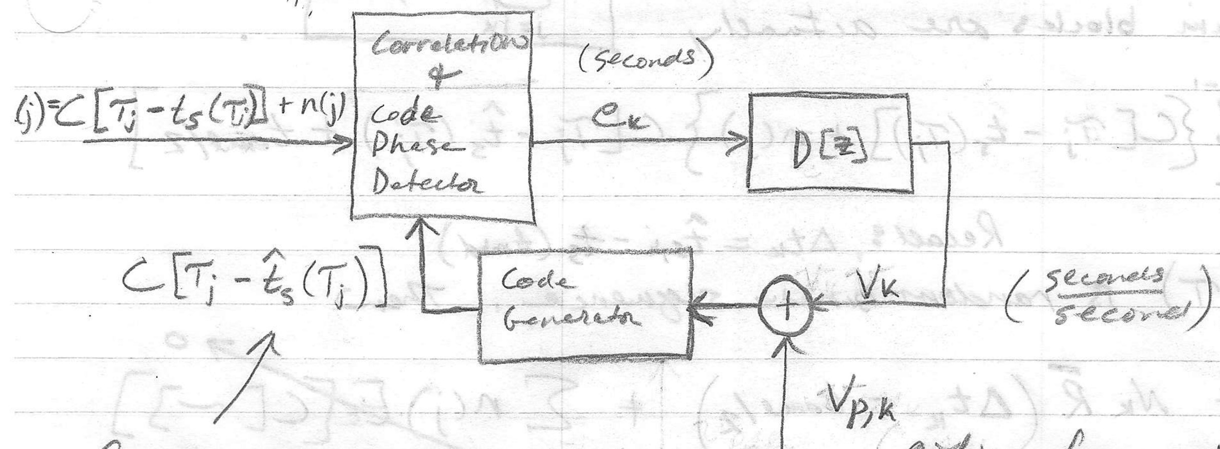
[Show big picture system before this to help orient students]

Lecture 16:

CODE TRACKING LOOPS



One may use a constant value here if the accumulation time within the code phase detector is short. In this case, set $\hat{t}_s(T_j) = \hat{t}_{s,k}$ over k^{th} subaccumulation interval.

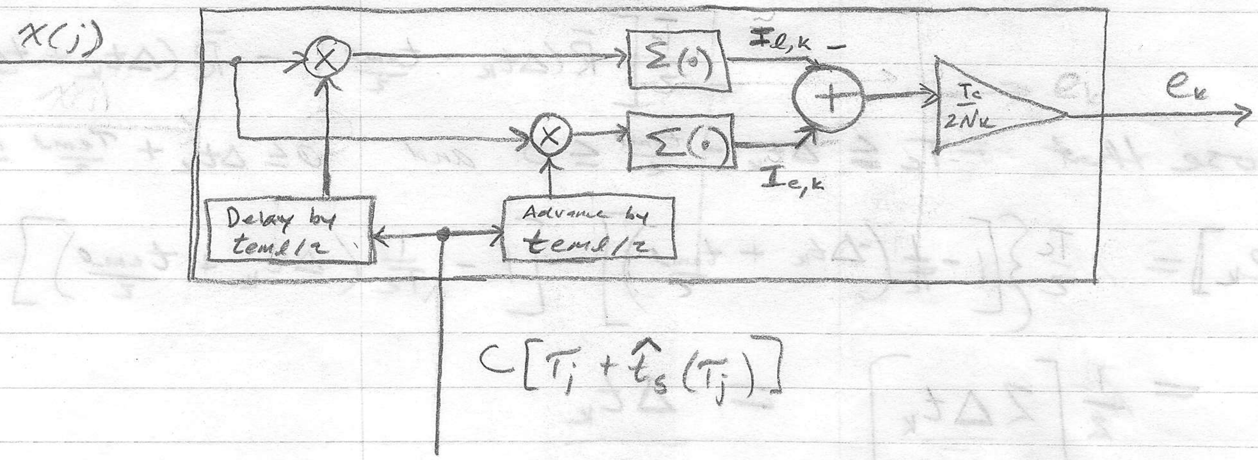
aiding from phase tracking loop (use $-V_{p,k}$ for high-side mixed IF data streams)

[we'll define this later in terms of V_k from PLL.]

Exploits fact that, absent code-carr. divergence, $t_s(T_j) = \beta\,\theta(T_j) + t_{s0}$

Code Phase Detector: We wish to design a code phase detector such that $E[e_k] = \Delta t_k \triangleq t_s(t_{mid_k}) - \hat{t}_{s,k}$ where $t_{mid_k} = \frac{1}{2}\left[\hat{t}_{s,k} + \hat{t}_{s,k+1}\right]$. Δt_k is the average code phase error over the k^{th} subaccumulation interval when $t_s(T_j) - \hat{t}_{s,k}$ changes linearly.

Consider the following correlation & phase detector block:



where the sum blocks are actually $\boxed{\sum_{j=j_k}^{j_k+N_k-1}(\circ)}$.

$$I_{e,k} = \sum_{j=j_k}^{j_k+N_k-1}\left\{C[T_j - t_s(T_j)] + n(j)\right\} C\left[T_j - \hat{t}_s(T_j) + t_{eml}/2\right]$$

Recall: $\Delta t_k = \hat{t}_{s,k} - t_s(t_{mid})$

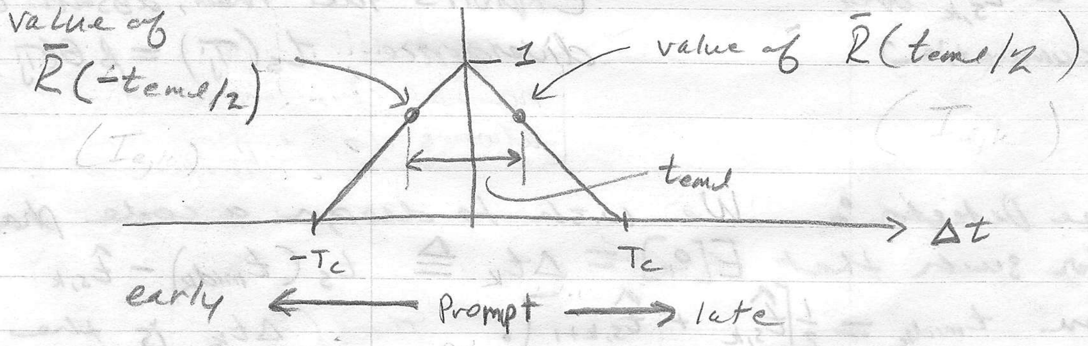
Suppose $C(T)$ is random binary sequence. Then

$$\underbrace{E_c[I_{e,k}]}_{\text{expectations over } C} \cong N_k \bar{R}\left(\Delta t_k - t_{eml}/2\right) + \sum n(j)\, \cancelto{0}{E_c\left[C[\sim]\right]}$$

Similarly, $E_c[I_{l,k}] \cong N_k \bar{R}\left(\Delta t_k + t_{eml}/2\right)$

$$\bar{R}(\Delta t) = \begin{cases} 1 - \frac{|\Delta t|}{T_c}, & |\Delta t| \le T_c \\ 0 & \text{else} \end{cases}$$

Recall the model of $\bar{R}(\Delta t)$:



Thus, when $\Delta t_k = 0$, $I_{e,k}$ picks off an early value of $\bar{R}$ and $I_{l,k}$ picks off a late value.

Now consider $E_c[e_k] = \frac{T_c}{2N_k}\left\{E_c[I_{e,k}] - E_c[I_{l,k}]\right\}$

$$= \frac{T_c}{2}\left[\bar{R}\left(\Delta t_k - \frac{t_{eml}}{2}\right) - \bar{R}\left(\Delta t_k + \frac{t_{eml}}{2}\right)\right]$$

Suppose that $-T_c \le \Delta t_k - \frac{t_{eml}}{2} \le 0$ and $0 \le \Delta t_k + \frac{t_{eml}}{2} \le T_c$. Then

$$E_c[e_k] = \frac{T_c}{2}\left\{\left[1 - \frac{1}{T_c}\left(-\Delta t_k + \frac{t_{eml}}{2}\right)\right] - \left[1 - \frac{1}{T_c}\left(\Delta t_k + \frac{t_{eml}}{2}\right)\right]\right\}$$

$$= \frac{1}{2}\left[2\Delta t_k\right] = \Delta t_k$$

Thus, the mean of the error signal e_k is Δt_k, just as we wanted.

[A full noise analysis yielding $\sigma_{\Delta t}^2 = E[e_k^2]$ is quite involved. See Sections 10.5 and 10.6 in Misra + Enge. ~~and see next lecture on CR bounds for TOA measurement errors.~~

$$\boxed{\text{Result: } \quad \sigma_{\Delta t}^2 \cong \frac{d B_{DLL} T_c^2}{2\left(\frac{C}{N_0}\right)} \quad \text{seconds}^2}$$

where $d = \dfrac{t_{eml}}{T_c}$ and B_{DLL} is DLL loop noise bandwidth ($\cong B_L$)

See graph on next page.

The detector introduced here is known as the coherent code phase det. There are many other code phase detection strategies, each with its strengths & weaknesses. (See Dierendonck in Blue Book.)

Loop Filter

Because GNSS code tracking loops are always aided by the PLL (or FLL), they require only a low-order loop filter and a small loop bandwidth $B_n = B_{DLL}$.

Adequate:

First-order loop with $D[z] = K = 4 B_n$

$$0.01 \leq B_n \leq 0.1$$

B_n only has to be wide enough to enable DLL to

1. track code-carrier divergence (very slow)
2. pull-in in a reasonable time (e.g., 100s)

Code Generation

To produce $C[\tau_j - \hat{t}_s(\tau_j)]$ at each τ_j given an input $V_k + V_{p,k}$, which remains const. from τ_{jk} to τ_{jk+1}, implement the following recipe for $\hat{t}_s(\tau_j)$:

$$\hat{t}_s(\tau_j) = \hat{t}_s(\tau_{jk}) - (V_k + V_{p,k})(\tau_j - \tau_{jk}), \quad \tau_{jk} \le \tau_j \le \tau_{jk+1}$$

Minus sign here because $(V_k + V_{p,k}) > 0$ indicates that the received spreading code appears <u>compressed</u> wrt the nominal code, so it's as if the nominal code start time moves <u>backwards in time</u>.

Note that $\hat{t}_s(\tau_{jk+1})$ still refers to the "same start time to which $\hat{t}_s(\tau_{jk})$ refers, even though a new code may have started in the interval $(\tau_{jk}, \tau_{jk+1}]$. This presents no problem algorithmically since the definition of $C[\tau]$ can handle values of τ longer than one nominal code interval $N_c \cdot T_c$:

$$C[\tau] = \underbrace{\sum_{i=-\infty}^{\infty} c_{\text{mod}(i,N_c)}}_{\text{cyclic code sequence repeats after } N_c} \Pi_{T_c}(\tau - iT_c)$$

Nonetheless, in practice it is convenient to make $\hat{t}_s(\tau_{jk+1})$ refer to the <u>closest</u> code start time as follows:

$$\hat{t}_s(\tau_{jk+1}) = \hat{t}_s(\tau_{jk}) - (V_k + V_{p,k})(\tau_{jk+1} - \tau_{jk}) + u_{k+1} \cdot N_c \cdot T_c$$

where $u_{k+1} = \text{round}\left[\dfrac{\tau_{jk+1} - \hat{t}_s(\tau_{jk})}{N_c \cdot T_c}\right]$

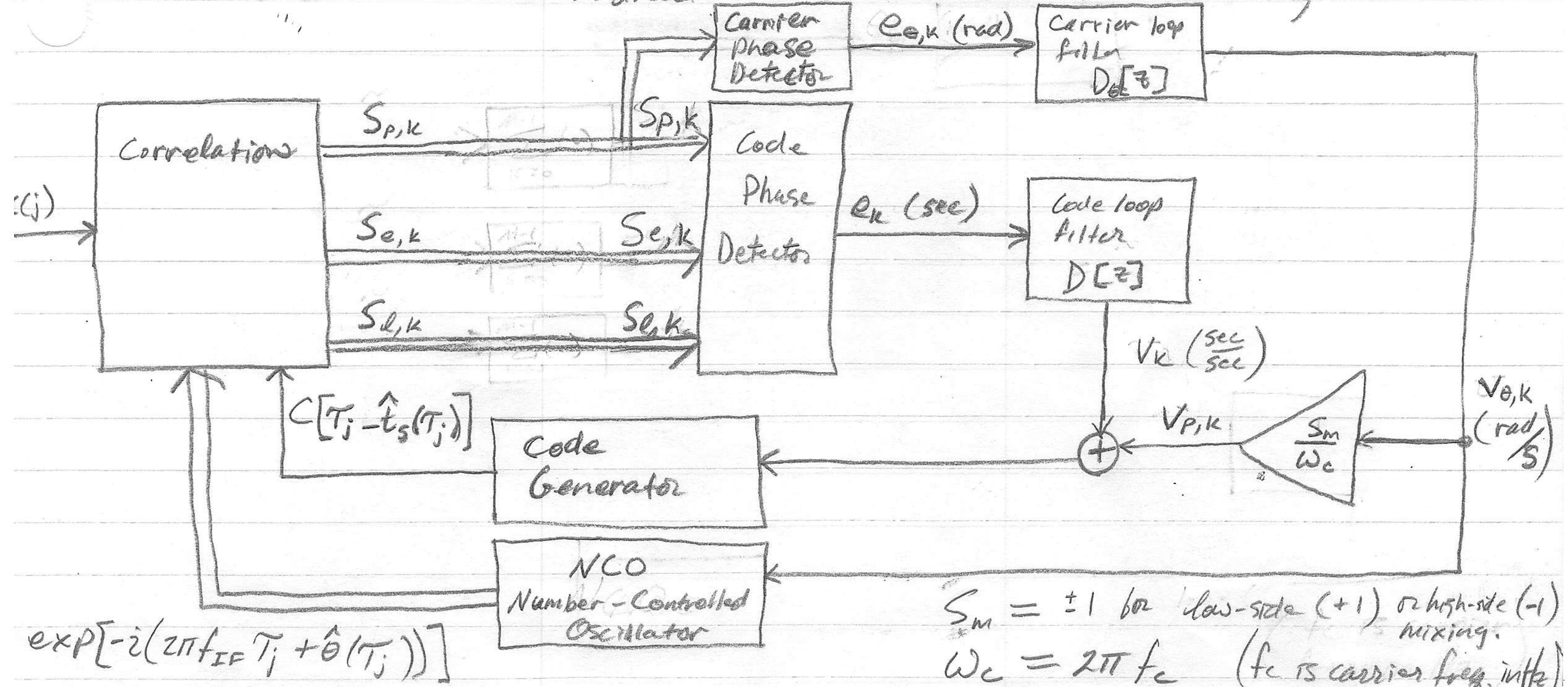
<u>Even easier</u>: Recognize that, in the absence of Doppler, one can assume that codes are separated by the code period $P_c = N_c \cdot T_c$. Thus, we could write

$$\hat{t}_s(\tau_{jk+1}) = \hat{t}_s(\tau_{jk}) + P_c \quad \text{(No Doppler)}$$

When Doppler is present, we can modify this as:

$$\hat{t}_s(\tau_{jk+1}) = \hat{t}_s(\tau_{jk}) + [1 - (V_k + V_{p,k})] \cdot P_c \quad \text{(With Doppler)}$$

Combined Code and Carrier Tracking



$S_m = \pm 1$ for low-side (+1) or high-side (−1) mixing.

$\omega_c = 2\pi f_c$ (f_c is carrier freq. in Hz)

$$\boxed{\Sigma} = \boxed{\sum_{j=j_k}^{j_k+N_k-1} (\cdot)}$$

$$x(j) = A(\tau_j)D[\tau_j - t_d(\tau_j)]C[\tau_j - t_s(\tau_j)]\cos[2\pi f_{IF}\tau_j + \theta(\tau_j)] + n(j)$$