The the correct values for the observer position were `xObs = 55.5 m, dObs = 12.4 m`. Note that to obtain these values within 0.1 meters requires searching for the values that minimize some norm of the difference (e.g., mean square) between the Doppler time history you extracted from the `trainout.wav` file and the time history produced by your code.

```
function [fDVec,tVec] = ...
      simulateTrainDoppler(fc, vTrain, t0, x0, xObs, dObs, delt, N, vs)
% simulateTrainDoppler : Simulate the train horn Doppler shift scenario.
%
% INPUTS
%
% fc ------ train horn frequency, in Hz
%
% vTrain -- constant along-track train speed, in m/s
%
% t0 ------ time at which train passed the along-track coordinate x0, in
%           seconds
%
% x0 ------ scalar along-track coordinate of train at time t0, in meters
%
% xObs ---- scalar along-track coordinate of observer, in meters
%
% dObs ---- scalar cross-track coordinate of observer, in meters (i.e.,
%           shortest distance of observer from tracks)
%
% delt ---- measurement interval, in seconds
%
% N ------- number of measurements
%
% vs ------ speed of sound, in m/s
%
%
% OUTPUTS
%
% fDVec --- N-by-1 vector of apparent Doppler frequency shift measurements as
%           sensed by observer at the time points in tVec
%
% tVec ---- N-by-1 vector of time points starting at t0 and spaced by delt
%           corresponding to the measurements in fDVec
%
%+------------------------------------------------------------------------------+
% References:
%
%
% Author:
%+==============================================================================+

tVec = [0:N-1]'*delt + t0;
fDVec = zeros(N,1);
rObs = [xObs; dObs];
vTrainVec = [vTrain;0];
```

```
for ii=1:N
  deltTOF = 0;
  for jj=1:5
    rTrain = [(tVec(ii) - deltTOF)*vTrain + x0; 0];
    deltTOF = norm(rTrain - rObs)/vs;
  end
  rG = rObs - rTrain;
  rGnorm = rG/norm(rG);
  vLos = -vTrainVec'*rGnorm;
  fr = fc/(1 + vLos/vs);
  fDVec(ii) = fr - fc;
end
```

Step 1: Frequency conversion to baseband. A nominal baseband frequency $f_{b,\text{nom}} = 0$ implies that the nominal frequency of the mixing signal is $f_{l,\text{nom}} = f_c$. With a fractional frequency error of $\beta = \Delta f / f$, the actual mixing signal frequency is

$$f_l = f_c\,(1+\beta)$$

We are told that $\beta < 0$, from which it follows that $f_l < f_c$. The actual baseband frequency f_b is thus

$$\begin{aligned} f_b &= f_c - f_l \\ &= f_c - f_c\,(1+\beta) \\ &= -\beta f_c > 0 \end{aligned}$$

Step 2: Apparent Doppler measurement via sampling. After frequency down-conversion and filtering, the received signals looks like

$$\cos(2\pi f_b t)$$

where f_b is given above and t is true time. We now sample this signal to measure its frequency. This step is exactly the same as the frequency measurement step of Problem 3 from Problem Set 1 except that the signal's frequency is f_b instead of f_c.

Substituting into the equation for f_m from Problem 3 (whose answer is copied below), we have

$$\begin{aligned} f_m &= f_b \left(\frac{1}{1+\beta} \right) \\ &= f_c \left(\frac{1-(1+\beta)}{1+\beta} \right) \\ &= f_c \left(\frac{1}{1+\beta} - 1 \right) \end{aligned}$$

Given that the nominal value of the baseband frequency was $f_{b,\text{nom}} = 0$, it follows that the apparent Doppler is

$$f_D \triangleq f_m - f_{b,\text{nom}} = f_m = f_c \left(\frac{1}{1+\beta} - 1 \right) = -f_c \left(\frac{\beta}{1+\beta} \right)$$

Thus, for the case $f_l < f_c$, the measured Doppler after downmixing and sampling is the same as the direct measurement of Doppler at f_c, as done in Problem Set 1, Number 3.

For $f_l > f_c$ (the high-side-mixed case) we have that $f_b < 0$. In this case, the positive frequency component of the real-valued sinusoid crosses over to the negative side of frequencies. But, at the same time, the negative frequency component crosses over to the positive side. As a result, the magnitude of the Fourier transform of the resulting signal is the same as in the case of $f_l < f_c$. If we attempt to measure the Doppler frequency using the above sampling approach, then we will always measure a positive Doppler frequency. However, we can distinguish high-side from low-side mixing as follows:

Slightly increase the frequency f_l and perform the same measurement again:

- If the measured f_D is now smaller in magnitude $\rightarrow$ low-side mixing.
- If the measured f_D is now larger in magnitude $\rightarrow$ high-side mixing.

Answer to Problem 3 from Problem Set 1

Suppose a signal $x(t) = \cos(2\pi f_c t)$ is intended to be sampled at a nominal interval Δt_{nom}, but the actual sampling interval Δt is different from this because the clock driving the sampler has a fractional frequency error β.

Due to the fractional frequency error, Δt is related to Δt_{nom} by

$$\Delta t = \Delta t_{\text{nom}} \left(\frac{1}{1+\beta} \right)$$

This makes sense: if our sampling clock is running fast ($\Delta f / f > 0$), then the actual sampling interval Δt will be shorter than the nominal interval Δt_{nom}.

The number of sample intervals you will measure in one period of the incoming signal (including fractional intervals) is $N = T/\Delta t$, where $T = 1/f_c$ is the true period. From this measurement, you'll infer a period

$$T_m = N \Delta t_{\text{nom}} = T \left(\frac{\Delta t_{\text{nom}}}{\Delta t} \right)$$

Thus, the measured frequency of $x(t)$ is

$$f_m = 1/T_m = f_c \left(\frac{\Delta t}{\Delta t_{\text{nom}}} \right) = f_c \left(\frac{1}{1+\beta} \right)$$

which yields an apparent Doppler

$$f_D \triangleq f_m - f_c = f_c \left(\frac{1}{1+\beta} - 1 \right) = -f_c \left(\frac{\beta}{1+\beta} \right)$$

A. We are given that

$$\begin{aligned} L_1 &= 1 \text{ dB} \\ &= 1.2589 \end{aligned}$$

For a passive element, we have that

$$\begin{aligned} F_1 &= L_1 \\ &= 1.2589 \end{aligned}$$

The gain is the reciprocal of loss:

$$\begin{aligned} G_1 &= \frac{1}{L_1} \\ &= -1 \text{ dB} \\ &= 0.7943 \end{aligned}$$

To compute the temperature T_1, we have that

$$\begin{aligned} T_1 &= (F_1 - 1)T_{in} \\ &= 0.2589 \times 290 \\ &= 75.0884 \text{ K} \end{aligned}$$

B. For the LNA, we are given that

$$\begin{aligned} F_2 &= 1.5 \text{ dB} \\ &= 1.4125 \end{aligned}$$

The temperature T_2 is then computed as

$$\begin{aligned} T_2 &= (F_2 - 1)T_{in} \\ &= 0.4125 \times 290 \\ &= 119.6359 \text{ K} \end{aligned}$$

Also, we are given that $G_2 = 50 \text{ dB} = 10^5$.

C. It is given that

$$L_3 = 6.48 \text{ dB}; \quad L_4 = 0.6 \text{ dB}; \quad L_5 = 9.8 \text{ dB}$$

The combined loss L_{345} can be computed by adding the individual losses in dB:

$$\begin{aligned} L_{345} &= L_3(\text{dB}) + L_4(\text{dB}) + L_5(\text{dB}) \\ &= 6.48 + 0.6 + 9.8 \text{ dB} \\ &= 16.88 \text{ dB} \\ &= 48.7528 \end{aligned}$$

Since all components are passive, $F_{345} = L_{345}$. The temperature T_{345} is given as

$$\begin{aligned} T_{345} &= (F_{345} - 1)T_{in} \\ &= 47.7528 \times 290 \\ &= 13848.3262 \text{ K} \end{aligned}$$

D. It is given that $T_A = 100$ K. Using Friis's formula, we get T_R as

$$\begin{aligned} T_R &= T_1 + \frac{T_2}{G_1} + \frac{T_{345}}{G_1 G_2} \\ &= 75.0884 + \frac{119.6359}{0.7943} + \frac{13848.3262}{0.7943 \times 10^5} \\ &= 225.8808 \text{ K} \end{aligned}$$

Thus, the system temperature is

$$\begin{aligned} T_S &= T_A + T_R \\ &= 100 + 225.8808 \text{ K} \\ &= 325.8808 \text{ K} \end{aligned}$$

The combined losses L_{345} contribute 0.1743 K to the system temperature, even though T_{345} is incredibly high. This is because of the large amplification provided by the LNA earlier in the signal chain.

E. The effective noise floor N_0 is computed as

$$\begin{aligned} N_0 &= kT_S \\ N_0(\text{dBW/Hz}) &= k(\text{dBW/K-Hz}) + T_S(\text{dBK}) \\ &= -228.6 + 10 \times \log_{10}(325.8808) \\ &= -228.6 + 25.1306 \\ &= -203.4694 \text{ dBW/Hz} \end{aligned}$$

For signal power $C = -162.5$ dBW

$$\begin{aligned} C/N_0 &= -162.5 - (-203.4694) \\ &= 40.9694 \text{ dB-Hz} \end{aligned}$$

For signal power $C = -154.5$ dBW

$$\begin{aligned} C/N_0 &= -154.5 - (-203.4694) \\ &= 48.9694 \text{ dB-Hz} \end{aligned}$$

F. Since the output of the 8-way splitter is the end of the RF chain, the $(C/N_0)_{sp}$ here is the same as the effective C/N_0 corresponding to T_S at the beginning of the cascade with ideal components thereafter. We know that the carrier power at the splitter output is $C_{sp} = G_1 G_2 G_{345} C$. Thus, to maintain the same carrier-to-noise ratio, we have that

$$\begin{aligned} N_{0,sp} &= G_1 G_2 G_{345} N_0 \\ &= -1 + 50 - 16.88 - 203.4694 \text{ dBW/Hz} \\ &= -171.3494 \text{ dBW/Hz} \end{aligned}$$

G. It is easiest to measure $N_{0,sp}$ as a ratio against a known carrier power. Assume that the antenna gain pattern of the receiving antenna is known. Now assume that a well-calibrated transmitter with a known antenna gain pattern is configured to emit a signal of known power toward the receiving antenna. Note that this must be done with the receiving antenna in its normal situation: outdoors in full view of the sky so that all contributors to antenna noise are included in $N_{o,sp}$.

Accounting for the known transmit and receive antenna gain patterns, and the distance between them, the carrier power C at the output of an isotropic antenna model can be computed. Since the gain of each element in the downstream receiver chain is known, the expected power of the carrier at the output of the splitter can also be computed. If G is the combined gain of all RF elements, then $C_{sp} = GC$.

Now we measure the carrier-to-noise ratio $(C/N_0)_{sp}$ of the calibration signal at the output of the splitter. If the calibration signal is a pure tone, its carrier-to-noise ratio can be easily calculated from a power spectrum. It it's a GNSS-like signal, then the signal must be tracked and the carrier-to-noise ratio calculated from the phase noise or the statistics of the complex (I and Q) accumulations. From this measurment, we can estimate $N_{0,sp}$ as

$$N_{0,sp} = \frac{C_{sp}}{\left(\frac{C}{N_0}\right)_{sp}}$$

Problem Statement: To measure the receiver temperature T_R of a GNSS receiver and antenna setup, a friend recommends placing the receiver's antenna in a RF test enclosure such as the one in the Radionavigation Laboratory (seen here `https://ramseytest.com/forensic-test-enclosures`), but cryogenically cooled down to 5 K, and then measuring the noise power in the raw samples generated by the receiver. The enclosure effectively isolates the antenna from environmental noise. The antenna is an active antenna consisting of a patch element, a (passive) filter, and an amplifier. Is this a valid approach for measuring T_R? Why or why not?

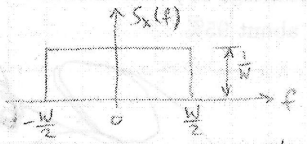
Answer: This would not be a valid approach for measuring T_R. Recall that the system temperature, to which the noise power in the raw samples is proportional, is $T_S = T_A + T_R$. Our friend's idea seems to be this: if we could drive T_A to zero without affecting T_R, then T_S becomes a proxy for T_R. Indeed, cooling the RF enclosure to 5 K would make T_A almost negligible. But it would also affect T_R! The noise figures of the filter and LNA that are built into the active antenna are temperature sensitive. And these components are part of the full cascade that contributes to T_R. Thus, unless we expected to always operate the receiver with the critical first filter and first LNA cryogenically cooled to 5 K, the noise power we'd measure would not be an accurate reflection of operational T_R.

Hypothetical system's spreading waveform: $S_X(f) = \frac{1}{W}\Pi\left(\frac{f}{W}\right)$

↳ two-sided bandwidth W Hz

a) Find autocorrelation function $R_X(\tau)$

$$R_X(\tau) = \mathcal{F}^{-1}\{S_X(f)\}$$

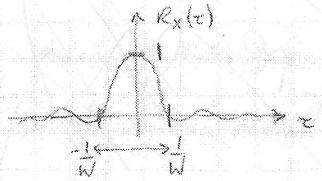


Use the time scaling property of Fourier transforms: $x(kt) = \mathcal{F}^{-1}\left\{\frac{1}{|k|}X\left(\frac{f}{k}\right)\right\}$

and recall that $\mathcal{F}^{-1}\{\Pi(f)\} = \text{sinc}(t)$.

Thus, $\boxed{R_X(\tau) = \text{sinc}(W\tau)}$ ✓

b) Width of main peak in $R_X(\tau)$



Rough Sketch

if $\tau = \frac{1}{W} \Rightarrow R_X(\tau) = \text{sinc}(1)$

$= 0$

Thus, width is $\frac{1}{W} + \frac{1}{W} = \frac{2}{W}$

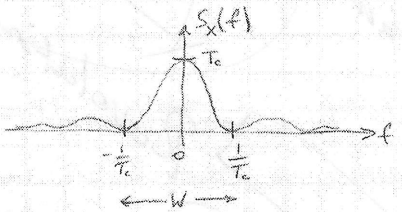
$\boxed{\text{Width} = \frac{2}{W}}$ ✓

c) Random Binary Sequence

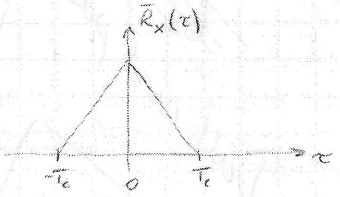
↳ $S_X(f) = T_c\,\text{sinc}^2(fT_c)$, which has a corresponding autocorrelation function

$$\bar{R}_X(\tau) = \Lambda\left(\frac{\tau}{T_c}\right) = \Pi\left(\frac{\tau}{T_c}\right) * \Pi\left(\frac{\tau}{T_c}\right), \quad \text{where} \quad \Lambda\left(\frac{\tau}{T_c}\right) = \begin{cases} 1 - \frac{|\tau|}{T_c} & -T_c < \tau < T_c \\ 0 & \text{otherwise} \end{cases}$$

↑ has an equivalent null-to-null bandwidth $B_t = W$



↳ $W = \frac{2}{T_c} \Rightarrow T_c = \frac{2}{W}$



$\text{Width} = 2T_c = 2\left(\frac{2}{W}\right) = \frac{4}{W}$

$\boxed{\text{Width} = \frac{4}{W}}$ ✓

$\Rightarrow$ Twice the width of main peak in $R_X(\tau)$

d) The peak (from first left zero-crossing to first right zero-crossing) of the autocorrelation function of the hypothetical waveform is half the width of the peak of the traditional random binary sequence autocorrelation function ($2/W$ vs. $4/W$). So one might expect that the hypothetical waveform would lead to more accurate measurements of the location τ_{peak} of the peak, which is related to pseudorange. On the other hand, one might note that the peak of the traditional waveform's autocorrelation function is *sharper* than the rounded $\text{sinc}(W\tau)$ peak of the hypothetical waveform's autocorrelation function, and thus one might conclude that the traditional waveform leads to more accurate code phase determination.

To resolve this puzzle, one may appeal to a theoretical analysis of the lower bound of pseudorange precision, specifically, the Cramer-Rao lower bound. This analysis reveals that pseudorange variance goes inversely as the so-called mean square bandwidth, which is given by

$$B_{\text{ms}} = \int_{-\infty}^{\infty} f^2 S_n(f) df$$

where $S_n(f)$ is the spreading waveform's normalized power spectral density (normalized in that it integrates to 1). The presence of the f^2 in this integral heavily weights frequencies far from 0.

If your receiver captures exactly W Hz of spectrum around the carrier, then the mean square bandwidth of the hypothetical system's spreading waveform is greater than that of the traditional waveform (truncated to a W-Hz width). But if your receiver captures a wide bandwidth (e.g., 10W as in the problem statement), then the sidelobes of the traditional spreading waveform's power spectrum will make its B_{ms} much wider than that of the hypothetical waveform. The larger B_{ms} is manifest in the sharp peak of the traditional waveform's autocorrelation function.

Note that the pseudorange variance also goes inversely as C/N_0, so that a higher C/N_0 leads to a lower variance. But this effect applies to both waveforms equally. Thus, under whatever C/N_0 arises in a given scenario, the waveform with the wider B_{ms} will enjoy greater precision in the measurement of τ_{peak}.

You are not expected to know, or be able to derive, the foregoing material at this stage of the class. But you should be able to decide which autocorrelation function would be more favorable for a simple case where τ_{peak} is measured by two samples spaced by an interval $\Delta\tau = 0.2/W$. The situation is shown in Fig. 1. Because the slope of the red function is larger in magnitude than the slope of the blue function at the points indicated, it follows that the red points are more sensitive to slight changes in the location τ_{peak} than the blue points. Thus, determination of τ_{peak} will be easier for the red function than for the blue funcion. And since C/N_0 will affect both functions equivalently, the more sensitive function (for this spacing of samples) is preferable regardless of the value of C/N_0.

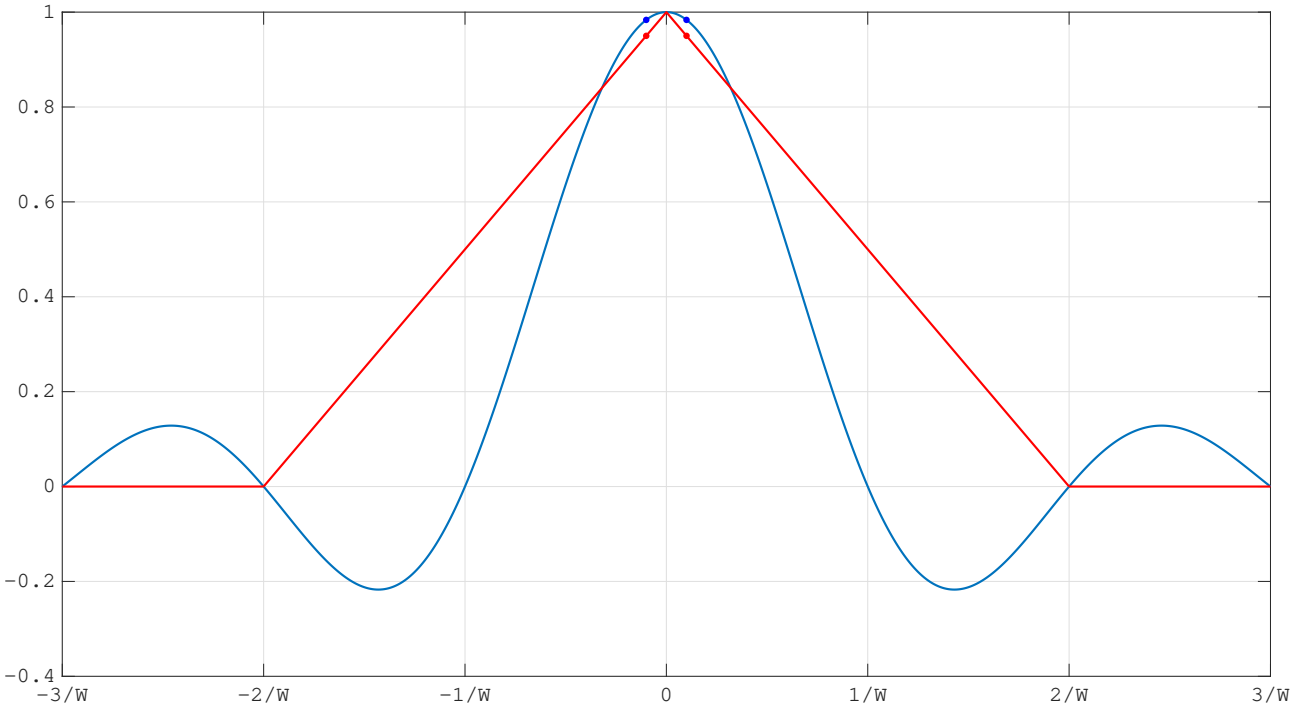


FIGURE 1. Comparison of autocorrelation functions for two different waveforms.

e) The random binary sequence, with its $T_c\text{sinc}(T_c f)$ power spectrum, has historically been preferred simply because it is easy to generate, both in the transmitter and the receiver. Implementing a waveform with a power spectrum equal to $(1/W)\Pi(f/W)$ would require stringent filtering and/or pulse shaping at the transmitter and pulse shaping at the receiver. Pulse shaping is a shaping of the support function $p(t)$ introduced in lecture. For the traditional random binary sequence, $p(t) = \Pi(t)$, which is nice and square and time-limited. For the hypothetical waveform, the support function would be a sinc-type function, which is impossible to generate exactly in practice because it is not time-limited.

One should not conclude from this that the traditional binary sequence is the best waveform to use for GNSS. In fact, recent research suggests that there are better candidates that are both easy to implement and more spectrally efficient. However, the hypothetical waveform introduced in this problem is not one of these promising candidates.

Let $r(t)$ denote the signal received at the receiver's antenna. In presence of a single multi-access interferer, this is given as

$$r(t) = r_S(t) + r_I(t) + n(t)$$

where, $r_S(t)$ is the desired signal, $r_I(t)$ is the interfering multi-access signal, and $n(t)$ is complex AWGN. We can represent the desired signal as

$$r_S(t) = \sqrt{P_S} C(t) D(t) \exp\left[j\theta(t)\right]$$

where, $C(t)$ is the spreading code with the given power spectral density $S_C(f)$, $D(t)$ are the data bits, and $\theta(t)$ is the carrier-phase. On receiving this signal, the receiver correlates it with a phase-aligned local replica $l(t) = C(t)\exp\left[j\theta(t)\right]$. The output of this correlator is

$$\begin{aligned} Y(t) &= r^*(t) \cdot l(t) \\ &= S(t) + I(t) + N(t) \end{aligned}$$

where

$$\begin{aligned} S(t) &= \sqrt{P_S} D(t) \\ I(t) &= r_I^*(t) C(t) \exp\left[j\theta(t)\right] \\ N(t) &= \left[n_I(t) - j n_Q(t)\right] C(t) \exp\left[j\theta(t)\right] \end{aligned}$$

As mentioned in the lecture, when the desired signal, interfering signal, and noise are statistically independent, we need only focus on $I(t)$ to analyze the impact of interference. The autocorrelation of $I(t)$ can be written

$$\begin{aligned} R_I(\tau) &= \mathbb{E}[I^*(t) I(t)] \\ &= \mathbb{E}[r_I^*(t) r_I(t-\tau)] \mathbb{E}[C^*(t) C(t-\tau)] \mathbb{E}[\exp\left[-j\theta(t)\right] \exp\left[j\theta(t-\tau)\right]] \end{aligned}$$

The PSD of $I(t)$ is the Fourier transform of $R_I(\tau)$, and is given as

$$\begin{aligned} S_I(f) &= \mathcal{F}[R_I(\tau)] \\ &= S_C(f) * S_{r_I}(f) * \delta(f + f_D) \\ &= S_C(f) * S_{r_I}(f + f_D) \end{aligned}$$

For simplicity, assume $f_D = 0$. Then

$$S_I(f) = S_C(f) * S_{r_I}(f)$$

In case of multi-access interference, we have that $S_{r_I}(f) = P_I S_C(f)$. Thus,

$$\begin{aligned} S_I(f) &= S_C(f) * P_I S_C(f) \\ &= P_I \int_{-\infty}^{\infty} S_C(f-\nu) S_C(\nu) \mathrm{d}\nu \end{aligned}$$

It is given that

$$S_C(f) = T_C \Pi(f T_C)$$

Notice that $S_C(f) = S_C(-f)$. Thus,

$$S_I(f) = P_I \int_{-\infty}^{\infty} S_C(\nu - f) S_C(\nu) \mathrm{d}\nu$$

As discussed in lecture, due to the low pass nature of the integrator, we need only focus on $I_0 \triangleq S_I(0)$.

$$\begin{aligned} I_0 &= P_I \int_{-\infty}^{\infty} S_C^2(\nu) \mathrm{d}\nu \\ &= P_I \int_{-\infty}^{\infty} [T_C \Pi(\nu T_C)]^2 \mathrm{d}\nu \\ &= P_I \times T_C^2 \times \frac{1}{T_C} \\ &= P_I T_C \end{aligned}$$

Note that this is greater than the multi-access interference seen in the traditional GPS spreading code. With M multi-access signals, the overall multi-access interference is

$$I_0 = (M-1) P_I T_C$$

For $I_0 > N_0$, we have that multi-access interference in greater than N_0 when

$$\begin{aligned} (M-1) P_I T_C &> N_0 \\ \Rightarrow M &> \frac{N_0}{P_I T_C} + 1 \end{aligned}$$

The plot you generate should look like Fig. 1 in Problem Set 2, or a 90-degrees rotated version of this. You'll be able to determine which signal lies along the x axis, and which along the y axis, by examining their amplitudes: the carrier for the C/A code has a higher amplitude than the one for the P(Y) code. The smallest chip interval for the C/A code is $1e6/1023 = 977.5ns$. The smallest chip interval for the C/A code is $1e6/10230 = 97.75ns$. The amplitude ratio of the C/A code to the P(Y) code is approximately 1.4.

Part (a). The value of N was chosen to be 1024, and the `randn` function in `MATLAB` was used to generate different pairs of random sequences. These sequences were used to compute $R_{ab}(0) = \sum_{n=1}^{N} a'_n b'_{n+0}$. The variance of $R_{ab}(0)$ obtained using 100 different pairs of random sequences was highly variable, but fairly close to the expected value of $N = 1024$. With 100,000 different pairs, the variance obtained was 1022, which is quite close to N. It might be argued that this value is less than that for a truly random sequence because `MATLAB`'s implementation of `randn` is not perfect.

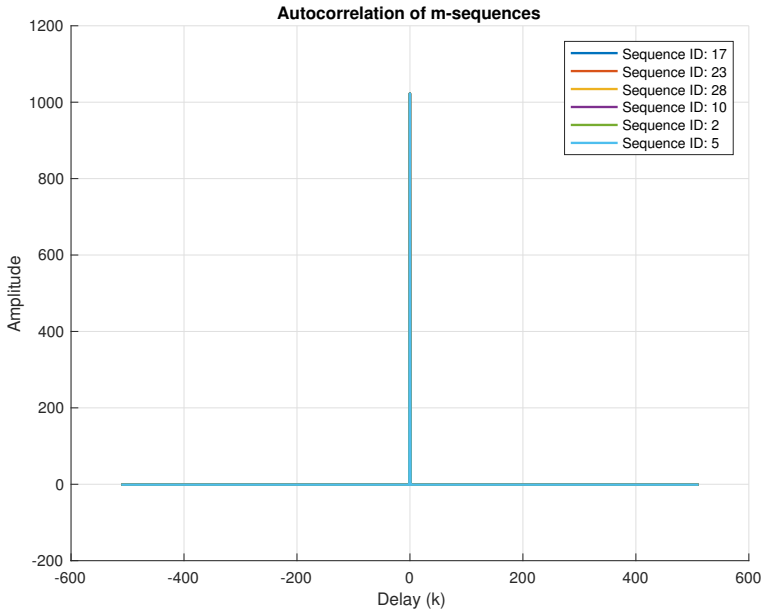


FIGURE 1. Autocorrelation for 6 different LFSR m-sequences.

Part (b). The function `generateLfsrSequence` was used to generate 6 randomly-chosen m-sequences with $n = 10$. Subsequently, the autocorrelation

$$R_a(k) = \sum_{n=1}^{N} a'_n a'_{n+k}$$

and the crosscorrelation

$$R_{ab}(k) = \sum_{n=1}^{N} a'_n b'_{n+k}$$

were computed for all combinations of m-sequences. Finally, the maximum crosscorrelation

$$\max_k R_{ab}(k)$$

was also computed. For $N = 1023$ and $M = 6$, the lower bound on maximum crosscorrelation is

$$N\sqrt{\frac{M-1}{MN-1}} = 29.2$$

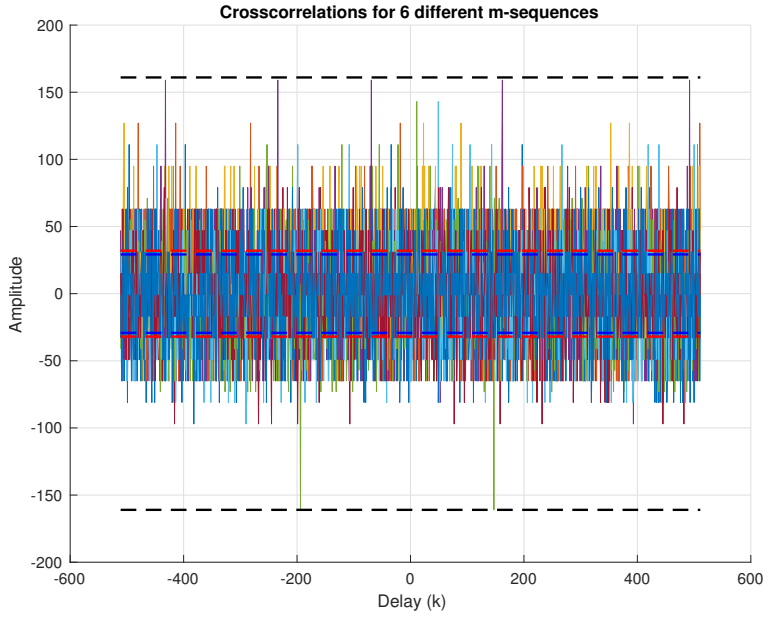


FIGURE 2. Cross-correlation for 6 different m-sequences with lower bound (blue-dashed), approximate lower bound (red-dashed), and empirical maximum auto-correlation (black-dashed).

and the approximate lower bound is

$$\sqrt{N} = 31.98$$

Figures 1 and 2 show the autocorrelation and crosscorrelation characteristics obtained for 6 different LFSR m-sequences. It is verified that the autocorrelation function is $(N+1)\delta(k) - 1$ and that the the absolute maximum crosscorrelation (161 in this case) is greater than the lower bound computed above. It is also noted that the approximate lower bound of $\sqrt{N}$ close to the exact lower bound for $M = 6$.

Part(c). Figures 3 and 4 show the autocorrelation and crosscorrelation characteristics of the three types of random sequences explored in this problem. The ratio of the maximum of $R_{X_1}(\tau_i)$ to the maximum of $R_{X_1,X_2}(\tau_i)$ for different types of sequences were found to be

$$\frac{\max_i R_{X_1}(\tau_i)}{\max_i R_{X_1,X_2}(\tau_i)} = \begin{cases} 8.9271, & \texttt{codeType = 'rand'} \\ 9.5938, & \texttt{codeType = 'pi'} \\ 8.0458, & \texttt{codeType = 'mseq'} \end{cases}$$

From Figure 3, it is clear that m-sequences have the best autocorrelation properties as they produce a Kronecker delta as the autocorrelation function. Figure 4 reveals that the random sequence derived from the digits of π have the best crosscorrelation properties in this case.

Not much can be concluded about the general superiority of any one of the three methods over the other two by comparison of a single set of three oversampled codes. However, we know this much: Assuming 'rand' and 'pi' produce perfectly random sequences, then for large N, both of these will yield crosscorrelations distributed as $R_{ab}(k) \sim \mathcal{N}(\mu, \sigma^2)$, where $\mu = 0$

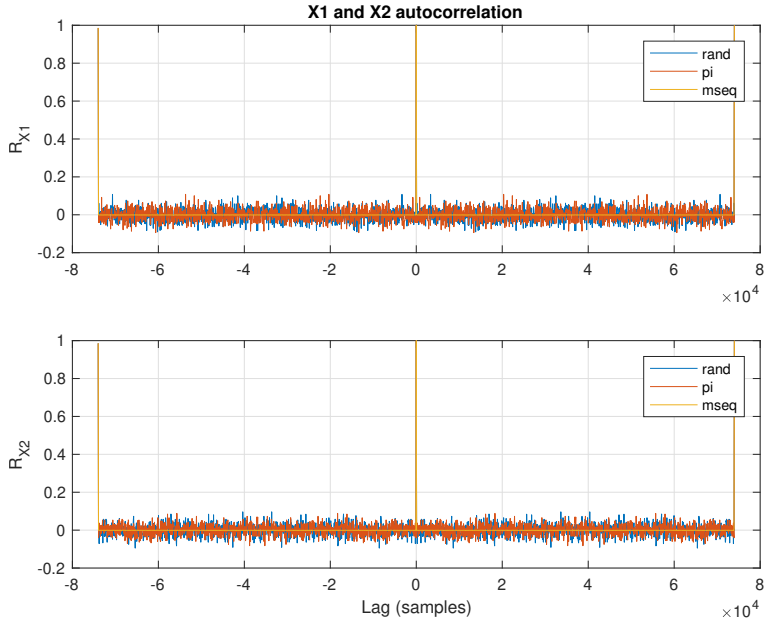


FIGURE 3. Comparison of autocorrelation for `rand`, `pi`, and `mseq` random sequences.

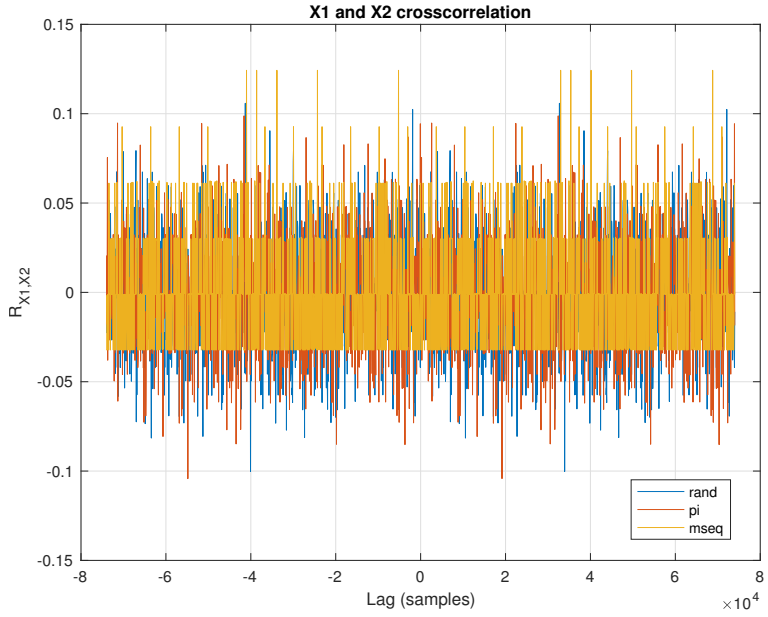


FIGURE 4. Comparison of crosscorrelation for `rand`, `pi`, and `mseq` random sequences.

and $\sigma^2 = N$. Note that this implies that crosscorrelation values for random sequences are unbounded. Similarly, the crosscorrelation values for general m-sequences is unbounded. Thus, any one of 'rand', 'pi', or 'mseq' would be problematic as sources for GNSS spreading codes.

From the given set of linear feedback shift register taps, we must first compute the autocorrelation of the sequences represented as (+/-1) to determine which of the taps actually generate m-sequences (i.e., maximal-length sequences). The taps that represent m-sequences will have -1 values for all indexes except one, which has a value of $2^9 - 1 = 511$. We find that $f_1(D)$, $f_2(D)$, and $f_4(D)$ are valid m-sequences while $f_3(D)$, $f_5(D)$ and $f_6(D)$ are not.

Now we can check which pairs of m-sequences could generate Gold codes; that is, which pairs could be *preferred pairs*. The crosscorrelation among all the the Gold codes in the family generated by a preferred pair (including the two preferred pair sequences themselves) will be 3-valued. For $n = 9$, these values are {-33, -1, 31}. Experimenting with the characteristic polynomials, we find that only $f_1(D)$ and $f_4(D)$ generate a crosscorrelation with only these three values. Thus, only these two sequences satisfy the crosscorrelation requirement for a preferred pair.