FEEDFORWARD CARRIER AND SYMBOL TIMING RECOVERY AND PHASE-INVARIANT SIGNALING FOR SOFTWARE-DEFINED RADIO

Robert H. Morelos-Zaragoza (San Jose State University, San Jose, CA 95192 USA; Email: rmorelos@helios.sjsu.edu); Eric Kreb (San Jose State University, San Jose, CA 95192 USA); Ericson Wen (San Jose State University, San Jose, CA 95192 USA); and Boon Wooi Tay (San Jose State University, San Jose, CA 95192 USA)

ABSTRACT

Methods that enable fast carrier and symbol timing recovery of multimode digital communication receivers are discussed. The paper is divided into two parts. In the first part, feedforward synchronization techniques are designed for the reception of multiple digital modulation formats, such as M-PSK and M-QAM. These techniques have the advantages that their mean acquisition time is extremely short compared to feedback synchronizers. We report on a feedforward phase recovery technique for M-PSK (M=2,4 and 8) and 16-QAM and a non-data-aided symbol timing recovery circuit that have been implemented in FPGA technology. In the second part of the paper, we introduce a signal-set design for digital communication without the need of an absolute phase reference at the receiver. Results are presented for quaternary constellations with asymmetrical points whose locations in the complex plane are determined by design parameters. Performance comparisons with conventional and differential modulation techniques are presented.

1. INTRODUCTION

Coherent reception of digitally modulated signals requires the receiver to be synchronous to the transmitter. Data-aided synchronization is a technique to achieve synchronization but it requires bandwidth expansion due to pilot symbols used to estimate the phase and timing of the received signal [1]. In non-data-aided synchronization techniques, in general, more time is needed to establish synchronization. Most of today's digital communication systems employ linear modulation formats using symmetric signal constellations that lend themselves to phase ambiguity [2].

Phase ambiguity needs to be resolved after the received complexed baseband symbols are estimated and results in additional receiver complexity. It is desirable to design a communication system that uses a signal set with the characteristic of not needing a phase reference and whose error performance is better than that of the well-known technique of differential modulation.

The paper is divided into two parts: In the first part, feedforward synchronization techniques are designed for the reception of multiple digital modulation formats, such as M-ary phase-shift keying (M-PSK) and M-ary quadrature amplitude modulation (M-QAM). In the second part, methods of constructing non-uniform signal sets are introduced. It is shown, with 4-ary modulation as an example, that the proposed methods allow phase-invariant signaling with a small performance loss when compared to traditional methods such as differential modulation.

2. FEEDFORWARD CARRIER RECOVERY

We consider a digital communication system employing linear amplitude and/or phase modulation with coherent detection at the receiver. In order to estimate the carrier phase, a group of L_0 incoming complex baseband symbols is collected. The value of L_0 is determined by the length of an *observation period*. Without data aid, the symbols are unknown to the synchronizer prior to both phase recovery and symbol timing recovery processes. For the purpose of phase recovery, both timing and frequency are assumed to be practically recovered and almost ideal. Therefore, in our design, the receiver architecture is such that properly synchronized symbols, extracted by the timing recovery circuit from the unsynchronized samples out of the analog-to-digital conversion stage, are fed to the phase estimator; while the frequency of the carrier signals is modeled as having a small offset with respect to the transmitter frequency and assumed to be much smaller than 1/T.

2.1 Matlab ™ model [3]

Assume that the format of the received complex baseband symbols is M-PSK. Raising to the M-power the incoming symbols, and taking the average of the result over the observation period, yields an estimation of the phase offset. However, for 16-QAM the phase estimator requires extra processing before estimating the phase of the symbols. In this work, a filter is used that selects high-energy symbols. Only the four highest-energy symbols out of the sixteen possible are selected for further processing. As a result, the subsequent phase estimator of 16-QAM becomes simply that of QPSK modulation. Under the plausible assumption of a uniform distribution of the transmitted symbols, the probability of getting one of these highest-energy symbols is ¼. We found this method to give good performance, with an appropriate length of observation period, L_0.

As the phase is estimated by observing a number L_0 of incoming symbols, the length of observation period is taken into consideration carefully. The frequency of the local oscillator at the demodulator is not perfectly matched to frequency of the carrier signal. The mismatch between the carrier frequency and the local oscillator's frequency is denoted here by Δf. An assumption is made here that Δf is much smaller than the symbol rate, 1/T. This assumption is reasonable and can be ensured with practical oscillators. The existence of Δf causes the received symbols to rotate at an angular speed of $2\pi\Delta f$ for every T seconds.

2.1.1 Carrier phase estimator

A feedforward non-data aided estimation algorithm was studied. The phase of the incoming complex baseband symbols (for each of the possible modulation formats: BPSK, QPSK, 8-PSK, and 16-QAM) is estimated with different parameters for each modulation mode. The estimate of the phase for PSK modulation modes can be expressed as

$$\hat{q} = \frac{1}{M}\arg\left\{\sum_{k=0}^{L_0-1} x^M(k)\right\}, \qquad (1)$$

where M is the number of constellation points for the corresponding modulation modes (for M=2, 4, and 8) and L_0 represents the number of observed symbols within an observation period. Depending on these two parameters, the model can be built either with a switching circuit to select the parameters corresponding to the specific modulation modes, or with one structure for each modulation mode. In our model, three structures of phase estimator have been built, as the length of the observation period varies depending on the modulation mode.

Note that Eq. (1) can be written in the form

$$\hat{q} = \frac{1}{M}\arg\left\{\sum_{k=0}^{L_0-1} r^M(k)e^{jMf(k)}\right\}. \qquad (2)$$

It follows from Eq. (2) that computation of the phase estimation can be performed by taking the M-th power of the magnitude of the incoming symbols and then multiplying by M the phase of the incoming symbols, before storing the results into memory for averaging over an observation period. Taking the average removes the effects of additive white Gaussian noise. Therefore a large value of L_0 minimizes the effects of noise, while, on the other hand, a small value of L_0 is required to accurately track variations in the carrier phase due to frequency offset. All phase estimates in all PSK modes are computed according to Eq. (2) in our model, with parameters M and L_0.

2.2 Phase tracking logic

A close look at Eq. (1) reveals that the phase estimate is computed by multiplying by factor of 1/M the results gathered within an observation period. Consequently, the phase estimate ranges from $-\pi/M$ to π/M. A circuit is needed to track the estimate of phase and produce a correct phase that ranges from $-\pi$ to π. This process is known as *phase unwrapping* and is accomplished by tracking the changes of sign of the phase estimate [1]. Whenever there is a change of sign from positive to negative, or viceversa, a corresponding phase offset is added to, or subtracted from, the phase estimate, respectively. The resulting phase is then used to compute the value of a counter-rotating phasor that multiplies the incoming symbols and corrects the phase offset.

2.2 FPGA implementation

An FPGA implementation of the Matlab Simulink™ [3] model was performed with Xilinx System Generator™ [4]. The conversion of the Matlab Simulink™ model into the FPGA model amounts to finding similar blocks in the Xilinx System Generator™ library. However, it was found that, with the restriction of limited hardware realization of most Matlab Simulink™ blocks, it was easier to compute the estimate of the phase with the incoming symbols in their in-phase and quadrature form, rather than in magnitude and phase form. Avoiding using the magnitude and phase form to compute the estimate of phase reduced the amount of blocks, and hence hardware complexity. A new representation of Eq. (1) was derived in the terms of their in-phase and quadrature components as follows:

$$x(k) = r(k)e^{jf(k)} = x_I(k) + jx_Q(k) = r(k)\cos(f(k)) + jr(k)\sin(f(k)) \qquad (3)$$

From Equation (3), the estimate of phase for each PSK modulation can be expressed by

$$\begin{aligned}\text{BPSK: } x^2(k) &= r^2(k)e^{j2f(k)} \\ &= r^2(k)\cos(2f(k)) + jr^2(k)\sin(2f(k) \\ &= \left[r^2(k)\cos^2(f(k)) - r^2(k)\sin^2(f(k))\right] + j2r(k)\sin(f(k))r(k)\cos(f(k))\end{aligned}$$

(4)

$$\text{QPSK: } x^4(k) = r^4(k)e^{j4f(k)}$$
$$= r^4(k)\cos(4f(k)) + jr^4(k)\sin(4f(k))$$
$$= \left\{\left[r^2\cos(2f(k))\right]^2 - \left[r^2\sin(2f(k))\right]^2\right\} + j2r^2\sin(2f(k))r^2\cos(2f(k))$$

(5)

$$\text{8PSK: } x^8(k) = r^8(k)e^{j8f(k)}$$
$$= r^8(k)\cos(8f(k)) + jr^8(k)\sin(8f(k))$$
$$= \left\{\left[r^4\cos(4f(k))\right]^2 - \left[r^4\sin(4f(k))\right]^2\right\} + j2r^4\sin(4f(k))r^4\cos(f(k))$$

(6)

Equations (4), (5), and (6), indicate how $x^M(k)$ can be computed from the in-phase and quadrature components of the incoming complex baseband symbols. Furthermore, the value $x^4(k)$, which refers to QPSK modulation, can be computed by using of the result of $x^2(k)$, while $x^8(k)$, which refers to 8-PSK modulation, can be computed from the result of $x^4(k)$. In other words, the phase estimator for the three PSK modes can be implemented within a single structure that is able to compute the estimate of phase for all the three PSK modes. The parameter M, which varies according to the corresponding M-PSK mode, is no longer needed in the FPGA hardware structure.

2.2.1 Phase tracking loop

Phase unwrapping was implemented with a state machine that tracks the sign of the phase estimate. The state changes when a change of sign is detected, and the output is set or cleared accordingly. This signal is used as a clock enable of the counters. This phase-unwrapping state machine consists of three states: State 0 is the initial state where the output is set at low state. As the estimate of phase rotates and is tracked, State 0 will change to State 1, as soon as change of sign is detected, and the output is set to high state at State 1 to enable all counters. The state then changes from 1 to 2, in the next clock cycle, where the output is set to low for N_c clock cycles before returning to State 0. Fig. 1 shows the state diagram of the phase-unwrapping circuit.

Data Width

As the FPGA architecture is a *fixed-point* architecture, all data computed in the phase estimator are represented in fixed point format. Since symbol energy is normalized to one, the number of bits for the integer part usually does not require more than 2 bits (MSB for sign and LSB for the magnitude). As a result, most bits are used as fraction to get precise values.

Simulations have been conducted with several data widths, starting from as high as 64 bits and finally down to 16 bits. A main reason to have model simulations with lower data width is to optimize the performance of the hardware. For example, a multiplier that requires latency of two clock cycles can only support up to 18-bit data inputs. A higher latency is required for a higher data width. Also, although an 18x18 multiplication will produce a 36-bit data, if full precision used, data can always be normalized to a smaller value.

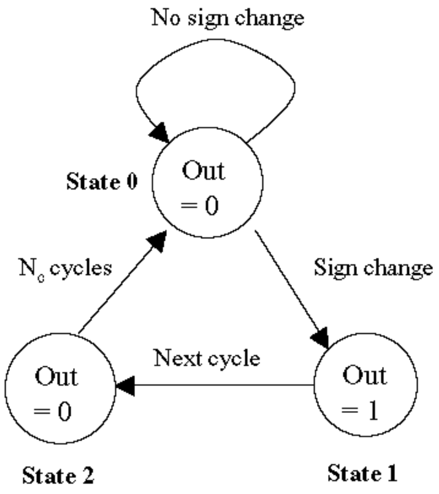

Figure 1: State diagram for phase unwrapping.

Delay

Delay is inevitable in a hardware implementation. From Equation (4), it is clear that we need three multipliers in parallel to compute $x^2(k)$. Each multiplier requires a latency of two clock cycles in the FPGA architecture. With three multipliers in parallel, the delay remains the same. However, as the computation becomes more complex from BPSK to 8-PSK, where the higher modulation mode relies on the lower modulation mode, delay increases. Also, we note that an adder and a constant-multiplier do not require any delay. Table 1 shows the number of multipliers, adders, and constant-multipliers required to compute the values $x^2(k)$, $x^4(k)$, and $x^8(k)$.

Even though all computations of the phase estimation circuit are based on the in-phase and quadrature components of the incoming symbols, it is necessary to calculate the phase out from the result of these computations. A cordic algorithm was implemented to perform the phase and magnitude calculation. In the current hardware architecture, there are two blocks that implement cordic algorithms. The first is used to compute the phase out from an observation period of incoming symbols, while the second one converts the complex baseband symbols in magnitude-phase form back to their in-phase and quadrature form. Both cordic blocks require three clock cycles each.

Table 2 shows the delay needed to compute the estimate of phase and the total delay required for producing the phase-recovered symbols, for each PSK mode.

3. TIMING RECOVERY

In a practical digital communication system, the receiver sampling frequency is independent of the transmitting clock. The timing estimator has the main task of distinguishing and interpolating the correct data symbols from the unsynchronized received samples. The timing estimator presented here operates under realistic conditions such as phase noise, clock jitter, and an irrational value of the ratio T/T_s of the symbol period T to the sample period T_s.

A feedforward non-data aided timing recovery architecture was selected due its low acquisition time and no overhead. Fig. 2 shows a block diagram of the timing recovery circuit.

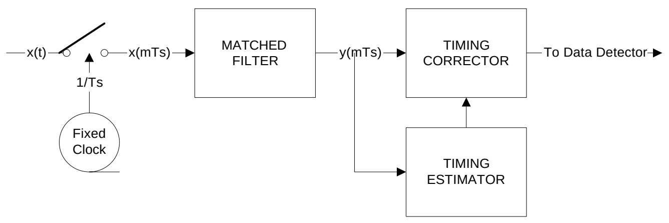

Figure 2: Block diagram of the timing recovery circuit

Without changing or adjusting the internal sampling clock, the proposed timing recovery circuit oversamples the incoming matched-filter samples and calculates an estimate of the timing offset value, τ. With the calculated values of τ, the timing corrector (a linear interpolator) constructs the synchronized symbol values.

The timing offset is calculated with the following equation:

$$\hat{t} = -\frac{T}{2p}\arg\left\{\sum_{k=0}^{NL_0-1} x^2\left(kT_s\right)e^{-j2pk/N}\right\}, \qquad (7)$$

where N is the number of samples taken between symbol periods, T is the symbol period, and L_o is the observation period. The numbers of samples per symbol period was chosen to be N=4, due to hardware complexity and computation time considerations. The above equation in this case reduces to the inverse tangent of the sum of the imaginary magnitudes (odd samples) over the sum of the real magnitudes (even samples).

The timing recovery circuit consists of four subparts: Timing error estimator (TEE), estimator of the fractional time delay μ, linear interpolation, and computation of the closest sample index, m_k. The TEE implements the formula mentioned above, while the μ computation truncates the integer part of the estimated timing offset τ. The variable μ is a parameter that the linear interpolator uses to compute the intermediate symbol between the closest two samples. Finally, m_k can be interpreted as a time shift, in the receiver clock domain, of the symbol to the correct sample time epoch.

4. PHASE-INVARIANT SIGNALING

In this section, a modulation scheme based on PSK is introduced. The main idea is to place signal points in the complex plane in such a way that a fully asymmetric constellation is obtained. This results in a phase-invariant signal set that offers advantages over conventional symmetric constellations such as QPSK or 4-PAM, as will be shown in a subsequent section. Several methods to construct quaternary asymmetric signal constellations have been investigated. These methods include pure phase rotations, as well as nonuniform amplitude distributions that can satisfy a given crest factor (CF).

The proposed methodology is able to construct new phase-invariant modulation schemes that are altered versions of symmetric PSK constellations. Extensions to QAM constellations have also been investigated. The key components of the design involve the minimization of the CF and the maximization of the Euclidean distance between adjacent symbols.

The signal set design has undergone several phases, beginning with phase invariance and constant amplitude, and ending with a phase-invariant signal set of non-constant amplitude that minimizes the probability of error when affected by an AWGN channel. Performance results for the bit error rate (BER) versus the bit energy-to-noise ratio, E_b/N_o, are presented later in the paper to compare the proposed phase-invariant signal constellations with both QPSK and differential QPSK (DPSK).

On the receiver side, the detector assumes knowledge of the potential symbols available and conducts a search for the shortest distance to a received signal. Without knowledge of the CF, other than through observation of the received signal, the receiver will have to determine the most likely symbol set that was sent.

4.1 Construction of phase-invariant (PI) signal sets

Several variations of construction methods based on an initial symmetrical QPSK signal constellation have been investigated and reported below. The first two variants hold amplitude constant while condensing the phase linearly and nonlinearly, analysis demonstrated the limited use of such methods. We refer to the linear scheme as *a-mapping* and to the nonlinear scheme as *b-mapping*. Figs. 3 and 4 show the input-output characteristics for the α- and β-mapping, respectively.

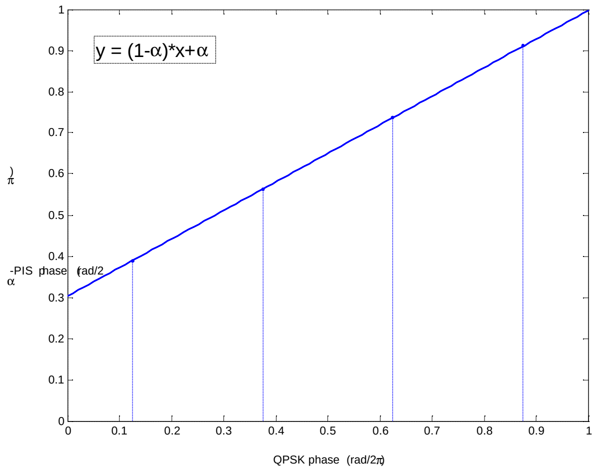

Figure 3: α-mapping.

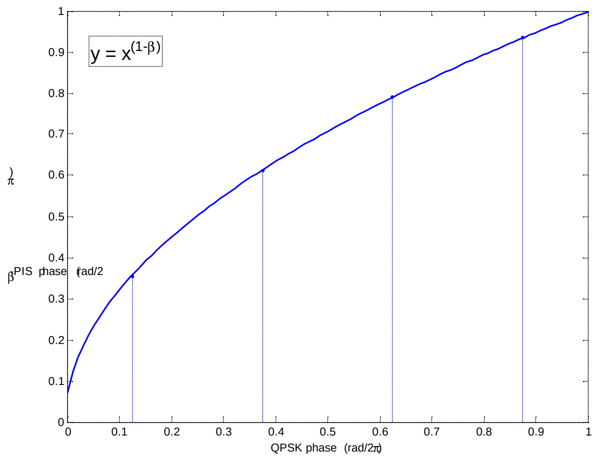

Figure 4: β-mapping.

The third and fourth variants use an amplitude mapping function, which trades off the CF of the signal. All of these variations use a dynamic ‘offset’ parameter for optimization through experimentation. The third variant is pure amplitude mapping, while the fourth adjusts the phases and amplitudes to obtain equidistance among adjacent neighbors. The fourth variant has not been fully developed yet. It includes elliptical and space-shift mappings.

The fifth variant is expected to give the best results. The input parameters are the CF and an equidistant adjacent symbol precision (EASP). According to the CF, a distribution function arranges the amplitudes of a subset of signal points. The selected signal points are ranked in accordance to their phases and are chosen to remain within EASP limits. Yet another variation is to consider permutations of phase arrangements. The arrangement that maximizes the Euclidean distance between nearest neighboring symbols is selected.

4.1.1 *Performance of a-mapping*

The performance of the linear α-mapping, with QPSK as initial signal set, was evaluated over an AWGN channel and its performance compared to coherent QPSK and differential QPSK (DPSK). The results are shown in Fig. 5. It is observed that, due to a reduction in the minimum Euclidean distance of the PI constellation, its error performance eventually becomes worse than that of DPSK. However, for relatively small values of α, it can be argued that the performance of quaternary PI signaling with (linear) α-mapping is better than DPSK, in a coded digital communication system for which the required BER is typically greater than 10^{-3} due to error correcting coding.

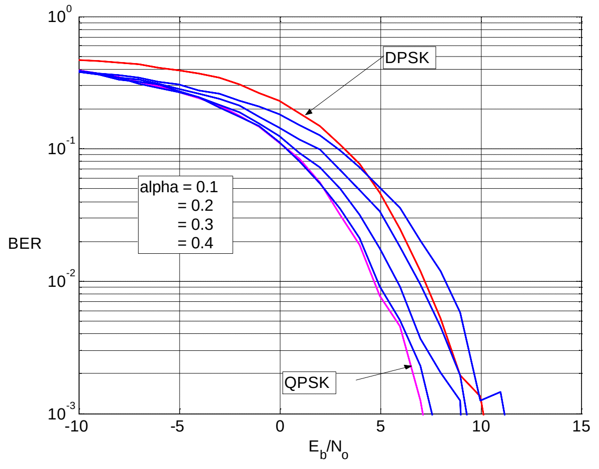

Figure 5: Error performance of PI α-QPSK.

4.1.2 b-mapping

The performance of the nonlinear scheme, where the phase is modified in a nonlinear fashion, is shown in Fig. 6. As it is evident from the figure, compared with the linear mapping, the error performance of the PI β-QPSK constellation improves considerably. For β-mapping, the amplitudes of the original QPSK constellation remain constant, while the phases are modified as indicated in Fig. 4. The Euclidean distance between adjacent signal points decreases, when β increases, resulting in a higher bit error rate compared to the symmetric QPSK constellation. However, it follows from the results that, over a large range of β, the PI signal set outperforms DPSK for values of practical BER above 10^{-3}.

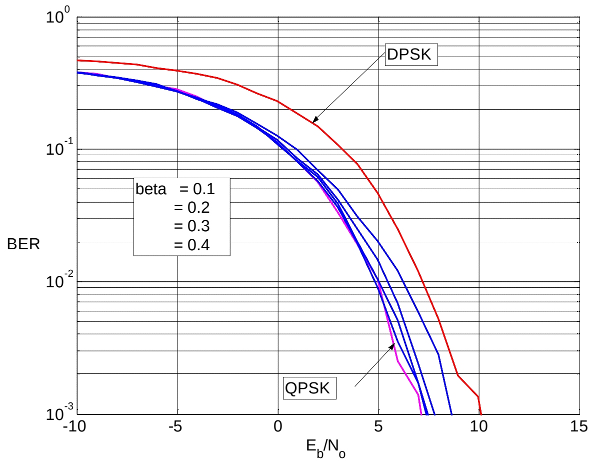

Figure 6: Error performance of PI β-QPSK.

5. CONCLUSIONS

In this paper, methods that enable fast synchronization of multimode digital communication receivers have been discussed. Two feedforward synchronization circuits, for phase and symbol timing recovery, have been designed and implemented in FPGA technology. The circuits work for a multiplicity of digital modulation formats, such as M-PSK and M-QAM. Also, a signal-set design for digital communication was introduced. An asymmetry is introduced in the signal constellation such that an absolute phase reference is not needed at the receiver. Performance results for quaternary PI constellations over AWGN channels show that the proposed modulation technique outperforms conventional differential modulation.

6. REFERENCES

[1] H. Meyr, M. Moeneclaey and S.A. Fetchel, *Digital Communication Receivers*, Wiley-Interscience, New York, 1998.

[2] J.G. Proakis, *Digital Communications*, 4th ed., McGraw-Hill, New York, 2001.

[3] Matlab™ and Simulink™ are registered trademarks of The MathWorks, Inc., Natick, MA.

[4] System Generator™ is part of XtremeDSP™, both registered trademarks of Xilinx, Inc., San Jose, CA.

Table 1

Modulation Mode	Multipliers	Adders	Constant-multiplier	Accumulated Delay
BPSK	3 in parallel	1	1	2
QPSK	3 in parallel	1	2 in parallel	4
8-PSK	3 in parallel	1	1	6

Table 2

Modulation Mode	Delay to estimate the phase	Total delay to produce phase-recovered data
BPSK	21	28
QPSK	23	30
8-PSK	25	32