REVISION HISTORY			
REV	DESCRIPTION (U)	DATE	APPROVED
-	Initial Release	12/12/05	L. Freitag

FHFSK
Coding and Modulation Specification

Part of the WHOI Multi-User Frequency-Hopping Underwater Acoustic Communication Protocol

Woods Hole Oceanographic Institution
266 Woods Hole Road
Woods Hole, MA 02540

401003-SPEC FHFSK Coding and Modulation Specification

SIZE	CAGE CODE	DWG NO
A	88846	401003-SPEC

SCALE		SHEET	
NONE		1 of	13

The work described in this document was performed by a number of people over the past several years. Primary contributors include Mark Johnson, James Preisig, Ryuji Suzuki, Sandipa Singh, Matt Grund, Milica Stojanovic and Lee Freitag. This document was written and edited by Lee Freitag and Sandipa Singh.

All questions, comments and corrections should be sent to:

Lee Freitag
Department of Applied Ocean Physics and Engineering
Woods Hole Oceanographic Institution
Mail Stop 18
Woods Hole, MA 02543
lfreitag@whoi.edu

The most recent version of this document is archived in Portable Document Form (PDF) at:
acomms.whoi.edu

This document is based on the original publication, *Multi-User Frequency-Hopping Underwater Acoustic Communication Protocol, Ver. 1.02 May 25, 2000,* which is now superseded by this document and WHOI document 401002-SPEC.

Release Notes for the Original Version of the FHFSK SPEC

Release Notes for Version 1.0

The initial release is intended to provide a complete description of the following portions of the specification:

- The segmenter.
- The data frame structure.
- Error correction coding.
- Modulation.
- Frequency-hop patterns.

At this time the supervisory frame description is for informational purposes only and is not intended for Implementation until after July, 2000. The supervisory frame descriptions constitute a Request for Comment from interested developers.

Interim testing is expected to be conducted as follows with respect to bands and user identification:

- A modem pair is programmed to use one pre-selected frequency band.
- A modem pair is programmed to use one pre-selected hopping pattern.

Interim testing among modems from different organizations is expected to be conducted as follows, with respect to bidirectional communication in the near-term:

- One modem is configured as receiver. One or more other modems are programmed to transmit datagrams at low duty-cycle.
- One modem is programmed to initiate transactions (unit 0). All other modems are configured as slaves and transmit only when they are addressed by unit 0.

Other protocols will be established and documented after use of the physical layer is validated among different organizations.

Release Notes for Version 1.01

Changes in this version include a reduced width for Band C.

- The upper edge of Band C is lowered.
- The hopping patterns for Band C are changed to 7 and 13.
- The number of sync bits for Band C are changed to 21 and 26.

Release Notes for Version 1.02

Changes in this version include a modification to the hopping pattern used for acquisition. The pattern for the synchronization sequence for user i is now changed to the pattern $i + 3$ for the 7 hop sequence and $i + 6$ for the 12 hop sequence. This imposes a practical limit of 3 users for the 7 hop case and 6 users for the 13 hop cases (per frequency band).

Table of Contents

Tables

Part I

Physical Layer Description

1 Introduction

This document describes a standard for low-rate, multi-user communication in the underwater acoustic environment.

1.1 Scope

The principal characteristics of the standard include:

1. Error-correction coding utilizing a rate 1/2 convolutional code.
2. Symbol rates of:
   (a) 160 symbols per second
   (b) 80 symbols per second
3. Three frequency regimes:
   (a) 7.6 to 12 kHz
   (b) 12.5 to 17 kHz
   (c) 23 to 30 kHz
4. Low-coincidence frequency hopping patterns which provide code-division multiple-access and channel clearing time in multipath environments.

The format of the data for a standard packet is described in WHOI document 401002-SPEC. The format of a mini packet is described in 401010-SPEC.

Section 2 describes error correction methods in detail. Frequency hop FSK modulation is described in Section 3.

In addition, a Matlab toolbox has been written to demonstrate and document the physical layer of the system. The Matlab modules include the segmenter, the error-correction coding and frequency-hop modulation layers. The toolbox is summarized in Appendix A and it outlines the capabilities of the ECC, modulation and demodulation tools.

2 Encoding

Frames can be of two types: supervisory and data. A data frame is used purely for data transfer, while a supervisory frame is used in all other cases, e.g. to request a connection, to set up the details of the data transfer and to set the link parameters. The supervisory frames are not currently implemented.

2.1 Encoder

After the data is formed into blocks with the header and CRC, each frame is whitened using a pseudorandom noise (PN) sequence, convolutionally coded, then bit-interleaved.

2.1.1 Data Whitening

The data is whitened using an m-sequence, which is generated using a shift register of length 11 and a generator polynomial $g(x) = 1 + x^2 + x^{11}$. The shift register is initialized with a one and 10 zeros. The whitening sequence generated in this fashion is logical-OR'd with the data prior to encoding with the convolutional code.

2.1.2 Convolutional Code

The whitened frame is convolutionally encoded using the same code as is employed in the IS-95 CDMA standard, a rate-1/2 encoder with constraint length 9 [2]. The generator polynomials are:

$$g_1(x)=1+x+x^2+x^3+x^5+x^7+x^8$$
$$g_2(x)=1+x^2+x^3+x^4+x^8$$

Prior to encoding 8 zeros are appended to the data as a tail for the convolutional encoder. These bits are discarded at the receiver.

2.1.3 Interleaver

The encoded frame is then block-interleaved using an interleaver depth of 40 symbols. This separates consecutive errors by greater than four constraint lengths.

The encoded data is padded to an even number of 40 symbol blocks, then formed into a matrix with 40 columns and *n* rows. The interleaver operates by then reading out the data columnwise to achieve the 40 symbol spacing.

3 Frequency-Hop Modulation

The modulation method is binary (one-of-two) frequency-shift keying (FSK) with frequency hopping.

3.1 Symbol Rates and Frequency Separation

The symbol rates are chosen to be integer multiples of each other and allow for an even number of A/D samples using common clock rates. Symbol rates of 80 and 160 sps. are used for all three-frequency bands. The symbol durations corresponding to the two rates are 12.5 and 6.25 milliseconds respectively. At the 80 symbol per second rate the separation between frequencies is fixed at 160 Hz. At the 160 symbol per second rate the separation is 320 Hz.

3.2 Signaling Bandwidth

The bandwidth occupied by the signals is defined by the number of hops in each hopping table and the frequency bin separation. The number of hops used is dependent on the band 7 or 13 for the lower two frequency bands, and 11 or 23 for the upper frequency band. The upper band edges are calculated based on this and listed in Table 1.

3.3 Shaping

This standard does not define a time-domain weighting function for the transmitted signal.

3.4 Frequency Bands

There are three frequency bands defined in this specification. The bandwidth is approximately 4 kHz for bands A and B and 7 kHz for band C. Band A is intended for compatibility with underwater telephone transducers (i.e. the UQC). The bands are summarized in Table 1 where F_l denotes the carrier of the lowest frequency bin and F_h is the highest carrier frequency. The total bandwidth is slightly larger than $F_h - F_l$ and is also different for each of the two symbol rates.

Table 1. Frequency Bands and Summary Information

Band	Rate sps	ΔF Hz	Hops	Synch Bits	F_l Hz	F_h Hz	T_{clear} sec
A	160	320	7	21	7680	11840	0.0375
A	80	160	13	26	7680	11680	0.150
B	160	320	7	21	12800	16960	0.0375
B	80	160	13	26	12800	16800	0.150
C	160	320	7	21	23040	27200	0.0375
C	80	160	13	26	23040	27040	0.150

The channel clearing time is a function of the number of hops and the symbol duration. The clearing time is defined as the period from the end of a symbol in a particular bin to the start of the next symbol in that same bin. The values for T_{clear} for each of the bands and rates is also listed in the table.

Table 2. Transducer and manufacturer list for each of the three bands.

Band	Model Number	Type	Manufacturer
A	SB31CT	Omni	EDO
	SP23LT	Conical	EDO
	249-10 Series	Toroidal	EDO
	ITC-3167	Hemispherical	ITC
	ITC-3013	Hemispherical	ITC
	ITC-3148	Hemispherical	ITC
	ITC-2044	Toroidal ±35° @9.5 kHz	ITC
	AT-12*xx*	Omni	Benthos
	TC-1010	Omni	RESON
	TC-1032	Conical, ±30° @10 kHz	RESON
	TC-1035	Toroidal ±35° @9.5 kHz	RESON
	TC-1037	Conical, ±30° @10 kHz	RESON
B	AT-18A1	Conical, ±30° @18 kHz	Benthos
	AT-18AT	Omni	Benthos
	AT-18DT 3 ring	Toroidal ±20° @18 kHz	Benthos
	249-17 Series	Toroidal	EDO
	6969-3500 2nd resonance	Omni	EDO
	ITC-3001	Conical, ±14° @17.5 kHz	ITC
C	ITC-6137	Toroidal ±14° @24 kHz	ITC
	ITC-2089	Toroidal ±38° @24 kHz	ITC
	AT-27BT	Omni	Benthos
	BT-2	Toroidal ±20° @25 kHz	BTECH

Commercial transducers identified as fitting these bands are listed in Table 2 as a convenience. In no way does this listing recommend or suggest any suitability of these transducers for any application but is instead a reference for the implementer. Some of the transducers listed here require electrical tuning or pre-emphasis to optimize the bandwidth.

3.5 Hopping Sequences

Two different hopping sequences are defined. Their selection depends upon the operating band as described in Table 1. There are *q* - 1 sequences of length *q* where *q* is a prime number. These sequences provide a minimum number of coincidences (exactly one with any other sequence) and may be generated as follows:

$$V_m = (v_{m,1},\ v_{m,2},\ \ldots, v_{m,q})$$

$$v_{1,k} = k - 1$$

$$v_{m,k} = v_{1,p+mr}$$

where $k = mp + r$ under **N**, $1 \leq m \leq q - 1$, $k \leq q$, and operates as defined under GF(*q*).

The $q = 7$ and q = 13 cases are tabulated in this document ($m = 0$ results in zero vector as a trivial case, which is excluded from the tables).

The tables are used as follows:

- User 1 employs the base pattern and sequentially steps through the bin pairs where bin pair one corresponds to the two lowest frequency bins defined in Table 1.
- User 2 employs pattern 2 and so on.

Table 3. Sequences for *q* = 7

m							
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

Table 4. Sequences for *q* = 13

m													
1	0	1	2	3	4	5	6	7	8	9	10	11	12
2	0	2	4	6	8	10	12	1	3	5	7	9	11
3	0	3	6	9	12	2	5	8	11	1	4	7	10
4	0	4	8	12	3	7	11	2	6	10	1	5	9
5	0	5	10	2	7	12	4	9	1	6	11	3	8
6	0	6	12	5	11	4	10	3	9	2	8	1	7
7	0	7	1	8	2	9	3	10	4	11	5	12	6
8	0	8	3	11	6	1	9	4	12	7	2	10	5
9	0	9	5	1	10	6	2	11	7	3	12	8	4
10	0	10	7	4	1	11	8	5	2	12	9	6	3
11	0	11	9	7	5	3	1	12	10	8	6	4	2
12	0	12	11	10	9	8	7	6	5	4	3	2	1

3.6 Packet Timing Acquisition

Timing acquisition of a data packet is done using a prescribed sequence of data bits modulated according to the hopping pattern for a particular user. The number of bits used for synchronization is shown in Table 1. In all cases an integer number of passes through the hopping table are used. The actual data to be modulated is simply a section of a pseudo-random sequence. The sequence used for generation of the data used to modulate the synchronization pattern is based on the shift-register sequence of length 10 which uses the polynomial $g(x) = 1+x^2+x^{10}$. The initial 10 outputs from the shift register are discarded and starting at the 11th point the data is segmented for each user's pattern.

Thus the first user in the 7 hop system uses 21 synchronization bits, which are the 11th through the 32nd bit from the shift register. User 2 uses the next 21 bits, and so on. The same sequence is employed for all of the different hopping patterns. The data bits for each hopping sequence and each user are tabulated in several include files for use with C programs, and in a Matlab file as well.

The hopping pattern for synchronization is different than that used for data. This reduces the correlation between the data and synchronization portions of a packet. The hop sequence for the data for user *i* is row *i* from the hopping table. The hop sequence for the synchronization bits is row *i* + 3 for the 7 hop case and row *i* + 6 for the 13 hop case. Thus a maximum of 3 users are supported per frequency band for the 7 hop case (160 sps) and a maximum of 6 users for the 13 hop case (80 sps).

3.7 References

[1] Wicker, Stephen B., Error Control Systems for Digital Communication and Storage, Prentice-Hall, 1995.

[2] Peterson, Roger L., R.E. Ziemer, and D. E. Borth, Introduction to Spread Spectrum Communications, Prentice-Hall, 1995.

A Matlab Frequency Hop Toolbox

The toolbox includes the functions required to break up a block of data into frames and then encode those frames for transmission. In addition, the toolbox provides all the functions necessary to take demodulated data from the frequency-hop receiver and decode it back to frames.

The frequency hop modulation and demodulation tools include the facilities to add synchronization bits to data frames, modulate the data to passband, then demodulate using a matched filter bank. The tools are provided as examples only and it is not in tended that actual implementations copy the methodology used here, which is provided only for demonstration. Demonstration of a method for performing timing acquisition is also provided. This function is called `fhacq.m`

A.1 Using the Toolbox

The toolbox is supplied as a reference for the standard presented in this document. The operations described in the previous sections are provided as Matlab functions in order to allow validation of actual implementations. These tools are thus used to both document the standard and to provide a means for checking the compliance of a given system with the standard.

The functions are contained in one directory for easy use. However, it must be added to the users' matlab path in order to execute the functions from another directory.

A.2 MEX files

Several functions are provided only as C source code, not in m-file format. These functions include the Viterbi decoder, the metric table generator and the CRC function. While these functions have been compiled for Linux systems and are provided, users of other operating systems will need to compile these functions for their system. Type `help mex` at the Matlab prompt for more information. Check the README file in src/vit within the distribution for additional information. After the mex files have been compiled for your system place them in the distribution directory.

The soft-decision Viterbi decoder is from a publicly available web site maintained by Phil Karn. This software may be used under the terms of the GNU Public License.

A.3 Demonstrations

The complete structure of the standard may be observed by studying the file `fskdemo.m`. To run a demonstration of one of the 6 possible band and rate combinations a set of parameter files have been created. Simply invoke the name of one of the parameter files, for example `fhfsk_a_160` and then type `fskdemo`. The scripts leave all of their output in the workspace to be examined by the user.

A.4 Toolbox Contents

The contents of the toolbox include the functions listed below:

Multi-User Frequency Hopping Communication Toolbox
Version 1.0 March 27, 2000

Segmenter
- mkframe - makes frames, given data
- rdframe - unpacks data/header info, given frames
- rdheader - unpacks a header word to the header info
- paddata - pads data
- crc - 16 bit CRC generator (mex)

Error Correction
- whitenframes - whiten a frame using PN sequence
- encodeframes - convolutionally encode and interleave frames
- decodeframes - de-interleave and viterbi decode frames
- conv_encode - convolutional encoder
- conv_decode - Viterbi decoder function
- vit29_decode - (2,1,9) Viterbi decoder (mex)
- mettab - metric table generator (mex)
- interleave - fixed-length block interleaver
- mlgen - Maximal-Length sequence generator

Frequency-Hopping FSK
- fhmod - modulate data to passband
- fhdemod - demodulate and apply filter bank
- fsk2prob - convert binary FSK to probability metric
- isnr - calculate SNR of FSK filter bank output

Utilities
- bit2word - convert a stream of bits into words
- word2bit - convert a word into a stream of bits
- oct2bin - Convert an octal number into binary
- bin2sym - convert packed binary data to symbols
- sym2bin - convert symbols to binary data

Demonstrations
- fhfsk_a_160 - generate and demodulate Band A, 160 sps
- fhfsk_a_80 - generate and demodulate Band A, 80 sps
- fhfsk_b_160 - generate and demodulate Band B, 160 sps
- fhfsk_b_80 - generate and demodulate Band B, 80 sps
- fhfsk_c_160 - generate and demodulate Band C, 160 sps
- fhfsk_c_80 - generate and demodulate Band C, 80 sps
- fskdemo - utility script for above demos

Mat Files
- sync - matrices of synchronization bits (pn7, pn11, pn13, pn23)
- table7c - 7 element hop matrix
- table11c - 11 element hop matrix
- table13c - 13 element hop matrix
- table23c - 23 element hop matrix

Contributions by S. Singh, M. Grund, L. Freitag, M. Johnson, M. Stojanovic, R. Suzuki and D. Kilfoyle.

B C-Code Include Files

The prime-number based hopping tables are in the file `include/tab.h`. The tables are in the form of static variables. The shortest table is shown below as an example

```
unsigned char **tab7 = {
{ 0, 1, 2, 3, 4, 5, 6},
{ 0, 2, 4, 6, 1, 3, 5},
{ 0, 3, 6, 2, 5, 1, 4},
{ 0, 4, 1, 5, 2, 6, 3},
{ 0, 5, 3, 1, 6, 4, 2},
{ 0, 6, 5, 4, 3, 2, 1},
};
```

The synchronization data bits are also available as a set of 5 files. Four of the files contain the sync data bits encoded as bytes, the other file (`sync.h`) contains the same data, but packed into words for a more compact representation. The file `sync11.h` is shown below. All of the files are in the same directory.

```
unsigned char **sync11 = {
{0,1,1,1,1,0,0,1,0,0,0,0,1,0,1,1,1,1,1,1,0,1},
{1,0,1,1,1,0,0,0,0,1,1,0,0,1,1,0,0,1,1,1,1,0},
{0,1,1,1,1,0,1,0,1,0,0,1,1,1,1,1,1,1,0,0,0,1},
{0,0,1,1,0,1,1,0,1,1,1,0,0,1,1,0,1,1,0,0,1,0},
{1,0,0,1,1,1,1,1,1,1,0,1,1,0,1,0,0,1,1,1,1,1},
{0,1,0,0,1,1,1,1,0,1,0,1,1,0,0,0,0,0,0,0,1,1},
{0,1,1,0,1,0,1,1,1,0,0,0,1,1,1,1,1,0,0,1,0,0},
{1,0,1,1,1,0,1,0,1,1,1,1,0,0,1,0,1,0,1,1,0,1},
{0,1,0,0,1,0,1,0,0,1,1,1,0,0,0,1,1,0,0,1,0,1},
{0,0,0,0,1,0,1,0,0,1,1,1,0,0,1,1,0,0,1,1,0,0},
{1,1,0,0,0,0,1,1,1,1,1,1,1,0,0,1,0,1,0,0,0,0},
}
```

SIZE	CAGE CODE	DWG NO	REV
A	**88846**	401003-SPEC	-
SCALE NONE			SHEET 13