

Practical Costas loop design

Designing a simple and inexpensive BPSK Costas loop carrier recovery circuit.

By Jeff Feigin

Binary phase shift keying (BPSK), in terms of noise immunity per unit bandwidth, is one of the most efficient binary data modulation techniques.

Yet, communications systems designers often neglect this option because the design of a BPSK demodulator is not as mathematically simple or straightforward as frequency shift keying (FSK). The prospect of having to apply thorough engineer-

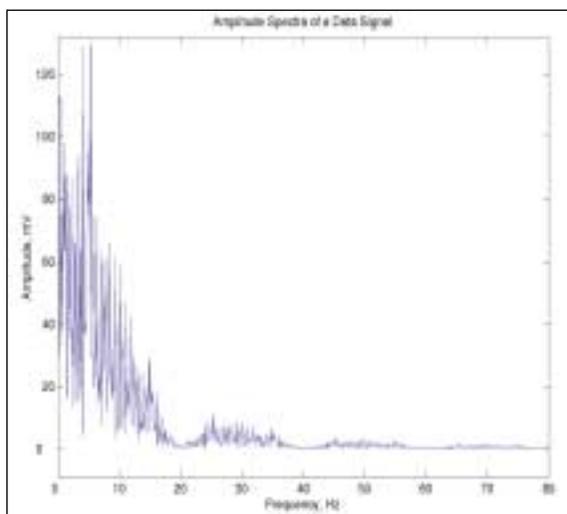


Figure 1. Amplitude spectra of a typical binary data signal.

ing rigor to the design of a BPSK demodulator can be daunting. However, it is unlikely that any such circuit will perform as well as it could if it were implemented without fully understanding and parameterizing its behavior.

Designing and implementing a Costas-loop carrier recovery circuit and demodulator can be done simply and inexpensively using only basic components.

BPSK Background

Simple BPSK modulation is the process of shifting a carrier's phase by 180° for one data symbol while not shifting it for the other — known as

'antipodal' phase shift modulation. The mathematical equation for this process is:

$$BPSK_N(t) = \cos\left(2\pi f_c t + DATA_N(t) \cdot \frac{\pi}{2}\right) \quad (1)$$

where $DATA_N$ is restricted to ± 1 and N is advanced at a much lower rate than the frequency of the carrier (the cosine function). Shifting the phase of a carrier (a sinusoid) by 180° is the same mathematical process as reversing the magnitude of a carrier for one symbol and not the other. With identical results, the following amplitude modulation process can be substituted, interchangeably:

$$BPSK_N(t) = DATA_N(t) \cdot \cos(2\pi f_c t) \quad (2)$$

Modulation theory

The modulation techniques in Equation 1 and Equation 2 are referred to as BPSK and double side-band, suppressed carrier-amplitude modulation (DSBSC-AM), respectively, and when the phase shift is restricted to 180° between opposing symbols, there is no difference.

As with the DSBSC-AM, the resultant BPSK RF spectra are simply the baseband spectra mirrored by the carrier frequency (see Figures 1, 2). The upper sideband (the half of the BPSK spectra that exists above the carrier) is identical to that of the modulating signal, except shifted up to where the carrier frequency was the DC point in the spectra of the original signal. The lower sideband (similarly, the part of the modulated signal that exists below the carrier) contains identical information to the upper sideband, except its spectra is a mirror image of the carrier.

A mathematically simple demodulation scheme multiplies the incoming RF signal by a coherent carrier (a carrier that is identical in frequency and phase to the carrier that originally modulated the BPSK signal). This is an application of the following trigonometric identity:

$$\cos(a) \cdot \cos(b) = \frac{1}{2} [\cos(a+b) + \cos(a-b)] \quad (3)$$

where the product of two cosine functions is the sum and the difference of the inner term of each. When two cosine functions representing periodic time-domain waveforms are multiplied together, the result is two new cosines; the sum of the two frequencies and the difference. Therefore, when the BPSK signal is multiplied by a cosine function identical to the one that modulated it, the original modulating data, plus the same BPSK signal at twice the carrier frequency, are produced. This is mathematically represented by:

$$BPSK_N(t) \cdot \cos(2\pi f_c t) = DATA_N(t) \cdot \cos(2\pi f_c t) \cdot \cos(2\pi f_c t) \quad (4)$$

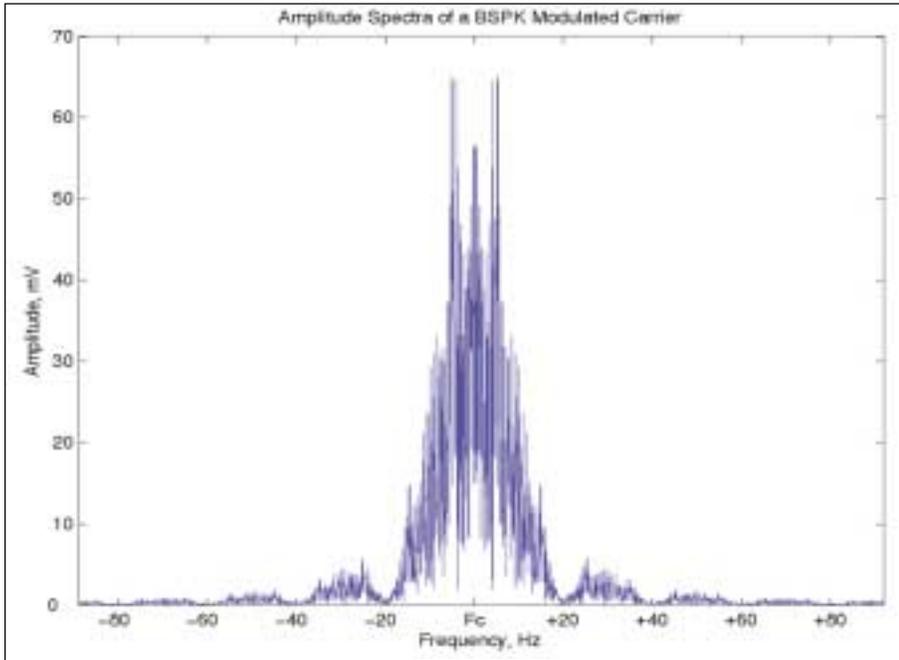


Figure 2. Amplitude spectra of a BPSK modulated carrier .

Applying the trigonometric identity from Equation 3, the result becomes:

$$\frac{1}{2} \left[\frac{DATA_N(t) \cdot \cos(0) + BPSK_N(t)}{\cos(2\pi ft)} \right] \quad (5)$$

Now, considering that the cosine of zero is one, the result of this multiplication is the modulating signal plus the BPSK signal shifted to twice its original frequency:

$$\frac{1}{2} [DATA_N(t) + BPSK_N(t) \cdot \cos(2\pi ft)] \quad (6)$$

A “brick wall” filter (an ideal low-pass filter) then isolates the demodulated data or “low side” product from the extraneous high-frequency or “high-side” product:

$$LPF \left[\frac{1}{2} [DATA_N(t) + BPSK_N(t) \cdot \cos(2\pi ft)] \right] = \frac{1}{2} DATA_N(t) \quad (7)$$

Equivalently, the block diagram for this mathematical operation is depicted in Figure 3.

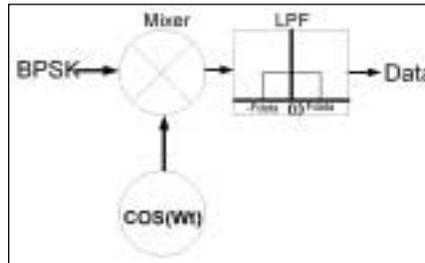


Figure 3. Block diagram of an ideal coherent demodulator .

Demodulation theory

On demodulation, the upper and lower sidebands, which are mirror images, will “fold” onto one another. The two sidebands of the modulating signal will coherently add while the random channel noise (in which the upper and lower sidebands are completely independent) will randomly add. The fact that the two identical modulation sidebands coherently add while the noise, which occupies the

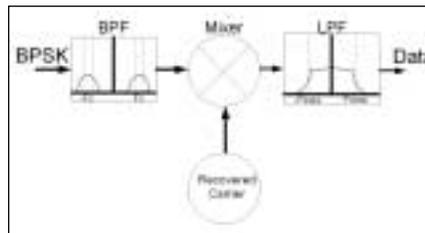


Figure 4. Block diagram of a more easily realized coherent receiver.

same bandwidth, adds according to a root mean squared (RMS) relationship means that the demodulated data will have an inherent signal-to-noise ratio (SNR) advantage, or “processing gain” of 3 dB above that of the BPSK signal.

The narrower the bandwidth of the data filter, the less noise will appear at its output. Too narrow of a filter will limit demodulated data output levels. For optimum performance, the ratio between signal and noise should be maximized. According to Nyquist’s first criteria, so as not interfere with the data signal at the center of each symbol period (the instant where the symbol value is decided), a square-shaped “brick wall” filter should have a bandwidth of no less than half the symbol rate. Such a channel filter must have unity response between DC and a frequency equal to half the symbol rate. This is the ideal channel filter, because it removes the most noise possible without reducing the amplitude of the desired signal at its sampling instants. It will produce an output SNR that is 3dB greater than that of the BPSK signal. Note that Nyquist specifies alternate filters, vestigial spectrum filter shapes such as the raised-cosine, which will achieve the same goal.

Practical BPSK demodulation

The previous mathematical description explains the principle behind coherent BPSK demodulation. Such a structure is straightforward and lends itself well to understanding the concepts. However, it is difficult to implement this circuit because its performance is poor when non-ideal components are used. With some modification (see Figure 4), the demodulator circuit becomes much more practical. The requirements placed on the building blocks of the structure become far less demanding to achieve good performance.

The more realizable BPSK demodulator is based on a practical mixer that is allowed to be imperfect. Such a mixer can be built using common semiconductor devices with small current requirements. Unlike an ideal multiplier, easily implementable mixers are subject to overloading and high-order non-linearities; undesired radio signals, which need not exist at even similar frequencies, can mix in a complicated manner to produce undesired interference that superimposes the desired demodulation product. The solution is to place a chan-

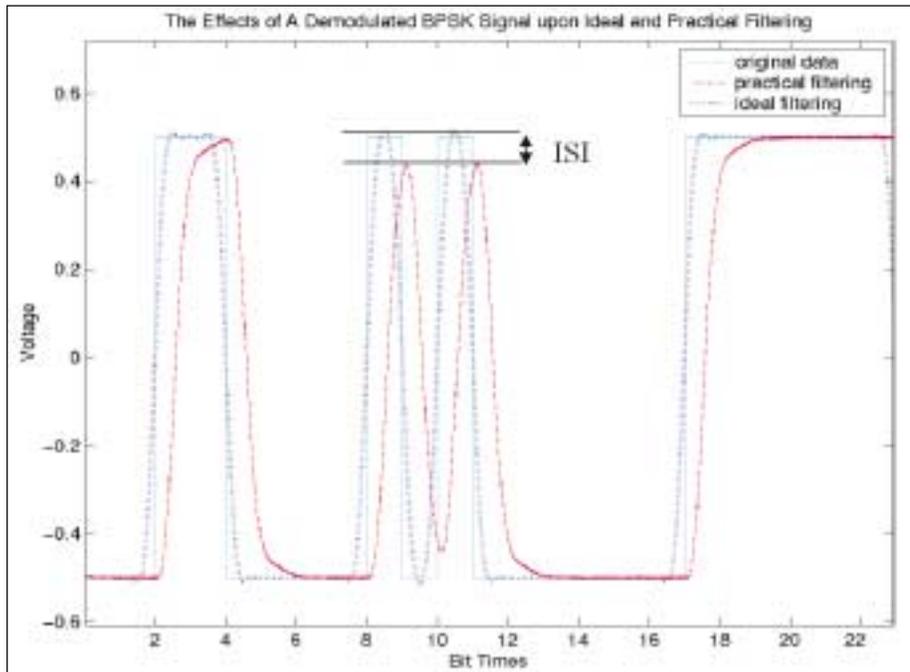


Figure 5. Resultant data waveform upon ideal vs. practical filtering in a demodulator.

nel filter before the mixer to preclude as much off-channel energy from reaching the mixer as possible. The most common inexpensive channel filters are sec-

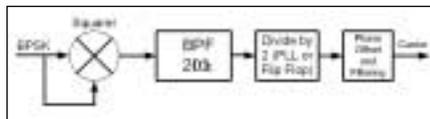


Figure 6. A square-then-divide carrier recovery circuit.

ond- and third-order surface acoustic wave (SAW) and ceramic type. However, neither type exhibits ideal rectangular characteristics.

Low-pass filter discussions

The low-pass filter, as with the ideal demodulator, is the data filter. Because the bandpass filter, ahead of the mixer, removes a great deal of unwanted noise, the low-pass filter requirements can be relaxed. While the perfect demodulator requires that the filter exhibit no intersymbol interference (a Nyquist filter), the practical design may use filters that trade cost, complexity, and size for some degree of SNR degradation.

For the purpose of a simple implementation, a three-pole Butterworth is used as the bandpass filter and a single-pole RC is used as the low-pass data filter for this analysis. The single-pole low-pass filter has a far more grad-

ual roll-off characteristic than ideal — about -20 dB per frequency decade. Therefore, it will not be possible to minimize intersymbol interference with

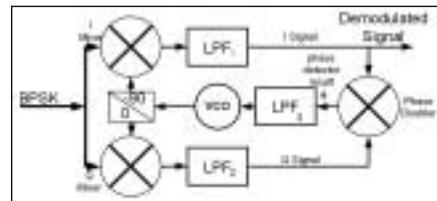


Figure 7. A Costas loop carrier recovery circuit.

allowing extra noise through; its -3 dB cutoff point must be defined such that it maximizes the SNR of the data signal. An optimization means that its

bandwidth must be wide enough to minimize ISI, while narrow enough to minimize noise. For an alternating (1,0,1,0,1...) data pattern, a -3 dB cutoff frequency equal to half the symbol rate will maximize SNR for the single-pole RC low-pass. The total SNR degradation, shown by simulation, is about -1.2 dB for a worst-case alternating data pattern, but found to be about -0.6 dB in the case of a random data pattern. This amount of degradation is acceptable for simple designs, but better filters are recommended if one wishes to improve the SNR.

Figure 5 is a comparison of the resultant demodulated BPSK data on ideal and practical demodulation. Ideal demodulation is that of Figure 3 where the “perfect filter” is implemented as a 10,001-tap raised cosine finite-impulse response (FIR) and the practical demodulator, from Figure 4, uses an infinite impulse response (IIR) three-pole Butterworth as the channel filter and a one-pole IIR structure as the data filter. Although the ideal demodulator does not faithfully reproduce the original signal, it does reproduce the entire signal to reach its peak value at the data-sampling instant. This is the only critical point, according to Nyquist (this is an acausal implementation). However, the result obtained using the more practical structure requires more than one bit time to reach its maximum level. This is the ISI degradation parameter. More noise than ideal would have to be allowed through the demodulator if one were to attempt to solve this problem with non-ideal filters.

Finally, the carrier must be recovered. Its frequency and phase needs to be exactly reproduced to optimally demodulate the BPSK signal. Unless there exists some connection or infor-

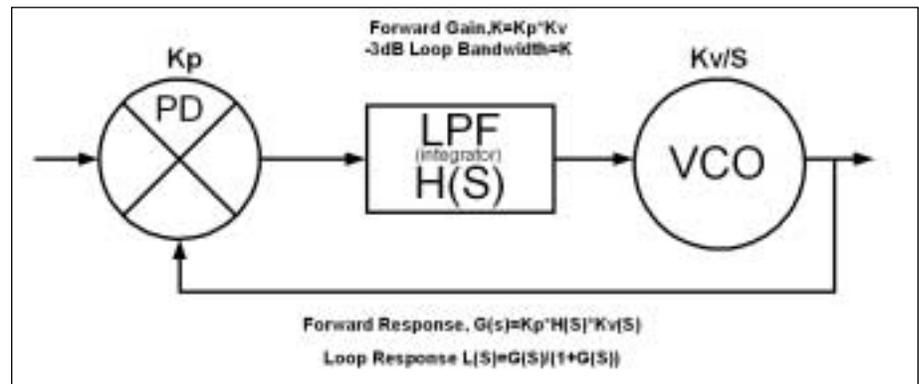


Figure 8. The second-order PLL.

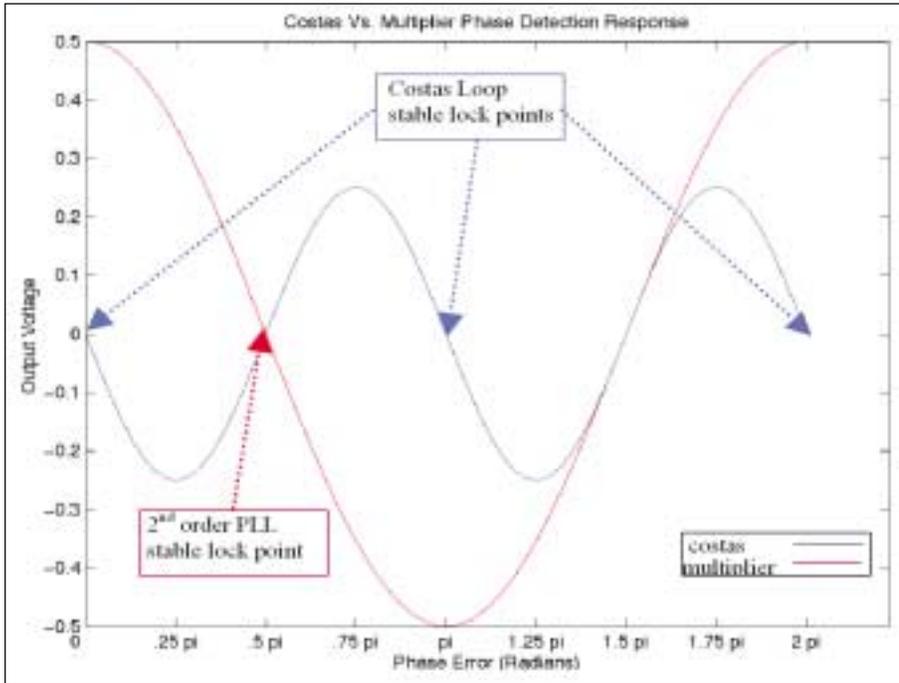


Figure 9. The output-input phase detection characteristics of multiplier and Costas-type phase detection.

mation path between the carrier that was used in modulation and the demodulator, a carrier recovery circuit is required for coherent demodulation.

Carrier recovery

The two common methods for BPSK carrier recovery are: 1) squaring the BPSK signal then dividing by two and 2) the 180° Costas loop. The first technique relies on the fact that, because

the BPSK modulation causes $\pm 180^\circ$ phase transitions, its second harmonic will be phase-modulated by an ambiguous $\pm 360^\circ$. The second harmonic is an unmodulated carrier at twice the frequency. Dividing this second harmonic by two will result in a theoretically phase-coherent carrier. The advantage of the squaring-then-divide circuit is that it is mathematically simple to analyze. However, in prac-

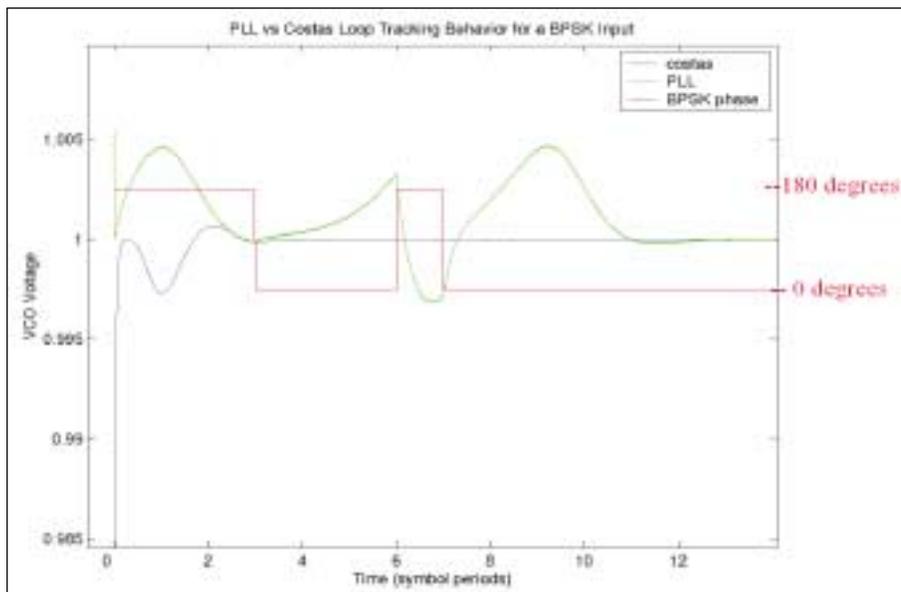


Figure 10. VCO tracking behaviors of a Costas loop and PLL with a BPSK reference input.

tice, controlling the phase offset will be somewhat complicated and layout-dependent; the recovered carrier takes a different path from the demodulator path, and this creates a time differential that will result in a phase error. Also, several filters are required, making it difficult to maintain proper phase over the range operating frequencies.

While the first method is a feed-forward technique, the Costas loop relies on feedback concepts related to the PLL. The Costas loop offers an inherent ability to self-correct the phase (and frequency) of the recovered carrier and, in the end, its implementation is no more complicated than the first technique. Its main disadvantage is involvement of a loop settling time.

Analyzing the Costas Loop

The mechanism of the Costas loop carrier recovery is to iterate its internally generated carrier – the VCO – into the correct phase and frequency based on the principle of coherency and orthogonality. The low-frequency product of a BPSK signal and its coherent carrier is the demodulated information, while the low-frequency component is completely canceled (there will be no low-frequency component at all) in the case of a BPSK signal multiplied by its orthogonal carrier (a carrier that is 90° out of phase with its coherent carrier). The coherent case has already been mathematically demonstrated in Equations 3 through 7. For the orthogonal case, the following trigonometric identity is presented:

$$\begin{aligned} \cos(a) \cdot \sin(b) \\ = \frac{1}{2} [\sin(a-b) + \sin(a+b)] \end{aligned} \quad (8)$$

representing the coherent BPSK carrier at a cosine function and its orthogonal carrier is a sine (or negative sine) function. The time-domain representation of this orthogonal multiplication is:

$$\begin{aligned} BPSK_N(t) \cdot \sin(2\pi f t) = DATA_N(t) \cdot \\ \cos(2\pi f t) \cdot \sin(2\pi f t) \end{aligned} \quad (9)$$

Applying the trigonometric identity from Equation 8, the result becomes:

$$\frac{1}{2} \left[\begin{aligned} &DATA_N(t) \cdot \sin(0) \\ &+ DATA_N(t) \cdot \sin(2\pi f t) \end{aligned} \right] \quad (10)$$

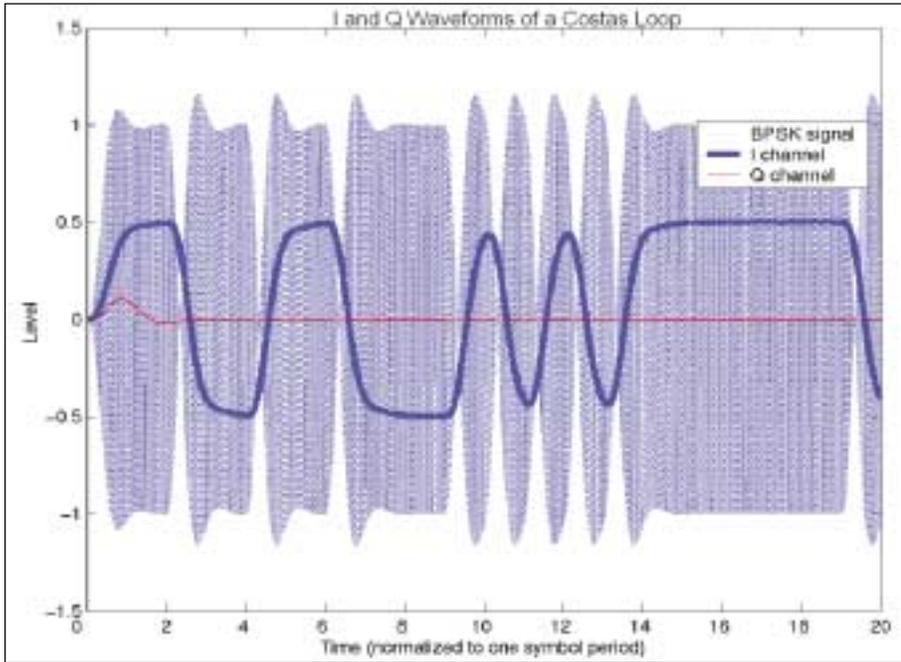


Figure 11. Costas loop simulation with a noise-free, band-limited BPSK input.

Now, considering that the sine of zero is zero, the product of this multiplication is only a “high side” component and the BPSK signal shifted by 90° and to a frequency twice that of what it was.

$$\frac{1}{2} DATA_N(t) \cdot \sin(2\pi ft) \quad (11)$$

Next, a low-pass filter removes the high-frequency component, and nothing remains:

$$LPF \left[\frac{1}{2} \cdot [DATA_N(t) \cdot \sin(2\pi ft)] \right] = 0 \quad (12)$$

The Costas loop is “locked” when it has adjusted its VCO phase and frequency (the initial conditions are random) until the ‘I’ signal is a maximum and the ‘Q’ signal is zero (in reality, the locked-loop ‘Q’ signal is close to zero, but not exactly zero). The third multiplier, the phase doubler, produces the product of the ‘I’ and ‘Q’ signals that sets the VCO input voltage. LPF₃’s purpose is only to remove spurious components and LPF₁/LPF₂ “high side” leakage — it is not meant to significantly contribute to the loop response and is often omitted in theoretical Costas loops block diagrams. LPF₁ not only serves the purpose of a data filter, but in combination with LPF₂ (these two

should be equal to avoid imbalances that will prolong settling time), it comprises a pseudo-integrator (a low-pass filter is related to an integrator). This allows the circuit to behave in a somewhat similar fashion as a second-order PLL (see Figure 8).

Carriers of interest

The carrier that is to become coherent when the loop settles is represented as a cosine function with some phase error. Therefore, the orthogonal carrier that leads the coherent carrier by 90° must be a negative sine function with the same phase error. Considering the incoming BPSK signal as a cosine with zero phase offset relative to time zero, a radial frequency of ω_{BPSK} , (the radial frequency is 2π times the periodic frequency) and the Costas loop VCO frequency to be ω_{VCO} with a phase error relative to the BPSK carrier of θ_{phase_error} , the resultant product of the ‘I’ mixer is represented by:

$$\begin{aligned} I_Mixer_Output &= \cos(\omega_{VCO}t + \theta_{phase_error}) \\ &\bullet BPSK_N(t) = \cos(\omega_{VCO}t + \theta_{phase_error}) \\ &\bullet DATA_N(t) \bullet \cos(\omega_{BPSK}t) \end{aligned} \quad (13)$$

For analysis purposes, because the modulating signal is binary data that reverses its magnitude, DATAN(t) is replaced by ± 1 and the identity of Equation 3 is applied:

$$\pm \frac{1}{2} \left[\begin{aligned} &\cos((\omega_{VCO} - \omega_{BPSK})t + \theta_{phase_error}) + \\ &\cos((\omega_{VCO} + \omega_{BPSK})t + \theta_{phase_error}) \end{aligned} \right] \quad (14)$$

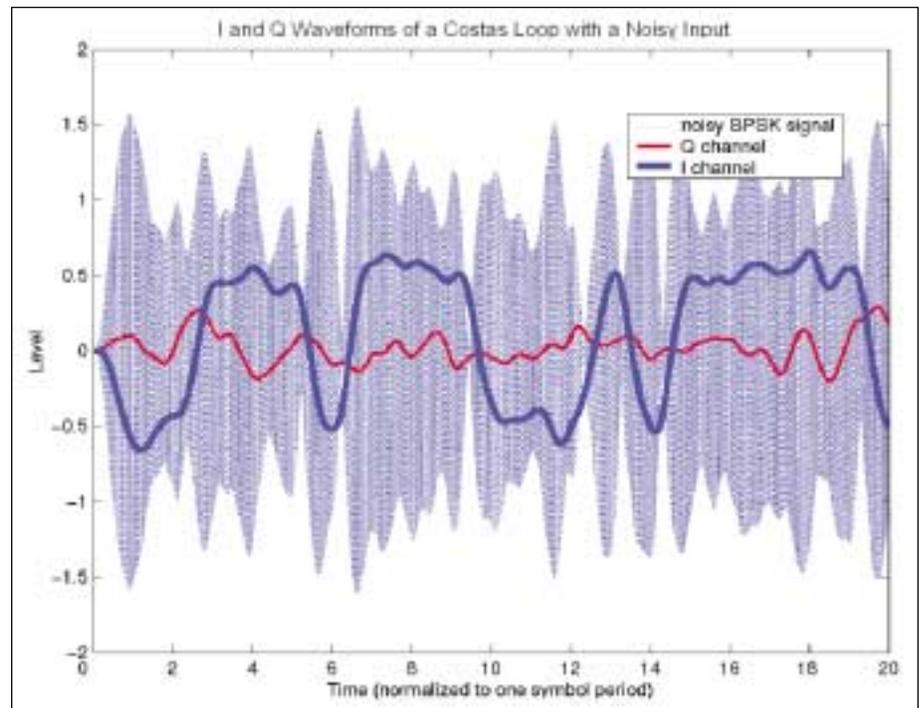


Figure 12. Costas loop simulation with a noisy, band-limited BPSK input.

LPF₁ removes the “high side” component and its output; the ‘I’ signal is represented as:

$$LPF_1(t) = \pm \frac{1}{2} \cos\left(\begin{matrix} (\omega_{vco} - \omega_{bpsk})t \\ +\theta_{phase_error} \end{matrix}\right) \quad (15)$$

Similarly, the ‘Q’ mixer produces the following product:

$$\begin{aligned} Q_Mixer_Output \\ = \sin(\omega_{vco}t + \theta_{phase_error}) \bullet BPSK_N(t) \quad (16) \\ = -\sin(\omega_{vco}t + \theta_{phase_error}) \bullet DATA_N(t) \end{aligned}$$

Applying Equation 8, and again substituting DATA_N(t) with ±1, the resultant ‘Q’ product is shown as:

$$\pm \frac{1}{2} \left[\begin{matrix} \sin((\omega_{vco} - \omega_{bpsk})t + \theta_{phase_error}) \\ \sin((\omega_{vco} + \omega_{bpsk})t + \vartheta_{phase_error}) \end{matrix} \right] \quad (17)$$

LPF₂ removes the “high side” component and its output; the ‘Q’ signal is represented as:

$$LPF_2(t) = \pm \frac{1}{2} \sin\left(\begin{matrix} (\omega_{vco} - \omega_{bpsk})t \\ +\theta_{phase_error} \end{matrix}\right) \quad (18)$$

Then, multiplying these two LPF results together, the phase doubler produces:

$$\begin{aligned} Phase_doubler(t) &= LPF_1 \bullet LPF_2 \\ &= \left[\pm \frac{1}{2} \cos((\omega_{vco} - \omega_{bpsk})t + \theta_{phase_error}) \right] \\ &\bullet \left[\pm \frac{1}{2} \sin((\omega_{vco} - \omega_{bpsk})t + \theta_{phase_error}) \right] \end{aligned} \quad (19)$$

Next, applying Equation 8:

$$= -\frac{1}{8} \left[\begin{matrix} \sin(0) \\ +\sin\left(2\left(\begin{matrix} (\omega_{vco} - \omega_{bpsk})t \\ +\theta_{phase_error} \end{matrix}\right)\right) \end{matrix} \right] \quad (20)$$

Then simplifying the output of the phase doubler, the phase detector result becomes:

$$\begin{aligned} Phase_detector_result &= \\ -\frac{1}{8} \sin\left(2\left(\begin{matrix} (\omega_{vco} - \omega_{bpsk})t \\ +\theta_{phase_error} \end{matrix}\right)\right) \end{aligned} \quad (21)$$

Further dissection

The phase detector result is then filtered by LPF₃, which removes extraneous loop products before being applied to the VCO. Again, this filter is not meant to significantly contribute to the Costas loop locking response — its response should be far outside the closed-loop response. From the result of Equation 21, it can be determined that the loop will correct itself, both in terms of frequency and phase.

And, by modifying Equation 21 to represent absolute phase difference (rather than phases that are relative to time zero), the phase detection response is found.

It is important to remember that all three multipliers compose the phase detector response. The phase doubler multiplier is not, by itself, “the phase detector.” In the case where the input signals have a peak value of unity, the phase detection response is described by Equation 22. The phase detector gain vs. amplitude dependency is mentioned for mathematical completeness, but such effects need not be thoroughly quantified because realistic “multiplier” phase detectors will be amplitude invariant.

Picking unity for the input and VCO amplitudes as the parameters for

phase-detector gain serve the purpose of an example gain. The phase-detection response is described by:

$$\begin{aligned} Costas_Phase_Detector &= \\ -\frac{1}{8} \sin(2\phi_{phase_difference}) &\approx K_p = \frac{\frac{1}{2}V}{r} \end{aligned} \quad (22)$$

This result is similar to that of a conventional multiplier-type phase detector whose output, based on a unity amplitude input, is:

$$\begin{aligned} Conventional_Phase_Detector \\ = \cos(\phi_{phase_difference}) &\approx K_p = \frac{\frac{1}{2}V}{r} \end{aligned} \quad (23)$$

Comparing these two results (see Figure 9), the Costas loop phase detection response is a sine function while the multiplier-type phase-detection response is a cosine function of the phase difference. The second-order PLL contains a low-pass filter that integrates (or pseudo-integrates, depending on the type of filter) the error signal

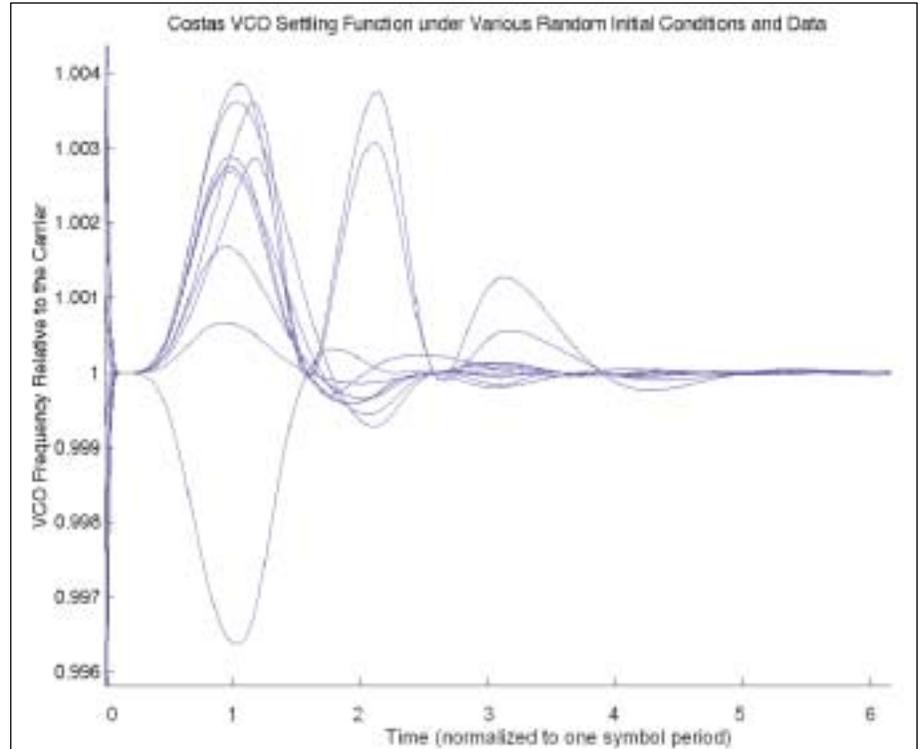


Figure 13. Ten Costas loop settling patterns under identical parameters, but randomized initial conditions and BPSK modulation data.

from the phase detector. The PLL is locked when the phase detector result is zero (near zero when the loop filter is not a true integrator), hence producing a DC constant at the input of the VCO.

The cosine response of the multiplier phase detector causes a lock when the phase error is 90° (because the cosine of 90° is zero).

The Costas loop, considering LPF₁ and LPF₂, acts similarly to a second-order loop (the combined effect of LPF₁ and LPF₂ adds a second pole to the loop response. The filtered ‘Q’ signal moves just slightly above or below zero and is multiplied by the filtered ‘I’ product). Its doubled-sine phase detection response allows two stable locking points: 180° phase error and zero degrees — both produce a redundant output that drives the VCO to the correct phase/frequency. Low-pass filters LPF₁/LPF₂ must pass the modulation (the direct result of filters that are too narrow is ISI) as:

$$2\omega_c \ll \omega_{LP1,2} \geq 2\pi B_M \quad (24)$$

where B_M, the modulation bandwidth, is half the data rate.

Before the loop has settled, whether a PLL or a Costas loop, the phase detection response must be one that, based on the phase relationship between the VCO and the input signal, guides the VCO to a stable locking phase and frequency. If one were to apply a signal whose phase is reversing by 180° to an ordinary PLL, the phase detector result would constantly reverse polarity and the phase error magnitude is unlikely to converge on any stable value (i.e., the PLL will “track” in opposite directions for opposite phases — see Figure 9). One might refer to a conventional phase detector as 360° periodic. This means that the phase of the incoming carrier would have to be modulated with 360° phase transitions (which is no phase transition at all because a sinusoidal carrier has a period of 360°) not to upset the tracking so that the loop error may converge.

Costas vs. conventional

Conversely, the Costas loop phase-detection response is 180° periodic—there are two stable tracking points. BPSK modulation shifts the Costas loop input by 180°, which is the next

period of the phase detection function, where the loop tracking response is identical. Therefore, the Costas loop is able to track a BPSK modulated carrier (loops can also be derived that track higher-order phase modulation schemes such as QPSK). The only catch is that the loop-phase doubled response means that it has a 50% chance of generating an upside-down carrier. Figure 10 displays simulation results of how a Costas loop vs. an ordinary PLL with similar loop parameters would behave with a BPSK signal as an input.

Because LPF₃ is not part of the control loop (and *not* the PLL loop filter), it must not have a frequency response that falls within the loop bandwidth. Its purpose is only to remove the excess noise products produced by the three previous multipliers and two imperfect filters. This filter constitutes an undesired S-plane pole that would cause the loop to oscillate, but if its response is far outside of the loop response, then it will not cause problems. A rule-of-thumb recommendation for a safe, out of the loop, response would be to set the pole of LPF₃ to a minimum of four times that of what the closed loop response would be without this filter.

Exactly how the VCO will settle depends on the initial phase and frequency of the VCO as it relates to the incoming BPSK signal, as well as to the noise characteristics. Although not apparent, the behavior of any practical implementation of this circuit will also be affected by the actual data that has been modulated. Real-world communications are usually band-limited, and the abrupt 180° phase shifts of BPSK, which the Costas loop is immune to, would require an infinite bandwidth.

A more realistic-version BPSK signal is one in which a bit transition will cause the carrier amplitude to slowly sweep from its current phase to the opposite phase through the zero-amplitude point. The phase-detector contribution to loop gain (although realistic phase detectors are not perfect multipliers, they still have minimum input level requirements) is diminished as the input signal level shrinks.

Every BPSK phase transition will cause a Costas loop “dropout” at and near the zero-crossing instant during the interval between the two discrete phase levels. If the loop is still in the locking phase at this point (i.e., when the VCO phase does not match that of

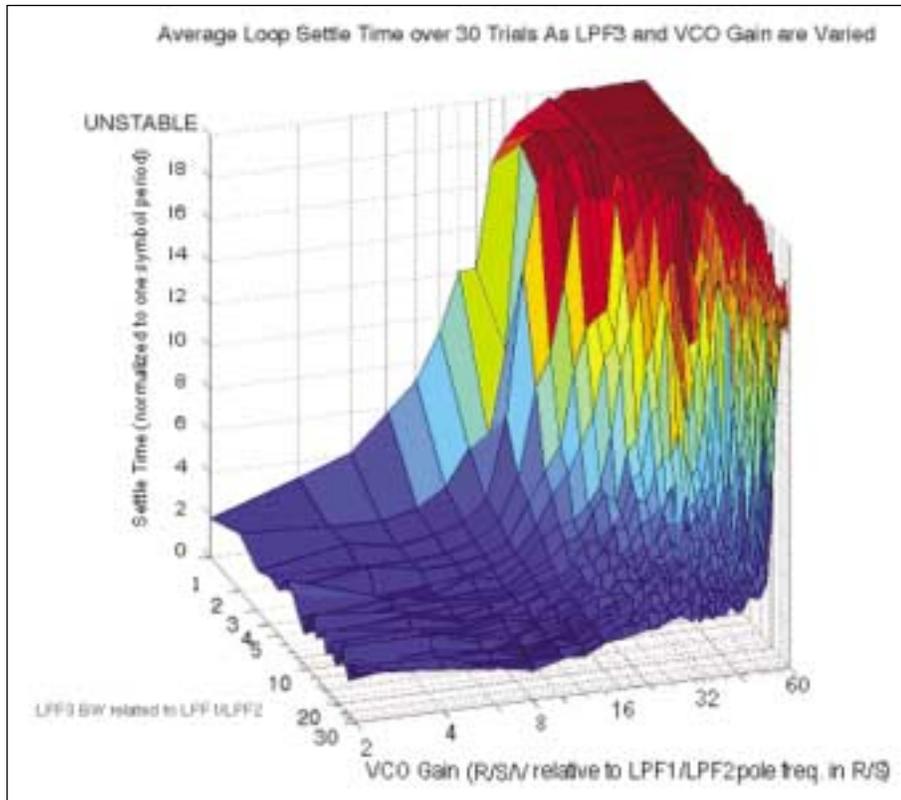


Figure 14. Averaged results of simulation comparing Costas loop settle time to the bandwidth of LPF₃ and VCO gain.

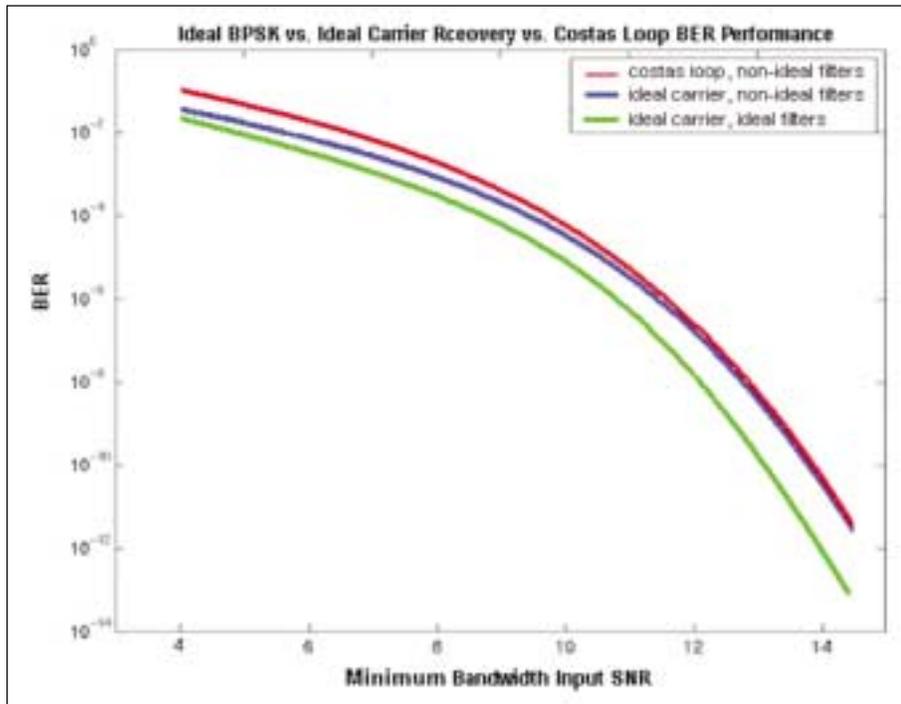


Figure 15. Results (interpolated) of simulation of Costas loop vs. ideal BER performance as input SNR varies.

the carrier), such a “glitch” could allow a phase slippage and may temporarily allow the loop to track in the wrong direction.

Other design issues include the effect of realistic (non-ideal) filters. Some “high side” product will always “leak” through and affect the circuit’s performance; their respective responses will not be identical, and there will be ISI (a 1-0-1-0 pattern will not quite produce 180° phase transitions). Further, it is not realistic to assume that the quadrature components of the VCO will have a perfect 90° offset or that the phase detector is an ideal multiplier free from DC offset. A second-order PLL analysis (where the loop filter is the same as LPF₁/LPF₂) of a carrier will approximate settling characteristics of a Costas loop, but a computer simulation is recommended if the designer needs accurate information. This is because “mathematical” building blocks may need to be substituted with commonly available and inexpensive components. Figures 11 and 12 show the simulated timing waveforms of a Costas loop operating under noise-free and noisy conditions, respectively.

Figure 13 is a plot of the VCO settling function where the loop parameters are identical for each run, but the VCO starting phase/frequency and the

modulation data are randomized over 10 trials. Realistic Costas loop behavior is somewhat chaotic for the reasons mentioned previously, depending on when BPSK phase transitions occur during the lock phase.

Design considerations of Costas loops

Similar to PLL design, the Costas loop design considerations are noise performance, settling time and a reliable lock range. As a demodulator, noise performance is maximized when the least amount of noise is allowed in the loop. This is accomplished by setting the LPF₁/LPF₂ response to their maximum SNR. This corresponds to a -3 dB cutoff equaling half the data rate for a single-pole RC. For loop settling purposes, this cutoff is also the minimum allowable for the loop filter. Additionally, this is an attractive choice because this filter also serves the purpose of a data filter.

The loop gain must now be set. Because LPF₁, one of the two identical legs of the loop filter, serves the dual purpose of also being the data filter and is required to pass BPSK modulation, a compromise has been made. According to [2], the critical damping point (the point where minimum settling time occurs) for a PLL using the same lag-type filter as LPF₁/LPF₂, is when the

pole of this filter equals the closed-loop bandwidth. Based on settling time to a particular threshold, simulation shows that setting the pole of lag-filter to half the DC forward gain results in a quicker lock (this is a point where the loop is slightly under-damped). Therefore, these same parameters were used in the Costas loop design as a starting point for simulation.

The Costas loop phase detector gain, under unity input conditions, is 1/4 V/r, as stated in Equation 22, so the VCO gain is the variable that needs to be determined. Solving for this parameter, the unity output VCO should have a gain of eight times the filter’s pole frequency, in terms of radians per second for a unity BPSK input for this theoretical circuit containing perfect multipliers as the phase detector:

$$VCO_Gain_{critically_damped} = \frac{\omega_{LPF1,LPF2}}{\frac{1}{2}K_P} \quad (25)$$

$$= 8 \bullet \omega_{LPF1,LPF2} r/s/v$$

Simulation confirms this result for the Costas loop (see Figure 14). The fastest achievable settle time is one in which the VCO has a gain of eight times that of the LPF₁/LPF₂ pole frequency with the above the phase-detector gain parameters and a random BPSK input. Using the Costas loop parameters presented here, where the filter poles and loop gain are all in a fixed relationship to the data rate, the regenerated carrier will settle in less than three bit times. Of course, if the phase detector has some gain or gain-function other than that of Equation 22, Equation 25 should be appropriately modified.

LPF₃ must then be specified. This filter should have its pole at a low-enough frequency that the Costas loop will not be too noisy nor be subject to carrier phase reversals in the presence of noise (the Costas loop is equally stable in both phases) while high enough that it doesn’t cause the loop to oscillate. Setting this pole to four times *K* (or eight times the LPF₁/LPF₂ pole) is the point in which this filter will negligibly affect on the loop; simulation shows this (see Figure 14). To be cautious, particularly at lower data rates where such a filter can more easily create problems, a factor of six times *K* (12 times the LPF₁/LPF₂ pole) is a better choice.

The phase detector and VCO gain control the reliable lock range. With the above design parameters, this Costas loop will always reliably lock so long as the VCO can produce the required carrier frequency. This value, the frequency range in which the loop may lock, is determined as follows:

$$Range_{VCO} \approx Gain_{phase_detector} \cdot Gain_{VCO} \quad (26)$$

A more practical limit on how far the carrier recovery circuit may “stretch” from its center frequency, however, depends on the width of the band-pass filter shown in Figure 4. Whatever the track and hold ability of the Costas loop, the carrier recovery range will never reach beyond that of the intermediate frequency (IF) filters.

Finally, noise performance must be considered. While demodulated BPSK (the data, itself) has an SNR that is 3 dB greater than that of the modulated BPSK (not considering ISI degradation due to non-ideal filtering), noise will cause the Costas loop to introduce even more noise of its own.

The reason is that a carrier recovery circuit produces a noisy carrier under noisy conditions. Figure 15 displays the bit error rate (BER) performance of ideal BPSK demodulation (Figure 3) vs. practical demodulation with ideal carrier recovery (Figure 4) vs. Costas loop demodulation, with a random data pattern. It is observed that this Costas loop demodulator performs well at regenerating the carrier until a low SNR input. At a BER of 10^{-4} (a value often specified for minimum system performance), non-ideal filtering contributes a 1.5 dB degradation while Costas non-ideal carrier recovery causes only an additional 0.6 dB demodulator loss. It is clear that, in the overall scheme, single-pole data filtering causes more SNR degradation than a Costas loop.

Implementation discussion

The most difficult structure to implement in a Costas loop, within reasonable cost, complexity, and performance, is a quadrature downconverter. Such a circuit requires that the incoming BPSK signal be split between two mixers (to perform the down-conversion multiplication) and independently multiplied by two signals of identical frequency, but differ in phase by 90° .

The greatest difficulty arises from

producing these two signals as well as maintaining their relative phase offset over a range of frequencies. Fortunately, modern integrated circuits provide an easy-to-use and inexpensive implementation of an I/Q demodulator. Such ICs are also attractive solutions because they provide additional mixers and amplifiers that facilitate the necessary building blocks required to build most of a receiver.

The remaining Costas loop building blocks are three low-pass filters, the “phase doubler” multiplier, and a single-phase VCO. As previously mentioned, all filters in this design are single-pole RC (this is for simplicity—the reader may wish to implement other types of filters for better performance).

One can implement the multiplier and VCO in a number of ways. The use of an integrated op amp-type multiplier and a separate VCO are one possibility. A double-balanced switching-type mixer is also a suitable choice, although its pseudo-multiplier characteristics will somewhat alter the Costas loop characteristics. However, conventional PLL logic gate phase detectors (such as the XOR) or devices that must be operated under heavy saturation (which causes limiting) are unsuitable. Devices that limit the Q-channel, but not the I-channel are perfectly acceptable.

Conclusions

The Costas loop, a cousin of the PLL, is an effective close-loop coherent demodulator. Though the PLL and Costas loop exhibit similar setting characteristics when configured under identical parameters, the latter can lock onto a carrier that is reversing in phase; A Costas loop regenerates the “phantom” BPSK carrier. Prediction of settling behavior may be estimated by appropriately substituting Costas loop parameters into a traditional PLL analysis, but simulation is required to accurately estimate its chaotic behavior.

Based on simulation, Costas loop settling time is minimized when the closed-loop bandwidth is twice that of its constituent RC filter, according to a slightly under-damped condition. It has also been found that a filter after the phase-doubler multiplier (the third multiplier) is effective in reducing loop noise when its response is kept far outside of the closed-loop bandwidth. Finally, referring to Figure 15, although the recovered carrier is noisy

when a Costas loop is presented with a noisy BPSK signal, the ISI created by simple data filters produces the most degradation in a simple Costas loop implementation.

RF

References

- [1] F. M. Gardner, “Phaselock Technique,” Wiley, New York, 1979.
- [2] B.P. Lathi, “Modern Digital and Analog Communications Systems 2nd Edition,” Oxford University Press, New York, 1983.

About the author

Jeffrey Feigin is an RF applications engineer at Analog Devices, where his responsibilities include RF IC applications support, systems-level IC design and reference design. Previously, Feigin was a design engineer with Zo&Co., in Skopje, Macedonia, where he designed microcontroller and RF circuitry for an urban wireless network. Prior to that, he was employed at Lincom Corp. as a contract engineer, where he developed an OPNET model of a TCP/IP over Milstar. In addition, while a research assistant at the Center for Wireless Information Network Studies, Feigin performed QoS analysis of wireless multimedia Internet traffic and protocols. Feigin is a graduate of Worcester Polytechnic Institute with a B.S.E.E. and M.S.E.E. For further information, contact Doug Grant doug.grant@analog.com