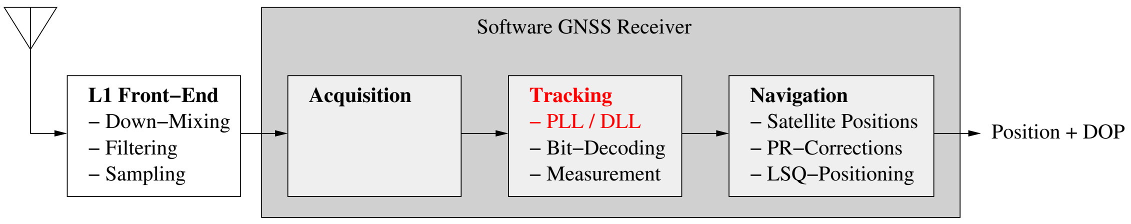
Lab6: Tracking

Satellite Navigation Laboratory, Summer Semester 2022

Institute for Communications and Navigation, Technical University of Munich

Contents

1 Introduction



We estimated the PRN sequence number of the received signal, the code-phase offset, and the Doppler shift in the acquisition process, by examining auto-correlation of the received signals and local replicas. Since acquisition is a global search method, which requires a large computational load, we track the changes of the code-phase offset and the Doppler shift once acquisition is successfully conducted (instead of repeating acquisition). In this session, we will discuss the Delay Lock Loop (DLL), the Phase Lock Loop (PLL), as well as the GNSS tracking loop (a combination of the DLL and the PLL).

2 Overview of Tracking Loops

As shown in Fig. 1, a signal tracking loop is similar to a feedback control system of autonomous vehicles. It consists of a plant, a discriminator (error computer), and a loop filter. First, the discriminator computes errors in the code-phase offset or carrier phase with the measured correlations as inputs. Using these computed errors, the filter give commands (control inputs) to the plant. With changed (controlled) parameters, the plant generates the outputs, such as local replicas of a PRN sequence or the carrier reference signals.

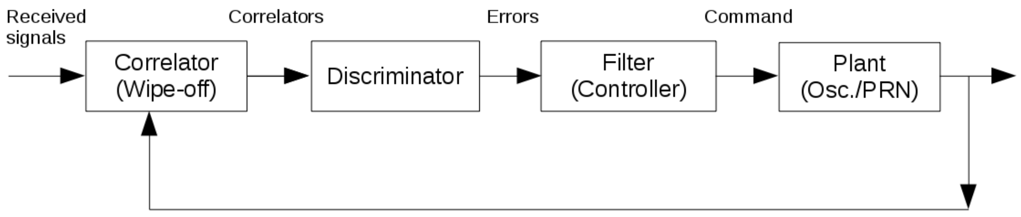


Figure 1: An overview of a tracking loop

2.1 Correlator

The discriminator takes correlators $\tilde{Z}$ as inputs, which can be described with correlation $\tilde{S}$ and noise $\tilde{\eta}$:

$$\tilde{Z} = \tilde{S} + \tilde{\eta}. \tag{1}$$

After the acquisition process, correlations can be modeled as $\tilde{S} = \sqrt{P_{rcv}}D\exp(j\Delta\theta)R(\Delta\tau)$ since the Doppler shift error is assumed to be almost zero ($\Delta f_D \sim 0$). With this result, correlators can be described as

$$\tilde{Z} = \tilde{S} + \tilde{\eta} = \sqrt{P_{rcv}}D\exp(j\Delta\theta)R(\Delta\tau) + \tilde{\eta}, \tag{2}$$

where $R(\Delta\tau)$ denotes the ambiguity function

$$R(\Delta\tau) = \frac{1}{T_{CO}}\int_0^{T_{CO}} x(t-\tau)x(t-\hat{\tau})dt. \tag{3}$$

Depending on when we take correlation measurements, we can have three sorts of the correlator measurements:

- The prompt correlator: $\tilde{Z}_P = \tilde{S}_P + \tilde{\eta}_P \cong \sqrt{P_{rcv}}D\exp(j\Delta\theta)R(\Delta\tau) + \tilde{\eta}_P$
- The early correlator: $\tilde{Z}_E = \tilde{S}_E + \tilde{\eta}_E \cong \sqrt{P_{rcv}}D\exp(j\Delta\theta)R(\Delta\tau - dT_c/2) + \tilde{\eta}_E$
- The late correlator: $\tilde{Z}_L = \tilde{S}_L + \tilde{\eta}_L \cong \sqrt{P_{rcv}}D\exp(j\Delta\theta)R(\Delta\tau + dT_c/2) + \tilde{\eta}_L$

By modifying $dT_c/2$ ($d \leq 1$), we can set different spacing for correlators.

2.2 Discriminator

A discriminator computes errors in the code phase delay or Doppler shift. The computed values are used as inputs of a loop filter.

2.3 Loop Filter (Controller)

A loop filter is used to reduce noise in discriminator outputs, and to generate commands for the plant. Filter transfer functions are complex functions of complex variables described in the Laplace domain:

Order	Input	$F(s)$	Bandwidth
1	Phase-jump	ω_n	$\omega_n/4$
2	Frequency-jump	$2\zeta\omega_n + \frac{\omega_n^2}{s}$	$\frac{\omega_n}{2}\left(\zeta + \frac{1}{4\zeta}\right)$
3	Frequency-ramp	$2\omega_n + \frac{2\omega_n^2}{s} + \frac{\omega_n^3}{s^2}$	$\omega_n/1.2$

where ζ is a damping ratio and ω_n is the natural frequency of the tracking loop.

The damping ratio and the bandwidth affect on the noise response. With a large damping ratio, we can have a small overshoot, but settling time is long. Generally, the damping ratio for GPS is set as $\frac{1}{\sqrt{2}}$. When bandwidth is small, a loop reacts slower, but noise can be more effectively suppressed.

Since the received signals are digitalized with a sample rate, the filters needs to be converted to the discrete time domain ($z-$plane) with the forward Euler transformation

$$s = \frac{1 - z^{-1}}{T_i z^{-1}}.$$

The order of the loop is determined with the highest order of z in $F(z)$.

3 Delay lock loop (DLL)

Without a priori information of data bits and carrier phase, the delay lock loop (DLL) is used to track the changes in the code-phase offset, and keep the local replica of the PRN code aligned with the received one, by controlling the clock of the local PRN code generator. As shown in Fig. 2a, the discriminator takes the early, prompt, and late correlators as inputs.

3.1 Coherent DLL

For the coherent DLL, we assume that the carrier phase is locked ($\Delta\theta = 0$), and the data bit is known. The early-minus-late discriminator is one of the discriminators that is used in the coherent DLL:

$$\begin{aligned} L_\tau &= \frac{1}{2}\left(Z_{E,I} - Z_{L,I}\right) \cdot D \\ &= \frac{1}{2}\left(S_{E,I} - S_{L,I}\right) \cdot D + \varepsilon_\tau \end{aligned} \tag{4}$$

Without considering the received power P_{rcv} in the correlation S, the discriminator can be rewritten as

$$\begin{aligned} L_\tau(\Delta\tau, d) &= \frac{1}{2}\left(I_E - I_L\right) + \varepsilon_\tau \\ &= \frac{1}{2}\left[R\left(\Delta\tau - \frac{dT_c}{2}\right) - R\left(\Delta\tau + \frac{dT_c}{2}\right)\right] + \varepsilon_\tau. \end{aligned} \tag{5}$$

3.2 Non-coherent DLL

Non-coherent DLL is used when the carrier phase is not locked. The followings are generally-used discriminators for the non-coherent DLL:

- Non-coherent early-minus-late power: $\frac{1}{2}\left[(I_E^2 + Q_E^2) - (I_L^2 + Q_L^2)\right]$
- Quasi-coherent dot product power: $\frac{1}{2}\left[(I_E - I_L)I_P + (Q_E - Q_L)Q_P\right]$
- Non-coherent early-minus-late envelope: $\frac{1}{2}\left[\sqrt{I_E^2 + Q_E^2} - \sqrt{I_L^2 + Q_L^2}\right]$

4 Phase Lock Loop (PLL)

Without a priori information of data bits, the phase lock loop (PLL) is applied to track the changes in the carrier, and keep the frequency and the phase of the local carrier synchronized with the received one. It controls the frequency of the numerically controlled oscillator (NCO) that generates the local replica of the carrier.

As shown in Fig. 2b, signal measurements are correlated before the discriminator to reduce the sensitivity to noise. The responding speed of the loop is dependent on the integration period of this correlation.

4.1 Costas discriminator

In the Costas discriminator, the navigation bits are squared, so it can be applied when the navigation bits are unknown:

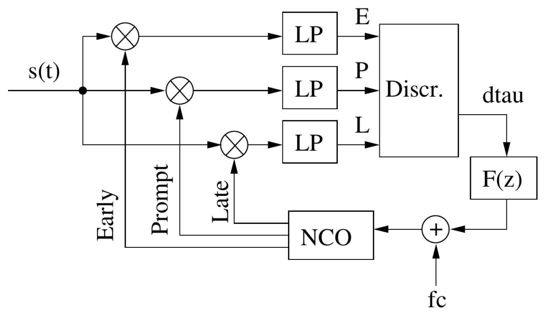
$$\begin{aligned} L_\theta &= Z_{P,I} \cdot Z_{P,Q} \\ &= S_{P,I} \cdot S_{P,Q} + \varepsilon_\theta \\ &= P_{rcv} D^2 R^2(\Delta\tau) \sin(\Delta\theta)\cos(\Delta\theta) + \varepsilon_\theta \end{aligned} \tag{6}$$

4.2 Navigation bit demodulation

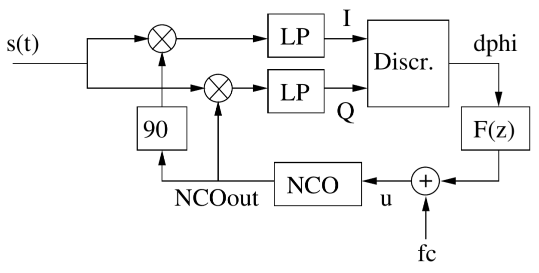
Once the DLL and the PLL are locked, we can assume $\Delta\tau \approx 0$ and $\Delta\theta \approx 0$. Hence, we can find each navigation bit using the inphase prompt correlation samples $S_{P,I} = \sqrt{P_{rcv}}D$. If the integration interval of correlation is 1 ms, 20 inphase samples should be used to determine one navigation bit since the bit duration is 20 ms.

5 GNSS tracking loop

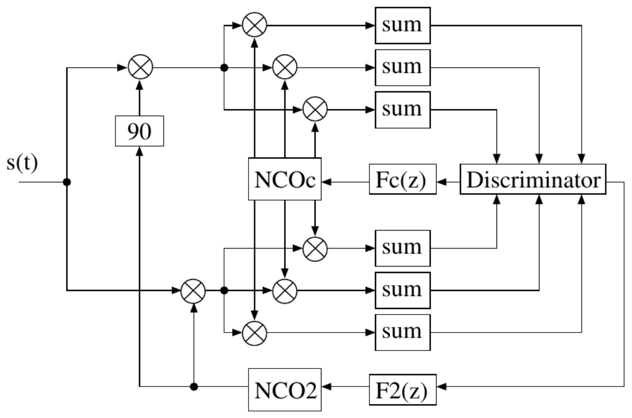
In a GNSS receiver, the code-phase and the carrier need to be tracked simultaneously. Fig. 2c shows the general tracking loop of a GNSS receiver with both the DLL and the PLL.



(a) A block diagram of the DLL. The NCO in the diagram is a PRN code generator, and the input of the loop $s(t)$ is the received signals modulated with the local replicas of the carrier



(b) A block diagram of the PLL. The NCO generates the carrier, and the input of the loop $s(t)$ is the received signals modulated with the prompt code



(c) GNSS Tracking Loop

References

[1] C. Günther, *Satellite Navigation.* Insitute for Communications and Navigation, Technical University of Munich, 2017.

[2] P. Misra and P. Enge, "Global positioning system: signals, measurements and performance second edition," *Massachusetts: Ganga-Jamuna Press*, 2006.

6 Homework

1. **Correlator spacing:** Explain the advantages and disadvantages of a small/large correlator spacing d of (5)

2. **Normalization of DLL discriminators:** Discriminators need to be normalized with the amplitude of received signals (normally unknown to the users) in practice. Assuming $\Delta\tau \approx 0$, explain how to normalize the coherent early-minus-late DLL discriminators in (5).

3. **PLL discriminator:** Which property should the PLL discriminators fulfill, considering that the GPS data bits are BPSK-modulated? Give an example of a proper discriminator of the PLL that satisfy the property (except for the Costas discriminator).
   Hint: Consider what happen to the carrier phase if the data bit is flipped.

4. **Normalization of the Costas discriminator:** Find the proper scaling factor for the Costas discriminator.

5. **Navigation bit demodulation:** Explain how to find the navigation bit D transmitted with the 20 inphase samples, using the maximum a posteriori approach. Assume the code phase and carrier phase are locked ($\Delta\tau \approx 0$, $\Delta\theta \approx 0$), and the same probability of $+1$ or -1 for the navigation bits.

7 Lab Tasks

1. **Coherent DLL discriminator:** : Implement the coherent early-minus-late discriminator in `dllDiscr.m`, then visualize the discriminator curve for each spacing factor $d \in \{1,\ 0.5,\ 0.1\}$ using `plotcoherentdllDiscr(d)`.

2. **Non-coherent DLL discriminator:** Extend `dllDiscr.m` with the three non-coherent DLL, then visualize them with `plotnoncoherentdllDiscr(d)`.

3. **Normalization of DLL discriminators:** Implement the normalized discriminators (derived in Homework 2) in `dllDiscr.m`, then test them with `plottruedllDiscr(d,CN0)`.

4. **PLL discriminator:** Implement two PLL discriminators (the one derived in Homework 3 and the Costas discriminator) in `pllDiscr.m`, then test them with `plottruepllDiscr(CN0)`.

5. **GNSS tracking loop:** Implement the navigation bit demodulation method (derived in Homework 5) in `detectBit.m`, and the loop filters in `dllLoopFilter.m` and `pllLoopFilter.m`.
   Then, test the entire GNSS tracking loop using `postProcessing('testIFdata.bin', sampling_frequency, intermediate_frequency)`. The sampling and intermediate frequencies can be found in `testIFdata.txt`.