In-phase orthogonal loop—Costas Loop

Advantages of Costas loop:

1.Costas loop works on f_c instead of $2f_c$, so when f_c is large, Costas loop is easier to realize

2.The output of in-phase loop $r_2(t)$ is the signal $f(t)$

3. Performance of carrier synchronization technique

1) Phase error: steady-state phase error, random phase error
2) Synchronization build time and hold time

Performance

1. steady-state phase error ϕ_e

(1) Narrowband filter is a simple single tuned loop with a fixed Q value. When the central frequency ω_0 is not equal to the carrier frequency ω_c, steady-state phase error $\Delta\phi$ is arose.

Let $\Delta\omega = |\omega_c - \omega_0|$

When $\Delta\omega$ is small

$$\Delta\phi = 2Q\frac{\Delta\omega}{\omega_0},\quad \text{Q}\downarrow,\quad \Delta\phi\downarrow$$

(2)*W*hen PLL is applied: $\Delta\phi = \phi_e = \dfrac{\Delta\omega}{k_0}$

k_0 is the DC gain of the PLL circuit. Apparently, $\Delta\phi$ can have a very small value as long as k_0 is large enough.

Performance

2. Random phase error

Assume Gaussian random noise is the case, we already know when SNR is high (SNR>>1), the phase distribution of the sum of a sinusoid signal and a gaussian noise passed through a narrowband system is a gaussian distribution.

Let $\varphi(0) = 0,$ then:

$$f(\phi) = \sqrt{\frac{d}{\pi}} e^{-d\phi^2}$$

$f(\phi)$ with $mean = 0,$ $\sigma_\phi^2 = \frac{1}{2d}$, we use σ_ϕ to assess the random phase error.

$$\sigma_\phi = \sqrt{\frac{1}{2d}}$$ – random phase jitter

Performance

Example：Narrowband filter is a single tuned loop.

The equivalent noise bandwidth: $B_n = \frac{\pi}{2}\frac{f_0}{Q}$

Let the noise's single-sideband PSD is N_0, then the noise power is $P_n = N_0 \cdot B_n$

$$\therefore\ d = \frac{P_S}{P_n} = \frac{P_s}{N_0} \cdot \frac{2Q}{\pi f_0}$$

then $\sigma_\phi = \sqrt{\frac{1}{2d}} = \sqrt{\frac{\pi N_0 f_0}{4QP_S}}$

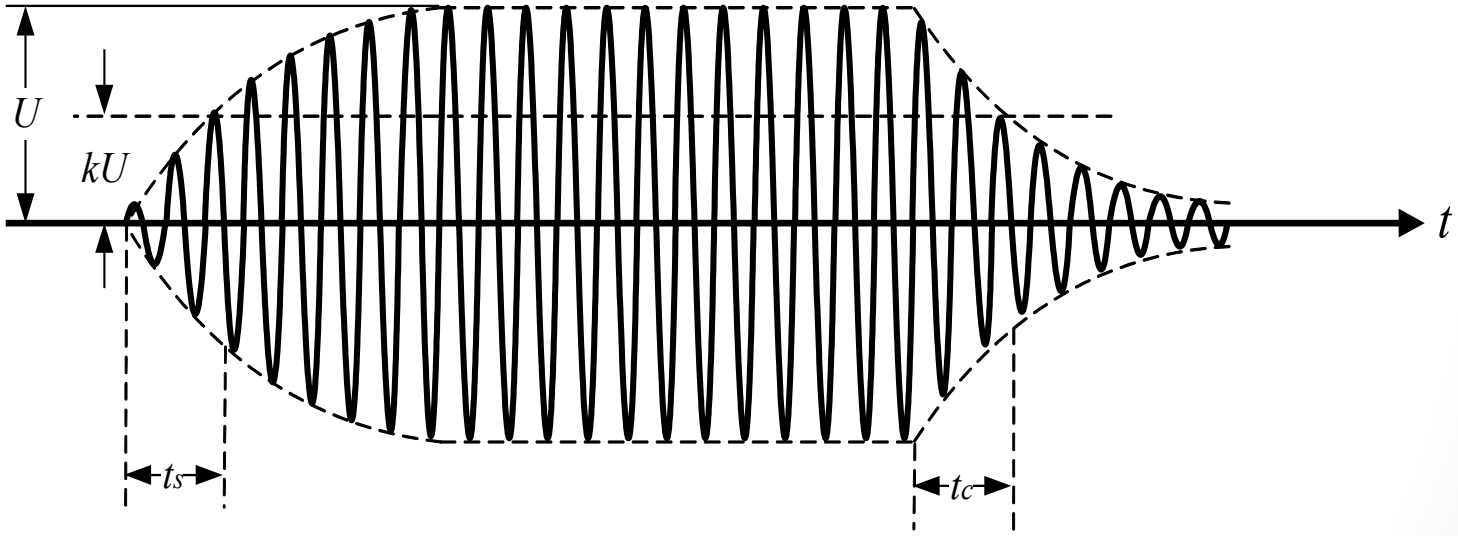
Apparently，Q↑, σ_ϕ↓.

Based on above discussion, we can see there is a trade-off between minimize the steady-state phase error and the random phase error.

Performance

3. Synchronization build time and hold time

Example: using single tuned loop to realize narrowband filtering



Performance

(1) When t = 0, the output voltage is:

$$u(t) = u_0\left(1 - e^{-\alpha t}\right)\cos\omega_0 t$$

$\alpha = \omega_0 / 2Q$ ω_0 is the resonant frequency

When $t = t_S$, synchronization is build when the amplitude goes to $Ku_0\ (0 < K < 1)$

As: $$Ku_0 = u_0\left(1 - e^{-\alpha t_s}\right)$$

So $t_s = \frac{1}{\alpha}\ln\frac{1}{1-k}$ ——synchronization build time

Performance

(2) When synchronization is build, cut off the input signal at $t = 0$, then the output is:

$$u(t) = Ue^{-\alpha t}\cos\omega_0 t$$

When $t = t_c$, synchronization is hold until the amplitude decreases to Ku_0

As $\mathrm{KU} = \mathrm{U}e^{-\alpha t_s}$

So $t_c = \frac{1}{\alpha}\ln\frac{1}{k}$ ——synchronization hold time

Performance

$$t_s = \frac{1}{\alpha} \ln \frac{1}{1-k} \qquad t_c = \frac{1}{\alpha} \ln \frac{1}{k}$$

$$\alpha = \omega_0 / 2Q$$

In practice, we want $t_s \downarrow$, $t_c \uparrow$. However, when Q $\uparrow$, $\alpha \downarrow$, both t_s and t_c $\uparrow$, vice versa. Therefore, we need consider both factors when designing the parameters.

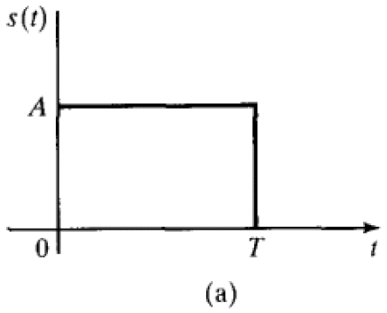
When using PLL, t_s is the lock time, t_c is the hold time. Their values depend on the parameters of circuits. Similarly, there is also a conflict when picking parameters. However, we can change the parameters after lock is build, therefore let $t_s \downarrow$ and $t_c \uparrow$.

Symbol Synchronization

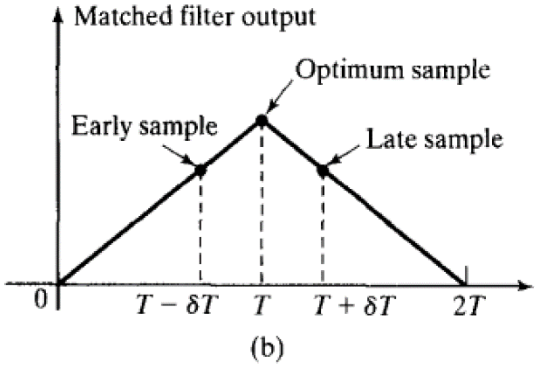
- In a digital communication system, the output of the receiving filter must be sampled periodically at the symbol rate and at the precise sampling time instance.
- To perform this periodic sampling, we need a clock signal at the receiver
- The process of extracting such a clock signal is called symbol synchronization or timing recovery
- One method is for the transmitter to simultaneously transmit the clock frequency along with the information signal. The receive can simply employ a narrowband filter or PLL to extract it. This method requires extra power and bandwidth and hence, but frequently used in telephone transmission systems.
- Another method is to extract the clock signal from the received data signal by using some kind of non-linear transformation.

Early-Late Gate Synchronization

- Basis Idea: exploit the symmetry properties of the output signal of matched filter or correlator



(a)



(b)

Figure 8.48 (a) Rectangular signal pulse and (b) its matched filter output.

- Due to the symmetry, the values of the correlation function at the early samples $t = T - \delta T$ and the late samples $t = T + \delta T$ are equal.
- Thus, the proper sampling time is the midpoint between $t = T - \delta T$ and $t = T + \delta T$

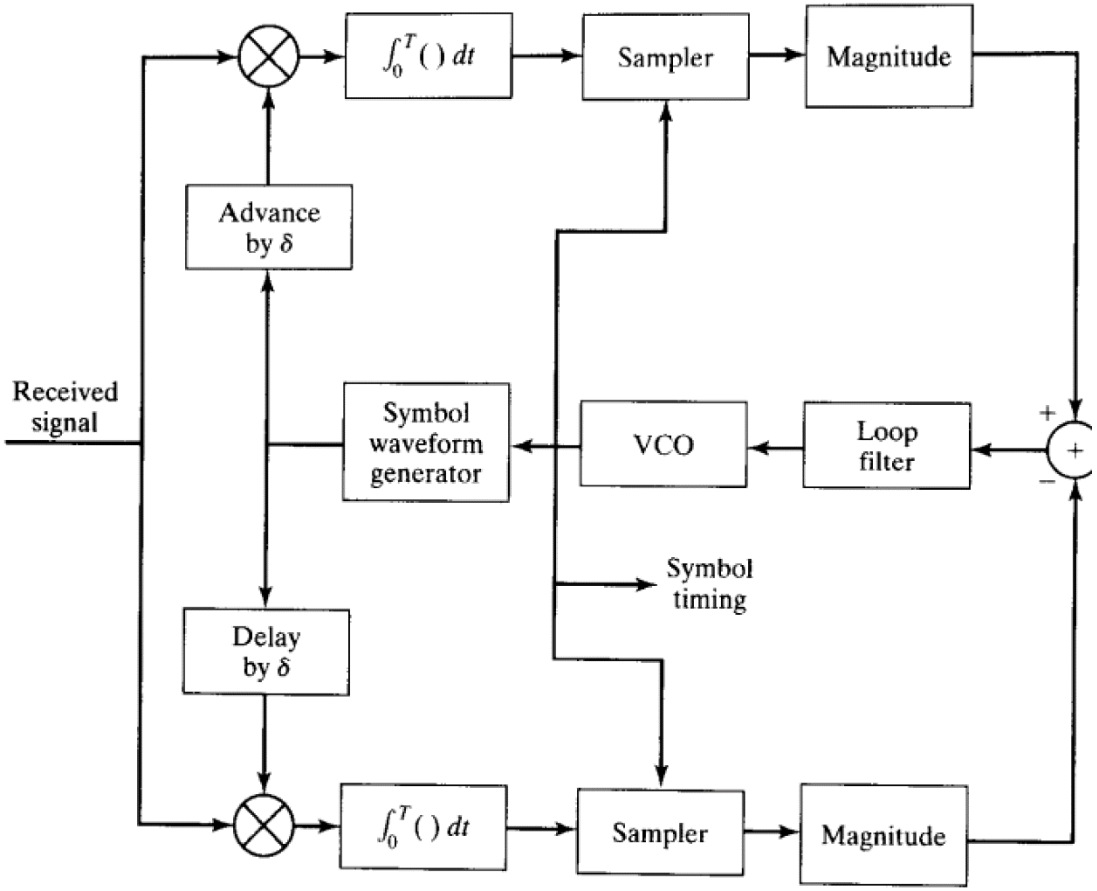
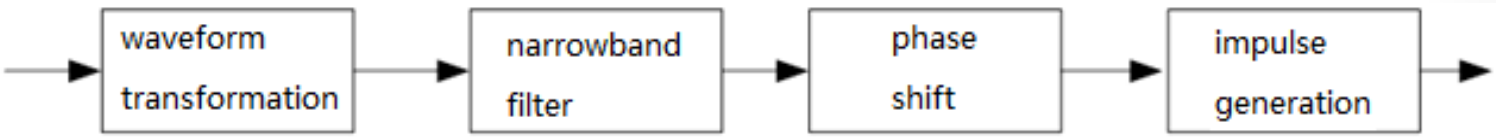


Figure 8.49 Block diagram of early–late gate synchronizer.

Nonlinear-transformation-based method

1. Nonlinear-transformation-based method

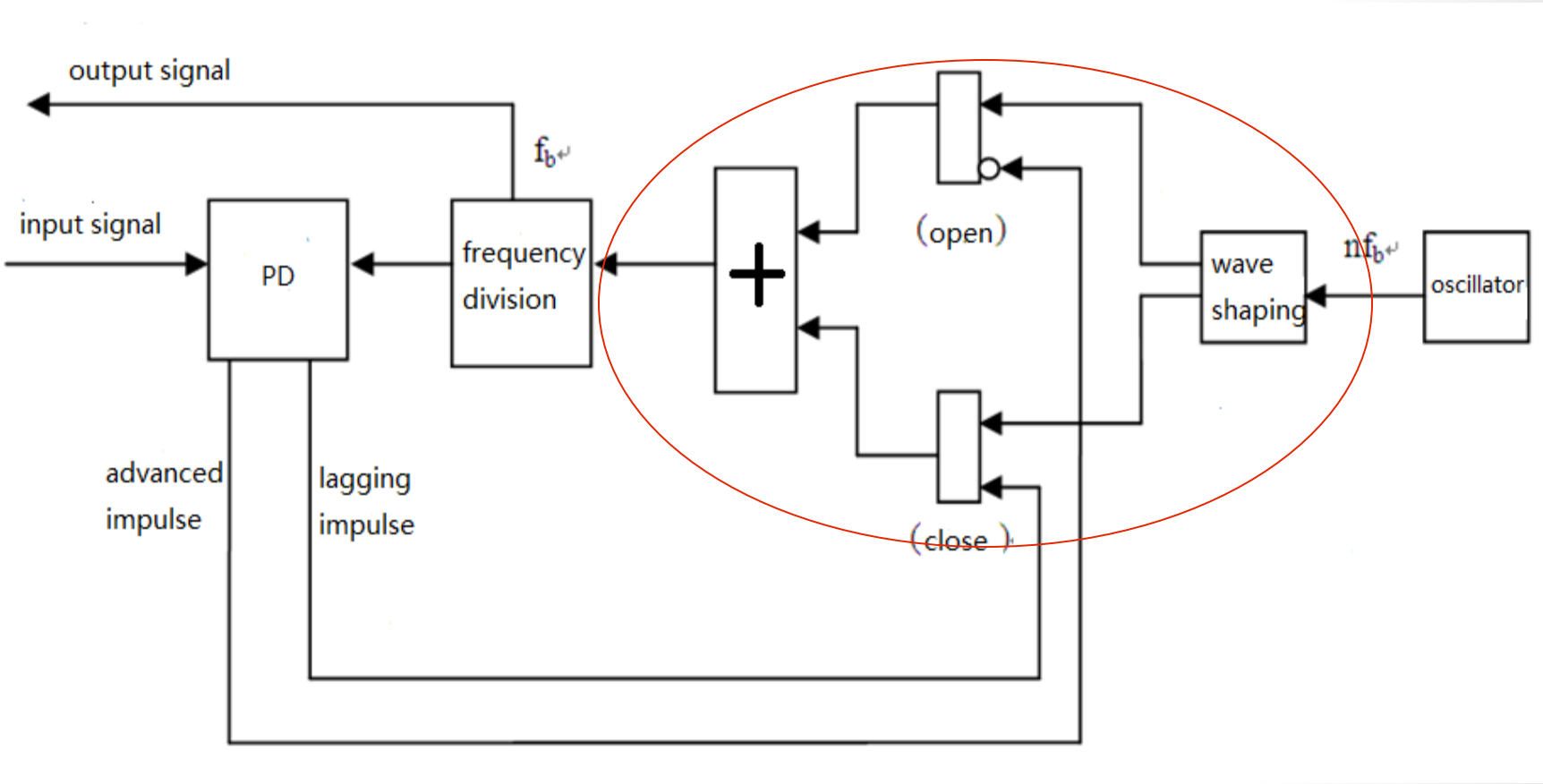


Some transformations can add synchronous signal with f=1/T to the original signal. For example, we can transform the signal to return-to-zero waveform. After narrowband filtering and phase shifting, we can generate the clock signal used for synchronization.

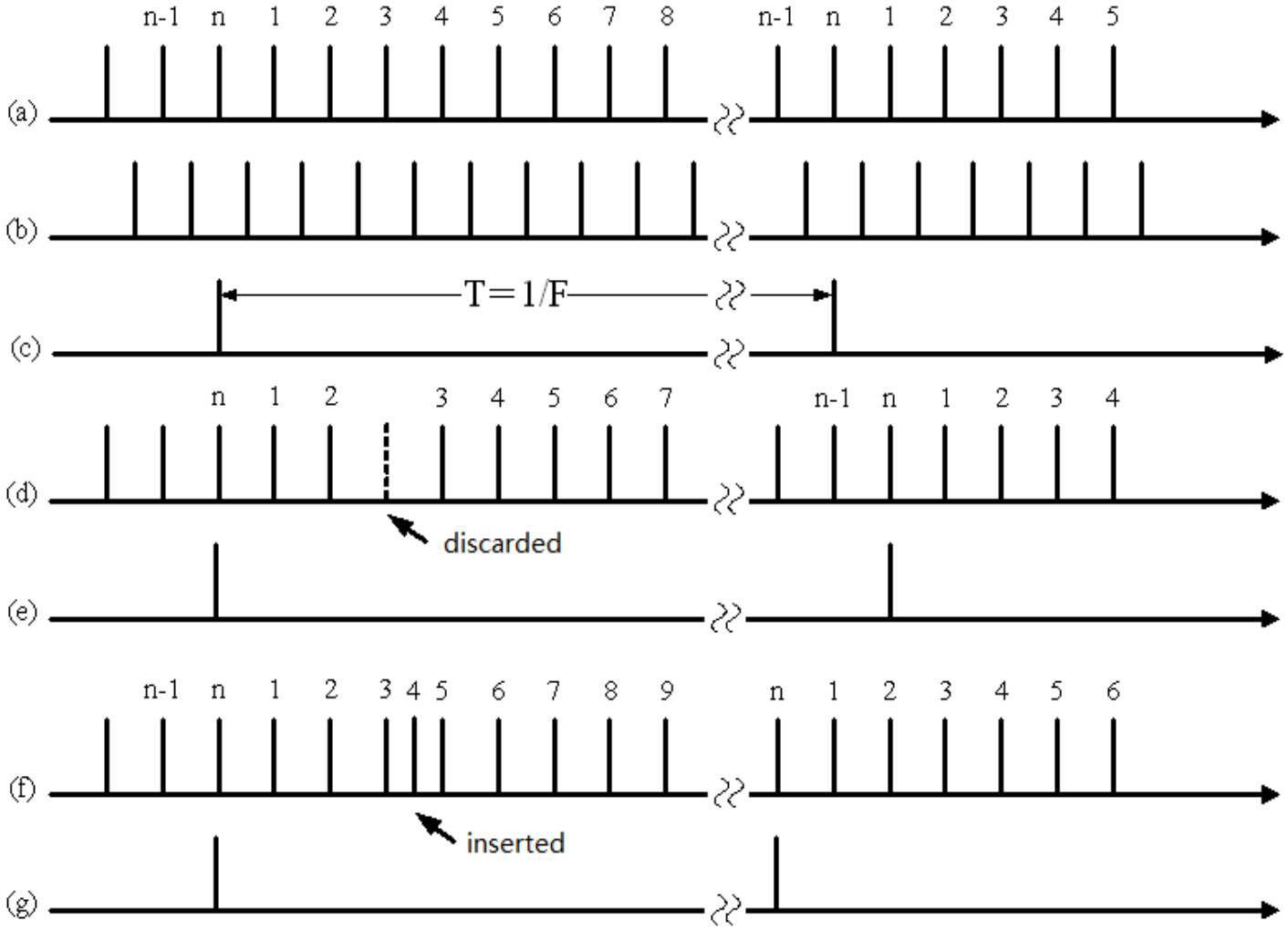
$$P_s(f) = f_s P(1-P)\left|G_1(f) - G_2(f)\right|^2 + f_s^2 \sum_{m=-\infty}^{\infty} \left|PG_1(mf_s) + (1-P)G_2(mf_s)\right|^2 \delta(f - mf_s)$$

Digital PLL (DPLL)

2. DPLL



Digital PLL (DPLL)



3. Performance of symbol synchronization system

—DPLL

1). Phase error

The phase error occurs because of for a DPLL, the phase cannot change with arbitrary small value, each time the phase can only be modified by 2π/n（n is the frequency division number）. Therefore the maximum of phase error is:

$\varphi = 2\pi/n(rad)$

In time domain, it is:

$$T_e = T_b / n(s)$$

Performance

2. Synchronization build time t_s

Synchronization build time is the maximum period from the system lost synchronization to the system back to synchronization state.

The maximum delay between the synchronous impulse and the received signal is $T_b/2$. Each time the DPLL can only modify the phase by $T_b/2n$, so synchronization is build after N times:

$$N = \frac{T_b/2}{T_b/n} = \frac{n}{2}$$

The PDLL can only modify the phase when the received code crosses zero point. As to random binary code, we can approximately think '01','10','11','00' appears with same probability. Therefore, the code will cross zero point with probability 0.5. Thus, a modification happens each $2T_b$ seconds.

Synchronization build time is:

$$t_s = 2T_b \cdot N = nT_b$$

Performance

3. Synchronization hold time t_c

If the input signal is interrupted after synchronization is build, since there is a frequency difference ΔF between the transmitter and the receiver, the phase of the synchronous signal will keep drifting. We consider the system is not synchronous any more if the phase is drifted by a centain value. This period is called synchronization hold time.

If the transmitter and the receiver have $T_1 = \frac{1}{F_1}$ and $T_2 = \frac{1}{F_2}$, separatively, then

$$|T_1 - T_2| = \left|\frac{1}{F_1} - \frac{1}{F_2}\right| = \frac{|F_2 - F_1|}{F_1 F_2} = \frac{\Delta F}{F_0^2}$$

$F_0 = \sqrt{F_1 F_2}$ and $T_0 = 1/F_0$

It can also be written as: $F_0 |T_1 - T_2| = \frac{\Delta F}{F_0}$

$$\therefore \quad \frac{|T_1 - T_2|}{T_0} = \frac{\Delta F}{F_0}$$

Performance

When frequency difference ΔF exists，the phase will drifted by $|T_1 - T_2|$after every T_0. If the system can only allow the drift to be as large as T_0/K (K is relevant with Pe), then we can calculate the synchronization hold time t_c as:

$$\frac{T_0/k}{t_c} = \frac{\Delta F}{F_0}$$

$$\therefore \qquad t_c = \frac{1}{\Delta F \cdot k}$$

If t_c fixed，then the system requires ΔF to be:

$$\Delta F = \frac{1}{t_c \cdot k}$$

If the oscillators in transmitter and receiver have the same stability, then the stability should not below:

$$\frac{\Delta F}{2F_0} = \pm\frac{1}{2t_c k F_0}$$

Performance

4. Synchronous bandwidth Δf_s

*S*ynchronous bandwidth is the range of ΔF.

The drift occurs during one symbol interval is：

$$\Delta T = |T_1 - T_2| = \frac{\Delta F}{F_0} T_0 = \frac{\Delta F}{F_0^2}$$

As mentioned, the DPLL modifies the phase every 2 symbols. Each time spends T_0/n seconds. To mentain synchronazation, obviously we need:

$$\Delta T \le \frac{T_0}{2n} = \frac{1}{2nF_0} \qquad \frac{\Delta F}{F_0^2} \le \frac{1}{2nF_0}$$

$$\therefore \qquad |\Delta f_s| = |\Delta F| \le \frac{F_0}{2n}$$

when $F_0 \to f_b$, we have: $|\Delta f_s| \le \frac{f_b}{2n}$