10.B APPENDIX: COSTAS LOOP FOR *N* PHASES

A general representation of a loop for carrier recovery with an n-phase signal is shown in Fig. 10.B.1. The biphase Costas loop of Fig. 10.7 is a special case of this loop.

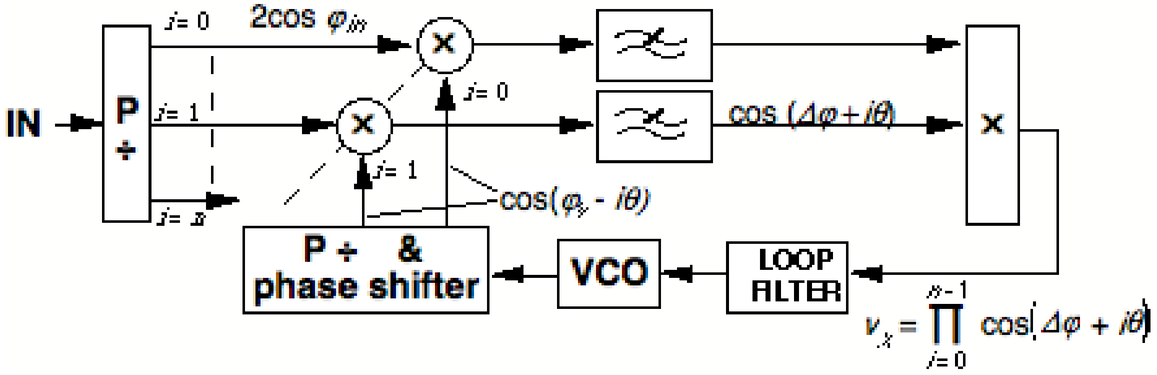


Fig. 10.B.1 *N*-phase Costas Loop

The number of multiplications of the incoming signal and the VCO output is *n*, the number of equally spaced phases (usually 2, 4, or 8) in the code, and the phase shifts given to the *n* VCO outputs are multiples of

$$\theta = \pi/n. \qquad (10.B.1)$$

After low passing, all of the multiplier outputs are multiplied together to produce v_x, as shown. The derivative of v_x with respect to the phase difference is

$$\frac{dv_x}{d\Delta\varphi} = -\left[\prod_{i=0}^{n-1}\cos(\Delta\varphi + i\theta)\right]\sum_{i=0}^{n-1}\frac{\sin(\Delta\varphi + i\theta)}{\cos(\Delta\varphi + i\theta)}\ , \qquad (10.B.2)$$

where, as before,

$$\Delta\varphi \equiv \varphi_{\text{in}} - \varphi_v\ . \qquad (10.B.3)$$

In order to concentrate on regions near the zero crossings of v_x, we define γ as the phase change from such a point, where $i = k$ for that point,

$$\gamma = \Delta\varphi + k\theta - \pi/2\ . \qquad (10.B.4)$$

Partly from William F. Egan, *Phase-Lock Basics*. Reprinted with permission of John Wiley and Sons, Inc.

Then the derivative near such a point is

$$\left.\frac{dv_x}{d\Delta\varphi}\right|_{\gamma\approx 0} \approx \left[\prod_{i=0}^{n-1}\cos\left(\gamma - k\theta + \frac{\pi}{2} + i\theta\right)\right]\frac{\cos\gamma}{\sin\gamma} \tag{10.B.5}$$

$$\approx(-1)^n\left[\prod_{j=-k}^{n-1-k}\sin(\gamma + j\theta)\right]\frac{1}{\sin\gamma} \tag{10.B.6}$$

The approximation of Eq. (10.B.5) is valid because, near $\gamma = 0$, one of the terms in the summation of (10.B.2) approaches infinity; that is the term for which the denominator is sin γ. Even though this term is much larger than the other terms in the summation, v_x does not approaching infinity because the denominator term is identical to a term within the preceding product of terms. Normally n is some power of 2 (2, 4, 8, 16) so the first factor in (10.B.6) is then equal to 1. Therefore, we will drop $(-1)^n$ from Eq. (10.B.6) at this point.

Note that, if $k = 0$, the first term in the product in Eq. (10.B.6) cancels the denominator on the right and the expression can then be written

$$\left.\frac{dv_x}{d\Delta\varphi}\right|_{\gamma=0} \approx \prod_{j=1}^{n-1}\sin(\gamma + j\theta). \tag{10.B.7}$$

If $k = 1$, the second term in the product in (10.B.6) cancels the right term, producing a form similar to (10.B.7) except $\sin[\gamma + (n-1)\theta]$ is replaced by $\sin[\gamma - \theta]$, equivalent to a phase change of $n\theta = \pi$ in that term and resulting in the expression being multiplied by -1. This occurs each time we advance k by 1 so we can write

$$\left.\frac{dv_x}{d\Delta\varphi}\right|_{\gamma\approx 0} \approx (-1)^k\left[\prod_{j=1}^{n-1}\sin(\gamma + j\theta)\right] = \pm C_x. \tag{10.B.8}$$

Thus the same sign occurs at the zero crossing whenever k is advanced by 2, that is for a change in the lock point of $2\theta = 2\pi/n$. For biphase, as we saw in Section 10.3, this is every 180°. For quadraphase there is a lock point each 90° and there is a lock point each 45° for eight phase. Thus changes between the allowed phases in a phase coded signal will not change the tuning voltage and we can lock to the carrier in spite of the lack of a spectral line there.

Just as a decoded biphase signal is available from one of the PDs in Fig. 10.7 (or Fig. 10.B.1 when n=1), so too can decoded QPSK be obtained from two of the four PDs in Fig. 10.B.1 when n=2. These outputs are proportional to the projections of the four possible vectors in Fig. 10.8 on the horizontal and vertical axes respectively. In both cases, which PD or PDs provide the decoded signal(s) is arbitrary so provision must be made for selection or control of the proper output.

Values of C_x are easily computed from (10.B.8) and are shown in Table 10.B for several values of n.

n	C_x
2	1
4	0.5
8	0.066
16	0.00052

Table 10.B Factor in K_p For Several Values of n

10.C APPENDIX: SYMBOL CLOCK RECOVERY

Once a digital signal has been demodulated and changed to binary levels, a clock is needed that is synchronized with the rate of change of the symbols. For a QPSK signal, for example, two binary streams will be generated and the values of the two streams at any given time represent a four-level signal. While each binary stream will change less often than the symbol rate, the basic (highest) frequency in each stream will equal the symbol frequency. A clock at this frequency is required for use in sampling the bit streams. Since the bit patterns of the streams are essentially random, there is no spectral line in these streams at the desired bit rate. As with carrier recovery, one can be created. A circuit for doing this is shown in Fig. 10.C.1.

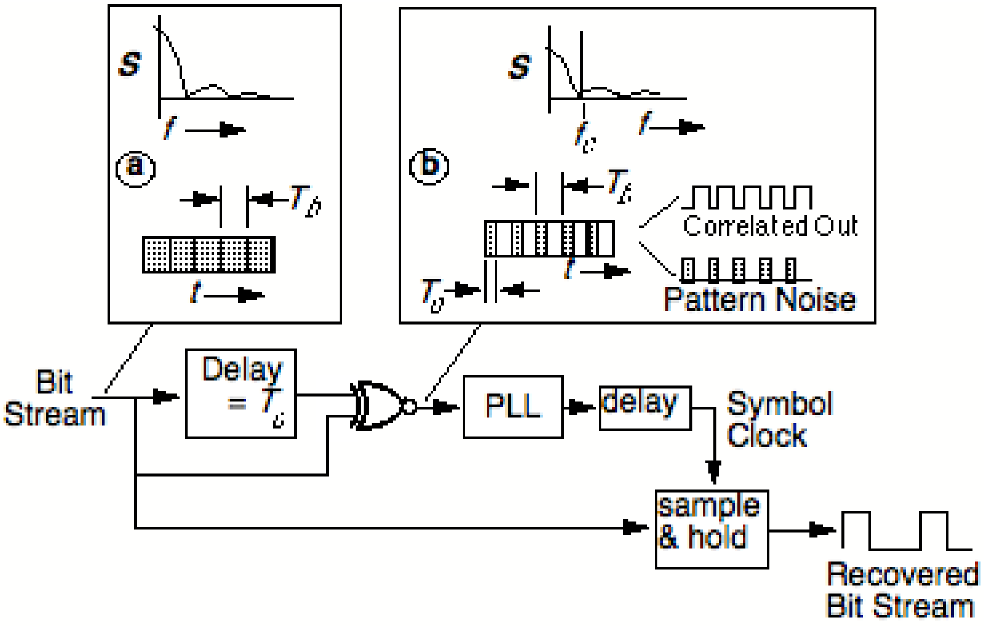


Fig. 10.C.1 Symbol Recovery

The incoming bit stream is represented at **a**. In the time domain, it consists of a series of data bits of unknown (random) value, usually 0 or 1 with equal probability. In the frequency domain, the power spectral density *S* has the shape characteristic of a single pulse of width T_b. There is no discrete spectral line since the stream is random. In order to develop a spectral line, the stream is delayed by T_d, approximately equal to half of T_b, and then enters an exclusive-or gate having an inverted output (ExNOR) with the undelayed version. (An analog multiplier can also be used.) During the delay period after the beginning of the pulse, the output from the ExNOR is the product of the two adjacent bits and is unknown (random). But, after T_d has passed, a one is produced since both inputs are from the same pulse. The resulting output at **b** can be decomposed into a pulse stream of frequency $1/T_b$ and a stream of random pulses. The random pulses produce "pattern noise" while the regular pulse stream is characterized by a discrete line at the bit rate. A PLL is employed to reproduce the bit frequency while filtering out the noise. A narrow filter could be employed to do this but would not be able to accommodate changes in bit rate. The PLL output, after an appropriate delay, can be used to drive the sample-and-hold circuit to sample the original pulse stream

10.D APPENDIX: ADPLL BY PULSE ADDITION AND REMOVAL, ADDITIONAL MATERIAL

10.D.1 Implementation of the Increment/Decrement Circuit

The ÷M to ÷2 interface is actually mechanized as shown in Fig. 10.D.1. When counting up, if the counter exceeds its modulus (capacity) it outputs a carry signal, which is an increment command to the ÷2. Conversely, when counting down, it can output a borrow signal, which is a decrement input.

10.D.2 Stability

Since the analog first-order loop is inherently stable, we look for stability problems due to the sampling effect that has been added by the digital implementation, such as we found with the simple loop of section 10.7.1.

We can identify three sampling processes: clocking at frequencies F_M and F_N and phase detection. Phase detection with an ExOR gate can be considered a sampling process because phase information is only acquired at the transitions of the output waveform (twice a cycle); no phase information is available between transitions. The sampling process at frequency f_s will not appreciably affect stability if

$$f_s >> \omega_L = K \,. \tag{10.D.1}$$

From Eq. (10.45), this requires that

$$MN >> 2, \tag{10.D.2}$$

in order that $F_M >> \omega_L$ cycle/rad, and that

$$MN >> 2F_M/F_N \,, \tag{10.D.3}$$

in order that $F_N >> \omega_L$ cycle/rad.

From Eqs. (10.45) to (10.47), the lowest input sampling frequency is

$$2f_{\text{in}} \geq 2(f_c - F_H) = \frac{F_N}{N} - \frac{F_M}{MN} = \frac{\omega_L}{2}\left(\frac{F_N}{F_M}M - 1\right). \tag{10.D.4}$$

This is much greater than ω_L if

$$M >> 2F_M/F_N. \tag{10.D.5}$$

Thus (10.D.2), (10.D.3), and (10.D.5) are sufficient to guarantee stability and, since M and N are usually large numbers (to reduce ripple), these conditions should be easily met if F_M is not too much greater than F_N.

10.D.3 Ripple Control

Ripple refers to jitter on the loop output due to variations in instantaneous frequency. Although the loop controls average frequency, it does so by removing pulses so, depending on the sequence in which this is done, different cycles may be of different lengths, even though the average frequency is correct. The loop can be designed so it will not produce a jitter of more than 1/N cycles at the output.

Assume initially that the output is at f_c so the PD duty factor is 50%, in the center of its range. The time during which the PD output is at a single state is one quarter of an output cycle (Fig. 3.2b) so, using Eq. (10.47),

$$T_{PD} = 1/(4f_c) = N/(2F_N)\,. \tag{10.D.6}$$

We can prevent more than one increment or decrement pulse from occurring at I during this period if we make the duration of the M count at least as long as T_{PD}. Thus we require

$$M/F_M \geq T_{PD}\,. \tag{10.D.7}$$

Combining these last two expressions, we obtain

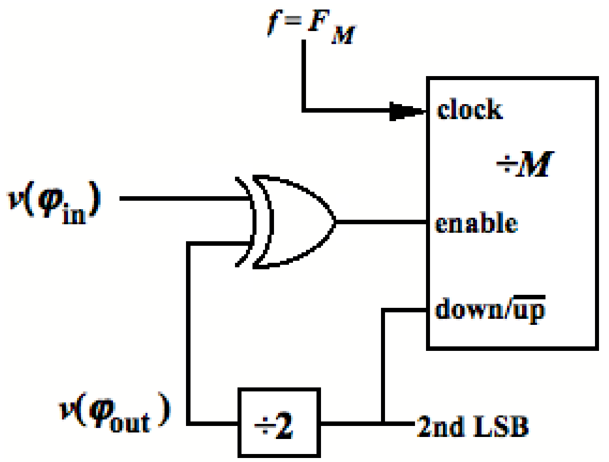
$$M \geq NF_M/(2F_N)\,. \tag{10.D.8}$$

If we meet this restriction, we limit the unnecessary outputs from the *M* divider to one in each direction during lock at center frequency. When the frequency is offset from center, increments or decrements are required but (10.D.8) will limit the number of ÷*M* outputs that are in the "wrong" direction to one because the width of the PD output during the time such outputs are generated is narrower than what is given by Eq. (10.D.6) and therefore Eq. (10.D.7) is still met by that period.

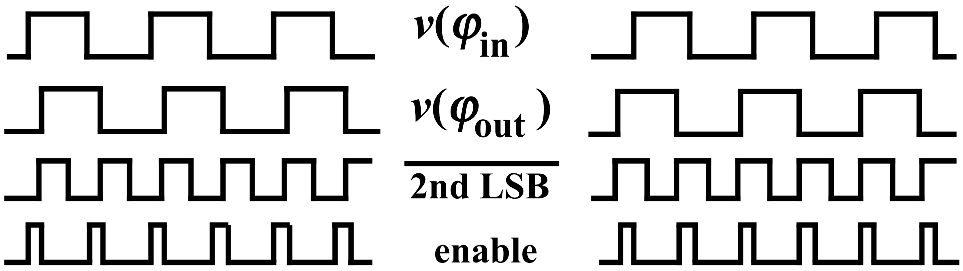
Additional circuitry can also be used for ripple control. The circuit in Fig. 10.D.2*a* inhibits up and down counting when the two inputs are in phase, the zero-error relationship for this circuit. As the phase moves away from this center value, up or down counting occurs in ÷M, but for short times when the deviation from center is small. When $v(\varphi_{in})$ and $v(\varphi_{out})$ are in phase, the ExOR output is zero and the ÷M is not enabled. If $v(\varphi_{out})$ becomes delayed, as in Fig. 10.D.2*b*, the enable occurs when the second least significant bit of the counter is 1 and its inverse causes the ÷M to count up. Conversely, in the case shown at Fig. 10.D.2*c*, it counts down. However, it only counts during the enable pulse, which would be narrow near band center, as opposed to counting all the time with the original realization. This scheme, however, reduces K_p and the tuning range[1] by 2.

---

[1] The application note [Troha, 1994] indicates that the range is reduced by (2 + 1/*M*), perhaps due to the details of how the enable and up/down functions work.



(a)



(b) (c)

Fig. 10.D.2 Ripple inhibiting circuit.

10.D.4 Second-Order ADPLLs

Second- and higher-order loops are also possible using this general design. Figure 10.D.3 is another representation of the system of Fig. 10.33. This time we have shown F_N as a second input. Previously we had ignored it because it was a constant. Now, however we wish to use that point to inject a signal. We have redefined K_p and K_v to accommodate this; but K is not affected by the redefinition and so neither is the loop response.

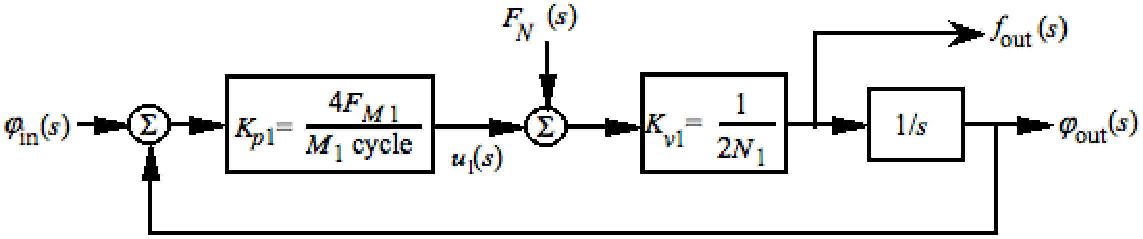

$F_N\ (s)$
$f_{\text{out}}\ (s)$
$\varphi_{\text{in}}(s)$
Σ
$K_{p1} = \frac{4F_{M1}}{M_1\ \text{cycle}}$
$u_1(s)$
Σ
$K_{v1} = \frac{1}{2N_1}$
$1/s$
$\varphi_{\text{out}}(s)$

Fig. 10.D.3 Mathematical block diagram rearranged.

10.D.4.1 Transfer Function

We will feed $f_{\text{out}}(s)$ through a second loop like this one and back to F_N to create a loop filter. We have added subscripts to Fig. 10.D.3 to differentiate this main loop (1) from the second loop that is used to generate the filter. F_N now changes from a constant clock to a function of s. The transfer function of the second loop is given by Eq. (2.25b) and we will multiply this by a constant, γ, giving

$$H_2(s) = \frac{\gamma}{1 + \frac{s}{K_2}}. \tag{10.D.9}$$

This will create a transfer function, between the PD output and f_{out}, of

$$K_{v1}K_{LF}F(s) = \frac{K_{v1}}{1 - K_{v1}H_2(s)} = K_{v1}\,\frac{1 + \frac{s}{K_2}}{1 - K_{v1}\gamma + \frac{s}{K_2}} \tag{10.D.10}$$

$$= \frac{K_{v1}}{1 - K_{v1}\gamma}\;\frac{1 + \frac{s}{K_2}}{1 + \frac{s}{(1 - K_{v1}\gamma)K_2}}. \tag{10.D.11}$$

Here we have used Eq. (2.22) with a minus sign on the feedback because the standard diagram has a minus at the summer but this implementation does not. By the way Eq. (10.D.10) is written, we are effectively abstracting K_v from the response to use for VCO gain constant.) From this, and our definition of K_{v1} in Fig. 10.D.3, we can see the following equivalencies to our standard parameters for a loop with a lag-lead filter.

Partly from William F. Egan, *Phase-Lock Basics*. Reprinted with permission of John Wiley and Sons, Inc.

$$K_{p1} = \frac{4F_{M1}}{M_1 \text{ cycle}} \tag{10.D.12}$$

$$K_{v1} = 1/(2N_1) \tag{10.D.13}$$

$$\omega_z = K_2 \tag{10.D.14}$$

$$\omega_p = K_2 (1 - K_{v1}\gamma) = K_2 \left(1 - \frac{\gamma}{2N_1}\right) \tag{10.D.15}$$

$$K_{\text{LF}} = \frac{1}{1 - K_{v1}\gamma} = \frac{1}{1 - \frac{\gamma}{2N_1}} \tag{10.D.16}$$

$$K_1 = K_{p1}K_{\text{LF}}K_{v1} = \frac{2F_{M1}}{M_1N_1(1-K_{v1}\gamma)\text{cycle}} = \frac{2F_{M1}}{M_1N_1(1-\gamma/(2N_1))\text{cycle}} \tag{10.D.17}$$

$$\omega_n^2 = K_1\omega_p = \frac{2F_{M1}K_2}{M_1N_1 \text{ cycle}} \tag{10.D.18}$$

Using K_1, ω_p, ω_z, and ω_n from above, ζ and α can be obtained from Eqs. (4.9) and (6.5). (That may be more efficient than expanding those expressions directly in terms of the circuit parameters.)

If we want an integrator-and-lead filter, we set $\gamma = 1/K_{v1} = 2N_1$ and obtain a filter transfer function, from (10.D.10), of

$$K'_{\text{LF}}F'(s) = \frac{1 + s/K_2}{s/K_2} = K_2 \frac{1 + s/K_2}{s}, \tag{10.D.19}$$

where primes are used to designate this particular choice of γ. The terms K_{v1} and ω_z are still given by (10.D.13) and (10.D.14). By comparing $K_{p1}K_{v1}K'_{\text{LF}}F'(s)/s$ to Eq. (4.23), we find

$$\omega_n^2 = K_2K_{p1}K_{v1} = \frac{2F_{M1}K_2}{M_1N_1 \text{ cycle}}, \tag{10.D.20}$$

which is the same as Eq. (10.D.18). The damping factor is

$$\zeta = \frac{1}{2}\frac{\omega_n}{\omega_z} = \sqrt{\frac{F_{M1}}{2K_2M_1N_1 \text{ cycles}}}\ . \tag{10.D.21}$$

Responses for either type of loop (with lag-lead or integrator-and-lead filter) can be obtained from the equations in Table 4.1 or the graphs of Chapters 6 and 7 using the parameters computed above.

Notice, in Eq. (10.D.11), that, if γ should exceed $1/K_v$, the pole would move to the right half plane and the filter would be unstable. However, because we are using numbers rather than analog components, the value of γ can be set precisely so we can obtain an integrator-and-lead filter without drifting into the right half plane. In fact, here is the first time we have met a true type-2 loop, one that really does have a second pole at zero rather than at some frequency small enough to be approximated as zero.

10.D.4.2 Realization

The multiplying factor γ can be realized by taking the output from the second loop before the final $\div N$, effectively multiplying the output frequency by N_2, the value of N in loop 2. A $\div L$ circuit is also placed at the input to the second loop, giving $\gamma = N_2/L$. The final structure is shown in Fig. 10.D.4.

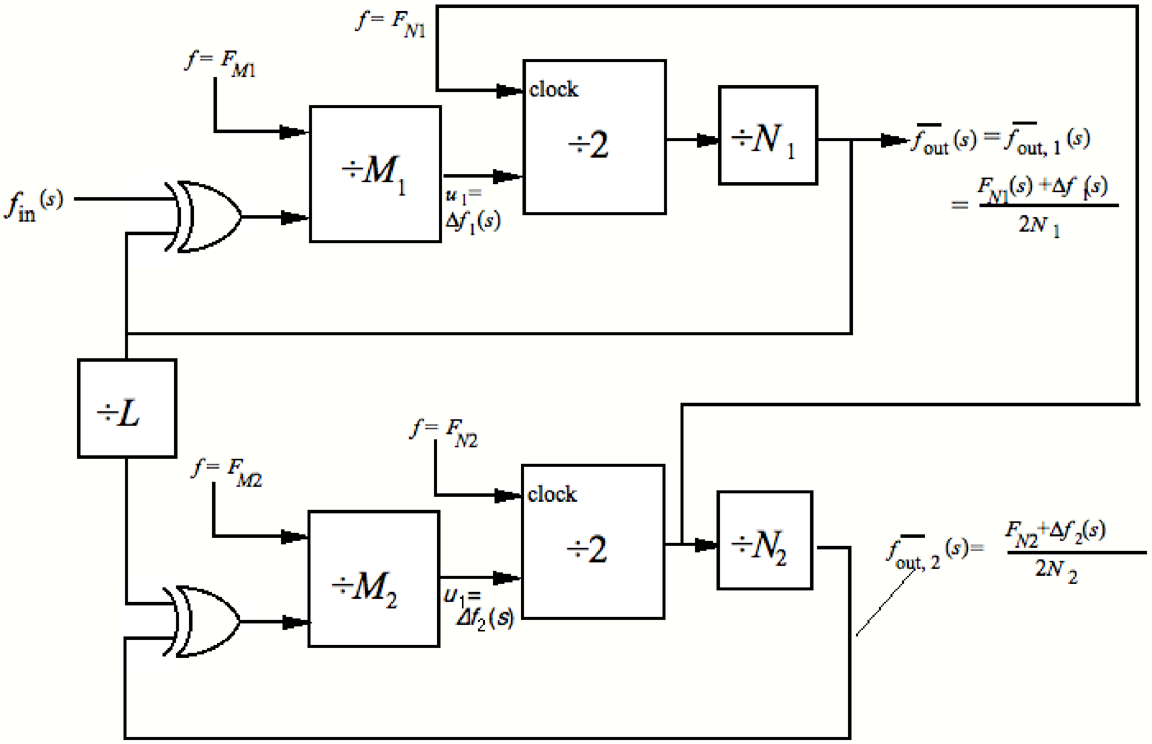


Fig. 10.D.4 Second-order ADPLL using pulse removal.

10.D.4.3 Tuning Range

The center frequency depends on the center value of $F_{N1}(s)$, which depends on F_{N2}. From Fig. 10.D.4, we can see this to be

$$f_c = \frac{\overline{F_{N1}(s)}}{2N_1} = \frac{F_{N2}}{4N_1}\ . \qquad (10.D.22)$$

The hold-in range could be set by either loop. From the expression for the output frequency (Fig. 10.D.4), a change in output frequency of Δf_{out} is accompanied by

$$\Delta f_1 = \Delta f_{\text{out}} 2N_1 - \Delta F_{N1} = \Delta f_{\text{out}} 2N_1 - \gamma\Delta f_{\text{out}} = \Delta f_{\text{out}}(2N_1 - \gamma). \qquad (10.D.23)$$

The corresponding phase change in loop 1 is

$$\Delta\varphi_1 = \frac{\Delta f_1}{K_{p1}} = \frac{\Delta f_{\text{out}} M_1 (2N_1 - \gamma)}{4F_{M1}\,\text{cycle}^{-1}} \qquad (10.D.24)$$

so we can see immediately that, for a type-2 loop, where $\gamma = 2N_1$, there is no phase change at the main-loop (loop 1) phase detector, as we would expect for such a loop.

Referring again to Fig. 10.D.4,

$$\frac{\Delta f_2}{2N_2} = \frac{\Delta f_{\text{out}}}{L}, \tag{10.D.25}$$

$$\Delta\varphi_2 = \frac{\Delta f_2}{K_{p2}} = \frac{\Delta f_{\text{out}} M_2 N_2}{2F_{M2} L \,\text{cycle}^{-1}}. \tag{10.D.26}$$

Neither $\Delta\varphi_1$ nor $\Delta\varphi_2$ can exceed one-quarter cycle, leading to a maximum value of Δf_{out} given by the smaller of the frequency changes with $\Delta\varphi = 1/4$ in Eq. (10.D.24) and (10.D.26),

$$\Delta f_{\text{out}} \le \min\left(\frac{F_{M2} L}{2M_2 N_2}, \frac{F_{M1}}{(2N_1 - \gamma) M_1}\right). \tag{10.D.27}$$

10.T APPENDIX: COMBINED PLL AND DLL

The combination of DLL and PLL discussed in Section 10.6 is illustrated in Fig. 10.T.1 [Lee, 1992].

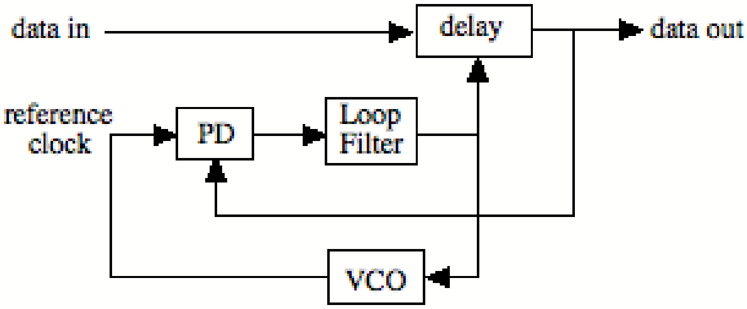
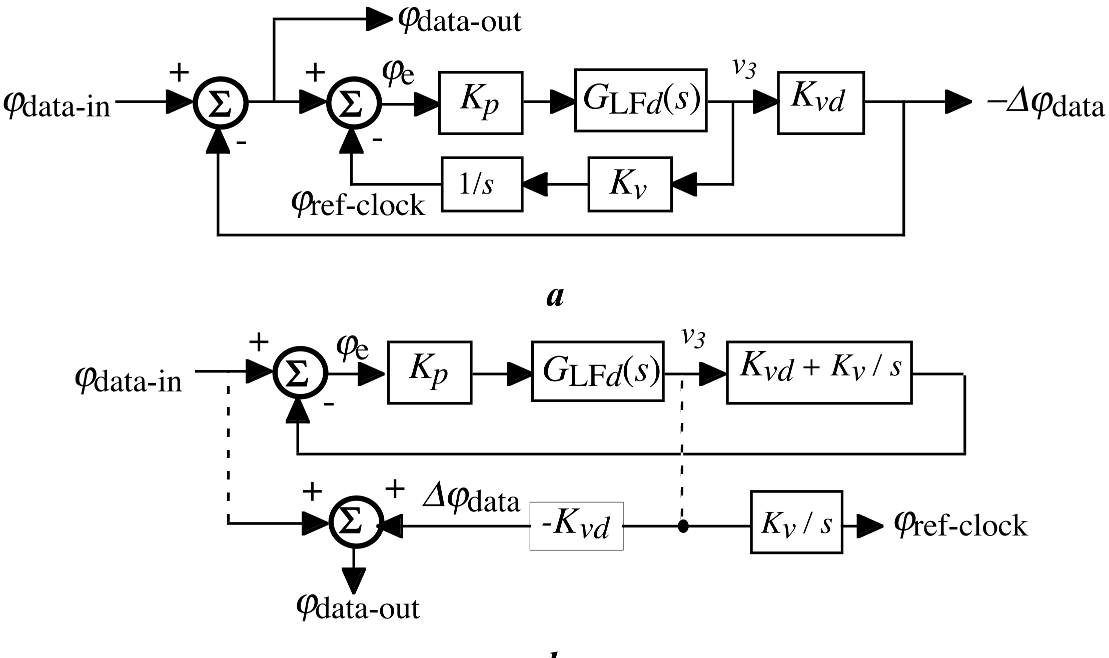


Fig. 10.T.1 Simultaneous synchronization of the data and extraction of its clock.

Now the mathematical block diagram must again show the reference clock input to the PD, something that we dropped in going from Fig. 10.23 to 10.24. The mathematical representation of Fig. 10.T.1 is shown in Fig. 10.T.2*a*. The feedback path in the inner loop represents the VCO, which is controlled by v_3. Its phase output is subtracted from $\varphi_{\text{data-out}}$ in the PD. By adding the two paths from v_3 to φ_e we obtain the loop shown in Fig. 10.T.2*b*. (The variables from the original loop, $\varphi_{\text{data-out}}$ and $\Delta\varphi_{\text{data}}$, no longer appear in the loop so they must be synthesized from other variables that do appear.)


$\varphi_{\text{data-out}}$
$\varphi_{\text{data-in}}$
+
-
+
-
Σ
φ_e
K_p
$G_{LFd}(s)$
v_3
K_{vd}
$-\Delta\varphi_{\text{data}}$
$1/s$
K_v
$\varphi_{\text{ref-clock}}$
a
$\varphi_{\text{data-in}}$
φ_e
K_p
$G_{LFd}(s)$
v_3
$K_{vd} + K_v / s$
$\Delta\varphi_{\text{data}}$
$-K_{vd}$
K_v / s
$\varphi_{\text{ref-clock}}$
$\varphi_{\text{data-out}}$

b

Fig. 10.T.2 Mathematical block diagram of combined DLL and PLL at *a* with equivalent diagram shown at *b*. (Dashed connections are external to the loop.)

The right-most block in the loop in Fig. 10.T.2.*b* is an integrator-and-lead element of the same form as Eq. (3.20). Assuming $G_{\mathrm{LF}d}(s)$ consists only of an integrator (i.e., $F(s) = 1$ sec^{-1} in Eq. 10.20), Fig. 10.T.2 becomes Fig. 10.T.3, where

$$\omega_z = K_v \,/\, K_{vd}. \tag{10.T.1}$$

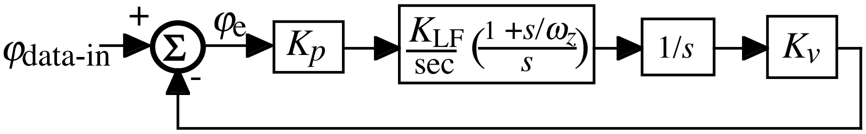

+
φ_e
$\varphi_{\text{data-in}}$
Σ
-
K_p
$\frac{K_{\mathrm{LF}}}{\mathrm{sec}}\left(\frac{1+s/\omega_z}{s}\right)$
$1/s$
K_v

Fig. 10.T.3 Combined DLL and PLL with integrator loop filter.

Here $1/C$ in Eq. (3.20) becomes K_{LF} sec^{-1}, so the natural frequency (see column *c* in Table 4.1) is given by

$$\omega_n^2 = K_{\mathrm{LF}} K_p K_v \,\mathrm{sec}^{-1} \tag{10.T.2}$$

and the damping factor is

$$\zeta = 0.5 \sqrt{\frac{K_{\mathrm{LF}} K_p}{K_v \,\mathrm{sec}}} K_{vd} \,. \tag{10.T.3}$$

This circuit has the response of a PLL with an integrator-and-lead filter. Thus we can use the stability requirements in Chapter 5. However, the responses in Chapters 6 and 7 do not describe directly the response of the recovered clock or the retimed data.