goGPS: open-source MATLAB software

Article *in* GPS Solutions · June 2015

DOI: 10.1007/s10291-015-0469-x

Antonio M. Herrera Olmo
Universidad de Jaén

Hendy Suhandri
Universität Stuttgart

Eugenio Realini
Geomatics Research & Development s.r.l.

Mirko Reguzzoni
Politecnico di Milano

GPS Solut
DOI 10.1007/s10291-015-0469-x

GPS TOOLBOX

goGPS: open-source MATLAB software

Antonio M. Herrera[1] · Hendy F. Suhandri[2] · Eugenio Realini[3] · Mirko Reguzzoni[4] · M. Clara de Lacy[1,5]

Received: 21 January 2015 / Accepted: 2 June 2015

Abstract goGPS is a positioning software application designed to process single-frequency code and phase observations for absolute or relative positioning. Published under a free and open-source license, goGPS can process data collected by any receiver, but focuses on the treatment of observations by low-cost receivers. goGPS algorithms can produce epoch-by-epoch solutions by least squares adjustment, or multi-epoch solutions by Kalman filtering, which can be applied to either positions or observations. It is possible to aid the positioning by introducing additional constraints, either on the 3D trajectory such as a railway, or on a surface, e.g., a digital terrain model. goGPS is being developed by a collaboration of different research groups, and it can be downloaded from http://www.gogps-project.org. The version used in this manuscript can be also downloaded from the GPS Toolbox Web site http://www.ngs.noaa.gov/gps-toolbox. This software is continues to evolve, improving its functionalities according to the updates introduced by the collaborators. We describe the main modules of goGPS along with some examples to show the user how the software works.

Keywords GNSS · Open-source software · Positioning · Navigation

The GPS Toolbox is a column dedicated to highlighting algorithms and source code utilized by GPS engineers and scientists. If you have an interesting program or software package you would like to share with our readers, please pass it along; e-mail it to us at gpstoolbox@ngs.noaa.gov. To comment on any of the source code discussed here, or to download source code, visit our website at http://www.ngs.noaa.gov/gps-toolbox. This column is edited by Stephen Hilla, National Geodetic Survey, NOAA, Silver Spring, Maryland, and Mike Craymer, Geodetic Survey Division, Natural Resources Canada, Ottawa, Ontario, Canada.

✉ Antonio M. Herrera
aherreraolmo@gmail.com

Hendy F. Suhandri
suhandri@nav.uni-stuttgart.de

Eugenio Realini
eugenio.realini@gmail.com

Mirko Reguzzoni
mirko.reguzzoni@polimi.it

M. Clara de Lacy
mclacy@ujaen.es

1 Departamento de Ingeniería Cartográfica, Geodesia y Fotogrametría, Escuela Politécnica Superior, Universidad de Jaén, Jaén, Spain

2 Institut für Navigation, Universität Stuttgart, Stuttgart, Germany

3 Geomatics Research and Development (GReD) srl, Milan, Italy

4 DICA, Politecnico di Milano, Milan, Italy

5 Centro de Estudios Avanzados de Ciencias de la Tierra (CEACTierra), Jaén, Spain

Introduction

The development of goGPS started in 2007 at Politecnico di Milano, Como campus (Italy). The first version of goGPS MATLAB was published as free and open-source software in 2009, under a GPLv3 license; the software has evolved steadily through the years, improving stability and performance. The development of an alternative version of goGPS, written in Java and published under a LGPLv3 license, was started in 2010 with the aim of providing a positioning library in a coding language more suitable for the implementation of positioning Web services (Realini et al. 2012). It is important to note goGPS does not require MATLAB special supporting toolboxes. The user only requires the Instrumental Control Supporting Toolbox when the connection of a GPS device to a COM port is desired. In this paper, we focus on the MATLAB version, from here on simply referred to as "goGPS".

goGPS processes single-frequency (L1) code and carrier phase observations either by epoch-by-epoch least squares adjustment (LSA) or by an extended Kalman filter (EKF), applied to either undifferenced or double-differenced observations, to produce solutions based on multiple epochs. goGPS can apply different observation-weighting strategies: based on satellite elevation or on tailored weight functions that exploit the known signal-to-noise ratio characteristics of low-cost receivers. The modular design of the goGPS EKF allows for the seamless addition of optional external data sources, e.g., pseudo-observations interpolated from digital terrain models (Realini and Reguzzoni 2013).

The general way of using a Kalman filter for GPS-based navigation is to apply it to epoch-by-epoch positions. These are typically estimated by LSA on code and phase observations: the filter acts a posteriori smoothing the resulting trajectory (Hofmann-Wellenhof et al. 2008). When this method is applied to low-cost devices, the resulting positioning is as accurate as that obtained by applying absolute positioning on phase-smoothed code measurements, i.e., typically with an error of few meters (Leick et al. 2015). By this approach, the effects of atmospheric delays and clock errors cannot be efficiently mitigated; this kind of receiver, in fact, is not designed for relative positioning, and the resulting trajectories are generally affected by large biases. The Kalman filter implemented in goGPS works in this way when the filter is used with code-only observations, either in stand-alone or in relative mode. When code and phase observations are used in stand-alone and double-difference modes, the goGPS EKF is applied on double-difference observations with respect to a reference station. The possibility of directly processing GPS observations allows us to remove most of the biases associated with absolute positioning, obtaining accuracies of less than 1 m.

goGPS acquires input observations either by reading standard RINEX files, from either low-cost or geodetic receivers, or by decoding receiver-specific binary formats. The decoding functions support the following low-cost modules: u-blox LEA-4T/5T/6T, NVS NV08C-CSM, SkyTraq S1315F-RAW, Fastrax IT03. For real-time functioning, the software can acquire reference station observations by decoding RTCM 3.1 data streams from an NTRIP caster; these observations are synchronized on the fly with those decoded from multiple low-cost receivers (from one to four) connected to USB (COM) ports. Changes in the satellite configurations, i.e., satellite additions/losses or new reference (pivot) satellite used for the double differences, and cycle slips are managed for all EKF modes. As of now, only the MATLAB version provides a graphical user interface.

goGPS mainly targets single-frequency, low-cost devices, but its code and phase modules are designed to work either in single-frequency or in dual-frequency mode. Therefore, it can be used also with dual-frequency receivers, even if it is not yet optimized to best exploit both frequencies, e.g., by computing useful data combinations (Teunissen and Kleusberg 1998).

We describe the main modules of goGPS, included in the official releases of the software since version 0.4.2. In particular, the following section deals with a summary of the theory implemented in the program: stand-alone positioning with LSA and relative positioning with EKF. Each topic is briefly summarized in a separate subsection. After that, tests and results are discussed.

Main functionalities of goGPS

This section summarizes the theory implemented in goGPS in the different positioning modes: stand-alone positioning with LSA and relative positioning with EKF, both in post-processing and real time. In the manual of the program, the parameters to set in both cases are shown.

Stand-alone positioning with LSA

This module uses pseudorange observations. The code observation equation at the epoch t is defined as:

$$P_{\mathrm{r}}^{\mathrm{s}}(t) = \rho_{\mathrm{r}}^{\mathrm{s}}(t) + c(\mathrm{d}t_{\mathrm{r}}(t) - \mathrm{d}t^{\mathrm{s}}(t)) + I_{\mathrm{r}}^{\mathrm{s}}(t) + T_{\mathrm{r}}^{\mathrm{s}}(t) + v_{\mathrm{r}}^{\mathrm{s}}(t) \tag{1}$$

where c is the speed of the light in vacuum, $\mathrm{d}t_{\mathrm{r}}$ and $\mathrm{d}t^{\mathrm{s}}$ are the receiver and satellite clock offsets, $T_{\mathrm{r}}^{\mathrm{s}}$ is the tropospheric delay and $I_{\mathrm{r}}^{\mathrm{s}}$ is the ionospheric delay, $v_{\mathrm{r}}^{\mathrm{s}}$ is code observation noise and $\rho_{\mathrm{r}}^{\mathrm{s}}$ is the geometric distance between satellite and receiver positions in Cartesian coordinates.

The atmospheric delays are removed by standard models, in this case the Klobuchar ionospheric model (Klobuchar 1987) and the Saastamoinen tropospheric model (Saastamoinen 1973). Since the satellite position $(X^{\mathrm{s}}, Y^{\mathrm{s}}, Z^{\mathrm{s}})$ is assumed to be known from the ephemerides, the satellite clock offset $\mathrm{d}t^{\mathrm{s}}$ is obtained from the navigation message, the atmospheric delays $I_{\mathrm{r}}^{\mathrm{s}}$ and $T_{\mathrm{r}}^{\mathrm{s}}$ are estimated from standard models, there are four unknowns: the receiver position coordinates $X_{\mathrm{r}}, Y_{\mathrm{r}}, Z_{\mathrm{r}}$ and the receiver clock offset dt_r. At each epoch of measurement, point positioning can thus be performed by having at least four observations, which means receiving the signal from at least four satellites. Since it is possible to estimate the receiver position by using single epoch observations, point positioning by code observations can be effectively used in kinematic positioning for navigation purposes or using LSA to estimate the unknowns during the observation session in post-processing.

The system of code observation equations is nonlinear; therefore, it is required to linearize it around an approximate value of the receiver coordinates. This approximated position is computed by performing three iterations of least squares on code pseudorange observations starting from the center of the earth.

A way to improve the result of stand-alone positioning is to use the corrections coming from a SBAS (satellite-based augmentation system). The European Geostationary Navigation Overlay Service (EGNOS) is a SBAS, developed to improve the performance of GPS and Galileo in the European region. EGNOS allows for improved accuracy in stand-alone positioning with respect to GPS-only solutions, by means of corrections to GPS orbits, satellite clocks and ionospheric delays. The EGNOS corrections are divided into two categories: fast and long term. The fast corrections are used to model rapidly changing terms such as the satellite clock errors; they are provided as corrections to be applied directly to the code pseudorange observations. The long-term corrections are for the slower changing terms, due to the ionosphere, the mis-modeling of the broadcast orbits and the long-term components of the satellite clock error. The fast corrections and the clock/orbit parts of the long-term corrections are common to all users, while the ionospheric corrections depend on the user position, since they are derived from a gridded model of the ionosphere.

goGPS MATLAB uses several of the standard EGNOS messages that are needed to achieve improved accuracy. These implemented messages are shown in Table 1. The formulation of the implemented corrections can be seen in (RTCA DO-229D 2006).

The implementation of EGNOS corrections in goGPS includes both fast and long-term corrections, but their availability depends on the data source. The most complete source of EGNOS data, that includes all corrections available, is the EGNOS Message Server (EMS). The EMS stores the augmentation messages broadcast by EGNOS in hourly text files and permits access to archived broadcast messages by FTP protocol. goGPS automatically connects to EMS, downloads the necessary data and uses them to apply the corrections to GPS orbits and observations. However, the corrections provided by the EMS are not available in real time, so they may be applied only for post-processing purposes. Some low-cost receivers, such as those based on the u-blox LEA-4T/5T/6T modules, provide some of the EGNOS corrections in real time by means of receiver-specific binary messages that can be enabled with the raw data output. In the case of u-blox receivers, EGNOS data are included in the NAV-SBAS message (UBX format), but they are limited to the fast corrections and the ionospheric corrections. The current version of goGPS supports only EMS; therefore, it can apply EGNOS corrections only in post-processing. Support for decoding and using corrections from the UBX-NAV messages has already been implemented and can be provided on request, but it is not included in the current official release.

Table 1 EGNOS messages implemented in goGPS

Type	Content
1	PRN mask assignments
2–5	Fast corrections
18	Ionospheric grid points mask
24	Mixed fast corrections/long-term satellite error corrections
25	Long-term satellite error corrections
26	Ionospheric delay corrections

Furthermore, goGPS includes a module to deal with several GNSS constellations that include the Russian system GLONASS, the Chinese BeiDou, the European Galileo and the Japanese QZSS. At a basic level, multi-constellation processing has to take into account at least the design differences between terrestrial reference systems and time systems. In addition, the presence of inter-system (receiver clock) biases (ISB) has to be assumed because of the alignment errors among the different time systems. Table 2 gives an overview of the multi-constellation design characteristics taken into account in goGPS.

The transformation between PZ-90.02 and WGS84 is valid from 20/09/2007 at 18:00 UTC (Revnivykh 2008). Its expression in meters is given by:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{\mathrm{ITRF2000}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{\mathrm{PZ-90.02}} + \begin{bmatrix} -0.36 \\ 0.08 \\ 0.18 \end{bmatrix} \quad (2)$$

No transformation between Galileo Terrestrial Reference Frame (GTRF) and WGS-84 is currently implemented in goGPS, but its agreement with WGS-84 is deemed to be sufficient for positioning with broadcast ephemeris (Hofmann-Wellenhof et al. 2008); the same is assumed for CGCS2000.

goGPS uses GPS time as a reference for the processing. Galileo time and QZSS time are aligned with GPS time, so there is no need to apply known time offsets beforehand. On the contrary, GLONASS time is aligned to UTC time, so the current number of leap seconds has to be applied. Actually there is a fixed 3-h offset between GLONASS

Table 2 Constellations, terrestrial reference systems (TRS) and time systems, as currently implemented in goGPS

System	TRS	Time system
GPS	WGS-84	GPST
GLONASS	PZ-90.02	UTC (GPST–leap sec)
Galileo	GTRF	GST (GPST)
BeiDou	CGCS2000	BDT (GPST–14 s)
QZSS	WGS-84	QZSST (GPST)

time and UTC time, but this is already accounted for in the time tag read from RINEX files. The start epoch of BeiDou time is 00:00:00 on January 1, 2006, UTC; therefore, the leap seconds on that moment (14 s) have to be taken into account to align BeiDou time with GPS time. In addition, ISBs are estimated by means of additional unknowns in a classical least squares problem. If, at any given epoch, i satellite systems are available, $i-1$ ISB parameters are added to the design matrix in a classical LSA. Equation 3 shows an example for code observations at an epoch with n GPS, m GLONASS and s QZSS satellites.

$$\begin{bmatrix} P_{\mathrm{r}}^{g1} \\ P_{\mathrm{r}}^{g2} \\ \vdots \\ P_{\mathrm{r}}^{gn} \\ P_{\mathrm{r}}^{\mathrm{r}1} \\ \vdots \\ P_{\mathrm{r}}^{\mathrm{rm}} \\ P_{\mathrm{r}}^{j1} \\ \vdots \\ P_{\mathrm{r}}^{js} \end{bmatrix} = \begin{bmatrix} a_X^{g1} & a_Y^{g1} & a_Z^{g1} & 1 & 0 & 0 \\ a_X^{g1} & a_Y^{g2} & a_Z^{g2} & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_X^{gn} & a_Y^{gn} & a_Z^{gn} & 1 & 0 & 0 \\ a_X^{\mathrm{r}1} & a_Y^{\mathrm{r}1} & a_Z^{\mathrm{r}1} & 1 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_X^{\mathrm{r}m} & a_Y^{\mathrm{r}m} & a_Z^{\mathrm{r}m} & 1 & 1 & 0 \\ a_X^{j1} & a_Y^{j1} & a_Z^{j1} & 1 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_X^{js} & a_Y^{js} & a_Z^{js} & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X-\tilde{X} \\ Y-\tilde{Y} \\ Z-\tilde{Z} \\ c\cdot \mathrm{d}t_{\mathrm{r}} \\ c.\delta_{\mathrm{GLO}} \\ c.\delta_{\mathrm{QZSS}} \end{bmatrix} + \underline{b} \tag{3}$$

where code observations are indicated by the notation P_{r}^{gk} ($k = 1,\ldots,n$ with n being the number of GPS satellites), $P_{\mathrm{r}}^{\mathrm{r}l}$ ($l = 1,\ldots,m$ with m being the number of GLONASS satellites) and P_{r}^{jq} ($q = 1,\ldots,s$ with s being the number of QZSS satellites); the a parameters represent the linearization coefficients of the pseudorange equation; $\tilde{X}$, $\tilde{Y}$, $\tilde{Z}$ are the approximate coordinates for the receiver antenna; c is the speed of light; the unknowns are expressed as X, Y, Z for the receiver antenna coordinates, $\mathrm{d}t_{\mathrm{r}}$ for the receiver clock error and δ_{GLO}, δ_{QZS} for the ISBs of GLONASS and QZSS, respectively, in this case, with respect to GPS. $\underline{b}$ is the vector of constant terms from the linearized pseudorange equations using initial or updated values of the parameters. goGPS can process single-frequency code and phase observations for all the currently available GNSS constellations (excluding IRNSS) read from RINEX v2.10-12 and v3.01 files. The satellite orbits and clocks are computed from the broadcast ephemeris and clock parameters. The constellations to be processed can be selected by means of the goGPS graphical user interface. Support for the multi-constellation low-cost receiver NV08C-CSM (by NVS Technologies AG) is also included. It is important to note that at the moment, the receiver must be updated with a demo firmware that provides raw data output not only for GPS and GLONASS, as with the standard firmware, but also for Galileo and QZSS.

Relative positioning with KF

In this section, we focus on the EKF used with phase and code observations, since this is the most original part of the software. The Kalman filter implemented in goGPS works on double-difference observations with respect to a reference station (Realini 2009).

In the goGPS KF, the quantities used as state variables for double-difference observations are position, velocity and ambiguities. A generic nth-order dynamic model can be written as

$$\begin{cases} \bar{X}(t+1) = \bar{X}(t) + \bar{X}'(t) \\ \bar{X}'(t+1) = \bar{X}'(t) + \bar{X}''(t) \\ \vdots \\ \bar{X}^{(n)}(t+1) = 0 + \bar{X}^{(n)}(t) + \varepsilon_{\bar{X}^{(n)}}(t+1) \\ N_{\mathrm{r}m}^{\mathrm{p}1}(t+1) = N_{\mathrm{r}m}^{\mathrm{p}1}(t) \\ \vdots \\ N_{\mathrm{r}m}^{\mathrm{ps}}(t+1) = N_{\mathrm{r}m}^{\mathrm{ps}}(t) \end{cases} \tag{4}$$

where $\bar{X}$ is the receiver position $[X_{\mathrm{r}}, Y_{\mathrm{r}}, Z_{\mathrm{r}}]$, $\varepsilon_{\bar{X}^{(n)}}$ is the model error and $N_{\mathrm{r}m}^{\mathrm{p}i}$ represents the double differences of ambiguities. Various types of motion can be modeled by using different values for n; therefore, $n=0$ means "rover is static," $n=1$ means "constant velocity," $n=2$ means "constant acceleration" and so on.

The transition matrix T will be defined as follows

$$T = \begin{bmatrix} \bar{T} & 0 \\ 0 & I \end{bmatrix} \tag{5}$$

with

$$\bar{T} = \begin{bmatrix} T_n & 0 & 0 \\ 0 & T_n & 0 \\ 0 & 0 & T_n \end{bmatrix} \tag{6}$$

and

$$T_n = \begin{bmatrix} 1 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & 1 & \ddots & & \vdots \\ \vdots & \ddots & 1 & 1 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & \ddots & 1 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 \end{bmatrix} \tag{7}$$

The phase observation equation that combines master and rover observations, with respect to the pivot satellite p and each satellite s, is written as

$$\lambda \phi_{rm}^{ps}(t) = \rho_{rm}^{ps}(t) - \lambda N_{rm}^{ps}(t) - I_{rm}^{ps}(t) + T_{rm}^{ps}(t) + \eta_{rm}^{ps}(t) \quad (8)$$

where $N_{rm}^{ps}(t)$ is the double-difference of ambiguities between rover and master, λ is the L1 wavelength, $\eta_{rm}^{ps}(t)$ is the phase measurement double-difference noise, $\rho_{rm}^{ps}(t), I_{rm}^{ps}(t), T_{rm}^{ps}(t)$ are the double difference of the geometric distance, the double differences of the ionospheric effect and the double differences of the tropospheric effect, respectively.

For level baselines less than 10 km, Eq. (8) can be approximated as

$$\lambda \phi_{rm}^{ps}(t) = \rho_{rm}^{ps}(t) - \lambda N_{rm}^{ps}(t) + \eta_{rm}^{ps}(t). \quad (9)$$

The observation vector can be written as

$$\underline{Y}_t^{(\text{phase})} = \begin{bmatrix} \vdots \\ \lambda \phi_{rm}^{ps}(t) + a_{xr}^{ps}\tilde{X}_r(t) + a_{yr}^{ps}\tilde{Y}_r(t) + a_{zr}^{ps}\tilde{Z}_r(t) - \tilde{\rho}_{rm}^{ps}(t) \\ \vdots \end{bmatrix} \quad (10)$$

where $a_{xr}^{ps}, a_{yr}^{ps}, a_{zr}^{ps}$ are the coefficients of the linearization of the double differences of the geometric distance for all s $\neq$ p where s represents visible satellites; consequently, the design matrix H becomes

$$H_t^{(\text{phase})} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{xr}^{ps}(t) & 0 & a_{yr}^{ps}(t) & 0 & a_{zr}^{ps}(t) & 0 & 0 & \cdots & \lambda & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \quad (11)$$

with a λ value in the column corresponding to the satellite s.

The phase measurement noise covariance matrix is the known covariance matrix of the double-difference observations. For low-cost receivers, the way of weighting observations takes into account the satellite elevation and/or carrier-to-noise ratio (C/N0). From the results of empirical tests, it has been decided to use an inverse square sine function to introduce the elevation information in the weighting function, which becomes

$$W(CN0, e) = \begin{cases} \dfrac{1}{\sin^2 e}\left(10^{-\frac{(CN0-s1)}{a}}\left(\left(\dfrac{A}{10^{-\frac{(s0-s1)}{a}}} - 1\right)\dfrac{CN0 - s1}{s0 - s1} + 1\right)\right) & CN0 < s1 \\ 1 & CN0 \geq s1 \end{cases} \quad (12)$$

where e is the elevation angle. The $s1$ parameter defines the threshold after which the weight is set to 1, it means the $C/N0$ value after which measurements are considered "good"; the $s0$ parameter defines the $C/N0$ value for which the function is "forced" to have the weight defined by the A parameter. The a parameter defines the bending of the curve. Sample values are $A = 30$, $s0 = 10$, $s1 = 50$ and $a = 20$. The user can choose between different weighting options: without weights, as function of elevation of the satellites, as function of $C/N0$ of the satellites and as a combination of both of them. More details can be seen in (Realini 2009).

The initialization of the goGPS KF starts with a first estimation of the rover approximate position by computing three iterations of linearized least squares adjustment on undifferenced code observations, starting from the center of the earth, resulting in a position with an error of some meters. This result is then used as an approximate position to apply stand-alone positioning as explained above. This position is then passed to the KF algorithm as the $X_r(0), Y_r(0), Z_r(0)$ state variables. The velocities $\dot{X}_r(0), \dot{Y}_r(0), \dot{Z}_r(0)$ can be initialized to zero, under the hypothesis that the rover starts its motion from a stationary state, or to a constant value equal to the cruise velocity of the rover.

As for the initialization of ambiguities, the first float number of cycles N_{rm}^{ps} is obtained from a least squares solution on code and phase observations, or, if the number of observations is not sufficient, by the equation

$$N_{rm}^{ps} = \frac{\rho_{rm}^{ps} - \lambda \phi_{rm}^{ps}}{\lambda} + \frac{\eta_{rm}^{ps}}{\lambda} \quad (13)$$

in which ρ_{rm}^{ps} is estimated as $\rho_{rm}^{ps} = P_{rm}^{ps} - v_{rm}^{ps}$, where P_{rm}^{ps} is the combined pseudorange from code measurements and v_{rm}^{ps} is the code double-difference noise. Thus, the estimated number of cycles $\hat{N}_{rm}^{ps}$ will be $\hat{N}_{rm}^{ps} = \frac{P_{rm}^{ps} - \lambda \phi_{rm}^{ps}}{\lambda} \rightarrow \sigma_{\hat{N}} \cong \sigma_v / \lambda \sim 5$ cycles.

The initial state covariance matrix is defined as a diagonal matrix in which we will have the position error variances, the velocity variances, the acceleration variances and the estimated variance of the initial ambiguity; it is introduced just on the lines corresponding to visible satellites and can be different from satellite to satellite (Realini 2009). Note that a high velocity variance can be used to model the acceleration of a body that starts moving from a stationary state. It is important to note that in principle also covariances between position errors could be provided.

The resolution of the ambiguities is based on the LAMBDA method. goGPS implements both versions 2 and 3 of the LAMBDA MATLAB code (http://gnss.curtin.edu.au/research/lambda.cfm) developed at Delft University of Technology and Curtin University (Teunissen 1995; De Jonge and Tiberius 1996; Verhagen and Li 2012). goGPS applies the LAMBDA method either in the epoch-by-epoch processing modes or in the EKF processing modes. In the latter case, the Kalman filter keeps continuous float ambiguity estimation; LAMBDA is then applied after the filter update step (Suhandri and Realini 2013).

Cycle slips are a fundamental aspect to deal with when trying to achieve high accuracy with low-cost GPS devices. Compared to high-level receivers, carrier phase measurements coming from these kinds of sensors are subject to various measurement errors, such as multipath, antenna phase center variation and noise due to the lower quality of the electronic components involved (Takasu and Yasuda 2008). These error sources, together with cycle slip occurrences, severely deteriorate the achievable positional accuracy.

In goGPS, the detection of cycle slips at time t is performed by comparing the KF ambiguity update at that time with an estimation of the number of cycles made on the basis of observations. Since according to the goGPS KF dynamic model the number of cycles should not change over time, then the updated ambiguity is

$$N_{rm}^{ps}(t|t-1) = \hat{N}_{rm}^{ps}(t-1). \tag{14}$$

Its variance $\sigma^2_{N_{rm}^{ps}(t|t-1)}$ is obtained from the covariance matrix $C_{t-1}^{e} = T_t C_{t-1}^{e} T_t^T$ at the position that corresponds to the satellite s. The new estimation of $\hat{N}_{rm}^{ps}(t|t-1)$ using observations at time t is

$$\hat{N}_{rm}^{ps}(t|t-1) = \frac{\rho_{rm}^{ps}(t|t-1) - \lambda\phi_{rm}^{ps}(t)}{\lambda}. \tag{15}$$

Then the following test is applied in order to detect cycle slips $\left|N_{rm}^{ps}(t|t-1) - \hat{N}_{rm}^{ps}(t|t-1)\right| < M \Rightarrow$ no cycle slip where M is an empirical threshold value in cycles set by the user; e.g., ten cycles would be equivalent to about a 2-m error.

Variometric approach for velocity estimation

goGPS includes a variometric approach applied to phase observations that is aimed at estimating the velocity of a stand-alone receiver (Traugott et al. 2008, Colosimo et al. 2011). This method exploits epoch-differenced phase observations to estimate the displacement vector from epoch t to epoch $t + \Delta t$.

Tests and results

In this section, some examples obtained by goGPS with different positioning modes are introduced. These tests correspond to the examples and data set included in the manual for the program. In this way, users will be able to reproduce the results by themselves.

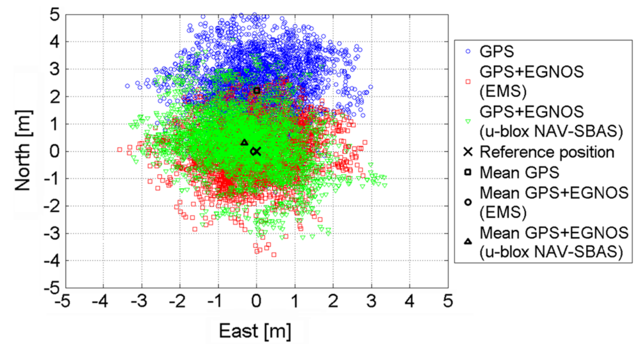


Fig. 1 Positioning results obtained from 1-hour single-frequency code observations by a low-cost receiver u-blox AEK-4T, 12 December 2012 (DOY 347)

Stand-alone positioning with EGNOS corrections

To check the positioning improvement achieved with the introduction of EGNOS corrections, the following experiment was performed. The standard u-blox antenna (ANN-MS, patch type) was connected to the low-cost receiver u-blox AEK-4T and placed on a point of known coordinates, on the rooftop of a building at the University of Jaen (Spain). Measurements were taken over 1 h with a 1-s sampling rate, on December 12, 2012. goGPS was used to post-process the acquired data according to the following three modes of operation: GPS only, GPS with EGNOS corrections from the logged u-blox NAV-SBAS message and GPS with EGNOS corrections downloaded from the EMS. The results are shown in Fig. 1. It can be seen that the GPS-only solution has a bias of about 2 m with respect to the known coordinates. The use of the fast and ionospheric corrections included in the u-blox NAV-SBAS message reduces the bias to about 1 m. Lastly, the use of the complete set of EGNOS corrections from the EMS (fast, long term and ionospheric) provides the best result, with the bias further reduced to less than 0.5 m (Herrera et al. 2012).

Multi-constellation support

As an example of the outcome of the multi-constellation implementation in goGPS, we present here the results of epoch-by-epoch processing of stand-alone single-frequency code observations from the JAXA station GMSD, belonging to the IGS Multi-GNSS Experiment (MGEX, http://igs.org/mgex/) network. The observations cover a time span of 24 h, on June 15, 2013, at 30-s rate. The elevation cutoff was set to 15°. Until epoch 246, the only available Galileo satellite was E19; from epoch 403, Galileo satellites E19 and E20 were available. The number of available BeiDou satellites at any given epoch ranged between 7 and 11 during this time span, including a total of

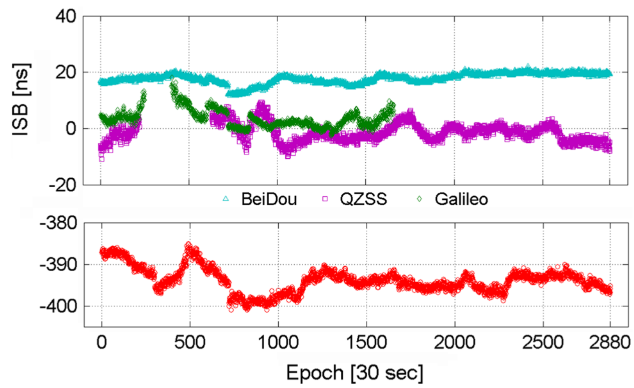


Fig. 2 Estimated inter-system biases (ISBs) with respect to GPS for BeiDou, QZSS, Galileo and GLONASS. IGS MGEX station "GMSD," June 15, 2013 (DOY 166)

Table 3 Mean and standard deviation of the estimated ISBs in Fig. 3

	Mean	Standard deviation
BeiDou	17.8 ns $\simeq$ 5.34 m	1.9 ns $\simeq$ 0.57 m
Galileo	3.5 ns $\simeq$ 1.05 m	3.2 ns $\simeq$ 0.96 m
GLONASS	-393.9 ns $\simeq$ 118.09 m	3.0 ns $\simeq$ 0.90 m
QZSS	-1.5 ns $\simeq$ -0.45 m	3.2 ns $\simeq$ 0.96 m

13 different PRNs: four geostationary Earth orbit (GEO) satellites (C1–C4), five medium Earth orbit (MEO) satellites (C6–C10) and four inclined geosynchronous satellite orbit (IGSO) satellites (C11–C14).

Figure 2 shows the estimated ISBs of BeiDou, QZSS, Galileo and GLONASS with respect to GPS; each of the four constellations was processed separately, in a dual-constellation configuration with GPS. Mean and standard deviation of the four constellations are presented in Table 3. The positioning mean difference with respect to the known coordinates of the GMSD station antenna and the standard deviation of these differences, for different combinations of constellations, are summarized in Fig. 3. The ionospheric delay was mitigated by applying the Klobuchar model derived from the broadcast ionospheric parameters for GPS, adjusted for the various frequencies. It should be pointed out that the GPS + Galileo and GPS + QZSS combinations give almost the same error as the GPS-only processing, since only two Galileo satellites not at the same time and one QZSS satellite are added, respectively.

Relative positioning with integer ambiguity resolution

In this case, stationary single-frequency relative positioning tests using an u-blox AEK-4T receiver are carried out. Its ANN-MS antenna was placed on a point of known coordinates on the rooftop of a building at the Politecnico di Milano, Italy, along with a geodetic base station. The baseline length was about 60 m. Processing settings include an elevation cutoff of 15°, double-differenced observation noise standard deviations of 30 cm (code) and 3 mm (phase) and an exponential weighting function based on satellite elevations. The LAMBDA version 3 code was used. We report here an example of the results obtained over a timespan of about 12 h, on April 5, 2013, with an observation rate of 15 s. Figure 4 compares the float results estimated by the goGPS EKF, and the integer result estimated by LAMBDA. The differences between the EKF final positioning and the ground truth are 1 cm (east), 2.8 cm (north) and 6.9 cm (up) for the float solution, and 0.6 cm (east), 2.9 cm (north) and 6.8 cm (Up) for the integer solution. The relatively large biases are deemed to be due to the undefined phase center of the low-cost antenna, which makes it difficult to properly place it on the point of known coordinates, as well as to its lack of a calibrated phase center offset.

Navigation

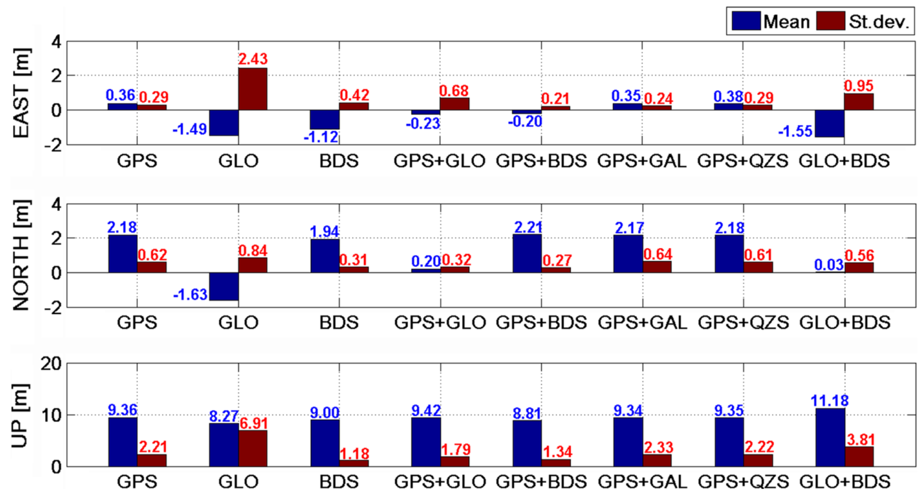
The performance of the navigation algorithm is presented by means of a path surveyed by car on an island in Osaka Bay, Japan, with good conditions of satellite visibility (a 10° elevation cutoff was adopted for the data processing).

An u-blox antenna was mounted on the car rooftop along with two dual-frequency receivers that were used for the accuracy assessment. The car was driven at approximately constant velocity along the whole path in order to be consistent with the Kalman filter dynamical model implemented in goGPS. The u-blox observations were processed in relative mode with respect to a VRS (virtual reference station) roughly located in the barycenter of the surveyed path. A comparison with the positioning results obtained by the dual-frequency receivers was performed, showing an overall accuracy of the order of half a meter. Therefore, these results improve the usual accuracy from 2 to 4 m to some decimeters. The interested reader can find additional details on the experimental test in Realini and Reguzzoni (2013).

Conclusions

The open-source positioning software goGPS has been introduced. This is the result of the collaboration between researchers from different universities and countries, made possible by the public availability of the goGPS code. The main functionalities of the software are summarized, and some examples are presented to demonstrate the capabilities of the software, in terms of achievable accuracy, by means of both stand-alone positioning and relative

Fig. 3 Accuracy of the epoch-by-epoch processing of multi-constellation single-frequency code observations by the IGS MGEX station "GMSD," June 15, 2013 (DOY 166)



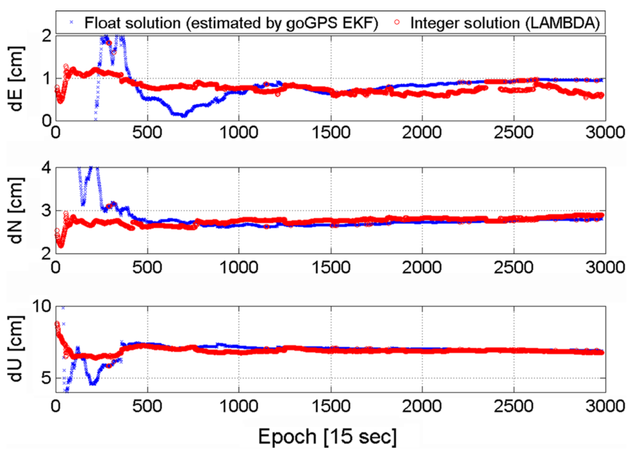


Fig. 4 *E*, *N*, *U* differences [cm] with respect to the known position using float and integer ambiguities

positioning. The examples included correspond to those presented in the manual. In this way, the user can replicate the results using the files delivered with the manual.

References

Colosimo G, Crespi M, Mazzoni A (2011) Real-time GPS seismology with a stand-alone receiver: a preliminary feasibility demonstration. J Geophys Res: Solid Earth (1978–2012), 116(B11)

De Jonge P, Tiberius C (1996) The LAMBDA method for integer ambiguity estimation: implementation aspects. Publications of the Delft Computing Centre, LGR-Series (12)

Herrera A, Caldera S, de Lacy MC, Realini E, Reguzzoni M (2012) An Analysis of the improvements introduced by EGNOS into the software goGPS for kinematic positioning using low cost receivers.7[a] Asamblea Hispano-Portuguesa de Geodesia y Geofísica, pp. 5–7

Hofmann-Wellenhof B, Lichtenegger H, Wasle E (2008) GNSS—global navigation satellite systems—GPS, GLONASS, Galileo, and more. Springer

Klobuchar JA (1987) Ionospheric time-delay algorithm for single-frequency GPS users. IEEE Trans Aerosp Electron Syst AES 23(3):325–331

Leick A, Rapoport L, Tatarnikov D (2015) GPS satellite surveying, 4th edn. New York, Wiley

Realini E (2009) goGPS free and constrained relative kinematic positioning with low cost receivers. Ph. D. thesis, http://www.researchgate.net/publication/237520116

Realini E, Reguzzoni M (2013) goGPS: open source software for enhancing the accuracy of low-cost receivers by single-frequency relative kinematic positioning. Meas Sci Technol 24(11):115010

Realini E, Yoshida D, Reguzzoni M, Raghavan V (2012) Enhanced satellite positioning as a web service with goGPS open source software. Appl Geomat 4(2):135–142

Revnivykh S (2008) GLONASS status and progress. In: Proceeding 48th Meeting of the Civil GPS Service Interface Committee (CGSIC) 2008. Savannah, GA, US

RTCA DO-229D (2006) Minimum Operational Performance Specification for Global Positioning System / Wide Area Augmentation System Airborne Equipment. Appendix A. Prepared by SC-159. December 13 2006

Saastamoinen J (1973) Contribution to the theory of atmospheric refraction. Bull Géod 107(1):13–14

Suhandri HF, Realini E (2013) Issues of different estimation models for epoch-by-epoch double-difference GPS observation equations: a comparative study. In: Proceedings of European Navigation Conference (ENC) 2013, Vienna, Austria

Takasu T, Yasuda A (2008) Evaluation of RTK-GPS Performance with low-cost single-frequency GPS Receivers, presented at International Symposium on GPS/GNSS 2008, November 11–14. Tokyo, Japan

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

Teunissen P, Kleusberg A (1998) GPS for geodesy. Springer, Berlin

Traugott J, Dell'Omo G, Vyssotski AL, Odijk D, Sachs G (2008) A time-relative approach for precise positioning with a miniaturized L1 GPS logger. Proc ION GNSS 2008, Institute of Navigation, September 16–19, Savannah, GA, US, pp. 1883–1894

Verhagen S, Li B (2012) LAMBDA—Matlab implementation, version 3.0. Delft University of Technology and Curtin University

Antonio M. Herrera Antonio M. Herrera is a Ph.D. candidate in the University of Jaen. He received his Master's Degree in Telecommunications Engineering in 2008 from Polytechnic School of Linares, University of Jaen. He is a member of "Microgeodesia Jaén" research group. His current research activity focuses on GNSS low-cost receivers, multi-constellations positioning and geodetic control ground deformation.

Hendy F. Suhandri Hendy F. Suhandri holds a master's degree in 2008 from the Geoengine program at the University of Stuttgart, Germany. Since 2009 he has been a research fellow at the Institute of Navigation, Stuttgart. His research focuses on stochastic modelling of GNSS observations, ambiguity resolution, and kinematic attitude determination.

Eugenio Realini Eugenio Realini got his Master's Degree in Environmental Engineering in 2005 and his Ph.D. in Geodesy and Geomatics in 2009, both from Politecnico di Milano (Italy). His research activities focus on GNSS positioning and GNSS-based tropospheric analysis. He is co-founder and developer of goGPS, an open source software application for GNSS positioning. He is now researcher at Geomatics Research & Development (GReD), a spin-off company of Politecnico di Milano.

Mirko Reguzzoni Mirko Reguzzoni is an assistant professor at Politecnico di Milano, where he graduated in Computer Engineering in 1999 and got a Ph.D. in Geodesy and Geomatics in 2004. He is also president of the International Service for the Geoid and scientific advisor of Geomatics Research & Development (GReD), a spin-off company of Politecnico di Milano. He mainly works on Earth gravity field modelling and in general on statistical data analysis, with applications to physical geodesy, geophysics and GNSS positioning. He is co-founder and developer of goGPS, an open source software package for GNSS positioning.

M. Clara de Lacy M. Clara de Lacy is an associate professor of Geodesy at the department of Cartographic, Geodetic and Photogrammetric Engineering in theUniversity of Jaén. She is a member of the UJA Geodesy Research Group. She is involved in research on GNSS positioning and navigation using GPS and Galileo, crustal deformation monitoring and engineering geodesy.