

# SYSTEMS AND COMPUTER ENGINEERING

**Transmission Techniques for Broadband  
Wireless Up-Link & Digital Interactive  
Broadcasting Services.**

**Phase III: DTV Channel Estimation from  
Laboratory and Off-Air DTV Signal Recordings**



**Carleton**  
UNIVERSITY

IC

LKC  
TK  
5103.2  
.E42  
2003

Department of Systems and Computer Engineering  
Carleton University, 1125 Colonel By Drive  
Ottawa, Ontario, Canada K1S 5B6

**Transmission Techniques for Broadband  
Wireless Up-Link & Digital Interactive  
Broadcasting Services.**

**Phase III: DTV Channel Estimation from  
Laboratory and Off-Air DTV Signal Recordings**

Prepared By

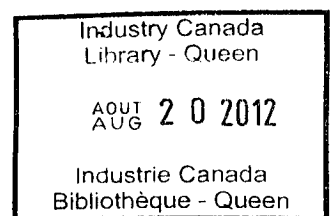
Mohamed S. El-Tanany, Ph.D., P.Eng  
Department of Systems and Computer Engineering  
Carleton University, 1125 Colonel By Drive  
Ottawa, Ontario, K1S 5B6

Prepared for

Communications Research Centre  
3701 Carling Avenue, Shirley Bay, Ottawa, Ontario  
Contract Number: 5005120  
Scientific Authority: Dr. Yiyan Wu

March, 2003

**CRC LIBRARY**  
**-06- 11 2003**  
**BIBLIOTHEQUE CRC**



# Table of Contents

<b>Executive Summary</b>	<b>1</b>
<b>I. Overview of the ATSC Digital Television Standard and Background</b>	<b>6</b>
DTV System Overview	6
RF Transmission Subsystem	8
Subsystem Overview	8
Frames, Fields And Segments	9
Channel Error Protection And Synchronization	10
DATA SEGMENT SYNC	15
The Data Field Sync	15
System Parameters	18
Modulation	20
Raised Cosine Filtering	21
Binary Shift Register Sequences	22
<b>II. Data Analysis and Channel Estimation Algorithms</b>	<b>24</b>
Introduction	24
Laboratory Experiments	24
Field Experiments	26
Structure of the data analysis programs	27
IF Processing	27
Baseband Processing	29
The Need for Time-Domain Windows	31
Program Description	34
Example Program Runs	35
<b>III. Channel Estimation Programs</b>	<b>45</b>

# Table of Contents

Main Program	46
Field Sync Calculator	53
Correlation with a down-converted reference/IIR Filtering	56
Correlation with a down-converted reference/SRRT Filtering	59
Correlation with a Baseband Reference	62
Binary to Decimal Conversion	66
IF_Filtering	67
Raised Cosine Window	68
Rectangular Window	69

## List of Figures

System Block Diagram	6
A Representative 8VSB Terrestrial Broadcast Transmitter.	8
Frames Fields and Segments; data organization for transmission	9
Data Randomizer	11
Reed Solomon coded word	11
Convolutional Interleaver. $M=4$ , $B=52$ , $N=208$ , RS-block= 207, $BM=N$	12
De-interleaver. $M=4$ , $B=52$ , $N=208$ , RS-block=207, $BM=N$	13
8 VSB Trellis Encoder, Precoder And Symbol Mapper	14
Trellis Code Interleaver	14
Data Segment Sync	15
VSB Channel Occupancy (Nominal)	20
Linear Feedback shift register circuit that can be used as a PNS generator	22
Example Laboratory Set-up for Channel Recording	25
Simplified Diagram of a Generic Set-Up That Could Be Used for the Recording of a Terrestrial DTV Signal.	26
Processing Steps Followed in Scanning a Recorded Data File.	28
Signal Processing Chain Filtering Options Based on an IF Reference Signal	30
Band-limiting, and non-ideal channel cause random data and the PN63 data to interfere with the desired PN511 sequence at the receiving end.	32
Signal Processing Chain and Options Based on a Baseband Reference Signal	33
Power spectral density of a data block taken from a laboratory recorded signal after passing through a multipath channel	36
Impulse Response based on a data block taken from a laboratory recorded signal after passing through a multipath channel	37
Auto-correlation Function of the Baseband Reference	38
Power spectral density for an off-air recording after IF filtering.	39
Estimated channel response from an off-air recording	41

Frequency response of an IF/8VSB filter, center frequency = 9 MHz	41
Impulse response from an off-air signal (off_air4) obtained using an IF reference, with SRRC filters after down conversion.	42
Impulse response from an off-air signal (off_air4) obtained using an IF reference, with Butterworth IIR filters after down conversion.	43
Impulse response from an off-air signal (off_air4) obtained using an IF reference, with SRRC filters after down conversion.	44
Four consecutive Impulse responses from an off-air signal (off_air4) .	45
Averaged impulse response based on the first four consecutive Impulse responses from an off-air signal (off_air4) .	46

# **Study of Transmission Techniques for Broadband Wireless Up-Link and Digital Interactive Broadcasting Service**

## **Phase III: Channel modeling based on measurements**

### **Executive Summary**

Orthogonal Frequency Division Multiplexing (OFDM) have been selected as modulation schemes for Digital Television Terrestrial Broadcasting (DTTB), Digital Audio Broadcasting (DAB), Asymmetric Digital Subscriber loops (ADSL) and Broadband Wireless Communications. The main advantages of OFDM are multi-path resistance under commonly encountered fading conditions and high spectrum efficiency. OFDM has two major drawbacks: a large peak-to-average envelope ratio, and heightened sensitivity to synchronization errors. The latter drawback may prove to be the decisive factor in whether OFDM can be used for interactive data transmission over broadband wireless LAN/MAN implementations, especially on the return link.

Other alternative modulations, which can cope with multi-path, include spread spectrum, as well as block transmission based on single carrier MQAM, with a frequency domain equalizer. The latter has been shown to perform nearly equivalent to OFDM in terms of complexity and multi-path resistance.

This particular study is to investigate the performance of transmission techniques that appear suitable for use on the return link of a broadband wireless LAN/MAN environment and for the terrestrial wireless interactive multimedia services.

In the Phase I and II study, sophisticated mathematical models have been established to analysis the impact of RF frequency offset, sampling speed error, and time delay error, as well as diversity gain that could be achieved for possible mobile implementation. Computer simulations were conducted to verify the accuracy of the mathematical models. It was proven that spectrum

overlapped single carrier QAM-type modulations can be used as the up-link for its hardware simplicity and low peak-to-average-power ratio. At the headend, OFDM-type of demodulation can be utilized to group demodulate the up-link carriers. The used of OFDM-type demodulation achieves high spectrum efficiency and reduces headend hardware complicity.

Previous simulations have been conducted using channel models that were developed for wireless cellular systems, mainly due o the lack of sufficient knowledge of the DTV channel. The Terrestrial DTV channels differ from those used for cellular systems due to:

- 1) Differences in network topologies. Current wireless systems use a cellular structure to allow for a high frequency re-use ratio. Terrestrial TV on the other hand is likely to use single frequency networks or on-channel repeaters to ensure more uniform coverage at relatively low transmitted power levels.
- 2) Different Frequency bands: Wireless cellular systems are more likely to use the 1900 MHz frequency band. Existing wireless LANs use mostly unlicensed bands @ 2.4 GHz or 5 GHz. Digital Terrestrial Tv on the other hand uses VHF and/or UHF frequencies. The propagation characteristics at these bands are much different from the propagation characteristics at higher bands. The differences are favorable for the TV bands due to lower large-scale loss, higher resistance to shadowing and stronger building penetration
- 3) Transmitted power levels: Cellular systems and wireless LANs use relatively low transmission power levels (in the order of 1 Watt); this limit is imposed by spectrum regulations for the bands in use. As such, the coverage area of a single transmitter is in the order of 1 km. TV transmitters on the other hand are much more powerful in terms of transmitted power levels. This coupled with low path loss ensures that the coverage area of a single transmitter is much wider than the coverage area of existing wireless systems.
- 4) transmitter antennas: Wireless systems are more likely to use sectored antennas for spectrum reuse and/or interference reduction purposes. TV transmitters antennas are more likely to be omni-directional.



5) Types of end user antennas: for wireless LANs or cellular systems the end-user antenna is a simple monopole omni-directional antenna. End-user TV antennas on the other hand can be directional.

The factors cited above lead to the reasonable expectation that the channel propagation characteristics for DTV uplink applications are sufficiently different from those encountered in existing cellular conditions. To this end, there exists a need for the development of new channel models that are directly applicable to the DTV conditions. The models are to be developed on the basis of experiments followed off-line signal processing. The signal processing functions are intended for the estimation of the channel impulse response based on field and/or laboratory measurements.

The work reported here is concerned mainly with the development of the algorithms needed for the off-line processing of data provided by CRC based on laboratory and field conducted experiments. The algorithms developed are also implemented and tested using Matlab. The impulse response models developed can be used for future uplink simulation and designs.

The measurements provide digital recordings of 8VSB digital TV signals which are generated according to the ATSC 8VSB standard. According to the standard, the transmitted signal contains a field-sync sequence that repeats every 24.2 milli-seconds. The signal processing functions considered here rely on this field-sync sequence for channel estimation. The DSP processing functions developed include IF filtering, field-sync signal synthesis, signal/ field-sync signal interpolation, IF carrier removal, matched filtering, time domain windowing, deconvolution and graphical representation of estimated results. The algorithms developed are provided in a Matlab channel estimation set of functions as described in table I.

**Table 1: Main building blocks of the channel estimation procedure**

<b>function name</b>	<b>Main processing tasks</b>
atsc_data_interact	This is the main interactive program where the user specifies experimental parameters and indicates various choices for signal processing options.
binary_to_float	Converts recorded data from the binary format used by the recording equipment into decimal integers

**Table 1: Main building blocks of the channel estimation procedure**

function name	Main processing tasks
IF_Filtering	Applies an IF filter to the measurement data at IF. A square-root raised cosine response is implemented according to the ATSC standard. The IF filter is used only if the option is selected by the user.
BB_SRRC_SRRC	Down converts the received signal to baseband. The down-converted signal is applied to a matched filter. The matched filter response is obtained by down conversion of an IF reference signal. Square-root raised cosine filters are used as part of the signal and reference down-converters.
rcwin	This function returns a time-domain raised cosine window with a roll-off ratio $\beta$ , in a column vector that contains N points.
recwin	This function returns a rectangular window that is " $N-2\beta$ " points wide, in a column vector that contains N points.
BB_IIR_SRRC	Down converts the received signal to baseband. The down-converted signal is applied to a matched filter. The matched filter response is obtained by down conversion of an IF reference signal. A Butterworth 4th order IIR filter is used as part of the signal down-converter. A square-root raised cosine filter is used for reference down-conversion.
BB_SRRC_REC	Down converts the received signal to baseband. The down-converted signal is applied to a matched filter. The matched filter response is obtained by directly synthesizing the field-sync signal at base-band. A square-root raised cosine filter is used as part of the signal down-converter.
atsc_if_ref_circ2	This function generates a modulated carrier based on the ATSC field-sync pseudo random sequence. The modulated carrier is filtered using a SRRC filter with a bandwidth and roll-off factor as specified in the standard's documents. The sampling rate and carrier frequencies are those used for data recordings.

The channel estimation algorithms and programs provide flexibility in terms of the recorded signals IF carrier frequency, 8VSB symbol rate and sampling frequency of the data recording equipment. The processing options provided are meant to obtain the best possible results under different conditions. For example, the IF filtering option is useful if the IF signal presented to the recording equipment is contaminated by thermal noise over a bandwidth that is larger than the signal bandwidth. The time domain windowing options are useful to lower the noise floor of the estimated impulse response

when the multipath delay is relatively large. The use of matched filters extracted from an IF reference is useful only when the field-sync signal is based on measurements instead of synthesis based on analytical models. A measured reference is necessary only if the sampling rate had to be very arbitrary.

The results of this study will help in design the wireless interactive link for digital television and broadband wireless services, and better understand the TDMA BW-LAN data capacity and network structure for LMCS/DS, MMDS, ISM and NII-band implementations.

The future benefits of the developments undertaken in this work include, but are not limited to the following:

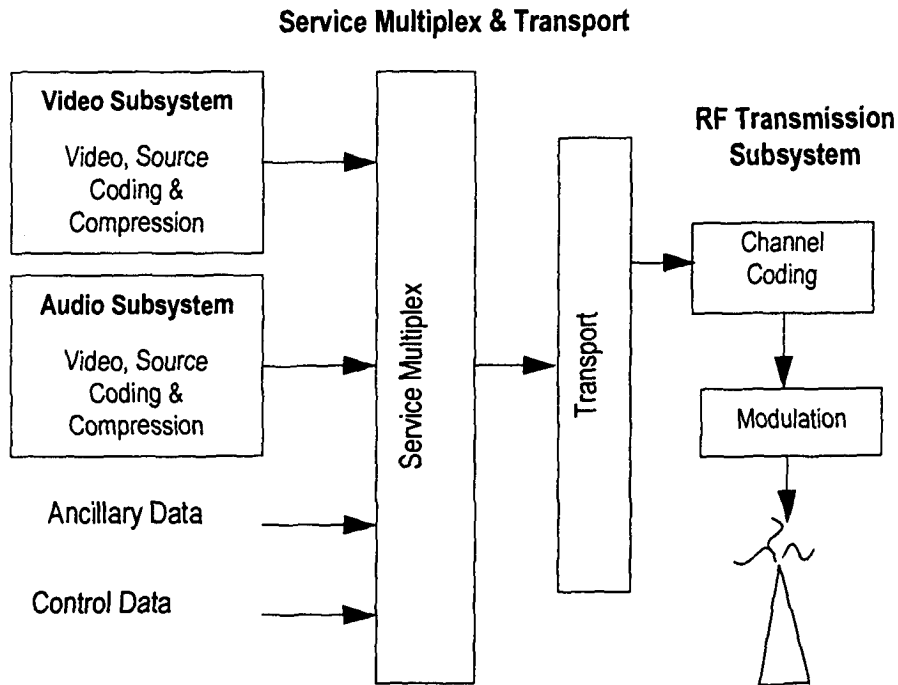
- 1) Establish dependable return link channel models for various locations, various receiving antenna patterns and varying channel conditions.
- 2) Establish statistical models for the temporal and spatial variations for the impulse response of the channel.
- 3) Determine the maximum and rms delay spread of the channel under various measurement conditions.
- 4) Utilize the information gained about the channel to quantify the error rate performance for broadband wireless up-link and digital interactive broadcasting service based on various modulation and up-link access techniques.
- 5) Extend the current set of algorithms and programs to other DTV standard signals such as OFDM.

# I Overview of the ATSC Digital Television Standard and Background

## I.1. DTV System Overview

The digital television system can be seen to consist of three subsystems.

1. Source coding and compression,
2. Service multiplex and transport, and
3. RF/Transmission.



**Fig. 1 SYSTEM BLOCK DIAGRAM**

The "Source coding and compression" subsystem is concerned with the bit rate reduction methods, also known as data compression, appropriate for application to the video, audio, and ancillary digital data streams.

The ancillary data stream includes control data, conditional access control data, and data associated

with the program audio and video services, such as closed captioning. The term *Ancillary data* can also refer to independent program services.

The purpose of *the audio/video coder* is to minimize the number of bits needed to represent the audio and video information.

The digital television system employs the MPEG-2 video stream syntax for the coding of video and the Digital Audio Compression (AC-3) Standard for the coding of audio.

*The Service multiplex and transport subsystem is concerned with:*

The means of dividing the digital data stream into “packets” of information,

The means of uniquely identifying each packet or packet type,

And the appropriate methods of multiplexing video data stream packets, audio data stream packets, and ancillary data stream packets into a single data stream.

In developing the transport mechanism, interoperability among digital media, such as terrestrial broadcasting, cable distribution, satellite distribution, recording media, and computer interfaces, was a prime consideration.

The digital television system employs the MPEG-2 transport stream syntax for the packetization and multiplexing of video, audio, and data signals for digital broadcasting systems.

*The RF/Transmission subsystem* is concerned with the channel coding and modulation.

The channel coder takes the data bit stream and adds additional information that can be used by the receiver to reconstruct the data from the received signal.

The modulation (or physical layer) uses the digital data stream information to modulate the transmitted signal.

The modulation subsystem offers two modes: a terrestrial broadcast mode (8 VSB), and a high data rate mode (16 VSB).

## I.2. I.2 RF TRANSMISSION SUBSYSTEM

### I.2.1 SUBSYSTEM OVERVIEW

Figure 2 depicts a representative block diagram of an 8VSB terrestrial broadcast transmitter.

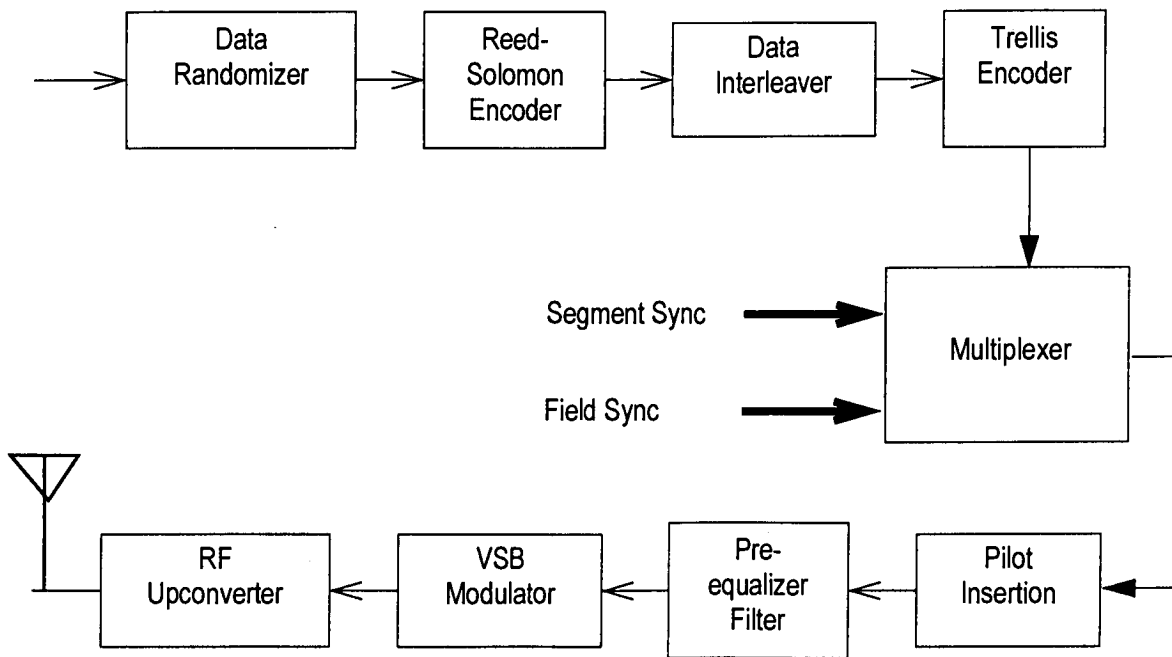


Fig. 2 a representative 8VSB terrestrial broadcast transmitter.

The key features of figure 2 are:

The terrestrial broadcast mode (known as 8 VSB) supports a payload data rate of 19.28... Mbps in a 6 MHz channel (The exact numbers are presented in section xxxx).

The input to the transmission subsystem from the transport subsystem is a 19.39... Mbps serial data stream comprised of 188-byte MPEG-compatible data packets.

The incoming data is randomized (for energy dispersal)

The randomized data is processed for forward error correction (FEC) in the form of Reed-Solomon (RS) coding (20 RS parity bytes are added to each packet)

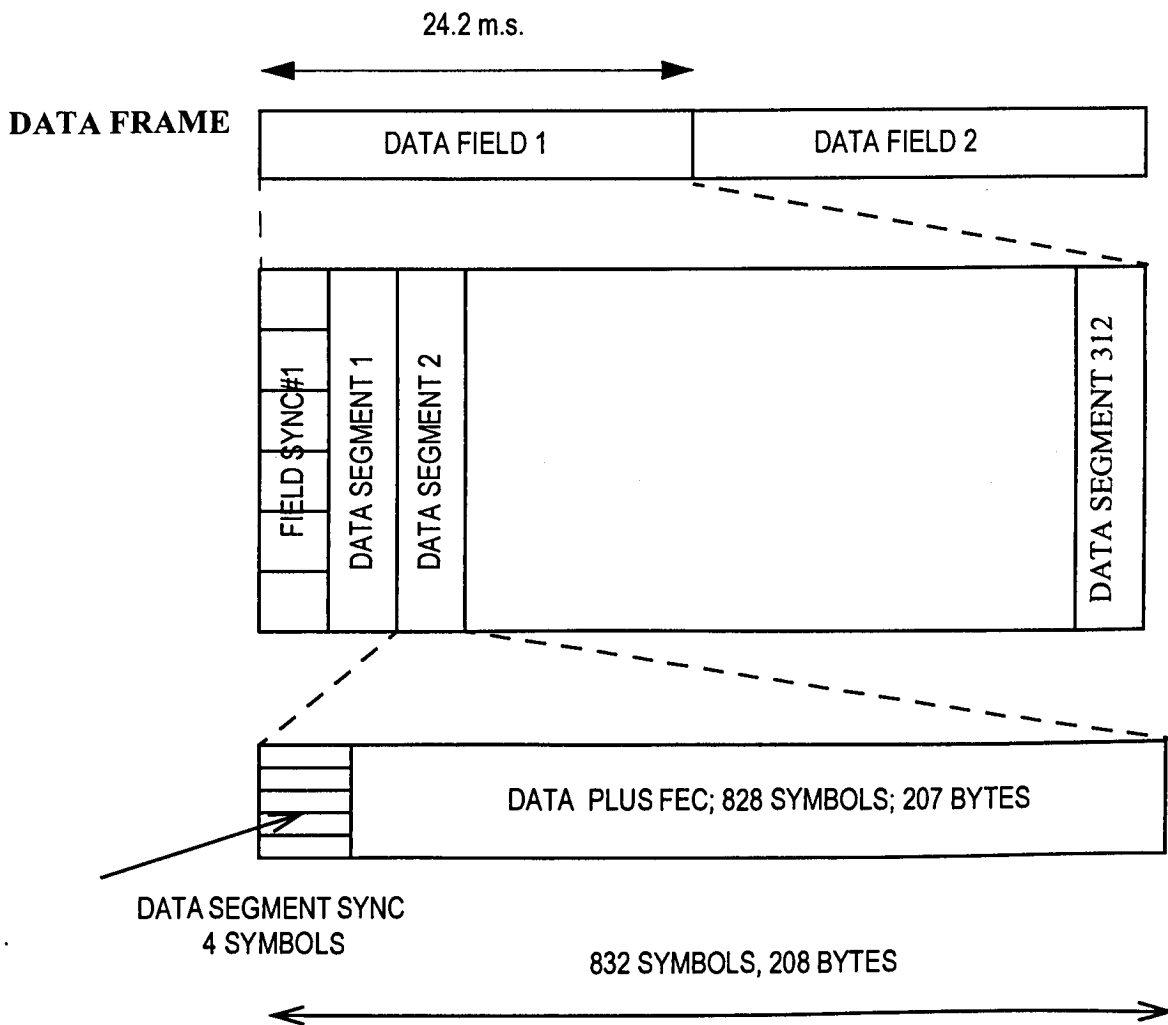
1/6 data field interleaving and 2/3 rate trellis coding are applied to the output of the R-S encoder.

The randomization and FEC processes are **NOT** applied to the sync byte of the transport packet

The data randomized and coded packets are formatted into Data Frames for transmission

Data Segment Sync and Data Field Sync are added.

### I.2.2 FRAMES, FIELDS AND SEGMENTS



**Fig. 3** Frames Fields and Segments; data organization for transmission

Figure 3 shows how the data is arranged for transmission. Each data frame consists of two data fields, each containing 313 data segments. The first data segment of each data field is a unique synchronization signal (Data Field Sync) and includes Pseudo Random training sequences which are

required for equalizer training and also used for channel estimation as demonstrated in chapter 2 of this report. The remaining 312 data segments each carry the equivalent of the data from one 188-byte transport packet plus its associated FEC overhead.

Each data segment consists of 832 symbols. The first 4 symbols are transmitted in binary form and provide segment synchronization. This data segment sync signal also represents the sync byte of the 188-byte MPEG-compatible transport packet. The remaining 828 symbols of each data segment carry data equivalent to the remaining 187 bytes of a transport packet and its associated FEC overhead. These 828 symbols are transmitted as 8 level signals and therefore carry 3 bits per symbol.

### **1.2.3 CHANNEL ERROR PROTECTION AND SYNCHRONIZATION**

#### **Data randomizer**

A data randomizer is used on all input data to randomize the data payload (not including Data Field Sync or Data Segment Sync, or RS parity bytes).

-The data randomizer XORs all the incoming data bytes with a 16-bit maximum length pseudo random binary sequence (PRBS) which is initialized at the beginning of the Data Field. The PRBS is generated in a 16-bit shift register that has 9 feedback taps.

Eight of the shift register outputs are selected as the fixed randomizing byte, where each bit from this byte is used to individually XOR the corresponding input data bit. The data bits are XORed MSB to MSB... LSB to LSB.

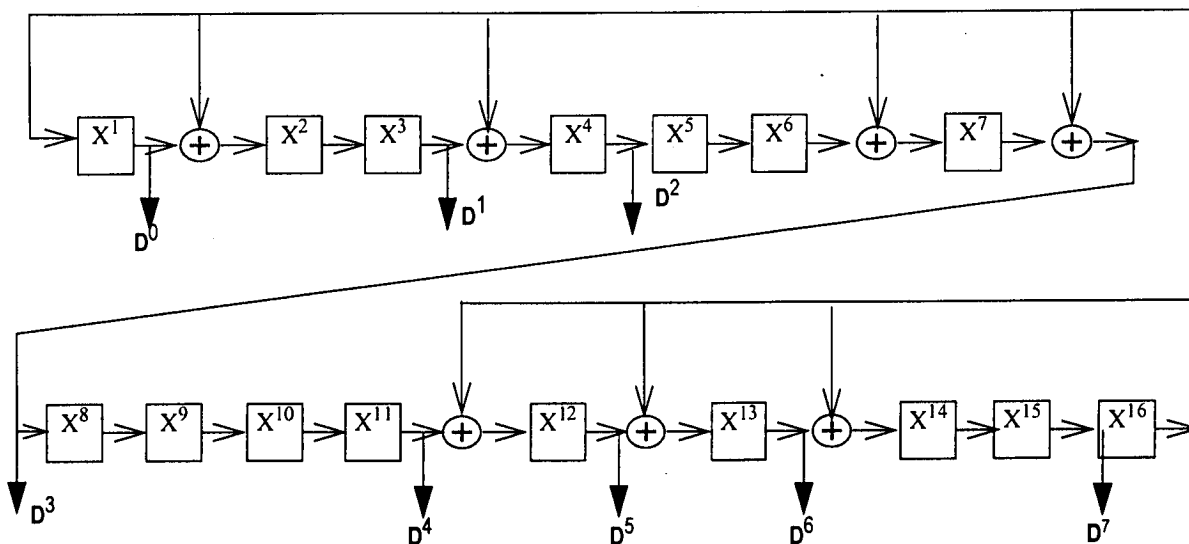
The randomizer generator polynomial is as follows:

$$G(X) = X^{16} + X^{13} + X^{12} + X^{11} + X^7 + X^6 + X^3 + X + 1$$

The initialization (pre-load) to F180 hex (1111000110000000 binary) occurs during the Data Segment Sync interval prior to the first Data Segment.



The randomizer generator polynomial and initialization is shown in figure 4.



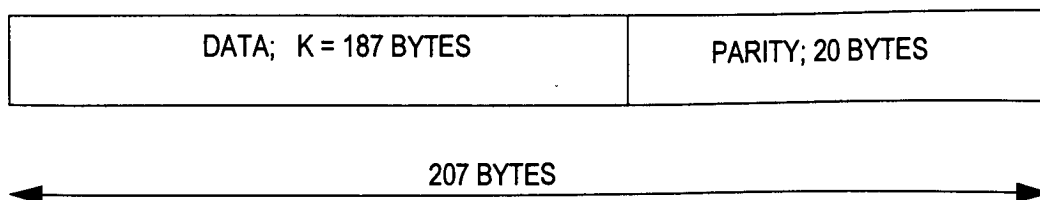
Initialization:  $x^{16} \dots x^1 = 1111000110000000$

Generator shifted with the Byte clock

One Byte of data is extracted per cycle

**Fig. 4 DATA RANDOMIZER**

## REED-SOLOMON ENCODER



**Fig. 5 Reed Solomon coded word**

The RS code used in the VSB transmission subsystem is a  $t = 10$  (207,187) code. In other words, the RS data block size is 187 bytes, with 20 RS parity bytes added for error correction as shown in figure 5. It should be noted that:

A total RS block size of 207 bytes is transmitted per Data Segment.

In creating bytes from the serial bit stream, the MSB shall be the first serial bit.

The 20 RS parity bytes shall be sent at the end of the Data Segment.

The parity generator polynomial for the R-S encoder is:

$$g(x) = x^{20} + 152x^{19} + 185x^{18} + 240x^{17} + 5x^{16} + 111x^{15} + 99x^{14} + 6x^{13} + 220x^{12} + 112x^{11} + 150x^{10} + 69x^9 + 36x^8 + 187x^7 + 22x^6 + 228x^5 + 198x^4 + 121x^3 + 121x^2 + 165x^1 + 174$$

## INTERLEAVING

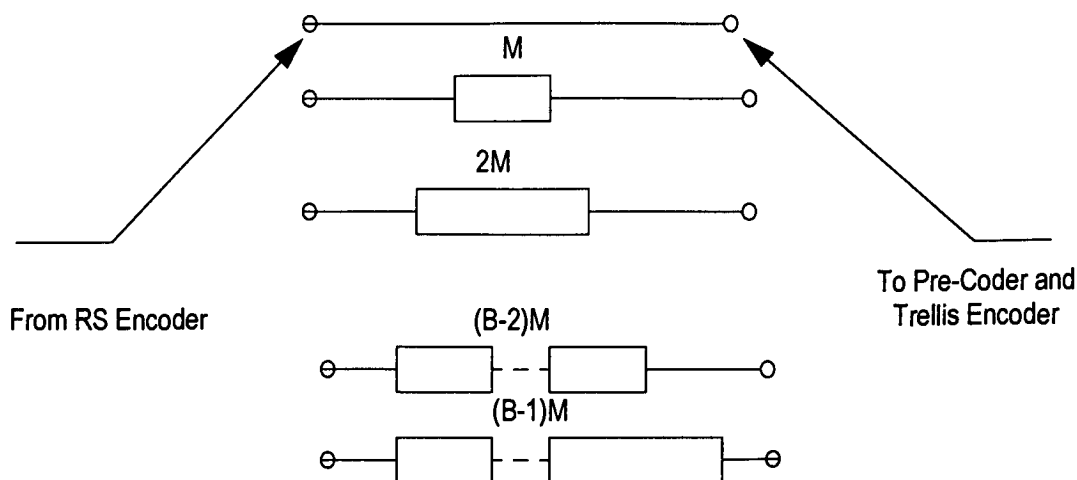


Fig. 6 CONVOLUTIONAL INTERLEAVER.  $M=4$ ,  $B=52$ ,  $N=208$ , RS-block= 207,  $BM=N$

The interleaver employed in the VSB transmission system is a 52 data segment convolutional byte interleaver as shown in figure 6. The complementary de-interleaver is shown in figure 7. Interleaving

is provided to a depth of about 1/6 of a data field (4 ms deep). Only data bytes shall be interleaved.

The interleaver shall be synchronized to the first data byte of the data field. Intra-segment interleaving is also performed for the benefit of the trellis coding process.

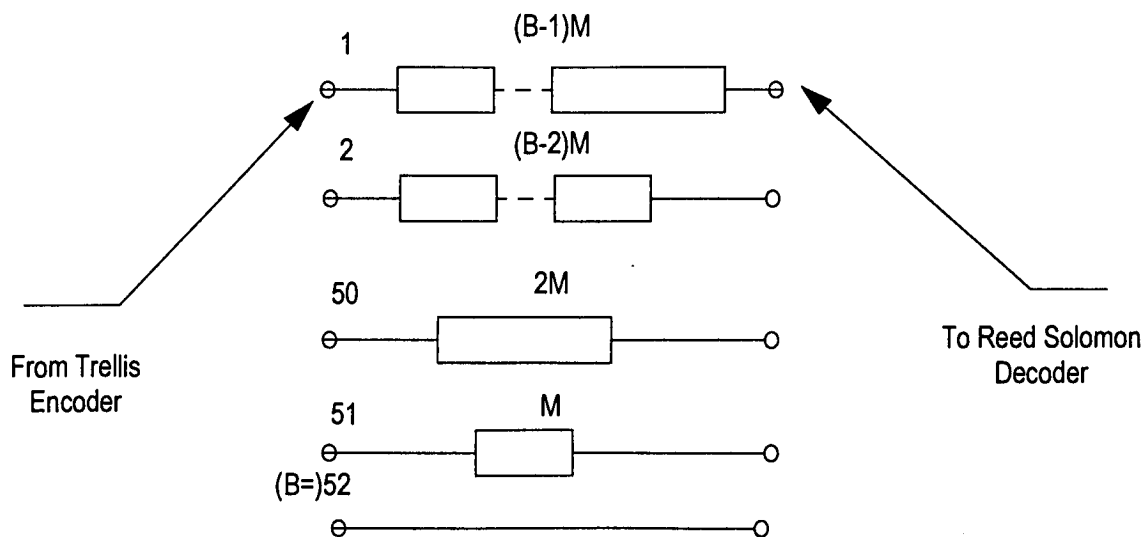


Fig. 7 DE-INTERLEAVER.  $M=4$ ,  $B=52$ ,  $N=208$ , RS-block= 207,  $BM=N$

## TRELLIS CODING

The 8 VSB transmission sub-system employs a 2/3 rate ( $R=2/3$ ) trellis code (with one unencoded bit which is precoded) as shown in figure 8.

That is, one input bit is encoded into two output bits using a 1/2 rate convolutional code while the other input bit is precoded.

The signaling waveform used with the trellis code is an 8-level (3 bit), one-dimensional constellation. The transmitted signal is referred to as 8 VSB.

Trellis code intra-segment interleaving shall be used. This uses twelve identical trellis encoders and precoders operating on interleaved data symbols.

The code interleaving is accomplished by encoding symbols (0, 12, 24, 36...) as one group, symbols (1, 13, 25, 37,...) as a second group, symbols (2, 14, 26, 38,...) as a third group, and so on for a total of

12 groups.

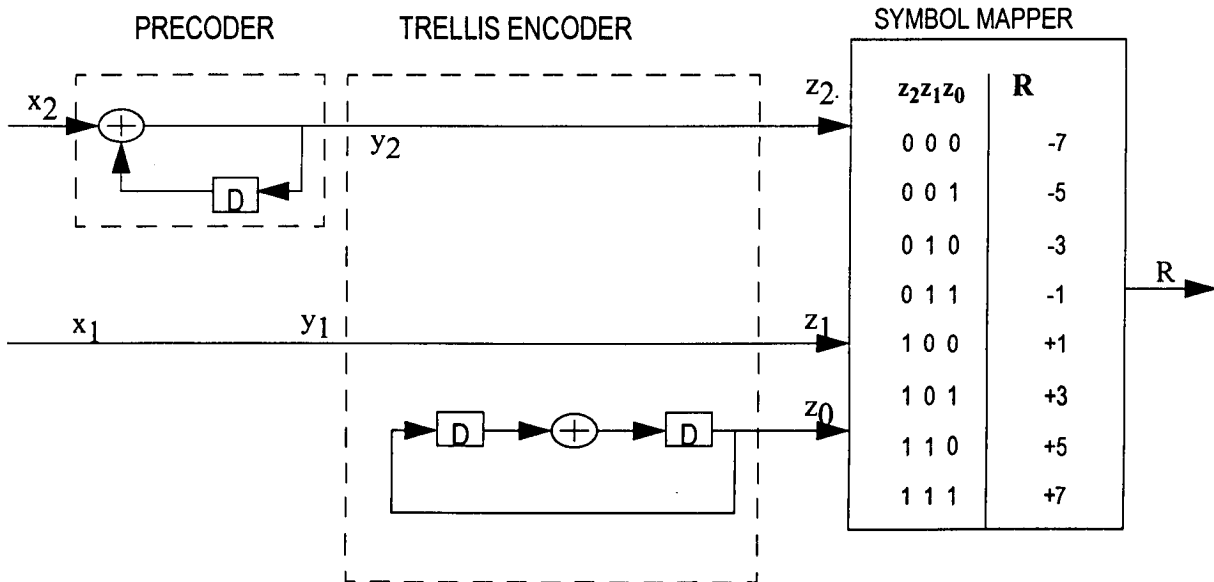


Fig. 8 8 VSB TRELLIS ENCODER, PRECODER AND SYMBOL MAPPER

Figure 9 shows the Trellis-Code interleaver.

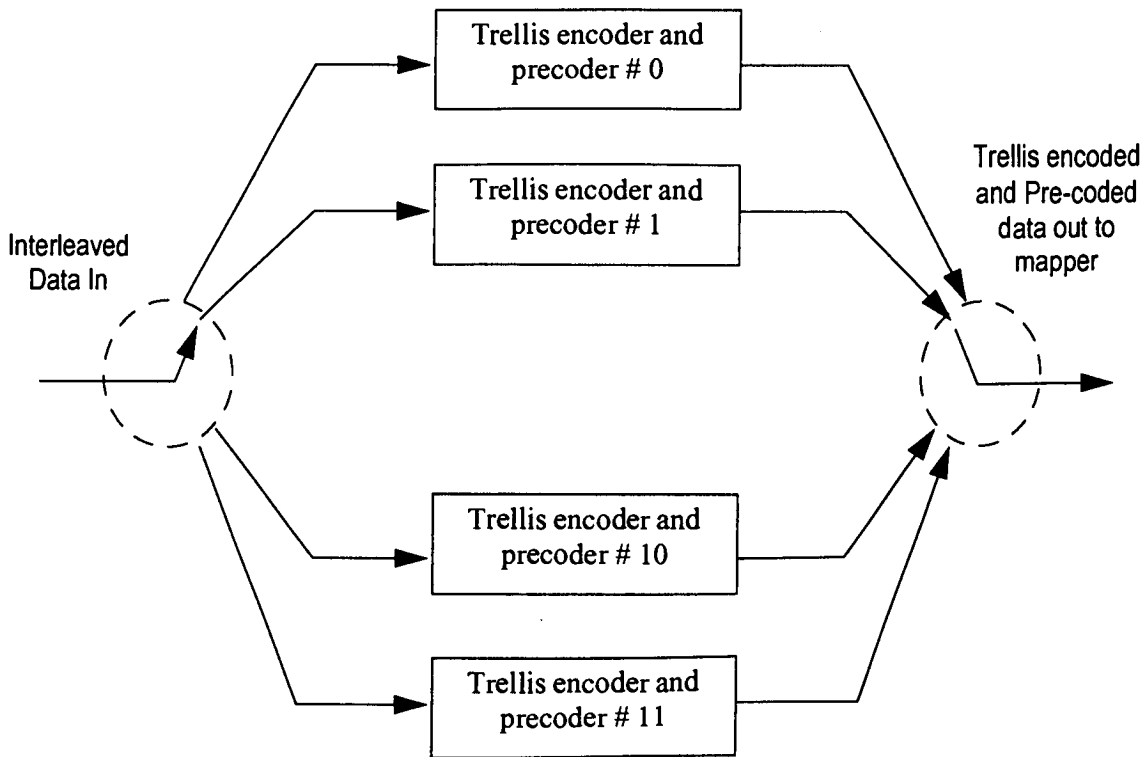


Fig. 9 TRELLIS CODE INTERLEAVER

#### I.2.4 DATA SEGMENT SYNC

The encoded trellis data is passed through a multiplexer that inserts the various synchronization signals (Data Segment Sync and Data Field Sync). A two-level (binary) 4-symbol Data Segment Sync shall be inserted into the 8-level digital data stream at the beginning of each Data Segment as depicted in figure 10.

In other words, the MPEG sync byte shall be replaced by Data Segment Sync..

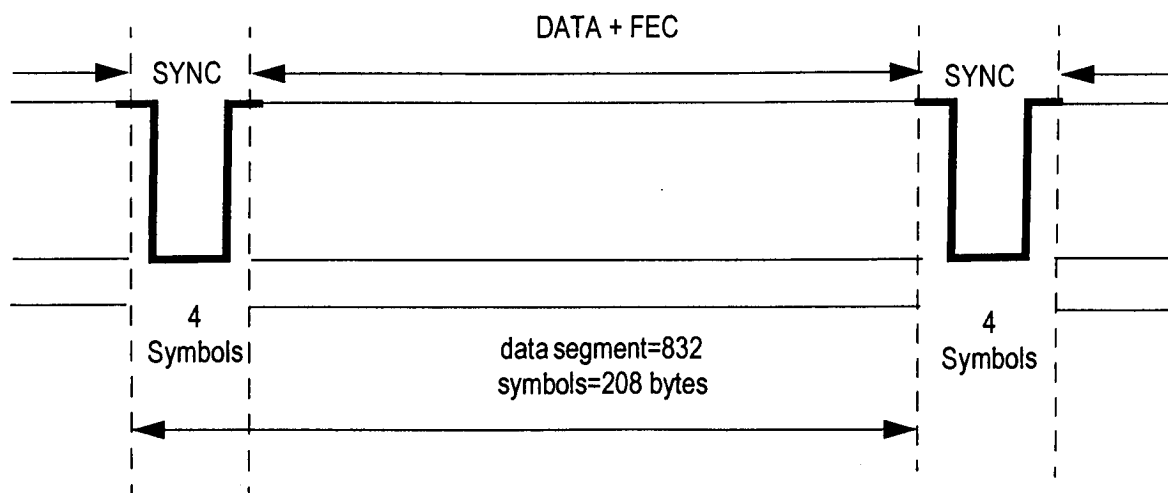


Fig. 10 Data Segment Sync

A complete segment will thus consist of 832 symbols: 4 symbols for Data Segment Sync, and 828 data plus parity symbols. The Data Segment Sync is binary (2-level).

The same sync pattern occurs regularly at 77.3  $\mu$ s intervals, and is the only signal repeating at this rate. Unlike the data, the four symbols for Data Segment Sync are not Reed-Solomon or trellis encoded, nor are they interleaved. The Data Segment Sync patterns a 1001 pattern as shown in figure 10

#### I.2.5 The Data Field Sync

Figure 4 shows the detailed structure of the data field sync. The features of the data field sync are:

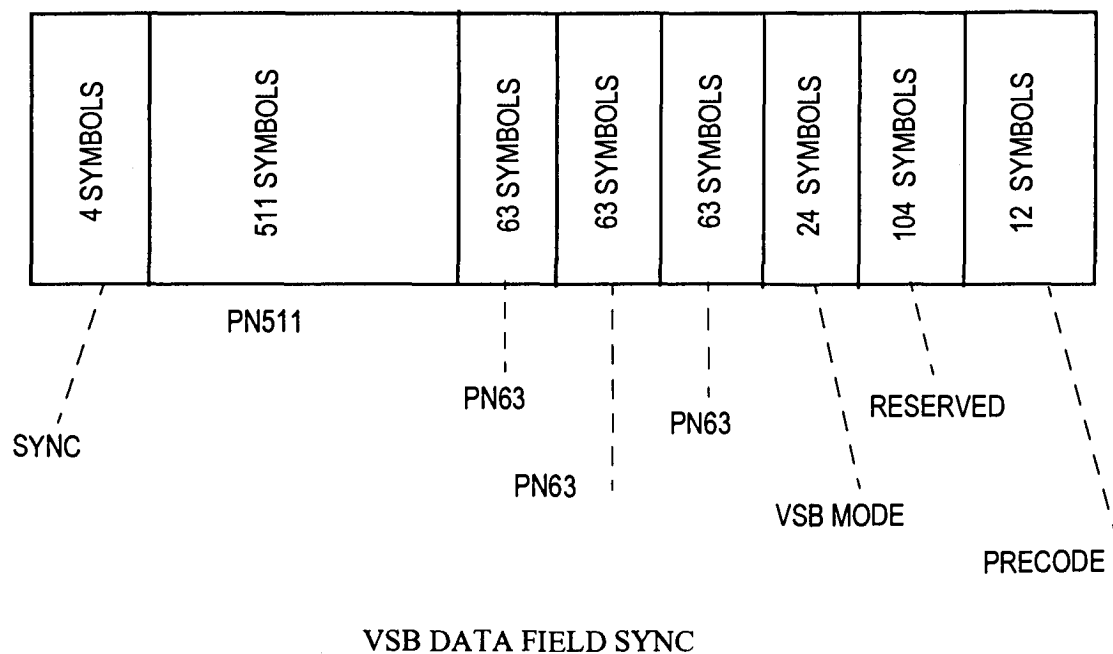
- Each Data Frame consists of two Data Fields, each containing 313 Data Segments.

The first Data Segment of each Data Field is a unique synchronizing signal (Data Field Sync) and includes the training sequence used by the equalizer in the receiver.

The remaining 312 Data Segments each carry the equivalent of the data from one 188-byte transport packet plus its associated FEC overhead. Each Data Segment consists of 832 symbols.

The first 4 symbols are transmitted in binary form and provide segment synchronization.

The remaining 828 symbols of each Data Segment carry data equivalent to the remaining 187 bytes of a transport packet and its associated FEC overhead. These 828 symbols are transmitted as 8-level signals and therefore carry three bits per symbol. Thus,  $828 \times 3 = 2484$  bits of data are carried in each Data Segment.



With reference to figure xxxxx note that:

**Sync** corresponds to Data Segment Sync and is defined as 1001.

**PN511:** This pseudo-random sequence is defined by the generator polynomial:

$$g(x) = x^9 + x^7 + x^6 + x^4 + x^3 + x + 1$$

with a pre-load value of 010000000.

**PN63:** This pseudo-random sequence is repeated three times. It is defined by the generator polynomial:

$$g(x) = x^6 + x + 1$$

with a pre-load value of 100111.

The middle PN63 is inverted on every other Data Field Sync.

**VSB mode:** These 24 bits determine the VSB mode for the data in the frame. The first two bytes are reserved. The suggested fill pattern is 0000 1111 0000 1111. The next byte is defined as:

P A B C P A B C where:

P A B C

1 1 0 0            16VSB

0 1 0 1            8 VSB\*

**Reserved:** The last 104 bits shall be reserved space.

**Precode:** In the 8 VSB mode, the last 12 symbols of the “Reserved” segment shall correspond to the last 12 symbols of the previous segment.

All sequences are pre-loaded before the beginning of the Data Field Sync.

Like the Data Segment Sync, the Data Field Sync is not Reed-Solomon or trellis encoded, nor is it interleaved.

### **Pilot addition**

A small in-phase pilot shall be added to the data signal. The frequency of the pilot shall be the same as the suppressed-carrier frequency.

A small (digital) DC level (1.25) shall be added to every symbol (data and sync) of the digital baseband data plus sync signal ( $\mp 1, \mp 3, \mp 5, \mp 7$ ) .

The power of the pilot is therefore  $10\log\left(\frac{1 + 9 + 25 + 49}{4 \times 1.25^2}\right) = 11.28 \text{ dB}$  below the average data signal power.

### I.3. System Parameters

The exact symbol rate of the transmission subsystem is given by:

$$S_r = \frac{4.5}{286} \times 684 = 10.762238 \text{ Msps} \quad (1)$$

The symbol rate must be locked in frequency to the transport rate. The transmission subsystem carries 2 information bits per trellis-coded symbol, so the gross payload is:

$$gpl = \frac{4.5}{286} \times 684 \times 2 = 21.524476 \text{ Mbps} \quad (2)$$

To find the net payload delivered to a decoder it is necessary to adjust (2) for the overhead of the Data Segment Sync, Data Field Sync, and Reed-Solomon FEC.

Upon doing this the net payload bit rate of the 8 VSB terrestrial transmission subsystem becomes:

$$npl = 21.524476 \times \frac{312}{313} \times \frac{828}{832} \times \frac{187}{207} = 19.289506 \text{ Mbps} \quad (3)$$

The factor of 312/313 accounts for the Data Field Sync overhead of one Data Segment per field. The factor of 828/832 accounts for the Data Segment Sync overhead of four symbol intervals per Data Segment, and the factor of 187/207 accounts for the Reed-Solomon FEC overhead of 20 bytes per Data Segment.

The calculation of the net payload bit rate of the high data rate mode is identical except that 16 VSB carries 4 information bits per symbol. Therefore, the net bit rate is twice that of the 8 VSB terrestrial



mode:

$$19.289506 \times 2 = 38.579012 \text{ Mbps} \quad (4)$$

To get the net bit rate seen by a transport decoder, however, it is necessary to account for the fact that the MPEG sync bytes are removed from the data stream input to the 8 VSB transmitter. This amounts to the removal of one byte per data segment. These MPEG sync bytes are then reconstituted at the output of the 8 VSB receiver. The net bit rate seen by the transport decoder is:

$$19.289506 \times \frac{188}{187} = 19.392658 \text{ Mbps} \quad (5)$$

The net bit rate seen by the transport decoder for the high data rate mode is:

$$19.392658 \times 2 = 38.785317 \text{ Mbps} \quad (6)$$

The frequency of a Data Segment is given in equation 2 below:

$$f_{seg} = \frac{S_r}{832} = \frac{10.762238}{832} = 12.935382 \times 10^3 \text{ data segments/second} \quad (7)$$

The Data Frame rate is given by equation (3) below:

$$f_{frame} = \frac{f_{seg}}{626} = \frac{12.935382 \times 10^3}{626} = 20.663549 \text{ frames per second} \quad (8)$$

Since a Frame contains two data fields, the number of data fields per second is given by

$$f_{data \text{ fields}} = 20.663549 \times 2 = 41.327099 \text{ data fields/second} \quad (9)$$

The time duration of a single data field is thus given by:

$$T_{data \text{ field}} = 1/f_{data \text{ fields}} = 24.197198 \text{ milli second} \quad (10)$$

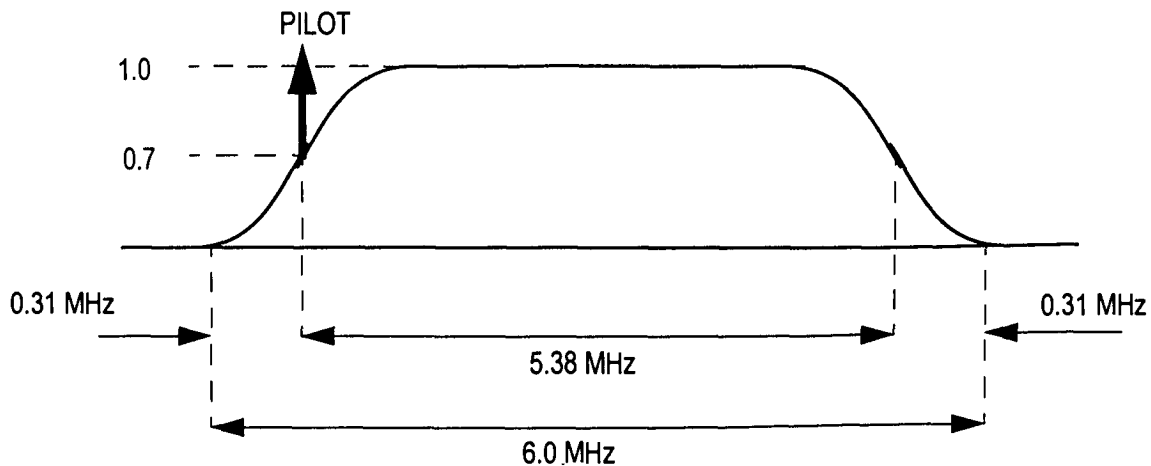
**Table 2: Parameters for VSB Transmission Modes**

Parameter	Terrestrial mode	High data rate mode
Channel bandwidth	6 MHz	6 MHz
Excess bandwidth	11.5%	11.5%

**Table 2: Parameters for VSB Transmission Modes**

Parameter	Terrestrial mode	High data rate mode
Symbol rate	10.76 Msymbols/s	10.76 Msymbols/s
Bits per symbol	3	4
Trellis FEC	2/3 rate	None
Reed-Solomon FEC	T=10 (207,187)	T=10 (207,187)
Segment length	832 symbols	832 symbols
Segment sync	4 symbols per segment	symbols per segment
Frame sync	1 per 313 segments	1 per 313 segments
Payload data rate	19.28 Mbps	38.57 Mbps
NTSC co-channel rejection	NTSC rejection filter in receiver	N/A
Pilot power contribution	0.3 dB	0.3 dB
C/N threshold	14.9 dB	28.3 dB

#### I.4. Modulation



**Fig. 11 VSB CHANNEL OCCUPANCY (NOMINAL)**

The 8-level symbols combined with the binary Data Segment Sync and Data Field Sync signals are used to suppressed-carrier modulate a single carrier.

Before transmission, most of the lower sideband shall be removed. The resulting spectrum is flat, except for the band edges where a nominal square root raised cosine response results in 620 kHz transition regions. The nominal VSB transmission spectrum is shown below.

At the suppressed-carrier frequency, 310 kHz from the lower band edge, a small pilot shall be added to the signal (the pilot power is 11.28 dB below the signal power).

The system performance is based on a linear phase raised cosine Nyquist filter response in the concatenated transmitter and receiver. The raised cosine response is essentially flat across the entire range.

### 1.5. Raised Cosine Filtering

The frequency response of the raised cosine filter **at baseband** is defined as follows:

$$w(f) = \begin{cases} T & 0 \leq |f| \leq \frac{1-\beta}{2T} \\ \frac{T}{2} \left\{ 1 + \cos \left[ \frac{\pi T}{\beta} \left( |f| - \frac{1-\beta}{2T} \right) \right] \right\} & \frac{1-\beta}{2T} \leq |f| \leq \frac{1+\beta}{2T} \\ 0 & |f| > \frac{1+\beta}{2T} \end{cases} \quad (11)$$

$$w(t) = \frac{\sin(\pi(t/T)) \cos((\pi\beta t)/T)}{\pi(t/T) \left[ 1 - ((2\beta t)/T)^2 \right]} \quad (12)$$

$0 \leq \beta \leq 1$  is the ROLL-OFF FACTOR of the filter

$T$  is the transmitted symbol period.

$w(f)$  is usually split evenly between the transmitter and the receiver. Therefore, the transmitter filter response is a SRRC in the frequency domain.

The bandwidth required for the transmission of  $R_s$  symbols/s, without ISI, is therefore:

$$BW = R_s \times (1 + \beta) \quad (13)$$

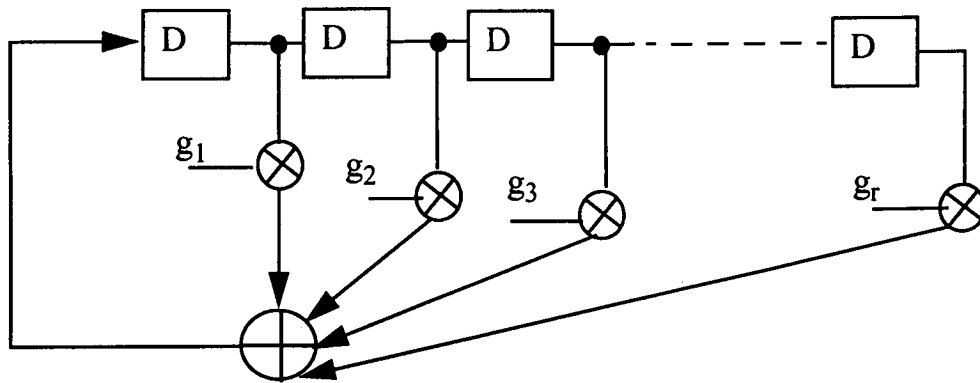
which is larger than the minimum bandwidth requirement. In other words, we have used an excess bandwidth for the sake of making the system more practical. The impulse response of a root-raised cosine filter is given by:

$$h(t) = \frac{4\beta}{\pi\sqrt{T}} \frac{\cos\left[(1+\beta)\pi\frac{t}{T}\right] + \frac{\sin\left[(1-\beta)\pi\frac{t}{T}\right]}{4\beta\frac{t}{T}}}{1 - \left(4\beta\frac{t}{T}\right)^2} \quad (14)$$

The roll-off factor used for 8VSB is

$$\beta = \frac{0.62}{(10.762/2)} \times 100 = 11.5 \% \quad (15)$$

#### I.6. Binary Shift Register Sequences:



**Fig. 12** Linear Feedback shift register circuit that can be used as a PNS generator

Binary shift register sequences can be generated using an  $r$ -stage binary shift register connected as shown in figure 12, where the coefficients  $g_1, g_2, \dots, g_r$  are binary (1 or 0). The addition is a modulo two operation.

The properties of the resulting PN sequence are dependent on the selected feedback coefficients.

The maximum possible sequence period for an r-stage shift register is  $(2^r - 1)$ .

Shift register sequences having the maximum possible period for an r-stage register are called maximal length sequences. m-sequences have the following well known properties:

an m-sequence contains one more one than zero

The modulo-two sum of an m-sequence, and any phase shift of the same sequence is another phase of the same m-sequence.

If a window of width r is slid along the sequence for  $N = 2^r - 1$  shifts, each r-tuple except the all zero tuple will occur exactly once.

The periodic autocorrelation function, of an m-sequence with period N, is two-valued:

$$\theta(k) = \begin{cases} 1.0 & k = LN \\ -\frac{1}{N} & (k \neq LN) \end{cases} \quad (16)$$

where L is any integer, and N is the sequence period

The power spectral density of c(t) is given by the Fourier transform of the autocorrelation function shown above. The result is:

$$S_c(f) = \sum_{m=-\infty}^{\infty} P_m \delta(f - mf_0) \quad (17)$$

where  $P_0 = 1/N^2$ ,  $P_m = [(N+1)/N^2] \text{sinc}^2(m/N)$  and  $f_0 = 1/NT_c$ . This spectrum consists of discrete spectral lines at all harmonics of  $1/NT_c$ . The envelope of the squared magnitude of these lines is given by  $[(N+1)/N^2] \text{sinc}^2(fT_c)$  except for the D.C. term which has an amplitude  $1/N^2$ .

## **II Data Analysis and Channel Estimation Algorithms**

### **II.1. Introduction**

This chapter discusses the signal processing chain used for the analysis of experimentally recorded data for the purpose of estimating the channel response. Two scenarios for data recordings have been encountered. The two scenarios correspond to Laboratory-type experiments, and field experiments. In the following subsections we discuss each of the two cases in order to identify the main variables likely to affect the signal processing functions discussed later on in this chapter.

### **II.2. Laboratory Experiments**

Figure 13 shows the block diagram of a laboratory set-up used by CRC during the summer of 2001 and fall of 2002, which resulted in the first set of data files analyzed under this contract. While the diagram is self explanatory, it is important to take note of the following:

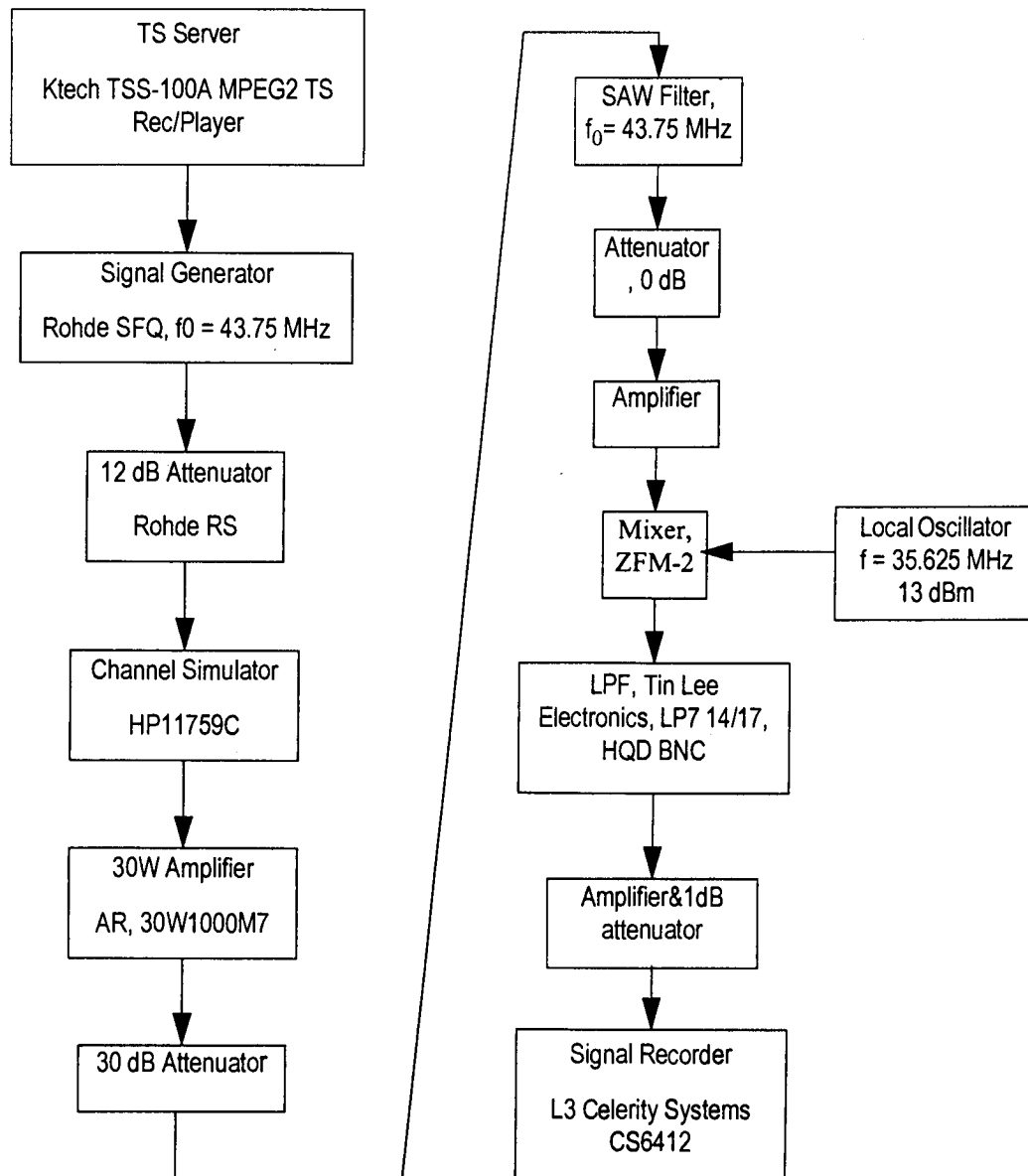
1) The 8VSB signal coming out of the DTV signal generator is applied to a Hewlett-Packard channel simulator for the purpose of intentionally introducing multipath.

2) The presence of a 43.75 MHz bandpass SAW filter and another lowpass filter in the receiving chain are expected to change the received DTV signal from its ideal (analytical) form. It should be noted that the use of such filters is unavoidable in this kind of experiments.

3) The received signal is down converted from an IF of 43.75 MHz to a lower IF of 8.125 MHz for recording purposes. This final IF frequency is not absolutely accurate. There will always be an amount of errors the magnitude of which depends on the stability of the oscillators involved in the set-up.

4) The received signal at a carrier of 8.125 MHz is recorded using the L3 Celerity Systems CS6412

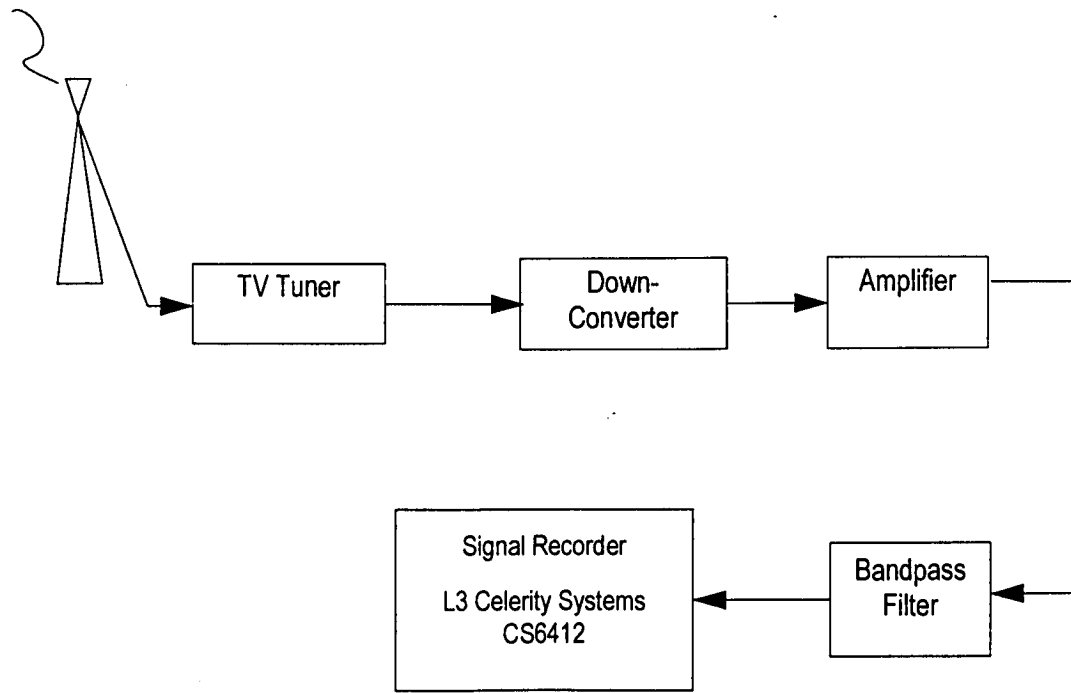
data recording system with a nominal resolution of 12 bits per sample. The actual resolution may be slightly less than 12-bits. The nominal sampling rate used for the first set of experiments was 32.5 MHz which is not an exact multiple of the 8VSB symbol rate. The sampling rate also has its own tolerance. The finite resolution of the data recording system introduces a noise-floor to the received signal (quantization noise).



**Fig. 13** Block Diagram of the Laboratory Set-up for Channel Recording which generated the first set of data in the summer of 2001 (Takada's Experiment)

Based on points 1-4 above we expected the recorded signal to have “built-in” uncertainties caused by: IF frequency errors, filtering distortion, quantization noise and sampling clock frequency errors.

### II.3. Field Experiments



**Fig. 14** Simplified Diagram of a Generic Set-Up That Could Be Used for the Recording of a Terrestrial DTV Signal.

Figure 14 shows a simplified set-up for the recording of emissions of an actual terrestrial DTV transmitter. The DTV signal may be intercepted using a commercial DTV tuner which brings the signal down to an IF of about 40-45 MHz. The tuner output is down converted to an IF frequency comparable to the 8VSB symbol rate, filtered and then applied to the data recording system. The signal recorded in such a scenario differs from a laboratory set-up signal in that:

- 1) The channel involved is a time-varying terrestrial channel which is not known beforehand.
- 2) The IF carrier frequency error is expected to be much larger given that the received signal had its carrier in the UHF band, and also due to the fact that commercial tuners are not expected to be very stable.
- 3) The received signal may also be corrupted by phase noise introduced by the commercial tuners.



- 4) The signal is also likely to be corrupted by the tuner thermal noise in light of the fact that the signal power is expected to be much lower than what is encountered in a laboratory environment
- 5) In addition to the factors above, we still expect errors caused by quantization noise and sampling clock frequency errors.

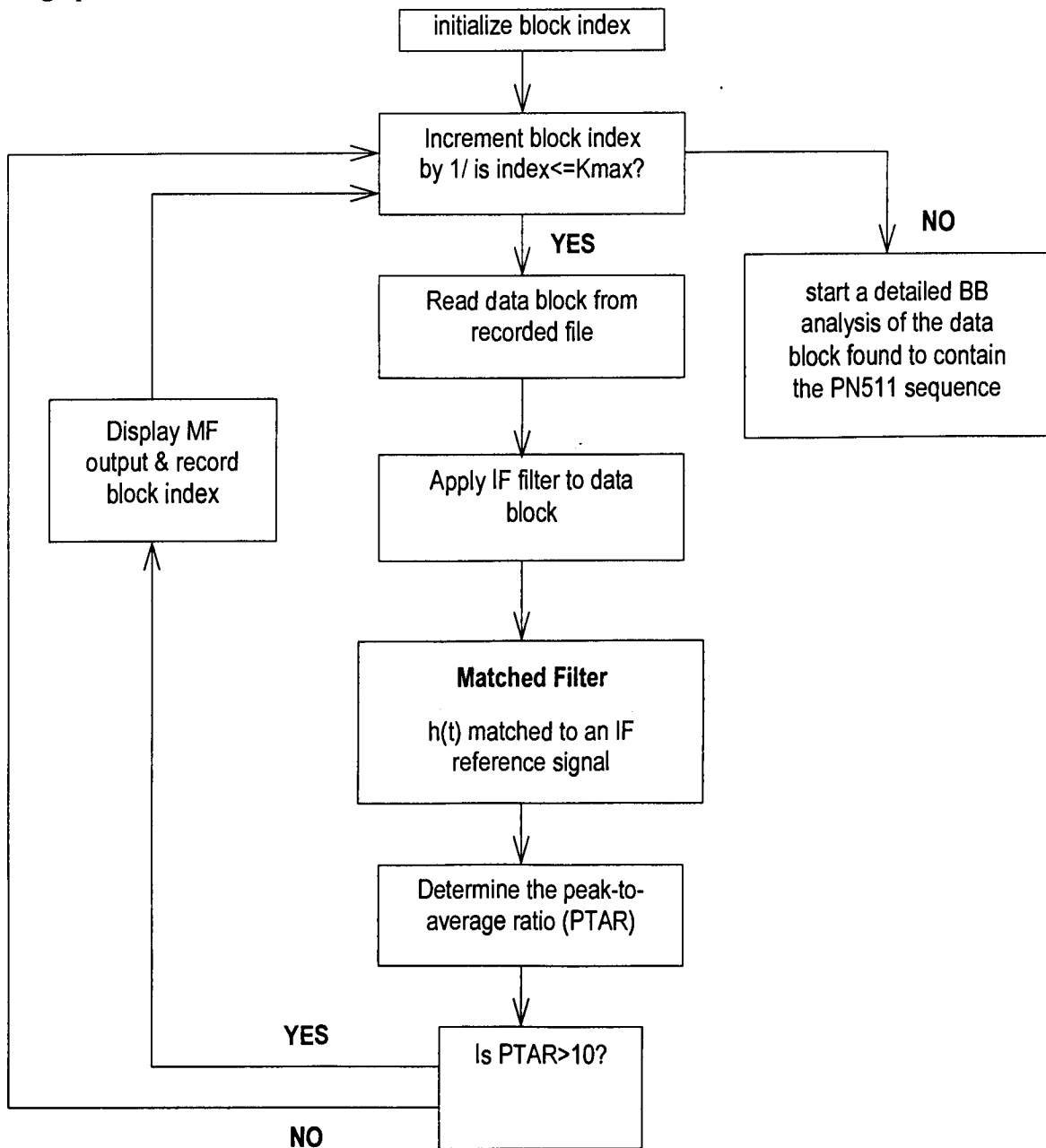
## **II.4. Structure of the data analysis programs**

### **II.4.1 IF Processing**

The main target for the data analysis is to estimate the channel response from field-recorded on-the-air DTV signals. The channel estimation relies entirely on the data field sync signals, and in particular the 511-bit PN sequence portion. The data-field sync is transmitted only once every 24.2 milliseconds (approximately), and the time duration of the PN sequence of concern is only  $511/10.762 \times 10^6 = 47.48 \mu s$ . As a result, only a very small portion of the recorded data (about 0.2%) is actually of use for channel estimation at this point (we are excluding any blind channel estimation algorithms at this stage). Therefore, the data analysis program has to perform the following tasks:

- 1) For a newly recorded file, the program must scan through the first data field, a 24.2 milli second, in order to locate the time position of the first 511-bit PN sequence. The scanning requires the application of an IF “matched filter” to the recorded data.
- 2) The recorded signal is not expected to exhibit a high signal-to-noise ratio always. As a result, it is necessary at times to apply an IF filtering function to the recorded data before doing any detailed analysis on the embedded 511 PN sequence. The program provides an option of IF filtering for cases where the received SNR is relatively low.
- 3) The channel response is obtained from a correlation function between the recorded data and a reference signal based on the 511 PN sequence. The reference signal can be obtained either through experiments as an IF signal, or synthesized based on analytical models extracted from the DTV standard document as explained in chapter I. We have elected to use the analytical models due to the higher accuracy that can be achieved and also due to the flexibility this approach provides in terms of

filtering options.



**Fig. 15** Processing Steps Followed in Scanning a Recorded Data File.

The objective is to determine the time location of the first pn511 sequence in the file.

4) Correlating the recorded IF signal with an IF reference produces approximate estimates for the channel multipath components, except that the results exhibit high frequency ringing at a rate of the

IF carrier frequency. This approach while fast in terms of processing yields results that are less obvious in terms of interpretation. The IF correlation is therefore used only to scan new files and to locate the first 511-bit PN sequence. Once this objective is met, the detailed analysis is carried out using baseband signals and baseband reference waveforms.

The preceding objectives have been incorporated in the data analysis program as shown in figure 15.

## II.4.2 Baseband Processing

Baseband signal processing is used for data analysis once the time position of the 511-bit PN sequence has been determined. The following functions, illustrated in figure 16, had to be implemented:

- 1) The recorded signal may need to be passed through an IF filter to for noise/interference limiting.
- 2) The filtered data is mixed (multiplication) with a complex carrier sequence of the form:

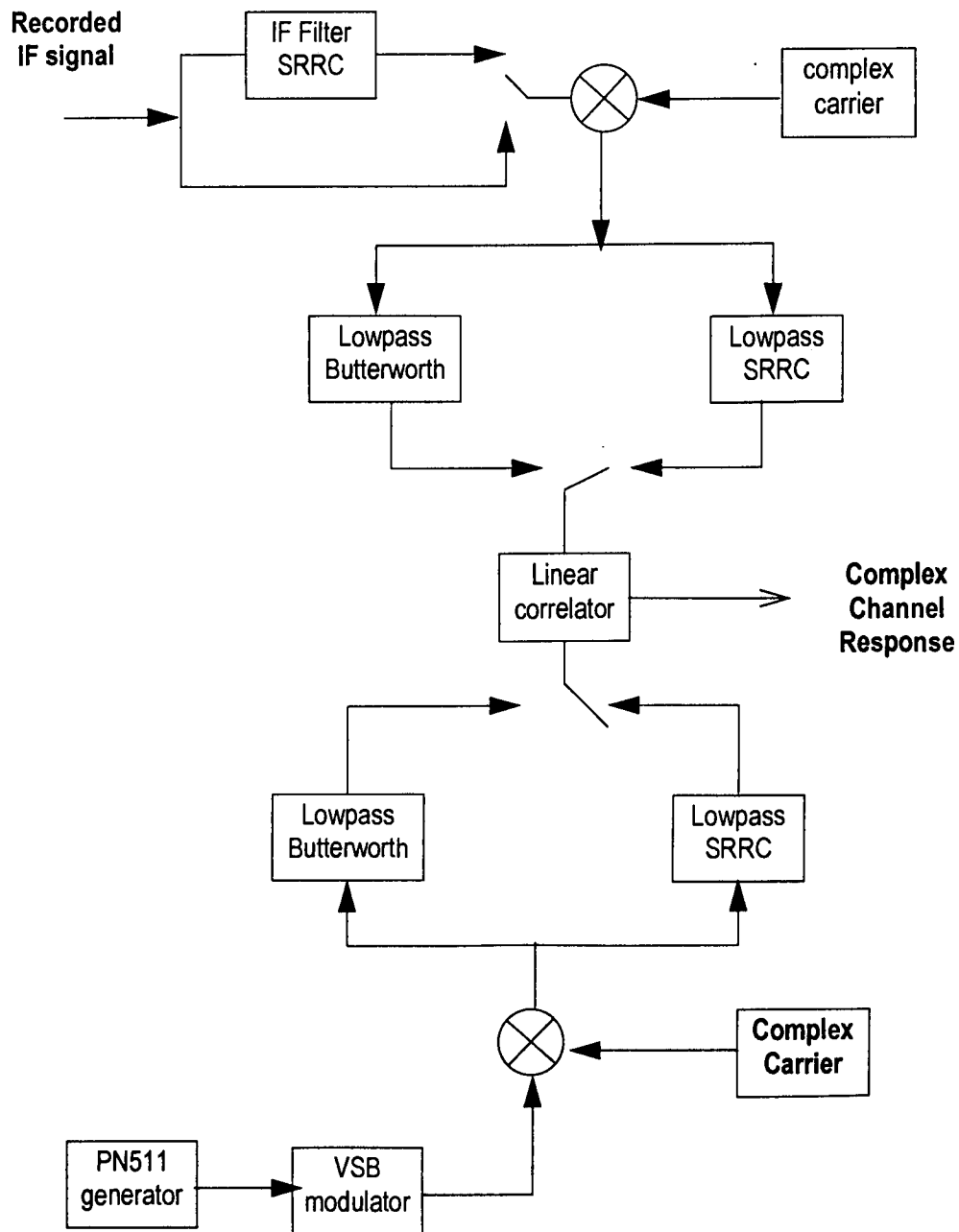
$$c(t) = \cos(2\pi f_0 t) + j\sin(2\pi f_0 t) = \exp(j2\pi f_0 t)$$

where  $f_0$  is the IF carrier frequency as seen at the input of the data recording system.

- 3) The mixer output contains the baseband signal components we are seeking in addition to a copy of the signal centered at twice the IF carrier frequency. Low pass filters are used to suppress the higher frequency terms. The type of lowpass filters used impacts the recovered baseband signal. The obvious choices for lowpass filters are: square-root raised cosine implemented as an FIR structure, or conventional IIR filters such as Butterworth lowpass.

The SRRC choice is most effective in terms of noise/interference limiting as it is matched to the spectrum of the recorded signal. A drawback of the SRRC is that it introduces further distortion to the signal in the event that the IF filtering function is used.

The IIR filtering option implements a 4th order Butterworth filter with a wide bandwidth (the bandwidth used is equal to 80% of the 8VSB symbol rate). The noise limiting effect of the IIR is slightly inferior. The signal distortion caused by the IIR filter is less significant compared to the previous case.



**Fig. 16 Signal Processing Chain Filtering Options Based on an IF Reference Signal**

4) The recovered baseband signal is next applied to a filter whose impulse response is matched to a

baseband reference signal. The baseband reference signal is obtained by passing a 511-bit PN sequence through an (analytical) 8VSB modulator running at the symbol rate of 10.762 M symbol/sec and which uses a carrier frequency identical to the IF carrier used for data recording. It should be noted that the matched filtering can also be realized by a linear correlation function.

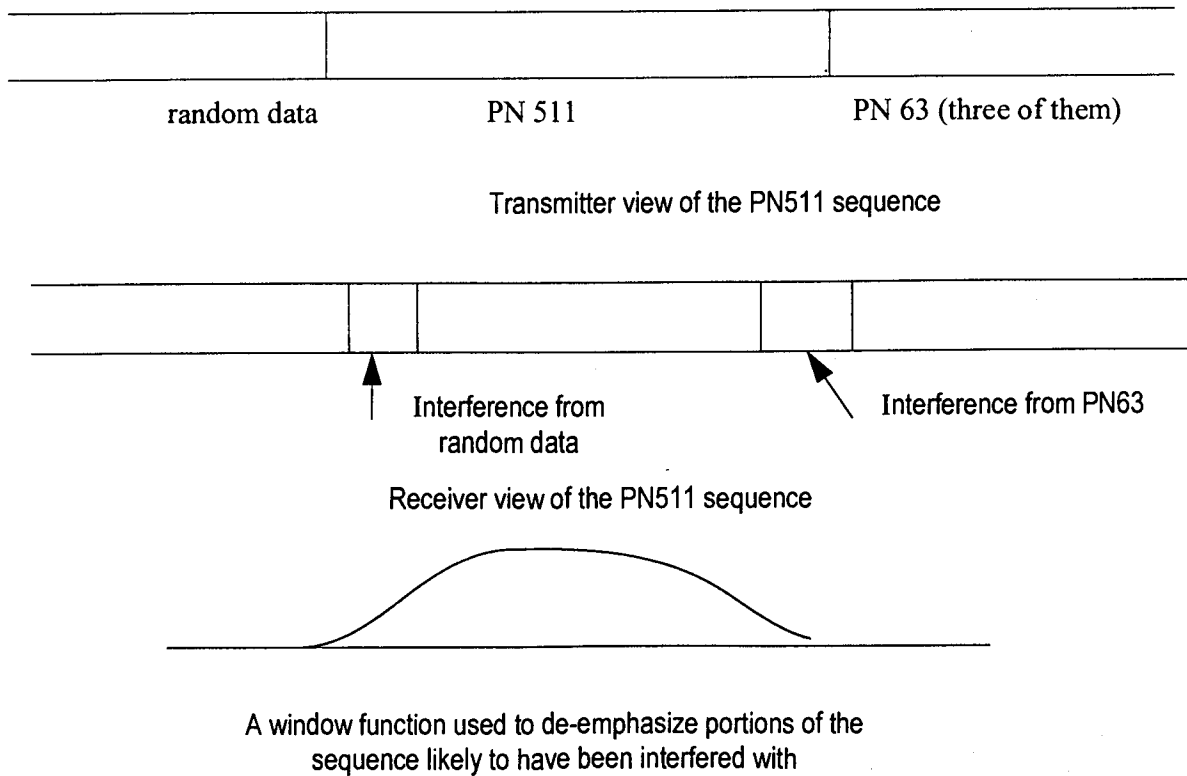
5) The output of the matched filter or its equivalent correlator is a sampled version of the convolution between the physical channel impulse response and the autocorrelation function of the reference baseband signal. When the reference autocorrelation function has low level side-lobes, the matched filter output can be used as an approximation for the channel response.

#### **II.4.3 The Need for Time-Domain Windows**

The 511-bit PN sequence used for analysis purposes is received once every 24.2 milli seconds. Looking back at the transmitter side, this sequence is preceded by random data and is followed by three PN sequences each of length 63 bits as shown in figure. The transmitter filters and other band-limiting electronics, in both the transmitter and receiver, will create interference from the unwanted data and PN sequences into the desired 511 bit PN sequence. What this implies is that it is not possible to separate the 511 sequence alone from the recorded data because the beginning and the tail-end of this sequence have been interfered with, even if the physical channel was ideal as shown in figure 17. When the physical channel also suffers from multipath, the extent of interference into the PN511 sequence becomes more significant. The interfering data and PN63 sequences are expected to elevate the noise floor of the estimated channel response.

One approach for dealing with interference is to multiply the PN511 signal on the receiving end (i.e. recorded data) by a time-domain window function designed to emphasize the central portion of the sequence and de-emphasize the ends which are expected to suffer from random interference. The same time-domain window function has to be applied to the baseband reference signal.

Several window functions have been experimented with. The most effective was found to be a raised cosine, time-domain window with a 10% roll-off factor. This was followed by a hamming window. A rectangular window that chops-off the edges of the sequence (10% on each end) was also experimented with but was not as effective as the raised cosine window.



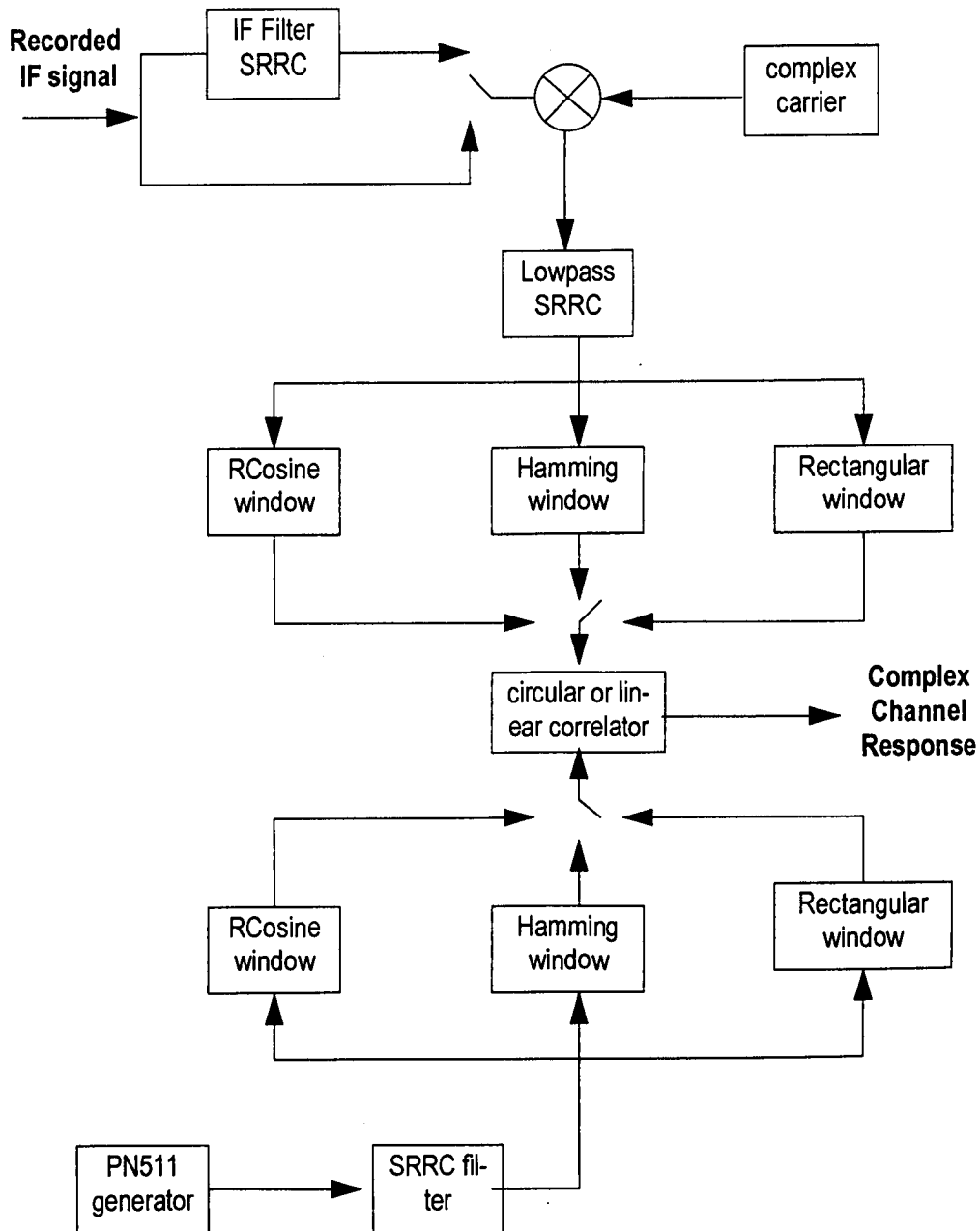
**Fig. 17** Band-limiting, and non-ideal channel cause random data and the PN63 data to interfere with the desired PN511 sequence at the receiving end.

The data analysis program has therefore been designed to incorporate an option for applying time domain window functions. The windows provided are the three types mentioned earlier: a raised cosine, a Hamming window and a rectangular window. The signal processing chain provided for this purpose is shown in figure 18.

With reference to this figure it should be noted that:

- 1) The lowpass filter used for down conversion of the signal is may be a square-root raised cosine FIR filter or a Butterworth IIR filter.
- 2) The reference is best when generated directly at baseband with no further filtering. Generating an IF reference followed by down conversion proved to produce more noisy results as will be shown

later on.



**Fig. 18** Signal Processing Chain and Options Based on a Baseband Reference Signal

3) The same type of time-domain window must be applied to both the down-converted signal and the reference.

4) A circular correlator, implemented using the Fast Fourier Transform (FFT) may be used in place of the linear correlator of figure 15. The circular correlation function was found, empirically, to result in a lower noise floor in the estimated impulse response of the channel in some instances.

## **II.5. Program Description**

The data analysis programs have been developed in Matlab version 5.3 and can of course be run on newer releases of Matlab. The programs require the signal processing tool box. The way the programs are run is described below.

Upon invoking the main program (*atsc\_data\_interact*) the user is prompted to select one of the two main tasks: (a) scanning of a new data file where the location of the PN511 sequence is not known, or (b) to carry out a detailed baseband analysis on a particular instance of the PN511 sequence with a known time location with respect to the beginning of the recorded file.

The user is then prompted to specify the particulars of the recorded data file which include the IF carrier frequency, the sampling frequency of the data recording equipment and the 8VSB data symbol rate.

To scan through a new file, the program starts reading and processing the recorded data in blocks of blocks of 2048 samples each with successive blocks overlapping by 1024 samples. For each 2048 sample block, the data is linearly correlated with the IF reference signal (generated by the matlab function *atsc\_if\_ref\_circ*). The correlator output is processed to find its peak-to-average ratio. When the data block is made mostly of random data the peak-to-average ratio is low (closer to 0 dB). If the data block contains a PN511 sequence, the peak-to-average ratio is large (10 dB or more).

When a large peak to average ratio is encountered the program displays the correlator output, records the index of the data block and pauses and waits for a carriage return to resume.

When the program has finished processing the first 24.2 milliseconds of the recorded file, it identifies the index of the block that yielded a channel response peak closer to the time-center of the correlator output. This segment is then processed further after down conversion to baseband. For detailed



baseband analysis the user is prompted to select the type of baseband filtering and also type of time-domain window in one case.

A modified version of the main program called “atsc\_data\_batch” is designed to carry out baseband analysis to produce several impulse responses at once. This function can aid in exploring the temporal variations of the impulse response. The resulting responses are also averaged on a power basis. The averaging is helpful in the reduction of the noise floor for the estimated response. It should be noted that the averaging is beneficial only when the channel is quasi-static, and the received signal-to-noise ratio is relatively low.

## II.6. Example Program Runs

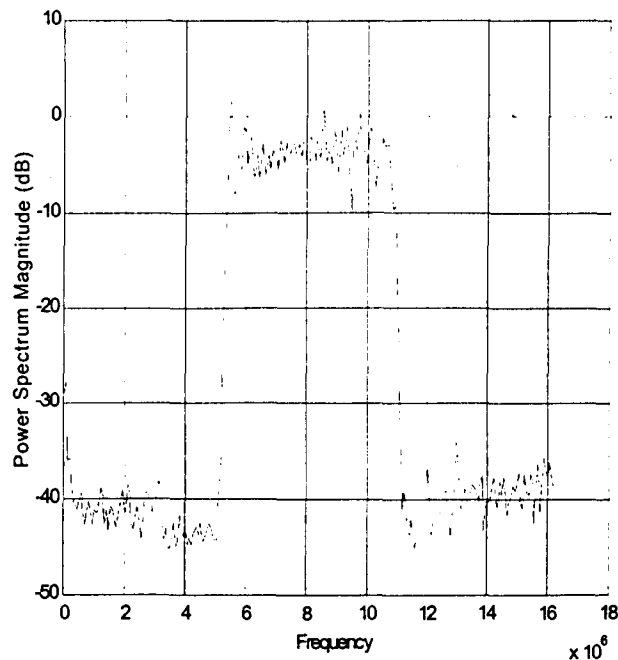
1) Here we will examine a laboratory recording obtained using the set-up of figure 13. The file to be examined corresponded to a channel with two relatively strong delayed multipath components which are -10 dB and -5 dB relative to the main cursor of the channel. The multipath delays were 5  $\mu$ s and 10  $\mu$ s respectively, relative to the main cursor. The file has been previously scanned and we know that the first instance of the PN511 sequence is in data block # 58. The program inputs were as shown below:

```
[a,ref_bb2,t]=atsc_data_interact;
type name of data file====> atsc_8125_1s_m6
sampling frequency in MHz====> 32.5
IF carrier frequency in MHz====> 8.125
Baud Rate in Ms/s====> 10.762
what do you wish to do: scan-file=s, examine a block=b> b
segment index to examine====> 58
frame index====> 2
apply IF filtering? yes=y, no=n > n
signal/reference filtering? IIR_IIR,SRRC_SRRC,SRRC_NONE====> SRRC_NONE
window type: hamming,hanning,rcosine,rectang====> rcosine
```

Here the program returns the raw-data block (no filtering is applied to the raw-data) and the baseband analytically generated reference signal in addition to graphical displays of the estimated channel response based on the selected filter and window functions.

Using the raw-data block we can calculate and plot the power spectral density of the recorded signal. The power spectral density should indicate, in qualitative terms, the level of noise that has been added to the signal.

Figure 19 shows the PSD obtained in this example. The figure suggests that the noise level is about 35 dB below the signal level. The PSD exhibits nulls which are caused by the channel multipath components.

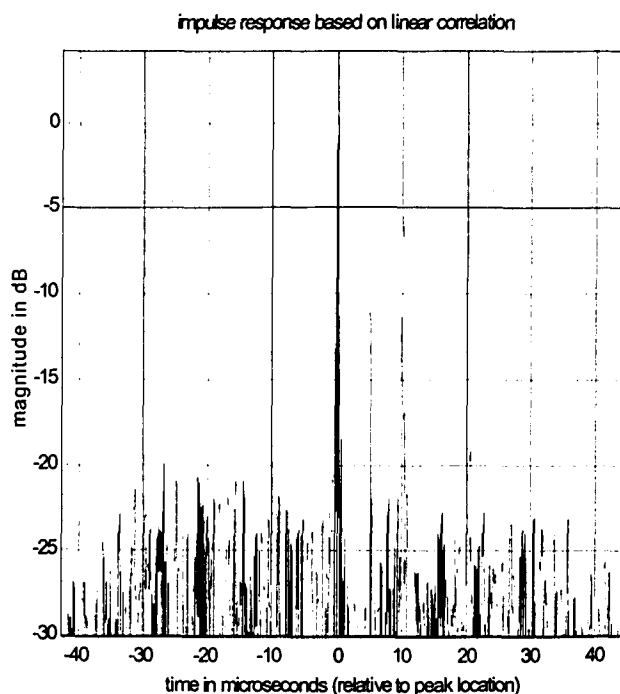


**Fig. 19** Power spectral density of a data block taken from a laboratory recorded signal after passing through a multipath channel

The estimated impulse response using a linear correlator is shown in figure 20. This result indicates the presence of two strong multipath components which are 5 and 10 microseconds away from the carrier. The stronger multipath is 5 dB below the carrier. The weaker component is about 11 dB

below the carrier; this level is about 1 dB off from what was programmed into the channel simulator. This difference can be caused by a number of reasons:

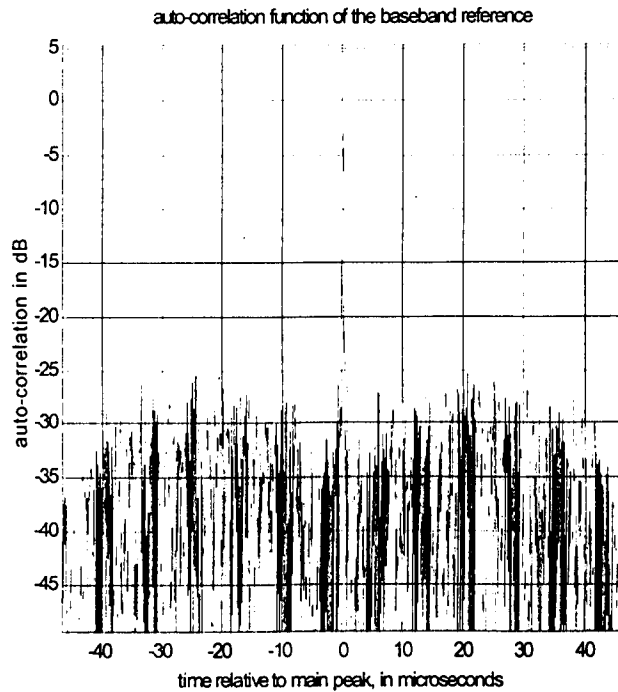
- a) The estimated response is in reality a convolution of the physical channel with the autocorrelation function of the reference signal. The result would be more accurate had it not been for the bandwidth limitations of the hardware used.
- b) The result is affected by the presence of thermal and quantization noise in addition to frequency errors.
- c) It is possible to obtain a more accurate result by averaging several channel responses. This would work only if the channel is not time varying.



**Fig. 20** Impulse Response based on a data block taken from a laboratory recorded signal after passing through a multipath channel

The “noise-like” floor appearing in the estimated response is due mainly to the sidelobes of the autocorrelation function of the PN511 sequence after it went through the 8VSB modulator. Other t

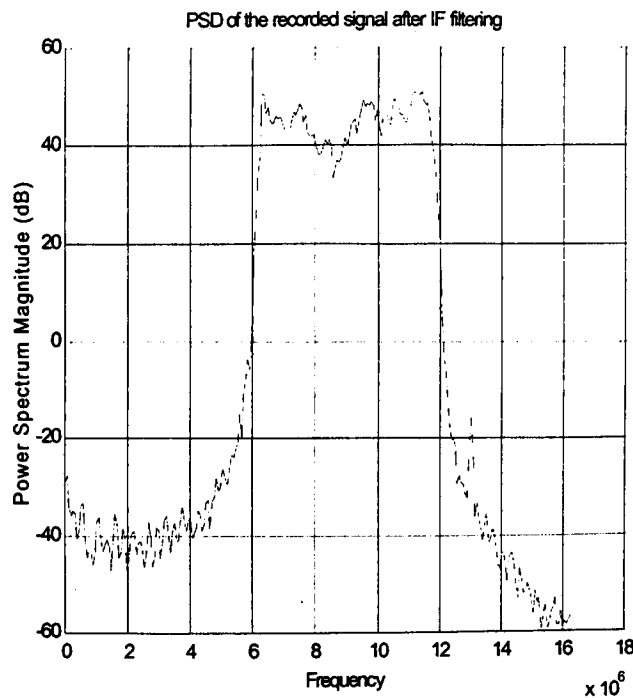
reasons include noise, quantization noise and data interference into the PN511 signal. To validate this point, figure 21 shows the auto-correlation function of an ideal baseband reference signal which exhibits a noise-like floor about 25 dB below its main peak.



**Fig. 21** Auto-correlation Function of the Baseband Reference

As another example we ran the program to analyze an off-air recording as shown below.

```
[a,ref_bb2,t]=atasc_data_interact;  
type name of data file====> off_air4  
sampling frequency in MHz====> 32.5  
IF carrier frequency in MHz====> 9  
Baud Rate in Ms/s====> 10.762  
what do you wish to do: scan-file=s, examine a block=b> b  
segment index to examine====> 50  
frame index====> 1  
apply IF filtering? yes=y, no=n > y  
signal/reference filtering? IIR_IIR,SRRC_SRRC,SRRC_NONE====> SRRC_NONE  
window type: hamming,hanning,rcosine,rectang====> rcosine
```



**Fig. 22** Power spectral density for an off-air recording after IF filtering. An Ideal SRRC/8VSB filter is used.

The power spectral density obtained from the raw data (no IF filtering) appeared slightly asymmetric and showed a noise floor about 30 dB below the signal level. In the current example the IF filtering option in the program is used.

The filtered signal PSD is shown as figure 22. The estimated impulse response of the channel is shown in figure 23. It exhibits strong multipath spreading in a 2 microseconds range from the channel main cursor. It is this multipath that created the spectral nulls apparent in figure 22.

The frequency response of the IF 8VSB filter, with a square-root raised cosine roll-off, is shown in figure 24. Figures 25 and 26 show the impulse response obtained using the remaining options for signal/reference filtering. Examination of these two figures by comparison to figure 20 or 23 one reaches the conclusion that a reference generated directly at baseband with no further filtering yields the best performance in terms of noise-floor at the correlator output.

Figure 27 is identical to 26 except for the type of window where a Hamming window is used instead of raised cosine. It appears that a Hamming window yields a higher noise floor closer to the main cursor. On the plus side, the noise floor is reduced as we get farther from the main cursor.

An example of the results generated by the function "atsc\_data\_batch" is shown as figures 28 and 29. The program has been used to estimate the first four impulse responses in the file "Off\_air4"

```
>>[a,ref_bb2,t,hce]=atsc_data_batch('Off_air4',32.5,9,10.762,50,[1:4]);  
>>apply IF filtering? yes=y, no=n > n  
>>signal/reference filtering? IIR_IIR,SRRC_SRRC,SRRC_NONE==> SRRC_NONE  
>>window type: hamming,hanning,rcosine,rectangular,none==> rcosine
```

Figure 28 shows all four impulse responses which appear to change slightly with the field index. The averaged response is shown in figure 29 and exhibits a noise floor that is a few dB below the noise floors shown in figure 28.

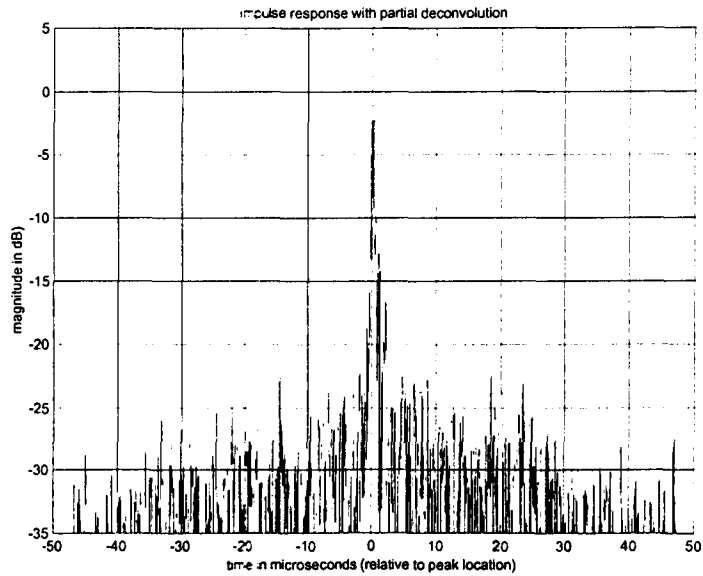


Fig. 23 Estimated channel response from an off-air recording

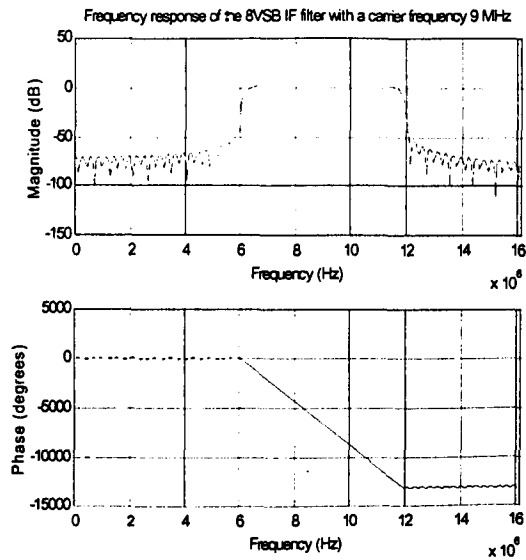
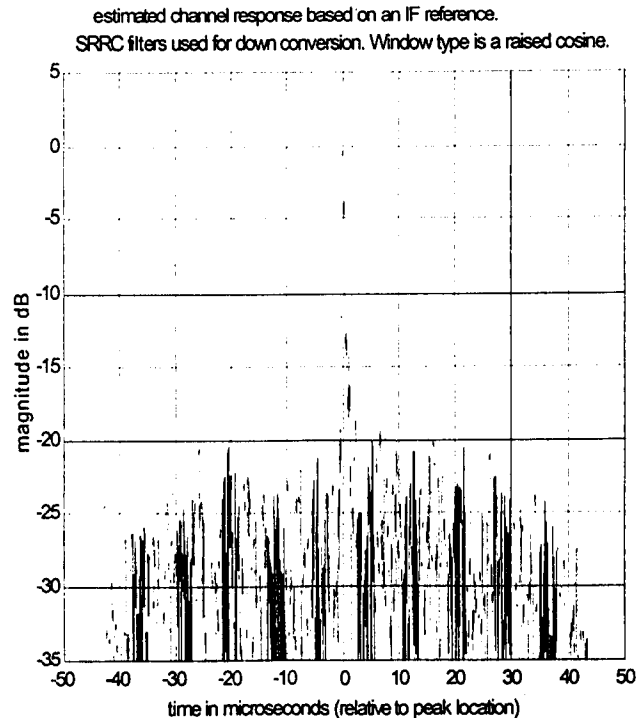


Fig. 24 Frequency response of an IF/8VSB filter, center frequency = 9 MHz



**Fig. 25** Impulse response from an off-air signal (off\_air4) obtained using an IF reference, with SRRC filters after down conversion.

```

atasc_data_interact;

type name of data file==> off_air4
sampling frequency in MHz==> 32.5
IF carrier frequency in MHz==> 9
Baud Rate in Ms/s==> 10.762
what do you wish to do: scan-file=s, examine a block=b> b
segment index to examine==> 50
frame index==> 1
apply IF filtering? yes=y, no=n > n
signal/reference filtering? IIR_IIR,SRRC_SRRC,SRRC_NONE==> SRRC_SRRC
window type: hamming,hanning,rcosine,rectang==> rcosine

```



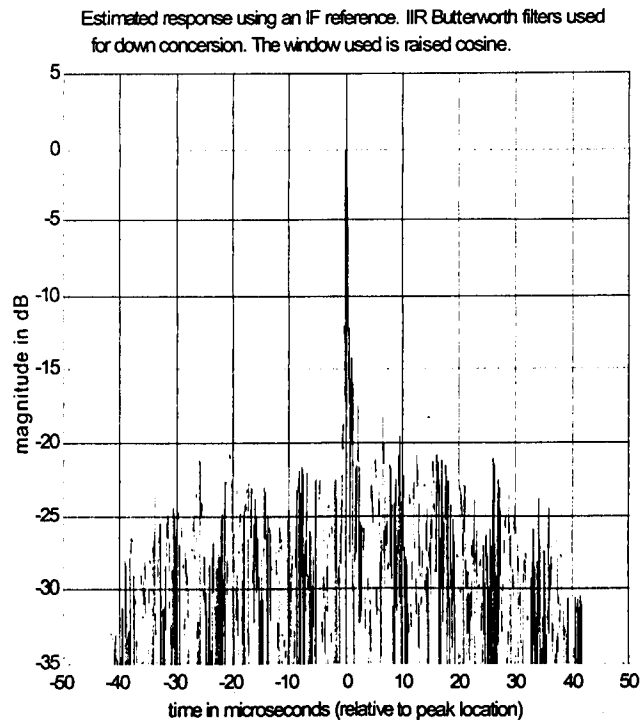


Fig. 26 Impulse response from an off-air signal (off\_air4) obtained using an IF reference, with Butterworth IIR filters after down conversion.

atsc\_data\_interact;

type name of data file====> off\_air4

sampling frequency in MHz====> 32.5

IF carrier frequency in MHz====> 9

Baud Rate in Ms/s====> 10.762

what do you wish to do: scan-file=s, examine a block=b> b

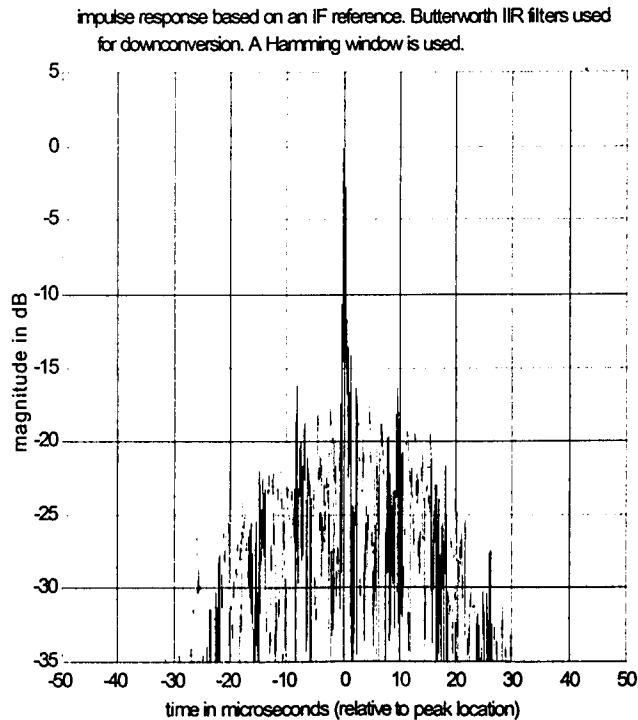
segment index to examine====> 50

frame index====> 1

apply IF filtering? yes=y, no=n > n

signal/reference filtering? IIR\_IIR,SRRC\_SRRC,SRRC\_NONE====> IIR\_IIR

window type: hamming,hanning,rcosine,rectang====> rcosine



**Fig. 27** Impulse response from an off-air signal (off\_air4) obtained using an IF reference, with SRRC filters after down conversion.

atsc\_data\_interact;

type name of data file====> off\_air4

sampling frequency in MHz====> 32.5

IF carrier frequency in MHz====> 9

Baud Rate in Ms/s====> 10.762

what do you wish to do: scan-file=s, examine a block=b> b

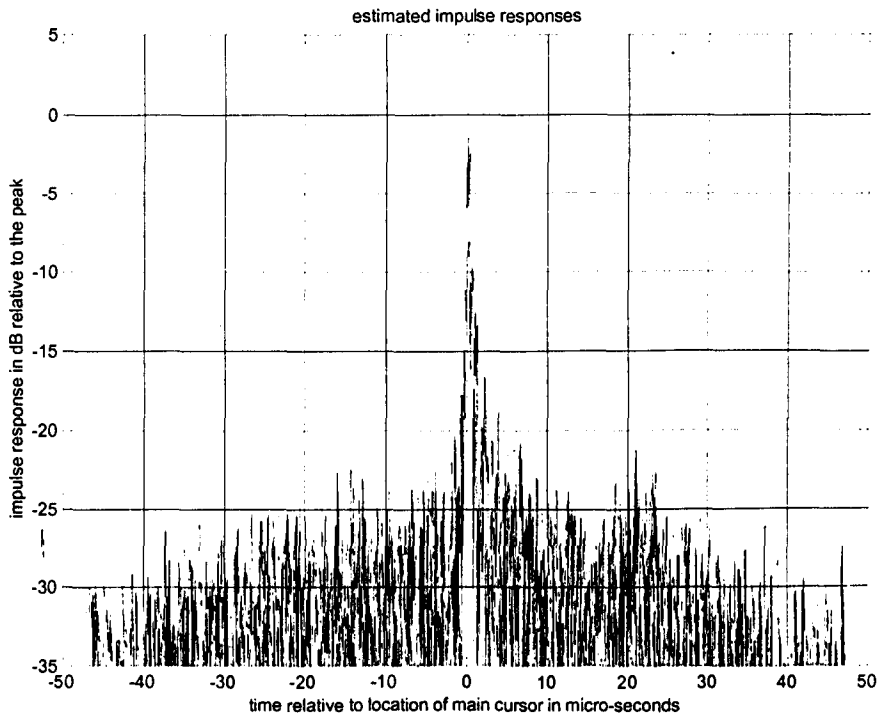
segment index to examine====> 50

frame index====> 1

apply IF filtering? yes=y, no=n > n

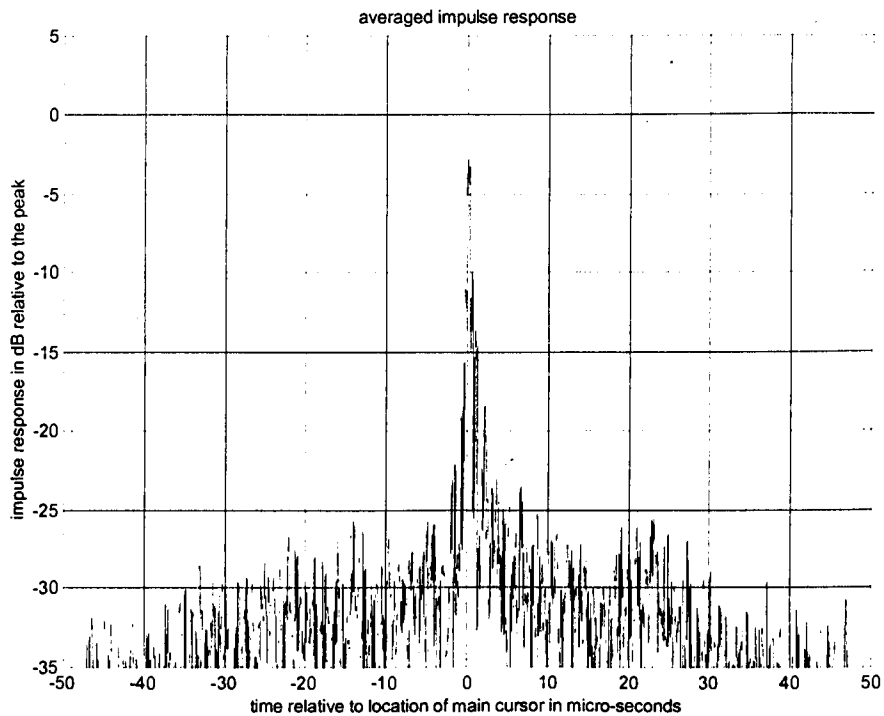
signal/reference filtering? IIR\_IIR,SRRC\_SRRC,SRRC\_NONE====> IIR\_IIR

window type: hamming,hanning,rcosine,rectang====> hamming



**Fig. 28** Four consecutive Impulse responses from an off-air signal (off\_air4) .

```
>>[a,ref_bb2,t,hce]=atasc_data_batch('Off_air4',32.5,9,10.762,50,[1:4]);
>>apply IF filtering? yes=y, no=n > n
>>signal/reference filtering? IIR_IIR,SRRC_SRRC,SRRC_NONE==> SRRC_NONE
>>>window type: hamming,hanning,rcosine,rectangular,none==> rcosine
```



**Fig. 29** Averaged impulse response based on the first four consecutive Impulse responses from an off-air signal (off\_air4) .

### **III Channel Estimation Programs**

### III.1. Main Program

#### **%MAIN PROGRAM**

%To use this program the folder that contains the data-files must be included  
%in the Matlab search path.

%

#### **%PROGRAM DESCRIPTION**

%Function of the program is to determine the position (time-wise) of the  
%firstPN511 sequence in a data file (to be specified by the user). If this time  
%position is known beforehand, the program is used to carry out detailed baseband  
%analysis of data segments to be identified by the user of the program.  
%The program can be used in one of two modes: a scanning mode and a detailed  
%baseband analysis mode as discussed below.

%

%

#### **%SCANNING MODE**

%

%(1)The program reads data from first 24.2 ms interval of the specified file in  
%blocks. The block size is 2048 samples (2 bytes each).The blocks are read  
%sequentially. Successive blocks have a 50% overlap.

%

%(2) Once a block is correctly transformed from binary to decimal integers, the  
%data block is correlated with a computed reference waveform.The correlation is  
%done on the data at IF (no down conversion). This is used during data scanning  
%because of processing simplicity.

%

%(3) To determine if the correlation is valid, the program compares the  
%correlation peak to average ratio to a preset threshold (determined empirically);  
%if the test is pass the program considers the correlation an approximation to the  
%impulse response. The result is displayed graphically together with the segment  
%index (i.e. data block index). There may be more than one hit for a given 24.2  
%m.s. interval.

%

%(4) Once the scan is completed, the user is prompted for the block index thought  
%most useful. The program revisits that segment for a more thorough investigation.  
%The selected index is then logged against the name of the data file

%

#### **%BLOCK ANALYSIS MODE (BASEBAND)**

%The baseband analysis includes several correlation options

%

%1)The data and reference are down-converted to baseband. The BB signal is  
%filtered using a Butterworth IIR and then cross-correlated with the baseband  
%reference.

%

%2)The data and reference are down-converted, filtered using SRRC filters that  
%are approximately matched to the Tx filter, and a cross-correlation is again  
%performed.

```

%
33)The data signal is down-converted and filtered as in option 2 above. The
%reference signal is generated directly at base-band. A cross-correlation followed
%by a partial-deconvolution operations are used to estimate the channel impulse
%response.
%
%Each of the three options yields a representation for the impulse response only
%if the data block contains most if not all of a PN511 sequence.
%
%REMARK: During the scan phase, the program deals exclusively with IF signals.
%The base-band processing is carried only when a segment index is selected for
%detailed analysis
%
%MAIN VARIABLES: INPUTS
%
%   file_data: The name of the recorded data file
%
%   f_sample: The sampling rate of the data recording system in MHz
%
%   f_carrier: The IF carrier frequency as seen by the data recording
%
%   system in MHz
%
%   f_baud: The symbol rate of the 8VSB system
%
%   selection: A character variable that determine whether the program
%
%   is to scan a new data
%
%   file to find out the time location of the first occurrence of the
%
%   PN511 sequence, or to perform detailed baseband analysis on a
%
%   particular segment of the file.
%
%   izoom: Index of the data segment to be analysed in detail, relative
%
%   to the beginning of the data field the sync segment is part of.
%
%
%   Iframe: This is the index of the data field that contains the data
%
%   segment to be analyzed.
%
%   ifreq: This is a character variable that tells the program whether
%
%   to apply IF filtering to the recorded data segment.
%
%   bb_filter: This is a character variable that indicates the baseband
%
%   filtering options for both the recorded signal and the computed
%
%   reference.
%
%
%MAIN VARIABLES: OUTPUTS
%
%   ref_bb2: This is the computed baseband reference obtained by passing
%
%   the PN511 sequence through an 8VSB equivalent lowpass filter.
%
%   reference: This is the computed IF reference, obtained by applying
%
%   the PN511 sequence to and 8VSB modulator utilizing the same carrier,
%
%   symbol rate and sampling frequency as the recorded signal.
%
%   W, w2, w3 and w4: These are the estimated impulse responses subject
%
%   to different types of filtering as indicated by the user defined
%
%   selections. Not all are present at the same time.
%
%   time_axis: This is the time scale for the final impulse response
%
%   (w4) which is obtained using a baseband
%
%   reference, circular correlation+time windowing.
%

```

```

%SYNTAX
%[a,ref_bb2,t,hce]=atasc_data_interact
%      a is the data block being analyzed, ref_bb2 is the computed baseband
%      reference, t is the time axis for the estimated channel response,
%      and hce is the impulse response of the channel (normalized to the
%      main cursor).
%
%RELATED PROGRAMS/ FUNCTIONS
%atasc_field_sync, binary_to_float,IF_Filtering, BB_IIR_SRRRC, BB_SRRRC_SRRRC,
%BB_SRRRC_REC
%
%fs=32.286713e6 for new data/ fs=32.5e6 for Takada's
%fc=9.0e6 for new data/ fc=8.125e6 for Takada's
%[a,ref_bb2,t,hce]=atasc_data_interact(file_data,f_sample,f_carrier,f_baud,
%selection,izoom,Iframe,ifreq)
%
function [a,ref_bb2,t,hce]=atasc_data_interact
%
clear all
close all
file_data=input('type name of data file==> ','s');
f_sample=input('sampling frequency in MHz==> ');
f_carrier=input('IF carrier frequency in MHz==> ');
f_baud=input('Baud Rate in Ms/s==> ');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%open data file containing measurement results
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[fid, message] = fopen(file_data,'r');
%if fopen attempt fails propmt user for a valid file name; limit number of
%re-tries to only 3

ifd=0;
while fid==-1
    fclose all
    fprintf(1,'%30s\n',[file_data ' not a valid file name'])
    file_data=input('type name of data file==> ','s');
    [fid, message] = fopen(file_data,'r');
    ifd=ifd+1;
    if ifd>=3
        break
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%generate a modulated carrier using the PN511 reference signal waveform
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
```



```

[ref_bb2,reference]=atsc_field_sync(f_sample,f_baud,f_carrier); %returns
%pn511 reference
%ref_bb2 is a baseband, unfiltered reference. reference: this is an IF version
ar=reference-mean(reference);
ar=rot90(ar,-1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%open log file with an "append" option
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[fid2 message]=fopen('scanfile1.txt','a');
fprintf(fid2,'%20s %20s\n\n','data file name = ',file_data);

%start processing data
a=zeros(4096,1); a_temp=zeros(2048,1); b_temp=zeros(2048,1);
selection=input('what do you wish to do: scan-file=s, examine a block=b> ','s');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%scan data file to identify data segments that contain a PN511 sequence
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

KMAX=(f_sample*1e6)*(24.2e-3)/(2^11)+2;%KMAX is number of segments in one
%frame (approximately 24.2 m.s.)

% Read first segment and store it in b_temp

x1=fread(fid,2^11,'ubit16');
[b_temp]=binary_to_float(x1); %convert binary file to floating point

seg_idxx=[]; %vector showing indices of segments containing the PN511
pk_loc=[]; %vector showing position of the correlation peak if one exists
pk_loc_offset=[]; %vector showing peak location relative to centre

if selection=='s'
for k=2:KMAX

%Read & interpret binary file data
x1=fread(fid,2^11,'ubit16');
[a_temp]=binary_to_float(x1); %convert binary file to floating point

%store data block in array a
a(2049:4096,1)=a_temp; a(1:2048,1)=b_temp;

%convolve data block with computed reference
w=conv(a-mean(a),flipud(ar-mean(ar)))/2048^2;

```

```

%detect the peak-to-average ratio of the convolution result
[ww, LP]=max(abs(w));
ww=ww/mean(abs(w));
threshold=0.0001*mean(abs(w));
w=10*log10(abs(w)+threshold);%threshold added to avoid taking the log of zero
%if the peak-to-average is large enough then record the iteration number
if ww>=10
    seg_indx=[seg_indx k]; %store segment index
    pk_loc=[pk_loc LP]; %store peak location
    pk_loc_offset=[pk_loc_offset LP-length(w)/2]; %store peak offset

    ref=a;
    plot(w)
    iteration=k
    fprintf(fid2,'%6.2f\n',iteration);
    fprintf(1,'<hit return to continue>');
    fprintf(1,'%\n\n')
    pause
else

end

b_temp=a_temp;

end
else
end

if selection=='s'
    fprintf(fid2,'sg_index=')
    fprintf(fid2,'2%6.2f\n',seg_indx)
    fprintf(fid2,'peak offsets=')
    fprintf(fid2,'2%6.2f\n',pk_loc_offset)
    [Loff_min,Loff_indx]=min(abs(pk_loc_offset));
    izoom=seg_indx(Loff_indx);
    Iframe=1;
else
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Pick a data segment (4096 samples) for detailed processing
%%%
if selection=='s'
    izoom=input('segment index to examine==> ');
    Iframe=input('frame index====> ');
else
    izoom=seg_indx(Loff_indx);

```

```

    Iframe=1;
end

fs=f_sample*1e6;
frame_offset=Iframe*floor(832*313*286/4.5/684e6*fs)*16;
%Note:The frame period=832*313*286/4.5/684e6 seconds. This
%is the interval between field sync segments

fprintf(fid2,'%20s %6.2f\n\n\n','selected index = ',izoom);
fprintf(fid2,'%20s %6.2f\n\n\n','selected frame index = ',Iframe);

frewind(fid)

%read the data segment selected

bits_to_skip=16*(izoom-2)*(2^11)+frame_offset;
x00=fread(fid,1,'ubit16',bits_to_skip); %this is a dummy read

x1=fread(fid,2^12,'ubit16');
[a]=binary_to_float(x1); %convert binary file to floating point

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%filter the IF signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ifreq=input('apply IF filtering? yes=y, no=n > ','s');
bb_filter=input('signal/reference filtering? IIR_IIR,SRRC_SRRC,SRRC_NONE==>
','s');

if ifreq=='y'
    [a]=IF_Filtering(f_sample,f_baud,f_carrier,a);
else
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%convolve with IF reference waveform and plot estimated response
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
w=conv(a-mean(a),flipud(ar-mean(ar)))/2048^2;
figure
plot(abs(w))
title('impulse response magnitude based on IF signals')
ylabel('magnitude (linear scale)')
xlabel('sample index')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%baseband processing
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

switch upper(bb_filter)
%CASE I: USES IIR LOWPASS FILTERS
case('IIR_IIR')
    [time_axis,wdB]=BB_IIR_SRRRC(f_sample,f_baud,f_carrier,ar,a);
    t=time_axis; hce=wdB;
%CASE II: USES SRRRC LOWPASS FIR FILTERS
case('SRRRC_SRRRC')
    [time_axis,wdB]=BB_SRRRC_SRRRC(f_sample,f_baud,f_carrier,ar,a);
    t=time_axis; hce=wdB;
%CASE III: try unfiltered reference
case('SRRRC_NONE')
    s1=ref_bb2;          % unfiltered PN511 sequence

[time_axis1,wdB,time_axis2,hcdB]=BB_SRRRC_REC(f_sample,f_baud,f_carrier,a,s1);
    t=time_axis2;hce=hcdB;
otherwise
    disp('unknown selection')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
ST=fclose(fid2);

```

## III.2. Field Sync Calculator

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %
%This function generates a modulated carrier based on the atsc PN511 reference
%pseudo random sequence that constitutes part of the field sync signal.
%The modulated carrier is filtered using a SRRC filter with a bandwidth and
%roll-off factor as specified in the standard's documents. The sampling rate
%and carrier frequencies are those used for data recordings. A baseband version of
%the field sync signal is also generated by this function.
%
%SYNTAX
%[ref_bb2,reference]=atsc_field_sync(f_sample,f_baud,f_carrier)
%
%ref_bb2,reference are the computed baseband and IF versions of the field-sync
%signal affected by the PN511 sequence.
%f_sample,f_baud,f_carrier are the sampling rate, 8VSB symbol rate and IF
%carrier frequency all expressed in MHz
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [ref_bb2,reference]=atsc_field_sync(f_sample,f_baud,f_carrier)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%generate PN511 sequence

A=[1 0 1 1 0 1 1 0 1 1];%feedback connections
PI=[0 0 0 0 0 0 0 1 0];%initial conditions

for k=1:511
    x=PI.*A(1:9);
    y=floor((sum(x)+.01)/2)*2;
    if y==sum(x)
        sb(k)=0;
    else
        sb(k)=1;
    end
    PI=[sb(k) PI(1:8)];
end
sb=fliplr(sb);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%make an oversampled, unfiltered baseband waveform. At this point
%we use OSR samples per symbol period where OSR is an integer
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
OSR=floor(f_sample/f_baud+.01); %number of samples per symbol(to nearest
%integer)
s1=zeros(1,511*OSR);
s1(1,1:OSR:511*OSR)=sign(sb-0.5);
%
%design a lowpass SRRC filter
OS_RATIO=floor(f_sample/f_baud+.011);
```

```

baud=f_baud*1e6;
[NUM, DEN] = rcosine(baud/2, baud/2*OS_RATIO, 'sqrt', .62/5.38,10);%SRRC FILTER

ar1=conv(NUM,s1); %filter reference
ar2=conv(NUM,ar1); ar2=ar2-mean(ar2); %
ref_bb=ar2(length(NUM):length(s1)+length(NUM)+1);%ref_bb now contains a full
%RC filter

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Set parameters using exactly OSR samples per symbol
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fs=f_sample*1e6; fc=f_carrier*1e6;
baud=f_baud*1e6;
f0=fc-baud/4;
tsif_nominal=1/(f_baud*1e6*OSR);%sampling interval assuming OSR samples
% per symbol (this is not necessarily equal
% to 1/f_sample

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Upconvert and filter reference, still assuming OSR samples per symbol
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f0=fc-baud/4;
[Bif,Aif]=rcosine(baud/2,baud/2*2*OSR,'sqrt',.62/5.38,10);%%
K=length(Bif)-1;
caif=exp(j*2*pi*(baud/4)*tsif_nominal*[-K/2:K/2]);% %%%
bif=caif.*Bif; % complex VSB filter @ baseband
figure
freqz(bif,Aif,256,f_baud*OSR*1e6)

reference=conv(bif,s1);%%%%
carrier=exp(j*2*pi*f0*tsif_nominal*[1:length(reference)]);%%%%
reference=real(reference.*carrier);%%%%
reference=reference/(2*max(reference))*(2^16);
figure
psd(reference,2048,f_baud*OSR*1e6)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Interpolate the IF reference signal in conformance with the sampling rate
%used for data recordings, f_sample
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tsif_measured=1/(f_sample*1e6);
time_nominal=[0:1:length(reference)-1]*tsif_nominal;
time_desired=[0:1:length(reference)-1]*tsif_measured;
reference2=interp1(time_nominal,reference,time_desired,'*linear');
%figure
%psd(reference2,2048,f_sample*1e6)

reference=reference2;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Interpolate the baseband reference in conformance with the sampling rate
%used for data recordings, f_sample
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tsif_measured=1/(f_sample*1e6);
time_nominal=[0:1:length(ref_bb)-1]*tsif_nominal;
time_desired=[0:1:length(ref_bb)-1]*tsif_measured;

ref_bb2=interp1(time_nominal,ref_bb,time_desired,'*nearest');

```

### III.3. Correlation with a down-converted reference/IIR Filtering

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Estimate channel impulse response from measured data. The estimation is based on:
%
%1- Generation of a complex carrier
%
%2-Down conversion to base-band of BOTH signal and reference.The downconverted
%signal and reference are processed using a lowpass filter implemented as a 8th-order
%Butterworth.
%
%3- The baseband signal and reference are subjected to a time-domain window
%
%4-The signal is then passed through a baseband filter with impulse response
%matched to the down-converted and filtered reference which corresponds to the
%PN511 portion of the field-sync signal
%
%The function displays the output of the matched filter (on a dB scale, relative
%to the peak)versus time measured from the position of the main cursor of the impulse
%response.
%
%SYNTAX
%[time_axis,wdB]=BB_IIR_SRRC(f_sample,f_baud,f_carrier,ar,a)
%
%OUTPUT VARIABLES: time_axis and wdB describe the impulse response estimate
%provided at the output of the matched filter.
%
%INPUT VARIABLES: f_sample,f_baud and f_carrier are the sampling rate, 8VSB
%symbol rate and the IF carrier frequency all expressed in MHz. ar is a Matlab
%computed reference signal according to the ATSC standard description for the
%field sync, and a is the measured data block.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [time_axis,wdB]=BB_IIR_SRRC(f_sample,f_baud,f_carrier,ar,a)
fc=f_carrier*1e6; baud=f_baud*1e6; bw=baud/2; f0=fc-bw/2;
K=length(ar);
fs=f_sample*1e6; ts=1/fs; ca=exp(-j*2*pi*f0*ts*rot90([1:1:K],-1));

%downconvert reference
[B,A]=butter(8,8/16.25); %lowpass IIR filter
ar1=ar.*ca; ar2=filter(B,A,ar1); ar2=ar2-mean(ar2);
ar2=ar2/2048;

%downconvert signal
K2=length(a);
```



```

ca=exp(-j*2*pi*f0*ts*rot90([1:1:K2],-1));
a1=a.*ca; a2=filter(B,A,a1); a2=a2-mean(a2);
a2=a2/2048;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Establish time reference

%convolve baseband signal and baseband reference
w2=conv(conj(a2),flipud(ar2)); w2=w2/max(abs(w2))*511;

figure
plot(abs(w2))
title('impulse response magnitude based on BB signals/ BW-IIR filter')
ylabel('magnitude (linear scale)')
xlabel('sample index')

w2=conv(conj(a2),flipud(ar2));
[wmax,Lw]=max(abs(w2));
a22=a2(Lw-length(ar2):Lw+1); %%%

%Apply time domain window
%
wintype=input('window type: hamming,hanning,rcosine,rectangular,none==> ','s');
%
Lwin=length(a22);
beta=floor(Lwin/8);

switch lower(wintype)
case('hamming')
    a22=a22.*hamming(Lwin);
    ar2=ar2.*hamming(length(ar2));
case('hanning')
    a22=a22.*hanning(Lwin);
    ar2=ar2.*hanning(length(ar2));
case('rcosine')
    a22=a22.*rcwin(Lwin,beta);
    ar2=ar2.*rcwin(length(ar2),floor(length(ar2)/8));

case('rectangular')
    a22=a22.*recwin(Lwin,beta);
    ar2=ar2.*recwin(length(ar2),floor(length(ar2)/8));
case('none')
    a22=a22.*ones(length(a22),1);
    ar2=ar2.*ones(length(ar2),1);

```

```

otherwise
    disp('unknown window selection')
end

%*****
%*****
w2=conv(conj(a22),flipud(ar2));
w2=w2/max(abs(w2))*511;
%set the time axis
[wmax,Lw]=max(abs(w2));
time_axis=[-Lw+1:1:length(w2)-Lw]/(f_sample);%time axis in microseconds
figure
plot(time_axis,abs(w2))%plot response on a linear scale
title('impulse response based on BB signals/ BW-IIR filter')
ylabel('magnitude (linear scale)')
xlabel('time in microseconds (relative to peak location)')

threshold=0.0001*mean(abs(w2));
wdB=20*log10(abs(w2)+threshold);%threshold added to avoid taking the log of
%zero
wdB=wdB-max(wdB);
figure
plot(time_axis,wdB)%plot response on a dB scale
title('impulse response based on BB signals/ BW-IIR filter')
ylabel('magnitude in dB')
xlabel('time in microseconds (relative to peak location)')
axis([-50 50 -35 5]);
grid on
zoom on

```

### III.4. Correlation with a down-converted reference/SRRT Filtering

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%Estimate channel impulse response from measured data. The estimation is based
%on:
%
%1- Generation of a complex carrier
%2- Down conversion to base-band of BOTH signal and reference. The downconverted
signal and reference are processed using a lowpass filter implemented as an FIR
with a square-root raised cosine response with a roll- off factor of 11.5 percent.
%3- The baseband signal and reference are subjected to a time-domain window
%4-The signal is then passed through a baseband filter with impulse response
%matched to the down-converted and filtered reference which corresponds to the
%PN511 portion of the field-sync signal
%
%The function displays the output of the matched filter (on a dB scale, relative
to the peak) versus time measured from the position of the main cursor of the impulse
response.
%
%SYNTAX:
%[time_axis,wdB]=BB_SRRC_SRRC(f_sample,f_baud,f_carrier,ar,a)
%
%OUTPUT VARIABLES: time_axis and wdB describe the impulse response estimate
%provided at the output of the matched filter.
%
%INPUT VARIABLES f_sample,f_baud and f_carrier are the sampling rate, 8VSB
%symbol rate and the IF carrier frequency all expressed in MHz. ar is a Matlab
%computed reference signal according to the ATSC standard description for the
%field sync, and a is the measured data block.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
function [time_axis,wdB]=BB_SRRC_SRRC(f_sample,f_baud,f_carrier,ar,a)
%%%
%multiply reference by a complex carrier
fc=f_carrier*1e6; baud=f_baud*1e6; bw=baud/2; f0=fc-bw/2;
K=length(ar);
fs=f_sample*1e6; ts=1/fs;
ca=exp(-j*2*pi*f0*ts*rot90([1:1:K],-1));
ar1=ar.*ca; %multiply reference by carrier

%multiply signal by a complex carrier
K2=length(a);
ca=exp(-j*2*pi*f0*ts*rot90([1:1:K2],-1));
a1=a.*ca; %multiply signal by carrier
```

```

%%%%
OS_RATIO=floor(f_sample/f_baud+.011);
[NUM, DEN] = rcosine(baud/2, baud/2*OS_RATIO, 'sqrt', .62/5.38,10);%SRRC
%FILTER
[H,F]=freqz(NUM,DEN,256,f_sample*1e6); %Frequency response of SRRC
figure
plot(F,abs(H))
title('frequency response of SRRC filter')

%filter signal and reference
ar2=conv(NUM,ar1); ar2=ar2-mean(ar2); %filter reference
a2=conv(NUM,a1); a2=a2-mean(a2); %filter signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%

%Establish time reference
w3=conv(conj(a2),flipud(ar2)); %convolve BB signal & BB reference
w3=w3/max(abs(w3))*511;

figure
plot(abs(w3))
title('impulse response magnitude based on BB signals/ SRRC filter')
ylabel('magnitude (linear scale)')
xlabel('sample index')

w3=conv(conj(a2),flipud(ar2));
[wmax,Lw]=max(abs(w3));
a22=a2(Lw-length(ar2):Lw+1); %%%

%%Apply window function to reference and signal
%
wintype=input('window type: hamming,hanning,rcosine,rectangular,none==> ','s');
%
Lwin=length(a22);
beta=floor(Lwin/8);

switch lower(wintype)
case('hamming')
    a22=a22.*hamming(Lwin);
    ar2=ar2.*hamming(length(ar2));
case('hanning')
    a22=a22.*hanning(Lwin);
    ar2=ar2.*hanning(length(ar2));

```

```

case('rcosine')
    a22=a22.*rcwin(Lwin,beta);
    ar2=ar2.*rcwin(length(ar2),floor(length(ar2)/8));

case('rectangular')
    a22=a22.*recwin(Lwin,beta);
    ar2=ar2.*recwin(length(ar2),floor(length(ar2)/8));
case('none')
    a22=a22.*ones(length(a22),1);
    ar2=ar2.*ones(length(ar2),1);
otherwise
    disp('unknown window selection')
end
%%%
%%
%%
w3=conv(conj(a22),flipud(ar2));
w3=w3/max(abs(w3))*511;
%set the time axis
[wmax,Lw]=max(abs(w3));
time_axis=[-Lw+1:1:length(w3)-Lw]/(f_sample);%time axis in microseconds
figure
plot(time_axis,abs(w3))%plot response on a linear scale
title('impulse response based on BB signals/ SRRC filter')
ylabel('magnitude (linear scale)')
xlabel('time in microseconds (relative to peak location)')

threshold=0.0001*mean(abs(w3));
wdB=20*log10(abs(w3)+threshold);%threshold added to avoid taking the log of
%zero
wdB=wdB-max(wdB);
figure
plot(time_axis,wdB)%plot response on a dB scale
title('impulse response based on BB signals/ SRRC filter')
ylabel('magnitude in dB')
xlabel('time in microseconds (relative to peak location)')
axis([-50 50 -35 5]);
grid on
zoom on

```

### III.5. Correlation with a Baseband Reference

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Estimate channel impulse response from measured data. The estimation is based
%on:
%1- Generation of a complex carrier
%
%2- Down conversion to base-band
%
%3- The baseband signal is subjected to a time-domain window
%
%4-The signal is then passed through a baseband filter with impulse response
%matched to the PN511 portion of the field-sync signal
%
%5-The filter output is de-convolved with the autocorrelation function of
%the field-sync signal.
%The function displays the outputs of both the matched filter and that of
%the de-convolver.
%
%Syntax:
%[time_axis1,wdB,time_axis2,hcdB]=BB_SRRC_REC(f_sample,f_baud,f_carrier,a,s1)
%
%Output variables: time_axis1 and wdB describe the impulse response estimate
%provided at the output of the matched filter. time_axis2 and hcdB describe the
%impulse response estimate provided at the output of the de-convolver that follows
%the matched filter.
%
%Input variables: f_sample,f_baud and f_carrier are the sampling rate, 8VSB
%symbol rate and the IF carrier frequency all expressed in MHz. a is the
%measured data block and s1 is a Matlab computed baseband reference signal
%according to the ATSC standard description for the field sync.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %

function
[time_axis1,wdB,time_axis2,hcdB]=BB_SRRC_REC(f_sample,f_baud,f_carrier,a,s1)
%
fc=f_carrier*1e6; baud=f_baud*1e6; bw=baud/2; f0=fc-bw/2;
OS_RATIO=floor(f_sample/f_baud+.011);
fs=f_sample*1e6; ts=1/fs;

%downconvert signal
K2=length(a);
ca=exp(-j*2*pi*f0*ts*rot90([1:1:K2],-1));
a1=a.*ca;

[NUM, DEN] = rcosine(baud/2, baud/2*OS_RATIO, 'sqrt', .62/5.38,10);%SRRC
```

```

%FILTER
% [H,F]=freqz(NUM,DEN,256,f_sample*1e6); %Frequency response of SRRC
a2=conv(NUM,a1); a2=a2-mean(a2); %filter signal

%Establish time reference
ar3=rot90(s1,-1);
w4=conv(conj(a2),flipud(ar3));
[wmax,Lw]=max(abs(w4));
a22=a2(Lw-511*OS_RATIO:Lw+1);

%Apply a time domain window to signal and to reference
wintype=input('window type: hamming,hanning,rcosine,rectangular,none==> ','s');
%
Lwin=length(a22);
beta=floor(Lwin/8);

switch lower(wintype)
case('hamming')
    a22=a22.*hamming(Lwin);
    ar3=ar3.*hamming(Lwin);
case('hanning')
    a22=a22.*hanning(Lwin);
    ar3=ar3.*hanning(Lwin);
case('rcosine')
    a22=a22.*rcwin(Lwin,beta);
    ar3=ar3.*rcwin(Lwin,beta);

case('rectangular')
    a22=a22.*recwin(Lwin,beta);
    ar3=ar3.*recwin(Lwin,beta);
case('none')
    a22=a22.*ones(length(a22),1);
    ar3=ar3.*ones(length(ar3),1);

otherwise
    disp('unknown window selection')
end

%Pass signal through a filter matched to the BB reference waveform
w4=conv(conj(a22),flipud(ar3));
w4=w4/max(abs(w4))*511;
%set the time axis
[wmax,Lw]=max(abs(w4));
time_axis=[-Lw+1:1:length(w4)-Lw]/(f_sample);%time axis in microseconds
threshold=0.0001*mean(abs(w4));

```

```

wdB=20*log10(abs(w4)+threshold);%threshold added to avoid taking the log of
%zero
wdB=wdB-max(wdB);

%Plot output of matched filter
figure
plot(time_axis,wdB)%plot response on a dB scale
ylabel('magnitude in dB')
%%    plot(time_axis,abs(w4))%plot response on a linear scale
%%    ylabel('magnitude (linear scale)')
xlabel('time in microseconds (relative to peak location)')
title('impulse response based on baseband reference')
axis([-50 50 -35 5]);
grid on
zoom on
time_axis1=time_axis;

%Deconvolution
w5=conv(conj(a22),flipud(ar3));
W5=fft(w5,length(w5));
AR3=fft(ar3,length(W5));
SIGMA=mean(abs(AR3.^2));
HC=W5./(abs(AR3.^2)+0.1*SIGMA);
% figure
% plot(abs(HC))
dig_bw=10.762*1.115/32.5;
dig_bw=floor(length(W5)*dig_bw);
dig_bw2=floor(dig_bw/2);
WM=[ones(1,dig_bw2) zeros(1,length(W5)-2*dig_bw2) ones(1,dig_bw2)];
HC=HC.*WM;

hc=ifft(HC);
threshold=0.01*mean(abs(hc));
hcdB=20*log10(abs(hc)+threshold); %threshold added to avoid taking the log of
%zero
hcdB=hcdB-max(hcdB);
[hcmax,Lw]=max(hcdB);

%Plot output of de-convolver
time_axis=[-Lw+1:1:length(hcdB)-Lw]/(f_sample);%time axis in microseconds
figure
plot(time_axis,hcdB)%plot response on a dB scale
title('impulse response with partial deconvolution')
ylabel('magnitude in dB')
xlabel('time in microseconds (relative to peak location)')
axis([-50 50 -35 5]);
grid on
zoom on

```



```
time_axis2=time_axis;
```

### III.6. Binary to Decimal Conversion

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This function converts binary data vector x1 into a decimal into decimal
%integers which are returned to vector a_temp.
%
%syntax:
%[a_temp]=binary_to_float(x1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [a_temp]=binary_to_float(x1)
%Read & interpret binary file data
%    x1=fread(fid,2^11,'ubit16');
x1_binary=dec2bin(x1);
x1_b_swap=[x1_binary(:,9:16) x1_binary(:,1:8)];
x1_12bits=x1_b_swap(:,5:16);
s=bin2dec(x1_12bits(:,1));
yp=bin2dec(x1_12bits(:,2:12));
yn=-bitcmp(bin2dec(x1_12bits(:,2:12)),11)-1;
x4=s.*yn+(1-s).*yp;
a_temp=x4-mean(x4);
```

### III.7. IF\_Filtering

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%This function applies an IF filtering DSP operation to a data block. The  
%filter used has a square-root raised cosine response with a bandwidth and  
%roll-off factor as specified by the ATSC digital TV standard. The center  
%frequency and the sampling rate are those used for data recordings.  
%  
%SYNTAX  
%[a]=IF_Filtering(f_sample,f_baud,f_carrier,a)  
%  
%The program filters the signal contained in vector a, and returns the result  
%to the same vector.  
%f_sample is the sampling rate, and f_carrier is the IF carrier frequency.  
%both expressed in MHz.  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%  
  
function [a]=IF_Filtering(f_sample,f_baud,f_carrier,a)  
fs=f_sample*1e6; baud=f_baud*1e6; fc=f_carrier*1e6;  
OS_RATIO=floor(f_sample/f_baud+.01);  
[Bif,Aif]=rcosine(baud/2,baud/2*(2*OS_RATIO),'sqrt',.62/5.38,10);  
tsif=1/fs; f0=fc;  
K=length(Bif)-1;  
caif=cos(2*pi*f0*tsif*[-K/2:K/2]);  
bif=caif.*Bif;  
figure  
freqz(bif,Aif,256,fs)  
a=conv(bif,a);  
a=a(1:2^12);
```

### III.8. Raised Cosine Window

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This function returns a window with a raised-cosine roll-off on both
%edges. The window is N-points wide, with a roll-off factor "beta"
%SYNTAX
%win2=rcwin(N,beta)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function win2=rcwin(N,beta)
n=0:1:beta;
win1=cos(pi*n/beta)+1;
N1=N-2*beta-2;
win2=[fliplr(win1) 2*ones(1,N1) win1];
win2=win2';
% figure
%% plot([0:N-1]',win2)
return
```

### III.9. Rectangular Window

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This function returns a rectangular window
%that is "N-2*beta" points wide, in a column vector that
%contains N points
%Syntax:
%win2=recwin(N,beta)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function win2=recwin(N,beta)

N1=N-2*beta;
win2=[zeros(1,beta) ones(1,N1) zeros(1,beta)];
win2=win2';
%figure
%stem([0:N-1]',win2)
return
```

TK5103.2 .E42 2003

Transmission techniques for  
broadband wireless up-link &  
digital interactive  
broadcasting [i.e.  
broadcasting] services :

**DATE DUE**[illegible]