

The GPSTk: an open source GPS toolkit

R. Benjamin Harris · Richard G. Mach

Published online: 5 October 2006
© Springer-Verlag 2006

Abstract Applied Research Laboratories, The University of Texas at Austin (ARL:UT) has established a cross platform open source software project called the GPSTk or the GPS Toolkit. The GPSTk consists of a library and collection of applications that support GPS research, analysis, and development. The code is released under the terms of the Lesser GNU Public License. The GPSTk supports a broad range of functionality. This includes reading and writing observations in standard formats, such as RINEX, BINEX, and SP3, ephemeris evaluation, position determination, receiver autonomous integrity monitoring (RAIM), atmospheric delay modeling, cycle slip detection and correction, and P-code generation. The GPSTk provides the core set of functionality that is used for GPS research and development at ARL:UT. ARL:UT has been involved with satellite navigation since Transit (the precursor to GPS) in the 1960s and is currently conducting research in a wide variety of GPS-related fields, including precise

surveys, monitor station networks, and ionospheric studies. The GPSTk is a community-wide resource for all users of GPS and GNSS technology. Participation is welcomed in all areas including: bug reports, new algorithms, suggestions for improvement, and contributions of additional functionality or applications. ARL:UT continually improves the library, shepherds community participation, and is committed to the project's development and maintenance.

Keywords GPSTk · GPS · Toolkit · Open source · ARL:UT · Applied Research Laboratories · Development tools · GNSS · RINEX

Introduction

Applied Research Laboratories, The University of Texas at Austin (ARL:UT) has established an open source software project called the GPS Toolkit, or the GPSTk. The GPSTk provides a library and collection of applications that support GPS research, analysis, and development. The code is released under the terms of the Lesser GNU Public License (LGPL, see <http://www.gnu.org/licenses/lgpl.html>). The LGPL grants the user a number of rights, notably the ability to choose whether to modify and redistribute the source with any application, whether closed or open source. The design of the GPSTk is object oriented and the code is written in platform independent ANSI C++. It has been installed and tested successfully under AIX, Linux, Solaris, and Windows.

The GPSTk distribution and current development version are available for download on SourceForge.net at <http://www.gpstk.org/>. To obtain the development

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 gps-toolbox@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.

R. B. Harris · R. G. Mach (✉)
Applied Research Laboratories,
The University of Texas at Austin,
10000 Burnet Road, Austin, TX 78758, USA
e-mail: rmach@arlut.utexas.edu

version of the source requires the Subversion version control system (<http://www.subversion.org>). The web-site also provides bug reporting, a feature request list, email groups, and a user manual for the applications. A user can generate the application programming interface (API) documentation from the code using the doxygen package (<http://www.doxygen.org>). The generated documentation can also be found on the GPSTk website.

This article presents information about the GPSTk that enables the reader to understand how it can fit into their tool set. It presents insight into the reasons and goals behind the GPSTk. It summarizes the library and application functionality and shows an example of using the GPSTk. It concludes with a description of future directions for the project.

Opening the GPSTk

ARL:UT has developed software to support GPS research and has fielded systems for two decades. During that time there have been several versions of GPS related shared libraries utilized among the internal projects. Over time, these libraries diverged into project specific customizations. ARL:UT recognized the value of the common library base since these shared libraries have been key to facilitating the work on new projects or new GPS research tools. Thus, ARL:UT undertook an effort to merge and modernize these libraries to form a cross platform, production quality, C++ library with an object oriented design.

After completing this effort, ARL:UT had developed a portable, modern library and tool set that serves as the basis for several projects at ARL:UT. This provides great benefit to internal development. We realized the value the GPSTk could bring to the greater GPS community considering the lack of open source projects of its kind. Most of the GPS open source projects are not research oriented but are aimed at navigation using a receiver generated position, velocity, and time solution (Tolman and Harris 2004). Closed source projects have drawbacks that limit their usefulness. They lack the benefits of transparency into the internal methodology, lack the ability for end-user enhancement to fit a project's needs, and lack the potential for reuse of existing functionality in a new application. In contrast, open source projects guarantee the user the ability to use, modify, and redistribute the source. If ARL:UT was to provide the GPSTk to the GPS community in support of research and development, we wanted to do so in a manner that provided the most value. Thus, ARL:UT established

the GPSTk as an open source project. This allows transparency and flexibility that is extremely important in research and academic areas in which we participate. It also allow ARL:UT to offer the GPS community the same benefit we have for internal projects.

ARL:UT chose the LGPL because of the flexibility and openness it provides. This choice of licensing does not limit the user's development environment, allowing the GPSTk to be used in any open or closed source project without requiring the project to open its source code. ARL:UT believes that the open source model will be beneficial to the GPSTk by allowing any of the users to contribute to the effort in a variety of ways. These contributions encompass both small and large ones, from reporting a bug or suggesting an improvement, to contributing new or improved functionality.

Design of the GPSTk

The design goals of the GPSTk library are portability, modularity, clarity, extensibility, and maintainability. These goals allow the GPSTk to maximize the audience and lifetime of the library while decreasing the costs associated with long-term maintenance. The GPSTk meets these goals by strict adherence to the ANSI C++ standard and Object Oriented Analysis and Design (OOA/D) techniques.

OOA/D contrasts with procedural design supported by the C and FORTRAN languages. In procedural programming, a function library is provided to the user. In Object Oriented (OO) programming, a 'class library' is provided. Each class is an independent module that can be invoked by the user as an object or extended by the user in the form of a new class. Classes can build upon each other through a number of object-oriented principles, such as inheritance, the use of metaclasses, polymorphism, and aggregation. The GPSTk library relies heavily on the Standard Template Library (STL, see <http://www.sgi.com/tech/stl/>), which is part of the ANSI standard for C++. The STL provides OO data structures (containers). These include linked lists, vectors, and standard operations such as the quicksort algorithm.

All of the GPSTk code includes documentation designed for extraction by doxygen, a freely available package that generates a HTML-based document set from the code itself. Like the GPSTk, doxygen is platform-independent. The doxygen documents are available on the GPSTk web site or can be easily generated from the code in the distribution.

To date, the applications of the GPSTk, with one exception, do not use graphical user interfaces but in-

stead use command-line input. This design choice makes the code extremely platform-independent. The entire distribution has been installed and tested under AIX (gcc), Linux (gcc), Solaris (Forte and gcc), and Windows (Visual Studio .NET 2003, Borland, and Cygwin/gcc). Support for other development environments should be readily achievable.

Description of the GPSTk

The GPSTk distribution contains a library, a collection of applications, and a test suite. The library consists of the base functionality used to develop applications that read, process, and write GPS data. The applications are standalone utilities that are valuable for examining and processing GPS data and for providing more detailed programming examples. The test suite includes programs that test parts of the library. The test programs provide not only validation of a new installation but also additional examples of the how to use the library in standalone programs.

The GPSTk library functionality

The GPSTk library supports a wide range of functionality, including, but not limited to, RINEX input and output, ephemeris handling (broadcast/RINEX or SP3), date and time conversions, matrix and vector algorithms, mathematical and statistical algorithms, tropospheric and ionospheric modeling, positioning and Receiver Autonomous Integrity Monitoring (RAIM), and support packages (exceptions, string utilities, and command line arguments). Each of these areas is briefly described in the following paragraphs.

RINEX and BINEX routines

The library provides for the reading and writing of data files in the RINEX format that is common to many projects. The RINEX specification version 2.1 is completely supported. The library also includes the capability to define custom ‘extended RINEX’ observation types that are useful for storing results and intermediate quantities in RINEX-like files.

Currently the library also supports low level BINEX input and output. Standard BINEX messages (see <http://www.binex.unavco.org>) are not interpreted by the library at this time. The routines can be used to parse custom BINEX formats. However, the user can write code that uses the existing capability to read or write a standard BINEX file with code that passes those quantities to or from other classes in the library.

Ephemeris processing

The library provides complete satellite ephemeris storage capabilities with input from both broadcast ephemerides (RINEX navigation files) and SP3 format files. The ephemeris storage classes encapsulate all the details of ephemeris handling. They require only input file names to load the ephemerides and implement the standard computational algorithms such as the satellite position algorithm defined in the IS-GPS-200 document (ARINC 2004), formerly called ICD-GPS-200. The ephemeris objects are designed using object-oriented methods (inheritance and polymorphism) so that users can easily switch usage between broadcast and precise ephemerides.

Time conversion

Dates and times are fundamental to GPS and to the GPSTk. The ‘DayTime’ class implements many useful date and time formats into a single very powerful object. These formats include Modified Julian Date, GPS time, and calendar dates. There are also text string-based routines that allow custom formatting of time tags, as well as text string interpretation.

Vector and matrix routines

The GPSTk library includes an extensive matrix and vector package built entirely using templates. The package includes the usual arithmetic operators like addition, subtraction, multiplication, and division. It also includes LU decomposition, singular value decomposition, Cholesky decomposition, and inversion of matrices. There are also several mathematical algorithms in the library that include statistics (1- or 2-sample), polynomial fitting, and Lagrange interpolation.

Tropospheric modeling

Several troposphere models are provided in the library. Their designs use inheritance and polymorphism so that the user may easily interchange models or create a new one. There are currently five models implemented that include Hopfield models (Hopfield 1970) and the New Brunswick UNB3 model (Collins and Langley 1997).

Positioning

Positioning and navigation provides autonomous pseudorange solutions implemented using either linearized least squares or the algebraic solution (Bancroft 1985). Additionally, single baseline and network dif-

ferential solutions that can achieve millimeter level accuracies are provided. To support data editing, a receiver autonomous integrity monitoring (RAIM) algorithm is also included.

Additional functionality

There is much more functionality in the GPSTk library than can be detailed here. There is a P-code generator that produces the bit sequence for P-code as defined by IS-GPS-200 (ARINC 2004) and a Runge Kutta integrator. An application framework is defined that allows programmers to easily create console applications. There are a wide variety of low-level support functionalities, such as exception handling, string utilities, formatted file handling, and command line argument processing that ease the development of console based programs.

The GPSTk applications

The GPSTk distribution includes several complete applications. These programs utilize the library to read, process, write, and display GPS data. These programs serve both as useful utilities and as detailed examples of programming using the GPSTk library. The GPSTk includes the following types of applications: RINEX utilities, RINEX plotting, positioning, cycle slip detection and estimation, residual analysis, date and time conversions, and ionospheric modeling. Each of these is described in more detail in the following paragraphs. More detail can be found in the user's manual available through the GPSTk website.

RINEX utilities

The RINEX utilities provide a set of applications that can be used to examine and manipulate RINEX observation files. They include capabilities to summarize a file, extract selected data into a simple columnar format, edit a file, and generate data residuals or corrections like ionospheric total electron content (TEC), elevation and azimuth, geometry-free phase, and wide-lane bias. Additionally, conversion from some Ash-tech, Novatel, and ITT receiver specific formats to RINEX is provided.

RINEX plotting

The RINEX plotting application is a menu-driven graphical interface to the RINEX utilities. This application takes the results of these utilities and displays them in a 2D scatter plot. For this application to work,

Perl (<http://www.perl.org>) and Tk (<http://www.tcl.sourceforge.net>) are required in addition to the base GPSTk requirements.

Positioning

The positioning applications include two different applications that perform standard pseudorange-based positioning and two that implement differential phase-based solutions. The pseudorange solutions are similar to autonomous GPS receiver solutions. The differential solutions can solve for a single baseline or an entire network of sites. Differential accuracies in the millimeter range can be achieved.

Cycle slip correction

The cycle slip application implements an algorithm based on the work of Blewitt (1990). This application detects and estimates the size of the cycle slip using the geometry-free and wide lane linear combinations of raw dual frequency pseudorange and carrier phase data. The results can be utilized by the RINEX utilities to produce a cycle slip-corrected RINEX observation file.

Residual analysis

A residual analysis application computes two types of measurement residuals using a single receiver or two receivers in a zero baseline configuration. For a single receiver, observed range deviations (observed minus expected pseudoranges) are computed using a broadcast or precise ephemeris. For the zero baseline, double differences across receivers and satellites are computed which are a function of only receiver noise and hardware biases. A satellite comparison tool differences satellite positions from two different sources. The outputs are suitable for plotting in gnuplot (<http://www.gnuplot.info>) or other plotting programs.

Time conversions

The time conversion utility is a simple application that converts one time format into a variety of other formats. This is handy when looking at two different file formats that contain different time representations like GPS week and seconds of week or year, month, day, hour, minute, and second.

Ionospheric modeling

The ionospheric modeling applications utilize the two frequency TEC estimate from the RINEX utilities and

compute a model of the ionosphere. This process includes estimating the biases in the TEC observation due to biases in the GPS receivers and satellites. The results are suitable for plotting in gnuplot or other plotting programs.

Getting started with the GPSTk

Building, installing, and starting programming with the GPSTk requires only a few simple steps. This installation process and a programming example are shown below. The GPSTk website includes more detailed instructions and additional programming examples.

Building the GPSTk

Building the GPSTk distribution requires only a few commands at the console. The use of jam simplifies the build process because it is an easy to use, platform independent, alternative to make (<http://www.perforce.com/jam/jam.html>). The program provides commands to compile, install, and test the source code. Precompiled binaries for the supported operating systems, including AIX, Linux, Solaris and Windows, are also available on the GPSTk website. In addition, it is possible to use the GNU autoconf/make tool set to build the source instead of jam.

The simplest way to build the GPSTk source is using jam on a UNIX or UNIX-like system. In that case, install jam and doxygen on your system and download the GPSTk compressed tar file from <http://www.gpstk.org>. The entire build sequence looks like the following within a shell:

tar xvzf gpstk-1.0.tar.gz	# decompress & expand tar file
cd gpstk	# change to gpstk directory
jam	# build library and applications
doxygen	# build the html documentation
su	# change to root/super user
jam -sPREFIX=/usr install	# install the library into /usr/local

This sequence will build and install the GPSTk dynamic and shared libraries, the header files, and the applications in the /usr/local directory. In addition a doc subdirectory is created with html-based documentation of the GPSTk library. Additional installation instructions for UNIX and Windows are located on the GPSTk website.

Library example code

The following code segment illustrates how easy it is to use the GPSTk library to process GPS data stored in RINEX format. The example demonstrates how the

expressive power of C++ is fully utilized by the GPSTk so that the user can focus on writing application-specific code. This example is based on Example 2 from the website and distribution. Four examples are available on the GPSTk web site and the complete source code is located in the example subdirectory of the distribution.

In Fig. 1, a RINEX observation file is parsed. The observations are combined to isolate multipath, which is then written to standard output. Lines 1 and 2 declare the C++ namespaces used by the GPSTk. Namespaces are used to prevent name clashes and to organize classes by topic. All of the GPSTk code is part of the namespace gpstk. On line 3, a C++ stream to read the RINEX observations is created. Lines 4 through 6 contain objects used to interpret the observation stream. Note that the GPS satellite associated with PRN 14 will be examined. Line 8 shows how a C++ stream is used to read the observations and to control a loop. Lines 11 through 14 demonstrate how to traverse the data structure of observations supplied from the observation stream. While the given data structure is specific to the GPSTk, the form is common to any program using the STL. The STL find function is used to search for observations for a given PRN and the bracket operator is used to pick out the observables. On line 15, the pseudorange minus phase multipath combination is formed. On line 16, the combination is written to standard output. The use of

```

Line No.
1   using namespace std;
2   using namespace gpstk;
3   RinexObsStream roffs("bahr1620.04o");
4   RinexObsData roe;
5   RinexObsData::RinexDatum dataobj;
6   RinexPrn prn(14, (RinexSystem) systemGPS);
7   double gamma = (L1_FREQ / L2_FREQ) * (L1_FREQ / L2_FREQ);
8   while (roffs >> roe)
9   {
10      RinexObsData::RinexPrnMap::iterator pointer =
11         roe.obs.find(prn);
12      if( itr != roe.obs.end() )
13      {
14         dataobj = (*itr).second[RinexObsHeader::P1];
15         double P1 = dataobj.data;
16         double P2 =
17             itr->second[RinexObsHeader::P2].data;
18         double L1= itr->second[RinexObsHeader::L1].data;
19         double mu = P1 - L1 * (C_GPS_M/L1_FREQ) -
20             2.0*(P1-P2)/(1-gamma);
21         cout << roe.time.GPSfullweek()
22             << roe.time.GPSSow()
23             << setprecision(11)
24             << mu << endl;
25      }
26   }

```

Fig. 1 Example user code

the DayTime class in roe.time is notable in that the time is transparently converted from RINEX format to GPS format.

The future of the GPSTk

ARL:UT is committed to supporting and enhancing the GPSTk for the foreseeable future since it serves as the basis of our projects. The plans are to regularly (approximately semiannually) post full releases that include enhancements and fixes to the code. The latest development version is also always available. Significant enhancements to the code base will be driven by ARL:UT internal development needs, user community contributions, and external sponsorship. Users within ARL:UT have already identified a number of near term goals. These include platform support for Mac OS X, ambiguity resolution, RINEX 2.11 and 3.0 support, other language bindings (Tcl, Tk, Perl, et al.), and software receiver capabilities.

In the long term, ARL:UT expects the library to be enhanced to support the changing GNSS environment. These changes encompass the new GPS signals such as M-code, L1C, L2C, and L5 and the upcoming European Galileo system (Onidi 2002). With its emphasis on fundamental observations, the GPSTk can provide the basis to explore and exploit these and other changes.

ARL:UT continues to foster an open source community that utilizes the capabilities of the GPSTk and helps lead the direction of the library. We have seen that the GPSTk is appealing to a broad range of audiences. Universities use the GPSTk to process GPS

data with open source code that facilitates learning. Embedded programmers develop software to perform GPS positioning, and to read, write and edit RINEX data files easily and quickly. Researchers find that this code is an excellent foundation for many GPS applications. It is our hope that university students, laboratory researchers, system engineers, and software developers will continue to contribute to, as well as benefit from, the GPS Toolkit. We have already seen many benefits to using this code and believe that the GPS community will continue to see similar results.

References

- ARINC Engineering Services, LLC (2004) Interface specification: Navstar GPS space segment/navigation user interfaces, IS-GPS-200, Revision D, 7 December (<http://www.navcen.uscg.gov/gps/geninfo/IS-GPS-200D.pdf>)
- Bancroft S (1985) An algebraic solution of the GPS equations. *IEEE Trans Aerosp Electron Syst* 21(7):56–59
- Blewitt G (1990) An automatic editing algorithm for GPS data. *Geophys Res Lett* 17(3):199–202
- Collins JP, Langley RB (1997) A tropospheric delay model for the user of the wide area augmentation system. Technical Report No. 187, Geodesy and Geomatics Engineering, University of New Brunswick
- Hopfield, HS (1970) Tropospheric effect on electromagnetically measured range: prediction from surface weather data. Applied Physics Laboratory, Johns Hopkins University, Baltimore
- Onidi O (2002) GALILEO is launched. European Commission, 25 September
- Tolman BW, Harris RB (2004) The GPS toolkit. Linux J, September, p 72