

# Tutorial 2

## Measurements analysis and error budget

Professors Dr. Jaume Sanz Subirana, Dr. J. M. Juan Zornoza  
and Dr. Adrià Rovira Garcia

Research group of Astronomy & Geomatics (gAGE)  
Universitat Politècnica de Catalunya (UPC)  
Barcelona, Spain



# Aim of this tutorial

- ⬆ This tutorial is devoted to analysing and assessing the GNSS measurements and their associated errors.

This is done over four groups of test cases:

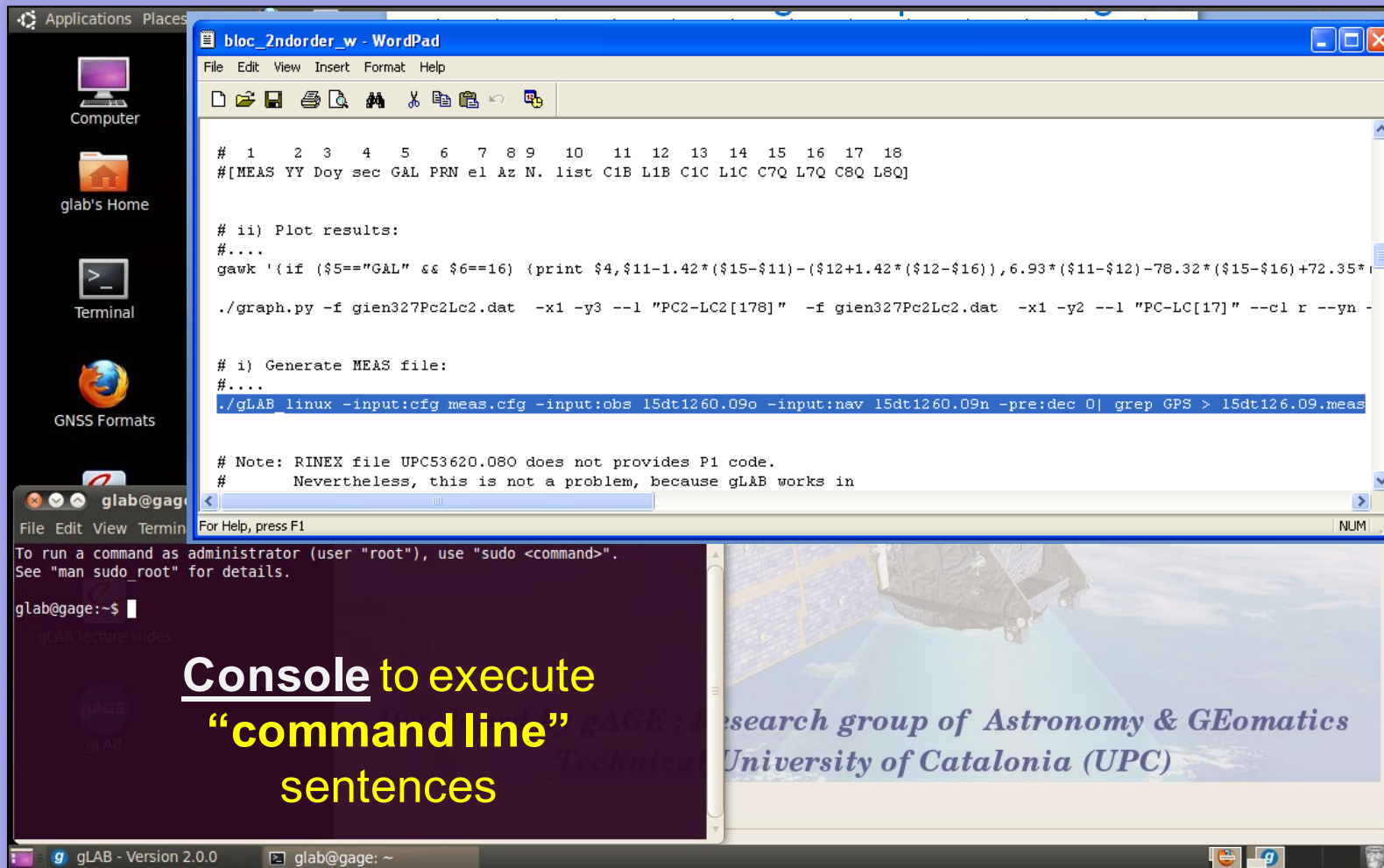
- 1.- Atmospheric propagation effects on GNSS signals. Analysis of the errors due to the troposphere and the ionosphere.
- 2.- Code measurement noise and multipath on C1, ionosphere-free and Melbourne-Wübbena combinations.
- 3.- Carrier smoothed code. Analysis of the code-carrier divergence of the ionosphere.
- 4.- Broadcast orbits and clocks accuracy versus the IGS precise products.

- ⬆ The laboratory exercises will be developed with actual GPS measurements, and processed with the **ESA/UPC GNSS-Lab Tool suite (gLAB)**, which is an interactive software package for GNSS data processing and analysis.
- ⬆ **All software tools** (including **gLAB**) and associated files for the laboratory session are included in the CD-ROM or USB stick associated with this tutorial.

# OVERVIEW

- **Introduction:** gLAB processing in command line.
- **Test cases on Atmospheric Propagation Effects:**  
Troposphere, Ionosphere.
- **Test cases on Measurement Noise and Multipath:** C1, Ionosphere-free and Melbourne Wübbena combinations.
- **Test cases on Carrier Smoothed Code:** Code-carrier divergence.
- **Test cases on Broadcast Orbits and Clocks Accuracy**

# gLAB processing in command line



```
# 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
#[MEAS YY Doy sec GAL PRN el Az N. list C1B L1B C1C L1C C7Q L7Q C8Q L8Q]

# ii) Plot results:
#....
gawk '{if ($5=="GAL" && $6==16) (print $4,$11-1.42*($15-$11)-($12+1.42*($12-$16)),6.93*($11-$12)-78.32*($15-$16)+72.35*($15-$16))}'

./graph.py -f gien327Pc2Lc2.dat -x1 -y3 --l "PC2-LC2[178]" -f gien327Pc2Lc2.dat -x1 -y2 --l "PC-LC[17]" --cl r --yn -

# i) Generate MEAS file:
#....
gLAB linux -input:cfg meas.cfg -input:obs 15dt1260.09o -input:nav 15dt1260.09n -pre:dec 0 | grep GPS > 15dt126.09.meas

# Note: RINEX file UPC53620.080 does not provides P1 code.
# Nevertheless, this is not a problem, because gLAB works in
```

glab@gage:~\$

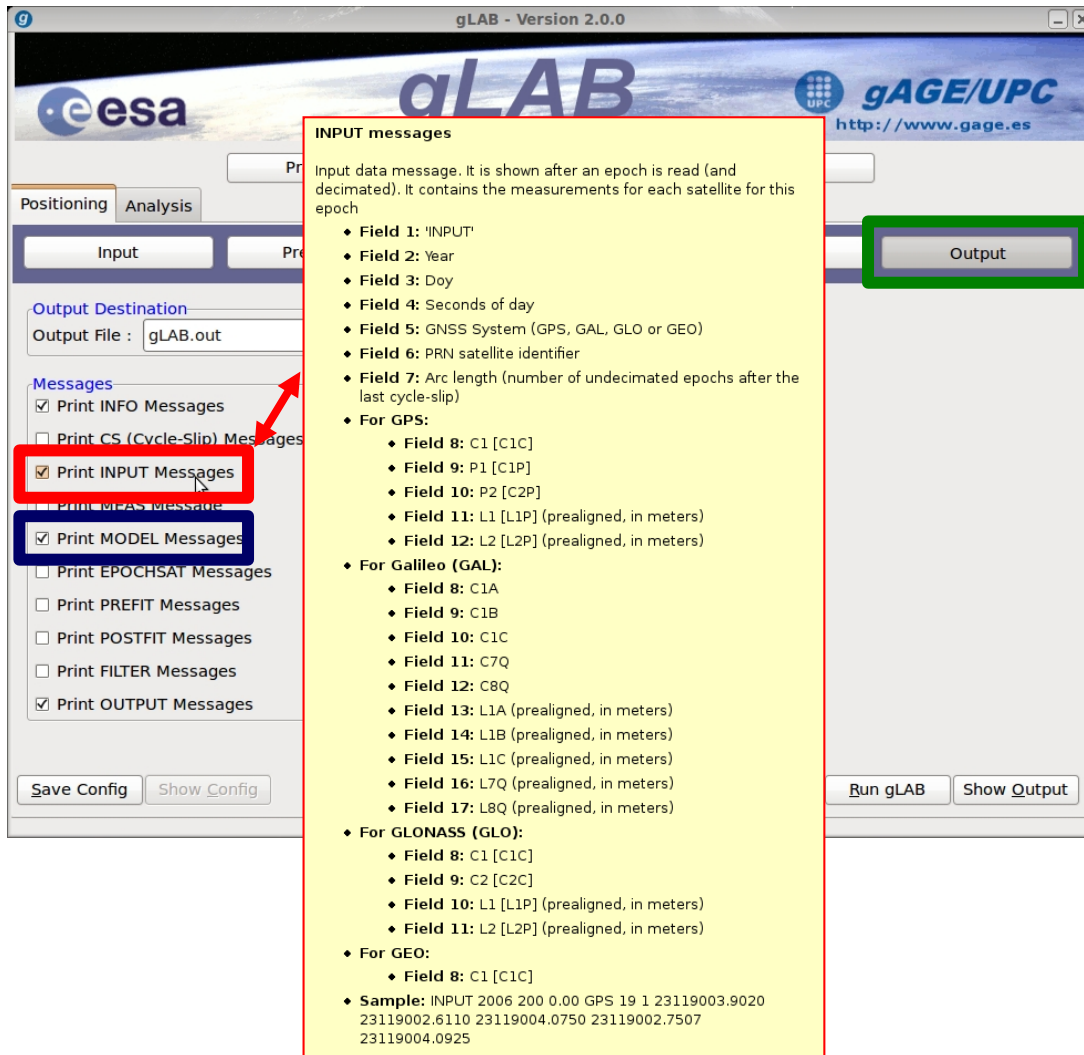
gLAB - Version 2.0.0

glab@gage: ~

gLAB research group of Astronomy & GEomatics  
Technical University of Catalonia (UPC)

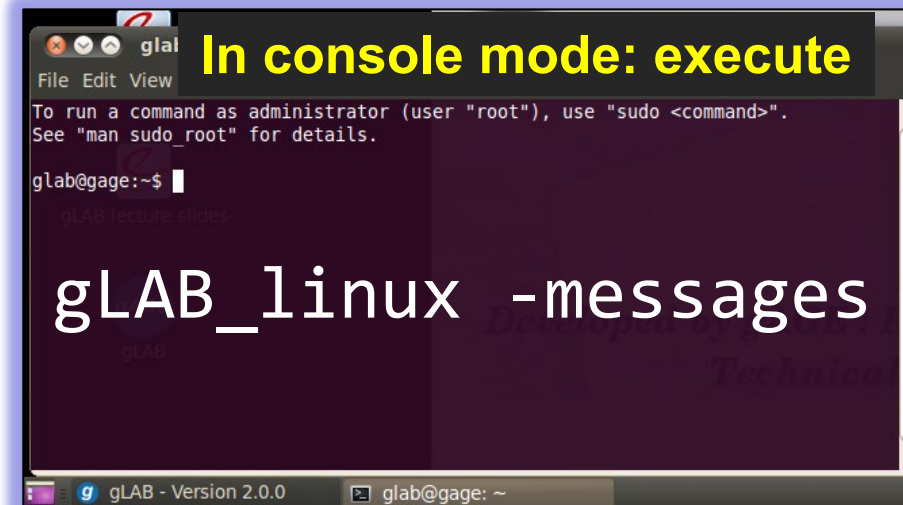
A “notepad” with the command line sentence is provided to facilitate the sentence writing: just “copy” and “paste” from notepad to the working terminal.

# gLAB processing in command line



The different messages provided by gLAB and its content can be found in the [OUTPUT] section.

By placing the mouse on a given message name, a tooltip appears describing the different fields.



# OVERVIEW

- ✦ **Introduction:** gLAB processing in command line.
- **Test cases on Atmospheric Propagation Effects:**  
Troposphere, Ionosphere.
- ✦ **Test cases on Measurement Noise and Multipath:** C1, Ionosphere-free and Melbourne Wübbena combinations.
- ✦ **Test cases on Carrier Smoothed Code:** Code-carrier divergence.
- ✦ **Test cases on Broadcast Orbits and Clocks Accuracy**

# Test cases on Atmospheric Propagation Effects: Troposphere, Ionosphere

# Troposphere

## Tropospheric delay

The troposphere is the atmospheric layer situated between the Earth's surface and an altitude of about 50 km.

The effect of the troposphere on GNSS signals appears as an extra delay in the measurement of the signal travelling from satellite to receiver.

The tropospheric delay does not depend on frequency and affects both the pseudo-range (code) and carrier phases in the same way. It can be modeled by:

- A **hydrostatic component**, composed of dry gases (mainly nitrogen and oxygen) in hydrostatic equilibrium. This component can be treated as an ideal gas. Its effects vary with the temperature and atmospheric pressure in a reasonably predictable manner, and it **is responsible for** about 90% of the delay.
- A **wet component** caused by the water vapor condensed in the form of clouds. It depends on the weather conditions and varies faster than the hydrostatic component and in a totally random way. **For high accuracy positioning, this component must be estimated together with the coordinates and other parameters in the navigation filter.**



# Exercise 1: Zenith Troposphere Delay estimation

**PPP Template:** Static positioning with dual freq. code & carrier (ionosphere-free combination PC,LC) + post-processed precise orbits & clocks.

The screenshot shows the gLAB - Version 2.0.0 interface. The 'Positioning' tab is selected, and the 'Input' sub-tab is highlighted with a yellow box. The 'Input Files' section contains fields for 'RINEX Observation File' (set to /home/gLAB/roap1810.09o), 'Show ANTEX' (checked), 'File' (set to /home/gLAB/igs05\_1525.atx), 'Orbit and Clock Source' (set to 'Precise (1 file)'), and 'SP3 File' (set to /home/gLAB/igs15382.sp3). Each of these fields has an 'Examine' button next to it. The 'Ionosphere Source (if activated)' section has a 'Show' checkbox. The 'Auxiliary Files' section has two 'Show' checkboxes for 'P1 - C1 Correction' and 'P1 - P2 Correction'. The 'Run gLAB' button is highlighted with a red box. The 'PPP Template' button is highlighted with a blue box. A large red number '3' is visible in the bottom right corner of the interface.

1. Select the **PPP Template**

2. **Upload data files:**

- Measurement : **roap1810.09o**
- ANTEX: **igs05\_1525.atx**
- Orbits & clocks: **igs15382.sp3**
- **SINEX**: **igs09P1538.snx**

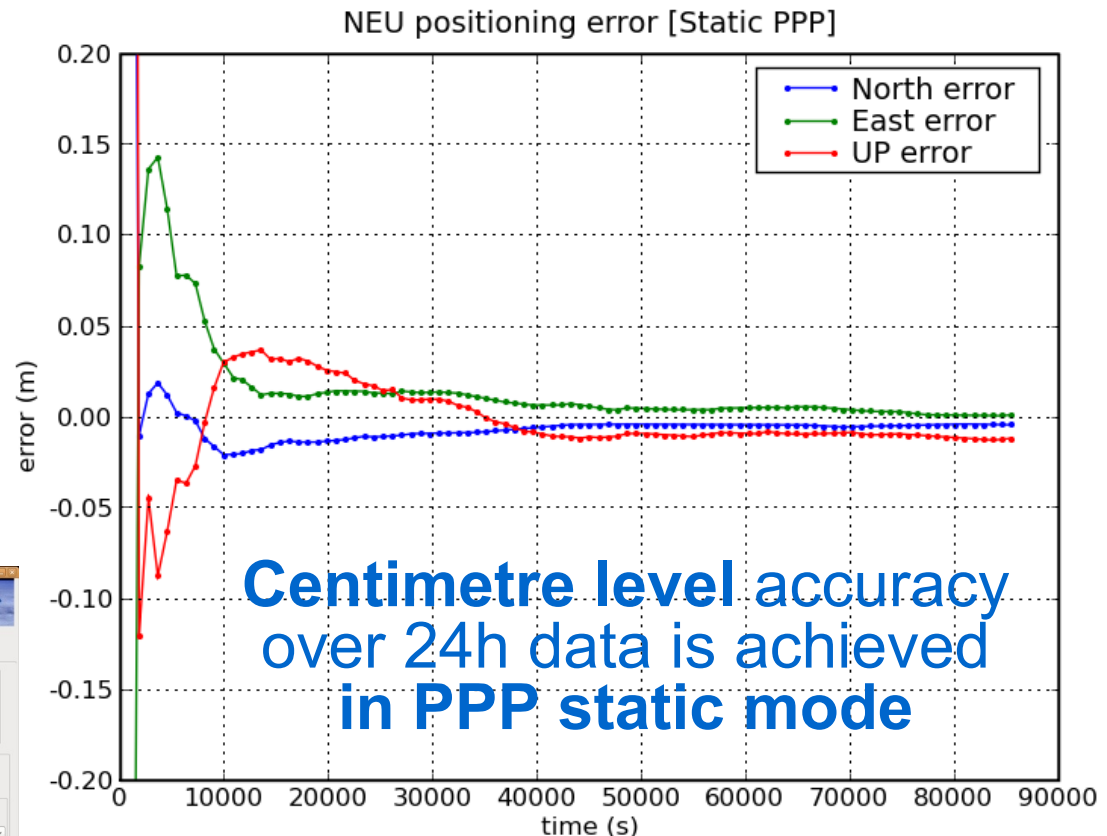
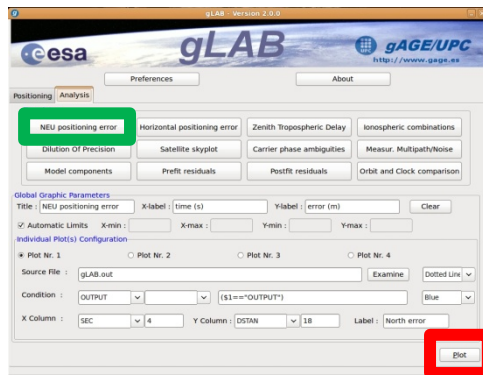
3. **RUN gLAB**

Default output file:  
**gLAB.out**

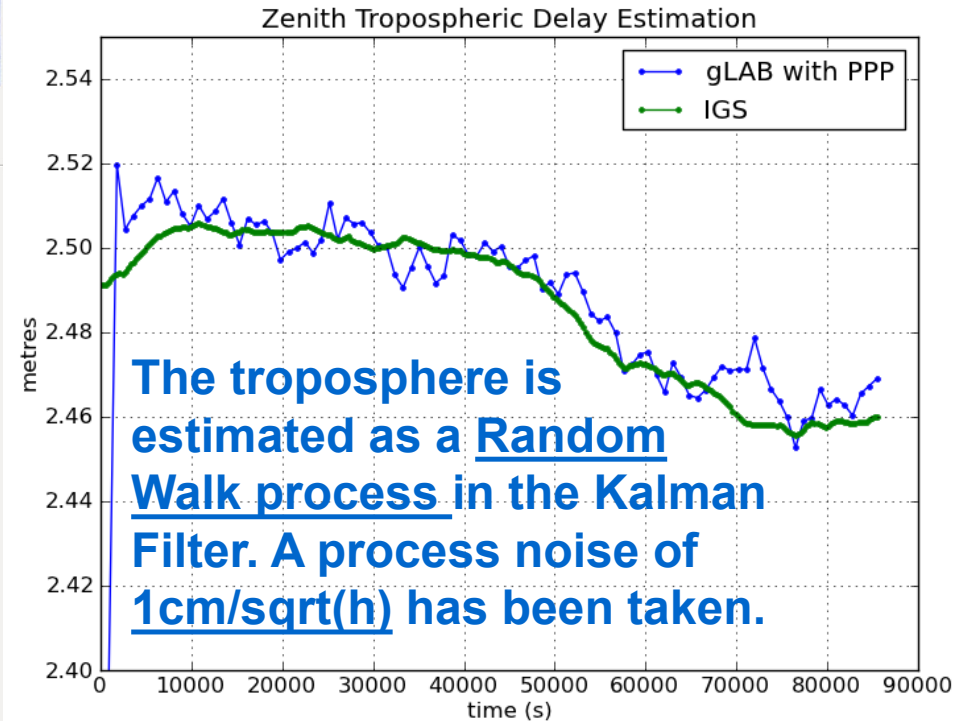
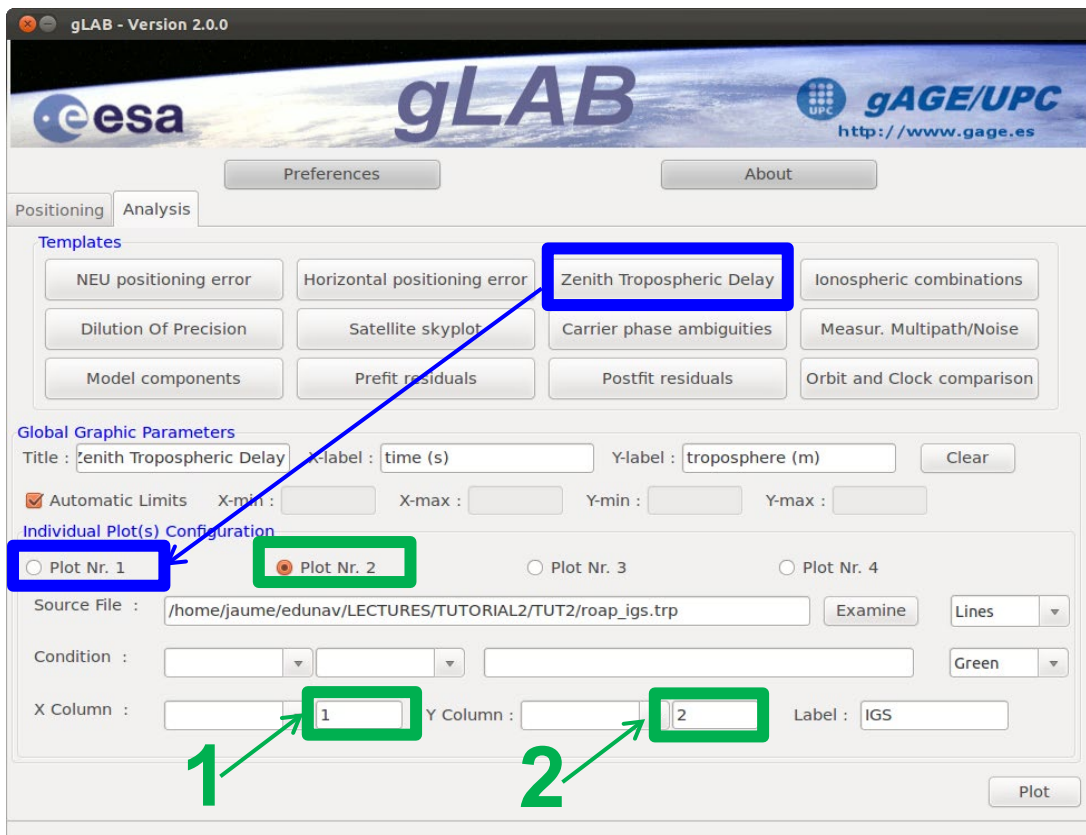
# Exercise 1: Zenith Troposphere Delay estimation

## Plotting Results

- Coordinates are taken as constants in nav. filter.
- Dual frequency Code and Carrier measurements.
- Precise orbits and clocks.
- Measurements modelling at the centimetre level.



# Exercise 1: Zenith Troposphere Delay estimation



<ftp://cddis.gsfc.nasa.gov/pub/gps/products/troposphere/new/2009/181/roap1810.09zpd.gz>

The ZTD in this file is given in mm of delay. Thus, it is converted to m to compare with gLAB results

```
grep ROAP roap1810.09zpd | gawk -F\: '{print $3}' | gawk '{print $1,$2/1000}' > roap_igs.trp
```

# Exercise 1: Zenith Troposphere Delay estimation

Equivalent command line sentences:

```
gLAB_linux -input:cfg gLAB_p51_ex6S.cfg
            -input:obs roap1810.09o
            -input:SP3 igs15382.sp3
            -input:ant igs05_1525.atx
            -input:snx igs09P1538.snx
```

```
graph.py -f gLAB.outK -x4 -y9 -s.- -c '($1=="FILTER")'
          -l "PPP Kinematic" -f gLAB.outS -x4 -y9 -s.-
          -c '($1=="FILTER")' -l "PPP Static"
          -f roap_igs.trp -s.- -l "IGS" --xl "time (s)"
          --yl "metres" --yn 2.40 --yx 2.55
          -t "Session 5.1, Ex6b: Zenith tropospheric delay Estimation"
```

# Exercise 1: Zenith Troposphere Delay estimation

## **Questions:**

- *What is the level of discrepancy between the estimated ZTD and the accurate IGS determination?*
- *What is the variation of ZTD in 24 hours due to the wet content?*
- *What is the proportion of wet component compared with the dry (or hydrostatic) component?*

# Ionosphere

## Ionospheric delay

The ionosphere extends from about  $60 \text{ km}$  over the Earth's surface up to more than  $2000 \text{ km}$ , with a sharp electron density maximum at around  $350 \text{ km}$ . The ionospheric refraction depends, among other things, on the location, local time and solar cycle (11 years).

- First order (~99.9%) ionospheric delay  $I_1$  depends on the inverse of squared frequency:

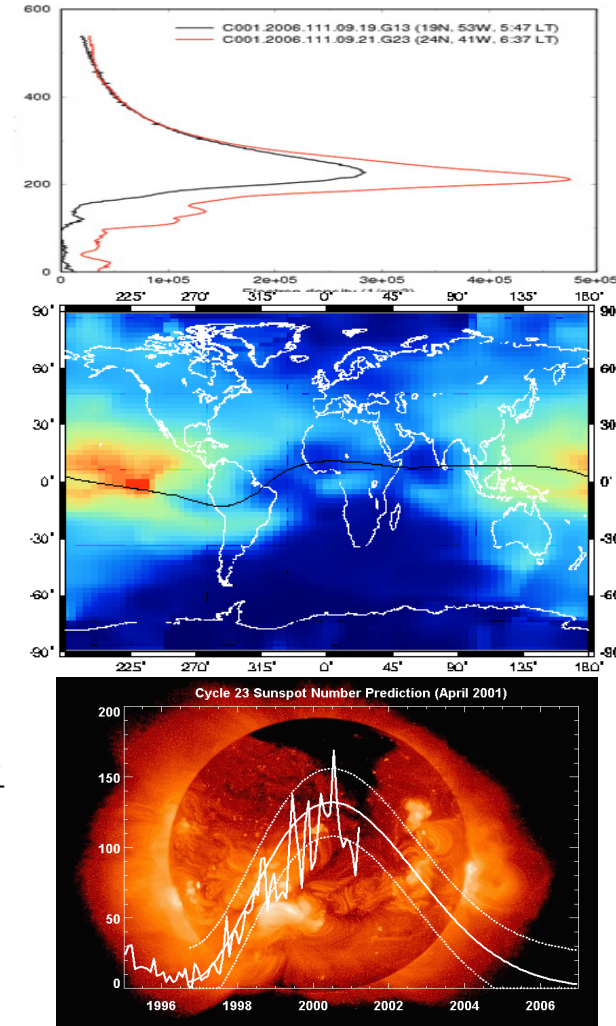
$$I_1 = \frac{40.3}{f^2} STEC$$

where  $STEC$  is the number of electrons per area unit along ray path (STEC: Slant Total Electron Content).

$$STEC = \int N_e ds$$

- Two-frequency receivers can remove this error source (up to 99.9%) using ionosphere-free combination of pseudo-ranges (PC) or carriers (LC).
- Single-frequency users can remove about 50-70% of the ionospheric delay using the Klobuchar model, whose parameters are broadcast in the GPS navigation message.

$$PC = \frac{f_1^2 P_1 - f_2^2 P_2}{f_1^2 - f_2^2}$$



# Exercise 2: Ionospheric delay analysis

## Depict the ionospheric delays for the different satellites in view from station amc2

- This is a simple exercise aimed to illustrate how to use gLAB to easily analyze GNSS measurements and their combinations.
- gLAB will be used to read the RINEX measurements file and to generate a “text” with the measurements provided in a columnar format (more suitable for making plots).
- From “text” file, compute and plot the ionospheric delay for a given satellite, by using code and carrier measurements at  $f_1$ ,  $f_2$ :

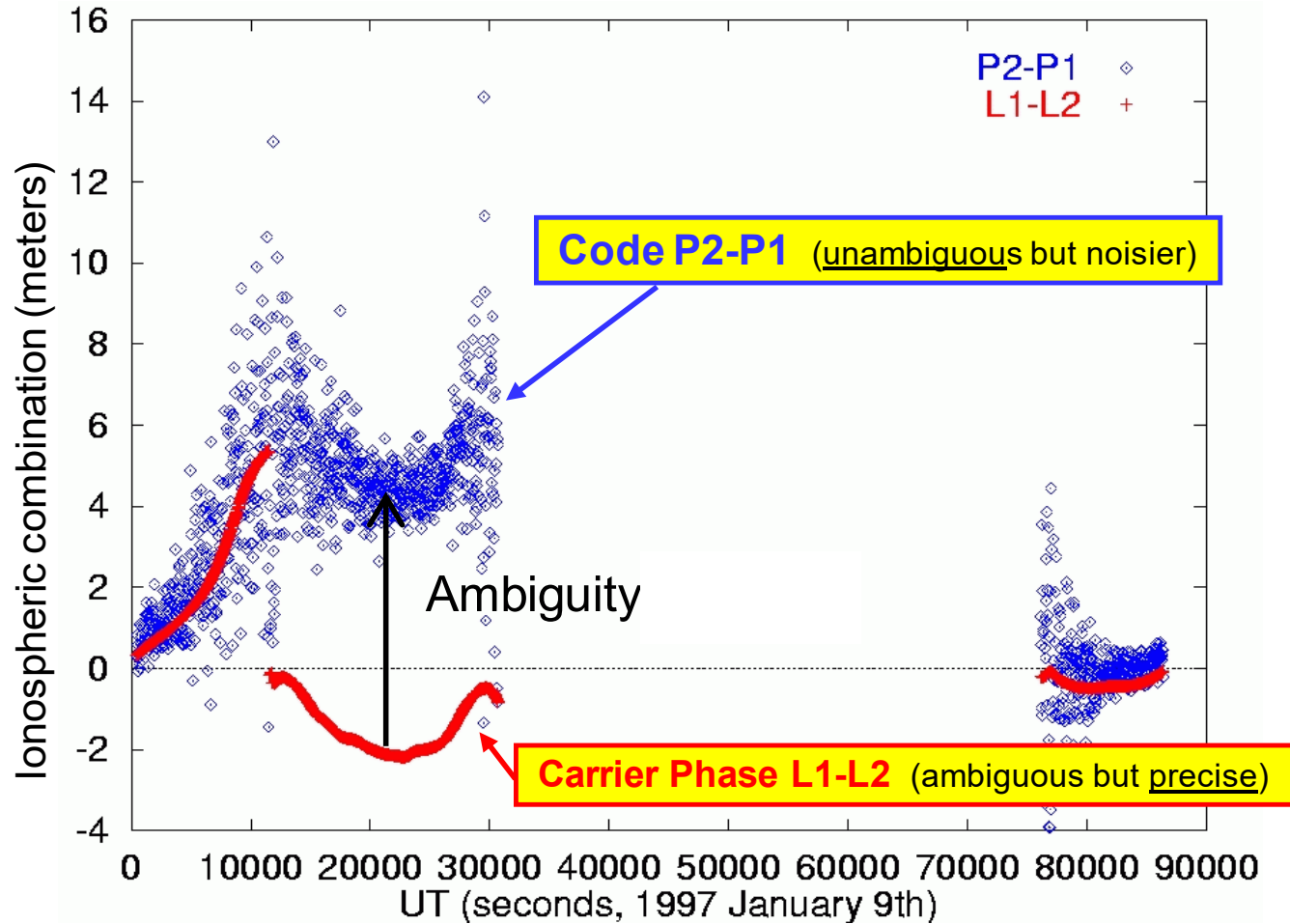
$$\begin{aligned} P_I &\equiv P_2 - P_1 = I + K_{21} \\ L_I &\equiv L_1 - L_2 = I + \text{Ambiguity} \end{aligned}$$

$$P_1 - L_1 = 2\bar{\alpha}_1 I + \text{ambiguity}_1$$

$$P_2 - L_2 = 2\bar{\alpha}_2 I + \text{ambiguity}_2$$



# Exercise 2: Ionospheric delay analysis



The target is to generate this plot to depict the ionospheric delay from code & carrier data

$$P_I \equiv P_2 - P_1 = I + K_{21}$$

$$L_I \equiv L_1 - L_2 = I + \text{Ambiguity}$$

$$P_1 - L_1 = 2\bar{\alpha}_1 I + \text{ambiguity1}$$

$$P_2 - L_2 = 2\bar{\alpha}_2 I + \text{ambiguity2}$$

$$\bar{\alpha}_1 = \frac{1}{\gamma_{21} - 1} = 1.546; \quad \bar{\alpha}_2 = 1 + \bar{\alpha}_1$$

$$\gamma_{21} = (f_1 / f_2)^2 = (154 / 120)^2$$



# Exercise 2: Ionospheric delay analysis

## Code measurement content

$$P_1 = \rho + \bar{\alpha}_1 (I + K_{21}) + \varepsilon_1$$

$$P_2 = \rho + \bar{\alpha}_2 (I + K_{21}) + \varepsilon_2$$

## Carrier measurement content

$$L_1 = \rho - \bar{\alpha}_1 I + B_1 + \zeta_1$$

$$L_2 = \rho - \bar{\alpha}_2 I + B_2 + \zeta_2$$

$$\bar{\alpha}_2 - \bar{\alpha}_1 = 1$$

$$\bar{\alpha}_1 = \frac{1}{\gamma_{21} - 1} = 1.546$$

$$\gamma_{21} = (f_1 / f_2)^2$$

$\rho$  Refers to all **non-dispersive terms**: geometric range, clocks, tropo. delay... (see [R-1]).

## Ionospheric delay

$$I = \frac{40.3(f_1^2 - f_2^2)}{f_1^2 f_2^2} 10^{16} STEC; (I \text{ is in } m \text{ of } L_1 - L_2 \text{ delay})$$

$$STEC = \int_{rec}^{sat} N_e dl, \quad (STEC \text{ is in } TECUs)$$

$$1 \text{ TECU} = 10^{16} e^- / m^2 = 0.10m \text{ of } L1 - L2 \text{ delay}$$

## Inter-frequency bias

$$K_{21} = K_{21,rec} - K_{21}^{sat}$$

As the satellite clocks are referred to the ionosphere-free combination of codes ( $P_C$ ), the  $K_{21}^{sat}$  cancels in such combination.

Note:  $TGD = -\bar{\alpha}_1 K_{21}^{sat}$  is broadcast in GPS nav. Message.

$$P_C = \frac{f_1^2 P_1 - f_2^2 P_2}{f_1^2 - f_2^2}$$

## Carrier ambiguities

$$B_i = \lambda_i N_i + b_i \quad N_i \text{ is an integer number. } b_i \text{ is a real number (fractional part of ambiguity)}$$

# Exercise 2: Ionospheric delay analysis

1.- Read RINEX file with gLAB and generate a “measurements file” in a columnar format (the easiest to manipulate and plot content):

→ Using the configuration file meas.cfg, READ the RINEX and generate the MEAS file:

```
gLAB_linux -input:cfg meas.cfg -input:obs coco0090.97o -input:nav brdc0090.97n > coco.meas
```

gLAB configuration file

```
-pre:dec 1
-print:none
-print:meas

--model:satphasecenter
--model:recphasecenter
--model:satclocks
--pre:cs:li
--pre:cs:bw
```

RINEX  
Measurement  
file

RINEX  
Navigation  
file

OUTPUT: measurement file in columnar format

[Id	YY	Doy	sec	GPS	PRN	e1	Az	N.	list	C1C	L1C	C1P	L1P	C2P	L2P]
1	2	3	4	5	6	7	8	9	10	11	xx	xx	14	15	16]

# Exercise 2: Ionospheric delay analysis

gLAB\_linux -input:cfg **meas.cfg** -input:obs **coco0090.97o** -input:nav **brdc0090.97n** > **coco.meas**

**-pre:dec 1**  
**-print:none**  
**-print:meas**

**--model:satphasecenter**  
**--model:recphasecenter**  
**--model:satclocks**  
**--pre:cs:li**  
**--pre:cs:bw**

**Input Files:**  
**coco0090.97o**  
**brdc0090.97n**

**Set default SPP config.**

**Set data decimation to 1s**  
**-pre:dec 1**

**Disable cycle-slip detectors:**  
**--pre:cs:li**  
**--pre:cs:bw**

**Disable:**  
**--model:satphasecenter**  
**--model:recphasecenter**  
**--model:satclocks**

**MEAS file**

**MEAS messages**  
 It provides the MEASurement values. It is shown after an epoch is read (and decimated). It contains the measurements for each satellite for this epoch.

- Field 1: 'MEAS'
- Field 2: Year
- Field 3: Day
- Field 4: Seconds of day
- Field 5: GNSS System (GPS, GAL, GLO or GEO)
- Field 6: PRN satellite identifier
- Field 7: Elevation of the satellite [degrees]
- Field 8: Azimuth of the satellite [degrees]
- Field 9: Number of Measurement(s)
- Field 10: Measurement identifier (as string)
- Field 11: Measurement(s) value [m]

**Sample:** MEAS 2010 081 300.00 GPS 30 30.00 240.00 6 C1C1L1C:C1P:L1P:C2P:L2P 20228715.3270 0.0000 0.0000 20228715.2722 20228714.8230 20228714.7005

# Exercise 2: Ionospheric delay analysis

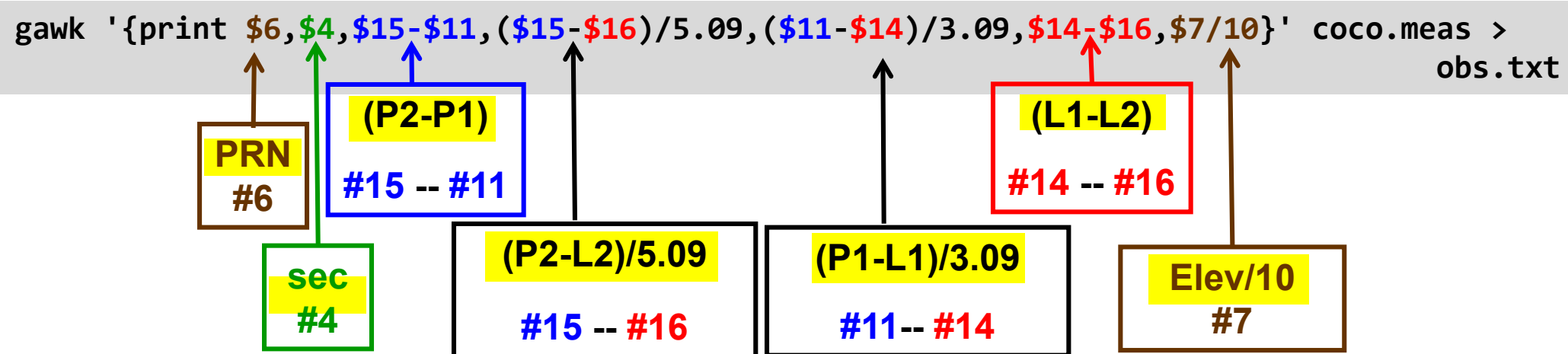
2.- Manipulate the file with the easy & powerful awk (or gawk) programming language (to compute the combinations of measurements):

→ From `coco.meas` file:

[Id	YY	Doy	sec	GPS	PRN	e1	Az	N.	list	P1	L1	P2	L2		
1	2	3	4	5	6	7	8	9	10	11	xx	xx	14	15	16

Compute different ionospheric combination of codes and carriers, and generate the `obs.txt` file containing the fields:

[PRN,sec, P2-P1, (P2-L2)/5.09, (P1-L1)/3.09, L1-L2, Elev/10]



# Exercise 2: Ionospheric delay analysis

## 3.-Plot results with “graph.py” (you can use the gnuplot as well)

→ From **obs.txt** file:

[PRN, sec, P2-P1, (P2-L2)/5.09, (P1-L1)/3.09, L1-L2, Elev/10]
1 2 3 4 5 6 7

Show in the same plot the following iono. delays for satellite **PRN01**:

P2-P1, (P2-L2)/5.09, (P1-L1)/3.09, L1-L2, Elev./10

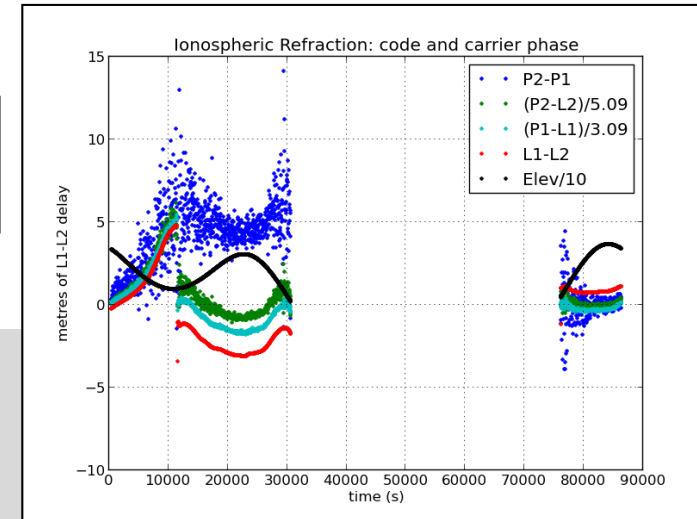
**Condition:**  
Select **PRN01**  
(from 1-st field)

**Fields to plot:**  
#2 (x-axis)  
versus  
#3 (y-axis)

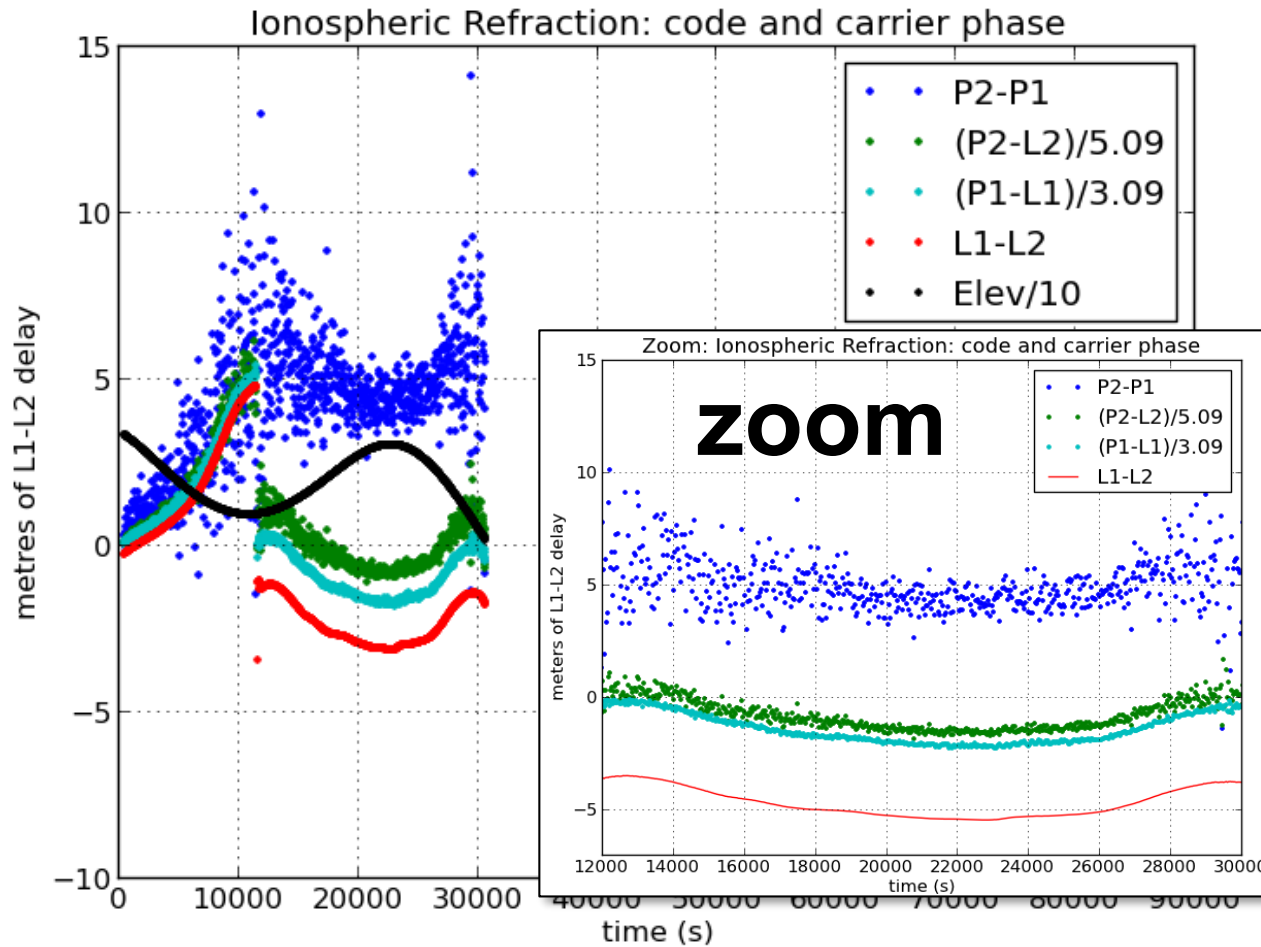
**Label:**  
P2-P1

**File to plot**

```
graph.py -f obs.txt -c '($1==01)' -x2 -y3 -l "P2-P1"  
-f obs.txt -c '($1==01)' -x2 -y4 -l "(P2-L2)/5.09"  
-f obs.txt -c '($1==01)' -x2 -y5 -l "(P1-L1)/3.09"  
-f obs.txt -c '($1==01)' -x2 -y6 -l "L1-L2"  
-f obs.txt -c '($1==01)' -x2 -y7 -l "Elev/10"  
--yn -10 --yx 15 --xl "time (s)" --yl "meters of L1-L2 delay"
```



# Exercise 2: Ionospheric delay analysis



$$P_I \equiv P_2 - P_1 = I + K_{21}$$

$$L_I \equiv L_1 - L_2 = I + \text{Ambiguity}$$

$$P_1 - L_1 = 2\bar{\alpha}_1 I + \text{ambiguity1}$$

$$P_2 - L_2 = 2\bar{\alpha}_2 I + \text{ambiguity2}$$

$$\bar{\alpha}_1 = \frac{1}{\gamma_{21} - 1} = 1.546; \quad \bar{\alpha}_2 = 1 + \bar{\alpha}_1$$

$$\gamma_{21} = (f_1 / f_2)^2 = (154 / 120)^2$$

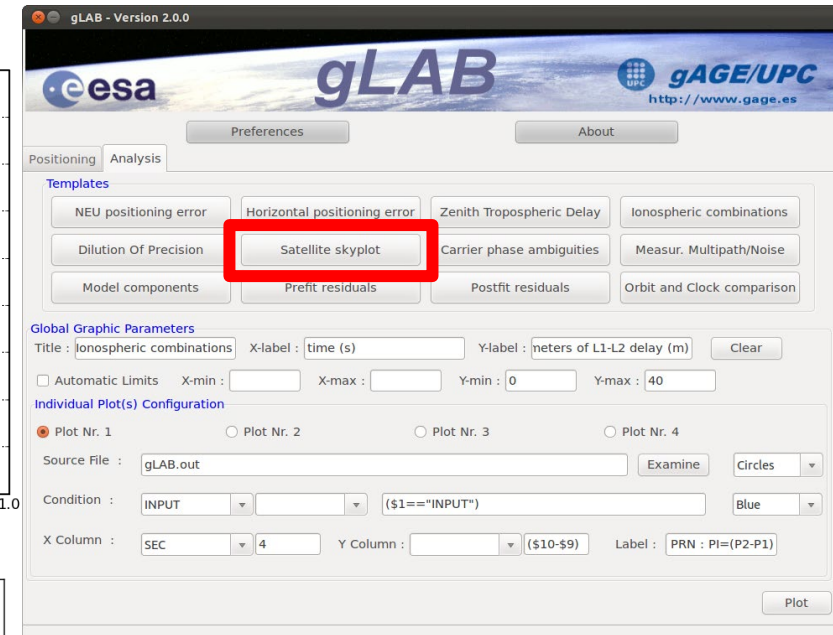
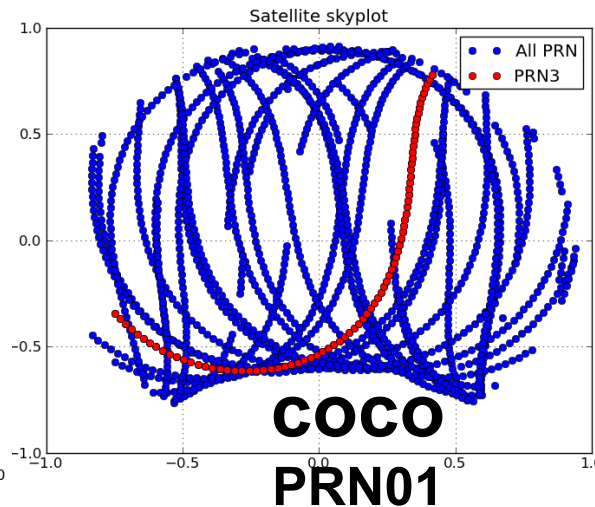
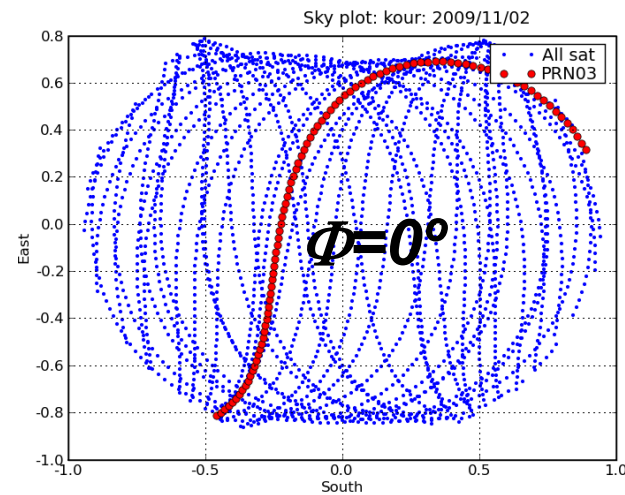
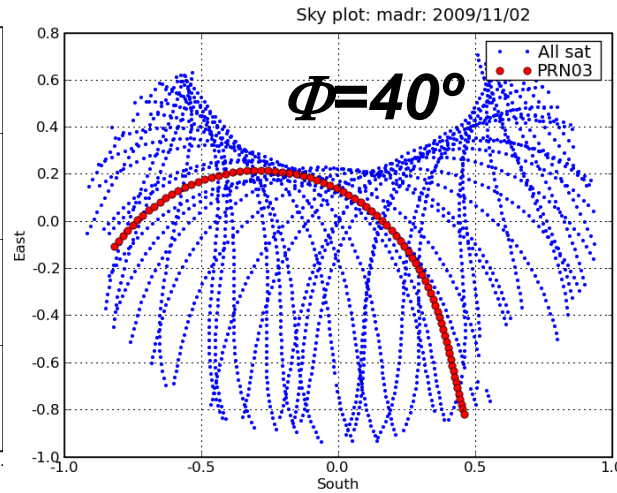
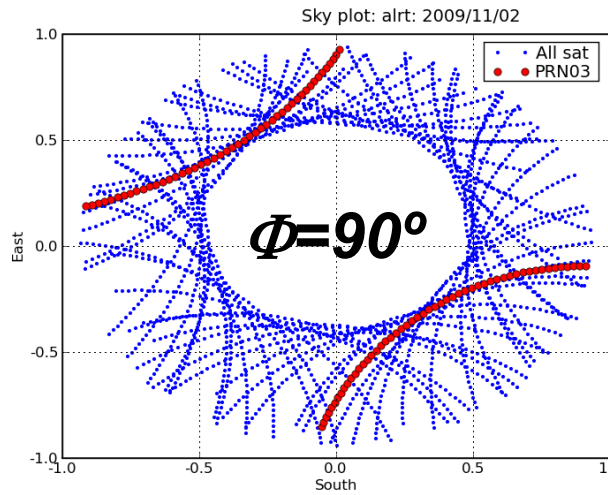
# Exercise 2: Ionospheric delay analysis

## **Questions:**

- *Justify the expressions used to depict the Ionospheric delay (see slides #16 to #18).*
- *Justify the factors of 5.09 and 3.09 used in the P2-L2 and P1-L1 combinations to give the results of L1-L2 delay in metres.*
- *Why is the STEC larger at low elevations?*



# Exercise 2: Ionospheric delay analysis



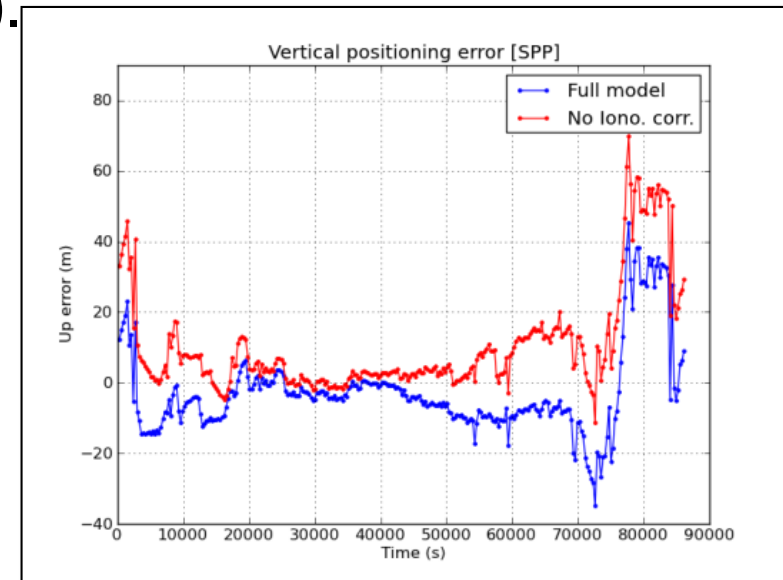
Sky plots at  
different  
latitudes



# Ex. 3. Halloween storm: October 2003

A severe ionospheric storm was experienced on October 29-31, 2003 producing an increase of the electron density which led to large ionospheric refraction values on the GPS signals. Such conditions were beyond the capability of the GPS Klobuchar model broadcast for single frequency users, producing large errors in the SPS (see details in [R-3]).

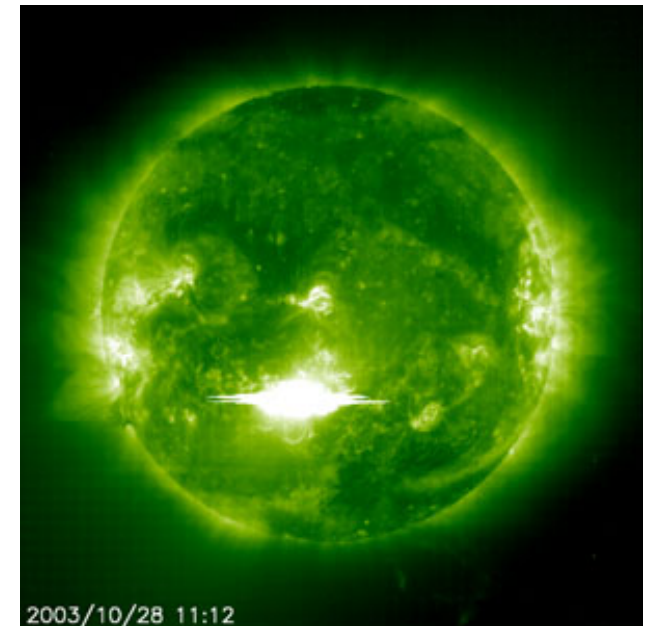
Dual frequency users, navigating with the ionospheric-free combination of GPS signals were not affected by such ionospheric errors, as the ionospheric refraction can be removed up to 99.9% using dual-frequency signals.



# Ex. 3.1: Solar Flare October 28, 2003

Before the storm, on October 28, 2003, an intense solar eruption (a Solar Flare) was detected around 11h UT in an active region which had grown one of the largest sunspots ever seen by the SOLar Helioscopic Observatory (SOHO). It appeared as a bright spike in the SOHO ultraviolet images.

This sudden enhancement of the solar radiation in the X-ray and extreme ultra-violet band produced a sudden increase in the ionospheric electron density on the daylight hemisphere, (see [R-3]).



# Ex. 3.1: Solar Flare October 28, 2003

**Exercise:** Analyze the effect of the Solar Flare on the Slant Total Electron Content (STEC) measurements of four permanent IGS receivers ankr, asc1, kour and qaq1, covering a wide range of longitude and latitude.

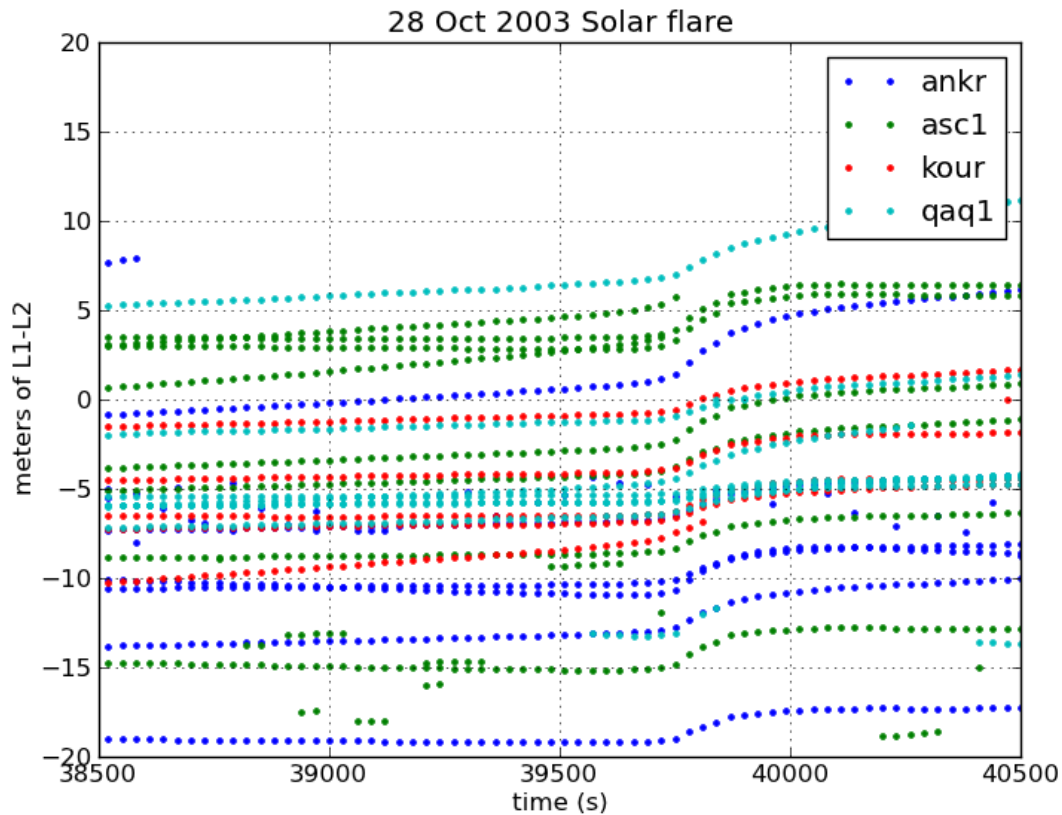
Data sets:

ankr3010.03o, asc13010.03o,  
kour3010.03o, qaq13010.03o





# Ex. 3.1: Solar Flare October 28, 2003





# Ex. 3.1: Solar Flare October 28, 2003

## **Questions:**

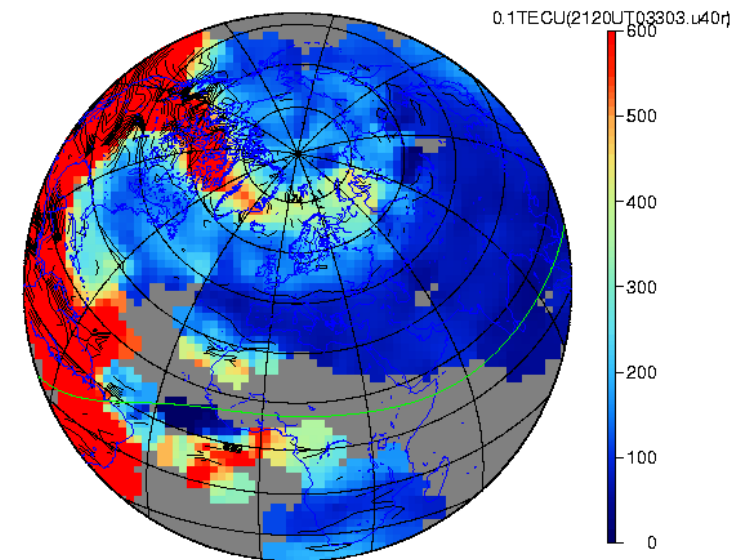
- *Why can we associate the sudden increase of TEC seen in the plots with a Solar Flare?*
- *What is the STEC variation rate due to the Solar Flare?*
- *Can all solar flares be seen in this way?*

# Ex. 3.2: Halloween storm analysis

Associated with the Solar Flare analysed in the previous exercise, a Coronal Mass Ejection occurred, which sent a large particle cloud impacting the Earth's magnetosphere about 19 hours later, on October 29.

Subsequent impacts were still occurring several hours later. This material interacted with the Earth's magnetosphere, and a Storm Enhancement Density (SED) appeared in North America and **this later** affected the northern latitudes in Europe. Extra large gradients of TEC associated with this phenomenon were also produced, degrading the GPS positioning performance.

The TEC evolution on October 30, 2003 (i.e., Day 303 of year 2003) can be seen in the movie TEC 2003oct30 anim.gif.



# Ex. 3.2: Halloween storm analysis

The measurement files `garl3010.03o`, `garl3020.03o`, `garl3030.03o`, `garl3040.03o`, `garl3050.03o`, `garl3060.03o` were collected by the permanent receiver garl in Empire, Nevada, USA (  $\phi = 40.42$  deg,  $\lambda = -119:36$  deg) from October 28 to November 2, 2003.

Using these files, plot the STEC for all satellites in view and discuss the range of such variations. Analyse, in particular, the satellite PRN 04 and calculate the maximum rate of STEC variation in mm=s of L1 delay. Add the elevation of satellite PRN 04 in the plot.

The associated broadcast navigation files are `brdc3010.03n`, `brdc3020.03n`, `brdc3030.03n`, `brdc3040.03n`, `brdc3050.03n`, `brdc3060.03n`.



# Ex. 3.2: Halloween storm analysis

The next commands read a RINEX file and generate a text file (in columnar format) that enables easy plotting of the measurements and their combinations:

1. Using the configuration file `meas.cfg`, READ the RINEX and generate the MEAS message with data format:

[Id	YY	Doy	sec	GPS	PRN	e1	Az	N.	list	C1C	L1C	C1P	L1P	C2P	L2P]
1	2	3	4	5	6	7	8	9	10	11	xx	13	14	15	16]

Execute:

```
gLAB_linux -input:cfg meas.cfg -input:obs amc23030.03o -input:nav brdc3030.03n > amc23030.03.meas
```

2. From `meas.txt` file,

Compute the ionospheric combination of codes:  $PI=P2-P1$ .

Generate the file `PI.txt` with the following content: [PRN, hour, PI, elevation]

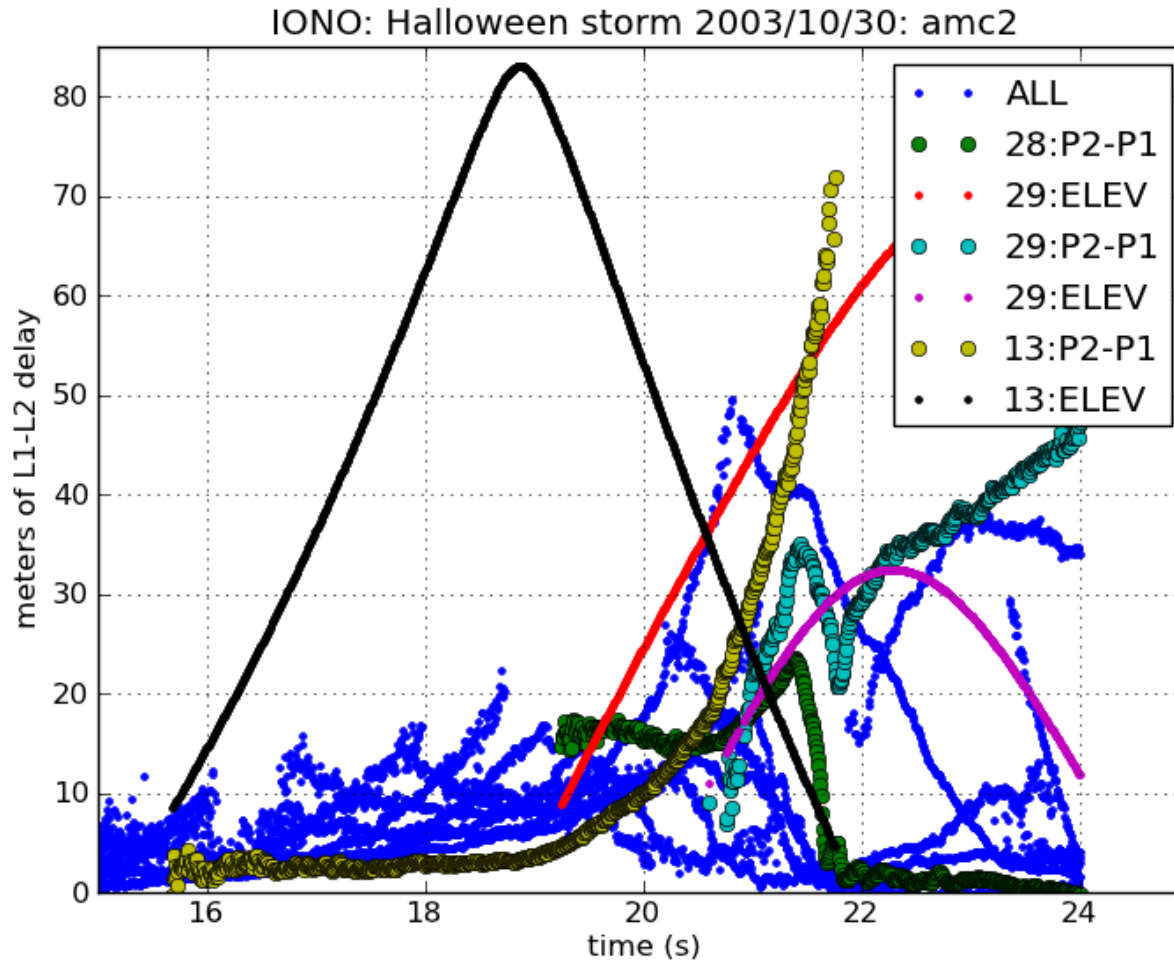
```
gawk '{print $6, $4/3600, $15-$13, $7}' amc23030.03.meas > PI.txt
```

3. From `PI.txt` file,

Plot the  $PI=P2-P1$  for time interval [15 to 24].hours. Show in the same graph: 1) ALL satellites, 2) PRN 13, 28 and 29, and 3) The elevation of each satellite.(13, 28 and 29)

```
graph.py -f PI.txt -x2 -y3 -l "ALL"  
-f PI.txt -c'($1==28)' -x2 -y3 -so -l "28:P2-P1" -f PI.txt -c'($1==28)' -x2 -y4 -l "28:ELEV"  
-f PI.txt -c'($1==29)' -x2 -y3 -so -l "29:P2-P1" -f PI.txt -c'($1==29)' -x2 -y4 -l "29:ELEV"  
-f PI.txt -c'($1==13)' -x2 -y3 -so -l "13:P2-P1" -f PI.txt -c'($1==13)' -x2 -y4 -l "13:ELEV"  
--xn 15 --xx 25 --yn 0 --yx 85 --xl "time (s)" --yl "meters of L1-L2 delay"
```

# Ex. 3.2: Halloween storm analysis



## Questions:

- What is the maximum STEC variation rate seen in the plots?
- How could we explain the shape of the STECs seen for satellites PRN28 and PRN29?



# Ex. 3.3: Halloween storm evolution

## Exercise: Analyze the ionospheric delays for 6 consecutive days including the Halloween storm

- This is a simple exercise aimed to illustrate the ionospheric **delay** variation during the Halloween storm. A period of 6 consecutive days (from October 28 to November 2, 2003) are analyzed using measurements collected in the “gar1” station in North America.
- The STEC variations are depicted from the geometry-free combination of codes  $P2-P1$ .

$$\text{Note: } P_2 - P_1 = I + K_{21}$$

# Ex. 3.3: Halloween storm evolution

[Id	YY	Doy	sec	GPS	PRN	e1	Az	N.	list	C1C	L1C	C1P	L1P	C2P	L2P]
1	2	3	4	5	6	7	8	9	10	11	xx	13	14	15	16]

## 1.- Read RINEX file:

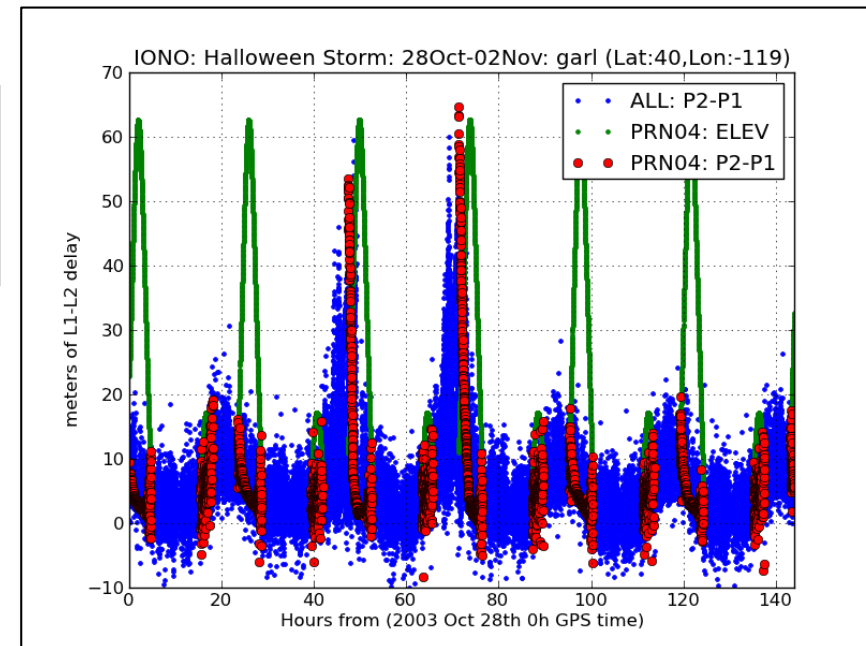
```
gLAB_linux -input:cfg meas.cfg -input:obs gar13010.03o -input:nav brdc3010.03n > gar13010.03.meas
gLAB_linux -input:cfg meas.cfg -input:obs gar13020.03o -input:nav brdc3020.03n > gar13020.03.meas
gLAB_linux -input:cfg meas.cfg -input:obs gar13030.03o -input:nav brdc3030.03n > gar13030.03.meas
gLAB_linux -input:cfg meas.cfg -input:obs gar13040.03o -input:nav brdc3040.03n > gar13040.03.meas
gLAB_linux -input:cfg meas.cfg -input:obs gar13050.03o -input:nav brdc3050.03n > gar13050.03.meas
gLAB_linux -input:cfg meas.cfg -input:obs gar13060.03o -input:nav brdc3060.03n > gar13060.03.meas
```

## 2.- Merge files and refer all the data to 0h of October 28th: Doy0301:

```
cat gar130?0.03.meas |gawk '{d=($3-301)*86400;$4=$4+d; print $6, $4/3600, $15-$13, $7}' > PI.txt
```

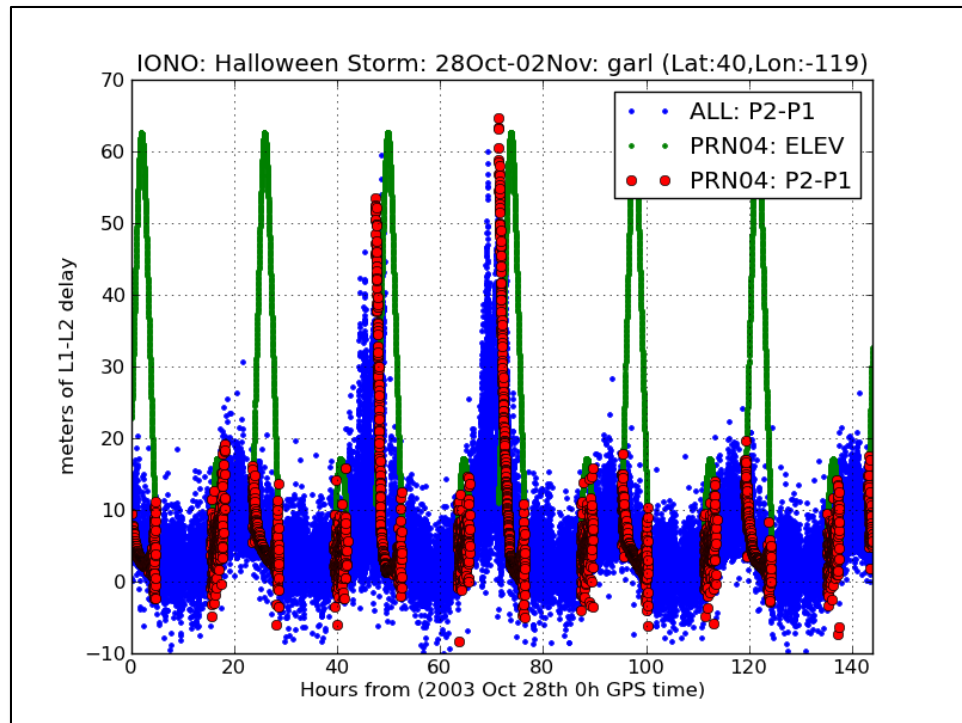
## 3.- Plot results:

```
graph.py -f PI.txt -x2 -y3 -l "ALL P2-P1"
-f PI.txt -c'($1==04)' -x2 -y4 -l "PRN04: ELEV"
-f PI.txt -c'($1==04)' -x2 -y3 -so -l "PRN04: P2-P1"
--xn 0 --xx 144
```



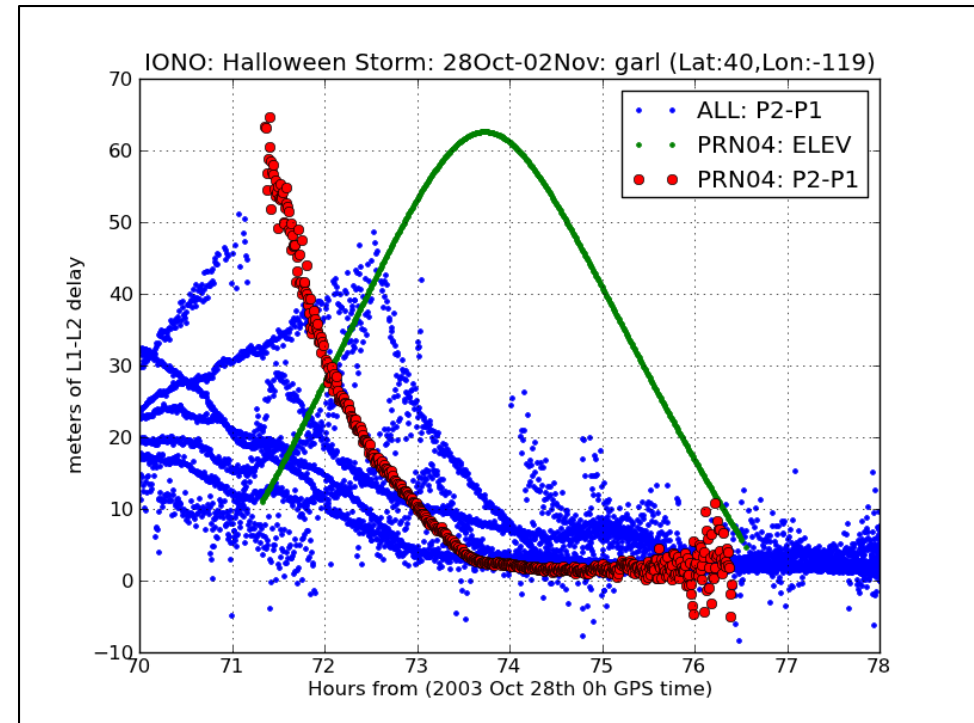
# Ex. 3.3: Halloween storm evolution

```
graph.py -f PI.txt -x2 -y3 -l "ALL P2-P1"  
-f PI.txt -c'($1==04)' -x2 -y4 -l "04: EL"  
-f PI.txt -c'($1==04)' -x2 -y3 -so -l "04"  
--xn 0 --xx 144
```



Zoom at time interval: **70 to 78 h**

```
graph.py -f PI.txt -x2 -y3 -l "ALL P2-P1"  
-f PI.txt -c'($1==04)' -x2 -y4 -l "04: EL"  
-f PI.txt -c'($1==04)' -x2 -y3 -so -l "04"  
--xn 70 --xx 78
```



# Ex. 3.3: Halloween storm evolution

## **Questions:**

- *What is the maximum STEC seen before and after the storm? And during the storm?*
- *The STEC of satellite PRN04 shows a flat pattern after time 74 hours which does not change with the elevation. Try to explain this phenomenon.*

# Ex. 3.4: Single frequency pos. effects

**Exercise:** Analyze the single frequency positioning solution under the Halloween storm.

The following steps are recommended:

1. Using files **amc23020.03o**, **brdc3030.03n** compute with gLAB the following solutions:
  1. Solution with full SPS modeling. Name output file as: **gLAB.out**
  2. Solution with the ionospheric corrections disabled → **gLAB1.out**
  3. Solution with the 2-freq. Ionosphere-free code (PC) → **gLAB2.out**
2. Plot results

**Note:** The **gLAB GUI** or the **command line sentences** can also be used .

A “notepad” with the command line sentence is provided to facilitate the sentence writing: just “copy” and “paste” from notepad to the working terminal



# Ex. 3.4: Full processing → gLAB.out

1. Compute SPP using files: amc23020.03o, brdc3030.03n



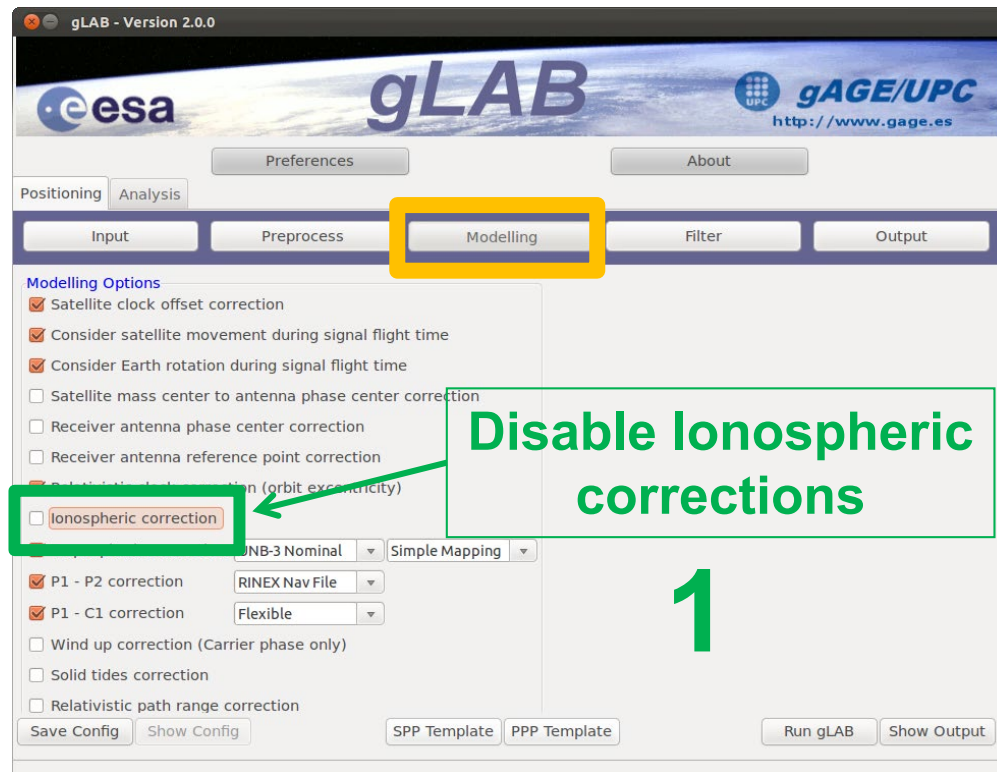
Equivalent command line sentence:

```
gLAB_linux -input:cfg gLAB_p1_Full.cfg  
-input:obs amc3030.03o  
-input:nav brdc3030.03n
```



# Ex. 3.4: Iono. Disabled → gLAB1.out

## 2. Reprocess the same files, with the iono. corrections disabled

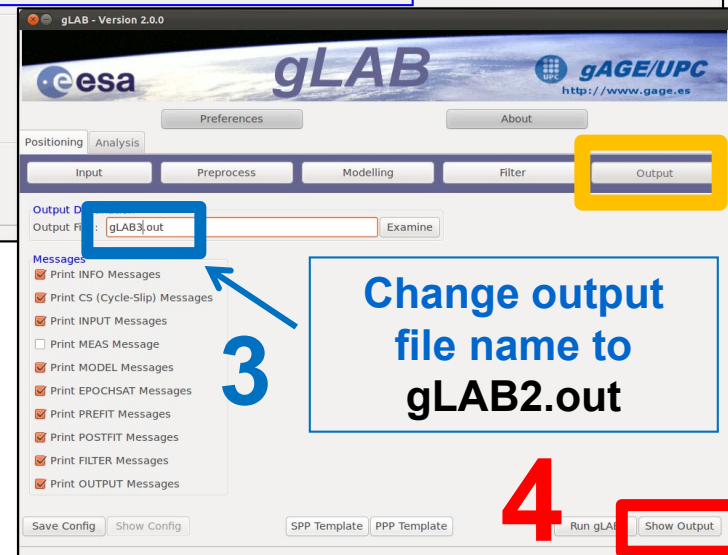
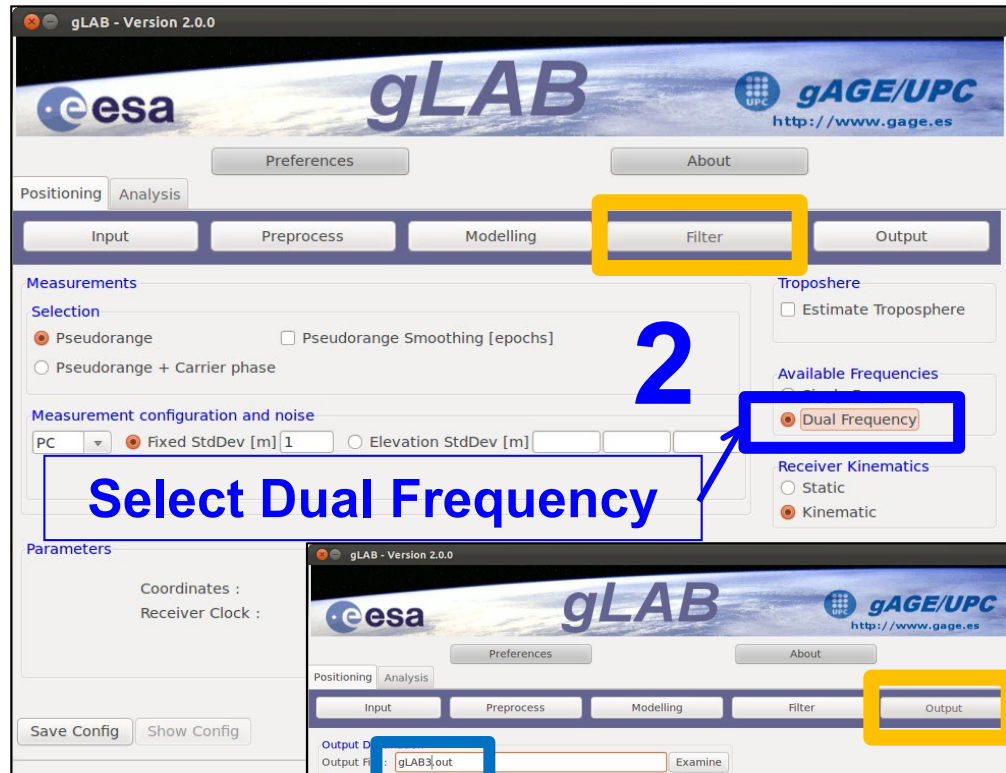
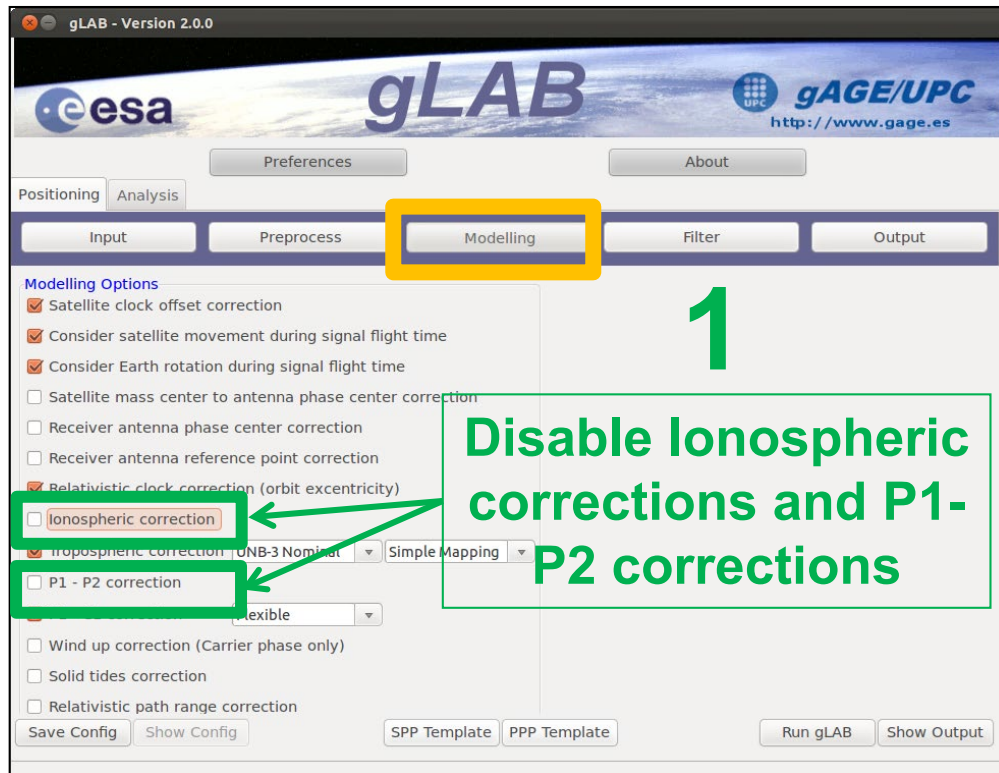


Equivalent command line sentence:

```
gLAB_linux -input:cfg gLAB_p1_NoIono.cfg  
-input:obs amc3030.03o  
-input:nav brdc3030.03n
```

# Ex. 3.4: Processing with PC → gLAB2.out

## 3. Reprocess the same files, but with 2-frequency ionosphere-free (PC)



Equivalent command line sentence:

```
gLAB_linux -input:cfg gLAB_p1_IFree.cfg  
-input:obs amc3030.03o  
-input:nav brdc3030.03n
```

# Ex. 3.4: Single freq. positioning effects

Execute in a single line: (or use the gLAB GUI)

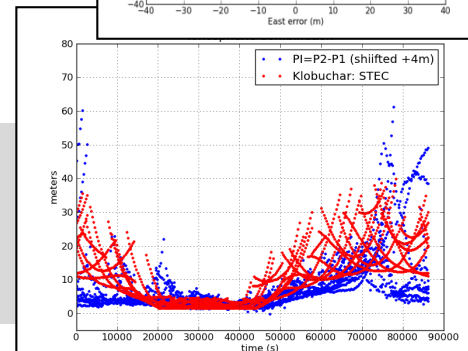
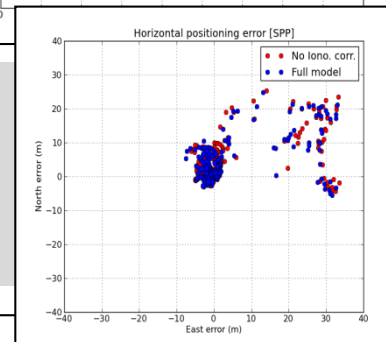
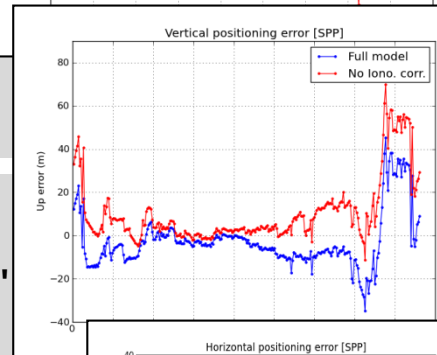
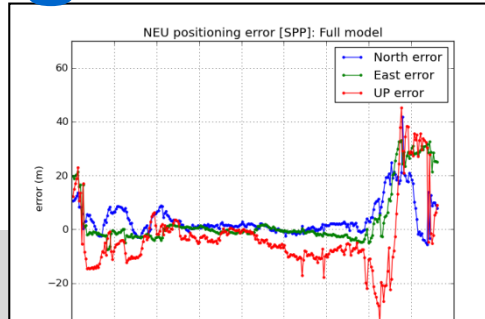
```
graph.py -f gLAB.out -x4 -y18 -s.- -c '($1=="OUTPUT")' -l "North error"
-f gLAB.out -x4 -y19 -s.- -c '($1=="OUTPUT")' -l "East error"
-f gLAB.out -x4 -y20 -s.- -c '($1=="OUTPUT")' -l "UP error"
--yn -40 --yx 70 --xl "time (s)" --yl "error (m)"
-t "NEU positioning error [SPP]: Full model"
```

```
graph.py -f gLAB.out -x4 -y20 -s.- -c '($1=="OUTPUT")' -l "Full model"
-f gLAB1.out -x4 -y20 -s.- -c '($1=="OUTPUT")' -l "No Iono." --cl
--yn -40 --yx 90 --xl "Time (s)" --yl "Up error (m)"
-t "Vertical positioning error [SPP]"
```

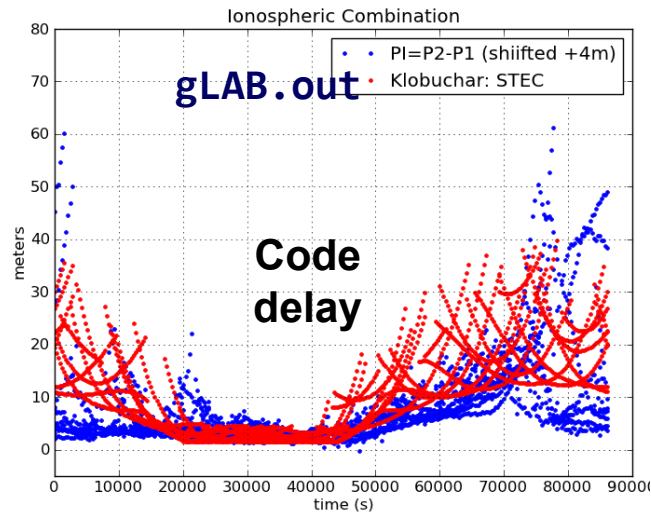
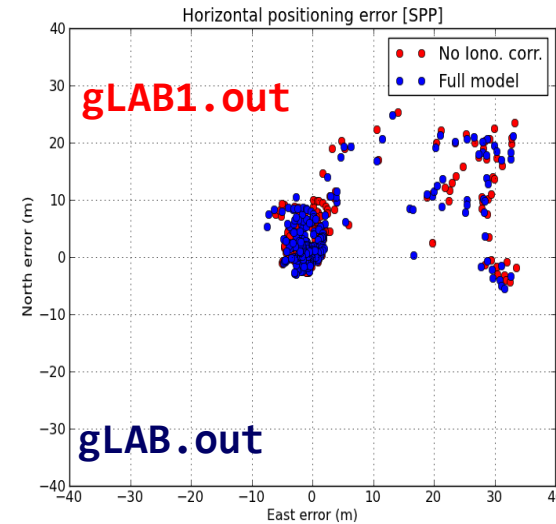
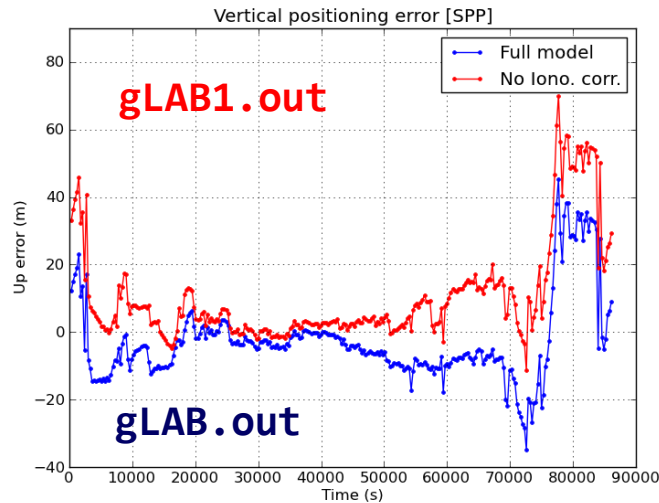
```
graph.py -f gLAB1.out -x19 -y18 -so -c '($1=="OUTPUT")' -l "No Iono." --cl r
-f gLAB.out -x19 -y18 -so -c '($1=="OUTPUT")' -l "Full mod" --cl b
--xl "East error (m)" --yl "North error (m)"
--xn -40 --xx 40 --yn -40 --yx 40
-t "Horizontal pos. error [SPP]"
```

```
graph.py -f gLAB.out -x4 -y'($10-$9+4)' -s.- -c '($1=="INPUT")'
-f gLAB.out -x4 -y25 -s.- -c '($1=="MODEL")' --cl r
--xl "time (s)" --yl "meters"
--yn -5 --yx 80 -t "Ionospheric Combination"
```

P2-P1 shifted +4 m



# Ex. 3.4: Single freq. positioning effects



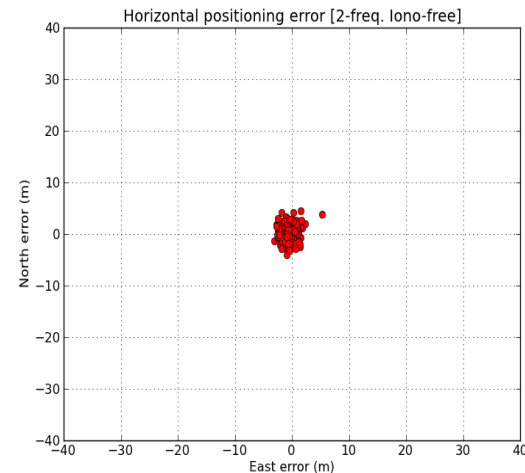
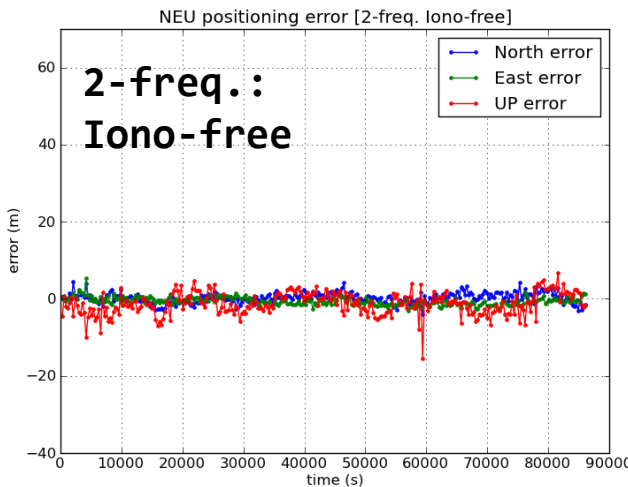
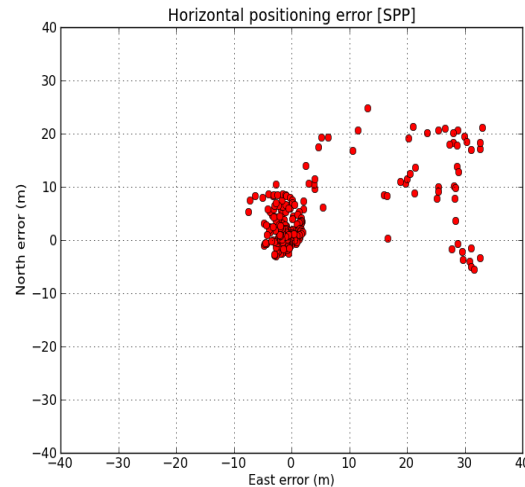
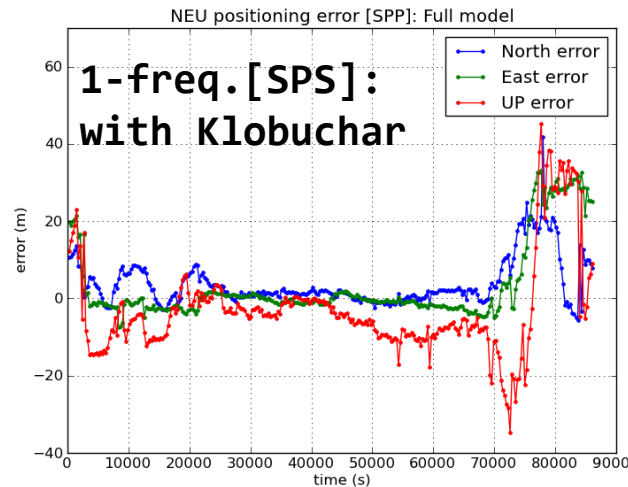
## Ionospheric correction (broadcast Klobuchar )

Ionospheric delays are larger at noon due to the higher insulation.

Klobuchar model is unable to mitigate the large ionospheric errors during the storm. Position domain errors reach up to 40 meters with Klobuchar corrections used.



# Ex. 3.4: Single freq. positioning effects



## Ionospheric correction (broadcast Klobuchar)

- The ionosphere-free combination (PC) of P1 and P2 codes is immune to the ionospheric storm.
- Although PC is three-times noisier than P1 or P2, it provides positioning accurate at the level of a few meters during the storm.

$$P_C = \frac{f_1^2 P_1 - f_2^2 P_2}{f_1^2 - f_2^2}$$



# Ex. 3.4: Single freq. positioning effects

## **Questions:**

- *Discuss the results seen in the previous plots. Is the Klobuchar model able to remove the ionospheric error during the storm? What was the level of the error?*
- *What is the position domain error level for single frequency users?*
- *And for dual frequency users navigating with the ionosphere-free combination?*
- *Discuss pros and cons of using the ionosphere-free combination against the single-frequency code with Klobuchar (consider nominal conditions and perturbed conditions).*

# OVERVIEW

- **Introduction:** gLAB processing in command line.
- **Test cases on Atmospheric Propagation Effects:**  
Troposphere, Ionosphere.
- **Test cases on Measurement Noise and Multipath:** C1, Ionosphere-free and Melbourne Wübbena combinations.
- **Test cases on Carrier Smoothed Code:** Code-carrier divergence.
- **Test cases on Broadcast Orbits and Clocks Accuracy**

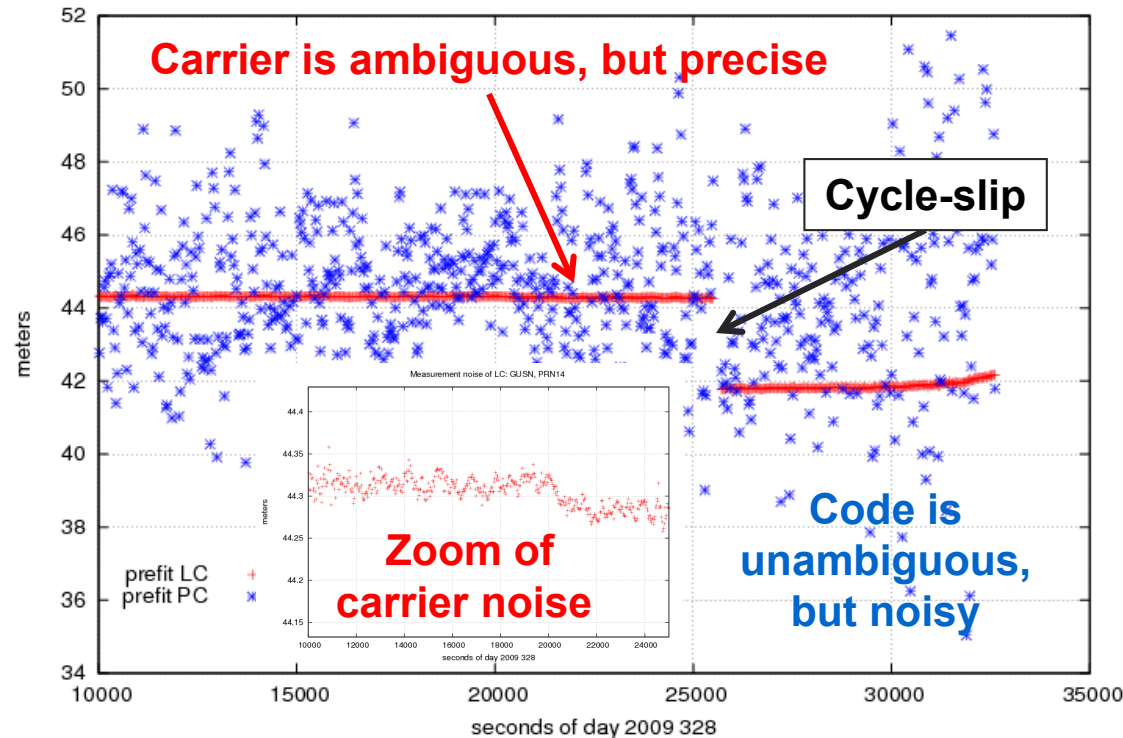


# Test cases on Measurement Noise and Multipath: C1, Iono-free and Melbourne Wübbena combinations.

# Measurement noise and multipath

## Code and carrier Measurements

Comparison of measurement noise of LC and PC: GUSN, PRN14



- Code measurements are unambiguous but noisy (meter level measurement noise).
- Carrier measurements are precise but ambiguous, meaning that they have few millimetres of noise, but also have unknown biases that could reach thousands of km.
- Carrier phase biases are estimated in the navigation filter along with the other parameters (coordinates, clock offsets, etc.). If these biases were fixed, measurements accurate to the level of few millimetres would be available for positioning. However, some time is needed to decorrelate such biases from the other parameters in the filter, and the estimated values are not fully unbiased.

**Note:** Figure shows the noise of **code** and **carrier** prefit-residuals, which are the input data for navigation equations.

# Exercise 4. Code multipath

The code multipath can be seen by plotting the difference of code and carrier ionosphere-free combinations (PC-LC). The evolution of this difference can be followed with a sampling rate of 1 Hz. Due to its geometric nature, the effect of multipath repeats with the receiver-satellite geometry.

The RINEX files **UPC33600.08N**, **UPC33610.08N**, **UPC33620.08N** contain observations at 1 Hz collected by a GPS receiver with fixed coordinates over the same period of time on three consecutive days. The corresponding navigation files are **BRD3600.08N**, **BRD3610.08N**, **BRD3620.08N**.

Using gLAB, read the RINEX files, plot the combination PC-LC and identify the effect of multipath. Analyse the measurements of satellites PRN20 and PRN25 in the time interval  $67000 < t < 68000$ s. Include the satellite's elevation in the plots.

# Exercise 4.1. PC Code multipath

Complete the following steps to depict the noise and multipath in the PC combination:

1. Read the RINEX files, execute:

```
gLAB linux -input:cfg meas.cfg -input:obs UPC33600.08 -input:nav BRD3600.08N > upc3360.meas
```

2. Verify the following field contents in the generated file upc3360.meas and others:

[Id	YY	Doy	sec	GPS	PRN	e1	Az	N.	list	C1C	L1C	C1P	L1P	C2P	L2P]
1	2	3	4	5	6	x	x	9	10	11	xx	13	14	15	16]

3. Compute the difference of code and carrier ionosphere-free combinations:  
(i.e. apply next equation) :

$$P_C = \frac{f_1^2 P_1 - f_2^2 P_2}{f_1^2 - f_2^2} = \frac{\gamma P_1 - P_2}{\gamma - 1};$$

$$L_C = \frac{f_1^2 L_1 - f_2^2 L_2}{f_1^2 - f_2^2} = \frac{\gamma L_1 - L_2}{\gamma - 1}$$

$$M_{P_C} = P_C - L_C$$

# Exercise 4.1. PC Code multipath

```
gawk 'BEGIN{g12=(154/120)**2}{if ($4>66000 && $4<69000)
    print $6,$4,(g12*$13-$15)/(g12-1)-(g12*$14-$16)/(g12-1),$7}' upc3360.meas >
                                         upc3360.LcPc
```

(Similarly for the other two files **upc3361.meas** and **upc3362.meas**.)

## 2. Plot the results:

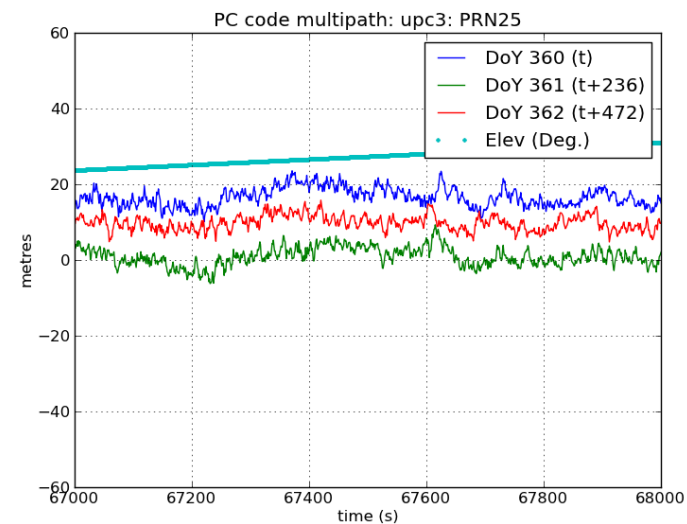
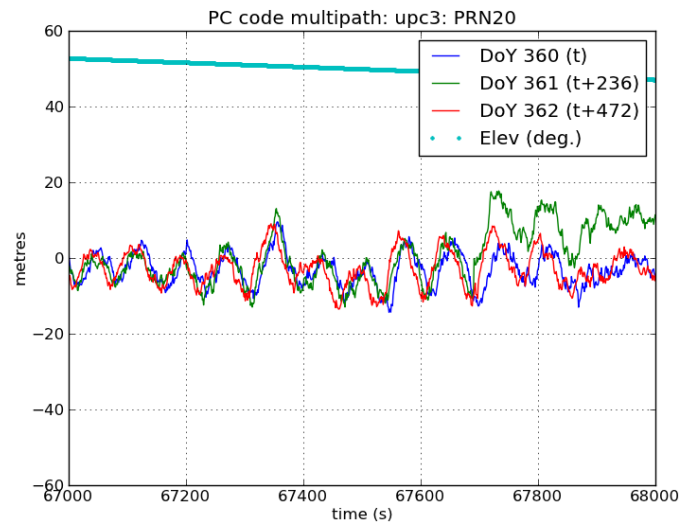
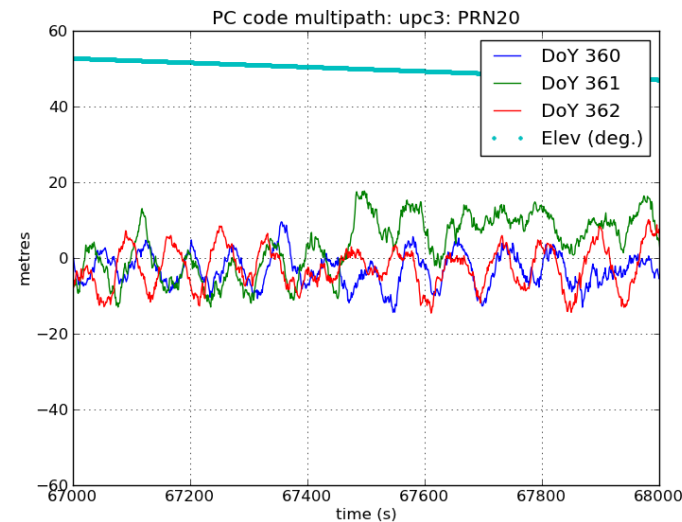
```
graph.py -f upc3360.LcPc -c '($1==20)' -x2 -y3 -s- -l "DoY 360"
        -f upc3361.LcPc -c '($1==20)' -x2 -y3 -s- -l "DoY 361"
        -f upc3362.LcPc -c '($1==20)' -x2 -y3 -s- -l "DoY 362"
        -f upc3360.LcPc -c '($1==20)' -x2 -y4 -l "Elev (deg.)"
```

- Repeat the same plots, but shift the plot of the second day by 3m 56s = 236s and the third day by 2x(3m 56s) = 472s.

```
graph.py -f upc3360.LcPc -c '($1==20)' -x2 -y3 -s- -l "DoY 360"
        -f upc3361.LcPc -c '($1==20)' -x'($2+236)' -y3 -s- -l "DoY 361"
        -f upc3362.LcPc -c '($1==20)' -x'($2+472)' -y3 -s- -l "DoY 362"
        -f upc3360.LcPc -c '($1==20)' -x2 -y4 -l "Elev (deg.)"
```

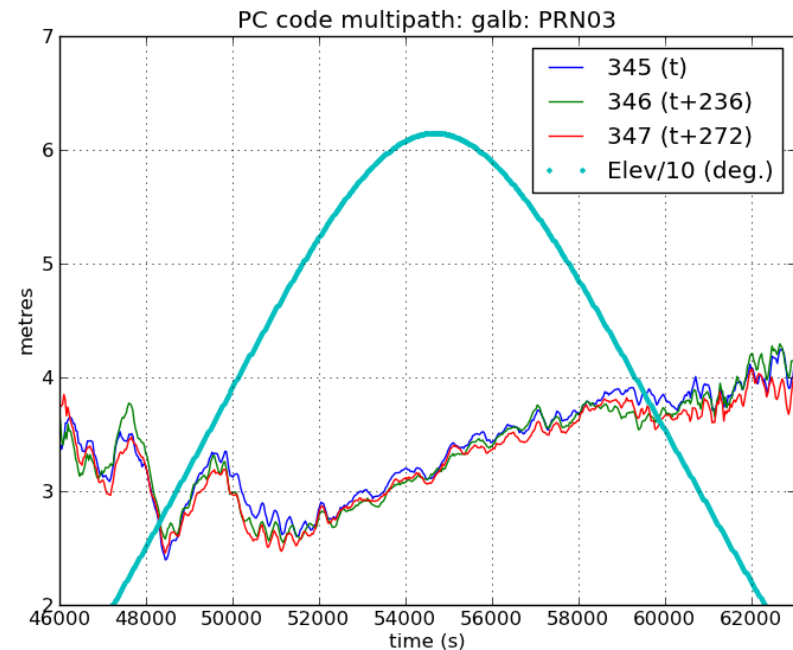
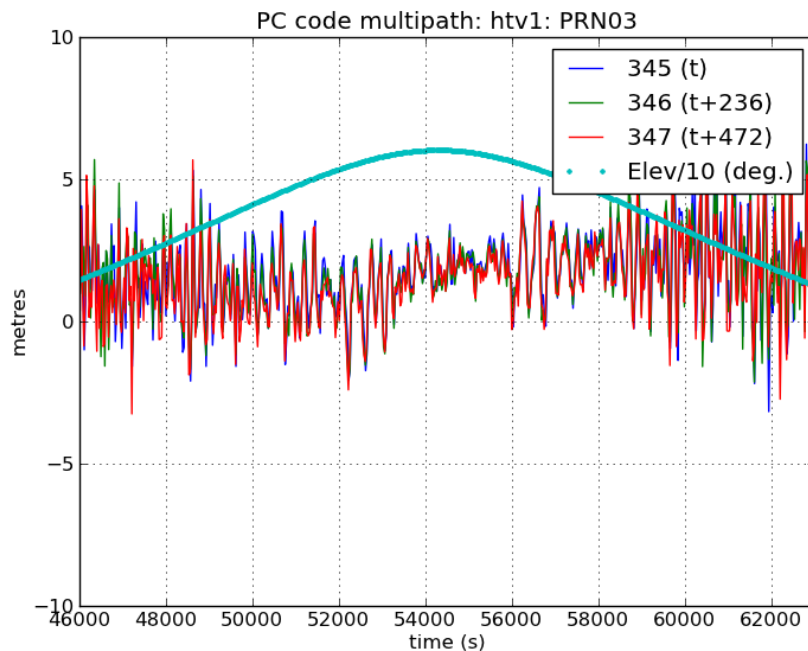
# Exercise 4.1. PC Code multipath

- What is the reason for the observed 3m 56s displacement between the graphs for two consecutive days?
- Repeat the previous plot for the satellite PRN25 and compare results.



# Exercise 4.1. PC Code multipath

**OPTIONAL:** Repeat the multipath analysis of the previous exercise using files **htv13450.04o**, **htv13460.04o**, **htv13470.04o** collected by the permanent receiver HTV1, and files **galb3450.04o**, **galb13460.04o**, **galb13470.04o** collected by the permanent receiver GALB. The associated broadcast orbit files are **brdc3450.04n**, **brdc3460.04n**, **brdc3470.04n**.





## Exercise 4.2. Melbourne-Wübbena multipath

Complete the following steps to depict the noise and multipath in the Melbourne Wübbena combination:

$$\left. \begin{aligned} P_N &= \frac{f_1 P_1 + f_2 P_2}{f_1 + f_2} = \frac{\sqrt{\gamma} P_1 + P_2}{\sqrt{\gamma} + 1}; \\ L_W &= \frac{f_1 L_1 - f_2 L_2}{f_1 - f_2} = \frac{\sqrt{\gamma} L_1 - L_2}{\sqrt{\gamma} - 1} \end{aligned} \right\} M_{MW} = L_W - P_N$$

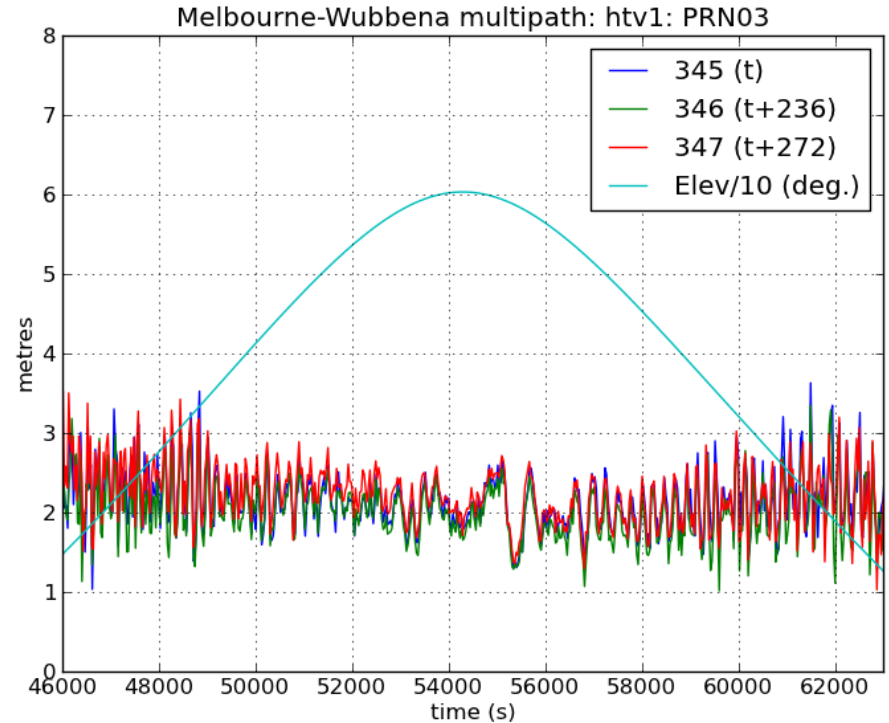
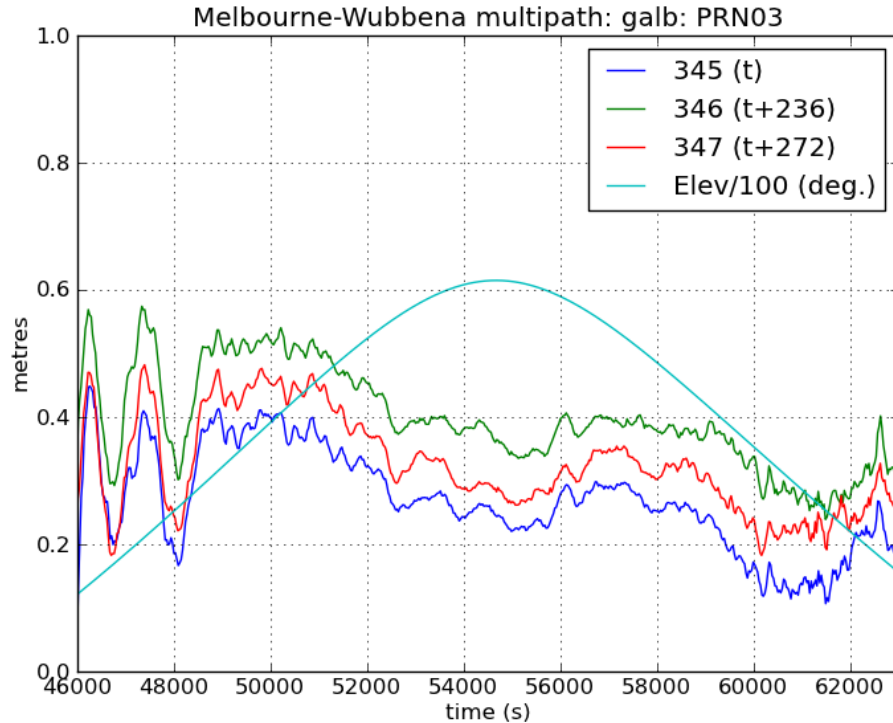
Execute (in a single line):

```
gawk 'BEGIN{s12=154/120}{print $6,$4,(s12*$14-$16)/(s12-1)-(s12*$13+$15)/(s12+1),$7}'  
galb3450.meas > galb3450.MW
```

(Similarly for the other files)

And plot results.

# Exercise 4.2. Melbourne-Wübbena multipath



# Exercise 4.3. C1 code multipath

The C1 code multipath and receiver noise can be depicted using the following combination (that removes all frequency dependent and not dependent terms):

$$M_{C_1} = C_1 - L_1 - 2\alpha(L_1 - L_2)$$

$$\alpha = \frac{f_2^2}{f_1^2 - f_2^2} = \frac{1}{\gamma - 1} = 1.545 \quad ; \quad \gamma = \left(\frac{77}{60}\right)^2$$

a) Generate the “meas” file:

Select, for instance, PRN03

Id	YY	Doy	sec	GPS	PRN	e1	Az	N.	list	C1C	L1C	C1P	L1P	C2P	L2P
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

```
gLAB_linux -input:cfg meas.cfg -input:obs UPC33510.080|gawk '{if ($6==03)print $0}'>upc3.meas
```

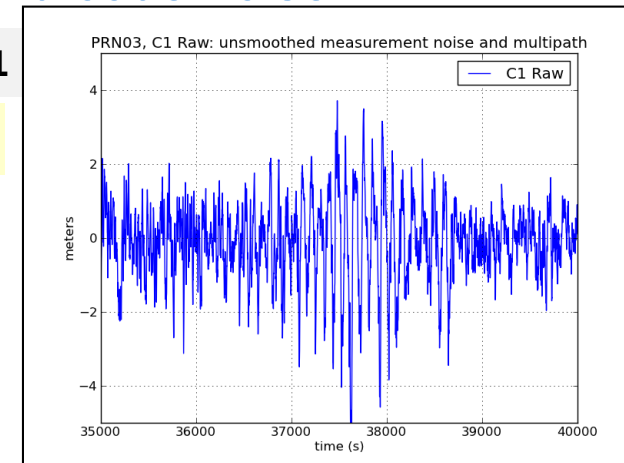
b) Using previous expression, compute the C1 multipath and code noise:

```
gawk '{print $4,$11-$14-3.09*($14-$16)-21.3}' upc3.meas> upc3.C1
```

**[\*] results are Shifted by “-21.3” to remove the carrier ambiguity**

c) Plot results for PRN03:

```
graph.py -f upc3.C1 -s- -l "C1 Raw" --xn 35000 --xx 40000
--yn -5 --yx 5 --xl "time (s)" --yl "meters"
-t "PRN03, C1 Raw measurement noise and multipath"
```



# OVERVIEW

- **Introduction:** gLAB processing in command line.
- **Test cases on Atmospheric Propagation Effects:**  
Troposphere, Ionosphere.
- **Test cases on Measurement Noise and Multipath:** C1, Ionosphere-free and Melbourne Wübbena combinations.
- **Test cases on Carrier Smoothed Code:** Code-carrier divergence.
- **Test cases on Broadcast Orbits and Clocks Accuracy**

# Test cases on Carrier Smoothed Code: Code-carrier divergence

# Carrier smoothed code

The noisy code  $P$  can be smoothed with the precise (but ambiguous) carrier  $L$  measurements. This carrier smoothing can be done in real-time applying the **Hatch filter**.

$$\hat{P}(k) = \frac{1}{n} P(k) + \frac{n-1}{n} (P(k-1) + L(k) - L(k-1))$$

where  $[n = k \text{ if } k < N]$ , and  $[n = N \text{ if } k \geq N]$ .

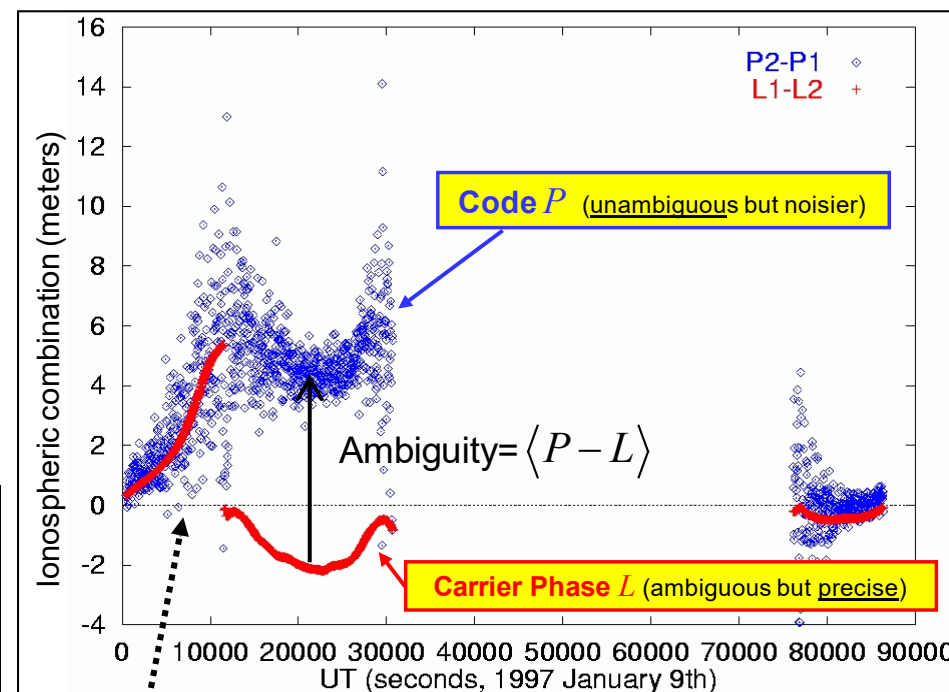
The algorithm is initialised with:  $\hat{P}(1) = P(1)$ .

The previous algorithm can be interpreted as real-time alignment of carrier with code:

$$\begin{aligned} \hat{P}(k) &= \frac{1}{n} P(k) + \frac{n-1}{n} (P(k-1) + L(k) - L(k-1)) = \\ &= L(k) + \langle P - L \rangle_{(k)} \end{aligned}$$

where

$$\langle P - L \rangle_{(k)} = \frac{1}{n} (P(k) - L(k)) + \frac{n-1}{n} \langle P - L \rangle_{(k-1)} = \frac{1}{n} \sum (P(k) - L(k))$$



# Carrier smoothed code

The noisy code can be smoothed with the precise (but ambiguous) carrier measurements. This carrier smoothing can be done in real-time applying the Hatch filter.

The smoothing depends on the time smoothing constant or filter length. The more the filter length is used, the more smoothed the code is, but (with single frequency measurements) a higher code-carrier divergence error is induced by the ionosphere.

This is because the ionospheric refraction has **an** opposite sign on code and carrier, its effect being twice on the difference of code and carrier. This double ionospheric refraction is propagated forward through the filter, producing a bias.

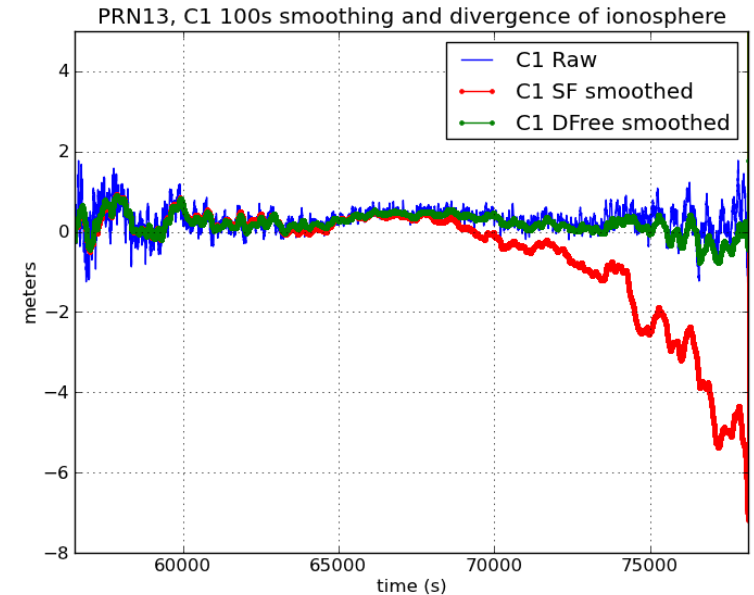
The error induced by the code-carrier divergence of the ionosphere on the single frequency smoothed codes is assessed in this exercise for different filter lengths.



# Ex.5.: Iono. Divergence on Smoothing

The target of this exercise is to analyze the error induced by the divergence of the ionosphere (between code and carrier) into the Single-Frequency (SF) carrier smoothed code.

The Divergence Free (Dfree) and the Ionosphere Free (IFree) smoothed codes will be compared with the SF one.



This effect will be analyzed analytically and tested with single and double frequency GPS measurements under large ionospheric gradients.

# Ex.5.: Iono. Divergence on Smoothing

**Time varying ionosphere induces a bias in the single frequency smoothed code** when it is averaged in the smoothing filter. This effect is analysed as follows:

Let:

$$\begin{aligned} P_1 &= \rho + I_1 + \varepsilon_1 \\ L_1 &= \rho - I_1 + B_1 + \varsigma_1 \end{aligned}$$

Where  $\rho$  includes all non dispersive terms (geometric range, clock offsets, troposphere) and  $I_1$  represents the frequency dependent terms (ionosphere and DCBs).  $B_1$  is the carrier ambiguity, which is constant along continuous carrier phase arcs, and  $\varepsilon_1, \varsigma_1$  account for code and carrier multipath and thermal noise.

therefore,

$$P_1 - L_1 = 2I_1 - B_1 + \varepsilon_1 \Rightarrow 2I_1 : \text{Code-carrier divergence}$$

*Note: the carrier noise  $\varsigma_1$  is neglected against code noise  $\varepsilon_1$ .*

Substituting  $P_1 - L_1$  in Hatch filter equation (see slide #61):

$$\begin{aligned} \hat{P}(k) &= L(k) + \langle P - L \rangle_{(k)} = \rho(k) - I_1(k) + B_1 + \langle 2I_1 - B_1 \rangle_{(k)} = \\ &= \rho(k) + I_1(k) + 2 \underbrace{\left( \langle I_1 \rangle_{(k)} - I_1(k) \right)}_{bias_I} \end{aligned}$$

$$\Rightarrow \hat{P}_1 = \rho + I_1 + bias_I + v_1$$

where, being the ambiguity term  $B_1$  a constant bias, thence  $\langle B_1 \rangle = B_1$ , and cancels in the previous expression

where  $v_1$  is the noise term after smoothing

# Ex.5.: Iono. Divergence on Smoothing

## Raw assessment of the induced bias on $P_1$ smoothed code by ionosphere:

- Let assume a simple model where the STEC vary linearly with time:

$$I_1(t) = I_{1_0} + I'_1 \cdot t \Rightarrow \boxed{bias_I = 2\left(\langle I_1 \rangle_{(k)} - I_1(k)\right) = -2\tau I'_1}$$

where  $\tau$  is the Hatch filter smoothing time constant (i.e.,  $\tau = N$  in previous eq.).

### Exercise:

*Prove the previous statement.*

*Solution:*

Let be  $f(t) \equiv I(t)$  and  $y(t) \equiv \langle I \rangle_{(t)}$ . The averaging in the Hatch filter can be implemented as:

$$y(t + \Delta T) = \frac{\tau - \Delta T}{\tau} y(t) + \frac{\Delta T}{\tau} f(t + \Delta T) \Rightarrow \frac{y(t + \Delta T) - y(t)}{\Delta T} + \frac{1}{\tau} y(t) = \frac{1}{\tau} f(t + \Delta T) \xrightarrow{\Delta T \rightarrow 0} y' + \frac{1}{\tau} y = \frac{1}{\tau} f(t)$$

Thence:

$$I_1(t) = I_{1_0} + I'_1 \cdot t \Rightarrow \langle I_1 \rangle_{(t)} = I_1(t) - \tau I'_1 (1 - e^{-t/\tau}) \Rightarrow \boxed{bias_I = 2\left(\langle I_1 \rangle_{(t)} - I_1(t)\right) \xrightarrow{t \rightarrow \infty} -2\tau I'_1}$$

# Ex.5.: Iono. Divergence on Smoothing

- **Divergence Free smoothing (DFree):**

With 2-frequency measurements, the ionosphere can be removed from a combination of two carriers:

$$P_1 - L_1 - 2\alpha(L_1 - L_2) = B_{12} + \varepsilon_1 \Rightarrow \hat{P}_1 = \rho + I_1 + \nu_{12}$$

$$\alpha = \frac{f_2^2}{f_1^2 - f_2^2}$$

$$B_{12} = -B_1 - 2\alpha(B_1 - B_2)$$

→ **DFree** smoothed code is not affected by iono. temporal gradients, being the ionospheric delay the same as in the original code  $P_1$ .

- **Ionosphere Free smoothing (IFree):**

Using both code and carrier 2-frequency measurements, it is possible to remove the frequency dependent effects using the ionosphere-free combination  $P_C, L_C$ :

$$\begin{cases} P_C = \rho + \varepsilon_C \\ L_C = \rho + B_C + \zeta_C \end{cases} \Rightarrow \hat{P}_C = \rho + \nu_C$$

→ **IFree** smoothed code is not affected by either spatial or temporal gradients, but is **3-times noisier** than the DFree, or the in the Single Freq. smoothed code.

$$P_C = \frac{f_1^2 P_1 - f_1^2 P_2}{f_1^2 - f_2^2}$$

$$L_C = \frac{f_1^2 L_1 - f_1^2 L_2}{f_1^2 - f_2^2}$$

# Ex. 5.1: Iono. Divergence on Smoothing

## 1.- Multipath and measurement noise assessment on raw code measurements:

The C1 code multipath and receiver noise can be depicted using the following combination (that removes all frequency dependent and not dependent terms):

$$M_{C_1} = C_1 - L_1 - 2\alpha(L_1 - L_2) \quad \alpha = \frac{f_2^2}{f_1^2 - f_2^2} = \frac{1}{\gamma - 1} = 1.545 \quad ; \quad \gamma = \left(\frac{77}{60}\right)^2$$

a) Generate the “meas” file for **PRN03**:

[Id	YY	Doy	sec	GPS	PRN	e1	Az	N.	list	C1C	L1C	C1P	L1P	C2P	L2P]
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16]

```
gLAB_linux -input:cfg meas.cfg -input:obs UPC33510.080|gawk '{if ($6==03)print $0}'>upc3.meas
```

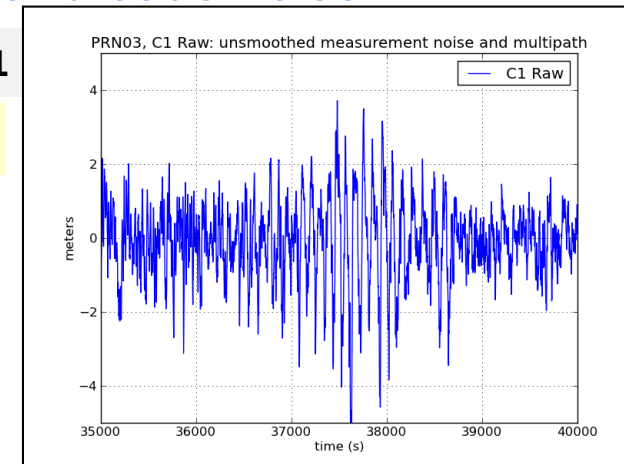
b) Using previous expression, compute the C1 multipath and code noise: :

```
gawk '{print $4,$11-$14-3.09*($14-$16)-21.3}' upc3.meas> upc3.C1
```

**[\*] results are Shifted by “-21.3” to remove the carrier ambiguity**

c) Plot the raw (unsmoothed) measurements for PRN03:

```
graph.py -f upc3.C1 -s- -l "C1 Raw" --xn 35000 --xx 40000  
--yn -5 --yx 5 --xl "time (s)" --yl "meters"  
-t "PRN03, C1 Raw measurement noise and multipath"
```



# Ex. 5.1: Iono. Divergence on Smoothing

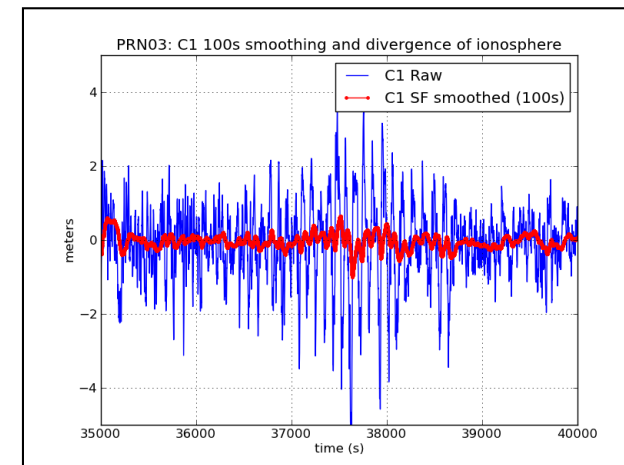
2. Apply the Hatch filter to smooth the code using a filter length of  $N = 100$  sample (as the measurements are at  $1\text{Hz}$ , this means 100 seconds smoothing). Then, as in the previous case, depict the multipath and noise of the smoothed code.

a) Smoothing code ( $T=100\text{sec}$ ):

```
gawk 'BEGIN{Ts=100}{if (NR>Ts){n=Ts}else{n=NR};  
      C1s=$11/n+(n-1)/n*(C1s+($14-L1p));L1p=$14;  
      print $4,C1s-$14-3.09*($14-$16)-21.3}' upc3.meas > upc3.C1s100
```

b) Plotting results and compare with the row C1.

```
graph.py -f upc3.C1 -s- -l "C1 Raw"  
-f upc3.C1s100 -s.- --cl r -l "C1 SF smoothed"  
--xn 35000 --xx 40000 --yn -5 --yx 5  
--xl "time (s)" --yl "meters"  
-t "PRN03: C1 100s smoothing and iono div."
```



# Ex. 5.1: Iono. Divergence on Smoothing

- 3 . Using 2-frequency carriers it is possible to generate a combination with the same ionospheric delay (the same sign) as the code to avoid the code-carrier divergence

$$L_{1DFree} = L_1 + 2\alpha(L_1 - L_2) = \rho + I_1 + B + \zeta$$

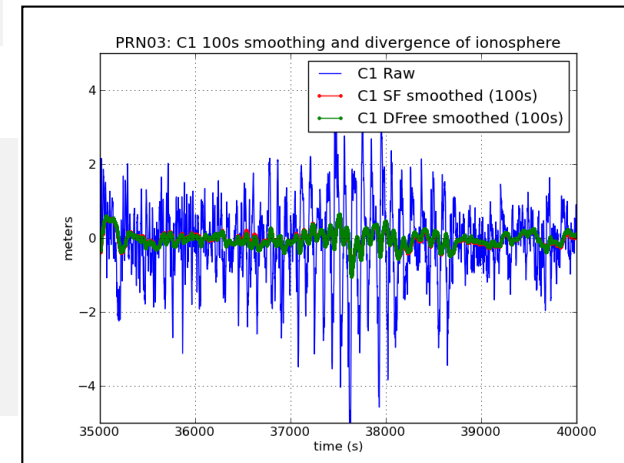
$$\alpha = \frac{f_2^2}{f_1^2 - f_2^2} = \frac{1}{\gamma - 1} = 1.545 \quad ; \quad \gamma = \left(\frac{77}{60}\right)^2$$

a) Apply the Hatch filter to compute the DFree smoothed code

```
gawk 'BEGIN{Ts=100}{if (NR>Ts){n=Ts}else{n=NR};  
C1f=$11;L1f=$14+2*1.545*($14-$16);  
C1fs=C1f/n+(n-1)/n*(C1fs+(L1f-L1p));L1p=L1f;  
print $4,C1fs-L1f-21.3}' upc3.meas > upc3.C1DFs100
```

b) Plot results and compare with the raw C1 code:

```
graph.py -f upc3.C1 -s- -l "C1 Raw"  
-f upc3.C1s100 -s- --cl r -l "C1 SF smoothed (100s)"  
-f upc3.C1DFs100 -s- --cl g -l "C1 DFree smooth(100s)"  
--xn 35000 --xx 40000 --yn -5 --yx 5  
--xl "time (s)" --yl "meters"
```





# Ex. 5.1: Ionos. Divergence on Smoothing

- 4 . Generate the ionosphere-free combinations of code and carrier measurements to compute the Ionosphere Free (IFree) smoothed code:

$$C_{IFree} \equiv P_C = \frac{\gamma P_1 - P_2}{\gamma - 1} \quad ; \quad L_{IFree} \equiv L_C = \frac{\gamma L_1 - L_2}{\gamma - 1} \quad \gamma = \left(\frac{77}{60}\right)^2$$

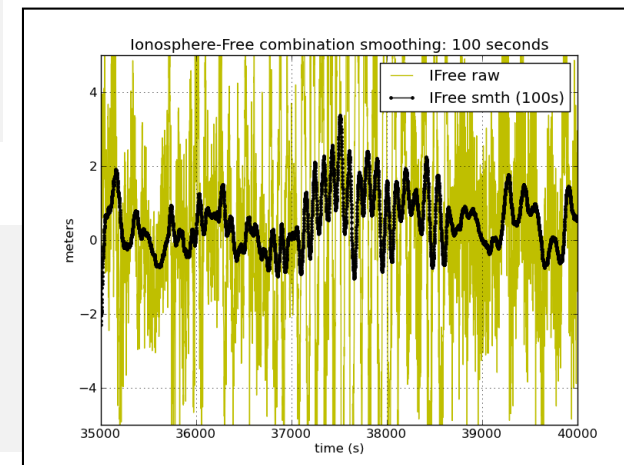
```
gawk 'BEGIN{g=(77/60)**2}{pc=(g*$13-$15)/(g-1); lc=(g*$14-$16)/(g-1);  
      print $4,pc-lc-3.5}' upc3.meas > upc3.PC
```

- Apply the Hatch filter to compute the IFree smoothed code

```
gawk 'BEGIN{g=(77/60)**2}{pc=(g*$13-$15)/(g-1);  
      lc=(g*$14-$16)/(g-1); if (NR>100){n=100}else{n=NR};  
      ps=1/n*pc+((n-1)/n*(ps+lc-lcp)); lcp=lc;  
      print $4,ps-lc-3.5}' upc3.meas > upc3.PCs100
```

- Plot results and compare with the unsmoothed PC:

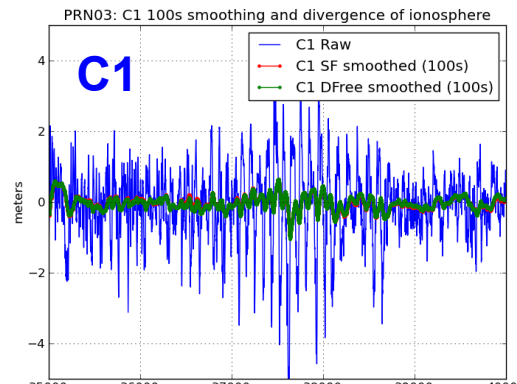
```
graph.py -f upc3.PC -s- -l "IFree raw"  
-f upc3.PCs100 -s.- --cl black -l "Ifree(100s)"  
--xn 35000 --xx 40000 --yn -5 --yx 5  
--xl "time (s)" --yl "meters"
```



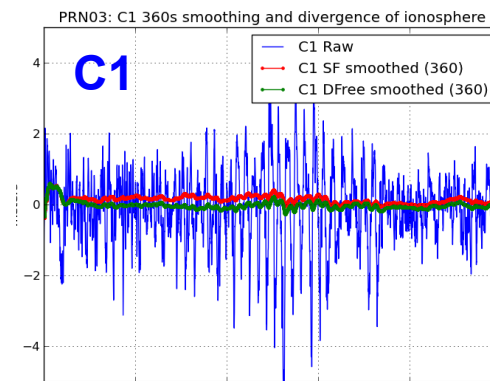
# Ex. 5.1: Iono. Divergence on Smoothing

5 . Repeat previous plots but using:  $N=360$ ,  $N=3600$  and compare results. Plot also the ionospheric delay (from L1-L2) (see more details in [R-1]):

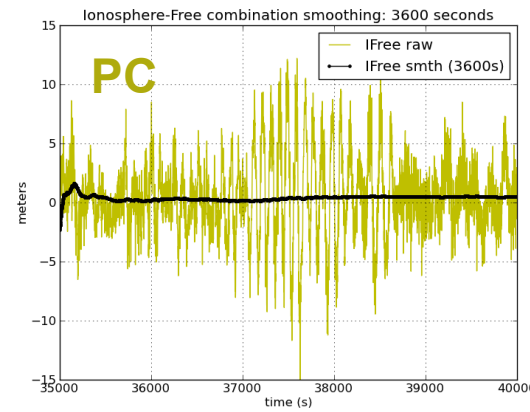
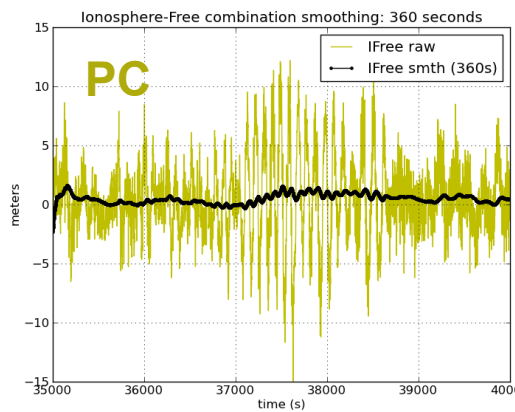
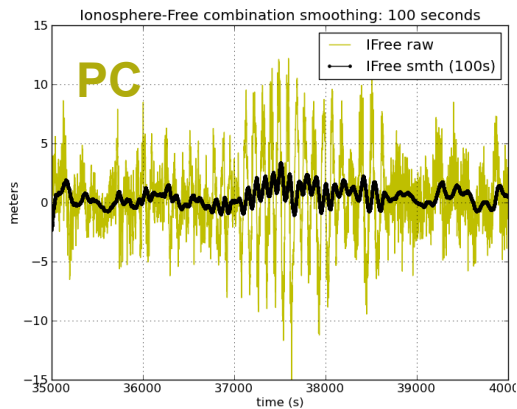
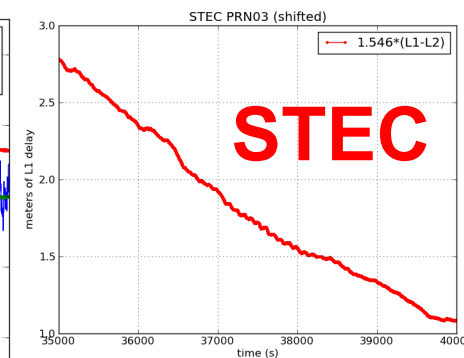
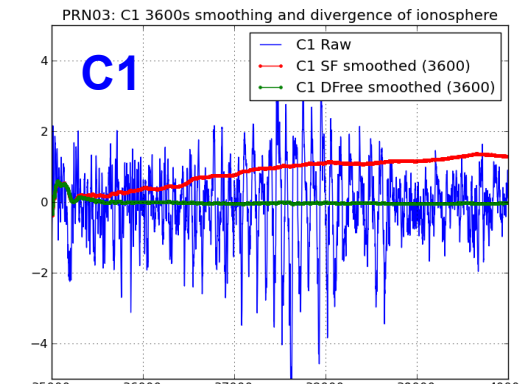
**T=100s**



**T=360s**



**T=3600s**



Note that the y-range in bottom row plots is 3 times larger than in top plots

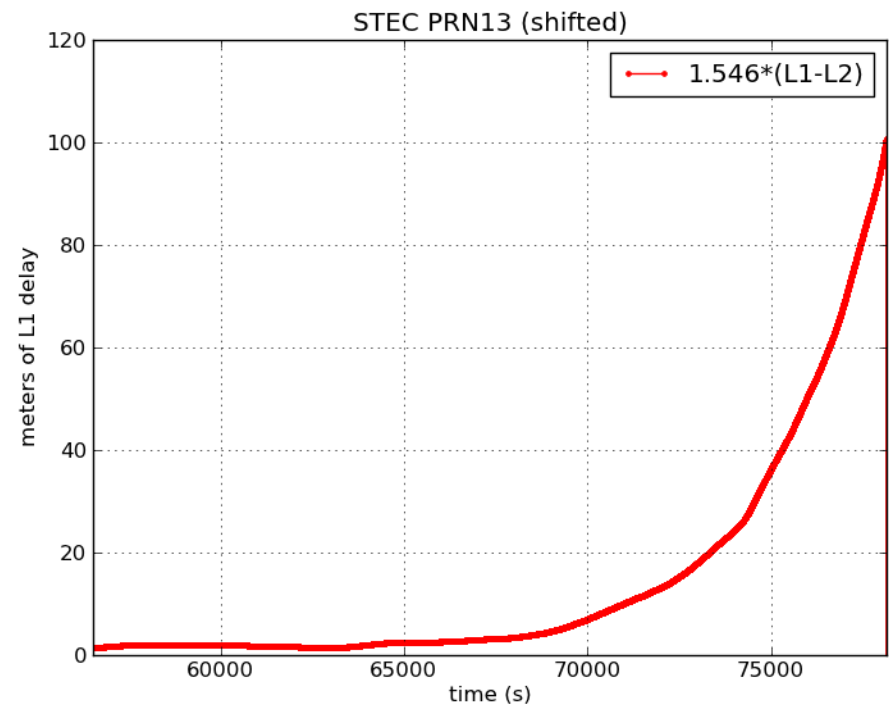
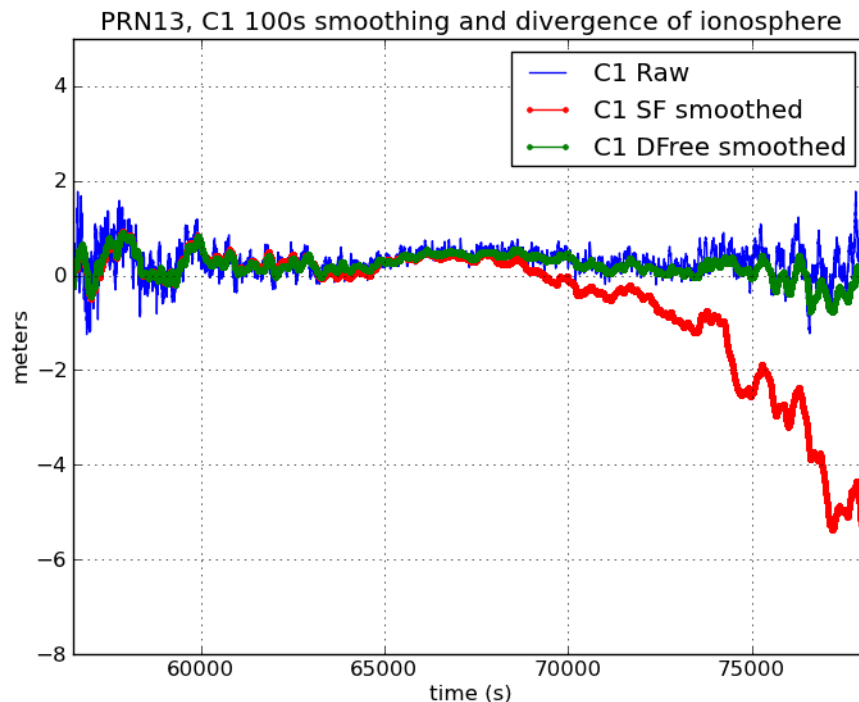
# Ex. 5.1: Iono. Divergence on Smoothing

## **Questions:**

- *Discuss a possible source of the large oscillations seen in the plots.*
- *Using the expressions of slide #65, estimate the expected accumulated ionospheric bias with smoothing time constant  $T=3600s$ .*
- *Is the Divergence-Free smoothing immune to ionospheric temporal gradients? And to spatial gradients? Why?*
- *Is the Ionosphere-Free smoothing immune to temporal gradients? And to spatial gradients?*

# Ex.5.2.: Assessment on Halloween storm

Repeat the previous exercise using the RINEX file `amc23030.03o_1Hz` collected for the station `amc2` during the Halloween storm. Take  $N=100$  (i.e. filter smoothing time constant  $\tau=100$  sec).



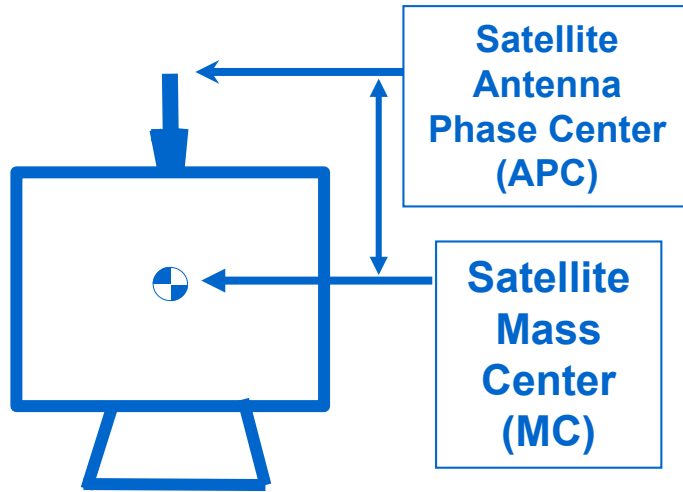
# OVERVIEW

- ✦ **Introduction:** gLAB processing in command line.
- ✦ **Test cases on Atmospheric Propagation Effects:**  
Troposphere, Ionosphere.
- ✦ **Test cases on Measurement Noise and Multipath:** C1, Ionosphere-free and Melbourne Wübbena combinations.
- ✦ **Test cases on Carrier Smoothed Code:** Code-carrier divergence.
- **Test cases on Broadcast Orbits and Clocks Accuracy**

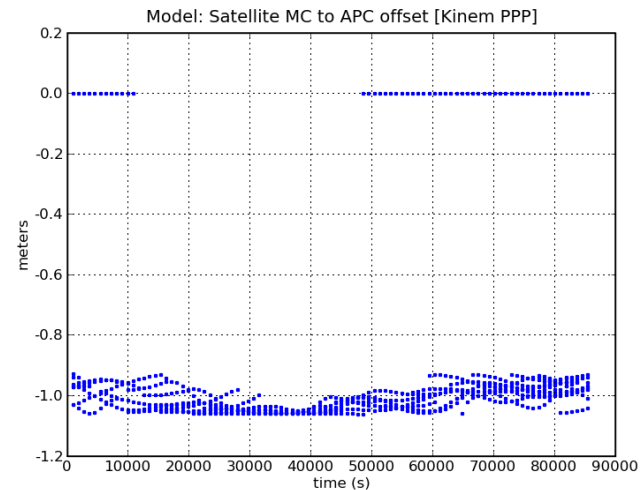
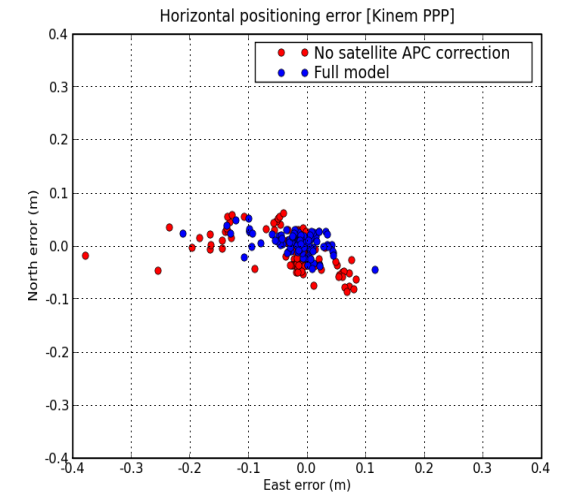
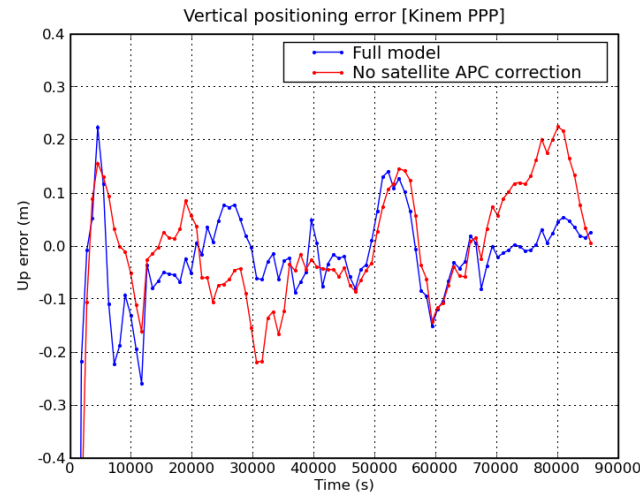
# Test cases on Broadcast Orbits and Clocks Accuracy

# Broadcast v.s. precise IGS Orbits and clocks

## Satellite Mass Center to Antenna Phase Center



Broadcast orbits are referred to the antenna phase center, but IGS precise orbits are referred to the satellite mass center.



## Satellite MC to APC:

The satellite MC to APC eccentricity vector depends on the satellite. The APC values used in the IGS orbits and clocks products are referred to the iono-free combination (LC, PC). They are given in the IGS ANTEX files (e.g., igs05.atx).



## Ex.6.: Broadcast orbits and clocks accuracy assessment using the IGS precise products as the accurate reference (i.e. the truth).

### Complete the following steps:

File **brdc0800.07n** contains the **orbits** and clocks data broadcast in the GPS navigation message. Files **cod14193.sp3**, **cod14193.clk** contain the precise orbits and clocks computed in post-process by “CODE” center (IGS precise orbits and clocks products program).

1. **Execute** the following sentence to compute the difference of satellite coordinates and clock offsets between both orbits and clocks sources:

```
gLAB_linux -input:nav brdc0800.07n -input:SP3 cod14193.sp3 -input:ant igs05_1402.atx >dif.tmp
```

2. **Select** the SATDIFF message of **dif.tmp** file:

```
grep SATDIFF dif.tmp > dif.out
```

SATDIFF message content is shown in the table beside.  
(see **gLAB\_linux -messages**).

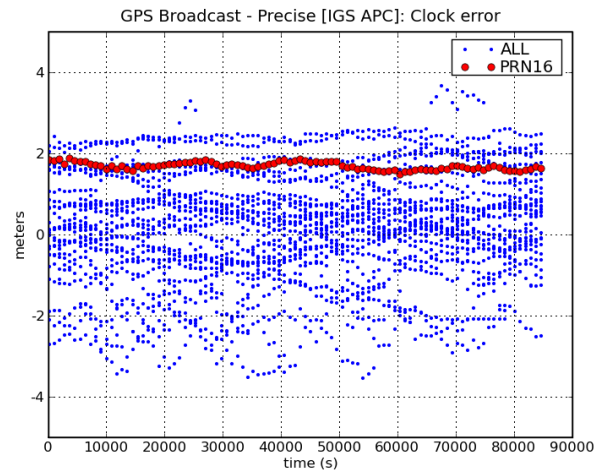
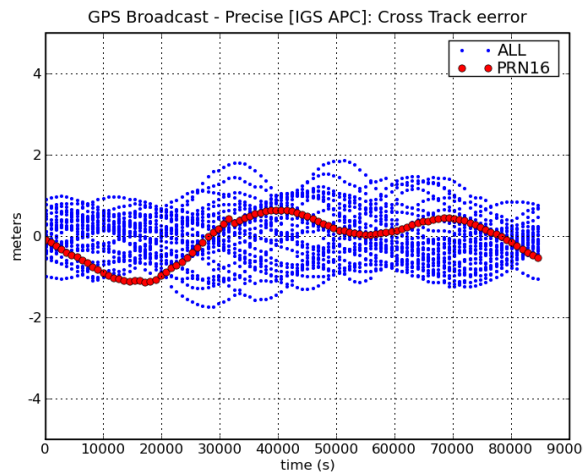
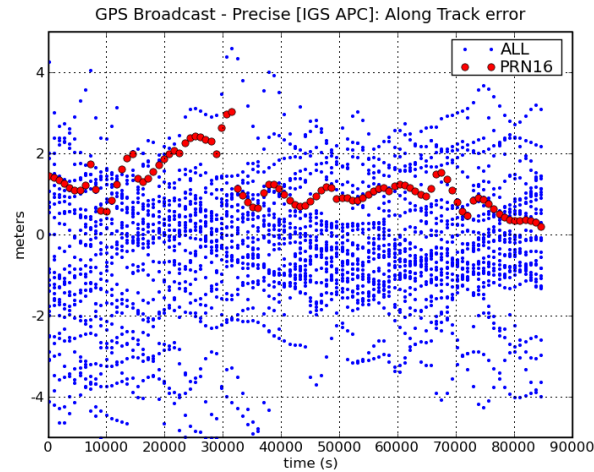
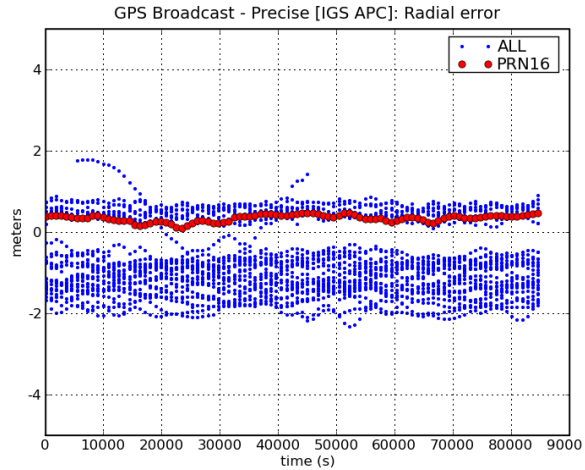
The IGS post-processed products are accurate at few cm level, therefore they can be taken as the truth.

3. **Plot** **dif.out** file as in the first exercise.

**Note:** 
$$SISRE = \sqrt{(\Delta Rad - \Delta Clk)^2 + \frac{1}{49}(\Delta Alon^2 + \Delta Cross^2)}$$

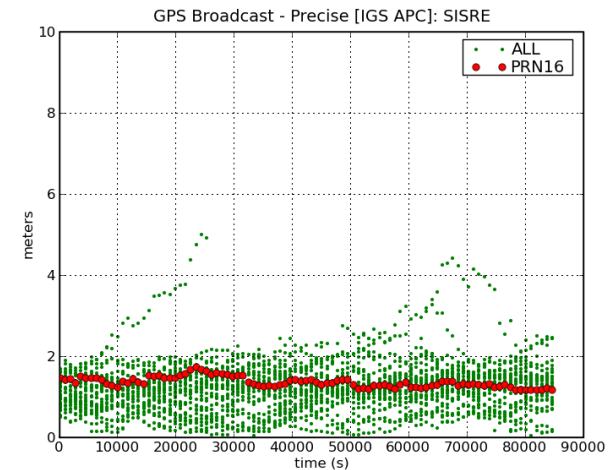
Field	Content
1	'SATDIFF'
2	Year
3	Doy (Days-of-Year)
4	Seconds of day (seconds)
5	GNSS System (GPS, GAL, GLO or GEO)
6	PRN satellite identifier
7	SISRE difference (meters)
8	SISRE orbit-only difference (meters)
9	3D orbit difference (meters)
10	<i>clkDiff</i> : Clock difference (meters)
11	<i>radDiff</i> : Radial position difference (meters)
12	<i>atDiff</i> : Along-track position difference (meters)
13	<i>ctDiff</i> : Cross-track position difference (meters)

# Ex.6.: Broadcast orbits and clocks accuracy assessment using the IGS precise products as the accurate reference (i.e. the truth).

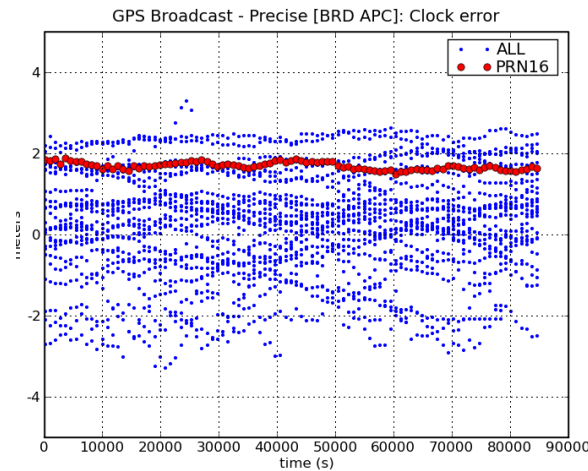
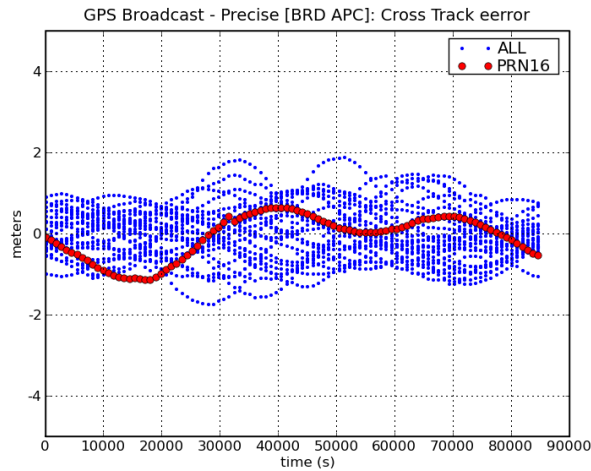
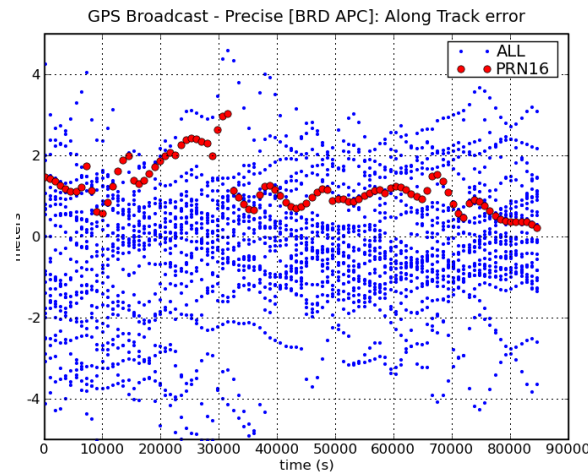
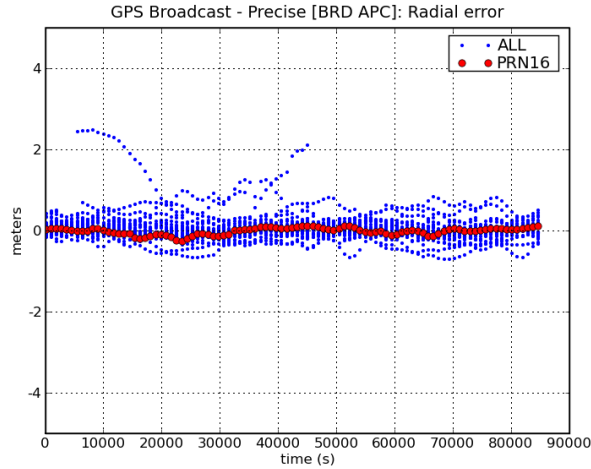


## Comments

- Meter level errors are found on broadcast orbits and clocks.
- The bias seen in the radial component is due to the different APC's used by the GPS ground segment (i.e. in broadcast orbits) and by IGS (precise products).
- This bias is compensated by a similar shift in clocks.
- For the Signal-In-Space-Range-Error (SISRE), please see the plot below.



# Ex.6.: Broadcast orbits and clocks accuracy assessment using the IGS precise products as the accurate reference (i.e. the truth).

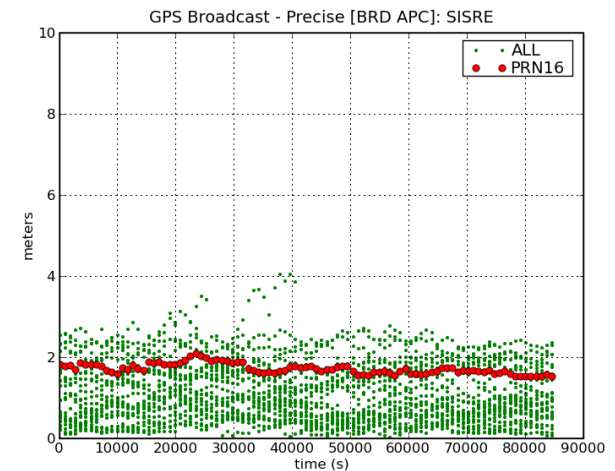


## Comments

The previous computations have been repeated, but using the ANTEX file `gps_brd.atx`, instead of `igs05_1402.atx`.

This new ANTEX file contains the GPS antenna phase center offsets used by the GPS ground segment, not the IGS ones.

- Notice that the biases in the radial component have disappeared.



## Ex.6.: Broadcast orbit and clock accuracy assessment using the IGS precise products as the accurate reference (i.e. the truth).

### **Questions:**

- *What is the level of error of the broadcast orbits and clocks?*
- *Why do the results improve using the `gps_brd.atx`, instead of `igs05_1402.atx` ANTEX file?*
- *Is there any correlation between radial orbit errors and clock errors?*
- *Which orbit errors are expected to affect the differential positioning more (radial, cross track, along track)?*

# Thanks for your attention

# Bibliography

- R-1: J. Sanz-Subirana, J.M. Jaun-Zoroza, M. Hernández-Pajares. GNSS Data processing. Volume-I and Volume-II. ESA Publications Division, 2012.
- R-2: J. Sanz-Subirana, J.M. Jaun-Zoroza, M. Hernández-Pajares. Tutorial on GNSS Data Processing Laboratory Exercises. ESAInternational Summer School on GNSS, 2010.
- R-3: Hernández-Pajares M., J.M. Juan, J. Sanz, "EGNOS Test Bed Ionospheric Corrections Under the October and November 2003 Storms", IEEE Transactions on Geoscience and Remote Sensing, Vol.43(10), pp.2283-2293, 2005.

# Acknowledgements

- ✦ The ESA/UPC GNSS-Lab Tool suite (gLAB) has been developed under the ESA Education Office contract N. P1081434.
- ✦ The other data files used in this study were acquired as part of NASA's Earth Science Data Systems and archived and distributed by the Crustal Dynamics Data Information System (CDDIS).
- ✦ To Adrià Rovira-Garcia for his contribution to the edition of this material and gLAB updating.