Satellite Navigation
Integrity and integer ambiguity resolution

AE4E08

Sandra Verhagen

Course 2010 - 2011, lecture 12
Today’s topics

- Integrity and RAIM
- Integer Ambiguity Resolution

- Study Section 7.4 – 7.6 (not LMS algorithm in 7.5.3)
Integrity: performance measures

**Integrity** = ability of a system to provide timely warnings to users when the system should not be used

- **HPE/VPE**: Horizontal/Vertical Position Error
  (not known; due to measurement noise and biases)

- **HPL/VPL**: Horizontal/Vertical Protection Level
  horizontal/vertical position is assured to be within region defined by HPL/VPL

- **HAL/VAL**: Horizontal/Vertical Alarm Limit
  position error that should result in an alarm being raised

- **TTA**: Time To Alarm
  time between occurrence of integrity event (position error too large) and alarm being raised
Integrity: performance measures

- **HPE/VPE**: Horizontal/Vertical Position Error
  (not known; due to measurement noise and biases)
- **HPL/VPL**: Horizontal/Vertical Protection Level
  horizontal/vertical position is assured to be within region defined by HPL/VPL (can be calculated)
- **HAL/VAL**: Horizontal/Vertical Alarm Limit
  position error that should result in an alarm being raised

required: \( P(XPE > XPL) < \) integrity risk

\( XPL > XAL \) → alarm, system unavailable
Stanford plots
http://waas.stanford.edu/metrics.html
GPS PRN 23 Anomaly, 1 Jan, 2004

East, North and Up differences – delf001s.04r

-500
-400
-300
-200
-100
0
100
200
300
400
500

Number of epochs

North
East
Up

at 18:30 (UT)

3 min.

Not noticed by US for 3 hours
Picked up by EGNOS
Alternative: check @receiver
Receiver Autonomous Integrity monitoring

Local Overall Model test statistic – delf001s.04r

Critical value

$\alpha = 0.01$

Immediate response by Overall Model test

Up to $14 \times 10^6$

Number of epochs

$= \text{full hour period}$
RAIM - Overall model test

RAIM: detect and correct for errors in GPS data at receiver

Overall model test: does $H_0$ provide good model?

Model: $\underline{y} = A\underline{x} + \underline{e}$

Measurements: $y \rightarrow \hat{x} = \left( A^T Q_{yy}^{-1} A \right)^{-1} A^T Q_{yy}^{-1} y$

$\hat{y} = A\hat{x}$

Residuals: $\hat{e} = y - \hat{y}$

“Mismatch”: $\sum_{q=m-n} = \hat{e}_o^T Q_{yy}^{-1} \hat{e}_o$
RAIM - Overall model test

RAIM: detect and correct for errors in GPS data @receiver

Overall model test: does $H_o$ provide good model?

\[
\overline{T}_q = m - n = \hat{e}_o^T Q_{yy}^{-1} \hat{e}_o
\]

$H_o : \overline{T}_q \sim \chi^2(m - n, 0)$

$H_a : y \in \mathbb{R}^m$
A Simple Ambiguity Resolution Example

Model:

\[
E(p) = \begin{pmatrix} 1 - \lambda & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \rho \\ a \end{pmatrix}, \quad
d(\varphi) = \begin{pmatrix} \sigma_\varphi^2 & 0 \\ 0 & \sigma_p^2 \end{pmatrix}, \quad a = \text{integer}
\]
Relative Positioning: Double Differencing

- elimination of **receiver clock errors**
- elimination of **initial receiver phase offsets**
- DD phase ambiguity is an **integer** number!

**DD code observation:**

$$\rho_{ur,i}^{(kl)} = \left(-1_r^{(kl)}\right)^T x_{ur} + \mu_i I_{ur}^{(kl)} + T_{ur}^{(kl)} + \varepsilon_{\rho_i,ur}^{(kl)}$$

**DD phase observation:**

$$\Phi_{ur,i}^{(kl)} = \left(-1_r^{(kl)}\right)^T x_{ur} - \mu_i I_{ur}^{(kl)} + T_{ur}^{(kl)} + \lambda_i N_{ur,i}^{(kl)} + \varepsilon_{\Phi_i,ur}^{(kl)}$$
Resolution of the DD ambiguities

- code observation: $\text{dm}$ precision

- phase observation: $\text{mm}$ precision,
  - but: $\text{receiver-satellite geometry}$ has to change considerably (long observation time) to solve position with mm-cm accuracy

- if DD ambiguities are resolved to $\text{integers}$ within a short time (or instantaneously), positions (and other parameters) can be solved with mm-cm accuracy
Precision of relative GPS positioning

two epochs of data: varying time span

code observations

phase observations

float

fixed

Satellite Navigation (AE4E08) – Lecture 12
Precision code vs. phase observations

- both RELATIVE positioning
- phase: provided that the integer ambiguity is KNOWN
Ionosphere-fixed, -float, -weighted model

- **Ionosphere-fixed model:**
  - no differential ionospheric delay parameters
  - observations may be corrected a priori for ionosphere
  - for short baselines only
  - can already be based on single-frequency data

- **Ionosphere-float model:**
  - estimation of differential ionospheric delays
  - no a priori corrections
  - for long baselines
  - based on at least dual-frequency data

- **Ionosphere-weighted model:**
  - ionosphere corrections from network RTK ‘subtracted’
  - for medium to long baselines
**GNSS model**

Observation equations:

\[
y = Aa + Bb + e, \quad a \in \mathbb{Z}^n; \quad Q_{yy}
\]

- \( y \) data vector
- \( a \) ambiguities
- \( b \) baseline coordinates & other unknowns
- \( Q_{yy} \) variance-covariance matrix of data
SUCCESSFUL INTEGER AMBIGUITY RESOLUTION

is the key to

FAST and PRECISE GNSS parameter estimation

(baseline coordinates, attitude angles, orbit parameters, atmospheric delays)
Integer estimation

- ‘float’ solution
  - estimate position and carrier ambiguities
  - validate float solution
- integer map
  - estimate integer ambiguities
- ‘fixed’ solution
  - estimate position (ambiguities fixed)
  - validate integer ambiguities
Float and fixed solution

Ambiguities not fixed

Ambiguities fixed
**Integer estimation**

integer map \( \hat{a} \in \mathbb{R}^n \rightarrow S(\hat{a}) = \tilde{a} \in \mathbb{Z}^n \)

no holes & no overlap
\( \Rightarrow \) there will always be **ONE** solution

translation invariant
Different choices of integer estimators

integer rounding

integer bootstrapping

integer least-squares

after their pull-in region
Ambiguity resolution

Integer ambiguities are derived from stochastic observations

Integer ambiguities are not deterministic but stochastic

\[ \hat{\mathbf{a}} \in \mathbb{R}^n \rightarrow S(\hat{\mathbf{a}}) = \tilde{\mathbf{a}} \in \mathbb{Z}^n \]

input (stochastic) \rightarrow \text{output (stochastic)}
Integer estimation

Optimal integer estimator: integer least-squares

\[ \tilde{\mathbf{a}} = \arg \min_{\mathbf{z} \in \mathbb{Z}^n} \| \hat{\mathbf{a}} - \mathbf{z} \|^2_{Q\hat{\mathbf{a}}} \]
Ambiguity search space: a (hyper-) ellipsoid

- **centered** at \( \hat{\mathbf{a}} \)
- **shape** governed by \( Q_{\hat{\mathbf{a}}\hat{\mathbf{a}}} \)
- find all integers \( z \) for which
  \[
  (\hat{\mathbf{a}} - z)^T Q_{\hat{\mathbf{a}}\hat{\mathbf{a}}}^{-1} (\hat{\mathbf{a}} - z) \leq \chi^2
  \]

- \( \chi^2 \) should be set such that search space contains at least one integer vector

- select the \( z \) which provides minimum

\[ a_1, a_2 \] (float solution)

○ candidate integer solution
Integer ambiguity resolution

- Float solution: least-squares
- Integer search: find integer solution with shortest weighted distance to float solution (weighted by variance-covariance matrix of float ambiguities)
- Search difficult due to correlations
- LAMBDA: transformation of search space to make it efficient
Example: Ambiguity search space
Two dimensions, geometry-free, short baseline

16 candidates for $a_1$, but only one for $a_2$ such that $(a_1, a_2)$ inside search space

→ Search is inefficient
Example: Ambiguity search space
Two dimensions, geometry-free, short baseline

After decorrelation
Number of candidates INSIDE search space is same

→ *Search is efficient*
Ambiguity estimation and success rate
Example based on real data (1000 epochs)

Distribution of original ambiguities

Distribution of transformed ambiguities

Success rate: 99%
= % of float ambiguities in pull-in region centered at correct integer value
Integer ambiguity resolution

Successful ambiguity resolution depends on precision of float solution, which depends on:

- baseline length (tropo + iono delays)
- satellite geometry
- precision of code and phase observations
- # frequencies

→ Change in satellite geometry helps (long duration)
LAMBDA method

Integer estimation:
• optimal : maximum success rate
• efficient : (near) real-time

Applicable to wide variety of models
• With or without relative satellite-receiver geometry
• Stationary or moving receivers
• With or without atmospheric delays
• Single- or multi-baseline
• One, two, three or more frequencies (any GNSS)
## Baseline models

### Parameters

<table>
<thead>
<tr>
<th></th>
<th>Geometry-free</th>
<th>Roving-receiver</th>
<th>Stationary-receiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ranges</td>
<td>N</td>
<td>N</td>
<td>C</td>
</tr>
<tr>
<td>Station coordinates</td>
<td>N</td>
<td>N</td>
<td>C</td>
</tr>
<tr>
<td>Ambiguities</td>
<td>C</td>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>Ionospheric delays</td>
<td>N*</td>
<td>N*</td>
<td>N*</td>
</tr>
</tbody>
</table>

N - New parameter introduced for each observation epoch  
C - Constant parameter for entire observation period  
*) - Long baselines only
Ambiguity Resolution Methods

- Search in the 3-dimensional position space (e.g. ambiguity function method); *Now deprecated*

- Linear combination of code and phase (using widelane/narrowlane combinations)
  - performance worse with AS
  - has been improved by LAMBDA: 2-dimensional ambiguity resolution/search problem
    → **Geometry-free model**
  - Search in the \( n \)-dimensional ambiguity space
    → **Geometry-based model**
Summary and outlook

- We covered it all! (except for the applications)

Next:
Applications: your presentations

Exam preparation: check blackboard!