SSC22-XII-06

A Complete Receiver for a Multi-h Continuous Phase Frequency Shift Keyed Signal in the Presence of Doppler

Brendan Hill, Nazia Mozaffar, Salwan Damman
Naval Information Warfare Center (NIWC) Pacific
53560 Hull Street, San Diego CA 92152; 619-553-3255
brendan.m.hill.civ@us.navy.mil

ABSTRACT

In the absence of signal imperfections other than the presence of additive white Gaussian (AWGN) noise, Continuous Phase Frequency Shift Keyed (CPFSK) signals with a changing modulation index (referred to in the literature as "multi-h") will have only a few possible starting and end phases at each symbol interval. This allows them to be represented in a trellis diagram where they inherently possess coding gain that can be realized with a Viterbi decoder. However, with a low earth orbit altitude small satellite exhibiting a large time-varying Doppler, the Viterbi decoder fails to maintain this coding gain advantage if the receiver does not also track timing, phase, and frequency offsets. An effective receiver implementation calculates timing and phase offsets for every possible received symbol and stores them in their own structure of metrics indexed by symbol time and phase state. In parallel with this process, a non data-aided band-edge frequency locked loop continuously tracks and corrects time-varying Doppler. This paper presents a design for a CPFSK receiver and the insights into how its frequency, phase, and timing loops function. It also explores the control loop design choices that affect the overall system performance.

INTRODUCTION

Suppose one wants to communicate from a ground station with a low earth orbit small satellite in a 500-mile altitude circular orbit. Suppose further that the satellite's overflight path will traverse the zenith of the ground station and that the satellite moves due west of the ground observer at its zenith point. If the carrier frequency is 2.45 GHz, the Doppler shift starts off at 57 kHz and eventually changes to -57 kHz (see Figure 1). Halfway through the overflight time, as the satellite reaches zenith, the rate of change of the Doppler frequency offset is -550 Hz/sec (see Figure 2). The maximum Doppler offset is observed as the satellite ascends from or descends to the horizon, whereas the maximum change of Doppler occurs as it reaches zenith.

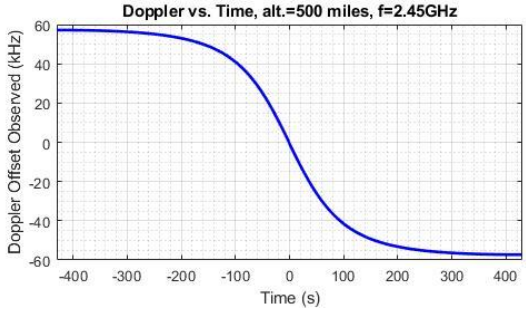

Figure 1: Doppler vs. Time Seen by Ground Observer for Small Satellite Orbit Scenario (t=0 at Zenith)

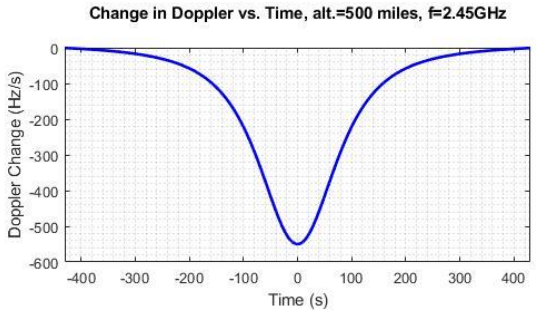

Figure 2: Rate of Doppler Change vs. Time Seen by Ground Observer for Small Satellite Orbit Scenario (t=0 at Zenith)

Continuous Phase Frequency Shift Keyed Signal Structure

Continuous Phase Frequency Shift Keyed signals possess a constant envelope structure because the frequency changes are achieved while maintaining the continuity of phase without abrupt changes in amplitude. This property allows the signal to drive a power amplifier in its nonlinear operating region beyond saturation, an efficient operating point, without exhibiting amplitude modulation to phase modulation (AM-PM) distortion.[1] Non-constant envelope signals require reducing the input power level to the amplifier in order to prevent clipping and spectral regrowth. This

reduces the efficiency of the power amplifier.[2] Furthermore, it is easier to control the occupied bandwidth of a constant envelope modulation scheme,[1] which makes it well suited for satellite communication applications.

For this paper, the authors selected a quaternary (4-ary) Continuous Phase Frequency Shift Keyed signal with a modulation index alternating between 2/8 and 3/8 and a modulation rate of 10 kHz. The changing modulation index between two values less than unity keeps the occupied bandwidth low, and presents a phase vs. symbol time trellis with improved coding gain.[3-4] This occurs because the trellis would repeat after fewer symbols without the changing modulation index. A longer trellis is analogous to a longer constraint length for a convolutional code.[5]

The receiver samples the signal at 20 samples per symbol (sps), for a sample rate of 200 ksps. This rate was chosen to make sure the half-sample rate exceeds the maximum expected Doppler frequency.

In the sections to follow, the major components of a classic CPFSK receiver are discussed with an emphasis on design choices that yield improved performance. The discussions on matched filters, the Viterbi decoder, the timing correction loop, and phase correction loop are considered in the absence of Doppler. Finally, the non data-aided band-edge FLL is introduced and its performance demonstrated, culminating in the presentation of a unified block diagram of the complete receiver.

MATCHED FILTERS

Each symbol can have one of four possible values. Matched filters, each synthesized with a complex conjugate of the reversed time series of each symbol, are used to maximize the signal to noise ratio at the end of the symbol period to select the maximum likelihood symbol. If the modulation indices are integers, then the information tones associated with each symbol would be mutually orthogonal. However, the system described in this paper uses alternating modulation indices, both of which are fractions. Therefore, the information tones are not mutually orthogonal. Each arriving symbol will produce a correlation peak with each matched filter. The correct matched filter is the one with the highest peak. If the absolute value of the matched filter is chosen as the basis for the symbol decision, it would ostensibly prevent incorrect decisions when the initial phase is nonzero. However, observe Figure 3, where it is clear that the cross-correlation peak between the "-3" symbol and the "-1" symbol is very close to the autocorrelation of the "-3" symbol. In the presence of noise, this is a likely source of error. The undesired peaks can be decreased if the real part of the cross correlation is used, as seen in Figure 4. To take advantage of this, one must maintain phase lock on the incoming signal, as any phase offset diminishes correct matched filter peak.

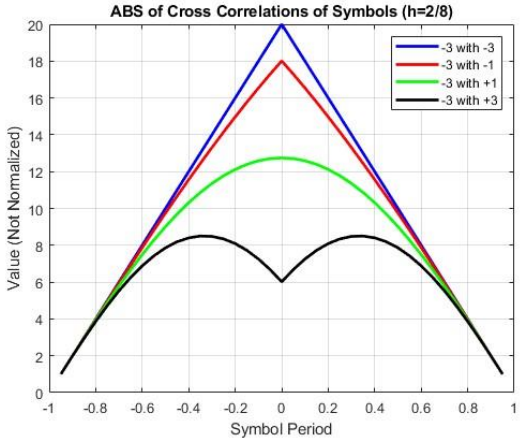

Figure 3: Absolute value of Matched Filter Outputs for h=2/8, 20 sps, and Incoming Symbol = "-3"

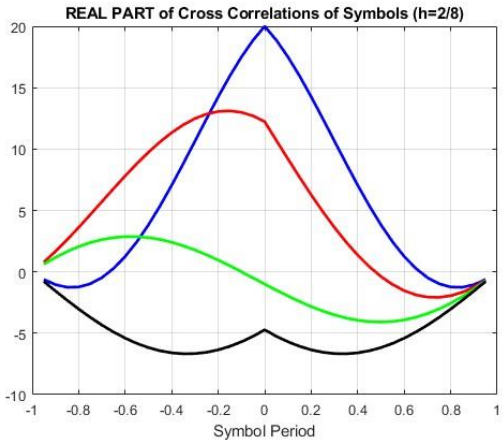

Figure 4: Real part of Matched Filter Outputs for h=2/8, sps = 20, and Incoming Symbol = "-3"

Achieving the maximum possible output from the matched filter necessitates the inclusion of the initial phase at the start of the symbol period. Since the modulation index is a fraction, most symbol transitions do not end at zero phase. This is compounded by the fact that the modulation index alternates between two values. The phase of the matched filter output must be rotated by this initial phase before taking the real part. In fact, the Viterbi decoder performs this de-rotation operation for every possible starting phase and every possible symbol as it builds up a table of metrics that it uses to find the maximum likelihood path. This will become clearer in the Viterbi decoder section.

VITERBI DECODER

In the phase trellis diagram of our 4-ary CPFSK multi-h signal (shown in Figure 5), different colors represent different arriving symbols. Blue paths signify the arrival of a "-3" symbol. Red represents "-1," green represents "+1," and black represents "+3." The trellis shows an assumed initial phase of zero radians. Starting with the second symbol period, there are eight possible initial phases. Starting with the third symbol period, each phase has four possible entry paths. The pattern repeats after four symbols.

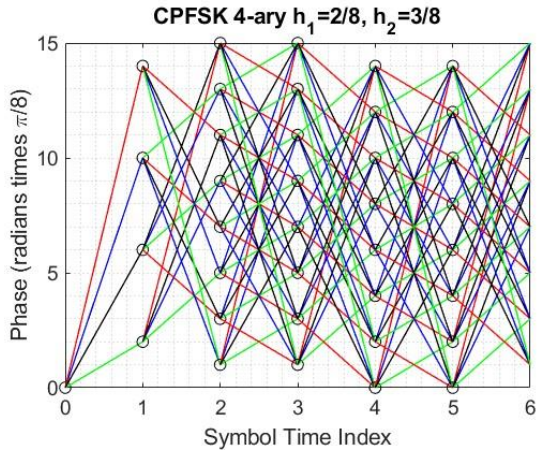

Figure 5: Trellis of 4-ary CPFSK, h_1=2/8, h_2=3/8 Starting at Zero Phase

A Viterbi decoder efficiently tracks the maximum likelihood path through a trellis by keeping track of which path through each state is the most likely. After a specified number of symbols arrives (a traceback length), the maximum likelihood symbol is chosen and its backward path through the trellis is traversed.

Summary of Viterbi Algorithm

The standard Viterbi algorithm consists of a few essential ingredients listed below.[5]

1) A path metric is computed for every possible path entering each state. The metric is the cumulative value of the real part of the correlations encountered along the path. Even though there can be multiple possible paths entering each state, only the path with the best metric is remembered.

2) Once the traceback number of symbol periods has elapsed, the algorithm starts at the end state with the highest metric at the end and walks backwards through the trellis following the one surviving path through each state. The corresponding symbols causing these transitions are the received symbols.

3) With each newly arriving symbol, the algorithm computes the maximum trellis from the current symbol back to the traceback length. It is possible with the arrival of a new symbol that the previous symbol time's maximum likelihood state may be eliminated because it does not have a path to the newest symbol time's maximum likelihood state.

4) The only "final" state (and symbol) decisions are the ones farther away from the present time than the traceback length. The traceback length thus represents a time delay between final decisions and current information.

Application of Viterbi Algorithm to Multi-h 4-ary CPFSK Signal

For a CPFSK multi-h receiver, the state is the starting phase at integer multiples of the symbol time. The signal is assumed to start at zero phase (as shown in Figure 5). This is achieved by choosing a preamble that results in a zero-phase state at the end of the preamble. Based on the alternating modulation indices chosen, the number of possible phase states are computed. The modulation index is the ratio of the information tone separation to the symbol rate, but it can also be thought of as the number of half-cycles of phase that a symbol traverses, as suggested by equation (1).

$$\Delta\varphi = \pi h\alpha, \quad \alpha \in (-3,-1,+1,+3) \tag{1}$$

If a "+1" symbol followed immediately by a "-1" symbol arrives, the total phase traversal of the two-symbol pair is either -π/8 or +π/8 radians depending on which modulation index is used first. This represents the minimum phase resolution of the system. There are 16 possible states, conveniently represented as positive integers between 0 and 15, representing multiples of π/8 radians.

Because the modulation index alternates between an even numerator to an odd numerator, there are four distinct periods of phase traversal for the system. In the first period, on even-numbered symbols, the modulation index is even so the start and end states are both even. This is referred to as the "E-to-E" period. This is followed by a symbol period where an even state transitions to an odd state because the modulation index numerator is odd ("E-to-O"). By similar reasoning, it can be seen that this is followed by an odd-to-odd ("O-to-O") period and then an odd-to-even ("O-to-E") period, and the cycle repeats. These transitions also describe the distinct structure of the trellis, and demonstrate that the trellis repeats after four symbol periods.

Table 1: Even-to-Even Transition Table for 4-ary CPFSK, h_1=2/8, h_2=3/8

E-TO-E	ARRIVING SYMBOL			
INIT. STATE	"-3"	"-1"	"+1"	"+3"
0	10	14	2	6
1				
2	12	0	4	8
3				
4	14	2	6	10
5				
6	0	4	8	12
7				
8	2	6	10	14
9				
10	4	8	12	0
11				
12	6	10	14	2
13				
14	8	12	0	4
15				

Consider the entries of Table 1, the "E-to-E" table. Only even initial states are possible, so the odd states' rows are grayed out. The only possible state changes are (-6, -2, +2, and +6), which are the product of the symbols times the numerator of the modulation index. If a "-3" symbol arrives at state 14, it maps to a state change of -6, which leads to a final state of 8. If the initial state is 14, and a "+3" arrives, the final state is 20, which is 4 in modulo-16 arithmetic.

Table 2: Even-to-Odd Transition Table for 4-ary CPFSK, h_1=2/8, h_2=3/8

E-TO-O	ARRIVING SYMBOL			
INIT. STATE	"-3"	"-1"	"+1"	"+3"
0	7	13	3	9
1				
2	9	15	5	11
3				
4	11	1	7	13
5				
6	13	3	9	15
7				
8	15	5	11	1
9				
10	1	7	13	3
11				
12	3	9	15	5
13				
14	5	11	1	7
15				

Table 3: Odd-to-Odd Transition Table for 4-ary CPFSK, h_1=2/8, h_2=3/8

O-TO-O	ARRIVING SYMBOL			
INIT. STATE	"-3"	"-1"	"+1"	"+3"
0				
1	11	15	3	7
2				
3	13	1	5	9
4				
5	15	3	7	11
6				
7	1	5	9	13
8				
9	3	7	11	15
10				
11	5	9	13	1
12				
13	7	11	15	3
14				
15	9	13	1	5

Table 4: Odd-to-Even Transition Table for 4-ary CPFSK, h_1=2/8, h_2=3/8

O-TO-E	ARRIVING SYMBOL			
INIT. STATE	"-3"	"-1"	"+1"	"+3"
0				
1	8	14	4	10
2				
3	10	0	6	12
4				
5	12	2	8	14
6				
7	14	4	10	0
8				
9	0	6	12	2
10				
11	2	8	14	4
12				
13	4	10	0	6
14				
15	6	12	2	8

As mentioned before, the Viterbi algorithm keeps track of the best metric entering each state. The matched filter outputs must be computed and de-rotated by the possible starting phases to find the most likely path entering a state. A matrix representation of these operations is shown in Figure 6. The incoming signal vector has *sps* samples per symbol. The matched filters for each possible symbol are columns of a *sps* x 4 matrix. The matrix product yields a 1 x 4 vector of the matched filter outputs. This vector is replicated as rows of an 8 x 4 matrix. Each row represents a possible initial state. A modified 8 x 8 identity matrix with non-zero elements along the main diagonal achieves the de-rotation. The matrix multiplication yields a matrix of matched filter output values. The columns represent the possible arriving symbol and the rows represent the presumed initial state. Note that in the example shown, the initial state is even, so the system is in either the "E-to-E" or "E-to-O" period at this symbol period. If the system

were in the "O-to-O" or "O-to-E" period, the de-rotation matrix elements would change to use odd initial states.

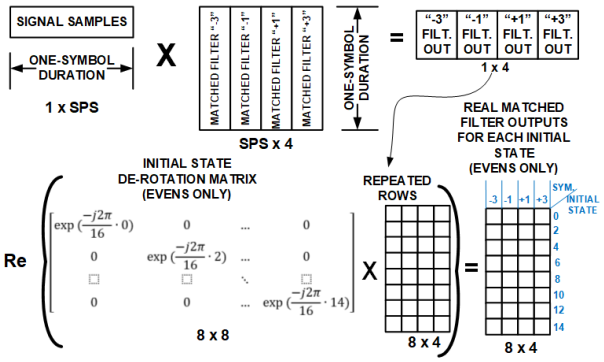

Figure 6: Matrix View of Calculating Symbol Correlations for Possible Initial States (Even Initial States Only)

Implementation of Path Metric of Viterbi Algorithm for 4-ary Multi-h CPFSK

An intuitive implementation of the path metric is a table with 16 rows (one for every initial state) and the traceback length number of columns. The metric table elements are initialized to negative infinity or some large negative value. The real part of a matched filter output might have a negative value, so initializing the table to all zeros could lead to bad decisions. Separate tables of the same size can store the path (or previous state) and the associated transmitted symbols. In a practical system, it would not be necessary to spend extra memory on both the paths and transmitted symbols tables because the trellis diagram shows that every combination of final state and transmitted symbol leads to only one possible previous state. In this paper, the paths table and transmitted data table are kept separate in order to simplify the discussion.

After the matched filter outputs are computed as described in Figure 6, the maximum likelihood path entering each state is computed. There are several steps to this process.

There are eight initial states to consider, each with four possible arriving symbols. For every possible initial state (s), and for every possible arriving symbol (α_n), that state's previously computed metric ($metric(s,n-1)$) is added to the real part of the matched filter output ($mf(\alpha_n)$) after it has been de-rotated by the assumed initial state. This new value is associated with the destination state that would be reached if the starting state is s and symbol α_n is received. This destination state is $s + h_{num} \cdot \alpha_n$. This equation is shown as (2).

$$test\_metric(s + h_{num}\alpha_n, n) = Re\left(mf(\alpha_n) \cdot \exp(\frac{-j2\pi s}{16})\right) + metric(s, n-1) \quad (2)$$

The factor h_{num} is the numerator of the modulation index of the current symbol period. The sum $s+h_{num}\alpha_n$ is performed using modulo 16 arithmetic, and it represents the next state if s is the initial state and symbol α_n arrives. Ordinarily, this expression wouldn't need to be calculated because the computer would already have transition tables (Table 1 through Table 4) in memory and would simple refer to the appropriate table and entry. The product of $Re\{ mf(\alpha_n) \cdot \exp(-j2\pi s/16)\}$ is an entry of the matrix shown on the right-hand side of the equals sign in Figure 6.

Equation (2) represents a test metric because the next step is to compare this value to the value currently stored at $metric(s+h_{num}\alpha_n,n)$. If the test metric is larger, it replaces the currently stored value in the metrics table. At the same time, the corresponding entries in the surviving paths table and the received data symbols table are updated. Equation (3) summarizes the updates.

$$\begin{aligned} &if(test\_metric(s + h_{num}\alpha_n, n) > metric(s + h_{num}\alpha_n, n)) \\ &metric(s + h_{num}\alpha_n, n) \leftarrow test\_metric(s + h_{num}\alpha_n, n) \\ &path(s + h_{num}\alpha_n, n) \leftarrow s \\ &rxdata(s + h_{num}\alpha_n, n) \leftarrow \alpha_n \end{aligned} \quad (3)$$

After the traceback number of symbol periods has elapsed, the metrics table, paths table, and transmitted data table are full. The computer works backwards to deduce the paths and transmitted data. An example of this process can be described with the aid of Table 5, Table 6, and Table 7.

Table 5: Cumulative Matched Filter Metrics After Eight Symbol Periods (SNR = 5 dB, 20 Samples Per Symbol)

	SYMBOL TIME								
STATE	0	1	2	3	4	5	6	7	8
0	20				99.439	103.947			154.361
1			25.4287	80.5403			139.016	150.145	
2		41.8419			80.6477	105.23			156.02
3			62.7894	65.2007			157.399	161.109	
4					74.261	124.151			161.641
5			63.5697	58.4192			147.645	162.691	
6		39.9309			95.3548	137.271			169.873
7			34.5386	43.9903			140.453	164.328	
8					108.736	131.942			171.313
9			40.1466	41.1749			148.068	167.002	
10		10.6708			109.488	127.95			170.155
11			43.6811	43.882			130.909	159.053	
12					112.233	117.077			167.303
13			44.5644	82.8096			145.679	146.984	
14		23.853			97.3791	108.327			158.999
15			46.4515	81.462			145.712	145.896	
	E-TO-E	E-TO-O	O-TO-O	O-TO-E	E-TO-E	E-TO-O	O-TO-O	O-TO-E	E-TO-E

Table 5 shows the first traceback of the metrics table. In this example, the traceback is set to eight symbols. For clarity, an extra column is added at the beginning representing the known initial state of "0" with an arbitrary value of "20" assigned to its metric. Any value greater than negative infinity, including zero, would suffice. Also observe the bottom row of the table shows the type of transition. "E-to-E" is first because the modulation index numerator is even and the initial state is 0, which is even.

The maximum likelihood state in symbol time 8 is state 8 with a metric of 171.313 (highlighted green). The state=8, time=8 entry in the paths table shows a value of 5, representing the previous state. The corresponding entry in the transmitted symbols table shows that the data was 2. The data (0,1,2,3) maps to symbols (-3,-1,+1,+3), so a "+1" symbol was transmitted at symbol time 8. Note that if the system were in state 5 at time 7 and a "+1" symbol is transmitted, the next state would be 5+(+1)*3=8.

Working backwards, it was already decided that the previous state was 5. So the time = 7, state = 5 entry in the metric table is highlighted in green. This metric value is only 162.691, which is not the highest metric at time = 7. The highest metric at this time is actually 167.002 at state 9 (highlighted yellow). That metric was ignored because there is no path from state 9 at time 7 to state 8 at time 8. A similar situation occurs at time 3, where the metric of the maximum likelihood path is not the highest metric at time 3. This example illustrates the need to keep track of the metric magnitudes and the paths.

Table 6: Surviving Paths Table After Eight Symbol Periods (SNR = 5 dB, 20 Samples Per Symbol)

	SYMBOL TIME							
STATE	1	2	3	4	5	6	7	8
0				13	10			3
1		10	3			10	3	
2	0			15	12			5
3		6	5			6	9	
4				13	10			7
5		2	3			8	3	
6	0			15	12			9
7		14	5			4	9	
8				1	10			5
9		6	3			6	3	
10	0			13	12			3
11		2	5			8	5	
12				15	10			3
13		6	3			6	7	
14	0			1	12			5
15		2	5			8	9	
	E-TO-O	O-TO-O	O-TO-E	E-TO-E	E-TO-O	O-TO-O	O-TO-E	E-TO-E

Table 7: Received Data Table After Eight Symbol Periods (SNR = 5 dB, 20 Samples Per Symbol)

	SYMBOL TIME							
STATE	1	2	3	4	5	6	7	8
0				2	3			1
1		0	1			0	1	
2	2			2	3			1
3		1	1			1	0	
4				0	0			1
5		2	2			1	2	
6	3			0	0			1
7		3	2			2	1	
8				0	1			2
9		2	3			2	3	
10	0			1	1			0
11		3	3			2	3	
12				1	2			3
13		0	0			0	3	
14	1			1	2			3
15		1	0			0	3	
	E-TO-O	O-TO-O	O-TO-E	E-TO-E	E-TO-O	O-TO-O	O-TO-E	E-TO-E

Once the traceback has been completed, the Viterbi decoder decides that the received data is 2,2,0,1,0,1,2,2. It only outputs the first symbol, 2. The metrics table, paths, table, and symbols table columns are shifted to the left. The rightmost metrics table column is initialized with negative infinity, and the rightmost data table and paths table columns are initialized with invalid values. The next received symbol is processed as in Figure 6, and the newest column of the metrics table is computed according to equations (2) and (3).

The longer the traceback length, the more chances there are for an errant path to be corrected. The penalty for this performance improvement is increased memory usage and a longer time delay between incoming data and final decisions. This penalty becomes intractable for a signal with a time-varying Doppler offset or even a symbol rate offset.

TIMING CONTROL LOOP

The initial Doppler frequency offset of 57 kHz for a 2.45 GHz carrier frequency means that the satellite initially moves with a velocity magnitude of nearly 7 km/s. This results in the 10 kHz symbol rate appearing to be $f_{sym} \cdot v/c$ = 10000.2327 Hz. Over time, the symbol rate offset will degrade the performance of the Viterbi decoder unless it possesses a tracking loop capable of detecting and cancelling this offset. Even if the symbol rate itself is perfect, the receiver will load the Viterbi decoder with bad decisions if the symbol boundaries are not correctly identified because that error will alter the measured starting and ending phases of the symbols.

The effect of a time offset in a CPFSK signal is to impart a phase offset, as shown in Figure 7. The effect of an error in the symbol rate is the accumulation of these phase offsets across multiple symbols.

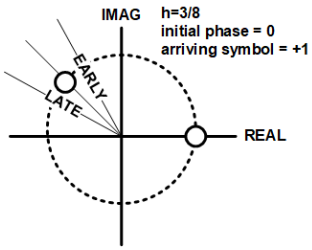

Figure 7: Scenario Demonstrating Timing Error Interpreted as Phase Error

The following is a discussion of a timing control loop using a polyphase interpolator. The architecture is modified from one designed for phase shift keyed signals in Harris' Multirate Signal Processing textbook.[6]

Timing Offset Measurement

In a PSK signal, the time offset can be found from an eye diagram. The ideal sampling instant is the point where the eye is open and the slope of the matched filter output is zero regardless of the arriving symbol. For early and late symbols, the sign of the matched filter output slope times the sign of the arriving symbol gives a consistent estimate of the timing offset.[6] Divide the measured slope by the sign of the arriving symbol to obtain the expressions. If this is a positive value, the signal has been sampled early.

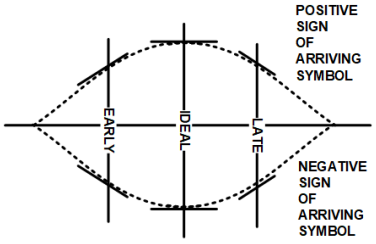

Figure 8: Eye Diagram View of PSK Symbol Timing Errors

Since CPFSK signal use multiple matched filters, one for every possible arriving symbol, the diagram in Figure 8 does not directly apply. However, for 4-ary CPFSK, there are two pairs of matched filters which are complex conjugates of each other. The imaginary portions of the matched filter outputs produce a similar eye diagram when they are input with only the imaginary portion of the input signal. Figure 9 demonstrates this for the matched filters for the h=2/8 modulation index. A similar pair of eye diagrams also exists for the h=3/8 modulation index. Timing offsets can be measured in a manner similar to how it is done for PSK signals.

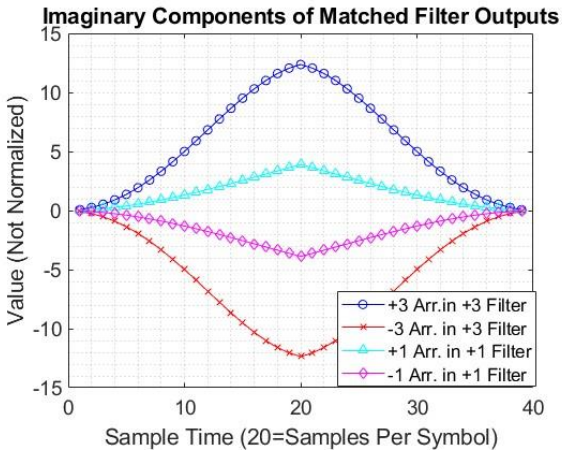

Figure 9: Imaginary Components of CPFSK 4-ary Matched Filter Outputs (for h=2/8)

A way to quickly measure the slope is to have a bank of derivative matched filters computed by convolving the matched filter coefficients with the sequence [1/2, 0, -1/2], and then removing the first and last elements in order to produce filters that are the same length as the matched filters. As there are different matched filters for each modulation index, so there are different derivative matched filters. The expression for the timing measurement involves the product of the imaginary portions of the matched filter and the derivative matched filter outputs. Since the correct matched filter is not known until after the Viterbi decoder finalizes a decision, a new table (the *timing* table) is needed to keep track of the measured timing offsets. Equation (3) must be updated to include this new table in the list of tables to update in the Viterbi decoder. The new update is reflected in equation (4). In this new expression, $dmf(\alpha_n)$ refers to the derivative matched filter output of symbol α_n, just as $mf(\alpha_n)$ denotes the matched filter output of symbol α_n. Also note that the measurements from the matched filter and derivative matched filter need to be de-rotated by the initial phase.

$$
\begin{aligned}
&if(test\_metric(s + h_{num}\alpha_n, n) > metric(s + h_{num}\alpha_n, n)) \\
&metric(s + h_{num}\alpha_n, n) \leftarrow test\_metric(s + h_{num}\alpha_n, n) \\
&path(s + h_{num}\alpha_n, n) \leftarrow s \\
&rxdata(s + h_{num}\alpha_n, n) \leftarrow \alpha_n \\
&timing(s + h_{num}\alpha_n, n) \leftarrow \\
&\quad Im\{mf(\alpha_n) \cdot \exp\left(\frac{-j2\pi s}{16}\right)\} \cdot Im\{dmf(\alpha_n) \cdot \exp\left(\frac{-j2\pi s}{16}\right)\}
\end{aligned}
\tag{4}
$$

When the Viterbi decoder selects the maximum likelihood paths from the current symbol back to the

traceback length, it also chooses the time offsets associated with those paths. It computes a weighted average value and reports this as the time error estimate $\tau(n)$ for symbol index n. It uses a constant scaling factor b less than unity. It multiplies the most recent measurement by b, the second most recent measurement by b^2, and so forth. This gives increased weight to the most recent measurement and yields the expression in equation (5). For the design in this paper, $b = 0.125$ was selected.

$$\tau(n) = \frac{1}{traceback} \sum_{i=0}^{traceback-1} timing(s_{ML\_i}, i) \cdot b^{i+1} \tag{5}$$

In equation (5), s_{ML_i} refers to the state at time index i associated with the maximum likelihood path. Note that the time indices i of the right-hand side of (5) refer to the columns of the timing metrics table rather than absolute time.

Polyphase Timing Interpolator

With the measurement task completed, the system needs a means to resample the incoming signal at a different sample time without causing any additional distortion to the signal. A bank of 32 polyphase filters defined from an oversampled impulse function accomplishes this task. Equation (6) gives the expression for the time series of each row of the polyphase filter.

$$g(row, n) = \frac{\sin\left(\frac{\pi(n \cdot nfilts + row)}{nfilts}\right)}{\left(\frac{\pi(n \cdot nfilts + row)}{nfilts}\right)} \tag{6}$$

In equation (6), *nfilts* is the number of filters in the polyphase interpolator, 32 in this case. It can be seen that in the zeroth row, the equation simplifies to $\sin(\pi n)/(\pi n)$, which is equivalent to a delta function. Sample index n is an integer range that needs to include zero so that the proper ideal sample time is included in the bank of filters. The time series of the first five rows of the polyphase filter are shown in Figure 10, where the complete oversampled impulse time series is also shown for clarity.

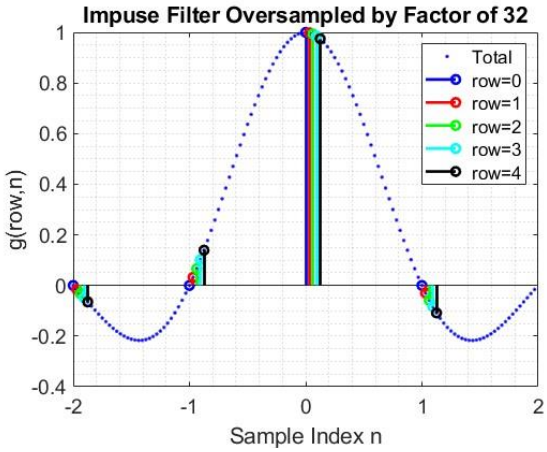

Figure 10: First Five Rows of 32-Bank Polyphase Impulse Filter from Equation (6)

The desired property of the polyphase filter bank is to have each row's filter possess a monotonically increasing delay as the row number increases. Increasing the row number by one increases the signal delay by 1/*nfilts* of a sample. The difference in delay between the zeroth row (perfect timing) and the last row is (*nfilts* -1)/*nfilts* of a sample, which is almost unity.

Operation of the Timing Control Loop

A proportional-integral filter smooths the timing measurements and adds them to a timing error accumulator, as shown in Figure 11. The coefficients of this filter k_{p_T} and k_{i_T} are chosen by phase locked loop design equations derived in the Appendix. A damping factor ζ and a loop bandwidth parameter η are chosen, and from these choices, the values of the coefficients are computed. For this particular timing control system, the values of $\zeta_T=2$ and $\eta_T=2\pi/250$ were selected.

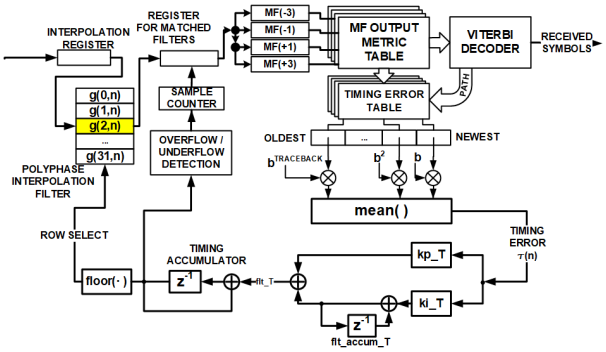

Figure 11: Detailed View of Timing Control Loop Embedded in Viterbi Decoder

The integer component of the timing accumulator selects the row of the polyphase filter bank. This is the row with the impulse filter taps that are closest to the measured timing offset. The accumulator overflows if its value

exceeds the number of filters in the polyphase filter bank. This happens if the control loop seeks to delay the signal by more than one sample. In this situation, an extra clock is added to the sample counter so that an extra sample of delay occurs before processing the next symbol. At the same time, the contents of the timing accumulator are reduced by *nfilts* to keep it within the one-sided open range of [0, *nfilts*). In a practical implementation, this is achieved naturally by representing the timing accumulator with a fixed-point field chosen so that its maximum value is a fraction less than *nfilts*. For this reason, *nfilts* is usually specified as a power of two.[6]

Similarly, if the timing accumulator falls below zero, the sample counter is reduced by one signifying that the signal needs to be advanced by one sample. At the same time, the accumulator contents are incremented by *nfilts* to keep it in the proper range. If the accumulator is structured as a fixed-point field such that it automatically rolls over as a positive number when it dips below zero, the logic block that detects this condition must be carefully designed to use the proper operator.

An improved timing estimate can be calculated by considering the two nearest rows.[6] A linear combination of their coefficients weighted by the non-integer portion of the timing accumulator produces this result.

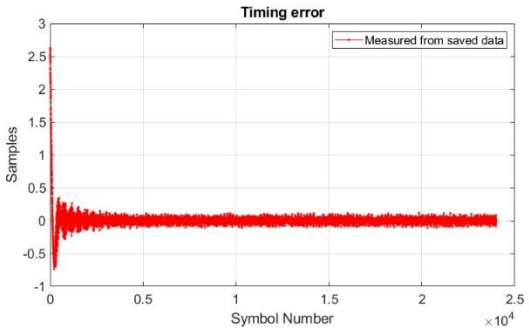

Figure 12: Response of Timing Control Loop to Symbol Rate Offset (10.003 kHz instead of 10 kHz)

A demonstration of the performance of this timing control loop is seen in the simulation results of Figure 12. The graph shows that for a symbol rate offset of 3 Hz, which represents a dynamically changing sample time offset, the control loop is able to lock onto and track the proper sampling time of the signal.

PHASE LOCKED LOOP

With the symbol timing corrected, the incoming signal may still possess a phase offset. The matched filter output of the maximum likelihood symbol after the initial phase has been de-rotated is the phase error. Once again, it is up to the Viterbi decoder to decide which initial state is correct (or which path) and which received symbol is correct. Final decisions do not happen until after the traceback length. Therefore, a separate phase metrics table needs to keep track of this information until the final decisions. Equation (7) shows an updated list of metric table updates which now includes the phase metric table (referred to as simply "phase" in the equation). The information is stored in same row and column as all the other metric tables. The information stored is the arctangent of $mf(\alpha_n)$, the matched filter output of the candidate symbol (α_n), after it has first been de-rotated by the phase corresponding to the candidate initial state *s*.

$$
\begin{aligned}
&if(test\_metric(s + h_{num}\alpha_n, n) > metric(s + h_{num}\alpha_n, n)) \\
&metric(s + h_{num}\alpha_n, n) \leftarrow test\_metric(s + h_{num}\alpha_n, n) \\
&path(s + h_{num}\alpha_n, n) \leftarrow s \\
&rxdata(s + h_{num}\alpha_n, n) \leftarrow \alpha_n \\
&timing(s + h_{num}\alpha_n, n) \leftarrow \\
&\quad Im\left\{mf(\alpha_n) \cdot \exp\left(\frac{-j2\pi s}{16}\right)\right\} \cdot Im\left\{dmf(\alpha_n) \cdot \exp\left(\frac{-j2\pi s}{16}\right)\right\} \\
&phase(s + h_{num}\alpha_n, n) \leftarrow atan\left(mf(\alpha_n) \cdot \exp\left(\frac{-j2\pi s}{16}\right)\right)
\end{aligned}
\tag{7}
$$

At each symbol time, when the Viterbi decoder recomputes the maximum likelihood path, the number of phase errors the Viterbi decoder has deduced is equal to the number of traceback symbols, which is the same number of timing error measurements it had to keep track of. A weighting is applied to the phase error measurements such that the newest measurement is weighted more than the older measurements. The same weighting that was used for the timing errors is used for the phase errors in order to determine a single unified phase error to feed to the phase locked loop. This is shown in equation (8).

$$
\Delta\theta(n) = \frac{1}{traceback} \sum_{i=0}^{traceback-1} phase(s_{ML\_i}, i) \cdot b^{i+1} \tag{8}
$$

Once again, the index *i* in the equation is a column of a table rather than an absolute symbol index. Earlier, it was stated that a timing error gets interpreted as a phase error. It is important to use different techniques for measuring time and phase offset so that the respective control loops do not positively reinforce each other and become unstable.

A second order phase locked loop, designed with a damping coefficient of $\zeta = 1/\sqrt{2}$ and a loop bandwidth η

= 2π/200 includes another proportional-integral filter and numerically controlled oscillator (see Figure 13). The coefficients of the proportional and integral branches of the filter, k_{p_θ} and k_{i_θ}, are computed using the selection of ζ and η and the equations in the Appendix.

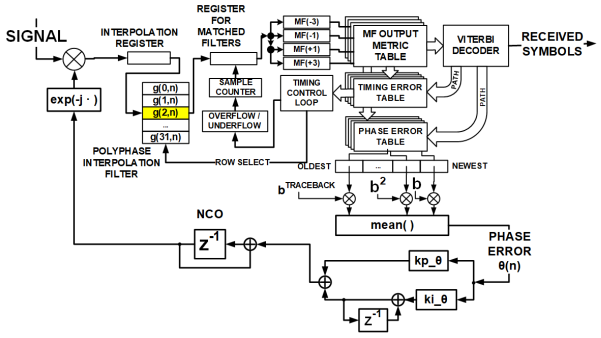

Figure 13: CPFSK Receiver with Detailed Phase Locked Loop

DOPPLER CORRECTION TECHNIQUE

Several techniques were considered for correcting the time-varying Doppler of the CFPSK signal. Since the signal uses a frequency change instead of a phase change for its information content, a delay-multiply Doppler detector does not work.[7] Measuring the difference in energy between upper and lower band-edge filters is a better approach. Optimum band-edge filters have a frequency response equal to the frequency derivative of the matched filter frequency response.[8] The CPFSK signal has multiple matched filters, so a composite matched filter comprised of the sum of the possible symbol matched filters can be used to create optimum band-edge filters. However, because the expected magnitude of the Doppler frequency far exceeds the modulation bandwidth of the signal, optimum band-edge filters will fail to capture the signal's energy. Instead, modified sloped-frequency response half-band band-edge filters are used to bring the signal into the processing bandwidth of the receiver.

Non Data-Aided Band-Edge FLL

The band-edge FLL design chosen for this CPFSK receiver uses a modified version of the sloped frequency response half-band band-edge filters first described in a paper submitted to the 2021 Small Satellite Conference.[9] Frequency-sloped half-band band-edge filters collect more energy for larger frequency offsets and gradually diminish the magnitude of the frequency error measurement as the signal approaches zero Doppler. The technique used to generate the filters was to create a ramp function defined in the frequency domain by taking the absolute value of the ratio of frequency to sample rate. Its inverse Fourier transform was computed and truncated to a length of 50 coefficients. The resulting time series was convolved with a 25-element positive and negative half-band filters and scaled to have unity gain near the half-sample frequencies, as shown in Figure 14.

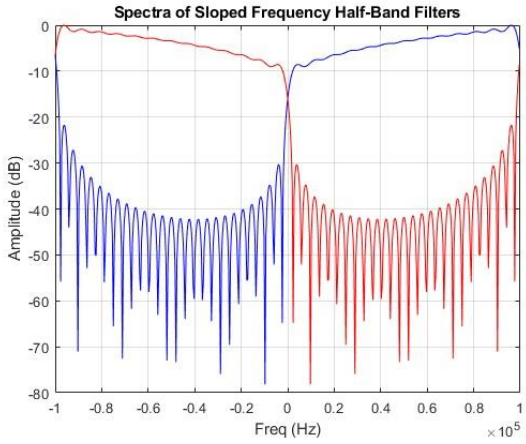

Figure 14: Original Sloped Half-Band Band-Edge Filter Spectra

The spectra of the original sloped half-band band-edge filters included ripple caused by the truncation of the ramp's inverse Fourier transform as well as the truncation of the ideal half-band filter prototype. To remove these ripple sources, a Kaiser window with β=7 was applied to the half-band prototype and separately applied to the frequency-domain ramp prior to its inverse Fourier transform. The resulting band-edge filters are shown in Figure 15.

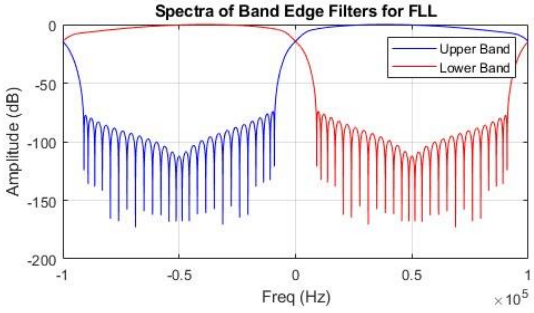

Figure 15: Modified Band-Edge Filters Used in FLL After Applying Kaiser Window to Ramp and Half-Band Prototypes

The sum and differences of the band-edge filter outputs create signals *cc(t)* and *ss(t)* as seen in the block diagram of Figure 16. Harris' derivation shows that the real part of the product of *cc(t)* and the conjugate of *ss(t)* is the difference in energy between the filters, which is proportional to the frequency offset.[8]

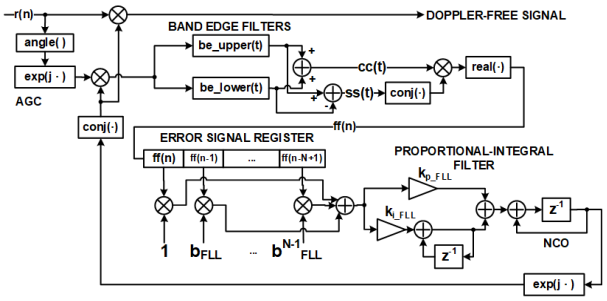

Figure 16: Non Data-Aided Band-Edge FLL for CPFSK Signals

There are key differences between the FLL described in Figure 16 and the description given in the previously submitted paper.[9] First, there is an automatic gain control (AGC) section to the FLL because the best choices for the damping ratio ζ and the loop bandwidth η are affected by the signal amplitude. An effective way to remove this sensitivity is to extract only the phase information from the incoming signal so that it has unity amplitude. This is achieved by the successive *angle(·)* and *exp(j ·)* blocks between the arriving signal and the mixer. Observe that a second mixer, omitting these blocks, provides the Doppler-free signal at the original amplitude. Next, frequency error measurements *ff(n)* are collected into a *N*-element register and weighted according to the age of the measurement. The choice of this register length is independent of the Viterbi traceback length or any other parameter in the other control loops, because the intention is to use a non data-aided frequency correction. A small factor less than unity, b_{FLL}, is successively applied to the measurement with each sample period. The purpose of this structure is to de-emphasize measurements caused by the inherent changing frequency of the signal of interest.

Performance of Non Data-Aided Band-Edge FLL

A simulation of a the FLL operating on a 4-ary CPFSK multi-h signal is shown in two figures. With the automatic gain control included, the FLL's proportional-integral filter was designed with $b_{FLL} = 0.9$, $\zeta = 0.01$, $\eta = 2\pi/2500$, and the equations in the Appendix. The measured vs. actual Doppler frequency offset is in Figure 17. The waterfall plot of the FLL-corrected signal is shown in Figure 18. Both plots show that within half of a second, the FLL locks onto the signal.

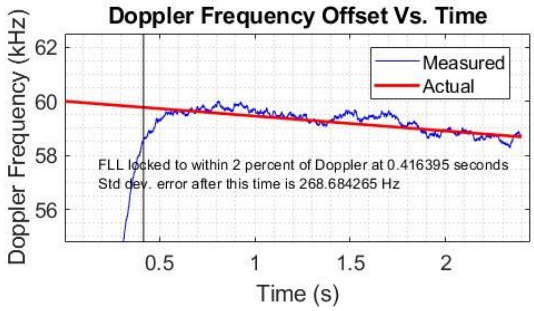

Figure 17: Simulation of Non Data-Aided Band-Edge FLL with 4-ary CPFSK (h_1=2/8, h_2=3/8), Initial Doppler=60 kHz; Doppler Rate of Change = -550 Hz/s, SNR=10 dB

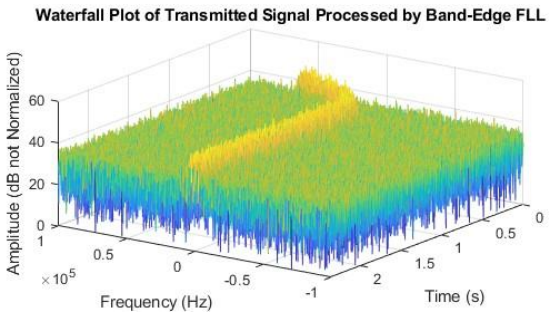

Figure 18: Waterfall Plot of Spectrum of Signal Under Influence of Sloped Half-Band Band-Edge FLL; Initial Doppler = 60 kHz, Doppler Rate of Change = -550 Hz/s, SNR = 10 dB

The inclusion of the Doppler correcting band-edge FLL with the rest of the receiver design is shown in Figure 19. The automatic gain control needed for reliable operation of the FLL is branched off the main path and precedes the FLL. The receiver may have its own separate AGC. Also notice the inclusion of a noise reducing filter following the FLL and preceding the rest of the receiver. In order for the system to approach the theoretical achievable symbol and bit error rates, the noise should be reduced to a bandwidth equal to the modulation rate. The proper place to put the noise reduction filter is after the FLL. Placing the filter before the FLL will result in the removal of the incoming signal when the Doppler shift exceeds the modulation bandwidth.

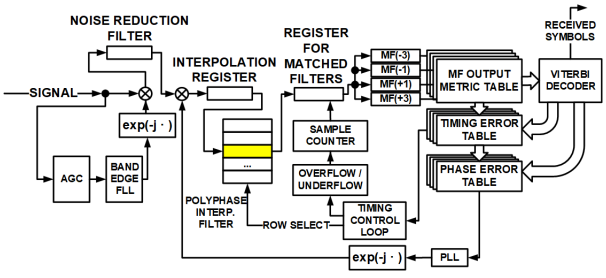

Figure 19: CPFSK Complete Receiver Diagram

CONCLUDING REMARKS

Further improvement to the performance of the complete receiver requires the inclusion of a means to detect the presence of a signal and an initial preamble or other known structure that can be used to quickly obtain a Doppler estimate. This estimate should initialize the modified half-band band-edge FLL in order to minimize the frequency locking time. The locked FLL oscillates between the different information bearing frequencies inherent in the CPFSK signal which can cause unintentional frequency modulation that will produce errors in the receiver. Instead of keeping the FLL in operation, a filtered version of the FLL error signal is used to determine if the FLL is in lock or not. When it is in lock, it can shut off its frequency accumulator update so that the data-aided phase locked loop can complete the operation of tracking the residual frequency offset.

Acknowledgments

The authors wish to acknowledge NIWC Pacific's Naval Innovative Science and Engineering (NISE) program committee for its continuing supporting for the project that led to this paper and to the papers submitted in previous Small Satellite Conference proceedings.

APPENDIX

A stable second-order analog phase-locked loop has a denominator with the structure shown in (A.1).

$$\frac{\theta_o(s)}{\theta_i(s)} = \frac{numerator}{s^2+2\zeta\omega_n s+\omega_n^2} \tag{A.1}$$

The digital phase-locked loop with proportional constant k_p and integral constant k_i has a block diagram shown in Figure 20 and a transfer function as expressed in (A.2).

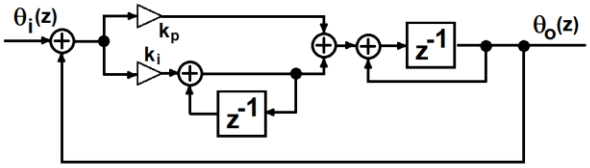

Figure 20: Second Order Digital Phase Locked Loop

$$\frac{\theta_o(z)}{\theta_i(z)} = \frac{(k_p+k_i)z-k_p}{z^2+(k_p+k_i-2)z+(1-k_p)} \tag{A.2}$$

In this derivation, the goal is to find a relationship between the proportionality constants k_p and k_i, the analog damping factor ζ, and the natural damping frequency ω_n. Our starting point is to replace the denominator of the analog system of (A.1) with an equivalent digital domain expression. The bilinear transform relating s to z is shown in (A.3).

$$s = \frac{2}{T_s}\left(\frac{z-1}{z+1}\right) \tag{A.3}$$

In (A.3), the sample period is T_s. Substituting the bilinear transform expression into (A.1) produces a new expression in terms of z as shown in (A.4).

$$\frac{\theta_o(z)}{\theta_i(z)} = \frac{numerator}{\left(\frac{z-1}{z+1}\right)^2+2\zeta\left(\frac{\omega_n T_s}{2}\right)\left(\frac{z-1}{z+1}\right)+\left(\frac{\omega_n T_s}{2}\right)^2} \tag{A.4}$$

Replacing the ratio $\omega_n T_s/2$ with η and multiplying the numerator and denominator by $(z+1)^2$, yields (A.5).

$$\frac{\theta_o(z)}{\theta_i(z)} = \frac{numerator}{(z-1)^2+2\zeta\eta(z+1)(z-1)+\eta^2} \tag{A.5}$$

Next, the powers of z are combined in the denominator of (A.5) and the result is compared to (A.2).

$$\frac{\theta_o(z)}{\theta_i(z)} = \frac{numerator}{z^2+z\left(\frac{2\eta^2-2}{1+2\zeta\eta+\eta^2}\right)+\left(\frac{1-2\zeta\eta+\eta^2}{1+2\zeta\eta+\eta^2}\right)} \tag{A.6}$$

Comparing (A.6) to (A.2), produces two equations with two unknowns:

$$1-k_p = \frac{1-2\zeta\eta+\eta^2}{1+2\zeta\eta+\eta^2} \tag{A.7}$$

$$k_p + k_i - 2 = \frac{2\eta^2+2}{1+2\zeta\eta+\eta^2}$$

Solving (A.7) for k_p and k_i completes the derivation, with the result in (A.8)

$$k_p = \frac{4\zeta\eta}{1+2\zeta\eta+\eta^2} \tag{A.8}$$

$$k_i = \frac{4\eta^2}{1+2\zeta\eta+\eta^2}$$

References

1. Andrisano, Oreste and Giorgio Corazza and Gianni Immovilli, "Effects of Nonlinear Power Amplifiers on the Spectrum of CPFSK Signals," MILCOM 1984 IEEE Military Communications Conference, Los Angeles, CA, October 1984.
2. Cripps, Steve C., RF Power Amplifiers for Wireless Communications, Artech House, Boston, 2005.
3. Mazur, Brian A. and Desmond P. Taylor, "Demodulation and Carrier Synchronization of Multi-h Phase Codes," IEEE Transactions on Communications, vol. COM-29, No. 3, March 1981.

4. Premji, Al-Nasir and Desmond P. Taylor, “A Practical Receiver Structure for Multi-h CPM Signals,” IEEE Transactions on Communications, vol. COM-35, No. 9, September 1987.

5. Forney, G. David, “The Viterbi Algorithm,” Proceedings of the IEEE, vol. 61, No. 3, March 1973.

6. Harris, Fred, Multirate Signal Processing, Prentice Hall Professional Technical Reference, Upper Saddle River, NJ, 2004, pages 409-416.

7. Hill, Brendan and Nazia Mozaffar and Salwan Damman, “A Comparison of Techniques for Non Data-Aided Carrier Tracking of Phase-Modulated Signals”, Small Satellite Conference, Logan UT, August 2020.

8. Harris, F., “Band Edge Filters: Characteristics and Performance in Carrier and Symbol Synchronization,” Proceedings of the 13th International Symposium on Wireless Personal Multimedia Communications, Recife, Brazil, October 11-14, 2010.

9. Hill, Brendan and Nazia Mozaffar and Salwan Damman, “Non Data-Aided Carrier Tracking Techniques for Continuous-Phase Frequency-Shift Keyed Signals”, Small Satellite Conference, Logan UT, August 2021.