Tutorial 9

Differential Positioning and carrier ambiguity fixing

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

gAGE/UPC
Research group of Astronomy & Geomatics
Technical University of Catalonia

Aim of this tutorial

- This tutorial is devoted to analysing and assessing the differential positioning with carrier phase measurements (L1, L2 and LC). Five different receivers and three baselines (of 7m, 18m and 15km) are considered.

- This study includes ambiguity fixing with the LAMBDA method and the analysis of different effects such as the geometry diversity and atmospheric propagation errors (troposphere and ionosphere).

- Two different implementations of differential positioning are considered:
  - Using time-tagged measurements the baseline vector is directly estimated.
  - Using computed differential corrections a user receiver is positioned.

  The effect of synchronization errors between the reference station and the user is also analysed for both implementations.

- **All software tools** (including ***gLAB***) and associated files for the laboratory session are included in the USB stick associated with this tutorial.

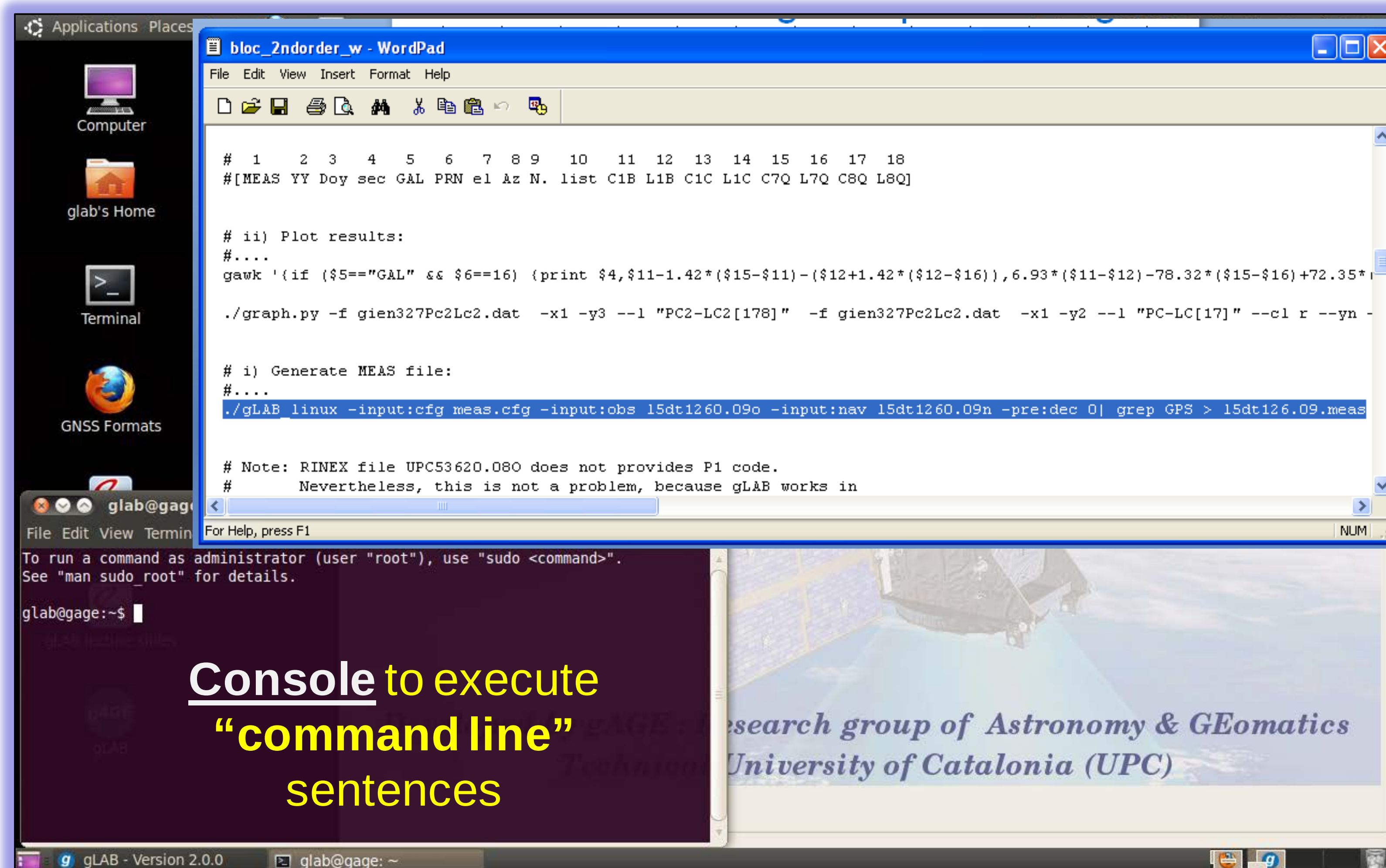
OVERVIEW

- **Introduction:** gLAB processing in command line
- **Preliminary computations:** data files & reference values
- **Session A:** Differential positioning of IND2-IND3 receivers
  (baseline: 18 metres)
- **Session B:** Differential positioning of IND1-IND2 receivers
  (baseline: 7 metres, but synchronization errors)
- **Session C:** Differential positioning of PLAN-GARR receivers
  (baseline: 15 km, Night time): tropospheric effects
- **Session D:** Differential positioning of PLAN-GARR receivers
  (baseline: 15 km, Day time): tropospheric and Ionospheric effects

OVERVIEW

- **Introduction: gLAB processing in command line**
- **Preliminary computations:** data files & reference values
- **Session A:** Differential positioning of IND2-IND3 receivers
  (baseline: 18 metres)
- **Session B:** Differential positioning of IND1-IND2 receivers
  (baseline: 7 metres, but synchronization errors)
- **Session C:** Differential positioning of PLAN-GARR receivers
  (baseline: 15 km, Night time): tropospheric effects
- **Session D:** Differential positioning of PLAN-GARR receivers
  (baseline: 15 km, Day time): tropospheric and Ionospheric effects

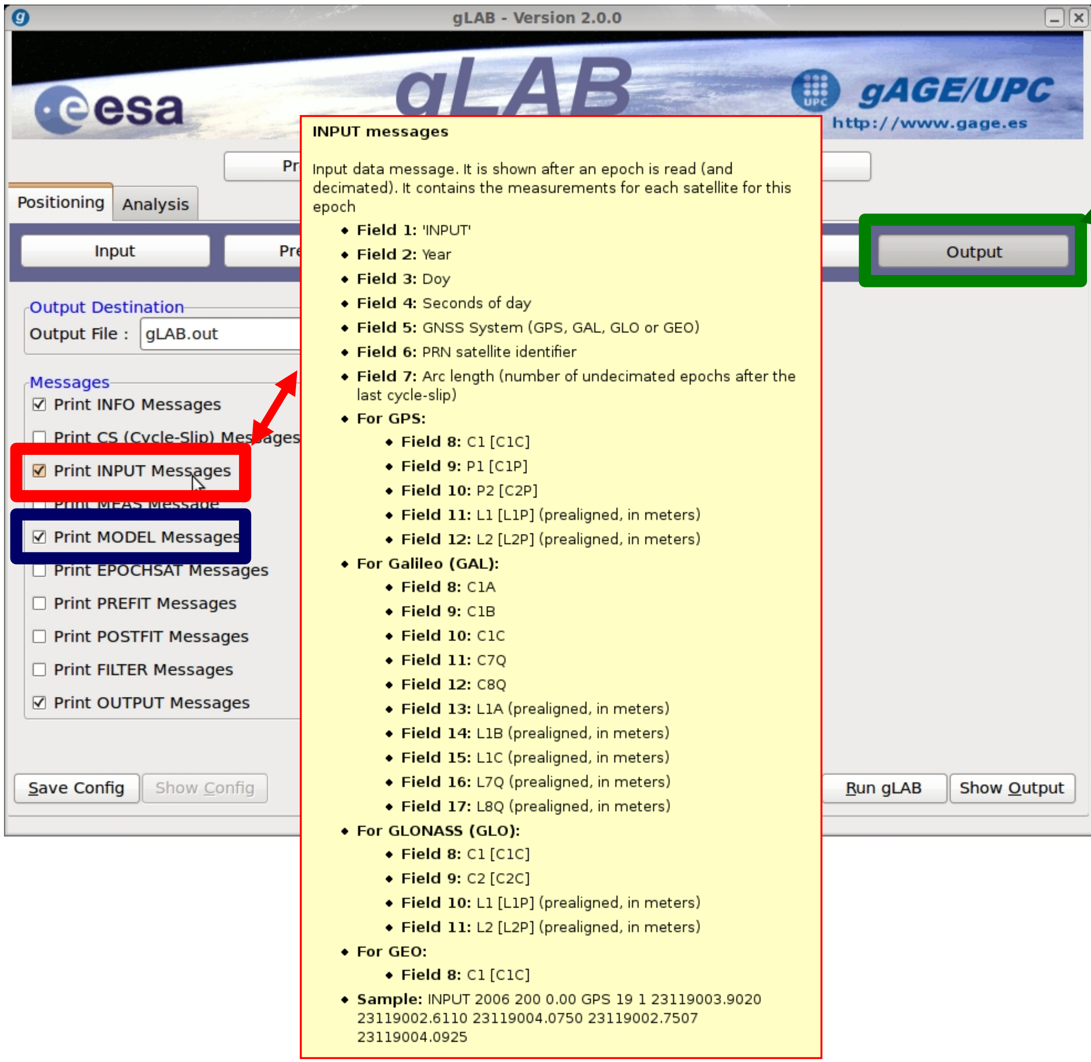
gLAB processing in command line



Console to execute **"command line"** sentences

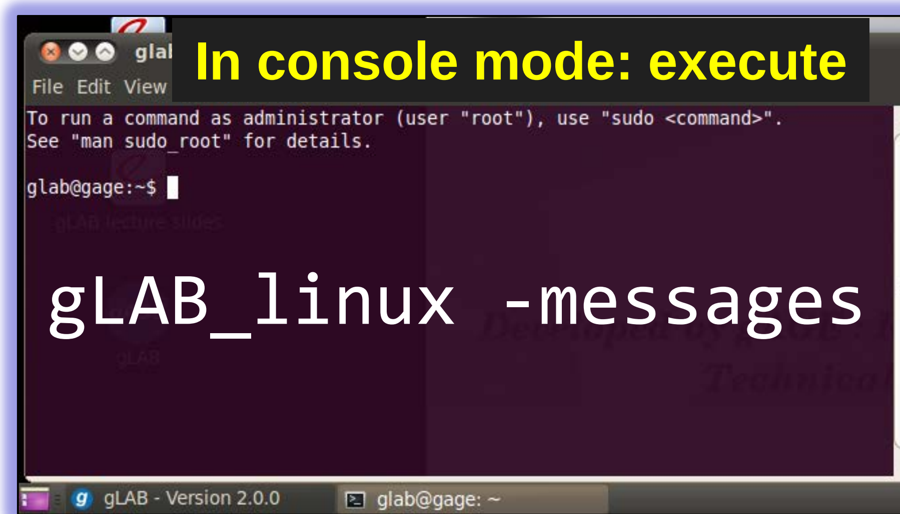
A **"notepad"** with the command line sentence is provided to facilitate the sentence writing: just **"copy" and " paste" from notepad to the working terminal.**

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
- **Preliminary computations:** data files & reference values
- **Session A:** Differential positioning of IND2-IND3 receivers

  (baseline: 18 metres)
- **Session B:** Differential positioning of IND1-IND2 receivers

  (baseline: 7 metres, but synchronization errors)
- **Session C:** Differential positioning of PLAN-GARR receivers

  (baseline: 15 km, Night time): tropospheric effects
- **Session D:** Differential positioning of PLAN-GARR receivers

  (baseline: 15 km, Day time): tropospheric and Ionospheric effects

Previous

Preliminary Computations

P. Preliminary computations

- This section is devoted to computing the reference values (receivers coordinates) and to preparing the data files to be used in the exercises.

- These data files will include the code and carrier measurements and the model components: geometric range, nominal troposphere and ionosphere corrections, satellite elevation and azimuth from each receiver…

- This data processing will be done with ***gLAB*** for each individual receiver.

- This preliminary processing will provide the baseline data files to perform computations easily using basic tools (such as **`awk`** for data files handling, to compute Double Differences of measurements) or using octave (MATLAB) scripts for the LAMBDA method implementation.

- Detailed **guidelines** for **self learning students** are provided in this tutorial and in its associated ***notepad*** text file.

P. Preliminary computations

P.1. Computation of reference values of receiver coordinates

Using gLAB and precise orbits and clocks, compute the PPP solution:

Note: the receivers were not moving (**static receivers**) during the data collection.

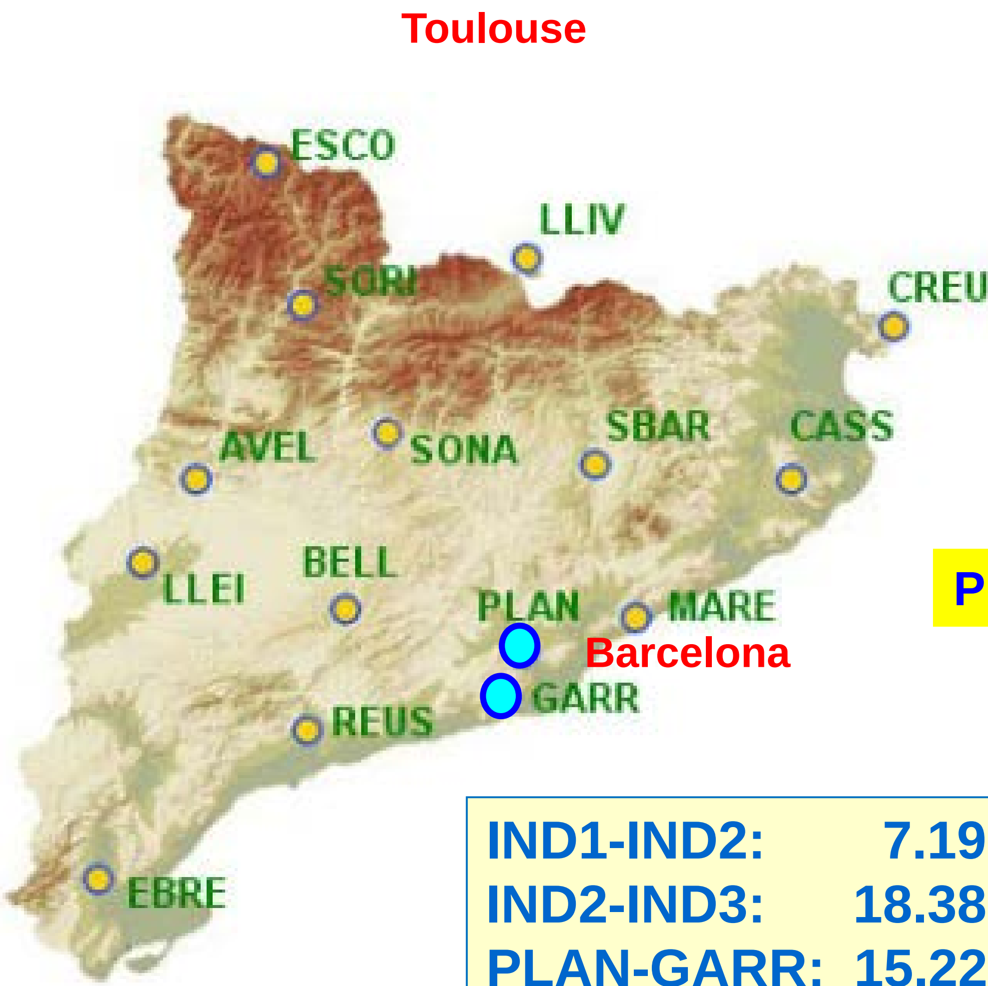
- Data files:
  - Measurements: PLAN0540.13O, GARR0540.13O, IND10540.13O, IND20540.13IO, IND30540.13O.
  - Orbits and clocks: brdc0540.13n, igs17286.sp3, igs17286.clk
  - ANTEX: igs08_1719.atx.
  - Configuration file (to compute LC APC coordinates): gLAB_2files_APC.cfg

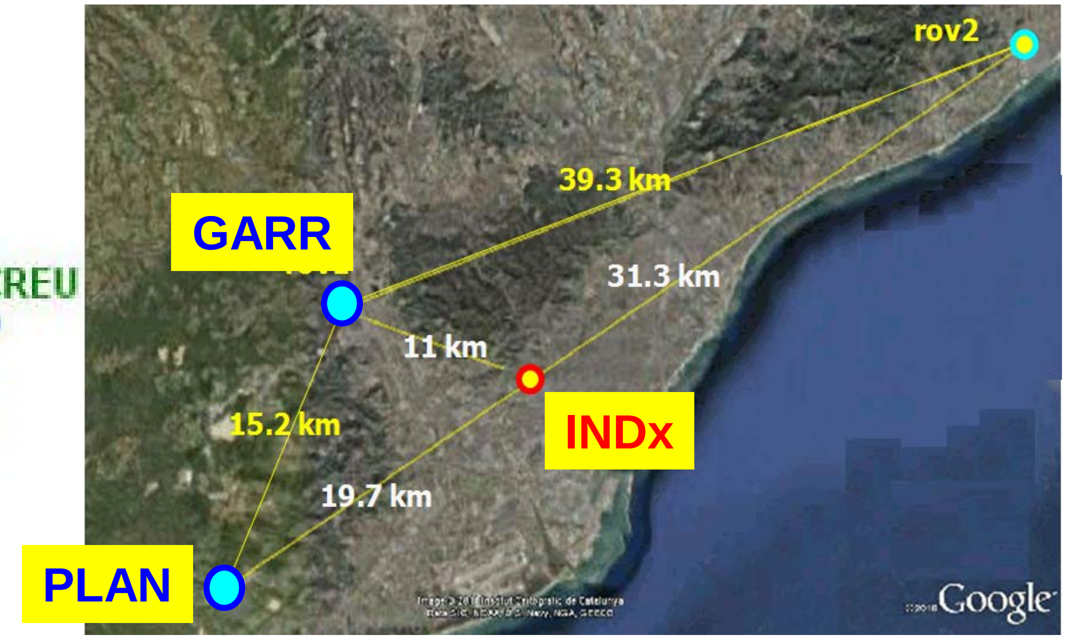
- Computation example:

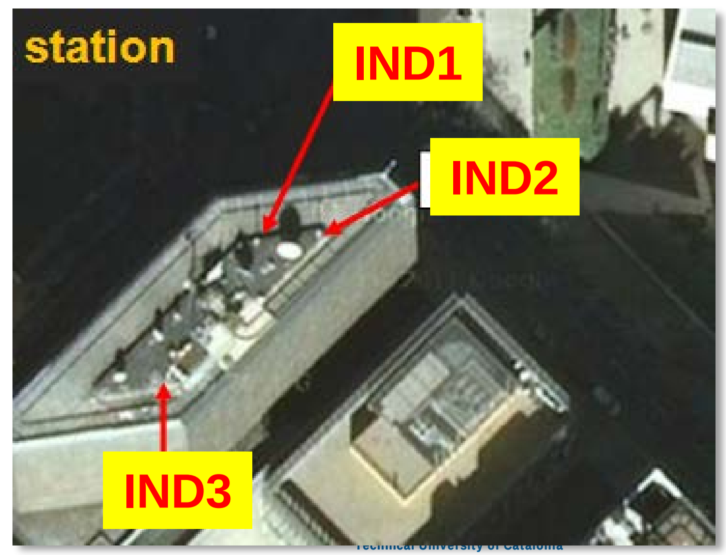
```
gLAB_linux -input:cfg gLAB_2files_APC.cfg  -input:obs PLAN0540.13O
           -input:orb igs17286.sp3 -input:clk igs17286.clk -input:ant igs08_1719.atx

grep OUTPUT gLAB.out | tail -1|gawk '{print "PLAN",$6,$7,$8,$15,$16,$17}' >> sta.pos
```

P. Preliminary computations







IND1-IND2:	**7.197 m**
IND2-IND3:	**18.380 m**
PLAN-GARR:	**15.228 km**

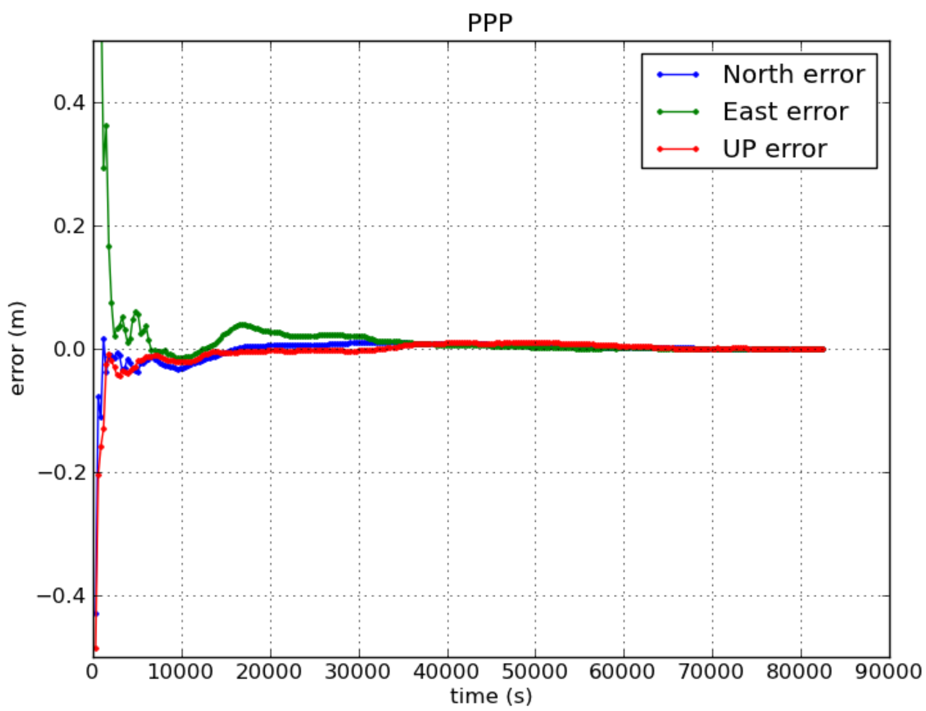
P.1. Computation of reference values of receiver coordinates

P. Preliminary computations

P.1. Computation of reference values of receiver coordinates

Plotting results:

```
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 -.5 --yx .5 --xl "time (s)" --yl "error (m)"  -t "PPP"
```



Note: the values of "APPROXIMATE COORDINATES written in RINEX files correspond to the precise APC of LC coordinates.

As a starting point, assume the same APC for L1 and LC

P. Preliminary computations

P.1. Computation of reference values of receiver coordinates

Plotting results:

```
more sta.pos

PLAN 4787328.7916 166086.0719 4197602.8893 41.418528940 1.986956885 320.0721
GARR 4796983.5170 160309.1774 4187340.3887 41.292941948 1.914040816 634.5682
IND1 4787678.1496 183409.7131 4196172.3056 41.403026173 2.193853893 109.5681
IND2 4787678.9809 183402.5915 4196171.6833 41.403018646 2.193768411 109.5751
IND3 4787689.5146 183392.8859 4196160.1653 41.402880392 2.193647610 109.5743
```

Question:
What is the expected accuracy of the computed coordinates?

P. Preliminary computations

P.1. Computation of reference values of receiver coordinates

Using octave (or MATLAB), compute the baseline length between the different receivers:

- Computation example:

```
octave

IND1=[ 4787678.1496 183409.7131 4196172.3056 ]
IND2=[ 4787678.9809 183402.5915 4196171.6833]

norm(IND1-IND2,2)
ans =  7.1969

exit
```

```
Results:

IND1-IND2:     7.197 m
IND2-IND3:    18.380 m
IND1-IND3:    23.658 m
PLAN-GARR:  15.228 km
PLAN-IND1:  17.386 km
IND1-GARR:  26.424 km
```

P. Preliminary computations

P.2. Model Components computation

- The script "**ObsFile.scr**" generates a data file with the following content

```
  1   2   3   4   5  6  7  8  9    10    11  12   13
[sta sat DoY sec P1 L1 P2 L2 Rho Trop Ion elev azim]
```

- Run this script for all receivers:

```
ObsFile.scr PLAN0540.13O brdc0540.13n
ObsFile.scr GARR0540.13O brdc0540.13n
ObsFile.scr IND10540.13O brdc0540.13n
ObsFile.scr IND20540.13O brdc0540.13n
ObsFile.scr IND30540.13O brdc0540.13n
```

- Merge all files into a single file:

```
cat ????.obs > ObsFile.dat
```

P. Preliminary computations

Selecting measurements: Time interval [14500:16500]

- To simplify computations, a time interval with always the same set of satellites in view and without cycle-slips is selected.
- Moreover an elevation mask of 10 degrees will be applied.

If the satellites change or cycle-slips appear during the data processing interval, care with the associated parameters handling must be taken in the navigation filter. Set up new parameters when new satellites appear and treat the ambiguities as constant between cycle-slips and white noise when a cycle-slip happens.

P. Preliminary computations

Selecting measurements: Time interval [14500:16500]

- Select the satellites with elevation over 10º in the time interval [14500:16500]

```
cat ObsFile.dat|gawk '{if ($4>=14500 && $4<=16500 && $12>10) print $0}' >
obs.dat
```

- Reference satellite (over the time interval [14500:16500])

Confirm that the satellite PRN06 is the satellite with the highest elevation
(this satellite will be used as the reference satellite)

obs.dat →

```
 1   2   3   4   5  6  7  8  9   10   11   12   13
[sta sat DoY sec P1 L1 P2 L2 Rho Trop Ion elev azim]
```

OVERVIEW

- **Introduction:** gLAB processing in command line
- **Preliminary computations:** data files & reference values
- **Session A:** Differential positioning of IND2-IND3 receivers
  (baseline: 18 metres)
- **Session B:** Differential positioning of IND1-IND2 receivers
  (baseline: 7 metres, but synchronization errors)
- **Session C:** Differential positioning of PLAN-GARR receivers
  (baseline: 15 km, Night time): tropospheric effects
- **Session D:** Differential positioning of PLAN-GARR receivers
  (baseline: 15 km, Day time): tropospheric and Ionospheric effects

Session A

Differential positioning of IND2- IND3 receivers

(baseline: 18 metres)

A. IND2- IND3 Differential positioning

A.1. **Double differences** between receivers and satellites computation

The script "**DDobs.scr**" computes the double differences between receivers and satellites from file **obs.dat**.

```
  1   2   3   4   5  6  7  8  9   10   11  12   13
[sta sat DoY sec P1 L1 P2 L2 Rho Trop Ion elev azim]
```

For instance, the following sentence:

```
DDobs.scr obs.dat IND2 IND3 06 03
```

generates the file

```
-------------------  DD_{sta1}_{sta2}_{sat1}_{sat2}.dat ---------------------------
   1    2    3    4    5   6    7    8    9   10    11    12     13    14  15  16  17
[sta1 sta2 sat1 sat2 DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2]
                                                                   <---- sta2 ---->
-----------------------------------------------------------------------------------
```

Where the elevation (EL) and azimuth (AZ) are taken from station #2.
and where (EL1, AZ1) are for satellite #1 and (EL1, AZ1) are for satellite #2.

A. IND2- IND3 Differential positioning

Compute the double differences between receivers **IND2 (reference)** and IND3 and satellites **PRN06 (reference)** and [PRN 03, 07,11, 16, 18, 19, 21, 22, 30]

```
DDobs.scr obs.dat IND2 IND3 06 03
DDobs.scr obs.dat IND2 IND3 06 07
DDobs.scr obs.dat IND2 IND3 06 11
DDobs.scr obs.dat IND2 IND3 06 16
DDobs.scr obs.dat IND2 IND3 06 18
DDobs.scr obs.dat IND2 IND3 06 19
DDobs.scr obs.dat IND2 IND3 06 21
DDobs.scr obs.dat IND2 IND3 06 22
DDobs.scr obs.dat IND2 IND3 06 30
```

Merge the files in a single file and sort by time:

```
cat DD_IND2_IND3_06_??.dat|sort -n -k +6 > DD_IND2_IND3_06_ALL.dat
```

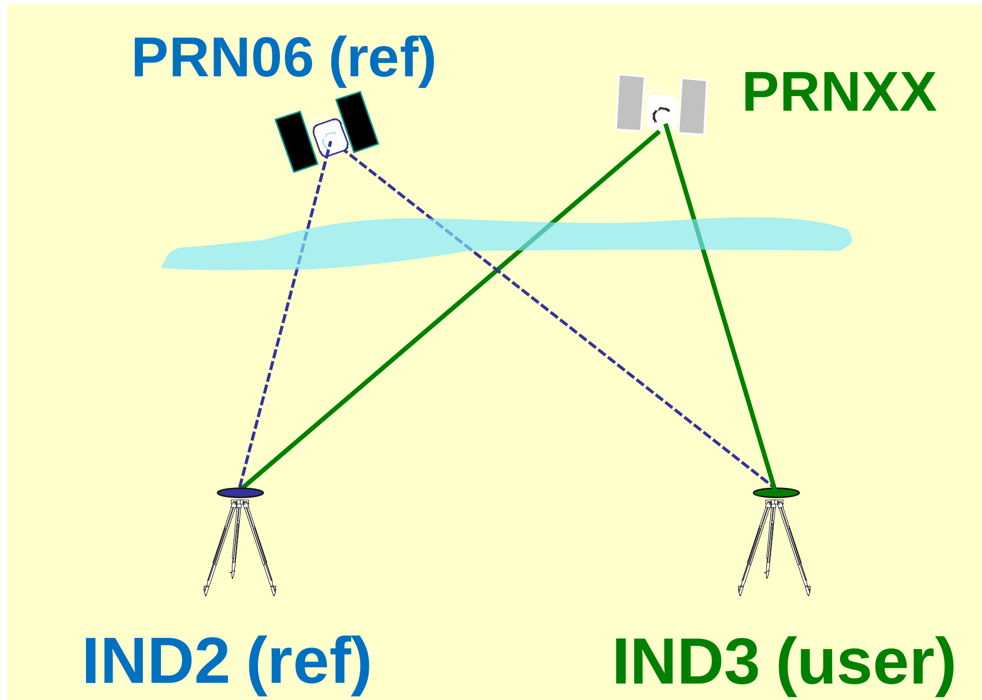
A. IND2- IND3 Differential positioning

```
------------------------------  DD_IND2_IND3_06_ALL.dat  ------------------------------
    1    2    3    4    5    6     7     8     9    10    11    12     13     14   15   16   17
[IND2 IND3 06 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2]
                                                                        <---- IND3 ---->
---------------------------------------------------------------------------------------
```

OUTPUT file

Where the elevation (EL) and azimuth (AZ) are taken from station **IND3** (the user)

and where, (EL1, AZ1) are for satellite PNR06 (reference) and (EL1, AZ1) are for satellite PRNXX



A.2. IND2-IND3 Baseline vector estimation with P1 code (using the time-tagged reference station measurements)

- In this exercise we will considerer an implementation of differential positioning where the user estimates the baseline vector using the time-tagged measurements of the reference station.

- This approach is **usually referred to as relative positioning** and can be applied in some applications where the coordinates of the reference station are not accurately known and where the relative position vector between the reference station and the user is the main interest. Examples are formation flying, automatic shipboard landing…

- Of course, the knowledge of the reference receiver location would allow the user to compute its absolute coordinates.

A.2. IND2-IND3 Baseline vector estimation with P1 code (using the time-tagged reference station measurements)

- This is a simple approach, but synchronism delays between the time tag measurements of the reference station and the user must be taken into account for real-time positioning.

- We will start positioning with the code C1 measurements, which is the simplest approach. Afterwards we will focus on positioning with L1 carrier by floating and fixing ambiguities.

- As the target is to perform differential positioning with carrier and carrier ambiguity fixing, we will work with double differences of measurements from the beginning (to have integer ambiguities), although these are not needed for code positioning.

A.2. IND2-IND3 Baseline vector estimation with P1 code (using the time-tagged reference station measurements)

Preliminary: Using octave (or MATLAB), and the receiver coordinates estimated before, **compute the baseline vector between IND2-IND3**. Give the results in the **ENU local sys**tem (at IND3).

```
IND2=[4787678.9809 183402.5915 4196171.6833]
IND3=[4787689.5146 183392.8859 4196160.1653]

IND3-IND2
      ans= 10.5337  -9.7056 -11.5180   (XYZ)
```

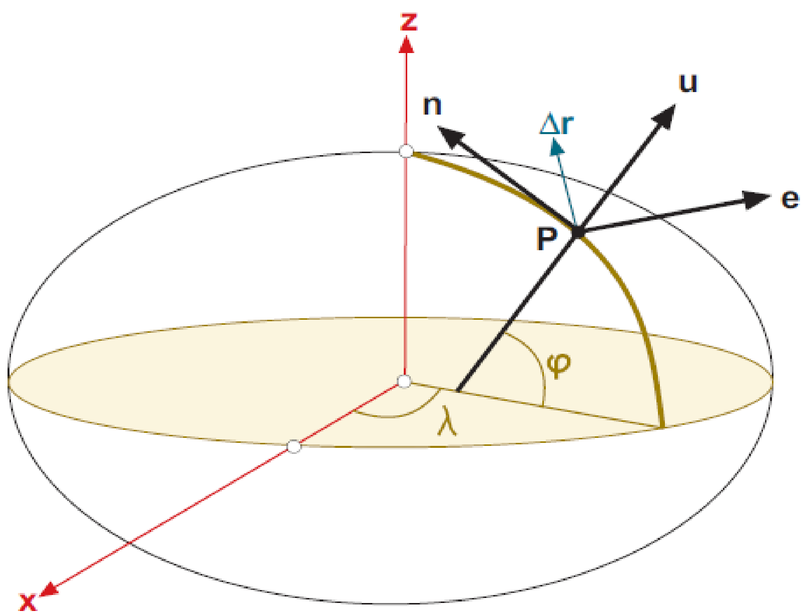
```
IND3 (lat and long):

 l=2.193647610*pi/180
 f=41.402880392*pi/180
```

```
R=[   -sin(l)             cos(l)      0   ;
   -cos(l)*sin(f)  -sin(l)*sin(f) cos(f);
    cos(l)*cos(f)   sin(l)*cos(f) sin(f)]

bsl_enu=R*(IND3-IND2)'
  ans   -10.1017 -15.3551 -0.0008 (ENU)
```

From ECEF (x,y,z) to Local (e,n,u) coordinates



$$\begin{bmatrix} \Delta \mathrm{e} \\ \Delta \mathrm{n} \\ \Delta \mathrm{u} \end{bmatrix} = \mathbf{R}_1[\pi/2 - \varphi]\, \mathbf{R}_3[\pi/2 + \lambda] \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}$$

$$\hat{\mathbf{e}} = (-\sin\lambda\,,\ \cos\lambda\,,\ 0)$$
$$\hat{\mathbf{n}} = (-\cos\lambda\sin\varphi\,,\ -\sin\lambda\sin\varphi\,,\ \cos\varphi)$$
$$\hat{\mathbf{u}} = (\cos\lambda\cos\varphi\,,\ \sin\lambda\ \cos\varphi\,,\ \sin\varphi)$$

$$\begin{bmatrix} \Delta \mathrm{e} \\ \Delta \mathrm{n} \\ \Delta \mathrm{u} \end{bmatrix} = \begin{bmatrix} -\sin\lambda & \cos\lambda & 0 \\ -\cos\lambda\sin\varphi & -\sin\lambda\sin\varphi & \cos\varphi \\ \cos\lambda\cos\varphi & \sin\lambda\cos\varphi & \sin\varphi \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}$$

A.2. IND2-IND3 Baseline vector estimation with P1 code (using the time-tagged reference station measurements)

A.2.1 Estimate the baseline vector between IND2 and IND3 receivers using the code measurements of file (DD_IND2_IND3_06_ALL.dat).

Note: Use the entire file (i.e. time interval [14500:16500]).

```
[DDP1]=[Los_k - Los_06]*[baseline]
```

Notation

$$\begin{bmatrix} DDP_1^{63} \\ DDP_1^{67} \\ \dots \\ DDP_1^{630} \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^3 - \hat{\boldsymbol{\rho}}^6\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7 - \hat{\boldsymbol{\rho}}^6\right)^T \\ \dots \\ -\left(\hat{\boldsymbol{\rho}}^{30} - \hat{\boldsymbol{\rho}}^6\right)^T \end{bmatrix} \mathbf{r}$$

$\mathbf{r} \equiv$ Baseline vector

$DDP_1^{k\,j} \equiv$ DDP1(involving satellites j and k)

$\hat{\boldsymbol{\rho}}^k \equiv$ Line-Of-Sight unit vector to satelite k

$$\hat{\boldsymbol{\rho}}^k \equiv \left[\cos(El_k)\sin(Az_k), \quad \cos(El_k)\cos(Az_k), \quad \sin(El_k)\right]$$

A.2.1. Baseline vector estimation with DDP1 (using with all epochs)

A.2. IND2-IND3 Baseline vector estimation with P1 code (using the time-tagged reference station measurements)

A.2.1 Estimate the baseline vector between IND2 and IND3 receivers using the code measurements of file (DD_IND2_IND3_06_ALL.dat).

Note: Use the entire file (i.e. time interval [14500:16500]).

```
[DDP1]=[Los_k - Los_06]*[baseline]
```

Notation

$$\begin{bmatrix} DDP_1^{63} \\ DDP_1^{67} \\ \dots \\ DDP_1^{630} \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^3 - \hat{\boldsymbol{\rho}}^6\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7 - \hat{\boldsymbol{\rho}}^6\right)^T \\ \dots \\ -\left(\hat{\boldsymbol{\rho}}^{30} - \hat{\boldsymbol{\rho}}^6\right)^T \end{bmatrix} \mathbf{r}$$

$DDP_1^{k\,j} \equiv$ DDP1(involving satellites j and k)

$$DDP_1^{k\,j} = DP_{1,usr}^{k\,j} - DP_{1,ref}^{k\,j} = \left(P_{1,usr}^{j} - P_{1,usr}^{k}\right) - \left(P_{1,ref}^{j} - P_{1,ref}^{k}\right)$$

$P_{1,ref}^{j}$ Measurements broadcast by the reference station.

A.2. IND2-IND3 Baseline vector estimation with P1 code (using the time-tagged reference station measurements)

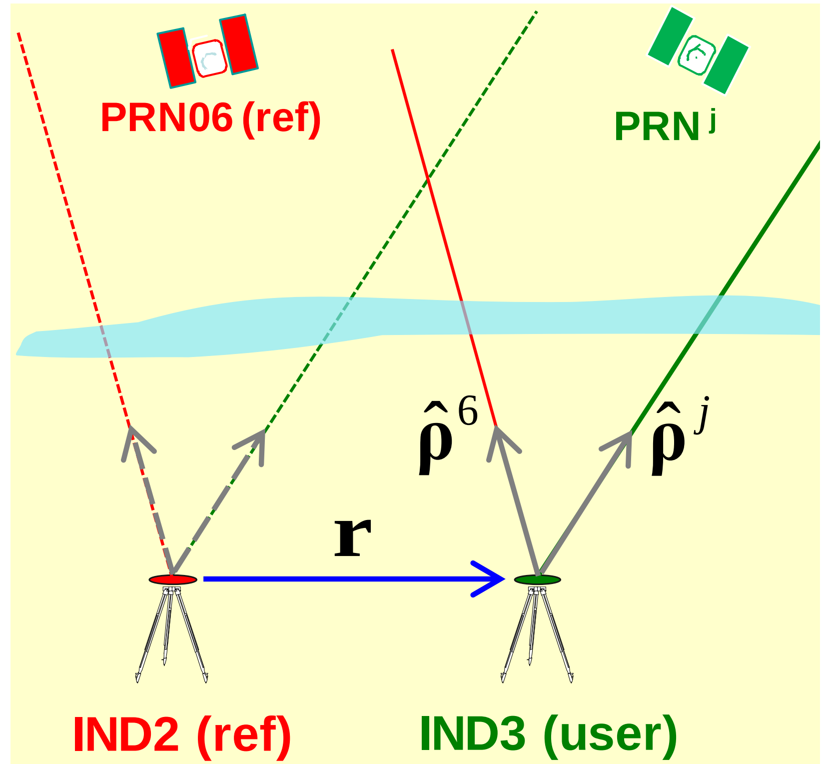
A.2.1 Estimate the baseline vector between IND2 and IND3 receivers using the code measurements of file (DD_IND2_IND3_06_ALL.dat).

Note: Use the entire file (i.e. time interval [14500:16500]).

```
[DDP1]=[Los_k-Los_06]*[baseline]
```

$$\begin{bmatrix} DDP_1^{63} \\ DDP_1^{67} \\ \cdots \\ DDP_1^{630} \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^3 - \hat{\boldsymbol{\rho}}^6\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7 - \hat{\boldsymbol{\rho}}^6\right)^T \\ \cdots \\ -\left(\hat{\boldsymbol{\rho}}^{30} - \hat{\boldsymbol{\rho}}^6\right)^T \end{bmatrix} \mathbf{r}$$

$$\hat{\boldsymbol{\rho}}^j \equiv \left[\cos(El_j)\sin(Az_j), \cos(El_j)\cos(Az_j), \sin(El_j)\right]$$



A.2.1. Baseline vector estimation with DDP1 (using with all epochs)

A.2. IND2-IND3 Baseline vector estimation with P1 code (using the time-tagged reference station measurements)

Justify that the next sentence builds the navigation equations system

See file content **in slide #21**

```
[DDP1]=[Los_k - Los_06]*[baseline]
```

```
cat DD_IND2_IND3_06_ALL.dat | gawk 'BEGIN{g2r=atan2(1,1)/45}
                    {e1=$14*g2r;a1=$15*g2r;e2=$16*g2r;a2=$17*g2r;
 printf "%14.4f %8.4f %8.4f %8.4f \n",
 $7, -cos(e2)*sin(a2)+cos(e1)*sin(a1),
     -cos(e2)*cos(a2)+cos(e1)*cos(a1), -sin(e2)+sin(e1)}' > M.dat
```

$$\begin{bmatrix} DDP_1^{63} \\ DDP_1^{67} \\ \cdots \\ DDP_1^{630} \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^3 - \hat{\boldsymbol{\rho}}^6\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7 - \hat{\boldsymbol{\rho}}^6\right)^T \\ \cdots \\ -\left(\hat{\boldsymbol{\rho}}^{30} - \hat{\boldsymbol{\rho}}^6\right)^T \end{bmatrix} \mathbf{r}$$

```
[DDP1]      [  Los_k - Los_06]
------      --------------------
-3.3762     0.3398 -0.1028 0.0714
-7.1131     0.1725  0.5972 0.0691
 4.3881    -0.6374  0.0227 0.2725
```

$$\hat{\boldsymbol{\rho}}^k \equiv \left[\cos(El_k)\sin(Az_k), \quad \cos(El_k)\cos(Az_k), \quad \sin(El_k)\right]$$

A.2. IND2-IND3 Baseline vector estimation with P1 code (using the time-tagged reference station measurements)

The receiver was not moving (static) during the data collection.

Thence, we can merge all the epochs in a single system to compute the static LS solution:

$$\begin{bmatrix} DDP_1^{6,3}(t_1) \\ DDP_1^{6,7}(t_1) \\ \cdots \\ DDP_1^{6,30}(t_1) \\ \cdots \\ DDP_1^{6,3}(t_n) \\ DDP_1^{6,7}(t_n) \\ \cdots \\ DDP_1^{6,30}(t_n) \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^3(t_1)-\hat{\boldsymbol{\rho}}^6(t_1)\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7(t_1)-\hat{\boldsymbol{\rho}}^6(t_1)\right)^T \\ \cdots \\ -\left(\hat{\boldsymbol{\rho}}^{30}(t_1)-\hat{\boldsymbol{\rho}}^6(t_1)\right)^T \\ \\ -\left(\hat{\boldsymbol{\rho}}^3(t_n)-\hat{\boldsymbol{\rho}}^6(t_n)\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7(t_n)-\hat{\boldsymbol{\rho}}^6(t_n)\right)^T \\ \cdots \\ -\left(\hat{\boldsymbol{\rho}}^{30}(t_n)-\hat{\boldsymbol{\rho}}^6(t_n)\right)^T \end{bmatrix} \mathbf{r}$$

$$\mathbf{y} = \mathbf{G}\,\mathbf{x}$$

Least Squares Solution

$$\mathbf{x} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{y}$$

$$\mathbf{P} = (\mathbf{G}^T\mathbf{G})^{-1}$$

```
[DDP1]=[Los_k - Los_06]*[baseline]
```

A.2. IND2-IND3 Baseline vector estimation with P1 code (using the time-tagged reference station measurements)

Solve the equations system using octave (or MATLAB) and assess the estimation error:

```
octave

load M.dat

y=M(:,1);
G=M(:,2:4);

x=inv(G'*G)*G'*y
x(1:3)'
-10.2909 -15.3856 -0.6511
```

```
bsl_enu =[-10.1017 -15.3551 -0.0008]
```

Estimation error:

```
x(1:3)-bsl_enu'
 -0.1891639885218108
 -0.0304617199913011
 -0.6502684114849081
```

A.2.1. Baseline vector estimation with DDP1 (using all epochs)

A.2. IND2-IND3 Baseline vector estimation with P1 code (using the time-tagged reference station measurements)

A.2.2. Repeat the previous computation, but using just the two epochs: $t_1=14500$ and $t_2=14515$.

- Selecting the two epochs:

```
cat DD_IND2_IND3_06_ALL.dat|gawk '{if ($6==14500||$6==14515) print $0}' >tmp.dat
```

- Building the equations system:

```
cat tmp.dat | gawk 'BEGIN{g2r=atan2(1,1)/45}
                    {e1=$14*g2r;a1=$15*g2r;e2=$16*g2r;a2=$17*g2r;
   printf "%14.4f %8.4f %8.4f %8.4f \n",
       $7, -cos(e2)*sin(a2)+cos(e1)*sin(a1),
           -cos(e2)*cos(a2)+cos(e1)*cos(a1), -sin(e2)+sin(e1)}' > M.dat
```


A.2.1. Baseline vector estimation with DDP1 (using all epochs)

A.2. IND2-IND3 Baseline vector estimation with P1 code (using the time-tagged reference station measurements)

Solving the equations system using octave (or MATLAB) and assessing the estimation error:

```
octave

load M.dat

y=M(:,1);
G=M(:,2:4);

x=inv(G'*G)*G'*y
x(1:3)'
-10.9525 -14.7363  -1.7780
```

```
bsl_enu =[-10.1017 -15.3551 -0.0008]

x(1:3)-bsl_enu'
   -0.850763748698302
    0.618803236835673
   -1.77716717481060 6
```

Questions:

1.- What is the level of accuracy?

2.- Why does the solution degrade when taking only two epochs?

A.2.1. Baseline vector estimation with DDP1 (using with all epochs)

A.3. IND2-IND3 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

A.3.1 Estimate the baseline vector between IND2 and IND3 receivers using the L1 carrier measurements of file (DD_IND2_IND3_06_ALL.dat).

Consider only the two epochs used in the previous exercise: t_1=14500 and t_2=14515

The following procedure can be applied:

1. **Compute the FLOATED solution**, solving the equations system with octave. Assess the accuracy of the floated solution.
2. **Apply the LAMBDA method to FIX the ambiguities**. Compare the results with the solution obtained by rounding directly the floated solution and by rounding the solution after decorrelation.
3. **Repair the DDL1 carrier measurements with the DDN1 FIXED** ambiguities and plot results to analyze the data.
4. **Compute the FIXED solution**.

A.3. IND2-IND3 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

A.3.1 Estimate the baseline vector between IND2 and IND3 receivers using the L1 carrier measurements of file (DD_IND2_IND3_06_ALL.dat).

```
[DDL1]= [Los_k - Los_06]*[baseline] + [ A ]*[lambda1*DDN1]
```

Notation (for each epoch t)

$$\begin{bmatrix} DDL_1^{6,3} \\ DDL_1^{6,7} \\ \dots \\ DDL_1^{6,30} \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^3 - \hat{\boldsymbol{\rho}}^6\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7 - \hat{\boldsymbol{\rho}}^6\right)^T \\ \dots \\ -\left(\hat{\boldsymbol{\rho}}^{30} - \hat{\boldsymbol{\rho}}^6\right)^T \end{bmatrix} \mathbf{r} + \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ & & \dots & \\ 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \, DDN_1^{6,3} \\ \lambda_1 \, DDN_1^{6,7} \\ \dots \\ \lambda_1 \, DDN_1^{6,30} \end{bmatrix}$$

$$\mathbf{y} = \mathbf{G}\,\mathbf{x}$$

Where the vector of unknowns $\mathbf{x}$ **includes the user coordinates and ambiguities**

A.3.1. Baseline vector estimation with DDL1 (using only two epochs)

A.3. IND2-IND3 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

The receiver was not moving (static) during the data collection.
Therefore, for each epoch we have the equations system:

$$\begin{bmatrix} DDL_1^{6,3}(t_1) \\ DDL_1^{6,7}(t_1) \\ \ldots \\ DDL_1^{6,30}(t_1) \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^3(t_1)-\hat{\boldsymbol{\rho}}^6(t_1)\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7(t_1)-\hat{\boldsymbol{\rho}}^6(t_1)\right)^T \\ \ldots \\ -\left(\hat{\boldsymbol{\rho}}^{30}(t_1)-\hat{\boldsymbol{\rho}}^6(t_1)\right)^T \end{bmatrix} \mathbf{r} + \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ & & \ldots & \\ 0 & 0 & \ldots & 1 \end{bmatrix} \begin{bmatrix} \lambda_1\, DDN_1^{6,3} \\ \lambda_1\, DDN_1^{6,7} \\ \ldots \\ \lambda_1\, DDN_1^{6,30} \end{bmatrix}$$

$$\mathbf{y}_1 = \mathbf{G}_1\,\mathbf{x}$$

y1:=y[t1]
G1:=G[t1]

$$\begin{bmatrix} DDL_1^{6,3}(t_2) \\ DDL_1^{6,7}(t_2) \\ \ldots \\ DDL_1^{6,30}(t_2) \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^3(t_2)-\hat{\boldsymbol{\rho}}^6(t_2)\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7(t_2)-\hat{\boldsymbol{\rho}}^6(t_2)\right)^T \\ \ldots \\ -\left(\hat{\boldsymbol{\rho}}^{30}(t_2)-\hat{\boldsymbol{\rho}}^6(t_2)\right)^T \end{bmatrix} \mathbf{r} + \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ & & \ldots & \\ 0 & 0 & \ldots & 1 \end{bmatrix} \begin{bmatrix} \lambda_1\, DDN_1^{6,3} \\ \lambda_1\, DDN_1^{6,7} \\ \ldots \\ \lambda_1\, DDN_1^{6,30} \end{bmatrix}$$

$$\mathbf{y}_2 = \mathbf{G}_2\,\mathbf{x}$$

y2:=y[t2]
G2:=G[t2]

```
[DDL1]=[Los_k - Los_06]*[baseline] + [ A ]*[lambda1*DDN1]
```

A.3. IND2-IND3 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

In the previous computation we have not taken into account the correlations between the double differences of measurements. This matrix will be used now, as the LAMBDA method will be applied to FIX the carrier ambiguities.

a) Show that the covariance matrix of DDL1 is given by $\mathbf{P_y}$

$$\mathbf{P_y} = 2\sigma^2 \begin{bmatrix} 2 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & 2 \end{bmatrix}$$

b) Given the measurement vectors (**y**) and Geometry matrices (**G**) for two epochs

y1:=y[t1] ; G1:=G[t1] ; Py

y2:=y[t2] ; G2:=G[t2] ; Py

show that the user solution and covariance matrix can be computed as:

P=inv(G1'*W*G1+G2'*W*G2);

where: **W=inv(Py)**

x=P*(G1'*W*y1+G2'*W*y2) ;

$$\mathbf{y = G\,x;} \quad \mathbf{W = P_y^{-1}}$$

$$\mathbf{x = (G^T W G)^{-1} G^T W y}$$

$$\mathbf{P = (G^T W G)^{-1}}$$

A.3.1. Baseline vector estimation with DDL1 (using only two epochs)

A.3. IND2-IND3 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

The script **MakeL1BslMat.scr** builds the equations system

```
[DDL1]=[ Los_k- Los_06]*[baseline] + [ A ]*[λ1*DDN1]
```

for the two epochs required t_1*=14500* and t_2*=14515*, using the input file **DD_IND2_IND3_06_ALL.dat** generated before.

Execute:

```
MakeL1BslMat.scr DD_IND2_IND3_06_ALL.dat 14500 14515
```

The **OUTPUT** are the files **M1.dat** and **M2.dat** associated with each epoch.

Where:
the columns of files **M.dat** are the vector **y** (first column) and Matrix **G** (next columns)

A.3. IND2-IND3 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

1. Computing the FLOATED solution (solving the equations system).

The following procedure can be applied

```
octave

load M1.dat
load M2.dat

y1=M1(:,1);
G1=M1(:,2:11);

y2=M2(:,1);
G2=M2(:,2:11);
Py=(diag(ones(1,7))+ones(7))*2e-4;
W=inv(Py);
```

```
P=inv(G1'*W*G1+G2'*W*G2);
x=P*(G1'*W*y1+G2'*W*y2);

x(1:3)'
-8.9463  -15.9102   -0.78636

bsl_enu =[-10.1017 -15.3551 -0.0008]

x(1:3)'-bsl_enu
ans= 1.1554  -0.555 -0.78556
```

A.3.1. Baseline vector estimation with DDL1 (using only two epochs)

A.3. IND2-IND3 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

2. Applying the LAMBDA method to FIX the ambiguities.

The following procedure can be applied **(justify the computations done)**
Compare the different results found:

```
octave

c=299792458;
f0=10.23e+6;
f1=154*f0;
lambda1=c/f1
 a=x(4:10)/lambda1;
 Q=P(4:10,4:10);
```

Decorrelation and integer LS search solution

```
[Qz,Zt,Lz,Dz,az,iZ] = decorrel (Q,a);
[azfixed,sqnorm] = lsearch (az,Lz,Dz,2);
afixed=iZ*azfixed;
sqnorm(2)/sqnorm(1)
ans =  3.3196897362350O
afixed(:,1)'
-8    20    -9    -8   -10     0    -8
```

Rounding directly the floated solution

```
round(a)'
-10    21    -4   -11    -4     5    -3
```

Rounding the decorrelated floated solution

```
afix=iZ*round(az);
-8    20    -9    -8   -10     0    -8
```

A.3.1. Baseline vector estimation with DDL1 (using only two epochs)

A.3. IND2-IND3 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

3. Repair the DDL1 carrier measurements with the DDN1 FIXED ambiguities and plot results to analyze the data.

```
octave
amb=lambda1*afixed(:,1);
save ambL1.dat amb
```

Using the previous file **ambL1.dat** and "DD_IND2_IND3_06_ALL.dat", generate a file with the following content:

```
----------------------------  DD_IND2_IND3_06_ALL.fixL1 ---------------------------------
   1     2    3    4    5    6     7     8     9    10     11    12     13     14   15   16   17      18
[IND2 IND3 06 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2  λ1*DDN1]
                                                                   <---- IND3 ---->
------------------------------------------------------------------------------------------
```

Note: This file is identical to file "DD_IND2_IND3_06_ALL.dat", but with the ambiguities added in the last field #18.

A.3.1. Baseline vector estimation with DDL1 (using only two epochs)

A.3. IND2-IND3 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

a) Generate a file with the satellite PRN number and the ambiguities:

```
grep -v \# ambL1.dat > na1
cat DD_IND2_IND3_06_ALL.dat|gawk '{print $4}'|sort -nu|gawk '{print $1,NR}’ >sat.lst
paste sat.lst na1 > sat.ambL1
```

b) Generate the "DD_IND2_IND3_06_ALL.fixL1" file:

```
cat DD_IND2_IND3_06_ALL.dat|
                  gawk 'BEGIN{for (i=1;i<1000;i++) {getline <"sat.ambL1";A[$1]=$3}}
  {printf "%s %02i %02i %s %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f
                                     %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f \n",
                 $1,$2,$3,$4,$5,$6,$7,$8,$9,$10,$11,$12,$13,$14,$15,$16,$17,A[$4]}' >
                                                              DD_IND2_IND3_06_ALL.fixL1
```


A.3.1. Baseline vector estimation with DDL1 (using only two epochs)

A.3. IND2-IND3 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

```
----------------------------  DD_IND2_IND3_06_ALL.fixL1 ------------------------------------
   1     2    3    4    5    6     7     8     9    10     11    12     13     14   15   16   17      18
[IND2 IND3 06 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2  λ1*DDN1]
                                                                  <---- IND3 ---->
-------------------------------------------------------------------------------------------
```

c) Make and discuss the following plots

```
graph.py -f DD_IND2_IND3_06_ALL.fixL1 -x6 -y'($8-$18-$11)'
       -so --yn -0.06 --yx 0.06 -l "(DDL1-lambda1*DDN1)-DDrho" --xl "time (s)" --yl "m"
```

```
graph.py -f DD_IND2_IND3_06_ALL.fixL1 -x6 -y'($8-$11)'
       -so --yn -5 --yx 5 -l "(DDL1)-DDrho" --xl "time (s)" --yl "metres"
```

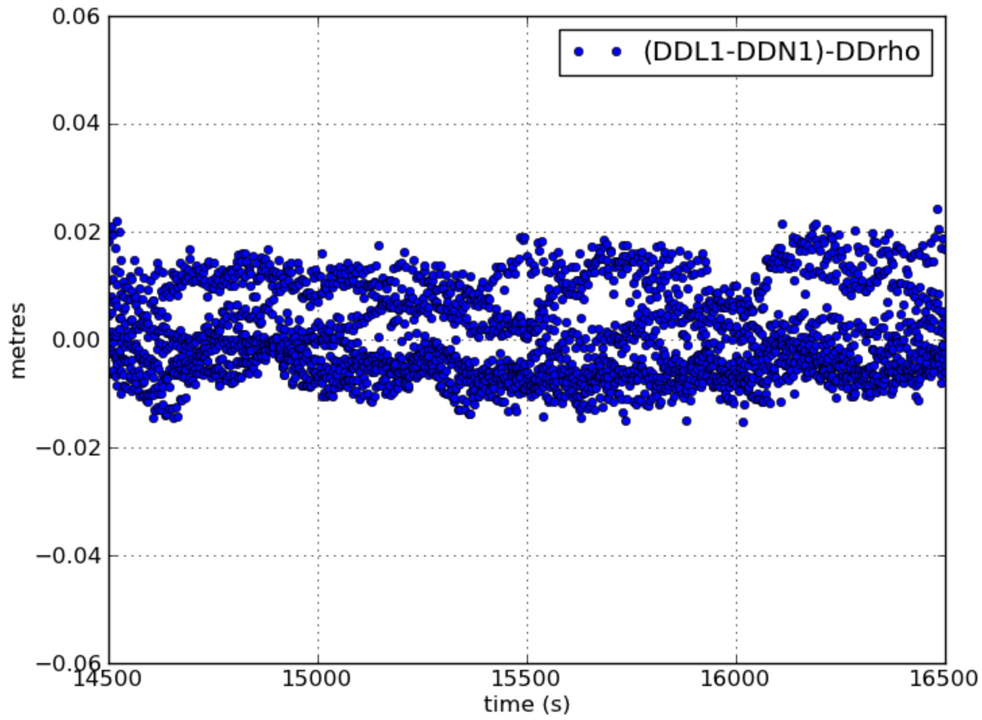
```
graph.py -f DD_IND2_IND3_06_ALL.fixL1 -x6 -y'($8-$18)'
       -so --yn -20 --yx 20 -l "(DDL1-lambda1*DDN1)" --xl "time (s)" --yl "metres"
```

A.3.1. Baseline vector estimation with DDL1 (using only two epochs)

A.3. IND2-IND3 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

```
----------------------------  DD_IND2_IND3_06_ALL.fixL1 --------------------------------
   1     2    3    4    5    6     7     8     9    10     11    12     13     14   15   16   17      18
[IND2 IND3 06 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2  λ1*DDN1]
                                                                         <---- IND3 ---->
-----------------------------------------------------------------------------------------
```

```
graph.py -f DD_IND2_IND3_06_ALL.fixL1
         -x6 -y'($8-$18-$11)'
         -so --yn -0.06 --yx 0.06
         -l "(DDL1-λ1*DDN1)-DDrho"
         --xl "time (s)" --yl "m"
```



Questions:
Explain what is the meaning of this plot.

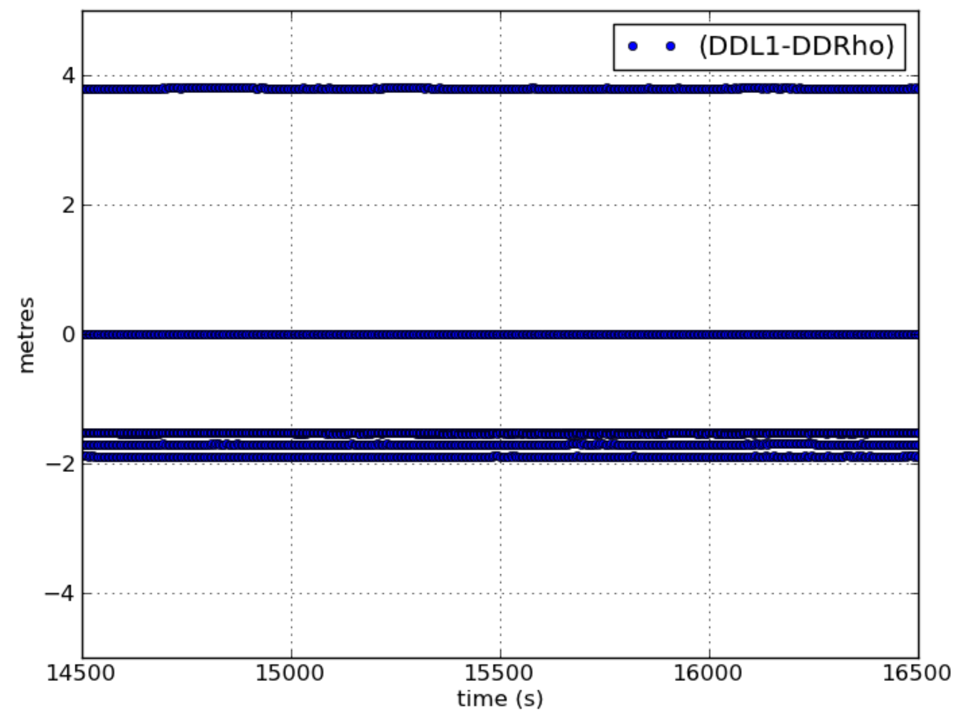
A.3.1. Baseline vector estimation with DDL1 (using only two epochs)

A.3. IND2-IND3 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

```
---------------------------- DD_IND2_IND3_06_ALL.fixL1 ------------------------------
   1    2   3   4   5   6    7    8    9    10    11    12     13     14  15  16  17     18
[IND2 IND3 06 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2  λ1*DDN1]
                                                                   <---- IND3 ---->
-------------------------------------------------------------------------------------
```

```
graph.py -f DD_IND2_IND3_06_ALL.fixL1
         -x6 -y'($8-$11)'
         -so --yn -5 --yx 5
         -l "DDL1-DDrho"
         --xl "time (s)" --yl "m"
```

Questions:
Explain what is the meaning of this plot.

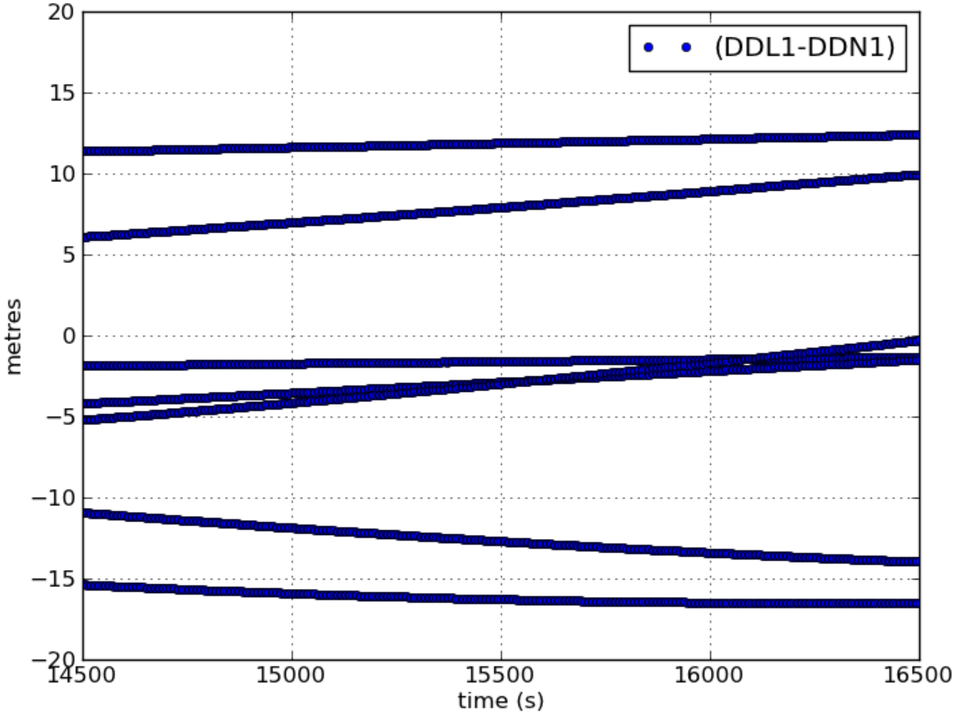


A.3.1. Baseline vector estimation with DDL1 (using only two epochs)

A.3. IND2-IND3 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

```
---------------------------- DD_IND2_IND3_06_ALL.fixL1 ------------------------------
   1    2   3   4   5   6    7    8    9   10    11    12     13    14  15  16  17      18
[IND2 IND3 06 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2  λ1*DDN1]
                                                                 <---- IND3 ---->
--------------------------------------------------------------------------------------
```

```
graph.py -f DD_IND2_IND3_06_ALL.fixL1
         -x6 -y'($8-$18)'
         -so --yn -20 --yx 20
         -l "(DDL1-λ1*DDN1)"
         --xl "time (s)" --yl "m"
```



Questions:

Explain what is the meaning of this plot.

A.3.1. Baseline vector estimation with DDL1 (using only two epochs)

A.3. IND2-IND3 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

1. Computing the FIXED solution (after FIXING ambiguities).

The following procedure can be applied

a) Build the equations system

```
[DDL1-lambda1*DDN1]=[Los_k - Los_06]*[baseline]
```

Note: it is the same system as with the code DDP1, but using "**DDL1-lambda1*DDN1**" instead of "**DDP1**"

```
cat DD_IND2_IND3_06_ALL.fixL1 | gawk 'BEGIN{g2r=atan2(1,1)/45}
                    {e1=$14*g2r;a1=$15*g2r;e2=$16*g2r;a2=$17*g2r;
 printf "%14.4f %8.4f %8.4f %8.4f \n",
  $8-$18, -cos(e2)*sin(a2)+cos(e1)*sin(a1),
         -cos(e2)*cos(a2)+cos(e1)*cos(a1), sin(e2)+sin(e1)}' > M.dat
```

A.3. IND2-IND3 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

Solve the equations system using octave (or MATLAB) and assess the estimation error:

```
octave

load M.dat

y=M(:,1);
G=M(:,2:4);

x=inv(G'*G)*G'*y
x(1:3)'
-10.1144 -15.3615 0.0031
```

```
bsl_enu =[-10.1017 -15.3551 -0.0008]
```

Estimation error:

```
x(1:3)-bsl_enu'
 -0.0127454057522200
 -0.0064270576494216
  0.0038663828567670
```

A.3. IND2-IND3 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

A.3.2. Using the DDL1 carrier with the ambiguities FIXED, compute the LS single epoch solution for the whole interval 145000< t <165000 with the program LS.f

Note: The program `LS.f` computes the Least Square solution for each measurement epoch of the input file (see the FORTRAN code `LS.f`)

The following procedure can be applied:

a) generate a file with the following content;

```
[Time], [DDL1-lambda1*DDN1], [ Los_k - Los_06]
```

where:

Time= seconds of day

DDL1-lambda1*DDN1= Prefit residulas (i.e., "y" values in program LS.f)

Los_k-Los_06 = The three components of the geometry matrix (i.e., matrix "a" in program LS.f)

A.3. IND2-IND3 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

```
[Time], [DDL1-lambda1*DDN1], [Los_k - Los_06]
```

The following sentence can be used

```
cat DD_IND2_IND3_06_ALL.fixL1 | gawk 'BEGIN{g2r=atan2(1,1)/45}
  {e1=$14*g2r;a1=$15*g2r;e2=$16*g2r;a2=$17*g2r;;printf "%s %14.4f
   %8.4f %8.4f %8.4f \n",$6,$8-$18,-cos(e2)*sin(a2)+cos(e1)*sin(a1),
        -cos(e2)*cos(a2)+cos(e1)*cos(a1),-sin(e2)+sin(e1)}' > L1model.dat
```

b) Compute the Least Squares solution

```
cat L1model.dat |LS > L1fix.pos
```

A.3. IND2-IND3 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

Plot the baseline estimation error

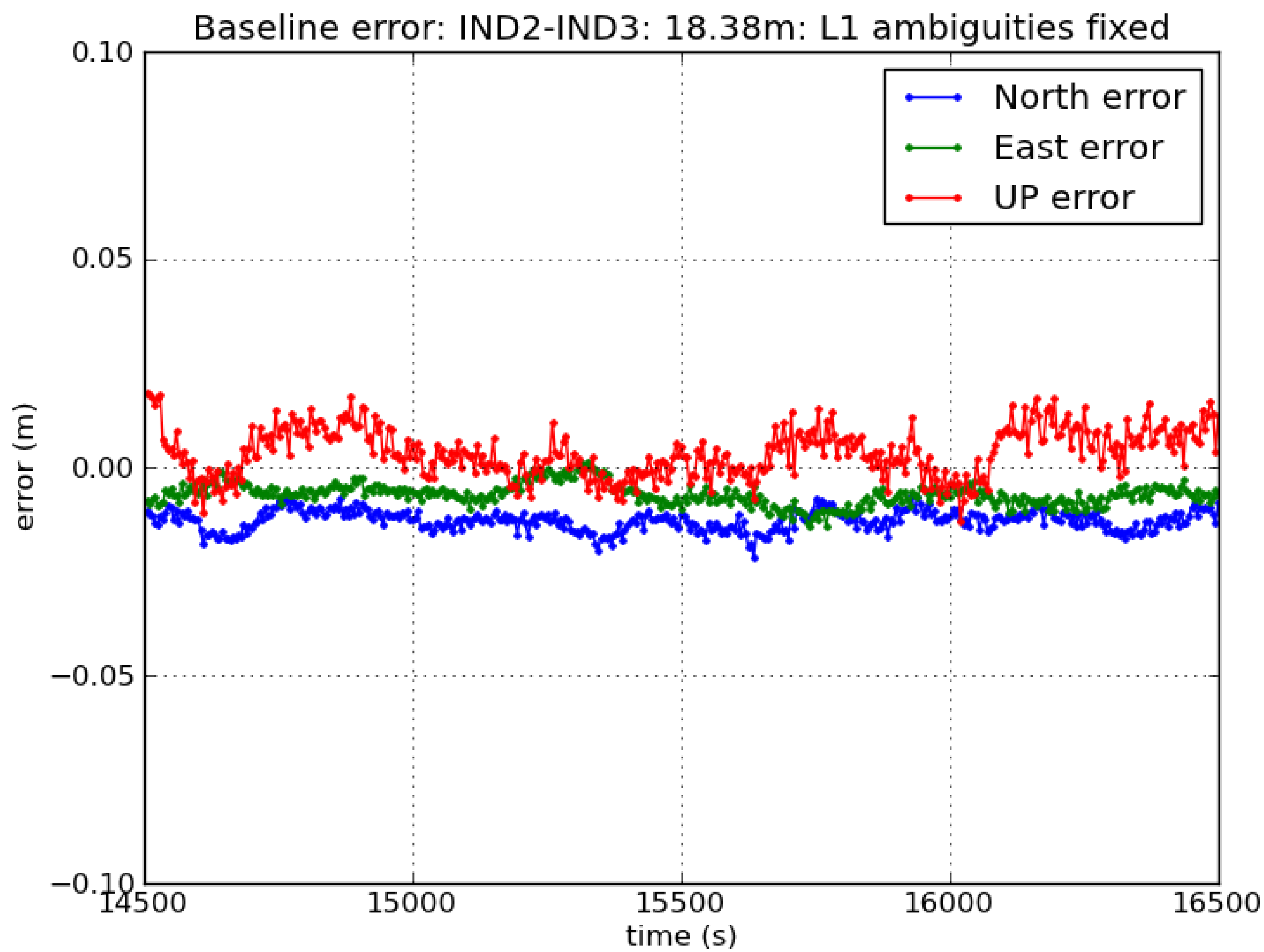
```
graph.py -f L1fix.pos -x1 -y'($2+10.1017)' -s.- -l "North error"
         -f L1fix.pos -x1 -y'($3+15.3551)' -s.- -l "East error"
         -f L1fix.pos -x1 -y'($4+0.0008)'  -s.- -l "UP error"
    --yn -.1 --yx .1 --xl "time (s)" --yl "error (m)" -t "Baseline error”
```

Note:

```
bsl_enu =[-10.1017 -15.3551 -0.0008]
```

A.3. IND2-IND3 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

Baseline estimation error after fixing ambiguities



A.3.2. Baseline vector estimation with DDL1 (single epoch LS, whole interval)

A.3. IND2-IND3 Baseline vector estimation (using the time-tagged reference station measurements)

A.3.3. Repeat previous computations, but using the Unsmoothed code P1.

i.e., compute the LS single epoch solution for the whole interval 145000< t <165000 with the program LS.f

The same procedure as in previous case can be applied, but using the code DDP1 instead of the carrier "DDL1-lambda1*DDN1"

a) generate a file with the following content;

```
[Time], [DDP1], [ Los_k - Los_06]
```

where:

Time= seconds of day

DDP1= Prefit residulas (i.e., "y" values in program lms1)

Los_k-Los_06 = The three components of the geometry matrix (i.e., matrix "a" in program LS.f)

A.3.3. Baseline vector estimation with DDP1 (single epoch LS, whole interval)

A.3. IND2-IND3 Baseline vector estimation (using the time-tagged reference station measurements)

```
[Time], [DDP1], [Los_k - Los_06]
```

The following sentence can be used

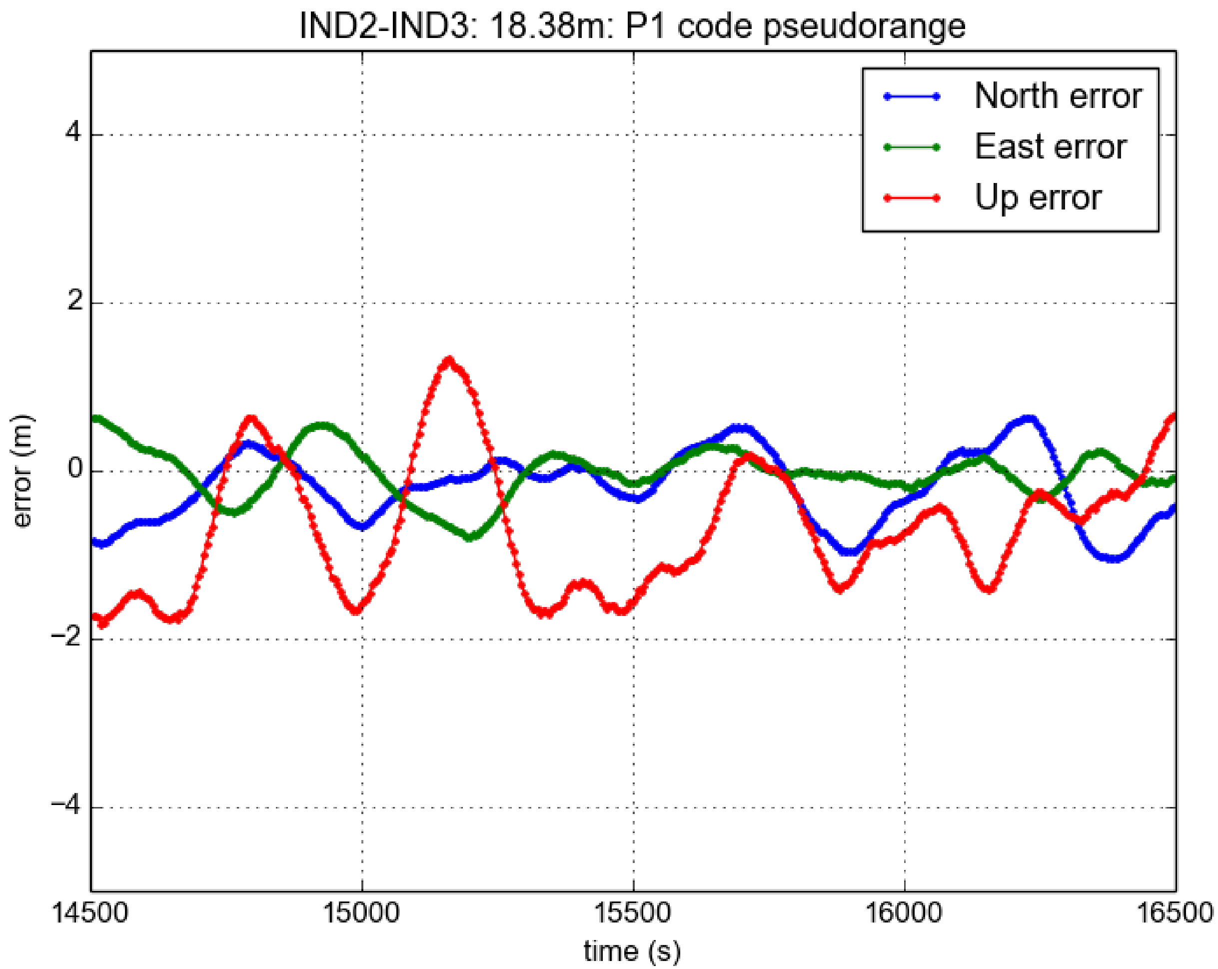
```
cat DD_IND2_IND3_06_ALL.fixL1 | gawk 'BEGIN{g2r=atan2(1,1)/45}
  {e1=$14*g2r;a1=$15*g2r;e2=$16*g2r;a2=$17*g2r;;printf "%s %14.4f
   %8.4f %8.4f %8.4f \n",$6,$7,-cos(e2)*sin(a2)+cos(e1)*sin(a1),
        -cos(e2)*cos(a2)+cos(e1)*cos(a1),-sin(e2)+sin(e1)}' > P1model.dat
```

b) Compute the Least Squares solution

```
cat P1model.dat |LS > P1.pos
```

A.3. IND2-IND3 Baseline vector estimation (using the time-tagged reference station measurements)

Baseline estimation error with the unsmoothed code



Questions:

1.- Discuss the results by comparing them with the previous ones with DDL1 carrier.

2.- Discuss the pattern seen in the plot.

A.4. IND2- IND3 Baseline vector estimation
Geometric diversity effect

Repeat the previous computations A.3., but using **two epochs more distant in time** : t_1=*14500* and t_2=*14600* (instead of t_2=*14515*).

Execute:

```
MakeL1BslMat.scr DD_IND2_IND3_06_ALL.dat 14500 14600
```

The **OUTPUT**
are the files **`M1.dat`** and **`M2.dat`** associated with each epoch.

Where:
the columns of files **`M.dat`** are the vector **`y`** (first column) and Matrix **`G`** (next columns)

A.4. IND2- IND3 Baseline vector estimation
Geometric diversity effect

Solving the equations system using octave (or MATLAB) and assessing the estimation error:

```
octave

load M1.dat
load M2.dat

y1=M1(:,1);
G1=M1(:,2:11);

y2=M2(:,1);
G2=M2(:,2:11);

W=inv(diag(ones(1,7))+ones(7))*2*1e-4;
P=inv(G1'*W*G1+G2'*W*G2);
x=P*(G1'*W*y1+G2'*W*y2);
10.9525 -14.7363  -1.7780
```

```
bsl_enu =[-10.1017 -15.3551 -0.0008]

x(1:3)-bsl_enu'
    0.331693264829917
    0.168847198925639
   -0.081327350481688
```

A.4. IND2- IND3 Baseline vector estimation
Geometric diversity effect

2. Applying the LAMBDA method to FIX the ambiguities.

The following procedure can be applied **(justify the computations done)**
Compare the different results found:

```
octave

c=299792458;
f0=10.23e+6;
f1=154*f0;
lambda1=c/f1
 a=x(4:10)/lambda1;
 Q=P(4:10,4:10);
```

Decorrelation and integer LS search solution

```
[Qz,Zt,Lz,Dz,az,iZ] = decorrel (Q,a);
[azfixed,sqnorm] = lsearch (az,Lz,Dz,2);
afixed=iZ*azfixed;
sqnorm(2)/sqnorm(1)
ans =  34.480193620474
afixed(:,1)'
-8   20   -9   -8  -10    0   -8
```

Rounding directly the floated solution

```
round(a)'
-8   19   -8   -9   -8    0   -9
```

Rounding the decorrelated floated solution

```
afix=iZ*round(az);
-8   20   -9   -8  -10    0   -8
```

A.3.1. Baseline estimation with L1
Geometry diversity effect.

A.4. IND2- IND3 Baseline vector estimation
Geometric diversity effect

Optional:

Repeat the computation taking t_1=*14500* **and** t_2=*15000*

Questions:

1.- Has the accuracy improved?

2.- Are the ambiguities well fixed?

3.- Has the reliability improved? Why?

A.5. IND2-IND3 differential positioning with P1 code (using the computed differential corrections)

- In the previous exercise we have considered an implementation of differential positioning where the user estimates the baseline vector from the time-tagged measurements of the reference station.

- In the next exercises, we will consider the common implementation of Differential positioning, where the reference receiver coordinates are accurately known and used to compute range corrections for each tracked satellite in view. Then, the user applies these corrections to improve the positioning.

- In the next example, a short baseline is processed (18 metres) and the range corrections are given as the measurements corrected by the geometric range. The differential atmospheric propagation errors can be assumed as zero for this very short baseline.

A.5.1. Differential positioning with DDP1 (using with all epochs)

A.5. IND2-IND3 differential positioning with P1 code (using the computed differential corrections)

- Unlike in the previous implementation, the synchronism errors between the time-tagged measurements will be not critical in this approach, as the differential corrections vary slowly.

- We will start positioning with the code C1 measurements, which is the simplest approach. Afterwards we will focus on positioning with L1 carrier by floating and fixing ambiguities.

- As the target is to perform differential positioning with carrier and carrier ambiguity fixing, we will work with double differences of measurements from the beginning (to have integer ambiguities), although they are not needed for code positioning.

A.5. IND2-IND3 differential positioning with P1 code (using the computed differential corrections)

A.5.1 Using code DDP1 measurements, estimate the coordinates of receiver IND3 taking IND2 as a reference receiver.

Justify that the associated equations system is given by:

```
[DDP1-DDRho]=[Los_k - Los_06]*[dr]
```

Notation

$$\begin{bmatrix} DDP_1^{6,3} - DD\rho^{6,3} \\ DDP_1^{6,7} - DD\rho^{6,7} \\ \dots \\ DDP_1^{6,30} - DD\rho^{6,30} \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^3 - \hat{\boldsymbol{\rho}}^6\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7 - \hat{\boldsymbol{\rho}}^6\right)^T \\ \dots \\ -\left(\hat{\boldsymbol{\rho}}^{30} - \hat{\boldsymbol{\rho}}^6\right)^T \end{bmatrix} \mathbf{dr}$$

$$\mathbf{dr} = \mathbf{r}_{IND3} - \mathbf{r}_{0,IND3}$$

$$DDP_1^{k\,j} \equiv \text{DDP1(involving satellites } j \text{ and } k\text{)}$$

$$\hat{\boldsymbol{\rho}}^k \equiv \text{Line-Of-Sight unit vector to satelite } k$$

$$\hat{\boldsymbol{\rho}}^k \equiv \left[\cos(El_k)\sin(Az_k) \quad \cos(El_k)\cos(Az_k) \quad \sin(El_k)\right]$$

A.5. IND2-IND3 differential positioning with P1 code (using the computed differential corrections)

A.5.1 Using code DDP1 measurements, estimate the coordinates of receiver IND3 taking IND2 as a reference receiver.

Justify that the associated equations system is given by:

```
[DDP1-DDRho]=[Los_k - Los_06]*[dr]
```

Notation

$$DDP_1^{k\,j} \equiv \text{DDP1(involving satellites } j \text{ and } k\text{)}$$

$$\begin{bmatrix} DDP_1^{6,3} - DD\rho^{6,3} \\ DDP_1^{6,7} - DD\rho^{6,7} \\ \cdots \\ DDP_1^{6,30} - DD\rho^{6,30} \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^3 - \hat{\boldsymbol{\rho}}^6\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7 - \hat{\boldsymbol{\rho}}^6\right)^T \\ \cdots \\ -\left(\hat{\boldsymbol{\rho}}^{30} - \hat{\boldsymbol{\rho}}^6\right)^T \end{bmatrix} \mathbf{dr}$$

$$\begin{aligned} DDP_1^{k\,j} - DD\rho^{k\,j} &= D\left(P_{1,usr}^{k\,j} - \rho_{usr}^{k\,j}\right) - D\left(P_{1,ref}^{k\,j} - \rho_{ref}^{k\,j}\right) \\ &= \left[\left(P_{1,usr}^{j} - \rho_{usr}^{j}\right) - \left(P_{1,usr}^{k} - \rho_{usr}^{k}\right)\right] - \left[\left(P_{1,ref}^{j} - \rho_{ref}^{j}\right) - \left(P_{1,ref}^{k} - \rho_{ref}^{k}\right)\right] \end{aligned}$$

$$PRC_1^j \equiv P_{1,ref}^j - \rho_{ref}^j$$

Computed corrections broadcast by the reference station.

A.5.1. Differential positioning with DDP1 (using with all epochs)

A.5. IND2-IND3 differential positioning with P1 code (using the computed differential corrections)

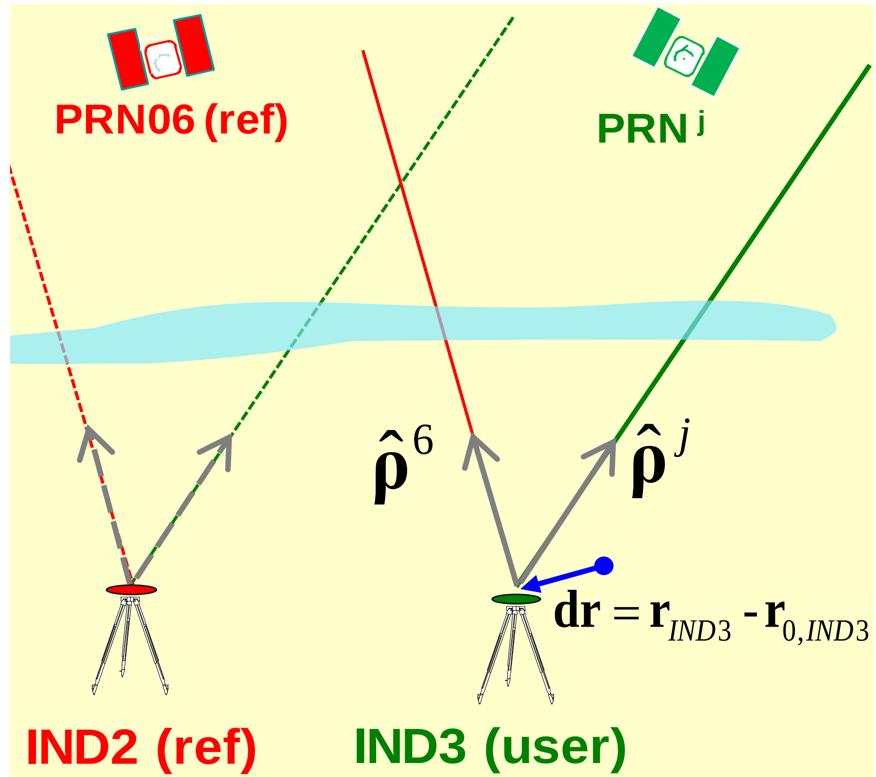
A.5.1 Using code DDP1 measurements, estimate the coordinates of receiver IND3 taking IND2 as a reference receiver.

Note: Use the entire file (i.e. time interval [14500:16500]).

```
[DDP1-DDRho]=[Los_k-Los_06]*[dr]
```

$$\begin{bmatrix} DDP_1^{6,3} - DD\rho^{6,3} \\ DDP_1^{6,7} - DD\rho^{6,7} \\ \dots \\ DDP_1^{6,30} - DD\rho^{6,30} \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^3 - \hat{\boldsymbol{\rho}}^6\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7 - \hat{\boldsymbol{\rho}}^6\right)^T \\ \dots \\ -\left(\hat{\boldsymbol{\rho}}^{30} - \hat{\boldsymbol{\rho}}^6\right)^T \end{bmatrix} \mathbf{dr}$$

$$\hat{\boldsymbol{\rho}}^j \equiv \left[\cos(El_j)\sin(Az_j), \cos(El_j)\cos(Az_j), \sin(El_j)\right]$$



A.5. IND2-IND3 differential positioning with P1 code (using the computed differential corrections)

Justify that the next sentence builds the navigation equations system

See file content **in slide #21**

```
[DDP1-DDRho]=[Los_k - Los_06]*[dr]
```

```
cat DD_IND2_IND3_06_ALL.dat | gawk 'BEGIN{g2r=atan2(1,1)/45}
                    {e1=$14*g2r;a1=$15*g2r;e2=$16*g2r;a2=$17*g2r;
 printf "%14.4f %8.4f %8.4f %8.4f \n",
  $7-$11, -cos(e2)*sin(a2)+cos(e1)*sin(a1),
          -cos(e2)*cos(a2)+cos(e1)*cos(a1), -sin(e2)+sin(e1)}' > M.dat
```

$$\begin{bmatrix} DDP_1^{6,3} - DD\rho^{6,3} \\ DDP_1^{6,7} - DD\rho^{6,7} \\ \cdots \\ DDP_1^{6,30} - DD\rho^{6,30} \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^3 - \hat{\boldsymbol{\rho}}^6\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7 - \hat{\boldsymbol{\rho}}^6\right)^T \\ \cdots \\ -\left(\hat{\boldsymbol{\rho}}^{30} - \hat{\boldsymbol{\rho}}^6\right)^T \end{bmatrix} \mathbf{dr}$$

```
[DDP1-DDRho]   [Los_k - Los_REF]
  -------     ---------------------
 -3.3762       0.3398 -0.1028 0.0714
 -7.1131       0.1725  0.5972 0.0691
  4.3881      -0.6374  0.0227 0.2725
```

$$\hat{\boldsymbol{\rho}}^k \equiv \left[\cos(El_k)\sin(Az_k) \quad \cos(El_k)\cos(Az_k) \quad \sin(El_k)\right]$$

A.5. IND2-IND3 differential positioning with P1 code (using the computed differential corrections)

The receiver was not moving (static) during the data collection.

Therefore, we can merge all the epochs in a single system to compute the static LS solution:

$$
\begin{bmatrix}
DDP_1^{6,3}(t_1) - DD\rho^{6,3}(t_1) \\
DDP_1^{6,7}(t_1) - DD\rho^{6,7}(t_1) \\
\dots \\
DDP_1^{6,30}(t_1) - DD\rho^{6,30}(t_1) \\
\\
DDP_1^{6,3}(t_n) - DD\rho^{6,3}(t_1) \\
DDP_1^{6,7}(t_n) - DD\rho^{6,7}(t_1) \\
\dots \\
DDP_1^{6,30}(t_n) - DD\rho^{6,30}(t_1)
\end{bmatrix}
=
\begin{bmatrix}
-\left(\hat{\boldsymbol{\rho}}^{3}(t_1) - \hat{\boldsymbol{\rho}}^{6}(t_1)\right)^T \\
-\left(\hat{\boldsymbol{\rho}}^{7}(t_1) - \hat{\boldsymbol{\rho}}^{6}(t_1)\right)^T \\
\dots \\
-\left(\hat{\boldsymbol{\rho}}^{30}(t_1) - \hat{\boldsymbol{\rho}}^{6}(t_1)\right)^T \\
\\
-\left(\hat{\boldsymbol{\rho}}^{3}(t_n) - \hat{\boldsymbol{\rho}}^{6}(t_n)\right)^T \\
-\left(\hat{\boldsymbol{\rho}}^{7}(t_n) - \hat{\boldsymbol{\rho}}^{6}(t_n)\right)^T \\
\dots \\
-\left(\hat{\boldsymbol{\rho}}^{30}(t_n) - \hat{\boldsymbol{\rho}}^{6}(t_n)\right)^T
\end{bmatrix}
\mathbf{dr}
\longrightarrow
\boxed{\mathbf{y} = \mathbf{G}\,\mathbf{x}}
$$

Least Squares Solution

$$
\mathbf{x} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{y}
$$

$$
\mathbf{P} = (\mathbf{G}^T\mathbf{G})^{-1}
$$

```
[DDPL1]=[Los_k - Los_06]*[baseline]
```

A.5. IND2-IND3 differential positioning with P1 code (using the computed differential corrections)

Solve the equations system using octave (or MATLAB) and assess the estimation error:

```
octave

load M.dat

y=M(:,1);
G=M(:,2:4);

x=inv(G'*G)*G'*y
x'
-0.1892  -0.0305  -0.6504
```

Absolute coordinates of IND3.

Taking into account that the ”a priori” coordinates of IND3 are:

```
IND3=[4787689.5146 183392.8859 4196160.1653 ]
```

Therefore the estimated absolute coordinates of IND3 are:

```
IND3+ x(1:3)'
 ans= 4787689.3254  183392.8554  4196159.5149
```

Note: as we have used the true coordinates of IND3 as the ”a priori” to linerize the model, the vector **x** provides the **estimation error** directly.

A.5.1. Differential positioning with DDP1 (using with all epochs)

A.5. IND2-IND3 differential positioning with P1 code (using the computed differential corrections)

A.5.2. Repeat the previous computation, but using just the two epochs: t_1=14500 and t_2=14515.

- Selecting the two epochs:

```
cat DD_IND2_IND3_06_ALL.dat|gawk '{if ($6==14500||$6==14515) print $0}' >tmp.dat
```

- Building the equations system:

```
cat tmp.dat | gawk 'BEGIN{g2r=atan2(1,1)/45}
                    {e1=$14*g2r;a1=$15*g2r;e2=$16*g2r;a2=$17*g2r;
 printf "%14.4f %8.4f %8.4f %8.4f \n",
 $7-$11, -cos(e2)*sin(a2)+cos(e1)*sin(a1),
         -cos(e2)*cos(a2)+cos(e1)*cos(a1),-sin(e2)+sin(e1)}' > M.dat
```


A.5.2. Estimate baseline vector with DDP1 (using only two epochs)

A.5. IND2-IND3 differential positioning with P1 code (using the computed differential corrections)

Solve the equations system using octave (or MATLAB) and assess the estimation error:

```
octave

load M.dat

y=M(:,1);
G=M(:,2:4);

x=inv(G'*G)*G'*y
x(1:3)'
-0.8509  0.6190 -1.7783
```

Absolute coordinates of IND3.

Taking into account that the ”a priory” coordinates of IND3 are:

```
IND3=[4787689.5146 183392.8859 4196160.1653 ]
```

Thence the estimated absolute coordinates of IND3 are:

```
IND3+ x(1:3)'
 ans= 4787688.6637  183393.5049  4196158.3870
```

Questions:

What is the level of accuracy?

Why does the solution degrade when taking only two epochs?

A.5.2. Differential positioning with DDP1 (using with all epochs)

A.6. IND2-IND3 differential positioning with L1 carrier (using the computed differential corrections)

A.6.1 Using DDL1 carrier measurements, estimate the coordinates of receiver IND3 taking IND2 as a reference receiver.

Consider only the two epochs used in the previous exercise: $t_1=14500$ and $t_2=14515$.

The following procedure can be applied:

1. **Compute the FLOATED solution**, solving the equations system with octave. Assess the accuracy of the floated solution.
2. **Apply the LAMBDA method to FIX the ambiguities**. Compare the results with the solution obtained by rounding directly the floated solution and by rounding the solution after decorrelation.
3. **Repair the DDL1 carrier measurements with the DDN1 FIXED** ambiguities and plot results to analyze the data.
4. **Compute the FIXED solution**.

A.6.1. Estimate IND3 coordinates with DDL1 (using only two epochs)

A.6. IND2-IND3 differential positioning with L1 carrier (using the computed differential corrections)

A.6.1 Estimate the coordinates of receiver IND3 taking IND2 as reference receiver, using the L1 carrier measurements of file (DD_IND2_IND3_06_ALL.dat)

```
[DDL1-DDRho]= [Los_k - Los_06]*[dr] + [ A ]*[lambda1*DDN1]
```

Notation

$$\begin{bmatrix} DDL_1^{6,3} - DD\rho^{6,3} \\ DDL_1^{6,7} - DD\rho^{6,7} \\ \cdots \\ DDL_1^{6,30} - DD\rho^{6,30} \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^3 - \hat{\boldsymbol{\rho}}^6\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7 - \hat{\boldsymbol{\rho}}^6\right)^T \\ \cdots \\ -\left(\hat{\boldsymbol{\rho}}^{30} - \hat{\boldsymbol{\rho}}^6\right)^T \end{bmatrix} \mathbf{dr} + \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \lambda_1\, DDN_1^{6,3} \\ \lambda_1\, DDN_1^{6,7} \\ \cdots \\ \lambda_1\, DDN_1^{6,30} \end{bmatrix}$$

$$\mathbf{y} = \mathbf{G}\,\mathbf{x}$$

Where the vector of unknowns $\underline{\mathbf{x}}$ **<u>includes the user coordinates and ambiguities</u>**

A.6. IND2-IND3 differential positioning with L1 carrier (using the computed differential corrections)

The receiver was not moving (static) during the data collection. Thence, for each epoch we have the equations system:

$$\begin{bmatrix} DDL_1^{6,3}(t_1)-DD\rho^{6,3}(t_1) \\ DDL_1^{6,7}(t_1)-DD\rho^{6,7}(t_1) \\ \cdots \\ DDL_1^{6,30}(t_1)-DD\rho^{6,30}(t_1) \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^3(t_1)-\hat{\boldsymbol{\rho}}^6(t_1)\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7(t_1)-\hat{\boldsymbol{\rho}}^6(t_1)\right)^T \\ \cdots \\ -\left(\hat{\boldsymbol{\rho}}^{30}(t_1)-\hat{\boldsymbol{\rho}}^6(t_1)\right)^T \end{bmatrix} \mathbf{dr} + \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 DDN_1^{6,3} \\ \lambda_1 DDN_1^{6,7} \\ \cdots \\ \lambda_1 DDN_1^{6,30} \end{bmatrix}$$

$$\mathbf{y}_1 = \mathbf{G}_1\,\mathbf{x}$$

y1:=y[t1]
G1:=G[t1]

$$\begin{bmatrix} DDL_1^{6,3}(t_2)-DD\rho^{6,3}(t_2) \\ DDL_1^{6,7}(t_2)-DD\rho^{6,7}(t_2) \\ \cdots \\ DDL_1^{6,30}(t_2)-DD\rho^{6,30}(t_2) \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^3(t_2)-\hat{\boldsymbol{\rho}}^6(t_2)\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7(t_2)-\hat{\boldsymbol{\rho}}^6(t_2)\right)^T \\ \cdots \\ -\left(\hat{\boldsymbol{\rho}}^{30}(t_2)-\hat{\boldsymbol{\rho}}^6(t_2)\right)^T \end{bmatrix} \mathbf{dr} + \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 DDN_1^{6,3} \\ \lambda_1 DDN_1^{6,7} \\ \cdots \\ \lambda_1 DDN_1^{6,30} \end{bmatrix}$$

$$\mathbf{y}_2 = \mathbf{G}_2\,\mathbf{x}$$

y2:=y[t2]
G2:=G[t2]

```
[DDL1-DDRho]=[Los_k - Los_06]*[dr] + [ A ]*[lambda1*DDN1]
```

A.6. IND2-IND3 differential positioning with L1 carrier (using the computed differential corrections)

In the previous computation we have not taken into account the correlations between the double differences of measurements. This matrix will be used now, as the LAMBDA method will be applied to FIX the carrier ambiguities.

a) Show that the covariance matrix of DDL1 is given by **$\mathbf{P_y}$**

$$\mathbf{P_y} = 2\sigma^2 \begin{bmatrix} 2 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & 2 \end{bmatrix}$$

b) Given the measurement vectors (y) and Geometry matrices (G) for two epochs

y1:=y[t1] ; G1:=G[t1] ; Py
y2:=y[t2] ; G2:=G[t2] ; Py

show that the user solution and covariance matrix can be computed as:

P=inv(G1'*W*G1+G2'*W*G2);

where: **W=inv(Py)**

x=P*(G1'*W*y1+G2'*W*y2) ;

$$\mathbf{y} = \mathbf{G}\,\mathbf{x}; \quad \mathbf{W} = \mathbf{P_y^{-1}}$$

$$\mathbf{x} = (\mathbf{G}^T\mathbf{W}\mathbf{G})^{-1}\mathbf{G}^T\mathbf{W}\mathbf{y}$$

$$\mathbf{P} = (\mathbf{G}^T\mathbf{W}\mathbf{G})^{-1}$$

A.6.1. Estimate IND3 coordinates with DDL1 (using only two epochs)

A.6. IND2-IND3 differential positioning with L1 carrier (using the computed differential corrections)

The script **MakeL1DifMat.scr** builds the equations system

$$[DDL1-DDRho]=[\ Los\_k-\ Los\_06]*[dr] + [\ A\ ]*[\lambda_1*DDN1]$$

for the two epochs required $t_1=14500$ and $t_2=14515$, using the input file **DD_IND2_IND3_06_ALL.dat** generated before.

Execute:

```
MakeL1DifMat.scr DD_IND2_IND3_06_ALL.dat 14500 14515
```

The **OUTPUT**

are the files **M1.dat** and **M2.dat** associated with each epoch.

Where:

the columns of files **M.dat** are the vector **y** (first column) and Matrix **G** (next columns)

A.6.1. Estimate IND3 coordinates with DDL1 (using only two epochs)

A.6. IND2-IND3 differential positioning with L1 carrier (using the computed differential corrections)

1. Computing the FLOATED solution (solving the equations system).

The following procedure can be applied

```
octave

load M1.dat
load M2.dat

y1=M1(:,1);
G1=M1(:,2:11);

y2=M2(:,1);
G2=M2(:,2:11);
Py=(diag(ones(1,7))+ones(7))*2e-4;
W=inv(Py);
```

```
P=inv(G1'*W*G1+G2'*W*G2);
x=P*(G1'*W*y1+G2'*W*y2);

x(1:3)'
0.9484  -0.3299  -0.8996
```

Taking into account that the ”a priori” coordinates of IND3 are: `IND3=[4787689.5146 183392.8859 4196160.1653]`

Thence the estimated absolute coordinates of IND3 are:

```
IND3+ x(1:3)'
4787690.4630 183392.5560  4196159.2657
```

A.6.1. Estimate IND3 coordinates with DDL1 (using only two epochs)

A.6. IND2-IND3 differential positioning with L1 carrier (using the computed differential corrections)

2. Applying the LAMBDA method to FIX the ambiguities.
Compare the results with the solution obtained by rounding the floated solution.
The following procedure can be applied **(justify the computations done)**

```
octave

c=299792458;
f0=10.23e+6;
f1=154*f0;
lambda1=c/f1
 a=x(4:10)/lambda1;
 Q=P(4:10,4:10);
```

Decorrelation and integer LS search solution

```
[Qz,Zt,Lz,Dz,az,iZ] = decorrel (Q,a);
[azfixed,sqnorm] = lsearch (az,Lz,Dz,2);
afixed=iZ*azfixed;
sqnorm(2)/sqnorm(1)
ans = 4.43344394778937
afixed(:,1)'
-8   20   -9   -8  -10    0   -8
```

Rounding directly the floated solution

```
round(a)'
-10   20   -4  -10   -5    4   -4
```

Rounding the decorrelated floated solution

```
afix=iZ*round(az);
-8   20   -9   -8  -10    0   -8
```

A.6.1. Estimate IND3 coordinates with DDL1 (using only two epochs)

A.6. IND2-IND3 differential positioning with L1 carrier (using the computed differential corrections)

Questions:

1. - Can the ambiguities be well fixed?

2. - Has the reliability improved? Why?

3. - The values found for the ambiguities are the same than in the previous case?

A.6.1. Estimate IND3 coordinates with DDL1 (using only two epochs)

A.6. IND2-IND3 differential positioning with L1 carrier (using the computed differential corrections)

3. Repair the DDL1 carrier measurements with the DDN1 FIXED ambiguities and plot results to analyze the data.

```
octave
amb=lambda1*afixed(:,1);
save ambL1.dat amb
```

Using the previous file **ambL1.dat** and "DD_IND2_IND3_06_ALL.dat", generate a file with the following content:

```
----------------------------  DD_IND2_IND3_06_ALL.fixL1 --------------------------------
   1     2    3    4    5    6     7     8     9    10     11    12     13     14   15   16   17      18
[IND2 IND3 06 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2  λ1*DDN1]
                                                                   <---- IND3 ---->
-----------------------------------------------------------------------------------------
```

Note: This file is identical to file "DD_IND2_IND3_06_ALL.dat", but with the ambiguities added in the last field #18.

A.6.1. Estimate IND3 coordinates with DDL1 (using only two epochs)

A.6. IND2-IND3 differential positioning with L1 carrier (using the computed differential corrections)

a) Generate a file with the satellite PRN number and the ambiguities:

```
grep -v \# ambL1.dat > na1
cat DD_IND2_IND3_06_ALL.dat|gawk '{print $4}'|sort -nu|gawk '{print $1,NR}' >sat.lst
paste sat.lst na1 > sat.ambL1
```

b) Generate the "DD_IND2_IND3_06_ALL.fixL1" file:

```
cat DD_IND2_IND3_06_ALL.dat|
                  gawk 'BEGIN{for (i=1;i<1000;i++) {getline <"sat.ambL1";A[$1]=$3}}
  {printf "%s %02i %02i %s %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f
                                      %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f \n",
                 $1,$2,$3,$4,$5,$6,$7,$8,$9,$10,$11,$12,$13,$14,$15,$16,$17,A[$4]}' >
                                                                 DD_IND2_IND3_06_ALL.fixL1
```

The ambiguities do not change. Therefore, the file DD_IND2_IND3_06_ALL.fixL1 generated in the previous exercise can be used here.

A.6.1. Estimate IND3 coordinates with DDL1 (using only two epochs)

A.6. IND2-IND3 differential positioning with L1 carrier (using the computed differential corrections)

4. Computing the FIXED solution (after FIXING ambiguities).

The following procedure can be applied

a) Build the equations system

```
[DDL1-DDRho-lambda1*DDN1]=[Los_k - Los_06]*[dr]
```

Note: this is the same system as with the code DDP1, but using "**DDL1-DDRho-lambda1*DDN1**" instead of "**DDP1**"

```
cat DD_IND2_IND3_06_ALL.fixL1 | gawk 'BEGIN{g2r=atan2(1,1)/45}
                    {e1=$14*g2r;a1=$15*g2r;e2=$16*g2r;a2=$17*g2r;
 printf "%14.4f %8.4f %8.4f %8.4f \n",
  $8-$11-$18, -cos(e2)*sin(a2)+cos(e1)*sin(a1),
          -cos(e2)*cos(a2)+cos(e1)*cos(a1), -sin(e2)+sin(e1)}' > M.dat
```

A.6.1. Estimate IND3 coordinates with DDL1 (using only two epochs)

A.6. IND2-IND3 differential positioning with L1 carrier (using the computed differential corrections)

Solve the equations system using octave (or MATLAB) and assess the estimation error:

```
octave

load M.dat

y=M(:,1);
G=M(:,2:4);

x=inv(G'*G)*G'*y
x
  -0.01278982304138015
  -0.00641700591386930
   0.00369003097108713
```

Absolute coordinates of IND3.

Taking into account that the ”a priori” coordinates of IND3 are:

```
IND3=[4787689.5146 183392.8859 4196160.1653 ]
```

Therefore the estimated absolute coordinates of IND3 are:

```
IND3+ x(1:3)'
 4787689.5018  183392.8795  4196160.1690
```

Question:
Is the accuracy similar to that in the previous case, when estimating the baseline vector?

A.6.1. Estimate IND3 coordinates with DDL1 (using only two epochs)

A.6. IND2-IND3 differential positioning with L1 carrier (using the computed differential corrections)

A.6.2. Using the DDL1 carrier with the ambiguities FIXED, compute the LS single epoch solution for the whole interval 145000< t <165000 with the program LS.f

Note: The program "LS.f" computes the Least Square solution for each measurement epoch of the input file (see the FORTRAN code "LS.f")

The following procedure can be applied

a) generate a file with the following content;

```
[Time], [DDL1-DDRho-lambda1*DDN1], [ Los_k - Los_06]
```

where:

Time= seconds of day

DDL1-DDRho-lambda1*DDN1= Prefit residulas (i.e., "y" values in program LS.f)

Los_k-Los_06 = The three components of the geometry matrix (i.e., matrix "a" in program LS.f)

A.6.2. Estimate IND3 coordinates with DDL1 (single epoch LS, whole interval)

A.6. IND2-IND3 differential positioning with L1 carrier (using the computed differential corrections)

```
[Time], [DDL1-DDRho-lambda1*DDN1], [Los_k - Los_06]
```

The following sentence can be used

```
cat DD_IND2_IND3_06_ALL.fixL1 | gawk 'BEGIN{g2r=atan2(1,1)/45}
  {e1=$14*g2r;a1=$15*g2r;e2=$16*g2r;a2=$17*g2r;;printf "%s %14.4f
   %8.4f %8.4f %8.4f \n",$6,$8-$11-$18,-cos(e2)*sin(a2)+cos(e1)*sin(a1),
        -cos(e2)*cos(a2)+cos(e1)*cos(a1),-sin(e2)+sin(e1)}' > L1model.dat
```

b) Compute the Least Squares solution

```
cat L1model.dat |LS > L1fix.pos
```

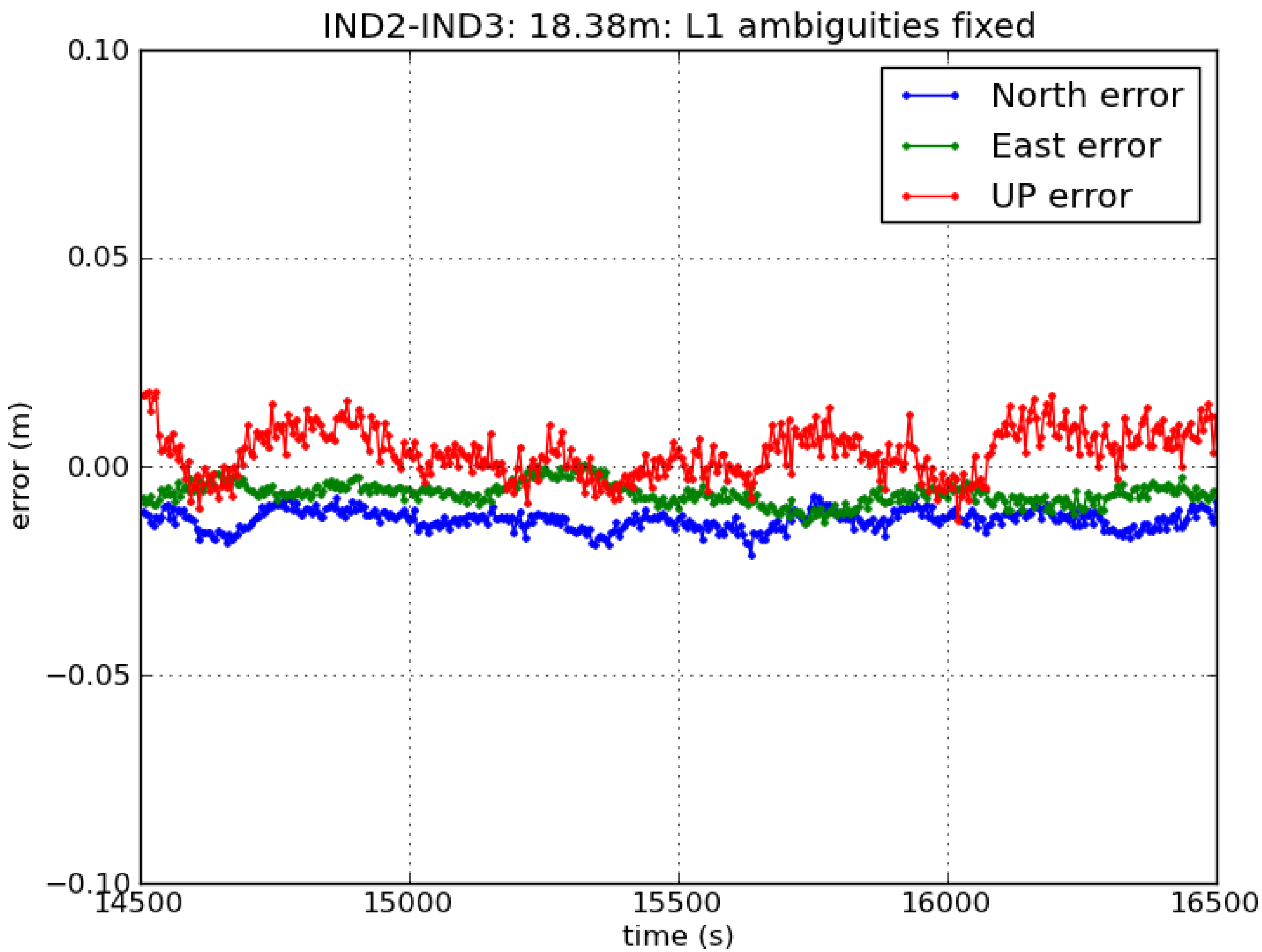
A.6. IND2-IND3 differential positioning with L1 carrier (using the computed differential corrections)

Plot the baseline estimation error

```
graph.py -f L1fix.pos -x1 –y2 -s.- -l "North error"
         -f L1fix.pos -x1 –y3 -s.- -l "East error"
         -f L1fix.pos -x1 –y4 -s.- -l "UP error"
  --yn -.1 --yx .1 --xl "time (s)" --yl "error (m)" -t “IND2-IND3: 18.38m:
L1 ambiguities fixed”
```

A.6. IND2-IND3 differential positioning with L1 carrier (using the computed differential corrections)

Differential Positioning error after fixing ambiguities



Question:
compare this plot with that obtained previously when estimating the baseline from the time-tagged measurements. Are the errors similar?

OVERVIEW

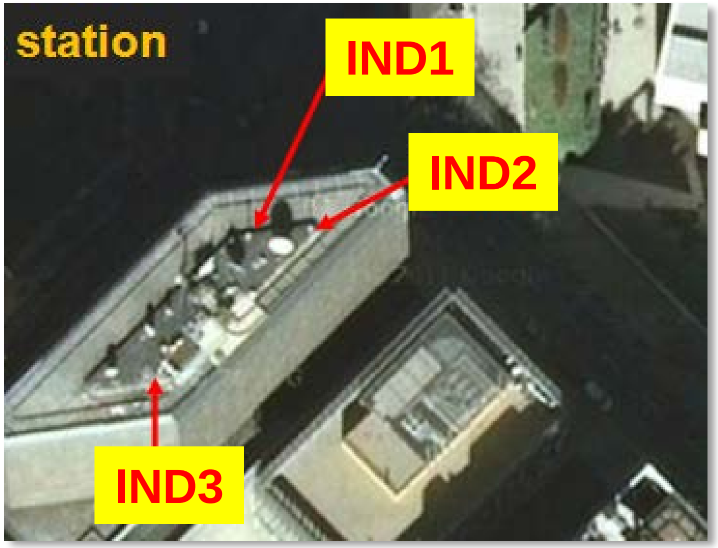
- **Introduction:** gLAB processing in command line
- **Preliminary computations:** data files & reference values
- **Session A:** Differential positioning of IND2-IND3 receivers (baseline: 18 metres)
- **Session B:** Differential positioning of IND1-IND2 receivers (baseline: 7 metres, but synchronization errors)
- **Session C:** Differential positioning of PLAN-GARR receivers (baseline: 15 km, Night time): tropospheric effects
- **Session D:** Differential positioning of PLAN-GARR receivers (baseline: 15 km, Day time): tropospheric and Ionospheric effects

Session B

Differential positioning of IND1- IND2 receivers
(baseline 7 metres and
not synchronised receivers)

B. Differential positioning of IND1- IND2 receivers

- The same exercises as in previous session will be repeated here for IND1 and IND2 receivers.
- These receivers are located in the same environment as IND2 and IND3 and the baseline is even shorter (7 metres, instead of 18 metres).
- The main difference in the receiver clock offset:
  - The receivers IND2 and IND3 apply clock steering and have a very short clock offset (just a tenth of nanoseconds), while the receiver IND1 has a large clock offset drift, accumulating up to 1 ms.
  - The effect of the synchronization errors on the two different implementations of differential positioning used in the previous session is one of the targets of this laboratory session.



IND1-IND2:	7.197 m
IND2-IND3:	18.380 m

B. IND1- IND2 Differential positioning

B.1. Double differences between receivers and satellites computation

The script "**DDobs.scr**" computes double differences between receivers and satellites from file **`obs.dat`**.

```
  1   2   3   4   5  6  7  8  9   10   11  12   13
[sta sat DoY sec P1 L1 P2 L2 Rho Trop Ion elev azim]
```

For instance, the following sentence:

```
DDobs.scr obs.dat IND1 IND2 06 03
```

generates the file

```
-------------------  DD_{sta1}_{sta2}_{sat1}_{sat2}.dat ---------------------------
   1    2    3    4   5   6    7    8    9   10    11    12     13    14  15  16  17
[sta1 sta2 sat1 sat2 DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2]
                                                                   <---- sta2 ---->
-----------------------------------------------------------------------------------
```

Where the elevation (EL) and azimuth (AZ) are taken from station #2.
and where (EL1, AZ1) are for satellite #1 and (EL1, AZ1) are for satellite #2.

B. IND1- IND2 Differential positioning

Compute the double differences between receivers **IND1 (reference)** and IND2 and satellites **PRN06 (reference)** and [PRN 03, 07,11, 16, 18, 19, 21, 22, 30]

```
DDobs.scr obs.dat IND1 IND2  06 03
DDobs.scr obs.dat IND1 IND2  06 07
DDobs.scr obs.dat IND1 IND2  06 11
DDobs.scr obs.dat IND1 IND2  06 16
DDobs.scr obs.dat IND1 IND2  06 18
DDobs.scr obs.dat IND1 IND2  06 19
DDobs.scr obs.dat IND1 IND2  06 21
DDobs.scr obs.dat IND1 IND2  06 22
DDobs.scr obs.dat IND1 IND2  06 30
```

Merge the files in a single file and sort by time:

```
cat DD_IND1_IND2_06_??.dat|sort -n -k +6 > DD_IND1_IND2_06_ALL.dat
```

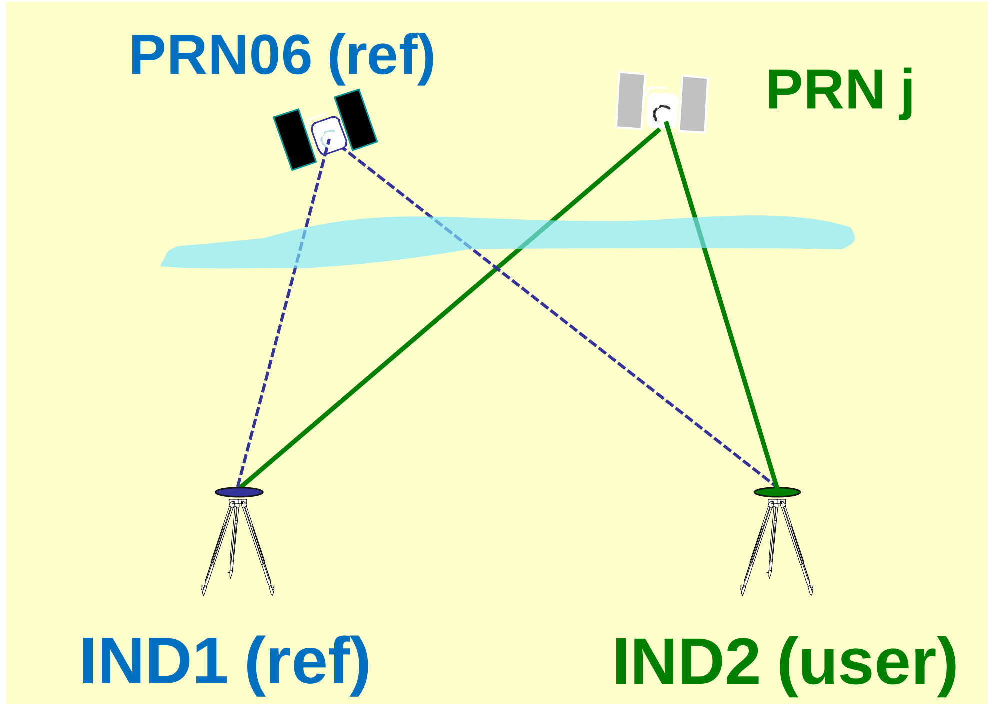
B. IND1- IND2 Differential positioning

```
------------------------------  DD_IND1_IND2_06_ALL.dat ------------------------------
   1    2    3    4    5    6     7     8     9    10     11    12     13     14   15   16   17
[IND1 IND2 06 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2]
                                                                    <---- IND2 ---->
-------------------------------------------------------------------------------------
```

OUTPUT file

Where the elevation (EL) and azimuth (AZ) are taken from station **IND2** (the user)

and where (EL1, AZ1) are for satellite PNR06 (reference) and (EL1, AZ1) are for satellite PRNXX



B.2. IND1-IND2 Baseline vector estimation with P1 code (using the time-tagged reference station measurements)

Preliminary: Using octave (or MATLAB), and the receiver coordinates estimated before, **compute the baseline vector between IND1-IND2**. Give the results in the **ENU local sys**tem (at IND2).

```
IND1=[4787678.1496 183409.7131 4196172.3056]
IND2=[4787678.9809 183402.5915 4196171.6833]

IND2-IND1
      ans= 0.8313 -7.1216  -0.6223   (XYZ)
```

```
IND2 (lat and long):

 l= 2.193768411 *pi/180
 f= 41.403018646 *pi/180
```

```
R=[   -sin(l)              cos(l)       0   ;
   -cos(l)*sin(f)  -sin(l)*sin(f) cos(f);
    cos(l)*cos(f)   sin(l)*cos(f) sin(f)]

bsl_enu=R*(IND2-IND1)'
     ans -7.1482 -0.8359 0.0070 (ENU)
```

B.3. IND1-IND2 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

B.3.1 Estimate the baseline vector between IND1 and IND2 receivers using the L1 carrier measurements of file (DD_IND1_IND2_06_ALL.dat).

Consider only the two epochs used in the previous exercise: t_1=14500 and t_2=14515

The following procedure can be applied:

1. **Compute the FLOATED solution**, solving the equations system with octave. Assess the accuracy of the floated solution.
2. **Apply the LAMBDA method to FIX the ambiguities**. Compare the results with the solution obtained by rounding directly the floated solution and by rounding the solution after decorrelation.
3. **Repair the DDL1 carrier measurements with the DDN1 FIXED** ambiguities and plot results to analyze the data.
4. **Compute the FIXED solution**.

B.3. IND1-IND2 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

B.3.1 Estimate the baseline vector between IND1 and IND2 receivers using the L1 carrier measurements of file (DD_IND1_IND2_06_ALL.dat).

```
[DDL1]= [Los_k - Los_06]*[baseline] + [ A ]*[lambda1*DDN1]
```

Notation

$$\begin{bmatrix} DDL_1^{6,3} \\ DDL_1^{6,7} \\ \ldots \\ DDL_1^{6,30} \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^3 - \hat{\boldsymbol{\rho}}^6\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7 - \hat{\boldsymbol{\rho}}^6\right)^T \\ \ldots \\ -\left(\hat{\boldsymbol{\rho}}^{30} - \hat{\boldsymbol{\rho}}^6\right)^T \end{bmatrix} \mathbf{r} + \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ & & \ldots & \\ 0 & 0 & \ldots & 1 \end{bmatrix} \begin{bmatrix} \lambda_1\, DDN_1^{6,3} \\ \lambda_1\, DDN_1^{6,7} \\ \ldots \\ \lambda_1\, DDN_1^{6,30} \end{bmatrix}$$

$$\mathbf{y} = \mathbf{G}\,\mathbf{x}$$

Where the vector of unknowns $\underline{\mathbf{x}}$ **<u>includes the user coordinates and ambiguities</u>**

B.3.1. Baseline vector estimation with DDL1 (using only two epochs)

B.3. IND1-IND2 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

The receiver was not moving (static) during the data collection.
Therefore, for each epoch we have the equations system:

$$\begin{bmatrix} DDL_1^{6,3}(t_1) \\ DDL_1^{6,7}(t_1) \\ \cdots \\ DDL_1^{6,30}(t_1) \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^3(t_1)-\hat{\boldsymbol{\rho}}^6(t_1)\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7(t_1)-\hat{\boldsymbol{\rho}}^6(t_1)\right)^T \\ \cdots \\ -\left(\hat{\boldsymbol{\rho}}^{30}(t_1)-\hat{\boldsymbol{\rho}}^6(t_1)\right)^T \end{bmatrix} \mathbf{r} + \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \lambda_1\, DDN_1^{6,3} \\ \lambda_1\, DDN_1^{6,7} \\ \cdots \\ \lambda_1\, DDN_1^{6,30} \end{bmatrix}$$

$$\mathbf{y}_1 = \mathbf{G}_1\, \mathbf{x}$$

y1:=y[t1]
G1:=G[t1]

$$\begin{bmatrix} DDL_1^{6,3}(t_2) \\ DDL_1^{6,7}(t_2) \\ \cdots \\ DDL_1^{6,30}(t_2) \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^3(t_2)-\hat{\boldsymbol{\rho}}^6(t_2)\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7(t_2)-\hat{\boldsymbol{\rho}}^6(t_2)\right)^T \\ \cdots \\ -\left(\hat{\boldsymbol{\rho}}^{30}(t_2)-\hat{\boldsymbol{\rho}}^6(t_2)\right)^T \end{bmatrix} \mathbf{r} + \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \lambda_1\, DDN_1^{6,3} \\ \lambda_1\, DDN_1^{6,7} \\ \cdots \\ \lambda_1\, DDN_1^{6,30} \end{bmatrix}$$

$$\mathbf{y}_2 = \mathbf{G}_2\, \mathbf{x}$$

y2:=y[t2]
G2:=G[t2]

```
[DDL1]=[Los_k - Los_06]*[baseline] + [ A ]*[lambda1*DDN1]
```

B.3. IND1-IND2 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

In the previous computation we have not taken into account the correlations between the double differences of measurements. This matrix will be used now, as the LAMBDA method will be applied to FIX the carrier ambiguities.

a) Show that the covariance matrix of DDL1 is given by **$\mathbf{P_y}$**

$$\mathbf{P_y} = 2\sigma^2 \begin{bmatrix} 2 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & 2 \end{bmatrix}$$

b) Given the measurement vectors (y) and Geometry matrices (G) for two epochs

y1:=y[t1] ; G1:=G[t1] ; Py

y2:=y[t2] ; G2:=G[t2] ; Py

show that the user solution and covariance matrix can be computed as:

P=inv(G1'*W*G1+G2'*W*G2);

where: **W=inv(Py)**

x=P*(G1'*W*y1+G2'*W*y2) ;

$$\mathbf{y = G\,x; \quad W = P_y^{-1}}$$

$$\mathbf{x = (G^T W G)^{-1} G^T W y}$$

$$\mathbf{P = (G^T W G)^{-1}}$$

B.3.1. Baseline vector estimation with DDL1 (using only two epochs)

B.3. IND1-IND2 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

The script **MakeL1BslMat.scr** builds the equations system

```
[DDL1]=[ Los_k- Los_06]*[baseline] + [ A ]*[λ1*DDN1]
```

for the two epochs required *t_1=14500* and *t_2=14515*, using the input file **DD_IND1_IND2_06_ALL.dat** generated before.

Execute:

```
MakeL1BslMat.scr DD_IND1_IND2_06_ALL.dat 14500 14515
```

The **OUTPUT**

are the files **M1.dat** and **M2.dat** associated with each epoch.

Where:

the columns of files **M.dat** are the vector **y** (first column) and Matrix **G** (next columns)

B.3.1. Baseline vector estimation with DDL1 (using only two epochs)

B.3. IND1-IND2 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

1. Computing the FLOATED solution (solving the equations system).

The following procedure can be applied

```
octave

load M1.dat
load M2.dat

y1=M1(:,1);
G1=M1(:,2:11);

y2=M2(:,1);
G2=M2(:,2:11);
Py=(diag(ones(1,7))+ones(7))*2e-4;
W=inv(Py);
```

```
P=inv(G1'*W*G1+G2'*W*G2);
x=P*(G1'*W*y1+G2'*W*y2);

x(1:3)'
 -8.8883   -2.2187   2.4998
bsl_enu =[-7.1482  -0.8359  0.0070]

x(1:3)'-bsl_enu
      ans=  -1.7401 -1.3828  2.4928
```

B.3. IND1-IND2 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

2. Applying the LAMBDA method to FIX the ambiguities.

The following procedure can be applied **(justify the computations done)**
Compare the different results found:

```
octave

c=299792458;
f0=10.23e+6;
f1=154*f0;
lambda1=c/f1
 a=x(4:10)/lambda1;
 Q=P(4:10,4:10);
```

Decorrelation and integer LS search solution

```
[Qz,Zt,Lz,Dz,az,iZ] = decorrel (Q,a);
[azfixed,sqnorm] = lsearch (az,Lz,Dz,2);
afixed=iZ*azfixed;
sqnorm(2)/sqnorm(1)
ans = 1.61389475957901
afixed(:,1)'
  17   -5   -2    4  -26    0   13
```

Rounding directly the floated solution

```
round(a)'
10  -11   13  -10   -8    7   10
```

Rounding the decorrelated floated solution

```
afix=iZ*round(az);
7  -22   16  -18   -3    1   -8
```

B.3. IND1-IND2 differential positioning with L1 carrier (using the computed differential corrections)

Questions:

1. - Can the ambiguities be fixed?

2. - Give a possible explanation about why the ambiguities cannot be fixed

B.3.1. Baseline vector estimation with DDL1 (using only two epochs)

Repeat previous processing, but using $t_1=14500$ and $t_2=15530$

B.3. IND1-IND2 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

The script **MakeL1BslMat.scr** builds the equations system

```
[DDL1]=[ Los_k- Los_06]*[baseline] + [ A ]*[λ1*DDN1]
```

for the two epochs required t_1*=14500* and ***t_2=15530***, using the input file **DD_IND1_IND2_06_ALL.dat** generated before.

Execute:

```
MakeL1BslMat.scr DD_IND1_IND2_06_ALL.dat 14500 15530
```

The **OUTPUT**

are the files **M1.dat** and **M2.dat** associated with each epoch.

Where:

the columns of files **M.dat** are the vector **y** (first column) and Matrix **G** (next columns)

B.3. IND1-IND2 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

1. Computing the FLOATED solution (solving the equations system).

The following procedure can be applied

```
octave

load M1.dat
load M2.dat

y1=M1(:,1);
G1=M1(:,2:11);

y2=M2(:,1);
G2=M2(:,2:11);
Py=(diag(ones(1,7))+ones(7))*2e-4;
W=inv(Py);
```

```
P=inv(G1'*W*G1+G2'*W*G2);
x=P*(G1'*W*y1+G2'*W*y2);

x(1:3)'
-6.78678   -0.7794    -0.2434
bsl_enu =[-7.1482 -0.8359 0.0070]

x(1:3)'-bsl_enu
      ans= 0.3614 0.0565 -0.2504
```

B.3.2. Baseline vector estimation with DDL1 (using t=14500, 15530)

B.3. IND1-IND2 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

2. Applying the LAMBDA method to FIX the ambiguities.

The following procedure can be applied **(justify the computations done)**
Compare the different results found:

```
octave

c=299792458;
f0=10.23e+6;
f1=154*f0;
lambda1=c/f1
 a=x(4:10)/lambda1;
 Q=P(4:10,4:10);
```

Decorrelation and integer LS search solution

```
[Qz,Zt,Lz,Dz,az,iZ] = decorrel (Q,a);
[azfixed,sqnorm] = lsearch (az,Lz,Dz,2);
afixed=iZ*azfixed;
sqnorm(2)/sqnorm(1)
ans =   1.10192131979339
afixed(:,1)'
9   -16    22   -10     7     9     9
```

Rounding directly the floated solution

```
round(a)'
8   -17    24   -12     9    10     9
```

Rounding the decorrelated floated solution

```
afix=iZ*round(az);
8   -17    24   -12     9    10     8
```

B.3. IND1-IND2 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

Questions:

1. - Can the ambiguities be fixed?

2. - Give a possible explanation about why the ambiguities cannot be fixed

B.3.2. Baseline vector estimation with DDL1 (using t=14500, 15530)

B.3. IND1-IND2 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

Repeat previous processing, but using $t_1=14500$ **and** $t_2=15000$

B.3. IND1-IND2 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

The script **MakeL1BslMat.scr** builds the equations system

```
[DDL1]=[ Los_k- Los_06]*[baseline] + [ A ]*[λ1*DDN1]
```

for the two epochs required t_1=14500 and t_2=15000, using the input file **DD_IND1_IND2_06_ALL.dat** generated before.

Execute:

```
MakeL1BslMat.scr DD_IND1_IND2_06_ALL.dat 14500 15000
```

The **OUTPUT**

are the files **M1.dat** and **M2.dat** associated with each epoch.

Where:

the columns of files **M.dat** are the vector **y** (first column) and Matrix **G** (next columns)

B.3. IND1-IND2 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

1. Computing the FLOATED solution (solving the equations system).

The following procedure can be applied

```
octave

load M1.dat
load M2.dat

y1=M1(:,1);
G1=M1(:,2:11);

y2=M2(:,1);
G2=M2(:,2:11);
Py=(diag(ones(1,7))+ones(7))*2e-4;
W=inv(Py);
```

```
P=inv(G1'*W*G1+G2'*W*G2);
x=P*(G1'*W*y1+G2'*W*y2);

x(1:3)'
-6.7640   -0.7441   -0.2256

bsl_enu =[-7.1482  -0.8359  0.0070]

x(1:3)'-bsl_enu
       ans=  0.3842 0.0918  -0.2326
```

B.3.3. Baseline vector estimation with DDL1 (using t=14500, 15000)

B.3. IND1-IND2 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

2. Applying the LAMBDA method to FIX the ambiguities.

The following procedure can be applied **(justify the computations done)**
Compare the different results found:

```
octave

c=299792458;
f0=10.23e+6;
f1=154*f0;
lambda1=c/f1
 a=x(4:10)/lambda1;
 Q=P(4:10,4:10);
```

Decorrelation and integer LS search solution

```
[Qz,Zt,Lz,Dz,az,iZ] = decorrel (Q,a);
[azfixed,sqnorm] = lsearch (az,Lz,Dz,2);
afixed=iZ*azfixed;
sqnorm(2)/sqnorm(1)
ans =  1.36905617725904
afixed(:,1)'
 9  -16   22  -10    7    9    9
```

Rounding directly the floated solution

```
round(a)'
8  -17   24  -12    9   10    9
```

Rounding the decorrelated floated solution

```
afix=iZ*round(az);
7  -18   26  -14   12   11    8
```

B.3.3. Baseline vector estimation with DDL1 (using t=14500, 15000)

B.3. IND1-IND2 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

OPTIONAL:
Repeat taking $t_1=14500$ and $t_2=17000$

Questions:
1. - Have the results improved?
2. - Has the reliability improved?
3. - Why it is not possible to fix the ambiguities?

B.3. IND1-IND2 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

Hint:

Check possible synchronism errors between the receivers' time tags.

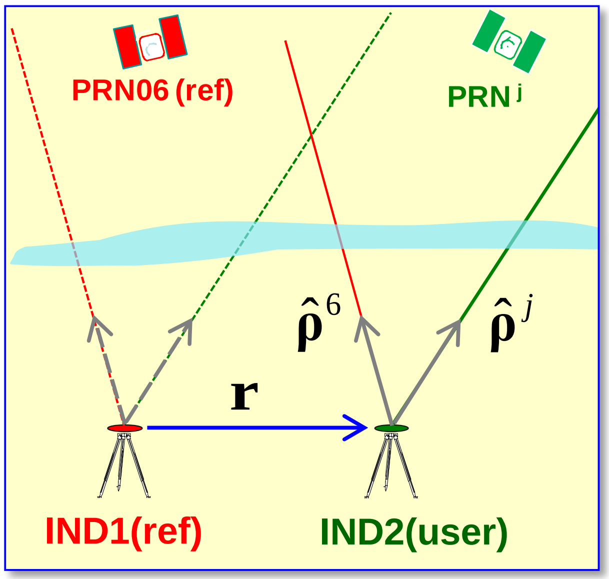
For instance, use the following sentence to compute the receiver clocks of IND1, IND2 and IND3 receivers with gLAB (the last field is the receiver clock offset):

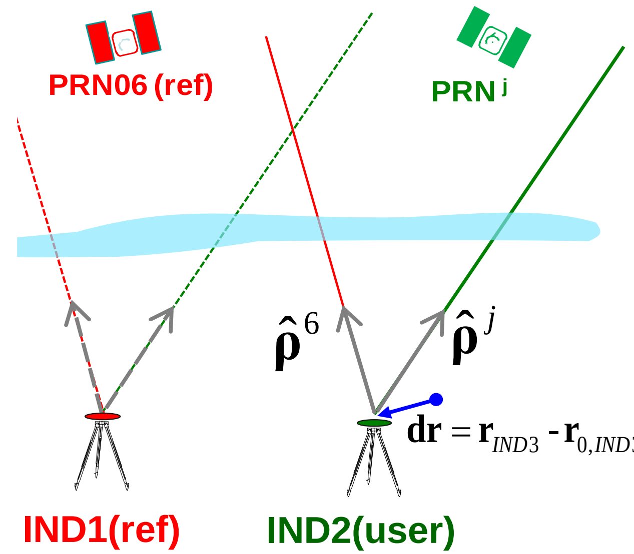
```
gLAB_linux -input:obs IND10540.13O -input:nav brdc0540.13n -pre:dec 1|grep FILTER
gLAB_linux -input:obs IND20540.13O -input:nav brdc0540.13n -pre:dec 1|grep FILTER
gLAB_linux -input:obs IND30540.13O -input:nav brdc0540.13n -pre:dec 1|grep FILTER
```

Questions:
Discuss how the relative receiver clock offset can affect the baseline estimation.

B.3. IND1-IND2 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

- In the previous exercise we have shown how the synchronism errors between the time-tagged measurements affect the ambiguity fixing when trying to estimate the baseline vector.





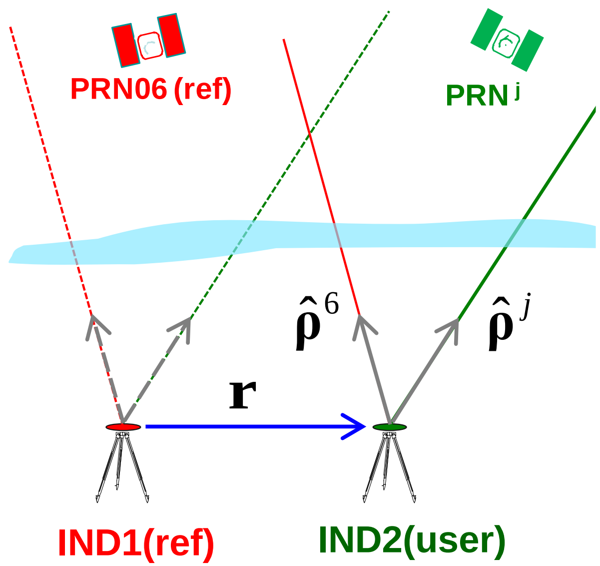
$L_{1,ref}^j$	**Time-tagged measurements** broadcast by reference station

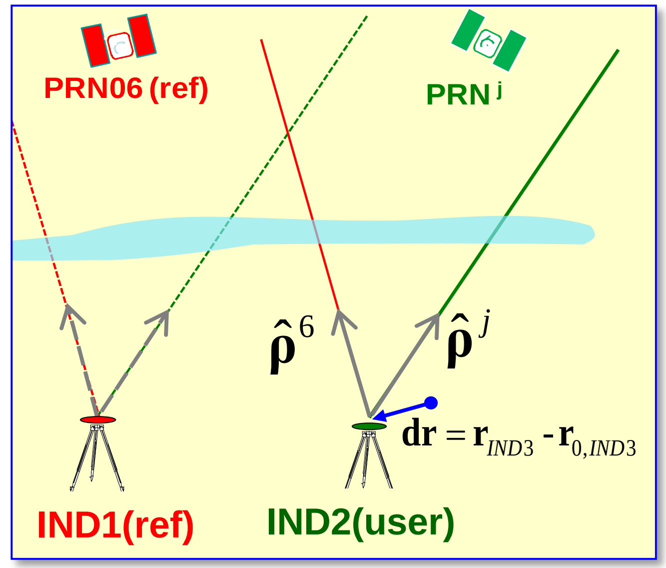
$$PRC_1^j \equiv L_{1,ref}^j - \rho_{ref}^j$$

Computed corrections broadcast by the reference station.

B.3. IND1-IND2 Baseline vector estimation with L1 carrier (using the time-tagged reference station measurements)

- Next we are going to repeat the differential positioning, but using the computed differential corrections. In this case, as the corrections vary slowly, the synchronization errors will not be an issue.





$L_{1,ref}^{j}$ Time-tagged measurements broadcast by reference station

$$PRC_1^j \equiv L_{1,ref}^j - \rho_{ref}^j$$

Computed corrections broadcast by the reference station.

B.3.3. Baseline vector estimation with DDL1.

B.4. IND1-IND2 differential positioning with L1 carrier (using the computed differential corrections)

B.4.1 Using DDL1 carrier measurements, estimate the coordinates of receiver IND2 taking IND1 as a reference receiver.

Consider only the two epochs used in the previous exercise: t_1=14500 and t_2=14530

The following procedure can be applied:

1. **Compute the FLOATED solution**, solving the equations system with octave. Assess the accuracy of the floated solution.
2. **Apply the LAMBDA method to FIX the ambiguities**. Compare the results with the solution obtained by rounding the floated solution directly and by rounding the solution after decorrelation.
3. **Repair the DDL1 carrier measurements with the DDN1 FIXED** ambiguities and plot results to analyze the data.
4. **Compute the FIXED solution**.

A.6.1. Estimate IND2 coordinates with DDL1 (using only two epochs)

B.4. IND2-IND3 differential positioning with L1 carrier (using the computed differential corrections)

B.4.1 Estimate the coordinates of receiver IND2 taking IND1 as the reference receiver, using the L1 carrier measurements of file (DD_IND1_IND2_06_ALL.dat)

```
[DDL1-DDRho]= [Los_k - Los_06]*[dr] + [ A ]*[lambda1*DDN1]
```

Notation

$$\begin{bmatrix} DDL_1^{6,3} - DD\rho^{6,3} \\ DDL_1^{6,7} - DD\rho^{6,7} \\ \cdots \\ DDL_1^{6,30} - DD\rho^{6,30} \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^3 - \hat{\boldsymbol{\rho}}^6\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7 - \hat{\boldsymbol{\rho}}^6\right)^T \\ \cdots\cdot \\ -\left(\hat{\boldsymbol{\rho}}^{30} - \hat{\boldsymbol{\rho}}^6\right)^T \end{bmatrix} \mathbf{dr} + \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ & & \ldots & \\ 0 & 0 & \ldots & 1 \end{bmatrix} \begin{bmatrix} \lambda_1\, DDN_1^{6,3} \\ \lambda_1\, DDN_1^{6,7} \\ \cdots \\ \lambda_1\, DDN_1^{6,30} \end{bmatrix}$$

$$\mathbf{y} = \mathbf{G}\,\mathbf{x}$$

Where the vector of unknowns $\underline{\mathbf{x}}$ **includes the user coordinates and ambiguities**

B.4.1. Estimate IND2 coordinates with DDL1 (using only two epochs)

B.4. IND1-IND2 differential positioning with L1 carrier (using the computed differential corrections)

The receiver was not moving (static) during the data collection. Therefore, for each epoch we have the equations system:

$$\begin{bmatrix} DDL_1^{6,3}(t_1) - DD\rho^{6,3}(t_1) \\ DDL_1^{6,7}(t_1) - DD\rho^{6,7}(t_1) \\ \ldots \\ DDL_1^{6,30}(t_1) - DD\rho^{6,30}(t_1) \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^3(t_1) - \hat{\boldsymbol{\rho}}^6(t_1)\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7(t_1) - \hat{\boldsymbol{\rho}}^6(t_1)\right)^T \\ \ldots \\ -\left(\hat{\boldsymbol{\rho}}^{30}(t_1) - \hat{\boldsymbol{\rho}}^6(t_1)\right)^T \end{bmatrix} \mathbf{dr} + \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ & & \ldots & \\ 0 & 0 & \ldots & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \, DDN_1^{6,3} \\ \lambda_1 \, DDN_1^{6,7} \\ \ldots \\ \lambda_1 \, DDN_1^{6,30} \end{bmatrix}$$

$$\mathbf{y}_1 = \mathbf{G}_1 \, \mathbf{x}$$

y1:=y[t1]
G1:=G[t1]

$$\begin{bmatrix} DDL_1^{6,3}(t_2) - DD\rho^{6,3}(t_2) \\ DDL_1^{6,7}(t_2) - DD\rho^{6,7}(t_2) \\ \ldots \\ DDL_1^{6,30}(t_2) - DD\rho^{6,30}(t_2) \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^3(t_2) - \hat{\boldsymbol{\rho}}^6(t_2)\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7(t_2) - \hat{\boldsymbol{\rho}}^6(t_2)\right)^T \\ \ldots \\ -\left(\hat{\boldsymbol{\rho}}^{30}(t_2) - \hat{\boldsymbol{\rho}}^6(t_2)\right)^T \end{bmatrix} \mathbf{dr} + \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ & & \ldots & \\ 0 & 0 & \ldots & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \, DDN_1^{6,3} \\ \lambda_1 \, DDN_1^{6,7} \\ \ldots \\ \lambda_1 \, DDN_1^{6,30} \end{bmatrix}$$

$$\mathbf{y}_2 = \mathbf{G}_2 \, \mathbf{x}$$

y2:=y[t2]
G2:=G[t2]

```
[DDL1-DDRho]=[Los_k - Los_06]*[dr] + [ A ]*[lambda1*DDN1]
```

B.4. IND1-IND2 differential positioning with L1 carrier (using the computed differential corrections)

In the previous computation we have not taken into account the correlations between the double differences of measurements. This matrix will be used now, as the LAMBDA method will be applied to FIX the carrier ambiguities.

$$\mathbf{P_y} = 2\sigma^2 \begin{bmatrix} 2 & 1 & \mathrm{L} & 1 \\ 1 & 2 & \mathrm{L} & 1 \\ \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M} \\ 1 & 1 & 1 & 2 \end{bmatrix}$$

a) Show that the covariance matrix of DDL1 is given by $\mathbf{P_y}$

b) Given the measurement vectors (y) and Geometry matrices (G) for two epochs

y1:=y[t1] ; G1:=G[t1] ; Py
y2:=y[t2] ; G2:=G[t2] ; Py

show that the user solution and covariance matrix can be computed as:

P=inv(G1'*W*G1+G2'*W*G2);

where: **W=inv(Py)**

x=P*(G1'*W*y1+G2'*W*y2) ;

$$\mathbf{y = G\,x;} \quad \mathbf{W = P_y^{-1}}$$

$$\mathbf{x = (G^T W G)^{-1} G^T W y}$$

$$\mathbf{P = (G^T W G)^{-1}}$$

B.4.1. Estimate IND2 coordinates with DDL1 (using only two epochs)

B.4. IND1-IND2 differential positioning with L1 carrier (using the computed differential corrections)

The script **`MakeL1DifMat.scr`** builds the equations system

```
[DDL1-DDRho]=[ Los_k- Los_06]*[dr] + [ A ]*[λ1*DDN1]
```

for the two epochs required $t_1=14500$ and $t_2=14530$, using the input file **`DD_IND1_IND2_06_ALL.dat`** generated before.

Execute:

```
MakeL1DifMat.scr DD_IND1_IND2_06_ALL.dat 14500 14530
```

The **OUTPUT**

are the files **`M1.dat`** and **`M2.dat`** associated with each epoch.

Where:

the columns of files **`M.dat`** are the vector **`y`** (first column) and Matrix **`G`** (next columns)

B.41. Estimate IND2 coordinates with DDL1 (using only two epochs)

B.4. IND1-IND2 differential positioning with L1 carrier (using the computed differential corrections)

1. Computing the FLOATED solution (solving the equations system).

The following procedure can be applied

```
octave

load M1.dat
load M2.dat

y1=M1(:,1);
G1=M1(:,2:11);

y2=M2(:,1);
G2=M2(:,2:11);
Py=(diag(ones(1,7))+ones(7))*2e-4;
W=inv(Py);
```

```
P=inv(G1'*W*G1+G2'*W*G2);
x=P*(G1'*W*y1+G2'*W*y2);

x(1:3)'
0.3132   -0.2648   0.6237
```

Taking into account that the "a priori" coordinates of IND2 are: `IND2=[4787678.9809 183402.5915 4196171.6833]`

Therefore the estimated absolute coordinates of IND3 are:

```
IND2+ x(1:3)'
4787679.2940  183402.3267  4196172.3070
```

B.4.1. Estimate IND2 coordinates with DDL1 (using only two epochs)

2. Applying the LAMBDA method to FIX the ambiguities.

The following procedure can be applied **(justify the computations done)**
Compare the different results found.

```
octave

c=299792458;
f0=10.23e+6;
f1=154*f0;
lambda1=c/f1
 a=x(4:10)/lambda1;
 Q=P(4:10,4:10);
```

Decorrelation and integer LS search solution

```
[Qz,Zt,Lz,Dz,az,iZ] = decorrel (Q,a);
[azfixed,sqnorm] = lsearch (az,Lz,Dz,2);
afixed=iZ*azfixed;
sqnorm(2)/sqnorm(1)
ans = 2.25895684415922
afixed(:,1)'
9   -17   22   -10   6   10   7
```

Rounding directly the floated solution

```
round(a)'
8   -17   22   -12   5   11   7
```

Rounding the decorrelated floated solution

```
afix=iZ*round(az);
9   -17   22   -10   6   10   7
```

A.6.1. Estimate IND2 coordinates with DDL1 (using only two epochs)

B.4. IND1-IND2 differential positioning with L1 carrier (using the computed differential corrections)

Questions:

1.- Can the ambiguities be fixed now? Why?
2.- Discuss why the synchronism errors affect the two differential positioning implementations.

B.4.1. Estimate IND2 coordinates with DDL1 (using only two epochs)

B.4. IND1-IND2 differential positioning with L1 carrier (using the computed differential corrections)

3. Repair the DDL1 carrier measurements with the DDN1 FIXED ambiguities and plot results to analyze the data.

```
octave
amb=lambda1*afixed(:,1);
save ambL1.dat amb
```

Using the previous the file **ambL1.dat** and "DD_IND1_IND2_06_ALL.dat", generate a file with the following content:

```
---------------------------- DD_IND2_IND3_06_ALL.fixL1 --------------------------------
   1     2    3    4    5    6     7     8     9    10     11    12     13     14   15   16   17      18
[IND1 IND2 06 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2  λ1*DDN1]
                                                                               <---- IND2 ---->
-----------------------------------------------------------------------------------------------
```

Note: This file is identical to file "DD_IND1_IND2_06_ALL.dat", but with the ambiguities added in the last field #18.

B.4.1. Estimate IND2 coordinates with DDL1 (using only two epochs)

B.4. IND1-IND2 differential positioning with L1 carrier (using the computed differential corrections)

a) Generate a file with the satellite PRN number and the ambiguities:

```
grep -v \# ambL1.dat > na1
cat DD_IND1_IND2_06_ALL.dat|gawk '{print $4}'|sort -nu|gawk '{print $1,NR}' >sat.lst
paste sat.lst na1 > sat.ambL1
```

b) Generate the "`DD_IND2_IND3_06_ALL.fixL1`" file:

```
cat DD_IND1_IND2_06_ALL.dat|
                 gawk 'BEGIN{for (i=1;i<1000;i++) {getline <"sat.ambL1";A[$1]=$3}}
  {printf "%s %02i %02i %s %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f
                                    %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f \n",
                $1,$2,$3,$4,$5,$6,$7,$8,$9,$10,$11,$12,$13,$14,$15,$16,$17,A[$4]}' >
                                                              DD_IND1_IND2_06_ALL.fixL1
```

B.4. IND1-IND2 differential positioning with L1 carrier (using the computed differential corrections)

```
----------------------------  DD_IND2_IND3_06_ALL.fixL1 --------------------------------
   1    2    3    4    5    6    7    8    9   10    11    12     13    14   15   16   17     18
[IND1 IND2 06 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2  λ1*DDN1]
                                                                  <---- IND2 ---->
------------------------------------------------------------------------------------------
```

c) Make and discuss the following plots

```
graph.py -f DD_IND1_IND2_06_ALL.fixL1 -x6 -y'($8-$18-$11)'
       -so --yn -0.06 --yx 0.06 -l "(DDL1-lambda1*DDN1)-DDrho" --xl "time (s)" --yl "m"
```

```
graph.py -f DD_IND1_IND2_06_ALL.fixL1 -x6 -y'($8-$11)'
       -so --yn -5 --yx 5 -l "(DDL1-Ddrho)" --xl "time (s)" --yl "metres"
```

```
graph.py -f DD_IND1_IND2_06_ALL.fixL1 -x6 -y'($8-$18)'
       -so --yn -10 --yx 10 -l "(DDL1-lambda1*DDN1)" --xl "time (s)" --yl "metres"
```

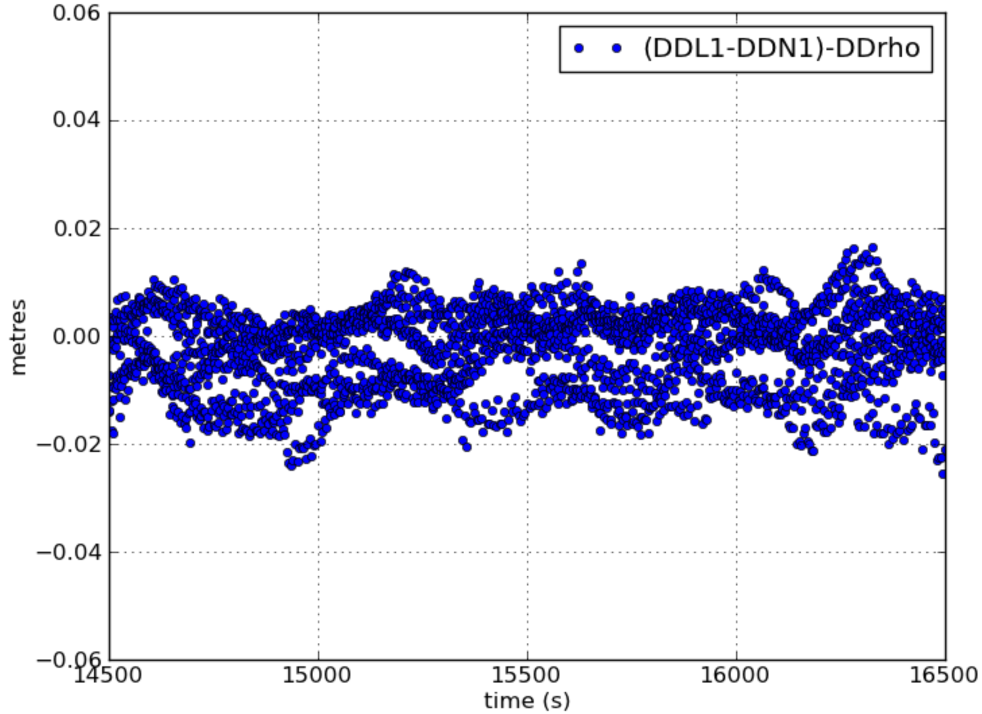

B.4.1. Estimate IND2 coordinates with DDL1 (using only two epochs)

B.4. IND1-IND2 differential positioning with L1 carrier (using the computed differential corrections)

```
----------------------------  DD_IND2_IND3_06_ALL.fixL1  ---------------------------------
  1     2    3    4    5    6     7     8     9    10     11    12     13     14   15   16   17      18
[IND1 IND2 06 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2  λ1*DDN1]
                                                                   <---- IND2 ---->
------------------------------------------------------------------------------------------
```

```
graph.py -f DD_IND1_IND2_06_ALL.fixL1
         -x6 -y'($8-$18-$11)'
         -so --yn -0.06 --yx 0.06
         -l "(DDL1-λ1*DDN1)-DDrho"
         --xl "time (s)" --yl "m"
```

Questions:
Explain what is the meaning of this plot.

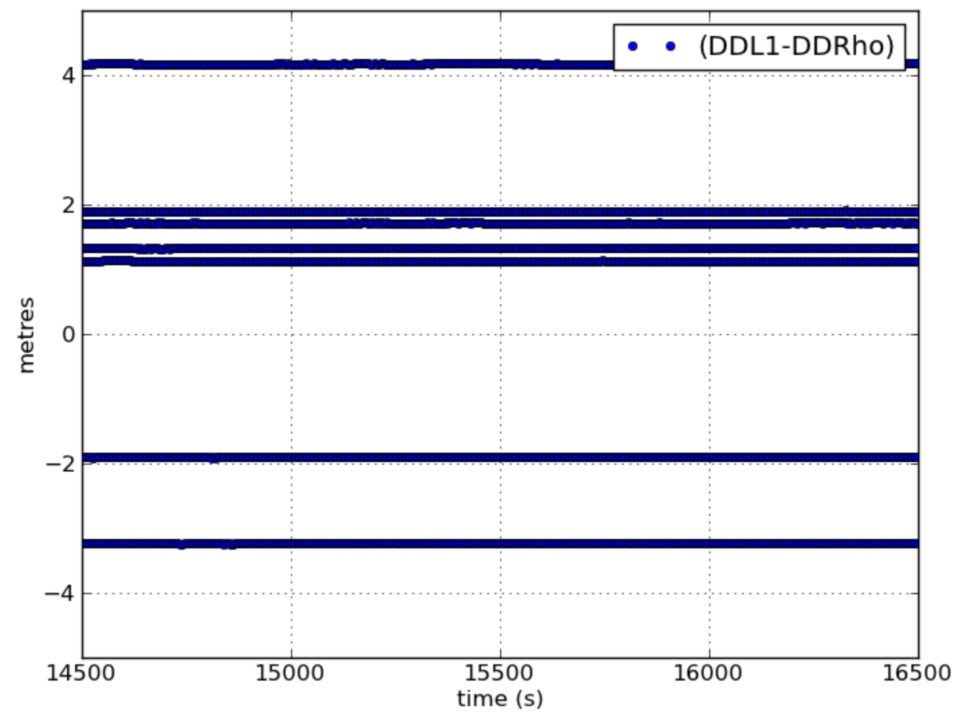


B.4.1. Estimate IND2 coordinates with DDL1 (using only two epochs)

B.4. IND1-IND2 differential positioning with L1 carrier (using the computed differential corrections)

```
----------------------------  DD_IND2_IND3_06_ALL.fixL1 ------------------------------------
   1     2    3    4    5    6     7     8     9    10     11    12     13     14   15   16   17       18
[IND1 IND2 06 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2  λ1*DDN1]
                                                                          <---- IND2 ---->
---------------------------------------------------------------------------------------------
```

```
graph.py -f DD_IND1_IND2_06_ALL.fixL1
         -x6 -y'($8-$11)'
         -so --yn -5 --yx 5
         -l ”(DDL1-DDrho)"
         --xl "time (s)" --yl "m"
```



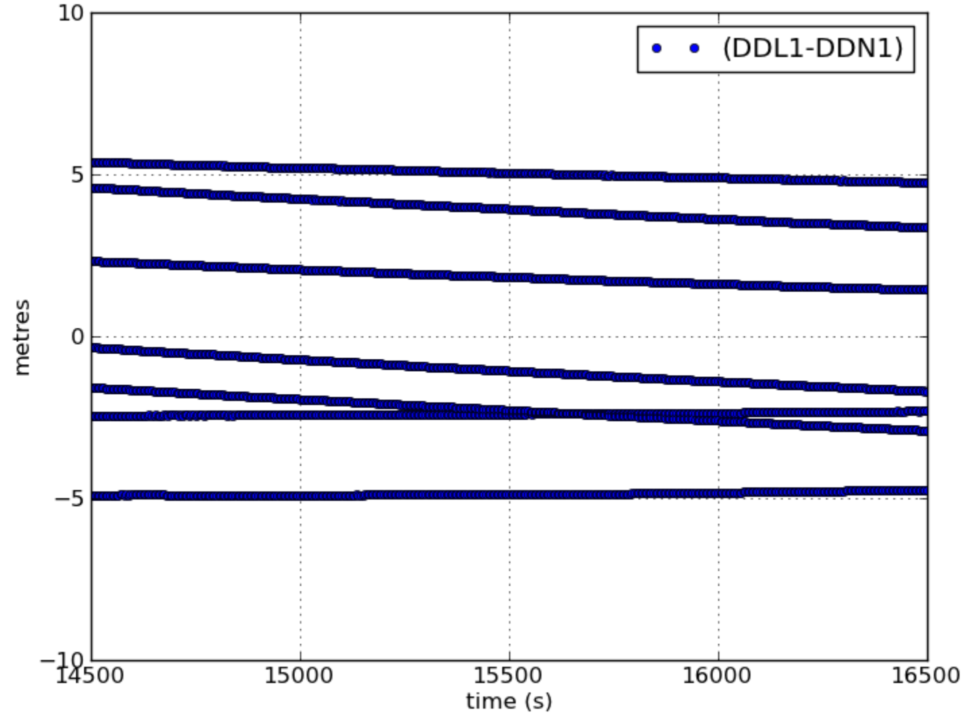
Questions:
Explain what is the meaning of this plot.

B.4.1. Estimate IND2 coordinates with DDL1 (using only two epochs)

B.4. IND1-IND2 differential positioning with L1 carrier (using the computed differential corrections)

```
----------------------------  DD_IND2_IND3_06_ALL.fixL1  ----------------------------------
   1     2   3   4   5   6    7    8    9    10    11    12     13     14  15  16  17     18
[IND1 IND2 06 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2  λ1*DDN1]
                                                                   <---- IND2 ---->
-----------------------------------------------------------------------------------------
```

```
graph.py -f DD_IND1_IND2_06_ALL.fixL1
         -x6 -y'($8-$18)'
         -so --yn -10 --yx 10
         -l "(DDL1-λ1*DDN1)"
         --xl "time (s)" --yl "m"
```



Questions:
Explain what is the meaning of this plot.

B.4.1. Estimate IND2 coordinates with DDL1 (using only two epochs)

B.4. IND1-IND2 differential positioning with L1 carrier (using the computed differential corrections)

4. Computing the FIXED solution (after FIXING ambiguities).

The following procedure can be applied

a) Build the equations system

```
[DDL1-DDRho-lambda1*DDN1]=[Los_k - Los_06]*[dr]
```

```
cat DD_IND1_IND2_06_ALL.fixL1 | gawk 'BEGIN{g2r=atan2(1,1)/45}
                    {e1=$14*g2r;a1=$15*g2r;e2=$16*g2r;a2=$17*g2r;
 printf "%14.4f %8.4f %8.4f %8.4f \n",
  $8-$11-$18, -cos(e2)*sin(a2)+cos(e1)*sin(a1),
         -cos(e2)*cos(a2)+cos(e1)*cos(a1), sin(e2)+sin(e1)}' > M.dat
```


B.4.1. Estimate IND2 coordinates with DDL1 (using only two epochs)

B.4. IND1-IND2 differential positioning with L1 carrier (using the computed differential corrections)

Solve the equations system using octave (or MATLAB) and assess the estimation error:

```
octave

load M.dat

y=M(:,1);
G=M(:,2:4);

x=inv(G'*G)*G'*y
x
  0.0118233916366036
  0.0016443593867621
 -0.0079900779585063
```

Absolute coordinates of IND3.

Taking into account that the ”a priori” coordinates of IND2 are:

```
IND2=[4787678.9809 183402.5915 4196171.6833]
```

Therefore the estimated absolute coordinates of IND2 are:

```
IND2+ x(1:3)'
 ans= 4787678.9927 183402.5931 4196171.6753
```

Question:

Is the accuracy similar to that in the previous case, when estimating the baseline vector? Why?

B.4.1. Estimate IND2 coordinates with DDL1 (using only two epochs)

B.4. IND1-IND2 differential positioning with L1 carrier (using the computed differential corrections)

B.4.2. Using the DDL1 carrier with the ambiguities FIXED, compute the LS single epoch solution for the whole interval 14500< t <16500 with the program LS.f

Note: The program "LS.f" computes the Least Square solution for each measurement epoch of the input file (see the FORTRAN code "LS.f")

The following procedure can be applied

a) generate a file with the following content;

```
[Time], [DDL1-DDRho-lambda1*DDN1], [ Los_k- Los_06]
```

where:

Time= seconds of day

DDL1 - DDRho-lambda1*DDN1= Prefit residulas (i.e., "y" values in program LS.f)

Los_k - Los_06 = The three components of the geometry matrix (i.e., matrix "a" in program LS.f)

B.4.2. Estimate IND2 coordinates with DDL1 (single epoch LS, whole interval)

B.4. IND1-IND2 differential positioning with L1 carrier (using the computed differential corrections)

```
[Time], [DDL1-DDRho-lambda1*DDN1], [Los_k - Los_06]
```

The following sentence can be used

```
cat DD_IND1_IND2_06_ALL.fixL1 | gawk 'BEGIN{g2r=atan2(1,1)/45}
  {e1=$14*g2r;a1=$15*g2r;e2=$16*g2r;a2=$17*g2r;;printf "%s %14.4f
   %8.4f %8.4f %8.4f \n",$6,$8-$11-$18,-cos(e2)*sin(a2)+cos(e1)*sin(a1),
        -cos(e2)*cos(a2)+cos(e1)*cos(a1),-sin(e2)+sin(e1)}' > L1model.dat
```

b) Compute the Least Squares solution

```
cat L1model.dat |LS > L1fix.pos
```

B.4. IND1-IND2 differential positioning with L1 carrier (using the computed differential corrections)

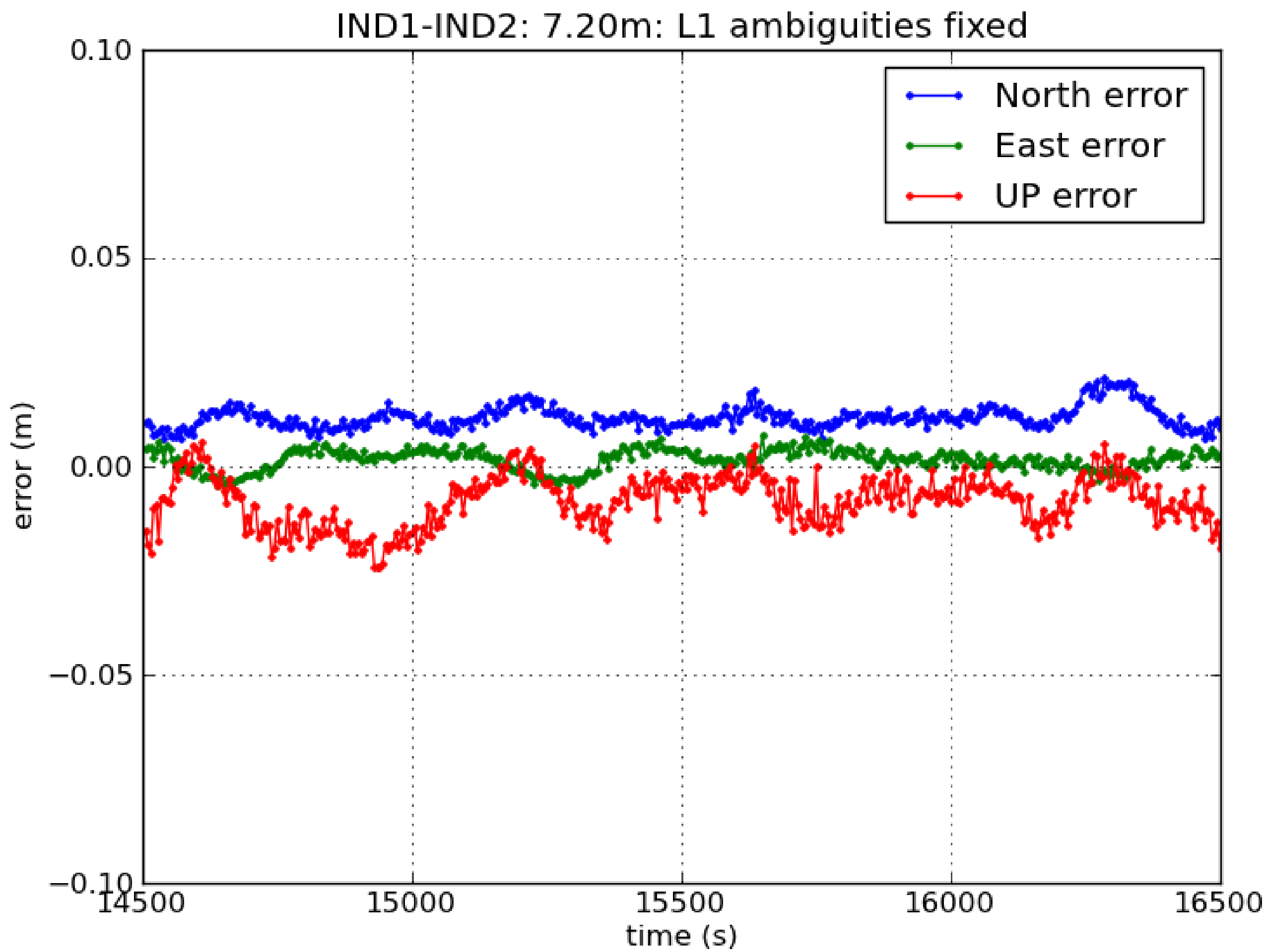
Plot the baseline estimation error

```
graph.py -f L1fix.pos -x1 -y2 -s.- -l "North error"
         -f L1fix.pos -x1 -y3 -s.- -l "East error"
         -f L1fix.pos -x1 -y4 -s.- -l "UP error"
  --yn -.1 --yx .1 --xl "time (s)" --yl "error (m)" -t "IND1-IND2: 7.20m:
L1 ambiguities fixed"
```


B.4.2. Estimate IND2 coordinates with DDL1 (single epoch LS, whole interval)

B.4. IND1-IND2 differential positioning with L1 carrier (using the computed differential corrections)

Differential Positioning error after fixing ambiguities



Question:
Discuss the accuracy achieved and the possible error sources that could affect this result (e.g. Antenna Phase Centres…)

B.4.2. Estimate IND2 coordinates with DDL1 (single epoch LS, whole interval)

B.4. IND1-IND2 differential positioning with L1 carrier (using the computed differential corrections)

B.4.3. Repeat previous computations, but using the Unsmoothed code P1.

i.e., compute the LS single epoch solution for the whole interval 14500< t <16500 with the program LS.f

The same procedure as in previous case can be applied, but using the code DDP1 instead of the carrier "DDL1-lambda1*DDN1"

a) generate a file with the following content;

```
[Time], [DDP1-DDRho], [ Los_k - Los_06]
```

where:

Time= seconds of day

DDP1 - DDrho= Prefit residuals (i.e., "y" values in program lms1)

Los_k - Los_06 = The three components of the geometry matrix (i.e., matrix "a" in program LS.f)

B.4.3. Estimate IND2 coordinates with DDP1 (single epoch LS, whole interval)

B.4. IND1-IND2 differential positioning with L1 carrier (using the computed differential corrections)

```
[Time], [DDP1-DDRho], [Los_k - Los_06]
```

The following sentence can be used

```
cat DD_IND1_IND2_06_ALL.fixL1 | gawk 'BEGIN{g2r=atan2(1,1)/45}
  {e1=$14*g2r;a1=$15*g2r;e2=$16*g2r;a2=$17*g2r;;printf "%s %14.4f
   %8.4f %8.4f %8.4f \n",$6,$7-$11,-cos(e2)*sin(a2)+cos(e1)*sin(a1),
        -cos(e2)*cos(a2)+cos(e1)*cos(a1),-sin(e2)+sin(e1)}' > P1model.dat
```

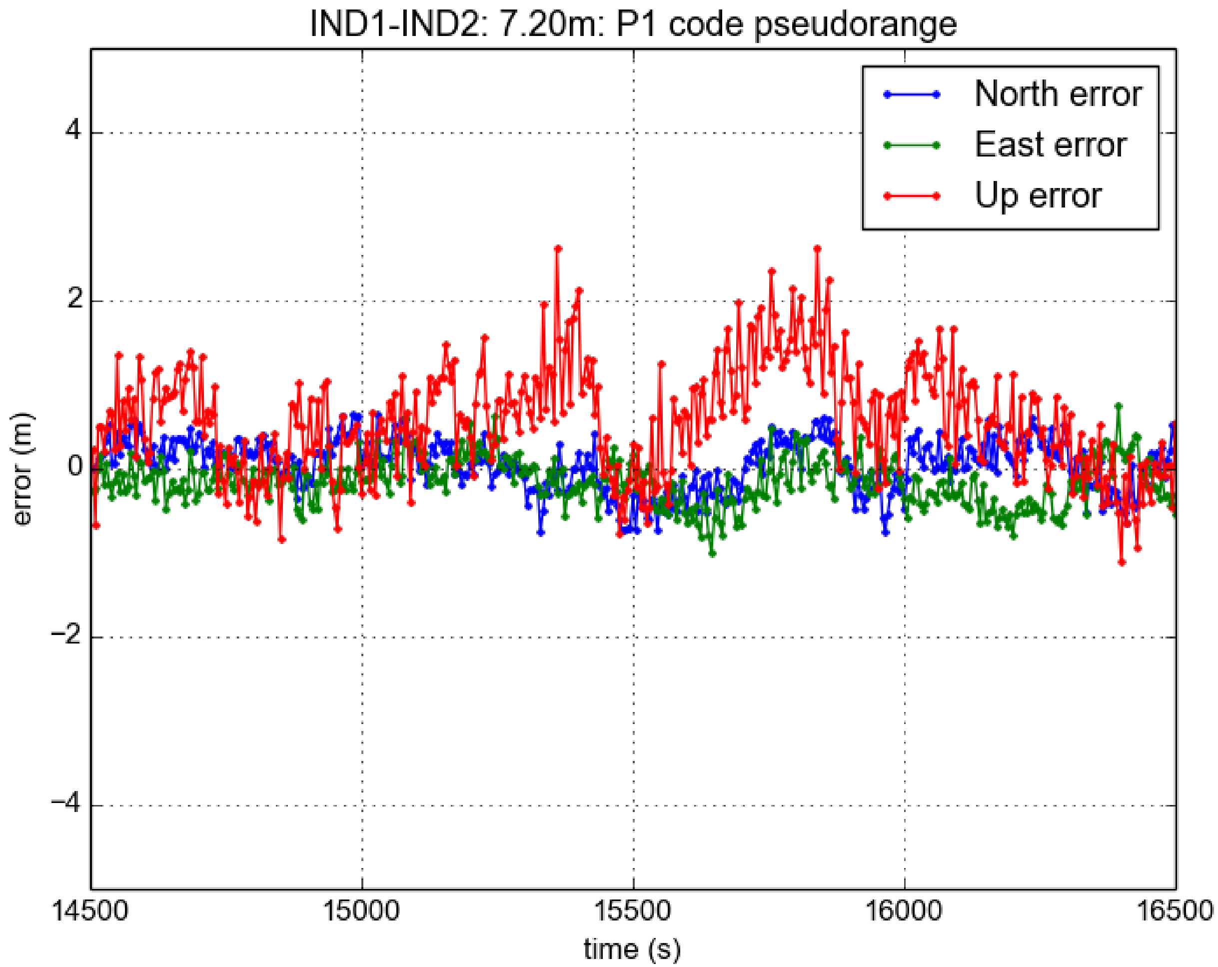
b) Compute the Least Squares solution

```
cat P1model.dat |LS > P1.pos
```

B.4.3. Estimate IND2 coordinates with DDP1 (single epoch LS, whole interval)

B.4. IND1-IND2 differential positioning with L1 carrier (using the computed differential corrections)

Positioning error with the P1 code



Question:
Discuss the results by comparing them with the previous ones with DDL1 carrier in the relative positioning implementation.

B.4.3. Estimate IND2 coordinates with DDP1 (single epoch LS, whole interval)

B.4.4. Repeat the previous computations, but for the baseline vector estimation and using the time-tagged measurements of the reference station, instead of the differential corrections.
That is, compute the LS single epoch solution for the whole interval 14500< t <16500 with the program LS.f

The same procedure as in previous exercise A.6.2 can be applied,
a) generate a file with the following content;

```
[Time], [DDL1], [ Los_k - Los_06]
```

where:
Time= seconds of day
DDL1= Prefit residulas (i.e., "y" values in program lms1)
Los_k - Los_06 = The three components of the geometry matrix (i.e., matrix "a" in program LS.f)

B.4.4. Baseline vector estimation with DDL1 (single epoch LS, whole interval)

B.4. IND1-IND2 differential positioning with L1 carrier

```
[Time], [DDL1-lambda1*DDN1], [Los_k - Los_06]
```

The following sentence can be used

```
cat DD_IND2_IND3_06_ALL.fixL1 | gawk 'BEGIN{g2r=atan2(1,1)/45}
  {e1=$14*g2r;a1=$15*g2r;e2=$16*g2r;a2=$17*g2r;;printf "%s %14.4f
   %8.4f %8.4f %8.4f \n",$6,$8-$18,-cos(e2)*sin(a2)+cos(e1)*sin(a1),
        -cos(e2)*cos(a2)+cos(e1)*cos(a1),-sin(e2)+sin(e1)}' > L1model.dat
```

b) Compute the Least Squares solution

```
cat L1model.dat |LS > L1fix.pos
```

Plot the baseline estimation error

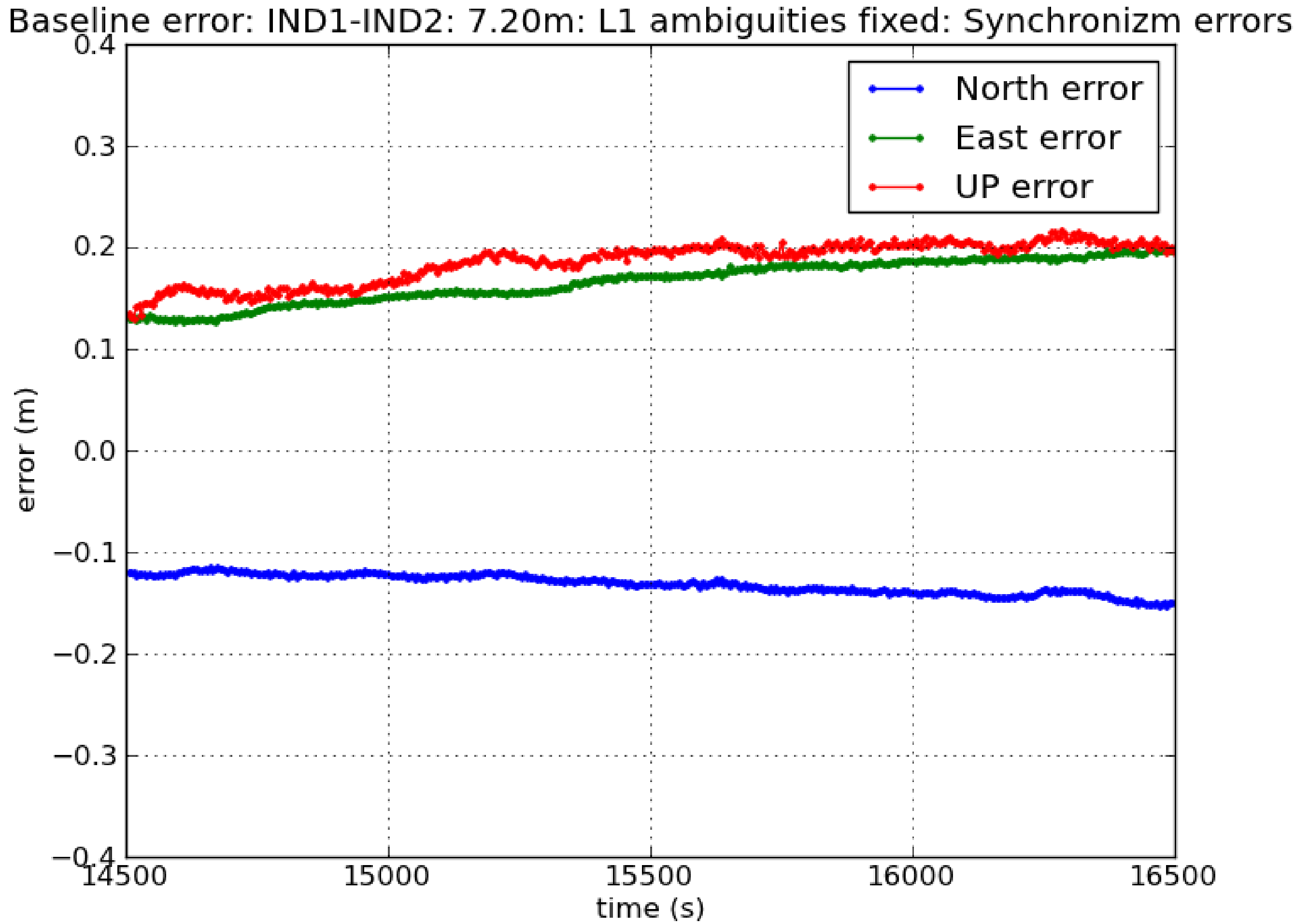
```
graph.py -f L1bslfix.pos -x1 -y'($2+7.1482)' -s.- -l "North error"
         -f L1bslfix.pos -x1 -y'($3+0.8359)' -s.- -l "East error"
         -f L1bslfix.pos -x1 -y'($4+0.0070)'  -s.- -l "UP error"
    --yn -.4 --yx .4 --xl "time (s)" --yl "error (m)" -t "Baseline error:
IND1-IND2: 7.20m: L1 ambiguities fixed: synchronism errors”
```

Note:

bsl_enu =[-7.1482 -0.8359 0.0070]

B.4. IND1-IND2 differential positioning with L1 carrier

Baseline estimation error after fixing ambiguities



Question:
Discuss why does the accuracy degrades respect to the previous case. Why this large error appears?

B.4.4. Baseline vector estimation with DDL1 (single epoch LS, whole interval)

OVERVIEW

- **Introduction:** gLAB processing in command line
- **Preliminary computations:** data files & reference values
- **Session A:** Differential positioning of IND2-IND3 receivers (baseline: 18 metres)
- **Session B:** Differential positioning of IND1-IND2 receivers (baseline: 7 metres, but synchronization errors)
- **Session C:** Differential positioning of PLAN-GARR receivers (baseline: 15 km, Night time): tropospheric effects
- **Session D:** Differential positioning of PLAN-GARR receivers (baseline: 15 km, Day time): tropospheric and Ionospheric effects

Session C

Differential positioning of PLAN- GARR receivers
(baseline: 15 km. Night time)

Analysis of differential tropospheric error effects

C. PLAN- GARR Differential positioning

- The previous exercises have been done over short baselines (less than 20 metres), where the errors introduced by the troposphere and ionosphere completely cancel when making differences of measurements.

- In this session we will consider a larger baseline (15 km) in order to assess the effect of the atmosphere on the ambiguity fixing and positioning.

- In this session we will consider Night time data in order to have a lower ionospheric error.

C. PLAN- GARR Differential positioning

C.1. **Double differences** between receivers and satellites computation

The script "**DDobs.scr**" computes the double differences between receivers and satellites from file **`obs.dat`**.

```
 1    2    3    4    5   6  7  8  9   10   11  12   13
[sta sat DoY sec P1 L1 P2 L2 Rho Trop Ion elev azim]
```

For instance, the following sentence:

```
DDobs.scr obs.dat PLAN GARR 06 03
```

generates the file

```
------------------- DD_{sta1}_{sta2}_{sat1}_{sat2}.dat ----------------------------
   1    2    3    4   5   6    7    8    9   10    11    12     13    14  15  16  17
[sta1 sta2 sat1 sat2 DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2]
                                                                   <---- sta2 ---->
-----------------------------------------------------------------------------------
```

Where the elevation (EL) and azimuth (AZ) are taken from station #2.

and where (EL1, AZ1) are for satellite #1 and (EL1, AZ1) are for satellite #2.

C. PLAN- GARR Differential positioning

Compute the double differences between receivers **PLAN (reference)** and GARR and satellites **PRN06 (reference)** and [PRN 03, 07,11, 16, 18, 19, 21, 22, 30]

```
DDobs.scr obs.dat PLAN GARR 06 03
DDobs.scr obs.dat PLAN GARR 06 07
DDobs.scr obs.dat PLAN GARR 06 11
DDobs.scr obs.dat PLAN GARR 06 16
DDobs.scr obs.dat PLAN GARR 06 18
DDobs.scr obs.dat PLAN GARR 06 19
DDobs.scr obs.dat PLAN GARR 06 21
DDobs.scr obs.dat PLAN GARR 06 22
DDobs.scr obs.dat PLAN GARR 06 30
```

Merge the files in a single file and sort by time:

```
cat DD_PLAN_GARR_06_??.dat|sort -n -k +6 > DD_PLAN_GARR_06_ALL.dat
```

C.2. PLAN-GARR differential positioning with L1 carrier (using the computed differential corrections)

C.2.1 Using DDL1 carrier measurements, estimate the coordinates of receiver GARR taking PLAN as a reference receiver.

Consider only the two epochs used in the previous exercise: $t_1=14500$ and $t_2=14590$.

The following procedure can be applied:

1. **Compute the FLOATED solution**, solving the equations system with octave. Assess the accuracy of the floated solution.
2. **Apply the LAMBDA method to FIX the ambiguities**. Compare the results with the solution obtained by rounding the floated solution directly and by rounding the solution after decorrelation.
3. **Repair the DDL1 carrier measurements with the DDN1 FIXED** ambiguities and plot results to analyze the data.
4. **Compute the FIXED solution**.

C.2. PLAN-GARR differential positioning with L1 carrier (using the computed differential corrections)

C.2.1 Estimate the coordinates of receiver GARR taking PLAN as the reference receiver, using the L1 carrier measurements of file (DD_PLAN_GARR_06_ALL.dat)

```
[DDL1-DDRho]= [Los_k - Los_06]*[dr] + [ A ]*[lambda1*DDN1]
```

Notation

$$\begin{bmatrix} DDL_1^{6,3} - DD\rho^{6,3} \\ DDL_1^{6,7} - DD\rho^{6,7} \\ \dots \\ DDL_1^{6,30} - DD\rho^{6,30} \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^{3} - \hat{\boldsymbol{\rho}}^{6}\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^{7} - \hat{\boldsymbol{\rho}}^{6}\right)^T \\ \dots \\ -\left(\hat{\boldsymbol{\rho}}^{30} - \hat{\boldsymbol{\rho}}^{6}\right)^T \end{bmatrix} \mathbf{dr} + \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ & & \dots & \\ 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \, DDN_1^{6,3} \\ \lambda_1 \, DDN_1^{6,7} \\ \dots \\ \lambda_1 \, DDN_1^{6,30} \end{bmatrix}$$

$$\mathbf{y} = \mathbf{G}\,\mathbf{x}$$

Where the vector of unknowns $\underline{\mathbf{x}}$ **<u>includes the user coordinates and ambiguities</u>**

C.2.1. Estimate GARR coordinates with DDL1 (using only two epochs)

C.2. PLAN-GARR differential positioning with L1 carrier (using the computed differential corrections)

The receiver was not moving (static) during the data collection.
Therefore for each epoch we have the equations system:

$$\begin{bmatrix} DDL_1^{6,3}(t_1)-DD\rho^{6,3}(t_1) \\ DDL_1^{6,7}(t_1)-DD\rho^{6,7}(t_1) \\ \cdots \\ DDL_1^{6,30}(t_1)-DD\rho^{6,30}(t_1) \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^3(t_1)-\hat{\boldsymbol{\rho}}^6(t_1)\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7(t_1)-\hat{\boldsymbol{\rho}}^6(t_1)\right)^T \\ \cdots \\ -\left(\hat{\boldsymbol{\rho}}^{30}(t_1)-\hat{\boldsymbol{\rho}}^6(t_1)\right)^T \end{bmatrix} \mathbf{dr} + \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ & & \dots & \\ 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} \lambda_1\, DDN_1^{6,3} \\ \lambda_1\, DDN_1^{6,7} \\ \cdots \\ \lambda_1\, DDN_1^{6,30} \end{bmatrix}$$

$$\mathbf{y}_1 = \mathbf{G}_1\, \mathbf{x}$$

y1:=y[t1]
G1:=G[t1]

$$\begin{bmatrix} DDL_1^{6,3}(t_2)-DD\rho^{6,3}(t_2) \\ DDL_1^{6,7}(t_2)-DD\rho^{6,7}(t_2) \\ \cdots \\ DDL_1^{6,30}(t_2)-DD\rho^{6,30}(t_2) \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^3(t_2)-\hat{\boldsymbol{\rho}}^6(t_2)\right)^T \\ -\left(\hat{\boldsymbol{\rho}}^7(t_2)-\hat{\boldsymbol{\rho}}^6(t_2)\right)^T \\ \cdots \\ -\left(\hat{\boldsymbol{\rho}}^{30}(t_2)-\hat{\boldsymbol{\rho}}^6(t_2)\right)^T \end{bmatrix} \mathbf{dr} + \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ & & \dots & \\ 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} \lambda_1\, DDN_1^{6,3} \\ \lambda_1\, DDN_1^{6,7} \\ \cdots \\ \lambda_1\, DDN_1^{6,30} \end{bmatrix}$$

$$\mathbf{y}_2 = \mathbf{G}_2\, \mathbf{x}$$

y2:=y[t2]
G2:=G[t2]

```
[DDL1-DDRho]=[Los_k - Los_06]*[dr] + [ A ]*[lambda1*DDN1]
```

C.2. PLAN-GARR differential positioning with L1 carrier (using the computed differential corrections)

In the previous computation we have not taken into account the correlations between the double differences of measurements. This matrix will be used now, as the LAMBDA method will be applied to FIX the carrier ambiguities.

$$\mathbf{P_y} = 2\sigma^2 \begin{bmatrix} 2 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & 2 \end{bmatrix}$$

a) Show that the covariance matrix of DDL1 is given by $\mathbf{P_y}$

b) Given the measurement vectors (y) and Geometry matrices (G) for two epochs

y1:=y[t1] ; G1:=G[t1] ; Py
y2:=y[t2] ; G2:=G[t2] ; Py

show that the user solution and covariance matrix can be computed as:

P=inv(G1'*W*G1+G2'*W*G2);

where: **W=inv(Py)**

x=P*(G1'*W*y1+G2'*W*y2) ;

$$\mathbf{y} = \mathbf{G}\,\mathbf{x}; \quad \mathbf{W} = \mathbf{P_y}^{-1}$$

$$\mathbf{x} = (\mathbf{G}^T\mathbf{W}\mathbf{G})^{-1}\mathbf{G}^T\mathbf{W}\mathbf{y}$$

$$\mathbf{P} = (\mathbf{G}^T\mathbf{W}\mathbf{G})^{-1}$$

C.2.1. Estimate GARR coordinates with DDL1 (using only two epochs)

C.2. PLAN-GARR differential positioning with L1 carrier (using the computed differential corrections)

The script **MakeL1DifMat.scr** builds the equations system

```
[DDL1-DDRho]=[ Los_k- Los_06]*[dr] + [ A ]*[λ1*DDN1]
```

for the two epochs required $t_1=14500$ and $t_2=14590$, using the input file **DD_PLAN_GARR_06_ALL.dat** generated before.

Execute:

```
MakeL1DifMat.scr DD_PLAN_GARR_06_ALL.dat 14500 14590
```

The **OUTPUT**
are the files **M1.dat** and **M2.dat** associated with each epoch.

Where:
the columns of files **M.dat** are the vector **y** (first column) and Matrix **G** (next columns)

C.2.1. Estimate GARR coordinates with DDL1 (using only two epochs)

C.2. PLAN-GARR differential positioning with L1 carrier (using the computed differential corrections)

1. Computing the FLOATED solution (solving the equations system).

The following procedure can be applied

```
octave

load M1.dat
load M2.dat

y1=M1(:,1);
G1=M1(:,2:13);

y2=M2(:,1);
G2=M2(:,2:13);
Py=(diag(ones(1,9))+ones(9))*2e-4;
W=inv(Py);
```

```
P=inv(G1'*W*G1+G2'*W*G2);
x=P*(G1'*W*y1+G2'*W*y2);

x(1:3)'
0.6879  -0.2712  -0.7924
```

Taking into account that the "a priori" coordinates of GARR are:

```
GARR=[4796983.5170 160309.1774
                        4187340.3887]
```

Therefore the estimated absolute coordinates of GARR are:

```
GARR+ x(1:3)'
4796984.2049 160308.9062 4187339.5963
```

C.2.1. Estimate GARR coordinates with DDL1 (using only two epochs)

C.2. PLAN-GARR differential positioning with L1 carrier (using the computed differential corrections)

2. Applying the LAMBDA method to FIX the ambiguities.

The following procedure can be applied **(justify the computations done)**
Compare the different results found.

```
octave

c=299792458;
f0=10.23e+6;
f1=154*f0;
lambda1=c/f1
 a=x(4:12)/lambda1;
 Q=P(4:12,4:12);
```

Decorrelation and integer LS search solution

```
[Qz,Zt,Lz,Dz,az,iZ] = decorrel (Q,a);
[azfixed,sqnorm] = lsearch (az,Lz,Dz,2);
afixed=iZ*azfixed;
sqnorm(2)/sqnorm(1)
ans = 1.19278115892607
afixed(:,1)'
-19337  130765326 -1759092 -1498083 130765325
   130765316  130765339  122888034  130765336
```

Rounding the floated solution directly

```
round(a)'  -19337  130765326 -1759092
 -1498083 130765325 130765316 130765339
  122888034 130765336
```

Rounding the decorrelated floated solution

```
afix=iZ*round(az);
-19337  130765326 -1759092 -1498083 130765325
   130765316  130765339  122888034  130765336
```

C.2.1. Estimate GARR coordinates with DDL1 (using only two epochs)

C.2. PLAN-GARR differential positioning with L1 carrier (using the computed differential corrections)

Questions:

1. Can the ambiguities be fixed with a certain degree of confidence?
2. Repeat the computations taking: t= 14500 and 14900.
3. Repeat the computations taking: t= 14500 and 15900.
4. Discuss why the ambiguities cannot be fixed.

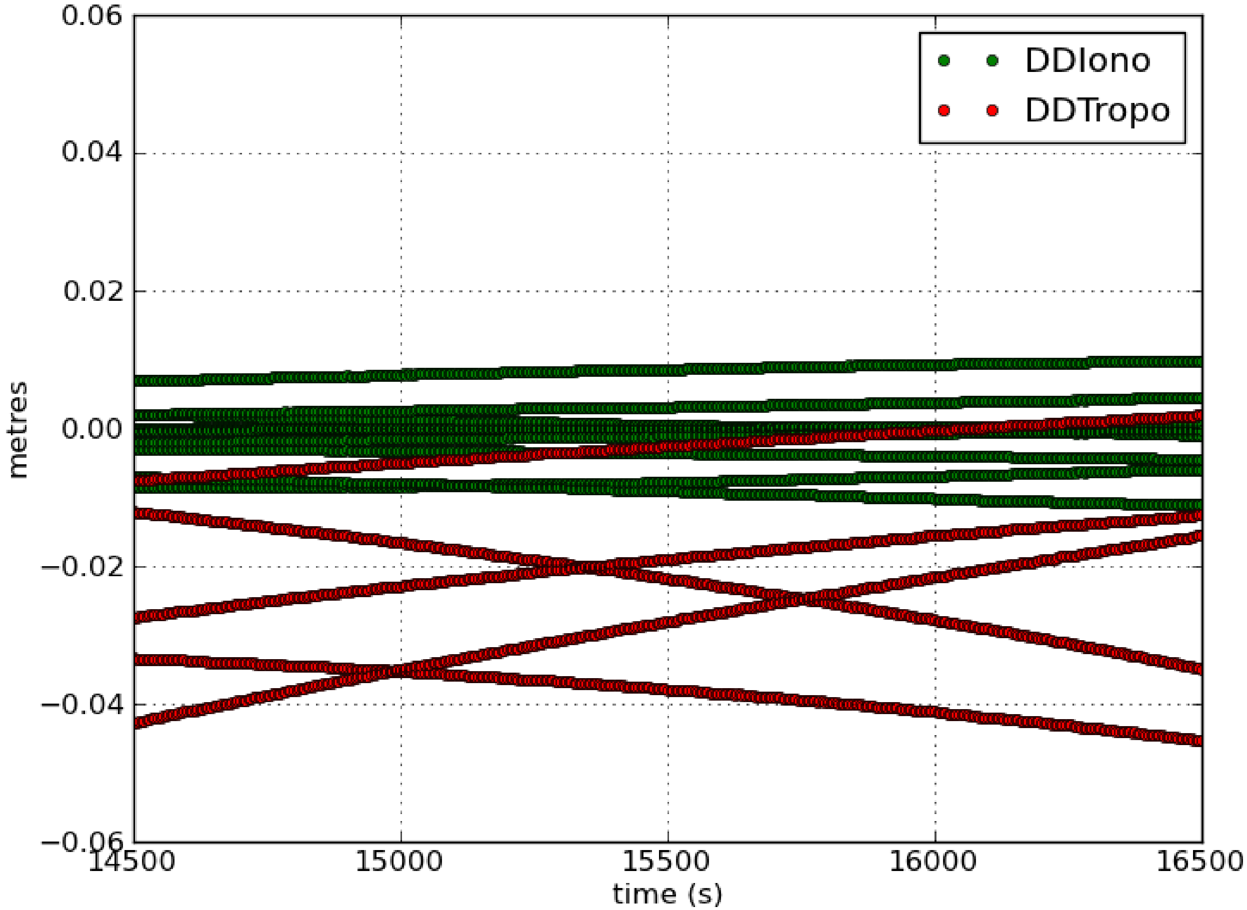
C.2. PLAN-GARR differential positioning with L1 carrier (using the computed differential corrections)

Hint:

Plot the differential Tropospheric and Ionospheric delays (from the gLAB model) and discuss their potential impact on the ambiguity fixing.

```
graph.py -f DD_PLAN_GARR_06_ALL.dat -x6 -y'13'
             -so --yn -0.06 --yx 0.06 –cl g -l "DDIono"
        -f DD_PLAN_GARR_06_ALL.dat -x6 -y'12'
             -so --cl r --yn -0.06 --yx 0.06
        -l "DDTropo" --xl "time (s)" --yl "metres"
```

C.2. PLAN-GARR differential positioning with L1 carrier (using the computed differential corrections)



C.2.1. Estimate GARR coordinates with DDL1 (using only two epochs)

C.3. PLAN-GARR differential positioning with L1 carrier

(using the computed differential corrections including troposphere)

Repeat the previous exercise, but correcting the measurements with the nominal tropospheric correction model.

C.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

C.3.1 Using DDL1 carrier measurements, estimate the coordinates of receiver GARR taking PLAN as a reference receiver and correcting troposphere.

Consider only the two epochs used in the previous exercise: t_1=14500 and t_2=14590

The following procedure can be applied:

1. **Compute the FLOATED solution**, solving the equations system with octave. Assess the accuracy of the floated solution.
2. **Apply the LAMBDA method to FIX the ambiguities**. Compare the results with the solution obtained by rounding the floated solution directly and by rounding the solution after decorrelation.
3. **Repair the DDL1 carrier measurements with the DDN1 FIXED** ambiguities and plot results to analyze the data.
4. **Compute the FIXED solution**.

C.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

The script **MakeL1DifTrpMat.scr** builds the equations system

```
[DDL1-DDRho-Trp]=[ Los_k- Los_06]*[dr] + [ A ]*[λ1*DDN1]
```

for the two epochs required $t_1=14500$ and $t_2=14590$, using the input file **DD_PLAN_GARR_06_ALL.dat** generated before.

Execute:

```
MakeL1DifTrpMat.scr DD_PLAN_GARR_06_ALL.dat 14500 14590
```

The **OUTPUT**

are the files **M1.dat** and **M2.dat** associated with each epoch.

Where:

the columns of files **M.dat** are the vector **y** (first column) and Matrix **G** (next columns)

C.3.1. Estimate GARR coordinates with **DDL1** (using tropospheric corrections)

C.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

1. Computing the FLOATED solution (solving the equations system).

The following procedure can be applied

```
octave

load M1.dat
load M2.dat

y1=M1(:,1);
G1=M1(:,2:13);

y2=M2(:,1);
G2=M2(:,2:13);
Py=(diag(ones(1,9))+ones(9))*2e-4;
W=inv(Py);
```

```
P=inv(G1'*W*G1+G2'*W*G2);
x=P*(G1'*W*y1+G2'*W*y2);

x(1:3)'
0.2140  -0.1732   0.1535
```

Taking into account that the "a priori" coordinates of GARR are:

```
GARR=[4796983.5170 160309.1774
                        4187340.3887]
```

Therefore the estimated absolute coordinates of GARR are:

```
GARR+ x(1:3)'
4796983.7310 160309.0042 4187340.5422
```

C.3.1. Estimate GARR coordinates with DDL1 (using tropospheric corrections)

C.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

2. Applying the LAMBDA method to FIX the ambiguities.

The following procedure can be applied **(justify the computations done)**
Compare the different results found.

```
octave

c=299792458;
f0=10.23e+6;
f1=154*f0;
lambda1=c/f1
 a=x(4:12)/lambda1;
 Q=P(4:12,4:12);
```

Decorrelation and integer LS search solution

```
[Qz,Zt,Lz,Dz,az,iZ] = decorrel (Q,a);
[azfixed,sqnorm] = lsearch (az,Lz,Dz,2);
afixed=iZ*azfixed;
sqnorm(2)/sqnorm(1)
ans = 2.47022808203678
afixed(:,1)'
-19333  130765338 -1759080 -1498083 130765319
     130765324  130765334 122888028 130765333
```

Rounding the floated solution directly

```
round(a)' -19334   130765336   -1759081
            -1498083   130765320 130765323
          130765334   122888029 130765334
```

Rounding the decorrelated floated solution

```
afix=iZ*round(az)
 -19333 130765338 -1759080 -1498083 130765319
      130765324 130765334 122888028 130765333
```

C.3.1. Estimate GARR coordinates with DDL1 (using tropospheric corrections)

C.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

Questions:

Can the ambiguities be fixed now?
Discuss, why?

C.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

3. Repair the DDL1 carrier measurements with the DDN1 FIXED ambiguities and plot results to analyze the data.

```
octave
amb=lambda1*afixed(:,1);
save ambL1.dat amb
```

Using the previous file **ambL1.dat** and "DD_PLAN_GARR_06_ALL.dat", generate a file with the following content:

```
----------------------------  DD_PLAN_GARR_06_ALL.fixL1 ------------------------------
  1    2   3   4   5   6    7    8    9   10    11   12    13    14  15  16  17    18
[PLAN GARR 06 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2  λ1*DDN1]
                                                                 <---- GARR ---->
-------------------------------------------------------------------------------------
```

Note: This file is identical to file "DD_PLAN_GARR_06_ALL.dat", but with the ambiguities added in the last field #18.

C.3.1. Estimate GARR coordinates with DDL1 (using tropospheric corrections)

C.3. PLAN-GARR differential positioning with L1 carrier

(using the computed differential corrections including troposphere)

a) Generate a file with the satellite PRN number and the ambiguities:

```
grep -v \# ambL1.dat > na1
cat DD_PLAN_GARR_06_ALL.dat|gawk '{print $4}'|sort -nu|gawk '{print $1,NR}' >sat.lst
paste sat.lst na1 > sat.ambL1
```

b) Generate the "DD_PLAN_GARR_06_ALL.fixL1" file:

```
cat DD_PLAN_GARR_06_ALL.dat|
                  gawk 'BEGIN{for (i=1;i<1000;i++) {getline <"sat.ambL1";A[$1]=$3}}
  {printf "%s %02i %02i %s %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f
                                          %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f \n",
                 $1,$2,$3,$4,$5,$6,$7,$8,$9,$10,$11,$12,$13,$14,$15,$16,$17,A[$4]}' >
                                                                DD_PLAN_GARR_06_ALL.fixL1
```

C.3. PLAN-GARR differential positioning with L1 carrier (using the computed differential corrections including troposphere)

```
----------------------------  DD_PLAN_GARR_06_ALL.fixL1 -------------------------------
   1    2   3   4   5   6    7    8    9   10     11    12     13    14  15  16  17      18
[PLAN GARR 06 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrp DDIon El1 Az1 El2 Az2  λ1*DDN1]
                                                                   <---- GARR ---->
----------------------------------------------------------------------------------------
```

c) Make and discuss the following plots

```
graph.py -f DD_PLAN_GARR_06_ALL.fixL1 -x6 -y'($8-$18-$11)'
       -so --yn -0.6 --yx 0.6 -l "(DDL1-lambda1*DDN1)-DDRho" --xl "time (s)" --yl "m"
```

```
graph.py -f DD_PLAN_GARR_06_ALL.fixL1 -x6 -y'($8-$18-$11-$12)'
 -so --yn -0.6 --yx 0.6 -l "(DDL1-lambda1*DDN1)-DDRho-DDTrp" --xl "time (s)" --yl "m"
```

```
graph.py -f DD_PLAN_GARR_06_ALL.fixL1 -x6 -y'($8-$18-$11-$12)'
 -so --yn -0.06 --yx 0.06 -l "(DDL1-lambda1*DDN1)-DDRho-DDTrp" --xl "time (s)" --yl "m"
```

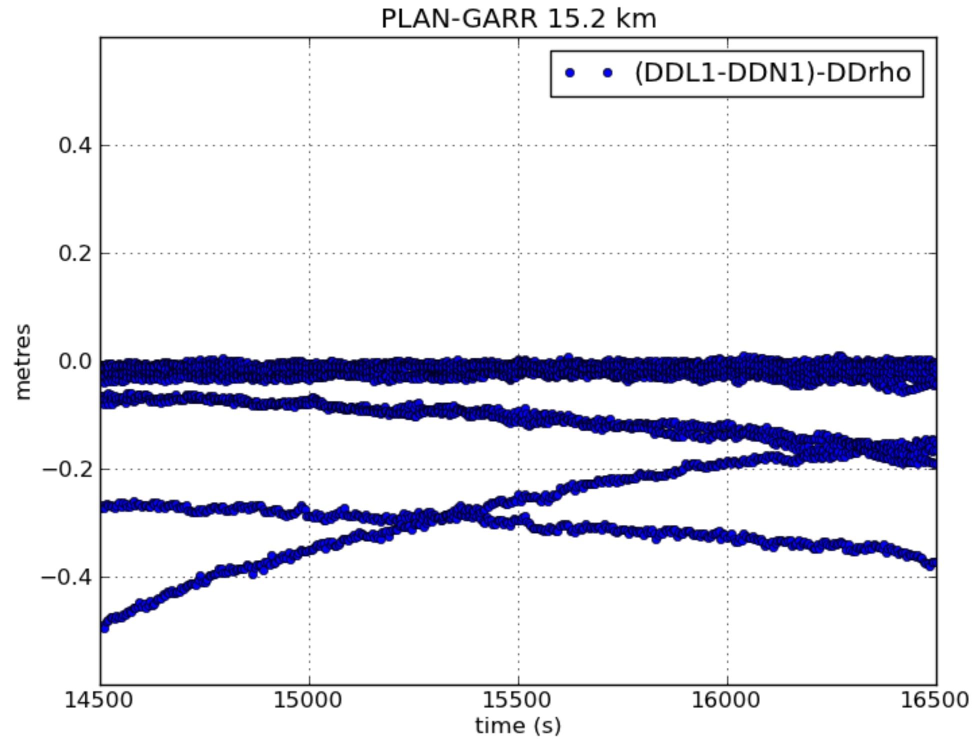
```
graph.py -f DD_PLAN_GARR_06_ALL.fixL1 -x6 -y'($8-$18)'
         -so --yn -15000 --yx 15000 -l "(DDL1-lambda1*DDN1)" --xl "time (s)" --yl "m"
```

C.3.1. Estimate GARR coordinates with DDL1 (using tropospheric corrections)

C.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

```
--------------------------- DD_PLAN_GARR_06_ALL.fixL1 ---------------------------
   1     2    3    4    5    6     7     8     9    10     11     12     13    14   15   16   17     18
[PLAN GARR 06 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2  λ1*DDN1]
                                                                     <---- GARR ---->
----------------------------------------------------------------------------------------
```

```
graph.py -f DD_PLAN_GARR_06_ALL.fixL1
         -x6 -y'($8-$18-$11)'
         -so --yn -0.6 --yx 0.6
         -l "(DDL1-λ1*DDN1)-DDRho"
         --xl "time (s)" --yl "m"
```



Questions:
Explain what is the meaning of this plot.

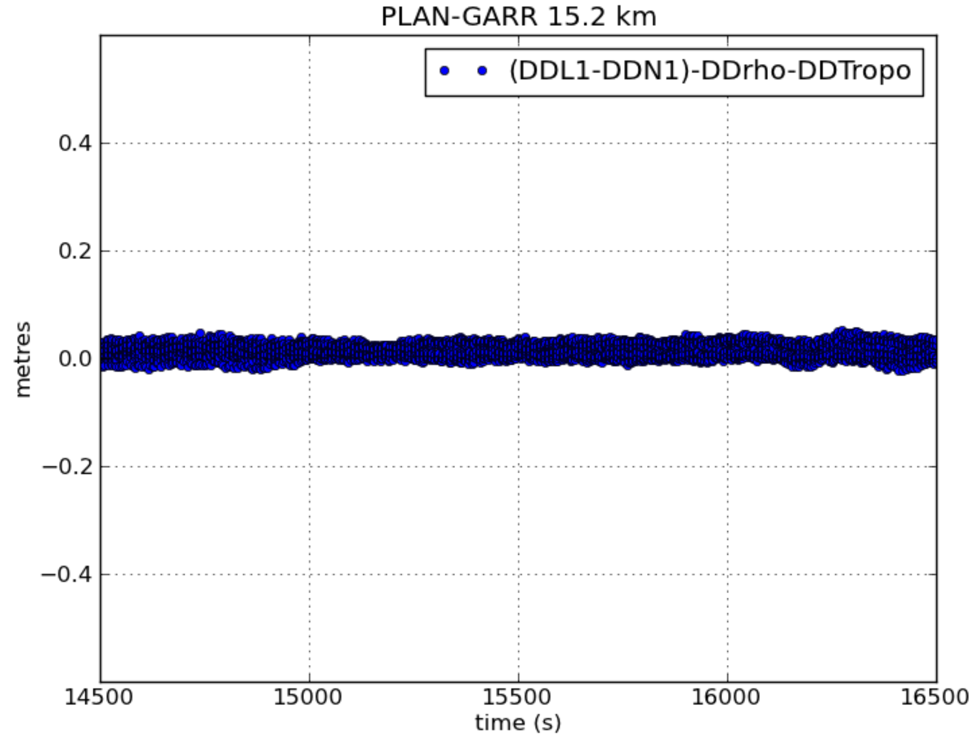
C.3.1. Estimate GARR coordinates with DDL1 (using tropospheric corrections)

C.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

```
----------------------------  DD_PLAN_GARR_06_ALL.fixL1  -------------------------------
  1     2    3    4    5    6     7     8     9    10     11    12     13     14   15   16   17      18
[PLAN GARR 06 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2  λ1*DDN1]
                                                                      <---- GARR ---->
-----------------------------------------------------------------------------------------
```

```
graph.py -f DD_PLAN_GARR_06_ALL.fixL1
         -x6 -y'($8-$18-$11-$12)'
         -so --yn -0.6 --yx 0.6
         -l "(DDL1-λ1*DDN1)-DDRho-DDTrp"
         --xl "time (s)" --yl "m"
```

Questions:
Explain what is the meaning of this plot.



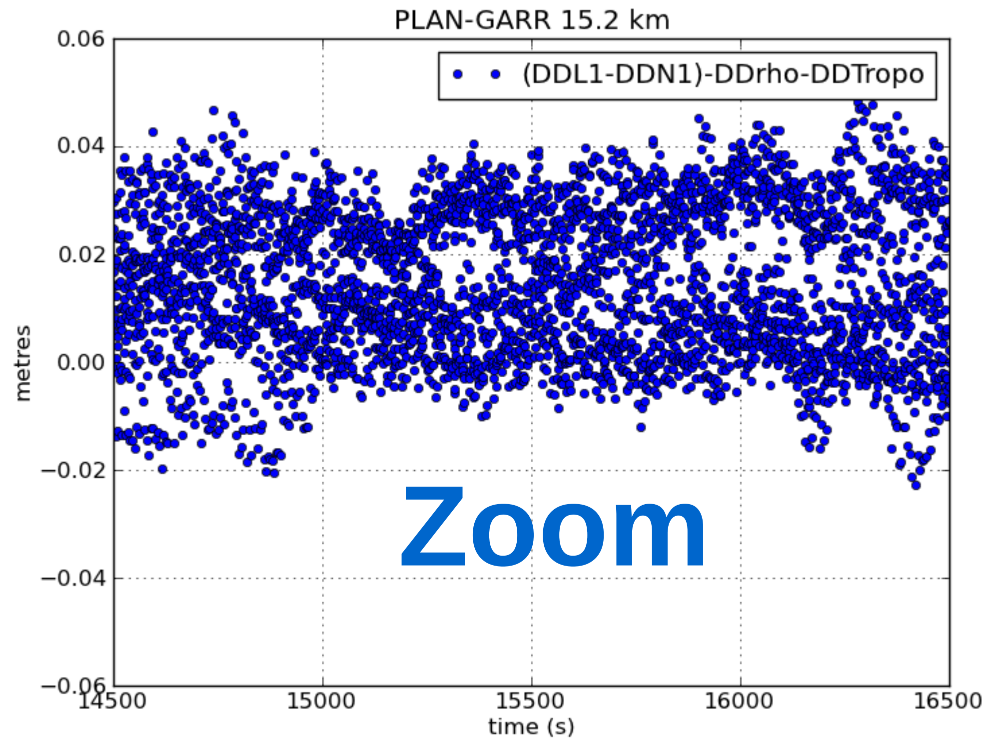
C.3.1. Estimate GARR coordinates with DDL1 (using tropospheric corrections)

C.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

```
----------------------------  DD_PLAN_GARR_06_ALL.fixL1 ------------------------------------
   1     2    3    4    5    6     7     8     9    10     11    12     13     14   15   16   17     18
[PLAN GARR 06 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2  λ1*DDN1]
                                                                          <---- GARR ---->
--------------------------------------------------------------------------------------------
```

```
graph.py -f DD_PLAN_GARR_06_ALL.fixL1
         -x6 -y'($8-$18-$11-$12)'
         -so --yn -0.06 --yx 0.06
         -l "(DDL1-λ1*DDN1)-DDRho-DDTrp"
         --xl "time (s)" --yl "m"
```

Questions:
Explain what is the meaning of this plot.

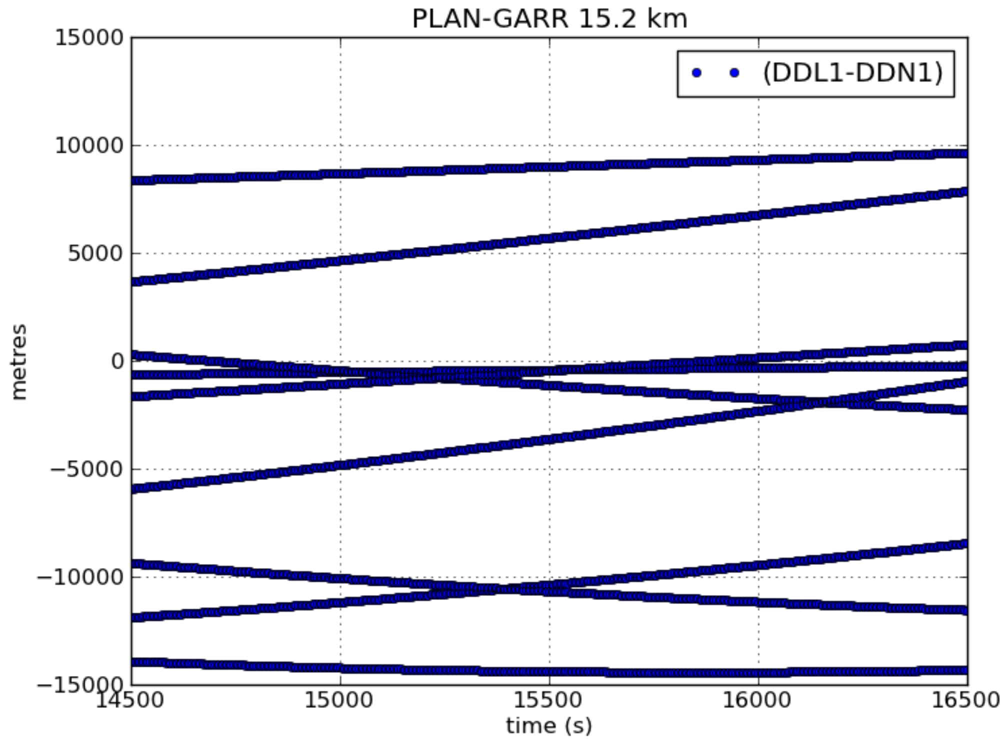


C.3.1. Estimate GARR coordinates with DDL1 (using tropospheric corrections)

C.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

```
---------------------------- DD_PLAN_GARR_06_ALL.fixL1 ----------------------------
  1     2    3    4    5    6     7     8     9    10     11    12     13     14   15   16   17     18
[PLAN GARR 06 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2  λ1*DDN1]
                                                                   <---- GARR ---->
-----------------------------------------------------------------------------------
```

```
graph.py -f DD_PLAN_GARR_06_ALL.fixL1
         -x6 -y'($8-$18)'
         -so --yn -15000 --yx 15000
         -l "(DDL1-λ1*DDN1)"
         --xl "time (s)" --yl "m"
```



Questions:
Explain what is the meaning of this plot.

C.3.1. Estimate GARR coordinates with DDL1 (using tropospheric corrections)

C.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

4. Computing the FIXED solution (after FIXING ambiguities).

The following procedure can be applied

a) Build the equations system

```
[DDL1-DDRho-DDTrp-lambda1*DDN1]=[Los_k - Los_06]*[dr]
```

```
cat DD_IND1_IND2_06_ALL.fixL1 | gawk 'BEGIN{g2r=atan2(1,1)/45}
                    {e1=$14*g2r;a1=$15*g2r;e2=$16*g2r;a2=$17*g2r;
 printf "%14.4f %8.4f %8.4f %8.4f \n",
  $8-$11-$12-$18, -cos(e2)*sin(a2)+cos(e1)*sin(a1),
         -cos(e2)*cos(a2)+cos(e1)*cos(a1), -sin(e2)+sin(e1)}' > M.dat
```

C.3.1. Estimate GARR coordinates with DDL1 (using tropospheric corrections)

C.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

Solve the equations system using octave (or MATLAB) and assess the estimation error:

```
octave

load M.dat

y=M(:,1);
G=M(:,2:4);

x=inv(G'*G)*G'*y
x
 -0.00290189178833524
  0.00354027112026342
  0.04277612243282228
```

Absolute coordinates of GARR.

Taking into account that the "a priori" coordinates of IND2 are:

```
GARR=[4796983.5170 160309.1774 4187340.3887]
```

Therefore the estimated absolute coordinates of GARR are:

```
GARR+ x(1:3)'
 ans= 4796983.5141 160309.1809 4187340.4315
```

C.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

C.3.2. Using the DDL1 carrier with the ambiguities FIXED, compute the LS single epoch solution for the whole interval 14500< t <16500 with the program LS.f

Note: The program `LS.f` computes the Least Square solution for each measurement epoch of the input file (see the FORTRAN code "LS.f")

The following procedure can be applied

a) generate a file with the following content;

```
[Time],[DDL1-DDRho-DDTrp-lambda1*DDN1],[Los_k - Los_06]
```

where:

Time= seconds of day

DDL1 – DDRho –DDTrp – lambda1*DDN1= Prefit residulas (i.e., "y" values in program LS.f)

Los_k – Los_06 = The three components of the geometry matrix (i.e., matrix "a" in program LS.f)

C.3.2. Estimate GARR coordinates with DDL1 (using tropospheric corrections)

C.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

```
[Time], [DDL1-DDRho-DDTrp-lambda1*DDN1], [Los_k - Los_06]
```

The following sentence can be used

```
cat DD_IND1_IND2_06_ALL.fixL1 | gawk 'BEGIN{g2r=atan2(1,1)/45}
  {e1=$14*g2r;a1=$15*g2r;e2=$16*g2r;a2=$17*g2r;;printf "%s %14.4f
   %8.4f %8.4f %8.4f \n",$6,$8-$11-$12-$18,-cos(e2)*sin(a2)+cos(e1)*sin(a1),
        -cos(e2)*cos(a2)+cos(e1)*cos(a1),-sin(e2)+sin(e1)}' > L1model.dat
```

b) Compute the Least Squares solution

```
cat L1model.dat |LS > L1fix.pos
```

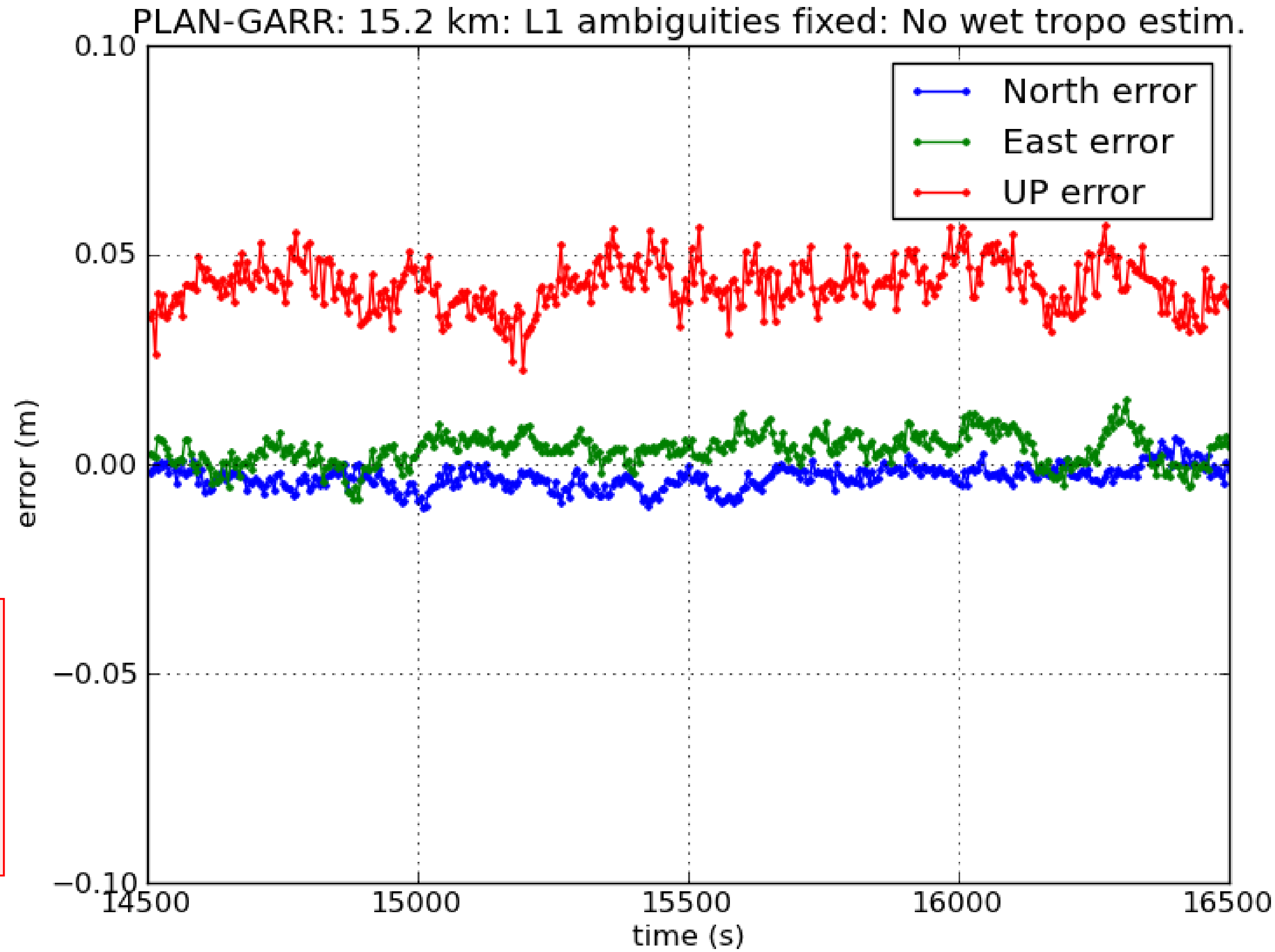
C.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

Plot the baseline estimation error

```
graph.py -f L1fix.pos -x1 -y2 -s.- -l "North error"
         -f L1fix.pos -x1 -y3 -s.- -l "East error"
         -f L1fix.pos -x1 -y4 -s.- -l "UP error"
  --yn -.1 --yx .1 --xl "time (s)" --yl "error (m)" -t "PLAN-GARR: 15.2
km: L1 ambiguities fixed: No wet tropo estim."
```

C.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

Differential Positioning error after fixing ambiguities



Question:
Discuss the possible sources of the bias found in the vertical component.

C.3. PLAN-GARR differential positioning with L1 carrier

(using the computed differential corrections including troposphere)

Remember that the reference coordinates have been taken relative to the Antenna Phase Centre in the ionosphere-free combination LC

Question:

Taking into account the following parameters of the PLAN and GARR antennas, **calculate the relative error introduced by the difference between the L1 and LC APC of both receivers** in the differential positioning.

According to the RINEX and ANTEX files, we have:

PLAN: TRM55971.00 (millimetres)

G01	1.29	-0.19	66.73	NORTH / EAST / UP
G02	0.38	0.61	57.6	

GARR: TRM41249.00 (millimetres)

G01	0.28	0.49	55.91	NORTH / EAST / UP
G02	0.15	0.46	58.00	

C.3.2. Estimate GARR coordinates with DDL1 (APC error effect)

C.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

Solution:

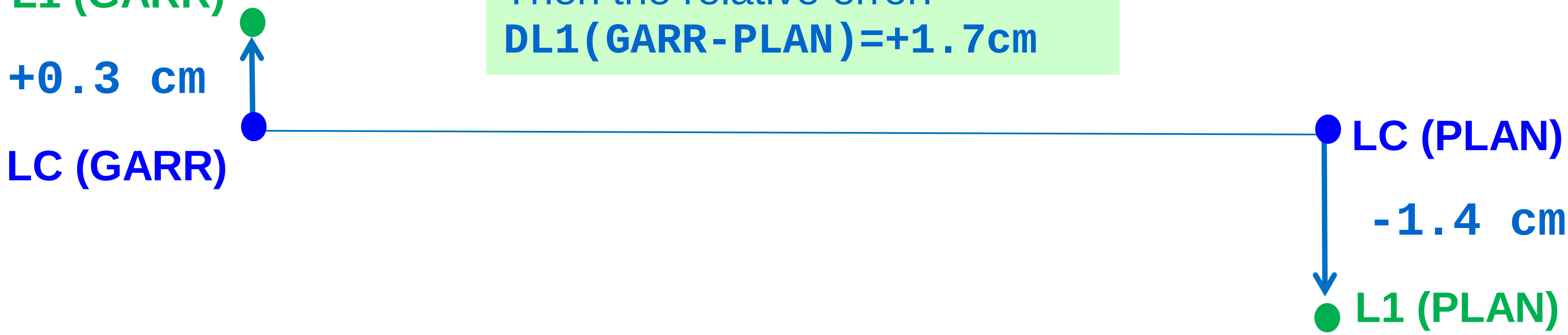
GARR:
`dL1=5.591 cm`
`dL2=5.800 cm`
`dLC=1/(g-1)*(g*dL1-dL2)=5.3cm`

Then:
`dL1=dLC+0.3 cm`

PLAN:
`dL1=6.673 cm`
`dL2=5.769 cm`
`dLC=1/(g-1)*(g*dL1-dL2)=8.1cm`

Then:
`dL1=dLC-1.4 cm`

Then the relative error:
`DL1(GARR-PLAN)=+1.7cm`



C.3.2. Estimate GARR coordinates with DDL1 (APC error effect)

C.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

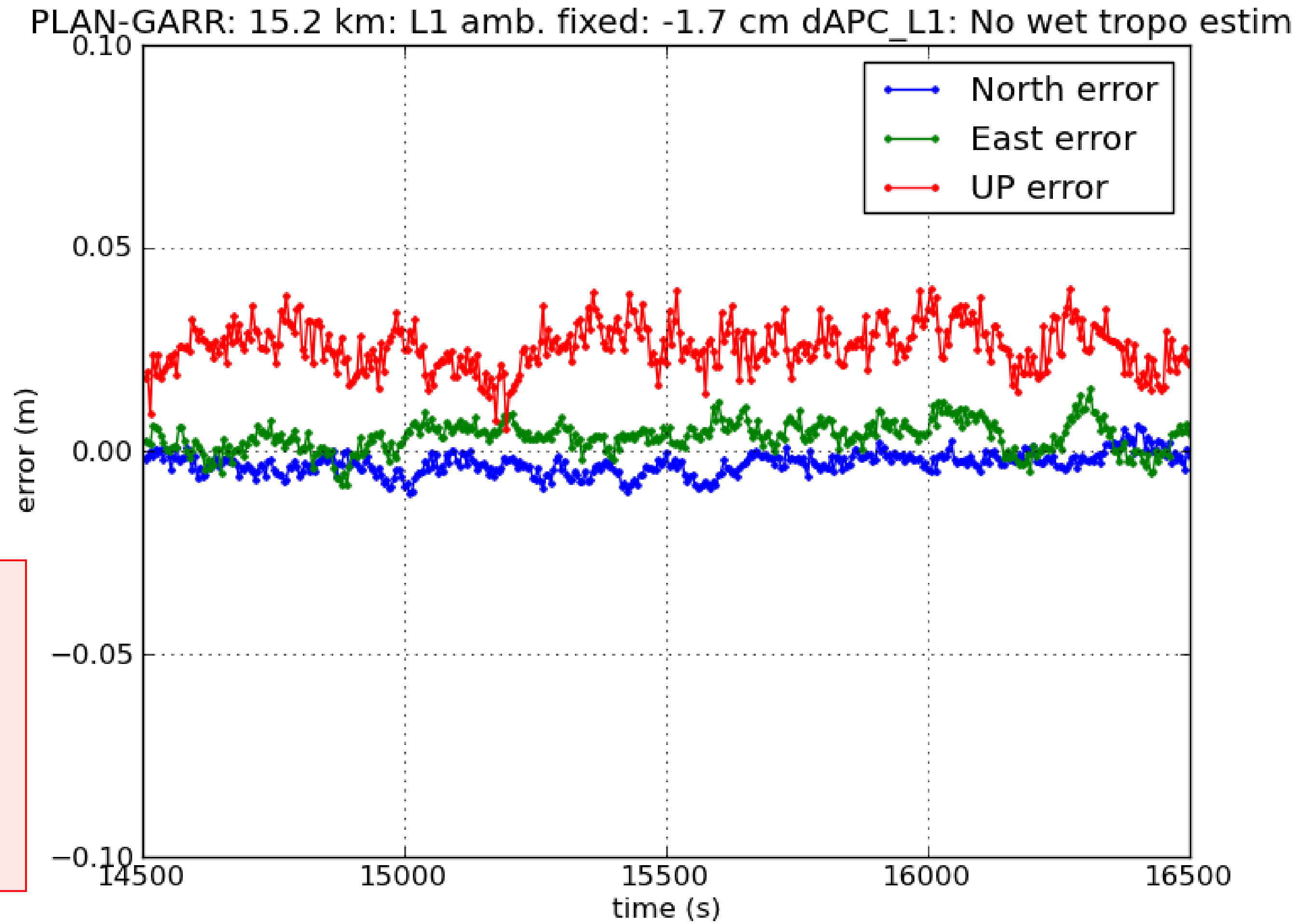
Repeat the positioning error plot, but correcting for the relative Antenna Phase Centres (APCs):

```
graph.py -f L1fix.pos -x1 –y2 -s.- -l "North error"
         -f L1fix.pos -x1 –y3 -s.- -l "East error"
         -f L1fix.pos -x1 –y '($4-0.017)' -s.- -l "UP error"
  --yn -.1 --yx .1 --xl "time (s)" --yl "error (m)" -t "PLAN-GARR: 15.2
km: L1 amb. fixed: -1.7 cm dAPC_L1: No wet tropo estim."
```


C.3.2. Estimate GARR coordinates with DDL1 (APC error effect)

C.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

Differential Positioning error after fixing ambiguities



Question:
Discuss on the remaining error sources which could explain the error bias found in the vertical component.

C.3.2. Estimate GARR coordinates with DDL1 (APC error effect)

C.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

C.3.3. Repeat the previous computations, but using the Unsmoothed code P1.
i.e., compute the LS single epoch solution for the whole interval 145000< t <165000 with the program `LS.f`

The same procedure as in previous case can be applied, but using the code DDP1 instead of the carrier "DDL1 – lambda1*DDN1"

a) generate a file with the following content;

```
[Time], [DDP1-DDRho-DDTrp], [ Los_k - Los_06]
```

where:
Time= seconds of day
DDP1 – DDRho-DDTrp= Prefit residulas (i.e., "y" values in program lms1)
Los_k – Los_06 = The three components of the geometry matrix (i.e., matrix "a" in program LS.f)

C.3.3. Estimate GARR coordinates
(using tropospheric corrections)

C.3. PLAN-GARR differential positioning
(using the computed differential corrections including troposphere)

```
[Time], [DDP1-DDRho-DDTrp], [Los_k - Los_06]
```

The following sentence can be used

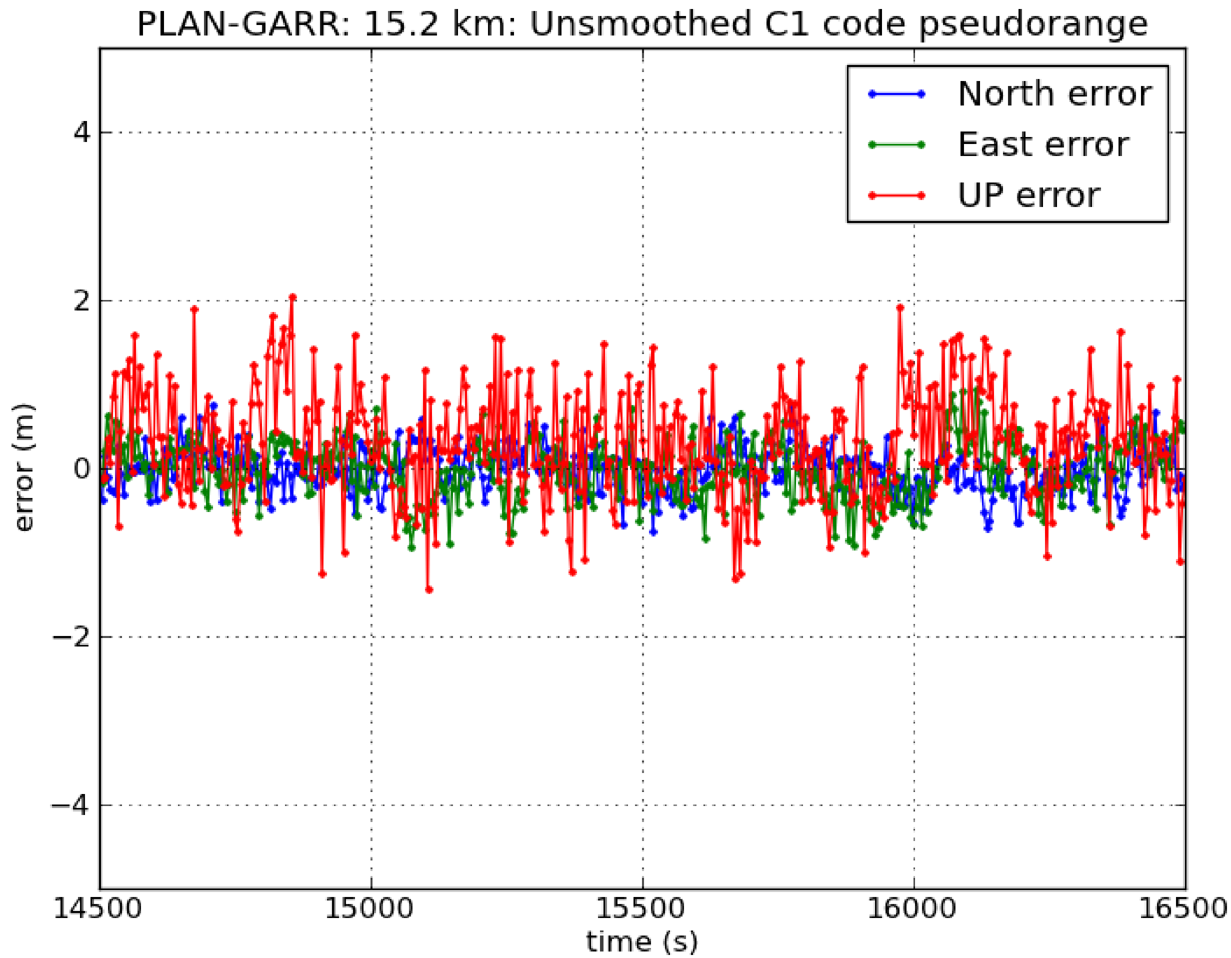
```
cat DD_IND1_IND2_06_ALL.fixL1 | gawk 'BEGIN{g2r=atan2(1,1)/45}
  {e1=$14*g2r;a1=$15*g2r;e2=$16*g2r;a2=$17*g2r;;printf "%s %14.4f
   %8.4f %8.4f %8.4f \n",$6,$7-$11-$12,-cos(e2)*sin(a2)+cos(e1)*sin(a1),
        -cos(e2)*cos(a2)+cos(e1)*cos(a1),-sin(e2)+sin(e1)}' > P1model.dat
```

b) Compute the Least Squares solution

```
cat P1model.dat |LS > P1.pos
```

C.3. PLAN-GARR differential positioning
(using the computed differential corrections including troposphere)

Positioning error with the unsmoothed code



Question:
Discuss the results by comparing them with the previous ones with DDL1 carrier.

OVERVIEW

- **Introduction:** gLAB processing in command line
- **Preliminary computations:** data files & reference values
- **Session A:** Differential positioning of IND2-IND3 receivers
  (baseline: 18 metres)
- **Session B:** Differential positioning of IND1-IND2 receivers
  (baseline: 7 metres, but synchronization errors)
- **Session C:** Differential positioning of PLAN-GARR receivers
  (baseline: 15 km, Night time): tropospheric effects
- **Session D:** Differential positioning of PLAN-GARR receivers
  (baseline: 15 km, Day time): tropospheric and Ionospheric effects

Session D

Differential positioning of PLAN- GARR receivers

(baseline: 15 km. Day time)

Analysis of differential tropospheric and Ionospheric error effects

D. Differential positioning of PLAN-GARR receivers

- The previous session has been carried out using measurements collected during the night time, when the effect of the ionosphere is lower.
- The effect of the ionosphere over a baseline of 15km (and in solar maximum conditions) will be assessed in this session using day-time measurements.
- The exercise will end with the computation of the unambiguous DDSTEC from dual-frequency carrier measurements (after fixing the ambiguities in both carriers).
- The solutions computed using the DDL1 carrier (with the ambiguity fixed) corrected by the unambiguous DDSTEC and corrected by the nominal Klobuchar model will be compared.
- Finally, the solution using the unambiguous DDLC carrier (iono-free combination) will be also computed to compare results.

D.1 Measurements selection

Selecting measurements: Time interval [39000:41300]

- Select the satellites within the time interval [39000:41300]. Exclude satellites PRN01 and PRN31 in order to have the same satellites over the whole interval

```
cat ObsFile.dat|gawk '{if ($4>=39000 && $4<=41300 && $2!=01 && $2!=31)
                                                          print $0}' > obs.dat
```

- Reference satellite (over the time interval [39000:41300])

Confirm that the satellite **PRN13** is the satellite with the highest elevation **(this satellite will be used as the reference satellite)**

obs.dat ➔

```
  1    2    3    4    5  6  7  8   9    10   11   12   13
[sta sat DoY sec P1 L1 P2 L2 Rho Trop Ion elev azim]
```

D.2. Computing Double Differences

Compute the double differences between receivers **PLAN (reference)** and GARR and satellites **PRN13 (reference)** and [PRN 02, 04, 07, 10, 17, 20, 23, 32]

```
DDobs.scr obs.dat PLAN GARR 13 02
DDobs.scr obs.dat PLAN GARR 13 04
DDobs.scr obs.dat PLAN GARR 13 07
DDobs.scr obs.dat PLAN GARR 13 10
DDobs.scr obs.dat PLAN GARR 13 17
DDobs.scr obs.dat PLAN GARR 13 20
DDobs.scr obs.dat PLAN GARR 13 23
DDobs.scr obs.dat PLAN GARR 13 32
```

Merge the files into a single file and sort by time:

```
cat DD_PLAN_GARR_13_??.dat|sort -n -k +6 > DD_PLAN_GARR_13_ALL.dat
```

D.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

D.3.1 Using DDL1 carrier measurements, estimate the coordinates of receiver GARR taking PLAN as a reference receiver and correcting troposphere.

Consider only the two epochs used in the previous exercise: $t_1=39000$ and $t_2=40500$.

The following procedure can be applied:

1. **Compute the FLOATED solution**, solving the equations system with octave. Assess the accuracy of the floated solution.
2. **Apply the LAMBDA method to FIX the ambiguities**. Compare the results with the solution obtained by rounding the floated solution directly and by rounding the solution after decorrelation.
3. **Repair the DDL1 carrier measurements with the DDN1 FIXED** ambiguities and plot results to analyze the data.
4. **Compute the FIXED solution**.

D.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

The script **MakeL1DifTrpMat.scr** builds the equations system

```
[DDL1-DDRho-Trp]=[ Los_k- Los_13]*[dr] + [ A ]*[λ1*DDN1]
```

for the two epochs required t_1=39000 and t_2=40500, using the input file **DD_PLAN_GARR_13_ALL.dat** generated before.

Execute:

```
MakeL1DifTrpMat.scr DD_PLAN_GARR_13_ALL.dat 39000 40500
```

The **OUTPUT**
are the files **M1.dat** and **M2.dat** associated with each epoch.

Where:
the columns of files **M.dat** are the vector **y** (first column) and Matrix **G** (next columns)

D.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

1. Computing the FLOATED solution (solving the equations system).

The following procedure can be applied

```
octave

load M1.dat
load M2.dat

y1=M1(:,1);
G1=M1(:,2:12);

y2=M2(:,1);
G2=M2(:,2:12);
Py=(diag(ones(1,8))+ones(8))*2e-4;
W=inv(Py);
```

```
P=inv(G1'*W*G1+G2'*W*G2);
x=P*(G1'*W*y1+G2'*W*y2);

x(1:3)'
 -0.3262  0.0268   0.09012
```

Taking into account that the ”a priori” coordinates of GARR are:

```
GARR=[4796983.5170 160309.1774
                        4187340.3887]
```

Therefore the estimated absolute coordinates of GARR are:

```
GARR+ x(1:3)'
4796983.1908 160309.2042 4187340.4789
```


D.3.1. Estimate GARR coordinates with DDL1 (using tropospheric corrections)

D.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

2. Applying the LAMBDA method to FIX the ambiguities.

The following procedure can be applied **(justify the computations done)**
Compare the different results found.

```
octave

c=299792458;
f0=10.23e+6;
f1=154*f0;
lambda1=c/f1
 a=x(4:11)/lambda1;
 Q=P(4:11,4:11);
```

Decorrelation and integer LS search solution

```
[Qz,Zt,Lz,Dz,az,iZ] = decorrel (Q,a);
[azfixed,sqnorm] = lsearch (az,Lz,Dz,2);
afixed=iZ*azfixed;
sqnorm(2)/sqnorm(1)
ans = 3.54169992923790
afixed(:,1)'
      -1372641    1731966    2313787     592316
       -878242    -401400    -475026    1855925
```

Rounding the floated solution directly

```
round(a)'
      -1372640    1731967    2313786     592317
       -878241    -401401    -475027    1855923
```

Rounding the decorrelated floated solution

```
afix=iZ*round(az)
      -1372641    1731966    2313787     592316
       -878242    -401400    -475026    1855925
```


D.3.1. Estimate GARR coordinates with DDL1 (using tropospheric corrections)

D.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

3. Repair the DDL1 carrier measurements with the DDN1 FIXED ambiguities and plot results to analyze the data.

```
octave
amb=lambda1*afixed(:,1);
save ambL1.dat amb
```

Using the previous the file **ambL1.dat** and "DD_PLAN_GARR_13_ALL.dat", generate a file with the following content:

```
----------------------------  DD_PLAN_GARR_13_ALL.fixL1 ----------------------------------
   1     2    3    4    5    6     7     8     9    10     11    12     13     14   15   16   17      18
[PLAN GARR 13 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2  λ1*DDN1]
                                                                  <---- GARR ---->
-----------------------------------------------------------------------------------------
```

Note: This file is identical to file "DD_PLAN_GARR_13_ALL.dat", but with the ambiguities added in the last field #18.

D.3.1. Estimate GARR coordinates with DDL1 (using tropospheric corrections)

D.3. PLAN-GARR differential positioning with L1 carrier

(using the computed differential corrections including troposphere)

a) Generate a file with the satellite PRN number and the ambiguities:

```
grep -v \# ambL1.dat > na1
cat DD_PLAN_GARR_13_ALL.dat|gawk '{print $4}'|sort -nu|gawk '{print $1,NR}' >sat.lst
paste sat.lst na1 > sat.ambL1
```

b) Generate the "DD_PLAN_GARR_13_ALL.fixL1" file:

```
cat DD_PLAN_GARR_13_ALL.dat|
                  gawk 'BEGIN{for (i=1;i<1000;i++) {getline <"sat.ambL1";A[$1]=$3}}
  {printf "%s %02i %02i %s %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f
                                        %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f \n",
                 $1,$2,$3,$4,$5,$6,$7,$8,$9,$10,$11,$12,$13,$14,$15,$16,$17,A[$4]}' >
                                                                 DD_PLAN_GARR_13_ALL.fixL1
```

D.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

```
----------------------------  DD_PLAN_GARR_13_ALL.fixL1 ------------------------------------
   1    2   3   4   5   6    7    8    9   10    11    12    13    14  15  16  17     18
[PLAN GARR 13 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrp DDIon El1 Az1 El2 Az2  λ1*DDN1]
                                                                 <---- GARR ---->
-------------------------------------------------------------------------------------------
```

c) Make and discuss the following plots

```
graph.py -f DD_PLAN_GARR_13_ALL.fixL1 -x6 -y'($8-$18-$11)'
       -so --yn -0.6 --yx 0.6 -l "(DDL1-lambda1*DDN1)-DDRho" --xl "time (s)" --yl "m"
```

```
graph.py -f DD_PLAN_GARR_13_ALL.fixL1 -x6 -y'($8-$18-$11-$12)'
-so --yn -0.6 --yx 0.6 -l "(DDL1-lambda1*DDN1)-DDRho-DDTrp " --xl "time (s)" --yl "m"
```

```
graph.py -f DD_PLAN_GARR_13_ALL.fixL1 -x6 -y'($8-$18-$12)'
    -so --yn -0.06 --yx 0.16 -l "(DDL1-lambda1*DDN1)-DDTrp " --xl "time (s)" --yl "m"
```

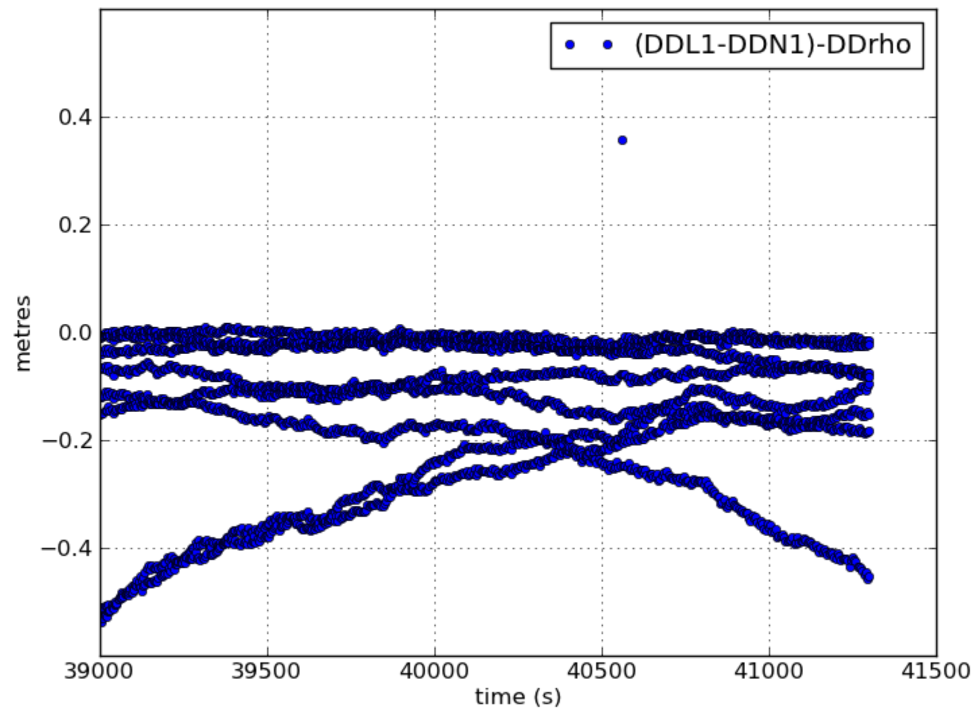
```
graph.py -f DD_PLAN_GARR_13_ALL.fixL1 -x6 -y'($8-$18)'
         -so --yn -15000 --yx 15000 -l "(DDL1-lambda1*DDN1)" --xl "time (s)" --yl "m"
```

D.3.1. Estimate GARR coordinates with DDL1 (using tropospheric corrections)

D.3. PLAN-GARR differential positioning with L1 carrier (using the computed differential corrections including troposphere)

```
----------------------------  DD_PLAN_GARR_13_ALL.fixL1 ------------------------------------
   1     2    3    4    5    6     7     8     9    10     11    12     13     14   15   16   17     18
[PLAN GARR 13 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2  λ1*DDN1]
                                                                    <---- GARR ---->
--------------------------------------------------------------------------------------------
```

```
graph.py -f DD_PLAN_GARR_13_ALL.fixL1
         -x6 -y'($8-$18-$11)'
         -so --yn -0.6 --yx 0.6
         -l "(DDL1-λ1*DDN1)-DDRho"
         --xl "time (s)" --yl "m"
```



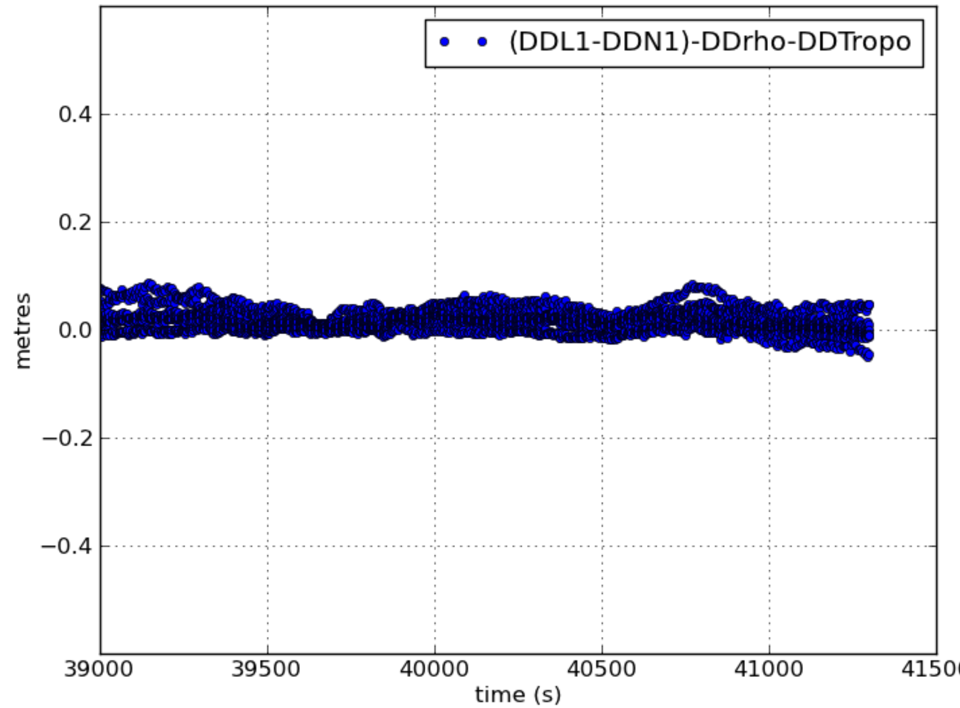
Questions:
Explain what is the meaning of this plot.

D.3.1. Estimate GARR coordinates with DDL1 (using tropospheric corrections)

D.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

```
---------------------------- DD_PLAN_GARR_13_ALL.fixL1 ----------------------------------
   1     2    3    4    5    6     7     8     9    10     11    12     13     14   15   16   17      18
[PLAN GARR 13 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2  λ1*DDN1]
                                                                        <---- GARR ---->
------------------------------------------------------------------------------------------
```

```
graph.py -f DD_PLAN_GARR_13_ALL.fixL1
         -x6 -y'($8-$18-$11-$12)'
         -so --yn -0.6 --yx 0.6
         -l "(DDL1-λ1*DDN1)-DDRho-DDTrp"
         --xl "time (s)" --yl "m"
```



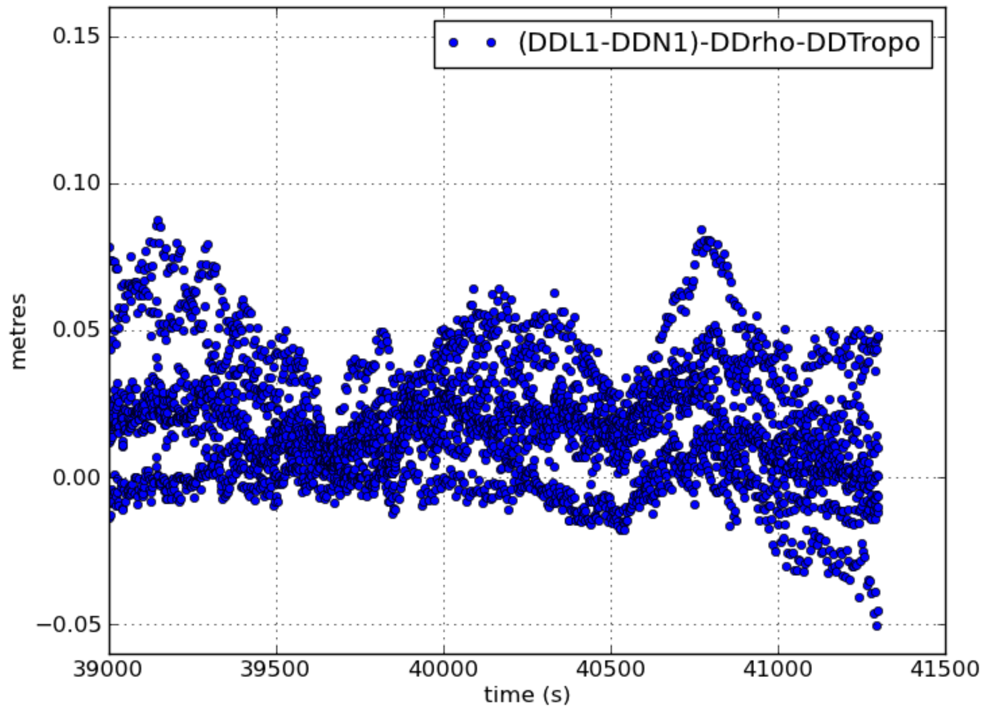
Questions:
Explain what is the meaning of this plot.

D.3.1. Estimate GARR coordinates with DDL1 (using tropospheric corrections)

D.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

```
----------------------------  DD_PLAN_GARR_13_ALL.fixL1 ------------------------------------
   1     2    3    4    5    6     7     8     9    10     11    12     13     14   15   16   17     18
[PLAN GARR 13 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2  λ1*DDN1]
                                                                        <---- GARR ---->
--------------------------------------------------------------------------------------------
```

```
graph.py -f DD_PLAN_GARR_13_ALL.fixL1
         -x6 -y'($8-$18-$11-$12)'
         -so --yn -0.06 --yx 0.16
         -l "(DDL1-λ1*DDN1)-DDRho-DDTrp"
         --xl "time (s)" --yl "m"
```



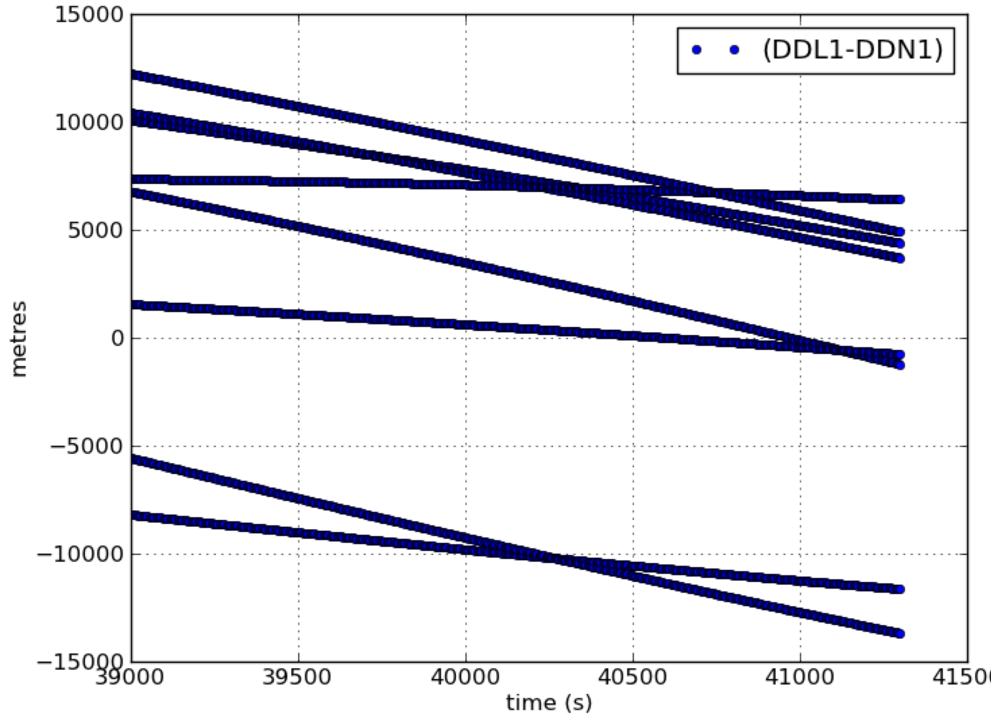
Questions:
Explain what is the meaning of this plot.

D.3.1. Estimate GARR coordinates with DDL1 (using tropospheric corrections)

D.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

```
----------------------------  DD_PLAN_GARR_13_ALL.fixL1 ------------------------------------
   1     2    3    4    5    6     7     8     9    10     11    12     13     14   15   16   17      18
[PLAN GARR 13 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2  λ1*DDN1]
                                                                       <---- GARR ---->
-----------------------------------------------------------------------------------------------
```

```
graph.py -f DD_PLAN_GARR_13_ALL.fixL1
         -x6 -y'($8-$18)'
         -so --yn -15000 --yx 15000
         -l "(DDL1-λ1*DDN1)"
         --xl "time (s)" --yl "m"
```



Questions:
Explain what is the meaning of this plot.

D.3.1. Estimate GARR coordinates with DDL1 (using tropospheric corrections)

D.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

4. Computing the FIXED solution for the whole interval (after FIXING ambiguities).

The following procedure can be applied

a) Build the equations system

```
[DDL1-DDRho-DDTrp-lambda1*DDN1]=[Los_k - Los_13]*[dr]
```

```
cat DD_PLAN_GARR_13_ALL.fixL1 | gawk 'BEGIN{g2r=atan2(1,1)/45}
                    {e1=$14*g2r;a1=$15*g2r;e2=$16*g2r;a2=$17*g2r;
 printf "%14.4f %8.4f %8.4f %8.4f \n",
  $8-$11-$12-$18, -cos(e2)*sin(a2)+cos(e1)*sin(a1),
          -cos(e2)*cos(a2)+cos(e1)*cos(a1), -sin(e2)+sin(e1)}' > M.dat
```


D.3.1. Estimate GARR coordinates with DDL1 (using tropospheric corrections)

D.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

Solve the equations system using octave (or MATLAB) and assess the estimation error:

```
octave

load M.dat

y=M(:,1);
G=M(:,2:4);

x=inv(G'*G)*G'*y
x
   0.00224133050672853
   0.00948643658103340
   0.04065938792819074
```

Absolute coordinates of GARR.

Taking into account that the "a priori" coordinates of IND2 are:

```
GARR=[4796983.5170 160309.1774 4187340.3887]
```

Therefore the estimated absolute coordinates of GARR are:

```
GARR+ x(1:3)'
 ans= 4796983.5192 160309.1869 4187340.4294
```


D.3.2. Estimate GARR coordinates with DDL1 (using tropospheric corrections)

D.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

D.3.2. Using the DDL1 carrier with the ambiguities FIXED, compute the LS single epoch solution for the whole interval *39000< t <41300* with the program LS.f

Note: The program `LS.f` computes the Least Square solution for each measurement epoch of the input file (see the FORTRAN code "LS.f")

The following procedure can be applied

a) generate a file with the following content;

```
[Time],[DDL1-DDRho-DDTrp-lambda1*DDN1],[Los_k - Los_13]
```

where:

Time= seconds of day

DDL1 – DDRho –DDTrp – lambda1*DDN1= Prefit residulas (i.e., "y" values in program LS.f)

Los_k – Los_13 = The three components of the geometry matrix (i.e., matrix "a" in program LS.f)

D.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

```
[Time], [DDL1-DDRho-DDTrp-lambda1*DDN1], [Los_k - Los_13]
```

The following sentence can be used

```
cat DD_PLAN_GARR_13_ALL.fixL1 | gawk 'BEGIN{g2r=atan2(1,1)/45}
  {e1=$14*g2r;a1=$15*g2r;e2=$16*g2r;a2=$17*g2r;;printf "%s %14.4f
   %8.4f %8.4f %8.4f \n",$6,$8-$11-$12-$18,-cos(e2)*sin(a2)+cos(e1)*sin(a1),
       -cos(e2)*cos(a2)+cos(e1)*cos(a1),-sin(e2)+sin(e1)}' > L1model.dat
```

b) Compute the Least Squares solution

```
cat L1model.dat |LS > L1fix.pos
```

D.3. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

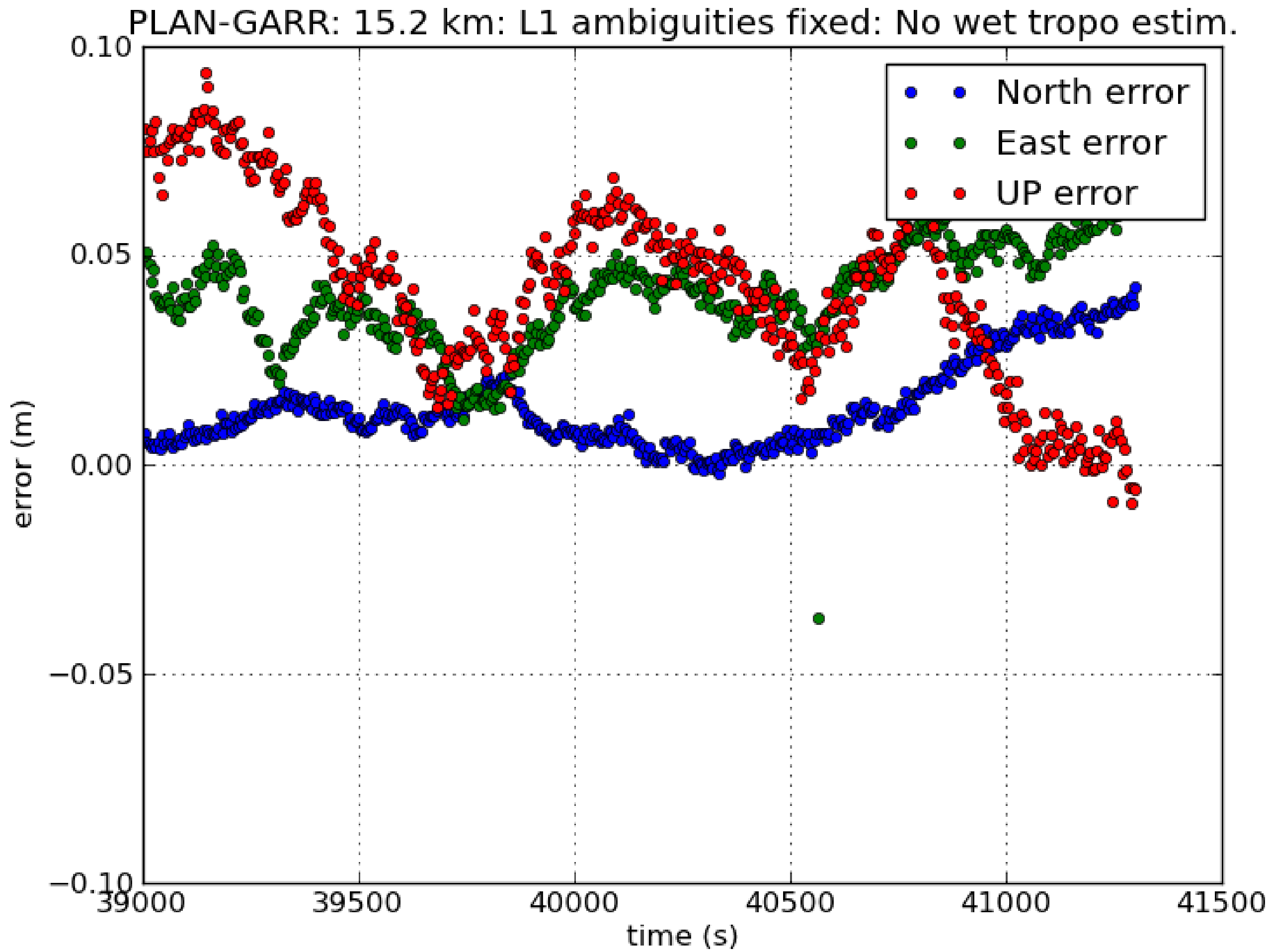
Plot the baseline estimation error

```
graph.py -f L1fix.pos -x1 –y2 -s.- -l "North error"
         -f L1fix.pos -x1 –y3 -s.- -l "East error"
         -f L1fix.pos -x1 –y4 -s.- -l "UP error"
  --yn -.1 --yx .1 --xl "time (s)" --yl "error (m)" -t "PLAN-GARR: 15.2
km: L1 ambiguities fixed: No wet tropo estim."
```

D.3. PLAN-GARR differential positioning with L1 carrier

(using the computed differential corrections including troposphere)

Differential Positioning error after fixing ambiguities



Question:
Discuss on the remaining error sources which could explain the error found in the North, East and Vertical components.

D.3.2. Estimate GARR coordinates with DDL1 (using tropospheric corrections)

D.4. PLAN-GARR differential positioning with L2 carrier

(using the computed differential corrections including troposphere)

Repeat the previous exercise, but positioning with the L2 carrier

D.4. PLAN-GARR differential positioning with L2 carrier (using the computed differential corrections including troposphere)

D.4.1 Using DDL2 carrier measurements, estimate the coordinates of receiver GARR taking PLAN as a reference receiver and correcting troposphere.

Consider only the two epochs used in the previous exercise: $t_1=39000$ and $t_2=40500$.

The following procedure can be applied:

1. **Compute the FLOATED solution**, solving the equations system with octave. Assess the accuracy of the floated solution.
2. **Apply the LAMBDA method to FIX the ambiguities**. Compare the results with the solution obtained by rounding the floated solution directly and by rounding the solution after decorrelation.
3. **Repair the DDL2 carrier measurements with the DDN2 FIXED** ambiguities and plot results to analyze the data.
4. **Compute the FIXED solution**.

D.4. PLAN-GARR differential positioning with L2 carrier
(using the computed differential corrections including troposphere)

The script **MakeL2DifTrpMat.scr** builds the equations system

$$[\mathrm{DDL2-DDRho-Trp}]=[\ \mathrm{Los\_k}-\ \mathrm{Los\_13}]*[\mathrm{dr}] + [\ A\ ]*[\lambda_2 *\mathrm{DDN2}]$$

for the two epochs required t_1=39000 and t_2=40500, using the input file **DD_PLAN_GARR_13_ALL.dat** generated before.

Execute:

```
MakeL2DifTrpMat.scr DD_PLAN_GARR_13_ALL.dat 39000 40500
```

The **OUTPUT**

are the files **M1.dat** and **M2.dat** associated with each epoch.

Where:

the columns of files **M.dat** are the vector **y** (first column) and Matrix **G** (next columns)

D.4. PLAN-GARR differential positioning with L2 carrier
(using the computed differential corrections including troposphere)

1. Computing the FLOATED solution (solving the equations system).

The following procedure can be applied

```
octave

load M1.dat
load M2.dat

y1=M1(:,1);
G1=M1(:,2:12);

y2=M2(:,1);
G2=M2(:,2:12);
Py=(diag(ones(1,8))+ones(8))*2e-4;
W=inv(Py);
```

```
P=inv(G1'*W*G1+G2'*W*G2);
x=P*(G1'*W*y1+G2'*W*y2);

x(1:3)'
-0.2949   0.0163   0.0567
```

Taking into account that the "a priori" coordinates of GARR are:

```
GARR=[4796983.5170 160309.1774
                         4187340.3887]
```

Therefore the estimated absolute coordinates of GARR are:

```
GARR+ x(1:3)'
4796983.2221 160309.1937 4187340.4454
```

D.4.1. Estimate GARR coordinates with DDL2 (using tropospheric corrections)

D.4. PLAN-GARR differential positioning with L2 carrier
(using the computed differential corrections including troposphere)

2. Applying the LAMBDA method to FIX the ambiguities.

The following procedure can be applied **(justify the computations done)**
Compare the different results found.

```
octave

c=299792458;
f0=10.23e+6;
f2=120*f0;
lambda2=c/f2
 a=x(4:10)/lambda2;
 Q=P(4:10,4:10);
```

Decorrelation and integer LS search solution

```
[Qz,Zt,Lz,Dz,az,iZ] = decorrel (Q,a);
[azfixed,sqnorm] = lsearch (az,Lz,Dz,2);
afixed=iZ*azfixed;
sqnorm(2)/sqnorm(1)
ans = 15.3627929427384
afixed(:,1)'
      -1075655   1343160   938718   468181
       -675593   -313616  -356299  1439836
```

Rounding the floated solution directly

```
round(a)'
 -1075654   1343161   938718   468182
  -675592   -313617  -356299  1439835
```

Rounding the decorrelated floated solution

```
afix=iZ*round(az)
      -1075655   1343160   938718   468181
       -675593   -313616  -356299  1439836
```

D.4.1. Estimate GARR coordinates with DDL2 (using tropospheric corrections)

D.4. PLAN-GARR differential positioning with L2 carrier

(using the computed differential corrections including troposphere)

3. Repair the DDL2 carrier measurements with the DDN2 FIXED ambiguities and plot results to analyze the data.

```
octave
amb=lambda2*afixed(:,1);
save ambL2.dat amb
```

Using the previous the file **ambL2.dat** and "DD_PLAN_GARR_13_ALL.fixL1", generate the a file with the following content:

```
-----------------------------  DD_PLAN_GARR_13_ALL.fixL1L2 ---------------------------------------
  1     2    3    4    5    6     7     8     9    10     11    12     13     14   15   16   17     18       19
PLAN GARR 13 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2 λ1*DDN1 λ2*DDN2
                                                                 <---- GARR ---->
---------------------------------------------------------------------------------------------------
```

Note: This file is identical to file "DD_PLAN_GARR_13_ALL.fixL1", but with the ambiguities added in the last field #19.

D.4.1. Estimate GARR coordinates with DDL2 (using tropospheric corrections)

D.4. PLAN-GARR differential positioning with L2 carrier
(using the computed differential corrections including troposphere)

a) Generate a file with the satellite PRN number and the ambiguities:

```
grep -v \# ambL2.dat > na2
cat DD_PLAN_GARR_13_ALL.dat|gawk '{print $4}'|sort -nu|gawk '{print $1,NR}' >sat.lst
paste sat.lst na2 > sat.ambL2
```

b) Generate the "DD_PLAN_GARR_13_ALL.fixL1L2" file:

```
cat DD_PLAN_GARR_13_ALL.fixL1|
                  gawk 'BEGIN{for (i=1;i<1000;i++) {getline <"sat.ambL2";A[$1]=$3}}
  {printf "%s %02i %02i %s %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f
                                %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f %14.4f \n",
              $1,$2,$3,$4,$5,$6,$7,$8,$9,$10,$11,$12,$13,$14,$15,$16,$17,$18,A[$4]}' >
                                                              DD_PLAN_GARR_13_ALL.fixL1L2
```

D.4. PLAN-GARR differential positioning with L2 carrier
(using the computed differential corrections including troposphere)

```
---------------------------- DD_PLAN_GARR_13_ALL.fixL1L2 -------------------------------------
  1     2   3   4   5   6     7    8    9   10    11    12     13    14  15   16  17     18      19
PLAN GARR 13 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2 λ1*DDN1 λ2*DDN2
                                                                <---- GARR ---->
-----------------------------------------------------------------------------------------------
```

c) Make and discuss the following plots

```
graph.py -f DD_PLAN_GARR_13_ALL.fixL1L2 -x6 -y'($10-$19-$11)'
       -so --yn -0.6 --yx 0.6 -l "(DDL2-lambda2*DDN2)-DDRho" --xl "time (s)" --yl "m"
```

```
graph.py -f DD_PLAN_GARR_13_ALL.fixL1L2 -x6 -y'($10-$19-$11-$12)'
 -so --yn -0.6 --yx 0.6 -l "(DDL2-lambda2*DDN2)-DDRho-DDTrp" --xl "time (s)" --yl "m"
```

```
graph.py -f DD_PLAN_GARR_13_ALL.fixL1L2 -x6 -y'($10-$19-$12)'
     -so --yn -0.06 --yx 0.16 -l "(DDL2-lambda2*DDN2)-DDTrp" --xl "time (s)" --yl "m"
```

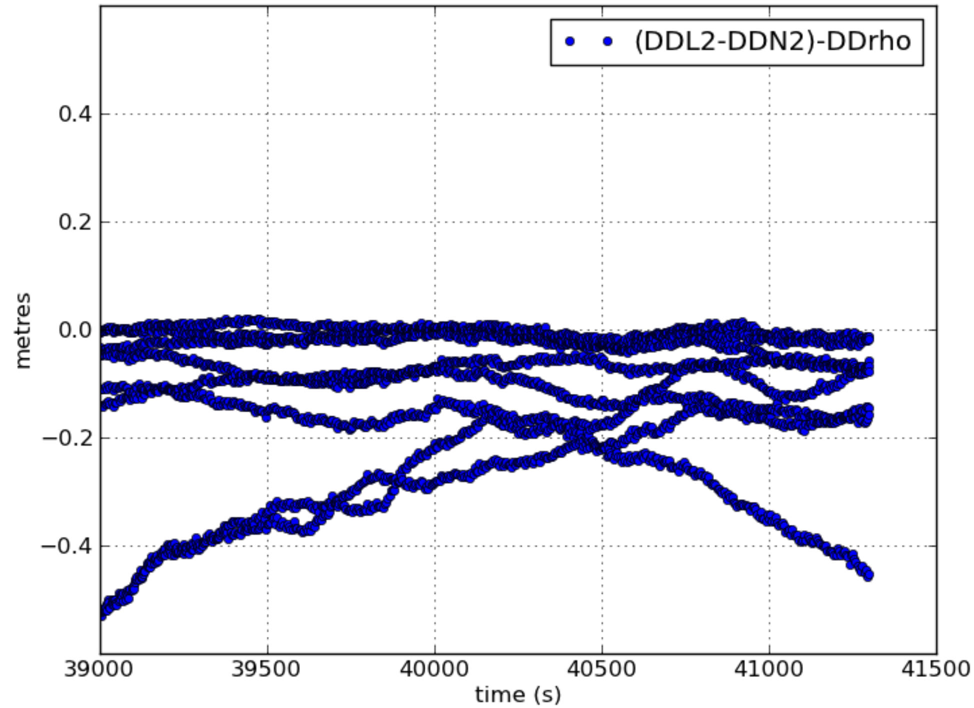
```
graph.py -f DD_PLAN_GARR_13_ALL.fixL1L2 -x6 -y'($10-$19)'
         -so --yn -15000 --yx 15000 -l "(DDL2-lambda2*DDN2)" --xl "time (s)" --yl "m"
```


D.4.1. Estimate GARR coordinates with DDL1 (using tropospheric corrections)

D.4. PLAN-GARR differential positioning with L2 carrier
(using the computed differential corrections including troposphere)

```
--------------------------- DD_PLAN_GARR_13_ALL.fixL1L2 -------------------------------------
 1     2    3    4    5    6     7     8    9    10     11     12     13     14   15   16   17     18       19
PLAN GARR 13 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2 λ1*DDN1 λ2*DDN2
                                                                 <---- GARR ---->
---------------------------------------------------------------------------------------------
```

```
graph.py -f DD_PLAN_GARR_13_ALL.fixL1L2
         -x6 -y'($10-$19-$11)'
         -so --yn -0.6 --yx 0.6
         -l "(DDL2-λ2*DDN2)-DDRho"
         --xl "time (s)" --yl "m"
```



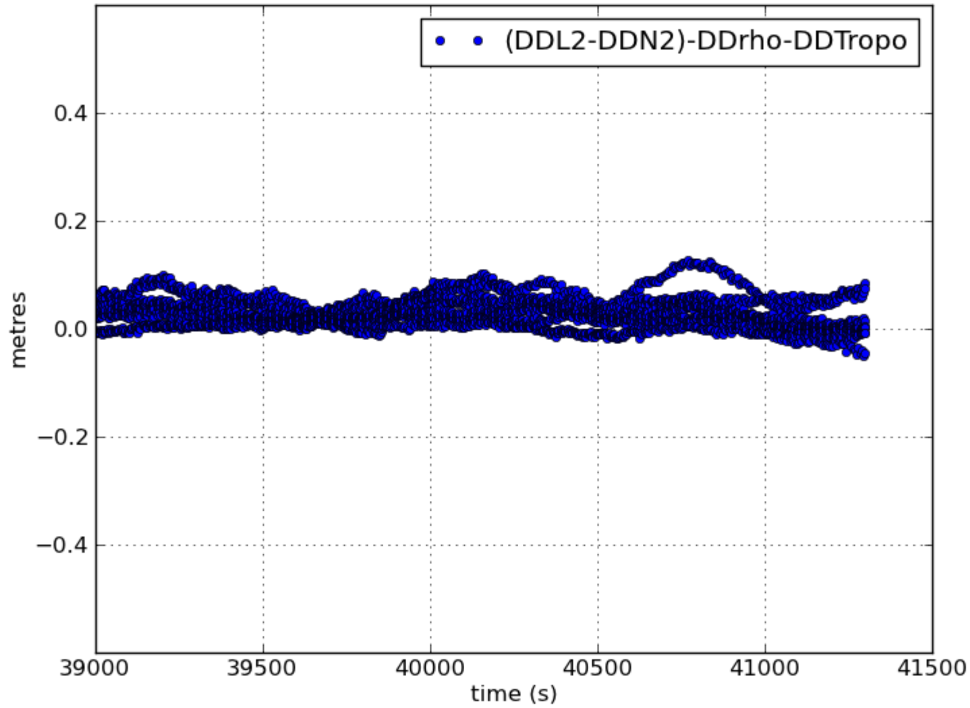
Questions:
Explain what is the meaning of this plot.

D.4.1. Estimate GARR coordinates with DDL2 (using tropospheric corrections)

D.4. PLAN-GARR differential positioning with L2 carrier
(using the computed differential corrections including troposphere)

```
---------------------------- DD_PLAN_GARR_13_ALL.fixL1L2 ------------------------------------
  1     2    3    4    5    6     7     8    9    10     11     12     13     14   15   16   17     18      19
PLAN GARR 13 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2 λ1*DDN1 λ2*DDN2
                                                                 <---- GARR ---->
---------------------------------------------------------------------------------------------
```

```
graph.py -f DD_PLAN_GARR_13_ALL.fixL1L2
         -x6 -y'($10-$19-$11-$12)'
         -so --yn -0.6 --yx 0.6
         -l "(DDL2-λ2*DDN2)-DDRho-DDTrp"
         --xl "time (s)" --yl "m"
```



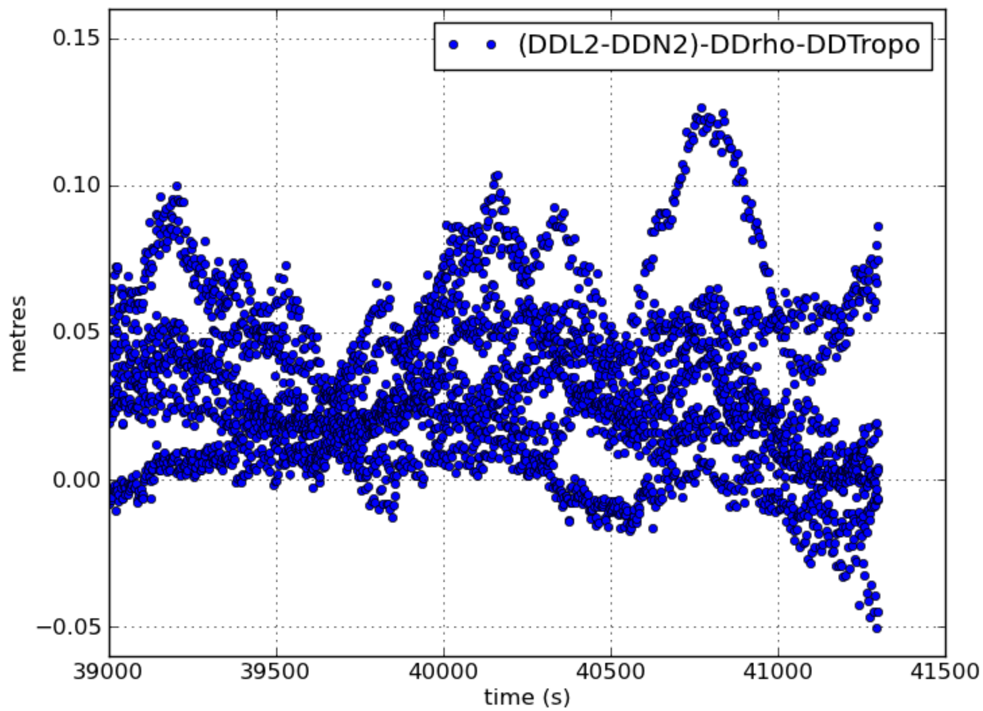
Questions:
Explain what is the meaning of this plot.

D.4.1. Estimate GARR coordinates with DDL2 (using tropospheric corrections)

D.4. PLAN-GARR differential positioning with L2 carrier (using the computed differential corrections including troposphere)

```
---------------------------- DD_PLAN_GARR_13_ALL.fixL1L2 -----------------------------------
  1    2    3    4    5    6    7    8    9    10    11    12    13     14   15   16   17    18       19
PLAN GARR 13 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2 λ1*DDN1 λ2*DDN2
                                                                    <---- GARR ---->
---------------------------------------------------------------------------------------------
```

```
graph.py -f DD_PLAN_GARR_13_ALL.fixL1L2
         -x6 -y'($10-$19-$11-$12)'
         -so --yn -0.06 --yx 0.16
         -l "(DDL2-λ2*DDN2)-DDRho-DDTrp"
         --xl "time (s)" --yl "m"
```



Questions:
Explain what is the meaning of this plot.

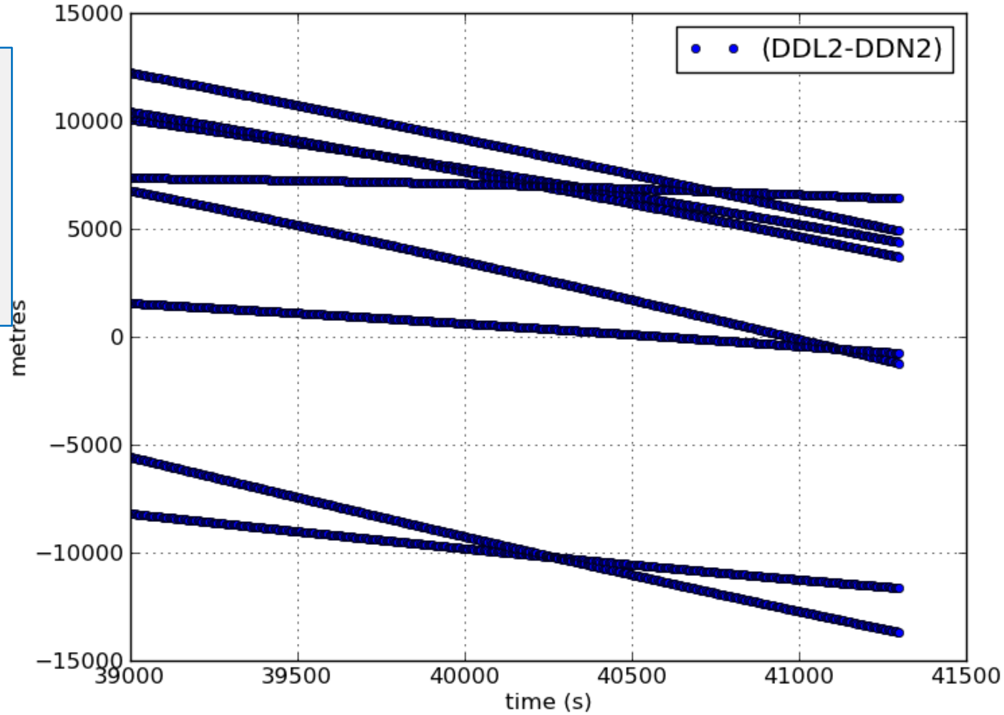
D.4.1. Estimate GARR coordinates with DDL2 (using tropospheric corrections)

D.4. PLAN-GARR differential positioning with L1 carrier
(using the computed differential corrections including troposphere)

```
---------------------------- DD_PLAN_GARR_13_ALL.fixL1L2 -------------------------------------
  1    2   3   4   5   6    7    8    9   10    11    12     13    14  15  16  17     18     19
PLAN GARR 13 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2 λ1*DDN1 λ2*DDN2
                                                                <---- GARR ---->
----------------------------------------------------------------------------------------------
```

```
graph.py -f DD_PLAN_GARR_13_ALL.fixL1L2
         -x6 -y'($10-$19)'
         -so --yn -15000 --yx 15000
         -l "(DDL2-λ2*DDN2)"
         --xl "time (s)" --yl "m"
```

Questions:
Explain what is the meaning of this plot.



D.4.1. Estimate GARR coordinates with DDL1 (using tropospheric corrections)

D.4. PLAN-GARR differential positioning with L2 carrier
(using the computed differential corrections including troposphere)

4. Computing the FIXED solution (after FIXING ambiguities).

The following procedure can be applied

a) Build the equations system

```
[DDL2-DDRho-DDTrp-lambda2*DDN2]=[Los_k - Los_06]*[dr]
```

```
cat DD_PLAN_GARR_13_ALL.fixL1L2 | gawk 'BEGIN{g2r=atan2(1,1)/45}
                    {e1=$14*g2r;a1=$15*g2r;e2=$16*g2r;a2=$17*g2r;
 printf "%14.4f %8.4f %8.4f %8.4f \n",
  $10-$11-$12-$19, -cos(e2)*sin(a2)+cos(e1)*sin(a1),
          -cos(e2)*cos(a2)+cos(e1)*cos(a1), -sin(e2)+sin(e1)}' > M.dat
```


D.4.1. Estimate GARR coordinates with DDL2 (using tropospheric corrections)

D.4. PLAN-GARR differential positioning with L2 carrier

(using the computed differential corrections including troposphere)

Solve the equations system using octave (or MATLAB) and assess the estimation error:

```
octave

load M.dat

y=M(:,1);
G=M(:,2:4);

x=inv(G'*G)*G'*y
x
    0.00312473328403573
    0.01680339170230549
    0.06303852755939099
```

Absolute coordinates of GARR.

Taking into account that the ”a priori” coordinates of IND2 are:

```
GARR=[4796983.5170 160309.1774 4187340.3887]
```

Therefore the estimated absolute coordinates of GARR are:

```
GARR+ x(1:3)'
 ans= 4796983.5201  160309.1942  4187340.4517
```

Question:

Is the accuracy similar to that in the previous case, when estimating the baseline vector?

D.4. PLAN-GARR differential positioning with L2 carrier

(using the computed differential corrections including troposphere)

D.4.2. Using the DDL2 carrier with the ambiguities FIXED, compute the LS single epoch solution for the whole interval 39000< t <41300 with the program LS.f

Note: The program `LS.f` computes the Least Square solution for each measurement epoch of the input file (see the FORTRAN code "LS.f")

The following procedure can be applied

a) generate a file with the following content;

```
[Time],[DDL2-DDRho-DDTrp-lambda2*DDN2],[Los_k - Los_13]
```

where:

Time= seconds of day

DDL2 – DDRho –DDTrp – lambda2*DDN2= Prefit residulas (i.e., "y" values in program LS.f)

Los_k – Los_13 = The three components of the geometry matrix (i.e., matrix "a" in program LS.f)

D.4. PLAN-GARR differential positioning with L2 carrier

(using the computed differential corrections including troposphere)

```
[Time], [DDL2-DDRho-DDTrp-lambda2*DDN2], [Los_k - Los_06]
```

The following sentence can be used

```
cat DD_PLAN_GARR_13_ALL.fixL1 | gawk 'BEGIN{g2r=atan2(1,1)/45}
  {e1=$14*g2r;a1=$15*g2r;e2=$16*g2r;a2=$17*g2r;;printf "%s %14.4f
   %8.4f %8.4f %8.4f \n",$6,$10-$11-$12-$18,-cos(e2)*sin(a2)+cos(e1)*sin(a1),
        -cos(e2)*cos(a2)+cos(e1)*cos(a1),-sin(e2)+sin(e1)}' > L2model.dat
```

b) Compute the Least Squares solution

```
cat L2model.dat |LS > L2fix.pos
```

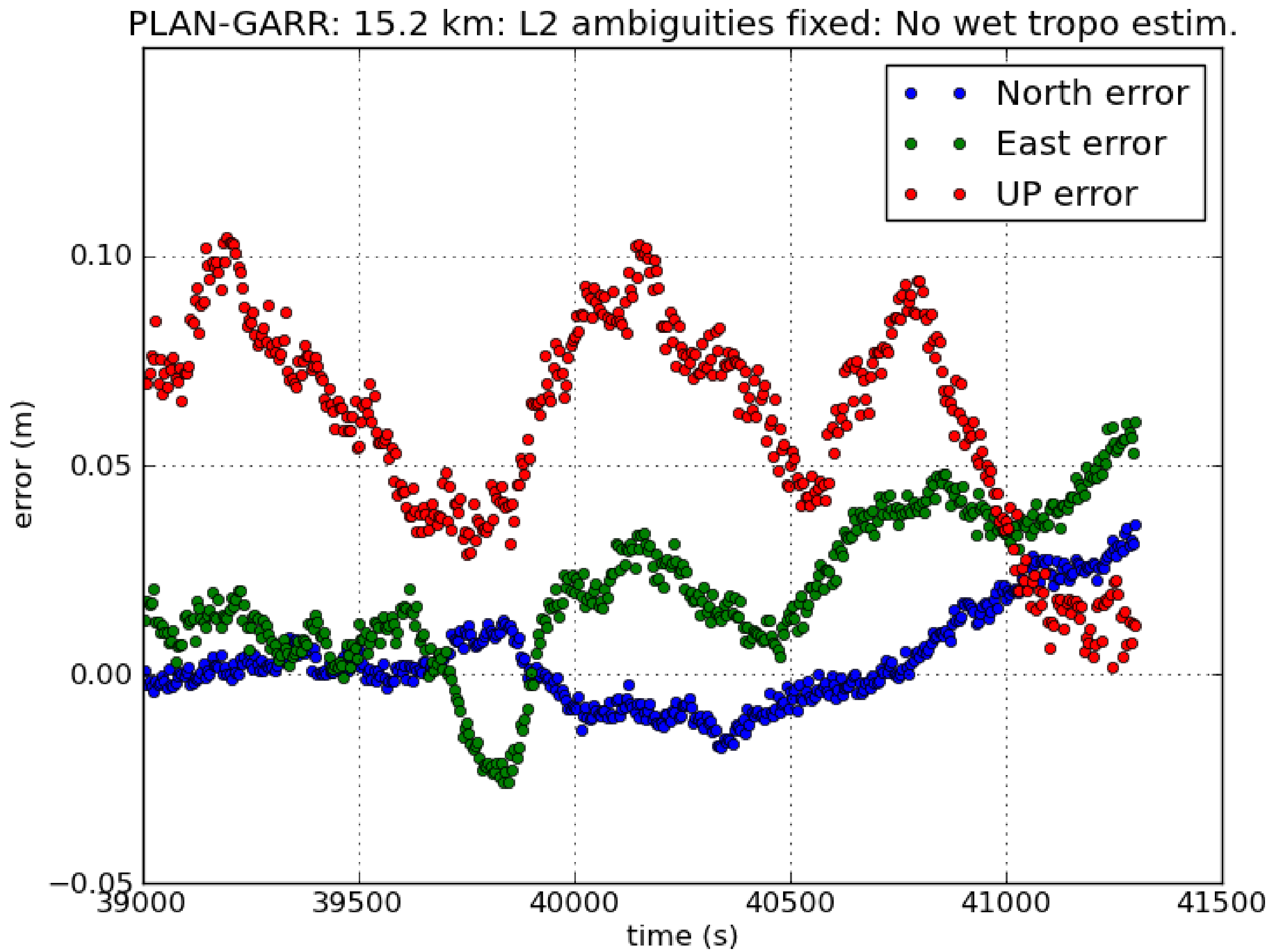
D.4. PLAN-GARR differential positioning with L2 carrier
(using the computed differential corrections including troposphere)

Plot the baseline estimation error

```
graph.py -f L2fix.pos -x1 -y2 -s.- -l "North error"
         -f L2fix.pos -x1 -y3 -s.- -l "East error"
         -f L2fix.pos -x1 -y4 -s.- -l "UP error"
  --yn -.05 --yx .15 --xl "time (s)" --yl "error (m)" -t "PLAN-GARR: 15.2
km: L2 ambiguities fixed: No wet tropo estim."
```

D.4. PLAN-GARR differential positioning with L2 carrier
(using the computed differential corrections including troposphere)

Differential Positioning error after fixing ambiguities



Question:
Discuss the possible sources of the bias found in the vertical component.

D.4.2. Estimate GARR coordinates with DDL2 (using tropospheric corrections)

D.5.1 PLAN-GARR differential positioning with L1 carrier (with ambiguity fixed and correcting from tropo. and **Klobuchar iono**.)

```
----------------------------- DD_PLAN_GARR_13_ALL.fixL1L2 ------------------------------------
  1     2    3    4    5    6     7     8    9    10     11     12     13     14   15   16   17      18      19
PLAN GARR 13 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2 λ1*DDN1 λ2*DDN2
                                                                 <---- GARR ---->
---------------------------------------------------------------------------------------------
```

Plot the unambiguous DDSTEC as a function of time and elevation, using Klobuchar model (it corresponds to the field #13 of file **DD_PLAN_GARR_13_ALL.fixL1L2**).

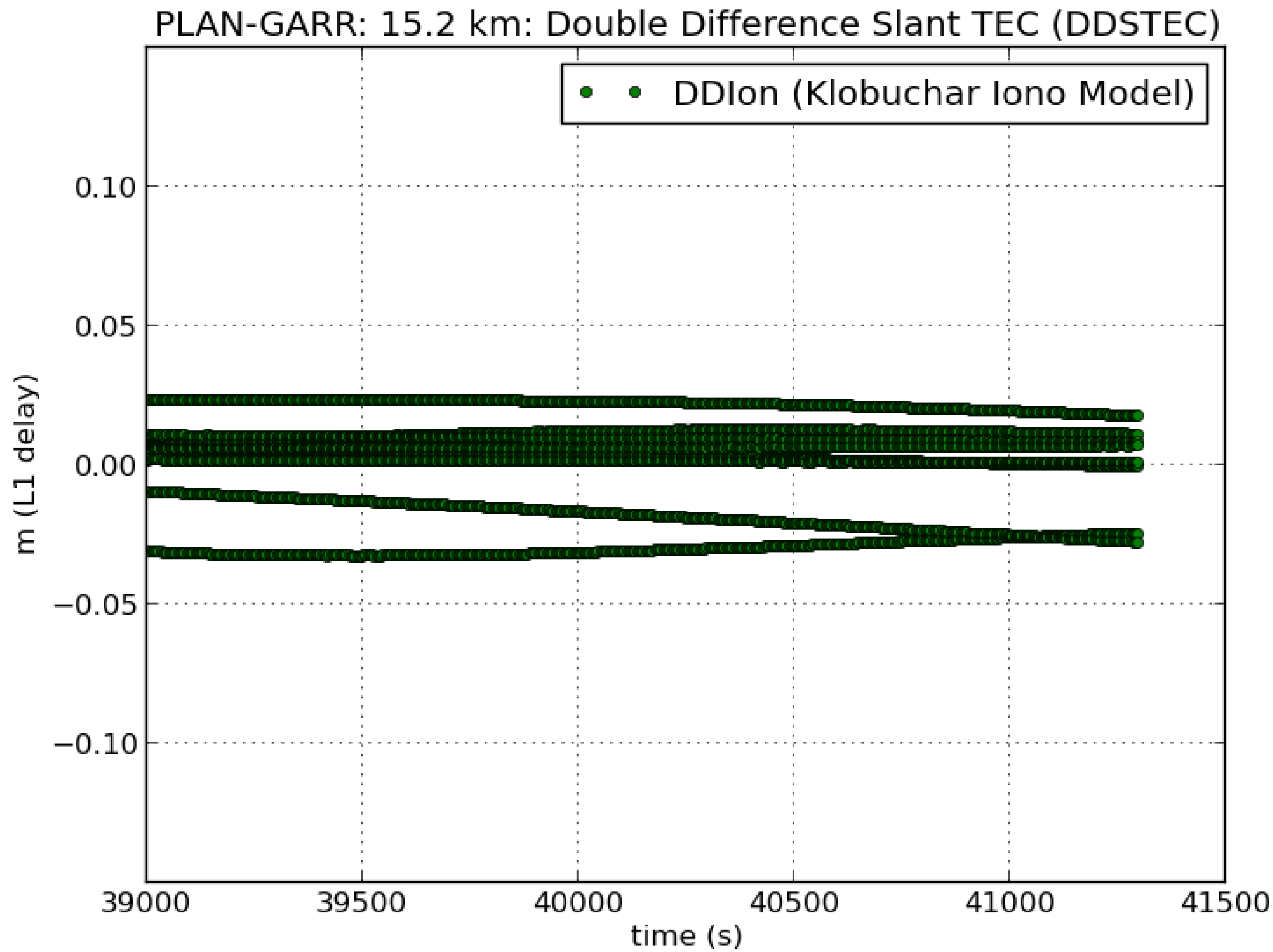
Execute:

```
graph.py -f DD_PLAN_GARR_13_ALL.fixL1L2 -x6 -y13
          -so -l "DDIon (Klobuchar Iono Model)" --xl "time (s)"
      --yl "(m L1 delay)" --yn -.1 --yx .1  -t "PLAN-GARR: 15.2 km: DDSTEC"
```

```
graph.py -f DD_PLAN_GARR_13_ALL.fixL1L2 -x16 -y13
          -so -l "DDIon (Klobuchar Iono Model)" --xl "elevation (deg.)"
      --yl "(m L1 delay)" --yn -.1 --yx .1  -t "PLAN-GARR: 15.2 km: DDSTEC"
```

D.5.1 PLAN-GARR differential positioning with L1 carrier
(with ambiguity fixed and correcting from tropo. and **Klobuchar iono**.)

Modelled DDIono (Klobuchar) as a function of time

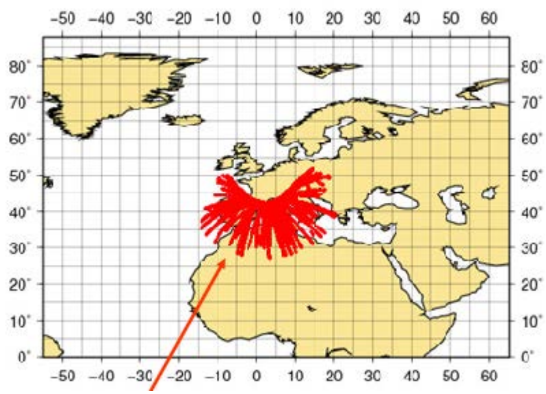


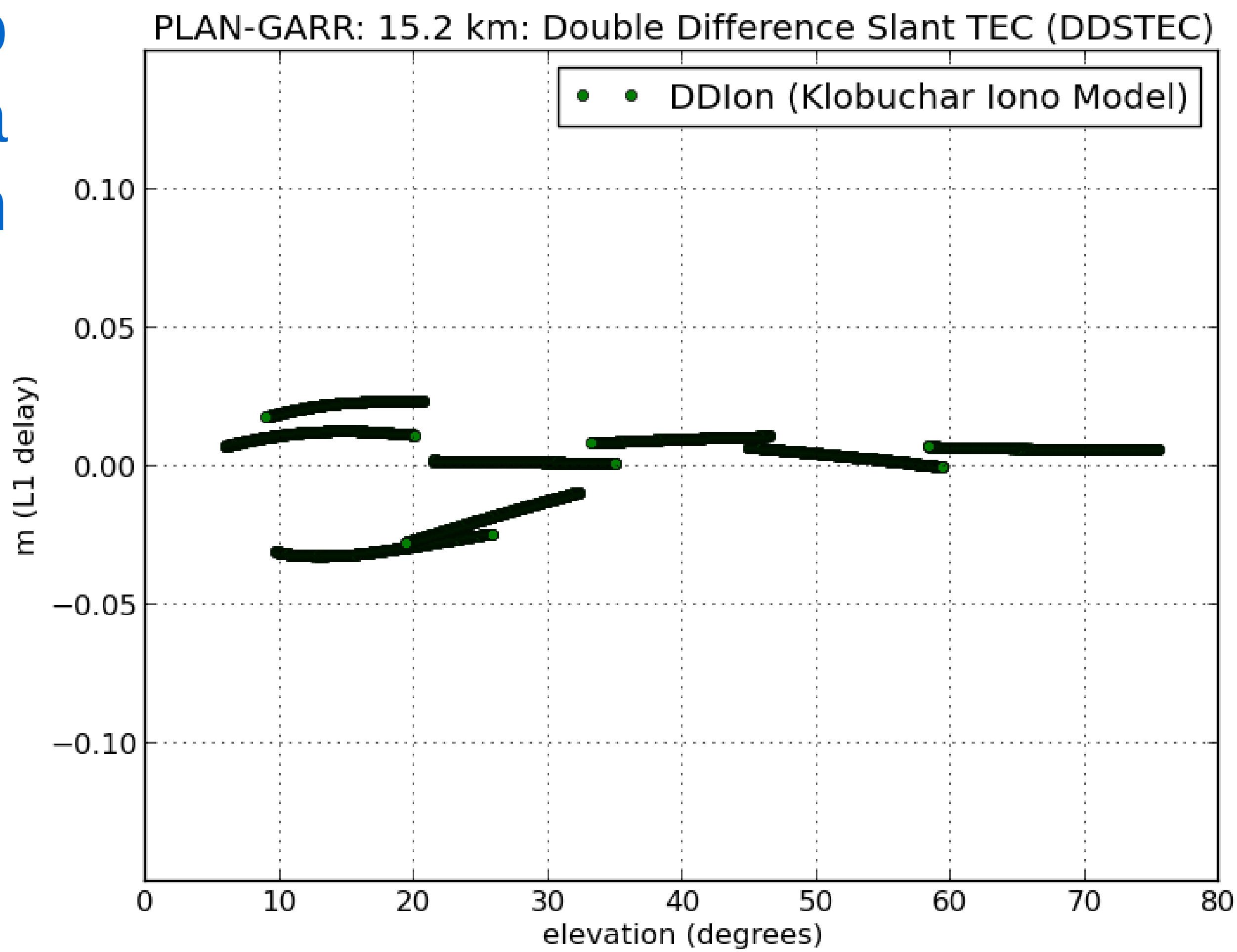
Question:
Discuss this plot

D.5.1 PLAN-GARR differential positioning with L1 carrier
(with ambiguity fixed and correcting from tropo. and **Klobuchar iono**.)

Modelled DDIono (Klobuchar) as a function of elevation

Question:
Why a larger dispersion is found at a low elevation?





D.5.1. Estimate GARR coordinates with DDL1 (using tropo & Iono Klobuchar)

D.5.1 PLAN-GARR differential positioning with L1 carrier (with ambiguity fixed and correcting from tropo. and **Klobuchar iono**.)

```
---------------------------- DD_PLAN_GARR_13_ALL.fixL1L2 ----------------------
  1    2   3   4   5   6    7    8   9   10    11    12     13    14  15  16  17    18     19
PLAN GARR 13 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2 λ1*DDN1 λ2*DDN
                                                              <---- GARR ---->
-------------------------------------------------------------------------------
```

Plot the prefit-residuals:

Prefit= DDL1-DDRho-Lambda1*DDN1-DDTropo+DDIon:

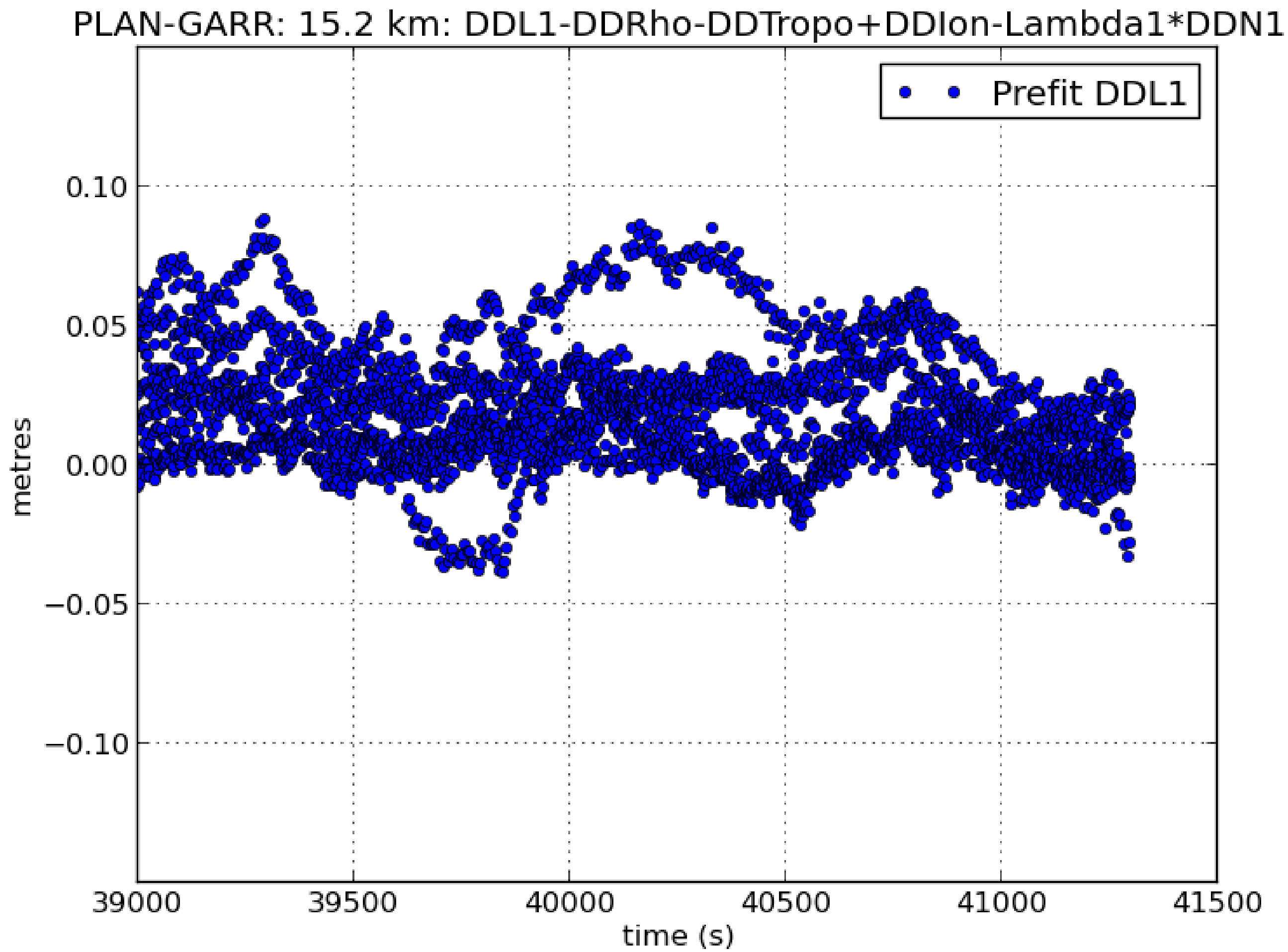
1.- As a function of time

```
graph.py -f DD_PLAN_GARR_13_ALL.fixL1L2
        -x6 -y'$8-$11-$18-$12+$13'
        -so -l "Prefit DDL1" --xl "time (s)"
        --yl "metres" --yn -.1 --yx .1
        -t "PLAN-GARR: 15.2 km: DDL1-DDRho-DDTropo+DDIon-Lambda1*DDN1"
```

D.5.1 PLAN-GARR differential positioning with L1 carrier

(with ambiguity fixed and correcting from tropo. and **Klobuchar iono**.)

DDL1 Pre-fit residuals as a function of time.



Question:
Discuss the noise seen in the plot.

D.5.1 PLAN-GARR differential positioning with L1 carrier

(with ambiguity fixed and correcting from tropo. and **Klobuchar iono**.)

D.5.1. Using the DDL1 carrier with the ambiguities FIXED, and correcting from both troposphere and Klobuchar ionosphere (DDSTEC), compute the LS single epoch solution for the whole interval 39000< t <41300 with the program LS.f.

Note: this correction (DD) is given in file DD_PLAN_GARR_13_ALL.fixL1L2 on the field “`DDIon`” (i.e. #13) in meters of L1 delay.

The following procedure can be applied

a) generate a file with the following content;

```
[Time], [DDL1-DDRho-DDTrp+DDIon-lambda1*DDN1], [Los_k-Los_13]
```

where:

Time= seconds of day

DDL1 – DDRho –DDTrp + DDIon– lambda1*DDN1= Prefit residulas (i.e., "y" values in program LS.f)

Los_k – Los_13= The three components of the geometry matrix,i.e. matrix "a" in LS.f program.

D.5.1 PLAN-GARR differential positioning with L1 carrier

(with ambiguity fixed and correcting from tropo. and **Klobuchar iono**.)

```
[Time], [DDL1-DDRho-DDTrp+DDIon-lambda1*DDN1], [Los_k - Los_13]
```

The following sentence can be used

```
cat DD_PLAN_GARR_13_ALL.fixL1L2 | gawk 'BEGIN{g2r=atan2(1,1)/45}
  {e1=$14*g2r;a1=$15*g2r;e2=$16*g2r;a2=$17*g2r;STEC1=$13;printf "%s %14.4f %8.4f
                      %8.4f %8.4f \n",$6,$8-$11-$18-$12+STEC1,
         -cos(e2)*sin(a2)+cos(e1)*sin(a1),-cos(e2)*cos(a2)+cos(e1)*cos(a1),
                                             -sin(e2)+sin(e1)}' > L1model_Klob.dat
```

b) Compute the Least Squares solution

```
cat L1model_Klob.dat |LS > L1fixKlob.pos
```

D.5.1 PLAN-GARR differential positioning with L1 carrier

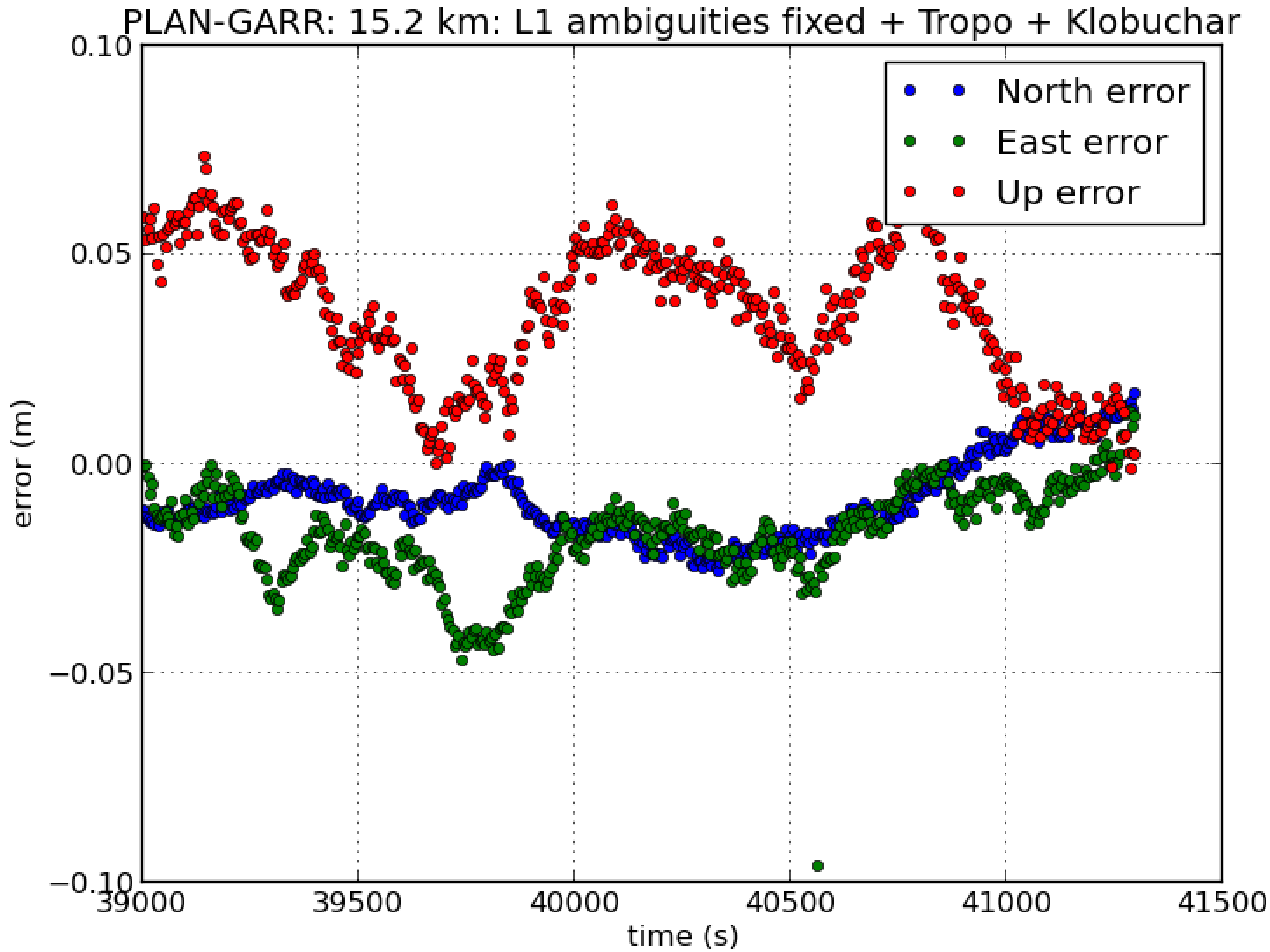
(with ambiguity fixed and correcting from tropo. and **Klobuchar iono**.)

Plot the positioning error

```
graph.py -f L1fixKlob.pos -x1 -y2 -s.- -l "North error"
         -f L1fixKlob.pos -x1 -y3 -s.- -l "East error"
         -f L1fixKlob.pos -x1 -y4 -s.- -l "UP error"
  --yn -.1 --yx .1 --xl "time (s)" --yl "error (m)" -t "PLAN-GARR: 15.2
km: L1 ambiguities fixed + Tropo + Klobuchar"
```

D.5.1 PLAN-GARR differential positioning with L1 carrier
(with ambiguity fixed and correcting from tropo. and **Klobuchar iono.**)

Differential Positioning error with L1, ambiguities fixed and troposphere with ionosphere (from Klobuchar) corrections



Question:
Discuss the results.

D.5.1. Estimate GARR coordinates with DDL1 (using tropo & Iono Klobuchar)

D.6. Unambiguous DDSTEC determination

```
----------------------------  DD_PLAN_GARR_13_ALL.fixL1L2 ------------------------------------
  1     2    3    4    5    6     7     8    9    10     11    12     13     14   15   16   17     18      19
PLAN GARR 13 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2 λ1*DDN1 λ2*DDN2
                                                                    <---- GARR ---->
---------------------------------------------------------------------------------------------
```

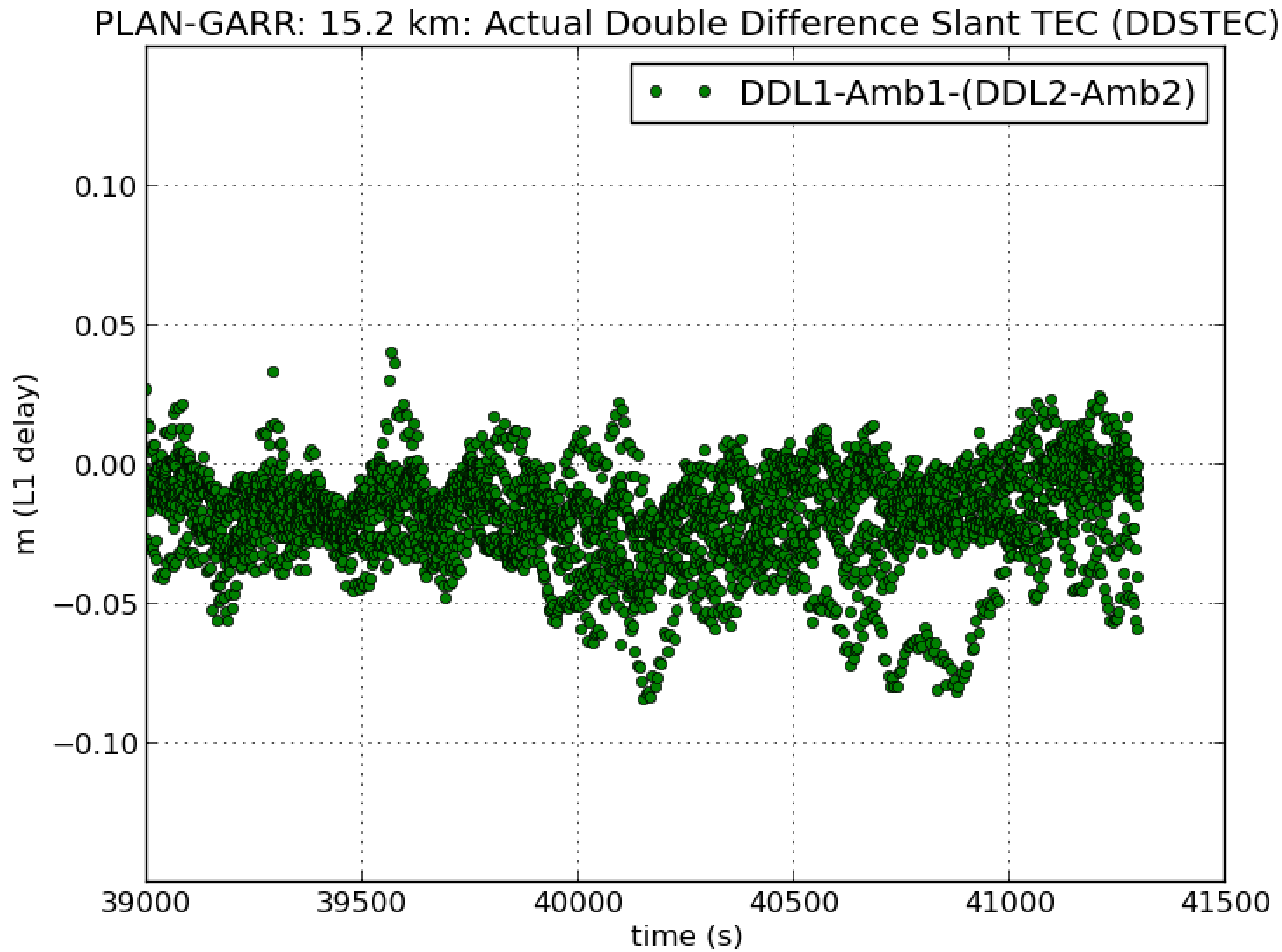
Using DDL1, DDL2 and the fixed ambiguities DDN1 and DDN2 obtained before, compute and plot the unambiguous DDSTEC as a function of time and elevation. Execute:

```
graph.py -f DD_PLAN_GARR_13_ALL.fixL1L2 -x6 -y'($8-$18-$10+$19)*1.546'
         -so -l "DDL1-Amb1-(DDL2-Amb2)" --xl "time (s)"
     --yl "(m L1 delay)" --yn -.1 --yx .1  -t "PLAN-GARR: 15.2 km: DDSTEC"
```

```
graph.py -f DD_PLAN_GARR_13_ALL.fixL1L2 –x16 -y'($8-$18-$10+$19)*1.546'
         -so -l "DDL1-Amb1-(DDL2-Amb2)" --xl "elevation (deg.)"
     --yl "(m L1 delay)" --yn -.1 --yx .1  -t "PLAN-GARR: 15.2 km: DDSTEC"
```

D.6. Unambiguous DDSTEC determination

Unambiguous DDSTEC as a function of time

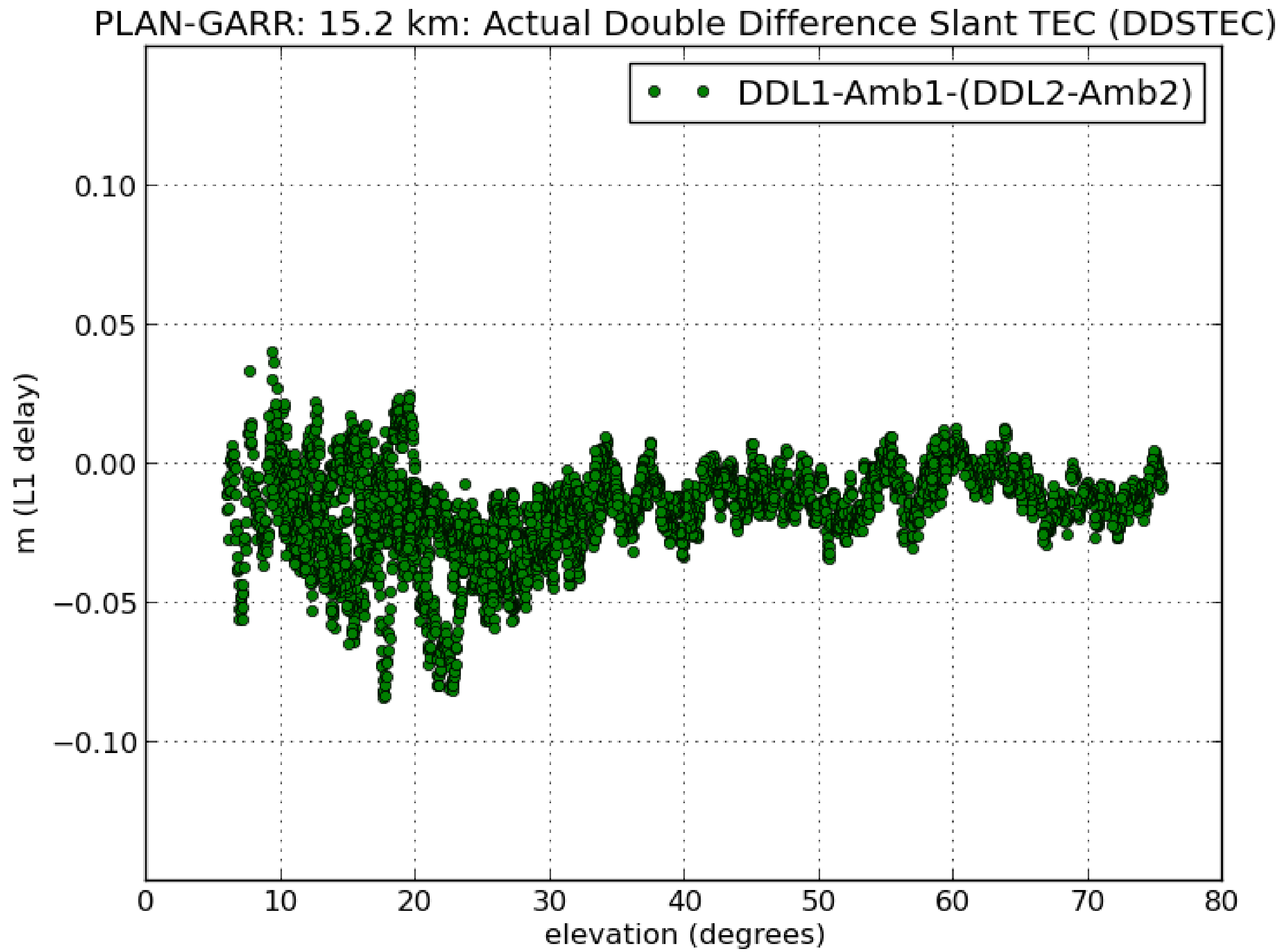


Question:
Discuss the noise seen in the plot.

D.6. Unambiguous DDSTEC determination

Unambiguous
DDSTEC
as a function
elevation

Question:
Why do we have an elevation-dependent pattern?



D.6. Unambiguous DDSTEC determination

```
------------------------------  DD_PLAN_GARR_13_ALL.fixL1L2 ----------------------------
  1     2    3    4    5    6     7     8    9    10     11     12      13     14   15   16   17     18      19
PLAN GARR 13 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2 λ1*DDN1 λ2*DDN
                                                                   <---- GARR ---->
-----------------------------------------------------------------------------------------
```

Plot the prefit-residuals:

Prefit= DDL1-DDRho-Lambda1*DDN1-DDTropo+alpha1*STEC:

1.- As a function of time

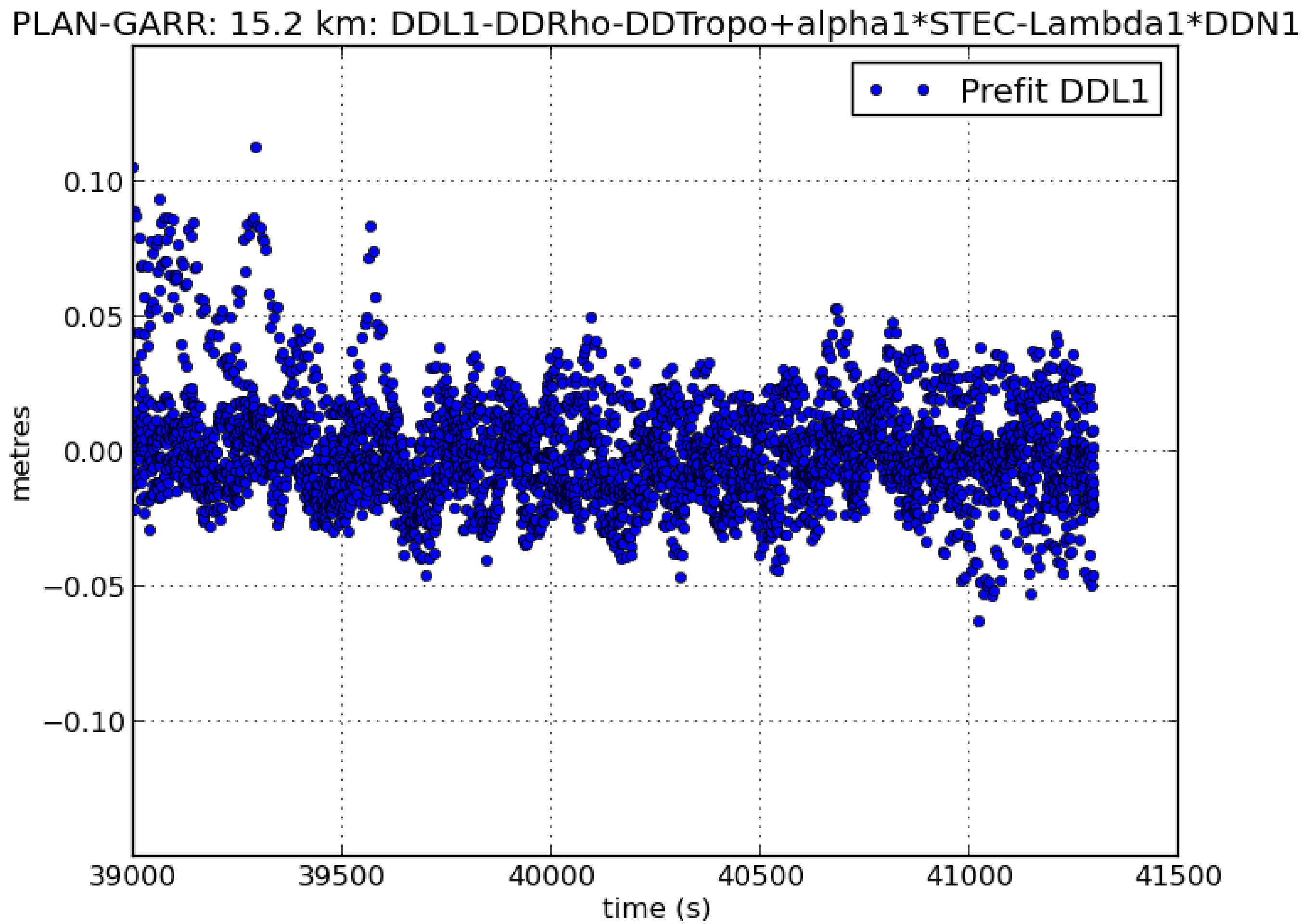
$$\bar{\alpha}_1 = \frac{f_2^2}{f_1^2 - f_2^2} = \frac{1}{\gamma - 1} = 1.546$$

$$\gamma = \left(\frac{77}{60}\right)^2$$

```
graph.py -f DD_PLAN_GARR_13_ALL.fixL1L2
        -x6 -y'$8-$11-$18-$12+1.546*($8-$18-$10+$19)'
        -so -l "Prefit DDL1" --xl "time (s)"
        --yl “metres" --yn -.15 --yx .15
        -t "PLAN-GARR: 15.2 km: DDL1-DDRho-Lambda1*DDN1-DDTropo+alpha1*STEC"
```

D.6. Unambiguous DDSTEC determination

DDL1 Pre-fit residuals as a function of time.



Question:
Discuss the noise seen in the plot.

D.6. Unambiguous DDSTEC determination

```
------------------------------  DD_PLAN_GARR_13_ALL.fixL1L2 ------------------------------------
  1    2   3   4   5   6    7    8   9   10    11    12    13    14  15  16  17    18     19
PLAN GARR 13 PRN DoY sec DDP1 DDL1 DDP2 DDL2 DDRho DDTrop DDIon El1 Az1 El2 Az2 λ1*DDN1 λ2*DDN2
                                                              <---- GARR ---->
-----------------------------------------------------------------------------------------------
```

Plot the prefit-residuals:

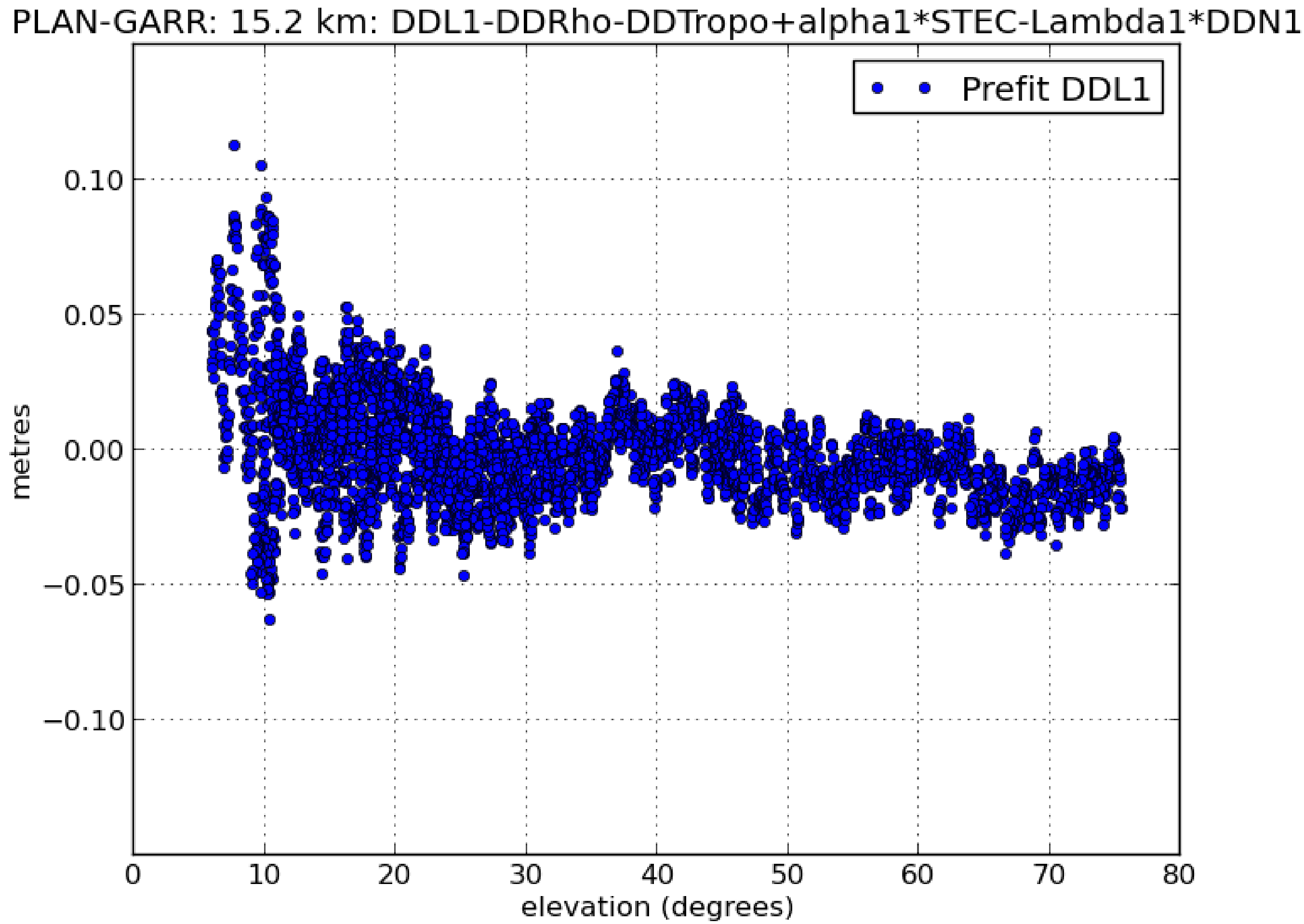
Prefit= DDL1-DDRho-Lambda1*DDN1-DDTropo+alpha1*STEC:

2.- As a function of elevation

```
graph.py -f DD_PLAN_GARR_13_ALL.fixL1L2
         -x16 -y'$8-$11-$18-$12+1.546*($8-$18-$10+$19)'
         -so -l "Prefit DDL1" --xl "elevation (deg.)"
         --yl "metres" --yn -.15 --yx .15
         -t "PLAN-GARR: 15.2 km: DL1-DDRho-Lambda1*DDN1-DDTropo+alpha1*STEC"
```

D.6. Unambiguous DDSTEC determination

DDL1 Pre-fit residuals as a function of elevation.



Question:
Why do we have an elevation-dependent pattern?

D.6.1 PLAN-GARR differential positioning with L1 carrier

(with ambiguity fixed and correcting from tropo. and **actual Iono.**)

D.6.1. Using the DDL1 carrier with the ambiguities FIXED, and correcting from both troposphere and ionosphere (DDSTEC), compute the LS single epoch solution for the whole interval 39000< t <41300 with the program LS.f

The following procedure can be applied

a) generate a file with the following content;

```
[Time], [DDL1-DDRho-DDTrp+α1*DDSTEC-lambda1*DDN1], [Los_k - Los_13]
```

where:

Time= seconds of day

DDL1 – DDRho –DDTrp + α_1*DDSTEC– lambda1*DDN1= Prefit residulas (i.e., "y" values in program LS.f)

Los_k – Los_13 = The three components of the geometry matrix (i.e., matrix "a" in program LS.f)

D.6.1. Estimate GARR coordinates with DDL1 (using tropo & actual Iono.)

D.6. PLAN-GARR differential positioning with L1 carrier

(with ambiguity fixed and correcting from tropo. and DDSTEC)

[Time], [DDL1-DDRho-DDTrp+α_1*DDSTEC-lambda1*DDN1], [Los_k - Los_13]

The following sentence can be used:

```
cat DD_PLAN_GARR_13_ALL.fixL1L2 | gawk 'BEGIN{g2r=atan2(1,1)/45}
 {e1=$14*g2r;a1=$15*g2r;e2=$16*g2r;a2=$17*g2r;STEC1=1.546*($8-$18-$10+$19);printf
              "%s %14.4f %8.4f %8.4f %8.4f \n",$6,$8-$11-$18-$12+STEC1,
        -cos(e2)*sin(a2)+cos(e1)*sin(a1),-cos(e2)*cos(a2)+cos(e1)*cos(a1),
                                           -sin(e2)+sin(e1)}' > L1model_stec.dat
```

b) Compute the Least Squares solution

```
cat L1model_stec.dat |LS > L1fixStec.pos
```

D.6. PLAN-GARR differential positioning with L1 carrier
(with ambiguity fixed and correcting from tropo. and DDSTEC)

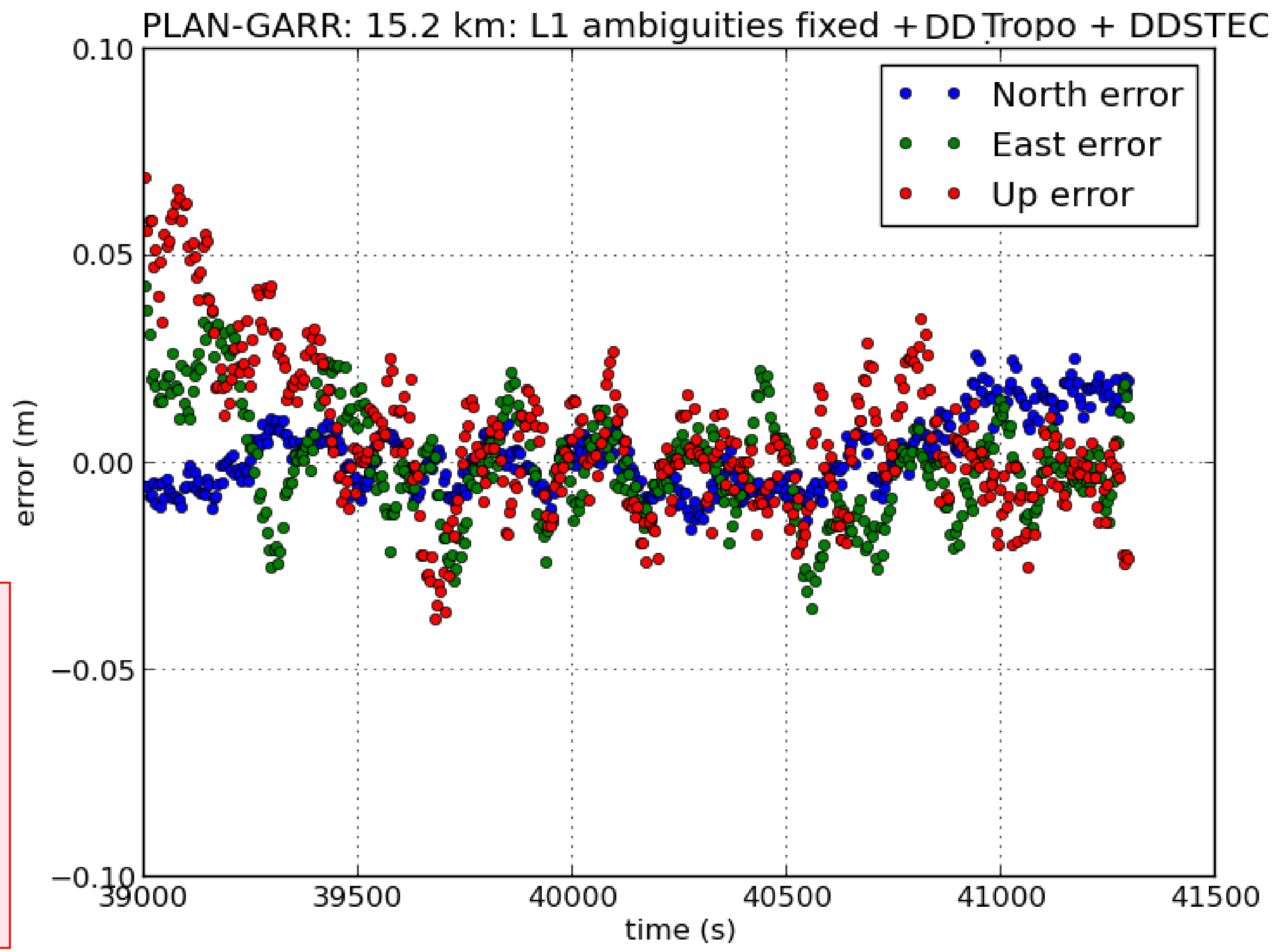
Plot the positioning error

```
graph.py -f L1fixStec.pos -x1 –y2 -s.- -l "North error"
         -f L1fixStec.pos -x1 –y3 -s.- -l "East error"
         -f L1fixStec.pos -x1 –y4 -s.- -l "UP error"
  --yn -.1 --yx .1 --xl "time (s)" --yl "error (m)" -t "PLAN-GARR: 15.2
km: L1 ambiguities fixed + Tropo + DDSTEC"
```

D.6.1. Estimate GARR coordinates with DDL1 (using tropo & DDSTEC corrections)

D.6. PLAN-GARR differential positioning with L1 carrier
(with ambiguity fixed and correcting from tropo. and DDSTEC)

Differential Positioning error with L1, ambiguities fixed with Tropo and DDSTEC removed.



Question:
Is any bias expected due to the L1-LC APCs, when removing the ionosphere using the unambiguous DDSTEC?

D.6.1. Estimate GARR coordinates with DDL1 (using tropo & DDSTEC corrections)

D.7. PLAN-GARR differential positioning with LC carrier
(with ambiguities fixed and correcting troposphere)

Repeat the previous exercise, but using the ionosphere free combination of carriers DDLC, with the ambiguities fixed.

The following procedure can be applied

a) generate a file with the following content;

```
[Time], [DD(LC-Amb)-DDRho-DDTrp], [Los_k - Los_13]
```

where:

Time= seconds of day

DD(LC-Amb) – DDRho –DDTrp = Prefit residulas (i.e., "y" values in program LS.f)

Los_k – Los_13 = The three components of the geometry matrix (i.e., matrix "a" in program LS.f)

D.7. Estimate GARR coordinates with DDLC FIXED (using tropo)

D.7. PLAN-GARR differential positioning with LC carrier
(with ambiguities fixed and correcting troposphere)

```
[Time], [DD(LC-Amb)-DDRho-DDTrp], [Los_k - Los_13]
```

```
cat DD_PLAN_GARR_13_ALL.fixL1L2 | gawk 'BEGIN{g2r=atan2(1,1)/45}
       {e1=$14*g2r;a1=$15*g2r;e2=$16*g2r;a2=$17*g2r;g=(77/60)**2;
       L1=$8-$18;L2=$10-$19;LC=(g*L1-L2)/(g-1);
       printf "%s %14.4f %8.4f %8.4f %8.4f \n", $6,LC-$11-$12,
      -cos(e2)*sin(a2)+cos(e1)*sin(a1),-cos(e2)*cos(a2)+cos(e1)*cos(a1),
                                              -sin(e2)+sin(e1)}' > LCmodel.dat
```

b) Compute the Least Squares solution

```
cat LCmodel.dat |LS > LCfix.pos
```

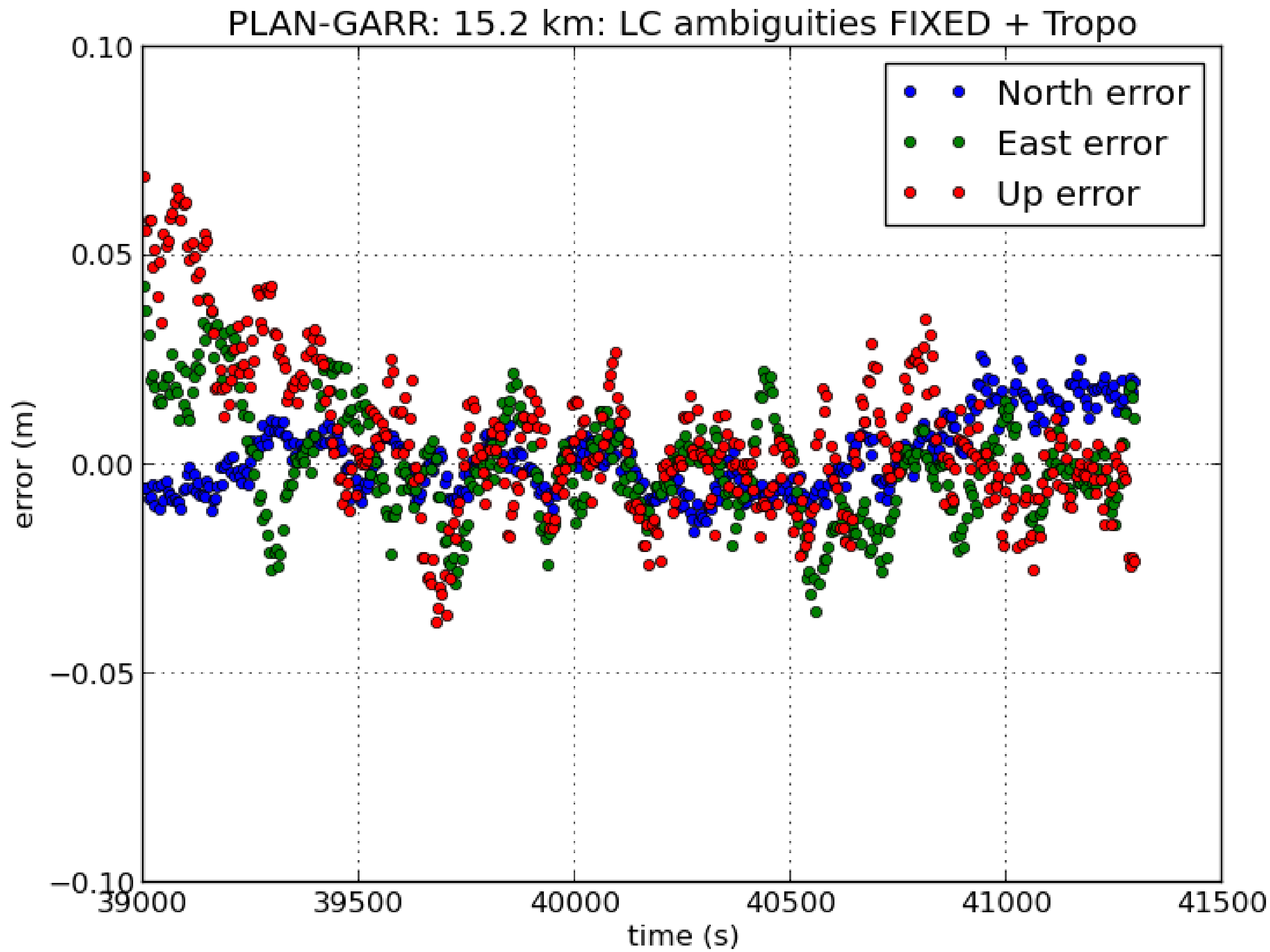
D.7. PLAN-GARR differential positioning with LC carrier (with ambiguities fixed and correcting troposphere)

Plot the positioning error

```
graph.py -f L1fixStec.pos -x1 –y2 -s.- -l "North error"
         -f L1fixStec.pos -x1 –y3 -s.- -l "East error"
         -f L1fixStec.pos -x1 –y4 -s.- -l "UP error"
  --yn -.1 --yx .1 --xl "time (s)" --yl "error (m)" -t "PLAN-GARR: 15.2
km: LC ambiguities FIXED + Tropo"
```

D.7. PLAN-GARR differential positioning with LC carrier
(with ambiguities fixed and correcting troposphere)

Differential Positioning error with DDLC, ambiguities fixed.



Question:
Compare this iono-free solution with that obtained with DDL1, removing the troposphere and ionosphere using the unambiguous DDSTEC.
Are the results the same? Why?

D.7. Estimate GARR coordinates with DDLC FIXED (using tropo)

Thanks for your attention

Acknowledgements

- To the University of Delft for the MATLAB files of LAMBDA method.
- To the INDRA company and Institut Cartogràfic de Catalunya for the data files of receivers IND1,IND2, IND2 and PLAN, GARR, respectively.
- To Adrià Rovira-Garcia for his contribution to the editing of this material and gLAB updating and integrating this learning material into the GLUE.

The ESA/UPC GNSS-Lab Tool suite (gLAB) has been developed under the ESA Education Office contract N. P1081434.