AAE 575

Introduction to Satellite Navigation and Positioning

Homework 3

November 11, 2011

Josh Wildey

Objective:

The first task at hand is to compute the ionosphere-free measurement calculations of the pseudorange and pseudorange rate using dual frequency measurement techniques. It is then required to estimate the frequency and clock biases using the given data. Then, from these measurements and calculations, determine the stability of the oscillator and use the carrier phase to smooth out the pseudorange measurement. The next task is to implement the Klobuchar model to estimate the Ionosphere delay and to calculate the corrected model and compare this model to the pseudorange measurements. Lastly, given another set of data, the tropospheric delay needs to be calculated by implementing the Sasstamoinen and Hopfield Models. These models are then to be used for comparing the mapping function between flat-earth, Misra and Enge, and Chao Mapping function coefficients.

Given Data:

files: rawdata.sv13, orbit.sv13, dualfreq.sv1, DH010600.MET

ECEF Location of CORS Station:

$X = -1288337.0539$ m

$Y = -4721990.4483$ m

$Z = 4078321.6617$ m

α_0 0.028×10^{-6}	α_1 -0.007×10^{-6}	α_2 -0.119×10^{-6}	α_3 0.199×10^{-6}
β_0 137.0×10^{3}	β_1 -49.0×10^{3}	β_2 -131.0×10^{3}	β_3 -262.0×10^{3}

ECEF Location of Durmid Hill GPS Station:

$\Phi = 33.390°$ N

$\lambda = 115.79°$ W

$h = 0$ m

Problem 1:

Part 1:

To calculate the ionosphere free pseudoranges the following equations were used:

$$\rho_{L1} = \rho^* + \frac{A}{f_{L1}^2}$$

$$\rho_{L2} = \rho^* + \frac{A}{f_{L2}^2}$$

Solving for A leads to a value of:

$$A = \frac{f_{L1}^2 f_{L2}^2}{f_{L1}^2 - f_{L2}^2}(\rho_{L2} - \rho_{L1})$$

Plugging this back into one of the equations and solving to the pseudorange, ρ^*, yields:

$$\rho^* = \frac{f_{L1}^2}{f_{L1}^2 - f_{L2}^2}\rho_{L1} - \frac{f_{L2}^2}{f_{L1}^2 - f_{L2}^2}\rho_{L2}$$

Similarly for the ionosphere free carrier phase scaled to the L1 frequency yields:

$$\varphi^* = \frac{f_{L1}^2}{f_{L1}^2 - f_{L2}^2}\varphi_{L1} - \frac{f_{L1} f_{L2}}{f_{L1}^2 - f_{L2}^2}\varphi_{L2}$$

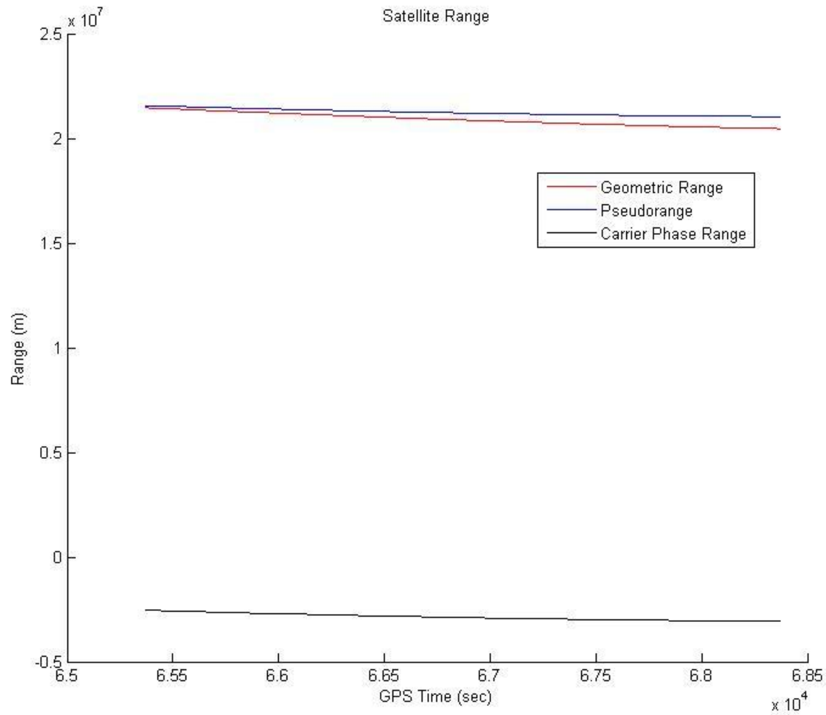


Figure 1- 1 Satellite Range

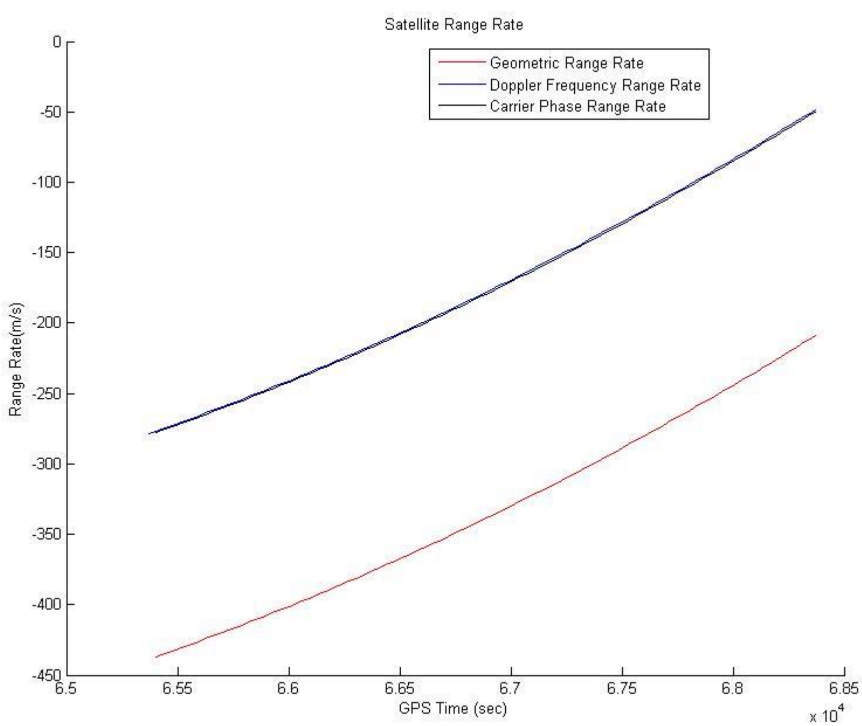


Figure 1-2 Satellite Range Rate

Part 2:

To determine the oscillator frequency bias, the following equation was used:

$$f_{b,S} = -f_D - \frac{\dot{r}}{\lambda}$$

To calculate the Line of Sight velocity the Frequency bias was multiplied by the wavelength of the frequency. Results can be seen in figures 1-3 and 1-4.

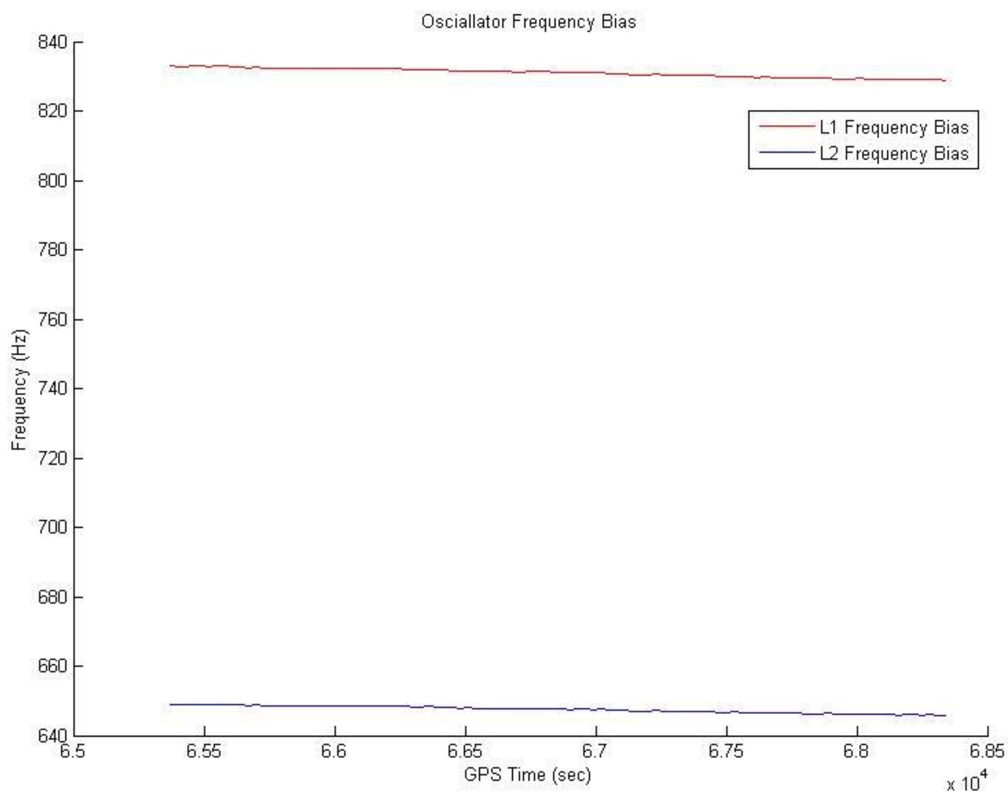


Figure 1-3 Oscillator Frequency Bias

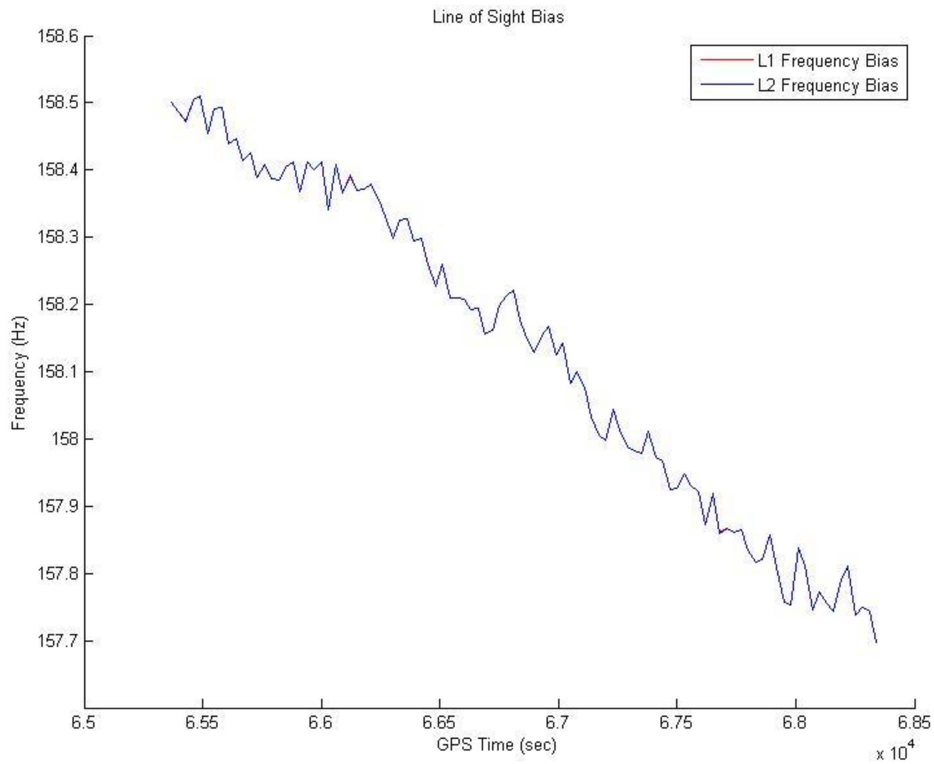


Figure 1-4 Line of Sight Bias

Problem 2 and 3:

To implement the Klobuchar model the following equations were used and were referenced from the IS-GPS-200E manual.

The ionospheric correction model is given by

$$T_{iono} = \begin{cases} F * \left[5.0 * 10^{-9} + (AMP)\left(1 - \frac{x^2}{2} + \frac{x^4}{24}\right)\right], & |x| < 1.57 \\ F * \left(5.0 * 10^{-9}\right) & , |x| \geq 1.57 \end{cases} \quad \text{(sec)}$$

where

T_{iono} is referred to the L1 frequency; if the user is operating on the L2 frequency, the correction term must be multiplied by γ (reference paragraph 20.3.3.3.3.2),

$$AMP = \begin{cases} \sum_{n=0}^{3} \alpha_n \phi_m^n, & AMP \geq 0 \\ \text{if } AMP < 0, & AMP = 0 \end{cases} \quad \text{(sec)}$$

$$x = \frac{2\pi\ (t - 50400)}{PER} \qquad \text{(radians)}$$

$$PER = \begin{cases} \sum_{n=0}^{3} \beta_n \phi_m^n, & PER \geq 72{,}000 \\ \text{if } PER < 72{,}000, & PER = 72{,}000 \end{cases} \quad \text{(sec)}$$

$$F = 1.0 + 16.0\,[0.53 - E]^3$$

and α_n and β_n are the satellite transmitted data words with n = 0, 1, 2, and 3.

Figure 20-4. Ionospheric Model (Sheet 1 of 3)

Other equations that must be solved are

$$\phi_m = \phi_i + 0.064\cos(\lambda_i - 1.617) \qquad \text{(semi-circles)}$$

$$\lambda_i = \lambda_u + \frac{\psi \sin A}{\cos \phi_i} \qquad \text{(semi-circles)}$$

$$\phi_i = \left\{ \begin{array}{ll} \phi_u + \psi \cos A, & |\phi_i| \le 0.416 \\ \text{if } \phi_i > +0.416, \text{ then } & \phi_i = +0.416 \\ \text{if } \phi_i < -0.416, \text{ then } & \phi_i = -0.416 \end{array} \right\} \qquad \text{(semi-circles)}$$

$$\psi = \frac{0.0137}{E + 0.11} - 0.022 \qquad \text{(semi-circles)}$$

$$t = 4.32\,(10^4)\,\lambda_i + \text{GPS time} \qquad \text{(sec)}$$

where

$0 \le t < 86400$: therefore, if $t \ge 86400$ seconds, subtract 86400 seconds;

if $t < 0$ seconds, add 86400 seconds.

Figure 20-4. Ionospheric Model (Sheet 2 of 3)

The following figure show the results.

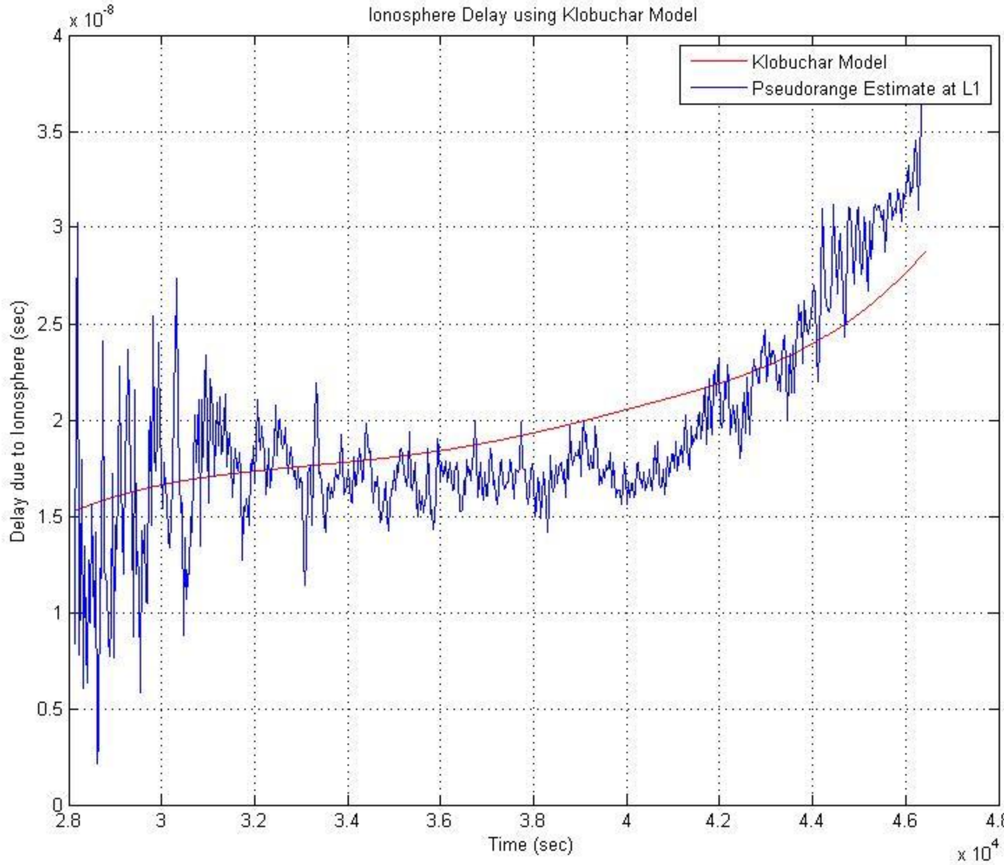


Figure 1- -5 Klobuchar Model and Pseudorange estimation

The pseudorange was estimated using the following equations:

$$A = \frac{f_{L1}^2 f_{L2}^2}{f_{L1}^2 - f_{L2}^2}(\rho_{L1} - \rho_{L2})$$

$$\tau_{Ionosphere\ Delay} = \frac{A}{c f_{L1}^2}$$

Although the pseudorange estimation is very noisy, the Klobuchar model can still be seen to model the pseudorange with the same general curvature. It is not perfect, but represents a decent model.

Problem 4:

To implement the Sasstamoinen model the following equations were used:

$$e_0 = 6.108 * RH * exp\left(\frac{17.15T - 4684}{T - 38.5}\right)$$

$$c\tau_{T,Z,Dry} = .0022777(1 + .0026\cos 2\phi + .0008H)P_0$$

$$c\tau_{T,Z,Wet} = .0022777\left(\frac{1255}{T_0} + .05\right)e_0$$

To implement the Hopfield Model the following equations were used:

$$h_d = 40136 + 148.72(T - 273.16)$$

$$h_w = 11000$$

$$c\tau_{T,Z,Dry} = 77.64 \times 10^{-6}\left(\frac{P_0 h_d}{5T}\right)$$

$$c\tau_{T,Z,Wet} = 0.373\left(\frac{e_0 h_w}{5T^2}\right)$$

The results are as follows:

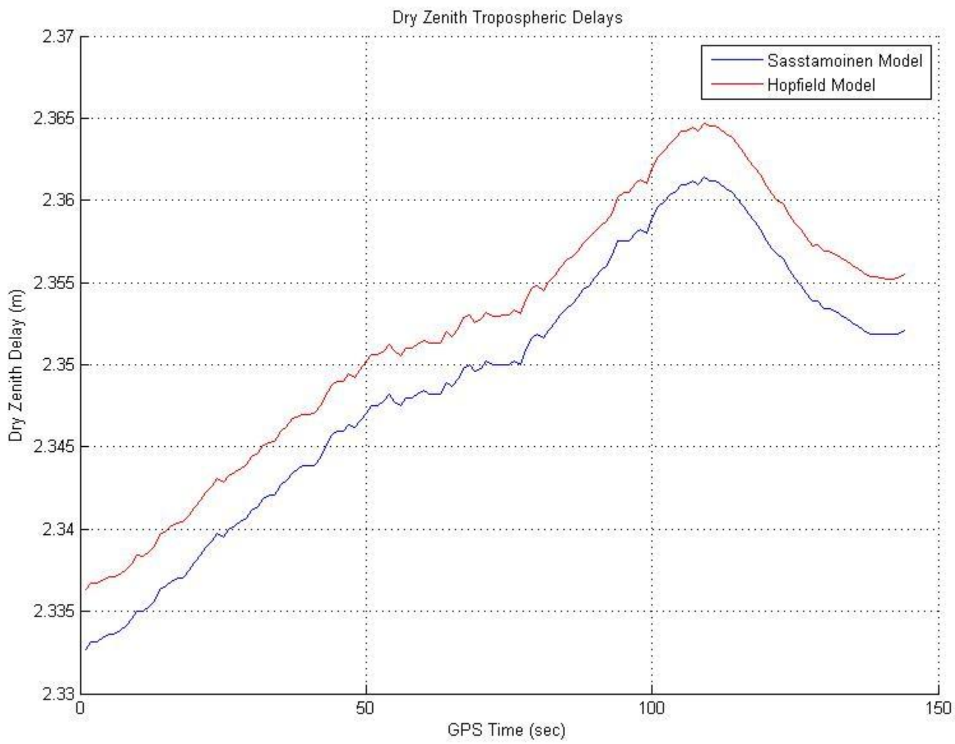


Figure 1-Error! No text of specified style in document.**6 Dry Zenith Delays**

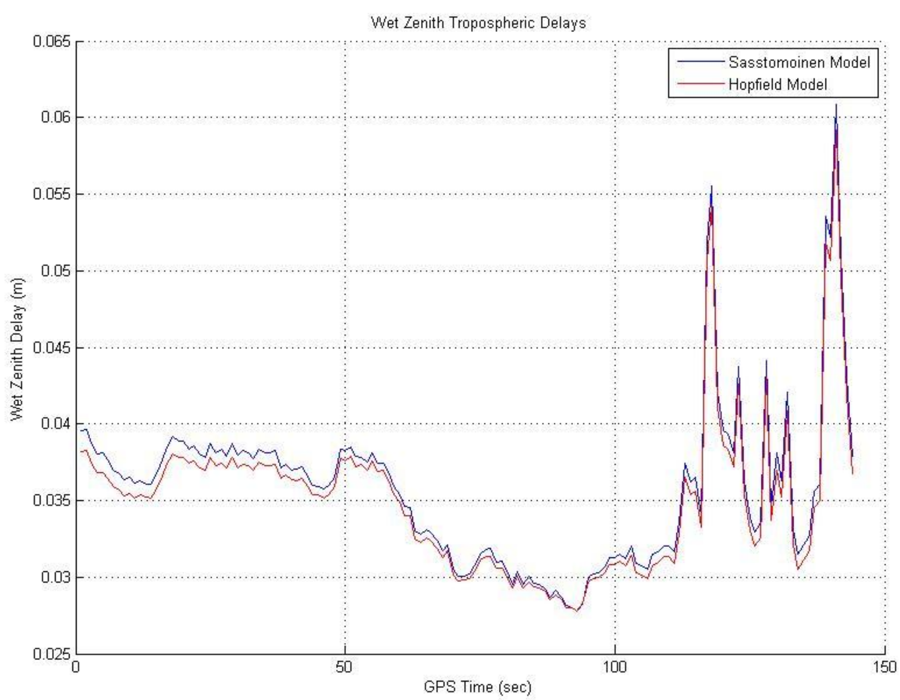


Figure 1-7 Wet Zenith Delays

Both models of tropospheric delays are very similar with only minor offsets in their values and could be used interchangeably for correct results.

Problem 5:

To compare the three different mapping functions the following coefficients and equations were used:

For simple flat-earth Mapping:

$$m(EL) = \frac{1}{\sin(EL)}$$

For Misra and Enge Mapping:

$$m(EL) = \frac{1}{\sqrt{1 - \left(\frac{\cos(EL)}{1.001}\right)^2}}$$

For Chao Mapping:

$$m_{Dry}(EL) = \frac{1}{\sin(EL) + \frac{0.00143}{\tan(EL) + 0.0445}}$$

$$m_{Wet}(EL) = \frac{1}{\sin(EL) + \frac{0.00035}{\tan(EL) + 0.017}}$$

The total tropospheric delay can be represented by the following equation:

$$\tau_{Total\ Tropospheric\ Delay} = \tau_{T,Z,Wet} m_{Wet}(EL) + \tau_{T,Z,Dry} m_{Dry}(EL)$$

The total tropospheric delay was calculated for elevation angles of 5°, 15°, 45°, and 85° and can be seen in the following figures.

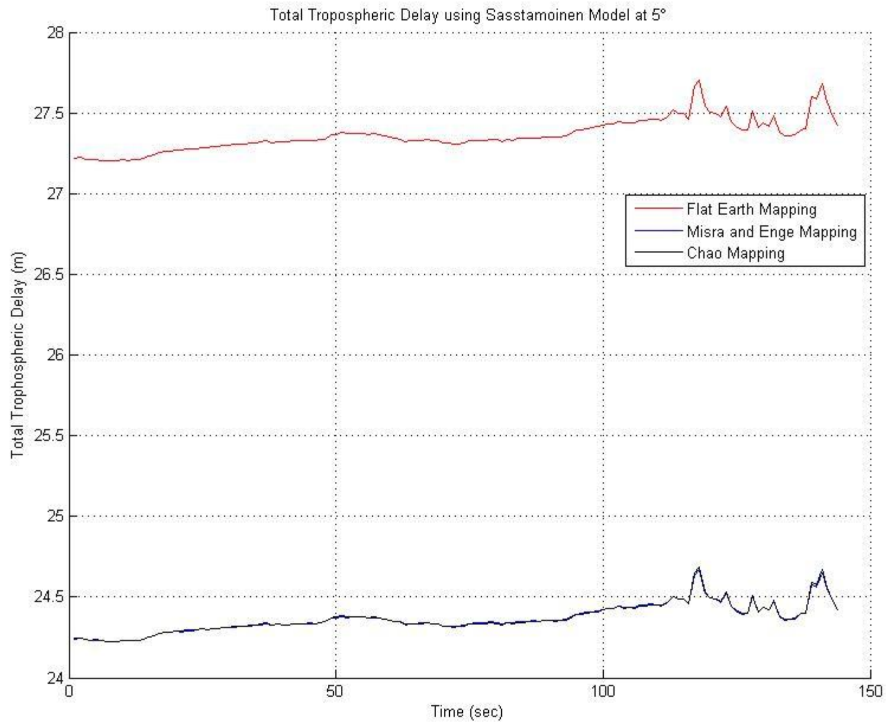


Figure 1-8

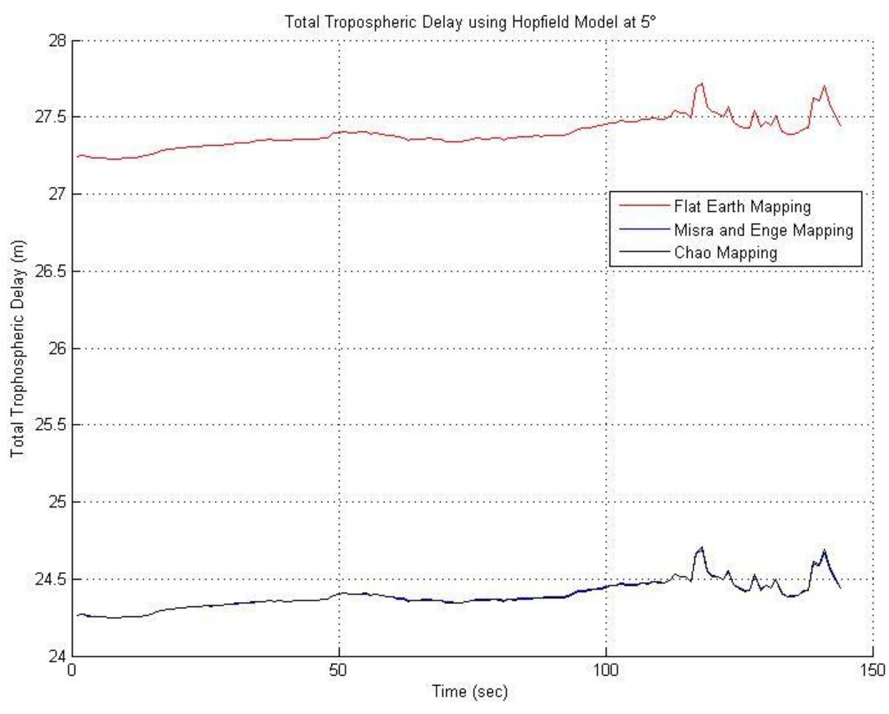
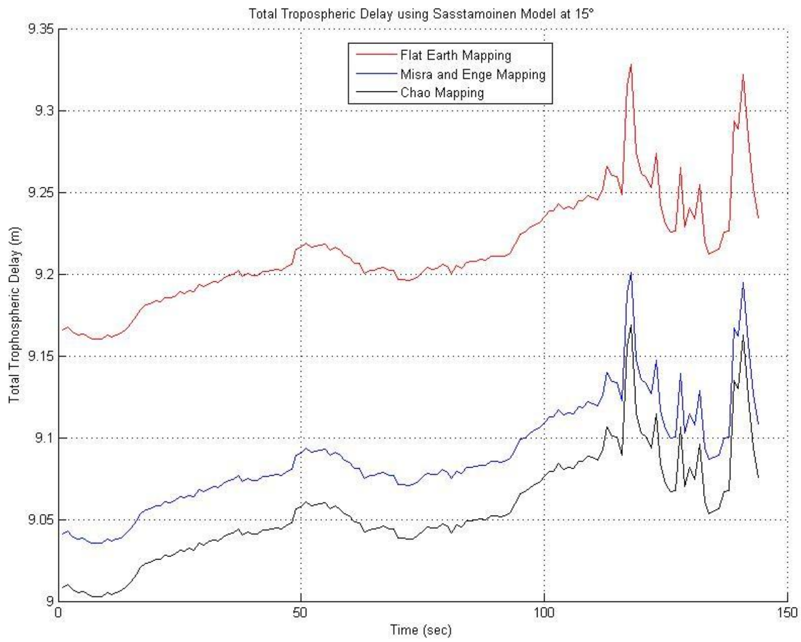


Figure 1-9


Total Tropospheric Delay using Sasstamoinen Model at 15°
Flat Earth Mapping
Misra and Enge Mapping
Chao Mapping
Total Trophospheric Delay (m)
Time (sec)
9.35
9.3
9.25
9.2
9.15
9.1
9.05
9
0
50
100
150

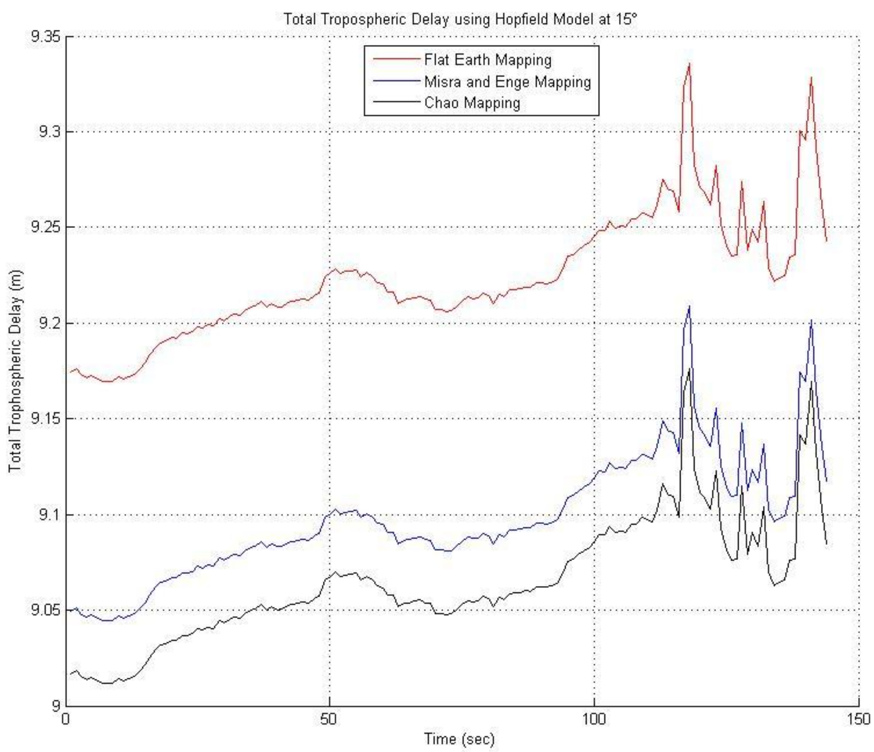

Total Tropospheric Delay using Hopfield Model at 15°
Flat Earth Mapping
Misra and Enge Mapping
Chao Mapping
Total Trophospheric Delay (m)
Time (sec)
9.35
9.3
9.25
9.2
9.15
9.1
9.05
9
0
50
100
150

Figure 1-10

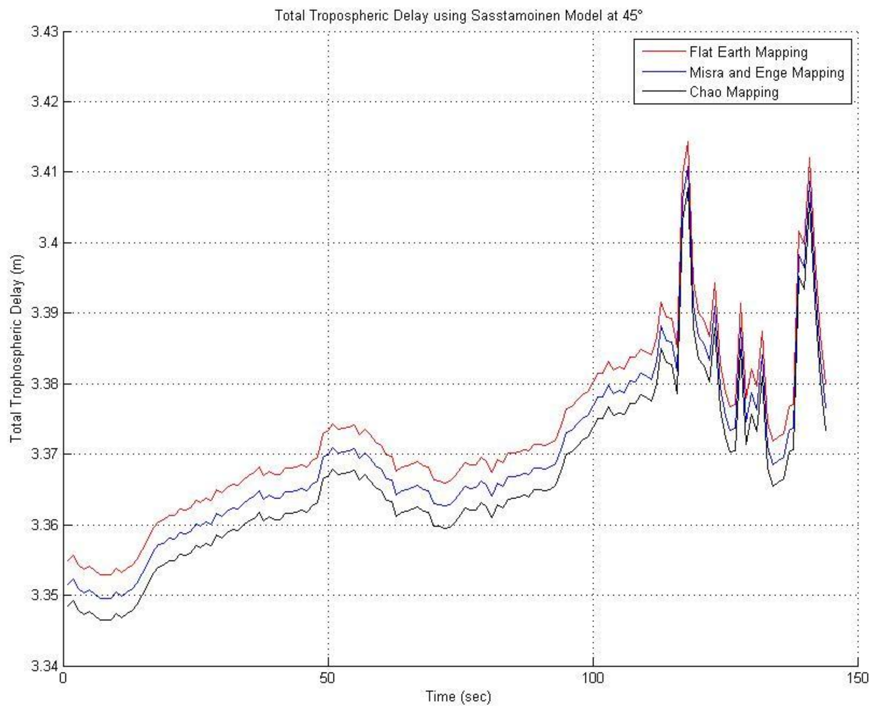


Figure 1-11

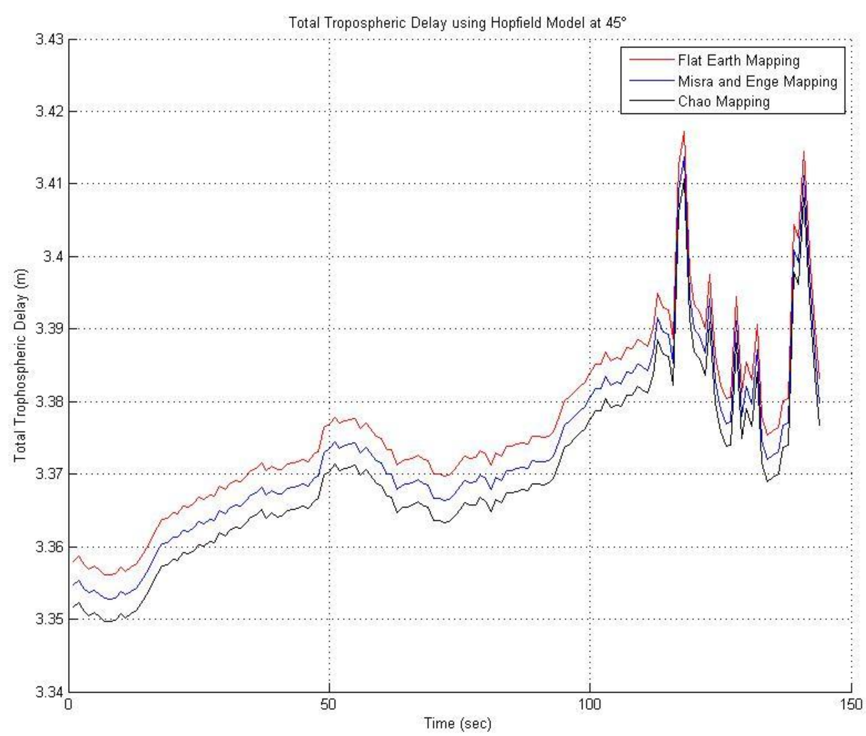


Figure 1-12

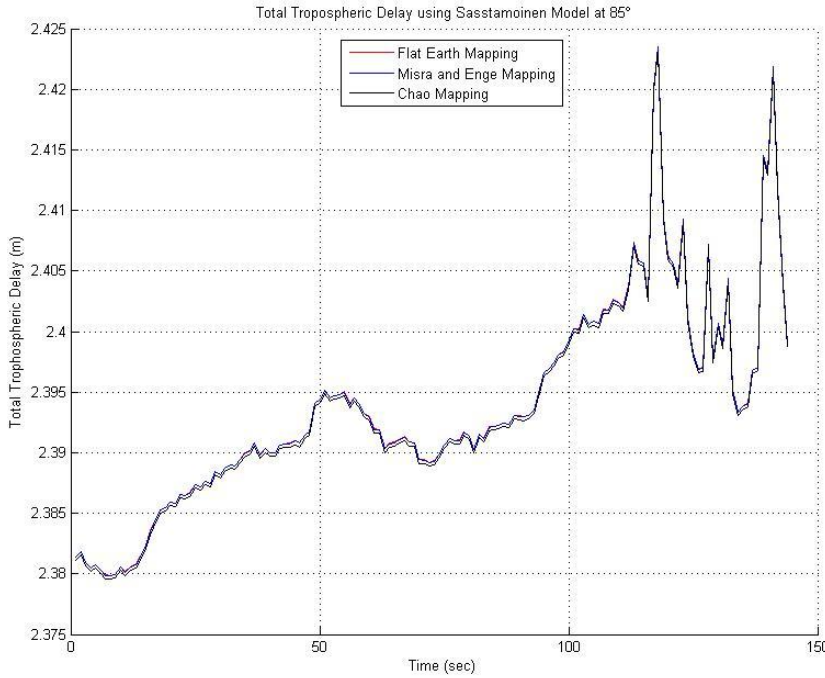


Figure 1-13

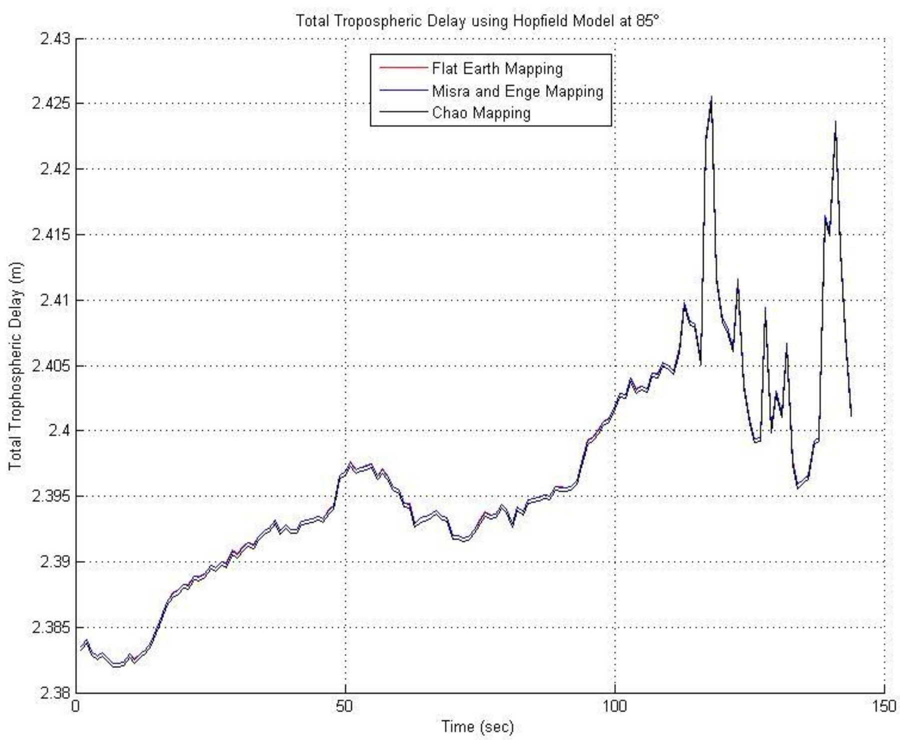


Figure 1-14

As shown in these figures, as the elevation angle gets closer and closer to 90° the different models become almost identical. At low elevation angles there are high error rates, especially with the flat earth mapping function as it is known to produce bad results below 15°. It is also observed that the actual delay decreases significantly as the elevation angle increases, again becoming more accurate and precise as the elevation angle increases. In the range between 5° and 85°, it is seen that the models still hold very similar shape and are just offset from each other.

Appendix A: Code

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Author: Josh Wildey
%  Class: AAE57500
%  Homework 3 - 11/4/2011
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all
clear all
clc

%%%%%%%%% Problem 1: %%%%%%%%%
% Load data and define Global Variables
load orbit0.sv13                %Load Data Set
load rawdata.sv13               %Load Data Set
load dualfreq.sv1               %Load Data set
load DH010600.MET               %Load Data set

xr = -1288337.0539;             %x-position of base station (m)
yr = -4721990.4483;             %y-position of base station (m)
zr = 4078321.6617;              %z-position of base station (m)

c = 299792458;                  %Speed of light (m/s)
fL1 = 1575.42e6;                %Frequency of L1 (Hz)
fL2 = 1227.60e6;                %Frequency of L2 (Hz)
lambdaL1 = c/fL1;               %Wavelength of L1 (m)
lambdaL2 = c/fL2;               %Wavelength of L2 (m)

GPStime = rawdata(:,1);         %SV 13 GPS time from given data set (s)
L1_Prange = rawdata(:,2);       %L1 Pseudorange from given data set (m)
L2_Prange = rawdata(:,3);       %L2 Pseudorange from given data set (m)
L1_Cphase = rawdata(:,4);       %L1 Carrier Phase from given data set (cycles)
L2_Cphase = rawdata(:,5);       %L2 Carrier Phase from given data set (cycles)
L1_DopFreq = rawdata(:,6);      %L1 Doppler Freqeuncy from data set (Hz)
L2_DopFreq = rawdata(:,7);      %L2 Doppler Frequency from data set (Hz)

GPStime_rx = orbit0(:,1);       %GPS time of signal reception (s)

% Calculate True Geometric Range
geo_range = sqrt((xr-orbit0(:,2)).^2+(yr-orbit0(:,3)).^2+(zr-
orbit0(:,4)).^2);

% Calculate True Geometric Range Rate
geo_range_rate = diff(geo_range)./diff(GPStime);

% Calculate Ionosphere Free Pseudorange
ionfree_Prange = (fL1^2/(fL1^2-fL2^2))*L1_Prange - (fL2^2/(fL1^2-
fL2^2))*L2_Prange;

% Calculate Phi*
phistar_L1 = (fL1^2/(fL1^2-fL2^2))*L1_Cphase-(fL1*fL2/(fL1^2-
fL2^2))*L2_Cphase;

% Calculate Carrier Phase Pseudorange
```

```
Cphase_range_L1 = lambdaL1*phistar_L1;

% Calculate Range Rate from Doppler Frequency
DopFreq_RangeRate = -lambdaL1*L1_DopFreq;

% Calculate Range Rate from Phi*
phistar_L1_RangeRate = lambdaL1*(diff(phistar_L1)./diff(GPStime));

figure(1)
hold on
plot(GPStime,geo_range,'r')
plot(GPStime,ionfree_Prange,'b')
plot(GPStime,Cphase_range_L1,'k')
title('Satellite Range'),xlabel('GPS Time (sec)'),ylabel('Range (m)')
legend('Geometric Range', 'Pseudorange', 'Carrier Phase Range')
hold off

figure(2)
hold on
plot(GPStime(2:length(GPStime)),geo_range_rate,'r')
plot(GPStime,DopFreq_RangeRate,'b')
plot(GPStime(2:length(GPStime)),phistar_L1_RangeRate,'k')
title('Satellite Range Rate'),xlabel('GPS Time (sec)'),ylabel('Range
Rate(m/s)')
legend('Geometric Range Rate', 'Doppler Frequency Range Rate', 'Carrier Phase
Range Rate')
hold off

% Calculate Frequency Bias (Part 2)
fL1_bias = -L1_DopFreq(1:length(geo_range_rate)) - geo_range_rate/lambdaL1;
fL2_bias = -L2_DopFreq(1:length(geo_range_rate)) - geo_range_rate/lambdaL2;

% Calculate Line of Sight Velocity (Part 2)
LoS_fL1 = lambdaL1*fL1_bias;
LoS_fL2 = lambdaL2*fL2_bias;

figure(3)
hold on
plot(GPStime(1:length(fL1_bias)),fL1_bias,'r')
plot(GPStime(1:length(fL2_bias)),fL2_bias,'b')
title('Osciallator Frequency Bias'),xlabel('GPS Time
(sec)'),ylabel('Frequency (Hz)')
legend('L1 Frequency Bias','L2 Frequency Bias')
hold off

figure(4)
hold on
plot(GPStime(1:length(LoS_fL1)),LoS_fL1,'r')
plot(GPStime(1:length(LoS_fL2)),LoS_fL2,'b')
title('Line of Sight Bias'),xlabel('GPS Time (sec)'),ylabel('Frequency (Hz)')
legend('L1 Frequency Bias','L2 Frequency Bias')
hold off

%%%%%%%%% Problem 2: %%%%%%%%%
```

```
El = dualfreq(:,3)/180;     %Elevation angle in units of semi-circles
Az = dualfreq(:,2)/180;     %Azimuth angle in units semi-circles

alpha = [0.028e-6,-0.007e-6,-0.119e-6,0.119e-6];
beta = [137e3,-49e3,-131e3,-262e3];

% Calculate Obliquity factor (dimensionless)
F = 1.0 + 16.0*(.53-El).^3;

% Calculate Earth's central angel between user position and earth
% projection of ionospheric intersection point (semi-circles)
psi = .0137./(El+.11) - .022;

% Convert Base station Earth Center Earth Fixed coordinates to latitude,
% longitude, altitude
lla = ecef2lla([xr yr zr]);
latR = lla(1)/180;  %Latitude of Rx in SC units
longR = lla(2)/180; %Longitude of Rx in SC units
altR = lla(3);      %Altitude of Rx in m

% Calculate Geodetic latitude of the earth projection of the ionospheric
% instersection point with anomoly correction based on IS-GPS-200E Spec
phii = latR + psi.*cosd(Az);
for q = 1:length(phii)
    if phii(q) > .416
        phii(q) = .416;
    elseif phii(q) < -.416
        phii(q) = -.416;
    else
        phii(q) = phii(q);
    end
end

% Calculate Geodetic longitude of earth projection of the ionospheric
% intersection point
lambdai = longR + psi.*sind(Az)./cosd(phii);

% Calculate Geomagnetic latitude of earth projection of ionospheric
% intersection
phim = phii + .064.*cosd(lambdai-1.617);

% Calculate local time with autocorrection based on IS-GPS-200E Spec
t = 4.32e4.*lambdai + dualfreq(:,1);
for q = 1:length(t)
    if t(q) >= 86400
        t(q) = t(q) - 86400;
    elseif t(q) < 0
        t(q) = t(q) + 86400;
    end
end

% Calculate A4 with autocorrection based on IS-GPS-200E Spec
PER = beta(1).*(phim.^0) + beta(2).*(phim.^1) + beta(3).*(phim.^2) +
beta(4).*(phim.^3);
for q = 1:length(PER)
```

```
    if PER(q) < 72000
        PER(q) = 72000;
    end
end

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

% Calculate A2 with autocorrection based on IS-GPS-200E Spec
AMP = alpha(1).*(phim.^0) + alpha(2).*(phim.^1) + alpha(3).*(phim.^2) +
alpha(4).*(phim.^3);
for q = 1:length(AMP)
    if AMP(q) < 0
        AMP(q) = 0;
    end
end

%Ionospheric correction model based on IS-GPS-200E Spec
Tiono = zeros(size(x));
for q = 1:length(t)
    if abs(x(q)) >= 1.57
        Tiono(q) = F(q)*5e-9;
    elseif abs(x(q)) < 1.57
        Tiono(q) = F(q)*(5e-9+AMP(q)*(1 - x(q)^2/2 + x(q)^4/24));
    end
end

figure(5)
plot(t,Tiono,'r')
title('Ionosphere Delay using Klobuchar Model'),xlabel('Time (sec)'),
ylabel('Delay due to Ionosphere (sec)')
grid on
hold on

%%%%%%%%% Problem 3: %%%%%%%%%

% Compute the Value of A
A = (fL1^2*fL2^2/(fL2^2-fL1^2)).*(dualfreq(:,4)-dualfreq(:,5));

% Compute Ionospheric Delay
tau_iono = A/(c*fL1^2);

plot(t,tau_iono)
legend('Klobuchar Model','Pseudorange Estimate at L1')
hold off

%%%%%%%%% Problem 4: %%%%%%%%%
%Local Variable Definitions
year = DH010600(:,1);
month = DH010600(:,2);
day = DH010600(:,3);
hour = DH010600(:,4);
minute = DH010600(:,5);
sec = DH010600(:,6);
pressure = DH010600(:,7);       % Surface Atmospheric Pressure (mbar)
```

```
temp = DH010600(:,8) + 273.15;  % Surface Atmospheric Temperature (deg K)
humidity = DH010600(:,9)/100;   % Relative Humidity

phi = 33.39;                    % Degrees N (Latitude)
lambda = 115.79;                % Degrees W (Longitude)
h = 0;                     % Altitude (meters)

% Calculate Water Vapor Partial Pressure
e0 = 6.108*humidity.*exp((17.15*temp - 4684)./(temp - 38.5));

% Calculate Sasstamoinen Zenith Delays
Sasst_dry = .0022777*(1 + .0026*cosd(2*phi) + .00028*h).*pressure;
Sasst_wet = .0022777*(1255./temp + .05).*e0;

% Calculate Heights for Hopfield Model
hd = 40136 + 148.72.*(temp - 273.16);
hw = 11000;

% Calculate Hopfield Model
hop_dry = 77.64e-6.*pressure.*hd./(5*temp);
hop_wet = .373*e0*hw./(5*temp.^2);

figure(6)
hold on, grid on
plot(Sasst_dry,'b')
plot(hop_dry,'r')
title('Dry Zenith Tropospheric Delays'),xlabel('GPS Time (sec)'),ylabel('Dry
Zenith Delay (m)')
legend('Sasstamoinen Model','Hopfield Model')

figure (7)
hold on, grid on
plot(Sasst_wet,'b')
plot(hop_wet,'r')
title('Wet Zenith Tropospheric Delays'),xlabel('GPS Time (sec)'),ylabel('Wet
Zenith Delay (m)')
legend('Sasstomoinen Model','Hopfield Model')

%%%%%%%%% Problem 5: %%%%%%%%%
% Local Variable Definitions
elev = [5 15 45 85];

% Calculate Flat-Earth Mapping
m = 1./sind(elev);

flat_delay_model_sasst = zeros(length(Sasst_dry),length(elev));
flat_delay_model_hop = zeros(length(Sasst_dry),length(elev));
for q = 1:length(elev)
    flat_delay_model_sasst(:,q) = Sasst_wet*m(q) + Sasst_dry*m(q);
    flat_delay_model_hop(:,q) = hop_wet*m(q) + hop_dry*m(q);
end

% Calculate Misra and Enge Mapping
mME = 1./(sqrt(1-(cosd(elev)./1.001).^2));
```

```
misra_delay_model_sasst = zeros(length(Sasst_dry),length(elev));
misra_delay_model_hop = zeros(length(Sasst_dry),length(elev));
for q = 1:length(elev)
    misra_delay_model_sasst(:,q) = Sasst_wet*mME(q) + Sasst_dry*mME(q);
    misra_delay_model_hop(:,q) = hop_wet*mME(q) + hop_dry*mME(q);
end

% Calculate Chao Mapping
mchao_dry = 1./(sind(elev)+.00143./(tand(elev)+.0445));
mchao_wet = 1./(sind(elev)+.00035./(tand(elev)+.017));

chao_delay_model_sasst = zeros(length(Sasst_dry),length(elev));
chao_delay_model_hop = zeros(length(Sasst_dry),length(elev));
for q = 1:length(elev)
    chao_delay_model_sasst(:,q) = Sasst_wet*mchao_wet(q) +
Sasst_dry*mchao_dry(q);
    chao_delay_model_hop(:,q) = hop_wet*mchao_wet(q) + hop_dry*mchao_dry(q);
end

figure(8)
hold on, grid on
plot(flat_delay_model_sasst(:,1),'r')
plot(misra_delay_model_sasst(:,1),'b')
plot(chao_delay_model_sasst(:,1),'k')
title('Total Tropospheric Delay using Sasstamoinen Model at
5\circ'),xlabel('Time (sec)'),ylabel('Total Trophospheric Delay (m)')
legend('Flat Earth Mapping','Misra and Enge Mapping','Chao Mapping')

figure(9)
hold on, grid on
plot(flat_delay_model_hop(:,1),'r')
plot(misra_delay_model_hop(:,1),'b')
plot(chao_delay_model_hop(:,1),'k')
title('Total Tropospheric Delay using Hopfield Model at 5\circ'),xlabel('Time
(sec)'),ylabel('Total Trophospheric Delay (m)')
legend('Flat Earth Mapping','Misra and Enge Mapping','Chao Mapping')

figure(10)
hold on, grid on
plot(flat_delay_model_sasst(:,2),'r')
plot(misra_delay_model_sasst(:,2),'b')
plot(chao_delay_model_sasst(:,2),'k')
title('Total Tropospheric Delay using Sasstamoinen Model at
15\circ'),xlabel('Time (sec)'),ylabel('Total Trophospheric Delay (m)')
legend('Flat Earth Mapping','Misra and Enge Mapping','Chao Mapping')

figure(11)
hold on, grid on
plot(flat_delay_model_hop(:,2),'r')
plot(misra_delay_model_hop(:,2),'b')
plot(chao_delay_model_hop(:,2),'k')
title('Total Tropospheric Delay using Hopfield Model at
15\circ'),xlabel('Time (sec)'),ylabel('Total Trophospheric Delay (m)')
legend('Flat Earth Mapping','Misra and Enge Mapping','Chao Mapping')
```

```
figure(12)
hold on, grid on
plot(flat_delay_model_sasst(:,3),'r')
plot(misra_delay_model_sasst(:,3),'b')
plot(chao_delay_model_sasst(:,3),'k')
title('Total Tropospheric Delay using Sasstamoinen Model at
45\circ'),xlabel('Time (sec)'),ylabel('Total Trophospheric Delay (m)')
legend('Flat Earth Mapping','Misra and Enge Mapping','Chao Mapping')

figure(13)
hold on, grid on
plot(flat_delay_model_hop(:,3),'r')
plot(misra_delay_model_hop(:,3),'b')
plot(chao_delay_model_hop(:,3),'k')
title('Total Tropospheric Delay using Hopfield Model at
45\circ'),xlabel('Time (sec)'),ylabel('Total Trophospheric Delay (m)')
legend('Flat Earth Mapping','Misra and Enge Mapping','Chao Mapping')

figure(14)
hold on, grid on
plot(flat_delay_model_sasst(:,4),'r')
plot(misra_delay_model_sasst(:,4),'b')
plot(chao_delay_model_sasst(:,4),'k')
title('Total Tropospheric Delay using Sasstamoinen Model at
85\circ'),xlabel('Time (sec)'),ylabel('Total Trophospheric Delay (m)')
legend('Flat Earth Mapping','Misra and Enge Mapping','Chao Mapping')

figure(15)
hold on, grid on
plot(flat_delay_model_hop(:,4),'r')
plot(misra_delay_model_hop(:,4),'b')
plot(chao_delay_model_hop(:,4),'k')
title('Total Tropospheric Delay using Hopfield Model at
85\circ'),xlabel('Time (sec)'),ylabel('Total Trophospheric Delay (m)')
legend('Flat Earth Mapping','Misra and Enge Mapping','Chao Mapping')
```