as there is sufficient stimulus to excite the tracking filter, then an output clock is produced. At this point, the filter output is low in amplitude and the system is vulnerable to long runs of the same encoded symbol. The system would therefore benefit from a short preamble (of say 6 phase reversal symbols), to stimulate the filter sufficiently to flywheel over this period.

As the tracking filter adapts to the input frequency, the stimulus becomes stronger and the filter output grows in amplitude, becoming less susceptible to encoded run lengths or data dropouts. When the system has fully adapted, the tracking filter is placed at the frequency of the input signal and exhibits the same desirable characteristics as for the fixed frequency filter, discussed in Chapter 2.

3.1.1. Tracking filter gain

A frequency estimator is used to re-tune the tracking filter to the correct input frequency. From equation (2-8), the Q of the tracking filter becomes:

$$Q_c = \frac{\pi \frac{f_c}{f_s}}{1-\sqrt{K_c}} \qquad \text{..... (3-1)}$$

A change in the centre frequency (f_c) produces a change of the filter Q, and a corresponding change in amplitude. Figure 3-3 shows the tracking filter swept across the frequency band. The change in amplitude can be approximated by a straight-line function over the region of interest, given by:

$$Y(f) = -10000\frac{f}{f_s} + 1300 \qquad \text{..... (3-2)}$$

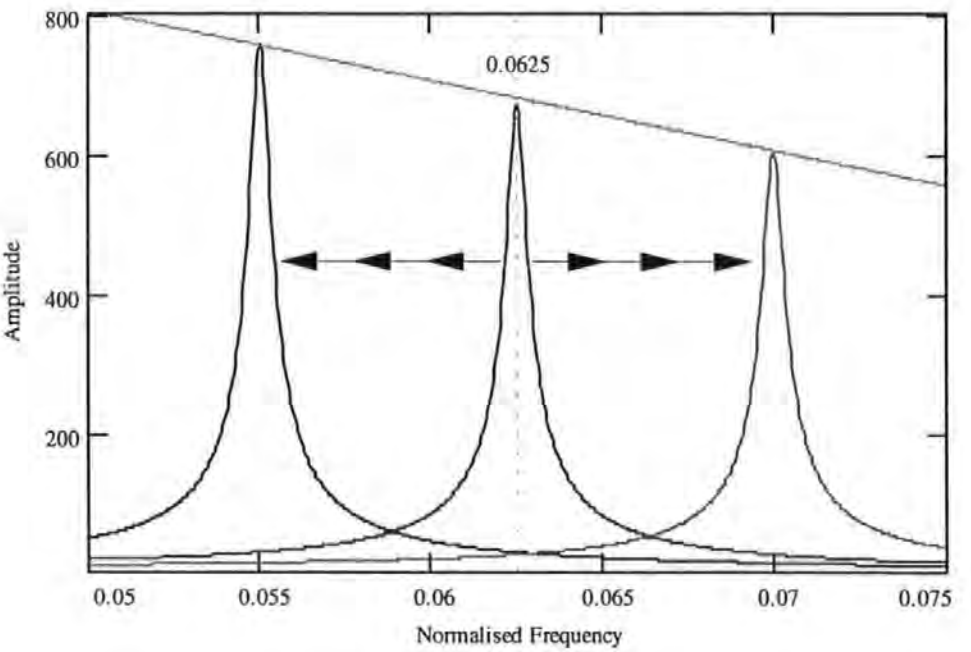


Figure 3-3 Tracking filter with no amplitude compensation

Given a frequency f, amplitude compensation can be applied, given by the function:

$$A(f) = \frac{(100 \cdot f_c) - 13}{(100 \cdot f) - 13} \qquad \text{..... (3-3)}$$

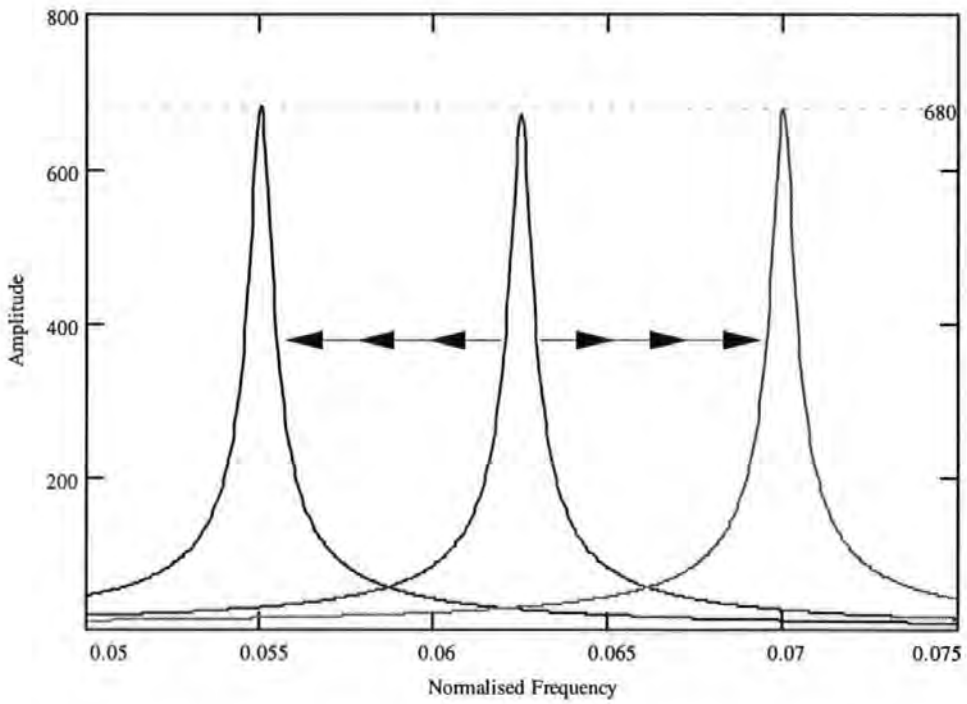


Figure 3-4 Tracking filter with amplitude compensation

The nominal frequency ($f_c = 0.0625$) has unity gain, while the applied amplitude compensation provides either gain for positive offsets, or attenuation for negative frequency offsets. Figure 3-4 shows the resulting constant amplitude across the tracking range.

3.1.2. Detection filter symmetry

It is important that the two detection filters have identical amplitude and frequency responses, in order to produce an accurate and symmetrical error control signal. Therefore, the positive frequency skirt of the lower detection filter must be an exact mirror image of the negative frequency skirt of the upper detection filter.

However, there are two problems in achieving this:

- The positive and negative skirts of the IIR filter have different rates of roll-off. This is particularly apparent with reference to the FDS analysis of the IIR filter, as shown in Figure 2-13.
- The Q of the detection filters, and hence amplitude, varies depending upon the centre frequency, since $Q = \frac{f_c}{BW}$.

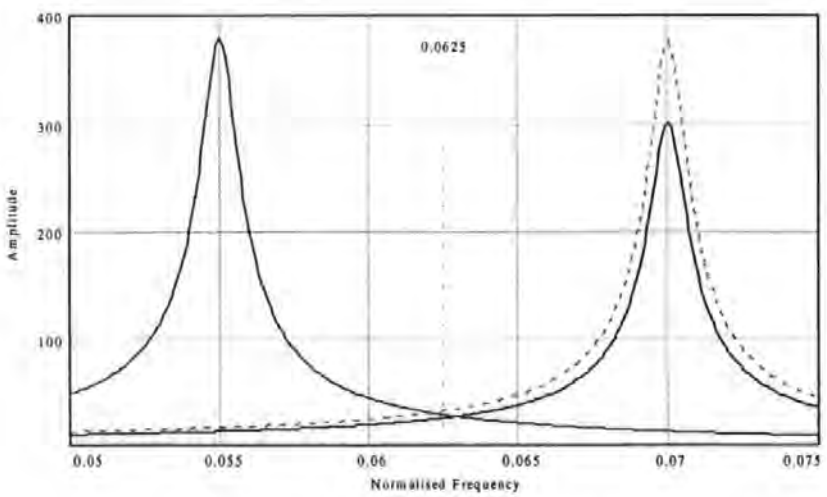


Figure 3-5 Response of non-symmetrical detection filters

Simply choosing coefficients to place detection filters, low $H_L(f)$ and high $H_H(f)$, results in uneven amplitude responses, as shown in Figure 3-5. The high filter can

be provided with gain, to peak its amplitude response to match that of the low filter, but the uneven roll-off skirts still result in a non-symmetrical response. Hence, the positive skirt of the low filter and the negative skirt of the high filter do not converge exactly at the nominal frequency of the tracking filter (f_c = 0.0625Hz).

Mathematically, the response of the low detection filter can be mirrored about the axis of the nominal-tracking filter, to give the desired response.

For example, given the low filter's transfer function $H_L(f)$ and a centre frequency f_c, a mirror image of $H_L(f)$ can be produced about the axis f_c to form a function $H_H(f)$.

$$H_H(f_c + \Delta) = H_L(f_c - \Delta) \qquad \text{for all values of } \Delta.$$

let $f = f_c + \Delta$ then $\Delta = f - f_c$

$$H_H(f_c + \Delta) = H_H(f)$$

$$H_L(f_c - \Delta) = H_L(f_c - (f - f_c))$$

$$H_H(f) = H_L(2f_c - f)$$

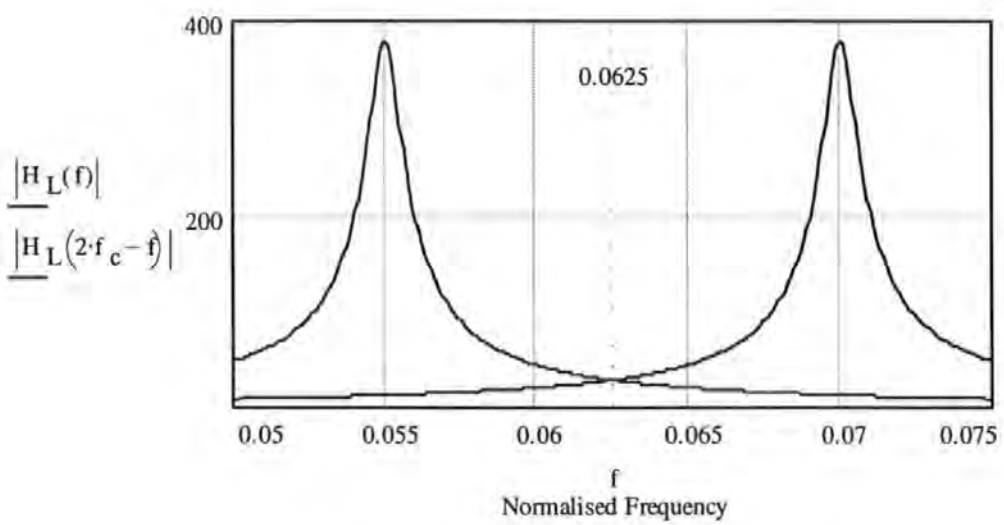


Figure 3-6 Ideal mirrored detection filters

However, this mathematical manipulation cannot be realised in practice and therefore an alternative scheme has been devised.

In analysing the response of the IIR filter coefficient (C), a cosine function is produced, spanning from dc to the sample rate ($f = 1$), given by equation (2-12) and plotted in Figure 3-7.

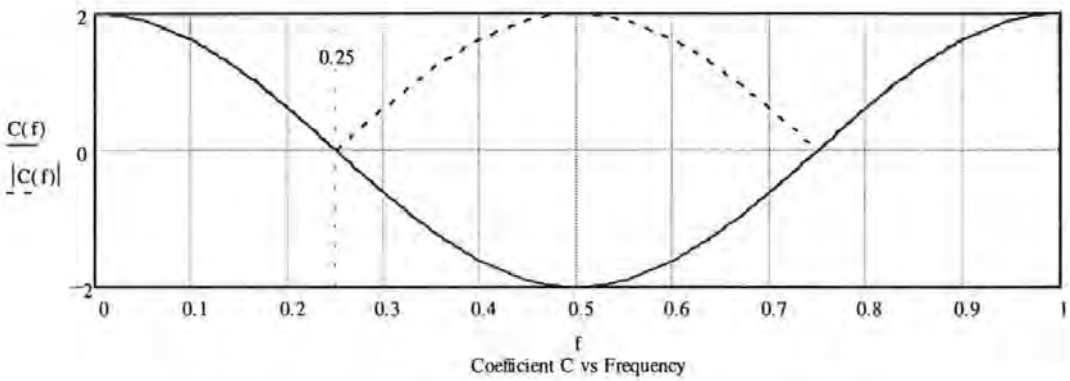


Figure 3-7 Cosine function of filter coefficient

It is interesting to observe that the cosine function has a null at 0.25 of the sample rate. The cosine curve between 0 and 0.25 is identical to that of 0.25 to 0.5, with the exception of a change in sign, from positive to negative. If the modulus of this cosine function is applied, then identical coefficients exist around 0.25 of the sample rate (and also at 0.75 of the sample rate). Hence, low and high detection filters can be placed above and below this frequency, which have identical coefficients (except for a sign change), thereby providing identical frequency and amplitude responses.

Low detection filter $f_L = f_c - \Delta$ and

High detection filter $f_H = f_c + \Delta$

The nominal centre frequency of the tracking filter (f_c = 0.0625Hz) is placed at this ¼ of the sample rate frequency, which is exactly where the identical responses of the detection filters skirts converge.

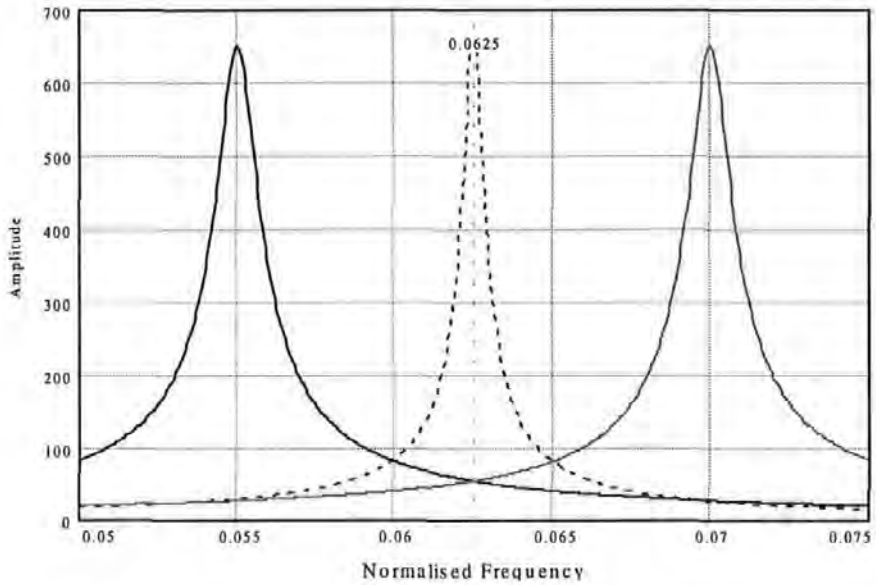


Figure 3-8 Symmetrical detection filters

To express the frequencies of the detection filters using the same frequency scale as the tracking filter, the sub-sample rate becomes $f_{sub} = \frac{f_s}{4}$.

From equation (2-12) the detection filter coefficients are given as:

Low Filter $$C_L = \sqrt{K_L} \cdot 2\cos\left(\frac{2\pi f_c}{\frac{f_s}{4}}\right) \quad \text{.....} \quad (3\text{-}4)$$

High Filter $$C_H = -C_L$$

A system has been designed using two sample rates. The tracking filter is, as usual, sampled at 16 samples/symbol, thereby minimising residual jitter in the recovered

clock, while the detection filters are sampled at $\frac{f_s}{4}$ to provide symmetrical frequency and magnitude responses, as plotted in Figure 3-8.

From equation (2-5) the transfer function of the detection filters become:

Low Filter $$H_L(f) = \left(1 - C_L e^{-j2\pi \frac{f}{\frac{f_s}{4}}} + K_L e^{-j4\pi \frac{f}{\frac{f_s}{4}}} \right)^{-1} \quad \ldots\ldots\ldots\ldots \quad (3\text{-}5)$$

High Filter $$H_H(f) = \left(1 - C_H e^{-j2\pi \frac{f}{\frac{f_s}{4}}} + K_H e^{-j4\pi \frac{f}{\frac{f_s}{4}}} \right)^{-1} \quad \ldots\ldots\ldots\ldots \quad (3\text{-}6)$$

The detection filters have identical bandwidths, and automatically adjust for a different Q, given by:

Low Filter $$Q_L = \frac{\pi \frac{f_L}{\frac{f_s}{4}}}{1 - \sqrt{K_L}} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (3\text{-}7)$$

High Filter $$Q_H = \frac{\pi \frac{f_H}{\frac{f_s}{4}}}{1 - \sqrt{K_H}} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (3\text{-}8)$$

where the second coefficient $K_L = K_H$ controls the filter bandwidths. The Q of the detection filters is lower than that of the tracking filter, so as to provide a wide detection range. Hence, the detection filters are provided with gain to provide a similar magnitude to that of the tracking filter, as plotted in Figure 3-8.

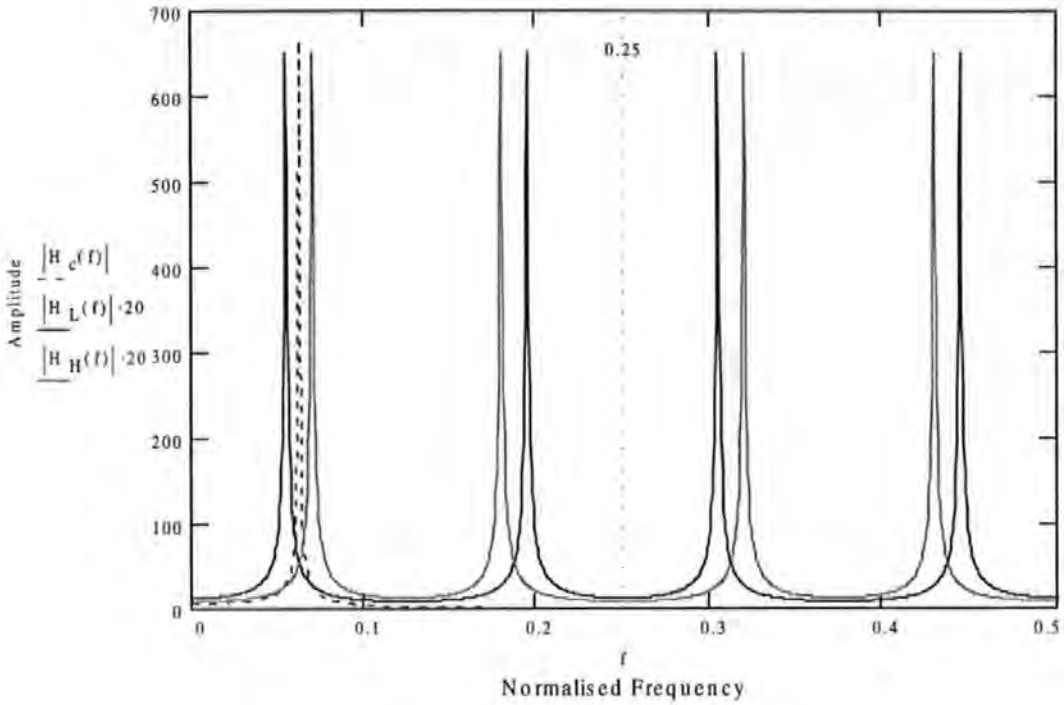


Figure 3-9 Repeat spectrum of detection filters

Figure 3-9 shows the frequency response of the tracking filter up to the Nyquist frequency *(f = 0.5)*. The response of the detection filters repeat around multiples of f_{sub}. Given that $f_{sub} = \frac{f_s}{4}$ and $f_s = 1$, then the spectrum repeats at 0.25, 0.5, 0.75, etc.

The required response, centred at 0.0625Hz (around 0Hz), repeats around 0.25Hz at 0.25 + 0.0625 = 0.3125Hz, the negative frequency image of which occurs at 0.25 – 0.0625 = 0.1875Hz. Similarly, the spectrum repeats again at 0.5Hz, the negative frequency image of which can be seen at 0.5 – 0.0625 = 0.4375Hz. The tracking filter response does not repeat around these frequencies, since it is sampled at $f_s = 1$. If the applied input signal comprises of rectangular pulses, then odd harmonics falling within the repeat spectra will alias back into the detection system, corrupting the error control signal. For this reason, rectangular pulses are

low-pass filtered to remove odd harmonics, before application to the detection system.

With reference to Figure 3-7, an observation worth noting is that a fixed filter placed at 0.25 f_s has a coefficient C = 0. For this special case, the filter architecture can be simplified as shown in Figure 3-10.

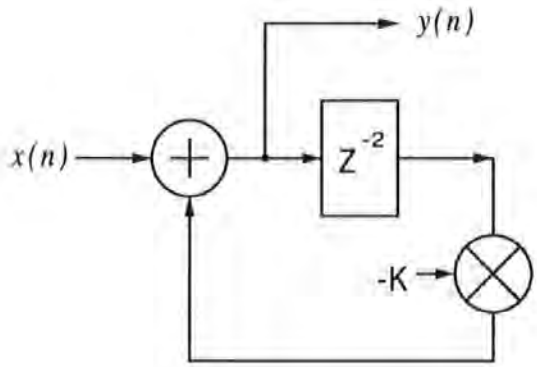


Figure 3-10 Simplified filter architecture

From equation (2-5) and with coefficient C = 0, the transfer function becomes:

$$H(f) = \left(1 + K\, e^{-j4\pi \frac{f}{f_s}}\right)^{-1} \qquad \text{..... (3-9)}$$

The sample rate of the filter is $f_s = 4 f_c$ and may be adjusted to tune the filter to the desired centre frequency. However, a sample rate of just 4 samples/cycle makes reconstruction of an analogue signal more difficult. For clock recovery applications it is desirable to maintain a large number of samples/symbol to reduce jitter in the recovered clock, therefore this architecture has not been adopted.

This simplified filter architecture results in the reduction of silicon utilisation within an FPGA or ASIC design and may be of benefit for bandpass filter applications.

3.1.3. Tracking control - Sine Input

The modulus output from the IIR detection filters are envelope detected and subtracted to produce an error control signal, which is proportional to the frequency offset. The error signal can then be applied to a function, or a look-up table, to determine the necessary parameters to re-tune the tracking filter to the correct input frequency.

The modulus of the detection filter's output is obtained by inverting negative value samples. This is a simple matter of inverting the MSB (Most Significant Bit), the sign bit in 2's complement. Averaging over a block of samples then filters the resulting positive cycles, producing a dc error control signal. For a periodic input, the averaging need only be over 4 symbols, but for data the block size must be increased, to average amplitude variations due to data run lengths. For symbols with a maximum run length of 7 (RLL7), the averaging block is increased to 32 symbols, although this can be reduced, for codes with shorter run lengths.

To determine an error signal calibration curve, a series of simulations have been performed to measure the error signal relative to frequency offset. Initially, results are obtained for a sinusoidal input over the tracking range ±10%, using detection filters with a Q=100 (Coefficient K = $^{31}/_{32}$). Due to the symmetrical detection filters, identical readings are obtained for both positive and negative offsets, except for a change in the sign, indicating the direction of the frequency offset.

The simulated error calibration curve is plotted in Figure 3-11, where e_n is given by:

$e_n = \pm(0, 373, 764, 1207, 1742, 2416, 3370, 4818, 7352, 12189, 17617)$

For convenience the horizontal scale is converted to percentage frequency offset, over the tracking range $n = \pm 10\%$.

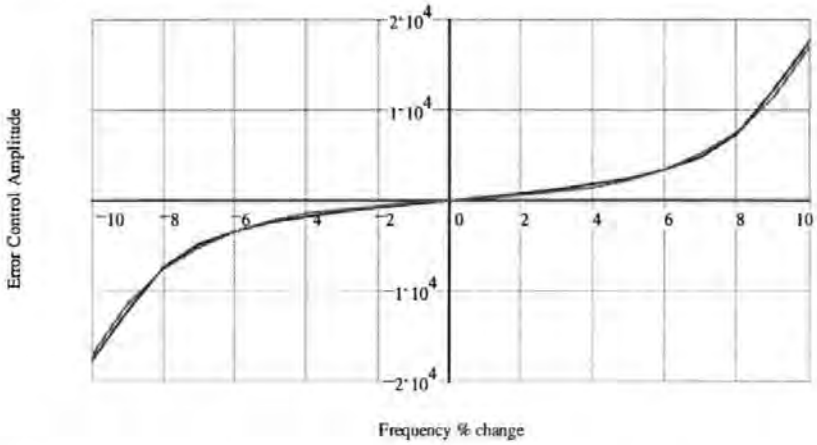


Figure 3-11 Error calibration over ±10% frequency offset

A function E_n is then determined to fit the simulated results, given by:

$$E_n = 17000 \cdot \frac{\sinh(0.4 \cdot n)}{\sinh(4)} \qquad \ldots\ldots\ldots\ldots\ldots \quad (3\text{-}10)$$

Figure 3-11 shows this function overlaid onto the simulated results.

The error curve may be expressed as a function of frequency f.

Given that the nominal centre frequency $f_c = 0.0625$ and the tracking frequency step size is $f_\delta = 0.000625$, then $f = f_c + n f_\delta$ and $n = \frac{f - f_c}{f_\delta}$.

Substituting for n into equation (3-10) yields:

$$E(f) = 17000 \cdot \frac{\sinh\left(0.4 \cdot \frac{f - f_c}{f_\delta}\right)}{\sinh(4)} \qquad \ldots\ldots\ldots\ldots\ldots \quad (3\text{-}11)$$

The error calibration curve is then plotted in terms of normalised frequency, as shown in Figure 3-12.

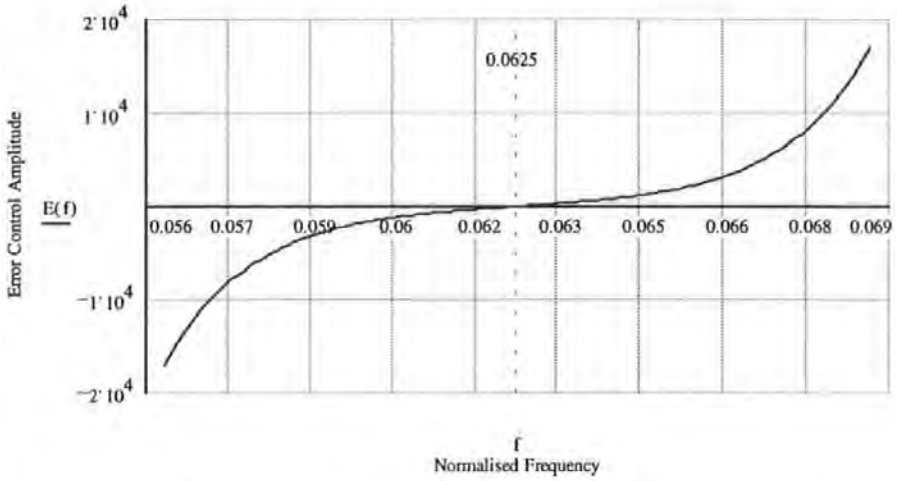


Figure 3-12 Error calibration as a function of frequency

It is now required to determine functions to express the Coefficient *(C)* and Gain *(A)* as a function of the error control signal *(E)*. These functions are applied to the tracking filter to re-tune the centre frequency to that of the input signal, and to apply the appropriate gain compensation to maintain a constant amplitude.

Rearranging equation (3-11) in terms of f:

$$E(f) = 17000 \frac{\sinh\left(0.4 \cdot \frac{f - f_c}{f_\delta}\right)}{\sinh(4)}$$

$$\sinh\left(0.4 \cdot \frac{f - f_c}{f_\delta}\right) = \frac{\sinh(4) \cdot E}{17000}$$

$$0.4 \cdot \frac{f - f_c}{f_\delta} = a\sinh\left(\frac{\sinh(4) \cdot E}{17000}\right)$$

$$f = \frac{f_\delta}{0.4} \cdot a\sinh\left(\frac{\sinh(4) \cdot E}{17000}\right) + f_c \qquad \text{..... (3-12)}$$

From equation (2-12), Coefficient C becomes:

$$C = \sqrt{K} \cdot 2\cos\left(\frac{2\pi f}{f_s}\right) \qquad \text{..... (3-13)}$$

where f is the tracking frequency over the range 0.05625 to 0.06875Hz (±10%) and f_c is the nominal centre frequency (zero offset).

Substituting for f into equation (3-13) yields:

$$C(E) = \sqrt{K} \cdot 2\cos\left(2\pi \frac{\frac{f_\delta}{0.4} \cdot a\sinh\left(\frac{\sinh(4) \cdot E}{17000}\right) + f_c}{f_s}\right)$$

$$C(E) = \sqrt{K} \cdot 2\cos\left[2\pi\left(\frac{5f_\delta}{2f_s} a\sinh\left(\frac{\sinh(4) \cdot E}{17000}\right) + \frac{f_c}{f_s}\right)\right] \qquad \text{..... (3-14)}$$

The resulting function is plotted in Figure 3-13.

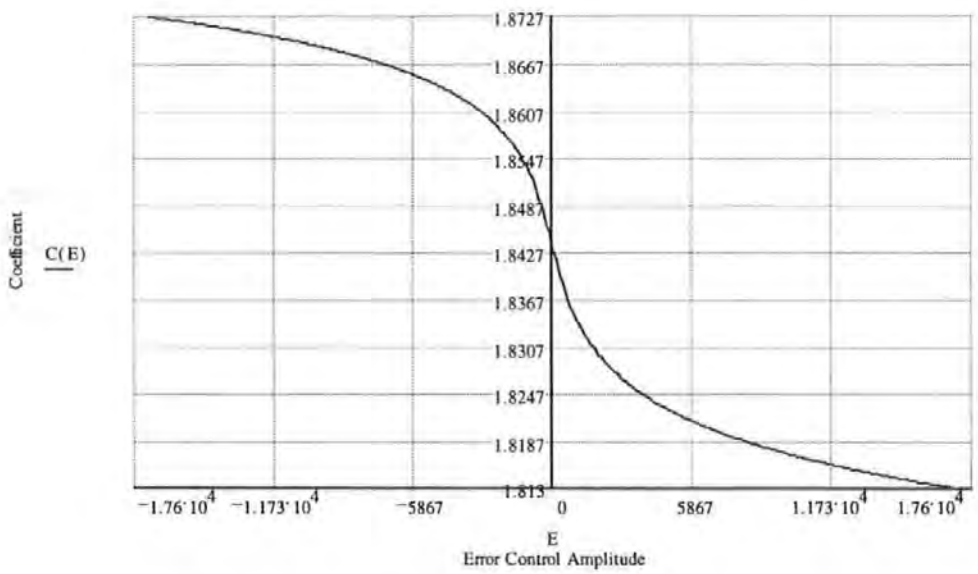


Figure 3-13 Coefficient vs. Error signal

A function must now be determined for the gain control.

Equation (3-3) expresses the gain control as a function of frequency:

$$A(f) = \frac{(100 \cdot f_c) - 13}{(100 \cdot f) - 13}$$

Given that the nominal centre frequency $f_c = 0.0625$ then:

$$A(f) = \frac{(100 \cdot 0.0625) - 13}{(100 \cdot f) - 13}$$

$$A(f) = \frac{-6.75}{(100 \cdot f) - 13}$$

Substituting equation (3-12) for f yields:

$$A(E) = \frac{-6.75}{100 \cdot \left(\frac{f_\delta}{0.4} \cdot a\sinh\left(\frac{\sinh(4) \cdot E}{17000} \right) + f_c \right) - 13} \quad \ldots\ldots\ldots\ldots \quad (3\text{-}15)$$

The resulting function is plotted in Figure 3-14.

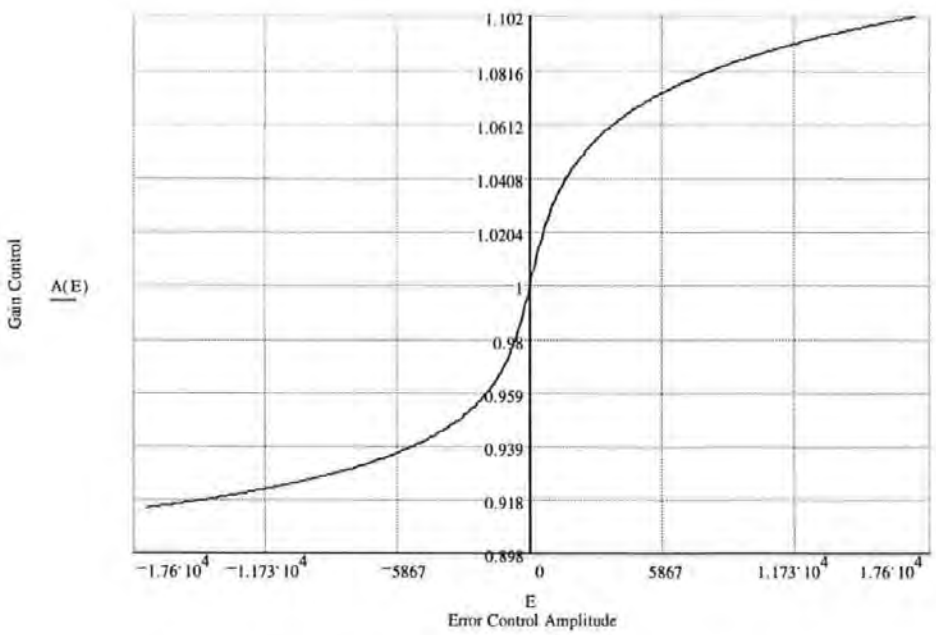


Figure 3-14 Gain control vs. Error signal

The functions for the coefficient and gain given by equations (3-14) and (3-15) can be applied to the adaptive clock recovery filter to track input frequency offsets. In a practical system, DSP processors do not contain trigonometric functions and therefore, the coefficient and gain must be applied by means of a look-up table.

The clock recovery filter can withstand a ±1% frequency offset, before the need to update the centre frequency. Therefore the error control signal is applied to a look-up table, divided into frequency bins with a 1% spacing. This is achieved by means of window comparators over the frequency range of ±10%. The tracking filter is therefore updated at 1% frequency increments, providing frequency hysteresis, thereby minimising short-term tracking variations due to noise.

The phase response of the IIR filter as shown in Figure 2-11, is relative to the frequency offset of the input signal. Due to the recursive nature of the IIR filter, short-term phase variations are averaged, minimising jitter in the recovered clock. While the filter is adapting to the input frequency, the phase of the recovered clock lags the input signal by a few samples for negative offsets, and leads for positive frequency offsets. When the tracking filter has fully adapted the phase of the recovered clock matches that of the input signal. At 16 samples/cycle these phase variations can be accommodated, however care must be taken to re-sample data at the centre of the data eye.

3.1.3.1. Test simulations – Sine Input

The adaptive system has been simulated, initially with a sinusoidal input, as shown in the illustration of Figure 3-15.

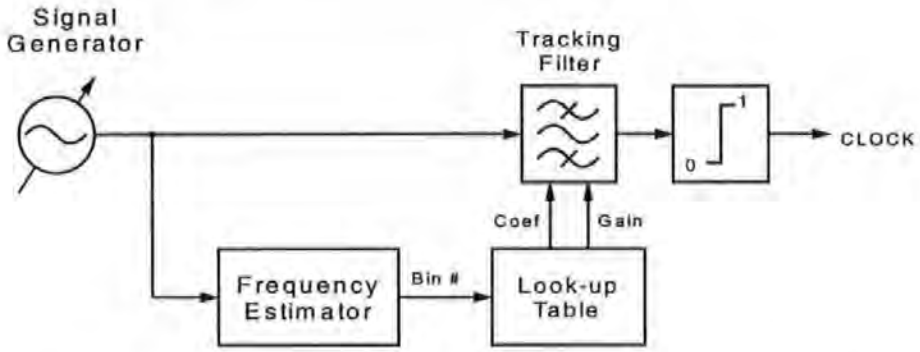


Figure 3-15 Overview of adaptive filter test model

The frequency estimator provides a bin number flag, indicating the currently applied frequency offset control (0 to ±10). The bin number is then used as a pointer in the look-up table to apply the appropriate coefficients and gain parameters to the tracking filter. When the system has fully adapted the bin number flag is stable and corresponds to the percentage offset from the nominal centre frequency.
Hence, the bin number flag is a converted quantized version of the error control signal and provides a convenient test point to measure system acquisition and tracking performance.

The simulation model, together with full simulation results for drifting frequency offsets, are included in Appendix 7.

Figure 3-16 provides an example simulation, indicating the tracking filter adapting to a +5% frequency offset. An off-tune fixed filter (f_c = zero offset) has a low amplitude falling to zero, while the amplitude of the adaptive filter steadily grows, as its centre frequency is progressively re-tuned, by means of the error control signal. The tracking filter takes just 7 input cycles to adapt to the input frequency, however the recovered clock is asserted within a single input cycle, indicating rapid clock acquisition.

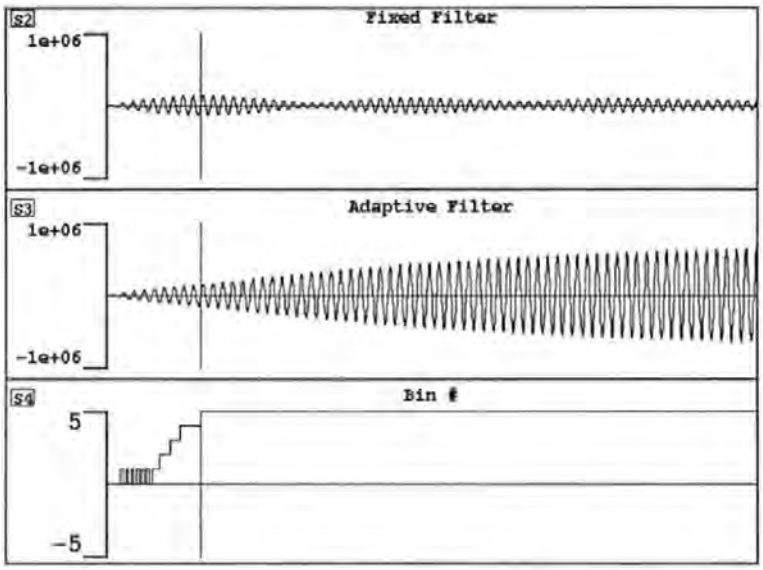


Figure 3-16 Tracking filter simulation - sinusoidal input

Figure 3-17 summarises a collection of simulation data to demonstrate the number of cycles taken for the tracking filter to adapt to an input frequency offset. Successful acquisition is rapidly achieved to within 1% of the input frequency, but the final 1% fine-tuning can take much longer as the frequency offset is increased. For a ±10% frequency offset, acquisition to 1% is achieved within 27 input cycles. However, due to averaging within the detection system, the final fine-tuning can take a further 43 cycles.

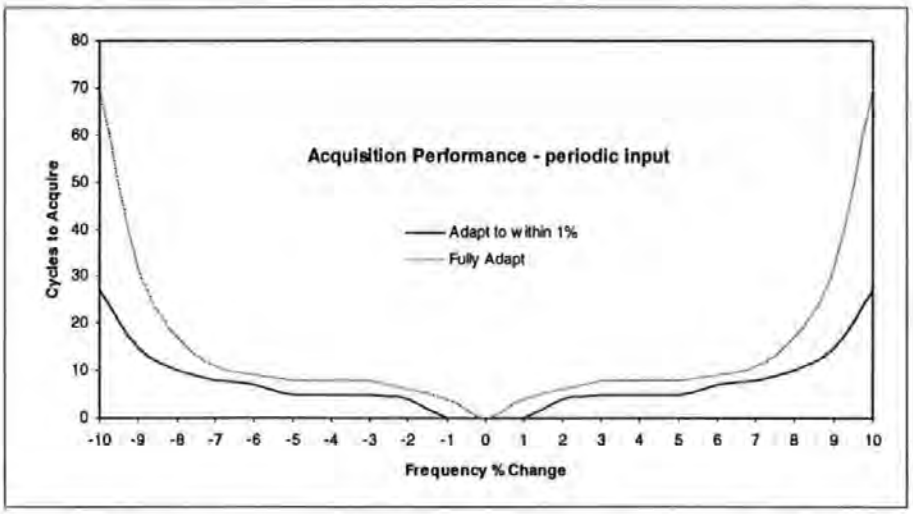


Figure 3-17 Acquisition performance for a periodic input

Since the tracking filter can tolerate a ±1% frequency offset, the final 1% fine-tuning is of little consequence to clock acquisition performance. The simulation of Figure 3-18 shows a filter pre-tuned to a 5% frequency offset, and then de-tuned by ±1%. Although there is a reduction in amplitude, the filter output is still strong, confirming tolerance to a ±1% frequency offset.

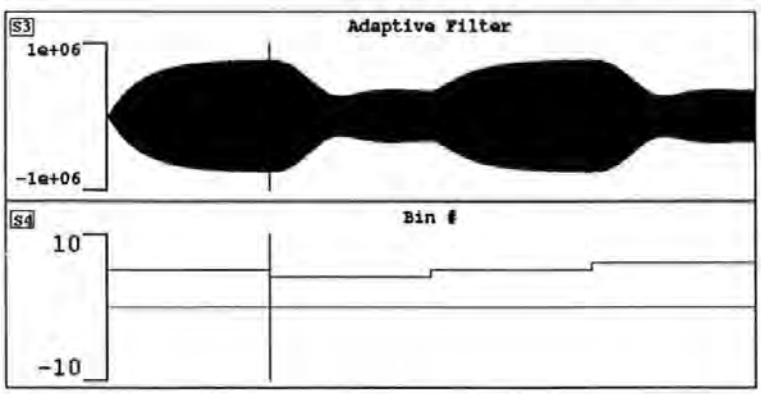


Figure 3-18 Tolerance to ±1% frequency offset

If the adaptive control is enabled, then the filter will track these frequency offsets and the amplitude will return to full strength.

3.1.4. Tracking control – Data input

The error calibration curve for a sinusoidal input demonstrates a smooth exponential error control signal, relative to an input frequency offset. For clock recovery applications, the retrieved signal comprise of non-periodic rectangular pulses and this gives rise to a number of problems.

Rectangular pulses contain odd harmonics, which alias back into the detection system and distorts the calibration of the error control signal.

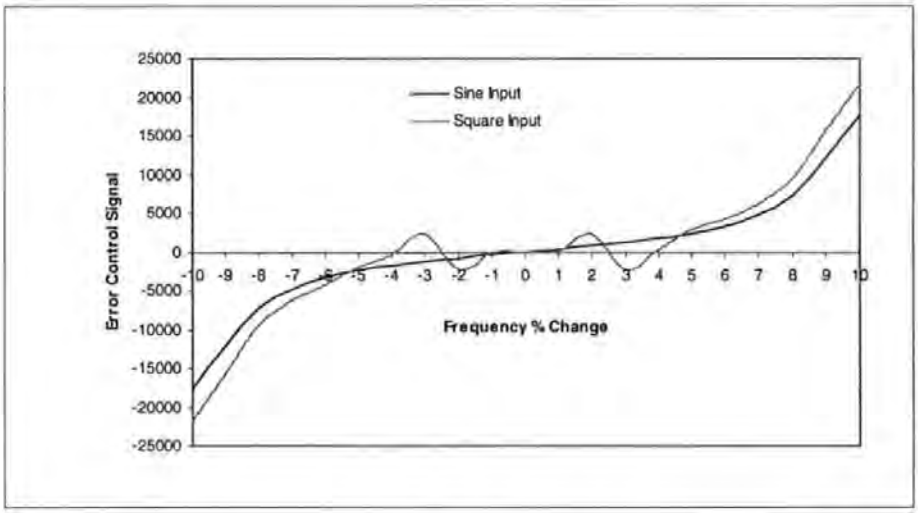


Figure 3-19 Error control for a signal generator input

The graph shown in Figure 3-19 is a collection of simulation data, demonstrating the effect of aliased harmonics from a square-wave input, corrupting the error control signal. This can be reduced by means of a simple 9-tap FIR low pass filter, to filter harmonics of the retrieved symbols, prior to application to the detection system, as shown Figure 3-20.

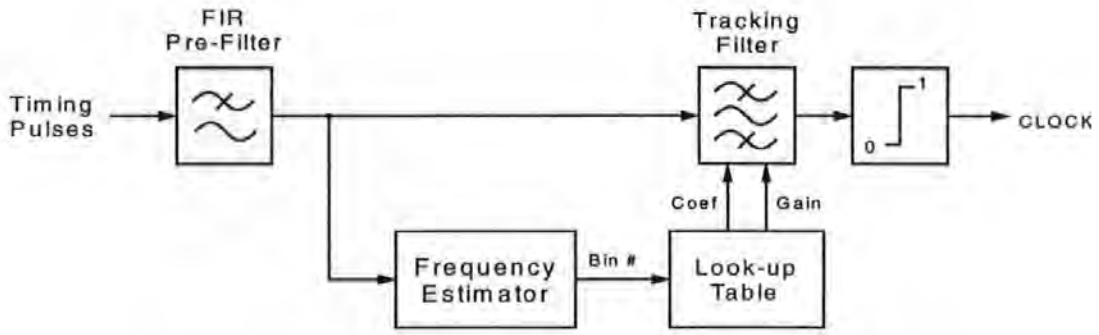


Figure 3-20 Adaptive filter for use with rectangular pulses

Figure 3-21 provides an FFT analysis, showing the frequency response of the pre-filter, containing a normalised –3dB cut-off frequency of 0.1Hz.

The coefficients for this filter are:

$b(0) = 0.0050699$, $b(1) = 0.029358$, $b(2) = 0.11074$, $b(3) = 0.21934$, $b(4) = 0.27097$, $b(5) = b(3)$, $b(6) = b(2)$, $b(7) = b(1)$, $b(8) = b(0)$.

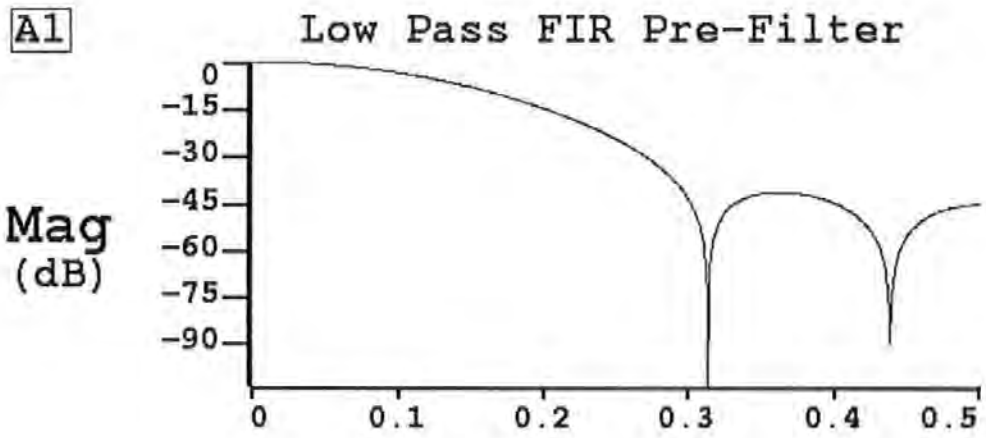


Figure 3-21 Pre-filter frequency response

The 3rd harmonic at 0.1875Hz is at –13dB and subsequent odd harmonics are below –40dB. A sharper cut-off frequency could be used to reduce further the 3rd harmonic, however care should be taken not to introduce ISI, which would result in closing of the data eye.

The simulated data of Figure 3-22 shows that the pre-filtered error control signal, for a square-wave input, closely approximates that of a sinusoidal input. Hence, the odd harmonics are suppressed and the integrity of the error control signal is restored.

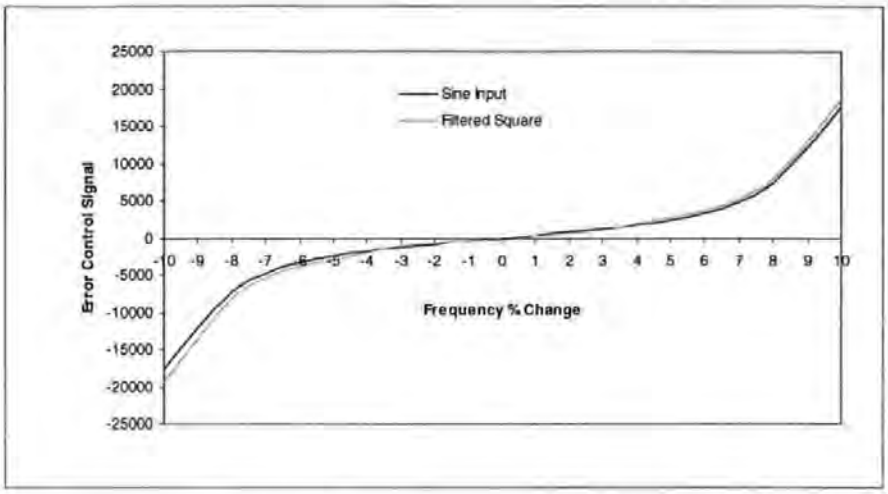


Figure 3-22 Error control signal with a pre-filter

The pre-filter has the added benefit of noise suppression and may also be used for input to the tracking filter, thereby reducing jitter in the recovered clock.

The eye diagram simulation of Figure 3-23 shows a large amount of gaussian noise applied to Manchester encoded data and the resulting eye diagram after passing through the pre-filter. It can be seen that the filtered eye is opened, reducing the probability of the recovered clock sampling errors due to noise.

In many applications, a matched input filter already exists for noise suppression purposes. Therefore, as long as the output signal from the matched filter is not converted to rectangular pulses, the filtered signal may be suitable for direct application to the adaptive clock recovery system, without the need for an additional pre-filter.

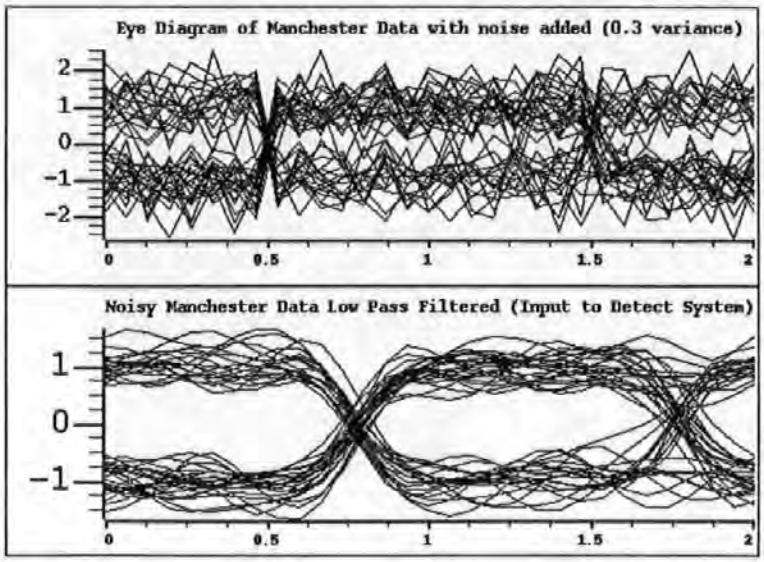


Figure 3-23 Eye diagram pre-filtered

The corresponding acquisition performance of the system is shown in Figure 3-24, demonstrating that the tracking system performs well, even in the presence of noise.

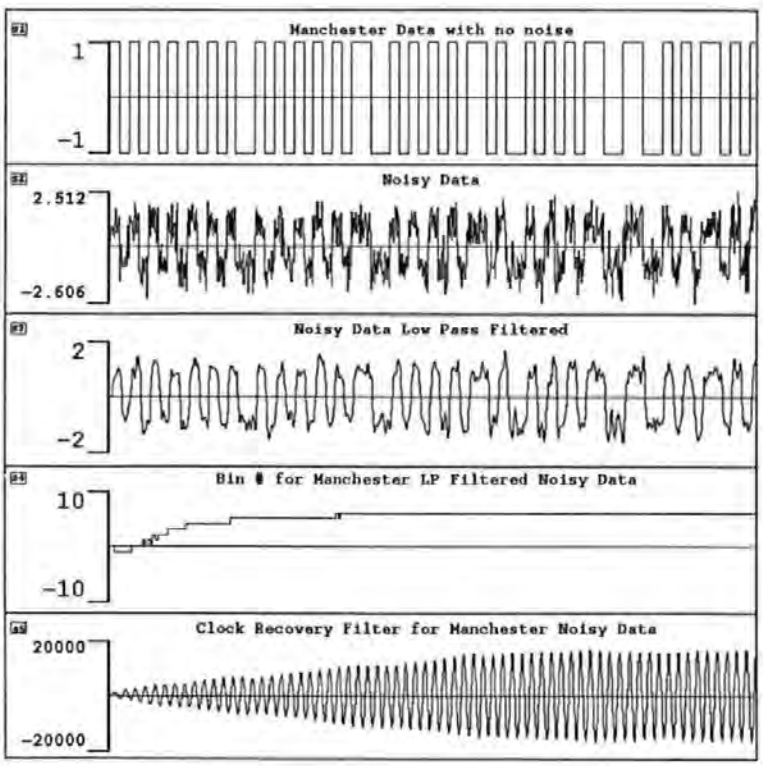


Figure 3-24 Acquisition performance with additive noise

Another problem introduced with data input is that encoded symbols are non-periodic. The run lengths produce amplitude variations at the output of the detection filters, which require greater averaging than for a periodic input. For data with a maximum run length of 7, the averaging block is increased to 32 symbols. Also, the greater the run length then the lower the averaged amplitude from the detection filters. Hence, the small amplitude signals from the detection filters become lost in noise, and cease to provide a meaningful error control signal. To overcome this problem the spacing of the detection filters are reduced proportionally. For Manchester data (containing a high clock content), the detection filters can continue to be placed at ±10% offsets, while for RLL3 and RLL7 encoded symbols the tracking range is reduced to ±7.5% and ±5% respectively, as plotted in Figure 3-25.

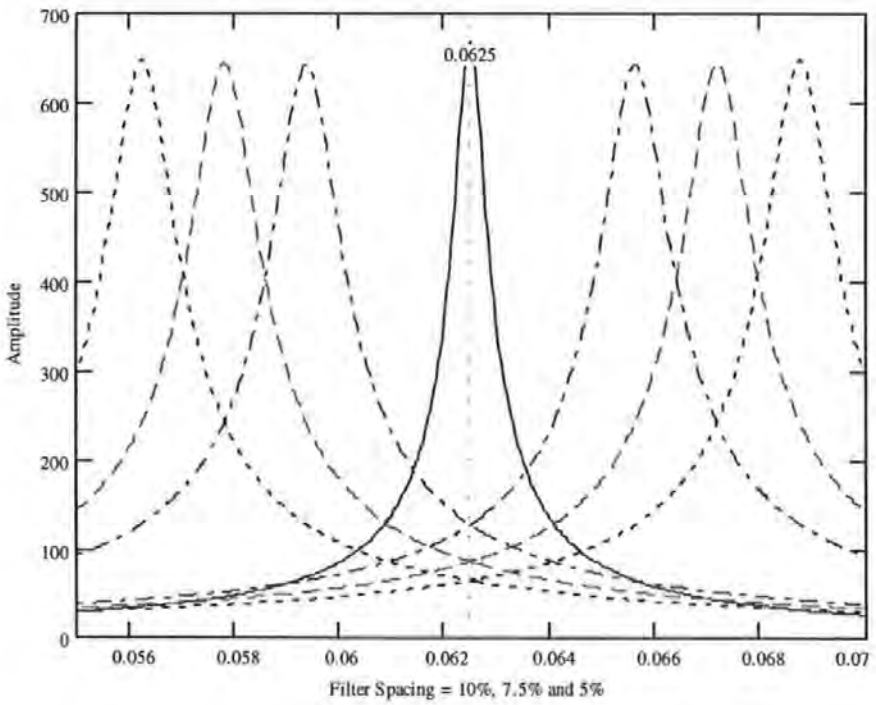


Figure 3-25 Reduction in detection filter spacing

Reducing the detection filter spacing provides a wider bandwidth overlap at the nominal centre frequency, and hence the input signal falls further into the bandwidth of the detection filters. If the average amplitude from the detection filters fall, due to encoded run lengths, then there is still sufficient signal strength so as to produce a reliable error control signal.

Changing the detection filter spacing also changes the error calibration, since for a given offset, the input signal is further in-band for a 5% spacing than for 10% spacing, resulting in an increased output amplitude. However, the run length of the encoded symbols also affects the amplitude of the detection filter outputs. Therefore, both the detection filter spacing and the encoded run length influence the error control signal. A corresponding error calibration is determined by simulation and the results are input into a look-up table for controlling the tracking filter.

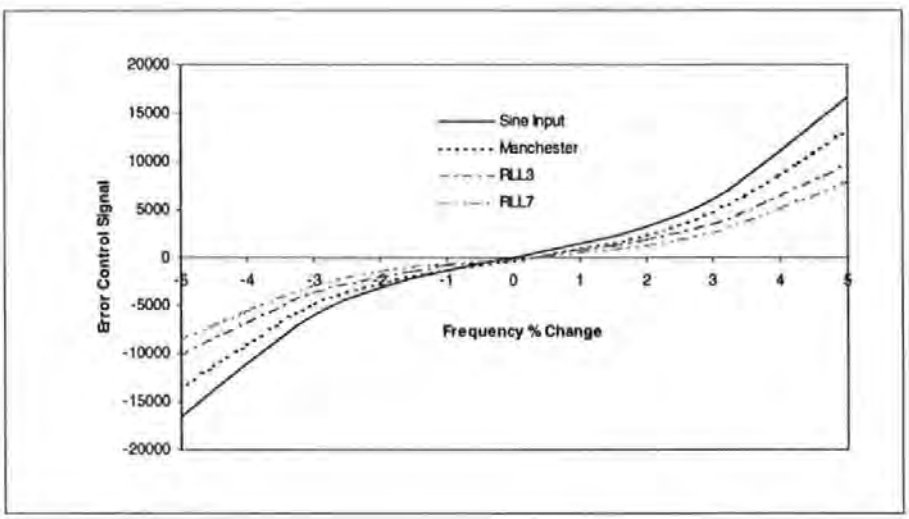


Figure 3-26 Error control signal for ±5% offset

The error calibration curve shown in Figure 3-26 has the detection filter spacing reduced to ±5% and demonstrates the increasing error deviation, as the run length of the encoded symbols is increased.

The error function of equation (3-10) is adjusted by means of variables a and x to achieve a fitted curve over the frequency range (n), to that of the simulated results.

$$E_n = 17000 \cdot \frac{\sinh(a \cdot n)}{\sinh(a \cdot x)} \qquad \text{..... (3-16)}$$

A modified error control function for RLL7 data over a ±5% detection range, is plotted in Figure 3-27, where $a = 0.48$ and $x = 6.5$. The function is overlaid onto the simulation data to demonstrate the fitted curve.

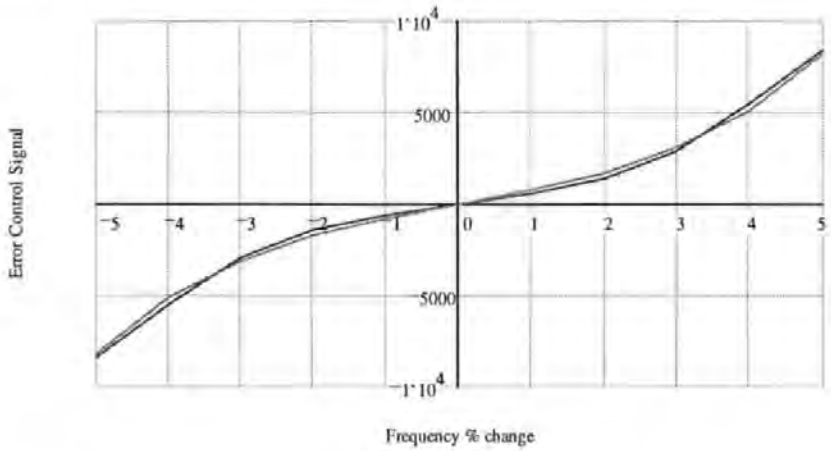


Figure 3-27 Error calibration function for RLL7 data

Similar functions may be plotted for Manchester data ($a = 0.48$ and $x = 5.5$) and RLL3 data ($a = 0.48$ and $x = 5.0$). A table relating the error control signal to percentage frequency offset can be generated from equation (3-16), or by means of open loop simulations.

3.2. Simulations

The adaptive clock recovery system has been simulated, using RLL data to analyse:

- **Acquisition Performance.**

Simulation results have shown significant improvements over that of a basic PLL. A PLL with a normalised loop bandwidth of 0.0015Hz requires up to 625 Manchester encoded symbols to acquire lock, over a ±2% acquisition range. Whereas the DSP-based system, adapts in just 34 cycles over a ±10% acquisition range. However, unlike a PLL, the recovered clock is acquired before the system has fully adapted. As for the fixed frequency filter, clock acquisition is achieved within a symbol period, from the first transition of the received symbols.

- **Flywheel Performance**

A narrow bandwidth PLL is necessary to provide flywheel memory. A normalised PLL bandwidth of 0.0015Hz provides a short flywheel period, of 7 symbols. Whereas the DSP-based adaptive system demonstrates similar flywheel performance to that of the fixed frequency filter. Once the system has fully adapted the clock flywheels for more than 80 symbol periods.

- **Tracking Performance**

For best results, the tracking range is reduced, depending upon the run length of the encoded symbols. For Manchester encoded data the tracking range can be up to ±10%, while for encoded data with a run length of 7, the tracking range is reduced to ±5%. Also, the tracking range has the same limits as the acquisition range.

3.2.1. Simulation model

Several hierarchy custom blocks have been created, for simulation of the adaptive clock recovery algorithm. An overview of the simulation model is shown in Figure 3-28. Full simulation models and results of the adaptive clock recovery algorithm can be seen in Appendix 7, key extracts of which are included in this chapter.

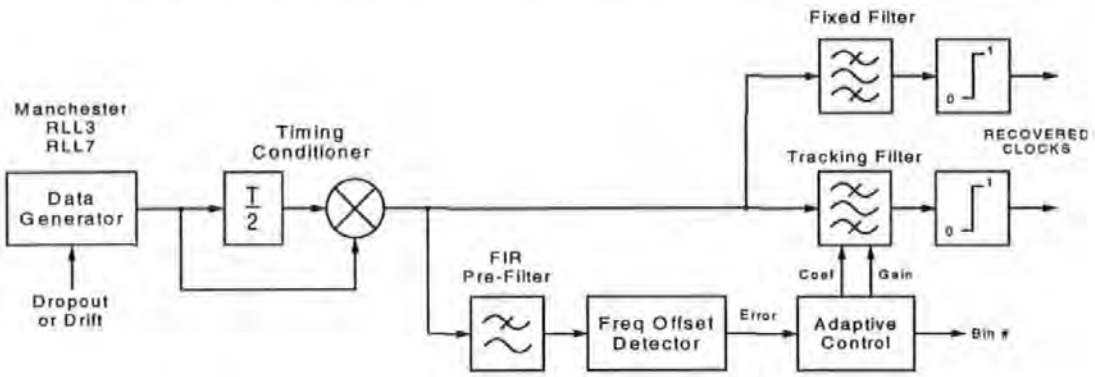


Figure 3-28 Overview of adaptive simulation model

Test simulations have revealed that many of SPWs standard library blocks are quantized to the nearest sample rate integer. For simulation of the fixed frequency clock recovery system, this is acceptable since 16 samples/symbol is used. However for the adaptive system, the symbol rate frequency deviates by up to ±10%, corresponding to 14.545 to 17.7777 samples/symbol, as indicated in Table 3-1. Therefore, the simulation model must operate at non-integer sample rates. For this reason SPWs standard random generator and encoders cannot be used. Instead, a special RLL Data Generator block has been created, which can be adjusted to provide either Manchester encoded data or PRBS data of the required maximum run length (RLL3 and RLL7 are used).

For RLL7 data, the data generator has provision for applying a short phase-reversal preamble at the start of a data burst, to aid clock acquisition.

To evaluate clock flywheel performance, the data generator output is subjected to a dropout, by holding the data for 80 symbol periods. For evaluation of acquisition and dropout performance, the data generator is clocked with a fixed frequency signal generator, pre-set to the desired frequency offset.

In order to simulate a drifting symbol rate frequency, the data generator is alternatively clocked from a VCO with a ramp control voltage, such that the VCO output is linearly frequency modulated across the entire tracking range. The frequency modulation can be initially inhibited, to allow full acquisition of the adaptive clock recovery system, before frequency tracking is applied. A "Delay and Multiply" timing conditioner is used to generate a symbol timing frequency component, which is then applied to the clock recovery system. A fixed frequency filter is also provided, in order that the benefits of the tracking system can be compared. The symbol timing frequency component is provided with gain to compensate for the lower Q of the detection filters, and then low-pass filtered to remove odd harmonics. The pre-filtered signal is applied to the frequency detector, which derives a frequency offset error signal. The error signal is translated to frequency offset bins by a custom adaptive control block, which applies the appropriate gain and coefficient to the adaptive filter, as tabulated in Table 3-1.

The calibration of the adaptive control block is specific to both detection filter spacing and the coding scheme.

Calibration data for Manchester, RLL3 and RLL7 (maximum run length) codes, relative to ±10%, ±7.5% and ±5% tracking ranges, are tabulated in Appendix 7.

% Offset Bin #	Symbol Rate Hz	Samples/ Symbol	Coef C	Gain A
+10	0.068750	14.5450	1.812735	1.10204
+9	0.068125	14.6788	1.816003	1.09090
+8	0.067500	14.8148	1.819242	1.08000
+7	0.066875	14.9523	1.822454	1.06931
+6	0.066250	15.0943	1.825638	1.05882
+5	0.065625	15.2380	1.828793	1.04854
+4	0.065000	15.3846	1.831920	1.03846
+3	0.064375	15.5339	1.835019	1.02857
+2	0.063750	15.6862	1.838090	1.01886
+1	0.063125	15.8415	1.844113	1.00934
0	0.062500	16.0000	1.844146	1.00000
-1	0.061875	16.1616	1.847132	0.99082
-2	0.061250	16.3625	1.850089	0.98182
-3	0.060625	16.4948	1.853017	0.97297
-4	0.060000	16.6666	1.855917	0.96428
-5	0.059375	16.8421	1.858788	0.95575
-6	0.587500	17.0212	1.861631	0.94737
-7	0.058125	17.2043	1.864445	0.93913
-8	0.057500	17.3913	1.867230	0.93103
-9	0.056875	17.5824	1.869986	0.92308
-10	0.056250	17.7777	1.872714	0.91525

Table 3-1 Simulation data look-up table

3.2.2. Simulation results

Adaptive clock recovery performance has been evaluated using Manchester encoded data and pseudo random NRZ data with maximum run lengths of 3 (RLL3) and 7 (RLL7), full simulation results of which are shown in Appendix 7.

3.2.2.1. Acquisition performance.

Simulations have been performed to measure the acquisition performance for an input signal with varying frequency offsets. The graph of Figure 3-29 is a collection of simulation data for the three test codes, demonstrating the number of symbols taken for the tracking filter to adapt to an offset input frequency.

The further the input signal is offset from the nominal, then the longer the tracking filter takes to adapt to the new input frequency. It can also be seen that the filter requires more symbols to adapt for codes with longer run lengths. This is because there are less symbol transitions to excite the detection filters and hence, the dc signal at the error detector takes longer to settle to a steady state.

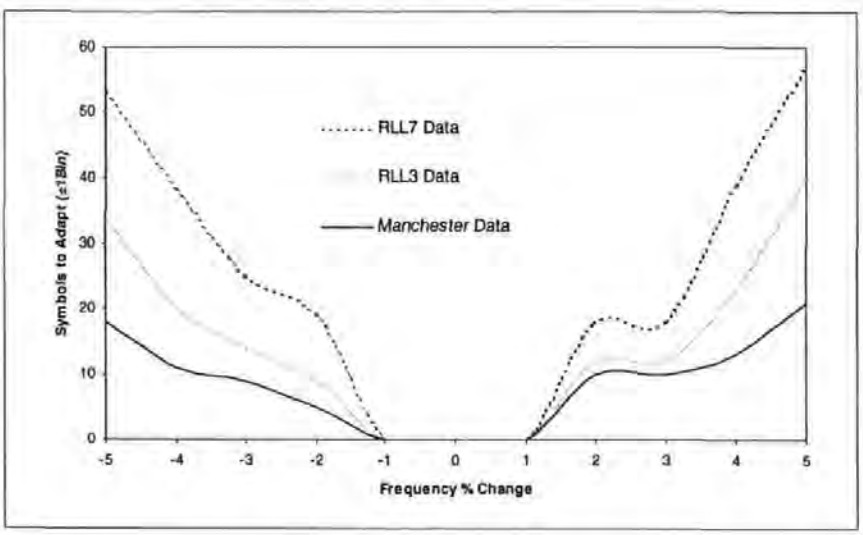


Figure 3-29 Acquisition performance - ±5% tracking range

For a 5% frequency offset, the tracking filter adapts to its optimum within 21 symbols for Manchester data, and 57 symbols for RLL7 encoded data. However, clock acquisition is achieved within a symbol period, since the tracking filter immediately rings at the frequency of the input symbols.

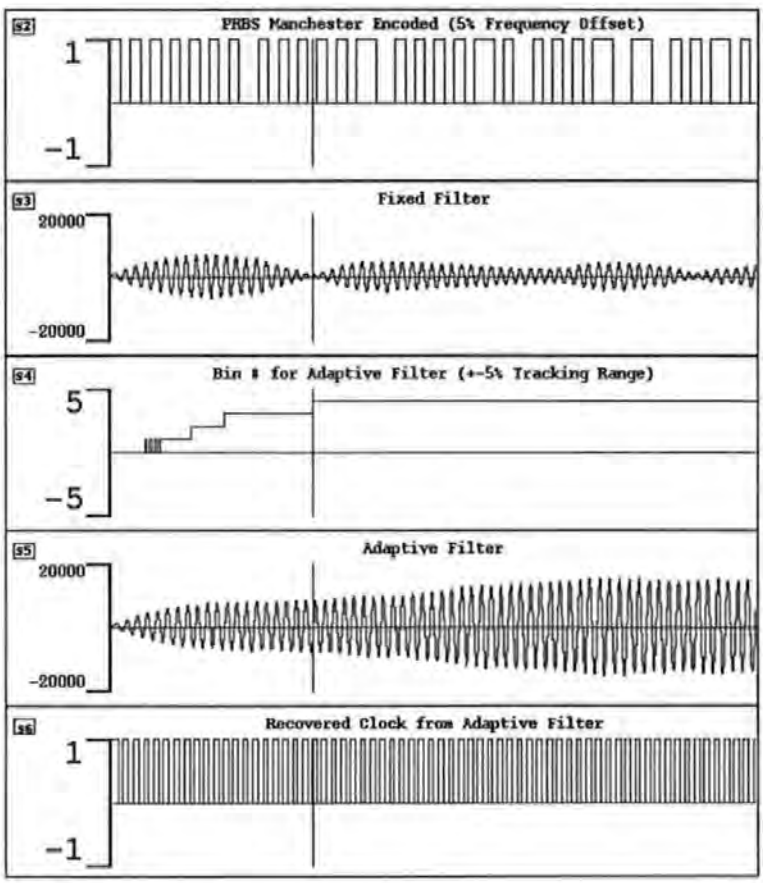


Figure 3-30 Acquisition performance ±5% range - Manchester data

The simulation of Figure 3-30 demonstrates the acquisition performance for Manchester encoded data. Initially, the filter is off-tune and the amplitude of the ringing is weak, but as long as the output of the filter does not decay to zero, then the hard limited output provides a continuous recovered clock. As the tracking filter adapts, the input frequency becomes increasingly within the filter bandwidth and the filter output subsequently grows in amplitude, reaching its maximum when the

filter has fully adapted to the new input frequency. The simulation of Figure 3-30 demonstrates the benefits of an adaptive filter, compared to that of a fixed frequency filter. The fixed filter remains continuously off-tune and the low-level filter output falls to zero, resulting in loss of the recovered clock

For RLL7 encoded data, it is possible for the first symbols to be retrieved to contain a long run length, in which case there are few transitions to stimulate the clock recovery system. For this reason, a short preamble containing a few phase-reversal symbols should be applied to stimulate the filter ringing, sufficient to flywheel over worst case run lengths, as shown in the simulation of Figure 3-31.

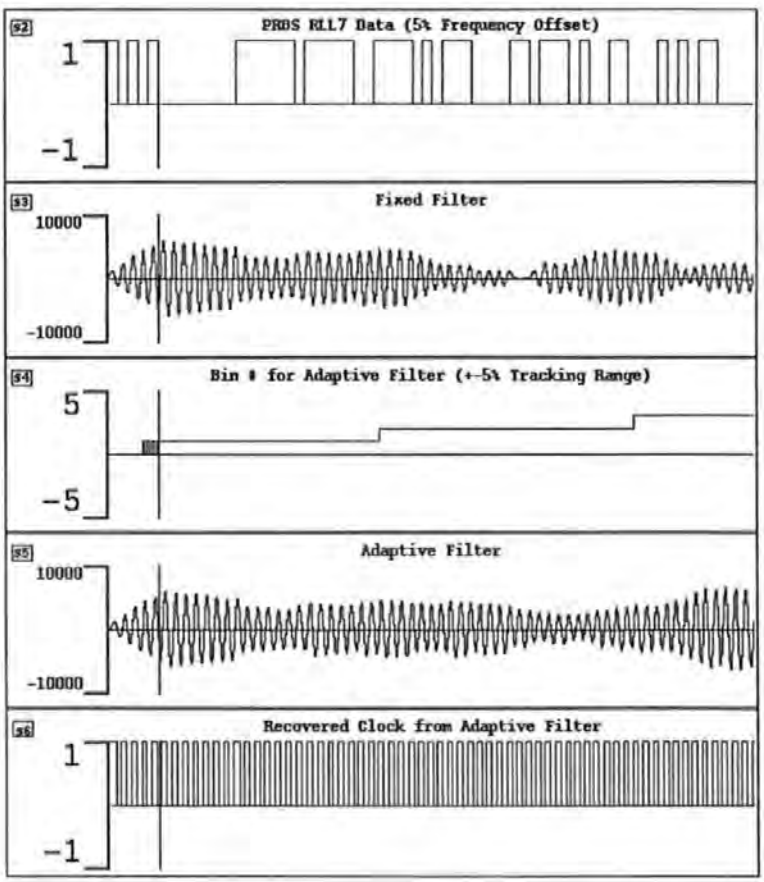


Figure 3-31 Acquisition performance ±5% range - RLL7 data with preamble

An acquisition range of ±5% is suitable for use with all three test codes, however, the range can be extended for use with codes of shorter run lengths.

Manchester encoded data have been applied to the detection system with a ±10% acquisition range. With this wide detection filter spacing, the residual low-level odd harmonics from the pre-filter become apparent, resulting in slight distortion of the error control signal. The tracking filter adapts over the range, but becomes non-symmetrical, due to low level harmonics aliasing into the detection system.

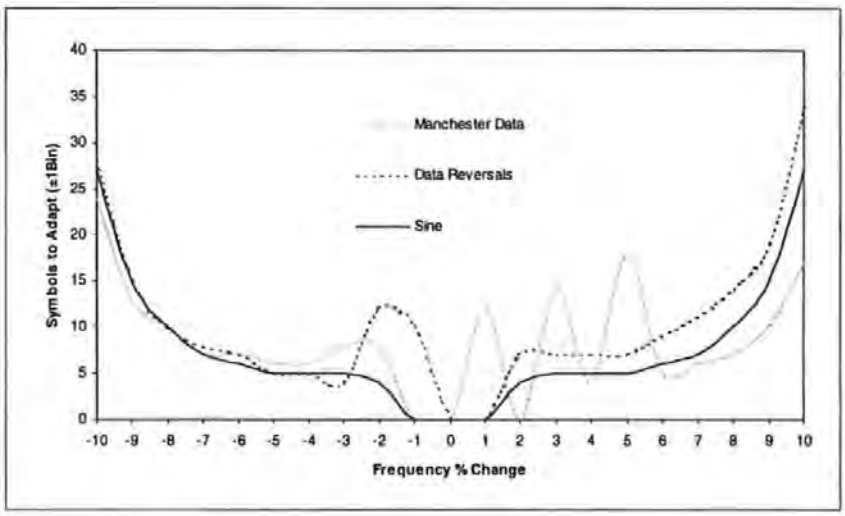


Figure 3-32 Acquisition performance - ±10% tracking range

The graph of Figure 3-32 is a collection of simulation data to demonstrate acquisition performance over a tracking range of ±10%. A sinusoidal input is free from harmonics and provides a symmetrical acquisition response, about the nominal input frequency. However, the acquisition response to a periodic, but phase-reversal data input, is distorted due to the residual harmonics from rectangular pulses. The symbol run lengths of the Manchester encoded data produce an even weaker signal at the output of detection filters, making the signal more vulnerable to low level

harmonic distortion. The system is less susceptible to random noise, which is averaged by the IIR filters and the envelope detectors, within the detection system.

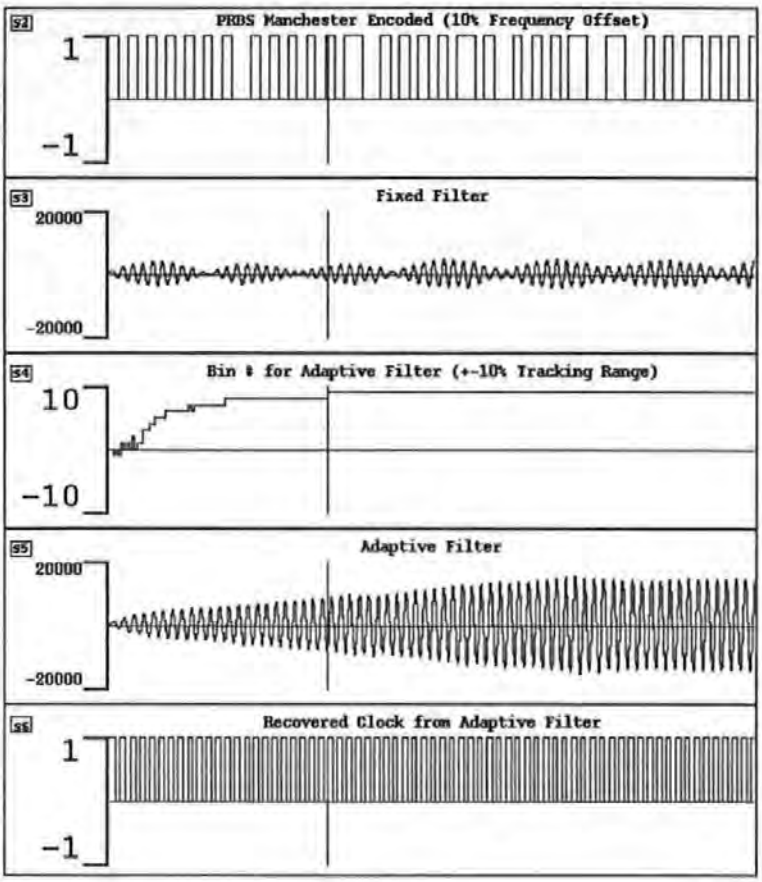


Figure 3-33 Acquisition performance ±10% range - Manchester data

The simulation of Figure 3-33 demonstrates the acquisition performance for Manchester encoded data with a +10% frequency offset input. The hard limited output from the tracking filter, provides a recovered clock within a symbol period of the first retrieved symbol. The amplitude at the output of the tracking filter becomes increasingly stronger, as the filter adapts to the offset frequency, while in contrast the off-tune fixed filter keeps falling to zero amplitude.

3.2.2.2. Flywheel performance

Once the filter has fully adapted to the frequency of the input symbols, the high-Q adaptive tracking filter exhibits similar flywheel performance to that of the fixed frequency clock recovery system, as discussed in Chapter 2.

During a data dropout, the absence of symbol transitions to the frequency offset detector results in the error signal decaying to zero. Hence, the coefficients to the tracking filter are modified, such that the centre frequency is progressively adjusted towards the nominal frequency (offset = 0). As well as the clock recovery filter, the detection filters themselves exhibit a flywheel effect over missing transitions. This, combined with averaging within the envelope detectors, provide damping to the drift in the error control signal, thereby aiding the return to the optimum frequency, when the input signal is re-applied. If the dropout is too long a period, then the centre frequency of the tracking filter is returned to the nominal frequency position and the system must re-acquire when the input signal is restored.

The simulation of Figure 3-34 shows the flywheel performance for Manchester encoded data when subjected to an 80-symbol dropout. The tracking filter has previously adapted to a +10% frequency offset. The tracking filter has drifted off-tune from the input frequency by only 2%, and at the end of the dropout period the flywheel clock remains in-phase when input symbols are re-applied, as shown in the zoomed view of Figure 3-35.

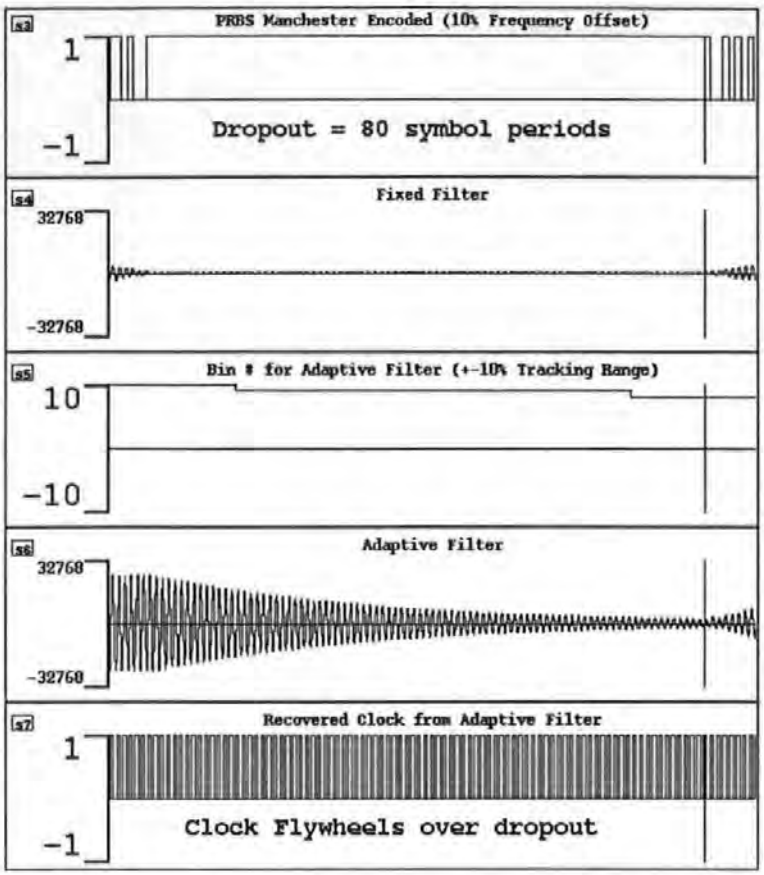


Figure 3-34 Flywheel performance +10% offset - Manchester data

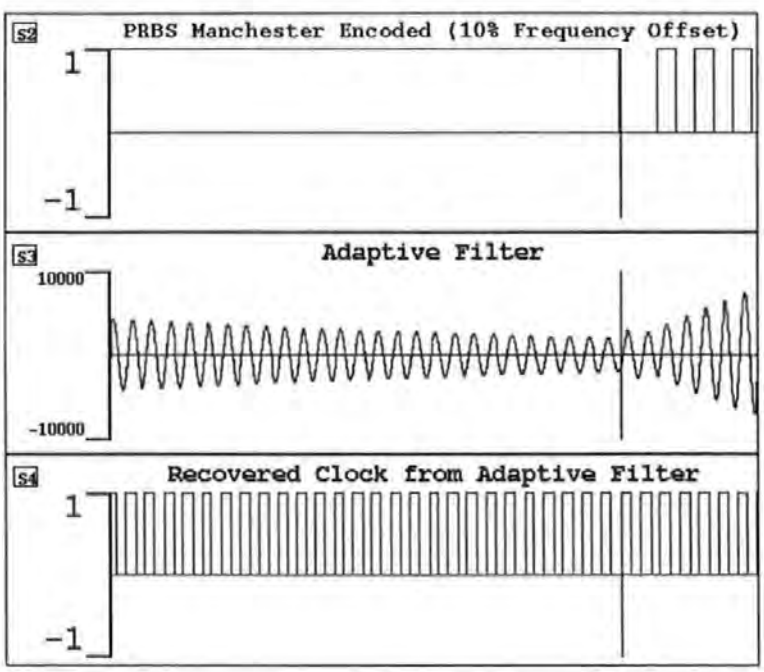


Figure 3-35 Flywheel performance zoomed in - Manchester data

For comparison, the fixed frequency filter is also shown in Figure 3-34, where it can be seen that the fixed filter has failed to grow in amplitude, and subsequently provides little flywheel performance.

Figure 3-36 shows the flywheel performance for random data with a run length =7, when subjected to an 80-symbol dropout. For RLL7 data the tracking range is reduced to ±5%, and the tracking filter has adapted to a +5% frequency offset.

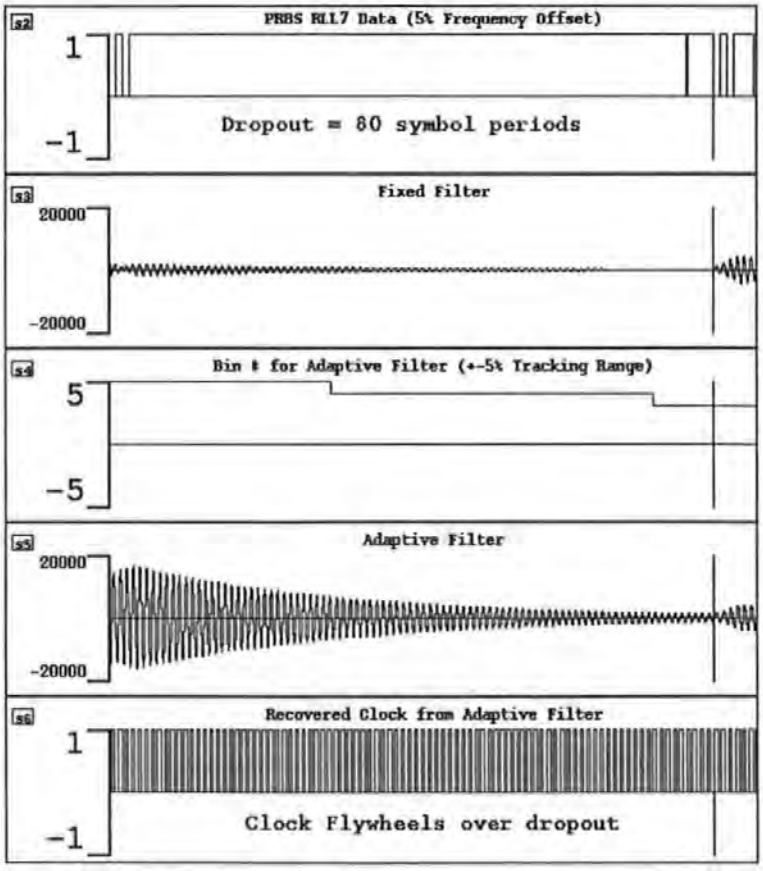


Figure 3-36 Flywheel performance +5% offset - RLL7 data

In the absence of input symbols, the adaptive control (Bin #) slowly starts to drift towards zero offset, and the adaptive filter successfully flywheels over the dropout period.

On close examination of the tracking filter at the end of the dropout period, the phase of the flywheel clock remains coincident with the re-applied input symbol transitions, as demonstrated in the simulation of Figure 3-37.

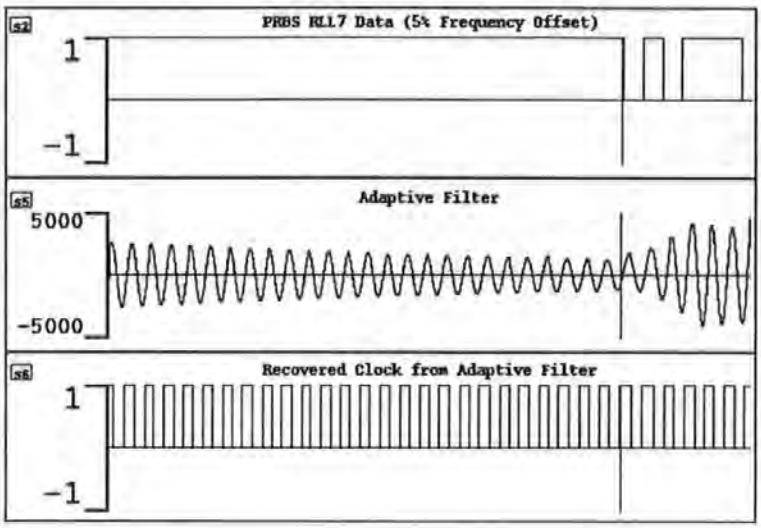


Figure 3-37 Flywheel performance zoomed in - RLL7 data

If the input signal is completely killed, then the adaptive control (Bin #), takes 240 symbols to decay to zero for Manchester data with a 10% offset, and 135 symbols for RLL7 data with a 5% offset. The recovered clock flywheels over these periods, however, the phase of the clock drifts from its optimum sampling point, at the centre of the data eye.

In all cases the drift is less than half a symbol period, therefore assuming an optimum data detection window, zero cycle slips are achieved. However, to guarantee optimum sampling, the dropout period should be restricted to 80 symbol periods. Full simulation results can be viewed in Appendix 7.

3.2.2.3. Tracking performance

The adaptive filter is capable of tracking over the entire acquisition range. As previously described, the maximum tracking range is limited relative to the run length of the coding scheme. For Manchester encoded data the tracking range is ±10%, while for data with a run length of 7 the tracking range is reduced to ±5%. In order to simulate a drifting symbol rate frequency, the data generator is clocked from a VCO with a ramp control voltage, such that the VCO output is linearly frequency modulated across the entire tracking range.

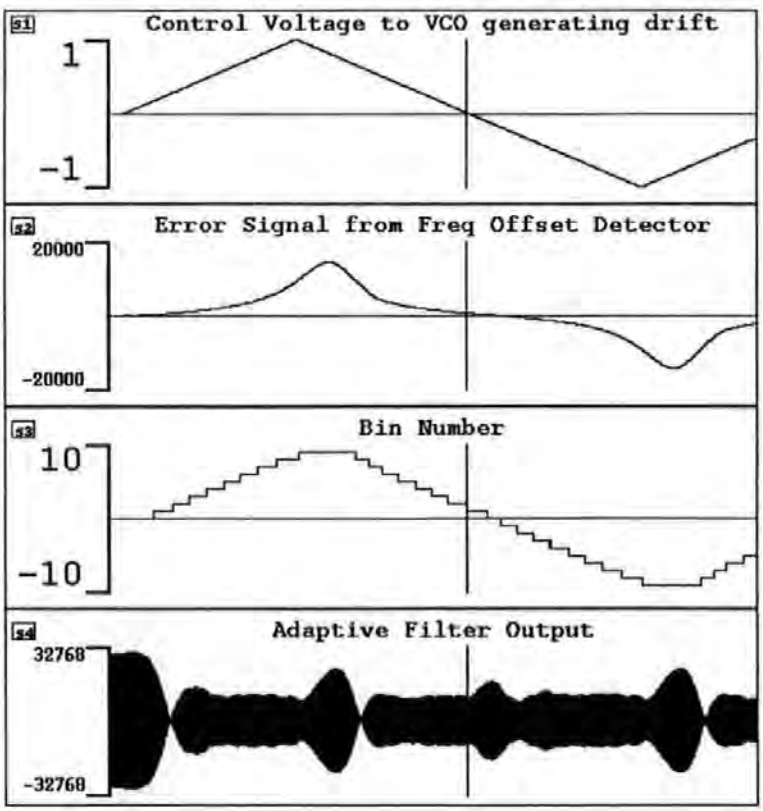


Figure 3-38 Tracking performance ±10% range - Sine input

The simulation of Figure 3-38 shows a sinusoidal signal whose frequency is drifting at a rate of 10% per sample. The drifting frequency signal is applied to the adaptive filter to monitor the tracking performance. The error signal (s2), at the output of the

frequency-offset detector, is identical to the theoretical error calibration response, shown in Figure 3-11. The output of the adaptive control block (s3), shows the quantized bin numbers tracking the frequency drift, successfully following the VCO control signal. The tracking signal is delayed from the VCO control signal by 27 cycles, due to delay elements within the simulation model. The adaptive control tracks the change in the input frequency, updating the coefficients and gain to the tracking filter. Closely comparing the frequency and phase of the adaptive filter output, with that of the drifting input confirms that the filter is successfully tracking the input signal. If the input frequency becomes stable anywhere within the tracking range, then the filter grows to its maximum amplitude.

During tracking, the phase of the recursive feedback within the filter lags the input by 2 or 3 samples for a negative tracking direction, and leads for a positive direction. If the frequency drift abruptly changes direction, the output of the tracking filter sharply drops in amplitude, while the recursive feedback adjusts to the change in direction of the input phase. However, close examination shows that the amplitude of the filter is sufficient to maintain a recovered clock, and the filter quickly re-adapts to full amplitude, as shown in Figure 3-39. The abrupt change in tracking direction is averaged by the detection system, producing an apparent short-term stability in frequency, at bins ± 10. This results in the filter output peaking in amplitude, just before a sharp drop, due the change in phase direction, as shown in Figure 3-38.

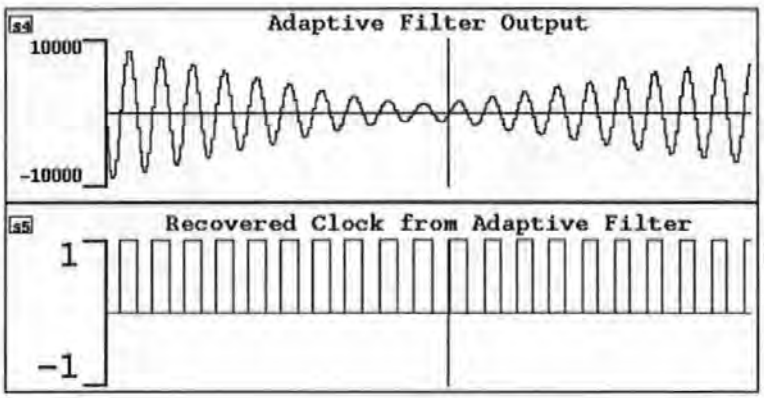


Figure 3-39 Abrupt change in tracking direction - Sine input

Similar tracking performance for data inputs may be observed, from the full simulation results presented in Appendix 7. The sample simulation of Figure 3-40 demonstrates the tracking performance for RLL7 data, with frequency drifting over the entire ±5% acquisition range.

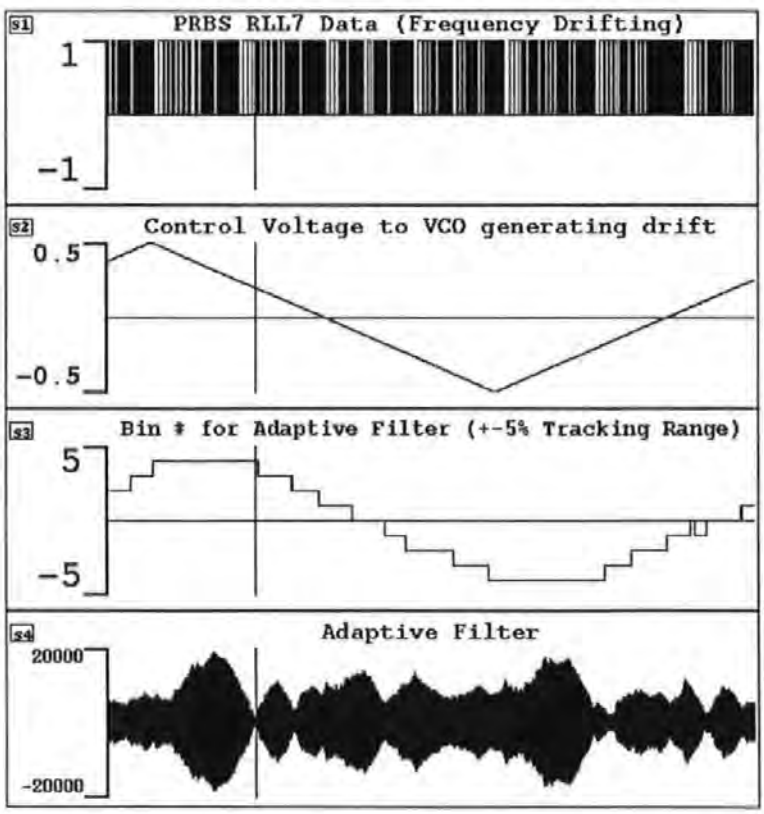


Figure 3-40 Tracking performance ±5% range – RLL7 data

The envelope of the tracking filter output (s4) tends to be more ragged, than that of a periodic input, due to the filter output decaying over long run lengths of the encoded symbols.

Zooming in at the worst-case filter output amplitude, when tracking abruptly changes direction, confirms that clock synchronisation is maintained, as shown in Figure 3-41. Phase variations of the recovered clock are only a few samples and assuming an optimum data detection window, then with a nominal 16 samples/symbol, zero cycle slips are achieved.

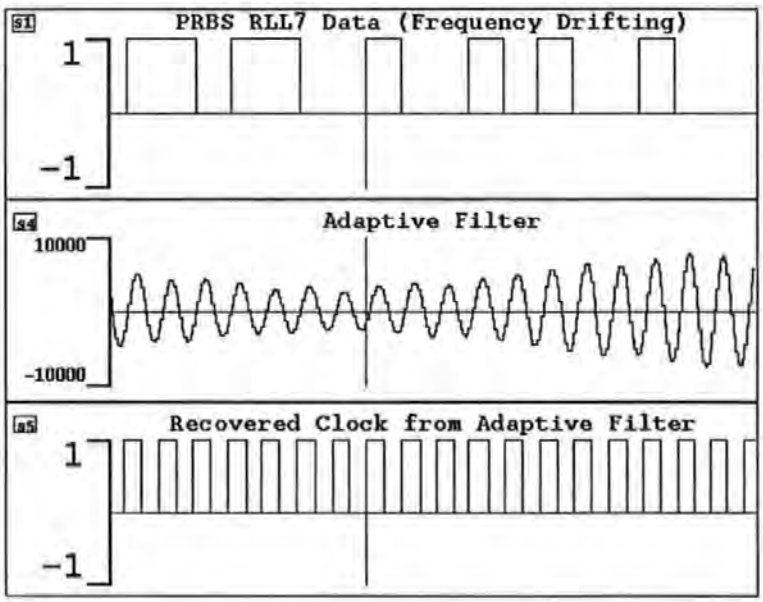


Figure 3-41 Abrupt change in tracking direction - RLL7 data

Similar results for Manchester encoded data (±5%, ±7.5%, ±10% ranges) and RLL3 data (±5%, ±7.5% ranges) can be viewed in Appendix 7.

Chapter 4

4. OFFT-based clock recovery

The results in Chapter 3 show that frequency estimation in an adaptive clock recovery system can be implemented using an iterative algorithm, based upon two symmetrical IIR detection filters. This approach achieves rapid clock acquisition, but is only capable of acquisition over a limited frequency offset range.

In DSP, a common method of estimating frequency from a time domain signal is the FFT (Fast Fourier Transform) algorithm. Figure 4-1 shows the concept of an adaptive clock recovery system, using an FFT to estimate the symbol rate frequency. The power spectrum output of the FFT is threshold detected, and the frequency bin with the greatest power corresponds to the unknown symbol rate. The returned frequency bin number is then used in a look-up table to apply new coefficients, to move the centre frequency of the adaptive filter to the correct symbol rate.

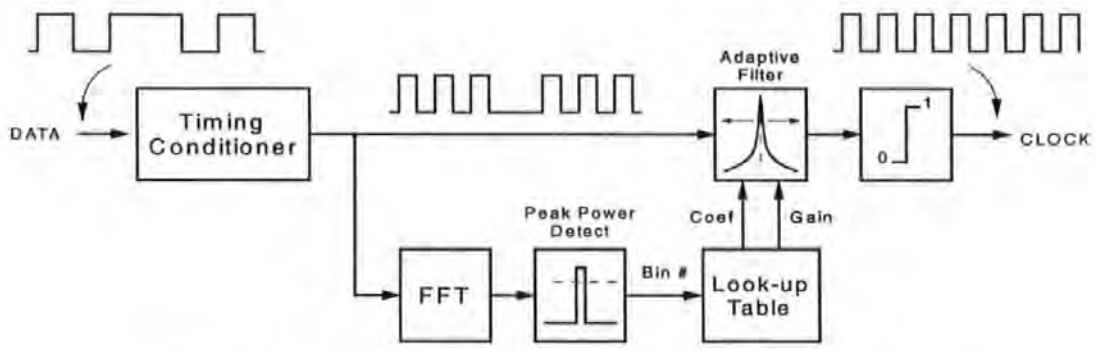


Figure 4-1 Concept of FFT-based adaptive clock recovery

A disadvantage of an FFT approach is that a block of samples must be processed, before a frequency estimate is obtained. This clearly limits clock acquisition performance for burst mode applications (necessitating a long preamble), but does have the advantage that large symbol rate offsets (from approximately dc to the Nyquist rate) can be estimated. This is useful for recovery of data where the symbol rate is not known, or applications where a large but constant symbol rate frequency offset exist.

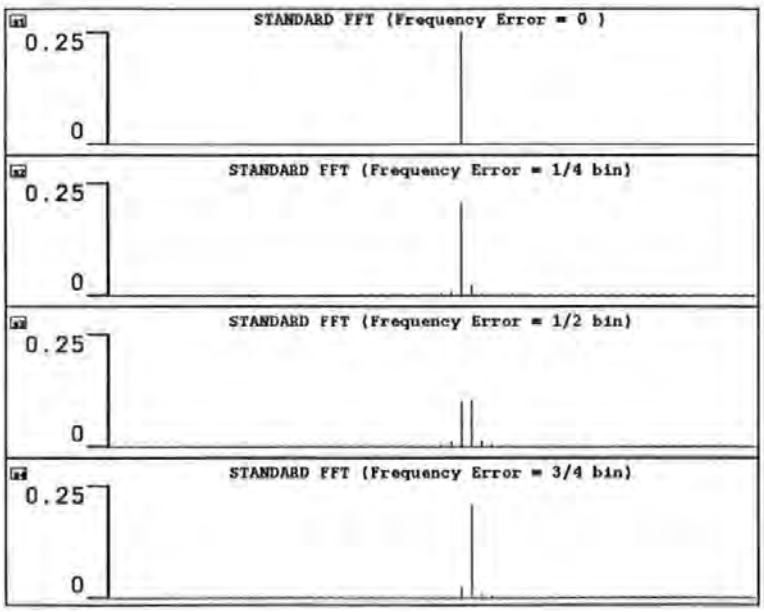


Figure 4-2 Power spectrum for standard FFT

Another problem estimating a frequency, based upon power threshold detection, is when a frequency offset exist, such that the power is shared equally between adjacent bins, as shown in the simulation of Figure 4-2. For this worst case (3rd trace), the power threshold must be reduced, and this increases the probability of false detection triggered by noise. Fortunately, the OFFT (Offset Fast Fourier Transform) algorithm [114] successfully overcomes this problem.

4.1. The OFFT algorithm

The FFT algorithm is an efficient method of evaluating the DFT (Discrete Fourier Transform). Appendix 8 contains the derivation of the FFT from a DFT using the DIT (Decimation in Time) method [115]. The FFT produces a frequency representation of the symbol-timing component and provides a convenient means of detecting and estimating the symbol rate frequency. However, when the signal power is shared equally between adjacent bins, the threshold must be reduced and this can result in false detection, particularly at low signal to noise ratios. The OFFT provides superior signal detection performance to that of a standard FFT, by introducing an asymmetric frequency response around the Nyquist frequency.

Equations for the DFT and the modified Offset DFT are given below.

DFT $$X(k)=\sum_{n=0}^{N-1} x_n \cdot W_N^{kn} \quad \ldots\ldots\ldots\ldots \quad (4\text{-}1)$$

Offset DFT $$X(k+c)=\sum_{n=0}^{N-1} x_n \cdot W_N^{(k+c)n} \quad \ldots\ldots\ldots\ldots \quad (4\text{-}2)$$

where $k = 0,1,2,\cdots\cdots, N\text{-}1$ and $W_N = e^{-j2\pi/N}$

The offset DFT is identical to the standard DFT, with the exception of an offset added to the twiddle factor, W_N. Therefore, a standard algorithm can be used for both the FFT and the OFFT, since only the twiddle factors change. However, it should be noted that an FFT often optimises the DFT for DSP implementation, by simplifying the coefficients (twiddle factors) to ±1and ±j.

For the OFFT, the addition of the offset invalidates this optimisation. Hence, a complete set of coefficients should be pre-calculated and stored as a look-up table in memory, for fast coefficient multiplication.

If the offset $c \equiv 0.25$ of a frequency bin, then for positive frequencies (dc to the Nyquist rate), the spectrum will be offset by ¼ of a frequency bin compared to that of the standard DFT. The negative frequency below dc is therefore offset by ¾ of a frequency bin. The spectrum about dc repeats around the sample rate, and the negative frequency with ¾ of a bin offset re-appears between the Nyquist rate and the sample rate, as shown in Figure 4-3.

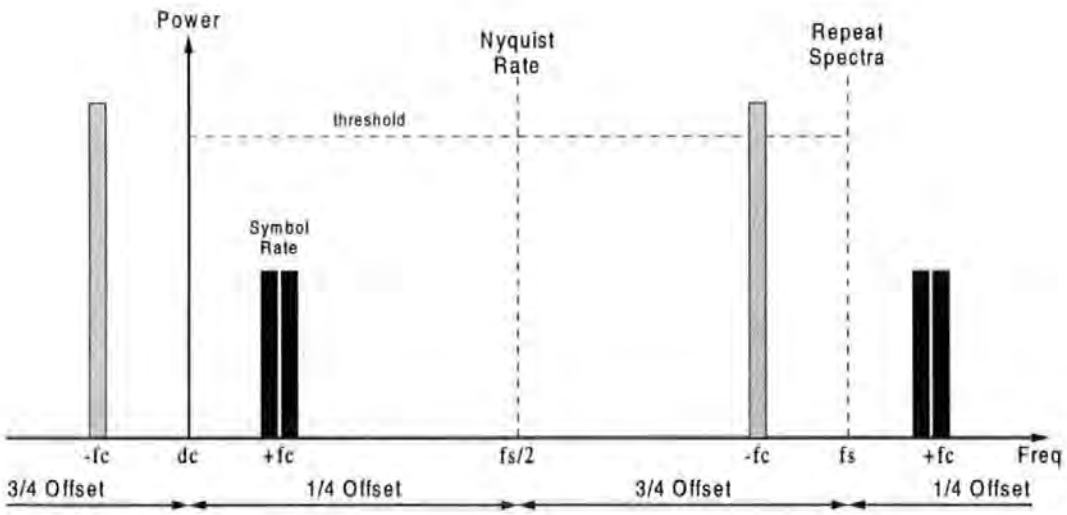


Figure 4-3 Spectrum of the OFFT

If the frequency offset of the symbol rate falls between frequency bins, then the power is shared equally between adjacent bins, and the threshold fails to detect the symbol-rate frequency component. However, with the asymmetry of the OFFT spectrum, a large frequency component appears at the negative frequency about the

sample rate, which is successfully threshold detected. Similarly, if the frequency offset of the symbol rate is such that the power is shared equally between adjacent bins at the negative frequency of the repeat spectra, then a frequency component will peak at the positive frequency above dc. Hence, a see-saw affect is achieved, where the frequency is detected either above or below the Nyquist frequency.

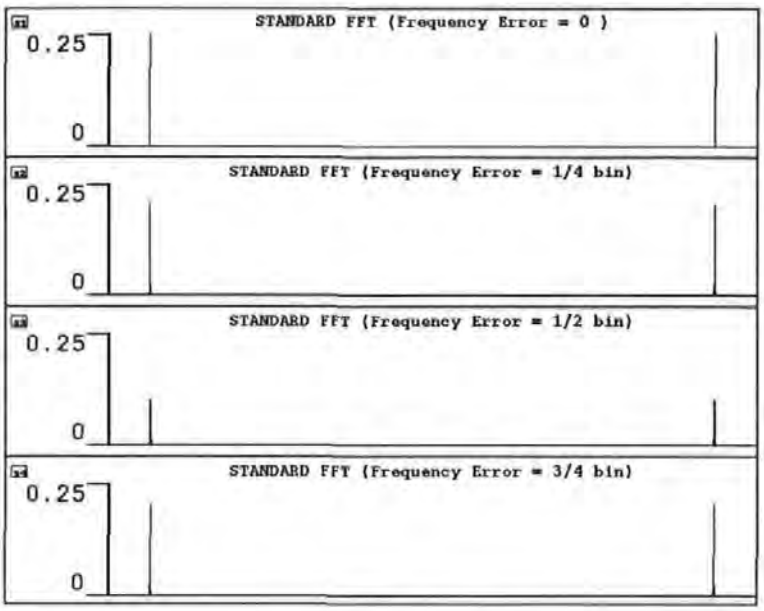


Figure 4-4 Simulation of FFT with frequency offsets

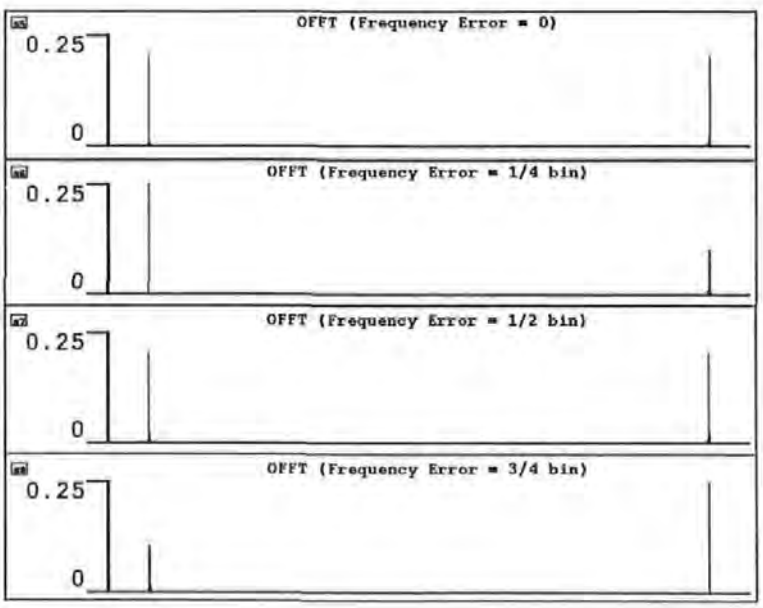


Figure 4-5 Simulation of OFFT with frequency offsets

The simulations shown in Figure 4-4 and Figure 4-5 are for a 1024-points, representing a frequency from dc to the sample rate. In each case, frequency error steps are introduced, equivalent to ¼ of a frequency bin. The worst-case response for the FFT, is shown in trace 3 of Figure 4-4, where the frequency error is such that the power is shared equally between adjacent bins, both above and below the Nyquist rate (*N/2*). Hence, for a power threshold set near the peak power, these signals would fail to be detected. Whereas, for the OFFT shown in Figure 4-5, a large spectral component exists either above or below the Nyquist rate, due to the asymmetric frequency response.

Once the OFFT bin number corresponding to the maximum power is detected, a test is performed to determine the absolute frequency of the symbol rate (f_c).

If $k_{max} < (N/2)-1$ then $f_{est1} = (k_{max} + 0.25) \cdot f_s/N$ Hz (4-3)

$k_{max} > (N/2)-1$ then $f_{est2} = (N - k_{max} - 0.25) \cdot f_s/N$ Hz (4-4)

Where k_{max} is the threshold detected maximum power, N is the number of points in the OFFT and f_s is the sample rate. This provides a maximum residual frequency error of ¼ of a bin ($f_{error} = 0.25 \times f_s / N$), compared to ½ a bin for a standard FFT. Closer inspection of the residual frequency error is shown in Figure 4-6, where maximum errors occur, when the input frequency is offset by zero and half a bin (top and 3rd trace). For zero frequency offset, the OFFT power threshold achieves two hits, one below and another above the Nyquist rate, producing residual errors of +0.25bin and –0.25 bin respectively.

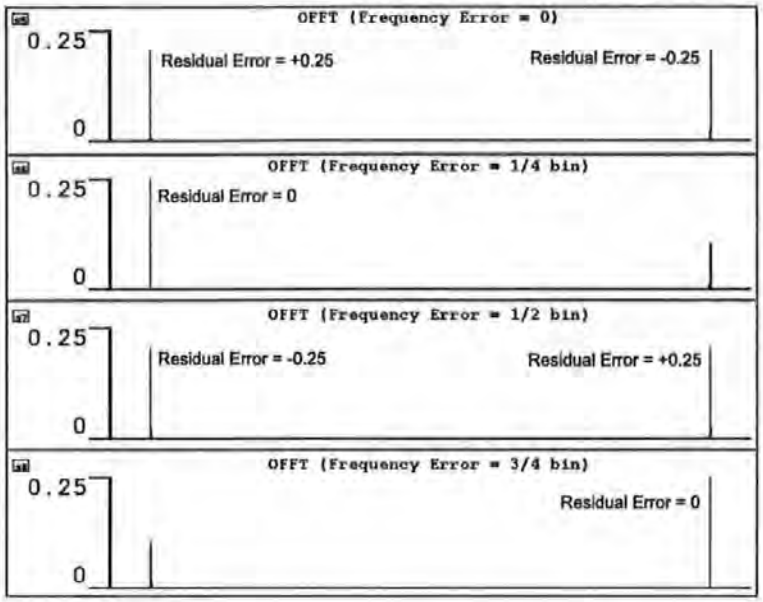


Figure 4-6 OFFT residual frequency errors

Similarly, when the input frequency is offset by ½ bin (trace 3), again two threshold hits are achieved, but this time the sign of the errors are reversed. Hence, when two power threshold hits are detected, one above and another below the Nyquist rate, the residual frequency error can be resolved by taking the average of the two.

If
$k_{max} < (N/2)-1$
and
$k_{max} > (N/2)-1$
then $f_{est3} = \frac{f_{est1} + f_{est2}}{2}$ Hz ..... (4-5)

The overall maximum residual frequency error is now reduced to just 1/8 of a frequency bin, given by:

$$f_{error} = \frac{1}{8} \times \frac{f_s}{N} \quad \text{.....} \quad (4\text{-}6)$$

and occurs when the input frequency is offset by 1/8, 3/8, 5/8 and 7/8 bin (i.e. 0.0 ±1/8 and 0.5±1/8).

The returned frequency estimate is then used to determine the required coefficient and gain to reposition the clock recovery filter to the correct symbol rate frequency, given by equations (2-12) and (3-3).

Since DSP processors do not contain trigonometric functions, a practical system requires the coefficients and gain to be pre-calculated for each of the OFFT bin values, and stored as a look-up table in memory.

The resolution, expressed as a percentage relative to the **nominal symbol rate**, is given by:

$$f_{resolution} = \left(\frac{f_{nom} + f_{error}}{f_{nom}} \cdot 100 \right) - 100 \ \% \qquad \ldots\ldots\ldots\ldots\ldots \quad (4\text{-}7)$$

For 16 samples/symbol, a 1024-point OFFT gives a resolution of 0.2%, which provides an accurate frequency estimate, used to reposition the adaptive filter. Given that an IIR clock recovery filter with a Q = 100 can tolerate a ±1% frequency offset, then a 512-point OFFT with a resolution of 0.4%, provides sufficient accuracy to reposition the adaptive filter. Hence, 32 symbols must be processed before a frequency estimate is obtained. In practice, twice this amount is required at the start of a burst, in order to guarantee a full OFFT buffer of symbols for processing. For burst mode applications, this may result in an unacceptably long synchronising preamble. However, once a frequency estimate is obtained, the clock recovery filter is immediately re-positioned (snapped) to the correct symbol rate frequency, and the benefit of rapid clock acquisition of a high-Q IIR filter is realised, as described in Chapter 2.

4.2. Simulations

The OFFT-based clock recovery system, has been simulated using RLL7 data, to analyse signal detection and frequency correction. Results have shown that the OFFT provides approximately 50% greater peak power than that of an FFT for worst-case frequency offsets. As a result, the signal power threshold can be increased, reducing the probability of false signal detection.

For simulation of the OFFT-based clock recovery system, several custom hierarchy blocks have been created. In particular, an OFFT block has been written in the C-language, compiled and linked to a symbol, which has been used in top level simulation systems. An overview of the simulation model is shown in Figure 4-7.

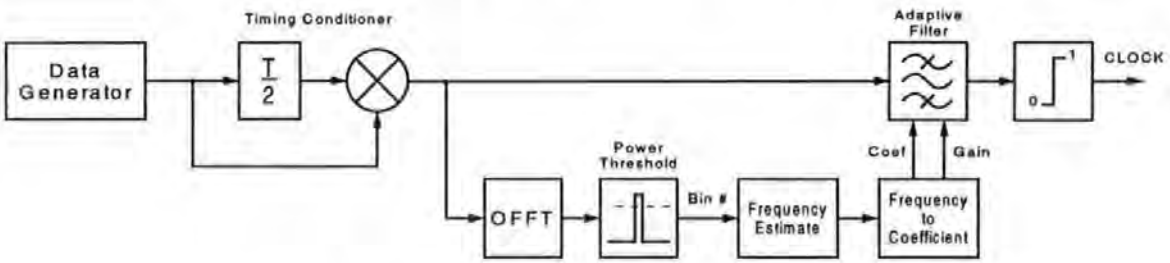


Figure 4-7 Overview of OFFT simulation model

The OFFT custom-coded block source code and the full simulation model are included in Appendix 9. The "Frequency Estimate" block is implementing using equations (4-3) to (4-5), which is then used by the "Frequency to Coefficient" block, by implemented equations (2-12) and (3-3), to determine the coefficient and gain for input to the adaptive filter. PRBS data generates a dc spectral component, therefore a guard band of 10 samples is provided, to prevent false power detection

triggered by dc. For data, harmonics are introduced and the noise floor increases due to modulation of the encoded run lengths, as shown in Figure 4-8.

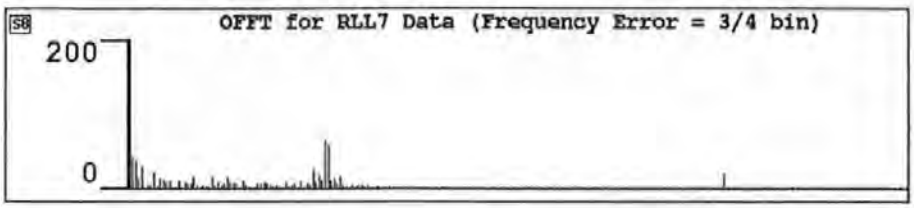


Figure 4-8 Simulation of OFFT with RLL7 data - zoomed in

With the reduced signal to noise ratio, it is apparent that reducing the power threshold would increase the probability of false signal detection. However, as previously established, the asymmetry of the OFFT allows a power threshold to be set well above the noise, and the signal is detected at the repeat spectra above the Nyquist rate, as shown in Figure 4-9.

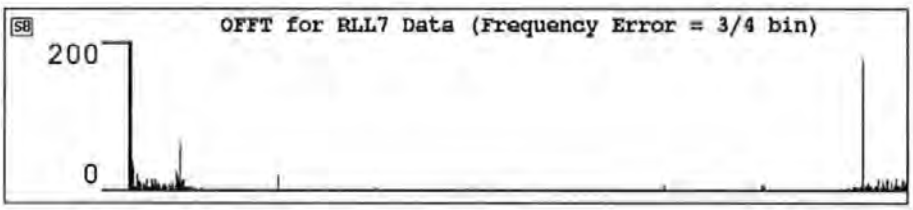


Figure 4-9 Simulation of 1024-point OFFT with RLL7 data

The simulation of Figure 4-10 shows the full OFFT-based clock recovery for RLL7 data, with a symbol rate offset of +20% (f_c = 0.075Hz). Bin number 947 is peak power detected, which is translated to a frequency estimate of 0.07495117Hz. On completion of processing the OFFT, the frequency estimate is translated to coefficient (Coef C = 1.778807) and gain (Gain = 1.226184) parameters, which are applied to the IIR filter. At this point, the filter is enabled and clock acquisition is almost instantaneous. Since the filter is re-positioned (snapped) to the centre

frequency of the input symbols, and also optimised for gain, the ringing of the filter rapidly grows to full amplitude, thereby maximising clock flywheel performance.

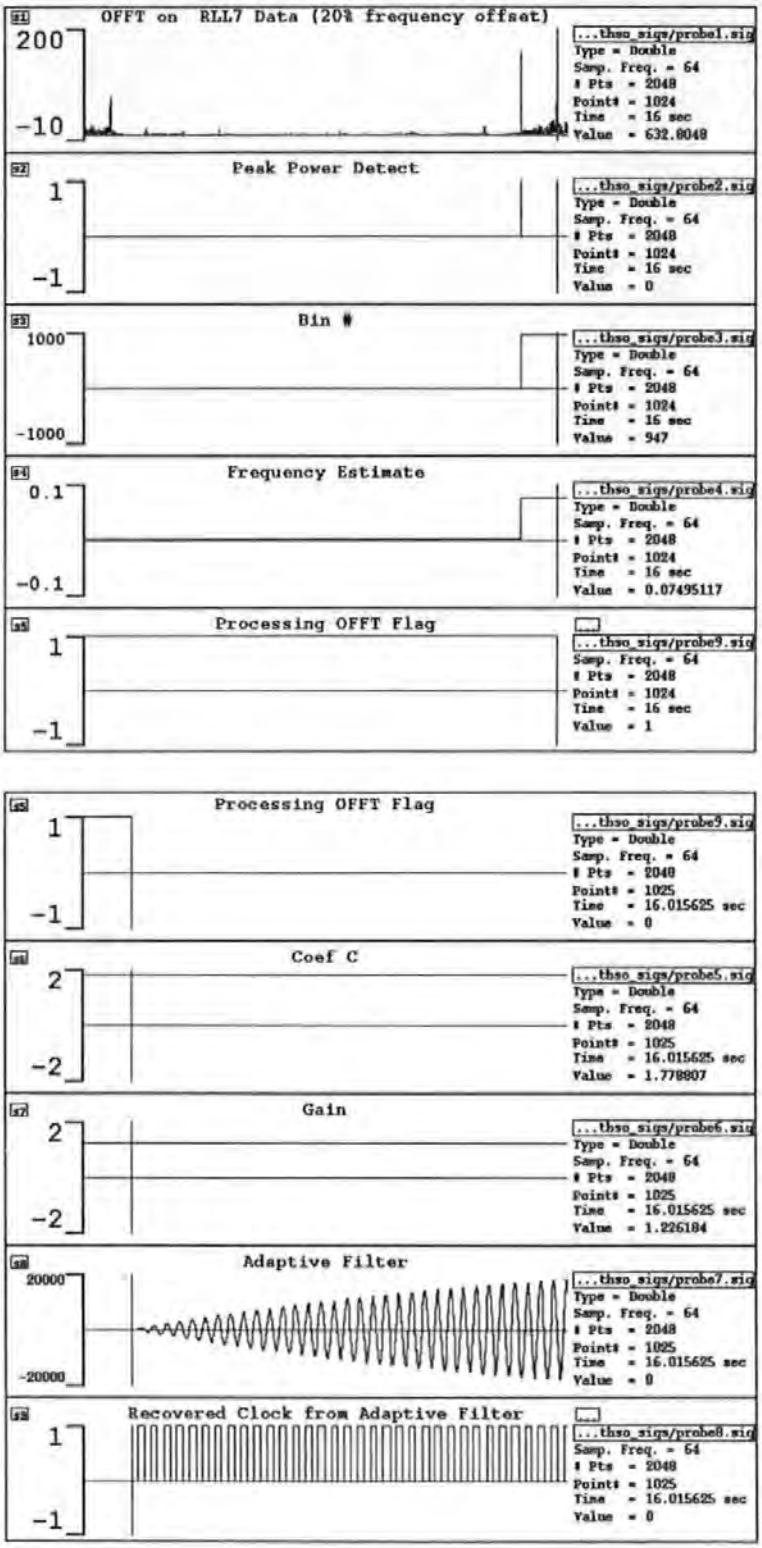


Figure 4-10 Simulation of clock recovery using OFFT

4.3. Test apparatus

The OFFT has been successfully implemented in real-time, utilising two TMS320C50 (80MHz) fixed-point DSP processors [43]. The author has designed and constructed a custom dual-processor DSP board, a block diagram of which is shown in Figure 4.11. Hardware information, including schematic diagrams and memory maps, are included in Appendix 10. The OFFT DSP assembler source code is included in Appendix 9.

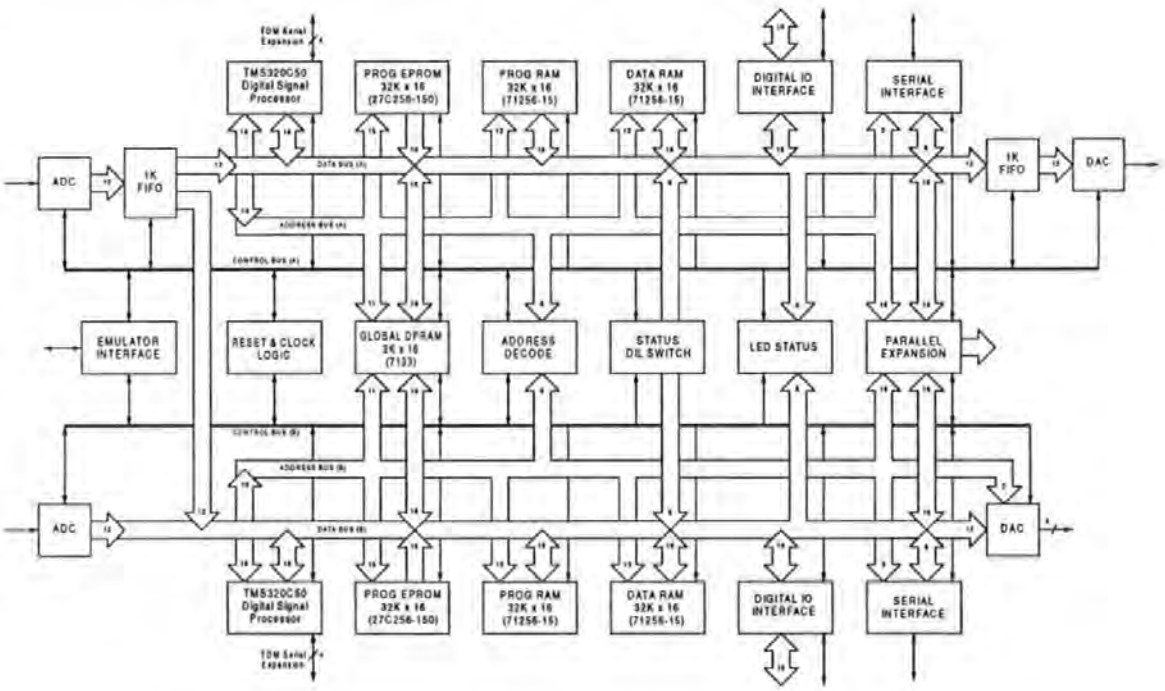


Figure 4.11 Dual-processor DSP hardware - Block diagram

The hardware consist of an 8-layer, extended Eurocard printed circuit board, based around two TMS320C50 fixed point DSP processors, jointly providing more than 80 MIPS (million instructions per second). A photograph of the board is shown in Figure 4-12.

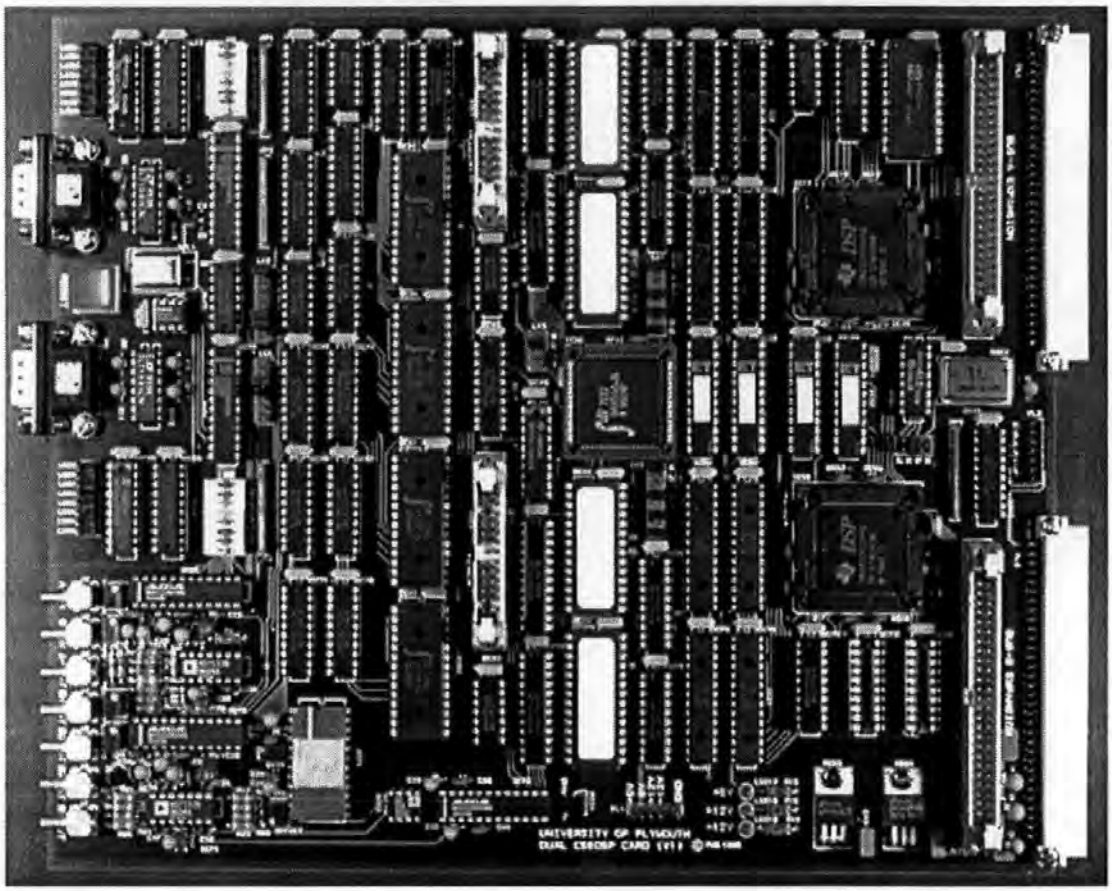


Figure 4-12 Dual-processor DSP board

The board has been developed as a general-purpose platform for the implementation of computationally intensive DSP algorithms, with particular consideration to the implementation of an OFFT.

Each processor provides independent processing channels and can pass information between each other, either via a high-speed synchronous serial port or via dual-port Global RAM. Each channel has 32K x 16 of EPROM and 32K x 16 of expansion memory for both programme space and data space, this is in addition to the processors 9K of on-chip memory.

Channel-A contains a high-speed (500K samples/second) 12-bit ADC and DAC. To reduce interrupt latency, channel-A's analogue converters are buffered with 1K-deep FIFOs. Hence, channel-A has been designed for block processing such as in an OFFT. For continuous block processing, channel-B can also gain access to channel-A's input FIFO. Hence, an OFFT can be processed by one channel, while the other channel is processing input samples from the input FIFO and vice versa, thus operating in a change-over mode, such that no input samples are lost.

Channel-B also has a high speed 12-bit ADC, but has no buffering, since it is intended for sample rate processing (such as filters). Channel-B contains a quad 12-bit DAC providing four independent analogue outputs. These outputs are useful for displaying, on an oscilloscope, intermediate test points within the software algorithms.

Both channels contain an RS232 serial interface and general-purpose Digital IO, LED's and DIL switches. A BIOS is programmed into the onboard EPROMs, to allow a remote dumb-terminal to control DSP software execution, or to display debug information. A standard IEEE 1149.1 (JTAG) emulator interface allows in-circuit software debugging [116]. Both channels have full bus expansion, brought out to ribbon cable connectors and the back plane connectors, allowing memory-mapped expansion.

The board has been designed in modular form, such that the processing power can be increased to more than 320MIPS, by plugging in additional cards. A maximum of 8 processors can communicate with each other, via a synchronous TDM (Time Division Multiplexing) serial port.

Although the board was originally developed as a general purpose DSP platform, it has been subsequently integrated into a commercial satellite hub station receiver, for reception of Internet packets. This application is briefly described in Section 5.3.

In this burst mode application, clock recovery is implemented as a fixed frequency IIR filter, as described in Chapter 2. An OFFT is also implemented, but for carrier frequency estimation and correction (demodulation). A satellite ½ rate encoded BPSK (Binary Phase Shift Keying) modulated signal is down converted to 64KHz IF (Intermediate Frequency), where it is sampled by channel-A's ADC at 256K samples/second. The two DSP processors perform 256-point OFFTs, and operate in flip-flop mode, thereby providing continuous block processing until the signal is detected and frequency corrected. The processors are then released from the OFFT algorithm, for implementing modem (includes the IIR filter clock recovery algorithm at 32Ksymbols/sec) and decoding algorithms [117].

For detailed information on this application, refer to the authors published paper, Appendix 11{7}.

4.3.1. DSP software implementation

DSP code is written in assembly language on a text editor, which is assembled, linked (with a link command file defining the hardware memory map) and then converted to Intel Hex, for programming the onboard EPROMs. Time critical DSP code is downloaded from EPROM to RAM, which is then executed at optimum processor speed (i.e. with zero wait states). The OFFT DSP assembler source code macro is included in Appendix 9.

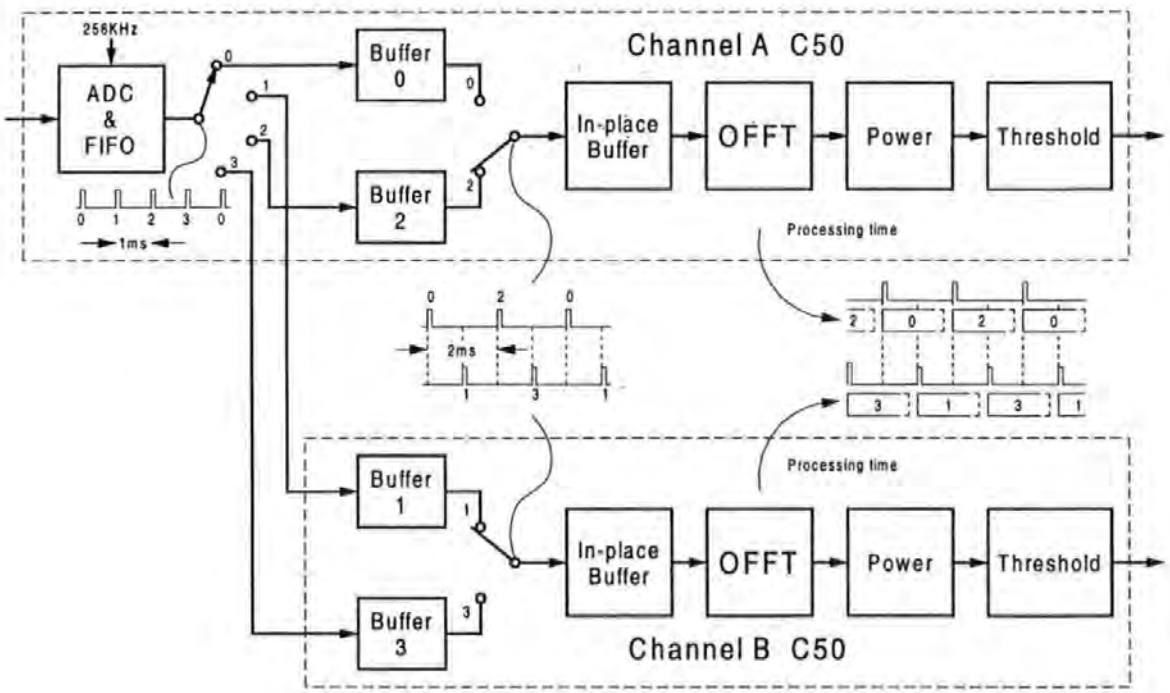


Figure 4-13 OFFT implementation using flip-flop dual buffers

A dual processor approach implementation of a real-time OFFT operating in a flip-flop mode, doubles the execution speed compared to that of a single processor, thereby maximising the acceptable sample rate, as shown in Figure 4-13.

The flip-flop operation allows the processing time of each DSP processor to be staggered, such that one channel is processing the OFFT, while the other is reading in new input samples. Hence, continuous OFFTs are performed with no loss of samples. A 512-point OFFT is executed, in less than 1ms. However, the time taken to calculate the power spectrum and threshold, extends the execution time to approximately 2ms, due to the overhead required for the processors to inter-communicate.

A typical FFT implementation uses an "in-place" algorithm, where the input buffer is over-written by samples from each butterfly. In many applications, input samples must be retained since they contain data symbols, which require subsequent decoding. Therefore, input samples are stored to an input buffer and then copied to an in-place buffer, for processing the OFFT. Once the signal is detected and acquired, the symbols are accessed from the input buffers for post processing. The input buffers are contained in dual-port RAM, to allow symbols to be accessed by either processor.

The FFT and OFFT algorithms are identical in structure, with the exception of the twiddle factors (coefficients). If an FFT is considered as an OFFT with zero offset, then a general algorithm can be constructed, which performs both the FFT and the OFFT, as illustrated in Figure 4-14. The twiddle factors are the same as for a standard FFT, with the exception of the added offset.

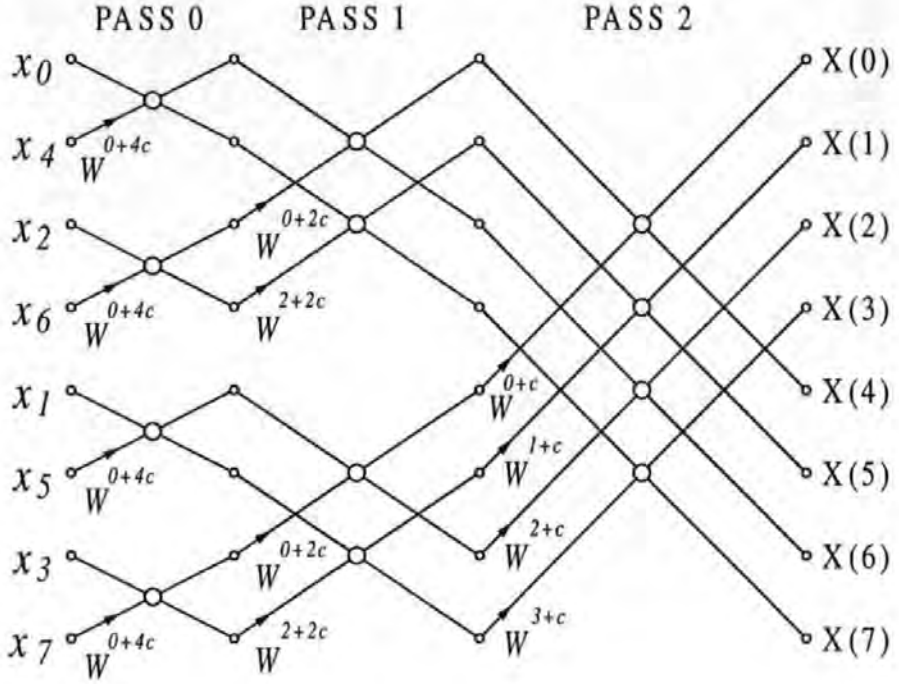


Figure 4-14 Flow diagram for an 8-point OFFT

For an *N*-point OFFT, there are *N-1* complex twiddle factors, containing values between 0 and 1. The TMS320C50 device is a fixed-point processor, therefore the twiddle values are converted into integer 2's complement format. The *N-1* twiddle factors are stored into working memory, with real and imaginary values interleaved.

Memory Address	Twiddle Factor	Pass
0	W^{0+4c}_{real}	0
1	W^{0+4c}_{imag}	0
2	W^{0+2c}_{real}	1
3	W^{0+2c}_{imag}	1
4	W^{2+2c}_{real}	1
5	W^{2+2c}_{imag}	1
6	W^{0+c}_{real}	2
7	W^{0+c}_{imag}	2
8	W^{1+c}_{real}	2
9	W^{1+c}_{imag}	2
10	W^{2+c}_{real}	2
11	W^{2+c}_{imag}	2
12	W^{3+c}_{real}	2
13	W^{3+c}_{imag}	2

Indexing the twiddle factors using indirect addressing, is then a simple matter of setting a pointer to the correct address, at the beginning of each pass.

A C++ programme has been written to automatically calculate and arrange the twiddle factors for an OFFT. The executable program requires the number of points and offset to be entered, and creates a text file (TWIDDLE.Q15). The assembler source code then references a software call to the text file, for inclusion of the twiddle factors. Source code for the executable file (TWIDDLE.EXE) is included in Appendix 9.

Samples from the ADC are stored sequentially into the input buffer and contain only real samples. The OFFT requires complex samples, where the imaginary samples are set to zero. Hence, the input (real) samples are copied into even memory locations of the in-place buffer, while the odd locations are set to zero. The complex samples stored into the in-place buffer, are shuffled during the processing of the OFFT, using bit reversed addressing, such that the output samples appear in natural order. Data shuffling is achieved with zero overhead on the TMS320C50, with the use of the INDX register and BR0+ (add index register with bit reverse carry) command. The interim results produced by the OFFT are scaled to prevent overflow and to maintain maximum dynamic range. Suitable scaling factors are 0.5 for each butterfly calculation, and 4 for the power spectrum output.

After the OFFT has been performed, the power spectrum is calculated and stored in the imaginary (odd) locations of the in-place OFFT buffer.

Memory Address	Input Buffer	Output Buffer
0	x_0	0
1	0	P(0)
2	x_1	0
3	0	P(1)
4	x_2	0
5	0	P(2)
6	x_3	0
7	0	P(3)
8	x_4	0
9	0	P(4)
10	x_5	0
11	0	P(5)
12	x_6	0
13	0	P(6)
...	...	...

Since the imaginary input samples are set to zero, the source code has the first-pass butterfly macro optimised for speed, by reducing the number of complex multiplications.

The last-pass butterfly macro is extended to calculate the power spectrum, with calls to subroutines to determine the peak power in each half of the spectrum.

The OFFT assembler source code macro is included in Appendix 9.

4.3.2. Real-time testing

The illustration of Figure 4-15 shows the OFFT real-time test arrangement. For diagnostic purposes it is desirable to view the power spectrum on an oscilloscope. This is achieved by writing the power output (real) samples to the output FIFO, which is then clocked out as a burst to the high speed DAC.

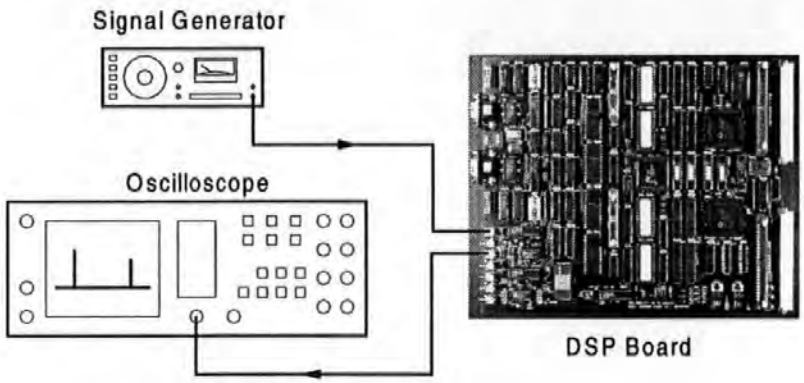


Figure 4-15 The OFFT test arrangement

The photograph of Figure 4-16 shows the practical test system. The dual-processor DSP board is plugged into an integrated power supply unit and a dumb-terminal is connected, to allow remote operation and the display of debug information.

Figure 4-16 Real-time OFFT test system

The photograph of Figure 4-17 shows the real-time power spectrum of the OFFT displayed on an oscilloscope, which is identical to the simulation results shown in Figure 4-5.

Figure 4-17 Real-time OFFT power spectrum output

The first sample written to the FIFO is of full-scale amplitude, so as to provide a pulse to trigger the oscilloscope time-base. As the frequency of the signal generator is varied, the asymmetric response of the OFFT can be viewed. The left-hand peak (just visible) has all its power confined to one frequency bin, and is twice the amplitude to that of the right hand peak, whose power is shared between adjacent frequency bins. Hence, in this case, the left-hand peak (below the Nyquist rate) is detected, and the corresponding bin number is used to determine the frequency estimate. The returned bin number is then be used in a look-up table, to apply new coefficient and gain parameters to the clock recovery IIR filter.

Chapter 5

5. Discussion of results

This final chapter reviews the problems associated with conventional clock recovery schemes and briefly summarises the work carried out in this investigation, with conclusions and recommendations for further work.

5.1. Background summary

Conventional clock recovery architectures are summarised in Section 1.1.7.
A closed-loop clock recovery algorithm is usually employed, for systems that exhibit symbol rate drift. In order that phase and frequency variations may be tracked, many closed-loop algorithms are based upon the basic PLL.

Analysis of the basic PLL in Appendix 1, together with PLL results presented in Appendix 2, demonstrate conflicting requirements. On the one hand, a wide loop bandwidth is required to aid rapid acquisition, but conversely a narrow loop bandwidth is necessary to reject noise and to provide flywheel memory, thereby retaining lock over symbol dropouts and encoded run lengths. Results have shown that whilst a basic PLL works well for a continuous and periodic input signal, performance for burst data applications is poor, both in terms of clock acquisition and flywheel performance. A narrow loop bandwidth restricts the acquisition range to less than ±1% and a training sequence of several hundred symbols is required to

achieve lock. Hence, for burst data, the basic loop is usually modified in a variety of forms, often containing several loops with dynamically controlled loop parameters, thereby aiding acquisition, tracking and flywheel performance. This inevitably leads to a complex system of loops, which can be difficult to implement and analyse.

The most successful of these closed-loop systems is the Quadricorrelator, which extends the acquisition range to ±5% and reduces the training sequence to approximately 30 phase-reversal symbols. Nevertheless, this preamble reduces data access time and occupies capacity, which could otherwise be used for data.

For systems that exhibit an accurate and stable symbol frequency, open-loop clock recovery algorithms are often employed. In general, open-loop algorithms demonstrate rapid clock acquisition and long clock flywheel periods, and are therefore desirable for burst data applications.

The high-Q passive resonator provides good clock recovery performance, with the exception of tracking symbol rate drift, and is often used as a benchmark against which other systems are compared. Unfortunately, practical considerations make passive resonator systems expensive to implement.

The DSP-based clock recovery algorithms, described in this work, are based upon the concept of a high-Q "ringing" filter, which eliminates many of the practical problems associated with the implementation of a passive resonator. The use of DSP allows system parameters to be re-configured "on-the-fly", and this is exploited to provide an adaptive high-Q tracking filter.

Results have shown that a DSP-based clock recovery algorithm provides rapid clock acquisition and long flywheel periods over encoded run lengths. This reduces the constraints currently imposed on recording codes, and could ultimately lead to new more efficient codes, providing increased storage capacity.

In burst mode applications it is necessary to achieve both rapid clock acquisition and good clock flywheel performance during a data burst. However, in some communications systems, at the end of a data burst the flywheel performance is no longer desirable. It is often necessary to quench the clock synchroniser as quickly as possible, to allow acquisition to a new data burst. In analogue systems, quenching of the clock synchroniser is not instantaneous and hence, a guard band is provided between successive burst transmissions.

A DSP-based clock recovery system has the advantage that registers can be immediately reset upon detecting the end of the burst thereby reducing the need for guard bands.

Due to the popularity of monolithic integrated circuits, many clock recovery circuits employ analogue techniques, which suffer from drift due to temperature change and ageing components. Digital solutions are often preferred over their analogue equivalents, since designs are usually more repeatable, reliable and robust.

DSP is a powerful, flexible and cost-effective platform for the implementation of many hitherto analogue processes. Common DSP hardware can be programmed to implement other processes as well as clock recovery (such as filtering, demodulation, error correction and control), thereby providing an integrated system.

The main problem of DSP processor-based systems is that algorithms are implemented sequentially in software (albeit very fast) and hence, a compromise must be met between the sample rate and the number of processor operations which can be performed within a sample period. Hence, DSP-based solutions tend to be used for low data rate systems.

To obtain a speed advantage over DSP processors, digital algorithms may be alternatively implemented using parallel logic, programmed into an FPGA or fabricated into dedicated integrated circuits. FPGAs are now widely available with capacities in excess of 130,000 gates and at speeds of 150MHz [12].

5.2. DSP-based clock recovery

With improvements in digital techniques, clock recovery has been successfully implemented using DSP. Instead of applying a direct digital implementation to an analogue model, it is often more constructive to consider a new approach.

This research investigates the use of DSP to implement novel clock recovery algorithms, all of which are based around the implementation of an IIR filter. Theoretical analysis, computer simulations and real-time DSP implementation have been used, to demonstrate the clock recovery features of a high-Q IIR "ringing" filter, which is exploited to provides rapid clock acquisition and long clock flywheel periods.

Three algorithms have been investigated:

- A fixed frequency clock recovery system.
- An adaptive tracking clock recovery system.
- An adaptive OFFT-based clock recovery system.

In addition, the IIR filter has been implemented within an FPGA and successfully incorporated into the clock recovery system of a commercial satellite receiver, utilising a data sub-carrier on an analogue FM TV broadcast.

5.2.1. Fixed frequency clock recovery system

Of the three algorithms investigated, the fixed frequency clock recovery filter has proven to be the most successful in terms of practical applications. This algorithm has been successfully implemented into a number of commercial satellite data modem receivers, developed at the University of Plymouth. Since early publications by the author, the algorithm has been adopted by a number of readers, who reporting-back, have claimed success in its implementation and performance [118].

This fixed frequency algorithm is best suited for communication systems, which usually contain an accurate and stable symbol rate. Clock acquisition is achieved at the first transition of the retrieved symbols, although a short preamble of 6 phase-reversal symbols is desirable to stimulate the filter sufficiently, so as to combat noise and worst case symbol run lengths. The system can withstand a data loss of 80 symbol periods, without losing clock synchronisation, while cascading filter sections can double this flywheel period. A filter Q of 100 is considered a good compromise, which provides good noise rejection, while still tolerating a symbol rate drift of ±1%. Statistical simulation results have shown good jitter rejection properties, with a PDF of better than 89% at zero phase jitter, when subjected to a gaussian phase jitter distribution.

The fixed frequency IIR filter, and its desirable clock recovery characteristics, form the basis of the adaptive algorithms investigated.

5.2.2. Adaptive tracking clock recovery system

The adaptive tracking system is probably best suited for digital magnetic recording, which can suffer from short-term symbol rate drift. The algorithm uses an iterative approach (sample-by-sample basis) based upon two wide-bandwidth symmetrically placed IIR detection filters, which are used to derive a frequency offset error signal. The error signal is then used to calculate the appropriate coefficients, so as to reposition the centre frequency of the IIR clock recovery filter, to the frequency of the input symbols. Clock recovery performance is then similar to that of the fixed frequency filter system.

In order to obtain identical mirror image detection filters, the detection system is sampled at ¼ of the system sample rate, so as to take advantage of the cosine response of the IIR filter coefficient, which changes sign but is otherwise identical, at ¼ of the sample rate. The tracking filter is placed nominally at the centre of the two detection filters. If the frequency of the retrieved symbols drifts from the nominal, towards one detection filter (and away from the other detection filter) then a higher output is achieved from the in-band detection filter. This imbalance in the detection filter outputs, produces an error signal, which is used to calculate new coefficients, so as to reposition the centre frequency and adjust the gain of the IIR tracking filter.

This algorithm is sensitive to change in system parameters, and of the three algorithms investigated is probably the least attractive to implement. The error

response is dependent upon the detection filter spacing and the encoded run length. Hence, a calibration curve or look-up table must be tailored to these system parameters. Simulation results have shown good tracking performance over the entire acquisition range, for a drifting symbol rate frequency. However, an abrupt change in the drift direction causes a corresponding reduction in the tracking filter amplitude, as the filter adapts to a change in phase direction. The reduction in amplitude does not fall to zero and continuity of the recovered clock is maintained.

Results have shown that the algorithm works well for recording codes with a high clock content, providing an acquisition range of ±10%. However, for codes with long run lengths, the reduction in stimulus to the low-Q IIR detection filters necessitates a reduction in the acquisition range. For codes with a run length of 7, an acquisition and tracking range of ±5% was found to provide reliable results. This is comparable with that of a conventional Quadricorrelator clock recovery system. Whereas the Quadricorrelator requires a training sequence of 30 phase reversal symbols, the DSP-based algorithm generates a recovered clock at the first retrieved transition. However, a short preamble of 6 phase-reversal symbols is recommended to combat noise and worst-case symbol run lengths. For low noise applications, a wide bandwidth fixed filter may prove more desirable, due to its simpler implementation. For example a filter Q of 25 allows ±4% symbol rate drift, while cascading filter sections can offset the reduction in flywheel performance.

5.2.3. Adaptive OFFT-based clock recovery system

The OFFT-based adaptive clock recovery system is best suited for applications that experience large symbol rate offsets.

The asymmetric frequency response of the OFFT, around the Nyquist frequency, provides an advantage over a basic FFT for frequency estimation. The OFFT power output is threshold detected and the resulting frequency estimate used to determine new coefficients, so as to move the adaptive clock recovery filter to the correct symbol rate frequency.

Unlike the tracking filter (which uses an iterative approach) the OFFT uses block processing, which compromises acquisition performance. However, the OFFT does have the advantage that large symbol rate offsets (almost dc to the Nyquist rate) can be detected, hence this algorithm may be used to provide clock recovery where the symbol rate is not known.

This work has refined the frequency estimate resolution, reducing the residual frequency error to just $^1/_8$ of a frequency bin (compared to ½ a bin for an FFT). For an adaptive clock recovery filter with a Q of 100, then a 512-point OFFT provides sufficient resolution to accurately reposition the adaptive filter. For a 512-point OFFT, a preamble of 64-symbols is required in order to determine the frequency estimate. The coefficients are then calculated and immediately applied to the adaptive filter, which is snapped to the correct symbol rate frequency.

Clock recovery performance is then identical to that of the fixed frequency filter system. The OFFT can then continue to monitor the symbol rate, providing frequency estimates every 32 symbols. Therefore, assuming that the symbol rate frequency has not drifted significantly during this 32-symbol period, then the adaptive filter is capable of tracking symbol rate-drift.

This algorithm is computationally intensive, requiring (in the real-time test system) two processors operating in a flip-flop mode, to provide continuous frequency monitoring.

For burst mode applications, it is unlikely that the symbol frequency will drift significantly over the duration of a short burst. Therefore, just a single frequency estimate can be provided on the preamble at the start of the burst, and the appropriate coefficients loaded into the adaptive filter. The processors can then be released from symbol frequency estimation and employed for other processes, such as decoding and error correction. This algorithm-sharing approach, operating on common hardware, can result in a more financially viable system.

The OFFT algorithm has been successfully used in a burst mode satellite modem, where it is employed for frequency estimation and demodulation of a BPSK signal.

5.2.4. FPGA implementation

A processor-based system is compromised by the speed of software operation, the program of which must be executed sequentially. To achieve a high-speed solution, whilst maintaining a degree of programmability, the IIR filter has been implemented into a low-cost Xilinx FPGA. The configuration programme is stored in ROM and downloaded to the FPGA at power-up, although it is also possible to reconfigure the gate array during operation.

The IIR filter requires coefficient multiplications, which are easily achieved in a DSP processor with a dedicated multiply register. However, multiplication using discrete logic (as in a gate array) is time consuming and occupies a lot of silicon. To avoid using conventional binary multipliers, the filter coefficients have been successfully implemented using a novel technique of binary-weighted coefficients. This is achieved by bit shifting the data words and adding or subtracting, to obtain the required coefficient values. Hence, the filter coefficients are implemented automatically at the sample rate without the need for sub-multiplication processing. This technique results in an approximate reduction of 50% in silicon utilisation and a 50% increase in speed performance.

The binary-weightings do impose quantization restrictions in the selection of the filter coefficients, however these restrictions are offset by the choice of sample rate, which provides fine-tuning to the desired centre frequency.

The FPGA implementation has been tailored to a commercial satellite data modem, operating at a sample rate of 2.59MHz and a data rate of 100Kbits/sec. The product is intended for consumer use and for cost reasons, the design has been targeted to the lowest specification device (2000 gates, 50MHz). Larger and faster devices are available (130,000 gates, 150MHz) and hence, there is much potential for improving performance and increasing sample rates.

A 16-bit dynamic range has been used throughout the filter, and simulation results have demonstrated negligible quantization errors for 16-bit coefficients. Further quantization analysis has shown that the coefficient can be reduced to just 9 bits, after which the system becomes unstable.

Due to the discrete binary-weighted coefficient steps, this approach would have limited value for an adaptive system. A further complication is that reprogramming the device "on-the-fly", requires a configuration time of 2msec. During this period the device is inoperable and would result in loss of clock synchronisation.

A tracking filter may be possible by retaining a fixed FPGA configuration and instead providing coherent adjustment of the sample rate frequency, to re-tune centre frequency of the filter. This is an idea that may warrant further investigation.

5.3. Applications

Algorithms discussed in this work have been successfully integrated into a number of commercial, satellite data communication applications. Section 2.5.2 briefly describes a distance learning application, where computer binary files are broadcast by a TV presenter to remote students, utilising a data sub-carrier on an analogue FM TV broadcast, as illustrated in Figure 5-1.

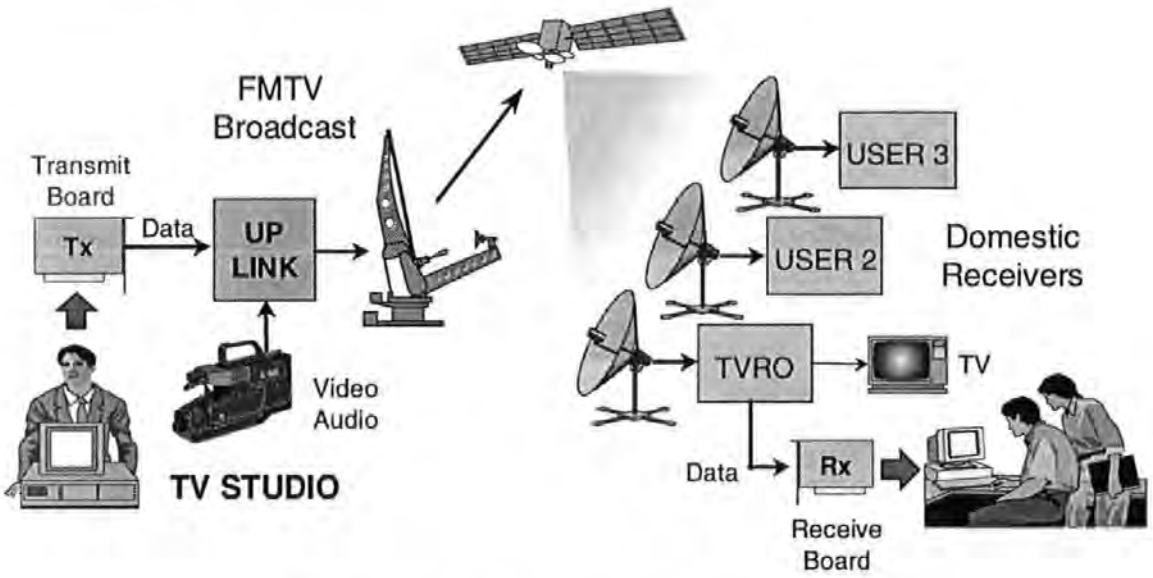


Figure 5-1 Distance learning system

In this application, the author has developed a PC-based digital modem receive board, which implements a fixed frequency clock recovery algorithm. Clock recovery at 100KHz is achieved within an FPGA, which is sampled at 2.59MHz, as described in Section 2.5.

For a full description of this application, readers are referred to the author's published paper [Appendix 11{4}].

As a result of this in-house (University of Plymouth) application, commercial investment has resulted in a new application, involving the high-speed delivery of internet data to client computers. This application is illustrated in Figure 5-2.

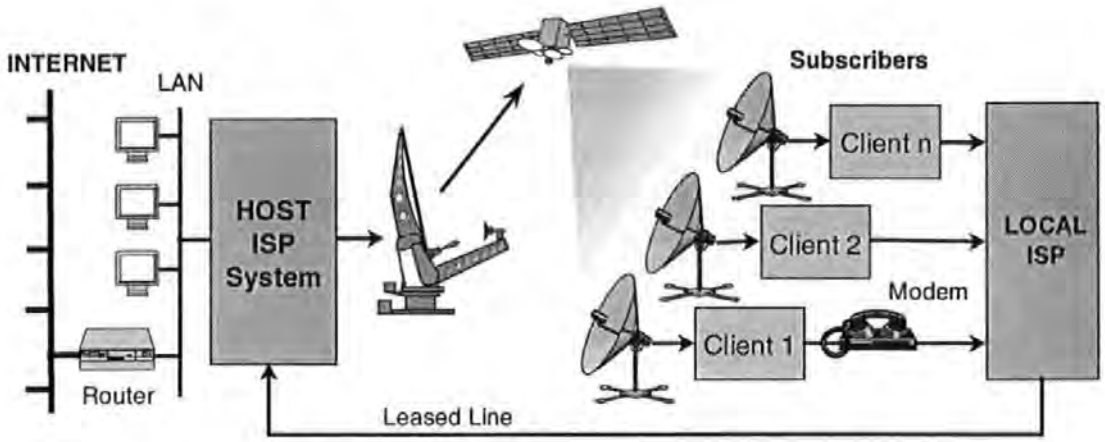


Figure 5-2 Satellite Internet Service Provision

The Satellite Internet Service Provision (SISP), allows client's outgoing packets, which are relatively few in number, to be routed via the slow telephone network to a satellite up-link, which has access to the internet backbone. The return packets are then delivered at 100Kbits/sec utilising a data sub-carrier on an FM TV broadcast, resulting in much faster delivery of complex www (world-wide-web) pages and other Internet services.

The SISP system has been successfully implemented commercially. BT at Madley, in the U.K., has provided the satellite up-link and the SISP data has been transmitted on a 7.74MHz sub-carrier on the EBN (European Business News) TV channel. Clients located in Jordan in the Middle East, have successfully accessed internet data, received from Eutelsat's Hotbird-I satellite. For further information on this application, refer to the author's published paper [Appendix 11{5}].

The SISP system illustrated in Figure 5-2, shows a one-way satellite system, where the (high-speed) internet data are broadcast on an analogue FM TV signal and the (low-speed) return-link is provided by a conventional phone modem. The natural progression to this hybrid scheme is to provide a full interactive satellite system.

Based upon the success of the SISP system, a commercial contract has resulted in the development, at the University of Plymouth, of a two-way Satellite Internet System, as shown in Figure 5-3.

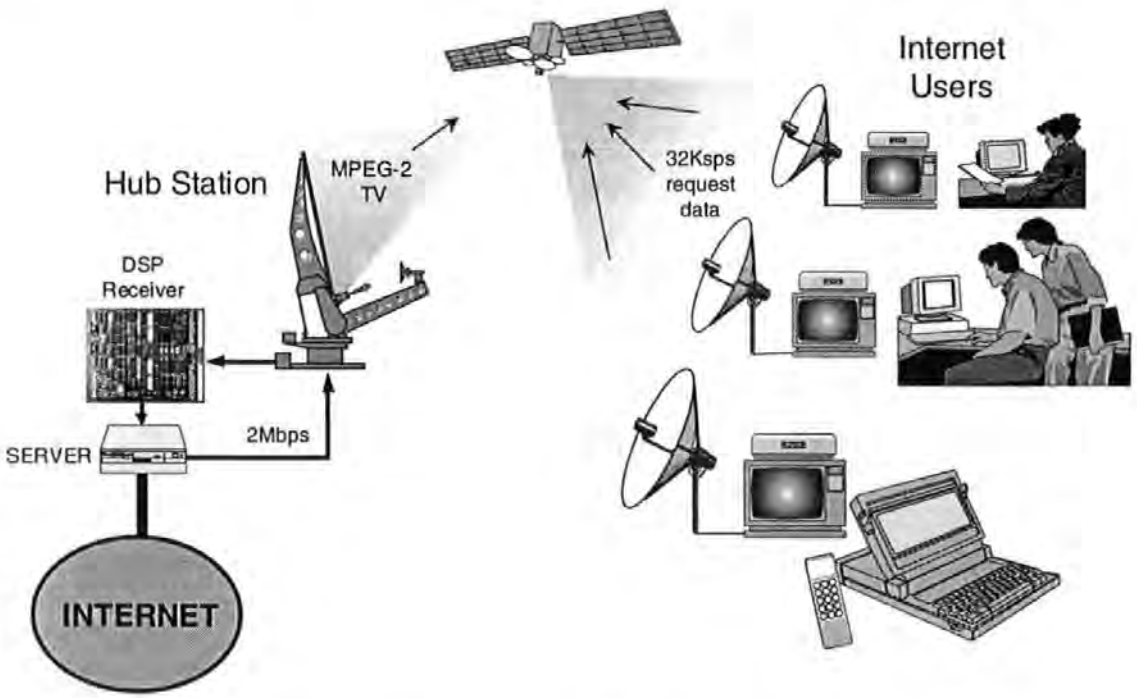


Figure 5-3 Two-way Satellite Internet System

While in general most systems are striving to increase data rates, in this unique application, the data rate of the return-link must be kept low, due to bandwidth limitations, and is therefore ideal for implementation using DSP.

Satellite capacity is purchased both in terms of signal power and bandwidth. In satellite TV broadcasting, the satellite transponder is dominated by the power and bandwidth of the TV channels. To allow interactive services, narrow return-link channels are placed within the spare capacity of the transponder. These low data rate channels are of low power (typically 7dB signal/noise) and narrow bandwidth, and therefore contribute little to the transponder payload. The return-link channels carry little traffic, since they are only used for request and acknowledgement data, hence, 32K symbols/sec is adequate.

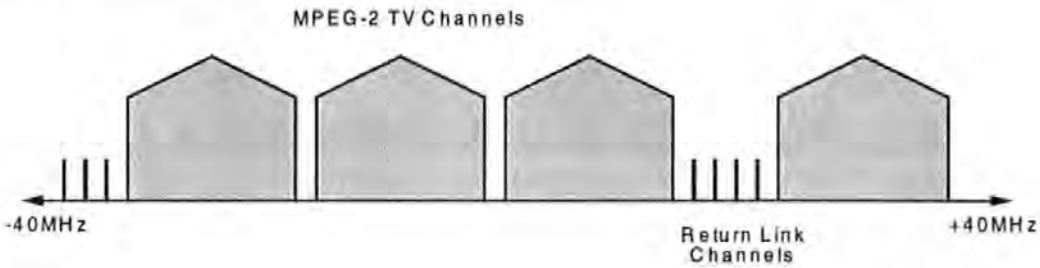


Figure 5-4 Satellite transponder frequency plan

A world-wide-web browser such as Netscape is used to request Internet pages. The request data are routed via a satellite return-link channel to a hub-station receiver, which in turn is connected to the Internet backbone, via a server computer.

The DSP receiver (consisting of the dual-processor DSP board described in Chapter 4 and Appendix 10) implements algorithms described in this thesis. A 256-point OFFT estimates the carrier frequency and is used to provide demodulation, while clock recovery is implemented using the fixed frequency IIR filter algorithm.

A preamble of just 32 symbols is used, to establish carrier frequency estimation, clock recovery and word synchronisation. DSP algorithms successfully acquire and decode the request packets, which are routed onwards to the Internet sites.

The requested internet pages from the accessed sites are routed back to the satellite hub-station, where they are multiplexed into the MPEG-2 digital TV transport-stream and broadcast at high data rates (up to 2Mbits/sec). All users receive the broadcast, but Ethernet address filtering only allows the client that requested the pages to decode the data, which is displayed on the users web browser. Hence, the system traffic is asymmetric, where request data are relatively few and can be sent at low data rates, but the requested internet pages can be huge and therefore benefit from delivery at high data rates.

This project is a world first and has received much acclaim from the satellite communications industry. A pilot scheme, funded by the BNSC (British National Space Centre) for satellite multimedia applications, has been successfully completed, the final report of which is referenced in [119].

Readers are also referred to the author's published paper [Appendix 11{7}].

5.4. Further work

This work has involved development of the algorithms, with real-time DSP implementation tailored to specific applications. Given available devices, further work can be carried out to determine the maximum sample and data rates possible. The algorithms may be optimised for speed, by for example reducing the number of samples per symbol and pipelined optimisation of DSP code.

In the case of an FPGA implementation, the speed is limited by the accumulative ripple-carry propagation delay of adders, used for the binary-weighted coefficients. The use of fast (150MHz) devices and look-ahead carry techniques, can significantly improve speed performance. The use of discrete binary-weighted coefficients, used in the FPGA, impose limitations for use in an adaptive system. However, in principle a tracking system is possible by providing coherent adjustment of the sample rate frequency derived from a VCO, resulting in repositioning the centre frequency of the filter.

While the algorithms have been simulated using recording codes, real-time testing on data magnetic recording systems could be implemented. Given that the algorithms demonstrate good acquisition and flywheel performance, the constraints currently imposed on recording codes can be relaxed. Hence, new recording codes may be developed, which could provide improved storage capacity.

Chapter 6

6. Conclusions

Taking the investigation as a whole, the concept of using DSP for clock recovery has proven successful for low data rate systems. The algorithms investigated, all of which are based upon a high-Q "ringing" IIR filter, have demonstrated good clock recovery properties, providing noise rejection, rapid acquisition and good flywheel performance. The author believes that a contribution to knowledge has been made in the following areas:

- Analysis, simulation and real-time testing have shown superior performance over that of conventional feedback systems, particularly in terms of clock acquisition. This considerably reduces the length of the synchronising preamble, improving access time and increasing storage capacity. Also the long flywheel performance demonstrated allows use with channel codes containing long run lengths, and maintains clock synchronisation over severe data dropouts.

- The fixed frequency algorithm has been simulated and implemented in real-time DSP processors. Clock acquisition is achieved at the first retrieved symbol transition and a symbol rate frequency offset of ±1% can be tolerated. The recovered clock flywheels over an 80-symbol dropout, without losing synchronisation. Cascading filter sections can extend this flywheel period. The

algorithm has been successfully incorporated into a number of commercial satellite data modem projects, and has proven reliable, even at low signal to noise ratios.

- The fixed frequency IIR filter algorithm has been implemented within an FPGA, using fast binary-weighted coefficient multipliers, providing a single chip solution to digital clock recovery. This technique provides both a speed advantage and a reduction in silicon utilisation. Hence, GEC-Plessey Semiconductors (at Plymouth) are investigating the use of this technique, for use in high-speed digital filter integrated circuits.

- An adaptive iterative tracking algorithm, based upon two symmetrically placed frequency offset detection filters, has been investigated. This algorithm demonstrates short-term tracking capabilities over a limited symbol frequency offset range. Simulation results have demonstrated improvements over that of a basic PLL, both in terms of acquisition time and range. Clock acquisition is achieved at the first retrieved symbol transition and the acquisition range can be up to ±10%. Also, the system demonstrates good clock flywheel performance, maintaining clock synchronisation over a dropout of 80 symbol periods.

- An adaptive algorithm, using an OFFT as a symbol rate frequency estimator, has proven successful. However, the block processing involved reduces acquisition

performance to 64 symbols. The system is capable of detecting symbol rate frequencies, from almost dc to the Nyquist frequency, and is therefore suitable for clock recovery of an unknown symbol rate. This algorithm is capable of tracking long-term symbol rate drift, updating the adaptive filter every 32 symbols. The OFFT algorithm has been implemented in real-time DSP and has been successfully integrated into a satellite burst-data modem, where it is used for carrier frequency estimation.

This investigation has resulted in the publication of seven refereed papers, relating to the developed algorithms and their applications. The referees remarks [120] and the subsequent feedback from readers [118], have been both positive and supportive.

The contribution to knowledge of this investigation has resulted in the implementation of the developed algorithms, applied to satellite data communication systems. These experimental systems have successfully undergone pilot-schemes to demonstrate the technology.

The following organisations have co-operated in this experimental activity: BNSC (British National Space Centre), ESA (European Space Agency), Eutelsat, Armstrong Electronics Ltd., Novella Satcoms Ltd. and Communicado Data Ltd.

References

[1] Wade G. *"Signal Coding and Processing"* 2nd Edition. Cambridge University Press. 1994. ISBN: 0-521-42336-8.

[2] Ifeachor E.C. and Jervis B.W. *"Digital Signal Processing, A Practical Approach"* Addison Wesley. 1993. ISBN: 0-201-54413-X.

[3] Eitan B., Caywood J., Lai S. and Sheperd W. *"Non-Volatile Memory Technoligies"* IEEE PAL Video. ISBN: 0-7803-1538-3.

[4] Watkinson J.R. *"CD: The 600 megabyte ROM"* Electronics and Wireless World. N°93, pp.1046-1049. 1987.

[5] Wood R.W. *"Magnetic Recording Systems"* Proceedings of the IEEE. Journal Vol.74, N°11, pp.1557-1569. 1986.

[6] Peek J.B.H. *"Communication aspects of the Compact Disk digital audio system"* IEEE Communications Mag. N°23, pp.7-15. 1985.

[7] Nakajima H. and Odaka K. *"A rotary-head high-density digital audio tape recorder"* IEEE Transactions on Consumer Electronics. CE-29, pp.430-437. 1983.

[8] Benoit H. *"Digital Television: MPEG-1, MPEG-2 and principles of the DVB System"* John Wiley & Sons. 1997. ISBN: 0-340-69190-5.

[9] Hahn H. and Stout R. *"The Internet Complete Reference"* Osborne McGraw-Hill. 1996. ISBN: 0-07-881980-6.

[10] Garg V.K. and Wilkes J.E. *"Wireless and Personal Communication Systems"* Prentice Hall. 1996. ISBN: 0-13-234626-5.

[11] Texas Instruments *"TMS320C8x Features"* Web Page: http://www.ti.com/sc/docs/dsp/products/c8x/features.htm

[12] Xilinx Publications. *"The Programmable Logic Data Book 9/96"* Document N° PN0010303.

[13] IEE Colloquium *"New synchronisation techniques for radio systems"* Professional Group E8 (Radio Communication Systems). Nov 1995. UK ISSN: 0963-3308. Ref. N°1995/220.

[14] Kent A., Williams J.G. (Exec Editors) *"Encyclopaedia of Computer Science and Technology"* Marcel Dekker Inc. Vol.24, Supp.9, Donnelly T., pp.239-285.

[15] Gardener F.M. *"Phaselock Techniques"* 2nd Edition. John Wiley & Sons. 1979. ISBN: 0-471-04294-3.

[16] Watkinson J. *"The Art of Digital Audio"* Focal Press. 1988. ISBN: 0-240-51270-7.

[17] VTC Inc. *"PRML Read Channel with PR4, 8/9 ENDEC, 4-Burst Servo. SSI32P4903 Preliminary Data Sheet"* Document N° VM65060 Rev 03.

[18] Myrica (UK) Ltd. *"Scotsman Model MCD750 Drive. Proposed Specification"*

[19] Wright G. and Stevens W.R. *"TCP/IP Illustrated. The implementation"* Vol.2. Addison-Wesley, 1995. ISBN: 0-301-63354-X.

[20] Uwano S. and Watanabe K. *"Symbol Synchronization Performance using Loop Switching Type Quenching Method"* Globecom'97 Proceedings. Vol.3, pp.1199-1203. ISBN: 0-7803-4198-8/97.

[21] Stremler F.G. *"Introduction to Communication Systems"* 3rd Edition. Addison Wesley. 1990. ISBN: 0-201-51651-9.

[22] Seigel P.H. *"Recording Codes for Digital Magnetic Storage"* IEEE Transactions on Magnetics. Vol. MAG-21, N° 5, pp.1344-1349. Sept 1985.

[23] Mathcad Inc. *"MathCad 6.0 Uses Guide"* Math Soft Europe. 1995. Publication N°VP6.00-231195-11899-AD.

[24] Orcad Systems Corporation *"Verification and Simulation Tools, Users Guide"* 1989. Document N°2880400B Ver.1.12.

[25] Alta SPW *"The DSP Framework"* Release 3.1.

BDE Users Guide/Tutorial.	Document N° 8010.
Designers/BDE Users Guide.	Document N° 8012.
Signal Calculator Users Guide.	Document N° 8013.
Communications Library Reference.	Document N° 8015.
FDS Users Guide.	Document N° FDS8000.

[26] Texas Instruments Publications *"TMS320C25 Users Guide"* 1986. ISBN: 2-86886-014-1.

[27] Tomlinson M. *"The Offset Fast Fourier Transform"* MOD SRDE Document N° 76025.

[28] Texas Instruments Publications. *"TMS320C5x Users Guide"* 1993. Document N° 2547301-9761 Rev D. SPRU056B.

[29] Savage J.E. *"Some simple self-synchronizing digital data scramblers"* Bell Systems. Technical Report, pp.449-487. Feb 1967.

[30] ANSI Standard. *"Recorded Magnetic Tape for Information Interchange"* (Manchester Code 1600 char/in) ANSI Standard X 3.39-1973.

[31] Miller J.W. *"Transmission System"* US Patent N° 3,108,261. Oct 1963.

[32] Miller J.W. *"The Miller Squared Code"* US Patent N° 4,027,335. May 1997.

[33] Franaszek P.A. *"Run Length Limited Variable Length coding with Error Propagation Limitation"* (RLL2,7 code). US Patent N° 3,689,899. Sept 1972.

[34] Itoh Y. and Nishimura K. *"Modulation-Demodulation Methods for digital Magnetic Recording"* Nikkei Electron. Vol.12, N°11, pp.1-164. 1978.

[35] Takasaki Y. *"Timing Extraction in Baseband Pulse Transmission"* IEEE Transactions on Communication Technology. Vol.COM-20, pp.877-884. Oct 1972.

[36] Greer W.T. *"Digital Phased-Locked Loops move into Analogue Territory"* Electronic Design. Vol.30, N°8, pp.95-100. March 1982.

[37] Bures K.J. *"Understanding timing recovery and jitter in digital transmission systems"* R.F. Design. Vol.15, N°11, pp.42-44, 47, 49, 51-53. Oct 1992. ISSN: 0163-321X.

[38] Briggman D., Hanke G., Langmann U. and Pottbacker A. *"Clock recovery circuits up to 20 Gbit/s for optical transmission systems"* IEEE MTT-S. International Microwave Symposium. Digest of Technical Papers, pp.1093-1096. 1994.

[39] Smithson P.M. and Tomlinson M. *"The Development of an Operational Computer Based Satellite Data Broadcasting System"*
IEE 10th International Conference on Digital Satellite Communications. ICDSC-10, Vol.2, pp.405-412. May 1995. ISBN: 0-85296-6393.

[40] Doi T.T., Tsuchiya Y., Tanaka M. and Watanabe N. *"A format of stationary-head digital audio recorder covering a wide range of applications"*
Journal Audio Engineering Society. Vol.28, p.931. Dec1980. Reprint 1677(H6).

[41] Siemens Matsushita *"SAW Components Data Book"*
Document N°B420-P2026-X-X-7600, pp.325-331. 1996.

[42] AEL Crystal Filters *"Frequency Control Products"* 1998.
Monolithic Crystal Filters, p.86

[43] Lai B. and Walker R.C. *"A monolithic 622Mb/sec clock extraction and data retiming circuit"* ISSCC Dig. Tech. Papers, pp.144-145. Feb 1991.

[44] Fleischmann B., Ruile W. and Riha G. *"Raleigh-mode SAW filters on quartz for timing recovery at frequencies above 1 GHz"* IEEE Ultrasonic Symposium. Conference Digest, pp.163-167. July 1984.

[45] Way W.I. *"High speed circuit technology for multi-gigabit/sec communication systems"* IEEE International Conference on Communications. Conference Proceedings, pp.313-318. June 1988.

[46] Gupta S.C. *"Phase Locked Loops"*
Proceedings of the IEEE. Vol.63, pp.291-306, Feb 1975.

[47] Lindsey W.C. and Chie C.M *"A Survey of Digital Phase-Locked Loops"*
IEEE Proceedings. Vol. 69, N°30, pp.410-431. April 1981.

[48] Razavi B. *"Monolithic Phase-Locked Loops and Clock Recovery Circuits"*
IEEE Press. 1996. ISBN: 0-7803-1149-3.

[49] Mullard Technical Handbook, *"High-speed CMOS logic HC/HCT Family, 4046 Monolithic PLL, 74HCT297 Digital PLL"* Book 4, Integrated Circuits, Part 5. 1986. Publication N°806055.

[50] Sharpe C.A. *"A 3-state phase detector can improve your next PLL design"*
EDN, pp.55-59. Sept 1976.

[51] Cessna J.R. and Levy J.D.M *"Phase noise and transition times for binary quantized digital phase-locked loop in white Gaussian noise"*
IEEE Transactions on Communication Technology. Vol. COM-20, pp.94-104. April 1972.

[52] Natali F.D. *"Accurate digital detection of angle modulation signals"*
EASCON Rec., pp.407-412. Oct 1968.

[53] Johnson M.G. and Hudson E.L. *"A variable delay line PLL for CPU-coprocessor synchronization"* IEEE Journal on Solid State Circuits. Vol.23, pp.1218-1223. Oct. 1988.

[54] Sonntag J. and Leonwich R. *"A monolithic CMOS 10MHz DPLL for burst mode retiming"* IEEE ISSCC Dig. Tech Papers, pp.194-195. Feb 1990.

[55] Kim B., Helman D.N. and Gray P.R. *"A 30-MHz Hybrid Analogue/Digital Clock Recovery Circuit in 2-μm CMOS"* IEEE Journal of Solid-State Circuits. Vol.SC-25, pp.1385-1394. Dec 1990.

[56] Miyazawa S. *"A BiCMOS PLL-Based Data Separator Circuit with High Stability and Accuracy"* IEEE Journal of Solid-State Circuits. Vol.SC-26, pp.116-121. Feb 1991.

[57] Shaeffer C.F. *"The zero-beat method of frequency discrimination"*
Proceedings of the IRE. Vol.30, pp.365-367, Aug. 1942.

[58] Richman D. *"Color-Carrier Reference Phase Synchronisation Accuracy in NTSC Color Television"* IRE proceedings. Vol.42, pp.106-133. Jan 1954.

[59] Bellisio J.A. *"A New Phased-Locked-Loop Timing Recovery Method for Digital Regenerators"*
IEEE International Communications Conference. Vol.1, pp.10-20. June 1976.

[60] Gardener F.M. *"Properties of Frequency Difference Detectors"*
IEEE Transactions on Communications. Vol.COM-33, N°2, pp.131-138. Feb. 1985.

[61] Messershmitt D.G. *"Frequency detectors for PLL acquisition in timing and carrier recovery"* IEEE Transactions on Communications. Vol.COM-27, pp.1288-1295. Sept 1979.

[62] Cordell R.R. *"A 50MHz Phase and Frequency Locked Loop"*
IEEE Journal of Solid-State Circuits. Vol. SC-14, pp.1003-1009. Dec 1979.

[63] Razavi B. and Sung J. *"A 2,5-Gb/sec 15-mW BiCMOS clock recovery circuit"* IEEE Symposium on VLSI Circuits Dig. Tech. Papers, p.83. 1995.

[64] Banu M. and Dunlop A. *"A 660Mb/s CMOS Clock Recovery circuit with Instantaneous Locking for NRZ Data and Burst-Mode Transmission"* IEEE International Solid-State Circuits Conference. ISSC, pp.102-103. Feb. 1993.

[65] Integrated Device Technology *"Product and Company Information on CD"* CD Ref. No. CD.COFP.00027. 1997.
Web Page: http://www.idt.com/products/fifo/Welcome.html

[66] Van Trees H.L. *"Detection, Estimation and Modulation Theory"* Part 1. Wiley & Sons Publishers. 1968.

[67] Bennett W.R. *"Statistics of Regenerative Digital Transmission"* BSTJ. Vol.37, pp1501-1542. Nov 1958.

[68] Stiffler J.J. *"Maximum Likelihood Symbol Synchronization"* Jet Propulsion Labs. SPS 37-35, Vol.4, pp.349-357. Oct 1965.

[69] Stiffler J.J. *"Theory of Synchronous Communications"* Prentice –Hall. 1971. Chapter 7. ISBN: 0-13-914739-X

[70] Mengali U. *"A Self Synchronizer Matched to the Signal Shape"* IEEE Transactions. AES-7, pp.686-693. July 1971.

[71] Layland J.W. *"Telemetry Bit Synchronization Loop"* Jet Propulsion Labs. California. SPS 34-46. Vol.3, pp.204-215. July 1967.

[72] Lindsey W.C. and Simon M.K. *"Telecommunication Systems Engineering"* Prentice-Hall. 1973. Chapter 9. ISBN: 0-486-66838-X.

[73] Simon M.K. *"Nonlinear Analysis of an Absolute Value Type of an Early-Late Gate Bit Synchronizer"* IEEE Transactions. Vol.COM-18, pp.589-596. 1970.

[74] Lindsey W.C. and Tausworth R.C. *"Digital Data-Transition Tracking Loops"* Jet Propulsion Labs. JPL SPS 37-50, Vol.3, pp.272-276. April 1968.

[75] Tausworthe R.C. *"Analysis and Design of the Symbol-Tracking Loop"* Jet Propulsion Labs. SPS 37-51, Vol.2, pp.145-147. May 1968.

[76] Gallopoulos A., Heegard C. and Siegel P. *"The power spectrum of run-length-limited codes"* IEEE Transactions on Communications. 1989. COM-37 Vol.9, pp.906-917.

[77] CCITT. *"Data Communications over the Telephone Network"* The International Telegraph and Telephone Consultative Committee Yellow Book. Vol.8, Fascicle 8.1, Rec.V.22, pp.84-85. Nov.1980.

[78] Lucky R.W., Salz J. and Weldon E.J. *"Principles of Data Communication"* McGraw-Hill. 1968.

[79] Jorgensen F. *"The complete handbook of magnetic recording"* Tab books. 4th Edition. 1996. ISBN: 0-070033045-X

[80] Mallison J.C. *"The Foundations of Magnetic Recording"* Academic Press 1987.

[81] Watkinson J. *"Coding for Digital Recording"* Focal Press. 1990. ISBN: 0-240-51293-6.

[82] Mallison J.C. and Miller J.W. *"Optimal Codes for Digital Magnetic Recordings"* The Institute of Electronic and Radio Engineers. Vol.47, N° 4, pp.172-176. April 1997.

[83] Ta-Mu Chien. *"Upper bound on the efficiency of DC-constrained codes"* Bell Systems. Technical Journal 49, pp.2267-2287. Nov 1970.

[84] Lindholm D.A. *"Power Spectra of Channel Codes for Digital Magnetic Recording"* IEEE Transactions on Magnetics. Vol.MAG-14, N°5. Sept 1978.

[85] Wood R., Lindholm D.A. and Hagg R.M. *"The bandwidth of magnetic record/reproduce heads"* IEEE Transactions on Magnetics. Vol.MAG-21, N°5, pp.1566-1568. Sept 1985.

[86] Davey P.J., Donnelly T. and Mapps D.J. *"Pulse slimming in Magnetic Recording using DSP Techniques"* Microprocessing and Microprogramming 37. Conference Proceedings Euromicro 92, pp.73-76. 1992.

[87] Nakagawa S. *"A Study on Detection Methods of NRZ Recording"* IEEE Transactions on Magnetics. Vol.MAG-16, N°1, pp.104-110. Jan 1980.

[88] Justesen J. *"Information rates and power spectra of digital codes"* IEEE Transactions on Information Theory. Vol.28, N°3, p.457. March 1982.

[89] Yokoyama K. *"Digital Video Tape Recorder"* NHK Technical Monograph. N°31. March 1982.

[90] Watkinson J. *"The Technology of the Ampex DCRSi"* IERE Conference on Video, Audio and Data Recording. University of York. March 1998.

[91] Engh J.T. *"The IBM Diskette and Diskette Drive"* IBM Journal. Res.Devel, Vol.25, N°5, pp.710-710. Sept 1981.

[92] SMPTE 226M *"Proposed American National Standard for component digital television recording 19mm type D-1 cassette format"*

[93] EBU Document *"Standard for recording digital television signals on magnetic tape in cassettes"* Technical Publication N°3252-E.

[94] Watkinson J. *"Splice-handling mechanism in DASH format"* 77th Audio Engineering Society Convention. 1985. Reprint N°1677 (H6).

[95] EBU Document. *"Specification of the Radio Data System (RDS) for VHF/FM Sound Broadcasters"* Technical Publication N° 3244. 1983.

[96] Feedback Telecommunications *"CloudFile 2000 Weather Satellite Receiving System 80-001"* 1994. Document N° 1160-80001.

[97] Kim G.Y. *"Broadband LAN technology"* Artech House. 1998. ISBN: 0-89006-291-9.

[98] Brush R. *"Design considerations for the D-2 PAL composite DVTR"* 7th IERE Video, Audio and Data Recording Conference. IERE Publication N°79, pp.141-148. 1988.

[99] Howell T. *"Channel Capacity for the IBM 3380"* International Symposium on Information Theory. IEEE Ann Arbor. 1986.

[100] Kobayashi H. *"A Survey of Coding Schemes for Transmission or Recording of Digital Data"* IEEE Transactions on Communications. Vol.COM-19, N°6, pp.1087-1100. Dec 1991.

[101] Mackintosh N,D. *"The Choice of Recording Codes"* IERE Conference on Video and Data Recording, pp.77-119. July 1979.

[102] Texas Instruments Publications. *"Digital Signal Processing Applications"* 1986. ISBN: 2-86886-009-5.

[103] Texas Instruments Publications *"The TTL Data Book for Design Engineers"* Vol.2, 74LS297 DPLL, pp.783-788. 1989. ISBN: 3-88078-078-1.

[104] Electronic Design. *"Applied Transform Theory"* pp.108-128. May 1995.

[105] Stroud K.A. *"Engineering Mathematics"* Macmillan Press. 1992. ISBN: 0-333-44887-1.

[106] Texas Instruments Publications. *"TMS9914A General Purpose Interface Bus (GPIB) Controller"* Document N° MPL7. Feb. 1984

[107] Xilinx Publications.
"Logic Cell Array Development System. Reference Manual" Vol 1.-0430398 01
"Logic Cell Array Development System. Reference Manual" Vol 2.-0430399 01
"Logic Cell Array Users Guide" 0430402 01
"Logic Cell Array Macro Library" 0430402 01

[108] Wade J.G., Roberts A. and Williams G. *"Multiplier-Less FIR Filter Design using a Generic Algorithm"* IEE Proceedings. Visual Image and Signal Processing. Vol.141, N°3, pp.175-180. 1994.

[109] Roberts A. and Wade J.G. *"Optimization of Finite Wordlength Filters using a Genetic Algorithm"* IEE Proceedings. ACEDC'94, pp.37-43. 1994.

[110] Wade J.G. and Roberts A. *"Ordering of FIR filter structures"* Electronic Letters. Vol. 30, N°17, pp.1293-1394. Aug 1994.

[111] Marven C. and Ewers G. *"A Simple Approach to digital Signal Processing"* John Wiley & Son. 1996. ISBN: 0-471-15243-9.

[112] Haykin S. *"Adaptive Filter Theory"* Prentice-Hall. 1986. ISBN: 0-13-397985-7.

[113] NG, Leung, Chung, Luk and Lau. *"Adaptive IIR Filter – A Genetic Search Approach"* IEEE. Signal Processing, pp.38-46. Nov 1996.

[114] Wade J.G. *"Offset FFT and its implementation on the TMS3020C25 processor"* Microprocessors and Microsystems 12, Vol.2, pp.76-82. 1998.

[115] Rabiner L. and Gold B. *"Theory and applications of Digital Signal Processing"* Prentice-Hall. 1975. ISBN: 0-13-914101-4.

[116] Texas Instruments Publications. *"TMS320C5x C Source Debugger Users Guide"* 1994.
Document N° 2547297-9761. Rev F. SPRU055B

[117] Slader J.T. *"TMS320C50 Satellite Data Receiver – Progress Report"* 1998.
University of Plymouth. Satellite Communications Research Centre.

[118] Reweben E. *"Personal Communications"* 1998. Synetcom, California.

[119] Tomlinson M., Winders R. and Hendry M. *"SMADP (Satellite Multimedia Applications Demonstration Programme) 2-Way Internet/Intranet Services by Satellite Project Report"* University of Plymouth.
Satellite Research Centre. 1998.

[120] Plumb (Prof.) B. and Globecom Referees *"Written personal communications relating to refereed publications"* 1994.

Bibliography

A Abbott W.L. and Cioffi J.M. *"Timing recovery for adaptive decision feedback equalization of the magnetic storage channel"* IEEE Globecom'90. Conference proceedings. Vol.3, pp.1794-1799. Nov 1990.

Adams J.W. and Wilson A.N. *"A new approach to FIR digital filters with fewer multipliers and reduced sensitivity"* IEEE Transactions on Circuits and Systems. Vol.CAS-30, N°5, pp.277-283. 1983.

Adams J.W. and Wilson A.N. *"Some efficient digital prefilter structures"* IEEE Transactions on Circuits and Systems. Vol.CAS-31, N°3, pp.260-265. 1984.

AEL Crystal Filters *"Frequency Control Products"* Monolithic Crystal Filters, p.86.

Agarwal R.C. and Sudhakar R. *"Multiplier-less design of FIR Filters"* IEEE ICASSP, pp.209-212. 1983.

Alta SPW. *"The DSP Framework"* Release 3.1.

BDE Users Guide/Tutorial.	Document N° 8010.
Designers/BDE Users Guide.	Document N° 8012.
Signal Calculator Users Guide.	Document N° 8013.
Communications Library Reference.	Document N° 8015
Tools Interface Reference.	Document N° 8016
FDS Users Guide.	Document N° FDS8000

Ananasso F. and Pennoni G. *"Clock synchronouous multicarrier demodulator for multi-frequency TDMA communication satellites"* IEEE International Conference on Communications. Supercomm ICC'90 proceedings. Vol.3, pp.1059-1063. 1990. Catalogue N° 90CH2829-0.

Anderson T.O. *"Digital Data-Transition Tracking Loops"* Proceedings of the International Telemetering Conference, pp.259-271. Oct. 1968.

Ansari R. and Liu B. *"Efficient sampling rate alteration using recursive (IIR) digital filters"* IEEE Transactions on Acoustic, Speech and Signal Processing. Vol.ASSP-31, N°6, pp.1366-1373. 1983.

ANSI Standard. *"Recorded Magnetic Tape for Information Interchange"* (Manchester Code 1600 char/in) ANSI Standard X 3.39-1973.

Avenhaus E. *"The Design of digital filters with coefficients of limited word length"* IEEE Transactions on Audio Electro Acoustics. Vol.AU-20, N°3, pp.206-212. 1972.

B Banu M. and Dunlop A. *"Clock Recovery Circuits with Instantaneous Locking"* Electronic Letters. Vol.28, N°23, pp.2127-2130. Nov. 1992.

Banu M. and Dunlop A. *"A 660Mb/s CMOS Clock Recovery circuit with Instantaneous Locking for NRZ Data and Burst-Mode Transmission"* IEEE International Solid –State Circuits Conference. ISSC, pp.102-103. Feb. 1993.

Baydan I.Ye., Karpukhin B.V., Pavlichenko Yu.A. and Spozito O.O. *"Effect of clock synchronization accuracy on digital signal regeneration fidelity"* Telecommunications and Radio Engineering. Part 1. Vol.40, N°3, pp.19-24.

Bellis F.A. *"Introduction to Digital Audio Recording"* IERE Radio and Electronic Engineering. Vol.53, N°10, pp361-368. Oct 1983.

Bellisio J.A. *"A New Phased-Locked-Loop Timing Recovery Method for Digital Regenerators"*. IEEE International Communications Conference. Vol.1, pp.10-20. June 1976.

Bennett W.R. *"Statistics of Regenerative Digital Transmission"* BSTJ. Vol.37, pp.1501-1542. Nov 1958.

Bennett W.R. and Davey J.R. *"Data Transmission"* McGraw-Hill. Chapter 5. 1965.

Benoit H. *"Digital Television: MPEG-1, MPEG-2 and principles of the DVB System"* John Wiley & Sons. 1997. ISBN: 0-340-69190-5.

Berkman L.N. and Girshov V.S. *"An estimate of the permissible inaccuracy of clock phasing of multi-channel modems"* Telecommunications and Radio Engineering. Part 1.Vol.43, N°8, pp.44-46. August 1998.

Bertram H.N. *"The Theory of Magnetic Recording"* Cambridge University Press. 1994.

Best R.E. *"Phase-Locked Loops"* 2nd Edition. McGraw-Hill. 1993.

Boas R. and Kac M. *"Inequalities for the Fourier Transforms of Positive Functions"* Duke Mathematical Journal. Vol.12, pp.189-206. 1945.

Bold G. *"A comparison of the time involved in computing the fast Hartley and Fast Fourier Transforms"*
IEEE Proceedings. Vol.12, N°.73, p.1863. 1985.

Briggman D., Hanke G., Langmann U. and Pottbacker A. *"Clock recovery circuits up to 20 Gbit/s for optical transmission systems"*
IEEE MTT-S. International Microwave Symposium. Digest of Technical Papers, pp.1093-1096. 1994.

Brigham E.O. *"The Fast Fourier Transform"* Prentice-Hall, 1974.

Brush R. *"Design considerations for the D-2 PAL composite DVTR"*
7th IERE Video, Audio and Data Recording Conference.
IERE Publication N°79, pp.141-148. 1988.

Bures K.J. *"Understanding timing recovery and jitter in digital transmission systems"* R.F. Design. Vol.15, N°11, pp.42-44, 47, 49, 51-53. Oct 1992.
ISSN: 0163-321X.

Burrus C.S. *"Digital filter structures described by distributed arithmetic"* IEEE Transactions on Circuits and Systems. Vol.CAS-24. N°12, pp.674-680. 1977.

Byrne C.J. *"Systematic jitter in a chain of digital regenerators"*
Bell Systems. Technical Report 42, pp.2679-2714. 1963.

Burrus C.S. and Parks T.W. *"DFT/FFT and Convolution algorithms: Theory and Implementation"* Wiley –Interscience Publications. 1985.

C Cahn C.R., Leimer D.K., Marsh F.J. Huntowski F.J. and Larue G.D. *"Software Implementation of a PN Spread Spectrum Receiver to Accommodate Dynamics"*
IEEE Transactions on Communication Technology.
Vol.COM-25, pp.832-840. August 1977.

CCITT. *"Data Communications over the Telephone Network"* The International Telegraph and Telephone Consultative Committee Yellow Book. Vol.8, Fascicle 8.1, Rec.V.22, pp.84-85. Nov.1980.

Cessna J.R. and Levy J.D.M *"Phase noise and transition times for binary quantized digital phase-locked loop in white Gaussian noise"*
IEEE Transactions on Communication Technology. Vol. COM-20, pp.94-104. April 1972.

Claasen T.A, Mecklenbrauker W.F. and Peek J.B. *"Effects of quantization and overflow in recursive digital filters"* IEEE Transactions on Acoustics, Speech and Signal Processing. Vol. ASSP-24, N°6, pp.517-528. 1965.

Cooley J.W. and Tukey J.W. *"An algorithm for the machine calculation of complex Fourier series"* Mathematics of Computation 19, pp.297-301. 1965.

Carl C. *"Relay Telemetry Modulation System Development"*
Jet Propulsion Labs. California. SPS 37-50. Vol.3, pp.326-331. April 1968.

Cordell R.R. *"A 50MHz Phase and Frequency Locked Loop"*
IEEE Journal of Solid-State Circuits. Vol. SC-14, pp.1003-1009. Dec 1979.

Costas J.P. *"Synchronous Communications"*
Conference Proceedings IRE. Vol.44, pp.1713-1718. Dec 1956.

Cristian F., Aghili F. and Strong R. *"Clock synchronization in the presence of omission and performance faults, and processor joints"* 16th International Symposium on Fault-Tolerant Computing Systems. pp.218-223. July 1986.
ISBN: 0 8186 0703 3.

D Dae Sun Oh, Won Gi Jeon, Yong Soo Cho, Hyung Woon Park and Ki Ho Kim. *"Convergence analysis of a PLL for a digital recording channel with an adaptive partial response equalizer"* IEEE Globecom'96, pp.979-983. Nov 1996.

Davey P.J. *"Channel Coding Techniques for Multiple Track Digital magnetic Recording"* PhD. Thesis. University of Plymouth. Oct 94.

Davey P.J., Donnelly T. and Mapps D.J. *"Pulse slimming in Magnetic Recording using DSP Techniques"* Microprocessing and Microprogramming 37.
Conference Proceedings Euromicro 92, pp.73-76. 1992.

Davey P.J., Donnelly T. and Mapps D.J. *"Two dimensional Coding for Multiple-Track Maximum-Liklihood Digital Magnetic Storage System"*
IEEE Transaction on Magnetics. Vol.3, pp.4212-4214. March 1994.

Deeley E.M. and Mitchell A. *"Optimization of nonlinear clock recovery circuits for digital recording"* IERE 5th International Conference on Video and Data Recording, pp.137-140. 1984. ISBN: 0 903748 55X.

Del Re E. and Fantacci R. *"Joint carrier and clock recovery for QPSK and MSK digital communications"* IEE Proceedings Communications, Speech and Vision. Vol. 36, N°. 3, pp.208-212. June 1989.

De Vito L. *"A 52MHz and 155MHz clock recovery PLL"* ISSCC Dig. Tech. Papers, pp.142-143. Feb 1991.

Didday R.L. and Lindsey W.C. *"Subcarrier Tracking methods and Communication System Design"* IEEE Transactions on Communication Technology. Vol. COM-16, pp.541-550. August 1968.

Doelz M.L., Heald E.T. and Martin D.L. *"Binary Data Transmission Techniques for Linear Systems"* Proceedings of the IRE. Vol.45, pp.656-661. May 1957.

Doi T.T., Tsuchiya Y., Tanaka M. and Watanabe N. *"A format of stationary-head digital audio recorder covering a wide range of applications"* Journal Audio Engineering Society. Vol.28, p.931. Dec1980. Reprint 1677(H6).

Donnelly T. *"Real-time Microprocessor Techniques for digital Multi-track Tape Recorder"* PhD. Thesis. Plymouth Polytechnic. 1989.

Donnelly T., Mapps D.J. and Wilson R. *"An Intelligent Microprocessor Interface for a Low-Cost Digital magnetic Tape Recorder"* Microprocessing and Microprogramming. 22, pp.333-338. 1988.

Duhamel P. and Vetterli M. *"Fast Fourier Transforms: A tutorial review and a state of the art"* Signal Processing. Elsevier Publishers B.V. Chapter 19. 1990.

Duttweiler D.L. *"The Jitter Performance of Phase-Locked Loops Extracting Timing from Baseband Data Waveforms"* BSTJ. Vol.55, pp.37-58. Jan 1976.

E EBU Document. *"Specification of the Radio Data System (RDS) for VHF/FM Sound Broadcasters"* Technical Publication N° 3244. 1983

EBU Document *"Standard for recording digital television signals on magnetic tape in cassettes"* Technical Publication N°3252-E.

Eitan B., Caywood J., Lai S. and Sheperd W. *"Non-Volatile Memory Technoligies"* IEEE PAL Video. ISBN: 0-7803-1538-3.

Electronic Design. *"Applied Transform Theory"* pp.108-128. May 1995.

Enam S.K and Abidi A.A. *"NMOS ICs for clock and data regeneration in gigabit-per second optical-fibre receivers"* IEEE Solid-State Circuits. Vol.SC-27, pp.1763-1774. Dec 1992.

Engh J.T. *"The IBM Diskette and Diskette Drive"* IBM Journal. Res. Devel, Vol.25, N°5, pp.710-710. Sept 1981.

F Feedback Telecommunications. *"CloudFile 2000 Weather Satellite Receiving System 80-001"* 1994. Document N° 1160-80001.

Fleischmann B., Ruile W. and Riha G. *"Raleigh-mode SAW filters on quartz for timing recovery at frequencies above 1 GHz"* IEEE Ultrasonic Symposium. Conference Digest, pp.163-167. July 1984.

Franaszek P.A. *"Run Length Limited Variable Length coding with Error Propagation Limitation"* US Patent N° 3,689,899. Sept 1972.

Franks L.E. and Bubrouski J.P. *"Statistical Properties of Timing Jitter in a PAM Timing Recovery Scheme"* IEEE Transactions on Communication Technology. Vol. COM-22, pp.913-920. July 1974.

G Garg V.K. and Wilkes J.E. *"Wireless and Personal Communication Systems"* Prentice Hall. 1996. ISBN: 0-13-234626-5

Gallopoulos A., Heegard C. and Siegel P. *"The power spectrum of run-length-limited codes"* IEEE Transactions on Communications. COM-37 Vol.9, pp.906-917. Sept 1989.

Gardener F.M. *"Clock and Carrier Synchronization: Prefilter and Antihang-up Investigation"* ESA CR-984. European Space Agency. Estec Holland. Nov 1997.

Gardener F.M. *"Phaselock Techniques"* 2nd Edition. John Wiley & Sons. 1979. ISBN: 0-471-04294-3

Gardener F.M. *"Properties of Frequency Difference Detectors"* IEEE Transactions Communications. Vol. COM-33, N°2, pp.131-138. Feb 1985.

Golomb S.W. *"Digital Communications with Space Applications"* Prentice Hall. Englewood Cliffs. 1964.

Grant I.S. and Phillips W.R. *"Electormagnetism"* John Wiley & Sons. 2nd Edition. 1990.

Grebene A.B. *"The monolithic phase locked loop a versatile building block"* IEEE Spectrum. Vol.8, pp.38-49. March 1971.

Greer W.T. *"Digital Phased-Locked Loops move into Analogue Territory"* Electronic Design. Vol.30, N°8, pp.95-100. March 1982.

Gupta S.C. *"Phase Locked Loops"*
Proceedings of the IEEE. Vol.63, pp.291-306, Feb 1975.

H Hahn H. and Stout R. *"The Internet Complete Reference"*
Osborne McGraw Hill. 1996. ISBN: 0-07-881980-6.

Haykin S. *"Adaptive Filter Theory"* Prentice-Hall. 1986.

Hartley R.V.L. *"A more symmetrical Fourier analysis to transmission problems"*
Proceedings of the IRE. Vol.30, pp.144-150. March 1942.

Hein P. and Scott J.W. *"Z-Domain Model for Discrete-Time PLLs"*
IEEE Transactions Circuits and Systems. Vol.35, pp.1393-1400, Nov 1988.

Hickling R.M. *"A single chip 2Gbit/s clock recovery subsystem for digital communications"* RF Technology. Expo88.
IEEE Proceedings, pp.493-497. Feb 1988.

Hogge C.R. *"A self-correcting clock recovery circuit"*
IEEE Lightwave Technology. Vol.LT-3, pp.1312-1314. Dec 1985.

Howell T. *"Channel Capacity for the IBM 3380"*
International Symposium on Information Theory. Ann Arbor. 1986

Hu T.H and Gray P.R. *"A Monolithic 480Mb/s Parallel AGC/Decision Clock Recovery Circuit in 1.2-μm CMOS"* IEEE Journal of Solid-State Circuits.
Vol.SC-28, pp.1314-1320. Dec 1993.

Huber W.D. *"Simultaneous and orthogonally interactive clock recovery and DC null equalization in high density magnetic recording"* IEEE Transactions on Magnetics. MAG-19, Vol.5, pp.1716-1718. April 1983. ISSN: 0018-9464.

I IEE Colloquium *"New synchronisation techniques for radio systems"*
Professional Group E8 (Radio Communication Systems). Nov 1995.
UK ISSN: 0963-3308. Ref. N°1995/220.

Ifeachor E.C. and Jervis B.W. *"Digital Signal Processing, A Practical Approach"* Addison-Wesley. 1993. ISBN: 0-201-54413-X.

Ifeachor E.C and Smithson P.M. *"A Practical Approach to Digital Signal Processing"* Feb 1995. London. UK.
IEE Proceedings. The Teaching of DSP in Universities.
Ref. N° 1995/035.

Iltis R.A. *"A Baesian maximum-likelihood sequence estimation algorithm for a priori unknown channels and symbol timing"* IEEE Journals on Selected Areas in Communications 10. Vol.3, pp.579-588. April 1992.

Imbeaux J.C. *"Performances of the delay-line multiplier circuit for clock and carrier synchronisation in digital satellite communications"* IEEE Journal in Selected Areas in Communications. Vol.SAC-1, N°1, pp.82-85. June 1983.

Integrated Device Technology *"Product and Company Information on CD"* CD Ref. No. CD.COFP.00027. 1997.
Web Page: http://www.idt.com/products/fifo/Welcome.html

Itoh Y. and Nishimura K. *"Modulation-Demodulation Methods for digital Magnetic Recording"* Nikkei Electron. Vol.12, N°11, pp.1-164. 1978.

Ishihara N. and Akazawa Y. *"A monolithic 156Mb/s Clock and Data Recovery PLL Circuit using Sample and Hold Techniques"* IEEE Journal of Solid-State Circuits. Vol.SC-29, pp.1566-1571. Dec 1994.

J Jackson T.J. *"Multi Track Performance on a Digital Magnetic Tape Systems: Experimental Study and Simulation using Parallel Processing Techniques"* PhD. Thesis. University of Plymouth. 1991.

Jackson T.J., Mapps D.J., Ifeachor E.C. and Donnelly T.
"A Real-Time Transputer-Based System for a Digital Recording Data Channel" Microprocessors and Microprogramming 25, pp.281-285. 1988.

Johnson M.G. and Hudson E.L. *"A variable delay line PLL for CPU-coprocessor synchronization"* IEEE Solid-State Circuits. Vol.23, pp.1218-1223. Oct 1988.

Jorgensen F. *"The complete handbook of magnetic recording"* Tab books. 4th Edition. 1996. ISBN: 0-07-033045-X

Justesen J. *"Information rates and power spectra of digital codes"* IEEE Transactions on Information Theory. Vol.28, N°3, p.457. March 1982.

K Kasai H. *"PCM jitter suppression by scrambling"* IEEE Transactions on Communication Technology. Vol.COM-22, N°8, pp.1114-1122. 1974.

Kautz W.H. *"Fibonacci Codes for Synchronisation Control"* IEEE Transactions on Information Theory. Vol.11, pp.284-92. April 1965.

Kent A., Williams J.G. *"Encyclopaedia of Computer Science and Technology"*. Marcel Dekker Inc. Vol.24, Supplement 9, Donnelly T., pp.239-285.

Kim B., Helman D.N., and Gray P.R. *"A 30-MHz Hybrid Analogue/Digital Clock Recovery Circuit in 2-μm CMOS"* IEEE Journal of Solid-State Circuits. Vol.SC-25, pp.1385-1394. Dec 1990.

Kim G.Y. *"Broadband LAN technology"* Artech House. 1998. ISBN: 0-89006-291-9.

Kobayashi H. *"A Survey of Coding Schemes for Transmission or Recording of Digital Data"* IEEE Transactions on Communications. Vol.COM-19, N°6, pp.1087-1100. Dec 1991.

Kodek D.M. *"Design of optimal finite wordlength FIR digital filters using integer programming techniques"* IEEE Transactions on Acoustic, Speech and Signal Processing. ASSP-28. Vol.3, pp.304-307. 1980.

L Lai B. and Walker R.C. *"A monolithic 622Mb/sec clock extraction and data retiming circuit"* ISSCC Dig. Tech. Papers, pp.144-145. Feb 1991.

Layland J.W. *"Optimum signals for Phased-Locked Loops"* IEEE Transactions on Communication Technology. Vol.COM-17, N°5, pp.526-531. Oct 1969.

Layland J.W. *"Telemetry Bit Synchronization Loop"* Jet Propulsion Labs. California. SPS 34-46. Vol.3, pp.204-215. July 1967.

Lee T.H. and Bulzacchelli J.F *"A 155-MHz clock recovery delay and phased-locked loop"* IEEE Solid-State Circuits. Vol.SC-27, pp.1736-1746. Dec 1992.

Lindsey W.C. and Chie C.M *"A Survey of Digital Phase-Locked Loops"* IEEE Proceedings. Vol.69, N°30, pp.410-431. April 1981.

Lindsey W.C. *"Synchronization Systems in Communication and Control"* Prentice-Hall. 1972.

Lindsey W.C. and Simon M.K. *"Telecommunication Systems Engineering"* Prentice-Hall. 1973. ISBN: 0-486-66838-X.

Lindsey W.C. and Tausworth R.C. *"Digital Data-Transition Tracking Loops"* Jet Propulsion Labs, California, SPS 37-50, Vol.3, pp.272-276. April 1968.

Lindholm D.A. *"Power Spectra of Channel Codes for Digital Magnetic Recording"* IEEE Transactions on Magnetics. Vol.MAG-14, N°5. Sept 1978.

Lucky R.W., Salz J. and Weldon E.J. *"Principles of Data Communication"* McGraw-Hill. 1968.

M Mackintosh N,D. *"The Choice of Recording Codes"* IERE Conference on Video and Data Recording, pp.77-119. July 1979.

Mallison J.C. *"The Foundations of Magnetic Recording"* Academic Press 1987.

Mallison J.C. and Miller J.W. *"Optimal Codes for Digital Magnetic Recordings"* The Institute of Electronic and Radio Engineers. Vol.47, N°4, pp.172-176. April 1977.

Mapps D.J. and McQuade E. *"Computer simulation of audio replay heads"* IERE Conference on Video and Data Recording. July 1979.

Mapps D.J., Donnelly T., Davey P.J. and Smith D.F. *"Signal Processing for present and future magnetic recording systems"* Journal of the Magnetic Society of Japan. Vol.21, Supp.52, PP.211-220. Oct. 1997.

Marven C. and Ewers G. *"A Simple approach to Digital Signal Processing"* Wiley Interscience Publication. 1996. ISBN: 0 471 15243 9.

Mathcad Inc. *"MathCad 6.0 Uses Guide"* Math Soft Europe. 1995. Publication N°VP6.00-231195-11899-AD.

Mattis W.E. and Uzunoglu V. *"A high speed modem with unique recovery and equalization"* 6th International Conference on Computers and Communications. Proceedings Cat. N° 87 0179-2, pp.150-155. 1987. ISBN: 0 8186 0765 3.

Mazo J.E. *"Jitter Comparison of Tones Generated by Squaring and forth-Power Circuits"* BSTN. Vol.57, pp.1489-1498. May 1978.

McEliece R.J. *"The Theory of Information and Coding"* Addison-Wesley. 1997.

McRae D.D. and Smith E.F. *"An Acquisition aid for the Digital Phase-Locked Loops"* Proceedings of the International Conference on Communications. Philadelphia, pp.14-22. June 1972.

Mee C.D. and Daniel E.D. *"Magnetic Recording"* Video, Audio and Instrumentation Recording. Vol.3. McGraw-Hill. 1988.

Mengali U. *"A Self Synchronizer Matched to the Signal Shape"* IEEE Transactions. Vol.AES-7 N°4, pp.686-693. July 1971.

Mengali U. *"Joint Phase and Timing Acquisition in Data-Transmission"* IEEE Transactions on Communication Technology. Vol.COM-25, pp.1174-1185. Oct 1997.

Messershmitt D.G. *"Frequency detectors for PLL acquisiton in timing and carrier recovery"* IEEE Transactions on Communications. Vol.COM-27, pp.1288-1295. Sept 1979.

Mitchell C.M., Evans N.E. and Black N.D. *"Use of software controlled timers for data synchronization in digital portable radio communication systems"* 3rd Bangor Symposium on Communications, pp.122-125. University of Wales. 1991

Miller J.W. *"Transmission System"* US Patent N° 3,108,261. Oct 1963.

Miller J.W. *"The Miller Squared Code"* US Patent N° 4,027,335. May 1997.

Millicker D.J. and Stanley R.D. *"2 Gbit/s timing recovery circuit using dielectric resonator filter"* Electronic Letters. Vol.23, N°14, pp.738-739. July 1987. ISSN: 0013-5194.

Miyazawa S. *"A BiCMOS PLL-Based Data Separator Circuit with High Stability and Accuracy"* IEEE Journal of Solid-State Circuits. Vol.SC-26, pp.116-121. Feb 1991.

Mosier R.F. and Clabaugh R.G. *"Kineplex, a Bandwidth-efficient Binary Transmission System"* Transactions of the AIEE (Communications and Electronics), Vol.76, pp.723-728. Jan 1958.

Myrica (UK) Ltd. *"Scotsman Model MCD750 Drive. Proposed Specification"* 1997.

Mueller K. and Muller M. *"Timing recovery in digital synchronous data receivers"* IEEE Transactions on Communications 24. Vol.5, pp.516-531. May 1976.

Mullard Technical Handbook, Book 4, Integrated Circuits, Part 5. 1986. *"High-speed CMOS logic HC/HCT Family"* Publication N°806055.

N Nakagawa S. *"A Study on Detection Methods of NRZ Recording"* IEEE Transactions on Magnetics. Vol.MAG-16, N°1, pp.104-110. Jan 1980.

Nakajima H. and Odaka K. *"A rotary-head high-density digital audio tape recorder"* IEEE Transactions on Consumer Electronics. CE-29, pp.430-437. 1983.

Natali F.D. *"Accurate digital detection of angle modulation signals"* EASCON Records, pp.407-412. Oct 1968.

NG, Leung, Chung, Luk and Lau. *"Adaptive IIR Filter – A Genetic Search Approach"* IEEE. Signal Processing, pp.38-46. Nov 1996.

O Oppenheim A.V. and Willsky A.S. *"Signals and Systems"* Prentice-Hall. 1983.

Orcad Systems Corporation *"Verification and Simulation Tools, Users Guide"* Document N°2880400B Ver.1.12. 1989.

P Palmer D., Ziperovich P., Wood R. and Howell T.D. *"Identification of nonlinear write effect using pseudo random sequences"* IEEE Transactions on Magnetics. Vol.MAG-23, N°5, pp.2377-2379. Sept 1987.

Parkinson J. *"Data Recording and its Problems"* Radio Communications. Vol.60, N°1, pp.32-35. 1984.

Pearson D.E. and Robinson J.A. *"Visual Communication at very low data rates"* IEEE Proceedings. Vol.73, N°4, pp.795-812.

Peek J.B.H. *"Communication aspects of the Compact Disk digital audio system"* IEEE Communications Mag. N°23, pp.7-15. 1985.

Pottbacker A. and Langmann U. *"An 8GHz silicon bipolar clock recovery and data regenerator IC"* IEEE Solid-State Circuits. Vol.29, pp.1572-1576. Dec 1994.

Pottbacker A., Langmann U. and Schreiber H.H. *"A Si bipolar phase and frequency detector IC for clock extraction up to 8Gb/s"* IEEE Solid-State Circuits. Vol.27, pp.1747-1751. Dec 1992.

Price R. and Green P.E. *"A Communication Technique for Multipath Channels"* Proceedings of the IRE. Vol.46, pp.555-570. March 1958.

Proakis J.G. and Manolakis D.G. *"Introduction to Digital Signal Processing"* Macmillan Press. 1988.

Q Qureshi S.U.H. *"Timing recovery for equalized partial-reponse systems"* IEEE Transactions on Communications. Vol.24, pp.1326-1331. Dec 1976.

R Rabiner L. and Gold B. *"Theory and applications of Digital Signal Processing"* Prentice-Hall. 1975. ISBN: 0-13-914101-4.

Ransijn H. and O'Connor P.O. *"A PLL-Based 2.5 Gb/s Clock and Data Regenerator IC"* IEEE Journal of Solid-State Circuits. Vol. SC-26, pp.1337-1344. Oct 1991.

Razavi B. *"Monolithic Phase-Locked Loops and Clock Recovery Circuits"* IEEE Press. 1996. ISBN: 0-7803-1149-3.

Razavi B. and Sung J. *"A 2,5-Gb/sec 15-mW BiCMOS clock recovery circuit"* IEEE Symposium on VLSI Circuits Dig. Tech. Papers, p.83. 1995.

Richman D. *"Color-Carrier Reference Phase Synchronisation Accuracy in NTSC Color Television"* IRE proceedings. Vol.42, pp.106-133. Jan 1954.

Roberts A. and Wade J.G. *"Optimization of Finite Wordlength Filters using a Genetic Algorithm"* IEE Proceedings. ACEDC'94, pp.37-43. 1994.

Roza E. *"Analysis of Phase-Locked Timing Extraction Circuits for Pulse Code Transmission"* IEEE Transactions on Communication Technology. Vol.COM-22, pp.1236-1249. Sept 1974.

S Sage A.P. and McBride A.L. *"Optimum Estimation of Bit Synchronization"*. IEEE Transactions on Aerospace and Electronic Systems. Vol. AES-5, N°3, pp.525-536. May 1969.

Salzberg B.R. *"Timing Recovery for Synchronous Data Transmission"* BSTJ. Vol.46, pp.593-622. March 1967.

Savage J.E. *"Some simple self-synchronizing digital data scramblers"* Bell Systems. Technical Report, pp.449-487. Feb 1967.

Seigel P.H. *"Recording Codes for Digital Magnetic Storage"*
IEEE Transactions on Magnetics. Vol.MAG-21, N°5, pp.1344-1349. Sept 1985.

Shaeffer C.F. *"The zero-beat method of frequency discrimination"*
Proceedings of the IRE. Vol.30, pp.365-367, Aug. 1942.

Shafiee H. *"Timing recovery for sampling detectors in digital magnetic recording"* IEEE International Conference on Communications, S18-25. 1996.

Shaft P. *"Optimum Design of the Nonlinearity in Signal Tracking Loops"*
Stanford Research Institute. California. May 1968.

Shannon C.E. *"A Mathematical Theory of Communications"*
Bell Systems. Technical Journal. Vol.27, N°3, pp.379-423, 623-656. July 1948.

Shannon C.E. *"Communication in the presence of noise"*
Proceedings of the IRE. Vol.37, pp.10-21. 1949.

Sharpe C.A. *"A 3-state phase detector can improve your next PLL design"*
EDN, pp.55-59. Sept 1976

Siemens Matsushita *"SAW Components Data Book"*
Document N°B420-P2026-X-X-7600, pp.325-331. 1996.

Sidnev A.N. *"Clock synchronization devices for high-speed receivers"*
Telecommunications and Radio Engineering, Part 2. Vol.47, pp.86-91. Feb 1992.
ISSN: 0040-2508.

Simon M.K. *"An Analysis of the Steady-State Phase Noise Performance of a Data-Transition Tracking Loop"*
Jet Propulsion Labs. California. SPS 37-55, pp.54-62. Feb 1969.

Simon M.K. *"Nonlinear Analysis of an Absolute Value Type of an Early-Late Gate Bit Synchronizer"* IEEE Transactions on Communication Technology.
Vol.COM-18, N°3, pp.589-597. Oct 1970.

Simon M.K. *"Optimization of the performance of a digital-Data Transition Tracking Loop"* IEEE Transactions on Communication Technology.
Vol.COM-18, N°3, pp.686-690. Oct 1970.

Sivov V.A., Pisarev V.A., Yu C. and Prytkov V.I. *"Synthesis of an optimal system of clock synchronization using a relay pilot-signal"* Telecommunications and Radio Engineering. Part 1, Vol.41, N°1, pp.28-33. Jan 1987.

Slader J.T. *"TMS320C50 Satellite Data Receiver – Progress Report"* University of Plymouth. Satellite Communications Research Centre. 1997.

Slader J.T., Smithson P.M. and Tomlinson M. *"A Data Reply Link for Satellite TV Applications"* IEEE Globecom'97 Proceedings. Vol.2, pp.1142-1146. 1997. ISBN: 0-7803-4198-8/97.

Slader J.T., Smithson P.M., Tomlinson M., Ambroze A.M., Wade J.G. *"A Novel Internet Delivery System using Asymmetric Satellite Channels"* The 49th International Astronautical Congress. Sept 1998. Document N° IAF-98-M.5.06.

Smith D.F. *"Enhanced Coding, Clock Recovery and Detection for a Magnetic Credit Card"* PhD. Thesis. University of Plymouth. May 1998.

Smith D.F., Donnelly T. and Mapps D.J. *"Fixed sample rate Bayesian detector in a variable speed magnetic channel"* IEEE International Conference on Magnetics. Intermag97 proceedings. Vol.33, pp.2797-2799. April 1997.

Smith E.F. *"Bit Synchronization"* IEEE Proceedings of the International Telemetering Conference. L.A. California, pp.539-552. Oct 1972.

Smithson P.M., Tomlinson M. and Donnelly T. *"DSP-Based Clock Recovery for a Digital Magnetic Recording Channel"* IEEE Globecom'94 Proceedings Vol.3, pp.1467-1471. 1994. ISBN: 0-7803-1820-X.

Smithson P.M., Tomlinson M. and Donnelly T. *"DSP-Based Clock Recovery Implemented in a Field Programmable Gate Array"* IEE Proceedings. New Synchronisation Techniques for Radio Systems. 1995. ISSN: 0963-3308, Ref. N° 1995/220.

Smithson P.M. and Tomlinson M. *"The Development of an Operational Computer Based Satellite Data Broadcasting System"* IEE 10th International Conference on Digital Satellite Communications. ICDSC-10, Vol.2, pp.405-412. May 1995. ISBN: 0-85296-6393

Smithson P.M. Slader J.T., Smith D.F. and Tomlinson M. *"The Development of an Operational Satellite Internet Service Provision"* IEEE Globecom'97 Proceedings. Vol.2, pp.1147-1151. Nov 1997. ISBN: 0-7803-4198-8/97.

SMPTE 226M *"Proposed American National Standard for component digital television recording 19mm type D-1 cassette format"*

Sonntag J. and Leonwich R. *"A monolithic CMOS 10MHz DPLL for burst mode retiming"* IEEE ISSCC Dig. Tech Papers, pp.194-195. Feb 1990.

Soyuer M. *"A Monolithic 2.3Gb/s 100mW Clock and Data Recovery Circuit in silicon Bipolar Technology"* IEEE Journal of Solid-State Circuits. Vol. SC-28, pp.1310-1313. Dec 1993.

Springett J.C. *"Telemetry and Command Techniques for Planetary Spacecraft"* Jet Propulsion Labs. California. Technical Report 32-495. Jan 1965.

Stiffler J.J. *"The Selection of Signals for Phase-Locked Loops"* IEEE Transactions on Communication Technology. Vol.COM-16, N°2, pp.239-244. April 1968.

Stiffler J.J. *"The Squaring loop Technique for binary PSK Synchronization"* Jet Propulsion Labs. SPS 37-26. Vol.4, pp.240-246. April 1964.

Stiffler J.J. *"Theory of Synchronous Communications"* Prentice–Hall. 1971. Chap 7. ISBN: 0-13-914739-X

Stiffler J.J. *"Maximum Likelihood Symbol Synchronization"* Jet Propulsion Labs. SPS 37-35, Vol.4, pp.349-357. Oct 1965.

Stremler F.G. *"Introduction to Communication Systems"* 3rd Edition. Addison Wesley. 1990. ISBN: 0-201-51651-9.

Stroud K.A. *"Engineering Mathematics"* Macmillan Press. 1992. ISBN: 0-333-44887-1.

Suehiro N. and Hatori M. *"Fast algorithms for the DFT and other sinusoidal transforms"* IEEE Transactions on Acoustic, Speech and Signal Processing. ASSP-34, pp.642-644. 1986.

Sun S.Y. *"An Analogue PLL-Based Clock and Data Recovery Circuit with High Input Jitter Tolerance"* IEEE Journal of Solid-State Circuits. pp.325-330. April 1989.

T Takasaki Y. *"Timing Extraction in Baseband Pulse Transmission"* IEEE Transactions on Communication Technology. Vol.COM-20, pp.877-884. Oct 1972.

Ta-Mu Chien. *"Upper bound on the efficiency of DC-constrained codes"* Bell Systems. Technical Journal 49, pp.2267-2287. Nov 1970.

Tausworthe R.C. *"Analysis and Design of the Symbol-Tracking Loop"* Jet Propulsion Labs. SPS 37-51, Vol.2, pp.145-147. May 1968.

Texas Instruments Publications *"The TTL Data Book for Design Engineers"* Vol.2, 74LS297 DPLL, pp.783-788. 1989.
ISBN: 3-88078-078-1.

Texas Instruments Publications *"TMS320C25 Users Guide"* 1986.
ISBN: 2-86886-014-1.

Texas Instruments Publications. *"Digital Signal Processing Applications"* 1986.
ISBN: 2-86886-009-5.

Texas Instruments Publications. *"TMS320C5x Users Guide"* 1993.
Document N° 2547301-9761 Rev D. SPRU056B

Texas Instruments Publications. *"TMS320C5x C Source Debugger Users Guide"* 1994. Document N° 2547297-9761 Rev F. SPRU055B

Texas Instruments *"TMS320C8x Features"*
Web Page: http://www.ti.com/sc/docs/dsp/products/c8x/features.htm

Texas Instruments Publications. *"TMS9914A General Purpose Interface Bus (GPIB) Controller"* Document N° MPL7. Feb. 1984

Tomlinson M. *"The Offset Fast Fourier Transform"*
MOD SRDE Document N° 76025.

Tomlinson M., Winders R. and Hendry M. *"SMADP (Satellite Multimedia Applications Demonstration Programme) 2-Way Internet/Intranet Services by Satellite Project Report"* University of Plymouth.
Satellite Research Centre. 1998.

U Uwano S. and Watanabe K. *"Symbol Synchronization Performance using Loop Switching Type Quenching Method"*
Globecom'97 Proceedings. Vol.3, pp.1199-1203. ISBN: 0-7803-4198-8/97.

V Van Horn J.H. *"A Theoretical Synchronization System for use with Noisy Digital Signals"* IEEE Transactions on Communication Technology.
Vol.COM-12, N°3, pp.82-90. Sept 1964.

Van Trees H.L. *"Detection, Estimation and Modulation Theory"* Part 1.
Wiley & Sons Publishers. 1968.

Viterbi A.J. *"Principles of Coherent Communication"* McGraw-Hill. 1996.

Viterbi A.J. and Omurua J.K. *"Principles of Digital Communication and Coding"* McGraw-Hill. 1971.

VTC Inc. *"PRML Read Channel with PR4, 8/9 ENDEC, 4-Burst Servo. SSI32P4903 Preliminary Data Sheet"* Document N° VM65060, Rev 03. 1997.

W Wade J.G. *"Signal Coding and Processing"* 2nd Edition.
Cambridge University Press. 1994. ISBN: 0-521-42336-8.

Wade J.G. *"Offset FFT and its implementation on the TMS3020C25 processor"* Microprocessors and Microsystems 12, Vol.2, pp.76-82. 1988.

Wade J.G. and Roberts A. *"Ordering of FIR filter structures"*
Electronic Letters. Vol. 30, N°17, pp.1293-1394. Aug 1994.

Wade J.G., Roberts A. and Williams G. *"Multiplier-Less FIR Filter Design using a Generic Algorithm"* Visual Image and Signal Processing.
IEE Proceedings. Vol.141, N°3, pp.175-180. 1994.

Wade J.G., Van-Eetvelt P. and Darwen H. "Synthesis of efficient low-order FIR filters from primitive sections"
IEE Proceedings. N°137, Part G, Vol.5, pp.367-72. 1990.

Wallace P., Bayruns R., Smith J., Laverick T. and Shuster R. *"A GaAs 1.5Gb/s clock recovery and data retiming circuit"* IEEE International Solid-State Circuits Conference. Cat. No.90CH2824-1, p.192. Feb 1990.

Widrow B. and Stearns S.D. *"Adaptive signal Processing"* Prentice-Hall. 1985.

Watkinson J. *"The Art of Digital Audio"*
Focal Press. 1988. ISBN: 0-240-51270-7.

Watkinson J. *"Coding for Digital Recording"*
Focal Press. 1990. ISBN: 0-240-51293-6.

Watkinson J. *"The Technology of the Ampex DCRSi"* IERE Conference on Video, audio and Data Recording. University of York. March 1998.

Watkinson J.R. *"CD: The 600 megabyte ROM"*
Electronics and Wireless World. N°93, pp.1046-1049. 1987.

Watkinson J. *"Splice-handling mechanism in DASH format"*
77th Audio Engineering Society Convention. 1985. Reprint N°1677 (H6).

Way W.I. *"High speed circuit technology for multi-gigabit/sec communication systems"* IEEE International Conference on Communications. Conference Proceedings, pp.313-318. June 1988.

Wintz P.A. and Hancock J.C. *"An Adaptive Receiver Approach to the Time Synchronization Problem"* IEEE Transactions on Communication Technology. Vol.COM-13, pp.90-96. March 1965.

Wintz P.A. and Luecke E.J. *"Performance of Optimum and Suboptimum Synchronizers"* IEEE Transactions on Communication Technology. Vol.COM 17, N°3, pp.380-389. June 1969.

Wood R.W. *"Magnetic Recording Systems"*
Proceedings of the IEEE. Vol.74, N°11, pp.1557-1569. 1986.

Wood R. Lindholm D.A. and Hagg R.M. *"The bandwidth of magnetic record/reproduce heads"* IEEE Transactions on Magnetics. Vol.MAG-21, N°5, pp.1566-1568. Sept 1985.

Wright G. and Stevens W.R. *"TCP/IP Illustrated. The implementation"* Vol.2. Addison-Wesley. 1995. ISBN: 0-301-63354-X.

X Xilinx Publications. *"The Programmable Logic Data Book 9/96"*
Document N° PN0010303.

Xilinx Publications.
"Logic Cell Array Development System. Reference Manual" Vol 1.-0430398 01.
"Logic Cell Array Development System. Reference Manual" Vol 2.-0430399 01.
"Logic Cell Array Users Guide" 0430402 01.
"Logic Cell Array Macro Library" 0430402 01.

Y Yokoyama K. *"Digital Video Tape Recorder"* NHK Technical Monograph. N°31. March 1982.

Z Zehavi E. and Wolf J.K. *"On run length codes"*
IEEE Transactions on Information Theory 34, Vol.1, pp.45-54. Jan 1988.