AD-A265 147

WHOI-91-38

Woods Hole Oceanographic Institution

WOODS HOLE OCEANOGRAPHIC INSTITUTION

1930

Multiple Convergence Zone Acoustic Telemetry Feasibility Test Report

by

Josko A. Catipovic
Keith von Der Heydt
John Stevens Merriam
Woods Hole Oceanographic Institution

and

Geir Helge Sandsmark
University of Trondheim (Norway)

November 1991

Technical Report

Funding was provided by the Office of Naval Technology under Grant No. N00014-90-C-0098.

Approved for public release; distribution unlimited.

WHOI-91-38

Multiple Convergence Zone Acoustic Telemetry Feasibility Test Report

by

Josko A. Catipovic
Keith von Der Heydt
John Stevens Merriam
Woods Hole Oceanographic Institution

and

Geir Helge Sandsmark
University of Trondheim (Norway)

Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543

Accesion For		
NTIS CRA&I	X	
DTIC TAB		
Unannounced		
Justification		
By		
Distribution /		
Availability Codes		
Dist	Avail and / or Special	
A-1		

November 1991

Technical Report

Funding was provided by the Office of Naval Technology under Grant No. N00014-90-C-0098.

Approved for Distribution:

Albert J Williams 3rd

Albert J. Williams 3rd, Chairman
Department of Applied Ocean Physics and Engineering

Multiple Convergence Zone Acoustic Telemetry Feasibility Test Report

Josko A. Catipovic
Keith von Der Heydt
John Stevens Merriam
Woods Hole Oceanographic Institution

Geir Helge Sandsmark
University of Trondheim (Norway)

1.0 Summary

This report describes a multiple CZ acoustic telemetry experiment conducted off the coast of California 1/28/90 - 2/2/90. The goal was to design a maximally robust high speed underwater modem suitable for data telemetry for submerged platforms and moorings.

Six modulation methods were used to transmit data at rates from 1 to 1000 baud, corresponding to bit rates up to 3kbit/sec. The modulation formats were:

1. Multiple Frequency Shift Keying (MFSK) and Binary Expurgated Modulation (BEXPERM)
2. Duobinary Frequency Shift Keying
3. Quadrature Phase -Shift Keying (QPSK)
4. 8 Quadrature Amplitude Modulation (8QAM)
5. Continuous Phase Modulation (CPM) 2DPM4 and 2CPFSK4
6. Trellis coded 8PSK

In addition, a large number of channel probe sequences was transmitted in order to estimate channel multipath, fluctuation dynamics and spatial diversity characteristics relevant to acoustic data telemetry.

The data was transmitted from a 1 kHz source suspended from the R/V McGaw, and received on a multichannel vertical array tended by the R/V Point Sur The multichannel data was digitally recorded using floating-point digitizers and stored on optical disk for further processing. Approximate transmission ranges were 70, 140, 200 and 250 km. Approximately 8 hrs of transmission were recorded at each data range.

2.0 Experiment hardware

The transmitter used digital waveform generation to cycle through the modulation formats and data rates using a minimum of dedicated hardware. The transmitter was based on a personal computer utilizing an AT&T DSP32C digital signal processor plug-in board with analog to digital (A/D) converters for real-time waveform generation. The receiver was 1 32-element 1 km long vertical array deployed between 500m and 1500 m. It was tethered through a 3 km tether to the R/V Point SUR, which housed the multichannel digital recorders. The R/V Point SUR also deployed a profiling CTD to measure the acoustic channel profile.

2.1 Transmitter

The transmitter block diagram is shown in Figure 1 Blocks of waveforms to be transmitted were precomputed and stored on disk. The transmitter program placed the blocks in memory and sequenced the A/D through a predetermined block sequence to generate the desired analog waveform. The A/D output was a $\mp 3V$ analog waveform supplied to an ELGAR 1751SL 1.7 kW power amplifier.[1] The power amp provided an analog output at 0 to 200 Vrms, which was fed to a 4:1 step-up transformer for more efficient power matching at the transducer.

The transducer was a Lockheed/Sanders Model 30 class IV flextensional transducer. Its specifications are shown in Figure 3 through 5. The voltage response is 145 - 147 db re 1 volt per μPa, yielding a maximum transmit level of 203-205 dB, well within the transducer power rating.[2] During the experiment, the transmitter was generally operated 5-8 dB below this peak level. Output RMS voltage levels are logged among the transmission parameters for each transmission block.

The transducer was deployed 100m below the surface. It was attached to the stern frame of the vessel and generally followed the pitching motion of the vessel. Vertical displacements of up to 2 m from the mean transmitter depth can be expected due to ship motion, particularly during the early transmitted sequences, when the logged sea state was relatively high (6'- 8'). Horizontal drift velocities as high as 0.9 knots were also observed during the experiment, and resultant Doppler shift of up to 2 knots is expected in the data set.

The transducer was deployed on a 250 m "CTD wire" cable. The 0.322" diameter cable contained three #19 awg conductors within a steel armor. The overall measured cable re-

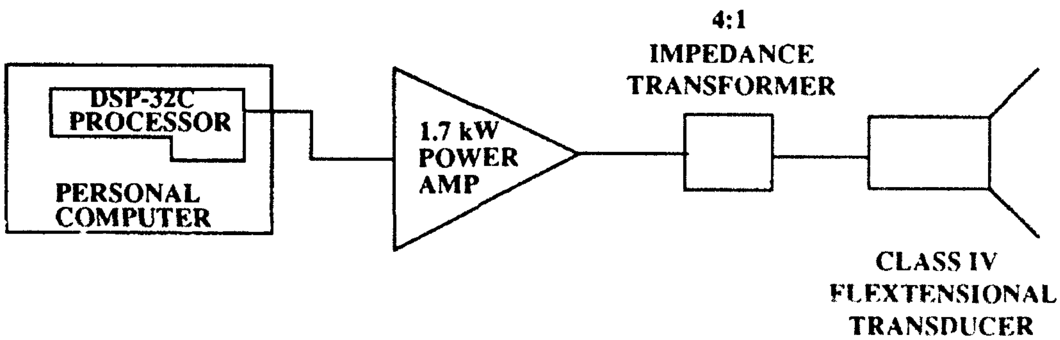

Figure 1: Waveform Transmitter Hardware

sistance is represented as a 11.56Ω resistance in series with the load. The cable is rated at a maximum voltage of 600V. However, our cable samples repeatedly withstood 1000V loads for extended periods of time, and no degradation or change in cable characteristics was noted after the cruise.

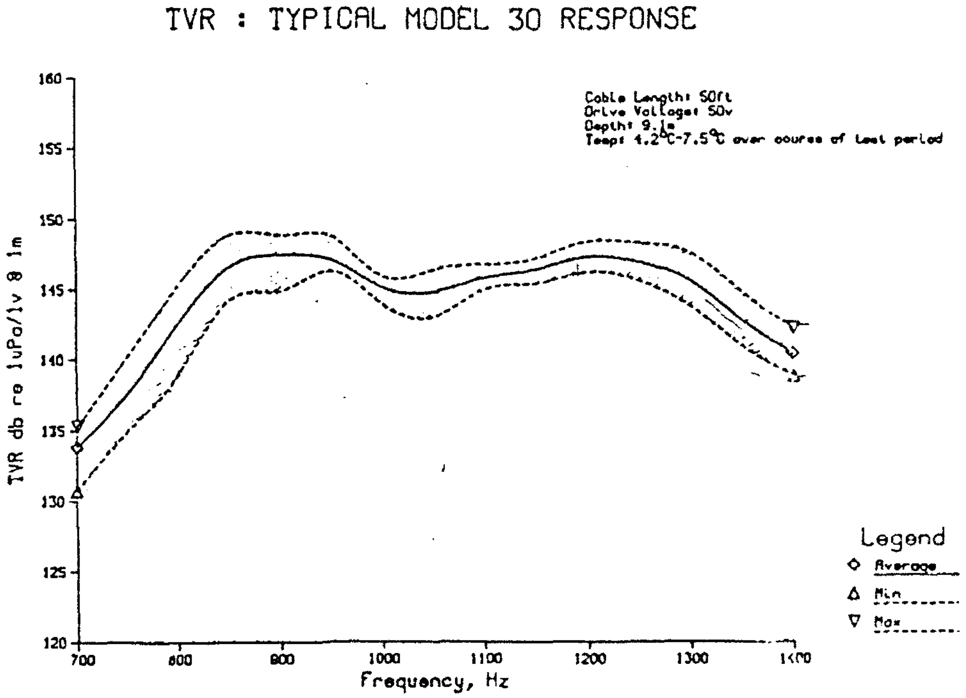

Figure 2: Model 30 Transducer transmit levels (courtesy Lockheed/Sanders)

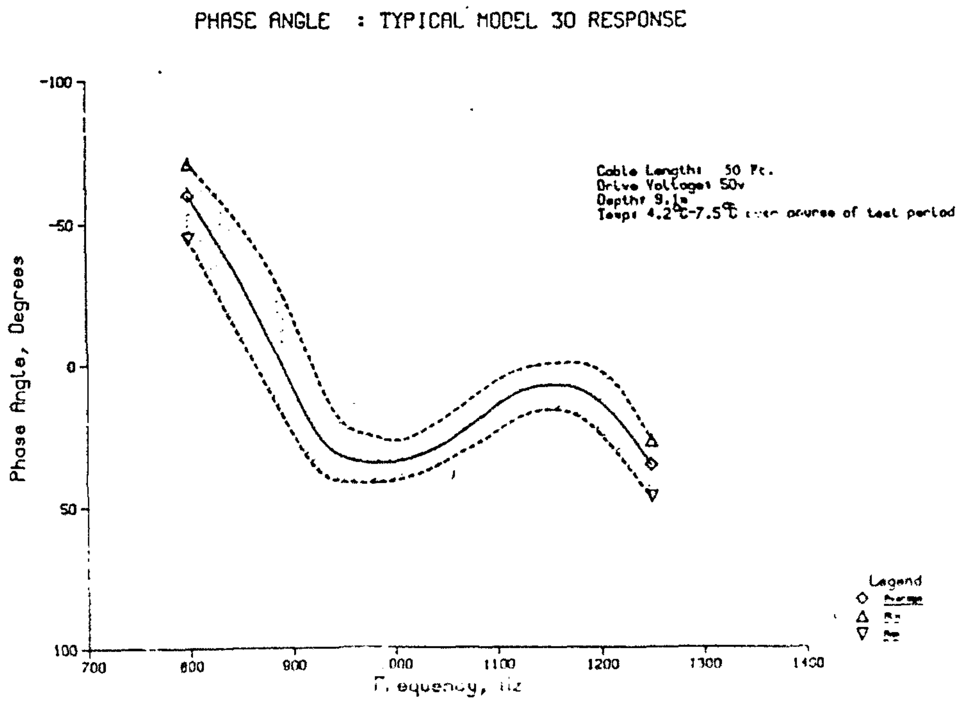

Figure 3: Model 30 transducer transmit phase angle (courtesy Lockheed/Sanders)

3.0 Waveform Generation

Experimental waveforms were stored on an NEC Powermate SX portable computer and transmitted via an AT Bus add-in board, AC5-C0-J, manufactured by Communications Automation and Control (CAC). The CAC board contains a 50 Mhz AT&T DSP32C Digital Signal Processor chip, 256 Kbyte 0 wait state ram and dual 14 bit D/A and A/D converters for general signal processing use. The board is I/O mapped into the AT Bus which can transfer data from the NEC PC's memory to the DSP32C's memory space at rates up to 1.5 Mbytes per second. To accomplish the data transfer, a PC program read raw data from the PC hard disk and passed the data to a program on the DSP32C which then clocked the data to one of the CAC board's D/A converters. The output sample rate was generated by a Syntest SI-101 Frequency Synthesizer box and was set to 4 Khz for this experiment.

Test waveforms (described in later sections) were generated on MATLAB, normalized to +/- 1.0, and stored as double precision floating point UNIX files. These were then converted to single precision binary floating point files using the MATLAB utility TRANSLATE386. A DOS program "MATCONV.EXE" was then used to generate a 16 bit integer DOS binary file from the binary floating point files. MATCONV scales the single precision floating point data to 14 bits to match the D/A converter on the CAC board but the data is saved as 16 bit integers on disk.

A data transmission program written in Microsoft C, '123_CNTL.EXE', was used to read the prestored binary waveforms and transfer the data to the CAC board. At start up, 123_CNTL.EXE downloads another program 'CACDAC' to the DSP32C. The two routines then interact through the use of two 'flip-flop' buffers in the DSP32C memory. This arrangement enables 123_CNTL to read a block of waveform data from disk and pass the data to one of the two buffers in the DSP memory while CACDAC clocks data out of the other buffer. Then, when CACDAC has clocked out that buffer's data, the two programs switch buffers. As long as 123_CNTL is able to read a block of disk data and send it to the CAC board before CACDAC has clocked out its buffer, then real time contraints have been met and the transmitted waveforms are continuous. Given the relatively slow 4K sample per second output rate, 123_CNTL only uses about 25% of its available time. Most of the time used by 123_CNTL is dependent on the hard disk access time. In other experiments the PC RAMDRIVE has been used to speed up reading the disk files and output rates of 60Khz have been accomplished.

CACDAC is written in C and compiled with the AT&T DSP32C C compiler. As mentioned, it handshakes with 123_CNTL to receive data to be transmitted. This data is sent to the D/A converter through the software programmable DSP32C serial I/O port using Direct Memory Access (DMA). The output sample frequency clocks the DMA address pointer through one of the two flip-flop buffers. The DMA controller is not capable of sending the address pointer back in a circular buffer fashion, so the software must periodically check the status of the DMA pointer and move it to another data buffer when the current buffer is exhausted. CACDAC handles this function and ensures that no discontinuities occur in the data.

Transmitted waveforms (e.g. 2DPM4 or 2CPFSK4 of section 4) were actually sequences of smaller binary waveform files which were designated "id1" files. A given waveform was then formed by placing the names of the .id1 files in an appropriate order in a so-called ".nam" (for name) file which is an ASCII file read by 123_CNTL. 123_CNTL accepts a .nam file as input and opens up as many as 255 of the .id1 files listed. 123_CNTL then cycles through the list of files reading in their data and transferring it to the DSP. Many of the waveforms transmitted contained the some of the same patterns, so it made sense to form the waveforms from collections of smaller ones. Furthermore, to prestore all of the waveforms transmitted as complete separate entities would have used too much disk space. The .id1 files themselves required a total of about 20 Mbytes of PC disk space.

Transmission of data for the experiment was done during 90 minute windows when the Heard Island experiment was not active. To automate the transmission process, DOS batch files were used to form sequences 1 through 12 (section 4.0) by calling 123_CNTL a number of times, each time with an appropriate .nam file as input. A given sequence would last approximately 65 minutes.

4.0 Receiver

4.0.1 Array description

Signals were recorded from a 32 channel array of hydrophones suspended vertically in the water column. The primary purpose of this array was the reception of the 57 Hz signal during the Heard Island Feasibility Experiment. Telemetry experiments were conducted during the periods that the Heard Island source was not being received. The array was electro-mechanically connected to the R/V Pt. Sur from Moss Landing Marine Labs, Moss Landing CA. Recording systems were housed and operated on board the Pt. Sur. The receiver configuration is shown in Figure 4.

The 32 hydrophone sensors in the array were identical, each consisting of a small cylindrical ceramic element with an internal preamp resulting in a sensitivity at its output of -170 dB re. 1 $\mu Pa/\sqrt{Hz}$The frequency response is nominally flat up to at least 2 kHz with an omnidirectional characteristic. The low end -3dB point was 10 Hz due to the sensor capacitance and the preamp input impedance. The sensors were supplied with bipolar power via the cable and each had an independent, unshielded twisted pair that transmitted the signal the length of the cable (from a minimum distance of 1.6km to a maximum of 3km from sensor #31) to the recording systems aboard ship. At the shipboard end, signals were tapped off via isolated BNC's on a patch box to a bank of differential input amplifiers.The input to each of these was shunted by a 1% 10k resistance. Each side of the differential amplifiers was connected to local common through a 1 megohm bias resistor. The inboard 100 feet of the cable was shielded and care was taken with grounding to minimize 60 Hz pickup. The driving concern with 60 Hz contamination was the proximity of the Heard Island signals and was clearly not an issue for telemetry purposes. The net effect however of cable impedances and the 10k shunt is that the signals provided to the acquisition system were attenuated by 4 to 7 dB at 1 kHz depending on transmission distance on the cable. This of course has no effect on signal-to-noise ratio but must be included in absolute level calculations.

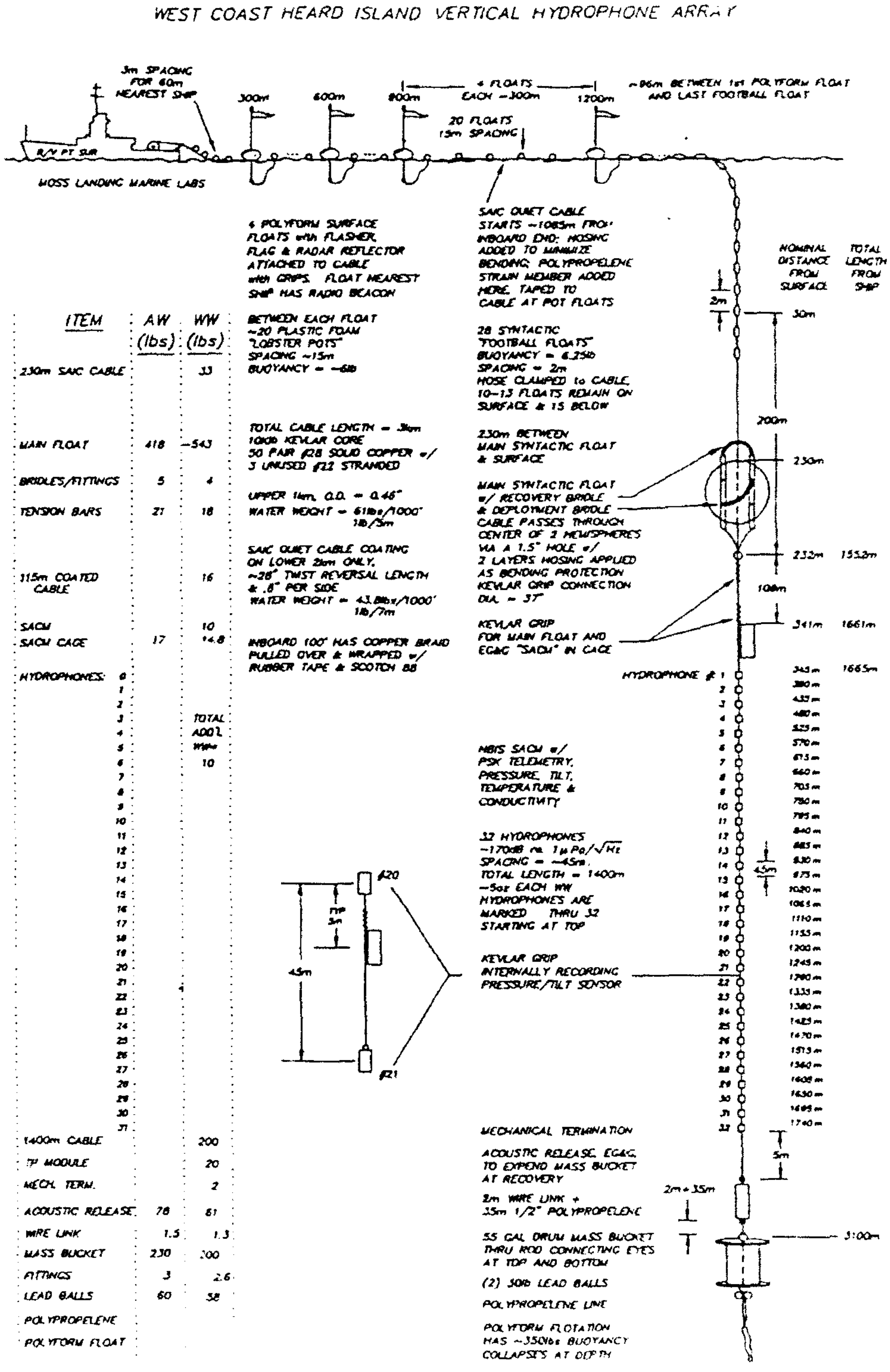

Figure 4: Receiver array configuration

A fixed gain of 20dB was applied to all channels at the differential amplfier, each of which drove 4 single ended unity gain buffers. This allowed some isolation among of each of the 3 independent recording systems used for the Heard Island work. Telemetry signals were recorded only on theWHOI multichannel system.

4.0.2 Array Deployment

The array was suspended in the water column to bracket the sound channel axis at about 800 meters with the top sensor (#0) at 350 meters depth and the bottom sensor,(#31) at 1750 meters. The sensors were linearly spaced every 45 meters along the 1400 meter length of the array. The suspension system was designed to hold the array at this depth and minimize the effect of swell motion observed to be about a 12-13 second period. The array was ballasted to maintain this attitude with a slack surface tether section of approximately 1.1km. Refer to the drawing of Figure xx for a details of the configuration. The ship continuously attempted to minimize strain on the cable by maneuvering very slightly ahead or astern to maintain an S-shaped surface tether configuration. If the ship maintained a strain of 100 pounds or more on the surface tether, the array would slowly rise and assume an increasing angle with the vertical at the top.

Two depth and inclination recording instruments were attached to the array as shown on the drawing. The upper unit telemetered data at a 2.5 second period in realtime to a logging system on the ship. The lower unit recorded data at 1 minute period internally. Unfortunately, data from neither of these units is available for most of the telemetry transmissions. Judging from the benign sea state during telemetry experiments and depth/ tilt data acquired previously during Heard Island transmissions, a tilt estimate of less than 5 degrees and depth variation of less than 50 meters can be made.

4.0.3 Data Recording System

The WHOI acquisition system is auto-gain-ranged, thereby offering a very large measurement range (>120 dB). The system consists of a a suite of cards, 1 per channel, on a common bus and interfaced to a generic PC/AT computer. Data were stored on optical disks but can be stored on any SCSI device. Each card is independent and includes the following functionality in order from input:

- programmable fixed gain block, 0 to 42 dB, 6 dB per step
- selectable 4 pole Butterworth high pass filter
- programmable 8 pole Butterworth low pass filter, 20 to 5120 Hz, 20 Hz steps
- 5 gain range amplifier (0,18,36,54,72 dB), with 13 bit ADC at each output
- microcontroller

When configuring an acquisition session, the PC communicates with all connected amplifier cards to program fixed gain, select the high pass filter, select the lowpass cutoff, initiate offset nulling operations and set other parameters related to the sampling process.

A common sample period pulse is provided to all channels for simultaneous sampling, i.e. there is no skew in sampling time across channels. For each sample the microcontroller selects the correct gain based on amplitude and slew rate, makes an offset correction and outputs a 16 bit data word upon being addressed by a high speed interface in the PC. The 16 bit sample consists of a 13 bit mantissa which is the output from one of the ADC's and a 3 bit gain word representing the dynamic gain applied to the signal for that sample.

The amplifier interface is essentially a programmable sequencer with FIFO buffering and the ability to transparently store data via DMA into the full 16 megabyte memory space of the PC/AT (ISA bus) machine. This ability touse **extended** memory space makes it possible to use multiple buffers and to devote significant amounts of memory space to achieve continuous acquisition rates at speeds in excess of 1 Mbyte/sec with an inexpensive computer.

As buffers fill, SCSI transfers from memory to the optical disk are initiated by the acquisition program. Using the Optimem disks, continuous acquisition can occur as fast as 200k samples per second. For the telemetry receptions, up to 32 channels were sampled at 4 kHz.

The acquisition parameters of concern for processing the telemetry data are the following:

- 4 kHz sampling rate

10 Hz low cut filter on GRA and 10 Hz low cut at the sensor for -6dB point at 10 Hz and -30 dB/oct rolloff below 10 Hz.

- 1400 Hz low pass -3dB point, 48 dB/oct rolloff
- 20 dB fixed gain for all data
- 4 to 7 dB attenuation at 1 kHz as a function of sensor position on the cable, i.e. -4 dB @ sensor #0 and -7 dB @ sensor \#31

4.0.4 Format of data on Optical Disk

All data for this experiment series was recorded on an Optimem 1000M optical disk drive using 12 inch PDO media with 1.2 gigabytes per side. The data from 16 files is contained on 4 disks. It amounts to a total of 8.8 GB. Refer to Appendix 4 for information on individual files. The file structure of these disks is **not compatible** with any operating system. A suite of DOS based programs has been written to read these disks. To date we have not read data directly with a UNIX based machine though theoretically that should be possible.

Each optical disk starts and ends with a disk header that is one sector long. Thus there is a disk header sector at the first logical block (LBA 0) and the last logical block, (LBA 1172499) for 1.2 GB per side disks. The file directory for a disk exists on the disk in 2 forms:

1. The "compact" directory which is stored backwards sequentially starting at the last LBA, i.e. at the location of the 2nd disk header. It is a series of directory entries, each one sector long, with one entry per data file. The compact directory is normally used since a listing of all files on the disk will be found in one place, i.e. the end of the disk. Typically then, LBA 1172499 would contain a disk header sector, LBA 1172498 would contain the directory entry of the first file written, LBA 1172497 for the second file, and so on.

2. Separate directory entries, each 1 sector long and identical to the entries in the compact area, are found preceding each file on the disk.

Thus, the procedure used during acquisition is to write data starting at LBA 0, while writing directory entries backward from the end. Two types of files are on the disks: *.DAT files which hold the data, and *.HDR files, which are ASCII files that contain info about the corresponding *.DAT files, eg phone sensitivity, sample rate, etc. The *.HDR files are independent files that have two sets of directory entries as do data files.

Data files from the TELEMETRY experiments are multiplexed and groundtrue (an "active" bit is a "0"). A data file is divided into records which have NO STANDARD LENGTH! The record length will be constant however, throughout any given data file. A data file is a contiguous sequence of records, each record beginning with a sector containing a record header,in which the record size, number of channels, etc.V are identified. The time to the microsecond of the first sample in any record and its number is recorded in the record header. Most of the information in record headers will be the same throughout a file. Following each record header sector will be a number of sectors containing multiplexed data strung out in this manner: chan 1 value, chan 2 value, \ldots, chan N value, chan 1 value, chan 2 value,\ldots etc. Each data value is two bytes in length and is written as an integer which means that if one examines a sequence of raw values, the least significant byte of a 2-byte value occurs first. There will be an equal number of values for all channels in one record. After the last sector in a record, the following sector will contain the next record header, followed by more data. In this way, all the data for one file occupies a contiguous area on the disk.

Information about the size of an entire data file is not contained in the record headers. This, along with information about the file's address, is found in the corresponding directory entry of the file. Appendix 2 contains the "C" language structures that describe the information held in directory entries and data record headers for data files. Also, a "C" routine is given that has been used to convert the stored 2-byte raw data floating point format to native single precision floating point.

4.1 Accessing Telemetry data from the optical disks

Presently, the only way to access data on optical disks is via the Powermate PC, with Internet hostname **necco.whoi.edu** in Bigelow 313. It is connected to WHOInet and can be accessed by all machines on the net. The network software on this machine is public domain NCSA, which does not allow a remote user to get an MS-DOS shell on the PC from a remote UNIX machine. This means somebody has to physically be sitting with the PC in order to get data. This isn't so bad, since someone has to to be there anyway to mount the optical disk required. The PC is equipped with a SCSI interface to the Optimem disk drives. Before powering up the PC, the Optimem disk drive must be on, so that the PC's initialisation routines will see the drive. A message, " WORM device found" will be displayed upon successful startup. The main programs for interacting with the optical disk are in the OPT directory, and are named: **odir** and **odrd**. A third program that is useful for browsing through record headers in an optical datafile is **od2gld.** These programs are all written in C, and come with many options. We will discuss only a few here, but a complete listing of options can be found at the start of **odrd.c** and **odir.c** .

ODIR

Just typing **odir** , in most cases, will provide the user with a list of all data files on the currently mounted optical disk. If a new disk has just been loaded into the drive, you may get an error message the first time \tt odir \rm is executed. Just execute it again. In rare cases, the directory area of the disk may have been clobbered, and some of the options to \tt odir \rm may have to be invoked. SEEK HELP!!.

OD2GLD:

Before reading a file, a user might like to browse through an optical disk file to find out useful parameters like the record length in bytes and time, the GMT time a record was acquired, etc. A program was written on the PC to convert optical disk files directly to GLD file format. (For a colorful discussion of "GLD" format refer to the document **Practical Guide to POLAR.WHOI.edu and ARCTIC.MIT.edu** found in Bigelow 313.) It probably will never be necessary to run this program on the PC, since a lot of floating point operations are done in converting 16 bit aquisition format to 32 bit floats, and doing this on a PC takes forever. However, **od2gld** , has a nice user interface, which displays record headers, file length info, interactively. Invoke the program this way:

od2gld filename

A menu appears with certain default settings which probably needn't be altered. You are then asked if you want to skip through the file. If you do, you can see record headers anywhere in the file. When you are done, kill the program (^ C) and delete the zerolength **filename.gld** file it created.

ODRD:

This program copies data from the optical disk to a disk file on the PC. The format of data on the disk, which is multiplexed, is preserved. The simplest invocation of odrd :

odrd -bi fname1 -bo fname2

would copy an entire data file, **fname1** to the PC as file **fname2** . This normally would result in an error because most files on the optical disk are huge, and the total storage for this purpose on the PC is about 30 Mbs. The following command:

odrd -bi fname1 -bo fname2 -st recno -rs howmany

where **recno** is the record number to start from, and **howmany** specifies the number of records to use, will be of more use. A typical TELEMETRY data file has 12 channels of data. If the user does not require to read all channels, there will be more storage space left on the PC to read in more records. Two options **-siftk** and -siftd are available for "sifting" through multiplexed data, and only extracting certain channels. For instance:

odrd -bi fname1 -bo fname2 -st recno -rs howmany -siftk

will result in the program requesting the user the channel numbers he wishes to keep. (-siftd asks which to delete). The resulting PC file for the sift options are in the original, multiplexed format, with headers adjusted to reflect the decreased number of channels.

OPTICAL (ODAS) FORMAT TO GLD FORMAT:

When the desired image of the optical diskfile is available on the PC, the next thing to do is to move it to a minicomputer or workstation. This can be accomplished over the network using ftp. On the PC, type **ftp hostname** , supplying the desired destination host. After logging in to **ftp**, tvpe **binary** , since a binary file tranfer is necessary. Then **send PCfilename** .

OD2GLDV:

If the destination machine is a microVAX, after the transfer is made, the program **od2gldv** , a VAX version of **od2gld** , can be run. Like **od2gld** , od2gldv , is user-friendly. Its menu contains the defaults for converting TELEMETRY date. Options for selecting channels at this point are also available. To do so, create a file with a text editor that includes pairs of channelnumber status , where status = 1 if the channel is to be included. In **od2gldv** 's main menu, enter the name of the created channel file. A GLD file will be created with the selected channels only, i.e. the number of rows will be equal to the number of selected channels.

4.1.1 MATLAB Format

A defacto data file format that is being used is the format forMATLAB *.mat files. This format is necessary to load external data intoMATLAB with MATLAB's **load** command. The format is very similar to GLD format: data is arranged in matrix form, and a small header PRECEDES the matrix. The differences are:

1. One *.mat file can hold multiple matrices (called "variables" in MATLAB). Each matrix, with its header, is just concatenated to the previous variable's in the *.mat file.

2. For each variable in a *.mat file, the header preceding the data is 20 bytes (5 four-byte long integers) long. This header contains information about the type of host machine, size of the matrix, whether it is real or complex data, and the length of the variable name. Following these five bytes is a string containing the variable name (terminated by a NUL character). This means that the header does not have a constant length. Its length depends on the number of characters in the variable name.

Native MATLAB matrices ("variables") are stored in column order in memory. GLD files are stored in row order. ALSO, all variables in MATLAB are converted to double precision (eight bytes per value). If you **save filename variablename** a variable in MATLAB , the variable is put on disk with the name filename.mat . The data stored will be in column order, with 8 bytes per value.

4.1.2 GLD To MATLAB Conversion

We have written a program, **gld2mat** , that converts a GLD file into *.mat format, for porting to MATLAB . The program requests input file name, and then asks for an output file name which should end in .mat . It then requests a variable name which will reference the entire GLD file once you are in MATLAB . After this the user is prompted for a destination machine type, supplying codes for PC, SUN, MAC, and VAX computers. You pick the desired code. A one variable *.mat file is then produced. The user can then start MATLAB , and load *.mat to obtain access to the GLDfiles's data as a variable. Two things will have happened. The header information supplied by **gld2mat** will have taken care of the row-wise versus column-wise orientation problem, and the size of the variable in MAT-

LAB will have doubled. Type whos to see the current MATLAB storage. If you later save this variable, the output *.mat file will be twice as long as the input file, and it will be column-oriented. As of 9/1990, there is no way to save a file in single precision, although "*they are working at it*". The double precision "feature" of MATLAB makes it difficult to work with a typical GLDfile, which may be on the order of 1 MByte or more. Once loaded into MATLAB , it consumes twice as much memory. Using the MATLAB clear command only frees the memory to the MATLAB process, the cleared memory is not made available to the Operating System. Typically, UNIX will slow down if MATLAB memory approaches the hardware limit, even if the user has cleared a huge variable. It is best to partition a huge variable, save the partitions, and exit MATLAB . Then re-invoke MATLAB , and load the saved partitions of data.

Another quick hint on **gld2mat** . If you wish to see your data values in a *.mat file from UNIX with an **od -f** command, you must use a variable name that is 3, or 7 characters long. The length of a *.mat header is not an even multiple of four unless you do this, and the subsequent float values will appear to be garbage.

4.1.3 Optical Disc To MATLAB Conversion

For those who wish to use optical disk data in MATLAB , without having to go through the steps of generating a GLD file, there are 2 routines, **od2matsp** and **od2matv** , for the SUN Sparcstation, and the microVAX, respectively. The friendly user interface to these programs is almost identical to that of **od2gld(v)**(see above).

Again, the sequence of events to move data from the optical disk to MATLAB on the SPARC station is:

1. In Bigelow 313, get help to install the right optical disk in the drive and use the PC program ODIR to look at the optical directory for your file

2. Use the PC program OD2GLD to check record times if desired tomake sure you will get the data you want.

3. Use the PC program ODRD to read data from the optical disk onto the PC hard disk....-probably no more than 30 MB at a time.

4. Use ftp {\em binary} at the PC to transfer the data to workstation of choice

5. Use OD2MATSP or OD2MATV to convert the data to native float in MATLAB format

5.0 Transmitted Data

The objective of the experiment was to evaluate a number of promising data modulation techniques at a wide range of data rates.The transmitted data is arranged into sequences approximately 70 min in duration. The 70 min duration derives from the recording capabilities of the receiver and the transmission interleave schedule with the Heard Island experiment, which was on-line for approximately 1 hour at 3 hour intervals, i.e from 01:00 to 02:00 PST, 04:00 to 05:00 PST, 07:00 to 08:00 PST etc. An additional 15 minutes of listen time was reserved to receive time-dispersed arrivals from Heard Island.

The data to be transmitted was organized into sequences. Each sequence is approximately 20 min in length. The sequences were in turn combined into 60-70 min. blocks for transmission. The two-step procedure was used to allow recombining the sequences in the field without having to manipulate the individual short segments. The structure of individual sequences is detailed below.

5.0.1 Sequence 1

This sequence contains 4PSK and 8QAM sequences. It is approximately 19 minutes in duration and organized as follows:

Modulation	Baud rate	Duration
4PSK	1 Hz	3 min 5 sec
4PSK	3.33 Hz	1 min 7 sec
4PSK	10 Hz	40.3 sec
4PSK	33.3 Hz	43.4 sec
4PSK	100 Hz	29.3 sec
4PSK	333 Hz	33.9 sec
4PSK	1 kHz	23.5 sec
8QAM	333 Hz	1 min 4.6 sec
8QAM	1 Hz	2 min 16 sec
8QAM	3.33 Hz	1 min 54.9 sec
8QAM	10 Hz	2 min 16.3 sec
8QAM	33.3 Hz	2 min 38.6 sec
8QAM	100 Hz	54.9 sec
8QAM	1 kHz	23.5 sec

Individual sequences are separated by1 sec quiet periods and are acessible by time-indexing from the block beginning. The individual waveforms consist of several signal files put together contiguously as shown in figure 6. Each waveform is started with a barker code transmitted with chip duration equal to the symbol duration for the actual waveform. Then follows a 1s silent interval before the data sequence is started with a preamble. Then follows one or more periods of a periodic data sequence derived from a maximum length shift register sequence. After the data sequence follows a 1 s. pause followed by the Barker followed by a new 1s pause.

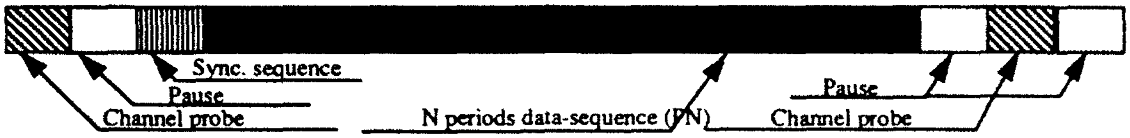

Figure 6. Transmitted signal format.

5.0.2 Sequence 2

This sequence contains channel probe sequences as well as MFSK and duobinary FSK transmissions. It is organized as follows:

Waveform	type/data rate	duration
500 Hz FM sweep	channel probe / 1 sec	90 sec
500 Hz FM sweep	channel probe / 5 sec	140 sec
1023 chip PN code	channel probe / 4 msec chip	368 sec
4 CW tones	channel probe / 50 Hz spacing	180 sec
16 CW tones	channel probe / 50 Hz spacing	180 sec
64 CW tones	channel probe / 10 Hz spacing	180 sec
500 Hz FM sweep	channel probe / 5 sec	140 sec
1023 chip PN code	channel probe / 4 msec chip	368 sec
FSK - 1 Hz	1 bit/sec	180 sec
4FSK- 1 Hz	4 bits/sec	180 sec
16FSK - 1 Hz	16 bits/sec	162 sec
32FSK - 1 Hz	32 bits/sec	162 sec
500 Hz FM sweep	channel probe / 5 sec	140 sec
1023 chip PN code	channel probe / 4 msec chip	368 sec
FSK - 10 Hz	10 bit/sec	180 sec
4FSK- 10 Hz	40 bits/sec	180 sec
16FSK - 10 Hz	160 bits/sec	162 sec
32FSK - 10 Hz	320 bits/sec	162 sec
Duobin. FSK - 10 Hz	20 bits/sec	210 sec
Duobin FSK - 100 Hz	200 bits/sec	210 sec

Individual sequences are separated by 5 sec quiet periods and are acessible by time-indexing from the block beginning.

5.0.3 Sequence 3

This sequence contains Digital Phase Modulation (DPM) transmissions at data rates from 3 1/3 to 1000 baud. Modulation index = 0.8.

Modulation	Baud rate	Duration
2DPM4	3.33 hz	6 min. 39.9 s
2DPM4	10 hz	2 min 15.3 s
2DPM4	33.3 hz	43.3 s
2DPM4	100 hz	29.1 s
2DPM4	333 hz	20.3 s
2DPM4	1 khz	13.3 s

5.0.4 Sequence 4

(sequence 15)

This sequence contains a 22 minute transmission of a 1 baud digital phase modulation with modulation index = 0.8. It is treated separately becuase of its duration.

Modulation	Baud rate	Duration
2DPM4	1 hz	21 min. 56 s

5.0.5 Sequence 5

This sequence contains Continuous Phase Frequency Shift Keying modulation (CPFSK) transmissions at baud rates from 3 1/3 to 1000. Modulation index = 0.8.

Modulation	Baud rate	Duration
2CPFSK4	3.33 Hz	6 min. 39.9 s
2CPFSK4	10 Hz	2 min 40.5 s
2CPFSK4	33.3 Hz	1 min 6.2 s
2CPFSK4	100 Hz	44.4 s
2CPFSK4	333 Hz	52.3 s
2CPFSK4	1 kHz	19.4 s

5.0.6 Sequence 6

This sequence contains a 22 min. transmission of a 1 baud CPFSK waveform with modulation index = 0.8.

Modulation	Baud rate	Duration
2CPFSK4	1 hz	21 min. 56 s

5.0.7 Sequence 7

This sequence is identical to sequence 3 except the modulation index = 1.0

(sequence 30 through 35)

Modulation	Baud rate	Duration
2DPM4	3.33 hz	6 min. 39.9 s
2DPM4	10 hz	2 min 15.3 s
2DPM4	33.3 hz	43.3 s
2DPM4	100 hz	29.1 s
2DPM4	333 hz	33.9 s
2DPM4	1 khz	13.3 s

5.0.8 Sequence 8

This sequence is identical to sequence 4 except the modulation index = 1.0.

(sequence 29)

Modulation	Baud rate	Duration
2DPM4	1 hz	21 min. 56 s

5.0.9 Sequence 9

This sequence is identical to sequence 5 except the modulation index = 1.0

(sequence 37 through 42)

Modulation	Baud rate	Duration
2CPFSK4	3.33 Hz	5 min. 15.3 s
2CPFSK4	10 Hz	2 min 40.5 s

2CPFSK4	33.3 Hz	1 min 6.2 s
2CPFSK4	100 Hz	44.4 s
2CPFSK4	333 Hz	52.3 s
2CPFSK4	1 kHz	19.4 s

5.0.10 Sequence 10

(sequence 36)

This sequence is identical to sequence 5 except the modulation index = 1.0

Modulation	Baud rate	Duration
2CPFSK4	1 hz	21 min. 56 s

5.0.11 Sequence 11

(sequence 44 through 49)

This sequence contains trellis coded 8PSK waveforms transmitted at baud rates from 3 1/3 to 1000.

Modulation	Baud rate	Duration
r=2/3 coded 8PSK	3.33 hz	6 min. 43.8 s
r=2/3 coded 8PSK	10 hz	2 min 16.6 s
r=2/3 coded 8PSK	33.3 hz	1 min 21.8 s
r=2/3 coded 8PSK	100 hz	29.2 s
r=2/3 coded 8PSK	333 hz	33.9 s
r=2/3 coded 8PSK	1 khz	13.3 s

5.0.12 (sequence 43)

This sequence contains a 1 baud trellis coded 8 PSK waveform

Modulation	Baud rate	Duration
r=2/3 coded 8PSK	1 hz	21 min. 19 s

6.0 Transmitted data schedule

Our goal was to transmit each of the above sequences at a set of ranges between 60 and 200 km in the deep ocean. Transmit times and ranges for each sequence are given in the chart below. The received data is indexed and retrievable by the transmit times. Individual waveform are referenced in time to the beginning of a sequence. In general, the recieved SNR is sufficiently high to allow sequence detection with band-pass filtering.

The ranges given in the table below are nominal and varied slightly due to ship drft rates. Accurate ship poositioins are available. The R/V Point Sur position is available from the headers of the received data, and R/V Wm. Mcgaw positipon is available from the ship log reproduced in the apendix. Sequences 5 through 12 were not transmitted at 110 nm nominar range because of rough weather forcasts and our desire to maximize transmission durations from the 140 nm mile range.

Sequence	40 nm	80 nm	110 nm	140 nm
Seq 1	1/28/91	1/31/91	2/1/91	2/1/91
	14:39 (3)	20:33(3)	08:25 (3)	17:55 (1)
Seq 2	Note 1	1/31/91	2/1/91	2/1/91
		21:35 (1)	09:25 (1)	14:55 (1)
Seq 3	1/31/91	1/31/91	2/1/91	2/1/91
	11:30 (2)	22:50 (2)	10:37 (2)	16:05 (2)
Seq 4	1/31/91	1/31/91	2/1/91	2/1/91
	14:45 (1)	32:20 (2)	11:00 (2)	18:30 (1)
Seq 5	1/31/91	2/1/91		2/1/91
	11:50 (2)	00:05 (2)		16.20 (2)
Seq 6	1/31/91	2/1/91		2/1/91
	15:05 (1)	00:40 (2)		18:50 (1)
Seq 7	Note 2	2/1/91		2/1/91
		01:20 (2)		16:40 (2)
Seq 8	1/31/9	2/1/91		2/1/91
	15:20 (1)	02:00 (2)		19:15 (1)
Seq 9	1/31/91	2/1/91		2/1/91
	8:30 (2)	02:35 (2)		19:15 (2)
Seq 10	Note 3	2/1/91		2/1/91
		03:05 (2)		19:43 (1)
Seq 11	1/31/91	2/1/91		2/1/91

	8:50 (2)	03:41 (2)	17:35 (2)
Seq 12	Note 3	2/1/91	2/1/91
		04:20 (2)	20:05 (1)

Note 1 This sequence was transmitted in part on 1/28/31. (Waveforms up to f1_1_3). The sequence was completed on 1/31/91 at 8:30 in a different location.

Note 2: This sequence was transmitted in part (Waveforms 30,31) onj 1/31/91 at 12:10. Due to equipment failure, it was completed at 14:30 at a range of 51 nm. (Waveform 32, 33, 34, 35). During the hour required to repair the equipment, the ship steamed to increase range to Pt. Sur.

Note 3: The two sequences were transmitted on 1/31/91 at 17:30 at a range of approx 60 nm. Seq 10 was transmitted once and seq 12 twice.

Note 4: Sequence 1 was transmitted twice at this range on 2/1/91. The first transmission was at 17:55 and the second at 20:15. (Sea state increased between first and second transmissions.)

7.0 Received data format

A thorough analysis of the received data is beyond the scope of this document. Several sample records were processed to verify received data quality and obtain preliminary estimates of channel multipath structure and stability.

8.0 Appendix 1: MATLAB code used for waveform generation

The details of each data-sequence are given below including excerpts of MATLAB code used to generate the different parts of the waveforms. The preamble and pause files are described in a separate section.

4PSK -1 bit/sec

The following sequence of files are transmitted contiguously:

4PSK 1 Hz symbolrate total duration 3 min. 5s:

Filename	signal type	Duration: Samples	seconds
f1311b01.mat	barker code	52000	13
f151ps1s.mat	silence(pause)	4000	1
f412qp01.mat	preamble (+ trans)	128001	31
f411qp01.mat	1 period data seq.	124000	31
f411qp01.mat	1 period data seq.	124000	31
f411qp01.mat	1 period data seq.	124000	31
f411qp01.mat	1 period data seq.	124000	31
f151ps1s.mat	silence(pause)	4000	4
f1311b01.mat	barker code	52000	13
f151ps1s.mat	silence(pause)	4000	4

The files containing the preamble and the data sequence were generated by the matlab m-file "seq4psk123.m". The preamble consists of 5 symbol interval of the carrier with phase -45 degrees followed by 5 symbols with +90 degrees shift relative to the previous symbol followed by another 5 symbols where the relative shift is 180 degrees. The preamble is concluded by a 13 element barker code. A 31 bit long PN-sequence is used as a data sequence. The symbols are generated from the PN-sequence by assigning the sequence directly to the quadrature component and assigning the same sequence advanced by 10 bits to the in-phase component. This symbol sequence is interpolated to yield a sampling frequency of 4 kHz by filtering it through a truncated 4 symbol intervals long cosine roll-off filter with roll-off factor 0.5. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> seq4psk123
 carrier frequency (Hz):
f0 = 1000
 symbol rate (Hz):
symbrat = 1
 output sampling frequency (Hz):
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 1 0
 roll-off factor:
betha = 0.5000
 (half) constraint length of roll-off filter:
corolnth = 2
 No of output symbols requested:
nsymb = 40
 Number of symbol intervals for tone:
nsil = 5
 Number of symbol intervals for 90 deg shifts:
n90 =5
 Number of symbol intervals for 180 deg shifts:
n180 = 5
ans =4-Jan-91
 Informative remark: time 15:10, 1Hz 4psk pn 40 symb, per=31, sh.reg: 5
ndel=fix((2^length(genpol)-1)/3);
ngen=nsymb+ndel;
pn=pnsequ(reginit,genpol,ngen);
clear dseq
dseq=preamb4psk(nsil,n90,n180,1+sqrt(-1),sqrt(-1));
```

```
dlen=length(dseq);
dseq(1+dlen:dlen+13)=[-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1];
dlen=length(dseq);
pn=2*(pn-.5*ones(pn));
dseq(1+dlen:dlen+nsymb)=pn(1+ndel:ngen)+sqrt(-1)*pn(1:nsymb);
iprat=fix(fs/symbrat);
cr=cosroll(betha,iprat,corolnth);
Warning: Divide by zero
Warning: Divide by zero
clear oseq oseqr
pack
oseq=interpfilt(cr,iprat,dseq);
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
???? Error using ==> *
Out of memory. Type HELP MEMORY for your options.
>> carr=exp((sqrt(-1)*2*pi*(f0/fs))*(0:999));
>>for k=1:272,oseq(k*1000-999:k*1000)=oseq(k*1000-999:k*1000).*conj(carr');end
>> oseqr=real(oseq);
>> oseqr=oseqr/max(abs(oseqr));
>> pnstrt=1+length(cr)+iprat*dlen;
>> pnper=(2^length(genpol)-1)*iprat;
>> pnstrt
pnstrt =128002
>> pnper
pnper =124000
>> head=oseqr(1:pnstrt-1);
>> body=oseqr(pnstrt:pnstrt+pnper-1);
>> save f411qp01 body
>> save f412qp01 head
>> exit
10799441 flops.
```

4PSK 3.33 Hz symbolrate total duration 1 min. 7s:

The following sequence of files are transmitted contiguously:

Filename	signal type	Duration: Samples	seconds
f1312b03.mat	barker code	15600	3.9
f151ps1s.mat	silence(pause)	4000	1
f422qp03.mat	preamble (+ trans)	38401	9.6
f421qp03.mat	1 period data seq.	37200	9.3
f421qp03.mat	1 period data seq.	37200	9.3
f421qp03.mat	1 period data seq.	37200	9.3
f421qp03.mat	1 period data seq.	37200	9.3
f421qp03.mat	1 period data seq.	37200	9.3
f151ps1s.mat	silence(pause)	4000	1
f1312b03.mat	barker code	15600	3.9
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seq4psk123.m". The preamble consists of 5 symbol interval of the carrier with phase -45 degrees followed by 5 symbols with +90 degrees shift relative to the previous symbol followed by another 5 symbols where the relative shift is 180 degrees. The preamble is concluded by a 13 element barker code. A 31 bit long PN-sequence is used as a data sequence. The symbols are generated from the PN-sequence by assigning the sequence directly to the quadrature component and assigning the same sequence advanced by 10 bits to the in-phase component. This symbol sequence is interpolated to yield a sampling frequency of 4 kHz by filtering it through a truncated 4 symbol intervals long cosine roll-off filter with roll-off factor 0.5. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> seq4psk123
 carrier frequency (Hz):
f0 = 1000
 symbol rate (Hz):
symbrat = 3.3333
 output sampling frequency (Hz):
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 1 0
 roll-off factor:
betha = 0.5000
 (half) constraint length of roll-off filter:
corolnth = 2
 No of output symbols requested:
nsymb = 40
 Number of symbol intervals for tone :
nsil = 5
 Number of symbol intervals for 90 deg shifts :
n90 = 5
 Number of symbol intervals for 180 deg shifts :
n180 = 5
ans =7-Jan-91
 Informative remark : carrier phase and amplitude for barker corrected time 09:35
ndel=fix((2^length(genpol)-1)/3);
ngen=nsymb+ndel;
pn=pnsequ(reginit,genpol,ngen);
clear dseq
dseq=preamb4psk(nsil,n90,n180,1+sqrt(-1),sqrt(-1));
dlen=length(dseq);
dseq(1+dlen:dlen+13)=[-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1]*(1+sqrt(-1));
dlen=length(dseq);
pn=2*(pn-.5*ones(pn));
dseq(1+dlen:dlen+nsymb)=pn(1+ndel:ngen)+sqrt(-1)*pn(1:nsymb);
iprat=fix(fs/symbrat);
cr=cosroll(betha,iprat,corolnth);
Warning: Divide by zero
Warning: Divide by zero
clear oseq oseqr
pack
oseq=interpfilt(cr,iprat,dseq);
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*dlen;
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
echo off
>> head=oseqr(1:pnstrt-1);
>> body=oseqr(pnstrt:pnstrt+pnper-1);
```

```
>> save f421qp03 body
>> save f422qp03 head
>> diary off
```

4PSK 10 Hz symbolrate total duration 40.3 s:

Filename	signal type	Duration: Samples	seconds
f1313b10.mat	barker code	5200	1.3
f151ps1s.mat	silence(pause)	4000	1
f432qp10.mat	preamble (+ trans)	12801	3.2
f431qp10.mat	1 period data seq.	25200	6.3
f431qp10.mat	1 period data seq.	25200	6.3
f431qp10.mat	1 period data seq.	25200	6.3
f431qp10.mat	1 period data seq.	25200	6.3
f431qp10.mat	1 period data seq.	25200	6.3
f151ps1s.mat	silence(pause)	4000	1
f1313b10.mat	barker code	5200	1.3
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seq4psk123.m". The preamble consists of 5 symbol interval of the carrier with phase -45 degrees followed by 5 symbols with +90 degrees shift relative to the previous symbol followed by another 5 symbols where the relative shift is 180 degrees. The preamble is concluded by a 13 element barker code. A 63 bit long PN-sequence is used as a data sequence. The symbols are generated from the PN-sequence by assigning the sequence directly to the quadrature component and assigning the same sequence advanced by 21 bits to the in-phase component. This symbol sequence is interpolated to yield a sampling frequency of 4 kHz by filtering it through a truncated 4 symbol intervals long cosine roll-off filter with roll-off factor 0.5. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> seq4psk123
 carrier frequency (Hz):
f0 = 1000
 symbol rate (Hz):
symbrat = 10
 output sampling frequency (Hz):
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 1
 roll-off factor:
betha = 0.5000
 (half) constraint length of roll-off filter:
corolnth = 2
 No of output symbols requested:
nsymb = 70
 Number of symbol intervals for tone :
nsil = 5
 Number of symbol intervals for 90 deg shifts :
n90 = 5
 Number of symbol intervals for 180 deg shifts :
n180 = 5
```

```
ans =7-Jan-91
 Informative remark : 10 Hz pn(p=63) preamb 5+5+5+bark time 10:15
ndel=fix((2^length(genpol)-1)/3);
ngen=nsymb+ndel;
pn=pnsequ(reginit,genpol,ngen);
clear dseq
dseq=preamb4psk(nsil,n90,n180,1+sqrt(-1),sqrt(-1));
dlen=length(dseq);
dseq(1+dlen:dlen+13)=[-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1]*(1+sqrt(-1));
dlen=length(dseq);
pn=2*(pn-.5*ones(pn));
dseq(1+dlen:dlen+nsymb)=pn(1+ndel:ngen)+sqrt(-1)*pn(1:nsymb);
iprat=fix(fs/symbrat);
cr=cosroll(betha,iprat,corolnth);
Warning: Divide by zero
Warning: Divide by zero
clear oseq oseqr
pack
oseq=interpfilt(cr,iprat,dseq);
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*dlen;
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
echo off
>> head=oseqr(1:pnstrt-1);
>> body=oseqr(pnstrt:pnstrt+pnper-1);
>> save f431qp10 body
>> save f432qp10 head
>> diary off
```

4PSK 33.3 Hz symbolrate total duration 43.4 s:

Filename	signal type	Duration: Samples	seconds
f1314b30.mat	barker code	1560	0.39
f151ps1s.mat	silence(pause)	4000	1
f442qp30.mat	preamble (+ trans)	5641	1.41
f441qp30.mat	1 period data seq.	30600	7.65
f441qp30.mat	1 period data seq.	30600	7.65
f441qp30.mat	1 period data seq.	30600	7.65
f441qp30.mat	1 period data seq.	30600	7.65
f151ps1s.mat	silence(pause)	4000	1
f1314b30.mat	barker code	1560	0.39
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seq4psk123.m". The preamble consists of 10 symbol interval of the carrier with phase -45 degrees followed by 10 symbols with +90 degrees shift relative to the previous symbol followed by another 10 symbols where the relative shift is 180 degrees. The preamble is concluded by a 13 element barker code. A 255 bit long PN-sequence is used as a data sequence. The symbols are generated from the PN-sequence by assigning the sequence directly to the quadrature component and assigning the same sequence advanced by 63 bits to the in-phase component. This symbol sequence is interpolated to yield a sampling frequency of 4 kHz by filtering it through a truncated 4 symbol intervals long cosine roll-off filter with roll-off factor 0.5. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init123
>> seq4psk123
f0 = 1000
 symbol rate (Hz):
symbrat = 33.3300
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 1 1 1 0
betha = 0.5000
corolnth = 2
 No of output symbols requested:
nsymb = 300
 Number of symbol intervals for tone :
nsil = 10
 Number of symbol intervals for 90 deg shifts :
n90 = 10
 Number of symbol intervals for 180 deg shifts :
n180 = 10
ans =7-Jan-91
 Informative remark : 33.33Hz pn(p=255) preamb 10 +10+10+bark time 10:35
ndel=fix((2^length(genpol)-1)/3);
ngen=nsymb+ndel;
pn=pnsequ(reginit,genpol,ngen);
clear dseq
dseq=preamb4psk(nsil,n90,n180,1+sqrt(-1),sqrt(-1));
dlen=length(dseq);
dseq(1+dlen:dlen+13)=[-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1]*(1+sqrt(-1));
dlen=length(dseq);
pn=2*(pn-.5*ones(pn));
dseq(1+dlen:dlen+nsymb)=pn(1+ndel:ngen)+sqrt(-1)*pn(1:nsymb);
iprat=fix(fs/symbrat);
cr=cosroll(betha,iprat,corolnth);
Warning: Divide by zero
Warning: Divide by zero
clear oseq oseqr
pack
oseq=interpfilt(cr,iprat,dseq);
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*dlen;
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
echo off
>> head=oseqr(1:pnstrt-1);
>> body=oseqr(pnstrt:pnstrt+pnper-1);
>> save f441qp30 body
>> save f442qp30 head
>> diary off
```

4PSK 100 Hz symbolrate total duration 29.3 s:

Filename	signal type	Duration: Samples	seconds
f1315b1h.mat	barker code	520	0.13
f151ps1s.mat	silence(pause)	4000	1
f452qp1h.mat	preamble (+ trans)	1881	0.47
f451qp1h.mat	1 period data seq.	20440	5.11
f451qp1h.mat	1 period data seq.	20440	5.11
f451qp1h.mat	1 period data seq.	20440	5.11
f451qp1h.mat	1 period data seq.	20440	5.11
f451qp1h.mat	1 period data seq.	20440	5.11
f151ps1s.mat	silence(pause)	4000	1
f1315b1h.mat	barker code	520	0.13
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seq4psk123.m". The preamble consists of 10 symbol interval of the carrier with phase -45 degrees followed by 10 symbols with +90 degrees shift relative to the previous symbol followed by another 10 symbols where the relative shift is 180 degrees. The preamble is concluded by a 13 element barker code. A 511 bit long PN-sequence is used as a data sequence. The symbols are generated from the PN-sequence by assigning the sequence directly to the quadrature component and assigning the same sequence advanced by 170 bits to the in-phase component. This symbol sequence is interpolated to yield a sampling frequency of 4 kHz by filtering it through a truncated 4 symbol intervals long cosine roll-off filter with roll-off factor 0.5. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init123
>> seq4psk123
f0 = 1000
 symbol rate (Hz):
symbrat = 100
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 1 0 0 0
betha = 0.5000
corolnth = 2
 No of output symbols requested:
nsymb = 600
 Number of symbol intervals for tone :
nsil = 10
 Number of symbol intervals for 90 deg shifts :
n90 = 10
 Number of symbol intervals for 180 deg shifts :
n180 = 10
ans =7-Jan-91
 Informative remark : 100Hz 4psk pn(p=511) preamb 10+10+10+bark time 10:45
ndel=fix((2^length(genpol)-1)/3);
ngen=nsymb+ndel;
pn=pnsequ(reginit,genpol,ngen);
clear dseq
dseq=preamb4psk(nsil,n90,n180,1+sqrt(-1),sqrt(-1));
dlen=length(dseq);
dseq(1+dlen:dlen+13)=[-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1]*(1+sqrt(-1));
dlen=length(dseq);
```

```
pn=2*(pn-.5*ones(pn));
dseq(1+dlen:dlen+nsymb)=pn(1+ndel:ngen)+sqrt(-1)*pn(1:nsymb);
iprat=fix(fs/symbrat);
cr=cosroll(betha,iprat,corolnth);
Warning: Divide by zero
Warning: Divide by zero
clear oseq oseqr
pack
oseq=interpfilt(cr,iprat,dseq);
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*dlen;
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
echo off
>> head=oseqr(1:pnstrt-1);
>> body=oseqr(pnstrt:pnstrt-1+pnper);
>> save f451qp1h body
>> save f452qp1h head
>> diary off
```

4PSK 333 Hz symbolrate total duration 33.9 s:

Filename	signal type	Duration: Samples	seconds
f1316b3h.mat	barker code	156	0.04
f151ps1s.mat	silence(pause)	4000	1
f462qp3h.mat	preamble (+ trans)	565	0.14
f461qp3h.mat	1 period data seq.	24564	6.14
f461qp3h.mat	1 period data seq.	24564	6.14
f461qp3h.mat	1 period data seq.	24564	6.14
f461qp3h.mat	1 period data seq.	24564	6.14
f461qp3h.mat	1 period data seq.	24564	6.14
f151ps1s.mat	silence(pause)	4000	1
f1316b3h.mat	barker code	156	0.04
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seq4psk123.m". The preamble consists of 10 symbol interval of the carrier with phase -45 degrees followed by 10 symbols with +90 degrees shift relative to the previous symbol followed by another 10 symbols where the relative shift is 180 degrees. The preamble is concluded by a 13 element barker code. A 2047 bit long PN-sequence is used as a data sequence. The symbols are generated from the PN-sequence by assigning the sequence directly to the quadrature component and assigning the same sequence advanced by 682 bits to the in-phase component. This symbol sequence is interpolated to yield a sampling frequency of 4 kHz by filtering it through a truncated 4 symbol intervals long cosine roll-off filter with roll-off factor 0.5. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init123
>> seq4psk123
f0 = 1000
 symbol rate (Hz):
symbrat = 333.3300
```

```
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 0 0 0 0 1 0
betha = 0.5000
corolnth = 2
 No of output symbols requested:
nsymb = 3000
 Number of symbol intervals for tone :
nsil = 10
 Number of symbol intervals for 90 deg shifts :
n90 = 10
 Number of symbol intervals for 180 deg shifts :
n180 = 10
ans =7-Jan-91
 Informative remark : 333 Hz 4psk pn(p=2047) pramb 3x10+bark time:11:00
ndel=fix((2^length(genpol)-1)/3);
ngen=nsymb+ndel;
pn=pnsequ(reginit,genpol,ngen);
clear dseq
dseq=preamb4psk(nsil,n90,n180,1+sqrt(-1),sqrt(-1));
dlen=length(dseq);
dseq(1+dlen:dlen+13)=[-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1]*(1+sqrt(-1));
dlen=length(dseq);
pn=2*(pn-.5*ones(pn));
dseq(1+dlen:dlen+nsymb)=pn(1+ndel:ngen)+sqrt(-1)*pn(1:nsymb);
iprat=fix(fs/symbrat);
cr=cosroll(betha,iprat,corolnth);
Warning: Divide by zero
Warning: Divide by zero
clear oseq oseqr
pack
oseq=interpfilt(cr,iprat,dseq);
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*dlen;
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
echo off
>> head=oseqr(1:pnstrt-1);
>> body=oseqr(pnstrt:pnstrt-1+pnper);
>> save f461qp3h body
>> save f462qp3h head
>> diary off
```

4PSK 1 kHz symbolrate total duration 23.5 s:

Filename	**signal type**	**Duration: Samples**	**seconds**
f1317b1k.mat	barker code	52	0.013
f151ps1s.mat	silence(pause)	4000	1
f472qp1k.mat	preamble (+ trans)	189	0.05
f471qp1k.mat	1 period data seq.	16380	4.10
f471qp1k.mat	1 period data seq.	16380	4.10
f471qp1k.mat	1 period data seq.	16380	4.10
f471qp1k.mat	1 period data seq.	16380	4.10
f471qp1k.mat	1 period data seq.	16380	4.10
f151ps1s.mat	silence(pause)	4000	1
f1317b1k.mat	barker code	52	0.013
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seq4psk123.m". The preamble consists of 10 symbol interval of the carrier with phase -45 degrees followed by 10 symbols with +90 degrees shift relative to the previous symbol followed by another 10 symbols where the relative shift is 180 degrees. The preamble is concluded by a 13 element barker code. A 4095 bit long PN-sequence is used as a data sequence. The symbols are generated from the PN-sequence by assigning the sequence directly to the quadrature component and assigning the same sequence advanced by 1365 bits to the in-phase component. This symbol sequence is interpolated to yield a sampling frequency of 4 kHz by filtering it through a truncated 4 symbol intervals long cosine roll-off filter with roll-off factor 0.5. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init123
>> seq4psk123
f0 = 1000
 symbol rate (Hz):
symbrat = 1000
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 0 1 0 1 0 0 1
betha = 0.5000
corolnth = 2
 No of output symbols requested:
nsymb = 6000
 Number of symbol intervals for tone :
nsil = 10
 Number of symbol intervals for 90 deg shifts :
n90 = 10
 Number of symbol intervals for 180 deg shifts :
n180 = 10
ans =7-Jan-91
 Informative remark : 1kHz 4psk pn(p=4095) preamb 3x10+bark time 11:15
ndel=fix((2^length(genpol)-1)/3);
ngen=nsymb+ndel;
pn=pnsequ(reginit,genpol,ngen);
clear dseq
dseq=preamb4psk(nsil,n90,n180,1+sqrt(-1),sqrt(-1));
dlen=length(dseq);
dseq(1+dlen:dlen+13)=[-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1]*(1+sqrt(-1));
dlen=length(dseq);
```

```
pn=2*(pn-.5*ones(pn));
dseq(1+dlen:dlen+nsymb)=pn(1+ndel:ngen)+sqrt(-1)*pn(1:nsymb);
iprat=fix(fs/symbrat);
cr=cosroll(betha,iprat,corolnth);
Warning: Divide by zero
Warning: Divide by zero
clear oseq oseqr
pack
oseq=interpfilt(cr,iprat,dseq);
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*dlen;
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
echo off
>> head=oseqr(1:pnstrt-1);
>> body=oseqr(pnstrt:pnstrt-1+pnper);
>> save f471qp1k body
>> save f472qp1k head
>> plot(head)
>> plot(body)
>> exit
 1542916 flops.
```

8QAM 333 Hz symbolrate total duration 1 min 4.6 s:

Filename	**signal type**	**Duration: Samples**	**seconds**
f1316b3h.mat	barker code	156	0.04
f151ps1s.mat	silence(pause)	4000	1
f462qp3h.mat	preamble (+ trans)	565	0.14
f561qa3h.mat	1 period data seq.	49140	12.29
f561qa3h.mat	1 period data seq.	49140	12.29
f561qa3h.mat	1 period data seq.	49140	12.29
f561qa3h.mat	1 period data seq.	49140	12.29
f561qa3h.mat	1 period data seq.	49140	12.29
f151ps1s.mat	silence(pause)	4000	1
f1316b3h.mat	barker code	156	0.04
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seq4psk123.m". The preamble is the same as for the 4PSKsequence of the same bitrate and is described above. A 4095 bit long PN-sequence is used as a data sequence. The symbols are generated from the PN-sequence by assigning groups of 3 adjacent bits to an 8QAM symbol. This symbol sequence is interpolated to yield a sampling frequency of 4 kHz by filtering it through a truncated 4 symbol intervals long cosine roll-off filter with roll-off factor 0.5. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init123
>> seq8qam123
f0 = 1000
 symbol rate (Hz):
symbrat = 333.3000
fs = 4000
 coefficients of generating function:
```

```
genpol = 1 0 0 0 0 0 1 0 1 0 0 1
betha = 0.5000
corolnth = 2
 No of output symbols requested:
nsymb = 5000
ans =9-Jan-91
 Informative remark : 1st attempt 8qam 333Hz pn(p=4095) time 16:00
nbit=nsymb*3;
pn=pnsequ(reginit,genpol,nbit);
clear dseq
dseq=map8qam(pn(1:3:nbit),pn(2:3:nbit),pn(3:3:nbit));
iprat=fix(fs/symbrat);
cr=cosroll(betha,iprat,corolnth);
Warning: Divide by zero
Warning: Divide by zero
clear oseq oseqr body
pack
oseq=interpfilt(cr,iprat,dseq);
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
 How many pn-periods?
 nper=1;
>> body=oseqr(pnstrt:pnstrt-1+nper*pnper);
>> save f561qa3h body
>> diary off
```

8QAM 1 Hz symbolrate total duration 2 min. 16 s

Filename	**signal type**	**Duration: Samples**	**seconds**
f1311b01.mat	barker code	52000	13
f151ps1s.mat	silence(pause)	4000	1
f412qp01.mat	preamble (+ trans)	128001	31
f511qa01.mat	1 period data seq.	60000	15
f511qa01.mat	1 period data seq.	60000	15
f511qa01.mat	1 period data seq.	60000	15
f511qa01.mat	1 period data seq.	60000	15
f511qa01.mat	1 period data seq.	60000	15
f151ps1s.mat	silence(pause)	4000	4
f1311b01.mat	barker code	52000	13
f151ps1s.mat	silence(pause)	4000	4

The files containing the preamble and the data sequence were generated by the matlab m-file "seq8qam123.m". The preamble is the same as for the 4PSKsequence of the same bitrate and is described above. A 15 bit long PN-sequence is used as a data sequence. The symbols are generated from the PN-sequence by assigning groups of 3 adjacent bits to an 8QAM symbol. This symbol sequence is interpolated to yield a sampling frequency of 4 kHz by filtering it through a truncated 4 symbol intervals long cosine roll-off filter with roll-off factor 0.5. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init123
```

```
>> seq8qam123
f0 = 1000
 symbol rate (Hz):
symbrat = 1
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 1
betha = 0.5000
corolnth = 2
 No of output symbols requested:
nsymb = 20
ans =12-Jan-91
 Informative remark : 8qam 1Hz pn(p=15) time 13:10
nbit=nsymb*3;
pn=pnsequ(reginit,genpol,nbit);
clear dseq
dseq=map8qam(pn(1:3:nbit),pn(2:3:nbit),pn(3:3:nbit));
iprat=fix(fs/symbrat);
cr=cosroll(betha,iprat,corolnth);
Warning: Divide by zero
Warning: Divide by zero
clear oseq oseqr body
pack
oseq=interpfilt(cr,iprat,dseq);
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
???? Error using ==> *
Out of memory. Type HELP MEMORY for your options.
>> cw=exp((sqrt(-1)*2*pi*(f0/fs))*(0:999));
>> for k=1:80,oseq(k*1000-999:k*1000)=oseq(k*1000-999:k*1000).*cw;end
>> pnstrt=1+length(cr);
>> pnper=(2^length(genpol)-1)*iprat;
>> oseqr=real(oseq);
>> oseqr=oseqr/max(abs(oseqr));
 How many pn-periods? >> nper
nper = 1
>> body=oseqr(pnstrt:pnstrt-1+nper*pnper);
>> save f511qa01 body
>> exit
 3310806 flops.
```

8QAM 3.33 Hz symbolrate total duration 1 min. 54.9 s:

Filename	signal type	Duration: Samples	seconds
f1312b03.mat	barker code	15600	3.9
f151ps1s.mat	silence(pause)	4000	1
f422qp03.mat	preamble (+ trans)	38401	9.6
f521qa03.mat	1 period data seq.	75600	18.9
f521qa03.mat	1 period data seq.	75600	18.9
f521qa03.mat	1 period data seq.	75600	18.9
f521qa03.mat	1 period data seq.	75600	18.9
f521qa03.mat	1 period data seq.	75600	18.9
f151ps1s.mat	silence(pause)	4000	1
f1312b03.mat	barker code	15600	3.9
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seq8qam123.m". The preamble is the same as for the 4PSKsequence of the same bi-trate and is described above. A 63 bit long PN-sequence is used as a data sequence. The symbols are generated from the PN-sequence by assigning groups of 3 adjacent bits to an 8QAM symbol. This symbol sequence is interpolated to yield a sampling frequency of 4 kHz by filtering it through a truncated 4 symbol intervals long cosine roll-off filter with roll-off factor 0.5. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init123
>> seq8qam123
f0 = 1000
 symbol rate (Hz):
symbrat = 3.3333
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 1
betha = 0.5000
corolnth = 2
 No of output symbols requested:
nsymb = 70
ans =15-Jan-91
 Informative remark : 3.333 Hz 8qam pn(p=63) time 11:50
nbit=nsymb*3;
pn=pnsequ(reginit,genpol,nbit);
clear dseq
dseq=map8qam(pn(1:3:nbit),pn(2:3:nbit),pn(3:3:nbit));
iprat=fix(fs/symbrat);
cr=cosroll(betha,iprat,corolnth);
Warning: Divide by zero
Warning: Divide by zero
clear oseq oseqr body
pack
oseq=interpfilt(cr,iprat,dseq);
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
 How many pn-periods? nper
nper = 1
body=oseqr(pnstrt:pnstrt-1+nper*pnper);
echo off
>> save f521qa03 body
>> exit
 28815522 flops.
```

8QAM 10 Hz symbolrate total duration 2 min 16.3 s:

Filename	signal type	Duration: Samples	seconds
f1313b10.mat	barker code	5200	1.3
f151ps1s.mat	silence(pause)	4000	1
f432qp10.mat	preamble (+ trans)	12801	3.2
f531qa10.mat	1 period data seq.	102000	25.5
f531qa10.mat	1 period data seq.	102000	25.5
f531qa10.mat	1 period data seq.	102000	25.5
f531qa10.mat	1 period data seq.	102000	25.5
f531qa10.mat	1 period data seq.	102000	25.5
f151ps1s.mat	silence(pause)	4000	1
f1313b10.mat	barker code	5200	1.3
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seq8qam123.m". The preamble is the same as for the 4PSKsequence of the same bitrate and is described above. A 255 bit long PN-sequence is used as a data sequence. The symbols are generated from the PN-sequence by assigning groups of 3 adjacent bits to an 8QAM symbol. This symbol sequence is interpolated to yield a sampling frequency of 4 kHz by filtering it through a truncated 4 symbol intervals long cosine roll-off filter with roll-off factor 0.5. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init123
>> seq8qam123
f0 = 1000
 symbol rate (Hz):
symbrat = 10
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 1 1 1 0
betha = 0.5000
corolnth = 2
 No of output symbols requested:
nsymb = 270
ans =12-Jan-91
 Informative remark : 8qam 10 Hz pn(p=255) time 12:10
nbit=nsymb*3;
pn=pnsequ(reginit,genpol,nbit);
clear dseq
dseq=map8qam(pn(1:3:nbit),pn(2:3:nbit),pn(3:3:nbit));
iprat=fix(fs/symbrat);
cr=cosroll(betha,iprat,corolnth);
Warning: Divide by zero
Warning: Divide by zero
clear oseq oseqr body
pack
oseq=interpfilt(cr,iprat,dseq);
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
 How many pn-periods? nper
```

```
nper = 1
body=oseqr(pnstrt:pnstrt-1+nper*pnper);
echo off
>> save f531qa10 body
```

8QAM 33.3 Hz symbolrate total duration 2 min. 38.6 s:

Filename	signal type	Duration: Samples	seconds
f1314b30.mat	barker code	1560	0.39
f151ps1s.mat	silence(pause)	4000	1
f442qp30.mat	preamble (+ trans)	5641	1.41
f541qa30.mat	1 period data seq.	122760	30.69
f541qa30.mat	1 period data seq.	122760	30.69
f541qa30.mat	1 period data seq.	122760	30.69
f541qa30.mat	1 period data seq.	122760	30.69
f541qa30.mat	1 period data seq.	122760	30.69
f151ps1s.mat	silence(pause)	4000	1
f1314b30.mat	barker code	1560	0.39
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seq8qam123.m". The preamble is the same as for the 4PSKsequence of the same bi-trate and is described above. A 1023 bit long PN-sequence is used as a data sequence. The symbols are generated from the PN-sequence by assigning groups of 3 adjacent bits to an 8QAM symbol. This symbol sequence is interpolated to yield a sampling frequency of 4 kHz by filtering it through a truncated 4 symbol intervals long cosine roll-off filter with roll-off factor 0.5. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init123
>> seq8qam123
f0 = 1000
 symbol rate (Hz):
symbrat = 33.3300
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 0 0 1 0 0
betha = 0.5000
corolnth = 2
 No of output symbols requested:
nsymb = 1100
ans =11-Jan-91
 Informative remark : 33.33 Hz 8qam pn(p=1023) time 13:00
nbit=nsymb*3;
pn=pnsequ(reginit,genpol,nbit);
clear dseq
dseq=map8qam(pn(1:3:nbit),pn(2:3:nbit),pn(3:3:nbit));
iprat=fix(fs/symbrat);
cr=cosroll(betha,iprat,corolnth);
Warning: Divide by zero
Warning: Divide by zero
clear oseq oseqr body
pack
oseq=interpfilt(cr,iprat,dseq);
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
```

```
???? Out of memory. Type HELP MEMORY for your options.
>> cw=exp(sqrt(-1)*2*pi*(f0/fs)*[0:999]);
>> for k=1:132,oseq(k*1000-999:k*1000)=oseq(k*1000-999:k*1000).*cw;end
>> pnstrt=1+length(cr);
>> pnper=(2^length(genpol)-1)*iprat;
>> oseqr=real(oseq);
>> oseqr=oseqr/max(abs(oseqr));
???? Error using ==> abs
Out of memory. Type HELP MEMORY for your options.
>> clear oseq
>> oseqr=oseqr/max(abs(oseqr));
>> nper=input('How many pn-periods? ');
nper = 1
>> body=oseqr(pnstrt:pnstrt-1+nper*pnper);
>> save f541qa30 body
>> diary off
```

8QAM 100 Hz symbolrate total duration 54.9 s:

Filename	signal type	Duration: Samples	seconds
f1315b1h.mat	barker code	520	0.13
f151ps1s.mat	silence(pause)	4000	1
f452qp1h.mat	preamble (+ trans)	1881	0.47
f551qa1h.mat	1 period data seq.	40920	10.23
f551qa1h.mat	1 period data seq.	40920	10.23
f551qa1h.mat	1 period data seq.	40920	10.23
f551qa1h.mat	1 period data seq.	40920	10.23
f551qa1h.mat	1 period data seq.	40920	10.23
f151ps1s.mat	silence(pause)	4000	1
f1315b1h.mat	barker code	520	0.13
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seq8qam123.m". The preamble is the same as for the 4PSKsequence of the same bitrate and is described above. A 1023 bit long PN-sequence is used as a data sequence. The symbols are generated from the PN-sequence by assigning groups of 3 adjacent bits to an 8QAM symbol. This symbol sequence is interpolated to yield a sampling frequency of 4 kHz by filtering it through a truncated 4 symbol intervals long cosine roll-off filter with roll-off factor 0.5. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init123
>> seq8qam123
f0 = 1000
 symbol rate (Hz):
symbrat = 100
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 0 0 1 0 0
betha = 0.5000
corolnth = 2
 No of output symbols requested:
nsymb = 1500
ans =10-Jan-91
 Informative remark : 8qam pn(p=1023) time 12:52 100Hz symbol rate
```

```
nbit=nsymb*3;
pn=pnsequ(reginit,genpol,nbit);
clear dseq
dseq=map8qam(pn(1:3:nbit),pn(2:3:nbit),pn(3:3:nbit));
iprat=fix(fs/symbrat);
cr=cosroll(betha,iprat,corolnth);
Warning: Divide by zero
Warning: Divide by zero
clear oseq oseqr body
pack
oseq=interpfilt(cr,iprat,dseq);
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
???? Error using ==> *
Out of memory. Type HELP MEMORY for your options.
>> fscw=exp(sqrt(-1)*2*pi*f0/fs*[0:999]);
>> for k=1:60,oseq(k*1000-999:k*1000)=oseq(k*1000-999:k*1000).*fscw;end
>> pnstrt=1+length(cr);
>> pnper=(2^length(genpol)-1)*iprat;
>> oseqr=real(oseq);
>> oseqr=oseqr/max(abs(oseqr));
 How many pn-periods? nper = 1
>> body=oseqr(pnstrt:pnstrt-1+nper*pnper);
>> save f551qa1h body
```

8QAM 1 kHz symbolrate total duration 23.5 s:

Filename	signal type	Duration: Samples	seconds
f1317b1k.mat	barker code	52	0.013
f151ps1s.mat	silence(pause)	4000	1
f472qp1k.mat	preamble (+ trans)	189	0.05
f571qa1k.mat	1 period data seq.	16380	4.10
f571qa1k.mat	1 period data seq.	16380	4.10
f571qa1k.mat	1 period data seq.	16380	4.10
f571qa1k.mat	1 period data seq.	16380	4.10
f571qa1k.mat	1 period data seq.	16380	4.10
f151ps1s.mat	silence(pause)	4000	1
f1317b1k.mat	barker code	52	0.013
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seq8qam123.m". The preamble is the same as for the 4PSKsequence of the same bitrate and is described above. A 4095 bit long PN-sequence is used as a data sequence. The symbols are generated from the PN-sequence by assigning groups of 3 adjacent bits to an 8QAM symbol. This symbol sequence is interpolated to yield a sampling frequency of 4 kHz by filtering it through a truncated 4 symbol intervals long cosine roll-off filter with roll-off factor 0.5. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init123
>> seq8qam123
f0 = 1000
 symbol rate (Hz):
symbrat = 1000
fs = 4000
 coefficients of generating function:
```

```
genpol = 1 0 0 0 0 0 1 0 1 0 0 1
betha = 0.5000
corolnth = 2
 No of output symbols requested:
nsymb = 5000
ans =12-Jan-91
 Informative remark : 8qam 1kHz pn(p=4095) time: 13:25
nbit=nsymb*3;
pn=pnsequ(reginit,genpol,nbit);
clear dseq
dseq=map8qam(pn(1:3:nbit),pn(2:3:nbit),pn(3:3:nbit));
iprat=fix(fs/symbrat);
cr=cosroll(betha,iprat,corolnth);
Warning: Divide by zero
Warning: Divide by zero
clear oseq oseqr body
pack
oseq=interpfilt(cr,iprat,dseq);
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
nper=input('How many pn-periods? ');
nper = 1
body=oseqr(pnstrt:pnstrt-1+nper*pnper);
echo off
>> save f571qa1k body
>> diary off
```

8.0.1 Sequence 2

Details of individual waveforms are given below. including excerpts of MATLAB code used to generate the waveforms:

500 Hz sweep - 1 sec duration

1 second linear FM sweep ranging from 800 Hz to 1.3 kHz. Generating code is:

```
a = sin(2*pi/5*(1:4000) + 2*pi/(16*4000)*(1:4000).^2);
a(12000) = 0;
```

This creates a 12000 point file containing a 4000 point representation of a linear sweep. The file is repeated 30 times for a total of 90 seconds

500 Hz sweep - 5 sec duration

1 second linear FM sweep ranging from 800 Hz to 1.3 kHz. Generating code is:

```
a = sin(2*pi/5*(1:20000) + 2*pi/(16*20000)*(1:20000).^2);
a(28000) = 0;
```

This creates a 28000 point file containing a 20000 point representation of a linear sweep. The file is repeated 20 times for a total of 140 seconds.

1023 PN code - standard tomography format (4 msec chip)W signals

1023-chip differential PSK coded sequence. The modulation angle is 88.2^o (88.2^o = atan ($\sqrt{1023}$)) to yield a 'period-matched angle[3]. This waveform and modulation format is intended to be compatible with classical multipath estimation techniques employed by acoustic tomography programs. Code used to generate the transmitted sequence is:

```
seed = [0 0 0 0 0 0 0 0 0 1];
chpol =[1 0 0 0 0 0 0 3 0 0];
data = pnseq(seed, chpol, 1023);          % data returns 1023 point PN sequence
f0 = sin((pi/2)*(1:16)+ 0);               %f0 = 16 point tone burst at 1 kHz
f1 = sin((pi/2)*(1:16)+ 176.418/57.29);   %f1 = f0 delayed by modulation angle
buf = [];
for i = 1:1023
if (data(i) == 0)
        buf =[buf f0];
else
        buf = [buf f1];
end
end
buf(20000) = 0;
```

The 5 second buffer is repeated 90 times for a total of 450 seconds.

4 tones - 50 Hz apart (1000, 1050, 1100, 1150 Hz)

Four CW tones transmitted at 1000 Hz, 1050 Hz 1100 Hz 1150 Hz. Generating code:

```
fourtone = sin(2*pi/4*(1:4000)) + sin(2*pi*(1050/4000)*(1:4000)) + sin(2*pi*(1100/
4000)*(1:4000) + pi) + sin(2*pi*(1150/4000)*(1:4000));
```

The 1 sec block is repeated 180 times for a total of 180 seconds.

16 tones - 10 Hz apart (1000 - 1150 Hz)

16 CW tones transmitted in the 1000 Hz - 1150 Hz band. generating code:

```
a = zeros(1,4000);
phaselog = [];
for i = 1000:10:1150
        phase = 2*pi*rand;
        phaselog = [phaselog phase];
        a = a + sin(2*pi*(i/4000)*(1:4000) + phase);
end
```

The 1 sec block is repeated 180 times for a total of 180 seconds.

64 tones - 10 Hz apart (760 - 1390 Hz)

64 CW tones transmitted in the760 Hz -1390 Hz band. Generating code:

```
sixfourtone = zeros(1:4000);
phaselog = [];
for i = 760:10:1390
        phase = 2*pi*rand;
```

```
        phaselog = [phaselog phase];
        sixfourtone = sixfourtone + sin(2*pi*(i/4000)*(1:4000) + phase);
  end
```

The 1 sec block is repeated 180 times for a total of 180 seconds.

FSK - 1 bit/sec - 100 Hz spacing (1000 - 1100 Hz)

This file contains a PN length 15 sequence modulated by a FSK carrier and transmitted at 1 baud. Generating code is:

```
seed = [0 0 0 1];
chpol =[1 0 0 1];
data = pnseq(seed, chpol, 15);
data(16) = 1;
synch = sin(2*pi/5*(1:4000) + 2*pi/(16*4000)*(1:4000).^2);
f0 = sin(2*pi/4*(1:4000));
f1 = sin(2*pi*1100/4000*(1:4000));
buf = [];
for j = 1:16
j
 if (data(j) == 0)
        buf = [buf f0];
 else
        buf = [buf f1];
end
if (rem(j,8) == 0)
        buf = [buf synch];
end
end
```

A 1 sec 500 Hz sweep is inserted every 8 seconds for synchronization purposes. Resultant file length is 72000 points (18 sec). It is transmitted 10 times for a total duration of 180 sec.

4FSK - 4 bits/sec 50 Hz spacing (850 - 1150 Hz)

This file contains a PN length 64 sequence modulated by 4FSK carriers and transmitted at 1 baud. Generating code is:

```
seed = [0 0 0 0 0 1];
chpol =[1 0 0 0 0 1];
data = pnseq(seed, chpol, 63);
data(64) = 1;                                    %last data symbol
synch = sin(2*pi/5*(1:4000) + 2*pi/(16*4000)*(1:4000).^2);
buf1 = [];
j = 1;
for k = 1:16
buf = zeros(1,4000);
for i=1:4
 freq = 2*i + data(j)
 j = j+1;
 buf = buf + sin(2*pi*(800+freq*50)/4000*(1:4000));
end
buf1 = [buf1 buf];
```

```
if (rem(k,8) == 0)
        bufl = [bufl synch];
end
end
```

A 1 sec 500 Hz sweep is inserted every 8 seconds for synchronization purposes. Resultant file length is 72000 points (18 sec). It is transmitted 10 times for a total duration of 180 sec.

16FSK - 16 bits/sec- 20 Hz spacing (760 - 1400 Hz)

This file contains a PN length 511 sequence modulated by 16FSK carriers and transmitted at 1 baud. Generating code is:

```
seed = [0 0 0 0 0 0 0 0 1];
chpol =[1 0 0 0 0 1 0 0 0];
data = pnseq(seed, chpol, 511);
synch = sin(2*pi/5*(1:4000) + 2*pi/(16*4000)*(1:4000).^2);
bufl = [];
j = 1;
for kk = 1:3
kk
for k=1:8
buf = zeros(1,4000);
for i=1:16
        freq = 2*i + data(j);
        j = j+1;
        buf= buf + sin(((2*pi*(740+(freq*20)))/4000)*(1:4000));
end
bufl = [bufl buf];
end
bufl = [bufl synch];
end
```

A 1 sec 500 Hz sweep is inserted every 8 seconds for synchronization purposes. Resultant file length is 72000 points (18 sec). It is transmitted 10 times for a total duration of 180 sec.

32FSK -32 bits/sec -10 Hz spacing

This file contains a PN length 511 sequence modulated by 32FSK carriers and transmitted at 1 baud. Generating code is:

```
seed = [0 0 0 0 0 0 0 0 0 1];
chpol =[1 0 0 0 0 0 0 3 0 0];
data = pnseq(seed, chpol, 1023);
synch = sin(2*pi/5*(1:4000) + 2*pi/(16*4000)*(1:4000).^2);
bufl = [];
j = 1;
for kk = 1:3
kk
```

```
for k=1:8
buf = zeros(1,4000);
for i=1:32
freq = 2*i + data(j);
j = j+1;
buf = buf + sin(((2*pi*(740+freq*10))/4000)*(1:4000));
end
buf1 = [buf1 buf];
end
buf1 = [buf1 synch];
end
```

A 1 sec 500 Hz sweep is inserted every 8 seconds for synchronization purposes. Resultant file length is 72000 points (18 sec). It is transmitted 10 times for a total duration of 180 sec.

FSK - 10 bit/sec - 50 Hz spacing

This file contains a PN length 1023 sequence modulated by an FSK carrier and transmitted at 10 baud. Generating code is:

```
seed = [0 0 0 0 0 0 0 0 0 1];
chpol =[1 0 0 0 0 0 0 1 0 0];
data = pnseq(seed, chpol, 1023);
synch = sin(2*pi/5*(1:4000) + 2*pi/(16*4000)*(1:4000).^2);
f0 = sin(2*pi/4*(1:400));
f1 = sin(2*pi*1100/4000*(1:400));
buf = [];
for j = 1:160
j
 if (data(j) == 0)
        buf = [buf f0];
 else
        buf = [buf f1];
end
if (rem(j,40) == 0)
        buf = [buf synch];
end
```

A 1 sec 500 Hz sweep is inserted every 8 seconds for synchronization purposes. Resultant file length is 80000 points. It is transmitted 8 times for a total duration of 160 sec.

4FSK - 40 bits/sec- 50 Hz spacing

This file contains a PN length 1023 sequence modulated by 4FSK carriers and transmitted at 10 baud. Generating code is:

```
seed = [0 0 0 0 0 0 0 0 0 1];
chpol =[1 0 0 0 0 0 0 1 0 0];
data = pnseq(seed, chpol, 1023);
synch = sin(2*pi/5*(1:4000) + 2*pi/(16*4000)*(1:4000).^2);
buf1 = [];
j = 1;
for k = 1:160
```

```
buf = zeros(1,400);
for i=1:4
 freq = 2*i + data(j);
 j = j+1;
 if (j > 1023)
        j = 1;
 end
 buf = buf + sin(2*pi*(800+freq*50)/4000*(1:400));
end
buf1 = [buf1 buf];
if (rem(k,80) == 0)
        buf1 = [buf1 synch];
end
end
```

A 1 sec 500 Hz sweep is inserted every 8 seconds for synchronization purposes. Resultant file length is 72000 points (18 sec). It is transmitted 10 times for a total duration of 180 sec.

16FSK - 160 bits/sec - 20 Hz spacing

This file contains a PN length 1023 sequence modulated by 16FSK carriers and transmitted at 10 baud. Generating code is:

```
seed = [0 0 0 0 0 0 0 0 0 1];
chpol =[1 0 0 0 0 0 0 1 0 0];
data = pnseq(seed, chpol, 1023);
synch = sin(2*pi/5*(1:4000) + 2*pi/(16*4000)*(1:4000).^2);
buf1 = [];
j = 1;
for kk = 1:3
kk
for k=1:80
buf = zeros(1,400);
for i=1:16
freq = 2*i + data(j);
j = j+1;
if (j>1023)
        j = 1;
end
buf = buf + sin(((2*pi*(740+freq*20))/4000)*(1:400));
end
buf1 = [buf1 buf];
end
buf1 = [buf1 synch];
end
```

A 1 sec 500 Hz sweep is inserted every 8 seconds for synchronization purposes. Resultant file length is 108000 points. It is transmitted 6 times for a total duration of 162 sec.

32 FSK - 320 bits/sec -10 Hz spacing

This file contains a PN length 1023 sequence modulated by 16FSK carriers and transmitted at 10 baud. Generating code is:

```
seed = [0 0 0 0 0 0 0 0 0 1];
```

```
chpol =[1 0 0 0 0 0 0 1 0 0];
data = pnseq(seed, chpol, 1023);
synch = sin(2*pi/5*(1:4000) + 2*pi/(16*4000)*(1:4000).^2);
buf1 = [];
j = 1;
for kk = 1:3
kk
for k=1:80
buf = zeros(1,400);
for i=1:32
freq = 2*i + data(j);
j = j+1;
if (j>1023)
        j = 1;
end
buf = buf + sin(((2*pi*(740+freq*10))/4000)*(1:400));
end
buf1 = [buf1 buf];
end
buf1 = [buf1 synch];
end
```

A 1 sec 500 Hz sweep is inserted every 8 seconds for synchronization purposes. Resultant file length is 108000 points. It is transmitted 6 times for a total duration of 162 sec.

Duobinary FSK: 20 bit/sec

This file contains a PN length 511 sequence modulated with duobinary FSK modulation at 20 baud. The band-shaping filter is a 1/2 cycle cosine waveform with nulls at 1005 and 1025 Hz. Generating code is:

```
f0 = sin(2*pi*1010/4000*(1:200));
f1 = sin(2*pi*1020/4000*(1:200));

seed = [0 0 0 0 0 0 0 0 1];
chpol =[1 0 0 0 0 1 0 0 0];
data = pnseq(seed, chpol, 511);

P(1) = data(1);
for i = 2:511
P(i) = abs(rem((data(i) - P(i-1)),2));
end

if (P(1) == 0)
        buf = f0;
else
        buf = f1;
end

k=1;

for j = 2:511
 if (P(j) == 0)
```

```
        if (buf(length(buf)-1) < 0)
                                    buf = [buf f0];
        else
                                    buf = [buf -f0];
        end
else
        if (buf(length(buf)-1) < 0)
                                    buf = [buf f1];
        else
                                    buf = [buf -f1];
        end
end
[j data(j) P(j)]

end
f_buf = fft(buf,2^17);
f_res = f_buf.*f_filt;
result = real(ifft(f_res,2^17));          %filter the modulated seq. with cos. filter
result = result/max(result);

synch = 0.5*(sin(2*pi/5*(1:4000) + 2*pi/(16*4000)*(1:4000).^2))';

for i = 1001:25600:100000
result(i:i+3999) = result(i:i+3999) + synch';
end
```

A 1 sec 500 Hz sweep is added to the sequence 6.15 sec for synchronization purposes. Resultant file length is 140000 points. It is transmitted 6times for a total duration of 210 sec.

200 bit/sec FSK 1 kHz and 1.1 kHz

This file contains a PN length 511 sequence modulated with duobinary FSK modulation at 200 baud. The band-shaping filter is a 1/2 cycle cosine waveform with nulls at 1050 and 1250 Hz. Generating code is:

```
f0 = sin(2*pi*1100/4000*(1:20));
f1 = sin(2*pi*1200/4000*(1:20));

seed = [0 0 0 0 0 0 0 0 0 1];
chpol =[1 0 0 0 0 0 0 1 0 0];
data = pnseq(seed, chpol, 1023);

P(1) = data(1);
for i = 2:1023
P(i) = abs(rem((data(i) - P(i-1)),2));
end

if (P(1) == 0)
        buf = f0;
else
        buf = f1;
end

k=1;
```

```
for j = 2:1023
 if (P(j) == 0)
        if (buf(length(buf)-1) < 0)
                buf = [buf f0];
        else
                buf = [buf -f0];
        end
else
        if (buf(length(buf)-1) < 0)
                buf = [buf f1];
        else
                buf = [buf -f1];
        end
 end
[j data(j) P(j)]

end
buf = [zeros(1,1000) buf zeros(1:1000) buf];

buf = [buf buf buf];

f_buf = fft(buf,2^17);
f_res = f_buf.*f_filt;
result = real(ifft(f_res,2^17));
result = result/max(result);

synch = 0.25*(sin(2*pi/5*(1:400) + 2*pi/(16*400)*(1:400).^2))';

for i = 1001:25600:140000
result(i:i+399) = result(i:i+399) + synch';
end
```

A 1 sec 500 Hz sweep is added to the sequence 6.15 sec for synchronization purposes. Resultant file length is 140000 points. It is transmitted 6times for a total duration of 210 sec.

8.1 Sequence 3

(sequence 16 through 21)

	2DPM4 3.33 hz symbolrate total duration 6 min. 39.9 s:		
Filename	**signal type**	**Duration: Samples**	**seconds**
f1312b03.mat	barker code	15600	3.9
f151ps1s.mat	silence(pause)	4000	1
f722dp03.mat	preamble (+ trans)	44400	11.1
f721dp03.mat	1 period data seq.	75600	18.9
f721dp03.mat	1 period data seq.	75600	18.9
f721dp03.mat	1 period data seq.	75600	18.9
f721dp03.mat	1 period data seq.	75600	18.9
f721dp03.mat	1 period data seq.	75600	18.9
f721dp03.mat	1 period data seq.	75600	18.9
f721dp03.mat	1 period data seq.	75600	18.9
f721dp03.mat	1 period data seq.	75600	18.9
f721dp03.mat	1 period data seq.	75600	18.9
f721dp03.mat	1 period data seq.	75600	18.9
f721dp03.mat	1 period data seq.	75600	18.9
f721dp03.mat	1 period data seq.	75600	18.9
f721dp03.mat	1 period data seq.	75600	18.9
f721dp03.mat	1 period data seq.	75600	18.9
f721dp03.mat	1 period data seq.	75600	18.9
f721dp03.mat	1 period data seq.	75600	18.9
f721dp03.mat	1 period data seq.	75600	18.9
f721dp03.mat	1 period data seq.	75600	18.9
f721dp03.mat	1 period data seq.	75600	18.9
f721dp03.mat	1 period data seq.	75600	18.9
f151ps1s.mat	silence(pause)	4000	1
f1312b03.mat	barker code	15600	3.9
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqdpm123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. A BPSK-modulated 13-element Barker code preceeds the preamble. A 63 bit long PN-sequence is used as a data sequence and fed to the DPM-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. The DPM-modulator has a 4RC (raised cosine) prefilter and the modulation index was 0.8. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> initdpm123
>> seqdpm123
f0 = 1000
 symbol rate (Hz):
symbrat = 3.3333
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 1
h = 0.8000
fidur = 4
 No of output symbols requested:
nsymb = 130
 Number of symbol intervals for tone :
nsil = 10
 Number of symbol intervals for clock sync :
nalt = 10
```

```
ans =21-Jan-91
 Informative remark : 3.333 hz 2dpm4 h=.8 pn(63) time:11:55
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=normdpmrc(fidur,iprat);
clear oseq oseqr body
pack
oseq=[ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat) dpm(dseq,h,cr,iprat)];
???? Out of memory. Type HELP MEMORY for your options.
>> oseq=ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat);
>> oseq=[oseq dpm(dseq(1:100),h,cr,iprat)];
>> oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
>> pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
>> pnper=(2^length(genpol)-1)*iprat;
>> oseqr=real(oseq);
>> oseqr=oseqr/max(abs(oseqr));
>> body=oseqr(pnstrt:pnstrt-1+pnper);
>> head=oseqr(1:pnstrt-1);
>> save f721dp03 body
>> save f722dp03 head
>> exit
 32113859 flops.
```

2DPM4 10 hz symbolrate total duration 2 min 15.3 s:

Filename	signal type	Duration: Samples	seconds
f1313b10.mat	barker code	5200	1.3
f151ps1s.mat	silence(pause)	4000	1
f732dp10.mat	preamble (+ trans)	14800	3.7
f731dp10.mat	1 period data seq.	25200	6.3
f731dp10.mat	1 period data seq.	25200	6.3
f731dp10.mat	1 period data seq.	25200	6.3
f731dp10.mat	1 period data seq.	25200	6.3
f731dp10.mat	1 period data seq.	25200	6.3
f731dp10.mat	1 period data seq.	25200	6.3
f731dp10.mat	1 period data seq.	25200	6.3
f731dp10.mat	1 period data seq.	25200	6.3
f731dp10.mat	1 period data seq.	25200	6.3
f731dp10.mat	1 period data seq.	25200	6.3
f731dp10.mat	1 period data seq.	25200	6.3
f731dp10.mat	1 period data seq.	25200	6.3
f731dp10.mat	1 period data seq.	25200	6.3
f731dp10.mat	1 period data seq.	25200	6.3
f731dp10.mat	1 period data seq.	25200	6.3
f731dp10.mat	1 period data seq.	25200	6.3
f731dp10.mat	1 period data seq.	25200	6.3
f731dp10.mat	1 period data seq.	25200	6.3
f731dp10.mat	1 period data seq.	25200	6.3
f731dp10.mat	1 period data seq.	25200	6.3
f151ps1s.mat	silence(pause)	4000	1
f1313b10.mat	barker code	5200	1.3
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqdpm123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. A BPSK-modulated 13-element Barker code preceeds the preamble. A 63 bit long PN-sequence is used as a data sequence and fed to the DPM-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. The DPM-modulator has a 4RC (raised cosine) prefilter and the modulation index was 0.8. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> initdpm123
>> seqdpm123
f0 = 1000
 symbol rate (Hz):
symbrat = 10
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 1
h = 0.8000
fidur = 4
 No of output symbols requested:
nsymb = 190
 Number of symbol intervals for tone :
nsil = 10
 Number of symbol intervals for clock sync :
nalt = 10
```

```
ans =21-Jan-91
 Informative remark : 10 hz 2dpm4 pn(63) time:11:25
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=normdpmrc(fidur,iprat);
clear oseq oseqr body
pack
oseq=[ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat) dpm(dseq,h,cr,iprat)];
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt-1+pnper);
head=oseqr(1:pnstrt-1);
echo off
>> save f731dp10 body
>> save f732dp10 head
>> diary off
```

2DPM4 33.3 hz symbolrate total duration 43.3 s:

Filename	**signal type**	**Duration: Samples**	**seconds**
f1314b30.mat	barker code	1560	0.39
f151ps1s.mat	silence(pause)	4000	1
f742dp30.mat	preamble (+ trans)	5640	1.41
f741dp30.mat	1 period data seq.	15240	3.81
f741dp30.mat	1 period data seq.	15240	3.81
f741dp30.mat	1 period data seq.	15240	3.81
f741dp30.mat	1 period data seq.	15240	3.81
f741dp30.mat	1 period data seq.	15240	3.81
f741dp30.mat	1 period data seq.	15240	3.81
f741dp30.mat	1 period data seq.	15240	3.81
f741dp30.mat	1 period data seq.	15240	3.81
f741dp30.mat	1 period data seq.	15240	3.81
f741dp30.mat	1 period data seq.	15240	3.81
f151ps1s.mat	silence(pause)	4000	1
f1314b30.mat	barker code	1560	0.39
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqdpm123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. A BPSK-modulated 13-element Barker code preceeds the preamble. A 127 bit long PN-sequence is used as a data sequence and fed to the DPM-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. The DPM-modulator has a 4RC (raised cosine) prefilter and the modulation index was 0.8. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> initdpm123
>> seqdpm123
```

```
f0 = 1000
 symbol rate (Hz):
symbrat = 33.3330
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 1 0 0
h = 0.8000
fidur = 4
 No of output symbols requested:
nsymb = 400
 Number of symbol intervals for tone :
nsil = 15
 Number of symbol intervals for clock sync :
nalt = 15
ans =21-Jan-91
 Informative remark : 33.333 hz 2dpm4 pn(127) time: 10:52
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=normdpmrc(fidur,iprat);
clear oseq oseqr body
pack
oseq=[ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat) dpm(dseq,h,cr,iprat)];
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt-1+pnper);
head=oseqr(1:pnstrt-1);
echo off
>> save f741dp30 body
>> save f742dp30 head
>> diary off
```

2DPM4 100 hz symbolrate total duration 29.1 s:

Filename	signal type	Duration: Samples	seconds
f1315b1h.mat	barker code	520	0.13
f151ps1s.mat	silence(pause)	4000	1
f752dp1h.mat	preamble (+ trans)	1480	0.37
f751dp1h.mat	1 period data seq.	10200	2.55
f751dp1h.mat	1 period data seq.	10200	2.55
f751dp1h.mat	1 period data seq.	10200	2.55
f751dp1h.mat	1 period data seq.	10200	2.55
f751dp1h.mat	1 period data seq.	10200	2.55
f751dp1h.mat	1 period data seq.	10200	2.55
f751dp1h.mat	1 period data seq.	10200	2.55
f751dp1h.mat	1 period data seq.	10200	2.55
f751dp1h.mat	1 period data seq.	10200	2.55
f751dp1h.mat	1 period data seq.	10200	2.55
f151ps1s.mat	silence(pause)	4000	1
f1315b1h.mat	barker code	520	0.13
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqdpm123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. A BPSK-modulated 13-element Barker code preceeds the preamble. A 255 bit long PN-sequence is used as a data sequence and fed to the DPM-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. The DPM-modulator has a 4RC (raised cosine) prefilter and the modulation index was 0.8. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> initdpm123
>> seqdpm123
f0 = 1000
 symbol rate (Hz):
symbrat = 100
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 1 1 1 0
h = 0.8000
fidur = 4
 No of output symbols requested:
nsymb = 800
 Number of symbol intervals for tone :
nsil = 10
 Number of symbol intervals for clock sync :
nalt = 10
ans =20-Jan-91
 Informative remark : 100 hz 2dpm4 h=0.8 input=pn(255),time =17:05
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=normdpmrc(fidur,iprat);
clear oseq oseqr body
```

```
pack
oseq=[ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat) dpm(dseq,h,cr,iprat)];
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt-1+pnper);
head=oseqr(1:pnstrt-1);
echo off
>> save f751dp1h body
>> save f752dp1h head
>> exit
 5066981 flops.
```

2DPM4 333 hz symbolrate total duration 20.3 s:

Filename	signal type	Duration: Samples	seconds
f1316b3h.mat	barker code	156	0.04
f151ps1s.mat	silence(pause)	4000	1
f762dp3h.mat	preamble (+ trans)	7404	1.85
f761dp3h.mat	1 period data seq.	12276	3.07
f761dp3h.mat	1 period data seq.	12276	3.07
f761dp3h.mat	1 period data seq.	12276	3.07
f761dp3h.mat	1 period data seq.	12276	3.07
f761dp3h.mat	1 period data seq.	12276	3.07
f151ps1s.mat	silence(pause)	4000	1
f1316b3h.mat	barker code	156	0.04
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqdpm123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. A BPSK-modulated 13-element Barker code preceeds the preamble. A 1023 bit long PN-sequence is used as a data sequence and fed to the DPM-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. The DPM-modulator has a 4RC (raised cosine) prefilter and the modulation index was 0.8. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> initdpm123
>> seqdpm123
f0 = 1000
 symbol rate (Hz):
symbrat = 333.3300
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 0 0 1 0 0
h = 0.8000
fidur = 4
 No of output symbols requested:
nsymb = 3100
 Number of symbol intervals for tone :
nsil = 300
 Number of symbol intervals for clock sync :
nalt = 300
```

```
ans =20-Jan-91
 Informative remark : 1st attemp dpm 4symb. constr.1 h=.8 input=pn(1023),time=16:30
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=normdpmrc(fidur,iprat);
clear oseq oseqr body
pack
oseq=[ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat) dpm(dseq,h,cr,iprat)];
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt-1+pnper);
head=oseqr(1:pnstrt-1);
>> diary off
```

2DPM4 1 khz symbolrate total duration 13.3 s:

Filename	signal type	Duration: Samples	seconds
f1317b1k.mat	barker code	52	0.013
f151ps1s.mat	silence(pause)	4000	1
f772dp1k.mat	preamble (+ trans)	188	0.047
f771dp1k.mat	1 period data seq.	8188	2.047
f771dp1k.mat	1 period data seq.	8188	2.047
f771dp1k.mat	1 period data seq.	8188	2.047
f771dp1k.mat	1 period data seq.	8188	2.047
f771dp1k.mat	1 period data seq.	8188	2.047
f151ps1s.mat	silence(pause)	4000	1
f1317b1k.mat	barker code	52	0.013
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqdpm123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. A BPSK-modulated 13-element Barker code preceeds the preamble. A 2.047 bit long PN-sequence is used as a data sequence and fed to the DPM-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. The DPM-modulator has a 4RC (raised cosine) prefilter and the modulation index was 0.8. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>>initdpm123
>>seqdpm123
f0=1000
symbolrate(Hz):
symbrat=1000
fs=4000
coefficientsofgeneratingfunction:
genpol=10000000010
h=0.8000
fidur=4
```

```
Noofoutputsymbolsrequested:
nsymb=4100
Numberofsymbolintervalsfortone:
nsil=15
Numberofsymbolintervalsforclocksync:
nalt=15
ans=21-Jan-91
Informativeremark:1khz2dpm4h=.8pn(2047)time:13:00
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil)cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq2*pn-1];
cr=normdpmrc(fidur,iprat);
clearoseqoseqrbody
pack
oseq=[ipfill([-11-11-1-111-1-1-1-1-1],iprat)dpm(dseq,h,cr,iprat)];
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt-1+pnper);
head=oseqr(1:pnstrt-1);
echooff
>>savef771dp1kbody
>>savef772dp1khead
>>exit
59144769flops.
```

8.2 Sequence 4

	2DPM4 1 hz symbolrate total duration 21 min. 56 s:		
Filename	**signal type**	**Duration: Samples**	**seconds**
f1311b01.mat	barker code	52000	13
f151ps1s.mat	silence(pause)	4000	1
f712dp01.mat	preamble (+ trans)	108000	27
f711dp01.mat	1 period data seq.	252000	63
f711dp01.mat	1 period data seq.	252000	63
f711dp01.mat	1 period data seq.	252000	63
f711dp01.mat	1 period data seq.	252000	63
f711dp01.mat	1 period data seq.	252000	63
f711dp01.mat	1 period data seq.	252000	63
f711dp01.mat	1 period data seq.	252000	63
f711dp01.mat	1 period data seq.	252000	63
f711dp01.mat	1 period data seq.	252000	63
f711dp01.mat	1 period data seq.	252000	63
f711dp01.mat	1 period data seq.	252000	63
f711dp01.mat	1 period data seq.	252000	63
f711dp01.mat	1 period data seq.	252000	63
f711dp01.mat	1 period data seq.	252000	63
f711dp01.mat	1 period data seq.	252000	63
f711dp01.mat	1 period data seq.	252000	63
f711dp01.mat	1 period data seq.	252000	63
f711dp01.mat	1 period data seq.	252000	63
f711dp01.mat	1 period data seq.	252000	63
f711dp01.mat	1 period data seq.	252000	63
f711dp01.mat	1 period data seq.	252000	63
f151ps1s.mat	silence(pause)	4000	1
f1311b01.mat	barker code	52000	13
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqdpm123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. A BPSK-modulated 13-element Barker code preceeds the preamble. A 2.047 bit long PN-sequence is used as a data sequence and fed to the DPM-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. The DPM-modulator has a 4RC (raised cosine) prefilter and the modulation index was 0.8. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> initdpm123
>> seqdpm123
f0 = 1000
 symbol rate (Hz):
symbrat = 1
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 1
h = 0.8000
fidur = 4
 No of output symbols requested:
nsymb = 75
 Number of symbol intervals for tone :
nsil = 5
 Number of symbol intervals for clock sync :
nalt = 5
```

```
ans =22-Jan-91
 Informative remark : 1 hz 2dpm4 h=.8 pn(63) time 12:30
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=normdpmrc(fidur,iprat);
clear oseq oseqr body
pack
oseq=[ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat) dpm(dseq,h,cr,iprat)];
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt-1+pnper);
head=oseqr(1:pnstrt-1);
echo off
>> save f712dp01 head
>> save f711dp01 body
>> diary off
```

Sequence 52CPFSK4 3.33 Hz symbolrate total duration 6 min. 39.9 s:

Filename	signal type	Duration: Samples	seconds
f1312b03.mat	barker code	15600	3.9
f151ps1s.mat	silence(pause)	4000	1
f822cf03.mat	preamble (+ trans)	44400	11.1
f821cf03.mat	1 period data seq.	75600	18.9
f821cf03.mat	1 period data seq.	75600	18.9
f821cf03.mat	1 period data seq.	75600	18.9
f821cf03.mat	1 period data seq.	75600	18.9
f821cf03.mat	1 period data seq.	75600	18.9
f821cf03.mat	1 period data seq.	75600	18.9
f821cf03.mat	1 period data seq.	75600	18.9
f821cf03.mat	1 period data seq.	75600	18.9
f821cf03.mat	1 period data seq.	75600	18.9
f821cf03.mat	1 period data seq.	75600	18.9
f821cf03.mat	1 period data seq.	75600	18.9
f821cf03.mat	1 period data seq.	75600	18.9
f821cf03.mat	1 period data seq.	75600	18.9
f821cf03.mat	1 period data seq.	75600	18.9
f821cf03.mat	1 period data seq.	75600	18.9
f821cf03.mat	1 period data seq.	75600	18.9
f821cf03.mat	1 period data seq.	75600	18.9
f821cf03.mat	1 period data seq.	75600	18.9
f821cf03.mat	1 period data seq.	75600	18.9
f821cf03.mat	1 period data seq.	75600	18.9
f151ps1s.mat	silence(pause)	4000	1
f1312b03.mat	barker code	15600	3.9
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqcpfsk123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. A BPSK-modulated 13-element Barker code preceeds the preamble. A 63 bit long PN-sequence is used as a data sequence and fed to the CPFSK-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. The CPFSK-modulator has a 4RC (raised cosine) prefilter and the modulation index was 0.75. NB! this sequnce is not a perfect period of a periodic CPFSK-sequence. Consequently there are a discontinuity at the end/dtart point of the body file f821cf03.mat.The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> initcpfsk123
>> seqcpfsk123
f0 = 1000
 symbol rate (Hz):
symbrat = 3.3333
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 1
h = 0.7500
fidur = 4
 No of output symbols requested:
nsymb = 130
 Number of symbol intervals for tone :
nsil = 10
 Number of symbol intervals for clock sync :
nalt = 10
ans =23-Jan-91
 Informative remark : 2cpfsk4rc h=.75 pn(63) time=17:00 neww attempt
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=rcsynth(fidur*iprat);
clear oseq oseqr body
pack
oseq=ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat);
oseq=[oseq cpfsk(dseq,h,cr,iprat)];
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
???? Error using ==> *
Out of memory. Type HELP MEMORY for your options.
>> cw=exp((sqrt(-1)*2*pi*(f0/fs))*(0:1199));
>> for k=1:163,oseq(k*1200-1199:k*1200)=oseq(k*1200-1199:k*1200).*cw;end
>> oseqr=real(oseq);
>> pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
>> pnper=(2^length(genpol)-1)*iprat;
>> oseqr=oseqr/max(abs(oseqr));
>> body=oseqr(pnstrt:pnstrt-1+pnper);
>> head=oseqr(1:pnstrt-1);
>> save f8212f03 body
>> save f8222f03 head
```

```
>> diary off
```

2CPFSK4 10 Hz symbolrate total duration 2 min 40.5 s:

Filename	signal type	Duration: Samples	seconds
f1313b10.mat	barker code	5200	1.3
f151ps1s.mat	silence(pause)	4000	1
f832cf10.mat	preamble (+ trans)	9600	2.4
f831cf10.mat	1 period data seq.	201600	50.4
f831cf10.mat	1 period data seq.	201600	50.4
f831cf10.mat	1 period data seq.	201600	50.4
f151ps1s.mat	silence(pause)	4000	1
f1313b10.mat	barker code	5200	1.3
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqcpfsk123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. A 63 bit long PN-sequence is used as a data sequence and fed to the CPFSK-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. This sequence is repeated 8 times to yield one period of a periodic CPFSK-sequence. The CPFSK-modulator has a 4RC (raised cosine) prefilter and the modulation index was 0.75. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> initcpfsk123
>> seqcpfsk123
f0 = 1000
symbrat = 10
fs = 4000
genpol = 1 0 0 0 0 1
clear reginit
reginit(length(genpol))=1;
h = 0.7500
fidur = 4
nsymb = 510
nsil = 10
nalt = 10
date
ans =25-Jan-91
tkst=input(' Informative remark : ','s');
 Informative remark : disp(tkst)
8 per. 10 hz 2cpfsk4rc pn(63) time=21:55
clear dseq
echo on
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=rcsynth(fidur*iprat);
clear oseq oseqr body
pack
oseq=cpfsk(dseq,h,cr,iprat);
```

```
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
???? Error using ==> *
Out of memory. Type HELP MEMORY for your options.
>> cw=exp((sqrt(-1)*2*pi*(f0/fs))*(0:999));
>> for k=1:212,oseq(k*1000-999:k*1000)=oseq(k*1000-999:k*1000).*cw;end
>> oseqr=real(oseq);
>> pnper=(2^length(genpol)-1)*iprat;
>> oseqr=oseqr/max(abs(oseqr));
>> pnstrt=1+length(cr)+iprat*(nsil+nalt);
>> body=oseqr(pnstrt:pnstrt-1+pnper*8);
>> head=oseqr(1:pnstrt-1);
>> save f831cf10 body
>> diary off
```

2CPFSK4 33.3 Hz symbolrate total duration 1 min 6.2 s:

Filename	signal type	Duration: Samples	seconds
f1314b30.mat	barker code	1560	0.39
f151ps1s.mat	silence(pause)	4000	1
f842cf30.mat	preamble (+ trans)	5640	1.41
f841cf30.mat	1 period data seq.	121920	30.48
f841cf30.mat	1 period data seq.	121920	30.48
f151ps1s.mat	silence(pause)	4000	1
f1314b30.mat	barker code	1560	0.39
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqcpfsk123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. The preamble is preceeded by a 13-element bpsk-modulated Barker code . A 127 bit long PN-sequence is used as a data sequence and fed to the CPFSK-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. This sequence is repeated 8 times to yield one period of a periodic CPFSK-sequence. The CPFSK-modulator has a 4RC (raised cosine) prefilter and the modulation index was 0.75. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> initcpfsk123
>> seqcpfsk123
f0 = 1000
 symbol rate (Hz):
symbrat = 33.3333
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 1 0 0
h = 0.7500
fidur = 4
 No of output symbols requested:
nsymb = 1116
 Number of symbol intervals for tone :
nsil = 15
 Number of symbol intervals for clock sync :
nalt = 15
ans =25-Jan-91
```

```
Informative remark : 8 pers. 2cpfsk4rc 33.33 hz pn(127), h=.75, time =21:00
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=rcsynth(fidur*iprat);
clear oseq oseqr body
pack
oseq=ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat);
oseq=[oseq cpfsk(dseq,h,cr,iprat)];
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt-1+pnper);
head=oseqr(1:pnstrt-1);
echo off
>> body=oseqr(pnstrt:pnstrt-1+pnper*8);
>> save f841cf30 body
>> save f842cf30 head
>> diary off
```

2CPFSK4 100 Hz symbolrate total duration 44.4 s:

Filename	signal type	Duration: Samples	seconds
f1315b1h.mat	barker code	520	0.13
f151ps1s.mat	silence(pause)	4000	1
f852cf1h.mat	preamble (+ trans)	1480	0.37
f851cf1h.mat	1 period data seq.	81600	20.4
f851cf1h.mat	1 period data seq.	81600	20.4
f151ps1s.mat	silence(pause)	4000	1
f1315b1h.mat	barker code	520	0.13
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqcpfsk123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. The preamble is preceeded by a 13-element bpsk-modulated Barker code . A 255 bit long PN-sequence is used as a data sequence and fed to the CPFSK-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. This sequence is repeated 8 times to yield one period of a periodic CPFSK-sequence. The CPFSK-modulator has a 4RC (raised cosine) prefilter and the modulation index was 0.75. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> initcpfsk123
>> seqcpfsk123
f0 = 1000
 symbol rate (Hz):
symbrat = 100
fs = 4000
```

```
 coefficients of generating function:
genpol = 1 0 0 0 1 1 1 0
h = 0.7500
fidur = 4
 No of output symbols requested:
nsymb = 2140
 Number of symbol intervals for tone :
nsil = 10
 Number of symbol intervals for clock sync :
nalt = 10
ans =25-Jan-91
 Informative remark : new 8 perids 2cpfsk4 100 hz pn(255) h=.75, time=20:45
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=rcsynth(fidur*iprat);
clear oseq oseqr body
pack
oseq=ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat);
oseq=[oseq cpfsk(dseq,h,cr,iprat)];
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt-1+pnper);
head=oseqr(1:pnstrt-1);
echo off
>> body=oseqr(pnstrt:pnstrt-1+pnper*8);
>> save f851cf1h body
>> save f852cf1h head
>> diary off
```

2CPFSK4 333 Hz symbolrate total duration 52.3 s:

Filename	**signal type**	**Duration: Samples**	**seconds**
f1316b3h.mat	barker code	156	0.04
f151ps1s.mat	silence(pause)	4000	1
f862cf3h.mat	preamble (+ trans)	564	0.141
f861cf3h.mat	1 period data seq.	98208	24.552
f861cf3h.mat	1 period data seq.	98208	24.552
f151ps1s.mat	silence(pause)	4000	1
f1316b3h.mat	barker code	156	0.04
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqcpfsk123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. The preamble is preceeded by a 13-element bpsk-modulated Barker code . A 1023 bit long PN-sequence is used as a data sequence and fed to the CPFSK-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz.This sequence is repeated 8 times to yield one period of a periodic CPFSK-sequence. The CPFSK-modulator has a 4RC (raised cosine) prefilter and the modulation index was 0.75. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> initcpfsk123
>> seqcpfsk123
>> initcpfsk123
>> seqcpfsk123
f0 = 1000
symbrat = 333.3300
fs = 4000
genpol = 1 0 0 0 0 0 0 1 0 0
h = 0.7500
fidur = 4
 No of output symbols requested:
nsymb = 8284
nsil = 15
nalt = 15
ans =25-Jan-91
 Informative remark : 8periods 333.33 hz 2cpfsk4 h=3/4 pn(1023) time= 20:05
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=rcsynth(fidur*iprat);
clear oseq oseqr body
pack
oseq=ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat);
oseq=[oseq cpfsk(dseq,h,cr,iprat)];
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt-1+pnper);
head=oseqr(1:pnstrt-1);
echo off
>> body=oseqr(pnstrt:pnstrt-1+pnper*8);
>> save f862cf3h head
>> save f861cf3h body
>> diary off
```

2CPFSK4 1 kHz symbolrate total duration 19.4 s:

Filename	signal type	Duration: Samples	seconds
f1317b1k.mat	barker code	52	0.013
f151ps1s.mat	silence(pause)	4000	1
f872cf1k.mat	preamble (+ trans)	136	0.034
f871cf1k.mat	1 period data seq.	65504	16.376
f151ps1s.mat	silence(pause)	4000	1
f1317b1k.mat	barker code	52	0.013
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqcpfsk123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. The preamble is preceeded by a 13-element bpsk-modulated Barker code . A 2047 bit long PN-sequence is used as a data sequence and fed to the CPFSK-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. This sequence is repeated 8 times to yield one period of a periodic CPFSK-sequence. The CPFSK-modulator has a 4RC (raised cosine) prefilter and the modulation index was 0.75. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>>initcpfsk123
>>seqcpfsk123
f0=1000
symbolrate(Hz):
symbrat=1000
fs=4000
coefficientsofgeneratingfunction:
genpol=10000000010
h=0.7500
fidur=4
Noofoutputsymbolsrequested:
nsymb=16476
Numberofsymbolintervalsfortone:
nsil=15
Numberofsymbolintervalsforclocksync:
nalt=15
ans=25-Jan-9
Informativeremark:8per.1khz2cpfsk4h=.75pn(2047)time=22:55
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil)cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq2*pn-1];
cr=rcsynth(fidur*iprat);
clearoseqoseqrbody
pack
oseq=ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 1 -1],iprat);
oseq=[oseq cpfsk(dseq,h,cr,iprat)];
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
```

```
body=oseqr(pnstrt:pnstrt-1+pnper*8);
head=oseqr(1:pnstrt-1);
>>save f871cf1k body
>>save f872cf1k head
>>exit
7364334flops.
```

8.3 Sequence 6

Filename	2CPFSK4 1 Hz symbolrate total duration 21 min. 56 s: signal type	Duration: Samples	seconds
f1311b01.mat	barker code	52000	13
f151ps1s.mat	silence(pause)	4000	1
f812cf01.mat	preamble (+ trans)	108000	27
f811cf01.mat	1 period data seq.	252000	63
f811cf01.mat	1 period data seq.	252000	63
f811cf01.mat	1 period data seq.	252000	63
f811cf01.mat	1 period data seq.	252000	63
f811cf01.mat	1 period data seq.	252000	63
f811cf01.mat	1 period data seq.	252000	63
f811cf01.mat	1 period data seq.	252000	63
f811cf01.mat	1 period data seq.	252000	63
f811cf01.mat	1 period data seq.	252000	63
f811cf01.mat	1 period data seq.	252000	63
f811cf01.mat	1 period data seq.	252000	63
f811cf01.mat	1 period data seq.	252000	63
f811cf01.mat	1 period data seq.	252000	63
f811cf01.mat	1 period data seq.	252000	63
f811cf01.mat	1 period data seq.	252000	63
f811cf01.mat	1 period data seq.	252000	63
f811cf01.mat	1 period data seq.	252000	63
f811cf01.mat	1 period data seq.	252000	63
f811cf01.mat	1 period data seq.	252000	63
f811cf01.mat	1 period data seq.	252000	63
f151ps1s.mat	silence(pause)	4000	4
f1311b01.mat	barker code	52000	13
f151ps1s.mat	silence(pause)	4000	4

The files containing the preamble and the data sequence were generated by the matlab m-file "seqcpfsk123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. A BPSK-modulated 13-element Barker code preceeds the preamble. A 63 bit long PN-sequence is used as a data sequence and fed to the CPFSK-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. The CPFSK-modulator has a 4RC (raised cosine) prefilter and the modulation index was 0.75. NB! this sequnce is not a perfect period of a periodic CPFSK-sequence. Consequently there are a discontinuity at the end/dtart point of the body file f821cf03.mat.The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> initcpfsk123
>> seqcpfsk123
```

```
f0 = 1000
 symbol rate (Hz):
symbrat = 1
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 1
h = 0.7500
fidur= 4
 No of output symbols requested:
nsymb = 75
 Number of symbol intervals for tone :
nsil = 5
 Number of symbol intervals for clock sync :
nalt = 5
ans =22-Jan-91
 Informative remark : 1 hz 2cpfsk4 h=.75 pn(63) time 16:40
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=rcsynth(fidur*iprat);
clear oseq oseqr body
pack
oseq=ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat);
oseq=[oseq cpfsk(dseq,h,cr,iprat)];
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt-1+pnper);
head=oseqr(1:pnstrt-1);
echo off
>> save f811cf01 body
>> save f812cf02 head
>> diary off
```

8.4 Sequence 7

Filename	2DPM4 3.33 hz symbolrate total duration 6 min. 39.9 s: signal type	Duration: Samples	seconds
f1312b03.mat	barker code	15600	3.9
f151ps1s.mat	silence(pause)	4000	1
f922dp03.mat	preamble (+ trans)	44400	11.1
f921dp03.mat	1 period data seq.	75600	18.9
f921dp03.mat	1 period data seq.	75600	18.9
f921dp03.mat	1 period data seq.	75600	18.9
f921dp03.mat	1 period data seq.	75600	18.9
f921dp03.mat	1 period data seq.	75600	18.9
f921dp03.mat	1 period data seq.	75600	18.9
f921dp03.mat	1 period data seq.	75600	18.9
f921dp03.mat	1 period data seq.	75600	18.9
f921dp03.mat	1 period data seq.	75600	18.9
f921dp03.mat	1 period data seq.	75600	18.9
f921dp03.mat	1 period data seq.	75600	18.9
f921dp03.mat	1 period data seq.	75600	18.9
f921dp03.mat	1 period data seq.	75600	18.9
f921dp03.mat	1 period data seq.	75600	18.9
f921dp03.mat	1 period data seq.	75600	18.9
f921dp03.mat	1 period data seq.	75600	18.9
f921dp03.mat	1 period data seq.	75600	18.9
f921dp03.mat	1 period data seq.	75600	18.9
f921dp03.mat	1 period data seq.	75600	18.9
f921dp03.mat	1 period data seq.	75600	18.9
f921dp03.mat	1 period data seq.	75600	18.9
f151ps1s.mat	silence(pause)	4000	1
f1312b03.mat	barker code	15600	3.9
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqdpm123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. A BPSK-modulated 13-element Barker code preceeds the preamble. A 63 bit long PN-sequence is used as a data sequence and fed to the DPM-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. The DPM-modulator has a 4RC (raised cosine) prefilter and the modulation index was 1.0. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init2dpm123
>> seqdpm123
f0 = 1000
 symbol rate (Hz):
symbrat = 3.3333
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 1
h = 1
fidur = 4
 No of output symbols requested:
nsymb = 130
 Number of symbol intervals for tone :
```

```
nsil = 10
 Number of symbol intervals for clock sync :
nalt = 10
ans =23-Jan-91
 Informative remark : 2dpm4rc h=1.0 pn(63) time 11:25
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=normdpmrc(fidur,iprat);
clear oseq oseqr body
pack
oseq=[ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat) dpm(dseq,h,cr,iprat)];
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
???? Error using ==> *
Out of memory. Type HELP MEMORY for your options.
>> cw=exp((sqrt(-1)*2*pi*(f0/fs))*(0:1199));
>> for k=1:63,oseq(k*1200-1199:k*1200)=oseq(k*1200-1199:k*1200).*cw;end
>> pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
>> pnper=(2^length(genpol)-1)*iprat;
>> oseqr=real(oseq);
???? Error using ==> real
Out of memory. Type HELP MEMORY for your options.
>> body=real(oseq(pnstrt:pnstrt-1+pnper));
>> head=real(oseq(1:pnstrt-1));
>> save f921dp03 body
>> save f922dp03 head
>> exit
 49343180 flops.
```

Filename	signal type	Duration: Samples	seconds
	2DPM4 10 hz symbolrate total duration 2 min 15.3 s:		
f1313b10.mat	barker code	5200	1.3
f151ps1s.mat	silence(pause)	4000	1
f932dp10.mat	preamble (+ trans)	14800	3.7
f931dp10.mat	1 period data seq.	25200	6.3
f931dp10.mat	1 period data seq.	25200	6.3
f931dp10.mat	1 period data seq.	25200	6.3
f931dp10.mat	1 period data seq.	25200	6.3
f931dp10.mat	1 period data seq.	25200	6.3
f931dp10.mat	1 period data seq.	25200	6.3
f931dp10.mat	1 period data seq.	25200	6.3
f931dp10.mat	1 period data seq.	25200	6.3
f931dp10.mat	1 period data seq.	25200	6.3
f931dp10.mat	1 period data seq.	25200	6.3
f931dp10.mat	1 period data seq.	25200	6.3
f931dp10.mat	1 period data seq.	25200	6.3
f931dp10.mat	1 period data seq.	25200	6.3
f931dp10.mat	1 period data seq.	25200	6.3
f931dp10.mat	1 period data seq.	25200	6.3
f931dp10.mat	1 period data seq.	25200	6.3
f931dp10.mat	1 period data seq.	25200	6.3
f931dp10.mat	1 period data seq.	25200	6.3
f931dp10.mat	1 period data seq.	25200	6.3
f931dp10.mat	1 period data seq.	25200	6.3
f931dp10.mat	1 period data seq.	25200	6.3
f151ps1s.mat	silence(pause)	4000	1
f1313b10.mat	barker code	5200	1.3
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqdpm123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. A BPSK-modulated 13-element Barker code preceeds the preamble. A 63 bit long PN-sequence is used as a data sequence and fed to the DPM-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. The DPM-modulator has a 4RC (raised cosine) prefilter and the modulation index was 1.0. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init2dpm123
>> seqdpm123
f0 = 1000
 symbol rate (Hz):
symbrat = 10
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 1
h = 1
fidur = 4
 No of output symbols requested:
nsymb = 190
 Number of symbol intervals for tone :
nsil = 10
 Number of symbol intervals for clock sync :
nalt = 10
```

```
ans =23-Jan-91
 Informative remark : 10 hz 2dpm4rc pn(63) h=1.0, time=11:10
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=normdpmrc(fidur,iprat);
clear oseq oseqr body
pack
oseq=[ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat) dpm(dseq,h,cr,iprat)];
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt-1+pnper);
head=oseqr(1:pnstrt-1);
echo off
>> save f931dp10 body
>> save f932dp10 head
>> diary off
```

2DPM4 33.3 hz symbolrate total duration 43.3 s:

Filename	signal type	Duration: Samples	seconds
f1314b30.mat	barker code	1560	0.39
f151ps1s.mat	silence(pause)	4000	1
f942dp30.mat	preamble (+ trans)	5640	1.41
f941dp30.mat	1 period data seq.	15240	3.81
f941dp30.mat	1 period data seq.	15240	3.81
f941dp30.mat	1 period data seq.	15240	3.81
f941dp30.mat	1 period data seq.	15240	3.81
f941dp30.mat	1 period data seq.	15240	3.81
f941dp30.mat	1 period data seq.	15240	3.81
f941dp30.mat	1 period data seq.	15240	3.81
f941dp30.mat	1 period data seq.	15240	3.81
f941dp30.mat	1 period data seq.	15240	3.81
f941dp30.mat	1 period data seq.	15240	3.81
f151ps1s.mat	silence(pause)	4000	1
f1314b30.mat	barker code	1560	0.39
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqdpm123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. A BPSK-modulated 13-element Barker code preceeds the preamble. A 127 bit long PN-sequence is used as a data sequence and fed to the DPM-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. The DPM-modulator has a 4RC (raised cosine) prefilter and the modulation index was 1.0. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init2dpm123
>> seqdpm123
```

```
f0 = 1000
 symbol rate (Hz):
symbrat = 33.3333
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 1 0 0
h = 1
fidur = 4
 No of output symbols requested:
nsymb = 400
 Number of symbol intervals for tone :
nsil = 15
 Number of symbol intervals for clock sync :
nalt = 15
ans =23-Jan-91
 Informative remark : 33.33 hz 2dpm4rc h= 1.0 pn(127) time=10:55
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=normdpmrc(fidur,iprat);
clear oseq oseqr body
pack
oseq=[ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat) dpm(dseq,h,cr,iprat)];
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt-1+pnper);
head=oseqr(1:pnstrt-1);
echo off
>> save f941dp30 body
>> save f942dp30 head
>> exit
 21090677 flops.
```

	2DPM4 100 hz symbolrate total duration 29.1 s:		
Filename	**signal type**	**Duration: Samples**	**seconds**
f1315b1h.mat	barker code	520	0.13
f151ps1s.mat	silence(pause)	4000	1
f952dp1h.mat	preamble (+ trans)	1480	0.37
f951dp1h.mat	1 period data seq.	10200	2.55
f951dp1h.mat	1 period data seq.	10200	2.55
f951dp1h.mat	1 period data seq.	10200	2.55
f951dp1h.mat	1 period data seq.	10200	2.55
f951dp1h.mat	1 period data seq.	10200	2.55
f951dp1h.mat	1 period data seq.	10200	2.55
f951dp1h.mat	1 period data seq.	10200	2.55
f951dp1h.mat	1 period data seq.	10200	2.55
f951dp1h.mat	1 period data seq.	10200	2.55
f951dp1h.mat	1 period data seq.	10200	2.55
f151ps1s.mat	silence(pause)	4000	1
f1315b1h.mat	barker code	520	0.13
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqdpm123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. A BPSK-modulated 13-element Barker code preceeds the preamble. A 255 bit long PN-sequence is used as a data sequence and fed to the DPM-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. The DPM-modulator has a 4RC (raised cosine) prefilter and the modulation index was 1.0. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init2dpm123
>> seqdpm123
f0 = 1000
 symbol rate (Hz):
symbrat = 100
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 1 1 1 0
h = 1
fidur = 4
 No of output symbols requested:
nsymb = 800
 Number of symbol intervals for tone :
nsil = 10
 Number of symbol intervals for clock sync :
nalt = 10
ans =23-Jan-91
 Informative remark : h=1.0 2dpm4rc 100 hz pn(255) time = 10:40
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=normdpmrc(fidur,iprat);
clear oseq oseqr body
```

```
pack
oseq=[ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat) dpm(dseq,h,cr,iprat)];
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt-1+pnper);
head=oseqr(1:pnstrt-1);
echo off
>> save f951dp1h body
>> save f952dp1h head
>> diary off
```

2DPM4 333 hz symbolrate total duration 33.9 s:

Filename	signal type	Duration: Samples	seconds
f1316b3h.mat	barker code	156	0.04
f151ps1s.mat	silence(pause)	4000	1
f962dp3h.mat	preamble (+ trans)	564	0.141
f961dp3h.mat	1 period data seq.	12276	3.069
f961dp3h.mat	1 period data seq.	12276	3.069
f961dp3h.mat	1 period data seq.	12276	3.069
f961dp3h.mat	1 period data seq.	12276	3.069
f961dp3h.mat	1 period data seq.	12276	3.069
f961dp3h.mat	1 period data seq.	12276	3.069
f961dp3h.mat	1 period data seq.	12276	3.069
f961dp3h.mat	1 period data seq.	12276	3.069
f961dp3h.mat	1 period data seq.	12276	3.069
f961dp3h.mat	1 period data seq.	12276	3.069
f151ps1s.mat	silence(pause)	4000	1
f1316b3h.mat	barker code	156	0.04
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqdpm123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. A BPSK-modulated 13-element Barker code preceeds the preamble. A 1023 bit long PN-sequence is used as a data sequence and fed to the DPM-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. The DPM-modulator has a 4RC (raised cosine) prefilter and the modulation index was 1.0. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init2dpm123
>> seqdpm123
f0 = 1000
 symbol rate (Hz):
symbrat = 333.3300
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 0 0 1 0 0
h = 1
fidur = 4
 No of output symbols requested:
nsymb = 3100
```

```
 Number of symbol intervals for tone :
nsil = 15
 Number of symbol intervals for clock sync :
nalt = 15
ans =23-Jan-91
Informative remark : 2dpm4, h=1.0 333.3 hz pn(1023), time : 10:10
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=normdpmrc(fidur,iprat);
clear oseq oseqr body
pack
oseq=[ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat) dpm(dseq,h,cr,iprat)];
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt-1+pnper);
head=oseqr(1:pnstrt-1);
echo off
>> save f961dp3h body
>> save f962dp3h head
>> diary off
```

2DPM4 1 khz symbolrate total duration 13.3 s:

Filename	signal type	Duration: Samples	seconds
f1317b1k.mat	barker code	52	0.013
f151ps1s.mat	silence(pause)	4000	1
f972dp1k.mat	preamble (+ trans)	188	0.047
f971dp1k.mat	1 period data seq.	8188	2.047
f971dp1k.mat	1 period data seq.	8188	2.047
f971dp1k.mat	1 period data seq.	8188	2.047
f971dp1k.mat	1 period data seq.	8188	2.047
f9771dp1k.mat	1 period data seq.	8188	2.047
f151ps1s.mat	silence(pause)	4000	1
f1317b1k.mat	barker code	52	0.013
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqdpm123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. A BPSK-modulated 13-element Barker code preceeds the preamble. A 2.047 bit long PN-sequence is used as a data sequence and fed to the DPM-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. The DPM-modulator has a 4RC (raised cosine) prefilter and the modulation index was 1.0. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init2dpm123
>> seqdpm123
f0 = 1000
```

```
 symbol rate (Hz):
symbrat = 1000
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 0 0 0 0 1 0
h = 1
fidur = 4
 No of output symbols requested:
nsymb = 4100
 Number of symbol intervals for tone :
nsil = 15
 Number of symbol intervals for clock sync :
nalt = 15
ans =23-Jan-91
Informative remark : 1khz 2dpm4rc h=1.0 pn(2047) time= 12:55
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=normdpmrc(fidur,iprat);
clear oseq oseqr body
pack
oseq=[ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat) dpm(dseq,h,cr,iprat)];
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt-1+pnper);
head=oseqr(1:pnstrt-1);
echo off
>> save f971dp1k body
>> save f972dp1k head
>> diary off
```

8.5 Sequence 8

Filename	2DPM4 1 hz symbolrate total duration 21 min. 56 s: signal type	Duration: Samples	seconds
f1311b01.mat	barker code	52000	13
f151ps1s.mat	silence(pause)	4000	1
f912dp01.mat	preamble (+ trans)	108000	27
f911dp01.mat	1 period data seq.	252000	63
f911dp01.mat	1 period data seq.	252000	63
f911dp01.mat	1 period data seq.	252000	63
f911dp01.mat	1 period data seq.	252000	63
f911dp01.mat	1 period data seq.	252000	63
f911dp01.mat	1 period data seq.	252000	63
f911dp01.mat	1 period data seq.	252000	63
f911dp01.mat	1 period data seq.	252000	63
f911dp01.mat	1 period data seq.	252000	63
f911dp01.mat	1 period data seq.	252000	63
f911dp01.mat	1 period data seq.	252000	63
f911dp01.mat	1 period data seq.	252000	63
f911dp01.mat	1 period data seq.	252000	63
f911dp01.mat	1 period data seq.	252000	63
f911dp01.mat	1 period data seq.	252000	63
f911dp01.mat	1 period data seq.	252000	63
f911dp01.mat	1 period data seq.	252000	63
f911dp01.mat	1 period data seq.	252000	63
f911dp01.mat	1 period data seq.	252000	63
f911dp01.mat	1 period data seq.	252000	63
f911dp01.mat	1 period data seq.	252000	63
f151ps1s.mat	silence(pause)	4000	1
f1311b01.mat	barker code	52000	13
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqdpm123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. A BPSK-modulated 13-element Barker code preceeds the preamble. A 2.047 bit long PN-sequence is used as a data sequence and fed to the DPM-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. The DPM-modulator has a 4RC (raised cosine) prefilter and the modulation index was 1.0. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init2dpm123
>> seqdpm123
f0 = 1000
 symbol rate (Hz):
symbrat = 1
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 1
h = 1
fidur = 4
 No of output symbols requested:
nsymb = 75
 Number of symbol intervals for tone :
nsil = 5
 Number of symbol intervals for clock sync :
nalt = 5
```

```
ans =23-Jan-91
 Informative remark : 1 hz 2dpm4 h=1.0 pn(63) time 09:30
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=normdpmrc(fidur,iprat);
clear oseq oseqr body
pack
oseq=[ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat) dpm(dseq,h,cr,iprat)];
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt-1+pnper);
head=oseqr(1:pnstrt-1);
echo off
>> save f911dp01 body
>> save f912dp01 head
>> fb=fft(body);
>> exit
 103079830 flops.
```

8.6 Sequence 9

Filename	2CPFSK4 3.33 Hz symbolrate total duration 5 min. 15.3 s: signal type	Duration: Samples	seconds
f1312b03.mat	barker code	15600	3.9
f151ps1s.mat	silence(pause)	4000	1
f8222f03.mat	preamble (+ trans)	26400	6.6
f8212f03.mat	1 period data seq.	604800	151.2
f8212f03.mat	1 period data seq.	604800	151.2
f151ps1s.mat	silence(pause)	4000	1
f1312b03.mat	barker code	15600	3.9
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqcpfsk123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. A BPSK-modulated 13-element Barker code preceeds the preamble. A 63 bit long PN-sequence is used as a data sequence and fed to the CPFSK-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. This sequence is repeated 8 times to yield one period of a periodic CPFSK-sequence. The CPFSK-modulator has a 2RC (raised cosine) prefilter and the modulation index was 0.75. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init2cpfsk123
>> seqcpfsk123
f0 = 1000
 symbol rate (Hz):
```

```
symbrat = 3.3333
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 1
h = 0.7500
fidur = 2
 No of output symbols requested:
nsymb = 510
 Number of symbol intervals for tone :
nsil = 10
 Number of symbol intervals for clock sync :
nalt = 10
ans =25-Jan-91
Informative remark : 8 per. 2cpfsk2rc h=3/4 pn(63), time =22:05
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=rcsynth(fidur*iprat);
clear oseq oseqr body
pack
oseq=cpfsk(dseq,h,cr,iprat);
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(nsil+nalt);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt-1+pnper*8);
head=oseqr(1:pnstrt-1);
>> save f8212f03 body
>> save f8222f03 head
>> diary off
```

2CPFSK4 10 Hz symbolrate total duration 2 min 40.5 s:

Filename	signal type	Duration: Samples	seconds
f1313b10.mat	barker code	5200	1.3
f151ps1s.mat	silence(pause)	4000	1
f8322f10.mat	preamble (+ trans)	9600	2.4
f8312f10.mat	1 period data seq.	201600	50.4
f8312f10.mat	1 period data seq.	201600	50.4
f8312f10.mat	1 period data seq.	201600	50.4
f151ps1s.mat	silence(pause)	4000	1
f1313b10.mat	barker code	5200	1.3
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqcpfsk123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. A 63 bit long PN-sequence is used as a data sequence and fed to the CPFSK-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. This sequence is repeated 8 times to yield one period of a periodic CPFSK-sequence. The CPFSK-modulator has a 2RC (raised cosine) prefilter and the modulation index was 0.75. The resulting sequence is modulated on to a 4 kHz carrier.

2CPFSK4 33.3 Hz symbolrate total duration 1 min 6.2 s:

Filename	signal type	Duration: Samples	seconds
f1314b30.mat	barker code	1560	0.39
f151ps1s.mat	silence(pause)	4000	1
f8422f30.mat	preamble (+ trans)	5640	1.41
f8412f30.mat	1 period data seq.	121920	30.48
f8412f30.mat	1 period data seq.	121920	30.48
f151ps1s.mat	silence(pause)	4000	1
f1314b30.mat	barker code	1560	0.39
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqcpfsk123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. The preamble is preceeded by a 13-element bpsk-modulated Barker code . A 127 bit long PN-sequence is used as a data sequence and fed to the CPFSK-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. This sequence is repeated 8 times to yield one period of a periodic CPFSK-sequence. The CPFSK-modulator has a 2RC (raised cosine) prefilter and the modulation index was 0.75. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init2cpfsk123
>> seqcpfsk123
f0 = 1000
 symbol rate (Hz):
symbrat = 33.3333
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 1 0 0
h = 0.7500
fidur = 2
 No of output symbols requested:
nsymb = 1031
 Number of symbol intervals for tone :
nsil = 15
 Number of symbol intervals for clock sync :
nalt = 15
ans =26-Jan-91
 Informative remark : 8 per. 2cpfsk2rc 33.33 hz pn(127), h=.75 time=12:30
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=rcsynth(fidur*iprat);
clear oseq oseqr body
pack
oseq=ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat);
oseq=[oseq cpfsk(dseq,h,cr,iprat)];
```

```
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt-1+pnper*8);
head=oseqr(1:pnstrt-1);
>> save f8412f30 body
>> save f8422f30 head
>> diary off
```

2CPFSK4 100 Hz symbolrate total duration 44.4 s:

Filename	signal type	Duration: Samples	seconds
f1315b1h.mat	barker code	520	0.13
f151ps1s.mat	silence(pause)	4000	1
f8522f1h.mat	preamble (+ trans)	1480	0.37
f8512f1h.mat	1 period data seq.	81600	20.4
f8512f1h.mat	1 period data seq.	81600	20.4
f151ps1s.mat	silence(pause)	4000	1
f1315b1h.mat	barker code	520	0.13
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqcpfsk123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. The preamble is preceeded by a 13-element bpsk-modulated Barker code . A 255 bit long PN-sequence is used as a data sequence and fed to the CPFSK-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. This sequence is repeated 8 times to yield one period of a periodic CPFSK-sequence. The CPFSK-modulator has a 2RC (raised cosine) prefilter and the modulation index was 0.75. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init2cpfsk123
>> seqcpfsk123
f0 = 1000
 symbol rate (Hz):
symbrat = 100
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 1 1 1 0
h = 0.7500
fidur = 2
 No of output symbols requested:
nsymb = 2140
 Number of symbol intervals for tone :
nsil = 10
 Number of symbol intervals for clock sync :
nalt = 10
ans =26-Jan-91
 Informative remark : 8 per 2cpfsk2rc 100 hz pn(255) h=.75 time=12:25
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
```

```
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=rcsynth(fidur*iprat);
clear oseq oseqr body
pack
oseq=ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat);
oseq=[oseq cpfsk(dseq,h,cr,iprat)];
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt-1+pnper*8);
head=oseqr(1:pnstrt-1);
echo off
>> save f8512f1h body
>> save f8522f1h head
>> diary off
```

2CPFSK4 333 Hz symbolrate total duration 52.3 s:

Filename	signal type	Duration: Samples	seconds
f1316b3h.mat	barker code	156	0.04
f151ps1s.mat	silence(pause)	4000	1
f8622f3h.mat	preamble (+ trans)	564	0.141
f8612f3h.mat	1 period data seq.	98208	24.552
f8612f3h.mat	1 period data seq.	98208	24.552
f151ps1s.mat	silence(pause)	4000	1
f1316b3h.mat	barker code	156	0.04
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqcpfsk123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. The preamble is preceeded by a 13-element bpsk-modulated Barker code . A 1023 bit long PN-sequence is used as a data sequence and fed to the CPFSK-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz.This sequence is repeated 8 times to yield one period of a periodic CPFSK-sequence. The CPFSK-modulator has a 2RC (raised cosine) prefilter and the modulation index was 0.75. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init2cpfsk123
>> seqcpfsk123
f0 = 1000
 symbol rate (Hz):
symbrat   333.3330
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 0 0 1 0 0
h = 0.7500
fidur = 2
 No of output symbols requested:
nsymb = 8684
```

```
 Number of symbol intervals for tone :
nsil = 15
 Number of symbol intervals for clock sync :
nalt = 15
ans =26-Jan-91
Informative remark : 8 per. 2cpfsk2rc h=.75 pn(1023) 333 hz, time=12:15
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=rcsynth(fidur*iprat);
clear oseq oseqr body
pack
oseq=ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat);
oseq=[oseq cpfsk(dseq,h,cr,iprat)];
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt-1+pnper*8);
head=oseqr(1:pnstrt-1);
echo off
>> save f8612f3h body
>> save f8622f3h head
>> diary off
```

2CPFSK4 1 kHz symbolrate total duration 19.4 s:

Filename	signal type	Duration: Samples	seconds
f1317b1k.mat	barker code	52	0.013
f151ps1s.mat	silence(pause)	4000	1
f8722f1k.mat	preamble (+ trans)	136	0.034
f8712f1k.mat	1 period data seq.	65504	16.376
f151ps1s.mat	silence(pause)	4000	1
f1317b1k.mat	barker code	52	0.013
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seqcpfsk123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. The preamble is preceeded by a 13-element bpsk-modulated Barker code . A 2047 bit long PN-sequence is used as a data sequence and fed to the CPFSK-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. This sequence is repeated 8 times to yield one period of a periodic CPFSK-sequence. The CPFSK-modulator has a 2RC (raised cosine) prefilter and the modulation index was 0.75. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init2cpfsk123
>> seqcpfsk123
f0 = 1000
 symbol rate (Hz):
```

```
symbrat = 1000
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 0 0 0 0 1 0
h = 0.7500
fidur = 2
 No of output symbols requested:
nsymb = 16476
 Number of symbol intervals for tone :
nsil = 15
 Number of symbol intervals for clock sync :
nalt = 15
ans =25-Jan-91
 Informative remark : 8 per. 1khz 2cpfsk2rc h=3/4 pn(2047) time=11:10
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=rcsynth(fidur*iprat);
clear oseq oseqr body
pack
oseq=ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat);
oseq=[oseq cpfsk(dseq,h,cr,iprat)];
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt-1+pnper*8);
head=oseqr(1:pnstrt-1);
echo off
>> save f8712f1k body
>> save f8722f1k head
>> exit
 67791866 flops.
```

8.7 Sequence 10

Filename	signal type	Duration: Samples	seconds
f1311b01.mat	barker code	52000	13
f151ps1s.mat	silence(pause)	4000	1
f8122f01.mat	preamble (+ trans)	108000	27
f8112f01.mat	1 period data seq.	252000	63
f8112f01.mat	1 period data seq.	252000	63
f8112f01.mat	1 period data seq.	252000	63
f8112f01.mat	1 period data seq.	252000	63
f8112f01.mat	1 period data seq.	252000	63
f8112f01.mat	1 period data seq.	252000	63
f8112f01.mat	1 period data seq.	252000	63
f8112f01.mat	1 period data seq.	252000	63
f8112f01.mat	1 period data seq.	252000	63
f8112f01.mat	1 period data seq.	252000	63
f8112f01.mat	1 period data seq.	252000	63
f8112f01.mat	1 period data seq.	252000	63
f8112f01.mat	1 period data seq.	252000	63
f8112f01.mat	1 period data seq.	252000	63
f8112f01.mat	1 period data seq.	252000	63
f8112f01.mat	1 period data seq.	252000	63
f8112f01.mat	1 period data seq.	252000	63
f8112f01.mat	1 period data seq.	252000	63
f8112f01.mat	1 period data seq.	252000	63
f8112f01.mat	1 period data seq.	252000	63
f8112f01.mat	1 period data seq.	252000	63
f151ps1s.mat	silence(pause)	4000	4
f1311b01.mat	barker code	52000	13
f151ps1s.mat	silence(pause)	4000	4

The files containing the preamble and the data sequence were generated by the matlab m-file "seqcpfsk123.m". The preamble consists of 10 symbol intervals with input = -1 followed by 10 alternating (1/-1) symbols. A BPSK-modulated 13-element Barker code preceeds the preamble. A 63 bit long PN-sequence is used as a data sequence and fed to the CPFSK-modulator which also interpolates the signal to yield a sampling frequency of 4 kHz. The CPFSK-modulator has a 2RC (raised cosine) prefilter and the modulation index was 0.75. NB! this sequnce is not a perfect period of a periodic CPFSK-sequence. Consequently there are a discontinuity at the end/dtart point of the body file f8212f03.mat.The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init2cpfsk123
>> seqcpfsk123
f0 = 1000
 symbol rate (Hz):
symbrat = 1
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 1
h = 0.7500
fidur = 2
 No of output symbols requested:
nsymb = 75
 Number of symbol intervals for tone :
nsil = 5
 Number of symbol intervals for clock sync :
```

```
nalt = 5
ans =23-Jan-91
 Informative remark : 2cpfsk2rc h=.75 pn(63) 1 hz time= 13:25
nbit=nsymb;
iprat=fix(fs/symbrat);
dseq=[-ones(1:nsil) cos(pi*[0:nalt-1])];
dlen=length(dseq);
pn=pnsequ(reginit,genpol,nbit);
dseq=[dseq 2*pn-1];
cr=rcsynth(fidur*iprat);
clear oseq oseqr body
pack
oseq=ipfill([-1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1],iprat);
oseq=[oseq cpfsk(dseq,h,cr,iprat)];
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:1:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(13+nsil+nalt);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt-1+pnper);
head=oseqr(1:pnstrt-1);
echo off
>> save f8112f01 body
>> save f8122f01 head
>> diary off
```

8.8 Sequence 11

	Coded 8PSK 3.33 hz symbolrate total duration 6 min. 43.8 s:		
Filename	**signal type**	**Duration: Samples**	**seconds**
f1312b03.mat	barker code	15600	3.9
f151ps1s.mat	silence(pause)	4000	1
f0228p03.mat	preamble (+ trans)	60001	15
f0218p03.mat	1 period data seq.	75600	18.9
f0218p03.mat	1 period data seq.	75600	18.9
f0218p03.mat	1 period data seq.	75600	18.9
f0218p03.mat	1 period data seq.	75600	18.9
f0218p03.mat	1 period data seq.	75600	18.9
f0218p03.mat	1 period data seq.	75600	18.9
f0218p03.mat	1 period data seq.	75600	18.9
f0218p03.mat	1 period data seq.	75600	18.9
f0218p03.mat	1 period data seq.	75600	18.9
f0218p03.mat	1 period data seq.	75600	18.9
f0218p03.mat	1 period data seq.	75600	18.9
f0218p03.mat	1 period data seq.	75600	18.9
f0218p03.mat	1 period data seq.	75600	18.9
f0218p03.mat	1 period data seq.	75600	18.9
f0218p03.mat	1 period data seq.	75600	18.9
f0218p03.mat	1 period data seq.	75600	18.9
f0218p03.mat	1 period data seq.	75600	18.9
f0218p03.mat	1 period data seq.	75600	18.9
f0218p03.mat	1 period data seq.	75600	18.9
f0218p03.mat	1 period data seq.	75600	18.9
f151ps1s.mat	silence(pause)	4000	1
f1312b03.mat	barker code	15600	3.9
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seq8pskspec.m". The preamble consists of 10 symbol interval of the carrier with phase -45 degrees followed by 10 symbols with +90 degrees shift relative to the previous symbol followed by another 10 symbols where the relative shift is 180 degrees. The preamble is concluded by a 13 element barker code. A 63 bit long PN-sequence is used as a data sequence and applied to a rate 2/3 convolutional coder. The symbols are generated from the coded sequence by assigning groups of 3 adjacent bits to an 8PSK symbol applying a special mapping rule according to the set partitioning used . This symbol sequence is interpolated to yield a sampling frequency of 4 kHz by filtering it through a truncated 4 symbol intervals long cosine roll-off filter with roll-off factor 0.5. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init8psk123
>> seq8pskcod123
f0 = 1000
 symbol rate (Hz):
symbrat = 3.3333
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 1
betha = 0.5000
corolnth = 2
 No of output symbols requested:
```

```
nsymb = 75
nsil = 10
n45 = 10
n180 = 10
ans =24-Jan-91
 Informative remark : 3.33 hz 8psk cod(2/3) pn(63) time =21:50
ngen=nsymb*2;
pn=pnsequ(reginit,genpol,ngen);
clear dseq
dseq=[ones(1:nsil) cos(pi*[1:n180]) cos(pi*n180)*exp(sqrt(-1)*pi/4*[1:n45])];
dseq=[dseq -1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1];
dlen=length(dseq);
dseq=[dseq mp8psk2by3cod(pn)];
iprat=fix(fs/symbrat);
cr=cosroll(betha,iprat,corolnth);
Warning: Divide by zero
Warning: Divide by zero
clear oseq oseqr
pack
oseq=interpfilt(cr,iprat,dseq);
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(dlen+3);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt+pnper-1);
head=oseqr(1:pnstrt-1);
echo off
>> save f0218p03 body
>> save f0228p03 head
>> diary off
```

	Coded 8PSK 10 hz symbolrate total duration 2 min 16.6 s:		
Filename	**signal type**	**Duration: Samples**	**seconds**
f1313b10.mat	barker code	5200	1.3
f151ps1s.mat	silence(pause)	4000	1
f0328p10.mat	preamble (+ trans)	20001	5
f0318p10.mat	1 period data seq.	25200	6.3
f0318p10.mat	1 period data seq.	25200	6.3
f0318p10.mat	1 period data seq.	25200	6.3
f0318p10.mat	1 period data seq.	25200	6.3
f0318p10.mat	1 period data seq.	25200	6.3
f0318p10.mat	1 period data seq.	25200	6.3
f0318p10.mat	1 period data seq.	25200	6.3
f0318p10.mat	1 period data seq.	25200	6.3
f0318p10.mat	1 period data seq.	25200	6.3
f0318p10.mat	1 period data seq.	25200	6.3
f0318p10.mat	1 period data seq.	25200	6.3
f0318p10.mat	1 period data seq.	25200	6.3
f0318p10.mat	1 period data seq.	25200	6.3
f0318p10.mat	1 period data seq.	25200	6.3
f0318p10.mat	1 period data seq.	25200	6.3
f0318p10.mat	1 period data seq.	25200	6.3
f0318p10.mat	1 period data seq.	25200	6.3
f0318p10.mat	1 period data seq.	25200	6.3
f0318p10.mat	1 period data seq.	25200	6.3
f0318p10.mat	1 period data seq.	25200	6.3
f0318p10.mat	1 period data seq.	25200	6.3
f151ps1s.mat	silence(pause)	4000	1
f1313b10.mat	barker code	5200	1.3
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seq8pskspec.m". The preamble consists of 10 symbol interval of the carrier with phase -45 degrees followed by 10 symbols with +90 degrees shift relative to the previous symbol followed by another 10 symbols where the relative shift is 180 degrees. The preamble is concluded by a 13 element barker code. A 63 bit long PN-sequence is used as a data sequence and applied to a rate 2/3 convolutional coder. The symbols are generated from the coded sequence by assigning groups of 3 adjacent bits to an 8PSK symbol applying a special mapping rule according to the set partitioning used . This symbol sequence is interpolated to yield a sampling frequency of 4 kHz by filtering it through a truncated 4 symbol intervals long cosine roll-off filter with roll-off facto- 0.5. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of tne matlab session(s) is given below:

```
>> init8psk123
>> seq8pskcod123
f0 = 1000
 symbol rate (Hz):
symbrat = 10
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 1
betha = 0.5000
corolnth = 2
 No of output symbols requested:
```

```
nsymb = 130
nsil = 10
n45 = 10
n180 = 10
ans =25-Jan-91
Informative remark : 10 hz 8psk cod(2/3) pn(63) time=09:20
ngen=nsymb*2;
pn=pnsequ(reginit,genpol,ngen);
clear dseq
dseq=[ones(1:nsil) cos(pi*[1:n180]) cos(pi*n180)*exp(sqrt(-1)*pi/4*[1:n45])];
dseq=[dseq -1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1];
dlen=length(dseq);
dseq=[dseq mp8psk2by3cod(pn)];
iprat=fix(fs/symbrat);
cr=cosroll(betha,iprat,corolnth);
Warning: Divide by zero
Warning: Divide by zero
clear oseq oseqr
pack
oseq=interpfilt(cr,iprat,dseq);
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(dlen+3);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt+pnper-1);
head=oseqr(1:pnstrt-1);
echo off
>> diary off
```

	Coded 8PSK 33.3 hz symbolrate total duration 1 min 21.8 s:		
Filename	**signal type**	**Duration: Samples**	**seconds**
f1314b30.mat	barker code	1560	0.39
f151ps1s.mat	silence(pause)	4000	1
f0428p30.mat	preamble (+ trans)	6001	1.5
f0418p30.mat	1 period data seq.	30600	7.65
f0418p30.mat	1 period data seq.	30600	7.65
f0418p30.mat	1 period data seq.	30600	7.65
f0418p30.mat	1 period data seq.	30600	7.65
f0418p30.mat	1 period data seq.	30600	7.65
f0418p30.mat	1 period data seq.	30600	7.65
f0418p30.mat	1 period data seq.	30600	7.65
f0418p30.mat	1 period data seq.	30600	7.65
f0418p30.mat	1 period data seq.	30600	7.65
f0418p30.mat	1 period data seq.	30600	7.65
f151ps1s.mat	silence(pause)	4000	1
f1314b30.mat	barker code	1560	0.39
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seq8pskspec.m". The preamble consists of 10 symbol interval of the carrier with phase -45 degrees followed by 10 symbols with +90 degrees shift relative to the previous symbol followed by another 10 symbols where the relative shift is 180 degrees. The preamble is concluded by a 13 element barker code. A 255 bit long PN-sequence is used as a data sequence and applied to a rate 2/3 convolutional coder. The symbols are generated from the coded sequence by assigning groups of 3 adjacent bits to an 8PSK symbol applying a special mapping rule according to the set partitioning used . This symbol sequence is interpolated to yield a sampling frequency of 4 kHz by filtering it through a truncated 4 symbol intervals long cosine roll-off filter with roll-off factor 0.5. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init8psk123
>> seq8pskcod123
f0 = 1000
 symbol rate (Hz):
symbrat = 33.3333
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 1 1 1 0
betha = 0.5000
corolnth = 2
 No of output symbols requested:
nsymb = 550
nsil = 10
n45 = 10
n180 = 10
ans =25-Jan-91
Informative remark : 33.33 hz 8psk cod(2/3) pn(255) time =10:00
ngen=nsymb*2;
pn=pnsequ(reginit,genpol,ngen);
clear dseq
dseq=[ones(1:nsil) cos(pi*[1:n180]) cos(pi*n180)*exp(sqrt(-1)*pi/4*[1:n45])];
dseq=[dseq -1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1];
dlen=length(dseq);
dseq=[dseq mp8psk2by3cod(pn)];
iprat=fix(fs/symbrat);
cr=cosroll(betha,iprat,corolnth);
Warning: Divide by zero
Warning: Divide by zero
clear oseq oseqr
pack
oseq=interpfilt(cr,iprat,dseq);
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(dlen+3);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt+pnper-1);
head=oseqr(1:pnstrt-1);
echo off
>> save f0418p30 body
>> save f0428p30 head
>> diary off
```

Filename	Coded 8PSK 100 hz symbolrate total duration 29.2 s: signal type	Duration: Samples	seconds
f1315b1h.mat	barker code	520	0.13
f151ps1s.mat	silence(pause)	4000	1
f0528p1h.mat	preamble (+ trans)	2001	0.5
f0518p1h.mat	1 period data seq.	20440	5.11
f0518p1h.mat	1 period data seq.	20440	5.11
f0518p1h.mat	1 period data seq.	20440	5.11
f0518p1h.mat	1 period data seq.	20440	5.11
f0518p1h.mat	1 period data seq.	20440	5.11
f0518p1h.mat	1 period data seq.	20440	5.11
f0518p1h.mat	1 period data seq.	20440	5.11
f0518p1h.mat	1 period data seq.	20440	5.11
f0518p1h.mat	1 period data seq.	20440	5.11
f0518p1h.mat	1 period data seq.	20440	5.11
f151ps1s.mat	silence(pause)	4000	1
f1315b1h.mat	barker code	520	0.13
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seq8pskspec.m". The preamble consists of 10 symbol interval of the carrier with phase -45 degrees followed by 10 symbols with +90 degrees shift relative to the previous symbol followed by another 10 symbols where the relative shift is 180 degrees. The preamble is concluded by a 13 element barker code. A 511 bit long PN-sequence is used as a data sequence and applied to a rate 2/3 convolutional coder. The symbols are generated from the coded sequence by assigning groups of 3 adjacent bits to an 8PSK symbol applying a special mapping rule according to the set partitioning used . This symbol sequence is interpolated to yield a sampling frequency of 4 kHz by filtering it through a truncated 4 symbol intervals long cosine roll-off filter with roll-off factor 0.5. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init8psk123
>> seq8pskcod123
f0 = 1000
 symbol rate (Hz):
symbrat = 100
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 1 0 0 0
betha = 0.5000
corolnth = 2
 No of output symbols requested:
nsymb = 1100
nsil = 10
n45 = 10
n180 = 10
ans =25-Jan-91
 Informative remark : 100 hz 8psk cod(2/3) pn(511) time= 09:50
ngen=nsymb*2;
pn=pnsequ(reginit,genpol,ngen);
clear dseq
dseq=[ones(1:nsil) cos(pi*[1:n180]) cos(pi*n180)*exp(sqrt(-1)*pi/4*[1:n45])];
```

```
dseq=[dseq -1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1];
dlen=length(dseq);
dseq=[dseq mp8psk2by3cod(pn)];
iprat=fix(fs/symbrat);
cr=cosroll(betha,iprat,corolnth);
Warning: Divide by zero
Warning: Divide by zero
clear oseq oseqr
pack
oseq=interpfilt(cr,iprat,dseq);
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(dlen+3);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt+pnper-1);
head=oseqr(1:pnstrt-1);
echo off
>> save f0518p1h body
>> save f0528p1h head
>> diary off
```

Coded 8PSK 333 hz symbolrate total duration 33.9 s:

Filename	signal type	Duration: Samples	seconds
f1316b3h.mat	barker code	156	0.04
f151ps1s.mat	silence(pause)	4000	1
f0628p3h.mat	preamble (+ trans)	601	0.15
f0618p3h.mat	1 period data seq.	12276	3.07
f0618p3h.mat	1 period data seq.	12276	3.07
f0618p3h.mat	1 period data seq.	12276	3.07
f0618p3h.mat	1 period data seq.	12276	3.07
f0618p3h.mat	1 period data seq.	12276	3.07
f0618p3h.mat	1 period data seq.	12276	3.07
f0618p3h.mat	1 period data seq.	12276	3.07
f0618p3h.mat	1 period data seq.	12276	3.07
f0618p3h.mat	1 period data seq.	12276	3.07
f0618p3h.mat	1 period data seq.	12276	3.07
f151ps1s.mat	silence(pause)	4000	1
f1316b3h.mat	barker code	156	0.04
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seq8pskspec.m". The preamble consists of 10 symbol interval of the carrier with phase -45 degrees followed by 10 symbols with +90 degrees shift relative to the previous symbol followed by another 10 symbols where the relative shift is 180 degrees. The preamble is concluded by a 13 element barker code. A 1023 bit long PN-sequence is used as a data sequence and applied to a rate 2/3 convolutional coder. The symbols are generated from the coded sequence by assigning groups of 3 adjacent bits to an 8PSK symbol applying a special mapping rule according to the set partitioning used . This symbol sequence is interpolated to yield a sampling frequency of 4 kHz by filtering it through a truncated 4 symbol intervals long cosine roll-off filter with roll-off factor 0.5. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>> init8psk123
>> seq8pskcod123
f0 = 1000
 symbol rate (Hz):
symbrat = 333.3300
fs = 4000
 coefficients of generating function:
genpol = 1 0 0 0 0 0 0 1 0 0
betha = 0.5000
corolnth = 2
 No of output symbols requested:
nsymb = 2100
nsil = 10
n45 = 10
n180 = 10
ans =25-Jan-91
Informative remark : 333 hz 8psk cod(2/3) pn(1023) time=10:10
ngen=nsymb*2;
pn=pnsequ(reginit,genpol,ngen);
clear dseq
dseq=[ones(1:nsil) cos(pi*[1:n180]) cos(pi*n180)*exp(sqrt(-1)*pi/4*[1:n45])];
dseq=[dseq -1 1 -1 1 -1 -1 1 1 -1 -1 -1 -1 -1];
dlen=length(dseq);
dseq=[dseq mp8psk2by3cod(pn)];
iprat=fix(fs/symbrat);
cr=cosroll(betha,iprat,corolnth);
Warning: Divide by zero
Warning: Divide by zero
clear oseq oseqr
pack
oseq=interpfilt(cr,iprat,dseq);
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(dlen+3);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt+pnper-1);
head=oseqr(1:pnstrt-1);
echo off
>> save f0618p3h body
>> save f0628p3h head
>> diary off
```

Coded 8PSK 1 khz symbolrate total duration 13.3 s:

Filename	signal type	Duration: Samples	seconds
f1317b1k.mat	barker code	52	0.013
f151ps1s.mat	silence(pause)	4000	1
f0728p1k.mat	preamble (+ trans)	201	0.05
f0718p1k.mat	1 period data seq.	8188	2.047
f0718p1k.mat	1 period data seq.	8188	2.047
f0718p1k.mat	1 period data seq.	8188	2.047
f0718p1k.mat	1 period data seq.	8188	2.047
f0718p1k.mat	1 period data seq.	8188	2.047
f151ps1s.mat	silence(pause)	4000	1
f1317b1k.mat	barker code	52	0.013
f151ps1s.mat	silence(pause)	4000	1

The files containing the preamble and the data sequence were generated by the matlab m-file "seq8pskspec.m". The preamble consists of 10 symbol interval of the carrier with phase -45 degrees followed by 10 symbols with +90 degrees shift relative to the previous symbol followed by another 10 symbols where the relative shift is 180 degrees. The preamble is concluded by a 13 element barker code. A 2047 bit long PN-sequence is used as a data sequence and applied to a rate 2/3 convolutional coder. The symbols are generated from the coded sequence by assigning groups of 3 adjacent bits to an 8PSK symbol applying a special mapping rule according to the set partitioning used . This symbol sequence is interpolated to yield a sampling frequency of 4 kHz by filtering it through a truncated 4 symbol intervals long cosine roll-off filter with roll-off factor 0.5. The resulting sequence is modulated on to a 4 kHz carrier. An edited transcript of the matlab session(s) is given below:

```
>>init8psk123
>>seq8pskcod123
f0=1000
symbolrate(Hz):
symbrat=1000
fs=4000
coefficientsofgeneratingfunction:
genpol=10000000010
betha=0.5000
corolnth=2
Noofoutputsymbolsrequested:
nsymb=4500
nsil=10
n45=10
n180=10
ans=25-Jan-91
Informativeremark:1khz8pskcod(2/3)pn(2047)time=10:25
ngen=nsymb*2;
pn=pnsequ(reginit,genpol,ngen);
cleardseq
dseq=[ones(1:nsil)cos(pi*[1:n180])cos(pi*n180)*exp(sqrt(-1)*pi/4*[1:n45])];
dseq=[dseq-11-11-1-111-1-1-1-1-1];
dlen=length(dseq);
dseq=[dseqmp8psk2by3cod(pn)];
iprat=fix(fs/symbrat);
cr=cosroll(betha,iprat,corolnth);
```

```
Warning:Dividebyzero
Warning:Dividebyzero
clearoseqoseqr
pack
oseq=interpfilt(cr,iprat,dseq);
oseq=oseq.*exp((sqrt(-1)*2*pi*(f0/fs))*(0:length(oseq)-1));
pnstrt=1+length(cr)+iprat*(dlen+3);
pnper=(2^length(genpol)-1)*iprat;
oseqr=real(oseq);
oseqr=oseqr/max(abs(oseqr));
body=oseqr(pnstrt:pnstrt+pnper-1);
head=oseqr(1:pnstrt-1);
echooff
>>savef0718p1kbody
>>savef0728p1khead
>>diaryoff
```

8.9 Sequence 12

	coded 8PSK 1 hz symbolrate total duration 21 min. 19 s		
Filename	**signal type**	**Duration: Samples**	**seconds**
f1311b01.mat	barker code	52000	13
f151ps1s.mat	silence(pause)	4000	1
f0128p01.mat	preamble (+ trans)	200001	50
f0118p01.mat	1 period data seq.	252000	63
f0118p01.mat	1 period data seq.	252000	63
f0118p01.mat	1 period data seq.	252000	63
f0118p01.mat	1 period data seq.	252000	63
f0118p01.mat	1 period data seq.	252000	63
f0118p01.mat	1 period data seq.	252000	63
f0118p01.mat	1 period data seq.	252000	63
f0118p01.mat	1 period data seq.	252000	63
f0118p01.mat	1 period data seq.	252000	63
f0118p01.mat	1 period data seq.	252000	63
f0118p01.mat	1 period data seq.	252000	63
f0118p01.mat	1 period data seq.	252000	63
f0118p01.mat	1 period data seq.	252000	63
f0118p01.mat	1 period data seq.	252000	63
f0118p01.mat	1 period data seq.	252000	63
f0118p01.mat	1 period data seq.	252000	63
f0118p01.mat	1 period data seq.	252000	63
f0118p01.mat	1 period data seq.	252000	63
f0118p01.mat	1 period data seq.	252000	63
f0118p01.mat	1 period data seq.	252000	63
f0118p01.mat	1 period data seq.	252000	63
f151ps1s.mat	silence(pause)	4000	4
f1311b01.mat	barker code	52000	13
f151ps1s.mat	silence(pause)	4000	4

The files containing the preamble and the data sequence were generated by the matlab m-file "seq8pskspec.m". The preamble consists of 10 symbol interval of the carrier with phase -45 degrees followed by 10 symbols with +90 degrees shift relative to the previous symbol followed by another 10 symbols where the relative shift is 180 degrees. The preamble is concluded by a 13 element barker code. A 63 bit long PN-sequence is used as a data sequence and applied to a rate 2/3 convolutional coder. The symbols are generated from the coded sequence by assigning groups of 3 adjacent bits to an 8PSK symbol applying a special mapping rule according to the set partitioning used . This symbol sequence is interpolated to yield a sampling frequency of 4 kHz by filtering it through a truncated 4 symbol intervals long cosine roll-off filter with roll-off factor 0.5. The resulting sequence is modulated on to a 4 kHz carrier.

9.0 Appendix 1: Cruise log transcript (Josko Catipovic)

The following is a verbatim transcript of the science log aboard the R/V William McGaw during the telemetry cruise, 1/28/91 - 2/2/91:

1/28/91

0900 Arrive dock at Moss Landing. Call R/V Point Sur. Contact with A. B. Baggerroer: array detached, will attempt to reattach 1t 10:30. Winds ~ 20 kts, seas ~ 3 ft, swell ~ 6 ft Agree to call back at 14:00.

14:10 Attempt to call Point Sur at 8290 MHz. No response.

14:30 Still not hooked up to the arrray, 6' - 8' swells. Waiting for heard Island source to go back online. Schedule next communication for 17:00 hrs and 18:00 backup time. Ship is at 35 degrees 31.3'N, 123 degrees 02.3'W. have 4'-6' swell, 2'-3' seas, wind NW ~ 25 kts Can not remain attached to the array

17:00 Point Sur requests 100' of shock cord. R/V Mcgaw will get under way at 20:00 and deliver the cord

19:00 All transmit gear is working. Backup PC has file reading problems. Primary unit OK. Power amp working OK.

20:00 Get underway to meet Point Sur. Point sur is at 35 degrees 33'N, 123 degrees 02'W

1/29/91

07:00 Passed shock cord to Point Sur. Steaming north to 60 km range.

13:00 Arrive station - rough weather - all 4 science crew sick. winds ~ 25 kts seas ~ 8ft

14:39 Start transmitting block1, - range ~40 nm

15:30 Stop transmitting block1, - range ~40 nm

17:42 start block2 - range ~37 nm ((log time error likely)

17:49 Stop block2 (likely not recorded last 15 min or so)

17:50 ABB calls HF: array trapped in port screw. Will proceed to Monterey on starboard screw. Mcgaw leaves station to rendes-vous with Point Sur.

23:00 Meet Point Sur @ ~ 35 degrees N 123 degrees W. They were able to retrieve the array and get underway

1/30/91

09:00 McGaw arrives at Moss Landing

13:00 Point Sur arrives Monterey

17:15 Point Sur leaves Monterey with repaired array - proceeding to 34 degrees N, 122 degrees W

18:00 McGaw leaves Moss Landing - weather forcast: 15 - 30 kt southerly, 12' - 17' seas

20:00 Talk to ABBPoint Sur only about 10 nm ahead. They will drop a sonobuoy at 06:30 tomorrow. Plan is to stop McGaw at 05:30. Point Sur is heading for 122 degrees W, 34 degrees N

1/31/91

06:15 Point Sur drops a sonobuoy at 34 degrees 20'N, 122 degrees 0.0' W

07:00 Point Sur reports sonobuoy working - expect us active at 08:30

08:00 McGaw arrives at station at 122degrees W, 34 degrees 57'N

08:30 Start transmittingnb4 - sea lion in water 100' from ship - does not seem to notice acoustic waveforms Mcgaw at 34 degrees 58'N 121 degrees 59.7'W Pt. Sur at 34 degrees 18.5'N 122 degrees 00.25'W Signal input level 7.5 v peak-to-peak Power amp output 495 Vrms

09:42 Stop transmitting. Pt. Sure receives 30+ dB SNR

11:30 Start transmitting nb3 - weather calm - array deployed and receiving

12:35 PC hung up on seq32 - Restart PC with seq 32, seq 33, seq 34, seq 35

12:55 swapped D/A board with a spare - transmitter back online. PC overheating was the likely problem. - Steaming course 340 degrees.

14:15 Ship at 122 degrees 07'W, 35 degrees 09.00'N, 51 nm course 174 to Pt. Sur

14:31 Start PC with seq 32, seq 33, seq 34, seq 35 (left over from previous transmission) 2 min. of files "left over" from 11:30 transmission. Follow by nb5

15:40 Stop transmitting

15:41 Pt. Sur confirms signal quality high. They have 12 phones left on the vertical array

15:50 Source secured on deck and move ~ 7 nm

16:30 Attempted to balance gain on the replacement board - no sucess - negative excursions ~ -2.1v, positive excursions ~ 2.2v

17:00 On station 59 nm from Pt. Sur

17:25 Talk to ABB . Heard source is down, i.e. we will be transmitting continuously - advise before each transmission

17:30 start nb6 - range 59 nm

18:40 stop nb6 - steam north 11.5 nm to 73 nm range

20:25 stop ship at 72 nm range - contact Pt. Sur

20:33 start transmisison of block1 - input signal 7.5v peak-to-peak

21:30 stop transmission - Pt Sur is at 34.20.2N, 122.04.2W Mcgaw is at 35.30.6N, 122.16.6W

21:35 start block2 - input signal is 7.5V peak-to-peak

22:41 stop block2 - call Pt. Sur to alert them to change recording reels

22:50 start block3

23:55 stop block3

2/1/91

00:01 check with J. Miller on Pt. Sur. Signal quality good to excellent(~20 dB)

(Jim Preisig begins watch)

00:05 start block4 signal ~7.5v peak-to-peak

01:11 finish block4 - reported signal level 15-40 dB at Pt. Sur

01:20 start block5

02:25 stop block5 - signal ~ 7.5V peak-to-peak

02:35 start block6 - signal ~7.5V peak-to-peak

03:41 stop block6

03:48 start block7

04:57 stop block7

05:15 talked to Keith at Point Sur. reported our signal level at 100 dB re μPa and SNR of 40 dB. (Noise floor ~ 60 dB). 11 channels received on the array. We are now steaming to a new position 108 nm from Point Sur. ETA 0815. We will contact Point Sur at ~ 08:15 to coordinate transmisison of next group of signals

(Josko Catipovic begins watch)

08:07 arrived at station 36.00N 122.35.10W

08:20 Contact with Pt. Sur. They are staerting to record - we'll ahve a disk change in approx 2-3 hrs. call pt. Sur after block2 re disk change

08:25 start block1 - signal is 9V eak-to-peak Power amp output 600V rms

08:26 start VCR recorder - wind calm since 1/31/91 06:30

08:30 wind picking up - ~10 kts

09:03 wind SE ~ 15 kts - occasional breakers ~1-2 ft swell from northwest

09:21 stop block1

09:25 start block2 - wind SE 10-15 kts, 1'-2' swell from NW, signal 9V peak-to-peakMcgaw at 36.00.95N, 122.35.85W

10:22 small seal at stern - having a good time - apparently oblivious to sound source. likes bread but not frozen fish

10:30 stop block2

10:35 talk to Keith - data quality excellent (30-40 dB) receiving on 11 channels

10:37 start block3 - 9 V peak-to-peak Pt. Sur is at 34.22.7N, 122.08.1W

11:42 stop block3 - McGaw is at 36.03.02N, 122.36.79W

11:45 talk to ABB Strong encouragement to move out to 4 convergence zones. Wil pick up gear and move to ~ 150 nm range

14:45 stopped @ 36:31.00N, 122.45.00W wind 15 kts, swell 2' seas 1'-2', range to Pt. Sur 132 nm Pt. Sur is receiving @ 25 dB SNR. wind there ~5 kts, 2' swell

14:55 start recording block2

15:00 winds SE 15-18 kts McGaw drigfting NE @ ~ 3/4 knot

15:15 ship pitching - estimate ∓1.5mexcursions with 6 sec period - winds and seas increasing ~ 18 kts 2'seas

15:20 Bridge check: ground speed 0.9 kts heading north

16:04 stop block2

16:05 start nb3

16:40 winds increasing to 18-20 knots seas 2-3 ft transducer cable angle ~20 degrees - extended additional 5m of cable to lower source to approx. 100 m depth

17:12 stop nb3

17:15 start nb4-2

18:20 stop nb4-2 - 20-30 dB SNR - received Noise SPL 50-60 dB. At Pt. Sur ~ 10 kts SE 3' swell Pt sur is at 34.23.25N, 122.09.1W

18:30 start nb5 - McGaw at 36.34.36N, 122.45.90W

19.37 stop nb5

19:43 start nb6-2 - wind SE ~20 kts, seas ~4'-5', swell 3'

20:40 wind SE ~ 25 kts, seas, 6' - 8' small swell, McGaw is at 36.36.04N, 122.46.50W

20:44 stop nb6-2 - Pt. Sur is at 34.23.5N, 122.10.4W

21.05 secured and heading to Moss Landing - wind ~25-30 kts, seas 6' - 8'

2/2/91

02:00	arrive Moss Landing
02:20	Secure at Moss Landing pier

10.0 Appendix 2: ReferencesThese are the "C" language structures that describe the information held in directory entries and data record

10.1 Directory Entry Format

```
\struct dir_pair
{
 long   ptr;                          /* pointer to file segment starting LBA */
                                      /* doesn't include dir entries */
 long   blks;                         /* # blocks in file segment */
};

struct dir_entry
{
 unsigned char dekey[4];              /* directory entry key code, "ODIR" */
 unsigned char fname[16];             /* ascii file name <=15 chars */
                                       /* terminated with a 00h */
 unsigned int detime[4];              /* date of entry */
 long   clba;                         /* this entry's LBA in comp dir area */
 long   dlba;                         /* this entry's LBA in disp dir area */
 long   bytes;                        /* total # bytes in this file */
 struct dir_pair dp[122];             /* pointer/length pairs */
 long   decont;                        /* continuation ptr or zero if none */
 unsigned char dekeyl[4];              /* repeat of key code */
};
```

10.2 Data Record Header

10.3

```
struct data_rec_hdr
{
 unsigned char rhkey[4];              /* header key, "DATA" */
 unsigned char proj[16];              /* project name, ascii */
 unsigned char extype[32];            /* exp type, ascii */
 unsigned int exp;                     /* exp number */
 unsigned int date[2];                /* ODAS date...(year, Jday) */
 unsigned int time[2];                /* ODAS time..(minutes, ms) */
 unsigned int ch;                     /* # channels */
 unsigned int bkch;                   /* # blocks per demuxed channel */
                                      /* if zero, data is not demuxed */
 long   npts;                          /* # sample periods */
```

```
float    rhfs;                          /* sample rate in Hz */
unsigned int rlen;                      /* record length in blks, includes */
                                        /* sector used for record header */
unsigned int rec;                       /* number of record that follows */
char    rhlat[16];                      /* lat, ascii DDD MM SS.T N or E */
char    rhlng[16];                      /* long, ascii DDD MM SS.T E or W */
unsigned int lpf[256];                  /* LPF setting (Hz) for up to 256 ch */
int     slew_delay;                     /* slew delay...GRA parameter */
unsigned int microsec;                  /* microseconds, <1000 */
unsigned int sample_mode;               /* 0 = normal, 1 = offset updated */
unsigned int preamp_gain;               /* linear gain other than ranged */
unsigned int sample_period;             /* sample period in microsec */
unsigned int buffer_num;                /* buffer number this record */
long    rectime;                        /* record time in microsec */
unsigned char other[382];               /* space left in record header */
                                        /* possibly used for sensor locs */
unsigned char rhkeyl[4];                /* end of rec header key "DATA" */
};
```

The following "C" routine has been used to convert the stored 2-byteraw data floating point format to native single precision floating point.

```
/*      normize.c
 *
 *      routine to normalize 2-byte raw data samples, GRA gain in upper
 *      3 bits of the integer and the 13 most significant ADC bits in
 *      the remaining 13 bits. The bit assignments are as follows:
 *      The gain and mantissa bits are gnd true. The mantissa is
 *      2's complement with bit 12 (msb) the sign bit. This format
 *      is same as for the old and new GRA's. The fixed gain of the
 *      GRA preamp is accommodated.
 *
 *      bit 15                  GRA bit 2 (msb)
 *      bit 14                  GRA bit 1
 *      bit 13                  GRA bit 0 (lsb)
 *      bit 12                  ADC bit 13 (msb) (sign bit)
 *      bit 11                  ADC bit 12
 *      bit 10                  ADC bit 11
 *      bit 9                   ADC bit 10
 *      bit 8                   ADC bit 9
 *      bit 7                   ADC bit 8
 *      bit 6                   ADC bit 7
 *      bit 5                   ADC bit 6
 *      bit 4                   ADC bit 5
 *      bit 3                   ADC bit 4
 *      bit 2                   ADC bit 3
 *      bit 1                   ADC bit 2
 *      bit 0                   ADC bit 1 (next to lsb from ADC, lsb not
used)
 *
 *
 *      The function performed:
 *      normalized value = (raw value<<3) * 10v / 4096 / 8 / gain / pag
```

```
*/

void normize(bufin,bufout,npts,pag)
unsigned *bufin;
float *bufout;
int npts,pag;
{
 int i,mantissa;
 unsigned gain;

 if(pag==1)
 {
 for(i=0;i<npts;i++)
 {
 mantissa = (int)(~bufin[i] << 3);
 gain = (~bufin[i] & 0xe000);
 switch(gain)
 {
        case(0x8000):                    /* gain = 4096 */
        bufout[i] = 7.4505805e-8 * mantissa;
        break;
        case(0x6000): /* gain = 512 */
        bufout[i] = 5.9604644e-7 * mantissa;
        break;
        case(0x4000):                    /* gain = 64 */
        bufout[i] = 4.7683715e-6 * mantissa;
        break;
        case(0x2000):                    /* gain = 8 */
        bufout[i] = 3.8146972e-5 * mantissa;
        break;
        case(0x0000):                    /* gain = 1 */
        bufout[i] = 3.0517578e-4 * mantissa;
        break;
        default:                         /* bogus gain value */
        bufout[i] = 0.0;
        break;
        }
 }
 }
 else
 {
 for(i=0;i<npts;i++)
 {
 mantissa = (int)(~bufin[i] << 3);
 gain = (~bufin[i] & 0xe000);
 switch(gain)
 {
        case(0x8000):                    /* gain = 4096 */
        bufout[i] = 7.4505805e-8 * mantissa/pag;
        break;
        case(0x6000): /* gain = 512 */
        bufout[i] = 5.9604644e-7 * mantissa/pag;
        break;
        case(0x4000):                    /* gain = 64 */
```

```
        bufout[i] = 4.7683715e-6 * mantissa/pag;
        break;
        case(0x2000):                    /* gain = 8 */
        bufout[i] = 3.8146972e-5 * mantissa/pag;
        break;
        case(0x0000):                    /* gain = 1 */
        bufout[i] = 3.0517578e-4 * mantissa/pag;
        break;
        default:                         /* bogus gain value */
        bufout[i] = 0.0;
        break;
        }
    }
  }
}
\end{verbatim}

\end{document}
```

11.0 Appendix 3: Heard Island Telemetry Optical Disk Data Files

HEARD ISLAND 1-2-3- TELEMETRY OPTICAL DISK DATA FILES

OPTIMEM DIRECTORY LIST UTILITY.....disk s/n 42734-1, SIDE = 1.2GB
01292228.hdr: 5248 bytes in 6 blocks @ LBA = 2 : 01/29/91 22:28
01292228.dat:1037010048 bytes in 1012706 blocks @ LBA = 9 : 01/29/91 22:28
159782 blocks left

OPTIMEM DIRECTORY LIST UTILITY.....disk s/n 42734-2, SIDE = 1.2GB
01300129.hdr: 5248 bytes in 6 blocks @ LBA = 2 : 01/30/91 01:29
01300129.dat: 679096448 bytes in 663181 blocks @ LBA = 9 : 01/30/91 01:37
01300224.hdr: 5120 bytes in 5 blocks @ LBA = 663191 : 01/30/91 02:24
01300224.dat: 2104320 bytes in 2055 blocks @ LBA = 663197 : 01/30/91 02:24
01311459.hdr: 5248 bytes in 6 blocks @ LBA = 665253 : 01/31/91 14:59
01311459.dat: 171069568 bytes in 167061 blocks @ LBA = 665260 : 01/31/91 14.59
01312232.hdr: 5248 bytes in 6 blocks @ LBA = 832322 : 01/31/91 22:32
01312232.dat: 347504768 bytes in 339361 blocks @ LBA = 832329 : 01/31/91 22:32
801 blocks left

OPTIMEM DIRECTORY LIST UTILITY.....disk s/n 42736-1, SIDE = 1.2GB
02010127.hdr: 5248 bytes in 6 blocks @ LBA = 2 : 02/01/91 01:27
02010127.dat: 388221056 bytes in 379123 blocks @ LBA = 9 : 02/01/91 01:28
02010424.dat: 386640896 bytes in 377579 blocks @ LBA = 379133 : 02/01/91 04:25
02010533.dat: 377191424 bytes in 368351 blocks @ LBA = 756713 : 02/01/91 05:36
47431 blocks left

OPTIMEM DIRECTORY LIST UTILITY.....disk s/n 42736-2, SIDE = 1.2GB
02010643.dat: 848877568 bytes in 828982 blocks @ LBA = 2 : 02/01/91 06:51
343514 blocks left

OPTIMEM DIRECTORY LIST UTILITY.....disk s/n 43422-1, SIDE = 1.2GB
02010926.hdr: 5248 bytes in 6 blocks @ LBA = 2 : 02/01/91 09:26
02010926.dat: 382708864 bytes in 373740 blocks @ LBA = 9 : 02/01/91 09:26
02011037.hdr: 5248 bytes in 6 blocks @ LBA = 373750 : 02/01/91 10:37
02011037.dat: 400032896 bytes in 390658 blocks @ LBA = 373757 : 02/01/91 10:37
02011149.hdr: 5248 bytes in 6 blocks @ LBA = 764416 : 02/01/91 11:49
02011149.dat: 417356928 bytes in 407576 blocks @ LBA = 764423 : 02/01/91 11:50
494 blocks left

OPTIMEM DIRECTORY LIST UTILITY.....disk s/n 43422-2, SIDE = 1.2GB
02011620.hdr: 5248 bytes in 6 blocks @ LBA = 2 : 02/01/91 16:23
02011620.dat:1174102144 bytes in 1146585 blocks @ LBA = 9 : 02/01/91 16:24
25903 blocks left

OPTIMEM DIRECTORY LIST UTILITY.....disk s/n 42737-1, SIDE = 1.2GB
02011955.hdr: 5248 bytes in 6 blocks @ LBA = 2 : 02/01/91 19:55
02011955.dat: 125998208 bytes in 123046 blocks @ LBA = 9 : 02/01/91 19:55
02012251.hdr: 5248 bytes in 6 blocks @ LBA = 123056 : 02/01/91 22:51
02012251.dat:1074095232 bytes in 1048922 blocks @ LBA = 123063 : 02/01/91 22:51
510 blocks left

OPTIMEM DIRECTORY LIST UTILITY.....disk s/n 42737-2, SIDE = 1.2GB
02020205.hdr: 5248 bytes in 6 blocks @ LBA = 2 : 02/02/91 02:05
02020205.dat:1007161472 bytes in 983557 blocks @ LBA = 9 : 02/02/91 02:06
188931 blocks left

12.0 Appendix 4: Optimem telemetry file headers

*** 1991 HEARD ISLAND DATA ACQUISITON SYSTEM ***
*** WITH OPTICAL DISK STORAGE ***

START OF DATA RECORDING.....
JDAY: 29 YEAR: 1991
GMT: 22:28:55.214
DISK #: 42734-1

PARAMETERS FROM FILE HEADER

PROJECT: HEARD ISLAND
EXPERIMENT TYFE: TELEMETRY
EXPERIMENT #: 12
TOTAL# CHANNELS: 32
SAMPLE RATE: 4000.000 Hz
SAMPLE PERIOD: 250 usec
DATA REC LENGTH: 1025 kbytes
LATITUDE: 35,29,21
LONGITUDE: 123,03,56

DATA BUFFER SIZE: 1048576
TIME/BUFFER: 4.096 secs
TIME TO DISK FULL: 1 hr, 18 min, 1 sec
PTS/CH/BUFFER: 16384

**** AFTER EXPERIMENT # 12 ****

BYTES WRITTEN: 1037010048
BLOCKS WRITTEN: 1012706
RECORDS WRITTEN: 988
START LBA: 9
FILE AREA DE @: 8
COMP AREA DE @: 1172497
BLOCKS REMAINING: 159782

*** 1991 HEARD ISLAND DATA ACQUISITON SYSTEM ***
*** WITH OPTICAL DISK STORAGE ***

START OF DATA RECORDING.....
JDAY: 30 YEAR: 1991
GMT: 1:37:11.454
DISK #: 42734-2

PARAMETERS FROM FILE HEADER

PROJECT: HEARD ISLAND
EXPERIMENT TYPE: TELEMETRY
EXPERIMENT #: 14
TOTAL# CHANNELS: 32
SAMPLE RATE: 4000.000 Hz
SAMPLE PERIOD: 250 usec
DATA REC LENGTH: 1025 kbytes
LATITUDE: 35,29,21
LONGITUDE: 123,03,56

DATA BUFFER SIZE: 1048576
TIME/BUFFER: 4.096 secs
TIME TO DISK FULL: 1 hr, 18 min, 1 sec
PTS/CH/BUFFER: 16384

**** AFTER EXPERIMENT # 14 ****

BYTES WRITTEN: 679096448
BLOCKS WRITTEN: 663181
RECORDS WRITTEN: 647
START LBA: 9
FILE AREA DE @: 8
COMP AREA DE @: 1172497
BLOCKS REMAINING: 509307

*** 1991 HEARD ISLAND DATA ACQUISITON SYSTEM ***
*** WITH OPTICAL DISK STORAGE ***

START OF DATA RECORDING.....
JDAY: 31 YEAR: 1991
GMT: 14:59:57.416
DISK #: 42734-2

PARAMETERS FROM FILE HEADER

PROJECT: HEARD ISLAND
EXPERIMENT TYPE: TELEMETRY
EXPERIMENT #: 15
TOTAL# CHANNELS: 1
SAMPLE RATE: 4000.000 Hz
SAMPLE PERIOD: 250 usec
DATA REC LENGTH: 129 kbytes
LATITUDE: 34,19,55
LONGITUDE: 122,00,00

DATA BUFFER SIZE: 131072
TIME/BUFFER: 16.384 secs
TIME TO DISK FULL: 17 hr, 53 min, 41 sec
PTS/CH/BUFFER: 65536

**** AFTER EXPERIMENT # 15 ****

BYTES WRITTEN: 171069568
BLOCKS WRITTEN: 167061
RECORDS WRITTEN: 1295
START LBA: 665260
FILE AREA DE @: 665259
COMP AREA DE @: 1172493
BLOCKS REMAINING: 340172

*** 1991 HEARD ISLAND DATA ACQUISITON SYSTEM ***
*** WITH OPTICAL DISK STORAGE ***

START OF DATA RECORDING.....
JDAY: 31 YEAR: 1991
GMT: 22:32:29.372
DISK #: 42734-2

PARAMETERS FROM FILE HEADER

PROJECT: HEARD ISLAND
EXPERIMENT TYPE: TELEMETRY
EXPERIMENT #: 17
TOTAL# CHANNELS: 11
SAMPLE RATE: 4000.000 Hz
SAMPLE PERIOD: 250 usec
DATA REC LENGTH: 1013 kbytes
LATITUDE: 34,19,55
LONGITUDE: 122,00,00

DATA BUFFER SIZE: 1036288
TIME/BUFFER: 11.776 secs
TIME TO DISK FULL: 1 hr, 5 min, 44 sec
PTS/CH/BUFFER: 47104

**** AFTER EXPERIMENT # 17 ****

BYTES WRITTEN: 347504768
BLOCKS WRITTEN: 339361

RECORDS WRITTEN: 335
START LBA: 832329
FILE AREA DE @: 832328
COMP AREA DE @: 1172491
BLOCKS REMAINING: 801

*** 1991 HEARD ISLAND DATA ACQUISITON SYSTEM ***
*** WITH OPTICAL DISK STORAGE ***

START OF DATA RECORDING.....
JDAY: 32 YEAR: 1991
GMT: 1:28:i.827
DISK #: 42736-1

PARAMETERS FROM FILE HEADER

PROJECT: HEARD ISLAND
EXPERIMENT TYPE: TELEMETRY
EXPERIMENT #: 19
TOTAL# CHANNELS: 12
SAMPLE RATE: 4000.000 Hz
SAMPLE PERIOD: 250 usec
DATA REC LENGTH: 769 kbytes
LATITUDE: 34,19,55
LONGITUDE: 122,00,00

DATA BUFFER SIZE: 786432
TIME/BUFFER: 8.192 secs
TIME TO DISK FULL: 3 hr, 28 min, 4 sec
PTS/CH/BUFFER: 32768

**** AFTER EXPERIMENT # 19 ****

BYTES WRITTEN: 388221056
BLOCKS WRITTEN: 379123
RECORDS WRITTEN: 493
START LBA: 9
FILE AREA DE @: 8
COMP AREA DE @: 1172497
BLOCKS REMAINING: 793365

*** 1991 HEARD ISLAND DATA ACQUISITON SYSTEM ***
*** WITH OPTICAL DISK STORAGE ***

START OF DATA RECORDING.....
JDAY: 32 YEAR: 1991
GMT: 4:25:24.695
DISK #: 42736-1

PARAMETERS FROM FILE HEADER

PROJECT: HEARD ISLAND
EXPERIMENT TYPE: TELEMETRY
EXPERIMENT #: 20
TOTAL# CHANNELS: 12
SAMPLE RATE: 4000.000 Hz
SAMPLE PERIOD: 250 usec
DATA REC LENGTH: 769 kbytes
LATITUDE: 34,19,55
LONGITUDE: 122,00,00

DATA BUFFER SIZE: 786432
TIME/BUFFER: 8.192 secs
TIME TO DISK FULL: 2 hr, 20 min, 45 sec
PTS/CH/BUFFER: 32768

**** AFTER EXPERIMENT # 20 ****

BYTES WRITTEN: 386640896
BLOCKS WRITTEN: 377579
RECORDS WRITTEN: 491
START LBA: 379133
FILE AREA DE @: 379132
COMP AREA DE @: 1172496
BLOCKS REMAINING: 415784

*** 1991 HEARD ISLAND DATA ACQUISITON SYSTEM ***

*** WITH OPTICAL DISK STORAGE ***

START OF DATA RECORDING.....
JDAY: 32 YEAR: 1991
GMT: 5:36:30.14
DISK #: 42736-1

PARAMETERS FROM FILE HEADER

PROJECT: HEARD ISLAND
EXPERIMENT TYPE: TELEMETRY
EXPERIMENT #: 21
TOTAL# CHANNELS: 12
SAMPLE RATE: 4000.000 Hz
SAMPLE PERIOD: 250 usec
DATA REC LENGTH: 769 kbytes
LATITUDE: 34,19,55
LONGITUDE: 122,00,00

DATA BUFFER SIZE: 786432
TIME/BUFFER: 8.192 secs
TIME TO DISK FULL: 1 hr, 13 min, 43 sec
PTS/CH/BUFFER: 32768

**** AFTER EXPERIMENT # 21 ****

BYTES WRITTEN: 377191424
BLOCKS WRITTEN: 368351
RECORDS WRITTEN: 479
START LBA: 756713
FILE AREA DE @: 756712
COMP AREA DE @: 1172495
BLOCKS REMAINING: 47431

*** 1991 HEARD ISLAND DATA ACQUISITON SYSTEM ***
*** WITH OPTICAL DISK STORAGE ***

START OF DATA RECORDING.....
JDAY: 32 YEAR: 1991
GMT: 6:51:4.721
DISK #: 42736-2

PARAMETERS FROM FILE HEADER

PROJECT: HEARD ISLAND
EXPERIMENT TYPE: TELEMETRY
EXPERIMENT #: 22
TOTAL# CHANNELS: 12
SAMPLE RATE: 4000.000 Hz
SAMPLE PERIOD: 250 usec
DATA REC LENGTH: 769 kbytes
LATITUDE: 34,19,55
LONGITUDE: 122,00,00

DATA BUFFER SIZE: 786432
TIME/BUFFER: 8.192 secs
TIME TO DISK FULL: 3 hr, 28 min, 4 sec
PTS/CH/BUFFER: 32768

**** AFTER EXPERIMENT # 22 ****

BYTES WRITTEN: 848877568
BLOCKS WRITTEN: 828982
RECORDS WRITTEN: 1078
START LBA: 2
FILE AREA DE @: 1
COMP AREA DE @: 1172498
BLOCKS REMAINING: 343514

*** 1991 HEARD ISLAND DATA ACQUISITON SYSTEM ***
*** WITH OPTICAL DISK STORAGE ***

START OF DATA RECORDING.....
JDAY: 32 YEAR: 1991
GMT: 9:26:23.219
DISK #: 43422-1

PARAMETERS FROM FILE HEADER

PROJECT: HEARD ISLAND
EXPERIMENT TYPE: TELEMETRY
EXPERIMENT #: 23
TOTAL# CHANNELS: 12
SAMPLE RATE: 4000.000 Hz
SAMPLE PERIOD: 250 usec
DATA REC LENGTH: 769 kbytes
LATITUDE: 34,21,22
LONGITUDE: 122,04,98

DATA BUFFER SIZE: 786432
TIME/BUFFER: 8.192 secs
TIME TO DISK FULL: 3 hr, 28 min, 4 sec
PTS/CH/BUFFER: 32768

**** AFTER EXPERIMENT # 23 ****

BYTES WRITTEN: 382708864

BLOCKS WRITTEN: 373740
RECORDS WRITTEN: 486
START LBA: 9
FILE AREA DE @: 8
COMP AREA DE @: 1172497
BLOCKS REMAINING: 798748

*** 1991 HEARD ISLAND DATA ACQUISITON SYSTEM ***
*** WITH OPTICAL DISK STORAGE ***

START OF DATA RECORDING.....
JDAY: 32 YEAR: 1991
GMT: 10:37:23.374
DISK #: 43422-1

PARAMETERS FROM FILE HEADER

PROJECT: HEARD ISLAND
EXPERIMENT TYPE: TELEMETRY
EXPERIMENT #: 24
TOTAL# CHANNELS: 12
SAMPLE RATE: 4000.000 Hz
SAMPLE PERIOD: 250 usec
DATA REC LENGTH: 769 kbytes
LATITUDE: 34,21,22
LONGITUDE: 122,04,98

DATA BUFFER SIZE: 786432
TIME/BUFFER: 8.192 secs
TIME TO DISK FULL: 2 hr, 21 min, 43 sec
PTS/CH/BUFFER: 32768

**** AFTER EXPERIMENT # 24 ****

BYTES WRITTEN: 400032896
BLOCKS WRITTEN: 390658
RECORDS WRITTEN: 508
START LBA: 373757
FILE AREA DE @: 373756
COMP AREA DE @: 1172495
BLOCKS REMAINING: 408080

*** 1991 HEARD ISLAND DATA ACQUISITON SYSTEM ***
*** WITH OPTICAL DISK STORAGE ***

START OF DATA RECORDING.....
JDAY: 32 YEAR: 1991
GMT: 11:50:7.264
DISK #: 43422-1

PARAMETERS FROM FILE HEADER

PROJECT: HEARD ISLAND
EXPERIMENT TYPE: TELEMETRY
EXPERIMENT #: 25
TOTAL# CHANNELS: 12
SAMPLE RATE: 4000.000 Hz
SAMPLE PERIOD: 250 usec
DATA REC LENGTH: 769 kbytes
LATITUDE: 34,21,22
LONGITUDE: 122,04,98

DATA BUFFER SIZE: 786432
TIME/BUFFER: 8.192 secs
TIME TO DISK FULL: 1 hr, 12 min, 21 sec
PTS/CH/BUFFER: 32768

**** AFTER EXPERIMENT # 25 ****

BYTES WRITTEN: 417356928
BLOCKS WRITTEN: 407576
RECORDS WRITTEN: 530
START LBA: 764423
FILE AREA DE @: 764422
COMP AREA DE @: 1172493
BLOCKS REMAINING: 494

*** 1991 HEARD ISLAND DATA ACQUISITON SYSTEM ***
*** WITH OPTICAL DISK STORAGE ***

START OF DATA RECORDING.....
JDAY: 32 YEAR: 1991
GMT: 16:24:21.505
DISK #: 43422-2

PARAMETERS FROM FILE HEADER

PROJECT: HEARD ISLAND
EXPERIMENT TYPE: TELEMETRY
EXPERIMENT #: 26
TOTAL# CHANNELS: 12
SAMPLE RATE: 4000.000 Hz
SAMPLE PERIOD: 250 usec
DATA REC LENGTH: 769 kbytes
LATITUDE: 34,21,22
LONGITUDE: 122,04,98

DATA BUFFER SIZE: 786432
TIME/BUFFER: 8.192 secs
TIME TO DISK FULL: 3 hr, 28 min, 4 sec
PTS/CH/BUFFER: 32768

**** AFTER EXPERIMENT # 26 ****

BYTES WRITTEN: 1174102144
BLOCKS WRITTEN: 1146585
RECORDS WRITTEN: 1491
START LBA: 9
FILE AREA DE @: 8
COMP AREA DE @: 1172497
BLOCKS REMAINING: 25903

*** 1991 HEARD ISLAND DATA ACQUISITON SYSTEM ***
*** WITH OPTICAL DISK STORAGE ***

START OF DATA RECORDING.....
JDAY: 32 YEAR: 1991
GMT: 19:55:17.719
DISK #: 42737-1

PARAMETERS FROM FILE HEADER

PROJECT: HEARD ISLAND
EXPERIMENT TYPE: TELEMETRY
EXPERIMENT #: 27
TOTAL# CHANNELS: 12
SAMPLE RATE: 4000.000 Hz
SAMPLE PERIOD: 250 usec
DATA REC LENGTH: 769 kbytes
LATITUDE: 34,23,14
LONGITUDE: 122,08,48

DATA BUFFER SIZE: 786432
TIME/BUFFER: 8.192 secs
TIME TO DISK FULL: 3 hr, 28 min, 4 sec
PTS/CH/BUFFER: 32768

**** AFTER EXPERIMENT # 27 ****

BYTES WRITTEN: 125998208
BLOCKS WRITTEN: 123046
RECORDS WRITTEN: 160
START LBA: 9
FILE AREA DE @: 8
COMP AREA DE @: 1172497
BLOCKS REMAINING: 1049442

*** 1991 HEARD ISLAND DATA ACQUISITON SYSTEM ***
*** WITH OPTICAL DISK STORAGE ***

START OF DATA RECORDING.....
JDAY: 32 YEAR: 1991
GMT: 22:51:42.966
DISK #: 42737-1

PARAMETERS FROM FILE HEADER

PROJECT: HEARD ISLAND
EXPERIMENT TYPE: TELEMETRY
EXPERIMENT #: 28
TOTAL# CHANNELS: 12
SAMPLE RATE: 4000.000 Hz
SAMPLE PERIOD: 250 usec
DATA REC LENGTH: 769 kbytes
LATITUDE: 34,23,14
LONGITUDE: 122,08,48

DATA BUFFER SIZE: 786432
TIME/BUFFER: 8.192 secs
TIME TO DISK FULL: 3 hr, 6 min, 13 sec
PTS/CH/BUFFER: 32768

**** AFTER EXPERIMENT # 28 ****

BYTES WRITTEN: 1074095232
BLOCKS WRITTEN: 1048922
RECORDS WRITTEN: 1364
START LBA: 123063

FILE AREA DE @: 123062
COMP AREA DE @: 1172495
BLOCKS REMAINING: 510

*** 1991 HEARD ISLAND DATA ACQUISITON SYSTEM ***
*** WITH OPTICAL DISK STORAGE ***

START OF DATA RECORDING.....
JDAY: 33 YEAR: 1991
GMT: 2:6:6.305
DISK #: 42737-2

PARAMETERS FROM FILE HEADER

PROJECT: HEARD ISLAND
EXPERIMENT TYPE: TELEMETRY
EXPERIMENT #: 29
TOTAL# CHANNELS: 12
SAMPLE RATE: 4000.000 Hz
SAMPLE PERIOD: 250 usec
DATA REC LENGTH: 769 kbytes
LATITUDE: 34,23,14
LONGITUDE: 122,08,48

DATA BUFFER SIZE: 786432
TIME/BUFFER: 8.192 secs

TIME TO DISK FULL: 3 hr, 28 min, 4 sec
PTS/CH/BUFFER: 32768

**** AFTER EXPERIMENT # 29 ****

BYTES WRITTEN: 1007161472
BLOCKS WRITTEN: 983557
RECORDS WRITTEN: 1279
START LBA: 9
FILE AREA DE @: 8
COMP AREA DE @: 1172497
BLOCKS REMAINING: 188931

13.0 References

[1] Elgar Corporation: 1751SL Instruction manual, Elgar Corporatuion, 9250 Brown Deer Road, San Diego, CA 92121 (619) 450-0085

[2] Lockheed/Sanders: Model 30 Flextensional Transducer calibration results, Lockheed/ Sanders ASW, 955 Perimeter Road, MAN - 06 P.O. Box 8500, Manchester NH 0308-8500

[3] K. Metzger, Jr.: Signal Processing Equipment and Techniques for Use in Measuring Ocean Acoustic Multipath Structures, Cooley Electronics Lab. Univ. Michigan, Ann Arbor, Tech. Report No 231, Dec 1983

DOCUMENT LIBRARY

March 11, 1991

Distribution List for Technical Report Exchange

Attn: Stella Sanchez-Wade
Documents Section
Scripps Institution of Oceanography
Library, Mail Code C-075C
La Jolla, CA 92093

Hancock Library of Biology &
Oceanography
Alan Hancock Laboratory
University of Southern California
University Park
Los Angeles, CA 90089-0371

Gifts & Exchanges
Library
Bedford Institute of Oceanography
P.O. Box 1006
Dartmouth, NS, B2Y 4A2, CANADA

Office of the International
Ice Patrol
c/o Coast Guard R & D Center
Avery Point
Groton, CT 06340

NOAA/EDIS Miami Library Center
4301 Rickenbacker Causeway
Miami, FL 33149

Library
Skidaway Institute of Oceanography
P.O. Box 13687
Savannah, GA 31416

Institute of Geophysics
University of Hawaii
Library Room 252
2525 Correa Road
Honolulu, HI 96822

Marine Resources Information Center
Building E38-320
MIT
Cambridge, MA 02139

Library
Lamont-Doherty Geological
Observatory
Columbia University
Palisades, NY 10964

Library
Serials Department
Oregon State University
Corvallis, OR 97331

Pell Marine Science Library
University of Rhode Island
Narragansett Bay Campus
Narragansett, RI 02882

Working Collection
Texas A&M University
Dept. of Oceanography
College Station, TX 77843

Library
Virginia Institute of Marine Science
Gloucester Point, VA 23062

Fisheries-Oceanography Library
151 Oceanography Teaching Bldg.
University of Washington
Seattle, WA 98195

Library
R.S.M.A.S.
University of Miami
4600 Rickenbacker Causeway
Miami, FL 33149

Maury Oceanographic Library
Naval Oceanographic Office
Stennis Space Center
NSTL, MS 39522-5001

Marine Sciences Collection
Mayaguez Campus Library
University of Puerto Rico
Mayaguez, Puerto Rico 00708

Library
Institute of Oceanographic Sciences
Deacon Laboratory
Wormley, Godalming
Surrey GU8 5UB
UNITED KINGDOM

The Librarian
CSIRO Marine Laboratories
G.P.O. Box 1538
Hobart, Tasmania
AUSTRALIA 7001

Library
Proudman Oceanographic Laboratory
Bidston Observatory
Birkenhead
Merseyside L43 7 RA
UNITED KINGDOM

50272-101

REPORT DOCUMENTATION PAGE	1. REPORT NO. WHOI-91-38	2.	3. Recipient's Accession No.

4. Title and Subtitle
Multiple Convergence Zone Acoustic Telemetry Feasibility Test Report

5. Report Date
November, 1991

6.

7. Author(s) Josko A. Catipovic, Keith von Der Heydt, John Stevens Merriam of Woods Hole Oceanographic Institution and Geir Helge Sandsmark of University of Trondheim (Norway)

8. Performing Organization Rept. No.
WHOI-91-38

9. Performing Organization Name and Address
Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543

10. Project/Task/Work Unit No.

11. Contract(C) or Grant(G) No.
(C) N00014-90-C-0098
(G)

12. Sponsoring Organization Name and Address
Office of Naval Technology

13. Type of Report & Period Covered
Technical Report

14.

15. Supplementary Notes
This report should be cited as: Woods Hole Oceanog. Inst. Tech. Rept., WHOI-91-38.

16. Abstract (Limit: 200 words)

This report describes a multiple CZ acoustic telemetry experiment conducted off the coast of California 1/28/90-2/2/90. The goal was to design a maximally robust high speed underwater modem suitable for data telemetry for submerged platforms and moorings. Six modulation methods were used to transmit data at rates from 1 to 1000 baud corresponding to bit rates up to 3kbit/sec. The modulation formats were:

1. Multiple Frequency Shift Keying (MFSK) and Binary Expurgated Modulation (BEX-PERM)
2. Duobinary Frequency Shift Keying
3. Quadrature Phase-Shift Keying (QPSK)
4. 8 Quadrature Amplitude Modulation (8QAM)
5. Continuous Phase Modulation (CPM) 2 DPM4 aand 2CPFSK4
6. Trellis coded 8PSK

In addition, a large number of channel probe sequences was transmitted in order to estimate channel multipath, fluctuation dynamics and spatial diversity characteristics relevant to acoustic data telemetry.

The data was transmitted from a 1 kHz source suspended from the R/V *McGaw*, and received on a multichannel vertical array tended by the RV *Point Sur*. The multichannel data was digitally recorded using floating-point digitizers and stored on optical [illegible]k for further processing. Approximate transmission ranges were 70, 140, 200 and 250km. Approximately 8 hrs of transmission were recorded at each data range.

17. Document Analysis a. Descriptors
acoustic communication
acoustic fluxuation
underwater modem

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
Approved for public release; distribution unlimited.	UNCLASSIFIED	128
	20. Security Class (This Page)	22. Price

(See ANSI-Z39.18) *See Instructions on Reverse* OPTIONAL FORM 272 (4-77) (Formerly NTIS-35) Department of Commerce