Polyphase Filterbanks for Symbol Timing Synchronization in Sampled Data Receivers

Michael Rice
Department of Electrical & Computer Engineering, Brigham Young University, Provo, UT 84602
mdr@ee.byu.edu 801 422 4469 (V) 801 422 0201 (F)
fred harris
Department of Electrical & Computer Engineering, San Diego State University, San Diego, CA 92182
fred.harris@sdsu.edu 619 594 6162 (V) 619 594 3112 (F)

Abstract— **Symbol timing synchronization is an important component in a receiver designed to recover data from a digitally modulated waveform. Current trends favor sampled data architectures to perform the synchronization, matched filtering, and detection required to recover the data. Symbol timing synchronizers in a sampled data receiver differ from their continuous-time counterparts: the functionality in a sampled data receiver is an adaptive interpolator as opposed to a sample and hold. This paper describes the use of a polyphase filterbank for symbol timing synchronizers in sampled data receivers. Using this approach, interpolation and matched filtering are rolled into a single operation that requires less resources than separate filters for matched filtering and interpolation. Interpolations are realized by filterbank index selection. Since the filterbank index plays the role of the fractional interpolation interval, the same loop control techniques used in other sampled data synchronizers can be used. Maximum likelihood timing synchronization techniques as well as "early-late gate" techniques can be incorporated into the polyphase filter bank in a natural way.**

I. Introduction

In an effort to produce a single reconfigurable communications platform that accommodates multiple incompatible waveforms, the software defined radio has emerged as a promising solution. At the core of software defined radio modems is a sampled data system that performs the detection and synchronization tasks formerly accomplished with analog hardware. As the transition from analog hardware to discrete-time processing progresses, care must be taken to avoid propagating the implementation compromises of the analog system. Each function should be examined anew to take advantage of any DSP functionality that has no counterpart in the analog world.

As an example, consider timing synchronization subsystems for MQASK modems. The complex baseband transmitted waveform may be represented by

$$s(t) = \sum_{k} a_k p\left(t - kT_s\right) \tag{1}$$

where a_k is the k-th complex valued M-ary symbol drawn from a constellation with average symbol energy E_s, $p(t)$ is the unit energy pulse shape that spans $2L$ symbols, and T_s is the reciprocal of the symbol rate. The optimum detector filters the received signal $r(t) = s(t-\tau) + w(t)$ (τ is the delay relative to the detector's time axis and $w(t)$ is zero-mean, white Gaussian noise) with a matched filter whose impulse response is $h(t) = p(-t)$. The resulting output $y(t) = r(t) * h(t)$ is sampled at the end of the symbol interval to produce the decision variable used for the making the symbol decision [1]. This requires knowledge not only the symbol rate $1/T_s$ but also the exact sampling instant within the symbol interval (i.e. the *timing phase*). Extracting this knowledge may be done either with continuous-time processing, discrete-time processing, or both.

In analog implementations, the timing synchronization subsystem determines the optimum times to sample the output of the analog matched filter. The sampling is usually controlled via a clock whose edges are phase aligned to the symbol boundaries. In a discrete-time implementation, the timing synchronizer is quite different. The data are already sampled so that timing synchronization is better understood as adaptive interpolation [2].

In a sampled data receiver, the received signal is sampled at a rate $1/T$ that satisfies the Nyquist sampling Theorem and is N times the symbol rate $1/T_s$ (e.g. $N = 2$ for a square-root raised cosine pulse shape). These samples $r(nT)$ are then filtered by the discrete-time matched filter with impulse response $h(nT) = p(-nT)$. The desire is to produce N samples of the matched filter output during each symbol interval such that one of the samples is as close to $y(kT_s - \tau)$ as possible.

There are two basic approaches to the problem. The first approach is illustrated in Figure 3 (a). This approach computes a timing error value which is used to adjust the phase of the voltage controlled clock (VCC) that triggers the ADC. As a result, the samples of $r(t)$ are aligned with the symbol boundaries. The systems described in [3]-[7] are of this type. This approach has the advantage that it produces samples that are aligned in both phase and frequency with the data clock (i.e., T and T_s are commensurate). The disadvantages of this approach include the requirement of a feedback path to the continuous-time part of the system, the inclusion of the matched filter transport delay in the feedback path of the loop, a higher level of phase noise (and hence timing jitter) contributed by a VCC (relative to a fixed rate clock), and the inability to place the ADC at the IF if the IF signal contains multiplexed signals whose symbol clocks are derived from independent sources.

The second approach, illustrated in Figure 3 (b), addresses these issues by sampling the received signal $r(t)$ at a fixed rate $1/T$ that is asynchronous with the data clock period $1/T_s$ (i.e. T and T_s are potentially incommensurate). The time delay τ is estimated solely from $y(nT)$. The problem is best understood when cast as an interpolation problem where the sample rate at the output of the interpolator is slightly different from the sample rate at the input of the interpolator [2]. Figure 1 demonstrates the scenario and provides a basis for the definitions to

0-7803-7625-0/02/$17.00 ©2002 IEEE.

be used later. The continuous-time matched filter output $y(t)$ is represented by the dashed line and the available samples are represented by the solid circles. The desired matched filter outputs are represented by the clear dots and occur every T_i seconds. Since the available samples occur at a slightly faster rate than $N = 2$ samples per symbol, T_i will not be exactly $2T$. The matched filter output corresponding to symbol a_k occurs at time kT_i. The sample $y(kT_i)$ is called the k-th interpolant. The index m_k which is the integer part of kT_i/T is called the basepoint index since the k-th interpolant occurs in between $y(m_kT)$ and $y((m_k+1)T)$. The fractional interval $\mu_k = kT_i/T - m_k$ specifies the interval between $y(m_kT)$ and $y((m_k+1)T)$.

There are two ways to compute the desired interpolants. The first way uses an interpolation filter to compute $y(kT_i)$ from the available samples $\ldots, y((m_k-1)T), y(m_kT), y((m_k+1)T), \ldots$ [8],[9, Chapter 7], [10, Chapters 4,9]. The interpolation filter $h_I(nT)$ is usually based on a polynomial (e.g. linear, piece-wise parabolic, or cubic) and computes the desired interpolant using [2]

$$y(kT_i) = \sum_{i=I_1}^{I_2} y((m_k-i)T)h_I((i+\mu_k)T) \qquad (2)$$

where the time span of the interpolation filter is the interval $I_1 \leq i \leq I_2$. An exceptionally efficient architecture for performing this is function is described in [11]. A timing controller is required to indicate which sample is the k-th basepoint index and the corresponding fractional interval μ_k.

The second approach uses polyphase partition of the matched filter to compute the desired interpolants. [12]-[16]. From this point of view, the desired matched filter output corresponding to the k-th symbol is

$$y(kT_i) = y(m_kT - \mu_kT). \qquad (3)$$

The basic idea is to upsample the received signal by a factor M by inserting $M-1$ zeros in between the samples of $r(nT)$ and *use an upsampled version of the matched filter to perform the interpolations.* As shown in the next section, the polyphase filter partitions the inter-sample interval into M subintervals. Delay selection is performed by choosing the output of the appropriate subfilter in the filterbank.

II. Polyphase Filterbanks for Timing Synchronization

The polyphase filter structure is illustrated by the following simple example illustrated in Figure 2. Suppose that the required timing resolution is MN parts per symbol and that samples of the phase-corrected complex baseband signal $r(nT)$ provide approximately N samples/symbol. The sequence $r(nT)$ is upsampled by a factor M by inserting $M-1$ zeros between each sample of $r(nT)$ to produce a new sequence $r(nT/M)$ that provides MN samples/symbol. The sequence forms the input to a matched filter whose impulse response is $h(nT/M)$ is also sampled at MN samples/symbol and spans $2L$ symbols. The timing control selects approximately N samples during each symbol period such that one of them is as close to the optimum sampling instant as possible. The output $y(nT/M)$ is given by

$$y\left(n\frac{T}{M}\right) = \sum_{l=-MNL}^{MNL} r\left((n-l)\frac{T}{M}\right) h\left(l\frac{T}{M}\right). \qquad (4)$$

The output is downsampled to produce N samples per symbol where one of the samples is as close to $y(kT_s - \tau)$ as the resolution allows. The polyphase decomposition is due to the fact that not all of the multiplies defined by (4) are required. Since

$$r\left(n\frac{T}{M}\right) = \begin{cases} r(nT) & n = 0, \pm M, \pm 2M, \ldots \\ 0 & \text{otherwise,} \end{cases} \qquad (5)$$

only every M-th value of $r(nT/M)$ in the FIR filter is non-zero. At the m-th time instant at the high sample rate, these non-zero values coincide with the filter coefficients

$$\ldots, h(-MT+m), h(m), h(MT+m), h(2MT+m), \ldots$$

so that the filter output may be expressed as

$$\sum_{i=-NL}^{NL} r((n-i)T)h\left(\left(i+\frac{m}{M}\right)T\right) = y\left(\left(n-\frac{m}{M}\right)T\right). \qquad (6)$$

This characteristic is illustrated in Figure 4 where a parallel bank of M filters, operating at the low sample rate $1/T$ is shown. Each filter in the filterbank is a downsampled version of the matched filter, except with a different index offset. The impulse response for $h_m(nT)$ is $h_m(nT) = h\left(nT + \frac{m}{M}T\right)$. The data samples $r(nT)$ form the input to all the filters in the filterbank simultaneously. The desired phase shift of the output is selected by connecting the output to the appropriate filter in the filterbank.

To see that the output of the m-th filter in the polyphase filter bank given by (6) does indeed produce the desired result, assume that the k-th basepoint index $m_k = n$. Then the output of the m-th filterbank is given by (6) with $n = m_k$:

$$y_m(m_k) = y\left(m_kT - \frac{m}{M}T\right) \qquad (7)$$

which is identical to (3) where the ratio m/M plays the role of the k-th fractional interval μ_k. In this way, the polyphase filterbank implements the interpolation with a quantized fractional interval. The degree of quantization is controlled by the number of polyphase filter stages in the filterbank.

The desired output is selected by a discrete-time phase locked loop (DPLL). When used in conjunction with a timing phase error detector and NCO control [2], the DPLL outputs the index associated with the proper phase of the matched filter output corresponding to the maximum eye opening.

Maximum likelihood timing error estimation can be incorporated into the polyphase filterbank in a natural way as illustrated in Figure 5. The polyphase implementation of the early-late gate detector is shown in Figure 5 (a). The outputs of polyphase filter stages immediately preceding and following the current filter stage are used to form the phase error with $\Delta\tau/T = 1/MN$. This approximation requires three polyphase stage filter outputs for each matched filter output (two for the early-late-gate error

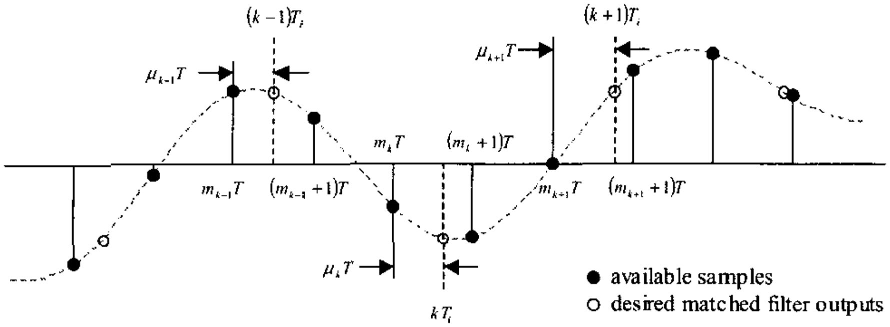


Fig. 1. Relationship between matched filter output, available samples, and interpolants. The continuous time matched filter output is indicated by the dashed line and the available samples (at approximately 2 samples/symbol) are indicated by the solid circles. The desired interpolants are indicated by the clear circles. Note that the position of the desired interpolants slides to the right as time progresses indicating that the sample rate is slightly higher than 2 samples/symbol.

and one for the actual matched filter output). The preceding and following polyphase filter outputs can also be used to form the first central difference which approximates the derivative of the matched filter output at the current filter stage as illustrated in Figure 5 (b). The matched filter output and its derivative are then combined to form the maximum likelihood error signal. Again, this error detector requires three polyphase stage outputs for each matched filter output. Instead of using two filters to compute the derivative matched filter output $\dot{y}(\tau)$, a single filter, whose coefficients are given by

$$\dot{h}_m(nT) = h_{m+1}(nT) - h_{m-1}(nT) \quad m = 1, 2, \ldots, M-2 \tag{8}$$

$$\dot{h}_0(nT) = h_1(nT) - h_{M-1}(nT) \tag{9}$$

$$\dot{h}_{M-1}(nT) = h_{M-2}(nT) - h_0(nT) \tag{10}$$

can be used [9]. This method is illustrated in Figure 5 (c) where $\dot{h}(nT_s)$ is the derivative matched filter. This structure requires only two polyphase filter stages for each matched filter output.

III. Simulation Results

Closed-loop simulation results are presented to demonstrate the main features of polyphase filterbanks for timing synchronization. The principle of operation does not depend on the type of timing error detector. The examples presented in this section use the maximum likelihood timing error detector (low signal-to-noise ratio approximation) to illustrate the ease with which the polyphase filterbank can be incorporated into the phase detector. The first simulated system is illustrated in Figure 6. The modulation is QPSK with square-root raised cosine pulse shapes with 25% excess bandwidth. Data samples at approximately $N = 2$ samples/symbol are processed by the polyphase matched filter (MF) and polyphase derivative matched filter (dMF) filterbanks each with $M = 32$ stages. The product of the two filterbank outputs form the timing error which is updated once per symbol. The timing error signal is upsampled by 2 and filtered by a proportional-plus-integrator loop filter which is required for a second order loop to track out the symbol clock frequency offset [17]. The loop filter output is used to control the increment in the counter (NCO) which underflows at the optimum timing instant when the loop has achieved lock.

In all simulations summarized in this paper the gain of the timing error detector was normalized to unity so that the loop characteristics were determined solely by the loop filter constants $K1$ and $K2$. The filter constants were chosen to produce a critically damped loop with a closed-loop single-sided noise bandwidth of 0.5% of the symbol rate.

Simulation results illustrating the phase step response and frequency step response for the closed-loop system are illustrated in Figures 7 and 8. In these plots the dark smooth lines are the timing error, NCO control, and polyphase index for alternating data and the gray lines are plots for random data with equally probable symbols. The gray lines illustrate the effect of modulation noise on loop performance (no additive channel noise was included in the simulations). For the phase step of $1/4T_s$, the three plots in Figure 7 demonstrate how the loop chooses successively larger filterbank indexes to successively increase the filterbank delay until the filterbank delay matches the delay in the data. Once the filterbank delay matches the clock delay, the polyphase index settles to a constant steady-state value (16 in this case) and the NCO control settles to approximately 1/2 since the loop is running at 2 samples/symbol.

For the frequency step (or phase ramp), data with a simulated sample clock offset 1/250 of the symbol clock was input to the loop. In this case, the optimum symbol sampling instant appears to slide through the data (as illustrated in Figure 1) at a rate equal to the timing offset. As a consequence, the optimum sampling instant slides as well and does so at a rate equal to the timing offset. This behavior is observed in polyphase index illustrated in the lower plot of Figure 8. Since the sample clock period T is less than $2T_s$, the optimum sample time delay increases with each sample. The loop tracks this by increasing the polyphase filter index. The sequence of polyphase indexes repeats every 250 symbols as a result of the symbol clock offset. This behavior matches exactly the behavior of the fractional interval μ reported in [8] for the loop using a separate polynomial-based interpolation filter.

IV. Conclusions

The use of a filterbank based on a polyphase decomposition of an upsampled version of the matched filter for timing synchronization has been described. The interpolations required for for symbol timing synchronization in a sampled data processor occur naturally in the filterbank. The filterbank index is proportional to the quantized fractional interval so that filter delay is adapted by changing to a different subfilter in the filterbank. Maximum likelihood timing error detectors can also be incorporated into the filterbank in a natural way leading to an efficient structure of realizing ML timing estimation.

REFERENCES

[1] J. G. Proakis, *Digital Communications*, McGraw-Hill, New York, fourth edition, 2001.

[2] F. M. Gardner, "Interpolation in digital modems — part I: Fundamentals," *IEEE Transactions on Communications*, vol. 41, no. 3, pp. 501–507, March 1993.

[3] A. Jennings and B. R. Clarke, "Data-sequence selective timing recovery for PAM systems," *IEEE Transactions on Communications*, vol. 33, no. 7, pp. 729–731, July 1985.

[4] W. G. Cowley and L. P. Sabel, "The performance of two symbol timing recovery algorithms for PSK demodulators," *IEEE Transactions on Communications*, vol. 42, no. 6, pp. 2345–2355, June 1994.

[5] A. N. D'Andrea and M. Luise, "Optimization of symbol timing recovery for QAM data demodulators," *IEEE Transactions on Communications*, vol. 44, no. 3, pp. 399–406, March 1996.

[6] D. W. Paranchych and N. C. Beaulieu, "Performance of a digital symbol synchronizer in cochannel interference and noise," *IEEE Transactions on Communications*, vol. 48, no. 11, pp. 1945–1954, November 2000.

[7] K. H. Mueller and M. Müller, "Timing recovery in digital synchronous data receivers," *IEEE Transactions on Communications*, vol. 34, no. 5, pp. 423–429, May 1976.

[8] L. Erup, F.M. Gamder, and R.A. Harris, "Interpolation in digital modems — part II: Implementation and performance," *IEEE Transactions on Communications*, vol. 41, no. 6, pp. 998–1008, June 1993.

[9] U. Mengali and A. N. D'Andrea, *Synchronization Techniques for Digital Receivers*, Plenum Press, New York, 1997.

[10] H. Meyr, M. Moenecleay, and S. Fechtel, *Digital Communication Receivers: Synchronization, Channel Estimation, and Signal Processing*, John Wiley & Sons, New York, 1998.

[11] C. Farrow, "A continuously variable digital delay element," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, Espoo, Finland, June 1988, pp. 2641–2645.

[12] f. j. harris and T. Weschselberger, "Multirate all-digital modems for transport of universal multiplex transport layer for digital compression," in *Proceedings of the National Cable Television Association (NCTA) Conference*, San Francisco, June 1993.

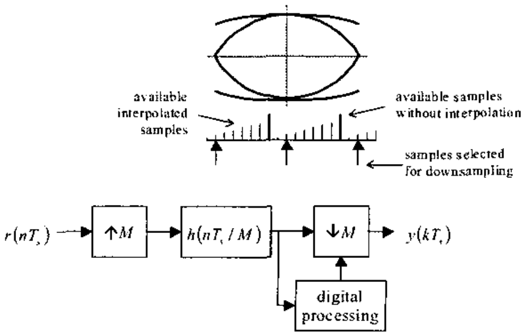
[13] Jim Tiernan fred harris, Don Becker, "Digital receiver for variable rate symbol synchronization," Patent US5504785, April 1996.

[14] C. Dick, M. Rice, and f. j. harris, "Synchronization in software radios: Carrier and timing recovery using FPGA's," in *Proceedings of the IEEE Symposium of Field Programmable Custom Computing Machines*, Napa Valley, CA, April 2000.

[15] f. j. harris, "Multirate digital filters used for timing recovery in digital receivers," in *Proceedings of the Asilomar Conference on Signals, Systems, and Computers*, 2000, pp. 246–251.

[16] f. harris and M. Rice, "Multirate digital filters for symbol timing synchronization in software defined radios," *IEEE Journal on Slected Areas in Communications*, vol. 19, no. 12, pp. 2346–2357, December 2001.

[17] F. M. Gardner, *Phaselock Techniques*, Wiley, 1979.



available interpolated samples
available samples without interpolation
samples selected for downsampling
$r(nT_s)$
$\uparrow M$
$h(nT_s/M)$
$\downarrow M$
$y(kT_s)$
digital processing

Fig. 2. Interpolation using upsampling, filtering at the high rate, and downsampling.

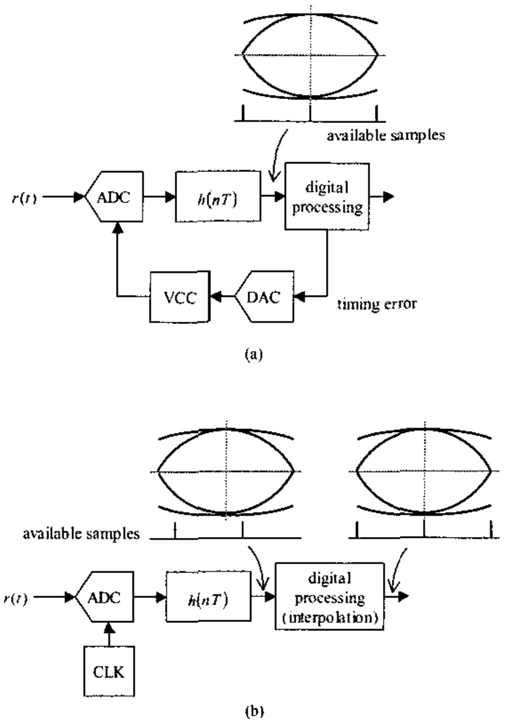


Fig. 3. The two basic discrete-time timing synchronization approaches.

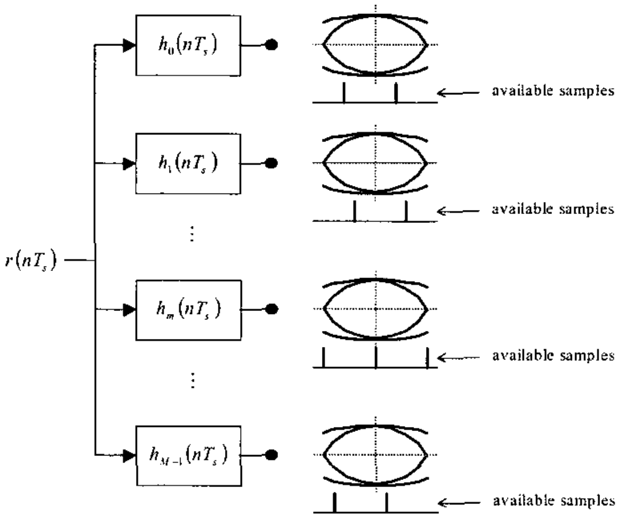


Fig. 4. Polyphase filterbank interpolator equivalent to the interpolator shown in Figure 2.

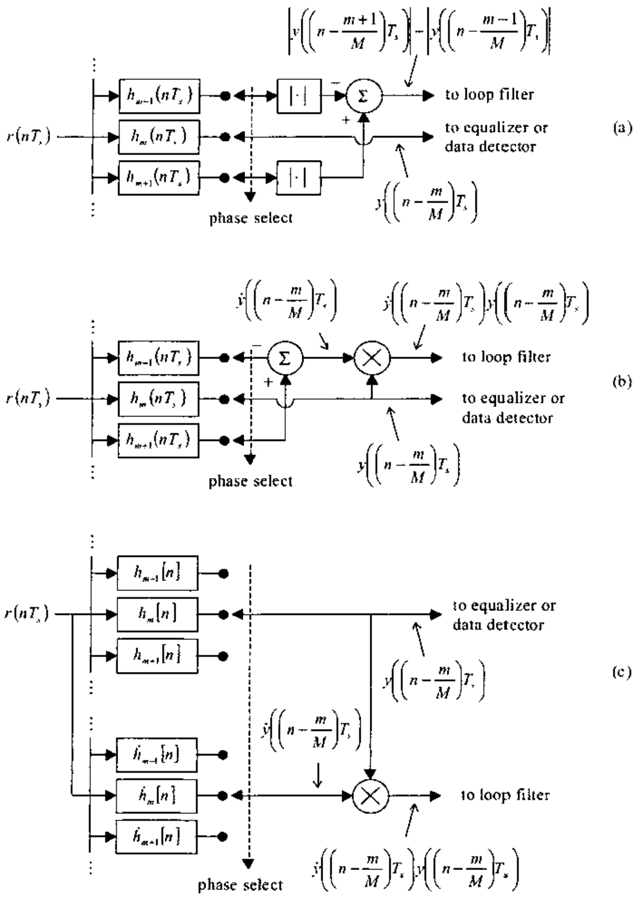

$\left|y\left(\left(n-\frac{m+1}{M}\right)T_s\right)\right|-\left|y\left(\left(n-\frac{m-1}{M}\right)T_s\right)\right|$
$r(nT_s)$
$h_{m-1}(nT_s)$
$h_m(nT_s)$
$h_{m+1}(nT_s)$
|·|
Σ
to loop filter
to equalizer or data detector
$y\left(\left(n-\frac{m}{M}\right)T_s\right)$
phase select
(a)
$\dot{y}\left(\left(n-\frac{m}{M}\right)T_s\right)$
$y\left(\left(n-\frac{m}{M}\right)T_s\right)\dot{y}\left(\left(n-\frac{m}{M}\right)T_s\right)$
(b)
$h_{m-1}[n]$
$h_m[n]$
$h_{m+1}[n]$
$\dot{h}_{m-1}[n]$
$\dot{h}_m[n]$
$\dot{h}_{m+1}[n]$
(c)

Fig. 5. Timing phase error detectors using a polyphase filter bank. (a) the early-late gate approximation; (b) ML phase error using two filters to compute the derivative (first central difference); (c) ML phase error using derivative matched filter.

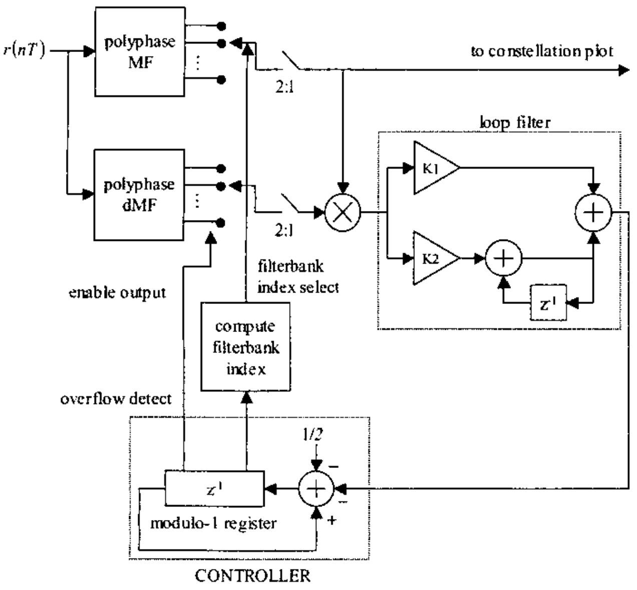


Fig. 6. Complete symbol timing synchronizing PLL operating at 2 samples/symbol using polyphase filterbanks.

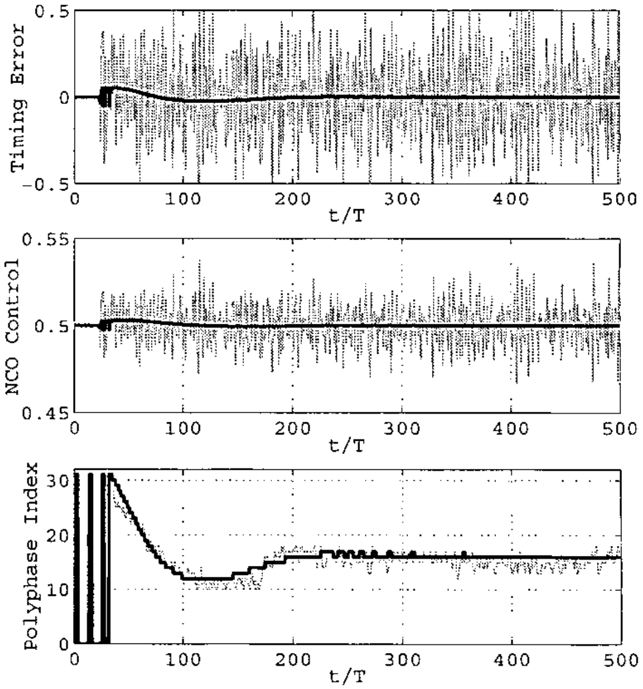


Fig. 7. Simulation results: Phase step response for a timing synchronization loop operating at 2 samples/symbol with a ML timing error detector and a 32-stage polyphase matched filter and derivative matched filter.

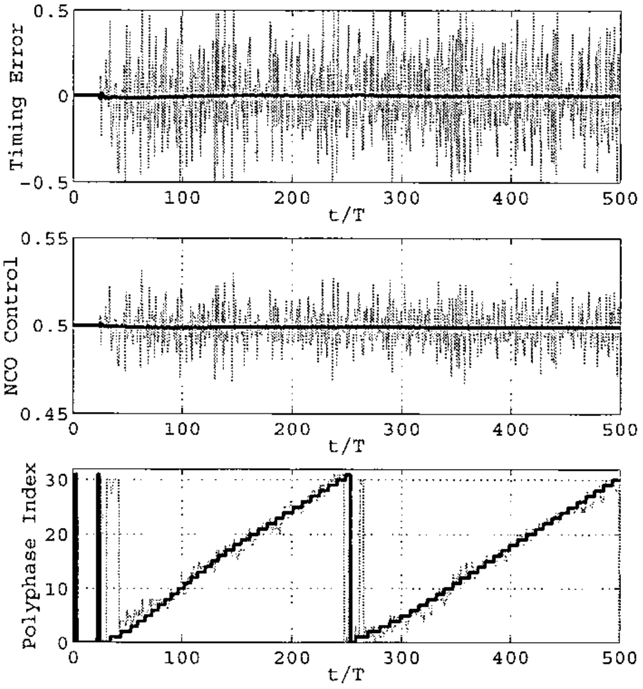


Fig. 8. Simulation results: Frequency step response for a timing synchronization loop operating at 2 samples/symbol with a ML timing error detector and a 32-stage polyphase matched filter and derivative matched filter.