ASYNCHRONOUS TIMING RECOVERY IN DSP BASED PSK MODEMS

B. KOBLENTS AND P. J. McLANE

DEPARTMENT OF ELECTRICAL ENGINEERING,
QUEEN’S UNIVERSITY, KINGSTON, ONTARIO
K7L 3N6

Abstract—While conventional analog modems employ some kind of clock wave regenerator circuit for synchronous timing recovery, in sampled modem receivers timing is recovered asynchronously to the incoming data stream, with no adjustment being made to the input sampling rate. All timing correction is accomplished by digital operations on the sampled data stream. A good timing error measurement algorithm is a zero crossing tracker proposed by Gardner. Digital, speech rate (2400 – 4800 bps) M-PSK modem receivers employing Gardner’s zero crossing tracker can achieve BER performance very close to theoretical maximums on the AWGN channel. Both rectangular pulse shaped, and Nyquist pulse shaped modem systems with excess bandwidth factors ranging from 100% to 60% are considered.

I. INTRODUCTION

We consider two low speed, fully digital modem families which have been implemented as real time firmware on Texas Instruments Digital Signal Processors. The first consists two 2400 bps rectangular pulse shape DPSK modems, one with uncoded D-QPSK signalling, and the other with interleaved and trellis coded 8-DPSK. The second family consists of a series of 4800 bps uncoded DQPSK modems with Raised Cosine pulse shaping with excess bandwidth factors ($\alpha$) ranging from 100% to 60%. Both modem families employ asynchronous timing recovery, whereby the modem receiver recovers timing phase from the oversampled input signal, leaving it's input A/D converter free running. Such timing recovery techniques for digitally implemented modems are discussed in [1]. Zero crossing type timing error detectors of the type described by Gardner [1][2], are used in both modem families to complete the timing recovery tracking loops.

II. ASYNCHRONOUS TIMING RECOVERY

In traditional analog modems, timing recovery is accomplished by regenerating a local clock wave to be synchronous with the data rate, and using this local clock wave to adjust the frequency and phase of the input A/D converters to be synchronous with the incoming data [1]. In a fully digital implementation, on the other hand, the input sampling rate can be asynchronous to the incoming data stream. In most cases, the input sampling rate is an integer multiple of the nominal symbol rate, but the receiver sampler is free running and never explicitly locked to the transmitter baud clock [3]. Timing correction is accomplished by aligning the decimator strobes shown in Figure 1, with the pulse peaks out of the matched filters. Such a concept is grossly illustrated in Figure 2. In that figure, the transmitter D/A clock is assumed to have a slightly lower frequency than the receiver A/D clock. Note how the shorter symbol epochs of the receiver approach and then depart from the transmitter symbol epochs, but by choosing appropriate receiver samples, the symbol epochs can be kept in synchronization. This approach requires that the receiver occasionally add samples. Conversely, if the receiver clock were the slower one, the receiver would occasionally drop samples [4]. If the oversampling rate is too low to permit the adding or dropping of samples with each correction, interpolation between samples is possible [1][4]. The asynchronous timing recovery method described above can
be considered equivalent to implementing a variable delay line in each of the I and Q rails of the receiver [1][4]. It permits the receiver A/D to be completely free running, with no corrections made to its sampling phase or frequency.

III. TIMING ERROR DETECTION: MAXIMUM LIKELIHOOD METHODS Vs. ZERO CROSSING TRACKERS

Timing error tracker designs suitable for bandlimited signals, and based on Maximum Likelihood Estimation (MLE) principles are described in various references [1][5][6][7]. The basic approach is shown in Figure 3 and consists of a matched filter and a differentiator. At the pulse peaks, the matched filter response is a maximum, and its derivative is zero. The tracker characteristic therefore has a null at the optimum decision time. The tracker expression is as follows

\[ u(n) = x'(n) \Delta(n) \]  

where \( u(n) \) is the tracker output, \( x'(n) \) is the derivative of the matched filter output, and \( \Delta(n) \) is the estimate of the data symbol, all for the \( n \)th symbol time.

A complication of the tracker in Figure 3 is the digital differentiator. In a strict sense, a sampled signal is discontinuous, and hence can not have a derivative. It is possible to reconstruct the derivative of a smoothly continuous bandlimited waveform from its samples. The derivative can then be expressed as the following infinite sum [1]

\[ x'(m) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} x(k) \left(-\frac{1}{2}\right)^m \Delta m - k \]  

Obviously, implementing a practical digital differentiator requires a truncation of the infinite sum. Truncations are not guaranteed to be well behaved, and can be causes of self noise in a synchronizer [1]. It is possible to obtain the derivative of the matched filter response not by differentiating the output of the matched filter, but by passing the signal through a separate filter, whose impulse response is the derivative of the matched filter response. That can yield a perfect derivative, but at the cost of doubling the filtering requirement. But even for a perfect derivative, there is still self noise due to ISI, since at the peak of the pulse of interest, all other pulses in the sequence have zero crossings, whose derivatives are non-zero. There are methods for overcoming this self noise, which involve designing modified derivative matched filters [5], or implementing a four point differentiator modified to cancel self noise from the adjacent terms [1][7]. All these approaches involve added complexity, and require that a minimum of four samples per symbol be processed in the timing error detector [1][7]. In addition, MLE based algorithms fail completely for rectangular pulse shaped systems, since such systems do not have smoothly continuous and bandlimited pulse shapes.

In [2], Gardner introduces a decision directed zero crossing tracker algorithm of the following form

\[ u(n) = x(n-1) \left( SGN(x(n)) - SGN(x(n-1)) \right) \]  

In the absence of timing error \( x(n-1) \) is a zero crossing, if there is a data transition between the \( n \)th and \( (n-1) \)st symbol. The terms \( SGN(x(n)) \) and \( SGN(x(n-1)) \) have two purposes. The first is to provide polarity information to the error signal, and the second is to eliminate the error signal if there is no data transition. For signals such as BPSK, QPSK, DBPSK and DQPSK, which have equal height pulse shapes for any data transition, the DD zero crossing tracker is entirely free of self noise for \( \alpha \) of 100%. For \( \alpha \) less than 100%, there is some self noise, getting worse with decreasing excess bandwidth.

Self noise is introduced if (3) is applied to modulation formats which utilize different height pulse shapes, as is true of M-PSK and M-DPSK with \( M > 4 \). But (3) can be adapted to Non-Data-Aided (NDA) operation, by replacing the terms \( SGN(x(n)) \) and \( SGN(x(n-1)) \) by \( x(n) \) and \( x(n-1) \), and by applying the
algorithm to both the in-phase and quadrature arms of the receiver, resulting in

\[ u(n) = x_i(n-\frac{1}{2})[x_i(n) - x_i(n-1)] + x_q(n-\frac{1}{2})[x_q(n) - x_q(n-1)] . \]  

(4)

This NDA variant of Gardner's zero crossing tracker can be shown to be free of self noise in the case of \( \alpha = 100\% \) for any PSK data transition, and not just those with equal pulse heights [4]. Both variants of Gardner's zero crossing tracker can be adapted to operate with rectangular pulse shapes [8].

It has been previously reported [9] that zero crossing trackers can actually outperform MLE based timing synchronizers in some applications. Despite not being theoretically optimum, Gardner's zero crossing tracker can provide good performance, at a complexity of only two samples per symbol from each of the I and Q rails of the receiver. We utilize it in our Nyquist pulse shaped modems, and an adaptation of it in our rectangular pulse shaped systems.

IV. STUDIES WITH RECTANGULAR PULSE
SHAPED MODEMS

We first provide a description of our family of rectangular pulse shaped systems, consisting of a 2400 bps uncoded DQPSK modem, and a trellis coded, convolutionally interleaved, 2400 bps 8-DPSK modem. Both systems have been implemented and tested on the TMS320 series of digital signal processors, and previously described in [10]. Figure 4 shows the block diagram representations of the 8-DPSK trellis coded transmitter and receiver, each implemented on a separate DSP chip. A third chip is used to implement the convolutional de-interleaver and Viterbi decoder. The design of the transmitter, receiver, and channel code are based on [8][11][12][13]. The DQPSK modem is similar, except that in place of a natural phase mapper and convolutional interleaver in the transmitter, there is a Grey code mapper, and final bit decisions are made in the receiver, so there is no need for a third DSP chip. The transmitter is based on 256 entry sine table [14]. Thirty two samples per symbol are employed, with eight samples per IF carrier cycle, resulting in four cycles of the IF carrier per symbol. Such a high carrier to data ratio is required to avoid aliasing problems at the demodulators. The differential phase modulator is based on the equation

\[ \theta_{k+1,m} = (\theta_{k,m} + \lambda_m) \mod 2\pi , \]  

(5)

where \( \lambda_m \) is the information phase output from the phase mapper, while \( \theta_{k+1,m} \) is the modulation phase for the \((k+1)st\) symbol. Phase modulation is achieved by using the output of (5) to initialize the eight sample per cycle carrier from one of eight (8-DPSK) or four (D-QPSK) starting positions. To achieve 2400 bps operation, a sampling rate of 38.4 KHz is required, leading to a symbol rate of 1200 baud and an IF carrier frequency of 4800 Hz.

The A/D converter at the input of the receiver also operates at 38.4 KHz, and is free running. The input samples are demodulated in quadrature by a numerical oscillator similar to the one in the transmitter, except that this time two pointers into the sine table deliver sampled sinusoids exactly 90° apart in phase. The matched filter for a rectangular pulse shape is an integrator, so the I and Q demodulated data streams are fed to the I and Q rail data integrators, as shown in Figure 4. The integrators are implemented as simple accumulators which run for a symbol duration (32 sample intervals) and then are reset, dumping their outputs to the timing and data detection blocks. The data detection, Doppler correction, interleaving and channel coding are described in [4][10][11][12][13][15]. They are beyond the scope of this paper and will not be discussed further.

The timing error detector used is a variation of Gardner's zero crossing tracker as described by Henley [8]. It uses mid-pulse to mid-pulse integrators, herein referred to as synchronization integrators, as shown in Figure 4. The scheme is based on the idea that a mid-pulse to mid-pulse integration should be
zero through an even pulse height data transition. The form of the timing error detector is identical to the Gardner NDA tracker of equation (4), with the synchronization integrators providing the \( x(n-N) \) terms, while the \( x(n) \) and \( x(n-1) \) terms are provided by the data integrators.

In the absence of timing offset, the double frequency terms produced in the mixers are completely averaged out in the integrators. With a timing offset, however, these double frequency terms are not fully averaged out, and can cause false error readings on data transitions not involving 180° phase shifts [4]. The timing jitter caused by the double frequency distortion adversely affect the BER performance of the rectangular pulse shape modems. By taking the sum of the absolute values of the synchronization integrators, and only proceeding to make timing measurements if the sum does not exceed a certain threshold as in (6) below, the timing jitter could be significantly reduced.

\[
\text{ABSUM} = \text{ABS}(x(n-\frac{1}{2})) + \text{ABS}(x(n-\frac{3}{2}))
\]

if \( \text{ABSUM} \leq \text{THRESHOLD} \) then make timing measurements

A summary of simulation results averaged over the SNR range of interest (9 to 20 dB, \( 2E_b/N_0 \)) is shown in Table 1. A threshold level of 60% of the maximum data accumulator value causes timing decisions to be skipped for 60% of the received symbols, and reduces the number of decision on pulses vulnerable to double frequency interference by 67%. Making the threshold tighter causes the algorithm to miss larger numbers of corrections on the desired 180° phase shifts. Similar behaviour is observed when the thresholding scheme is added to the DSP system. Timing jitter is reduced with an attendant improvement in performance.

The timing error signal is then filtered with a 50 Hz first order IIR loop filter, and applied to the timing correction mechanism. If the timing error signal exceeds a threshold for timing correction, then the receiver either drops or adds a sample to the next symbol's integration period. Since the system is highly oversampled, there is no need for interpolation between sampling points [1]. Note timing is corrected without altering the frequency or phase of the receiver input A/D converter, which remains free running.

The rectangular pulse shape modems have been tested over a Gaussian noise channel. Our first experimental result is for the 2400 bps uncoded DQPSK system. A 100 symbol setup sequence was used to establish receiver timing. After that a long sequence of pseudo random symbols was transmitted and the errors counted. The results of the experiment are given in Figure 5, which shows three curves: the performance of the system before any changes are introduced, the performance with the thresholding scheme as in equation (6), and a computer simulation with perfect timing which serves as the ideal case. Note that the thresholding scheme reduces implementation losses by roughly half a dB, to roughly 1 dB in \( E_b/N_0 \). Experimental results for the trellis coded 8-DPSK modem incorporating thresholding of equation (6) and loop filtering are shown in Figure 6. With proper choice of loop filter bandwidth, implementation losses are reduced to a small fraction of a dB.

V. STUDIES WITH RAISED COSINE PULSE SHAPED MODEMS

The MSAT signal standard will use Raised Cosine (RC) pulse shapes with \( \alpha = 60\% \) and DQPSK modulation with conventional convolutional coding. As such, a family of 4800 bps modems has been developed with \( \alpha \) ranging from 100\% to 60\%. The transmitter and receiver block diagrams are shown in Figure 7. As in the rectangular pulse shaped systems Gardner's NDA zero crossing tracker is employed for timing error measurement, except that this time no separate synchronization filters are required, as the mid points of a symmetric pulse shaped response are ISI free zero crossings, at least in the \( \alpha = 100\% \) case. For the excess bandwidth range of study (100\% to 60\%) ISI induced self noise can be controlled sufficiently with properly tight loop filtering, so that performance is
not significantly degraded. The thresholding scheme employed in the rectangular pulse shaped modems is not needed here, since the data filters attenuate the double frequency terms from the mixers.

The RC pulse shaped modem employs eight samples per baud. The IF carriers are again generated by table lookup NCO's with each sample incrementing the sine table pointers by 45°, resulting in one IF carrier cycle per baud, and an IF carrier rate of 2400 Hz. With fewer samples per baud, a timing correction mechanism consisting solely of adding or dropping samples from the next symbols timing epoch may be too coarse under some conditions. A second timing correction scheme has therefore been developed for the RC pulse shaped modems. This is the polyphase filter interpolated timing recovery scheme, and it's operation is illustrated in Figure 8. Upon startup, the receiver is initialized to utilize data filter $F_1$, which assumes zero timing offset. A timing offset produces a mismatch between the incoming signal and the receiver data filters. As a timing error large enough to warrant correction is detected, the receiver can switch coefficients of the data filter to those of $F_2$, so that the receiver data filters are once more matched to the incoming signal. As the timing error grows, filters $F_3$ and $F_4$ are loaded. Finally, the filters are reset to $F_3$ and a sample added to the next symbol epoch. If the timing drift is in the other direction, a sample is dropped, and the filter sequence $F_4$, $F_3$, $F_2$, $F_1$ is employed. Because the new filter responses lie in between the samples of the original response, they can be obtained by interpolation between the it's samples. The interpolation can be made as fine grained as desired, by simply implementing the receiver filters oversampled to the desired interpolation factor, $M$, and taking successive $M$th taps out of the oversampled response. These turn out to be the branches of the $M$-branch polyphase filter decomposition of the oversampled interpolating filter [1].

Filter coefficient interpolation timing correction has been implemented in simulation. Figure 9 shows timing error in the noise free simulation case without interpolation. Figure 10 shows the same perturbation sequence, but this time with filter coefficient interpolation. Note that timing error is kept much smaller if interpolation is used. Results for the initial DSP implementation employing only the coarser timing correction scheme lacking interpolation are good enough not to require any additional work. Apparently with eight samples per symbol, the impulse response samples are closely enough spaced not to require any interpolation.

Performance testing has been carried out on the AWGN channel. Figure 11 shows BER results for the 60% excess bandwidth case. Results for the 70%, 80%, 90% and 100% cases are similar. In all cases, implementation losses are kept well under 1 dB. The filter coefficient interpolation timing correction scheme may be incorporated in future extensions of the RC pulse shaped systems with lower oversampling rates and less excess bandwidth. For example McLane, Tay and Choy have incorporated it into their 40% excess bandwidth coherent BPSK modem [16], which has four samples per baud.

VI. CONCLUSIONS

Fully digital asynchronous timing recovery has been demonstrated to give excellent results in laboratory experiments on speech rate (2400 and 4800 bps) modems with both rectangular and RC pulse shaping. Gardner's zero crossing tracker timing error detector has been applied to rectangular and RC pulse shaped speech rate modems with good results on the additive noise channel. The NDA form of the zero crossing tracker has been found to be immune to pattern noise for any data phase transition, making it ideal for timing error detection in M-PSK and M-DPSK systems with $M > 4$. In RC pulse shaped systems with $\alpha$ down to 60% ISI induced self noise is sufficiently controlled by loop filtering. For systems with a limited number of samples per symbol, polyphase filter interpolation can be used to improve performance.
ACKNOWLEDGEMENTS

Michel Belanger of Queen's University is thanked for his help in the implementation and testing of the modems described in this report.

REFERENCES


Figure 1. Typical digital PSK structure.

ASYNCHRONOUS TIMING CORRECTION
Rx sampling rate faster than Tx. Rx adds samples to keep up.

Figure 2. Asynchronous timing correction.

Table 1. Simulation results for synch. integrator holding scheme.

<table>
<thead>
<tr>
<th>HIGH THRESHOLD</th>
<th>NUMBER OUT OF RANGE</th>
<th>NUMBER OF &quot;BAD&quot; TRANITIONS GUESSED AT 180 degrees phase shifts</th>
<th>NUMBER OF 180 DEGREE PHASE SHIFTS MISSED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(as percentage of</td>
<td>(as percentage of these)</td>
<td>(as percentage of these)</td>
</tr>
<tr>
<td></td>
<td>Tx data integrator</td>
<td>values)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>threshold)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>309</td>
<td>0</td>
<td>19</td>
<td>0</td>
</tr>
<tr>
<td>119</td>
<td>16</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>107</td>
<td>25</td>
<td>45</td>
<td>2</td>
</tr>
<tr>
<td>96</td>
<td>41</td>
<td>30</td>
<td>4</td>
</tr>
<tr>
<td>84</td>
<td>48</td>
<td>43</td>
<td>4</td>
</tr>
<tr>
<td>78</td>
<td>40</td>
<td>42</td>
<td>4</td>
</tr>
<tr>
<td>72</td>
<td>53</td>
<td>28</td>
<td>4</td>
</tr>
<tr>
<td>68</td>
<td>58</td>
<td>23</td>
<td>4</td>
</tr>
<tr>
<td>60</td>
<td>60</td>
<td>29</td>
<td>4</td>
</tr>
<tr>
<td>48</td>
<td>75</td>
<td>16</td>
<td>28</td>
</tr>
<tr>
<td>36</td>
<td>79</td>
<td>4</td>
<td>26</td>
</tr>
</tbody>
</table>
Figure 4. Transmitter and receiver block diagrams for rectangular pulse shaped trellis coded 8-DPSK modem.

Figure 5. BER performance of rectangular pulse shaped DQPSK modem.

Figure 6. Performance of rectangular pulse shaped trellis coded 8-DPSK modem.
Figure 7. Raised Cosine pulse shaped modem transmitter and receiver block diagrams.

Figure 8. Polyphase Filter interpolated timing recovery.
Figure 9. Timing error signal in simulation without interpolation.

Figure 10. Timing error signal in simulation with interpolation, and the same timing perturbation sequence as in Figure 9.

Figure 11. BER performance of Raised Cosine pulse shaped modem system with 60% excess bandwidth.