NEW OPTIMIZATIONS FOR CARRIER SYNCHRONIZATION IN SINGLE CARRIER SYSTEMS

Kiran Gunnam, Kanu Chadha, Mark Yeary*

Schlumberger, Sugarland, TX 77478, USA *School of ECE, University of Oklahoma, Norman, OK 73019, USA

ABSTRACT

Single carrier digital communication systems such as M-PSK and M-QAM require the use of carrier synchronization in frequency and phase. In this paper, we present new optimizations that simplify the core operations involved in the synchronization process, namely the implementation of the numerically controlled oscillator (NCO) and the phase detector. These mathematical optimizations are generic in nature and can be applied to various DSP implementations such as fixed-point and floating-point implementations. One of the benefits of these optimizations is that look up tables for calculating typical trigonometry functions such as the complex exponential and arc tangent are not required. Results of some example implementations on the Texas Instruments fixed point DSP TMS320VC54x and the Analog Devices floating point DSP SHARC2106x are reported in the results section of this paper.

keywords: Phase Locked Loop (PLL), Carrier Synchronization, Arc Tangent, Numerically Controlled Oscillator (NCO), CORDIC.

1. INTRODUCTION

In conventional digital receiver architectures, the sampling of the received symbols is synchronized with the incoming data symbols using analog phase locked loop techniques. However, in modern architectures, the received signal is sampled by a fixed analog to digital converter (ADC) clock, which is not synchronous with the clock of the transmitter's digital to analog converter (DAC). The timing and phase tracking are usually performed by interpolation and rotation of symbols in the complex base band domain. The complex rotation is achieved using a phase detector and a Numerically Controlled Oscillator (NCO).

This paper presents a simplified realization for the implementation of the phase detector and NCO for single carrier systems. The rest of the paper is organized as follows: Section 2 explains typical base band coherent receiver architecture; Section 3 presents the new method to implement an NCO and the method to implement the phase detector; and finally, Section 4 gives the performance comparison and sample implementation results.

2. BASE-BAND COHERENT RECEIVER

The front-end receiver processing involves analog to digital conversion, down conversion and symbol shaping as shown in Fig. 1. An analog band pass filter precedes the Analog to Digital Converter (ADC) to prevent aliasing. A stable crystal oscillator (XTAL) generated the sampling clock of ADC (F_s). The ADC is followed by a digital down converter. Generally F_s is chosen to be $4 \cdot F_c$ where F_c is the carrier frequency to have zero-IF at the base-band. The mixing is done using sequences of $[1 \ 0 \ -1 \ 0]$ and $[0 \ 1 \ 0 \ -1]$ to obtain the in-phase (I) and out-of phase (Q) components. To reduce the matched filter order requirements, multi-rate processing is a standard technique and the filter is split into two or more decimating stages, which does include a square root raised cosine (SRRC) filter here, and see also [1].



Fig. 1. Typical Front End Processing for a Single Carrier System.

The base-band symbol processing is performed using a coherent receiver and a typical receiver is shown in Fig. 2. The coherent receiver operates on symbols over sampled by a factor of 2 or 4. Many blocks in receiver use decision-directed algorithms. Decision directed carrier recovery is one of the optimal methods to implement a digital PLL [2]. The decision device is known as the *slicer* and it makes a hard decision on the input symbols. The carrier acquisition block is not shown in the above figure and will be realized using standard techniques using a synchronization sequence that has suitable autocorrelation properties such as the Barker sequence. In packet-based systems, the sync sequence is transmitted for the beginning of each packet for the frame synchronization. In continuous transmission systems, sync sequence is inserted in transmitted symbol stream at regular intervals. An equalization block may be needed for channels characterized by fading, Doppler shift and other non-linear effects. This equalization block can be any of the standard methods Least Mean Squares (LMS), Recursive Least Squares (RLS) or Constant Modulus Algorithm (CMA) [2].

A re-sampler is used to estimate the value of the signal at the desired time instant by using interpolation techniques [2, 3] based on the timing offset calculated by the timing tracker. A phase



Fig. 2. Typical Coherent Receiver at the Base-band

tracker will track the carrier phase and consists of phase detector, a loop filter and a Numerically Controlled Oscillator (NCO). A phase detector will have the re-sampled symbol and the slicer output and gives the estimate of phase offset in the carrier due to impairments. A loop filter will be used to smooth the phase error estimates and control the behavior of phase tracking loop. The phase adjustment is done by multiplying the re-sampled signal by a complex phasor generated by the NCO from the estimated phase offset in the carrier. This estimated phase also drives the time tracker and consists of a timing estimator (such as Mueller & Muller or Gardner [3]) and a loop filter. The system equations are defined below. The resampler output, adjusted by the time tracker is given by

$$x(nT + \hat{\tau}_n) = x_i(nT + \hat{\tau}_n) + jx_i(nT + \hat{\tau}_n) \tag{1}$$

Here T is the baseband sampling time and is $T_s/4$ (i.e. symbols over sampled by a factor of 4). The variable T_s is defined as the symbol time interval and is $1/f_{sym}$, where f_{sym} is defined as the symbol frequency. This interpolation is based on delta derived from τ_n [3]. Thus, $e_T(n) = Re\{[x((n-1)T + \tau_{n-1} - x(nT + \tau_n)]x * ((n-1/2)T + \tau_n)\}$, where $\tau_{n+1} = \tau_n + K_1e_T(n) + K_2 \sum_{i=0}^n e_T(i)$. Here, K_1 and K_2 are the parameters of the loop filter designed for the desired time tracking against oscillabor drift and noise. By letting $\mu_n = \tau_n/T_s$, the interpolation coefficients can be determined by solving $h_0 = h_3 = 0.5\mu_n^2 - 0.5\mu_n$, $h_1 = -h_0 - \mu_n + 1$ and $h_2 = -h_0 + \mu_n$. Moving forward, the baseband adjustment for the phase rotation is described as $z1(n) = x(nT + \hat{\tau}_n)e^{j\phi(n)} = z1_i(n) + jz1_q(n)$, where

$$z1_{i}(n) = x_{i}(nT + \hat{\tau}_{n})\cos(\phi(n)) + x_{i}(nT + \hat{\tau}_{n})\sin(\phi(n))$$
(2)

$$z1_q(n) = x_i(nT + \hat{\tau}_n)\cos(\phi(n)) - x_i(nT + \hat{\tau}_n)\sin(\phi(n))$$
 (3)

The decision slicer quantizes the received vector to appropriate constellation points. For example, the QPSK slicer is nothing but a "method 1" sign operation, defined by: $\hat{s}(n) = \hat{s}_i(n) + j\hat{s}_q(n) = \operatorname{sign}(z1_i(n)) + j\operatorname{sign}(z1_q(n))$, where $\operatorname{sign}(t) = 1$ if $t \ge 0$ and -1 if t < 0. The phase detector is defined by $z(n) = x(nT + \hat{\tau}_n)e^{-j\phi(n)}\hat{s}^*(n)$ and $e_\phi(n) = \arg(z(n)) = \tan^{-1}(z_q(n)/z_i(n))$. The phase tracking loop is defined by $\phi(n) = \phi(n-1) + \phi_c(n-1)$ and $\phi_c(n) = K_3 e_\phi(n) + K_4 \sum_{i=0}^n e_\phi(j)$.

Here K_3 and K_4 are the parameters of the loop filter designed for desired phase tracking against oscillator drift and noise. The mapper produces the demodulated bit stream based on the slicer output.

3. PROPOSED OPTIMIZATIONS

3.1. Numerically Controlled Oscillator

The sine and cosine values need to be generated in the process of decoding a symbol. There are several methods to generate sine and cosine, such as Taylor series, look up table and CORDIC methods. CORDIC computation adopts only primitive arithmetic operations - algebraic addition and shift - instead of multiplication. However these methods are iterative in nature and are better suited for fully parallel hardware implementation [3,4]. The authors in [5] performs calculations in polar coordinates to simplify the phase tracking and still requires a CORDIC computation. However polynomial based methods are faster and map better on programmable DSP units that have dedicated single cycle MAC (Multiply and Accumulate) units. These methods usually require coordinate mapping and typically 6 MACs and a total of 12 MACs are required for both sine and cosine. A table look up combined with interpolation is faster, however, this method takes up precious resource in terms of on chip memory. In this paper, we present an optimization that is faster and requires no tables. This method requires only two MACs to compute the sine and cosine values. It is a reasonable assumption that the equalization block takes care of the changes due to fading and Doppler shift so that the phase tracker needs to compensate for phase noise due to oscillators and channel noise. The phase drift at the transmitter oscillator and receiver oscillator may contribute additive phase noise. It is assumed that the oscillators used in TX DAC and RX ADC have the same ppm for simplicity. For a typical value of 250 for the oscillator, $f_{carrier}/f_{symbol} = 16$ and $e_{\phi,max}(n) = 2\pi/125$. Figure 3 shows the input to the phase detector and its output which has components due to noise and oscillator drift. All the figures illustrated later are based on QPSK simulation for the above parameters and Eb/No is 8 dB (with AWGN noise). The results are valid for other Eb/No values and other single carrier systems.

In addition, noise will also introduce phase noise, and it is a general practice to design the loop filter to track the phase changes



Fig. 3. Phase Detector Input and Output.

up to half of the decision boundary for a symbol period. If the tracking loop is designed correctly and running smoothly, the phase correction needed for a symbol period is less than $2\pi/125$ most of the time. Since the magnitude of the required phase correction term $\phi_c(n)$ is limited to $e_{\phi,max}(n)$ most of the time, the following first order approximation for the sine and cosine is derived.

$$\begin{aligned} \sin(\phi(n)) &= \sin(\phi(n-1) + \phi_c(n-)) & (4) \\ &= \sin(\phi(n-1))\cos(\phi_c(n-1)) \\ &+ \cos(\phi(n-1))\sin(\phi_c(n-1)) \\ &\approx \sin(\phi(n-1)) + \cos(\phi(n-1))\phi_c(n-1) \end{aligned}$$

Similarly, $\cos(\phi(n)) \approx \cos(\phi(n-1)) - \sin(\phi(n-1))\phi_c(n-1)$ It is assumed $\sin(\theta) \approx \theta$ and $\cos(\theta) \approx 1$, when $\theta \leq 2\pi/125$. However, this optimization does not work because of gradual accumulation of the phase error from the error contribution in the sine and cosine. This optimization can be used in continuous sine and cosine waveform generation and not for phase tracking. However, the increasing error magnitude suggests that the next symbol can be used as an estimate for the current symbol calculation.

Therefore, if we use the cosine calculation of the current symbol instead of the past symbol for the sine calculation of the present symbol, we get a more accurate sine value and thus, reduce the propagation and accumulation of error for both sine and cosine. Equations (5) and (6) define these quantities. The cosine waveform of the NCO for both methods is shown in Fig. 4

$$\cos(\phi(n)) = \cos(\phi(n-1)) - \sin(\phi(n-1))\phi_c(n-1)$$
 (5)

$$\sin(\phi(n)) = \sin(\phi(n-1)) + \cos(\phi(n))\phi_c(n-1)$$
 (6)

Since the frame synchronizer has to supply the phase offset at the beginning of a packet decoding, it can also supply the sine and cosine of the initial phase offset. Note that Frame synchronizer will run only once for a packet, while the NCO has to run for each symbol.

3.2. Phase Detector

The phase detector generates the phase error signal from the received symbols, as monitored before and after the slicer (Fig. 3). When $\theta \le \pi/8$, then $\tan(\theta) \approx \theta$. Here, $\pi/8$ is half of the decision boundary for a symbol period for QPSK. For a higher order modulation scheme, the decision boundary becomes smaller, and there by this approximation is more valid. For BPSK, a small 2nd order



Fig. 4. Cosine Waveform of the NCO for Both Methods.

polynomial is sufficient to generate the arc tangent for the range of 0 to $\pi/4$. Furthermore, the phase error may be approximated as:

$$e_{\phi}(n) = \tan^{-1}(z_q(n)/q_i(n)) \approx q_p(n)/z_i(n)$$
 (7)

For different Eb/No and oscillator drift (ppm) simulations for different single carrier systems, we observe that the range of this variable is from 0.3 to 3 for most of the time. So instead of doing a cumulative subtraction method or Newton Raphson method to calculate the inverse as in case of fixed-point division, a small lookup table (in case of hardware implementations) or a Polynomial fit for five ranges [0.3-1.0] [1.0-1.5],[1.5-2.0],[2.0-2.5] and [2.5-3.0] will be sufficient. This polynomial, p(i), is represented below.

$$\frac{1}{x} = \sum_{i=0}^{k=5} p(i)x^{6-i} \tag{8}$$

Each polynomial is optimized for a particular range for 16 bit variable Q format coefficients for MSE of 3e-5. The MSE for the phase detector with above arc tan and inverse optimizations is $\approx 1e-4$.

4. DISCUSSION AND RESULTS

The optimizations for the coherent receiver were implemented on two different QPSK design platforms. One using a fixed point DSP, the TI TMS 320-VC54x, and another using the Analog Devices floating point DSP SHARC-2106x. With related optimizations across the other blocks in coherent receiver and variable Qformats, the total number of clock cycles to decode a QPSK symbol required on C5402 is reduced from 287 to 188, thus resulting a 33% savings in the MIPs requirement for the base-band coherent receiver. Similar savings are obtained for the 2106x implementation (Table 1). Figures 5 and 6 present the simulation results with new optimizations and Figure 7 shows that there is no performance degradation in terms of BER. Clock cycle estimate for regular (Taylor series for Trigonometric and Newton-Raphson for inverse) methods is based on the optimized assembly routines supplied by TI and Analog Devices. In conclusion, new optimizations which simplify the core operations involved in the carrier synchronization process for single carrier systems are presented. These optimizations are useful for DSP based receiver implementation

 Table 1. Clock Cycle Performance for the TMS320VC54x and SHARC2106x Processors. "Ma" denotes the No of MACs, and so forth, M:Multiplies, A:Addition L:Logical Operations.

Metric	Ma, M, A, L	Ma, M, A, L	Clock cycles	Clock cycles	Clock cycles	Clock cycles
			for C54x	for C54x	for 2106x	for 2106x
Method	Regular	Proposed	Regular	Proposed	Regular	Proposed
Sine	6, 0, 0, 2	1,0,0,0	24	4	25	4
Cosine	6, 0, 0, 2	1,0,0,0	24	4	25	4
Atan	6, 0, 0, 1	0,0,0,0	30	0	30	0
Inverse	16, 16, 0, 2	5,0,0,4	58	22	12	12

and reduce the MIPs requirements while maintaining same BER performance.



Fig. 5. Input Constellation to Re-sampler.



Fig. 6. Constellation for Slicer Input.



Fig. 7. BER for QPSK System with New Optimizations.

5. REFERENCES

- M. Yeary, W. Zhang, and J.Q. Trelewicz, "A computationally efficient decimation filter design for embedded systems," *IEEE IMTC*, vol. 2, pp. 913-916, May 2004.
- [2] U. Mengali and A. D'Andrea, Synchronization Techniques for Digital Receivers, Plenum Press, New York, 1997.
- [3] F. Gardner, "Interpolation in digital modems," *IEEE Trans. on Comm.*, vol. 41, no. 3, pp. 501-507, March 1993.
- [4] R. Andraka, "A survey of CORDIC algorithms for FPGA's," 6th International Symposium on FPGA's, pp. 191-200, Monterey, CA, Feb. 22-24 1998.
- [5] M. Yeary, et al., "Design of a CORDIC processor for mixedsignal A/D conversion," *IEEE Trans. on Instrumentation and Measurement*, vol. 51, no. 4, pp. 804-809, Aug 2002.
- [6] M. Bouchere *et al.*, "Low-complexity carrier-phase estimator suited to on-board implementation," *IEEE Trans. on Communications*, vol. 48, no. 9, pp. 1451-1454, Sept 2000.

V - 664