Timing Recovery Algorithm Options for Modems with Rectangle Shaping Filters

fred harris (fred.harris@sdsu.edu),[1] Richard Bell (rcbell@eng.ucsd.edu)[2],
Vamsi Krishna (vamsi765@gmail.com)[1]
[1]San Diego State University, San Diego, CA, USA
[2]University of California, San Diego, CA, USA

ABSTRACT

Symbol timing synchronization is the process of identifying the sample position aligned with the peak value of a matched filter's or correlator's output waveform. Many signaling waveforms are continuous and for those signals, the correlation waveform amplitude and derivative are also continuous. For these signals the location of the correlator peak is identified by its zero derivative. Timing recovery loops are simple servomechanisms designed to seek the zero derivative. The correlator waveform for the common rectangle signaling shape does not have a continuous derivative at its peak value and thus does not have a zero valued derivative at its peak. In this paper we examine and modify the standard timing loop to enable its operation with the rectangle signaling wave shape.

I. INTRODUCTION

Over the recent past, digital signal processing has taken a lead role in implementing the baseband signal processing tasks in consumer and commercial DSP based radios. This proves to be so in the Software Defined Radio community and in particular true in the GNU radio development environment. One area that has benefited by the DSP insertion is the symbol timing synchronization process. This process identifies the sample location corresponding to the correlator peak. For many signaling wave shapes the correlator amplitude and derivative are continuous at its peak. For these signals the peak is identified by its zero derivative. The maximum likelihood timing estimator identifies the peak by locating its zero derivative.

Two common implementation techniques have been developed for timing synchronization. The first entails a single matched filter typically operating at 2-samples per symbol and a polyphase interpolating filter which resamples the input series to the filter to align the output sample of the filter with the maximum eye opening time position. The second entails a polyphase matched filter with each arm of the filter matched to a different offset increment between the data sample position and the maximum eye opening position. The control mechanism to shift the interpolator to its correct sample time offset or to shift the polyphase match filter bank to the filter arm matching the sample time offset is a servomechanism, the phase locked loop, that moves the control pointer to place output time samples at the peak location of the correlator or matched filter. As indicated in Figure 1, we traditionally identify the peak location as the zero derivative location in correlator main lobe. We operate the control system to move towards the zero derivative location.

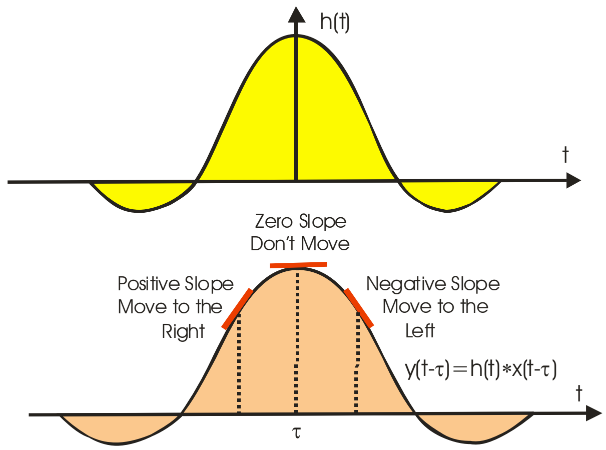

Figure 1. SQRT Nyquist shaping filter impulse response and output of matched filter with derivatives at different time locations and PLL control system response to measured derivative

If we identify the output of the matched filter as y(t), then the maximum likelihood statistic we must minimize is the signal to-noise ratio (SNR) scaled term shown in equation (1).

$$\min_{\text{wrt } \tau} \; E\{\tanh[SNR \cdot y(t\text{-}\tau)] \cdot \dot{y}(t\text{-}\tau)\} \qquad (1)$$

$$\cong \begin{cases} SNR \cdot E\{y(t\text{-}\tau) \cdot \dot{y}(t\text{-}\tau)\}\text{: Low SNR} \\ E\{\text{sgn}(y(t\text{-}\tau)) \cdot \dot{y}(t\text{-}\tau)\}\text{: High SNR} \end{cases} \qquad (2)$$

We recognize that the tanh exhibits two distinct regions of variation with respect to its argument. We call these two regions, the Low SNR and the High SNR regions. In these regions the tanh(x) functions can be approximated by x and by sign(x) respectively so equation (1) can be approximated by the terms in equation (2). Notice incidentally, that the maximum likelihood (ML) estimator drives to zero the $\dot{y}$ term in the $y \cdot \dot{y}$ product while the Gardner Loop, sometimes called the minimum likelihood estimator, drives to zero the y term in the $y \cdot \dot{y}$ product. Tracking the zero crossings is the reason the alternate samples from a matched filter were once referred to as data samples and timing samples respectively. The ML loop requires separate matched filters to estimate y and $\dot{y}$ while the Gardner loop estimates both y and $\dot{y}$ from a single matched filter. While we can track either the peaks or the zero crossings of the eye diagram we prefer to track the peaks because they have lower variance than the zero crossings.

We now address the problem. The rectangle shaping filter doesn't fit the option just discussed. As shown in Figure 2, the matched filter output for a rectangle doesn't have a zero derivative at its peak, and the derivate of the shaping filter isn't very useful, it doesn't have a zero crossing at time 0.

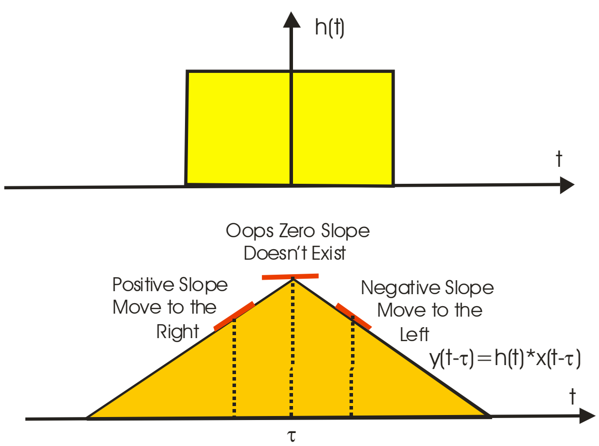

Figure 2. Rectangle shaping filter impulse response and output of matched filter with derivatives at different time locations and PLL control system response to measured derivative

As it happens, there are many modulation formats that use rectangle shaping filters. Frequency Shift Key (FSK), Continuous Phase Modulation (CPM), Frequency Hopping (FH) are some simple examples. We have to revisit the timing recovery mechanism for this class of modulation signals.

II. RECTANGLE SHAPING FILTERS

Our first realization about the rectangle shaping filter is that while we want the derivative of the matched filter's output we can't use the derivative of the matched filter to determine the desired derivative. We must go elsewhere! Where? Well oddly enough to the output of the matched filter. In the old analog days as well as in the current DSP days with spread spectrum signals, we form three matched filters, called Early, Prompt, and Late. The difference of the sample values from the early and late filters formed a slope estimate centered on the prompt sample. The sample values at the output time τ from the three rectangle shape matched filter responses E, P, L can be seen in the upper subplot of Figure 3.

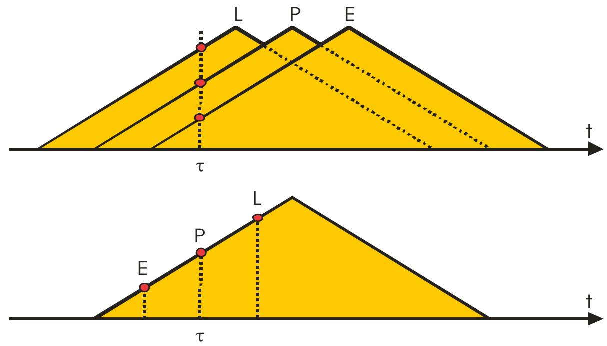

Figure 3. Time response at specific times t_k of three successive time offset rectangle pulse matched filters and time response at three successive time intervals bracketing time τ of a single rectangle pulse matched filter.

As indicated in the lower subplot of Figure 3, these same sample values can be formed from successive time offset samples of a single rectangle shape matched filter response. The rectangle pulse likely contains many samples, for argument's sake, say 32 samples. We can use these successive sample in a simple three tap filter to compute y and $\dot{y}$ in successive time intervals as they are formed by the matched filter. This architecture is shown in Figure 4.

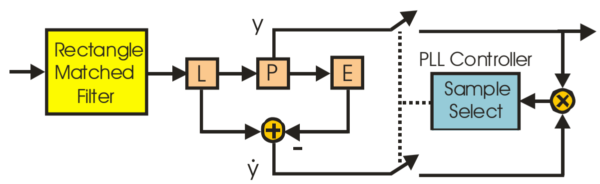

Figure 4. Early-Prompt-Late gates formed from three successive samples output from rectangle matched filter. Product $y \cdot \dot{y}$ controls sample switch to move E and L samples to up-slope and down-slope of matched filter output.

Here the $y \cdot \dot{y}$ product advances the sample switch when its short term average is positive and retards the sample switch when its short time average is negative. For the example being discussed here, the switch would close every 32 samples and the count would occasionally increase by 1 to 33 to advance or would decrease by 1 to 31 to retard the sample positions. The timing recovery loop we describe here acquired timing on a preamble formed by 200 alternating ones. Figure 5. shows the effect of the $y \cdot \dot{y}$ product being positive and directing the sample counter to move forward towards the matched filter maximum output. The positive $y \cdot \dot{y}$ product increases the content of the phase accumulator and when

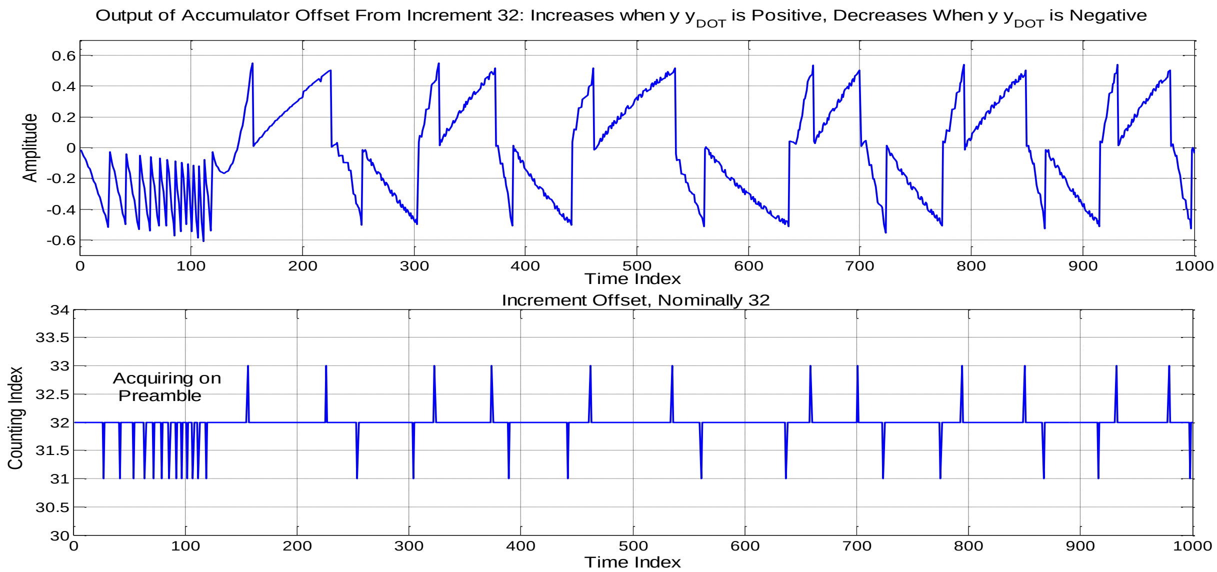

Figure 5. Top subplot: PLL phase accumulator response to $y \cdot \dot{y}$ product formed from three tap delay line of Figure 4. Bottom subplot: sample switch counter, nominally 32 samples, occasionally becomes 33 to move detection sample to down-slope of matched filter response and occasionally becomes 31 to move detection sample to up-slope of matched filter response.

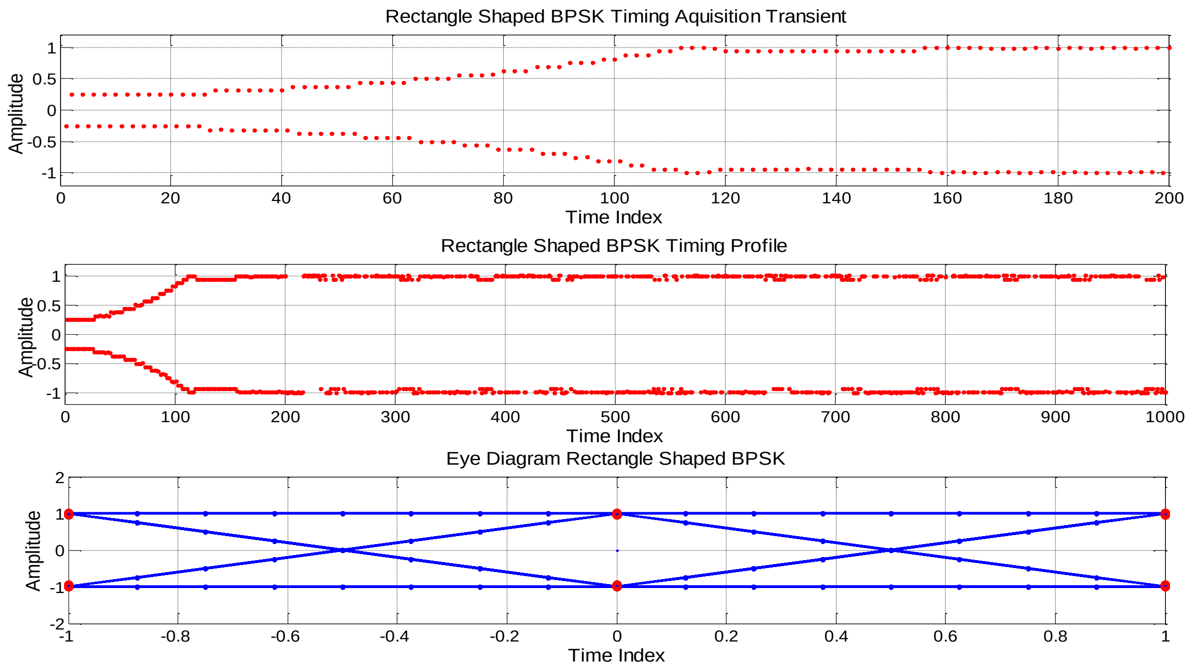

Figure 6. Top subplot: Matched filter sample selection to acquire peak values of rectangle shaping filter matched filter, acquisition during alternate sign preamble. Center subplot: steady state matched filter sample selection showing dithering sample selection process selecting samples either side of peak value of matched filter output. Bottom subplot: eye diagram of sample values extracted from matched filter by Early-Late gate control of PLL phase accumulator

sufficiently large changes the step increment from 32 to 33. This happens twice as seen in the lower subplot of figure 5. As the result of the shifting forward in time the samples now reside past the matched filter peak on the negative slope part of its response. The $y \cdot \dot{y}$ product is now negative and starts reducing the content of the phase accumulator. When the accumulator is sufficiently negative the step increment is reduced from 32 to 31 to bring the detected samples back to the left side of the matched filter peak value. Since there is not a sample location for which $y \cdot \dot{y}$ the product is exactly zero, the dithering of the sample position, first to the left of the peak and then to right of peak will be a periodic behavior of this type of timing recovery loop. Figure 6 shows the sample values taken from the matched filter output during the preamble acquisition and in steady state where we can see the small amplitude increments as the loop dithers to the left and right side of the peak amplitude. Since there are 32 samples per symbol in this example, the amplitude dither will always be less than 3% of the nominal peak to peak amplitude. We also see the eye diagram of the samples formed at symbol rate and can see the small dither increments related to the sample increment dither.

III- TIMING RECOVERY WITH INTERPOLATOR

In the previous section we relied on the large number of samples per rectangle shaped pulse to simply select the sample closest to the peak value of the matched filter output. This works fine when the number of samples is large, in particular for 32 samples per symbol. What would happen if we only had 5-sample per symbol? The granularity in time increment would not be sufficient for timing recovery. We modify the design of the previous section to include improved timing resolution with the aid of an interpolator. This alternate design option is shown in Figure 7. The Interpolate contains 32 3-tap sets of weights. Here the $y \cdot \dot{y}$ product interpolates between selected sample points output by the matched filter and only shift sample point increments upon overflow or underflow of the interpolator pointer. The interpolator can also be used to suppress the index dither we noted as being characteristic of the $y \cdot \dot{y}$ product control of counting increment between output samples. Figure 8 shows the 32 sets of 3-tap interpolator coefficients formed by 32-to-1 polyphase partition of a 96-tap prototype interpolator filter. Figure 9 shows the 3-tap interpolated response to selected input offsets as well as the interpolated output for the full 5-tap rectangle matched filter. Here we see the smooth continuous peak response formed by the interpolator.

Figure 10 shows the effect of the $y \cdot \dot{y}$ product being positive and directing the phase accumulator to increase. As it increases it is interpolating the interval between samples till it overflows. Upon overflow the increment counter must change from 32 sample count to 33 sample count and then return to 32. This is seen to happen three times during signal acquisition on the preamble. We note that the $y \cdot \dot{y}$ product has gone to near zero by this time. There is one additional counting index to 33 and then to 31 and the counter no longer changes. The dither in the earlier design has been replaced with shifts to different interpolating filter in the 32-path interpolator.

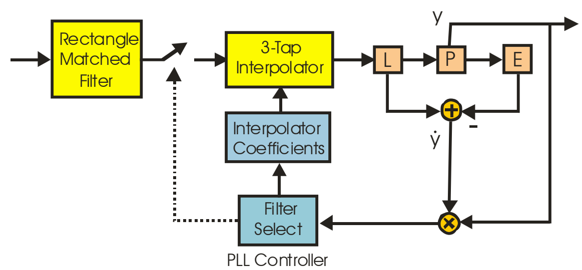

Figure 7. Timing recovery aided by interpolator to improve time increment resolution

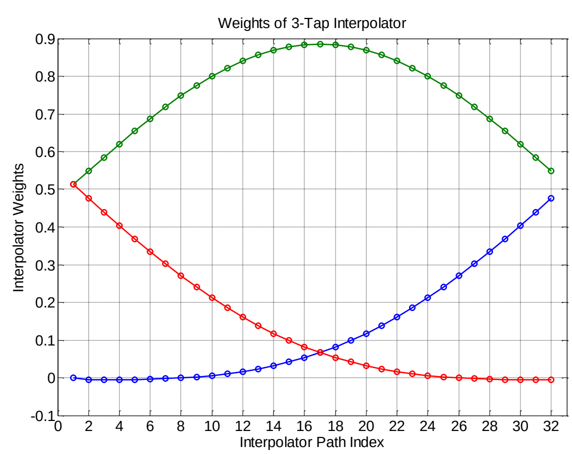

Figure 8. Interpolator weights, 32-sets of 3-taps per path.

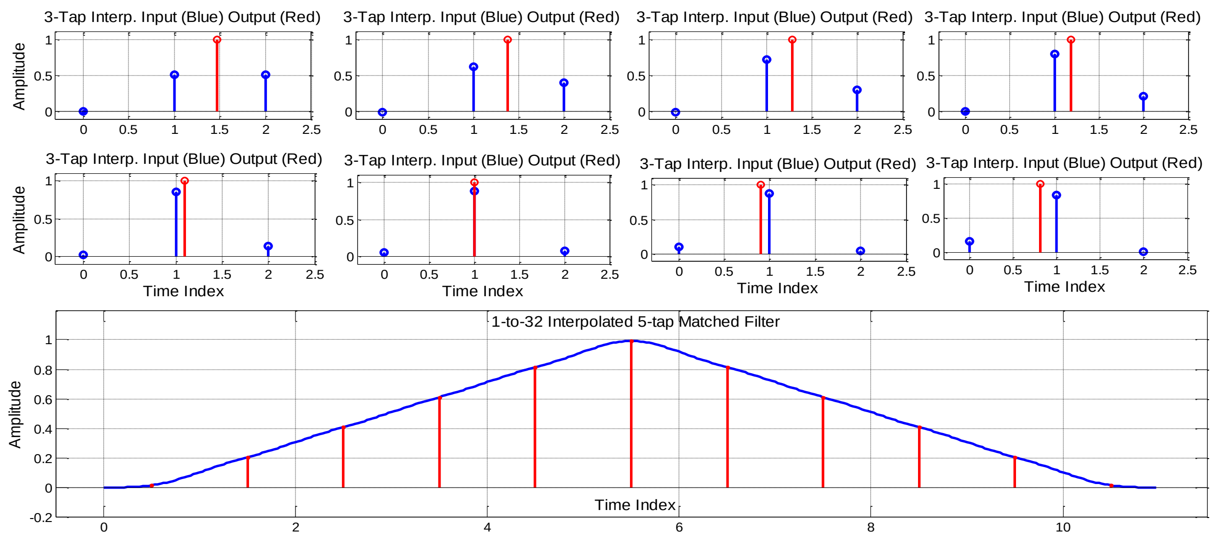

Figure 9. Interpolator response for three tap equalizer: different offsets and interpolated correlator of 5-tap rectangle matched filter.

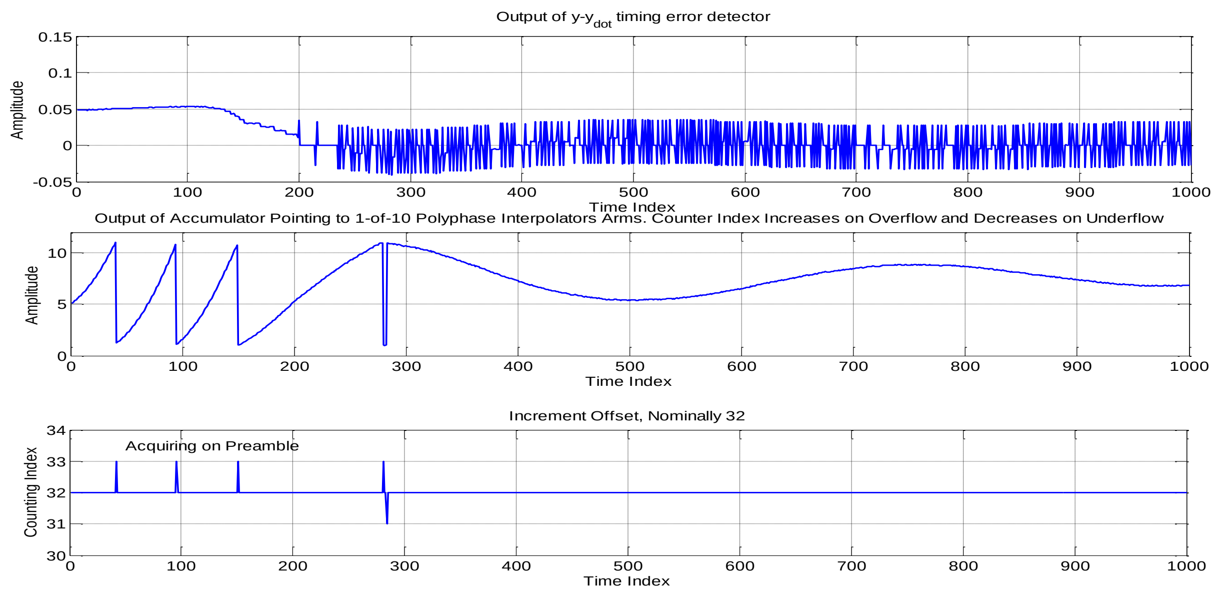

Figure 10. Top Subplot: $y \cdot \dot{y}$ product feeding PLL phase accumulator. The positive values cause phase accumulator overflow, seen in the center subplot, which invokes a counter change from 32 to 33 as seen in the bottom subplot.

IV CLOSING COMMENTS

We have reviewed the timing recover process of a modem using rectangle shaped symbols and described the problem seeking the peak of the matched filter using the standard $y \cdot \dot{y}$ PLL. We retreated to the classic early-late gate version of the timing loop and showed the loop hunting caused by toggling either side of the correlation peak. We then inserted a 3-tap 32-path interpolator that formed a continuous derivative interpolated version of the matched filter output which suppressed the hunting in the loop response.

V REFERENCES

[1] F.M. Gardner, "A BPSK/QPSK Timing Error Detector for Sampled data Receivers", IEEE Trans on Communications, vol. COM-34, no. 5, pp. 423-429, 1986

[2] fred harris, Multirate Signal Processing for Communication Systems, Prentice Hall