EVERYTHING YOU WANTED TO KNOW ABOUT DOUBLE DIFFERENTIAL ENCODERS BUT WERE AFRAID TO ASK

Erik Perrins
Department of Electrical Engineering & Computer Science
University of Kansas
Lawrence, KS 66045
esp@ieee.org

ABSTRACT

The existing offset quadrature phase shift keying (OQPSK) differential encoder in IRIG-106 is a curious scheme with a rather mysterious origin. In this paper, an alternative scheme known as *double differential encoding* is proposed. In many aspects, the proposed scheme has equivalent performance to the existing scheme: it successfully resolves the 4-phase ambiguity introduced by most carrier phase tracking loops and it also produces two decoded bit errors for each detection error. However, the proposed scheme has a number of conceptual advantages: it can be derived easily from first principles, it decouples the operations of even-bit/odd-bit demultiplexing and differential encoding, and it greatly simplifies the overly-complicated binary-to-ternary symbol mapping for OQPSK. It is also demonstrated to have tangible benefits, such as improved performance in systems with error control coding.

INTRODUCTION

Offset quadrature phase-shift keying (OQPSK), like many other modulations, can suffer from a phase ambiguity problem at the receiver. During transmission, the signal experiences a phase shift that must be corrected by the receiver. The phase shift can be resolved to a certain degree by a carrier phase recovery scheme. However, due to the rotational symmetry of the signal constellation, there are a discrete number of outcomes that appear equally correct to the carrier phase recovery scheme. The only method for deciding which of these phase shifts is absolutely correct is to transmit data that is known to the receiver; the absolute phase shift can be identified in the known data and corrected in the remainder of the data that carries information to the receiver.

In the absence of known data, a method of coping with an uncorrectable phase ambiguity is to transmit the information with *phase shifts*, rather than *absolute phases*. Since phase shifts are preserved in the received signal, they can be recovered and the transmitted information can be obtained. This type of signaling is called *differential encoding*.

In this paper, a differential encoding scheme is proposed for use with the OQPSK-type modulations known as the "ARTM Tier I" waveforms in IRIG-106. The organization and development of this paper closely parallels that in [1], and an excellent graphical and written overview of the phase ambiguity problem can be found there. The existing differential encoding scheme in [1] is referred to as the "ARTM differential encoder" in this paper. The ARTM differential encoder has a non-obvious form and one can only guess at its motivation since the original documentation has proved to be elusive [1]. The scheme proposed here, on the other hand, can be derived easily from first principles. It is also less rigid than the existing scheme since the order of operations between even-bit/odd-bit demultiplexing and differential encoding are interchangeable. Another advantage of the proposed scheme is that it greatly simplifies the binary-to-ternary symbol mapping for OQPSK. This mapping is part of the continuous phase modulation (CPM) model for OQPSK and its *shaped* cousin, SOQPSK. For these reasons, the double differential encoder is a "natural" companion to OQPSK.

The advantages pointed out so far are mainly conceptual. In terms of performance, the proposed scheme is equal to the existing scheme in coping with the 4-phase phase ambiguity of OQPSK. Like the existing scheme, it also produces two decoded bit errors for each detection error. Therefore, with existing applications there is no performance improvement with the proposed scheme. However, simulation results are provided which show double differential decoding has an advantage over ARTM differential encoding in serially concatenated coded systems with iterative detection. Thus, in addition to the conceptual advantages of the proposed scheme, there are tangible benefits as well.

The double differential encoder and decoder are first explained. The two are then characterized in terms of the 4-phase ambiguity, initial values, error propagation, and post-multiplexed decoding. Finally, the binary-to-ternary relationship is developed and simulation results are given for the coded system.

DOUBLE DIFFERENTIAL ENCODER

Consider the *information bearing* sequence $\boldsymbol{a}$, where a_n is the n-th bit from this sequence. A *differentially encoded* sequence $\boldsymbol{b}$ can be derived from the original sequence $\boldsymbol{a}$ on a bit-by-bit basis via the operation

$$b_n = a_n \oplus b_{n-1}, \quad \text{where} \quad a_n, b_n \in \{0,1\}, \qquad n \in \{0,1,2,\cdots\}, \tag{1}$$

and $\oplus$ is the binary XOR operator. The differential encoding rule in (1) can be summarized as "change phase on 1" since an input of $a_n = 1$ causes the output value to change relative to the previous value. The opposite formulation of "change phase on 0" can be used with equivalent results. If $\boldsymbol{b}$ in turn undergoes another differential encoding operation, the resulting bits are

$$c_n = b_n \oplus c_{n-1} \tag{2}$$

and $\boldsymbol{c}$ is referred to as a *double differentially encoded* [2] sequence. Substituting (1) for b_n and (2) for c_{n-1} yields

$$\begin{aligned} c_n &= \underbrace{a_n \oplus b_{n-1}}_{b_n} \underbrace{\oplus b_{n-1} \oplus c_{n-2}}_{c_{n-1}} \\ &= a_n \oplus c_{n-2} \end{aligned} \tag{3}$$

which shows that $\boldsymbol{b}$ does not need to be brought into existance since $\boldsymbol{c}$ can be obtained directly from $\boldsymbol{a}$.

Another way to obtain $\boldsymbol{c}$ is to demultiplex the original bit sequence $\boldsymbol{a}$ into even- and odd-indexed bits

$$\left.\begin{aligned} e_k &= a_k \\ o_{k+1} &= a_{k+1} \end{aligned}\right\} \qquad k \in \{0,2,4,\cdots\}$$

and then apply a *single differential encoder* to each subsequence

$$\begin{aligned} I_k &= e_k \oplus I_{k-2} \\ Q_{k+1} &= o_{k+1} \oplus Q_{k-1}. \end{aligned} \tag{4}$$

The sequences formed by I_k and Q_{k+1} are ready to drive the I (inphase, "real," or "cosine") and Q (quadrature, "imaginary," or "sine") subcarriers of the modulator, respectively. Notice that I_k and Q_{k+1} are simply

Table 1: Axis rotations that result from the phase ambiguity β.

β	+I′	+Q′
0	+I	+Q
$\pi/2$	−Q	+I
π	−I	−Q
$3\pi/2$	+Q	−I

the even- and odd-indexed bits of $\boldsymbol{c}$. Therefore, the order of the differential encoding and even/odd demultiplexing operations are interchangeable in the case of the double differential encoder. This is not the case for the ARTM differential encoder [1]

$$\begin{aligned} I_k &= e_k \oplus \overline{Q}_{k-1} \\ Q_{k+1} &= o_{k+1} \oplus I_k \end{aligned} \tag{5}$$

where the bits must be demultiplexed into even and odd bits before being differentially encoded. The $\overline{(\cdot)}$ operator is the logical complement.

DOUBLE DIFFERENTIAL DECODER

The transmitter sends the complex baseband OQPSK-type signal $s(t)$. The demodulator receives the signal

$$r(t) = s(t)e^{j\phi(t)} + w(t)$$

where $w(t)$ is additive noise and $\phi(t)$ is a time-varying phase shift that arises from transmission delay, channel conditions, motion between the transmitter and receiver, phase and frequency offsets between the transmit and receive oscillators, etc. Assuming perfect operation of the phase tracking loop, a *synchronized* version of the received signal is available, i.e.

$$r'(t) = s(t)e^{j\beta} + w(t), \qquad \beta \in \{0, \pi/2, \pi, 3\pi/2\},$$

where β is the (unknown) phase ambiguity. The inphase and quadrature subcarriers in the demodulator are I' and Q'. Unfortunately, when the phase ambiguity is present, the transmitted subcarriers are reassigned on the receiving end by the carrier phase recovery system. Table 1 shows the assignments between I' and Q' and I and Q as a function of β (the information in the table is taken from [1, Table M-1] and is depicted graphically in [1, Figure M-3]).

The motivation for the using differential encoding is that the original information sequence $\boldsymbol{a}$ can be recovered in spite of the axis reassignments in Table 1. The even- and odd-indexed bits at the demodulator are

$$\begin{aligned} e'_k &= I'_k \oplus I'_{k-2} \\ o'_{k+1} &= Q'_{k+1} \oplus Q'_{k-1}. \end{aligned} \tag{6}$$

These can be multiplexed into a single sequence $\boldsymbol{a}'$. As was the case at the encoder, the order of the differential decoding and even/odd multiplexing operations is interchangeable. Again, this is proven not to be the case with the ARTM differential decoder in [1]. The differential decoding rule using $\boldsymbol{c}'$ directly is

$$a'_n = c'_n \oplus c'_{n-2}. \tag{7}$$

The correct operation of (6) and especially (7) is now formally demonstrated for the double differential decoder with OQPSK.

IMMUNITY TO 4-PHASE AMBIGUITY

The first case for the phase ambiguity is $\beta = 0$, which does not require explicit proof since $I'_k = I_k$ and $Q'_{k+1} = Q_{k+1}$. Therefore, (6) gives the correct result by definition.

$$\underline{\text{Case II:}}\ \ \beta = \pi/2 \quad \Rightarrow \quad \begin{cases} I'_k & = \overline{Q}_{k-1} \\ Q'_{k+1} & = I_k. \end{cases}$$

In this case, the outputs of the two single differential decoders are

$$e'_k = I'_k \oplus I'_{k-2} = \overline{Q}_{k-1} \oplus \overline{Q}_{k-3} = Q_{k-1} \oplus Q_{k-3} = o_{k-1}$$

and

$$o'_{k+1} = Q'_{k+1} \oplus Q'_{k-1} = I_k \oplus I_{k-2} = e_k.$$

The correct bits are recovered with a delay of one bit time relative to the $\beta = 0$ case.

$$\underline{\text{Case III:}}\ \ \beta = \pi \quad \Rightarrow \quad \begin{cases} I'_k & = \overline{I}_k \\ Q'_{k+1} & = \overline{Q}_{k+1}. \end{cases}$$

In this case, the outputs of the two single differential decoders are

$$e'_k = I'_k \oplus I'_{k-2} = \overline{I}_k \oplus \overline{I}_{k-2} = I_k \oplus I_{k-2} = e_k$$

and

$$o'_{k+1} = Q'_{k+1} \oplus Q'_{k-1} = \overline{Q}_{k+1} \oplus \overline{Q}_{k-1} = Q_{k+1} \oplus Q_{k-1} = o_{k+1}.$$

The correct bits are recovered with no delay relative to the $\beta = 0$ case.

$$\underline{\text{Case IV:}}\ \ \beta = 3\pi/2 \quad \Rightarrow \quad \begin{cases} I'_k & = Q_{k-1} \\ Q'_{k+1} & = \overline{I}_k. \end{cases}$$

In this case, the outputs of the two single differential decoders are

$$e'_k = I'_k \oplus I'_{k-2} = Q_{k-1} \oplus Q_{k-3} = o_{k-1}$$

and

$$o'_{k+1} = Q'_{k+1} \oplus Q'_{k-1} = \overline{I}_k \oplus \overline{I}_{k-2} = I_k \oplus I_{k-2} = e_k.$$

The correct bits are recovered with a delay of one bit time relative to the $\beta = 0$ case.

In all cases the decoder recovers the correct bit sequence. The decoding is delayed by one bit time in half of the cases. This minor delay is inconsequential relative to the typical transmission delay between the transmitter and the receiver. The above cases have been formulated in terms of e_k, o_{k+1}, I_k, and Q_{k+1}; the problem can also be formulated in terms of a_n and c_n, where the end result is the same.

INITIAL VALUES

Using the notation established in [1], $\langle \cdot \rangle$ denotes an unknown initial value and $|| \cdot ||$ denotes computed values influenced by unknown initial values. Applying the single differential encoders in (4) for $k = 0, 2, 4, \cdots$ results in

$$\begin{aligned} ||I_0|| &= e_0 \oplus \langle I_{-2} \rangle \\ ||Q_1|| &= o_1 \oplus \langle Q_{-1} \rangle \\ ||I_2|| &= e_2 \oplus ||I_0|| \\ &\vdots \end{aligned}$$

which shows that the initial values influence the entire set of differential bits at the transmitter. However, the decoders in (6) each use two such influenced values on the right-hand side of the equation. Therefore, the influence of the initial values cancels out in the decoding operation. The exceptions to this rule are the first two bits out of the differential decoder (i.e. $k = 0$)

$$||e_0'|| = I_0' \oplus \langle I_{-2}' \rangle$$
$$||Q_1'|| = Q_1' \oplus \langle Q_{-1}' \rangle$$

where the influence of the decoder's own initial value assumptions is present. These initial values are flushed out of the decoder for $k > 0$.

ERROR PROPAGATION

Since the received code symbols influence more than one decoded bit, the double differential decoder incurs the same bit error penalty experience by other differential decoders. Suppose the symbol I_k' is in error; we label this value ϵ_k. A sequence of decoding operations produces

$$\begin{aligned} e_k' &= \epsilon_k \oplus I_{k-2}' && \text{(decoded bit error)} \\ o_{k+1}' &= Q_{k+1}' \oplus Q_{k-1}' && \text{(correct decoded bit)} \\ e_{k+2}' &= I_{k+2}' \oplus \epsilon_k && \text{(decoded bit error)} \\ o_{k+3}' &= Q_{k+3}' \oplus Q_{k+1}' && \text{(correct decoded bit)} \\ &\vdots \end{aligned}$$

which results in two decoded bit errors for a single detection error. The two bit errors are spread over three bit times. The same is true when Q_{k+1}' is in error.

When two consecutive detection errors occur, such as I_k' and Q_{k+1}', a total of four decoded bit errors occur. If two detection errors occur with a correct bit in between, such as errors in I_k' and I_{k+2}' or Q_{k_1}' and Q_{k+3}', only two decoded bit errors will occur and will be spread over *five* bit times.

The net result for the double differential encoder is the same as for the ARTM differential encoder. Both result in two bit errors for a single detection error. Both have a certain case where two closely-spaced detection errors result in only two bit errors. The rest of the time, both produce four bit errors for two detection errors, six bit errors for three detection errors, etc.

DECODING OF MULTIPLEXED CODE SYMBOLS

Since the original data came from one sequence $\boldsymbol{a}$, the separate decoded output sequences $\boldsymbol{e}'$ and $\boldsymbol{o}'$ in the demodulator are typically multiplexed into a single sequence $\boldsymbol{a}'$. One can also attempt to multiplex $\boldsymbol{I}'$ and $\boldsymbol{Q}'$ into a single coded sequence $\boldsymbol{c}'$. For the ARTM differential decoder (5), it was shown in [1] that multiplexing the coded bits *before* differential decoding does not work; this is because the post-multiplex decoder is shown to be recursive and a total of three unknown initial values can be identified. Since the decoder is recursive, the influence of the unknown initial values persists for the duration of the decoding process and the entire decoded data set is affected.

For the double differential decoder, the post-multiplex decoder is given by (7). From startup, the decoding process is

$$||a_0'|| = c_0' \oplus \langle c_{-2}' \rangle$$
$$||a_1'|| = c_1' \oplus \langle c_{-1}' \rangle$$
$$a_2' = c_2' \oplus c_0'$$
$$\vdots$$

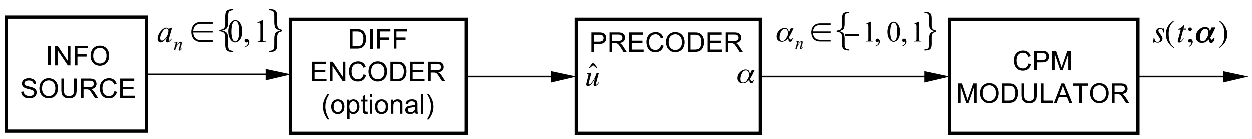

Figure 1: Block diagram of the CPM Model for OQPSK.

where it is clear that the post-multiplex decoder is not recursive and the influence of the two unknown initial values disappears after the first two output bits. This is a distinct difference between the double differential encoder and the ARTM differential encoder and is advantageous from a system architecture point-of-view.

BINARY-TO-TERNARY SYMBOL MAPPING

The CPM equivalent model is convenient for conceptualizing OQPSK and its SOQPSK relatives. This model is shown in Figure 1. The required elements in this model are the information source, the binary-to-ternary *precoder*, and the CPM modulator; the differential encoder is optional. The input to the precoder is the generic antipodal bit $\hat{u}_n \in \{\pm 1\}$ and the output is the ternary symbol $\alpha_n\{-1, 0, 1\}$; the two are related by the curious function [3]

$$\alpha_n = \frac{1}{2}(-1)^{n+1}\hat{u}_{n-1}\left(\hat{u}_n - \hat{u}_{n-2}\right). \tag{8}$$

The current symbol α_n is a function of the three most recent input bits, $\hat{u}_n$, $\hat{u}_{n-1}$, and $\hat{u}_{n-2}$, and also depends on whether n is even or odd. Therefore, the precoder can be described by an 8-state trellis with 16 branches, or a 4-state *time-varying* trellis with 8 branches and different sections for n-even and n-odd. More information on the trellis is given shortly.

It is easily verified that α_n assumes values from a ternary alphabet over time; however, there are some constraints on the binary-to-ternary relationship [3]:

1. While α_n is viewed as being *ternary*, in any given symbol interval α_n is actually drawn from one of two *binary* alphabets, $\{0, +1\}$ or $\{0, -1\}$.

2. When $\alpha_n = 0$, the binary alphabet for α_{n+1} switches from the one used for α_n, when $\alpha_n \neq 0$ the binary alphabet for α_{n+1} does not change.

3. A value of $\alpha_n = +1$ cannot be followed by $\alpha_{n+1} = -1$, and vice versa (this is implied by the previous constraint).

The last (redundant) constraint is often quoted, but it is not as strict as the second constraint and has led to confusion in the past.[1]

[1]For some applications, the answer to a question such as the following is important: How many distinct length-3 sequences can be generated by (8)? To answer this question, one can generate the $3^3 = 27$ possible *unconstrained* ternary sequences and then discard the 10 sequences that have a $+1$ followed by a -1 and vice versa. This method of counting was used, for example, in [2]. However, among the 17 sequences that remain are $[-1, 0, -1]$ and $[+1, 0, +1]$, which violate the "switch alphabets after $\alpha_n = 0$" constraint. Therefore, there are only 15 possible length-3 constrained ternary sequences. The counting error grows sharply as the sequence length increases. At first glance, this may seem insignificant, but this counting discrepancy is the very reason [4] for the 3 dB performance loss of the detection scheme in [2].

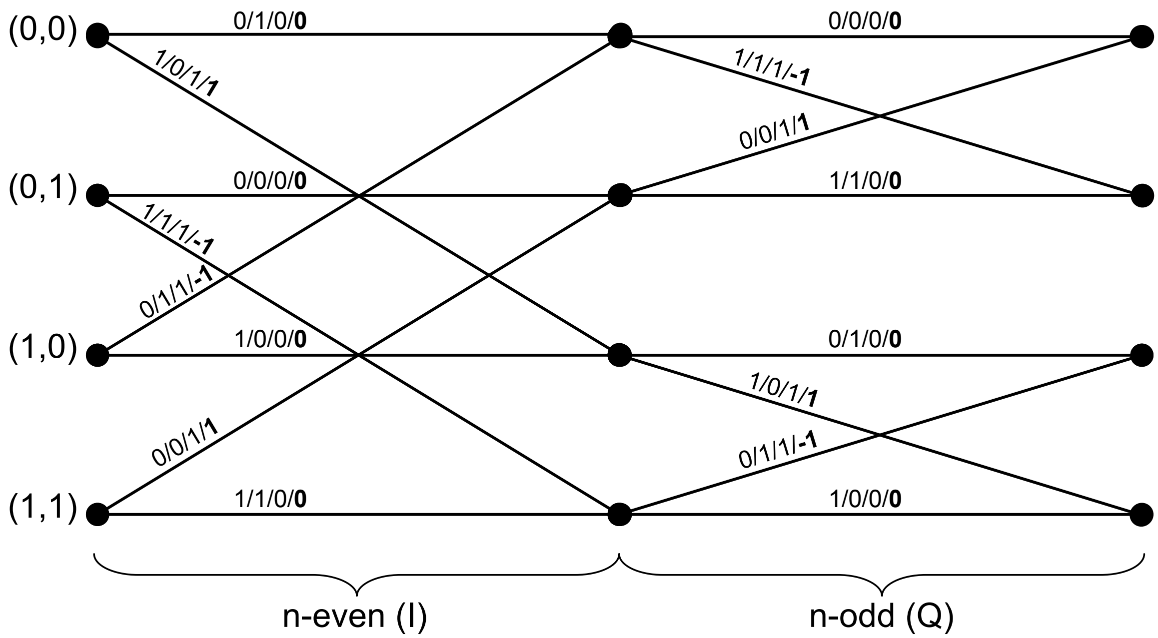

Figure 2: Four-state time-varying trellis of the CPM model for OQPSK. Each branch is labeled with 4 values: a_n [no differential encoder]/a_n [ARTM differential encoder]/ a_n [double differential encoder]/ α_n [in bold].

The precoder in (8) can be combined with the double differential encoder in (3) to produce a "double differential precoder." The steps are given in the Appendix and the resulting expression is

$$\alpha_n = (-1)^n a_n \hat{c}_{n-1} \hat{c}_{n-2}, \qquad a_n \in \{0,1\}, \quad \hat{c}_n \in \{\pm 1\}, \tag{9}$$

which is a function of the most recent *information bit*, a_n, the two previous differential (antipodal) bits, $\hat{c}_{n-1}$ and $\hat{c}_{n-2}$, and n-even/n-odd. Therefore, it has the same trellis structure as before but with different state variables. With further manipulations, (9) can be reduced to

$$\alpha_n = (-1)^{S_n} a_n, \qquad a_n, S_n \in \{0,1\}, \tag{10}$$

where

$$S_{n+1} = \big(S_n + \alpha_n + 1\big) \bmod 2. \tag{11}$$

A detailed derivation of (10) is also given in the Appendix. All of the binary-to-ternary constraints are clearly visible in (10) and (11); since the current bit a_n is drawn from a binary alphabet, α_n is also drawn from a binary alphabet whose "sign" is controlled by the *sign state* S_n; the switching rule for the binary alphabets is "switch on $\alpha_n = 0$," which is exactly how (11) works. Furthermore, an elegant side effect of the double differential precoder is that the *original information bits* are easily identified in the ternary symbol sequence: $a_n = 0$ always maps to $\alpha_n = 0$ and $a_n = 1$ always maps to $\alpha_n = \pm 1$. Therefore, the "change phase on 1" rule is clearly visible in the differential precoder.

The 4-state time-varying trellis of the precoder is shown in Figure 2. The trellis simultaneously describes *three* different system configurations:

- *No differential encoder*. Here the input bit is a_n and the state variable is the ordered pair (a_{n-2}, a_{n-1}) for n-even and (a_{n-1}, a_{n-2}) for n-odd.
- *ARTM differential encoder*. Here the input bit is a_n and the state variable is the ordered pair (I_{n-2}, Q_{n-1}) for n-even and (I_{n-1}, Q_{n-2}) for n-odd.

- *Double differential encoder*.[2] Here the input bit is a_n and the state variable is the ordered pair (I_{n-2}, Q_{n-1}) for n-even and (I_{n-1}, Q_{n-2}) for n-odd, which can also be expressed as (c_{n-2}, c_{n-1}) and (c_{n-1}, c_{n-2}) for n-even and n-odd, respectively.

The established convention is that the inphase (I) bit is always the most-significant and the quadrature (Q) bit is always least-significant [3]. Each branch in Figure 2 is labeled with 4 values: a_n [no differential encoder]/a_n [ARTM differential encoder]/ a_n [double differential encoder]/ α_n [in bold]. In all cases, the original information sequence $\boldsymbol{a}$ serves as the input; however, in each case the mapping between a_n and α_n is different.

The wisdom of the ordering convention for the state variable is evidenced by the fact that each branch has a fixed α_n, regardless of the system configuration. If one looks at things from the point-of-view of the unit circle this makes perfect sense. The OQPSK signal has 4 phase states, with certain rules for when each element of the state can change. Regardless of how these states map to the actual information sequence $\boldsymbol{a}$, the relationship between state transitions (branches) and phase changes (α_n) is fixed.

OTHER ADVANTAGES/APPLICATIONS OF THE DOUBLE DIFFERENTIAL PRECODER

In addition to resolving the 4-phase ambiguity that results in most carrier phase tracking loops, a differential (recursive) relationship is needed in such applications as serially concatenated coding with iterative detection. In this application, differentially encoded (S)OQPSK serves as the inner code in a concatenated coding scheme. Figure 3 shows simulated bit error rate (BER) results for the optimal rate-1/2 (7,5) convolutional code when it is concatenated with SOQPSK-TG and iteratively demodulated and decoded. Three SOQPSK systems are used: no differential encoder, ARTM differential encoder, and double differential encoder. The results with no differential encoder are included to highlight the well-known fact that large interleaver gains are only achieved when the inner code is recursive [6]. The advantage that the double differential encoder has over the ARTM differential encoder may be slight (0.07 dB), but it does represent another benefit of this type of differential encoding. The BER curve for the double differential encoder was publishhed previously in [5].

CONCLUSIONS

We have proposed another type of differential encoding, known as *double differential encoding*, for use with ARTM Tier I (and other OQPSK-type) modulations. We have shown that this differential encoding scheme can resolve the 4-phase ambiguity that results from most carrier tracking loops. We have also characterized the initial condition and error propagation properties of this encoder. Unlike the differential encoder currently recommended by IRIG-106, we have shown that the order of operations between encoding/decoding and even-odd multiplexing/demultiplexing are interchangeable for the proposed scheme. Furthermore, we have shown that the double differential encoder results in a simple and elegant relationship in the binary-to-ternary mapping that is used in the CPM model for OQPSK. Finally, we have demonstrated that the double differential encoder outperforms the existing IRIG-106 differential encoder in serially concatenated coded systems with iterative detection.

[2] Based on (10), the double differential precoder itself can be represented with a *two* state trellis. However, such a trellis is not suitable for detection since it does not account for all of the memory elements in Figure 1; in particular, the memory in the CPM modulator is missing. As it turns out, the 4-state trellis in Figure 2 simultaneously accounts for the memory in the precoder *and* a full-response CPM modulator [5].

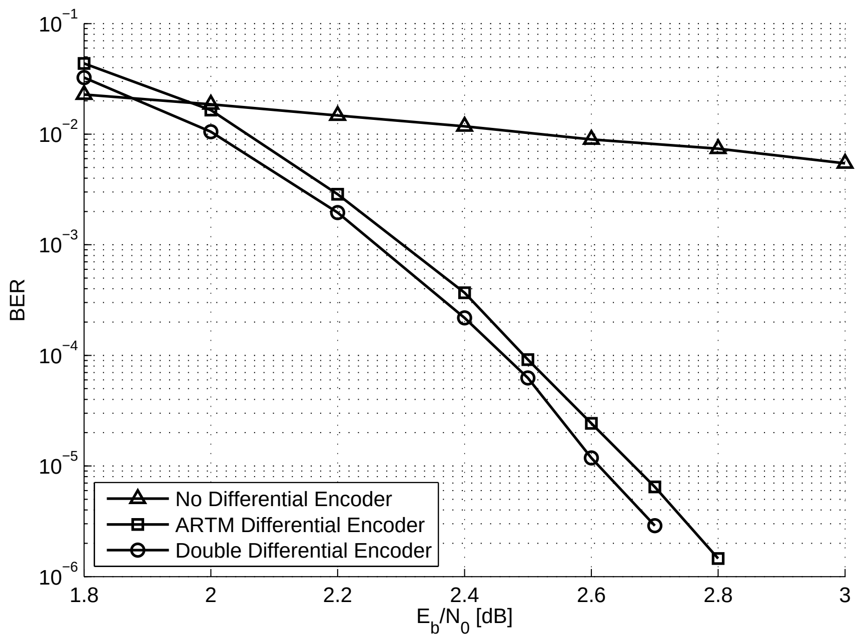

Figure 3: Performance of serially concatenated SOQPSK-TG with various differential encoding configurations.

APPENDIX

To begin, we modify some of the above expressions for binary antipodal variables. A generic Boolean variable $u_n \in \{0, 1\}$ can be converted into an antipodal variable $\hat{u}_n \in \{\pm 1\}$ by

$$\hat{u}_n = 2u_n - 1 \tag{12}$$

or

$$-\hat{u}_n = (-1)^{u_n}. \tag{13}$$

Also, the double differential encoder in (3) can be expressed in terms of antipodal bits with

$$\hat{c}_n = (-\hat{a}_n) \cdot \hat{c}_{n-2}, \qquad \hat{a}_n, \hat{c}_n \in \{\pm 1\} \tag{14}$$

where, as before, the "change phase on 1" rule is in effect. We now derive (9) by substituting $\hat{c}_n$ for $\hat{u}_n$ in (8), which yields

$$\begin{aligned} \alpha_n &= \frac{1}{2}(-1)^{n+1}\hat{c}_{n-1}(\underbrace{\hat{c}_n}_{-\hat{a}_n\hat{c}_{n-2}} - \hat{c}_{n-2}) \\ &= (-1)^{n+1}\underbrace{\frac{-\hat{a}_n - 1}{2}}_{-a_n}\hat{c}_{n-1}\hat{c}_{n-2} \\ &= (-1)^n a_n \hat{c}_{n-1}\hat{c}_{n-2} \end{aligned} \tag{15}$$

where (12) and (14) are also applied. These steps give the desired result.

Next, we show how (15) simplifies to (10). Substituting (14) with itself recursively gives

$$\begin{aligned}\hat{c}_{n-1}\hat{c}_{n-2} &= \prod_{i=-\infty}^{n-1} (-\hat{a}_i) \\ &= \prod_{i=-\infty}^{n-1} (-1)^{a_i} \\ &= (-1)^{\sum_{i=-\infty}^{n-1} a_i} \end{aligned} \tag{16}$$

where (13) is also applied. Inserting (16) into (15) results in

$$\begin{aligned}\alpha_n =& (-1)^{n+\sum_{i=-\infty}^{n-1} a_i} a_n \\ =& (-1)^{S_n} a_n\end{aligned}$$

where the *sign state* S_n is defined by either of the recursions

$$\begin{aligned}S_{n+1} &= \big(S_n + a_n + 1\big) \bmod 2 \\ &= \big(S_n + \alpha_n + 1\big) \bmod 2.\end{aligned}$$

These recursions are equivalent because “switch alphabets on $a_n = 0$” is the same as “switch alphabets on $\alpha_n = 0$” in the case of the double differential encoder.

REFERENCES

[1] *Appendix M, IRIG Standard 106-04: Telemetry Standards*, Range Commanders Council Telemetry Group, Range Commanders Council, White Sands Missile Range, New Mexico, 2004. (Available on-line at http://www.ntia.doc.gov/osmhome/106.pdf).

[2] M. K. Simon, “Multiple-bit differential detection of offset QPSK,” *IEEE Transactions on Communications*, vol. 51, pp. 1004–1011, June 2003.

[3] M. K. Simon, *Bandwidth-Efficient Digital Modulation With Application to Deep-Space Communication*. New York: Wiley, 2003.

[4] E. Perrins, R. Schober, M. Rice, and M. K. Simon, “Multiple bit differential detection of shaped offset QPSK.” To be published in *IEEE Transactions on Communications*.

[5] E. Perrins and M. Rice, “Serially concatenated ARTM Tier I waveforms with iterative detection,” in *Proceedings of the International Telemetering Conference*, October 2005.

[6] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, “Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding,” *IEEE Transactions on Information Theory*, vol. 44, pp. 909–926, May 1998.