ASE 389P-7
Problem set 3

Alejandro Moreno

October 28, 2022

1 Problem 1

1.1 Instruction

In lecture we introduced the concept of code-carrier divergence by considering the effect of the dispersive ionosphere on a simplified signal impinging on a GNSS receiver:

$$r(t) = C(t)sin(2\pi f_c t) \tag{1}$$

Where $C(t)$ is a spreading waveform with null-to-null bandwidth B_1 of approximately 2 MHz and f_c is a GNSS L-band carrier frequency (somewhere between 1.2 and 1.6 GHz). We subsequently expressed $r(t)$ as a function of the Fourier transform $\tilde{C}(f) = \mathcal{F}[C(t)]$:

$$r(t) = \int_{-\infty}^{\infty} \frac{\tilde{C}(f)}{2j}[e^{j2\pi(f+f_c)t} - e^{j2\pi(f-f_c)t}]df \tag{2}$$

We see from this equation that $r(t)$ is made up of two monochromatic signals that get weighted by $\tilde{C}(f)/2j$ (a complex number) and integrated over all f. For a particular value of f, the first monochromatic signal looks like

$$\frac{\tilde{C}(f)}{2j}e^{j2\pi(f+f_c)t} \tag{3}$$

If this signal were to pass through the ionosphere, it would experience an excess phase delay given by

$$\Delta\tau_{p,H} = \frac{-K}{(f+f_c)^2} \tag{4}$$

Thus, the received signal corresponding to 3 could be written

$$\frac{\tilde{C}(f)}{2j}e^{j2\pi(f+f_c)(t-\Delta\tau_{p,H})} \tag{5}$$

Following this reasoning, we can rewrite 2 as

$$r(t) = \int_{-\infty}^{\infty} \frac{\tilde{C}(f)}{2j}[e^{j2\pi(f+f_c)(t-\Delta\tau_{p,H})} - e^{j2\pi(f-f_c)(t-\Delta\tau_{p,L})}]df \tag{6}$$

where $\Delta\tau_{p,H}$ is the ionospheric delay (excess phase delay) for the frequency constituent at the "high" frequency $f + f_c$ and

$$\Delta\tau_{p,L} = \frac{-K}{(f - f_c)^2} \tag{7}$$

is the ionospheric delay for the frequency constituent at the "low" frequency $f - f_c$.

With some work it can be shown that 6 reduces to

$$r(t) = C(t - \Delta\tau_g)sin[2\pi f_c(t - \Delta\tau_p)] \tag{8}$$

where $\Delta\tau_g = K/f_c^2$ is the so-called group delay and $\Delta\tau_p = -K/f_c^2$ is the so-called phase delay, with $K = 40.3TEC/c$. Derive this latter expression for $r(t)$ from 6.

Hints: Plug the expressions for $\Delta\tau_{p,L}$ and $\Delta\tau_{p,H}$ into 6. Distribute the $(f+f_c)$ and the $(f-f_c)$. Recognize that the effective range of the integral is small compared to f_c (explain why). This allows you to approximate a term $K/(f_c^2 - f^2)$, which emerges as you simplify, as K/f_c^2 over the range of integration.

1.2 Solution

Plugging the expressions for $\Delta\tau_{p,L}$ and $\Delta\tau_{p,H}$ into 6.

$$r(t) = \int_{-\infty}^{\infty} \frac{\tilde{C}(f)}{2j}[e^{j2\pi(f+f_c)(t-\frac{-K}{(f+f_c)^2})} - e^{j2\pi(f-f_c)(t-\frac{-K}{(f-f_c)^2})}]df$$

Distributing the $(f + f_c)$ and the $(f - f_c)$.

$$\begin{aligned}
r(t) &= \int_{-\infty}^{\infty} \frac{\tilde{C}(f)}{2j}\left[e^{j2\pi[(f+f_c)t+\frac{K}{(f+f_c)}]} - e^{j2\pi[(f-f_c)t+\frac{K}{(f-f_c)}]}\right]df \\
&= \int_{-\infty}^{\infty} \frac{\tilde{C}(f)}{2j}\left[e^{j2\pi[(f+f_c)t+\frac{K}{(f+f_c)}]} - e^{j2\pi[(f-f_c)t+\frac{K}{(f-f_c)}]}\right]df \\
&= \int_{-\infty}^{\infty} \frac{\tilde{C}(f)}{2j}\left[e^{j2\pi[(f+f_c)t+\frac{K}{(f+f_c)}\frac{(f-f_c)}{(f-f_c)}]} - e^{j2\pi[(f-f_c)t+\frac{K}{(f-f_c)}\frac{(f+f_c)}{(f+f_c)}]}\right]df \\
&= \int_{-\infty}^{\infty} \frac{\tilde{C}(f)}{2j}\left[e^{j2\pi[(f+f_c)t+\frac{K(f-f_c)}{(f^2-f_c^2)}]} - e^{j2\pi[(f-f_c)t+\frac{K(f+f_c)}{(f^2-f_c^2)}]}\right]df
\end{aligned}$$

Recognizing that the effective range of the integral is small compared to f_c allows to approximate $K/(f_c^2 - f^2) \approx K/f_c^2$. Note that the carrier frequency is GHz but the bandwidth of the spreading code $\tilde{C}(f)$ is only MHz. Therefore, it is safe to assume the approximation is going to hold strongly.

$$
\begin{aligned}
r(t) &= \int_{-\infty}^{\infty} \frac{\tilde{C}(f)}{2j}\left[e^{j2\pi[(f+f_c)t+\frac{K(f-f_c)}{f_c^2}]} - e^{j2\pi[(f-f_c)t+\frac{K(f+f_c)}{f_c^2}]}\right] df \\
&= \int_{-\infty}^{\infty} \frac{\tilde{C}(f)}{2j}\left[e^{j2\pi[f(t-\frac{K}{f_c^2})+f_c(t+\frac{K}{f_c^2})]} - e^{j2\pi[f(t-\frac{K}{f_c^2})-f_c(t+\frac{K}{f_c^2})]}\right] df \\
&= \int_{-\infty}^{\infty} \frac{\tilde{C}(f)}{2j} e^{j2\pi f(t-\frac{K}{f_c^2})}\left[e^{j2\pi f_c(t+\frac{K}{f_c^2})} - e^{-j2\pi f_c(t+\frac{K}{f_c^2})}\right] df \\
&= C(t-\Delta\tau_g)sin[2\pi f_c(t-\Delta\tau_p)]
\end{aligned}
$$

2 Problem 2

2.1 Instruction

Show that the group velocity vg and the phase index of refraction np are related by $v_g = c\eta_p$ for small group and phase velocity departures from the speed of light c.

2.2 Solution

3 Exercise 3

3.1 Instruction

If a carrier wave meets an ionospheric layer with a refractive index of 0, then the group velocity tends to zero and the phase velocity tends to infinity as the wave is reflected off the layer. What electron density η_e would be required for reflection at the GPS L1 frequency? Research typical electron density values to determine whether there is any chance of such a reflection occurring. What η_e is required for reflection of the Ham Radio band at 50 MHz? What are the chances of reflection at this frequency? For all reflections, assume a normally incident wave (a wave with zenith angle 0).

3.2 Solution

The index of refraction can be, generally, written as

$$\eta = \sqrt{1 - \frac{\eta_e e^2}{4\pi^2 f^2 \epsilon_0 m}} = \sqrt{1 - \frac{40.3\eta_e}{f^2}} \tag{9}$$

where η_e is the electron density, e is the charge magnitude of electron, ϵ_0 is the permitibity of free space and m is the mass of electron. If the right hand side of 9 is close to 1, then it is possible to use the following approximation.

$$\eta = 1 - \frac{40.3\eta_e}{f^2} \tag{10}$$

As it can be observed in Figure 1, the maximum electron density one can get is in the neighborhood of $10^{12}[m^{-3}]$. Thus, the refraction index for a GPS signal can be modeled as 10 since $\eta \approx 1$ at those frequencies.

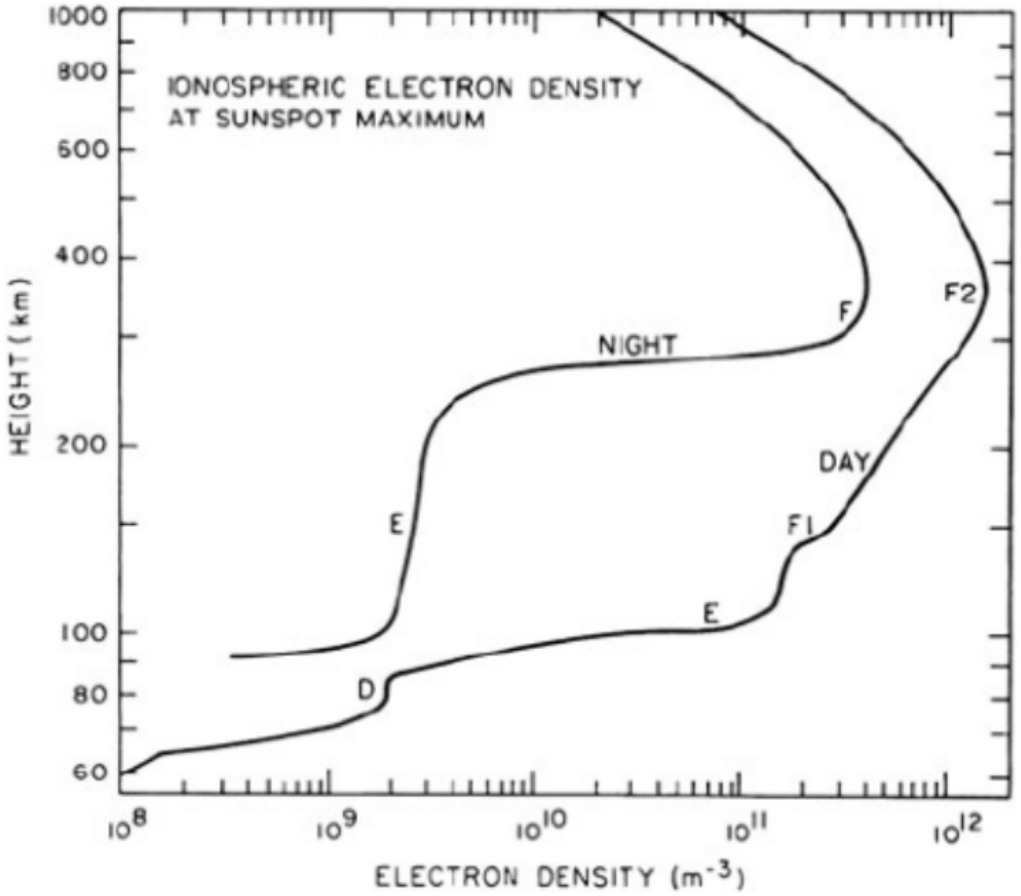

Figure 1: Idealized electron density distribution in the Earth's ionosphere. The curves indicate thedensities to be expected at sunspot maximum in temperate latitudes. Peak sunspot activity occursat 11-year intervals, most recently peaking in 2001 (cycle 23) and 2012 (cycle 24). Cycle 24 wasabout a year late and rather weak [see Janardhan et al. (2015)]. From J. V. Evans and T. Hagfors, Radar Astronomy, 1968. McGraw-Hill Education

The electron density one would require for a signal at $f = 1575.42MHz$ to be reflected is close to $\eta_e = 6 \times 10^{16}[m^{-3}]$.

However, if the signal was at a frequency of $50MHz$, the electron density necessary for it to be reflected is $\eta_e = 6 \times 10^{13}[m^{-3}]$. Note that in this case, the electron density required for the signal to be reflected is much closer to the maximum shown by the graph. Nevertheless, the refraction index for the Ham Radio when the electron density is at its maximum is $\eta \approx 0.98$. So, it is not likely for signals at this frequencies to be reflected.

The calculations show that $f \approx 6MHz$ is the critical frequency at which the ionosphere starts fully reflecting the signals.

4 Problem 4

4.1 Instruction

Download the gzipped archive **gridDataUt6.tar.gz** from http://radionavlab.ae.utexas.edu/datastore/gnssSigProc
Un-zip and un-tar this file to get access to the underlying files. The ***.log** files are columnar-format log files produced by the GRID receiver, a powerful software-defined radionavigation receiver under development in the UT Radionavigation Laboratory. The ***def.txt** files describe the data in the corresponding ***.log** files. For now, you'll only need to pay attention to **channel.log**, whose format is described in **channeldef.txt**. You'll find that in this data set there were four L2C-capable GPS satellites visible over the entire 27-minute data capture interval: those with TXIDs (PRN identifiers) 7, 8, 19, and 28. For this data set, GRID tracked the signal type GPS L1 CA on L1 and GPS L2 CL on L2. Draw the file **channel.log** (or, equivalently, the file channel.mat) into Matlab and operate on the data to generate a plot like the one shown in Figure 5.9 in the Misra and Enge textbook for TXID 7. The plot should include both code- and carrier-derived ionospheric delay time histories. Generate a second plot showing both the code- and carrier-derived total electron content (TEC) seen by the signals from the satellite with TXID 7. Express the value of TEC in TECU. Add a constant offset to the carrier TEC so that it best matches the code TEC in a least-squares sense.

Answer the following questions:

- What could explain the negative values in the final TEC estimates? Is this physically possible?
- The ionospheric delay for TXID 7 changes significantly over the 27-minute data capture interval. How is this possible if the GPS satellites move on slow 12-sidereal-hour orbits? (You may wish to inspect the file **navsol.log** for additional clues.)

Hints:

- Make sure you match pseudorange and carrier phase measurements from the L1 C/A signals with their L2 CL counterparts taken simultaneously.
- You'll want to write a Matlab script to automate the process of drawing in the data and generating the plots for this problem because you'll need to repeat the procedure in a later problem.
- The Matlab files *.mat contain the same data as the corresponding *.log files but in a convenient Matlab format. The following Matlab command sequence (1) draws channel.mat directly into Matlab, (2) stores the contents of the file in the matrix M, (3) finds all the row indices corresponding to the L1 C/A signal of the satellite with TXID 5, (4) loads all the C/N0 measurements from these rows into C_N0Vec_txid05:

```
>> load channel.mat
>> M = channel'; // Transpose to get in columnar format
>> iidum = find(M(:,13) == 0 & M(:,14) == 5);
>> C_N0Vec_txid05 = M(iidum,9);
```

4.2 Solution

4.2.1 Main function

```
clc; close all; clear all;

% Load the data
load("gridDataUt6/channel.mat");
M = channel';

% Calculate the Ionopheric delay and plot it
TXID = 7;
N_L1 = 5609020;
N_L2 = 5609020;
[t, I_L1_code, I_L2_code, I_L1_carrier, I_L2_carrier] = ...
                                dualFreqIonoDelay(M, TXID, N_L1, N_L2);
fig = plotIonoDelay(t, I_L1_code, 'code', NaN);
fig = plotIonoDelay(t, I_L1_carrier, 'carrier', fig);
```

4.2.2 Helper function

```
function [t, I_L1_code, I_L2_code, I_L1_carrier, I_L2_carrier] = ...
                                 dualFreqIonoDelay(M, TXID, N_L1, N_L2)
week2sec = 7 * 24 * 60 * 60;
c = physconst('LightSpeed');
f_L1 = 1575.42 * 1e6;
f_L2 = 1227.60 * 1e6;
lambda_L1 = c / f_L1;
lambda_L2 = c / f_L2;

GPS_L1_CA = 0;
GPS_L2_CL = 2;

% load L1
iidum = find(M(:,13) == GPS_L1_CA & M(:,14) == TXID &...
             M(:,3) ~= 9999 & M(:,10) == 1 & M(:,11) == 0);
ORT_week = M(iidum,3);
ORT_sec_week = M(iidum,4);
ORT_frac_sec = M(iidum,5);
t_L1 = week2sec*ORT_week + ORT_sec_week + ORT_frac_sec;
rho_L1 = M(iidum,8);
phi_L1 = M(iidum,7);

% load L2
iidum = find(M(:,13) == GPS_L2_CL & M(:,14) == TXID  &...
             M(:,3) ~= 9999 & M(:,10) == 1 & M(:,11) == 0);
```

```
ORT_week = M(iidum,3);
ORT_sec_week = M(iidum,4);
ORT_frac_sec = M(iidum,5);
t_L2 = week2sec*ORT_week + ORT_sec_week + ORT_frac_sec;
rho_L2 = M(iidum,8);
phi_L2 = M(iidum,7);

% Preprocess (IDK if the L1 and L2 samples coincide temporarly)
t0 = max(t_L1(1), t_L2(1));
tf = min(t_L1(end), t_L2(end));
t = linspace(t0, tf, 1e5);
rho_L1_intrp = spline(t_L1, rho_L1, t);
phi_L1_intrp = spline(t_L1, phi_L1, t);
rho_L2_intrp = spline(t_L2, rho_L2, t);
phi_L2_intrp = spline(t_L2, phi_L2, t);

% Ionospheric delay [Code]
I_L1_code = f_L2^2 / (f_L1^2 - f_L2^2) * (rho_L2_intrp - rho_L1_intrp);
I_L2_code = f_L1^2 / (f_L2^2 - f_L1^2) * (rho_L1_intrp - rho_L2_intrp);

% Ionospheric delay [Carrier]
I_L1_carrier = f_L2^2 / (f_L1^2 - f_L2^2) * ...
       (lambda_L1*(phi_L1_intrp - N_L1) - lambda_L2*(phi_L2_intrp - N_L2));
I_L2_carrier = f_L1^2 / (f_L2^2 - f_L1^2) * ...
       (lambda_L2*(phi_L2_intrp - N_L2) - lambda_L1*(phi_L1_intrp - N_L1));

end
```

4.2.3 Results

Figure 2 shows the ionospheric delay calculated by dual-frequency GPS.

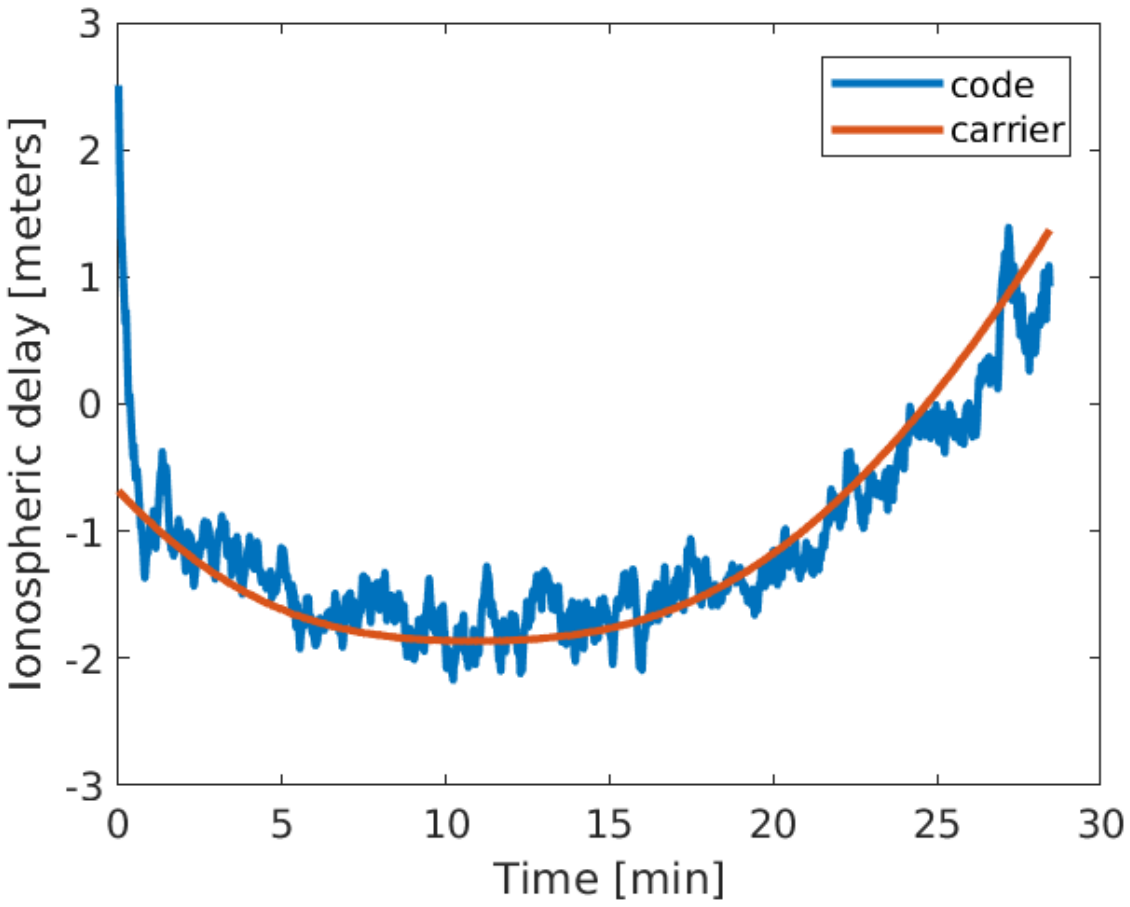

Figure 2: Ionospheric delay measured using code and carrier dual-frequency GPS.

The values shown in Figure 2 are negative. This is not physically possible. So, the must be related to some delay in the electronics of the physical transmitter or receiver.

The rapid change in ionospheric delay could be due to the fact that the data seems to be comming from a receiver in LEO orbit. This can be seen by analyzing the navsol.log file. The height seem to vary from 100 to 600 km and the velocity seems to range from 7500 to 8000 m/s. Knowing that Dr. Humphreys has a receiver in the ISS I wouldn't be surprised if the data would be comming from there. Thus, the rapid movement of the receiver through the ionosphere explains the fast fluctuation of ionospheric delay.

5 Problem 5

5.1 Instruction

Write a function in Matlab for computing the ionospheric delay from a model of the ionosphere. Your function should adhere to the interface described on the next page (which you can copy and paste as comments to your function). Only develop calculations for the broadcast (Klobuchar) model. The function can later be augmented to accommodate other model types. You can learn about the broadcast model on pages 168-169 of the Misra and Enge text. More details can be

found on pages 128-130 of the GPS interface specification (IS) IS-GPS-200F.pdf posted on Canvas. You will also need to write your own function for computing satellite elevation and azimuth angles. Assume the WGS84 model for the shape of the Earth. Note that the GPS IS uses semicircles as its angular measure, which is a strange convention employed back in the 1970s to reduce memory and computation requirement. The sin and cos functions in the IS (e.g., in Fig. 20-4) are meant to operate on semicircles, unless otherwise indicated. Thus, when the IS writes, e.g., $cos(\lambda_i - 1.1617)$, where λ_i is given in semicircles, you can implement this in Matlab as $cos((\lambda_i - 1.1617)\pi)$.

5.2 Solution

```
function [delTauG] = getIonoDelay(ionodata,fc,rRx,rSv,tGPS,model)
% getIonoDelay : Return a model-based estimate of the ionospheric delay
%                experienced by a trans-ionospheric GNSS signal as it
%                propagates from a GNSS SV to the antenna of a terrestrial
%                GNSS receiver.
%
% INPUTS
%
% ionodata ------- Structure containing a parameterization of the
%                  ionosphere that is valid at time tGPS. The structure is
%                  defined differently depending on what ionospheric model
%                  is selected:
%
%                  broadcast --- For the broadcast (Klobuchar) model, ionodata
%                                is a structure containing the following fields:
%
%                       alpha0 ... alpha3 -- power series expansion coefficients
%                                            for amplitude of ionospheric delay
%                       beta0 ... beta3 -- power series expansion coefficients
%                                          for period of ionospheric plasma densit
%                                          cycle
%
%
% Other models TBD ...
%
% fc ------------- Carrier frequency of the GNSS signal, in Hz.
%
% rRx ------------ A 3-by-1 vector representing the receiver antenna position
%                  at the time of receipt of the signal, expressed in meters
%                  in the ECEF reference frame.
%
% rSv ------------ A 3-by-1 vector representing the space vehicle antenna
%                  position at the time of transmission of the signal,
%                  expressed in meters in the ECEF reference frame.
%
```

```
% tGPS ------------ A structure containing the true GPS time of receipt of
%                   the signal. The structure has the following fields:
%                   week -- unambiguous GPS week number
%                   seconds -- seconds (including fractional seconds) of the
%                   GPS week
%
% model ----------- A string identifying the model to be used in the
%                   computation of the ionospheric delay:
%                   broadcast --- The broadcast (Klobuchar) model.
%
% Other models TBD ...
%
% OUTPUTS
%
% delTauG --------- Modeled scalar excess group ionospheric delay experienced
%                   by the transionospheric GNSS signal, in seconds.
%
%+-----------------------------------------------------------------------+
% References: For the broadcast (Klobuchar) model, see IS-GPS-200F
% pp. 128-130.
%
%+=======================================================================+
wgs84 = wgs84Ellipsoid('meter');
[lat,lon,h] = ecef2geodetic(wgs84, rRx(1), rRx(2), rRx(3));
[az,elev,slantRange] = ecef2aer(rSv(1), rSv(2), rSv(3), lat, lon, h, wgs84);
lambda_u = lon/180; % user geodetic longitude (semi-circles)
phi_u    = lat/180; % user geodetic latitude (semi-circles)
A        = az/180;  % azimuth angle between user and satellite, measured clockwise
                    % positive from the true North (semi-circles)
E        = elev/180;% elevation angle between user and satellite (semi_circle)

% earth's central angle between the user position and the earth
% projection of ionospheric intersection point (semi-circles)
Psi = 0.0137/(E+0.11) - 0.022;

% geodetic latitude of the earth projection of the ionospheric
% intersection point (semi-circles)
phi_i = phi_u + Psi * cos(A*pi);
phi_i = max(-0.416, phi_i);
phi_i = min(0.416, phi_i);

% geodetic longitude of the earth projection of the ionospheric
% intersection point (semi-circles)
lambda_i = lambda_u + Psi*cos(A*pi)/cos(phi_i*pi);
```

```
% geomagnetic latitude of the earth projection of the ionospheric
% intersection point (mean ionospheric height assumed 350 km) (semi-circles)
phi_m = phi_i + 0.064*cos((lambda_i - 1.617)*pi);

% local time (sec)
t = 4.32*(10^4)*lambda_i + tGPS.seconds;
while t > 86400
    t = t - 86400;
end
while t < 0
    t = t + 86400;
end

PER = ionodata.broadcast.beta0 * phi_m^0 + ...
      ionodata.broadcast.beta1 * phi_m^1 + ...
      ionodata.broadcast.beta2 * phi_m^2 + ...
      ionodata.broadcast.beta3 * phi_m^3;
PER = max(72000, PER);


AMP = ionodata.broadcast.alpha0 * phi_m^0 + ...
      ionodata.broadcast.alpha1 * phi_m^1 + ...
      ionodata.broadcast.alpha2 * phi_m^2 + ...
      ionodata.broadcast.alpha3 * phi_m^3;
AMP = max(0, AMP);

% phase (radians)
x = 2*pi*(t - 50400)/PER;

% obliquity factor (dimensionless)
F = 1 + 16*(0.53 - E)^3;

% Estimate the ionospheric delay
if abs(x) < 1.57
    T_iono = F*(5e-9 + AMP * (1 - (x^2)/2 + (x^4)/24));
else
    T_iono = F * 5e-9;
end

if fc == 1227.44 * 1e6
    gamma = (77/60)^2;
    T_iono = gamma * T_iono;
end

delTauG = T_iono;
```

end

5.3 Results

The solution I got for the conditions provided was $delTauG = 23.3269ns$, which translates to about 7 meters.

6 Problem 6

6.1 Instruction

Download the gzipped archive gridDataUt1.tar.gz from http://radionavlab.ae.utexas.edu/datastore/gnssSigProcCo
Repeat the steps in problem 4 for this data set. Do this for TXID 29 and for TXID 31. Note that the data capture interval is much shorter, and the ionospheric delay changes much less significantly, than in the original data set for problem 4.

The following broadcast ionospheric parameters are valid for the data in gridDataUt1.tar.gz:

```
alpha0: 4.6566e-009
alpha1: 1.4901e-008
alpha2: -5.9605e-008
alpha3: -5.9605e-008
beta0: 79872
beta1: 65536
beta2: -65536
beta3: -393220
```

Assume a true GPS time of receipt given by

```
week = 1490
seconds = 146238.774036515
```

and a static ECEF receiver antenna location given in meters by

```
X = 1101972.5309609
Y = -4583489.78279095
Z = 4282244.3010423
```

Use the function you wrote for problem 5 to determine the ionospheric delay as calculated by the broadcast model for TXIDs 29 and 31, with the ECEF SV position of TXID 29 at time of transmission given in meters by

```
X = 24597807.6872883
Y = -3065999.1384585
Z = 9611346.77939927
```

and the ECEF SV position of TXID 31 at time of transmission given in meters by

```
X = 2339172.27088689
Y = -16191391.3551878
Z = 21104185.0481546
```

Compare the ionospheric delay as calculated by the broadcast model for TXIDs 29 and 31 with your corresponding plots for these two. How much difference is there between the model-calculated and the empirical ionospheric delays? To what do you attribute this discrepancy?

6.2 Solution

7 Problem 7

7.1 Instruction

In lecture, we noted that the phase shift across an ionospheric blob (irregularity) of size a (in meters) in excess of the phase shift across a layer of average background density of the same size is $\Delta\phi_0 = ar_e\lambda\Delta\eta_e$ where r_e is the classical electron radius in meters, λ is the wavelength of the carrier in meters, and $\Delta\eta_e$ is the excess electron density within the blob as compared to the average background density, in electrons per cubic meter. Derive this expression from expressions for the refractive index η and the excess phase delay $\Delta\tau_p$ given in lecture.

7.2 Solution

The excess delay is defined as follows

$$\Delta\tau_p = \frac{1}{c}\int_{SV}^{RX} [\eta(l) - 1]\, dl \tag{11}$$

It is possible to think of scintillation as some electron density anomaly (blob) in the path of the signal. Therefore, it is possible to write something like

$$\Delta\tau_p = \frac{1}{c}\int_{SV}^{RX} [\eta_{avg}(l) - 1]\, dl + \frac{1}{c}\int_{blob} [\eta_{blob}(l) - 1]\, dl$$

$$\begin{aligned}
\Delta\tilde{\tau}_p &= \frac{1}{c}\int_{blob} [\eta_{blob}(l) - 1]\, dl \\
&= \frac{1}{c}\int_0^a \left[(1 - \frac{r_e\Delta\eta_e\lambda^2}{2\pi}) - 1\right] dl \\
&= -\frac{r_e\Delta\eta_e\lambda^2}{2\pi c}\int_0^a 1 dl \\
&= -\frac{ar_e\Delta\eta_e\lambda^2}{2\pi c}
\end{aligned}$$

Recognizing that $\Delta\tilde{\tau}_p$ is the excess delay in seconds associated with the scintillation effect of the blob. Then it is possible to compute the phase shift as follows. First, $c\Delta\tilde{\tau}_p$ is the delay in meters. Then, $\frac{c\Delta\tilde{\tau}_p}{\lambda}$ is the delay expressed as a fraction of wave length. Finally the phase shift can be obtained through

$$\Delta\phi_0 = \frac{2\pi c \Delta\tilde{\tau}_p}{\lambda} = -a r_e \Delta\eta_e \lambda$$

8 Problem 9

9 Instruction

Download the Cornell Scintillation Simulation Toolkit from http://radionavlab.ae.utexas.edu/datastore/websiteFil
Read the instructions in UserGuide.pdf for generating synthetic scintillation. Experiment with the graphical user interface by typing guiscint at the Matlab prompt. You'll be able to generate a time history of the complex channel response function $z(t)$ by setting the parameters S4 and τ_0 and pressing 'Simulate'. The quantity Te above the bar chart is the mean time between differentially-detected bit errors, which serves as a proxy for the mean time between cycle slips. Notice how Te changes as you experiment with different values of S4, τ_0, and C/N0. Write your own Matlab function to calculate the S4 index and the decorrelation time τ_0 corresponding to a given scintillation time history produced by the scintillation simulator and stored in scintDat.mat. Test your function to see how well your calculated S4 and τ_0 values match the values that were used as parameters in generating the data.

9.1 Solution

```
function [S4,tau0] = computeS4AndTau0(zkhist,tkhist)
% computeS4AndTau0 : Compute the scintillation index S4 and the decorrelation
%                    time tau0 corresponding to the input complex channel
%                    response function time history zkhist.
%
% INPUTS
%
% zkhist ------ Nt-by-1 vector containing the normalized complex scintillation
%               time history in the form of averages over Ts with sampling
%               interval Ts. zkhist(kp1) is the average over tk to tkp1.
%
% tkhist ------ Nt-by-1 vector of time points corresponding to zkhist.
%
%
% OUTPUTS
%
% S4 ---------- Intensity scintillation index of the scintillation time history
%               in zkhist, equal to the mean-normalized standard deviation of
%               the intensity abs(zkhist).^2.
%
% tau0 -------- The decorrelation time of the scintillation time history in
%               zkhist, in seconds.
%
%
```

```
%+------------------------------------------------------------------------------+
% References:
%
%
%+==============================================================================+
alpha = abs(zkhist);
I = alpha.^2;
S4_squared = (mean(I.^2) - mean(I)^2) / mean(I)^2;
S4 = sqrt(S4_squared);

z_bar = mean(zkhist);
xi = zkhist - z_bar;
[R,lags] = xcorr(xi,'normalized');
idx = find(abs(R(find(lags==0):end)) < R(find(lags==0))*exp(-1), 1);
tau0 = tkhist(idx);
```