gAGE research group of Astronomy and Geomatics
Barcelona, Spain

Phase-based GNSS Data Processing (PPP) with the GPSTk

Salazar D., Sanz-Subirana, J., and Hernandez-Pajares M.

gAGE/UPC, Barcelona, Spain

Contents

- GPSTk overview
- Some current features
- GNSS Data Structures
- Simple code processing example
- PPP processing example
- Conclusions and future work

GPSTk overview

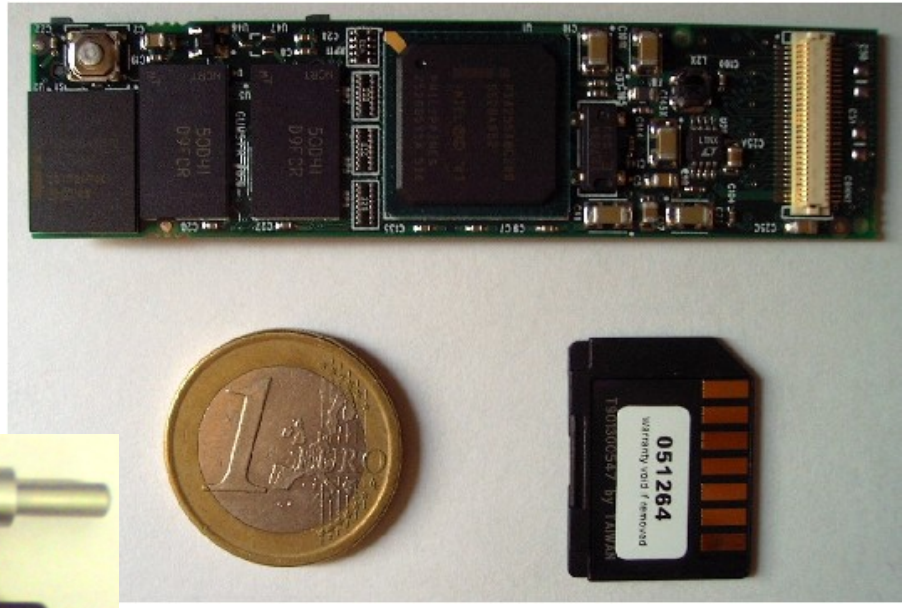
- GPSTk is:
  - A **library** to write GNSS software
  - Includes example **applications**.
- GPSTk is **Free Software** (LGPL):
  - Both non-commercial *and*
  - commercial applications.

GPSTk overview

- Written in ISO-standard C++ (portable).
- Operative System portability:
  - Works in Windows, Linux, Mac OSX, AIX, Solaris, etc.
- Hardware portability:
  - Big and (very) small systems, both 32 and 64 bits.

GPSTk portability

GUMSTIX Miniature computer



NOKIA Internet Tablets

gAGE research group of Astronomy and Geomatics
Barcelona, Spain

GPSTk overview

- It was initiated at the Advanced Research Laboratories at Texas University (ARL:UT)
- Now it has several official developers around the world.
- Project website:

http://www.gpstk.org

Some current features

- Time conversions.
- RINEX files reading/writing:
  - Observation
  - Ephemeris
  - Meteorological
- Ephemeris computation:
  - Broadcast
  - SP3.

Some current features

- Mathematical tools: Matrices, vectors, interpolation, numeric integration, LMS, W-LMS, Kalman filter, etc.
- Application development support:
  - Exceptions handling.
  - Command line framework.
- Full support for Antex files.

Some current features

- Several tropospheric models: Saastamoinen, Goad-Goodman, New Brunswick, etc.
- Classes for precise modeling: Phase centers, Wind-up, gravitational delay, etc.
- Tidal models:
  - Solid tides
  - Ocean loading
  - Pole tides

GNSS Data Structures

- GNSS Data Structures (GDS) were designed to help in GNSS data processing.
- They address GNSS ***data management*** problems.
- **NOTE**: *Currently only available in the development repository!!!*

GNSS Data Structures

- Almost all GNSS data may be identified (or indexed) by:
  - **Source**: Receiver logging data.
  - **Satellite**: GPS, GALILEO, GLONASS...
  - **Epoch**: Time the data belongs to.
  - **Type**: C1, P1, L2, etc, and other types.

Key point: *Save Meta-Information:*

Save 4 indexes to identify

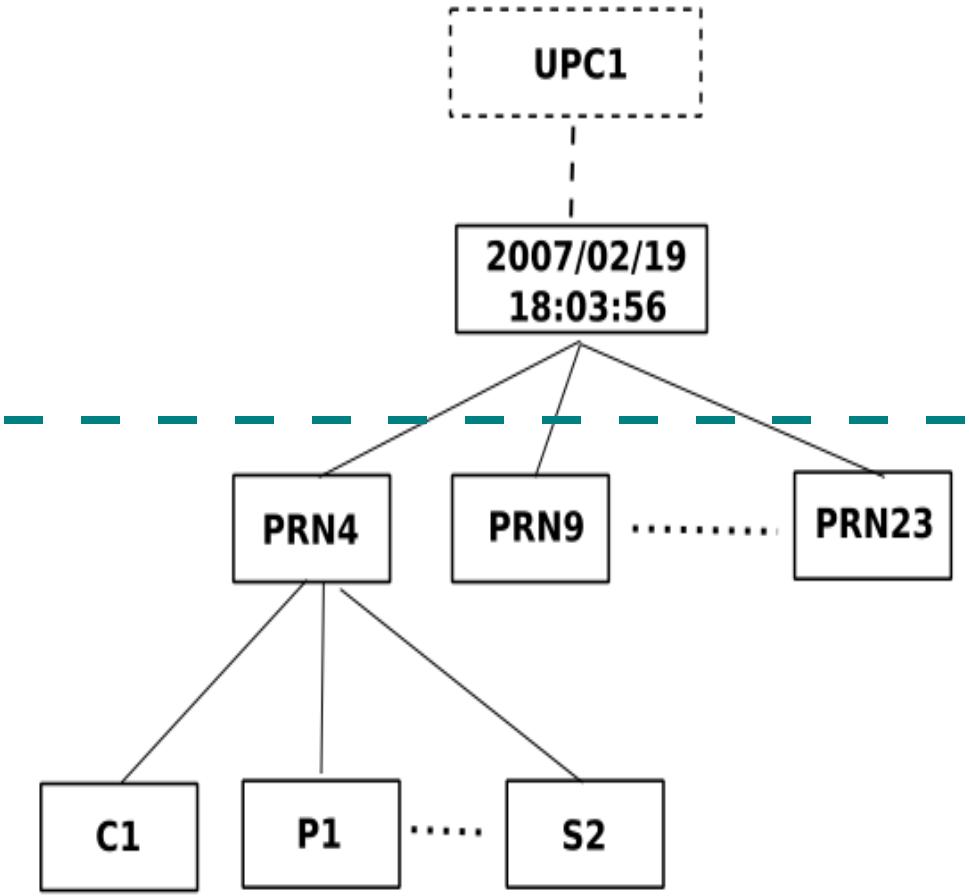
each piece of data

GNSS Data Structures

- **Source**: Implemented as class **SourceID**.
- **Satellite**: Implemented as class **SatID**.
  - Several systems: GPS, Galileo, Glonass...
- **Epoch**: Implemented as class **DayTime**.
- **Type**: Implemented as class **TypeID**.
  - Includes basic observations: C1, P1, etc.
  - Other “types”: Relativity, tropo, etc.
  - *You can create new types as needed!!!.*

Implementation

- GDS have two parts:



Header: *Common* information

"trunk" of tree

"branches" of tree

Body: *Specific* information

Implementation

- GDS provide several methods to ease handling. For instance:

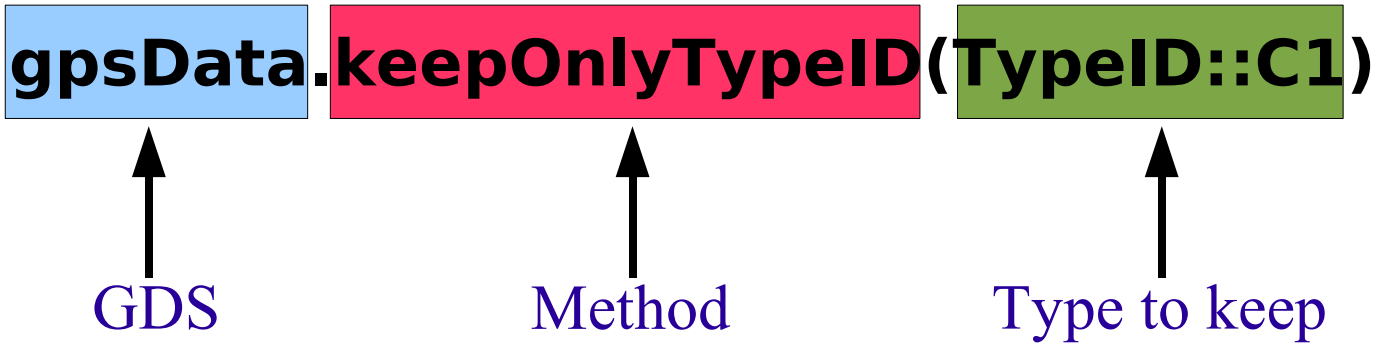
Keep only C1 observable in GDS:

gpsData.keepOnlyTypeID(TypeID::C1)

Implementation

- GDS provide several methods to ease handling. For instance:

Keep only C1 observable in GDS:



Processing paradigm

- Operator "**>>**" is *overloaded*:

  We *redefine it* in C++ to make data "*flow*" from one processing step to the next

Processing paradigm

Example:

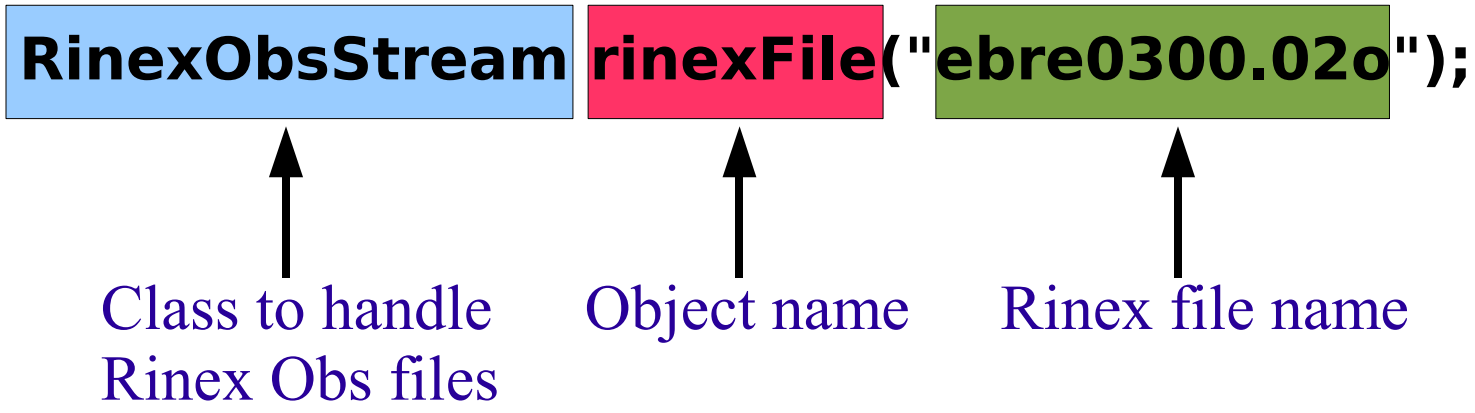
Declare object "rinexFile" to take data out of Rinex observation file:

RinexObsStream rinexFile("ebre0300.02o");

Processing paradigm

Example:

Declare object "rinexFile" to take data out of Rinex observation file:



Processing paradigm

Example:

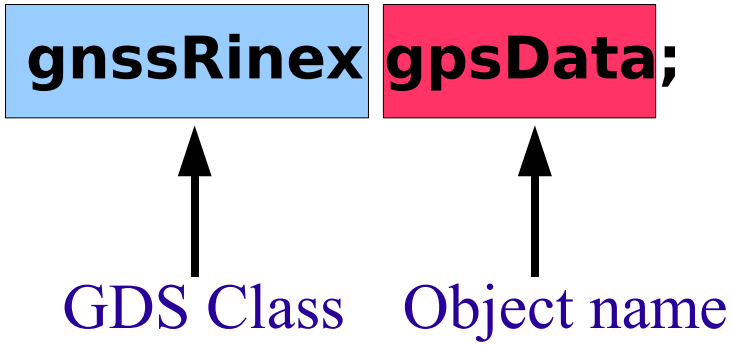
*Declare object "gpsData". This is a **GDS**:*

gnssRinex gpsData;

Processing paradigm

Example:

*Declare object "gpsData". This is a **GDS**:*



Processing paradigm

Example:

*Combine everything and take **one epoch** of data **out** of Rinex file **into** GDS:*

```
RinexObsStream rinexFile("ebre0300.02o");
gnssRinex gpsData;

rinexFile  >>  gpsData;
```

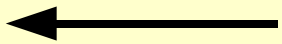
Processing paradigm

Example:

*Combine everything and take **one epoch** of data **out** of Rinex file **into** GDS:*

```
RinexObsStream rinexFile("ebre0300.02o");
gnssRinex gpsData;

rinexFile >> gpsData;
```



Processing line

Processing paradigm

Example:

*Combine everything and take **one epoch** of data **out** of Rinex file **into** GDS:*

```
RinexObsStream rinexFile("ebre0300.02o");
gnssRinex gpsData;

rinexFile >> gpsData;
```

Processing line

"Flow" operator

Processing paradigm

- **Main Idea**: GNSS data processing becomes like an "*assembly line*".
  - The GDS "*flows*" from one "*workstation*" to the next, like a car in factory.
  - And like the car, the GDS *changes* along the processing line.

- For the developer, the GDS is like a "white box" holding all the information needed, and properly indexed.

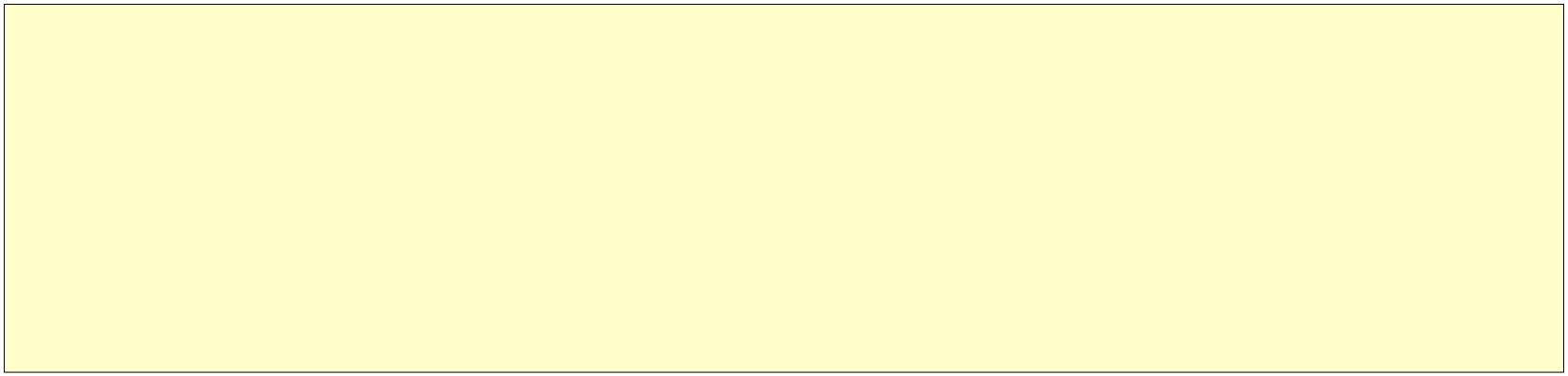
Processing *code-based* GPS data with the GPSTk and GDS

A simple example:

C1 code observable +
least-mean squares solver
Plain standard GPS processing

C1 + LMS

- Get data out of Rinex file, ***one epoch at a time***
  - Get data into GDS (*gpsData*)
  - Do it until Rinex file ends.



C1 + LMS

- Get data out of Rinex file, ***one epoch at a time***
  - Get data into GDS (*gpsData*)
  - Do it until Rinex file ends.

```
while ( rinexFile >> gpsData )
{

}
```

C1 + LMS

- Get data into modeling object *gpsModel*
  - *GpsModel* was previously initialized:
    - Ephemeris, tropospheric model, Klobuchar ionospheric model, etc.
  - Initialization phase is not shown here.

```
while ( rinexFile >> gpsData )
{

}
```

C1 + LMS

- Get data into modeling object *gpsModel*
  - *GpsModel* was previously initialized:
    - Ephemeris, tropospheric model, Klobuchar ionospheric model, etc.
  - Initialization phase is not shown here.

```
while ( rinexFile >> gpsData )
{
   gpsData >> gpsModel
}
```

C1 + LMS

- Resulting GDS with original data and modeled data gets into solver object *lmsSolver*
  - *lmsSolver* is from *SolverLMS* class.
  - *lmsSolver* was previously initialized.
  - Initialization phase is not shown here.

```
while ( rinexFile >> gpsData )
{
   gpsData >> gpsModel
}
```

C1 + LMS

- Resulting GDS with original data and modeled data gets into solver object *lmsSolver*
  - *lmsSolver* is from *SolverLMS* class.
  - *lmsSolver* was previously initialized.
  - Initialization phase is not shown here.

```
while ( rinexFile >> gpsData )
{
   gpsData >> gpsModel >> lmsSolver;
}
```

C1 + LMS

- Resulting GDS with original data and modeled data gets into solver object *lmsSolver*
  - *lmsSolver* is from *SolverLMS* class.
  - *lmsSolver* was previously initialized.
  - Initialization phase is not shown here.

```
while ( rinexFile >> gpsData )
{
    gpsData >> gpsModel >> lmsSolver;
}
```

← Processing line

C1 + LMS

- **All GNSS data processing is done!!!**
  - Print results to the screen.
  - C++ *cout* object is used to print to screen.
  - Solution is inside *lmsSolver*: Ask for type.

```
while ( rinexFile >> gpsData )
{
   gpsData >> gpsModel >> lmsSolver;




}
```

C1 + LMS

- **All GNSS data processing is done!!!**
  - Print results to the screen.
  - C++ *cout* object is used to print to screen.
  - Solution is inside *lmsSolver*: Ask for type.

```
while ( rinexFile >> gpsData )
{
   gpsData >> gpsModel >> lmsSolver;

   cout << lmsSolver.getSolution(TypeID::dx) << " ";
   cout << lmsSolver.getSolution(TypeID::dy) << " ";
   cout << lmsSolver.getSolution(TypeID::dz) << endl;
}
```

Processing phase-*based* GPS data with the GPSTk and GDS

Example of *Precise Point Positioning (PPP)*

PPP (static, forward mode)

```
correctObservables.setExtraBiases(tides);
```

Set “correcting” object with tide information.
It also includes RX antenna phase center and eccentricity

gAGE research group of Astronomy and Geomatics
Barcelona, Spain

PPP (static, forward mode)

```
correctObservables.setExtraBiases(tides);

gpsData >> basicGPSModel
```

Compute basic components of model

gAGE research group of Astronomy and Geomatics
Barcelona, Spain

PPP (static, forward mode)

```
correctObservables.setExtraBiases(tides);

gpsData >> basicGPSModel
        >> removeEclipsedSatellites
```

Remove satellites in eclipse

gAGE research group of Astronomy and Geomatics
Barcelona, Spain

PPP (static, forward mode)

```
correctObservables.setExtraBiases(tides);

gpsData >> basicGPSModel
        >> removeEclipsedSatellites
        >> gravitationalDelay
```

Compute gravitational delay

gAGE research group of Astronomy and Geomatics
Barcelona, Spain

PPP (static, forward mode)

```
correctObservables.setExtraBiases(tides);

gpsData >> basicGPSModel
        >> removeEclipsedSatellites
        >> gravitationalDelay
        >> computeSatellitePhaseCenter
```

Compute the effect of satellite phase centers

gAGE research group of Astronomy and Geomatics
Barcelona, Spain

PPP (static, forward mode)

```
correctObservables.setExtraBiases(tides);

gpsData >> basicGPSModel
        >> removeEclipsedSatellites
        >> gravitationalDelay
        >> computeSatellitePhaseCenter
        >> correctObservables
```

Correct observables from tides, RX phase center, etc

gAGE research group of Astronomy and Geomatics
Barcelona, Spain

PPP (static, forward mode)

```
correctObservables.setExtraBiases(tides);

gpsData >> basicGPSModel
        >> removeEclipsedSatellites
        >> gravitationalDelay
        >> computeSatellitePhaseCenter
        >> correctObservables
        >> windUp
```

Compute wind-up effect

gAGE research group of Astronomy and Geomatics
Barcelona, Spain

PPP (static, forward mode)

```
correctObservables.setExtraBiases(tides);

gpsData >> basicGPSModel
        >> removeEclipsedSatellites
        >> gravitationalDelay
        >> computeSatellitePhaseCenter
        >> correctObservables
        >> windUp
        >> computeTropo
```

Compute tropospheric effect (Niell model)

PPP (static, forward mode)

```
correctObservables.setExtraBiases(tides);

gpsData >> basicGPSModel
        >> removeEclipsedSatellites
        >> gravitationalDelay
        >> computeSatellitePhaseCenter
        >> correctObservables
        >> windUp
        >> computeTropo
        >> computeLinearCombinations
```

Compute common linear combinations: PC, LC, LI, ...

gAGE research group of Astronomy and Geomatics
Barcelona, Spain

PPP (static, forward mode)

```
correctObservables.setExtraBiases(tides);

gpsData >> basicGPSModel
        >> removeEclipsedSatellites
        >> gravitationalDelay
        >> computeSatellitePhaseCenter
        >> correctObservables
        >> windUp
        >> computeTropo
        >> computeLinearCombinations
        >> markCSLI
        >> markCSMW
```

Marck cycle slips: Two complementary algorithms

gAGE research group of Astronomy and Geomatics
Barcelona, Spain

PPP (static, forward mode)

```
correctObservables.setExtraBiases(tides);

gpsData >> basicGPSModel
        >> removeEclipsedSatellites
```

Compute prefit residuals

```
        >> computeSatellitePhaseCenter
        >> correctObservables
        >> windUp
        >> computeTropo
        >> computeLinearCombinations
        >> markCSLI
        >> markCSMW
        >> computePrefitResiduals
```

PPP (static, forward mode)

```
correctObservables.setExtraBiases(tides);

gpsData >> basicGPSModel
        >> removeEclipsedSatellites
```

Decimate data if epoch is not a multiple of 900 seconds

```
        >> computeSatellitePhaseCenter
        >> correctObservables
        >> windUp
        >> computeTropo
        >> computeLinearCombinations
        >> markCSLI
        >> markCSMW
        >> computePrefitResiduals
        >> decimateData
```

gAGE research group of Astronomy and Geomatics
Barcelona, Spain

PPP (static, forward mode)

```
correctObservables.setExtraBiases(tides);

gpsData >> basicGPSModel
        >> removeEclipsedSatellites
        >> computeSatellitePhaseCenter
        >> correctObservables
        >> windUp
        >> computeTropo
        >> computeLinearCombinations
        >> markCSLI
        >> markCSMW
        >> computePrefitResiduals
        >> decimateData
        >> baseChange
```

Change from ECEF to ENU reference frame

gAGE research group of Astronomy and Geomatics
Barcelona, Spain

PPP (static, forward mode)

```
correctObservables.setExtraBiases(tides);

gpsData >> basicGPSModel
        >> removeEclipsedSatellites
```

Solve equations with a properly adjusted Kalman filter

```
        >> computeSatellitePhaseCenter
        >> correctObservables
        >> windUp
        >> computeTropo
        >> computeLinearCombinations
        >> markCSLI
        >> markCSMW
        >> computePrefitResiduals
        >> decimateData
        >> baseChange
        >> pppSolver;
```

PPP (static, forward mode)

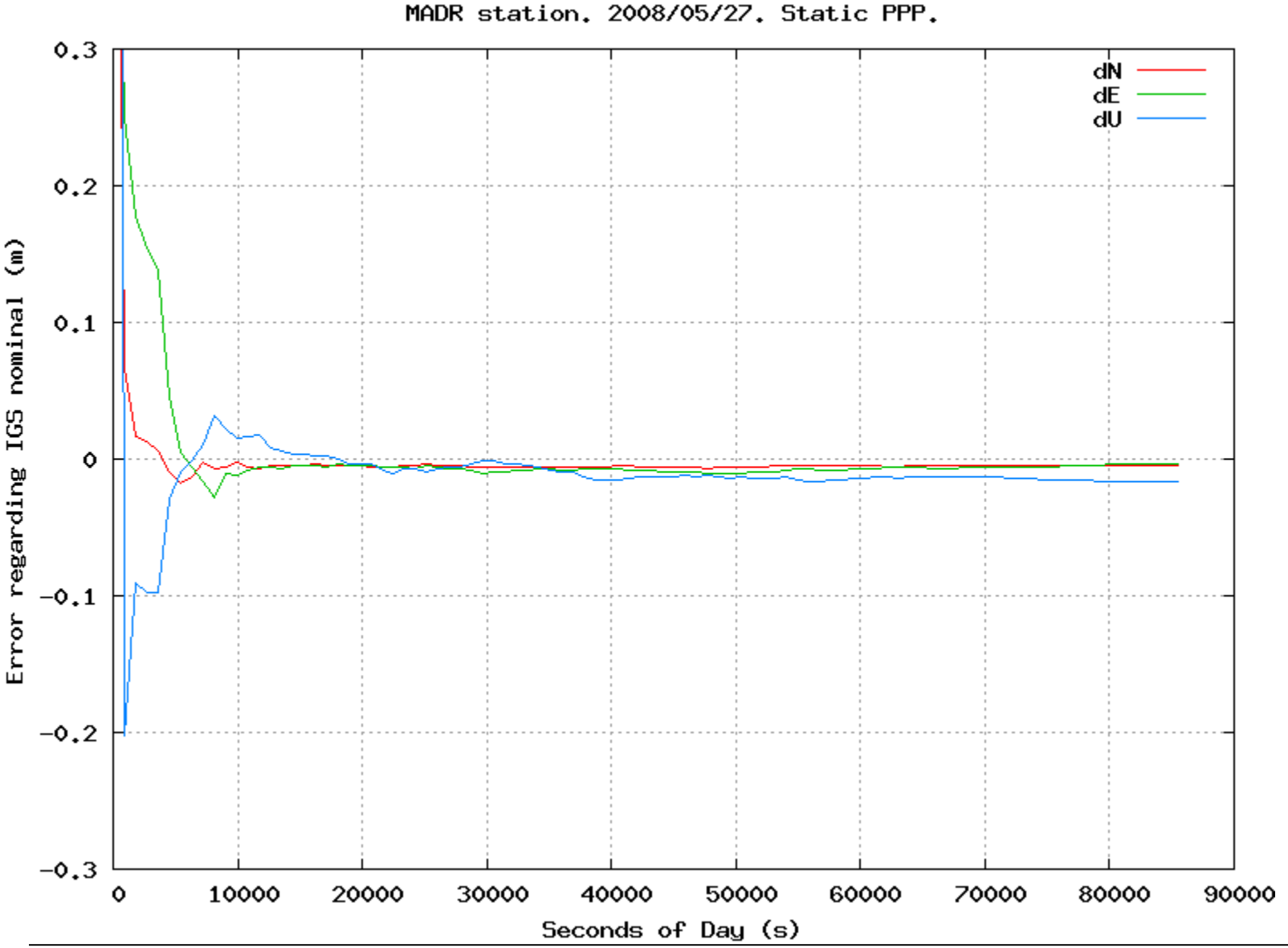
```
correctObservables.setExtraBiases(tides);

gpsData >> basicGPSModel
        >> removeEclipsedSatellites
        >> gravitationalDelay
        >> computeSatellitePhaseCenter
        >> correctObservables
        >> computeLinearCombinations
        >> markCSLI
        >> markCSMW
        >> computePrefitResiduals
        >> decimateData
        >> baseChange
        >> pppSolver;
```

All this processing is repeated for each epoch

gAGE research group of Astronomy and Geomatics
Barcelona, Spain

PPP: MADR 2008/05/27. static. forward



PPP (kinematic, forward mode)

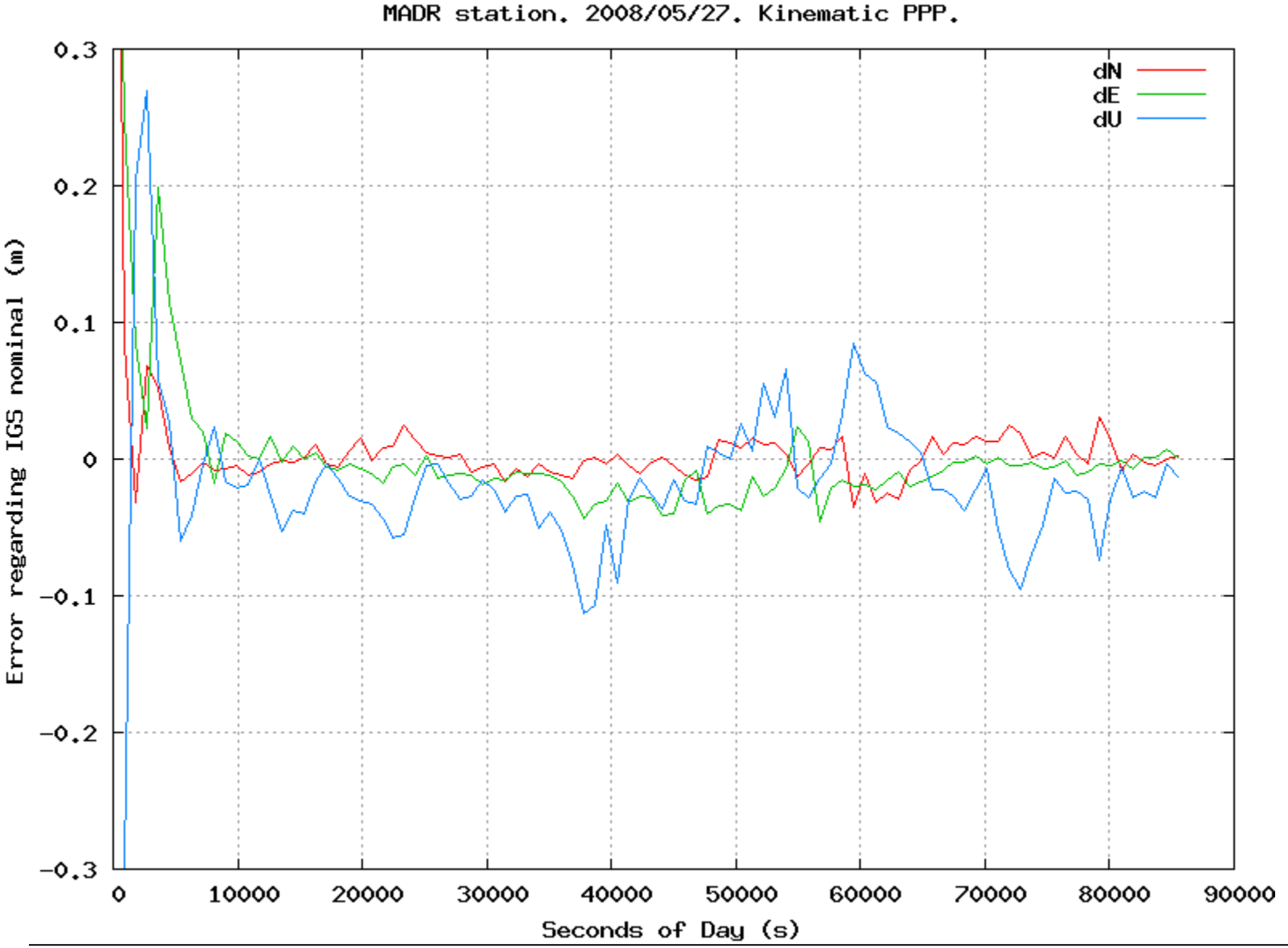
```
WhiteNoiseModel  newCoordinatesModel(100.0);

pppSolver.setCoordinatesModel(&newCoordinatesModel);

correctObservables.setExtraBiases(tides);

gpsData >> basicGPSModel
        >> removeEclipsedSatellites
        ...
        ...
        >> pppSolver;
```

PPP:MADR 2008/05/27, kinematic, forward



PPP (static, forward-backward)

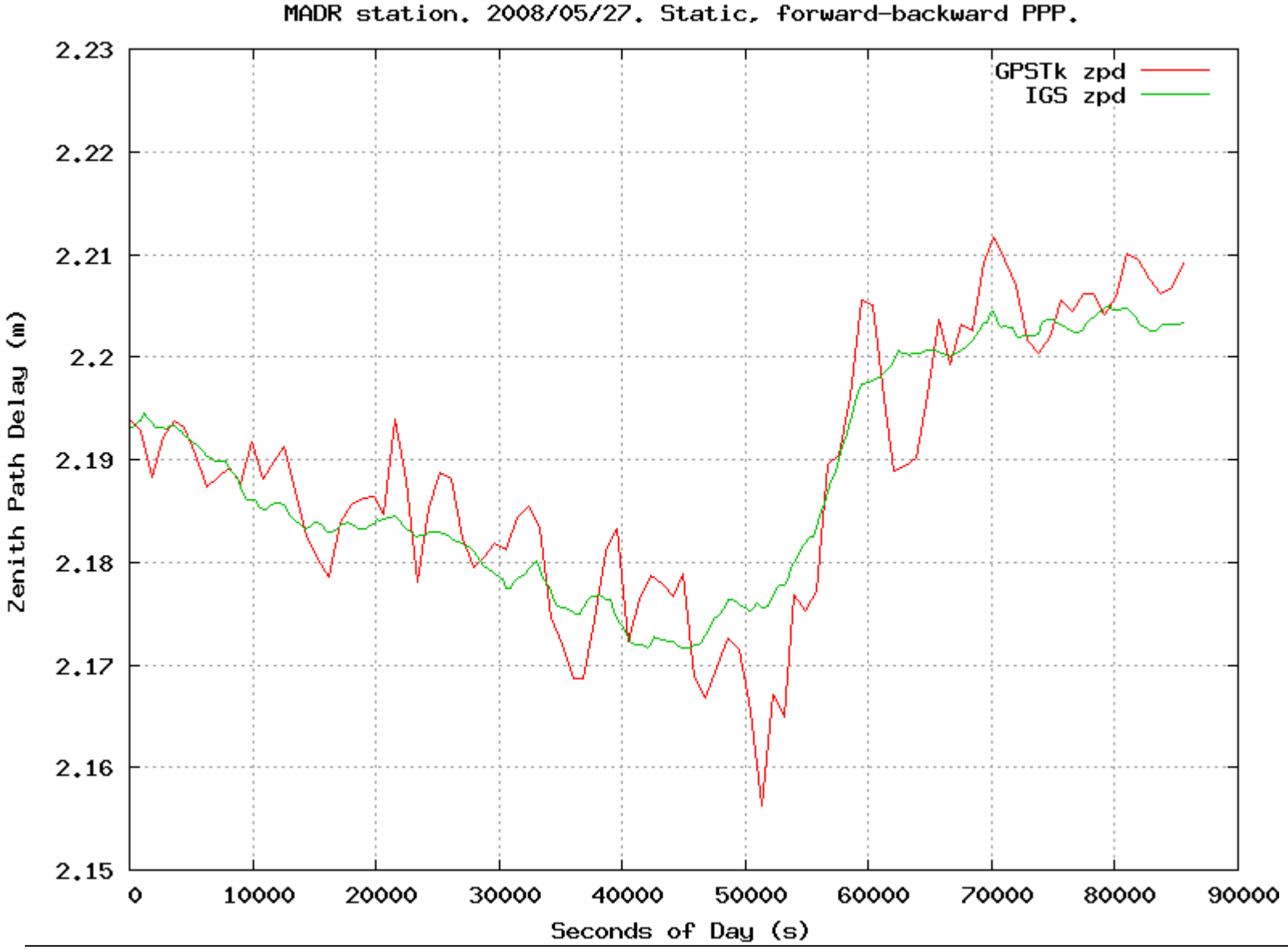
```
SolverPPPFB  fbpppSolver;

   correctObservables.setExtraBiases(tides);

   gpsData >> basicGPSModel
           >> removeEclipsedSatellites
           ...
           >> fbpppSolver;
   ...
fbpppSolver.ReProcess(4);

while ( fbpppSolver.LastProcess(gpsData) )
{
   cout << fbpppSolver.getSolution(TypeID::dLat) << " ";
   cout << fbpppSolver.getSolution(TypeID::dLon) << " ";
   cout << fbpppSolver.getSolution(TypeID::dH) << " ";
}
```

PPP: MADR 2008/05/27, static, fw/back



Conclusions and future work

- GPSTk is already a solid base to work upon, saving tedious work to the researcher.
- GDS are providing a powerful (yet flexible and easy to use) processing framework.
- PPP results using GDS are in agreement with other GNSS processing software.
- There are several areas for improvement:
  - RINEX version 3 handling.
  - IONEX files processing.
  - Robust outlier detection classes.
  - More sophisticated models.
  - Other GNSS processing strategies (RTK).

Thanks for your attention and time!

Dagoberto.Jose.Salazar@upc.edu