Introduction to BKG Ntrip Client (BNC) Usage

Leoš Mervart

GPS Solutions Inc., Boulder
Institute of geodesy, TU Prague

Frankfurt, March 2012

BNC's technical details

BNC from the programmer's point of view

- BNC source consists currently of approximately 50.000 lines of code
- approximately 90 % is C++, 10 % standard C
- BNC uses a few third-party pieces of software (first of all the RTCM decoders/encoders and a matrix algebra library)
- Qt library is used for
  1. GUI,
  2. networking,
  3. threads,
  4. containers, streams, files, ...

BNC is intended to be

- user-friendly
- cross-platform
- easily modifiable (by students, GNSS beginners)
- useful (at least a little bit ...)

BNC source code

Algorithms used in BNC are intended to be

- correct, but
- as simple as possible

```
//
////////////////////////////////////////////////////////////////////////////
void bncModel::kalman(const Matrix& AA, const ColumnVector& ll,
                      const DiagonalMatrix& PP,
                      SymmetricMatrix& QQ, ColumnVector& dx) {

  Tracer tracer("bncModel::kalman");

  int nObs = AA.Nrows();
  int nPar = AA.Ncols();

  UpperTriangularMatrix SS = Cholesky(QQ).t();

  Matrix SA = SS*AA.t();
  Matrix SRF(nObs+nPar, nObs+nPar); SRF = 0;
  for (int ii = 1; ii <= nObs; ++ii) {
    SRF(ii,ii) = 1.0 / sqrt(PP(ii,ii));
  }

  SRF.SubMatrix   (nObs+1, nObs+nPar, 1, nObs) = SA;
  SRF.SymSubMatrix(nObs+1, nObs+nPar)          = SS;

  UpperTriangularMatrix UU;
  QRZ(SRF, UU);

  SS = UU.SymSubMatrix(nObs+1, nObs+nPar);
  UpperTriangularMatrix SH_rt = UU.SymSubMatrix(1, nObs);
  Matrix YY  = UU.SubMatrix(1, nObs, nObs+1, nObs+nPar);

  UpperTriangularMatrix SHi = SH_rt.i();

  Matrix KT  = SHi * YY;
  SymmetricMatrix Hi; Hi << SHi * SHi.t();

  dx = KT.t() * ll;
  QQ << (SS.t() * SS);
}
```

svn archive

BNC source code may be downloaded from the svn archive using a command

A) `svn co http://software.rtcm-ntrip.org/svn/trunk/BNC`

or

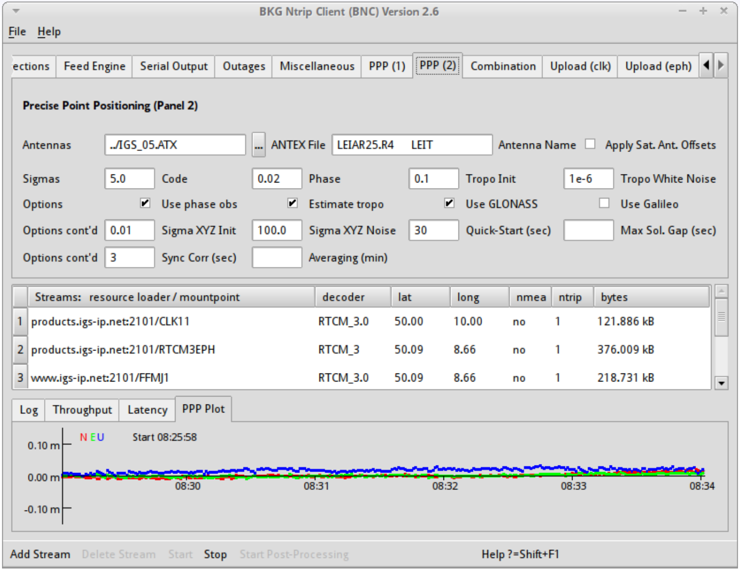
B) `svn co https://software.rtcm-ntrip.org/svn/trunk/BNC`

Option A) is a read-only access. Option B) is for the developers (read-write access). When the source code is downloaded using the `https` (secure protocol) currently two additional sub-directories are retrieved:

- combination
- rinex

The sub-directory "combination" contains the source code of the BNC module that performs the combination of PPP corrections streams provided by several analysis centers (more about the combination algorithms below).
The sub-directory "rinex" contains the module for the post-processing PPP client that uses the RINEX files as input (this directory is not yet made public because it is still under development).

Precise Point Positioning with PPP



Precise Point Positioning with PPP (cont.)

BNC provides a good framework for the PPP client (observations, orbits, and corrections stand for disposal).
Main reasons for the PPP module in BNC have been:

- monitoring the quality of incoming data streams (primarily the PPP corrections)
- providing a simple easy-to-use tool for the basic PPP positioning

The PPP facility in BNC is provided in the hope that it will be useful.

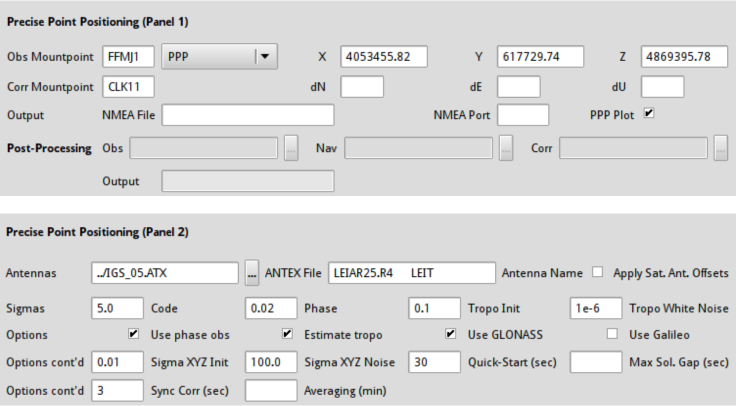
- The mathematical model of observations and the adjustment algorithm are implemented in such a way that they are (according to our best knowledge) correct without any shortcomings, however,
- we have preferred simplicity to transcendence, and
- the list of options the BNC users can select is limited.

$\Rightarrow$ Commercial PPP clients may outperform BNC in some aspects.

We believe in a possible good coexistence of the commercial software and open source software.

PPP Options

- single station, SPP or PPP
- real-time or post-processing
- processing of code and phase ionosphere-free combinations, GPS, Glonass, and Galileo



Combination of PPP Corrections

User's dilemma:

There are so many different data streams with PPP corrections available on NTRIP caster. Which one should I use?

Combination of PPP Corrections

User's dilemma:

There are so many different data streams with PPP corrections available on NTRIP caster. Which one should I use?

Use the combined one!

Combination of PPP Corrections

User's dilemma:

There are so many different data streams with PPP corrections available on NTRIP caster. Which one should I use?

Use the combined one!

Are the combined corrections always the best?

Combination of PPP Corrections

User's dilemma:

There are so many different data streams with PPP corrections available on NTRIP caster. Which one should I use?

Use the combined one!

Are the combined corrections always the best?

It depends ...

The combination of the PPP corrections

- increases the reliability, but
- may (slightly) decrease the quality because the combination algorithm is (currently) not fully correct (the combination algorithms must neglect information that is not provided in correction streams).

Simple combination of PPP corrections

The simple algorithm for the combination

- subtract the analysis-center specific biases
- computes the mean over all corrections for each satellite

Simple combination of PPP corrections

The simple algorithm for the combination

- subtract the analysis-center specific biases
- computes the mean over all corrections for each satellite

However, for PPP clock accuracy on DD level matters!

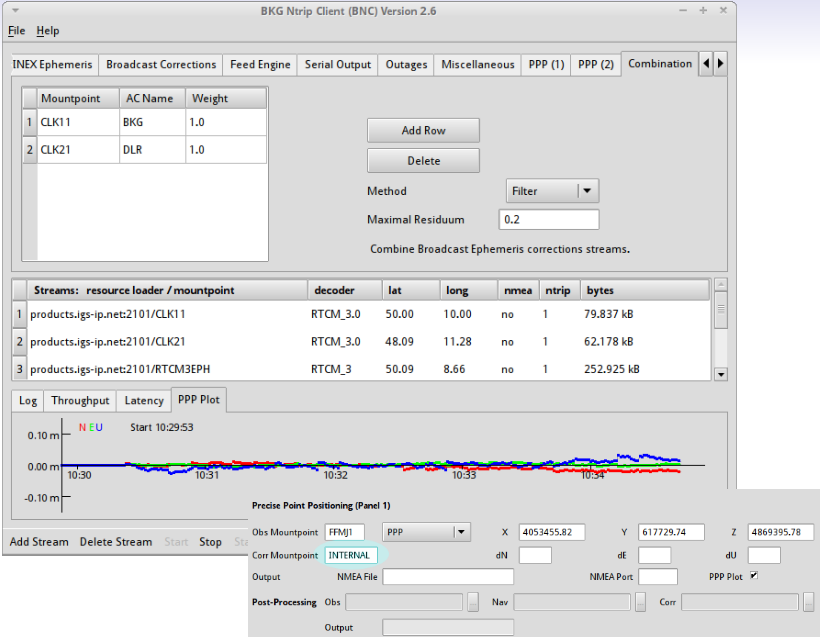
Observation equations for two satellites i, j and two epochs t_1, t_2:

$$
\begin{aligned}
L^i(t_1) &= \varrho^i(t_1) + c\,\delta(t_1) - c\,\delta^i(t_1) + b^i \\
L^j(t_1) &= \varrho^j(t_1) + c\,\delta(t_1) - c\,\delta^j(t_1) + b^j \\
L^i(t_2) &= \varrho^i(t_2) + c\,\delta(t_2) - c\,\delta^i(t_2) + b^i \\
L^j(t_2) &= \varrho^j(t_2) + c\,\delta(t_2) - c\,\delta^j(t_2) + b^j
\end{aligned}
$$

Eliminating the clock parameters $\delta(t_1), \delta(t_2)$ and the ambiguities (biases) b^i, b^j is equivalent to forming a double difference (between two satellites and between two epochs):

$$
L^{ij}(t_1) - L^{ij}(t_2) = \varrho^{ij}(t_1) - \varrho^{ij}(t_2) - c\ \underbrace{\left[(\delta^i(t_1) - \delta^j(t_1)) - (\delta^i(t_2) - \delta^j(t_2))\right]}_{\text{DD Clock}}
$$

Combination Options



Combination using Kalman filtering

The combination is performed in two steps

1. The satellite clock corrections that refer to different broadcast messages (different IODs) are modified in such a way that they all refer to common broadcast clock value (common IOD is that of the selected "master" analysis center).
2. The corrections are used as pseudo-observations for Kalman filter using the following model (observation equation):

$$c_a^s = c^s + o_a + o_a^s$$

where

c_a^s	is the clock correction for satellite s estimated by the analysis center a,
c^s	is the resulting (combined) clock correction for satellite s,
o_a	is the AC-specific offset (common for all satellites), and
o_a^s	is the satellite and AC-specific offset.

The three types of unknown parameters c^s, o_a, o_a^s differ in their stochastic properties: the parameters c^s and o_a are considered to be epoch-specific while the satellite and AC-specific offset o_a^s is assumed to be a static parameter.

Combination Results