1. Introduction

goGPS is a software package designed to perform GNSS positioning, either in post-processing or real-time. It is open-source software governed by the GNU General Public License.

goGPS is developed in MATLAB and it is aimed at providing a tool useful for carrying out GNSS-related research. It provides a platform for testing new algorithms and experimental functionalities. A Java version is also available from the goGPS software repository.

The main features of the software are:

- L1 positioning (relative and undifferenced)
- Multi-constellation support – GPS, GLONASS, Galileo, BeiDou, QZSS
- Integer ambiguity resolution using the LAMBDA method (for GPS only; GLONASS ambiguity resolution is not supported)
- EGNOS (SBAS) support through EMS service (automatic download of EMS files)
- Post-processing of RINEX v2.11, v2.12, v3.01 and goGPS binary data files
- Real-time processing (only for relative positioning, using an NTRIP caster)
- RTCM 3.x support
- u-blox LEA-xT support
- SkyTraq S1315F-RAW support
- Fastrax IT03 support
- NVS BINR support
- Plotting solutions in MATLAB figures and/or Google Earth
- Works on Windows and UNIX machines

2. Requirements

goGPS is being developed and tested in both MATLAB and Java environments on Windows and UNIX (Linux / Mac OS X) platforms. Note that the MATLAB GUI appearance is significantly influenced by the operating system and versions of MATLAB and Java being used. In some cases, the font size might appear small and the background color might be different than expected (i.e. different tones of grey). And sometimes the layout of GUI objects may be slightly different.

The following elements are needed in order to use goGPS with MATLAB:

- Windows or UNIX-based operating system
- MATLAB 7.6 or greater

The following elements are also needed for specific tasks:

For post-processing tasks	For real-time tasks
• RINEX observation file for the roving receiver • RINEX observation file for the master station (for relative positioning only) • RINEX navigation file (at the moment only broadcast ephemeris are supported) • LAMBDA v3 toolbox from Verhagin and Li (2012); v2.0 will be used if v3.0 is not installed. LAMBDA v3 use is recommended (especially for the LAMBDA example, see Installation section). (goGPS binary data saved during a real-time session can be used instead of RINEX files)	• Mathworks Instrument Control Toolbox • a GNSS receiver providing raw data to a COM port on the PC with appropriate drivers installed (currently u-blox UBX, Fastrax IT03, SkyTraq and NVS BINR binary protocols are supported) • GNSS permanent station(s) broad-casting raw data in RTCM 3.x format through NTRIP protocol (at least '1002' or '1004' messages) when differential positioning with pseudo-range or phase observations is used. • Google Earth (optional)

When working in real-time, goGPS can automatically plot the positioning solution on Google Earth. In order to use this feature, it is necessary to install Google Earth in advance. If error ellipses plotted on Google Earth produce odd output on Windows, please switch the Google Earth rendering engine to DirectX.

3. Installation

The goGPS software can be downloaded from

http://www.gogps-project.org/

To install goGPS simply unzip the goGPS_v0.4.3.zip file into a folder. It is also recommended to install the LAMBDA v3.0 toolbox by downloading the lambda-3.0.zip file from

http://gnss.curtin.edu.au/research/lambda.cfm

Copy all of the .m files from the lambda-3.0/src folder into the goGPS_v0.4.3/goGPS/positioning/lambda/lambda_v3 folder. If LAMBDA v3.0 is not installed, goGPS will use the LAMBDA v2.0 toolbox provided with goGPS.

To run goGPS, make sure you have a working MATLAB environment. Open the MATLAB application and set the current folder to the unzipped goGPS_v0.4.3/goGPS folder. Then type "goGPS" in the MATLAB command window and press [Return].

4. goGPS Processing Modes

goGPS can work in post-processing and real-time mode. The functional schemes for both processing modes are given in Figures 1 and 2.

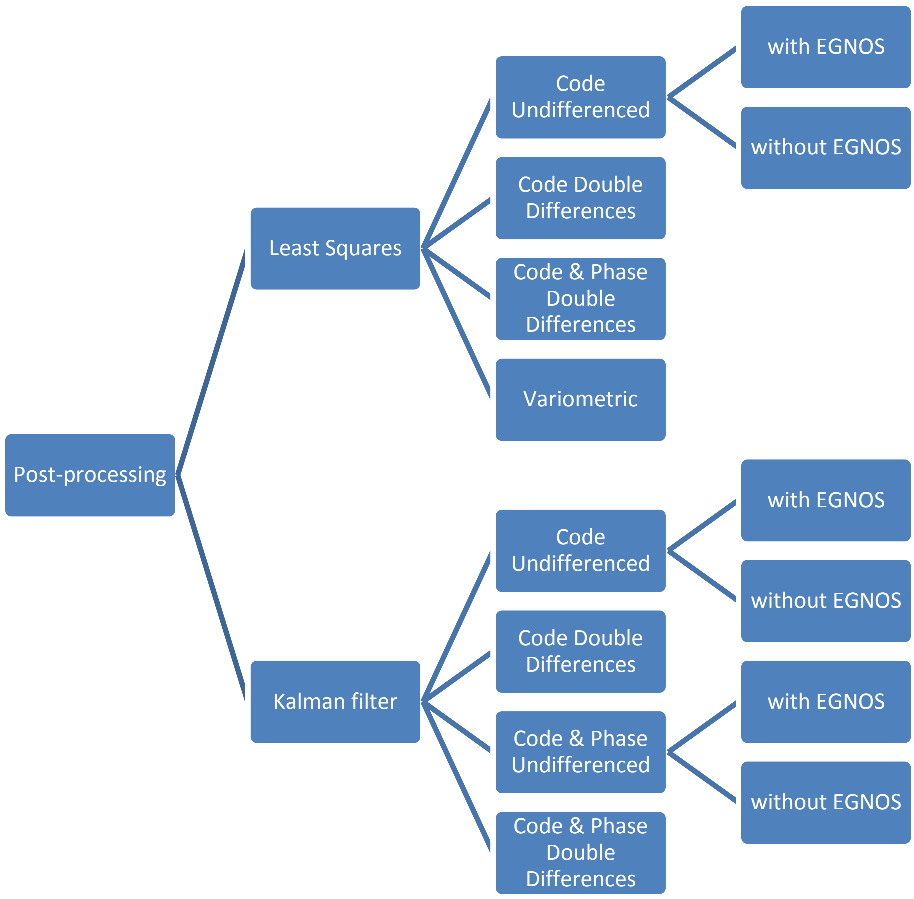


Figure 1: Scheme of the post-processing mode.

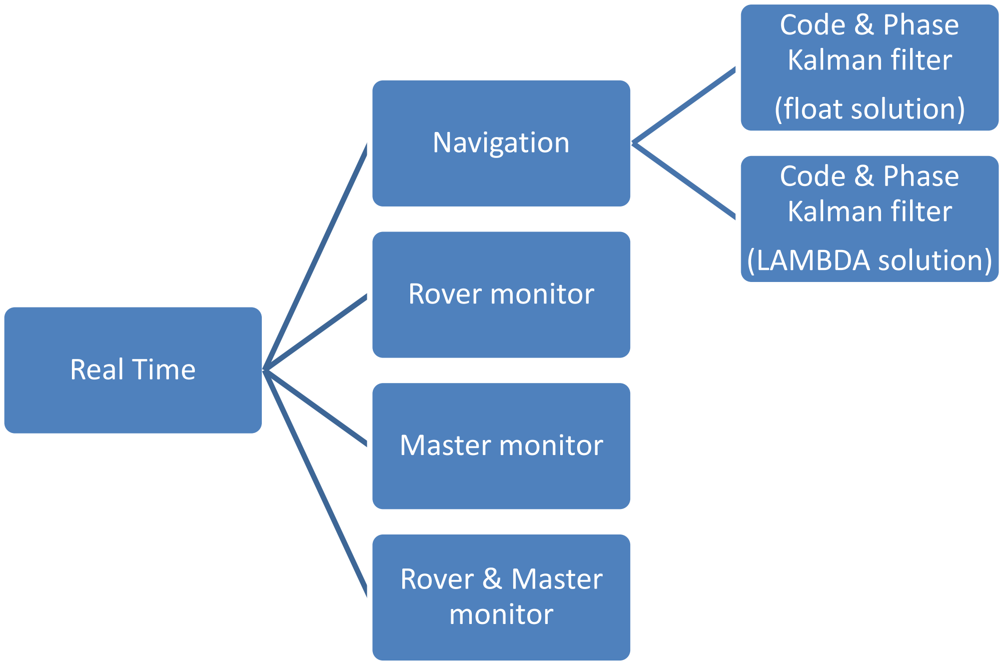


Figure 2: Scheme of the real-time mode.

5. goGPS Graphical User Interface (GUI)

The default goGPS graphical user interface (GUI) is shown in Figure 3:

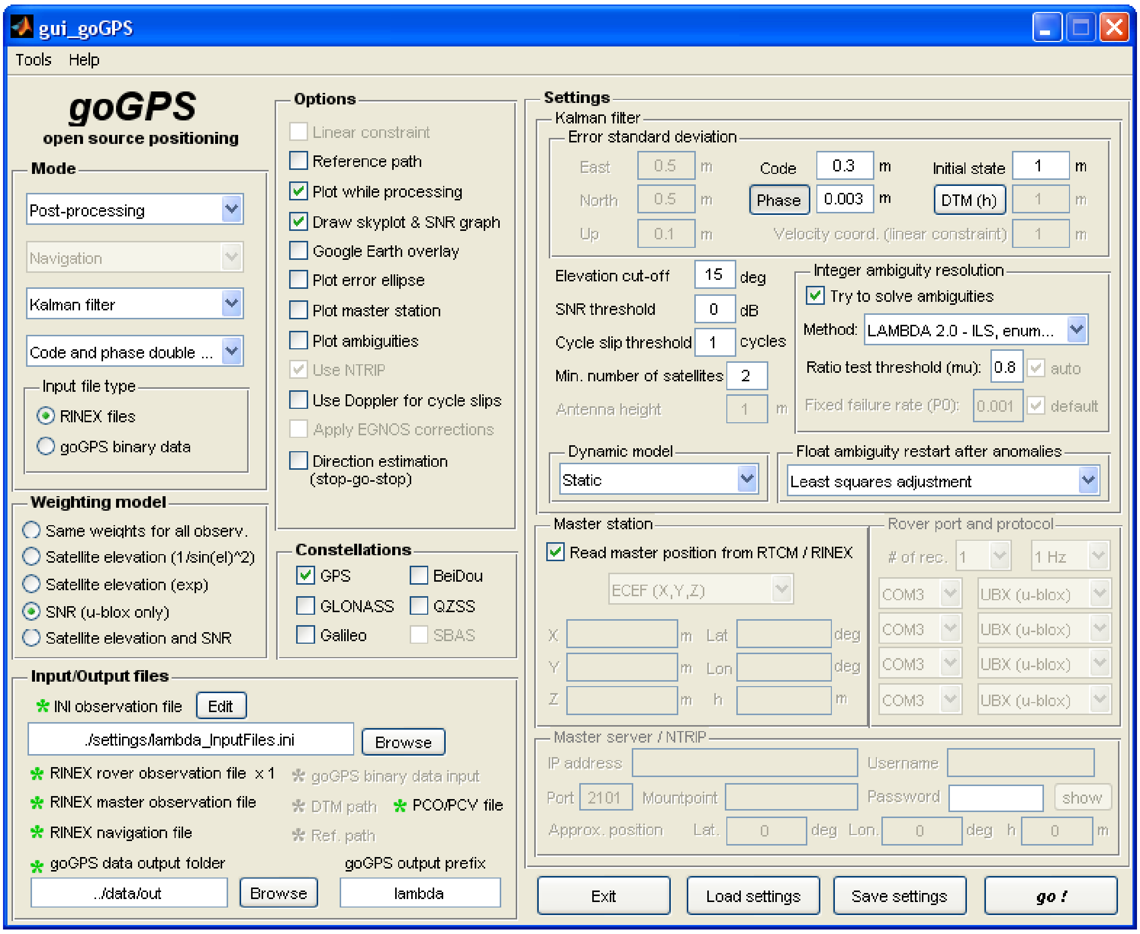
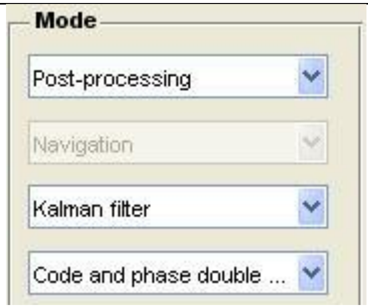
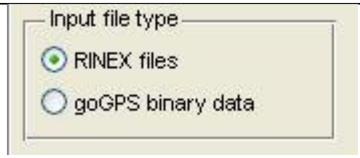
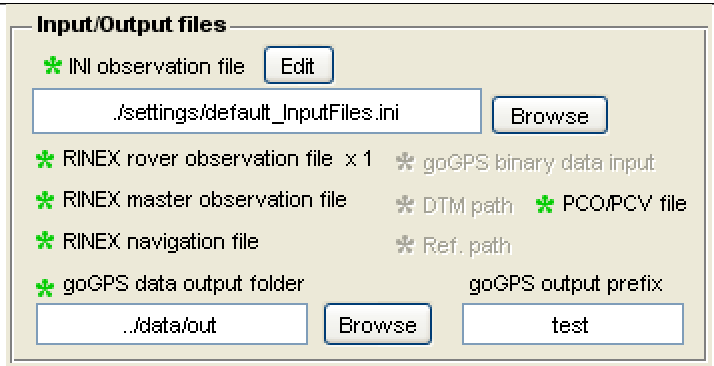
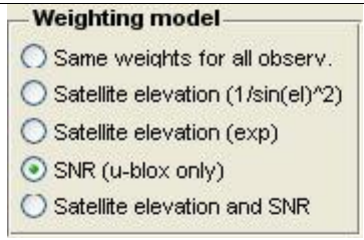


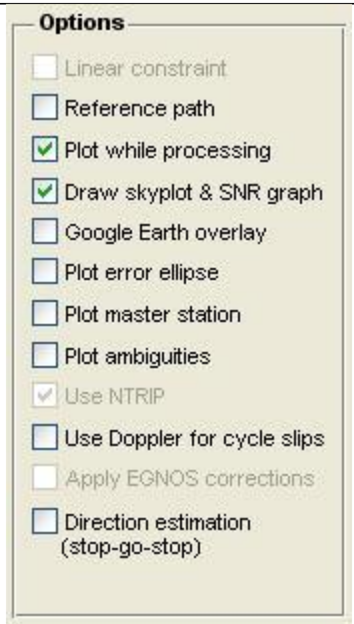
Figure 3: goGPS default MATLAB GUI (on Windows)

The different elements of the GUI are described in detail in the following table.

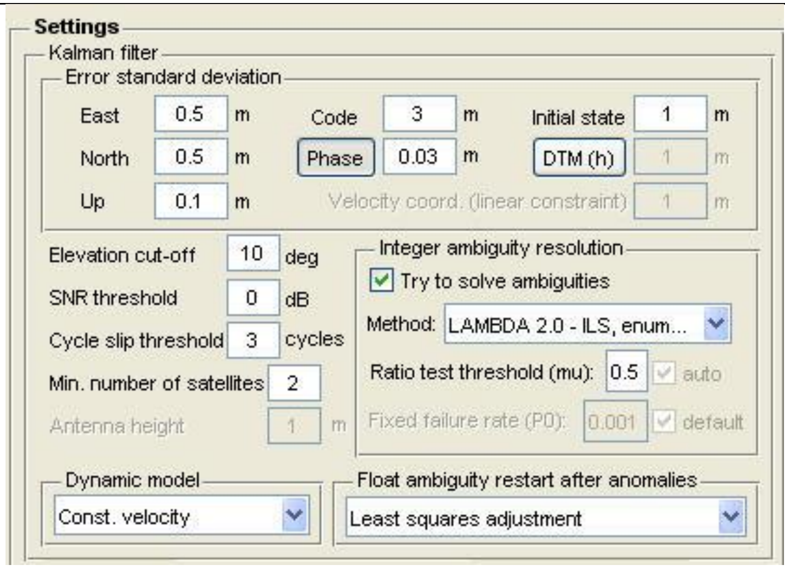
GUI Area	Options	Functions
Mode		The first selection menu in the Mode area specifies whether Post-processing or Real-time mode is to be used. All positioning/monitoring modes are given in the above schemes in Figures 2 and 3. For Post-processing mode, two processing options available: • Least-squares • Kalman filter The Least-squares option allows for selecting the following four types of observables to process: • Code undifferenced • Code double-difference • Code and phase double differences (for Lambda) • Variometric approach for velocity estimation (Colosimo et al. 2011) The Kalman filter options allows for selecting the following four types of observations to process: • Code undifferenced • Code double-difference • Code and phase undifferenced • Code and phase double differences For Real-time mode, four options are available: • Navigation: Receive and synchronize raw data streams from one rover receiver and one master station, and perform positioning on-the-fly by Kalman filter applied to code and phase double differenced observations. • Rover monitor: Display the raw data received by the

		rover station and save in binary format, to be used in post-processing mode later. It means the Rover monitor only activates the Rover port selector and a supported receiver needs to be connected. • Master monitor: Display the raw data received by the master station and in binary format, to be used in post-processing mode later. It requires an internet connection to receive information from the defined permanent station. • Rover and Master monitor: The Rover monitor and Master monitor options combined.
Input file type		In post-processing mode, users can select either RINEX or binary data files generated by goGPS (proprietary format) as input. goGPS stores the data collected in a goGPS binary data format during the real time functions. In real-time mode, this area is not active (dimmed).
Input/Output files		In Post-processing mode, the user can specify the INI observation settings file, the goGPS data output folder, and the goGPS output prefix used for the output file names. In Real-time mode, the user can only specify the goGPS data output folder, and the goGPS output file name prefix. The INI observation settings file is not selectable. If the user clicks the “Edit” button for the INI observation file, an interface appears that allows the user to directly modify the INI file within MATLAB (alternatively, it can also be

		externally modified using other text editing software). The INI file contains the names of all the required input files. More details about the INI editor are given at the end of this section. The color of the asterisk beside each file type indicates the following status: • red: a required filename or path was not set • yellow: a required filename or path was set, but it does not exist on the file system (i.e. MATLAB could not find it) • green: the filename or path is correctly set.
Weighting model	 	The user can select different observation weighting functions. The signal-to-noise (SNR)-based weighting model is valid only for receivers from u-blox AG. In general the option “satellite elevation (1/sin(el)^2)” is recommended. The “Satellite elevation and SNR” option applies a weighting function that combines the (1/sin(el)^2) model and the SNR-based model (therefore this option as well is valid only for receivers from u-blox AG). More information about these options is available in Realini and Reguzzoni (2013).
Constellations	 	Selects the satellite constellations to be included in the processing. The integer ambiguity resolution by the LAMBDA method can be applied only if all the available satellites use the same frequency (e.g. single-system processing, excluding GLONASS; GPS+Galileo; GPS+QZSS; etc.).

Options		"Linear constraint": Applies the goGPS Kalman filter on double differenced code and phase observations by constraining the solution to be estimated along a linear path. This option can be useful, for example, when estimating the position of a train moving on a railroad of known geometry. The "Reference path" option must be selected first to use this option. "Reference path": Used to load a reference (piece-wise) path, defined by a data file specified in the RefPath section of the INI settings file; this path can be used for two different purposes: to constrain post-processing solutions (if "Linear constraint" is selected), or to make goGPS compute the positioning accuracy with respect to the reference path (i.e. by computing mean and standard deviation of the 2D and 3D perpendicular distance of each estimated position from the path). . "Use NTRIP": Enabled by default and makes goGPS act as an NTRIP client for connecting to an NTRIP caster to receive an RTCM 3.x stream. The user can select this option for real-time processing tasks by relative positioning with a VRS (Virtual Reference Station). If the option is not selected, goGPS will directly connect to the specified server without using the NTRIP protocol (e.g. if the RTCM stream is simply sent to an open TCP/IP port). The option is selectable only when you use a real-time mode that involves receiving raw data from a permanent station. "Use Doppler for cycle slips": Detect cycle slips by comparing

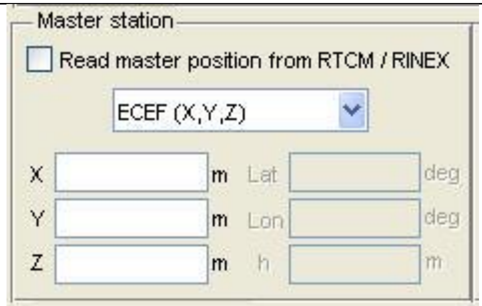
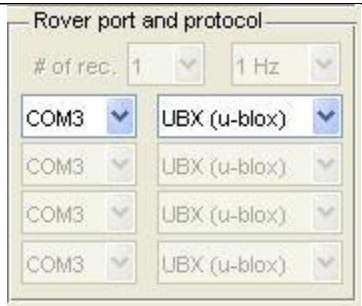
		the observed phase range at time *t+1* with the phase range reconstructed by means of Doppler measurements, i.e. the phase range at time *t* minus the Doppler measurement at time *t*. If the option is not selected, the detection of cycle slips will be performed by the standard method, i.e. by comparing the ambiguities derived from the phase observation minus the Kalman-predicted range with those predicted by the Kalman filter (see Realini and Reguzzoni, 2013) "Apply EGNOS corrections": Download and read EGNOS EMS files for the timespan of interest, check if the first epoch position falls within EMS grids and, if yes, apply EGNOS SBAS corrections during the undifferenced processing. This option is available (selectable) only in undifferenced processing mode. "Direction estimation (stop-go-stop)": Estimate a direction (azimuth) when processing data surveyed by means of the "stop-go-stop" method, which consists of: 1. Keeping the receiver antenna stationary for some time, with the dynamic model set to "Static" 2. Moving the antenna along the direction to be measured, switching the dynamic model to "Constant velocity" 3. Keeping the receiver antenna stationary at the other end of the movement line, setting again the dynamic model to "Static" Note that the dynamic model can be changed during a survey by setting the Dynamic model option to "Variable" in real-time rover monitoring modes.

		The other options are for the graphics utilities and are self-explanatory.
Settings: Kalman filter		These settings are selectable when using Kalman filtering in post-processing mode. Some of these options are also selectable for post-processing with least squares and are listed at the end of this section. "Error standard deviation" settings: Sets the following error standard deviations for Kalman filter processing: • "East", "North", "Up" represent the standard deviation of the dynamic model of the Kalman filter • "Code" and "Phase" represent the standard deviation of Code and Phase observations, respectively • "Initial state" represents the standard deviation of the initial position of the Kalman filter • "DTM(h)" represents the standard deviation of the Digital Terrain Model (DTM) to be used as a pseudo-observation; this option is toggled on/off by clicking on the DTM(h) button. General Kalman filter settings: • Elevation cut-off: 15º is recommended for low-cost receivers. Should not be set lower than 5º. This setting is also selectable for post-processing with least squares. • SNR threshold: observations with signal-to-noise ratios

		less than the threshold will be discarded from processing. The recommended value is 0 dB (i.e., use do not discard observations). Higher value (e.g., 30 dB or more) should only be used for surveys with very degraded observation quality. • Cycle slip threshold: observation discontinuities larger than the threshold will be considered cycle slips. Recommend using 1 cycle for geodetic receivers, static surveys with low-cost receivers and Doppler-based cycle slip detection, and 3 cycles for kinematic surveys with low-cost receivers. The default value is 3 cycles. • Minimum number of satellites: after initialization the Kalman filter prediction and update will be applied when the number of available satellites is equal to or greater than this minimum value. At least 4 satellites are still required for initializing the Kalman filter. "Integer ambiguity resolution" settings: These settings are based on the LAMBDA method (Teunissen 1995; Verhagen 2012) and are only selectable when double difference phase observations are used in post-processing mode. The available settings depend on the LAMBDA method used. The validation of the selected integer is carried out by means of the ratio test threshold. See Verhagen and Li (2012) for more information about these settings. The following LAMBDA methods are available when selecting "Try to resolve ambiguities":

		• LAMBDA v2.0 – ILS, enumeration: only the Ratio test threshold can be set with this method; the default value is 0.5. • LAMBDA v3.0 – ILS, search-and-shrink & enumeration: values for both the "Ratio test threshold" and "Fixed failure rate" can be set with these methods. Select the "auto" check box to have the LAMBDA software choose the ratio test threshold. Select the "default" checkbox to use the LAMBDA software's default value for the fixed failure rate. • LAMBDA 3.0 – Integer rounding & Integer bootstrapping: no settings can be specified for these methods. • LAMBDA 3.0 – Partial ambiguity resolution: values for the "Min. success rate" can be set with this method. Select the "default" checkbox to use LAMBDA's default value. "Dynamic model" settings: In post-processing mode, the following settings are available for the dynamic model used in the Kalman filter: • Const. velocity • Const. acceleration • Static • Variable: when used in post-processing, this selection is used to post-process a dataset surveyed with a variable dynamic model (see the real-time cases described below).

		In real-time navigation mode, the variable dynamic model is used to let the user change the dynamic model on-the-fly. In real-time modes with a rover monitor, the following dynamic models are available: • Constant: uses a constant dynamic model • Variable: manually switches between different dynamic models; this option is used during surveys, to make goGPS record when the receiver was stationary and when it was moving (either at "constant velocity" or "constant acceleration"). In practice, this procedure creates a file that stores the selected dynamic mode for each epoch of survey, which can be loaded during post-processing by selecting the "Variable" dynamic model option. Note: the "stop-go-stop" case explained above is a particular case of survey made by a variable dynamic model. "Ambiguity restart after anomalies" settings: When a new satellite rises or a cycle slip occurs, it is necessary to estimate a new ambiguity. The user can select one of the following methods for this: • Observed code–phase difference: uses difference between code and phase observations • Kalman-predicted code–phase difference: uses difference between Kalman-predicted range and phase observations • Least squares adjustment: uses least squares adjustment of all code and phase observations; this method is

		recommended. The following options are also available when post-processing with least squares: • Elevation cut-off • SNR threshold (only for code double differences) • Integer ambiguity resolution with LAMBDA (only for code and phase double differences) • Phase error standard deviation (only for code and phase double differences)
Settings: Master station		For post-processing mode with double difference observations or real-time navigation mode, the user can specify the position of the master station by either selecting “Read master position from RTCM/RINEX” to get the position from the real-time RTCM stream or the RINEX file header, or by manually entering the position as geocentric Cartesian ECEF (X,Y,Z) coordinates or as geodetic ellipsoidal (lat,lon,h) coordinates.
Settings: Rover port		Defines up to four COM ports where supported low-cost receivers can be simultaneously connected. Supports u-blox LEA-xT, SkyTraq S1315F-RAW, Fastrax IT03 or NVS NV08C-CSM protocols.

Settings: Master server / NTRIP		Configures the NTRIP client to connect to an NTRIP caster defined by the server IP address, port, mountpoint, user and password. The approximate position of the rover receiver can be set either manually (for running a Master monitor just to log RTCM data), or computed and set automatically by goGPS (when running a real-time mode that includes a rover receiver).
Buttons		Button descriptions: “Exit”: closes the goGPS application GUI. “Save settings”: allows the user to save the current GUI configuration settings in .mat files. “Load settings”: allows the user to load previously saved GUI configurations from .mat files. “***go!***”: runs goGPS with the current settings.

6. The INI Editor

The INI file editor is shown in Figure 4. It can be used to define the paths and names of the input files in the INI file. The entire INI file is displayed on the right side of the window for editing.

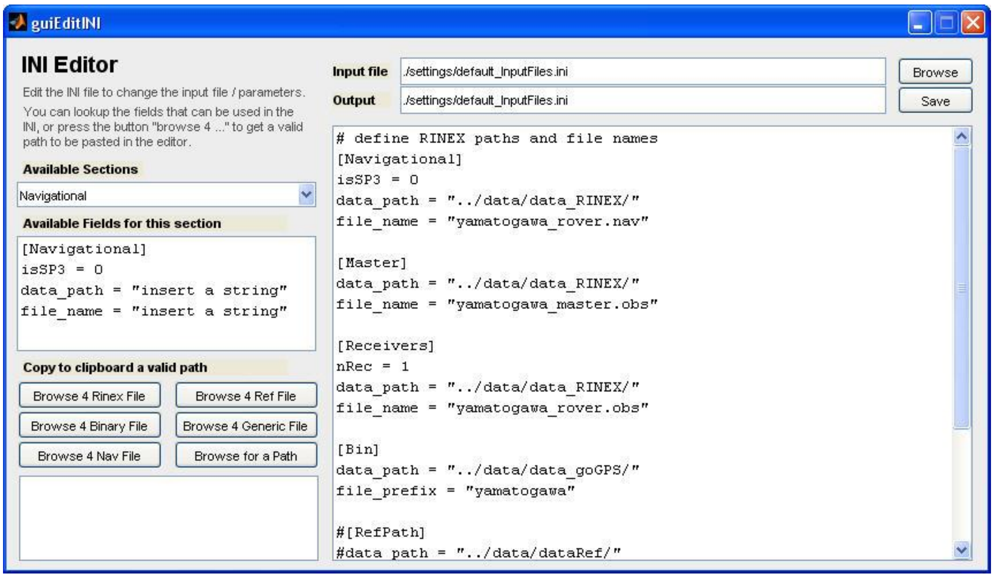


Figure 4: Input files editor

At the top of the INI Editor window, “Input file” and “Output file” define which INI file is loaded as input and which file changes will be saved to, respectively. The Browse button will browse the file system to select an input INI file instead of entering it manually. The Save button will save the INI file currently being edited to the specified output file.

The buttons under “Copy to clipboard a valid path” can be used to select valid paths by browsing the file system instead of entering the path manually, which may introduce errors. When a file is selected, the corresponding path and filename are copied to the clipboard, allowing the user to paste them directly in the INI file editor in the right side of the window. The clipboard contents are shown in the rectangular window in the bottom left corner of the INI Editor window.

There following are the different sections of the INI file in which the different types of files are specified. The generic content of the different sections can be viewed using the “Available Sections” menu.

- Navigation section: Specifies the path and name of the RINEX navigation (.nav) file. The “isSP3” flag is set to zero if broadcast ephemerides are used or set to one when precise ephemerides in SP3 format are used. Note that, at the moment, goGPS can read SP3 files and interpolate satellite positions and clocks, but it cannot actually use the precise orbits. This is due to the fact that it does not apply the satellite phase center offsets yet.

- Master section: Specifies the path and name of the RINEX observation file for the master station.

- Receivers section. Specifies the path(s) and name(s) of the RINEX observation file(s) for the rover receiver(s). In addition, the total number of rover receivers is specified by the “nRec” parameter. This feature was added in order to support planned multi-receiver modes that are still under development. Therefore, “nRec” should always be set to “1” for all the processing modes described in this document.

- Bin section: Specifies the path and name of the goGPS binary file which is formed from the real-time observation and navigation data collected by the user. This file is only used as an alternative to RINEX files.

- RefPath section: Specifies the path and filename defining the “reference path” to be used for the linear constraint positioning option. The RefPath file is created by saving a MATLAB .mat file containing two matrices: one matrix named “ref_path” that defines the path node ECEF coordinates, and one binary matrix named “mat_path” that defines which nodes are linked to which (0: not linked; 1: linked). Example:

```
>> ref_path

ref_path =

   1.0e+06 *

   4.398269203924325   0.703867421376948   4.550191012564839
   4.398275703583665   0.703857251123236   4.550186334656104
   4.398283094011108   0.703865821881468   4.550177921846528
   4.398276618611170   0.703875996818621   4.550182575743193
   4.398270047414793   0.703886240642199   4.550187310996588
   4.398262653590095   0.703877655928451   4.550195729212146

>> mat_path

mat_path =

     0     1     0     0     0     1
     1     0     1     0     0     0
     0     1     0     1     0     0
     0     0     1     0     1     0
     0     0     0     1     0     1
     1     0     0     0     1     0
```

- DTM section: Specifies the folder containing the files for the DTM model in post-processing mode. Note that goGPS requires DTM ASCII grid files to be divided into tiles in order to be loaded and used during processing. The tiles can be generated automatically using the function “dtm_bulk_load” in the MATLAB command window. This function requires the header of the files to be separated from the gridded values: each file header should be stored in a *filename.hdr* file, and each corresponding grid should be stored in a *filename.dtm* file. All the files composing the required DTM should be stored in a folder, which is passed as an argument to "dtm_bulk_load". For example, given the ASCII grid file "*test.grd*" with the following contents:

```
ncols         4
nrows         6
xllcorner     0.0
yllcorner     0.0
cellsize      50.0
NODATA_value  -9999
-9999 -9999 5 2
-9999 20 100 36
3 8 35 10
32 42 50 6
88 75 27 9
13 5 1 -9999
```

The header file "*test.hdr*" will contain:

```
ncols         4
nrows         6
xllcorner     0.0
yllcorner     0.0
cellsize      50.0
NODATA_value  -9999
```

And the dtm file "*test.dtm*" will contain:

```
-9999 -9999 5 2
-9999 20 100 36
3 8 35 10
32 42 50 6
88 75 27 9
13 5 1 -9999
```

- Variometric section: Specifies the time step for the variometric approach for velocity estimation option in post-processing mode.
- PCO_PCV_file section: Specifies the path and name of the antenna phase center offset (PCO) / phase center variation (PCV) file, in ANTEX (.ATX) format. A standard file (I08.ATX) is provided in the "data/stations" folder.

7. Examples

In this section, examples of different processing modes are described to clarify the operation of goGPS and illustrate its use. For each example an INI file is provided containing the necessary parameters to replicate the example. In general, the default options of the program have been used but the user can change them according to his particular use.

7.1 Post-processing: Least-squares, code undifferenced, with and without EGNOS

The main goal of this example is to evaluate the impact of using augmentation systems, like EGNOS, in post-processing mode and using least-squares adjustment with code observations.

Open the MATLAB application and set the current folder to the “goGPS” subfolder of the installed goGPS_v0.4.3 application. Now type in the MATLAB command "goGPS" and press [Return].

To select the corresponding setting options and input files, click on the button “Load settings” and select “egnos”. Change the output prefix “egnos” to “example_EGNOS” (see Figure 5) and click on the button “go!”. The MATLAB command line output expected is shown in Figure 6.

When the program execution has finished, the following output files are created in the folder “../data/out”:

example_EGNOS_cov_ENU.txt	Position covariance matrices per epoch in ENU
example_EGNOS_cov_XYZ.txt	Position covariance matrices per epoch in XYZ
example_EGNOS_NMEA.txt	NMEA output strings for each epoch
example_EGNOS_position.mat	Complete position information for each epoch
example_EGNOS_position.txt	Complete position information for each epoch
example_EGNOS_report.pdf	Single page report of position solution
example_EGNOS.kml	KML file for displaying positions in Google Earth
example_EGNOS_conf_000.bin	Satellite configuration (for internal use)
example_EGNOS_dop_000.bin	Dilution of precision (for internal use)
example_EGNOS_kal_000.bin	Kalman filter matrices (for internal use)
example_EGNOS_sat_000.bin	Satellite topocentric coordinates (for internal use)

To compare the GPS positions with EGNOS corrections and the GPS positions without EGNOS corrections, type in the MATLAB command "goGPS" and press [Return]. The program loads the last configuration, so to process without EGNOS corrections change the output prefix to “example_GPS” and unmark the “Apply EGNOS corrections” option (see Figure 7). Then click on the button “go!”.

Finally to generate the plots in Figure 8 comparing the two solutions, copy the graph_EGNOS.m file from the examples folder into the goGPS folder. Then type the

MATLAB command “graph_EGNOS.m” and press [Return]. The plots in Figure 8 should appear in a new figure window.

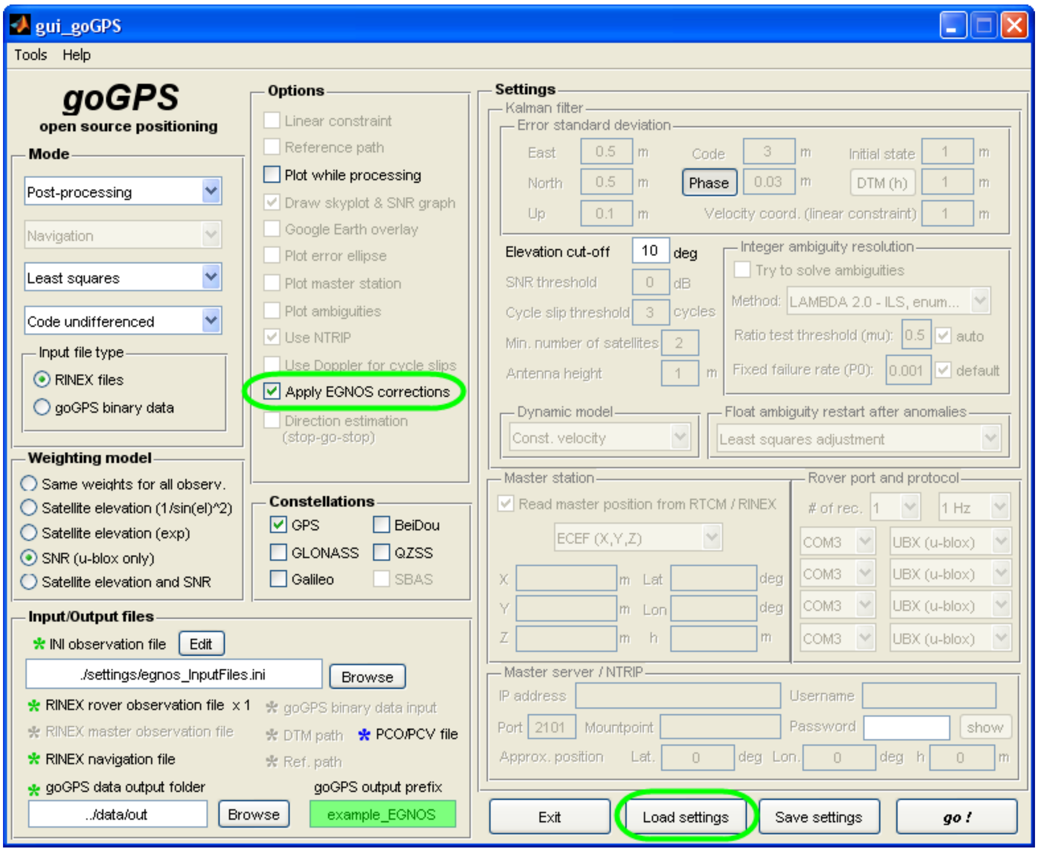
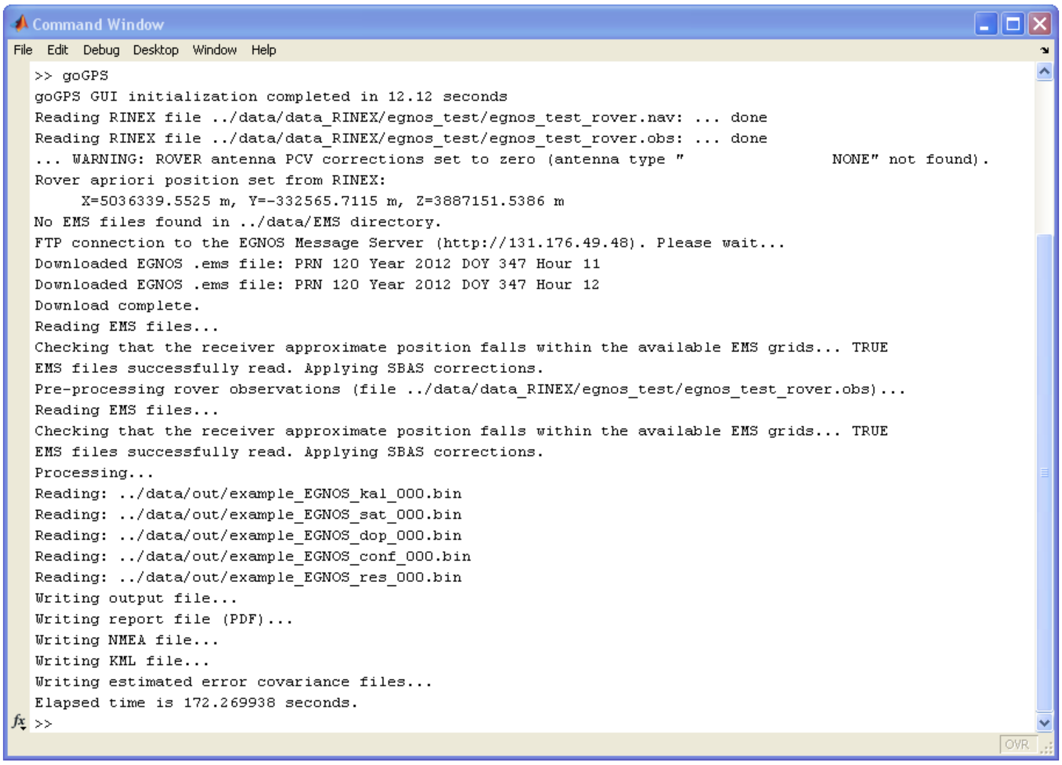


Figure 5




```
>> goGPS
goGPS GUI initialization completed in 12.12 seconds
Reading RINEX file ../data/data_RINEX/egnos_test/egnos_test_rover.nav: ... done
Reading RINEX file ../data/data_RINEX/egnos_test/egnos_test_rover.obs: ... done
... WARNING: ROVER antenna PCV corrections set to zero (antenna type "                NONE" not found).
Rover apriori position set from RINEX:
     X=5036339.5525 m, Y=-332565.7115 m, Z=3887151.5386 m
No EMS files found in ../data/EMS directory.
FTP connection to the EGNOS Message Server (http://131.176.49.48). Please wait...
Downloaded EGNOS .ems file: PRN 120 Year 2012 DOY 347 Hour 11
Downloaded EGNOS .ems file: PRN 120 Year 2012 DOY 347 Hour 12
Download complete.
Reading EMS files...
Checking that the receiver approximate position falls within the available EMS grids... TRUE
EMS files successfully read. Applying SBAS corrections.
Pre-processing rover observations (file ../data/data_RINEX/egnos_test/egnos_test_rover.obs)...
Reading EMS files...
Checking that the receiver approximate position falls within the available EMS grids... TRUE
EMS files successfully read. Applying SBAS corrections.
Processing...
Reading: ../data/out/example_EGNOS_kal_000.bin
Reading: ../data/out/example_EGNOS_sat_000.bin
Reading: ../data/out/example_EGNOS_dop_000.bin
Reading: ../data/out/example_EGNOS_conf_000.bin
Reading: ../data/out/example_EGNOS_res_000.bin
Writing output file...
Writing report file (PDF)...
Writing NMEA file...
Writing KML file...
Writing estimated error covariance files...
Elapsed time is 172.269938 seconds.
>>
```

Figure 6

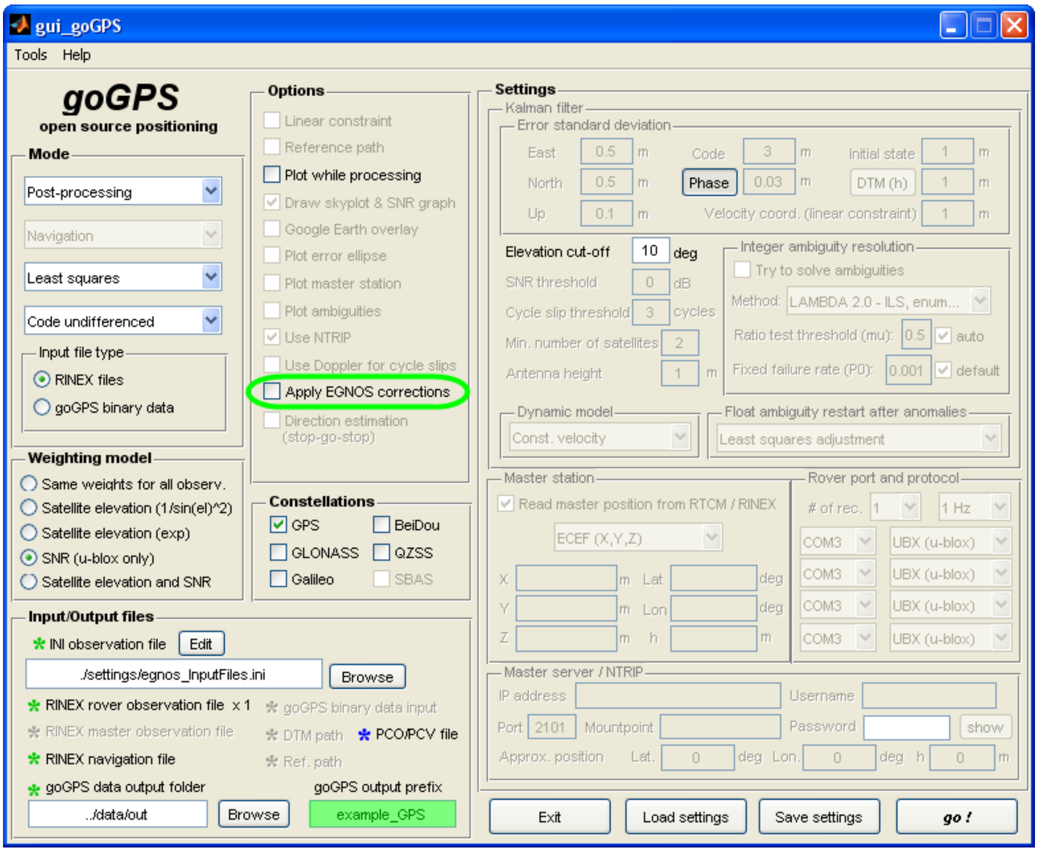


Figure 7

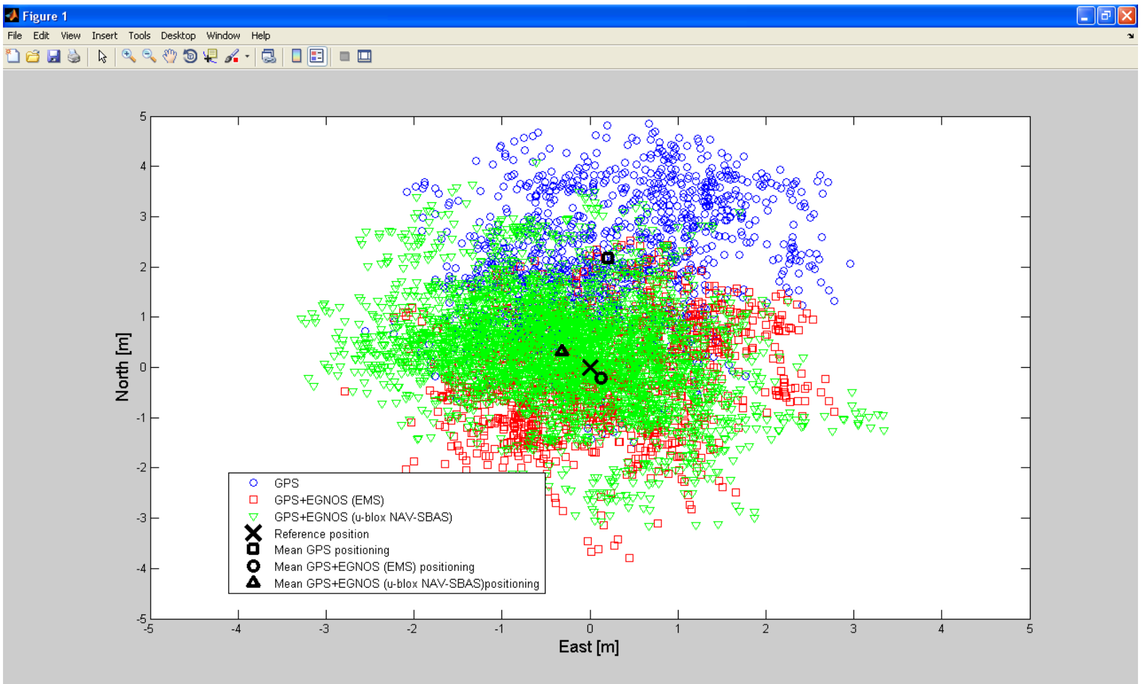


Figure 8

7.2 Post-processing: Least-squares, code undifferenced, multi-constellation

The objective of this example is to compare the position estimated for a point with its "true position" using different constellations in post-processing. The estimated position is computed by least-squares adjustment using code observations.

GPS

Open the MATLAB application and set the current folder to the "goGPS" subfolder of the installed goGPS_v0.4.3 application. Now type in the MATLAB command "goGPS" and press [Return].

To select the corresponding setting options, click on the button "Load settings" and select the "multiGNSS" file. Once the settings are loaded, change the output prefix "multiGNSS" to "GPS" and make sure that only the GPS constellation is selected (see Figure 9). Then click on the button "go!".

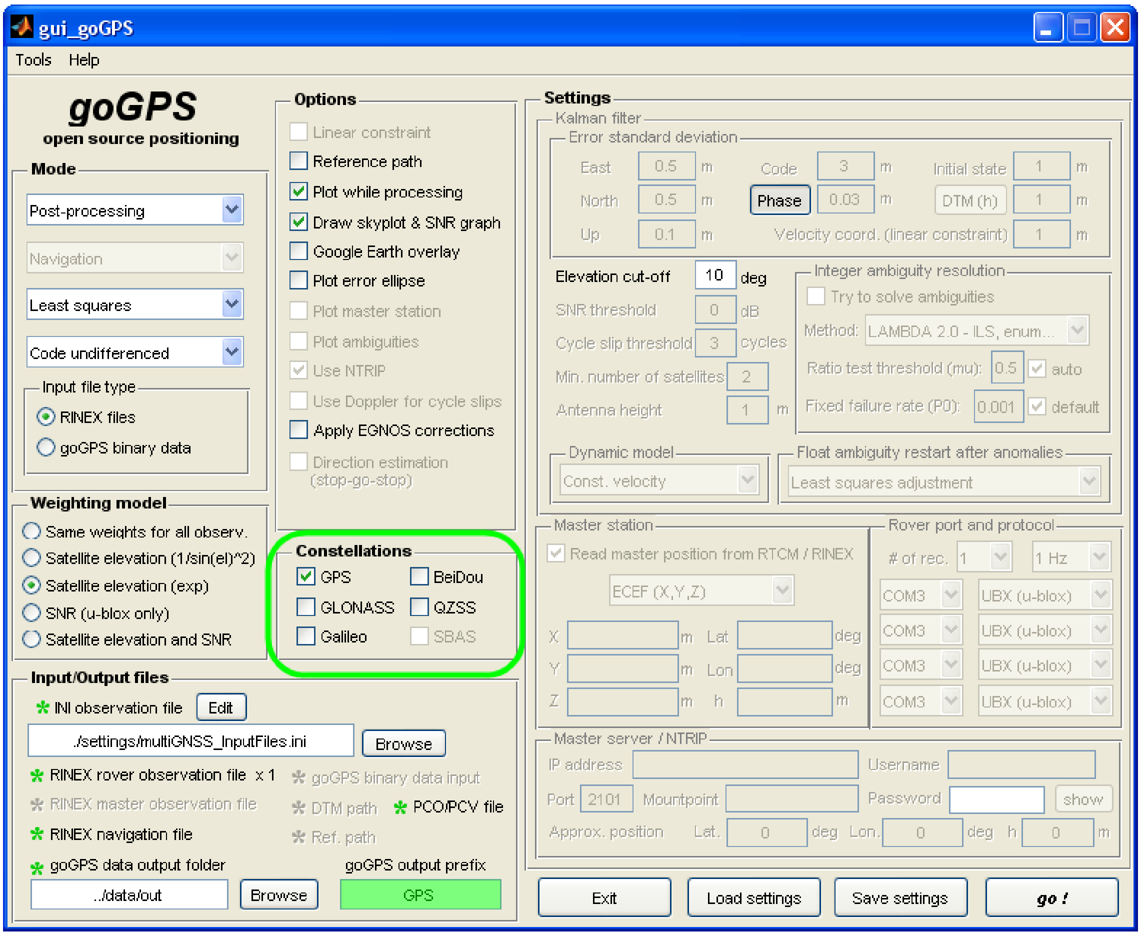


Figure 9

GLONASS

When the program is finished processing GPS, type in the MATLAB command "goGPS" and press [Return]. The program loads the last configuration, so to process GLONASS data change the output prefix "GPS" to "GLO", uncheck the GPS option in the constellation box, and check the GLONASS option. Then click on the "go!" button (see Figure 10).

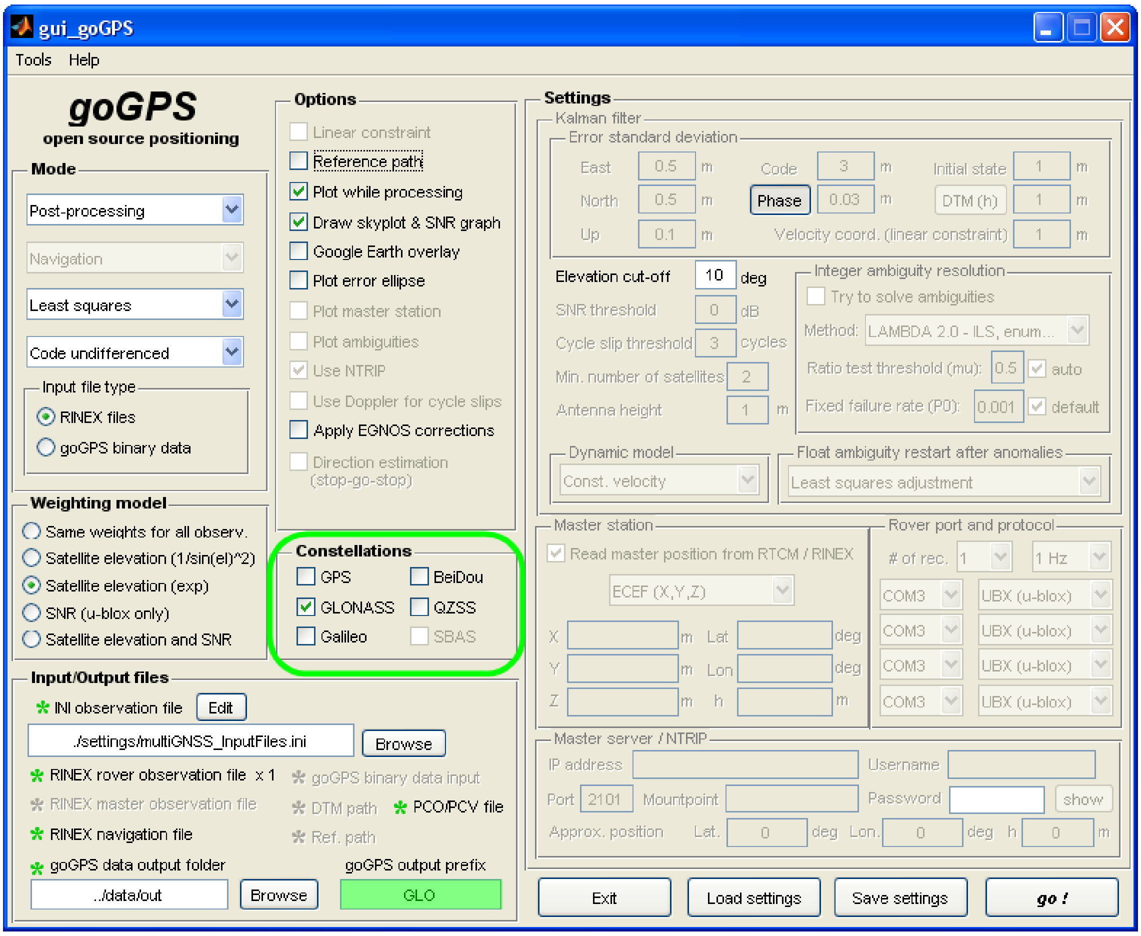


Figure 10

BeiDou

When the program is finished processing GLONASS, type in the Matlab command "goGPS" and press [Return]. To process BeiDou data, change the output prefix “GLO” to “BDS”, uncheck the GLONASS option in the constellation box and check the BeiDou option. Then click the “go!” button (see Figure 11).

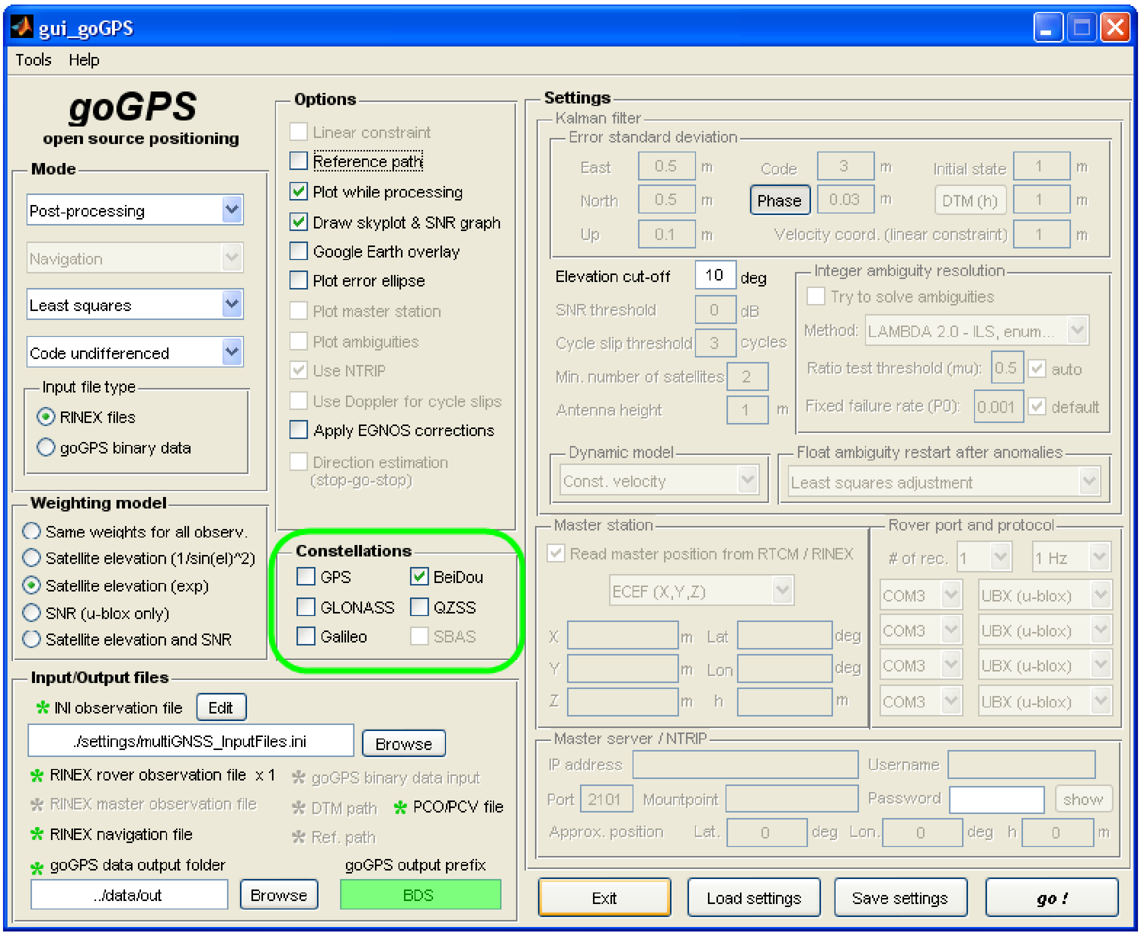


Figure 11

GPS + GLONASS

When the program is finished processing BeiDou, type in the Matlab command "goGPS" and press [Return]. To process both GPS and GLONASS data, change the output prefix “BDS” to “GPSGLO”, uncheck the BeiDou option in the constellation box and check the GPS and GLONASS options. Then click the “go!” button (see Figure 12).

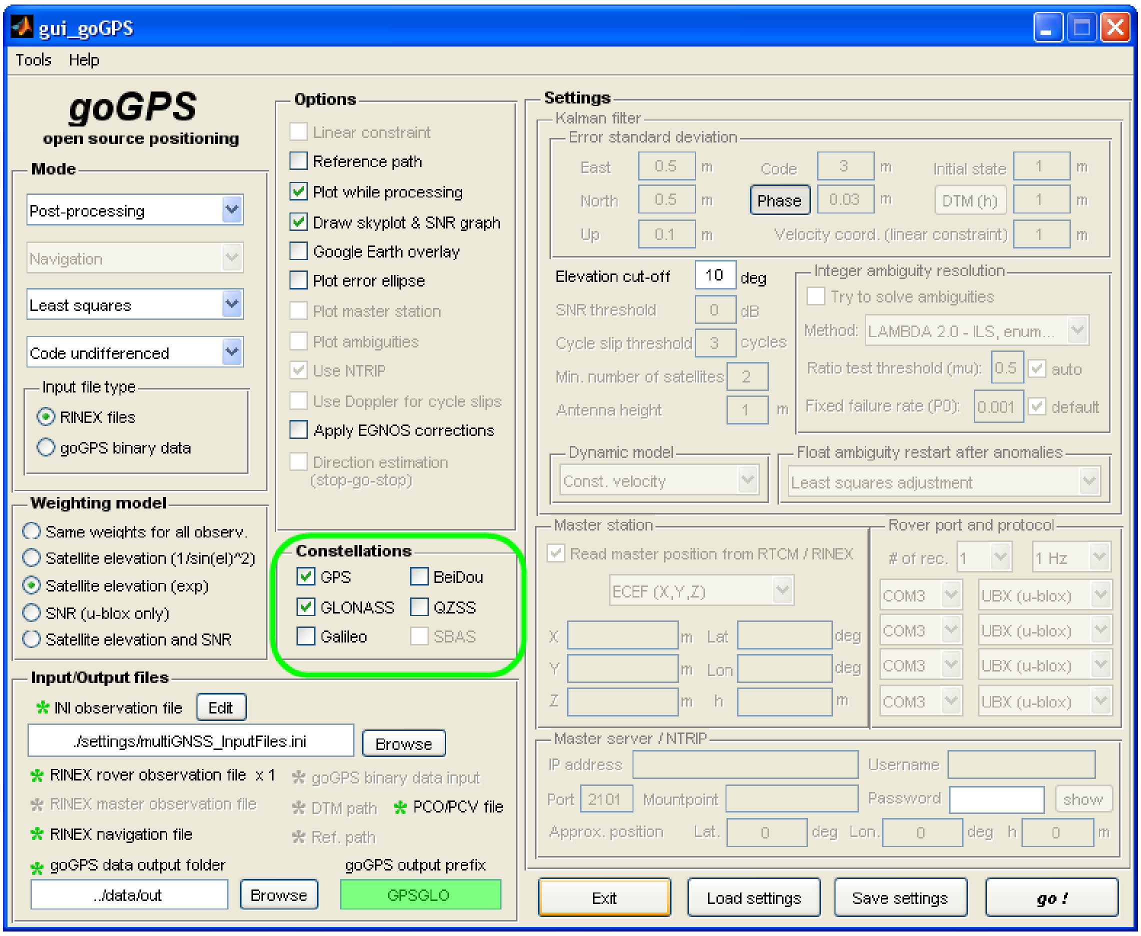


Figure 12

GPS + BeiDou

When the program is finished processing BeiDou, type in the Matlab command "goGPS" and press [Return]. To process both GPS and BeiDou data, change the output prefix “GPSGLO” to “GPSBDS”, uncheck the GLONASS option in the constellation box and check the GPS and BeiDou options. Then click the “go!” button (see Figure 13).

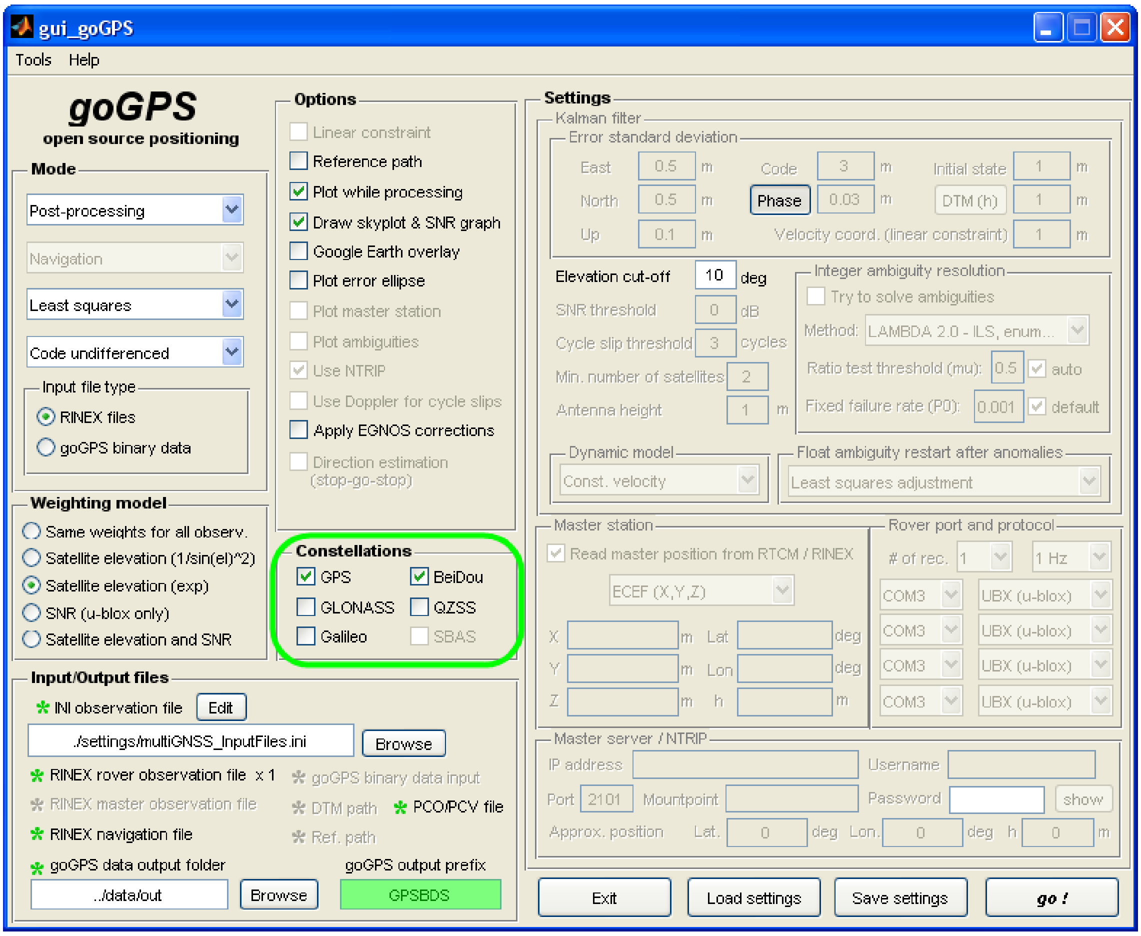


Figure 13

GPS + Galileo

When the program is finished processing GPS and BeiDou, type in the Matlab command "goGPS" and press [Return]. To process both GPS and Galileo data, change the output prefix “GPSBDS” to “GPSGAL”, uncheck the BeiDou option in the constellation box and check the GPS and Galileo options. Then click the “go!” button (see Figure 14).

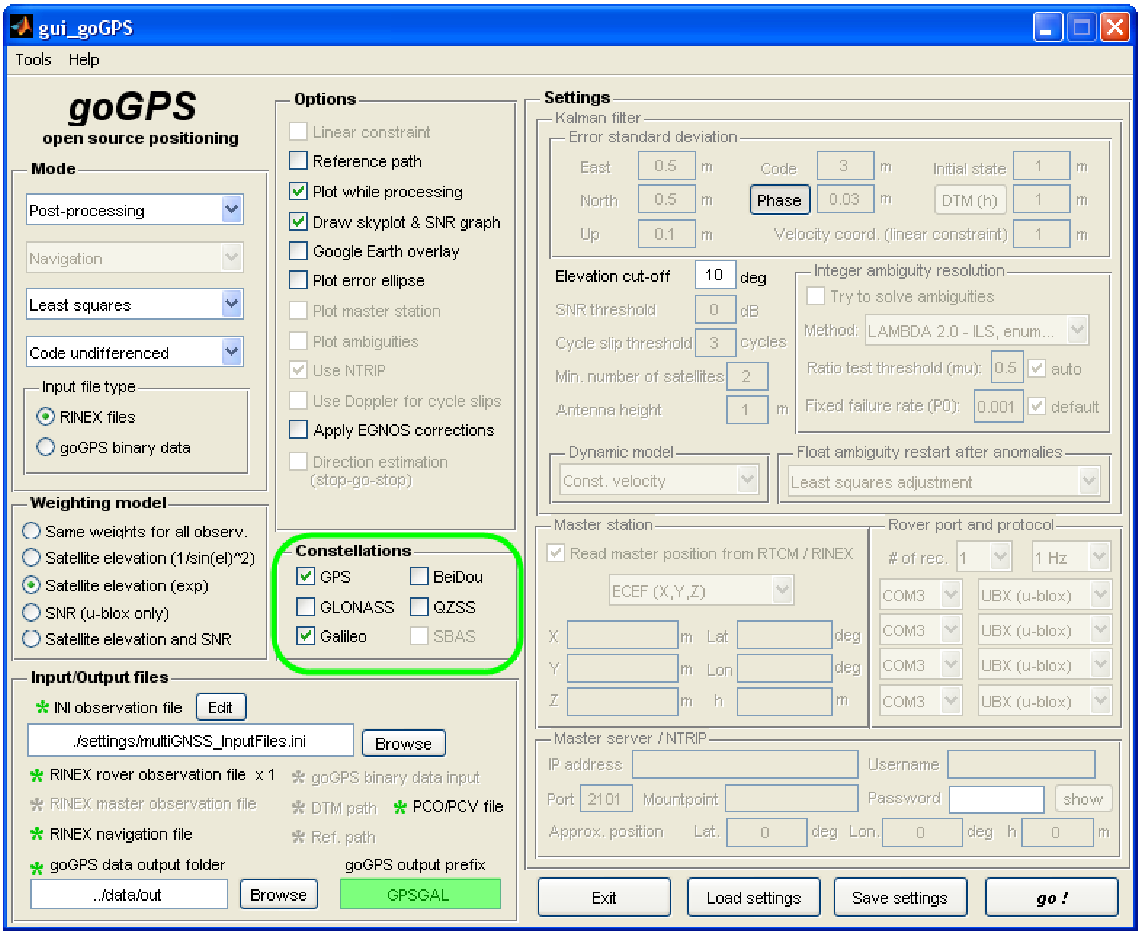


Figure 14

GPS + QZSS

When the program is finished processing GPS and Galileo, type in the Matlab command "goGPS" and press [Return]. To process both GPS and QZSS, change the output prefix “GPSGAL” to “GPSQZS”, uncheck the Galileo option in the constellation box and check the GPS and QZSS options. Then click the “go!” button (see Figure 15).

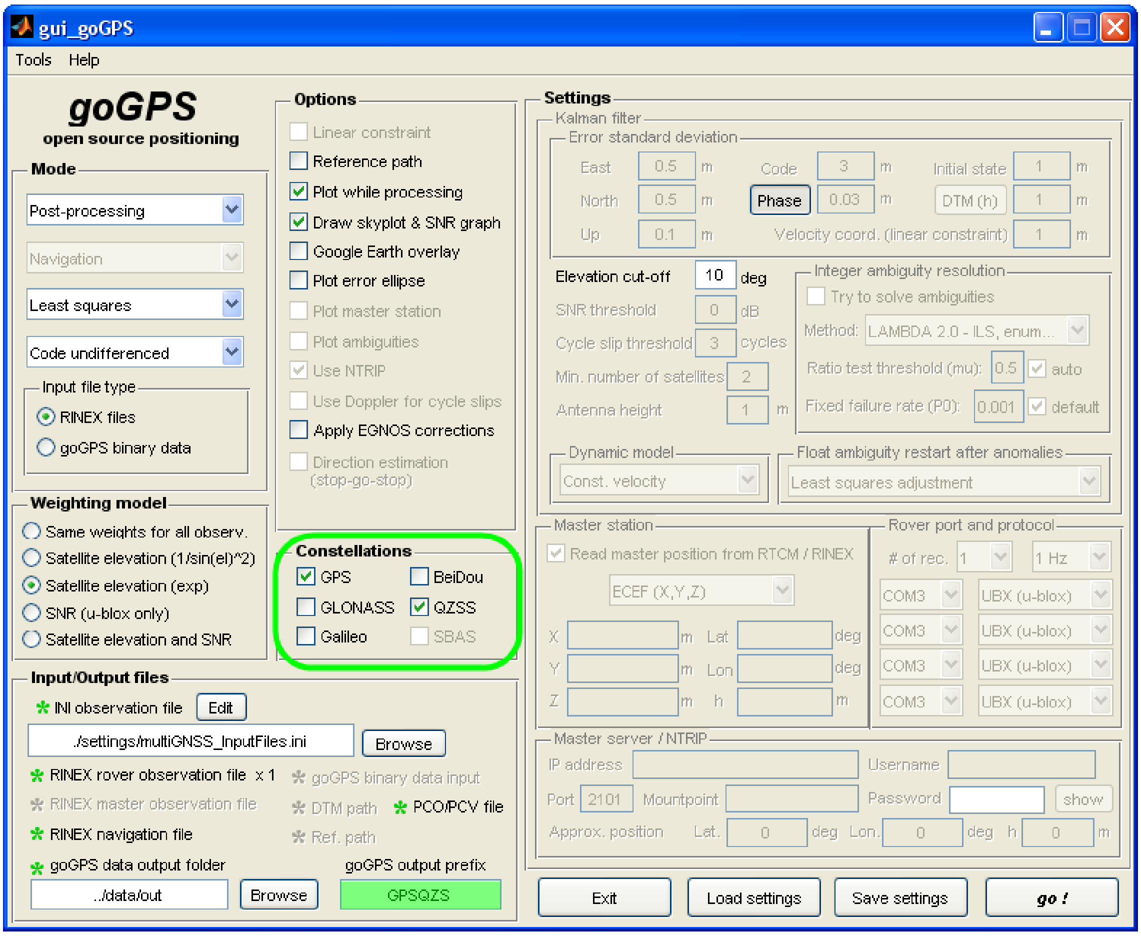


Figure 15

GLONASS + BeiDou

And finally, when the program is finished processing GPS and QZSS, type in the Matlab command "goGPS" and press [Return]. To process both GLONASS and QZSS, change the output prefix “GPSQZS” to “GLOBDS”, uncheck the GPS and QZSS options in the constellation box and check the GLONASS and BeiDou options. Then click the “go!” button (see Figure 16).

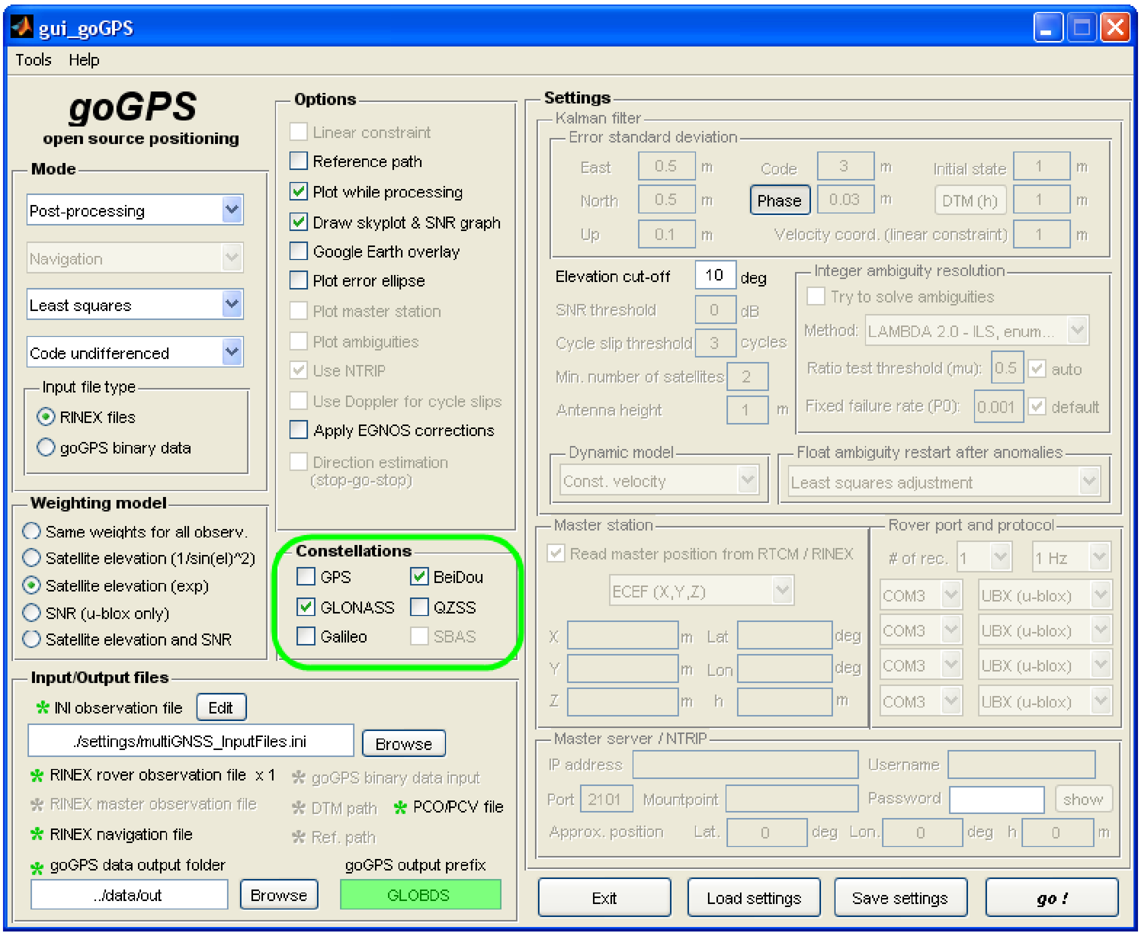


Figure 16

There should now be eight solutions with which to generate Figure 17. To do that, copy the graph_multiGNSS.m file from the “examples” folder to the goGPS folder. Type in the MATLAB command “graph_multiGNSS” and press [Return]. The plots in Figure 17 should appear in a new figure window.

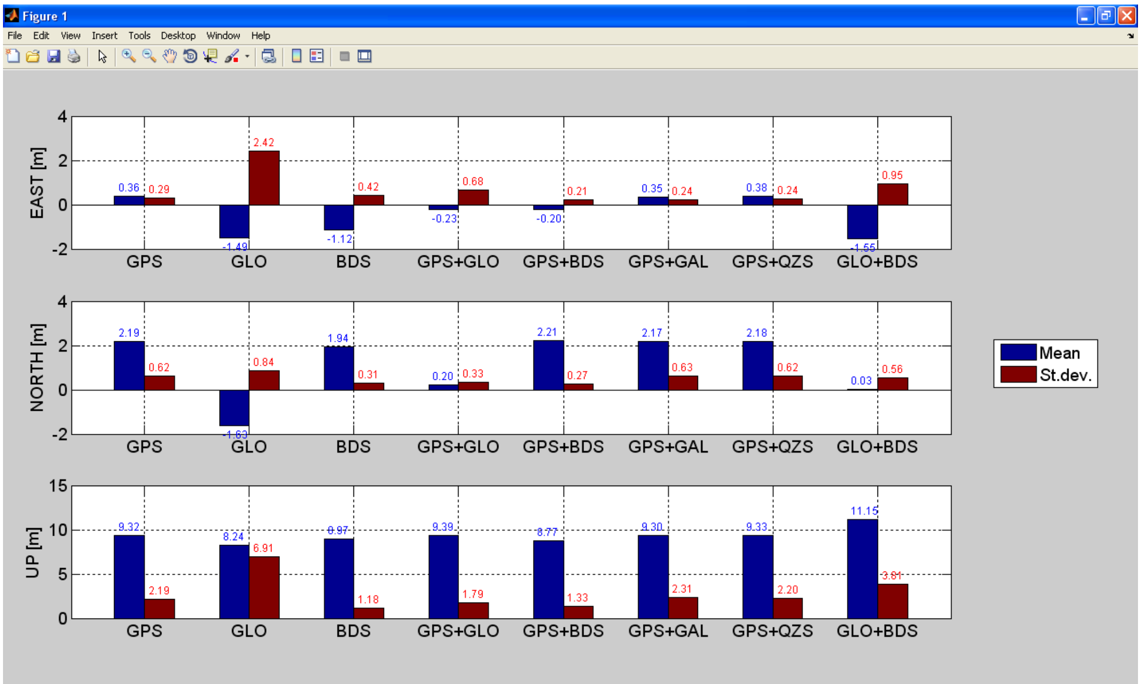


Figure 17

7.3 Post-processing: Kalman filter, code & phase double difference, integer ambiguity resolution by LAMBDA method

The objective of this example is to show a comparison between the coordinates estimated in post-processing of phase observations with float ambiguities and integer ambiguities resolved using the LAMBDA method.

Open the MATLAB application and set the current folder to the “goGPS” subfolder of the installed goGPS_v0.4.3 application. Type in the MATLAB command "goGPS" and press [Return]. The window in Figure 18 should appear.

To select the corresponding setting options click on the button “Load settings” and select the “lambda” file. Then click on the button “go!” to process with LAMBDA integer ambiguity resolution.

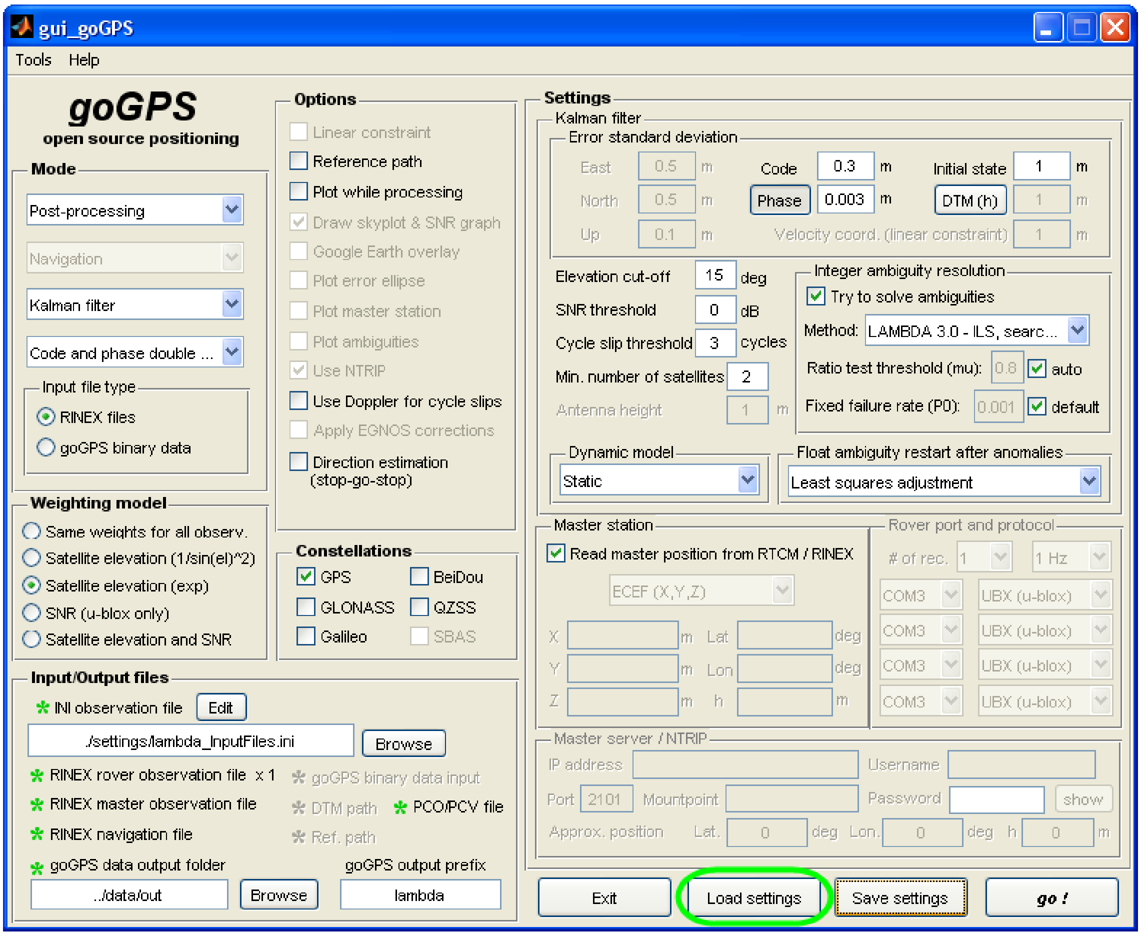


Figure 18

If LAMBDA version 3 is not installed, and error message will explain how to obtain it and install it. The GUI settings will then be automatically set to use LAMBDA 2.0 - ILS. Click the button “go!” to begin processing again with LAMBDA v2.0.

When the program is finished, multiple output files are created in the folder “../data/out”. To compare the float positioning results with the positioning results obtained with integer ambiguities estimated by LAMBDA method with positioning results using float ambiguities, run goGPS again with two parameters changed. Type in the command "goGPS" and press [Return]. The program loads the last configuration, so you only need to change the output prefix “lambda” to “float” and uncheck the option “Try to solve ambiguities” (see Figure 19). Then click on the button “go!”.

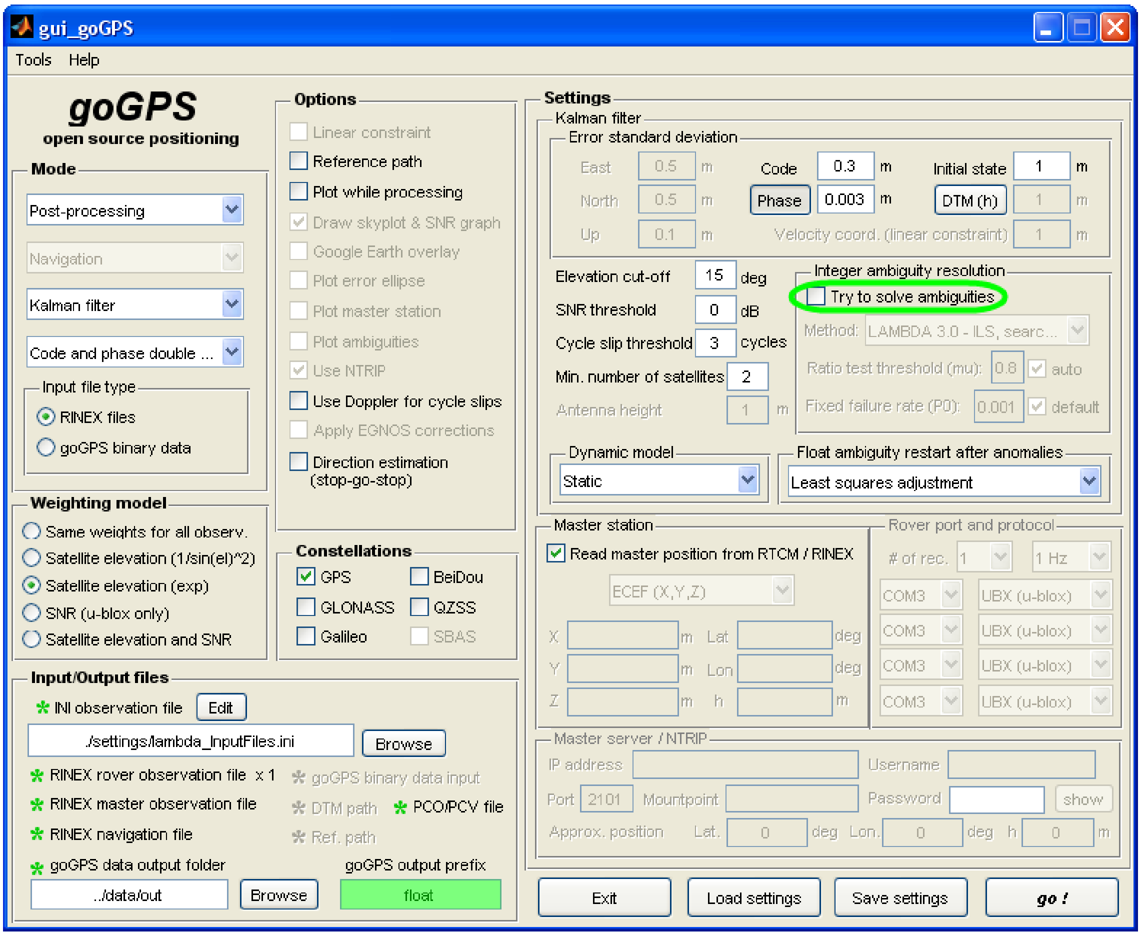


Figure 19

Finally to compare the two solutions, copy "graph_LAMBDA.m" from the examples folder to the folder "goGPS". Then type in the MATLAB command "graph_ LAMBDA" and press the [Return] button. The corresponding local coordinate convergence is shown in Figure 20. Since a Kalman filter with a static model for the receiver dynamics is used, the final estimated receiver position is the one for the last epoch of the plotted time series.

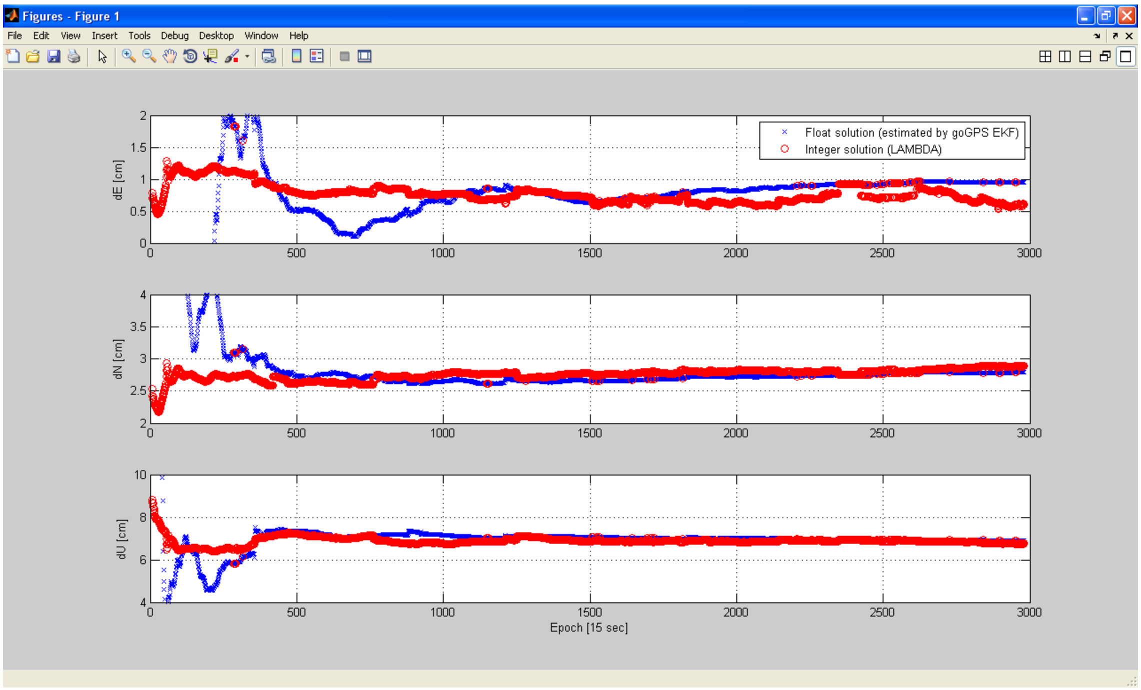

Figures - Figure 1
File Edit View Insert Tools Debug Desktop Window Help
Float solution (estimated by goGPS EKF)
Integer solution (LAMBDA)
dE [cm]
dN [cm]
dU [cm]
Epoch [15 sec]

Figure 20

7.4 Real-Time: Navigation

goGPS allows real time data collection. To do that, open the MATLAB application and set the current folder to the “goGPS” subfolder of the installed goGPS_v0.4.3 application. Then type in the command "goGPS" and press [Return]. The window in Figure 21 should appear.

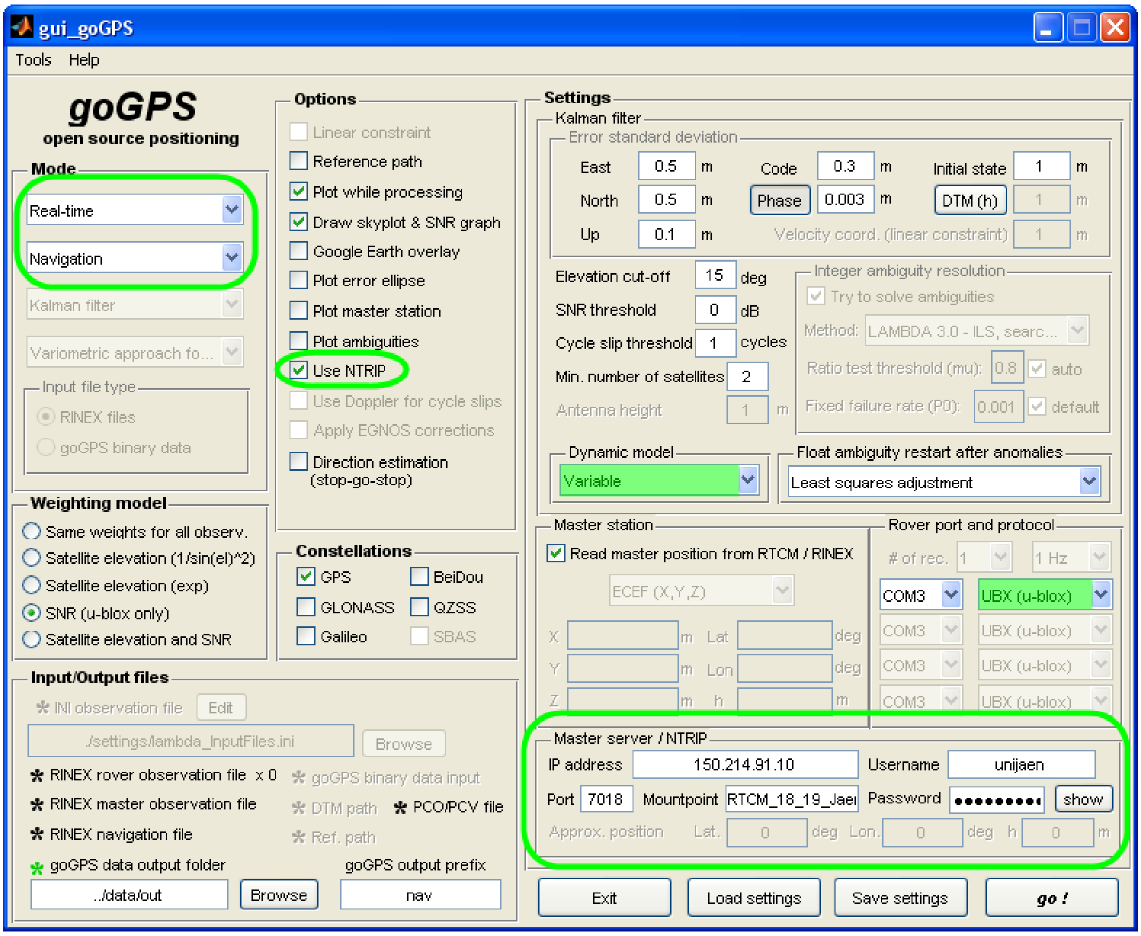


Figure 21

Navigation mode activates the settings area of the GUI and the Master Server / NTRIP options for differential positioning using the NTRIP protocol and Rover port. The user can connect a low cost receiver to the COM port and define appropriate Master server parameters if differential positioning is used. This mode allows the user to display a graph of the positions obtained at each epoch, see Figure 22.

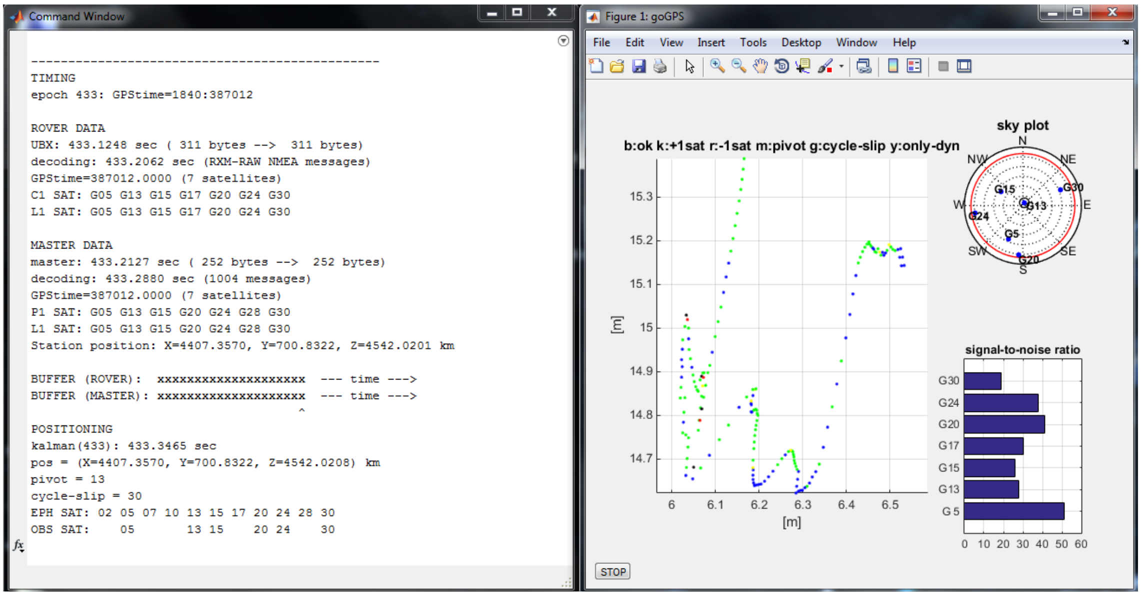


Figure 22

References

Colosimo G, Crespi M, Mazzoni A (2011) Real-time GPS seismology with a stand-alone receiver: A preliminary feasibility demonstration. Journal of Geophysical Research: Solid Earth (1978‑2012), 116(B11).

Herrera A, Suhandri H, Realini E, Reguzzoni M, de Lacy MC (2015) goGPS: open-source MATLAB software. GPS Solutions, doi: 10.1007/s10291-015-0469-x.

Realini E, Reguzzoni M (2013) goGPS: open source software for enhancing the accuracy of low-cost receivers by single-frequency relative kinematic positioning. Measurement Science and Technology 24(11):115,010.

Realini E (2009) Ph. D. thesis: goGPS free and constrained relative kinematic positioning with low cost receivers.

Teunissen P (1995) The least-squares ambiguity decorrelation adjustment: a method for fast GPS integer ambiguity estimation. Journal of Geodesy 70(1-2):65–82.

Verhagen S, Li B. (2012) LAMBDA software package: Matlab implementation, version 3.0. Delft University of Technology and Curtin University, Perth, Australia.