Problem 1:

Equations used to do Problem 1:

Ionosphere free Pseudorange:

$$\rho^* = \frac{{f_{L1}}^2}{{f_{L1}}^2 - {f_{L2}}^2}\rho_{L1} - \frac{{f_{L2}}^2}{{f_{L1}}^2 - {f_{L2}}^2}\rho_{L2} \quad \text{.....(1)}$$

$$\rho^* = 2.546\rho_{L1} - 1.546\rho_{L2} \quad \text{.....(1A)}$$

$$\rho^* = r^i(t) + c \times t_{b,R,\rho} \quad \text{.....(1C)}$$

Equation (1C) helps us calculate receiver clock bias ($t_{b,R,\rho}$)from the ionosphere-free pseudorange measurements.

Ionosphere free Carrier phase at L1:

$$\varphi^* = \frac{{f_{L1}}^2}{{f_{L1}}^2 - {f_{L2}}^2}\phi_{L1} - \frac{f_{L1}f_{L2}}{{f_{L1}}^2 - {f_{L2}}^2}\phi_{L2} \quad \text{.....(2)}$$

$$\phi^* = 2.5457\phi_{L1} - 1.9837\phi_{L2} \quad \text{.....(2A)}$$

$${\phi_{L1}}^* = r \times \frac{1}{\lambda_{L1}} + t_{b,R,\varphi} f_{L1} \quad \text{.....(2B)}$$

Equation (2B) helps us calculate receiver clock bias ($t_{b,R,\rho}$) from the ionosphere-free carrier-phase measurements.

$$A = \frac{{f_{L1}}^2 * {f_{L2}}^2}{{f_{L2}}^2 - {f_{L1}}^2}(\rho_{L1} - \rho_{L2}) \quad \text{.....(3)}$$

Where A = 40.3*TEC.

Now calculating pseudorange at L1 or L2 frequency:

$$\rho_{L1} = \rho^{*} + \frac{A}{f_{L1}{}^{2}} \text{..... ....} (4)$$

$$\rho_{L2} = \rho^{*} + \frac{A}{f_{L2}{}^{2}} \text{..... ....} (4A)$$

True range was determined from:

$$r_{TRUE\_GEOMETRIC\_RANGE} = \sqrt{(X_R - X_S)^2 + (Y_R - Y_S)^2 + (Z_R - Z_S)^2} \text{..... ....} (5)$$

Oscillator frequency bias ($f_{b,S}$) at each L1 & L2 Doppler measurements was calculated with the following equation:

$$f_{b,S} = -f_D - \frac{\dot{r}}{\lambda} \text{..... ....} (6)$$

Following equations were used to complete part 6 of Problem 1:

$$\frac{\lambda}{2pi}\left[\varphi(t_2) - \varphi(t_1)\right] = \left[\rho(t_2) - \rho(t_1)\right] = \left[r(t_2) - r(t_1)\right] \text{..... ....} (7)$$

$$\rho(t_K) = \hat{\rho}_K(t_0) + \frac{\lambda}{2pi}\left[\varphi(t_K) - \varphi(t_0)\right] \text{..... ....} (7A)$$

$$\hat{\rho}_K(t_0) = \frac{1}{K}\sum_{i=0}^{K}\rho(t_i) - \frac{\lambda}{2pi}\left[\varphi(t_i) - \varphi(t_0)\right] \text{..... ....} (7B)$$

Next page shows Problem 1 Part 1's plots:

Part 1:

Figure 1 shows true range vs. GPS measured range. True range was calculated using equation (5). Pseudo-range was calculated using equation (4) while carrier phase range was calculated by (lamda_1* L1 Carrier Phase) from the data that was provided to us.

Figure 1:

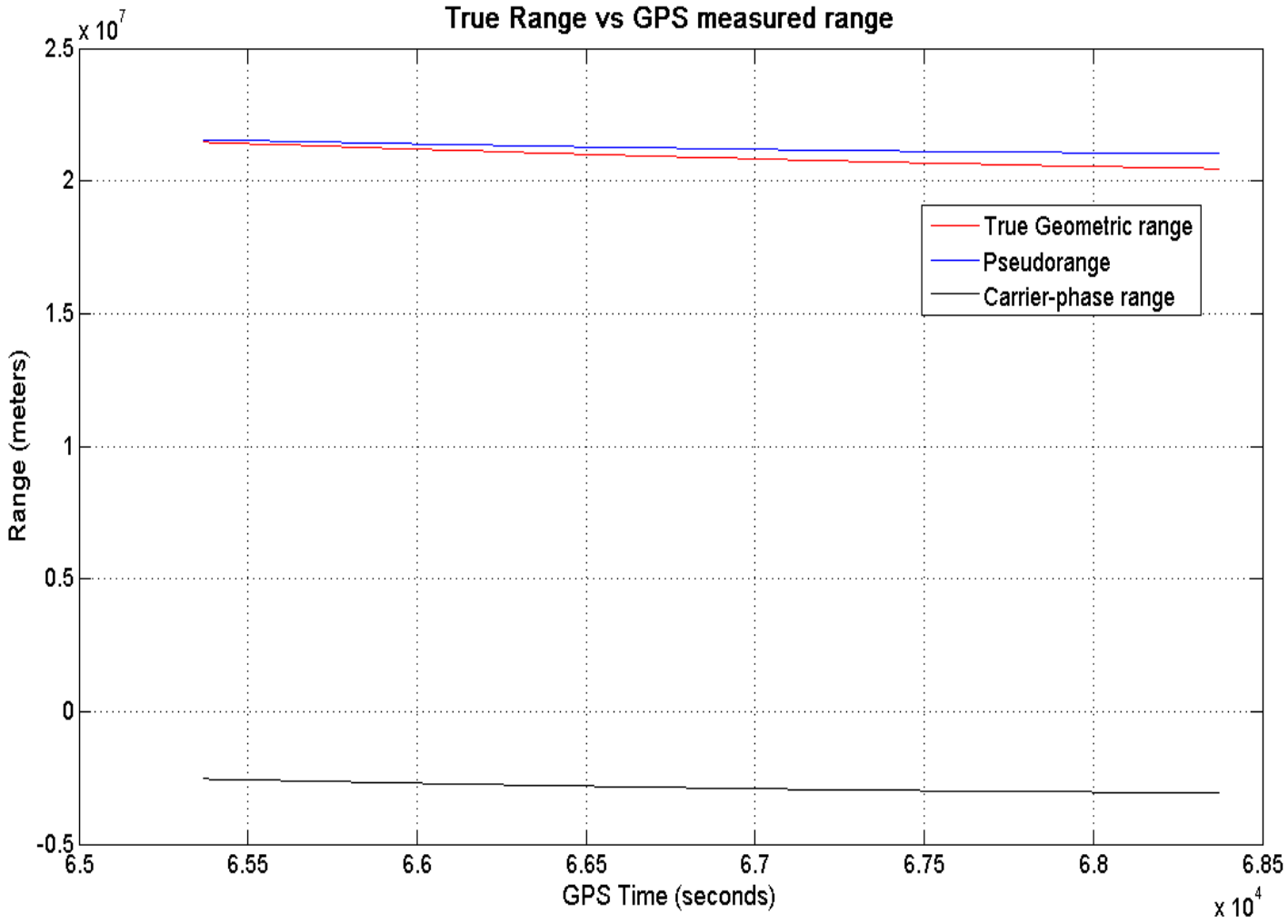


Figure 2:

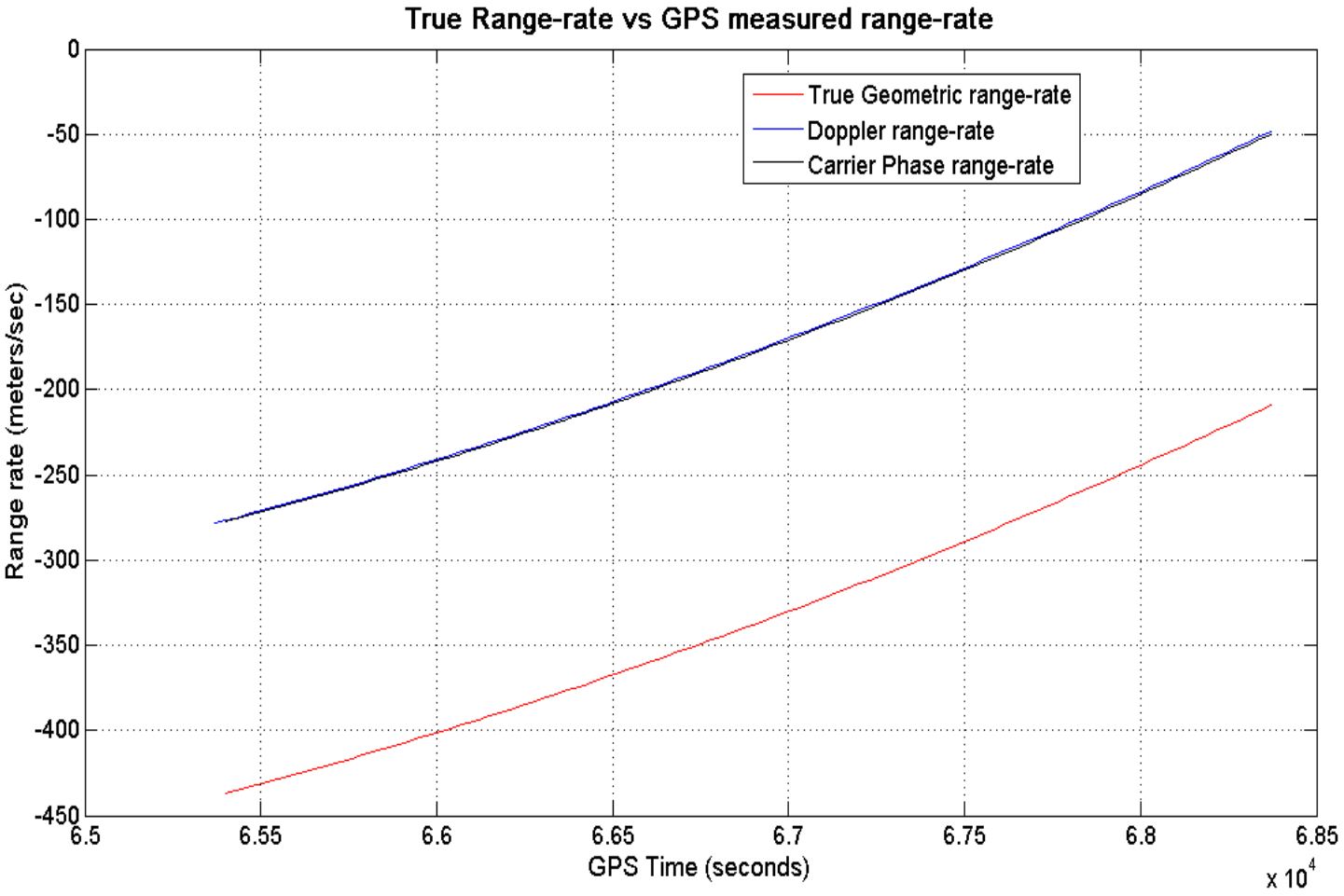


Figure 2 shows true range rate plot vs. GPS measured range rates. True range rate (or true Geometric range rate) was calculated by taking the time derivative of equation (5). Doppler range rate was found by multiplying lamda_1 with L1 Doppler frequency (given in Hz). Carrier phase range rate was found by taking time-derivate of ionosphere-free carrier phase and multiplying it by lamda_1. As can be seen in the plot, carrier phase range rate and Doppler range rate are almost equal.

Part 2:

Next page shows the oscillator frequency biases in both the L1 and L2 Doppler measurements (figure 3) and equivalent line of sight velocity bias in (m/sec) is shown in figure 4.

Figure 3:

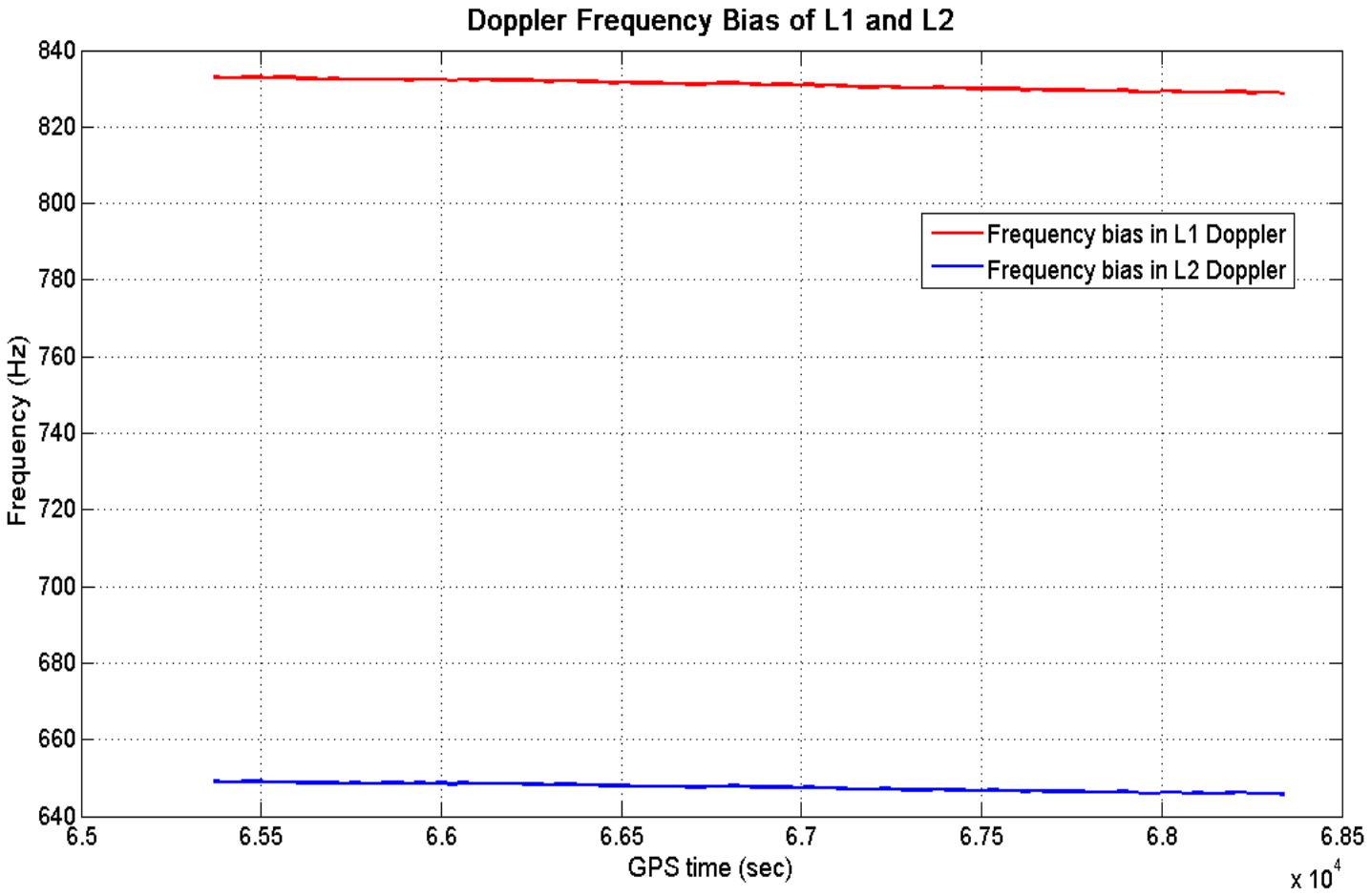


Figure 4:

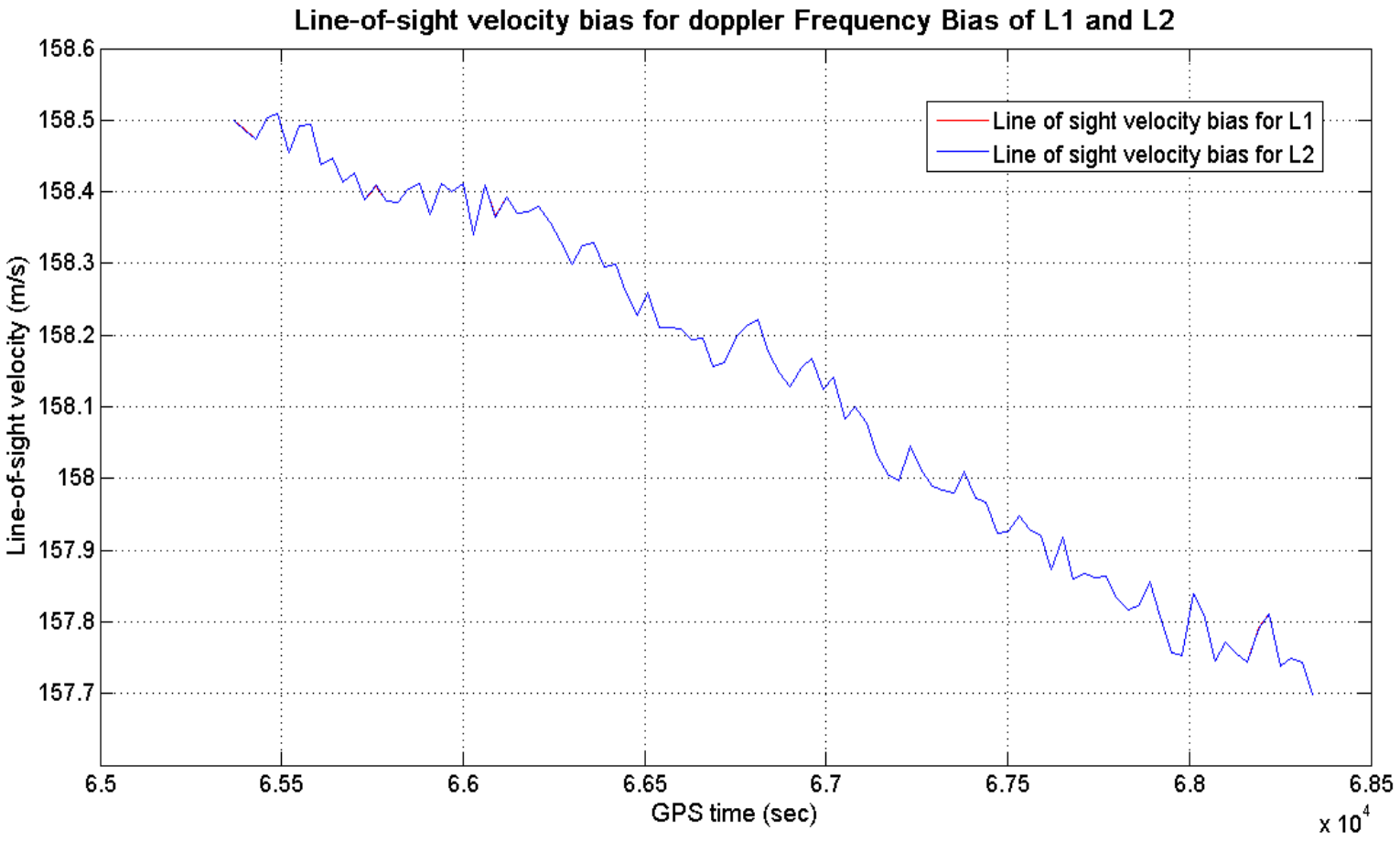


Figure 3 shows oscillator frequency biases in both L1 & L2 Doppler frequencies plotted as a function of GPS time. Oscillator frequency biases were estimated by using equation (6) which was evaluated at L1 and then L2 doppler measurements which were provided to us by the professor.

Figure 4 shows equivalent line of sight velocity bias. Since L1 & L2 doppler measurements came from the same oscillator, hence the converted line of sight velocity is the same for both L1 & L2 measurements.

Part 3 & 4:

Figure 5:

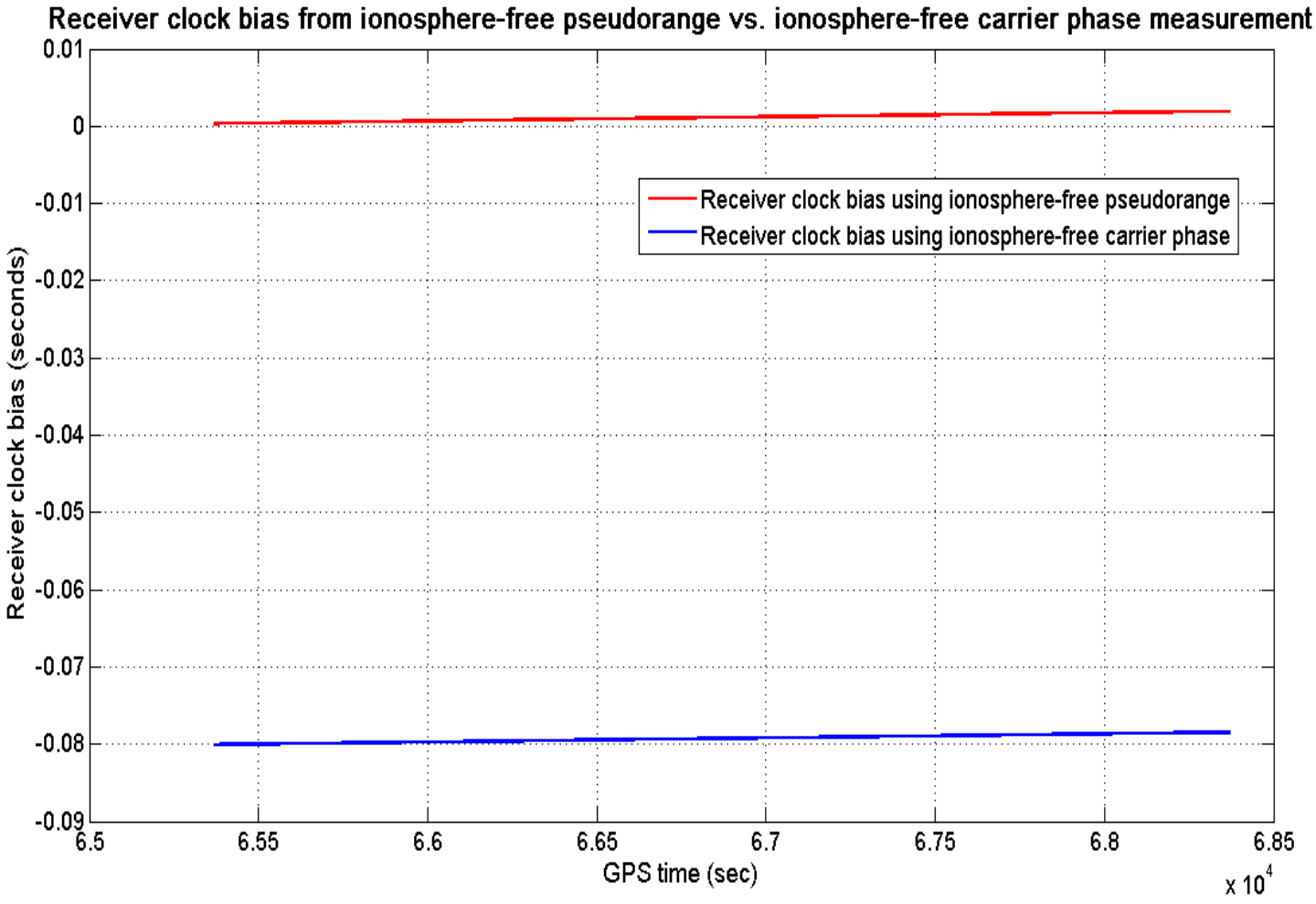


Figure 5 shows Receiver clock bias estimated from ionosphere-free pseudorange measurements vs. receiver clock bias which was estimated from ionosphere-free carrier phase measurements. Equation (1C) was used to calculate receiver clock bias from ionosphere-free pseudorange measurements while equation (2B) was used to estimate receiver clock bias from ionosphere-free carrier phase measurements. Receiver clock bias is given in seconds.
The reason for difference between estimated clock bias plot from ionosphere-free carrier phase measurements vs. plot for clock bias estimated from ionosphere-free carrier phase measurements is because the carrier-phase measurements suffers from an unknown integer ambiguity. The overall rising trend for both plots is same.

Figure 6 (shown below) shows a comparison between receiver clock bias rates computed from ionosphere-free pseudorange & carrier phase measurements vs. **L1** Doppler frequency bias. Clock bias **rates** were computed by taking a time derivate of receiver clock biases shown in figure 5.

When taking the time-derivative of receiver clock biases, we end up with clock bias rates which are dimensionless. However in order to compare receiver clock bias rates (dimensionless quantity) with L1 Doppler frequency bias (which have dimensions of Hertz), I ended up multiplying computed dimensionless clock rates with L1carrier frequency ($f\_{L1} = c/lamda\_{L1}$ =1575.42 Mhz). This way comparison between receiver clock bias rates from ionosphere-free pseudorange & ionosphere-free carrier-phase vs. Doppler frequency bias of L1 have common dimensions of Hertz.

It can be seen in figure 6 that both receiver clock bias rates computed from ionosphere-free pseudorange measurements & ionosphere-free carrier-phase measurements match very nicely with each other. Receiver clock bias rate from ionosphere-free pseudorange measurements is more smooth while clock bias rate computed from ionosphere-free pseudorange measurements has more noise present.

As can be seen in the figure, both receiver clock bias rates also match very nicely with Doppler frequency bias of L1. They are off by only a few hertz while all 3 plots follow a common decreasing trend as time passes by.

Figure 6:

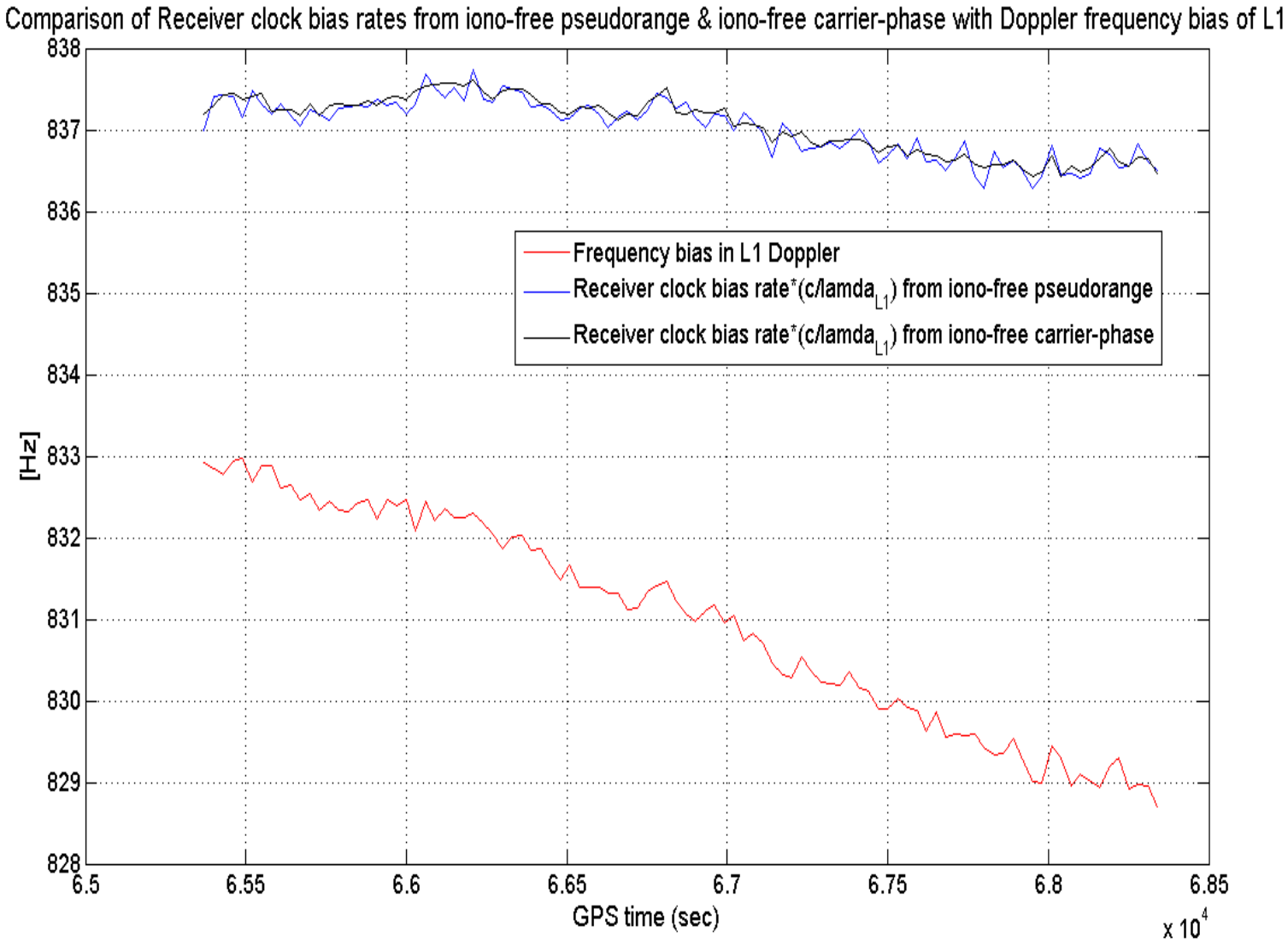


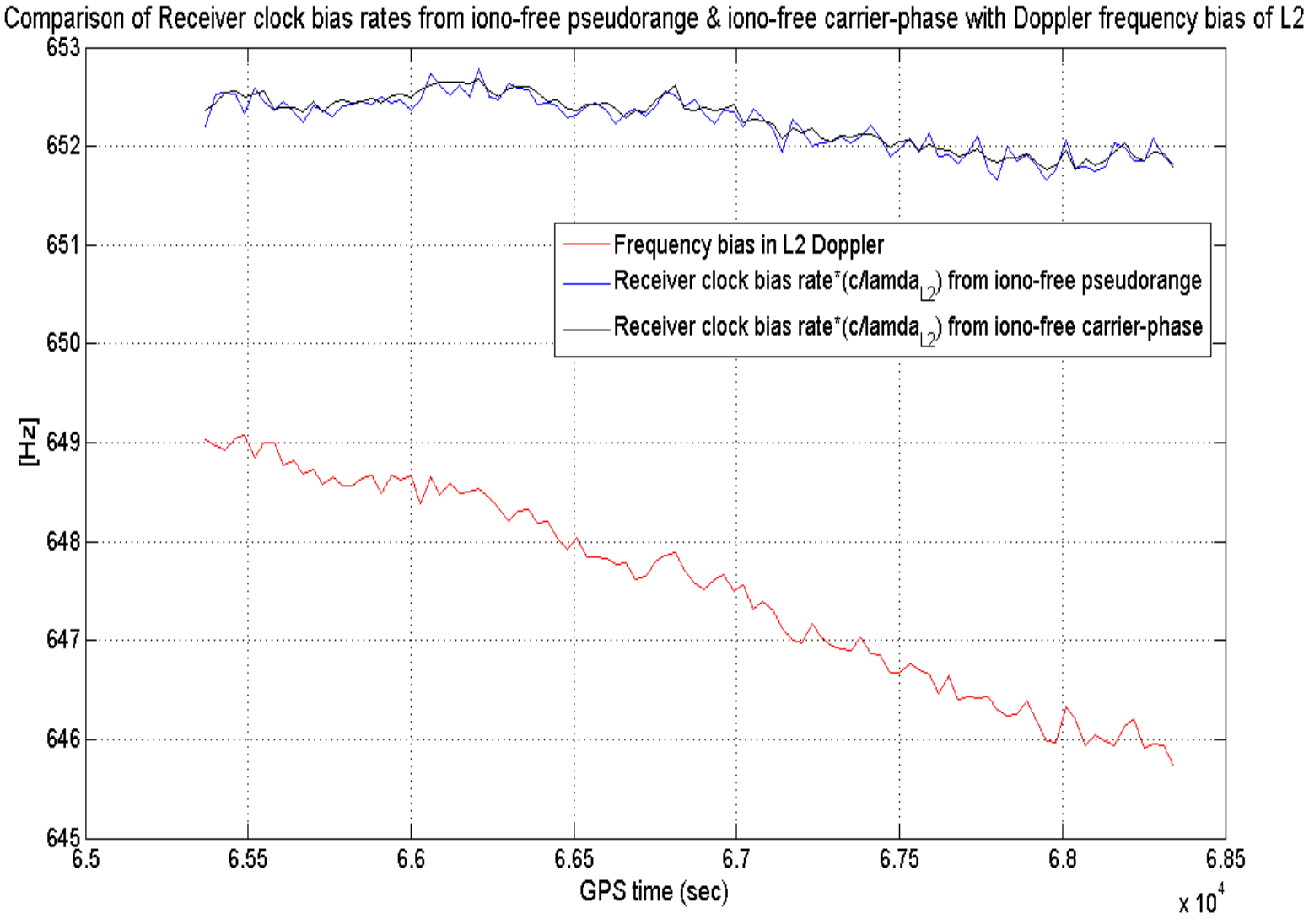
Figure 7 (shown below) shows a comparison between receiver clock bias rates computed from ionosphere-free pseudorange & carrier phase measurements vs. **L2** Doppler frequency bias. Clock bias **rates** were computed by taking a time derivate of receiver clock biases shown in figure 5.

When taking the time-derivative of receiver clock biases, we end up with clock bias rates which are dimensionless. However in order to compare receiver clock bias rates (dimensionless quantity) with L2 Doppler frequency bias (which have dimensions of Hertz), I ended up multiplying computed dimensionless clock rates with L2 carrier frequency ($f\_{L2} = c/lamda\_{L2}$ =1227.60 Mhz). This way comparison between receiver clock bias rates from ionosphere-free pseudorange & ionosphere-free carrier-phase vs. Doppler frequency bias of L2 have common dimensions of Hertz.

It can be seen in figure 7 that both receiver clock bias rates computed from ionosphere-free pseudorange measurements & ionosphere-free carrier-phase measurements match very nicely with each other. Just like in figure 6, receiver clock bias rate from ionosphere-free pseudorange measurements is more smooth while clock bias rate computed from ionosphere-free pseudorange measurements has more noise present.

Just like in figure 6, it can be seen in the figure 7, that both receiver clock bias rates also match very nicely with Doppler frequency bias of L2. They are off by only a few hertz while all 3 plots follow a common decreasing trend as time passes by.

Figure 7:



Part 6:

Part 6 illustrates the use of carrier phase to smooth the pseudorange measurements. We know that the carrier measurement is much more precise but it suffers from an unknown integer ambiguity. However, as long as this ambiguity does not change (i.e. no cycle slips) the change in phase between two time steps, and the change in pseudorange between two time steps, will both be unambiguous estimates of the change in geometric range between those two times. However, the error in the carrier phase difference would be much lower. This is illustrated in equation (7) shown on page 2.

One approach to combining the 2 measurements, for some set of data starting at time *t0* is to generate an accurate estimate of the pseudorange at the start of the data set ($\hat{\rho}_K(t_0)$), by averaging all data up to time t_k.

The smoothed pseudorange at time t_k was computed by using equation (7A). The estimate of ($\hat{\rho}_K(t_0)$) was found by averaging the pseudorange measurements available up to time t_k. After they have been corrected to the value at time *t0* using the phase measurement given by equation (7B).

Figure 8 (shown below) shows the estimated $\hat{\rho}_K(t_0)$ which was generated using ionosphere-free values of **rho** and **phi** at **L1** carrier frequency values.

Figure 8:

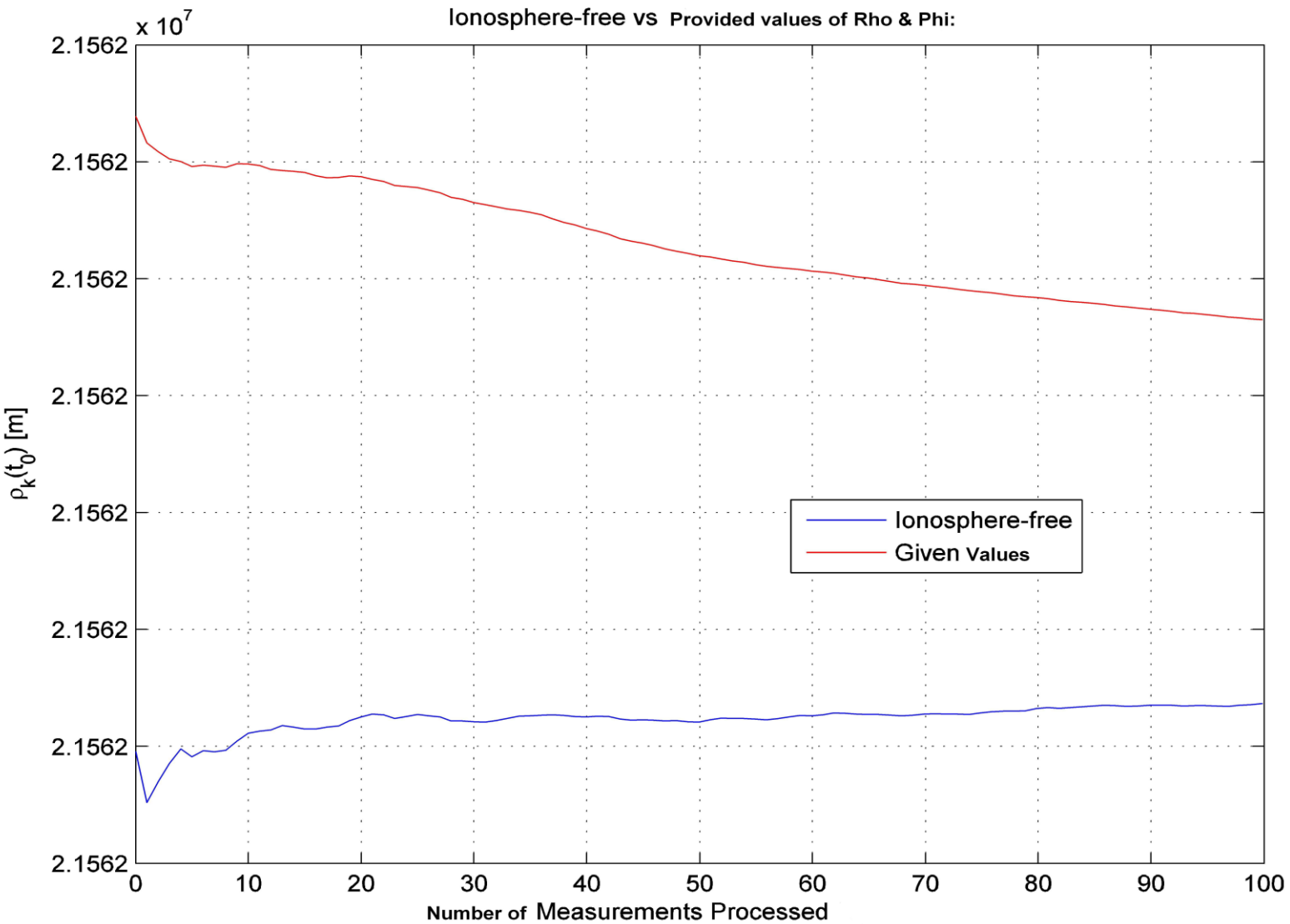


Figure 8 shows that after combining code and carrier, the pseudorange plot becomes much smoother as oppose to pseudorange plots which are generated when not combining code and carrier. The phase measurements, since they give less error or noise in their measurements helps us reduce the error in pseudorange measurements after implementing the combining of code and carrier method as illustrated by equations 7, 7A & 7B.

Problem 2:

Equations used in **Klobuchar Model** to predict Ionosphere delay:

$$\psi(EL) = \frac{0.0137}{EL + 0.11} - 0.022 \quad \text{.....(8)}$$

$$\phi_I \approx \phi_R + \psi * \cos(AZ) \quad \text{.....(9)}$$

$$\lambda_I \approx \lambda_R + \frac{\psi * \sin(AZ)}{\cos(\phi_I)} \quad \text{.....(10)}$$

$$\phi_M = \phi_I + 0.064 * \cos(\lambda_I - 1.617) \quad \text{.....(11)}$$

$$A_2 = \sum_{n=0}^{3} \phi_n \phi_M{}^n \quad \text{.....(12)}$$

$$A_4 = \sum_{n=0}^{3} \beta_n \phi_M{}^n \quad \text{.....(13)}$$

$$t = 43200\,\lambda_I + t_{GPS} \quad \text{.....(14)}$$

$$x = \frac{2\pi * (t - 50400)}{A_4} \quad \text{.....(15)}$$

$$\tau_Z = 5 \times 10^{-9} + A_2 \left[1 - \frac{x^2}{2} + \frac{x^4}{24} \right] \quad if \quad |x| \le 1.57 \quad \text{.....(16)}$$

$$F = 1.0 + 16 * (0.53 - EL)^3 \quad \text{.....(17)}$$

$$\tau_{IONOSPHERE} = F\,\tau_Z \quad \text{.....(18)}$$

Problem 3:

Equations used to compute the ionosphere delay at **L1** frequency from pseudorange measurements at L1 & L2 are as follows:

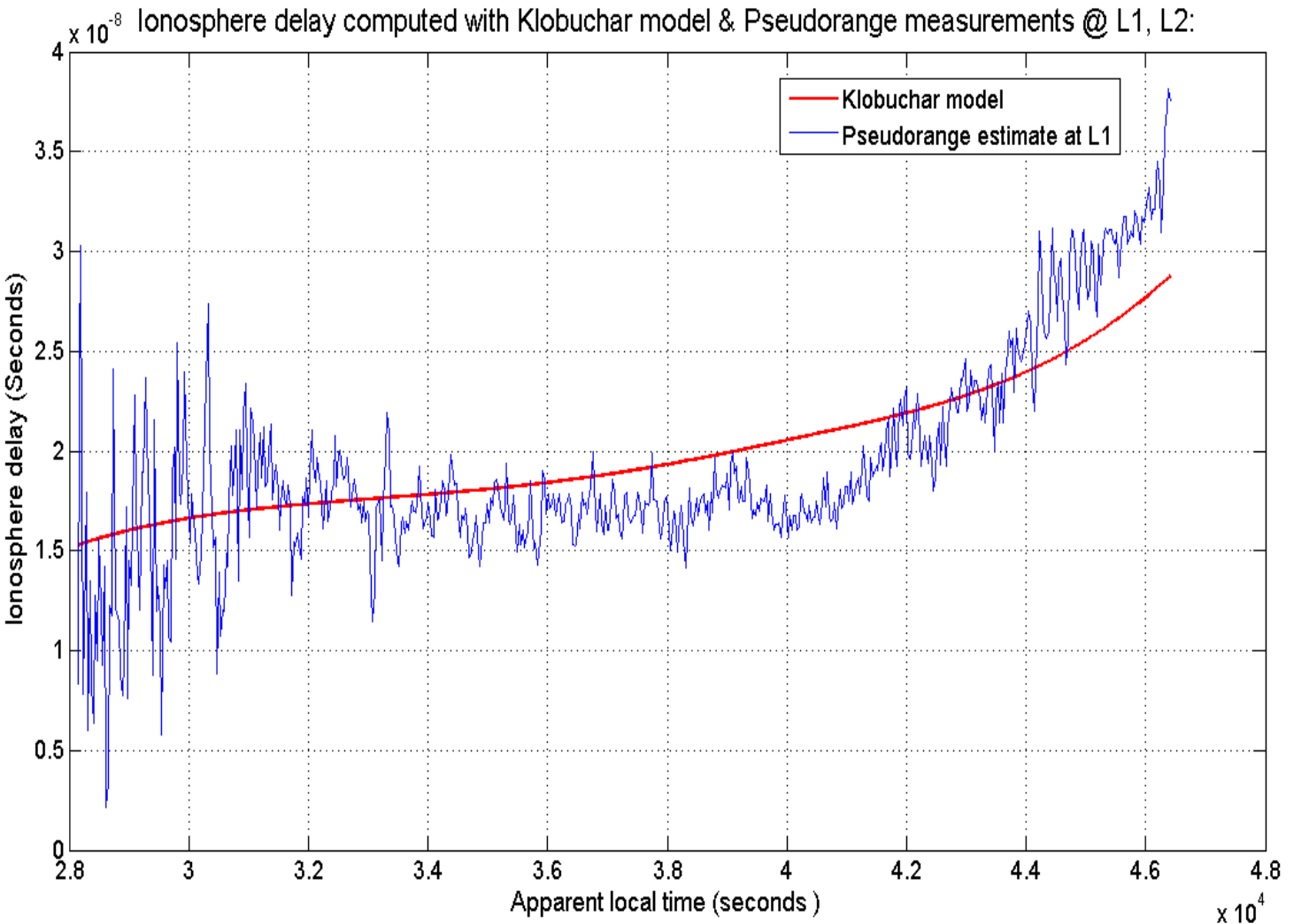
$$A = \frac{f_{L1}^{\ 2} * f_{L2}^{\ 2}}{f_{L2}^{\ 2} - f_{L1}^{\ 2}}(\rho_{L1} - \rho_{L2}) \text{.....} \text{.....} \text{.....} \text{.....} (19)$$

$$\tau_{IONOSPHERE\ \_delay\_AT\ \_L1} = \frac{A}{c * f_{L1}^{\ 2}} \text{.....} \text{.....} \text{.....} \text{.....} (20)$$

Using the above equations from Problem 2 & 3, Ionosphere delay computed using the Klobuchar model and ionosphere delay at **L1** frequency computed from pseudorange measurements at L1 & L2 is shown in figure 9 below. It can be seen that ionospheric delay predicted from Klobuchar model matches fairly well with the ionospheric delay which was computed from pseudorange measurements at L1 and L2.

Since the pseudorange measurements used in computing Ionosperic delay estimate at L1 frequency has noise and measurement errors present in it, hence that is why the ionospheric delay at L1 frequency (blue color) looks jittery and shaky. Overall the Klobuchar model nicely predicts the ionosphere delay measurements data. The ionosphere delay is given in seconds here.

Figure 9:



Problem 4:

Equations used in **Sasstamoinen Model** to predict the wet and dry zenith tropospheric delay:

$$e_0 = 6.108 * RH * \exp\left[\frac{17.15T - 4684}{T - 38.5}\right] \text{..... ....(21)}$$

$$c\tau_{T,Z,D} = 0.0022777 * (1 + 0.0026 * \cos(2\phi)) P_0 \text{..... ....(22)}$$

$$c\tau_{T,Z,W} = 0.0022777 * (\frac{1255}{T} + 0.05) e_0 \text{..... ....(23)}$$

Equations used in **Hopfield Model** to predict the wet and dry zenith tropospheric delay:

$$h_d = 40136 + 148.72 * (T - 273.16) \text{..... ....(24)}$$

$$c\tau_{T,Z,D} = 77.64 \times 10^{-6} * (\frac{P_0 h_d}{5T}) \text{..... ....(25)}$$

$$h_W = 11000 \ \ meters \text{..... ....(26)}$$

$$c\tau_{T,Z,W} = 0.373 * (\frac{e_0 h_W}{5T^2}) \text{..... ....(27)}$$

Next page shows plots of both dry & wet zenith tropospheric delay from Saastamoinen and the Hopfield models:

Figure 10:

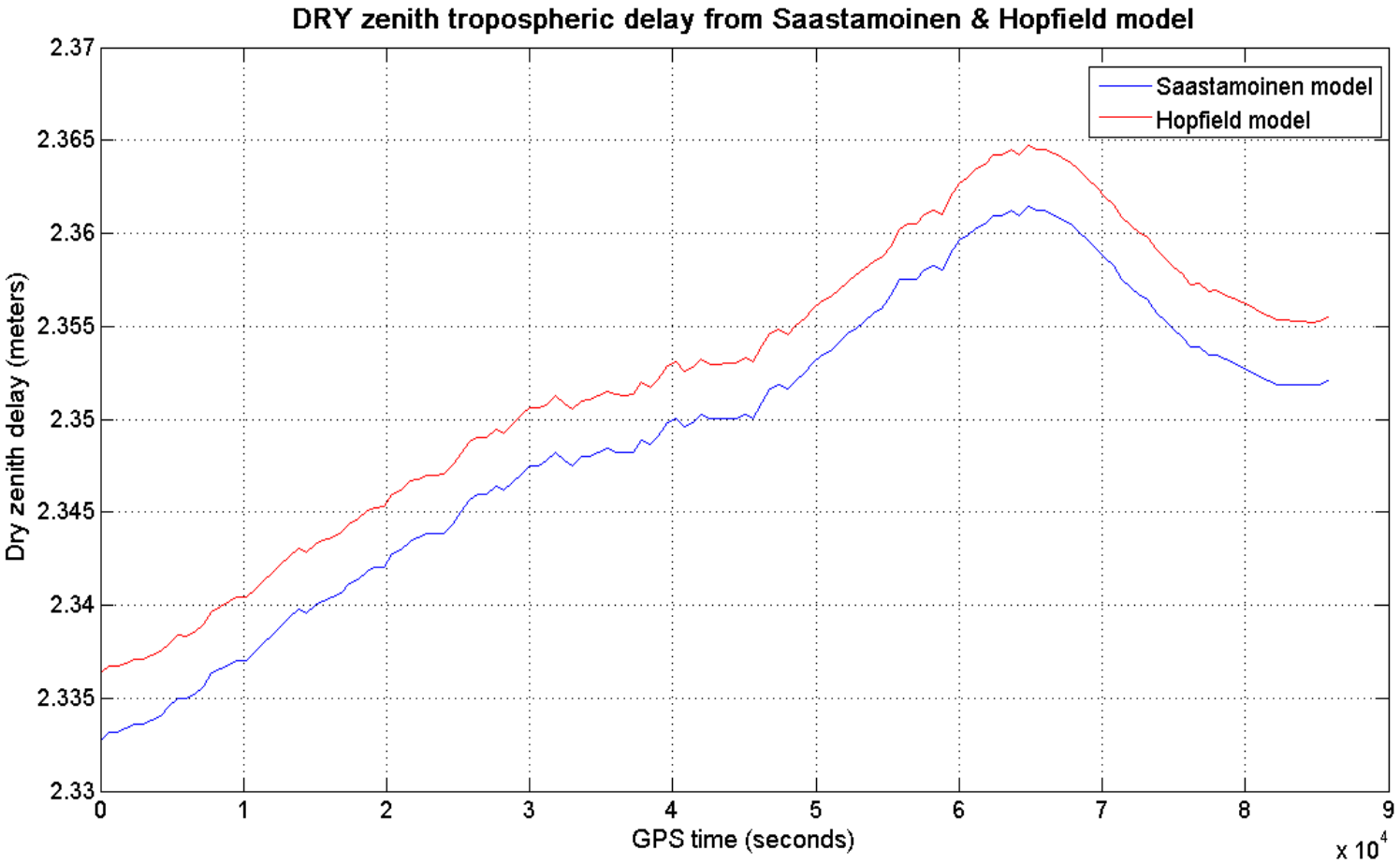


Figure 11:

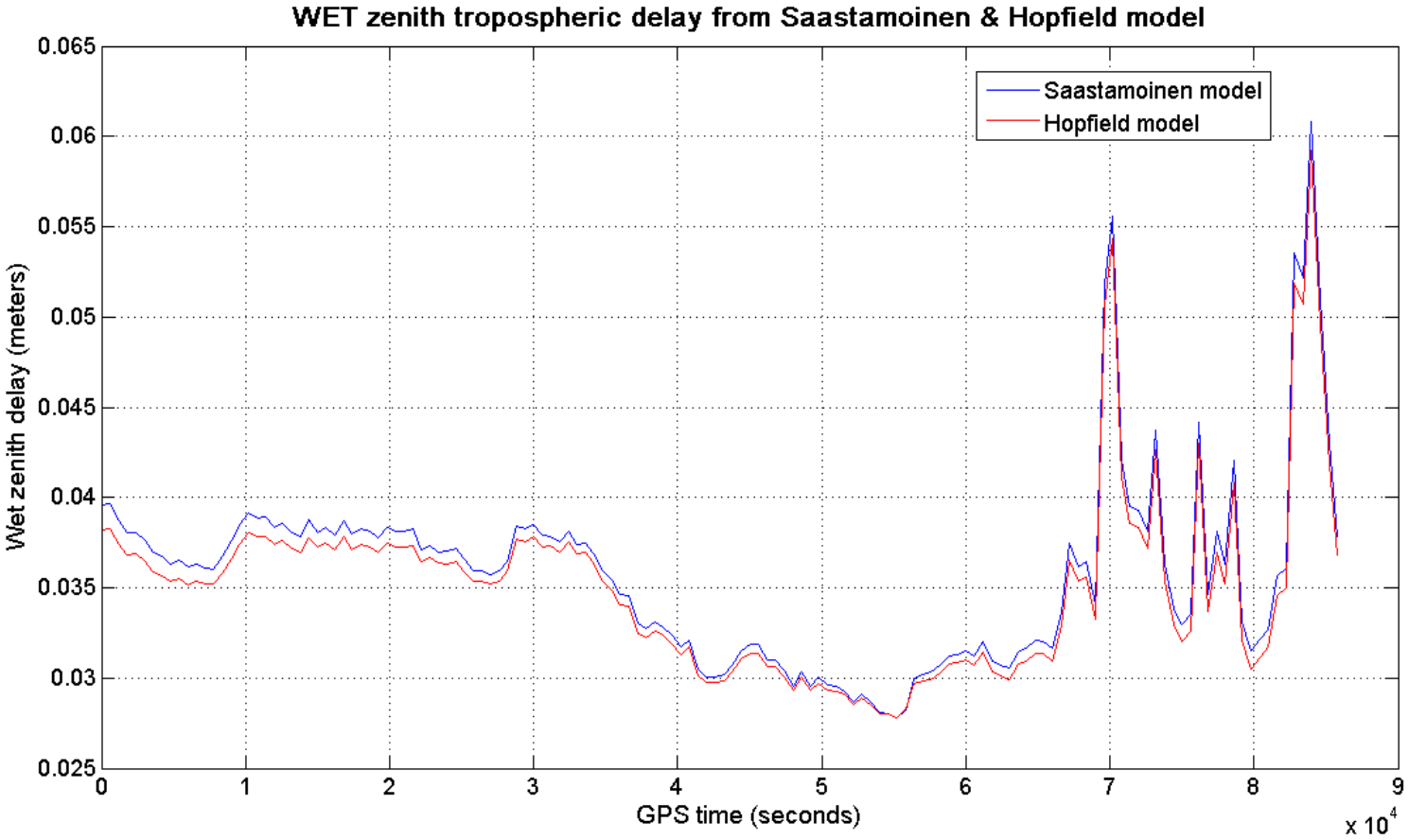


Figure 10 shows the **DRY** zenith tropospheric delay using both the Saastamoinen and the Hopfield model. It can be seen that the dry zenith troposphere delay predictions from both models match each other very well and for all practical purposes, give almost identical results.

Figure 11 shows the **WET** zenith tropospheric delay using both the Saastamoinen and the Hopfield model. It can be seen that the wet zenith troposphere delay predictions from both models also match each other very well and for all practical purposes, give almost identical results. After studying both figures 10 & 11, it is concluded that both the Saastamoinen and the Hopfield models give very identical results.

Problem 5:

Simple *Flat-Earth* mapping functions will be used to compute **total** tropospheric delay:

$$m_W(EL) = m_D(EL) = \frac{1}{\sin(EL)} \quad \text{.....(28)}$$

Mapping functions given in *Misra & Enge* will be used to compute **total** tropospheric delay:

$$m_W(EL) = m_D(EL) = \frac{1}{\sqrt{1-\left(\frac{\cos(EL)}{1.001}\right)^2}} \quad \text{.....(29)}$$

Separate mapping functions for wet & dry delays given by *Chao* will be used to compute **total** tropospheric delay:

$$m_W(EL) = \frac{1}{\sin(EL)+\left(\frac{0.00035}{Tan(EL)+0.017}\right)} \quad \text{.....(30A)}$$

$$m_D(EL) = \frac{1}{\sin(EL)+\left(\frac{0.00143}{Tan(EL)+0.0445}\right)} \quad \text{.....(30B)}$$

$$\tau_{Total\_Tropospheric\_Delay} = \tau_{T,Z,W} * m_W(EL) + \tau_{T,Z,D} * m_D(EL) \quad \text{.....(31)}$$

Figure 12:

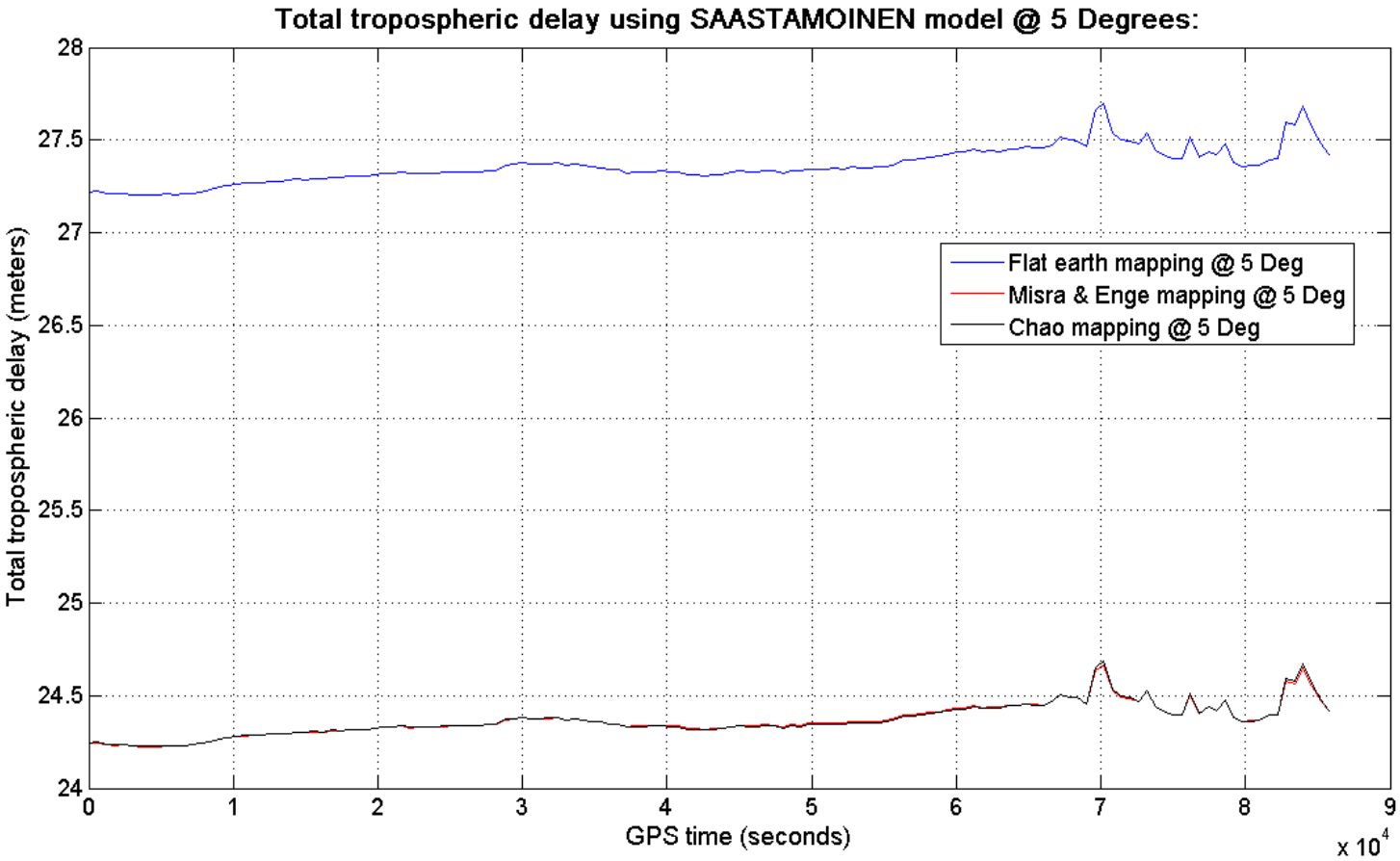


Figure 13:

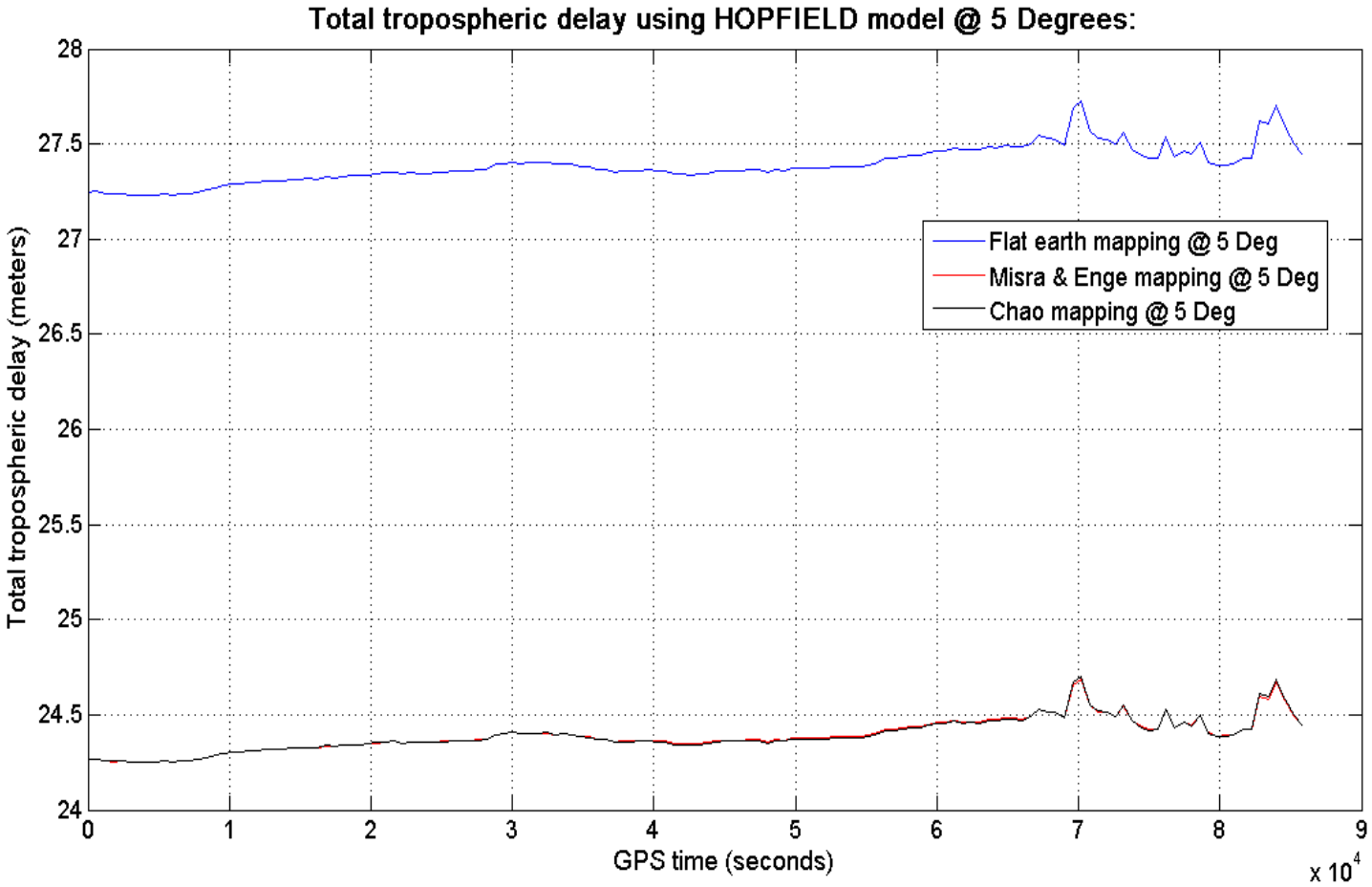


Figure 14:

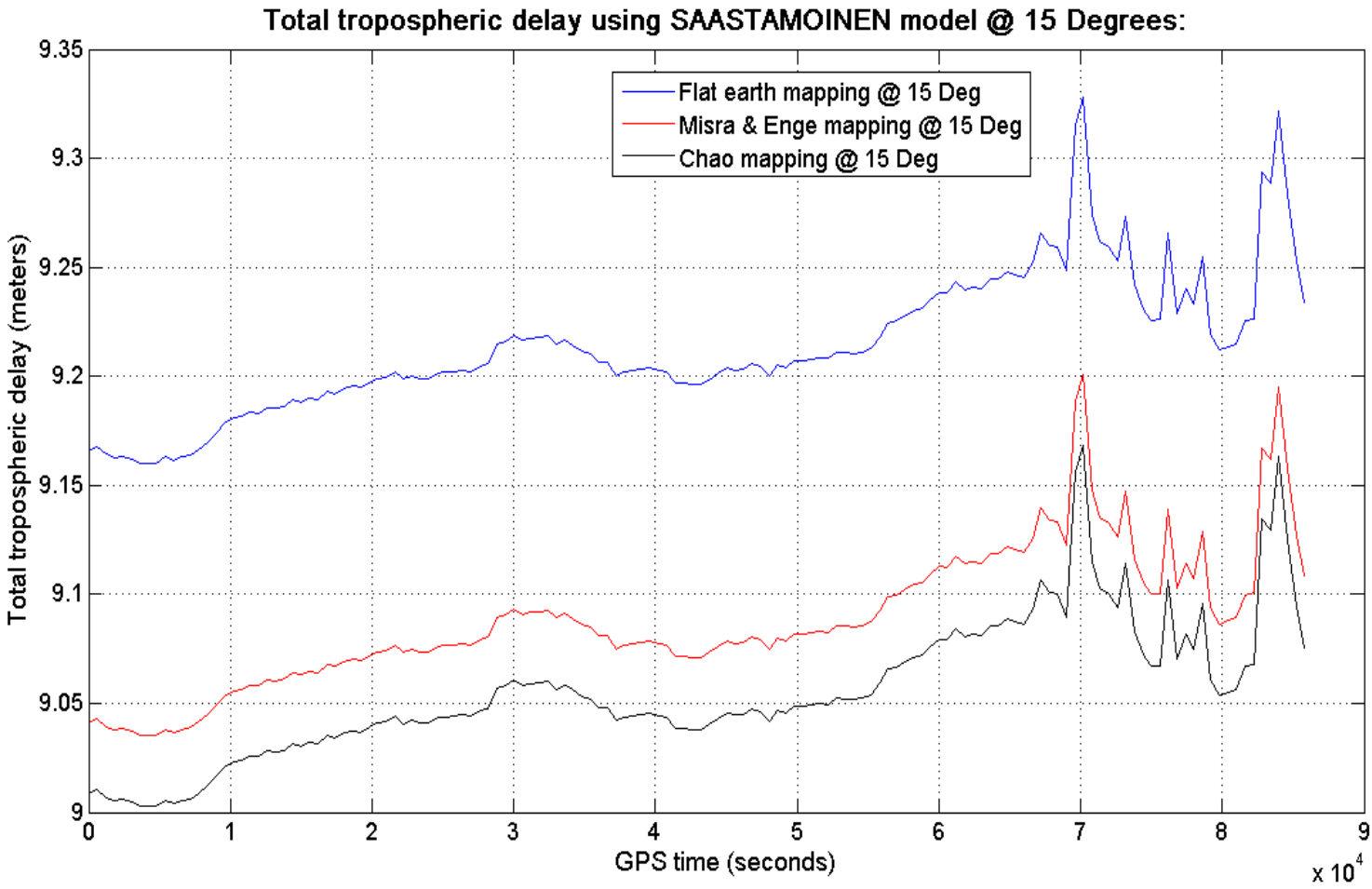


Figure 15:

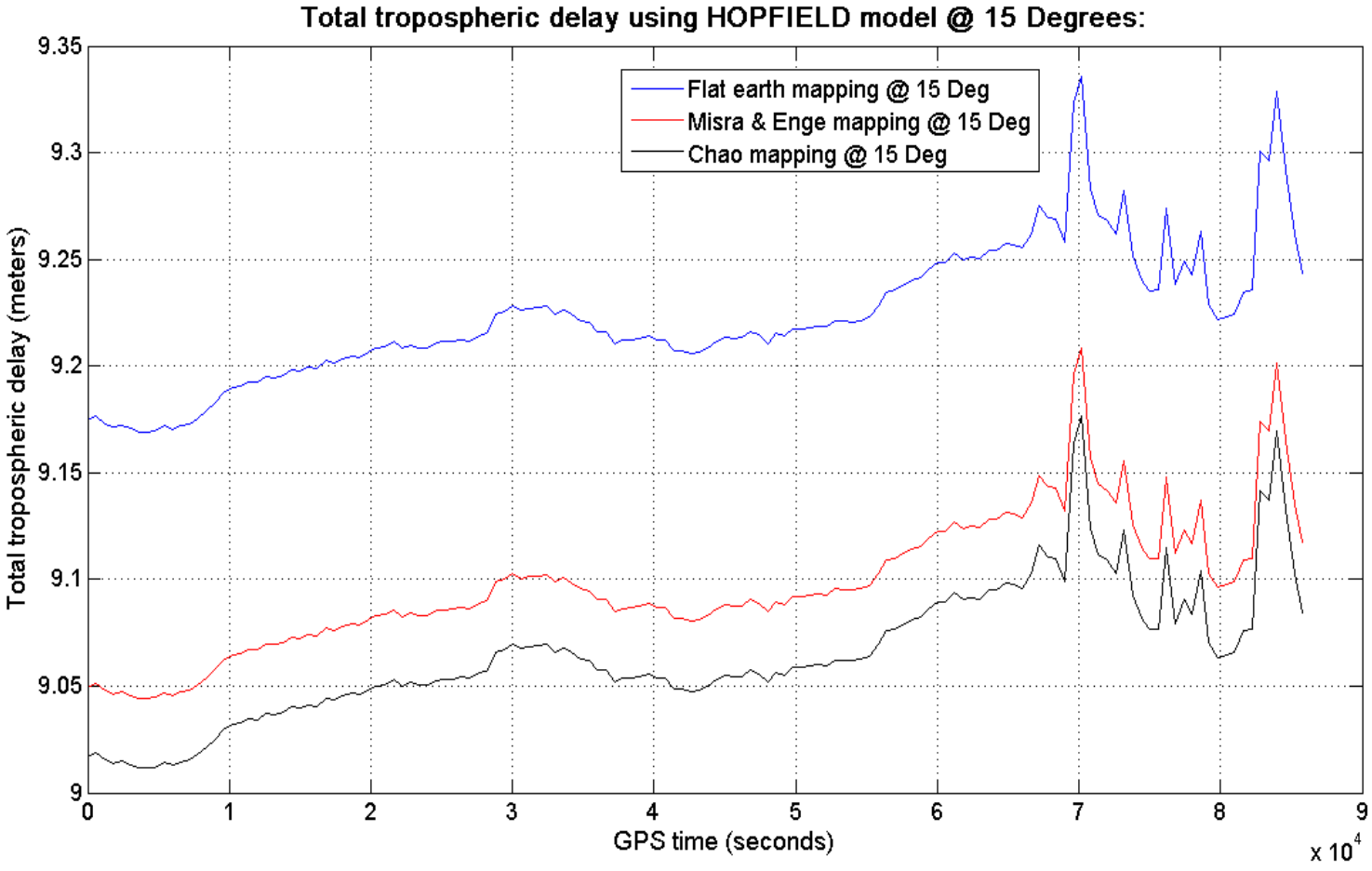


Figure 16:

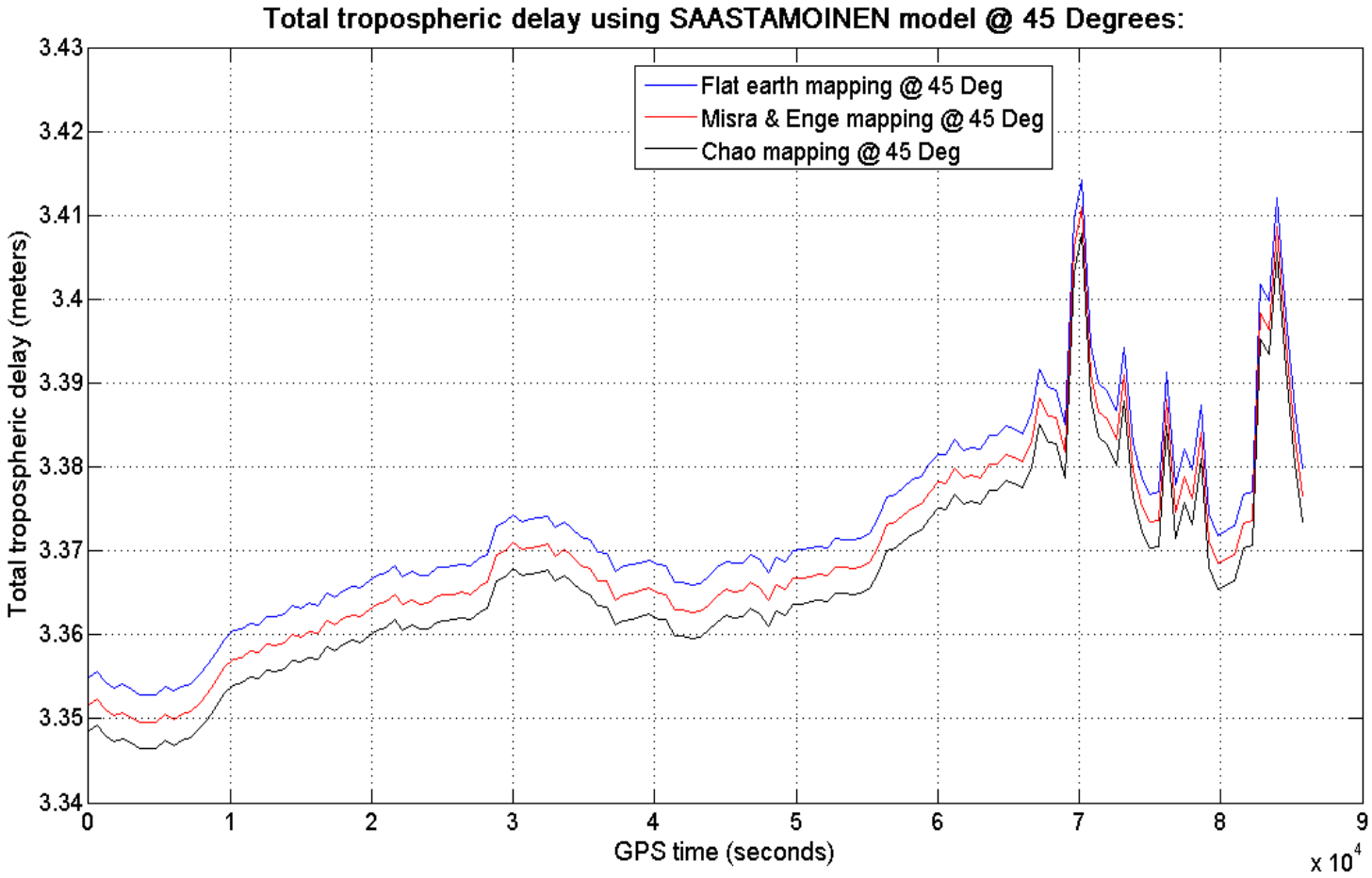


Figure 17:

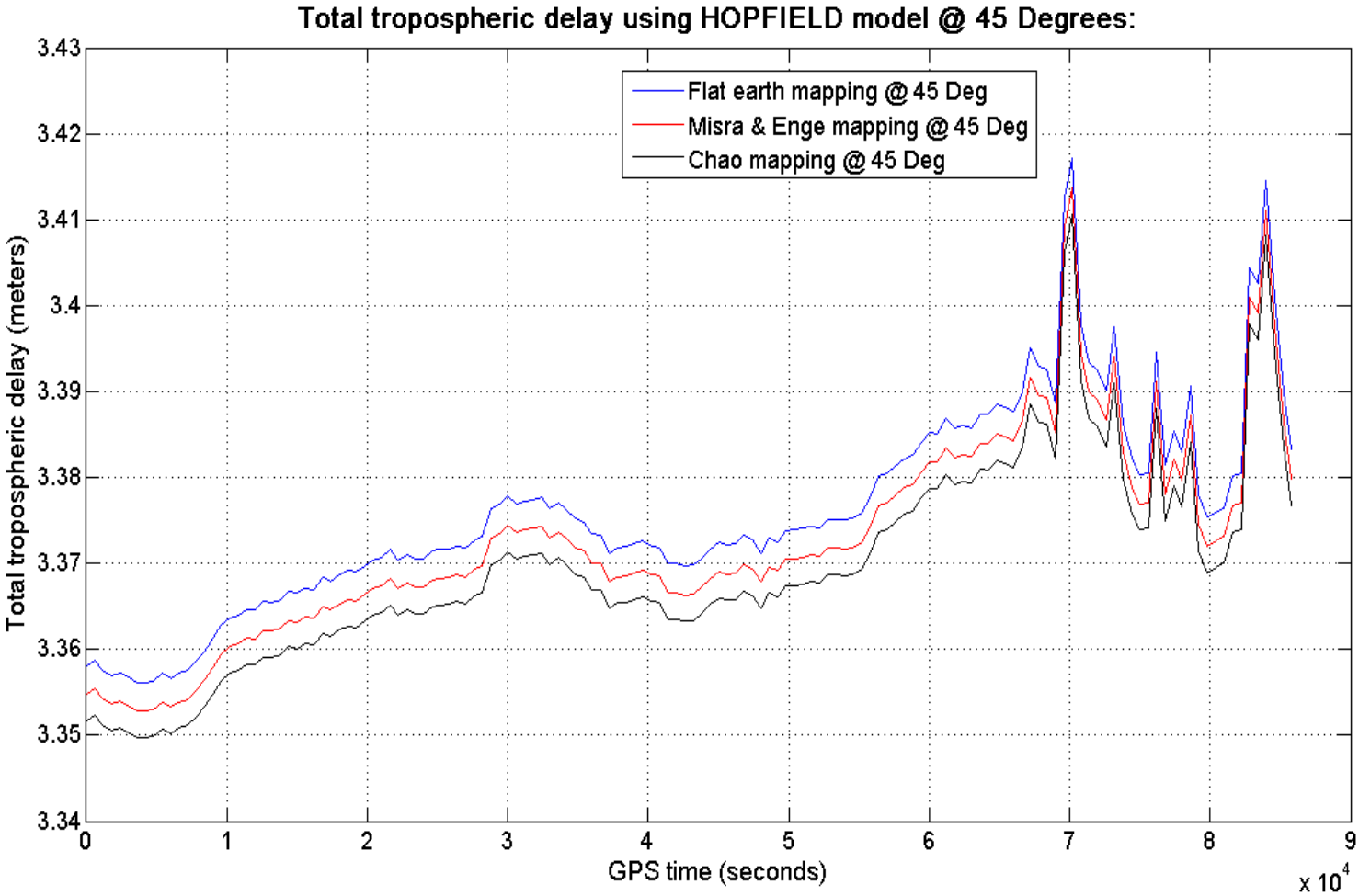


Figure 18:

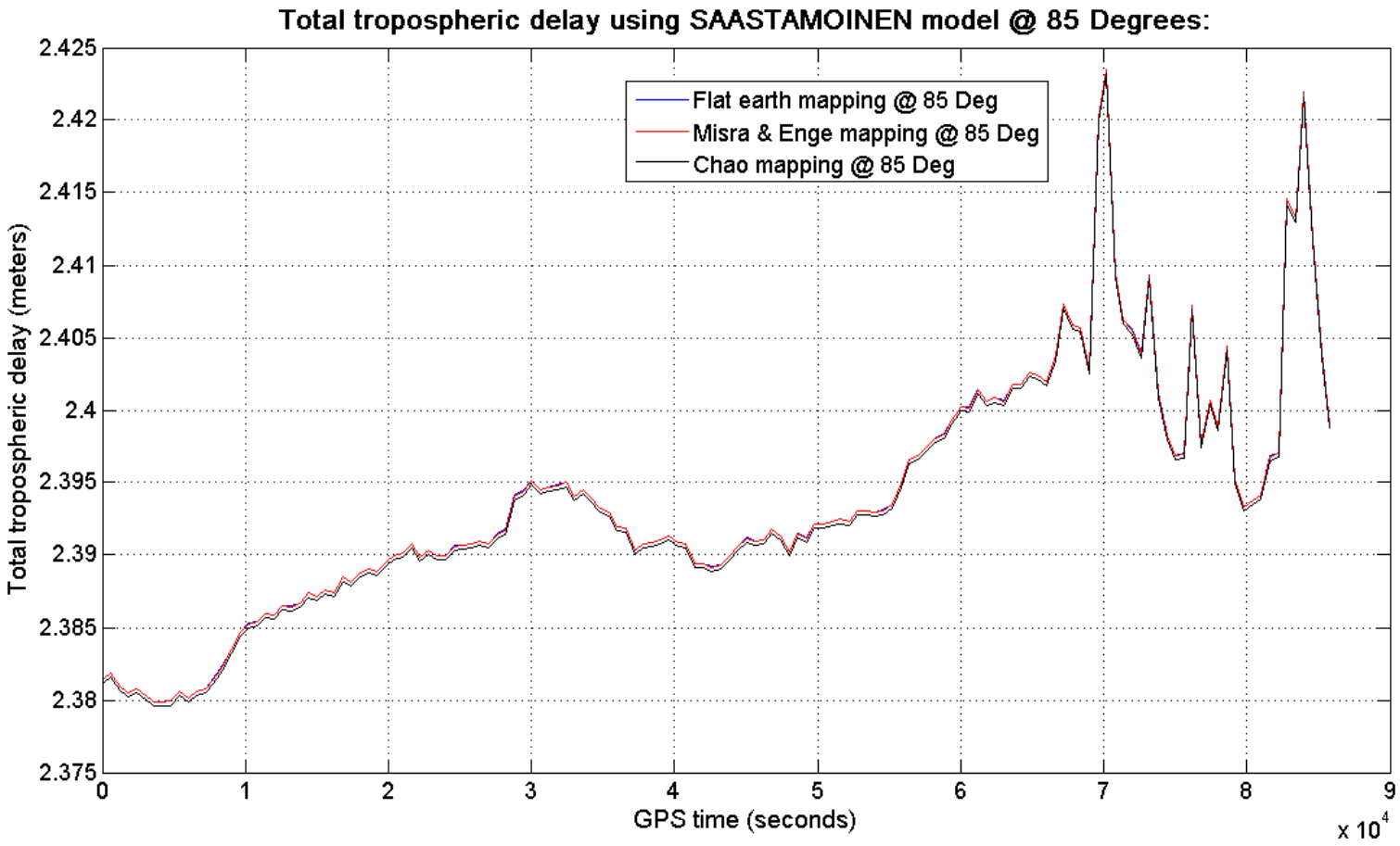
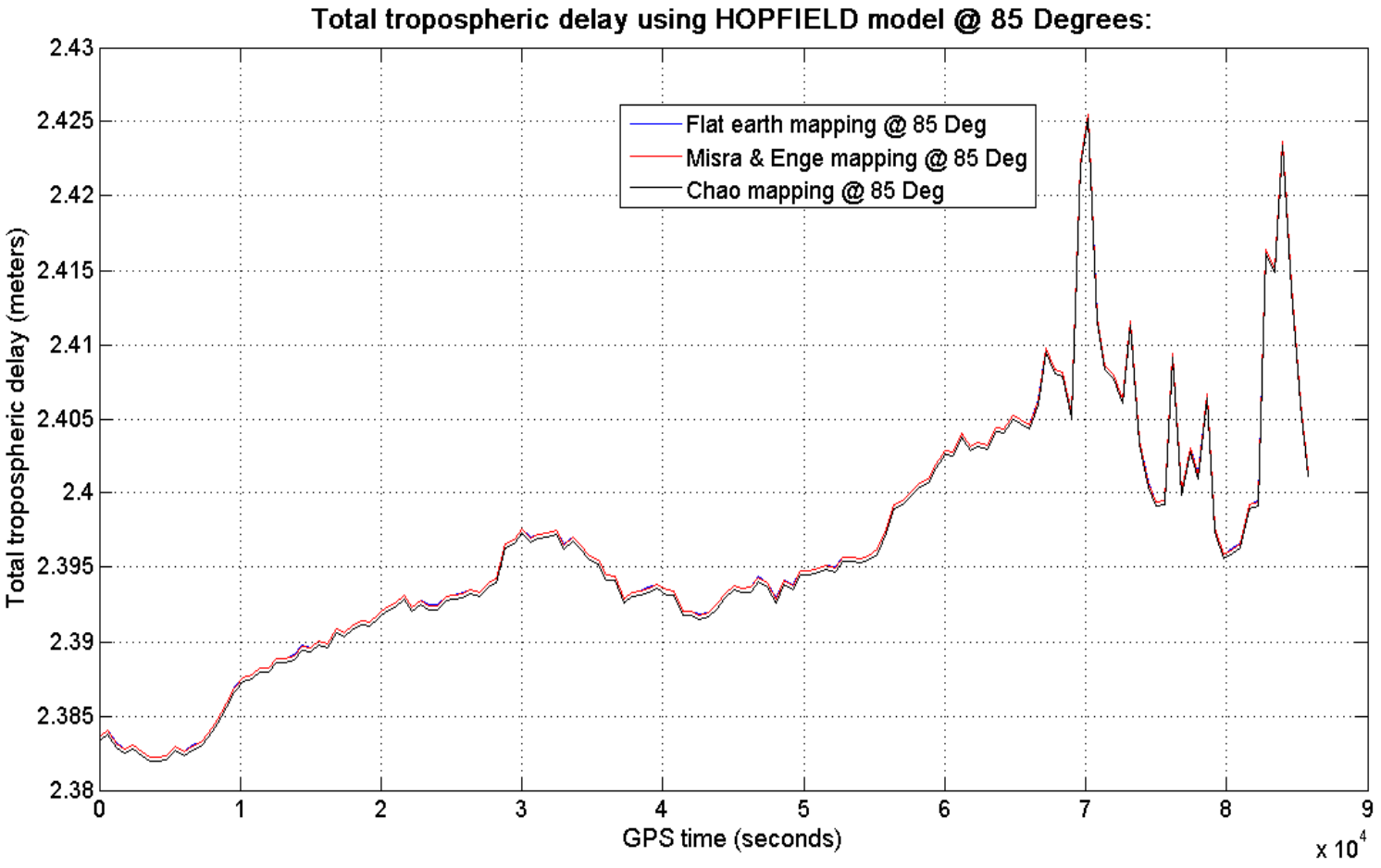


Figure 19:



Figures 12 & 13 show the total troposphere delay at elevation angle of **5°** for both Saastamoinen and the Hopfield models using 3 different mapping functions (i.e. flat earth, Misra & Enge, and Chao). Mapping functions in Misra/Enge & Chao give identical total troposphere delay (given in meters) while total troposphere delay using flat earth mapping function does not match total delay results given by mapping functions of Misra/Enge & Chao. The reason that total delay using flat earth mapping function does not match with delay computed by Misra/Enge & Chao is because the flat earth mapping function does **NOT** work well for elevation angles which are < 15°.
As can be seen in figures 12 & 13, both Saastamoinen and the Hopfield models predict almost identical results for total troposphere delay.

Figures 14 & 15 show the total troposphere delay at elevation angle of **15°** for both Saastamoinen and the Hopfield models using 3 different mapping functions (i.e. flat earth, Misra & Enge, and Chao). Mapping functions in Misra/Enge & Chao give identical total troposphere delay and now at elevation angle of 15°, total troposphere delay using flat earth mapping function comes closer (but not identical) to matching the results that are given by mapping functions of Misra/Enge & Chao. It is to be noted that as the satellite elevation angle increased from 5° to 15°, the total troposphere delay has decreased, on average, by roughly 2.5 times when compared to total delay at 5°. As can be seen in figures 14 & 15, both the Saastamoinen and the Hopfield models continue to predict almost identical results for total troposphere delay.

Figures 16 & 17 show the total troposphere delay at satellite elevation angle of **45°** for both Saastamoinen and the Hopfield models using 3 different mapping functions (i.e. flat earth, Misra & Enge, and Chao). This time all 3 mapping functions give almost identical total troposphere delay with very little difference between the 3 mapping functions' predictions of total delay. Now as the satellite elevation angle increased from 5° to 45°, the total troposphere delay has decreased, on average, by roughly 8 times when compared to total delay at 5°. Both Saastamoinen and the Hopfield models continue to predict almost identical results for total troposphere delay.

Finally figures 18 & 19 show the total troposphere delay at satellite elevation angle of **85°** for both Saastamoinen and the Hopfield models using 3 different mapping functions (i.e. flat earth, Misra & Enge, and Chao). Now all 3 mapping functions, for all practical purposes, give identical total troposphere delay. Now as the satellite elevation angle increased from 5° to 85°, the total troposphere delay has decreased, on average, by roughly 10.6 times when compared to total delay at 5°.

This exercise nicely proves that at low elevation angles, total troposphere delay is very high and as the satellite elevation angle continues to increase, the total tropospheric delays continues to go decrease. This exercise also shows that for high elevation angles (greater than 15 degrees), all 3 mapping functions give almost identical results.

Finally it has been discretely proven that both Saastamoinen and the Hopfield models predict almost identical results for total troposphere delay. ☺

Appendix A:

MATLAB code used to solve **problem 1**:

```
% AAE 575. HW 3. Rizwan Qureshi.
% PROBLEM 1:

clear all; clc; close all;

load orbit0_sv_13.mat;
load rawdata_sv_13.mat;

c = 299792458;  % speed of light in [m/sec]
f1 = 1575.42e06;
f2 = 1227.60e06;
lamda1 = c/(1575.42e06);
lamda2 = c/(1227.60e06);

orbit_sv_13 = orbit0;
raw_data_sv_13 = rawdata;

GPS_t1 = raw_data_sv_13(:,1);   % [secs]

L1_P_range = raw_data_sv_13(:,2);   % [meters]
L2_P_range = raw_data_sv_13(:,3);   % [meters]
L1_C_phase = raw_data_sv_13(:,4);    % [cycles]
L2_C_phase = raw_data_sv_13(:,5);    % [cycles]
L1_D_freq = raw_data_sv_13(:,6);    % [Hz]   [i.e. cycles/second]
L2_D_freq = raw_data_sv_13(:,7);    % [Hz]

% Time when satellite PRN13 is tracked
GPS_t2 = orbit_sv_13(:,1);   % [secs]. GPS_t1 = GPS_t2

% Satellite PRN 13's x,y,z position coordinates:
x_s = orbit_sv_13(:,2);   % [meters]
y_s = orbit_sv_13(:,3);   % [meters]
z_s = orbit_sv_13(:,4);   % [meters]

% Location of Receiver which is on the GROUND:
x_R = -1288337.0539;    % [meters]
y_R = -4721990.4483;   % [meters]
z_R = 4078321.6617;    % [meters]



%
% phi_star_L1 = f1^2/(f1^2-f2^2)*phi_L1 - f1*f2/(f1^2-f2^2)*phi_L2
% or
% phi_star_L1 = 2.5457*phi_L1 - 1.9837*phi_L2;

% PART 1:

%Compute True Geometric range (r(t)):
r_true = sqrt([(ones(1,101)'*x_R - x_s).^2 + (ones(1,101)'*y_R - y_s).^2 +
(ones(1,101)'*z_R - z_s).^2]) ;  % TRUE Geometric range [meters]

% TRUE Geometric range rate:
True_Geometric_range_rate = diff(r_true)./diff(GPS_t1);  % [m/sec]
```

```
% Ionosphere free pseudorange
rho_star_R = 2.546*L1_P_range - 1.546*L2_P_range;   % [meters]

% A value:
A=((1227.6e6^2)*(1575.42e6^2)/(1227.6e6^2-1575.42e6^2))*(L1_P_range-L2_P_range);

% Pseudorange: rho_L1
pseudorange_at_L1 = [rho_star_R] + [A./(1575.42e6^2)];    % same as given L1_P_range
variable!. L1_P_range is = L2_P_range for all purposes

% Ionosphere free carrier phase
phi_star_L1 = [2.5457*L1_C_phase] - [1.9837*L2_C_phase];   %[cycles]

carrier_phase_range_L1 = lamda1* phi_star_L1;  % gives nearly SAME plot as Justin's
formula

%Doppler range rate @ L1   JUSTIN proposed formula
doppler_range_rate_L1 = -lamda1*L1_D_freq;   % meters* cycles/sec = [meters/sec]

%phi_L1 range rate:
phi_L1_range_rate=(c/f1)*[diff(phi_star_L1)./diff(GPS_t1)];    %meters*(cyles/seconds) =
[m/sec]

% Part 2:

f_L1_bias = -L1_D_freq(1:length(True_Geometric_range_rate)) -
True_Geometric_range_rate/lamda1;
f_L2_bias = -L2_D_freq(1:length(True_Geometric_range_rate)) -
True_Geometric_range_rate/lamda2;

L_of_sight_L1=lamda1*f_L1_bias;    %[m/sec]
L_of_sight_L2=lamda2*f_L2_bias;   %[m/sec]


%Part 3:
format long
% Receiver clock bias from pseudorange
receiver_bias_pseudo_r= [rho_star_R - r_true]/c;   % [seconds]
Receiver_clock_bias_rate=diff(receiver_bias_pseudo_r)./diff(GPS_t1);  %[dimensionless]

%Part 4:
%Receiver clock bias from ionosphere-free carrier phase:

receiver_bias_carrier_phase = (1/f1)*[phi_star_L1 - r_true/lamda1];  % [seconds]
receiver_clock_biase_rate_phi = diff(receiver_bias_carrier_phase)./diff(GPS_t1);   %
[dimensionless]

%Part 1:
figure(9)
plot(GPS_t1,r_true,'r')
hold on
plot(GPS_t1,pseudorange_at_L1,'b')
plot(GPS_t1,carrier_phase_range_L1,'k')
xlabel('GPS Time (seconds)')
ylabel('Range (meters)')
title('True Range vs GPS measured range')
legend('True Geometric range', 'Pseudorange', 'Carrier-phase range')
grid on

figure(10)
```

```
plot(GPS_t1(2:101),True_Geometric_range_rate,'r')
hold on
plot(GPS_t1,doppler_range_rate_L1,'b')
plot(GPS_t1(2:101),phi_L1_range_rate,'k')
xlabel('GPS Time (seconds)')
ylabel('Range rate (meters/sec)')
title('True Range-rate vs GPS measured range-rate')
legend('True Geometric range-rate','Doppler range-rate', 'Carrier Phase range-rate')
grid on

% Part 2:
figure(11)
plot(GPS_t1(1:100),f_L1_bias,'r')
hold on
plot(GPS_t1(1:100),f_L2_bias,'b')
title('Doppler Frequency Bias of L1 and L2')
xlabel('GPS time (sec)')
ylabel('Frequency (Hz)')
legend('Frequency bias in L1 Doppler', 'Frequency bias in L2 Doppler')
grid on

figure(12)
plot(GPS_t1(1:100),L_of_sight_L1,'r')
hold on
plot(GPS_t1(1:100),L_of_sight_L2,'b')
title('line-of-sight velocity bias for doppler Frequency Bias of L1 and L2')
xlabel('GPS time (sec)')
ylabel('Line-of-sight velocity (m/s)')
legend('Line of sight velocity bias for L1', 'Line of sight velocity bias for L2')
grid on

%
%Part 3:
figure(13)
plot(GPS_t1,receiver_bias_pseudo_r,'r')
title('Receiver clock bias from ionosphere-free pseudorange measurement')
xlabel('GPS time (sec)')
ylabel('Receiver clock bias (seconds)')
grid on


figure(14)
plot(GPS_t1(1:100),Receiver_clock_bias_rate,'r')
title('Receiver Clock bias rate from ionosphere-free pseudorange measurement')
xlabel('GPS time (sec)')
ylabel('dimensionless quantity')
grid on

figure(15)
plot(GPS_t1(1:100),f_L1_bias,'r')
hold on
plot(GPS_t1(1:100),f1*Receiver_clock_bias_rate,'b')
title('Comparison of Receiver clock bias rate from iono-free pseudorange with Doppler
frequency bias of L1')
xlabel('GPS time (sec)')
ylabel('[Hz]')
grid on
legend('Frequency bias in L1 Doppler', 'clock bias rate*(c/lamda_L_1)')

figure(16)
plot(GPS_t1(1:100),f_L2_bias,'r')
hold on
```

```
plot(GPS_t1(1:100),f2*Receiver_clock_bias_rate,'b')
title('Comparison of Receiver clock bias rate from iono-free pseudorange with Doppler
frequency bias of L2')
xlabel('GPS time (sec)')
ylabel('[Hz]')
grid on
legend('Frequency bias in L2 Doppler', 'clock bias rate*(c/lamda_L_2)')

%Part 4:
figure(17)
plot(GPS_t1,receiver_bias_pseudo_r,'r')
hold on
plot(GPS_t1,receiver_bias_carrier_phase,'b')
title('Receiver clock bias from ionosphere-free pseudorange vs. ionosphere-free carrier
phase measurement')
xlabel('GPS time (sec)')
ylabel('Receiver clock bias (seconds)')
legend('Receiver clock bias using ionosphere-free pseudorange','Receiver clock bias using
ionosphere-free carrier phase')
grid on

figure(18)
plot(GPS_t1(1:100),f_L1_bias,'r')
hold on
plot(GPS_t1(1:100),f1*Receiver_clock_bias_rate,'b')
plot(GPS_t1(1:100),f1*receiver_clock_biase_rate_phi,'k')
title('Comparison of Receiver clock bias rate from iono-free pseudorange & iono-free
carrier-phase with Doppler frequency bias of L1')
xlabel('GPS time (sec)')
ylabel('[Hz]')
grid on
legend('Frequency bias in L1 Doppler', 'Receiver clock bias rate*(c/lamda_L_1) from iono-
free pseudorange','Receiver clock bias rate*(c/lamda_L_1) from iono-free carrier-phase')

figure(19)
plot(GPS_t1(1:100),f_L2_bias,'r')
hold on
plot(GPS_t1(1:100),f2*Receiver_clock_bias_rate,'b')
plot(GPS_t1(1:100),f2*receiver_clock_biase_rate_phi,'k')
title('Comparison of Receiver clock bias rate from iono-free pseudorange & iono-free
carrier-phase with Doppler frequency bias of L2')
xlabel('GPS time (sec)')
ylabel('[Hz]')
grid on
legend('Frequency bias in L2 Doppler', 'Receiver clock bias rate*(c/lamda_L_2) from iono-
free pseudorange','Receiver clock bias rate*(c/lamda_L_2) from iono-free carrier-phase')


%Part 6:
figure(20)
plot(GPS_t1,receiver_bias_pseudo_r,'r')
hold on
plot(GPS_t1,receiver_bias_carrier_phase,'b')
title('Receiver clock bias from ionosphere-free pseudorange vs. ionosphere-free carrier
phase measurement')
xlabel('GPS time (sec)')
ylabel('Receiver clock bias (seconds)')
legend('Receiver clock bias using ionosphere-free pseudorange','Receiver clock bias using
ionosphere-free carrier phase');grid on
```

MATLAB code used to solve **problems 2 & 3**:

```
% AAE 575. HW 3. Rizwan Qureshi.
% PROBLEM 2 & 3:

clear all; clc; close all;

load dualfreq_sv1.mat;

sv_1_K_model = dualfreq;

GPS_t1 = sv_1_K_model(:,1);   % [secs]

sv_1_AZ = sv_1_K_model(:,2)/180;   % [Azimuth angle DEG]
sv_1_EL = sv_1_K_model(:,3)/180;   % [Elevation angle. DEG]

L1_P_range = sv_1_K_model(:,4);    % [meters]
L2_P_range = sv_1_K_model(:,5);     % [meters]

L1_C_phase = sv_1_K_model(:,6);   % [cycles]
L2_C_phase = sv_1_K_model(:,7);   % [cycles]

% Location of Receiver which is on the GROUND:
x_R = -1288337.0539;    % [meters]
y_R = -4721990.4483;   % [meters]
z_R = 4078321.6617;    % [meters]

%Determine Geodetic lat and longitude of Receiver:
lla  = ecef2lla([x_R, y_R, z_R]);
latitude_R = lla(1)/180;   % [Deg]
longitude_R = lla(2)/180;  % [Deg]
height_R = lla(3);    % [meters]

% Determine Psi using Approximation:
% psi_deg = [(445./(sv_1_EL + 20))-4];   % Psi [Deg]
psi_deg = [0.0137./(sv_1_EL+0.11)] - 0.022;

% Determine latitude of IPP:
phi_I_Deg = [latitude_R*ones(1,611)]' + [(psi_deg.*cosd(sv_1_AZ))]; % Latitude of IPP
[deg]

% Determine longitude of IPP:
lamda_I_Deg = [longitude_R*ones(1,611)]' + [(psi_deg.*(sind(sv_1_AZ)))./cosd(phi_I_Deg)];
% Longitude of IPP [deg]

% Determine Geomagnetic latitude of the IPP:
phi_M_deg = phi_I_Deg + [0.064*cosd(lamda_I_Deg-1.617)];  %[deg]

%Now compute A2 and A4:

format long

% A2 and A4 in SC units. Prof said the inputs MUST be in SC units:
A2 =  [0.028e-6*(phi_M_deg).^0] + [-0.007e-6*(phi_M_deg).^1] + [-0.119e-6*(phi_M_deg).^2]
+ [0.119e-6*(phi_M_deg).^3];
A4 =  [137.0e3*(phi_M_deg).^0] + [-49.0e3*(phi_M_deg).^1] + [-131.0e3*(phi_M_deg).^2] +
[-262.0e3*(phi_M_deg).^3];


t = [43200*(lamda_I_Deg)]+ GPS_t1;
```

```
x = [2*pi*(t-50400)]./A4;

% delay_Z = 5e-9 + (A2.*cos(x));
delay_Z = 5e-9 + [A2.*(1- (x.^2)/2 + (x.^4)/24)];  %Gives same answer if using cos(x)!


% F semi-circle model
F = 1.0 + (16*(0.53-sv_1_EL).^3);

Ionosphere_delay_K_model = F.* delay_Z;   % in [Seconds]
%
figure(1)
plot(t,Ionosphere_delay_K_model)
xlabel('Apparent local time (seconds )')
ylabel('Delay due to Ionosphere (Seconds)')
title('Ionosphere delay using Klobuchar model ')
grid on

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Problem 3  %%%%%%%%%%%%%%%%%%%%%%

c = 299792458;  % speed of light in [m/sec]
f_L1 = 1575.42e06;
f_L2 = 1227.60e06;

% A defined on slide W8&9-36. A = 40.3*TEC
A = [(f_L1^2*f_L2^2)/(f_L2^2-f_L1^2)]*[L1_P_range - L2_P_range]; % [m/sec^2]

% Use A to get delay: See formula on slide W8&9-26:
Dual_freq_delay = [1/(c*f_L1^2)]*A;   % [sec]

figure(2)
plot(t,Dual_freq_delay)
xlabel('Apparent local time (Hours)')
ylabel('Delay due to Ionosphere (Seconds)')
title('Pseudorange estimate of delay at L1')
grid on

figure(3)
plot(t,Ionosphere_delay_K_model,'r', 'linewidth', 1.5)
hold on
plot(t,Dual_freq_delay,'o')
xlabel('Apparent local time (seconds )')
ylabel('Ionosphere delay (Seconds)')
grid on
legend('Klobuchar model', 'Pseudorange estimate at L1')


figure(4)
plot(t,Ionosphere_delay_K_model,'r', 'linewidth', 1.5)
hold on
plot(t,Dual_freq_delay)
xlabel('Apparent local time (seconds )')
ylabel('Ionosphere delay (Seconds)')
title('Ionosphere delay computed with Klobuchar model & Pseudorange measurements @ L1,
L2:')
grid on
legend('Klobuchar model', 'Pseudorange estimate at L1')
```

MATLAB code used to solve **problems 4 & 5**:

```
% AAE 575. HW 3. Rizwan Qureshi.
% PROBLEM 4 & 5:

clear all; clc; close all;

%%%%%%%%%%%% Problem 4 %%%%%%%%%%%%%%%%%%%

load DH010600.mat;

data_used_in_models = DH010600;

GPS_t1 = data_used_in_models(:,6);   % [secs]

P0 = data_used_in_models(:,7);   % surface atmospheric pressure [mbar]
T0_C = data_used_in_models(:,8);   % Surface atmospheric temperature. [Celcius]
T0_K = T0_C + 273.15;                %  Surface atmospheric temperature. [Kelvin]
RH = data_used_in_models(:,9)/100;     %Relative humidity converted into decimals

phi = 33.390;     % [Deg]

phi_array = [ones(1,144)*phi]';    % [Deg]

%Compute water vapor e0:
e0 = 6.108*RH.*exp((17.15*T0_K-4684)./(T0_K-38.5));    % [mbar]

%Compute Sasstamoinen Model DRY and WET Zeneith Delay:

Dry_delay_S_model = 0.0022777*(1+0.0026*cosd(2*phi_array)).*P0;     % [meters]
Wet_delay_S_model = 0.0022777*(1255./T0_K + 0.05).*e0;    %[ meters]

% Now compute Hopfield Model DRY and WET Zeneith Delays:

hd = 40136 + 148.72*(T0_K - 273.16);   % [meters]
hw = 11000;        % [meters]

Dry_delay_H_model = (77.64e-6*P0.*hd)./(5*T0_K);    % [meters]
Wet_delay_H_model = (0.373*e0*hw)./(5*(T0_K.^2));    % [meters]


%%%%%%%%%%%% Problem 5 %%%%%%%%%%%%%%%%%%%


%% simple flat-Earth mapping functions and TOTAL delay for S and H models:

El = [5 15 45 85]';

md = 1./(sind(El));
mw = 1./(sind(El));

n = 1;
while n<=4
Total_delay_Flat_Earth_S_model(:,n) = md(n)* Dry_delay_S_model + mw(n)*Wet_delay_S_model;
%[meters]
Total_delay_Flat_Earth_H_model(:,n) = md(n)* Dry_delay_H_model + mw(n)*Wet_delay_H_model;
%[meters]

n = n+1;
```

```
end

%% Mapping functions given in Misra & Enge & TOTAL delay for S & H models:

md_m = 1./sqrt(1-((cosd(El)./1.001)).^2);
mw_m = 1./sqrt(1-((cosd(El)./1.001)).^2);

n = 1;
while n<=4
Total_delay_Misra_S_model(:,n) = md_m(n)* Dry_delay_S_model + mw_m(n)*Wet_delay_S_model;
%[meters]
Total_delay_Misra_H_model(:,n) = md_m(n)* Dry_delay_H_model + mw_m(n)*Wet_delay_H_model;
%[meters]

n = n+1;
end


%% Mapping functions given by Chao & TOTAL delay for S & H models:

md_c = 1./[(sind(El) + (0.00143./(tand(El)+0.0445)))];
mw_c = 1./[(sind(El) +(0.00035./(tand(El)+0.017)))];

n = 1;
while n<=4
Total_delay_Chao_S_model(:,n) = md_c(n)* Dry_delay_S_model + mw_c(n)*Wet_delay_S_model;
%[meters]
Total_delay_Chao_H_model(:,n) = md_c(n)* Dry_delay_H_model + mw_c(n)*Wet_delay_H_model;
%[meters]


n = n+1;
end

%% Plots for Problem 4:

time = linspace(0,85800,144)';

figure(1)
plot(time,Wet_delay_S_model,'b')
hold on
plot(time,Wet_delay_H_model,'r')
xlabel('GPS time (seconds)')
ylabel('Wet zenith delay (meters)')
title('WET zenith tropospheric delay from Saastamoinen & Hopfield model')
legend('Saastamoinen model','Hopfield model')
grid on

figure(2)
plot(time,Dry_delay_S_model,'b')
hold on
plot(time,Dry_delay_H_model,'r')
xlabel('GPS time (seconds)')
ylabel('Dry zenith delay (meters)')
title('DRY zenith tropospheric delay from Saastamoinen & Hopfield model')
legend('Saastamoinen model','Hopfield model')
grid on


%% Plots for Problem 5:
```

```
figure(3)
plot(time,Total_delay_Flat_Earth_S_model(:,1),'b')
hold on
plot(time,Total_delay_Misra_S_model(:,1),'r')
plot(time,Total_delay_Chao_S_model(:,1),'k')
xlabel('GPS time (seconds)')
ylabel('Total tropospheric delay (meters)')
title('Total tropospheric delay using SAASTAMOINEN model @ 5 Degrees:')
grid on
legend('Flat earth mapping @ 5 Deg','Misra & Enge mapping @ 5 Deg','Chao mapping @ 5
Deg')


figure(4)
plot(time,Total_delay_Flat_Earth_H_model(:,1),'b')
hold on
plot(time,Total_delay_Misra_H_model(:,1),'r')
plot(time,Total_delay_Chao_H_model(:,1),'k')
xlabel('GPS time (seconds)')
ylabel('Total tropospheric delay (meters)')
title('Total tropospheric delay using HOPFIELD model @ 5 Degrees:')
grid on
legend('Flat earth mapping @ 5 Deg','Misra & Enge mapping @ 5 Deg','Chao mapping @ 5
Deg')


figure(5)
plot(time,Total_delay_Flat_Earth_S_model(:,2),'b')
hold on
plot(time,Total_delay_Misra_S_model(:,2),'r')
plot(time,Total_delay_Chao_S_model(:,2),'k')
xlabel('GPS time (seconds)')
ylabel('Total tropospheric delay (meters)')
title('Total tropospheric delay using SAASTAMOINEN model @ 15 Degrees:')
grid on
legend('Flat earth mapping @ 15 Deg','Misra & Enge mapping @ 15 Deg','Chao mapping @ 15
Deg')


figure(6)
plot(time,Total_delay_Flat_Earth_H_model(:,2),'b')
hold on
plot(time,Total_delay_Misra_H_model(:,2),'r')
plot(time,Total_delay_Chao_H_model(:,2),'k')
xlabel('GPS time (seconds)')
ylabel('Total tropospheric delay (meters)')
title('Total tropospheric delay using HOPFIELD model @ 15 Degrees:')
grid on
legend('Flat earth mapping @ 15 Deg','Misra & Enge mapping @ 15 Deg','Chao mapping @ 15
Deg')

figure(7)
plot(time,Total_delay_Flat_Earth_S_model(:,3),'b')
hold on
plot(time,Total_delay_Misra_S_model(:,3),'r')
plot(time,Total_delay_Chao_S_model(:,3),'k')
xlabel('GPS time (seconds)')
ylabel('Total tropospheric delay (meters)')
title('Total tropospheric delay using SAASTAMOINEN model @ 45 Degrees:')
grid on
```

```
legend('Flat earth mapping @ 45 Deg','Misra & Enge mapping @ 45 Deg','Chao mapping @ 45
Deg')

figure(8)
plot(time,Total_delay_Flat_Earth_H_model(:,3),'b')
hold on
plot(time,Total_delay_Misra_H_model(:,3),'r')
plot(time,Total_delay_Chao_H_model(:,3),'k')
xlabel('GPS time (seconds)')
ylabel('Total tropospheric delay (meters)')
title('Total tropospheric delay using HOPFIELD model @ 45 Degrees:')
grid on
legend('Flat earth mapping @ 45 Deg','Misra & Enge mapping @ 45 Deg','Chao mapping @ 45
Deg')


figure(9)
plot(time,Total_delay_Flat_Earth_S_model(:,4),'b')
hold on
plot(time,Total_delay_Misra_S_model(:,4),'r')
plot(time,Total_delay_Chao_S_model(:,4),'k')
xlabel('GPS time (seconds)')
ylabel('Total tropospheric delay (meters)')
title('Total tropospheric delay using SAASTAMOINEN model @ 85 Degrees:')
grid on
legend('Flat earth mapping @ 85 Deg','Misra & Enge mapping @ 85 Deg','Chao mapping @ 85
Deg')


figure(10)
plot(time,Total_delay_Flat_Earth_H_model(:,4),'b')
hold on
plot(time,Total_delay_Misra_H_model(:,4),'r')
plot(time,Total_delay_Chao_H_model(:,4),'k')
xlabel('GPS time (seconds)')
ylabel('Total tropospheric delay (meters)')
title('Total tropospheric delay using HOPFIELD model @ 85 Degrees:')
grid on
legend('Flat earth mapping @ 85 Deg','Misra & Enge mapping @ 85 Deg','Chao mapping @ 85
Deg')
```