Posting Date: November 18, 2015.

Problem 1: In the lecture, we modeled a discrete-time GNSS signal exiting the RF front-end as

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

with the following definitions, where all time-based quantities are in units of seconds

τ_j	sample time expressed in receiver time
$A(\tau_j)$	signal amplitude
$D(\tau) = \sum_{i=-\infty}^{\infty} d_i \Pi_{T_d}(\tau - iT_d)$	± 1-valued navigation data bit stream
d_i	navigation data bits
Π_{T_d}	unit pulse of width T_d
T_d	navigation data bit interval
t_d	start time of some data bit reference epoch
$C(\tau) = \sum_{i=-\infty}^{\infty} c_{\mathrm{mod}(i,N_c)} \Pi_{T_c}(\tau - iT_c)$	spreading (ranging) code sequence
N_c	number of chips in spreading (ranging) code sequence
$t_s(\tau_j)$	code phase expressed as a code start time
c_i	± 1-valued spreading (ranging) code chip values
f_{IF}	intermediate frequency in Hz
$\theta(\tau_j)$	beat carrier phase in radians
$f_D(\tau_j) = \frac{1}{2\pi}\frac{d[\theta(\tau)]}{d\tau}\Big\vert_{\tau=\tau_j}$	apparent Doppler frequency shift in Hz
$n(j)$	element of a zero-mean discrete-time Gaussian white noise sequence with $\mathbb{E}[n(j)n(k)] = \sigma_n^2\delta_{j,k}$
$\delta_{j,k}$	Kronecker delta function
T	uniform sample interval: $T = \tau_{j+1} - \tau_j$

The signal's carrier-to-noise ratio C/N_0 is related to σ_n^2 by

$$C/N_0 = \frac{A^2}{4\sigma_n^2 T}.$$

Complex subaccumulations $\tilde{S}_k$ are computed within a GNSS receiver according to the following recipe:

$$\tilde{S}_k = \sum_{j=j_k}^{j_k+N_k-1} x(j)\exp\left\{-i\left[2\pi f_{IF}\tau_j + \hat{\theta}(\tau_j)\right]\right\} C[\tau_j - \hat{t}_{s,k}].$$

In this recipe, $i = \sqrt{-1}$, $\hat{t}_{s,k}$ is the estimate of the code start time, $\hat{\theta}(\tau_j)$ is the estimate of the beat carrier phase at sample time τ_j, N_k is the number of samples that participate in the kth accumulation (note that N_k is approximately constant across different k), and j_k is the minimum value of the index j that respects the bound $\hat{t}_{s,k} \le \tau_j$.

Whereas the above is a *recipe* for computing $\tilde{S}_k$, we would also like to have a *model* for $\tilde{S}_k$ that relates it to code and carrier phase errors. We can use such a model to design acquisition and tracking algorithms. In the lecture, we considered the case of a random spreading code $C(\tau)$ and we claimed that under this assumption, $\tilde{S}_k$ can be modeled as

$$\tilde{S}_k[\Delta t_k, \Delta\theta(\tau_j)] = \frac{N_k \bar{A}_k d_i}{2}\bar{R}(\Delta t_k)\left\{\frac{1}{N_k}\sum_{j=j_k}^{j_k+N_k-1}\exp\left[i\Delta\theta(\tau_j)\right]\right\} + n_k.$$

Define all the quantities in this equation that have not been previously defined in this problem. Then, show how this model can be derived from the foregoing recipe for $\tilde{S}_k$ under the assumption of a random spreading code and the previously-defined noise model for $n(j)$. Derive the mean and variance of n_k in terms of σ_n and show that $\mathbb{E}[n_k^* n_l] = 0$ for $k \neq l$.

Hint: The signal component of $\tilde{S}_k$ will boil down to a summation involving three functions of time: $A(\tau_j)$, $R_c[t_s(\tau_j) - \hat{t}_{s,k}] = C[\tau_j - t_s(\tau_j)]C[\tau_j - \hat{t}_{s,k}]$, and $\exp[i\Delta\theta(\tau_j)]$. Assume that $A(\tau_j)$ and $R_c[t_s(\tau_j) - \hat{t}_{s,k}]$ vary slowly enough over the accumulation that they can be modeled as constant in τ_j. Justify the choice of $\bar{A}_k$ and $\bar{R}(\Delta t_k)$ as these constants. Assume that for the modeled $\tilde{S}_k$, we take the expectation of R_c with respect to the underlying code, which we have assumed is perfectly random.

Problem 2: Derive the expression for carrier-to-noise ratio

$$C/N_0 = \frac{A^2}{4\sigma_n^2 T}$$

introduced above under the assumption that the analog-to-digital converter that produces the samples $x(j)$ is preceded by a bandpass anti-aliasing filter with (one-sided) noise-equivalent bandwidth $B_{ne} = 1/2T$. In other words, assume that the bandpass analog signal is sampled at exactly the Nyquist frequency according to the bandpass sampling theorem. Also, assume that the noise component of the analog signal is spectrally flat (white) with two-sided power density $N_0/2$.

Hint: Use the definition of C/N_0 and your results for modeling noise in the conversion from analog-to-digital signals from Homework Assignment #4.

Problem 3: In the lecture, we noted that the summation in the model for $\tilde{S}_k$ is closely related to the discrete-time coherence function

$$C_{\text{coh}}(N) = \left| \frac{1}{N} \sum_{j=0}^{N-1} \exp\left[i\Delta\theta(\tau_j)\right] \right|,$$

where N is the number of samples that participate in a coherent accumulation.

(a) To prevent the signal power from being lost due to variations in $\Delta\theta(\tau_j)$, the value of the coherence function must be maintained near unity. Explain in your own words why this is so. To accompany your explanation, draw a figure that illustrates the concept of signal amplitude degradation due to variations in $\Delta\theta(\tau_j)$.

(b) In the lecture, we defined coherence time as $\tau_{\text{coh}} = TN_{\text{coh}}$, where T is the sampling interval and N_{coh} is the value of N for which the quantity

$$\mathbb{E}\left[C_{\text{coh}}^2(N)\right] = \frac{1}{N^2} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} \mathbb{E}\left[\exp\left\{i\left[\Delta\theta(\tau_j) - \Delta\theta(\tau_k)\right]\right\}\right]$$

drops to some specified value. For our purposes, we will define this value to be 0.5. Here, $\mathbb{E}[\xi]$ denotes the expected value of the random variable ξ. Derive this expression for $\mathbb{E}\left[C_{\text{coh}}^2(N)\right]$ from the foregoing definition of $C_{\text{coh}}(N)$.

Problem 4: Write a function in MATLAB that computes the coherence. Your function should adhere to the following interface

```
function [Ccoh] = computeCoherence(DeltaThetaVec,N)
% computeCoherence : Compute the value of the discrete-time coherence function
%                     Ccoh(N).
%
%
% INPUTS
%
% DeltaThetaVec ----- Ns-by-1 vector representing a sampled carrier phase
%                     error time history, in rad.
%
% N ----------------- The number of samples that will be used to evaluate
%                     the coherence Ccoh(N).
%
%
% OUTPUTS
%
% Ccoh -------------- The value of the discrete-time coherence function for
%                     the first N samples of DeltaThetaVec.
%
%+------------------------------------------------------------------------------+
% References:
%
%
%+==============================================================================+
```

(a) Develop a simulation that employs your function `computeCoherence` to examine the coherence of three common processes that drive variations in the time history $\Delta\theta(\tau_j)$:

White phase noise: $\Delta\theta(\tau_j)$ is a sample from a discrete-time white Gaussian sequence with

$$\mathbb{E}[\Delta\theta(\tau_j)\Delta\theta(\tau_m)] = \sigma_\theta^2 \delta_{j,m}.$$

The units of σ_θ are radians.

White frequency noise (phase random walk): The instantaneous Doppler frequency error $\Delta\omega(\tau_j)$ is a sample from a discrete-time white Gaussian sequence with

$$\mathbb{E}[\Delta\omega(\tau_j)\Delta\omega(\tau_m)] = \sigma_\omega^2 \delta_{j,m}.$$

Here, $\Delta\theta(\tau_j)$ is the integral of the time history $\Delta\omega(\tau_j)$. This process is sometimes called phase random walk because the integrated white frequency noise "walks" around randomly. The units of σ_ω are radians per sampling interval.

White frequency rate noise (frequency random walk): The instantaneous Doppler frequency rate error $\Delta\alpha(\tau_j)$ is a sample from a discrete-time white Gaussian sequence with

$$\mathbb{E}[\Delta\alpha(\tau_j)\Delta\alpha(\tau_m)] = \sigma_\alpha^2 \delta_{j,m}.$$

Here, $\Delta\omega(\tau_j)$ is the integral of the time history $\Delta\alpha(\tau_j)$, and $\Delta\theta(\tau_j)$ is the integral of the time history $\Delta\omega(\tau_j)$. This process is sometimes called frequency random walk because the integrated white frequency rate noise "walks" around randomly. The units of σ_α are radians per sampling interval squared.

(b) Using Monte-Carlo-type simulations, estimate the coherence time $\tau_{\text{coh}} = TN_{\text{coh}}$ for the white frequency noise and white frequency rate noise processes with $\sigma_\omega = 0.01$ radians per sampling interval and $\sigma_\alpha = 0.0001$ radians per sampling interval squared. In other words, generate a large number of realizations of the random process $\Delta\theta(\tau_j)$ using MATLAB's random number generator and then look at the mean of $C^2_{\text{coh}}(N)$ to estimate the coherence time. Assume that the sampling interval $T = 1$ ms. Use simple Euler-type integration to derive $\Delta\theta(\tau_j)$ from $\Delta\omega(\tau_j)$ and $\Delta\alpha(\tau_j)$.

(c) Experiment with the value of $C_{\text{coh}}(N)$ for white phase noise with, say, $\sigma_\theta = 0.8$ radians, and several different values of N. Why doesn't it make sense to estimate the coherence time for the white phase noise process?

Problem 5: In the lecture, we considered the following general acquisition statistic

$$Z = \frac{1}{M}\sum_{l=1}^{N}|S_l|^2 = \frac{1}{M}\sum_{l=1}^{N}\left[\left|\sum_{k=1}^{M}\tilde{S}_k\right|_l^2\right]$$

where M is the number of subaccumulations summed coherently to get accumulations S_l, and N is the number of accumulations S_l summed non-coherently to get Z. Assuming code phase error $\Delta t_k \approx 0$, the normalized subaccumulations $\tilde{S}_k$ can be modeled as

$$\tilde{S}_k = \rho_k \tilde{c}_k + n_k,$$

where the amplitude coefficient $\rho_k = \frac{N_k \bar{A}_k}{2\sigma_{IQ}}$, $\tilde{c}_k$ is a complex-valued scalar given by

$$\tilde{c}_k = \frac{1}{N_k}\sum_{j=j_k}^{j_n+N_k-1}\exp\left[i\Delta\theta(\tau_j)\right],$$

and $n_k = n_{I_k} + jn_{Q_k}$, with

$$n_{I_k} \sim \mathcal{N}(0,1), \quad n_{Q_k} \sim \mathcal{N}(0,1), \quad \mathbb{E}[n_{I_k}n_{I_i}] = \mathbb{E}[n_{Q_k}n_{Q_i}] = \delta_{k,i}, \quad \mathbb{E}[n_{I_k}n_{Q_i}] = 0, \quad \forall k, i.$$

We can state the acquisition hypotheses as follows:

$$\begin{aligned} H_0 &: \ \tilde{S}_k = n_k \\ H_1 &: \ \tilde{S}_k = \rho_k\tilde{c}_k + n_k. \end{aligned}$$

To formulate a Neyman-Pearson hypothesis test, we will need to derive the probability density functions $p(Z|H_0)$ and $p(Z|H_1)$.

(a) Show that $p(Z|H_0) = \chi^2_{2N}$, i.e., show that $p(Z|H_0)$ is a chi-square distribution with $2N$ degrees of freedom. Thus, under H_0, Z has mean $\mathbb{E}[Z|H_0] = 2N$ and variance $\text{var}(Z|H_0) = 4N$. Plot $p(Z|H_0)$ for N raging from 1 to 4 (all plots on the same figure).

(b) Show that $p(Z|H_1) = \chi^2_{2N}(\lambda)$, i.e., show that $p(Z|H_1)$ is a noncentral chi-square distribution with $2N$ degrees of freedom and non-centrality parameter $\lambda = NM\rho^2$, where we assume $\rho^2 = \rho_k^2|\tilde{c}_k|^2$ is constant for all participating k. Thus, under H_1, Z has mean $\mathbb{E}[Z|H_1] = 2N + \lambda$ and variance $\text{var}(Z|H_1) = 4(N + \lambda)$. Plot $p(Z|H_1)$ for N raging from 1 to 4 and M ranging from 1 to 4 (16 total plots). Put all plots for the same M on the same figure. Thus, you will have a total of 4 figures for H_1.

Hint: Let X_i be N independent, normally-distributed random variables with means μ_i and variances σ_i^2. Then, the random variable

$$Z \triangleq \sum_{i=1}^{N} \left(\frac{X_i}{\sigma_i}\right)^2$$

has a noncentral chi-square distribution with N degrees of freedom and non-centrality parameter

$$\lambda = \sum_{i=1}^{N} \left(\frac{\mu_i}{\sigma_i}\right)^2.$$

In the special case that $\mu_i = 0$, $\forall i$, then Z has a chi-square distribution with N degrees of freedom.

Problem 6: Write a MATLAB function to calculate the null hypothesis and alternative hypothesis probability density functions and the decision threshold corresponding to a GNSS signal acquisition problem. Your function should adhere to the following interface:

```
function [pZ_H0,pZ_H1,lambda0,Pd,ZVec] = performAcqHypothesisCalcs(s)
% performAcqHypothesisCalcs --- Calculate the null-hypothesis and alternative
% hypothesis probability density functions and the decision threshold
% corresponding to GNSS signal acquisition with the given inputs.
%
% Z is the acquisition statistic
%
%             N  |  M           |^2
% Z = (1/M)* sum | sum Stilde_k |
%            l=1 | k=1          |
%                |              |l
%
%                 _                _
%             N  | / M    \2   /M    \2|
%   = (1/M)* sum | |sum Ik| + |sum Qk| |
%            l=1 | |k=1   |    |k=1   | |
%                |_\      /    \      /_|l
%
% where Stilde_k = rho_k*exp(j*Delta_phi_k) + n_k
%
% and Ik = rho*cos(Delta_phi_k) + n_Ik,
%     Qk = rho*sin(Delta_phi_k) + n_Qk
%
% and nIk ~ N(0,1) and nQk ~ N(0,1), with E[nIk nQi] = 1 for k = i and 0 for
% k != i. The amplitude rho is related to familiar parameters N, A, and
% sigma_IQ by rho = (N*A)/(2*sigma_IQ), i.e., it is the magnitude of the I,Q
% vector normalized by sigma_IQ.
%
% Under H0, the statistic X is distributed as a chi square distribution with
% 2*N degrees of freedom. Under H1, X is distributed as a noncentral chi
```

```
% square distribution with lambda = N*M*rho^2 and 2*N degrees of freedom.
%
% The total number of cells in the search grid is assumed to be nCells =
% nCodeOffsets*nFreqOffsets, where nFreqOffsets = 2*fMax*Tcoh, with Tcoh the
% total coherent integration time Tcoh = M*Tsub.
%
% INPUTS
%
% s -------- A structure containing the following fields:
%
%         C_N0dBHz ------- Carrier to noise ratio in dB-Hz.
%
%         Tsub ----------- Subaccumulation interval, in seconds.
%
%         M -------------- The number of subaccumulations summed coherently to
%                          get accumulations.
%
%         N -------------- The number of accumulations summed noncoherently to
%                          get Z.
%
%         fMax ----------- Frequency search range delimiter.  The total
%                          frequency search range is +/- fMax.
%
%         nCodeOffsets --- Number of statistically independent code offsets in
%                          the search range.
%
%         PfaAcq --------- The total acquisition false alarm probability.
%                          This is the probability that the statistic Z
%                          exceeds the threshold lambda in any one of the
%                          search cells under the hypothesis H0.  One can
%                          derive the false alarm probability for each search
%                          cell from PfaAcq.
%
%         ZMax ----------- The maximum value of Z that will be considered.
%
%         delZ ----------- The discretization interval used for the
%                          independent variable Z.  The full vector of Z
%                          values considered is thus ZVec = [0:delZ:ZMax].
%
% OUTPUTS
%
% pZ_H0 ---------- The probability density of Z under hypothesis H0.
%
% pZ_H1 ---------- The probability density of Z under hypothesis H1.
%
% lambda0 -------- The detection threshold.
%
% Pd ------------- The probability of detection.
%
% Zvec ----------- The vector of Z values considered.
%
%+------------------------------------------------------------------------------+
% References:
%+==============================================================================+
```

You may use the script `topPerformAcqHypothesisCalcs.m` found on iLearn to test your function. Use your function and the following values for some of the sub-fields of `s`:

```
s.PfaAcq = 0.0001;
s.fMax = 7000;
s.nCodeOffsets = 1023*5;
s.ZMax = 1000;
s.delZ = 0.1;
```

to plot the following two curves:

(a) Set $M = 1$ and plot the N required to achieve a probability of detection $P_D = 0.95$ or more for each of the C/N_0 values in the set $27, 30, 33, 36, 39, 42, 45, 48, 51$.

(b) Set $N = 1$ and plot the M required to achieve a probability of detection $P_D = 0.95$ or more for each of the C/N_0 values in the set $5, 10, 15, ..., 45, 50$.

What conclusions can you draw about the relative performance of coherent vs. non-coherent integration?

Hint: You may find the function `chi2pdf`, `ncx2pdf`, `chi2inv`, and `ncx2cdf` from MATLAB's statistics toolbox useful in implementing `performAcqHypothesisCalcs`.