G-Nut

Software for GNSS processing

Pavel Vaclavovic, Jan Dousa

Research Institute of Geodesy and Cartography and Topography
Geodetic Observatory Pecny

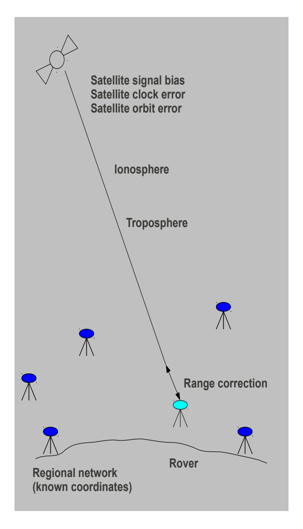
18.4.2012 / GOP seminar

Outline

1. **Introduction to PPP**
   - OSR vs SSR approaches
   - Observation equations
   - Adjustment procedure
   - Precise point positioning correction model
2. **G-Nut library**
   - Introduction
   - IO and data structures
   - Applications
     - PPP client
   - Future planes for G-Nut applications
3. **Summary**

OSR vs SSR approaches

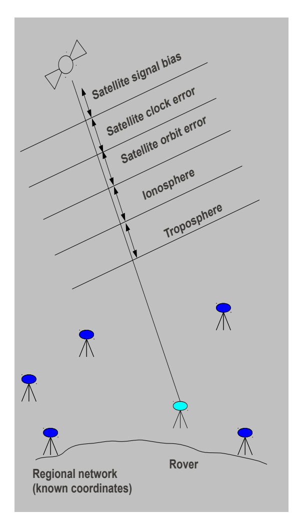
Observation Space Representation



- All errors are lumped to one parameter
- Double difference are formed
- Implemented in Real Time Kinematic
- Raw data
  and corrections are transmitting to rover
- Distance from
  reference station is main limiting factor
- Regional network needed for determine distance dependent error (FKP, VRS, ...)
- High rate of data transmission (1 Hz)
- Nature integer ambiguity

OSR vs SSR approaches

State Space Representation



- Implemented
  in Precise Point Positioning
- GNSS error components have different characteristic ⇒ better to divide them
- Better modelling
  and interpolation of individual error
- Optimization
  of communication bandwidth
- Not nature integer ambiguity

OSR vs SSR approaches

Standardization

- Standards for OSR (network RTK) well designed (Version 3)
- RTCM SC104 - SSR Working Group
  - Established in 2007
  - Main goal is to develop messages for SSR
- Status of SSR messages
  - 1^{st} stage
    - precise orbits, satellite clocks, satellite code biases
    - for dual frequency real time PPP
    - finished in 2011
  - 2^{nd} stage
    - vertical ionosphere TEC, satellite phase biases
    - for single frequency real time PPP
    - under development
  - 3^{rd} stage
    - slant ionosphere TEC and troposphere
    - for PPP-RTK
    - under development

OSR vs SSR approaches

Standardization

- Standards for OSR (network RTK) well designed (Version 3)
- RTCM SC104 - SSR Working Group
  - Established in 2007
  - Main goal is to develop messages for SSR
- Status of SSR messages
  - 1^{st} stage
    - precise orbits, satellite clocks, satellite code biases
    - for dual frequency real time PPP
    - finished in 2011
  - 2^{nd} stage
    - vertical ionosphere TEC, satellite phase biases
    - for single frequency real time PPP
    - under development
  - 3^{rd} stage
    - slant ionosphere TEC and troposphere
    - for PPP-RTK
    - under development

OSR vs SSR approaches

Standardization

- Standards for OSR (network RTK) well designed (Version 3)
- RTCM SC104 - SSR Working Group
  - Established in 2007
  - Main goal is to develop messages for SSR
- Status of SSR messages
  - 1^{st} stage
    - precise orbits, satellite clocks, satellite code biases
    - for dual frequency real time PPP
    - finished in 2011
  - 2^{nd} stage
    - vertical ionosphere TEC, satellite phase biases
    - for single frequency real time PPP
    - under development
  - 3^{rd} stage
    - slant ionosphere TEC and troposphere
    - for PPP-RTK
    - under development

Introduction to Precise Point Positioning

Observation equations

- General GNSS equations

$$
\begin{aligned}
P^i_{Af} &= \rho^i_A + c\delta_A - c\delta^i + I^i_{Af} + T^i_A + cB_{Af} - cB^i_f \\
L^i_{Af} &= \rho^i_A + c\delta_A - c\delta^i - I^i_{Af} + T^i_A + cb_{Af} - cb^i_f + \lambda_f N^i_{Af}
\end{aligned}
$$

- Modified PPP equations

$$
\begin{aligned}
P^i_{A3} &= \rho^i_A + c\delta_A + Mztd_A + cB_{Af} \\
L^i_{A3} &= \rho^i_A + c\delta_A + Mztd_A + cb_{Af} - cb^i_f + \lambda_f N^i_{Af} \\
P_3 &= \frac{1}{f_1^2 - f_2^2}(f_1^2 P_1 + f_2^2 P_2) \\
L_3 &= \frac{1}{f_1^2 - f_2^2}(f_1^2 L_1 + f_2^2 L_2)
\end{aligned}
$$

Introduction to Precise Point Positioning

Observation equations

- General GNSS equations

$$
\begin{aligned}
P^i_{Af} &= \rho^i_A + c\delta_A - c\delta^i + I^i_{Af} + T^i_A + cB_{Af} - cB^i_f \\
L^i_{Af} &= \rho^i_A + c\delta_A - c\delta^i - I^i_{Af} + T^i_A + cb_{Af} - cb^i_f + \lambda_f N^i_{Af}
\end{aligned}
$$

- Modified PPP equations

$$
\begin{aligned}
P^i_{A3} &= \rho^i_A + c\delta_A + Mztd_A + cB_{Af} \\
L^i_{A3} &= \rho^i_A + c\delta_A + Mztd_A + cb_{Af} - cb^i_f + \lambda_f N^i_{Af} \\
P_3 &= \frac{1}{f_1^2 - f_2^2}(f_1^2 P_1 + f_2^2 P_2) \\
L_3 &= \frac{1}{f_1^2 - f_2^2}(f_1^2 L_1 + f_2^2 L_2)
\end{aligned}
$$

Introduction to Precise Point Positioning

Adjustment procedure

- Classical formula of Kalman filter
- Square root covariance filter (SRCF)
- Square root information filter (SRIF)
- Backward Kalman smoothing

Adjustment procedure

Kalman filter

- Prediction
  State variables are linearly transformed from epoch t_0 to epoch t_1

$$
\begin{aligned}
x_1 &= D x_0 + e \\
Q_1 &= D Q_0 D^T + Q_e
\end{aligned}
$$

- Update
  Estimate state variables based on new measurements

$$
\begin{aligned}
\hat{x} &= x_1 + K(l - A x_1) \\
K &= Q_1 A^T (Q_l^{-1} + A Q_1 A^T)^{-1} \\
Q &= Q_1 - K A Q_1
\end{aligned}
$$

- Important: Kalman filter does not guarantee positive-definiteness covariance matrix

Adjustment procedure

Kalman filter

- Prediction
  State variables are linearly transformed from epoch t_0 to epoch t_1

$$\begin{aligned} x_1 &= Dx_0 + e \\ Q_1 &= DQ_0D^T + Q_e \end{aligned}$$

- Update
  Estimate state variables based on new measurements

$$\begin{aligned} \hat{x} &= x_1 + K(l - Ax_1) \\ K &= Q_1A^T(Q_l^{-1} + AQ_1A^T)^{-1} \\ Q &= Q_1 - KAQ_1 \end{aligned}$$

- Important: Kalman filter does not guarantee positive-definiteness covariance matrix

Adjustment procedure

Kalman filter

- Prediction
  State variables are linearly transformed from epoch t_0 to epoch t_1

$$
\begin{aligned}
x_1 &= Dx_0 + e \\
Q_1 &= DQ_0D^T + Q_e
\end{aligned}
$$

- Update
  Estimate state variables based on new measurements

$$
\begin{aligned}
\hat{x} &= x_1 + K(l - Ax_1) \\
K &= Q_1A^T(Q_l^{-1} + AQ_1A^T)^{-1} \\
Q &= Q_1 - KAQ_1
\end{aligned}
$$

- Important: Kalman filter does not guarantee positive-definiteness covariance matrix

Adjustment procedure

Square root covariance filter

- Does not propagate covariance matrix but its square root (Cholesky decomposition)
- Use QR decomposition

Modified kalman update:

$$
\begin{aligned}
H &= Q_l + A Q_1 A^T \\
Q_1 &= S_1^T S_1, \quad Q = S^T S, \quad Q_l = S_l^T S_l, \quad H = S_H^T S_H
\end{aligned}
$$

$$
QR\left(\begin{bmatrix} S_l & 0 \\ S_1 A^T & S_1 \end{bmatrix}\right) \Longrightarrow \begin{bmatrix} S_H & K S_H^T \\ 0 & S \end{bmatrix}
$$

$$
\hat{x} = x_1 + K(l - A x_1)
$$

- Covariance matrix Q is always positive-definiteness

Adjustment procedure

Square root covariance filter

- Does not propagate covariance matrix but its square root (Cholesky decomposition)
- Use QR decomposition

Modified kalman update:

$$\begin{aligned} H &= Q_l + AQ_1A^T \\ Q_1 &= S_1^T S_1, \quad Q = S^T S, \quad Q_l = S_l^T S_l, \quad H = S_H^T S_H \end{aligned}$$

$$QR\left(\begin{bmatrix} S_l & 0 \\ S_1 A^T & S_1 \end{bmatrix}\right) \Longrightarrow \begin{bmatrix} S_H & KS_H^T \\ 0 & S \end{bmatrix}$$

$$\hat{x} = x_1 + K(l - Ax_1)$$

- Covariance matrix Q is always positive-definiteness

Adjustment procedure

Square root information filter

- Propagate square root of information matrix ($A^T PA$)
- Use QR decomposition

Modified kalman update:

$$Q_x^{-1} = A^T PA = R_x^T R_x$$

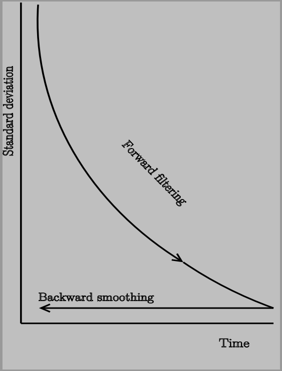
$$QR \left(\begin{bmatrix} A & l \\ R_x & z \end{bmatrix} \right) \Longrightarrow \begin{bmatrix} R_x & z \\ 0 & \rho \\ 0 & 0 \end{bmatrix}$$

$$\hat{x} = R_x \diagdown z$$

- Covariance matrix Q is always positive-definiteness

Adjustment procedure

Backward Kalman smoothing



- RTS (Rauch Tung Striebel) smoother
  1. Compute filtered quantities $x_{k|k-1},\ x_{k|k},\ Q_{k|k-1},\ Q_{k|k}$
  2. Compute smoothed quantities $x_{k|n},\ Q_{k|n}$

$$
\begin{aligned}
x_{k|n} &= x_{k|k} + K_k(x_{k+1|n} - x_{k+1|k}) \\
Q_{k|n} &= Q_{k|k} + K_k(Q_{k+1|n} - Q_{k+1|k})K_k^T \\
K_k &= Q_{k|k} Q_{k+1|k}^{-1}
\end{aligned}
$$

Introduction to Precise Point Positioning

Precise point positioning correction model

- Phase wind-up
  - For phase measurement only
  - Antenna rotation ⇒ change carrier phase up to one cycle
  - Important during "noon" or "midnight turn"
  - Almost negligible for double difference, but significant in PPP
- Phase center offset and variation
  - Need to be applied when using IGS orbit products
  - Must be consistent with terrestrial frame
- Solid Earth tides
- Ocean loading
- Earth rotation parameters
  - Required as long as inertial frame is used
- Relativistic effects

Introduction to Precise Point Positioning

Precise point positioning correction model

- Phase wind-up
  - For phase measurement only
  - Antenna rotation $\Rightarrow$ change carrier phase up to one cycle
  - Important during "noon" or "midnight turn"
  - Almost negligible for double difference, but significant in PPP
- Phase center offset and variation
  - Need to be applied when using IGS orbit products
  - Must be consistent with terrestrial frame
- Solid Earth tides
- Ocean loading
- Earth rotation parameters
  - Required as long as inertial frame is used
- Relativistic effects

G-Nut library
Introduction

- Flexible library for building application according individual requirements
- Primary designed for Linux OS
- Developed in C++ taking advantages of OOP
- Flexible configuration by XML format
- Support multi-GNSS (GPS, GLONASS, Galileo, ...)
- Support filter and LSQ processing (real-time stream or data file)
- As much as possible use C++ standard library
- GUI will be developed independently for prominent applications
- Flexible IO classes supporting various input/output strategies

G-Nut library

IO and data structures

- Unique system for file or stream
  Different decoders has been developed, but users use common virtual method for input or output (gio class)
  Inheritance:

$$gio \begin{cases} gfile \\ gtcp \end{cases}, \quad gtcp \longrightarrow gntrip$$

- Common approach for real time and post processing
  System of C++ containers (usually map)
  (few items = real time, manny items = post processing)

G-Nut library

Applications

- gNut ntrip client
- gNut ntrip server
- gNut ntrip pipe
- PPP client

Application

PPP client

- Current status
  - Static or kinematic positioning
  - Code + phase observations
  - Float ambiguity
  - Basic mapping function for ZTD
  - Post processing and real time mode
- Future plans
  - Decrease convergence time and overcome data gap
  - Post processing based on LSQ
  - Enhance ZTD mapping functions
  - Enhance models (PCV, troposphere ...)
  - Integer ambiguity resolution
  - Multi GNSS - GPS, GLONASS, Galileo
  - Use up to date IERS conventions

Application

PPP client

- Current status
  - Static or kinematic positioning
  - Code + phase observations
  - Float ambiguity
  - Basic mapping function for ZTD
  - Post processing and real time mode
- Future plans
  - Decrease convergence time and overcome data gap
  - Post processing based on LSQ
  - Enhance ZTD mapping functions
  - Enhance models (PCV, troposphere ...)
  - Integer ambiguity resolution
  - Multi GNSS - GPS, GLONASS, Galileo
  - Use up to date IERS conventions

PPP client

XML configuration


```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>

<config>
    <ppp>
        <settings
          tropo="true"
          static="true"
          phase="true"
          save_for_smooth="false"
          filter="SRF"
          smooth="kalman"
        />

        <inputs
          observations="../../data/sample/gope0700.12o"
          orbits_nav=""
          orbits="ID0|file://../../data/sample/ig116786.sp3|sp3"
          clocks="ID1|file://../../data/sample/igs16786.clk_30s|rinexc"
          BNC_rtcm=""
          PCV="ID3|file://../../data/sample/igs08_1664.atx|atx"
          filter_result="result.forw"
          smooth_result="result.back"
        />
    </ppp>
</config>
```

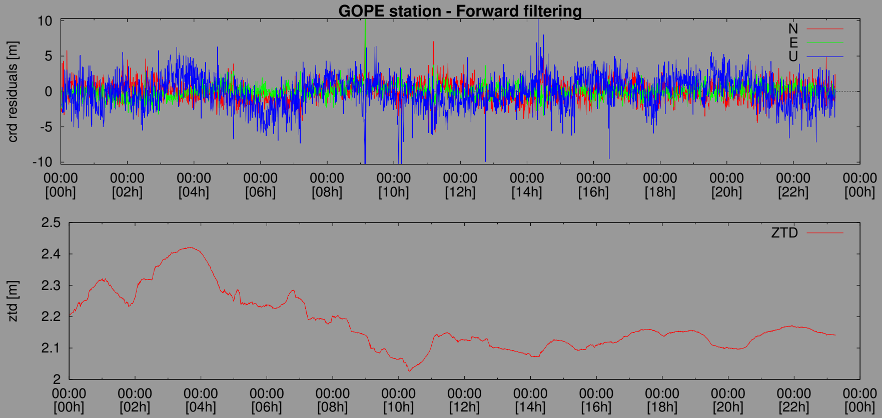
PPP client

Processing strategy

1. Compute apriory coordinates and receiver clock from code data (depends if station is static or kinematic)
2. Parameters prediction
3. Preprocessing
   - Cycle slip detection
   - Receiver clock jump detection
   - Outliers identification
4. Parameters update
   - Kalman filter
   - Square root covariance filter
   - Square root information filter
5. Backward smoothing

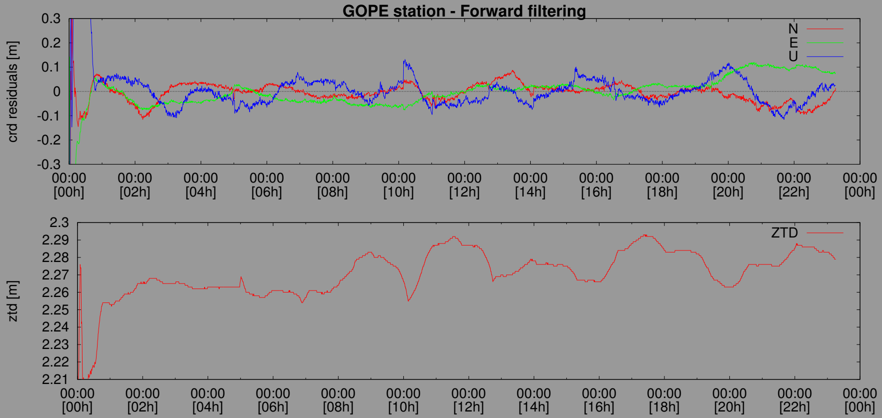
PPP client

Results: Code (IGS orbit/clock) 30s



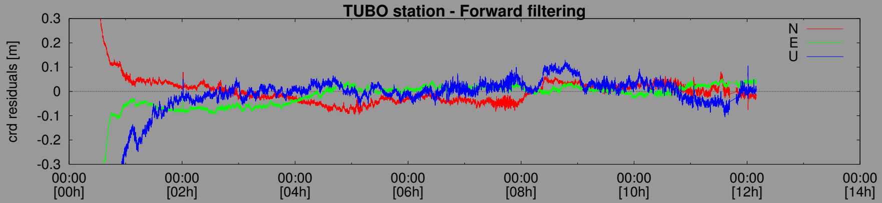
PPP client

Results: Code + Phase (IGS orbit/clock) 30s



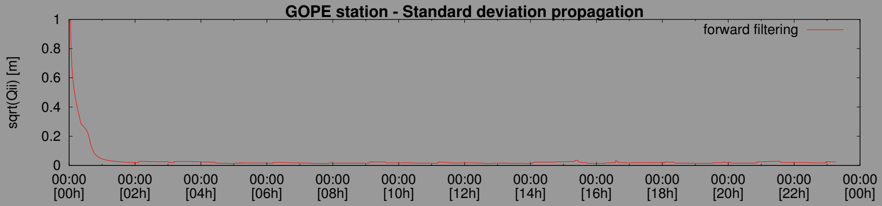
PPP client

Results: Code + Phase (BNC-RTCM correction) 1s



PPP client

Results: Code + Phase (IGS orbit/clock)



Future planes for G-Nut applications (milestones)

- Real time position calculation
  - Static or kinematic applications (including high-rate >1Hz data)
- (Near) real time ZTD determination and suport for regional troposphere modelling
- Online PPP service
- Real time satellite clock corrections
- Service for ambiguity resolution in PPP - UPD estimation
- Regional augmentation (PPP-RTK, fast amb. resolution)

Summary

- If we want to be in contact with scientific community we have to develop our own software
- G-Nut is software library for wide use
- Individual application will be built according requirements
- Most application will be based on Precise Point Positioning
- See www.pecny.cz(GNSS/software) for all information