Numerical Linear Algebra in the Global Positioning System

Xiao-Wen Chang

Joint work with Chris Paige

School of Computer Science

McGill University, Montreal, QC, Canada

It is now possible to find out where you are.

Gilbert Strang
The Mathematics of GPS,
SIAM News of June 1997

Outline

- Computation of GPS relative position estimates
  - Introduction to GPS
  - The mathematical model
  - A recursive least squares method
  - Real data tests
- Computation of a test statistic in data quality control
  - Generalized likelihood ratio test statistic
  - A numerically stable method

Part I:

Computation of GPS Relative Position Estimates

Collaborator: Lan Yin, Cimmetry Systems, Inc., Montreal.

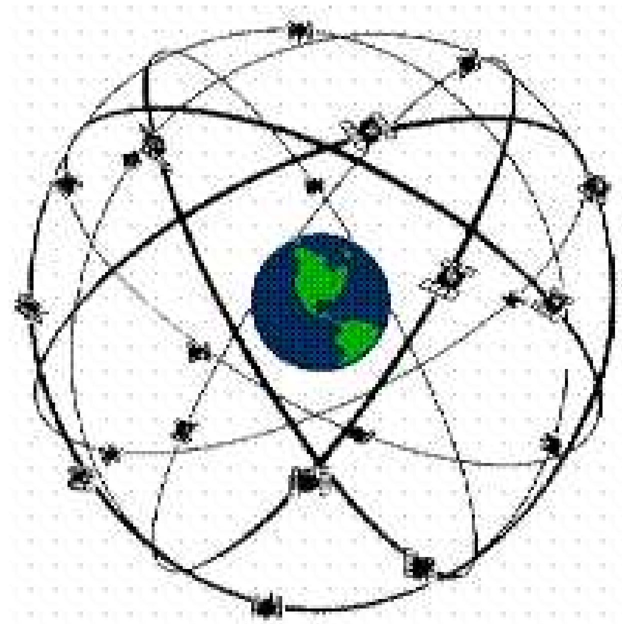
Introduction to GPS (6)

What Is the Global Positioning System (GPS) ?

An all-weather, worldwide, continuous coverage, satellite based navigation system, operated by the U. S. military.

GPS Segments:

- **Space:** 24 Satellites, in 6 orbits at 20,200 km altitude.
- **Control:** 1 master control station, 6 monitor stations, 4 ground antennas.
- **User:** receivers and users.

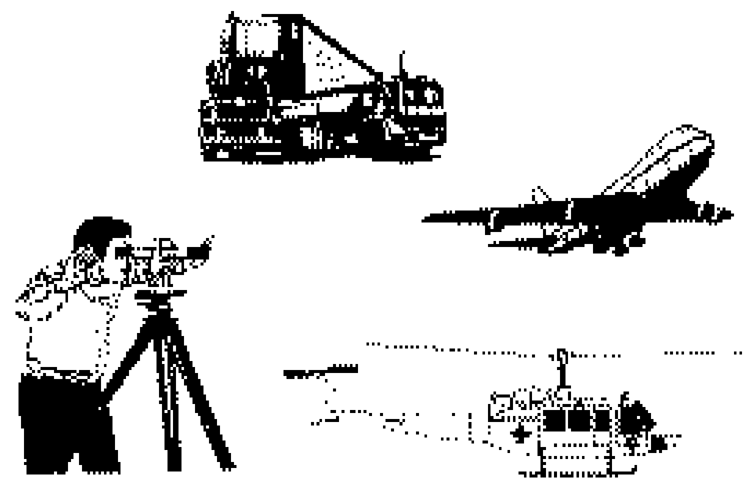


GPS Applications

GPS Military Uses:

GPS has become important for nearly all military operations and weapons systems.

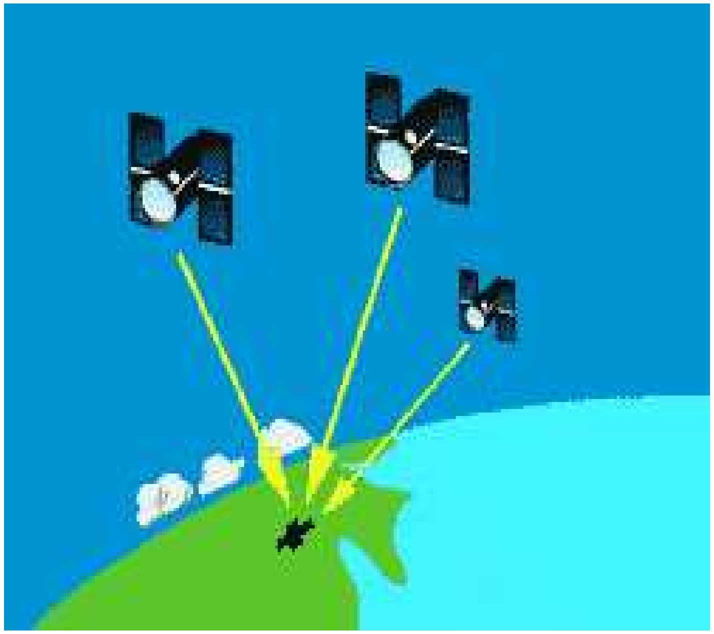
GPS Civil Uses:



- Land/sea/air/space navigation
- Mapping / GIS
- Surveying
- Search and rescue
- Recreation
- Intelligent vehicle highway systems, ...

How GPS Works

- The basis of GPS is "trilateration".
- In principle, if the distances from 3 satellites to a receiver can be measured, the receiver position can be determined.



GPS Signal & Measurements:

Each satellite transmits signals on two frequencies: L1 & L2. Superimposed on the carriers are C/A, P codes & navigation data.

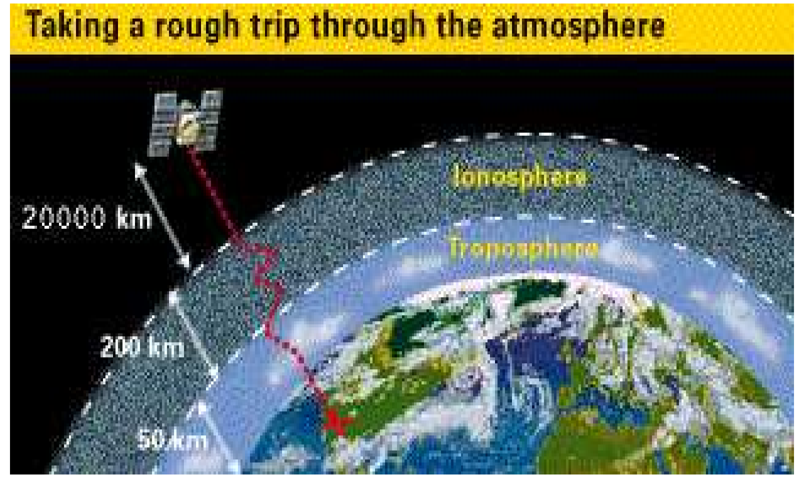
Measurements:

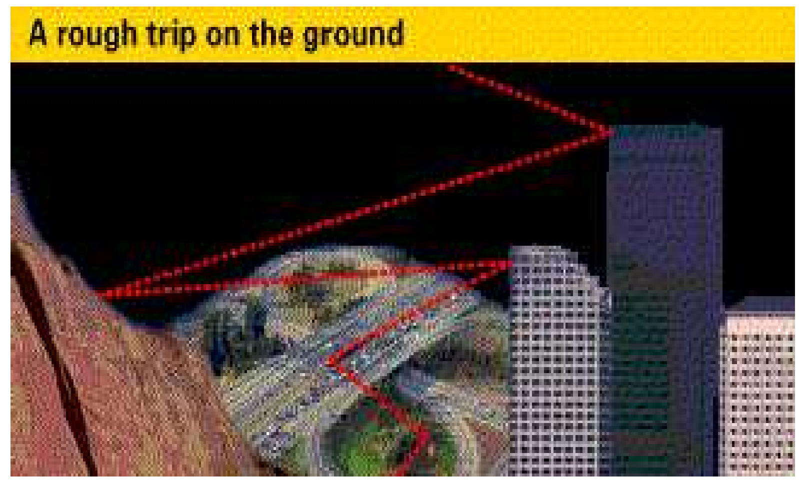
• Code measurements; • carrier phase measurements.

Carrier phase measurements are much more **complicated** but much more **accurate** than code measurements.

GPS Signal Errors

- Satellite clock errors
- Satellite hardware delay
- Satellite orbit errors
- Ionospheric reflection
- Tropospheric reflection
- Multipath errors
- Receiver clock errors
- Receiver hardware delay
- Noise errors

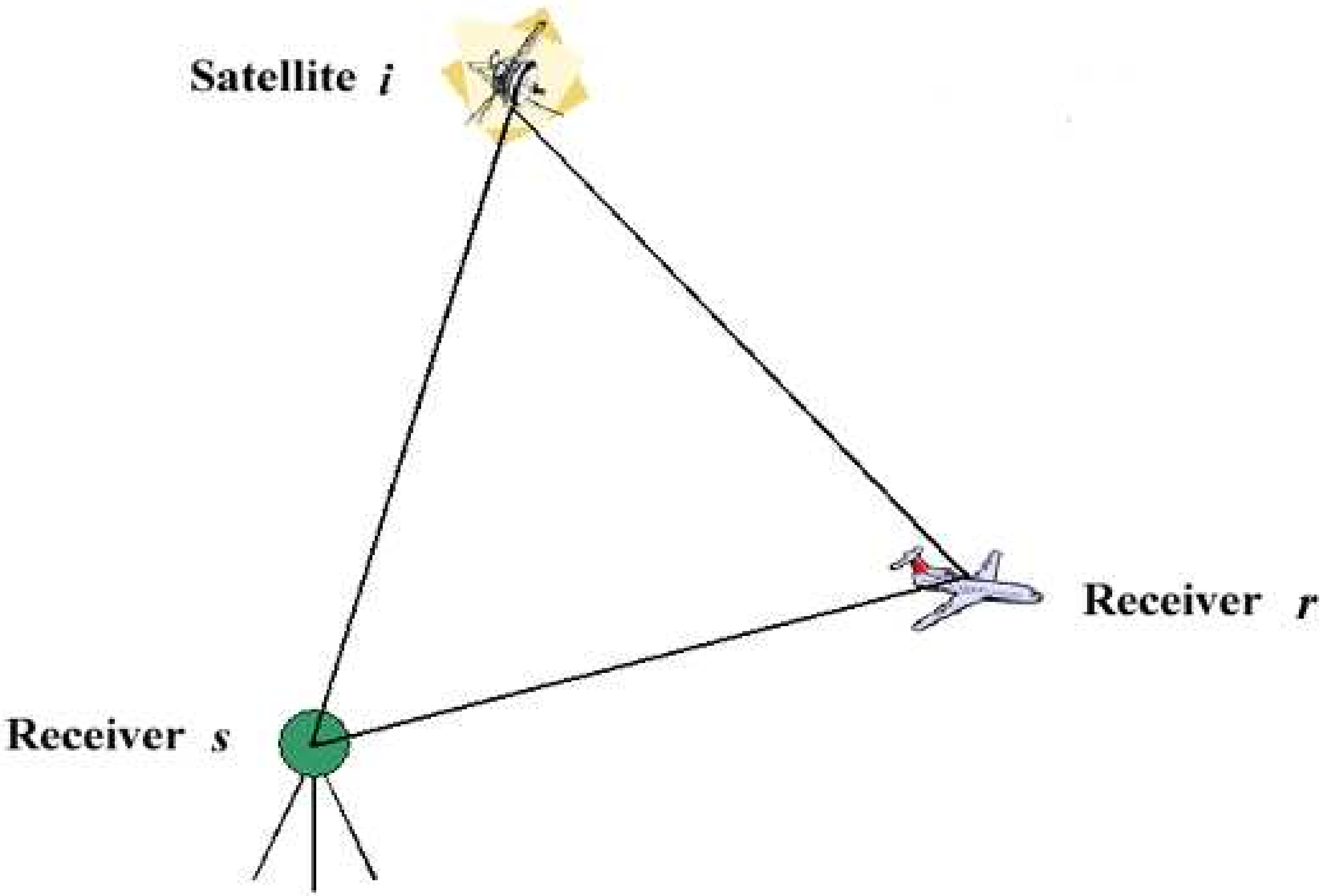




The multipath errors and noise errors of **code** measurements are usually $\sim 100\times$ larger than those of **carrier phase** measurements.

Physical Setting

- **Point positioning**: (simple, but precision is lower)
  - One receiver.
- **Relative positioning**: (complicated, but precision is high)
  - One stationary receiver with a known position; one (roving) receiver with a position to be determined.
  - They track the GPS signals at the same time.
  - They are close enough (10 km say) — the received signals have almost the **same** ionospheric and tropospheric errors.
  - For real time applications, there is a radio link between them.



Geometry for two Receivers and one Satellite.

Motivations and Goal

- Typical approach used in GPS:
  Least squares based on measurement eqn. $\boldsymbol{y}_k = A_k \boldsymbol{x}_k + \boldsymbol{v}_k$;
  Kalman filtering based on

  $$\boldsymbol{y}_k = A_k \boldsymbol{x}_k + \boldsymbol{v}_k \text{ (meas eqn)}, \quad \boldsymbol{x}_{k+1} = F_k \boldsymbol{x}_k + \boldsymbol{w}_k \text{ (dynamic eqn)}$$

  Sometimes the dynamic eqns are **artificially constructed**.

- Most methods given in GPS literature do not address the computer implementation issues.

- Efficiency is important—particularly for real time applications.
  Numerical reliability is important—ill-conditioned problems may arise.

Goal: present an efficient and numerically reliable approach for relative positioning.

The Mathematical Model (3)

Different combinations of measurements can be used for point and relative positioning.

We consider **relative positioning** based on **code and carrier phase measurements** from **L1** signal.

Single difference (SD) technique:
Difference the carrier (and code) measurements from the same satellite at two receivers to eliminate some common errors: *satellite clock error, satellite hardware delay, satellite orbit error, ionospheric reflection and tropospheric reflection.*

Single Difference (SD) Measurement Equations

At time t_k, for signal from satellite i

$$
\begin{aligned}
\text{carrier:}\quad & \phi_k^i = \lambda^{-1}(\boldsymbol{e}_k^i)^T\boldsymbol{x}_k + \alpha^i + \beta_k^\phi + \mu_k^i, && \mu_k^i \sim \mathcal{N}(0, \sigma_\phi^2) \\
\text{code:}\quad & \rho_k^i = \lambda^{-1}(\boldsymbol{e}_k^i)^T\boldsymbol{x}_k \qquad\quad + \beta_k^\rho + \nu_k^i, && \nu_k^i \sim \mathcal{N}(0, \sigma_\rho^2)
\end{aligned}
$$

ϕ_k^i: SD carrier phase measurement; ρ_k^i: SD code measurement;

λ: wavelength; μ_k^i: carrier phase noise; ν_k^i: code noise;

$\boldsymbol{x}_k$: **baseline** vector from receivers s to r, *to be determined*;

$\boldsymbol{e}_k^i$: unit vector from the midpoint of $\boldsymbol{x}_k$ to satellite i;

α^i: **SD integer ambiguity**, *constant but unknown*;

β_k^ϕ: SD receiver clock error and hardware delay for **carrier**;

β_k^ρ: SD receiver clock error and hardware delay for **code**.

Single Difference Measurement Equations, cont

$$\begin{array}{lll} \text{carrier} & \phi_k^i = \lambda^{-1}(\boldsymbol{e}_k^i)^T \boldsymbol{x}_k + \alpha^i + \beta_k^\phi + \mu_k^i, & \mu_k^i \sim \mathcal{N}(0, \sigma_\phi^2) \\ \text{code} & \rho_k^i = \lambda^{-1}(\boldsymbol{e}_k^i)^T \boldsymbol{x}_k + \beta_k^\rho + \nu_k^i, & \nu_k^i \sim \mathcal{N}(0, \sigma_\rho^2) \end{array}$$

Suppose there are m visible satellites. Then we have $2m$ such eqs. Write these in the matrix-vector form ($\boldsymbol{e} = [1, \ldots, 1]^T$):

$$\begin{array}{lll} \text{carrier} & \boldsymbol{y}_k^\phi = \boldsymbol{E}_k \, \boldsymbol{x}_k + \boldsymbol{a} + \boldsymbol{e} \, \beta_k^\phi + \boldsymbol{v}_k^\phi, & \boldsymbol{v}_k^\phi \sim \mathcal{N}(\boldsymbol{0}, \sigma_\phi^2 I_m) \\ \text{code} & \boldsymbol{y}_k^\rho = \boldsymbol{E}_k \, \boldsymbol{x}_k + \boldsymbol{e} \, \beta_k^\rho + \boldsymbol{v}_k^\rho, & \boldsymbol{v}_k^\rho \sim \mathcal{N}(\boldsymbol{0}, \sigma_\rho^2 I_m) \end{array}$$

Usually assume $\boldsymbol{v}_k^\phi$ and $\boldsymbol{v}_l^\rho$ $(k, l = 1, 2, \ldots)$ are uncorrelated.

Note: $\boldsymbol{E}_k$ depends on $\boldsymbol{x}_k$, so the model is nonlinear.
Use our estimate of $\boldsymbol{x}_{k-1}$ to compute an approximation to $\boldsymbol{E}_k$ — usually good enough. From now on, assume $\boldsymbol{E}_k$ is known.

A Recursive LS Method (10)

Background for the LS estimation

A standard linear model (A has full column rank):

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{v}, \quad \boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}),$$

The best linear unbiased estimate (BLUE) of $\boldsymbol{x}$ solves

$$\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2.$$

Solve the LS problem by (Householder) QR factn (see Golub '65):

$$\boldsymbol{Q}^T \boldsymbol{A} = \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{0} \end{bmatrix}, \quad \boldsymbol{Q} = [\boldsymbol{Q}_1, \boldsymbol{Q}_2] \text{ orthogonal}, \quad \boldsymbol{R} \text{ upper triangular}$$

The LS solution and residual satisfy

$$\hat{\boldsymbol{x}} = \boldsymbol{R}^{-1}\boldsymbol{Q}_1^T \boldsymbol{y}, \qquad \mathcal{E}\{\hat{\boldsymbol{x}}\} = \boldsymbol{x}, \qquad \operatorname{cov}\{\hat{\boldsymbol{x}}\} = \sigma^2(\boldsymbol{R}^T\boldsymbol{R})^{-1}$$

$$\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{Q}_2\boldsymbol{Q}_2^T\boldsymbol{y}, \qquad \mathcal{E}\{\boldsymbol{r}\} = \boldsymbol{0}, \qquad \operatorname{cov}\{\boldsymbol{r}\} = \sigma^2 \boldsymbol{Q}_2^T\boldsymbol{Q}_2.$$

Orthogonal transformation approach for position estimation

Use orthogonal transformations to recursively estimate $\boldsymbol{x}_k$ based on the model

$$
\begin{aligned}
\text{carrier} \quad & \boldsymbol{y}_k^{\phi} = \boldsymbol{E}_k\,\boldsymbol{x}_k + \boldsymbol{a} + \boldsymbol{e}\,\beta_k^{\phi} + \boldsymbol{v}_k^{\phi}, \quad && \boldsymbol{v}_k^{\phi} \sim \mathcal{N}(\mathbf{0}, \sigma_{\phi}^2 I_m), \\
\text{code} \quad & \boldsymbol{y}_k^{\rho} = \boldsymbol{E}_k\,\boldsymbol{x}_k + \boldsymbol{e}\,\beta_k^{\rho} + \boldsymbol{v}_k^{\rho}, && \boldsymbol{v}_k^{\rho} \sim \mathcal{N}(\mathbf{0}, \sigma_{\rho}^2 I_m).
\end{aligned}
$$

$$k = 1, 2, \ldots.$$

- Orthogonal transformations are numerically reliable.
- Orthogonal transformations can keep the noise vectors uncorrelated.

Eliminating β_k^ϕ and β_k^ρ from the model:

Let $\boldsymbol{P}$ be a Householder transformation: $\boldsymbol{P}\boldsymbol{e} = \sqrt{m}\,[1, 0, ..., 0]^T$.

Apply $\quad \boldsymbol{P} \equiv \begin{bmatrix} \boldsymbol{p}^T \\ \bar{\boldsymbol{P}} \end{bmatrix}$ to $\quad \boldsymbol{y}_k^\phi = \boldsymbol{E}_k\,\boldsymbol{x}_k + \boldsymbol{a} + \boldsymbol{e}\,\beta_k^\phi + \boldsymbol{v}_k^\phi$:

$$\begin{bmatrix} \boldsymbol{p}^T \boldsymbol{y}_k^\phi \\ \bar{\boldsymbol{P}}\boldsymbol{y}_k^\phi \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}^T \boldsymbol{E}_k \\ \bar{\boldsymbol{P}}\boldsymbol{E}_k \end{bmatrix} \boldsymbol{x}_k + \begin{bmatrix} \boldsymbol{p}^T \\ \bar{\boldsymbol{P}} \end{bmatrix} \boldsymbol{a} + \begin{bmatrix} 1 \\ \boldsymbol{0} \end{bmatrix} \sqrt{m}\beta_k^\phi + \begin{bmatrix} \boldsymbol{p}^T \boldsymbol{v}_k^\phi \\ \bar{\boldsymbol{P}}\boldsymbol{v}_k^\phi \end{bmatrix}.$$

Drop the 1st equation

$$\bar{\boldsymbol{P}}\boldsymbol{y}_k^\phi = \bar{\boldsymbol{P}}\boldsymbol{E}_k\boldsymbol{x}_k + \bar{\boldsymbol{P}}\boldsymbol{a} + \bar{\boldsymbol{P}}\boldsymbol{v}_k^\phi, \qquad \bar{\boldsymbol{P}}\boldsymbol{v}_k^\phi \sim \mathcal{N}(\boldsymbol{0}, \sigma_\phi^2 \boldsymbol{I}_{m-1}).$$

Since $\bar{\boldsymbol{P}}$ is $(m-1) \times m$, $\quad \boldsymbol{a} \in \Re^m \quad$ cannot be determined.

Introduce the double difference **integer** ambiguity vector

$$\boldsymbol{z} \equiv [\alpha^2 - \alpha^1, \alpha^3 - \alpha^1, \ldots, \alpha^m - \alpha^1]^T \in \Re^{m-1},$$

where satellite 1 is chosen to be the **'reference satellite'**.

Can show

$$\bar{\boldsymbol{P}}\boldsymbol{a} = \boldsymbol{F}\boldsymbol{z}, \quad \boldsymbol{F} = \boldsymbol{I}_{m-1} - \frac{1}{m-\sqrt{m}}\boldsymbol{e}\boldsymbol{e}^T \text{ nonsingular.}$$

So $\bar{\boldsymbol{P}}\boldsymbol{y}_k^{\phi} = \bar{\boldsymbol{P}}\boldsymbol{E}_k\boldsymbol{x}_k + \bar{\boldsymbol{P}}\boldsymbol{a} + \bar{\boldsymbol{P}}\boldsymbol{v}_k^{\phi}$ becomes

$$\bar{\boldsymbol{P}}\boldsymbol{y}_k^{\phi} = \bar{\boldsymbol{P}}\boldsymbol{E}_k\boldsymbol{x}_k + \boldsymbol{F}\boldsymbol{z} + \bar{\boldsymbol{P}}\boldsymbol{v}_k^{\phi}, \qquad \bar{\boldsymbol{P}}\boldsymbol{v}_k^{\phi} \sim \mathcal{N}(\boldsymbol{0}, \sigma_\phi^2 \boldsymbol{I}_{m-1}). \tag{1}$$

Similarly, applying $\bar{\boldsymbol{P}}$ to $\boldsymbol{y}_k^{\rho} = \boldsymbol{E}_k\boldsymbol{x}_k + \boldsymbol{e}\,\beta_k^{\rho} + \boldsymbol{v}_k^{\rho}$ gives

$$\bar{\boldsymbol{P}}\boldsymbol{y}_k^{\rho} = \bar{\boldsymbol{P}}\boldsymbol{E}_k\boldsymbol{x}_k + \bar{\boldsymbol{P}}\boldsymbol{v}_k^{\rho}, \qquad \bar{\boldsymbol{P}}\boldsymbol{v}_k^{\rho} \sim \mathcal{N}(\boldsymbol{0}, \sigma_\rho^2 \boldsymbol{I}_{m-1}). \tag{2}$$

Let $\sigma = \frac{\sigma_\phi}{\sigma_\rho}$. Combine (1) and (2):

$$\begin{bmatrix} \bar{\boldsymbol{P}}\boldsymbol{y}_k^{\phi} \\ \sigma\bar{\boldsymbol{P}}\boldsymbol{y}_k^{\rho} \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{P}}\boldsymbol{E}_k \\ \sigma\bar{\boldsymbol{P}}\boldsymbol{E}_k \end{bmatrix}\boldsymbol{x}_k + \begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{0} \end{bmatrix}\boldsymbol{z} + \begin{bmatrix} \bar{\boldsymbol{P}}\boldsymbol{v}_k^{\phi} \\ \sigma\bar{\boldsymbol{P}}\boldsymbol{v}_k^{\rho} \end{bmatrix}, \quad \begin{bmatrix} \bar{\boldsymbol{P}}\boldsymbol{v}_k^{\phi} \\ \sigma\bar{\boldsymbol{P}}\boldsymbol{v}_k^{\rho} \end{bmatrix} \sim \mathcal{N}(\boldsymbol{0}, \sigma_\phi^2 \boldsymbol{I})$$

For simplicity, ignore the noise vector and use '≈' to replace '='.

$$\begin{bmatrix} \bar{\boldsymbol{P}}\boldsymbol{y}_k^{\phi} \\ \sigma\bar{\boldsymbol{P}}\boldsymbol{y}_k^{\rho} \end{bmatrix} \approx \begin{bmatrix} \bar{\boldsymbol{P}}\boldsymbol{E}_k \\ \sigma\bar{\boldsymbol{P}}\boldsymbol{E}_k \end{bmatrix} \boldsymbol{x}_k + \begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{z} \tag{3}$$

Transform the coefficient matrix of $\boldsymbol{x}_k$ to upper triangular

Compute the QR factorization:

$$\boldsymbol{Q}_k^T \begin{bmatrix} \bar{\boldsymbol{P}}\boldsymbol{E}_k \\ \sigma\bar{\boldsymbol{P}}\boldsymbol{E}_k \end{bmatrix} = \begin{array}{cl} \begin{bmatrix} \boldsymbol{R}_k \\ \boldsymbol{0} \end{bmatrix} & \begin{array}{l} 3 \\ 2m-5 \end{array} \\ 3 & \end{array}$$

Note: Make full use of the structure of the matrix for efficiency.

Then apply $\boldsymbol{Q}_k^T$ to (3), with obvious notation:

$$\begin{bmatrix} \boldsymbol{y}_k \\ \bar{\boldsymbol{y}}_k \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{R}_k \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{x}_k + \begin{bmatrix} \boldsymbol{G}_k \\ \bar{\boldsymbol{G}}_k \end{bmatrix} \boldsymbol{z}$$

$$\begin{bmatrix} \boldsymbol{y}_k \\ \bar{\boldsymbol{y}}_k \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{R}_k \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{x}_k + \begin{bmatrix} \boldsymbol{G}_k \\ \bar{\boldsymbol{G}}_k \end{bmatrix} \boldsymbol{z}$$

Combine these for $k = 1, 2, \ldots,$ and reorder:

$$\left[\begin{array}{c} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_k \\ \hline \bar{\boldsymbol{y}}_1 \\ \vdots \\ \bar{\boldsymbol{y}}_k \end{array}\right] \approx \left[\begin{array}{ccc|c} \boldsymbol{R}_1 & & & \boldsymbol{G}_1 \\ & \ddots & & \vdots \\ & & \boldsymbol{R}_k & \boldsymbol{G}_k \\ \hline & & & \bar{\boldsymbol{G}}_1 \\ & & & \vdots \\ & & & \bar{\boldsymbol{G}}_k \end{array}\right] \left[\begin{array}{c} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_k \\ \hline \boldsymbol{z} \end{array}\right] \tag{4}$$

First estimate $\boldsymbol{z}$ from the lower part of (4);
then estimate $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ from the upper part of (4).

Recursively computing the LS estimate of z $\begin{bmatrix} \bar{G}_1 \\ \vdots \\ \bar{G}_k \end{bmatrix} z \approx \begin{bmatrix} \bar{y}_1 \\ \vdots \\ \bar{y}_k \end{bmatrix}$

Suppose at t_{k-1} we have obtained the orthogonal transformations:

$$U_{k-1}^T \begin{bmatrix} \bar{G}_1 \\ \vdots \\ \bar{G}_{k-1} \end{bmatrix} = \begin{bmatrix} S_{k-1} \\ \hline 0 \end{bmatrix}, \qquad U_{k-1}^T \begin{bmatrix} \bar{y}_1 \\ \vdots \\ \bar{y}_{k-1} \end{bmatrix} = \begin{bmatrix} b_{k-1} \\ \hline \bar{b}_{k-1} \end{bmatrix},$$

U_{k-1}: orthogonal, S_{k-1}: nonsingular upper triangular.

At t_k, after obtaining $\bar{G}_k$ and $\bar{y}_k$, perform

$$\tilde{U}_k^T \begin{bmatrix} S_{k-1} \\ \bar{G}_k \end{bmatrix} = \begin{bmatrix} S_k \\ \hline 0 \end{bmatrix}, \qquad \tilde{U}_k^T \begin{bmatrix} b_{k-1} \\ \bar{y}_k \end{bmatrix} = \begin{bmatrix} b_k \\ \hline \hat{b}_k \end{bmatrix}$$

$\tilde{U}_k$: orthogonal, S_k : nonsingular upper triangular.

Thus $\begin{bmatrix} \bar{\boldsymbol{G}}_1 \\ \vdots \\ \bar{\boldsymbol{G}}_k \end{bmatrix} \boldsymbol{z} \approx \begin{bmatrix} \bar{\boldsymbol{y}}_1 \\ \vdots \\ \bar{\boldsymbol{y}}_k \end{bmatrix}$ is transformed to $\begin{bmatrix} \boldsymbol{S}_k \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{z} \approx \begin{bmatrix} \boldsymbol{b}_k \\ \hat{\boldsymbol{b}}_k \end{bmatrix}$.

Compute the LS estimate $\boldsymbol{z}_k$ of $\boldsymbol{z}$ at t_k by solving

$$\boldsymbol{S}_k \boldsymbol{z}_k = \boldsymbol{b}_k.$$

It is easy to show

$$\mathrm{cov}\{\boldsymbol{z}_k\} = \sigma_\phi^2 (\boldsymbol{S}_k^T \boldsymbol{S}_k)^{-1}.$$

Remarks:

- Here we regarded $\boldsymbol{z}$ as a real vector.
- To get centimeter accuracy quickly, we have to fix $\boldsymbol{z}$ as a vector of integers. Then $\boldsymbol{z}$ will be regarded as known.
- The LAMBDA method (Teunissen '93) uses $\mathrm{cov}\{\boldsymbol{z}_k\}$ to fix $\boldsymbol{z}$.

Computing the LS estimates of $\boldsymbol{x}_1, \ldots \boldsymbol{x}_k$

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_k \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{R}_1 & & \\ & \ddots & \\ & & \boldsymbol{R}_k \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_k \end{bmatrix} + \begin{bmatrix} \boldsymbol{G}_1 \\ \vdots \\ \boldsymbol{G}_k \end{bmatrix} \boldsymbol{z}$$

We compute $\boldsymbol{x}_{1|k}, \boldsymbol{x}_{2|k}, \ldots, \boldsymbol{x}_{k|k}$, the LS estimates of $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k$ at time t_k by solving the upper triangular systems

$$\boldsymbol{R}_j \boldsymbol{x}_{j|k} = \boldsymbol{y}_j - \boldsymbol{G}_j \boldsymbol{z}_k, \qquad j = 1, \ldots, k.$$

Remarks:

- These can be solved in any order.
- $\boldsymbol{x}_{j|k}$ for $j \leq k-1$ is called the smoothed estimate of $\boldsymbol{x}_j$.
- For real time applications, we may only want to find $\boldsymbol{x}_{k|k}$.

Computing the error covariance matrices $\mathrm{cov}\{\boldsymbol{x}_{j|k} - \boldsymbol{x}_j\}$

The equations used for estimating $\boldsymbol{x}_j$ and $\boldsymbol{z}$ at time t_k:

$$\begin{bmatrix} \boldsymbol{y}_j \\ \boldsymbol{b}_k \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_j & \boldsymbol{G}_j \\ \boldsymbol{0} & \boldsymbol{S}_k \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_j \\ \boldsymbol{z} \end{bmatrix} + \underbrace{\begin{bmatrix} \boldsymbol{u}_j \\ \boldsymbol{w}_k \end{bmatrix}}_{\text{noise}}, \quad \begin{bmatrix} \boldsymbol{u}_j \\ \boldsymbol{w}_k \end{bmatrix} \sim \mathcal{N}(\boldsymbol{0}, \sigma_\phi^2 I). \tag{5}$$

Let $\boldsymbol{Z}_{j|k}^T \begin{bmatrix} \boldsymbol{R}_j & \boldsymbol{G}_j \\ \boldsymbol{0} & \boldsymbol{S}_k \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{j|k} & \boldsymbol{0} \\ \bar{\boldsymbol{R}}_{j|k} & \boldsymbol{S}_{j|k} \end{bmatrix}$, $\quad \boldsymbol{Z}_{j|k}$: Givens rotations; $\boldsymbol{R}_{j|k}, \ \boldsymbol{S}_{j|k}$: U.T.

Apply $\boldsymbol{Z}_{j|k}^T$ to (5):

$$\boldsymbol{Z}_{j|k}^T \begin{bmatrix} \boldsymbol{y}_j \\ \boldsymbol{b}_k \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{j|k} & \boldsymbol{0} \\ \boldsymbol{R}_{j|k} & \boldsymbol{S}_{j|k} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_j \\ \boldsymbol{z} \end{bmatrix} + \boldsymbol{Z}_{j|k}^T \begin{bmatrix} \bar{\boldsymbol{u}}_j \\ \boldsymbol{w}_k \end{bmatrix}.$$

Can show

$$\mathrm{cov}\{\boldsymbol{x}_{j|k} - \boldsymbol{x}_j\} = \sigma_\phi^2 (\boldsymbol{R}_{j|k}^T \boldsymbol{R}_{j|k})^{-1}.$$

Remarks (2)

- **Cycle slip**

  Cycle slip is the change in ambiguities mainly caused by temporary obstructions of the satellite signal.
  We need incorporate an algorithm for cycle slip detection.

- **Number of satellites**

  There may be different number of satellites at different epochs, e.g., satellite rising and setting, cycle slip.
  But we can modify our algorithm to handle such cases.

  Two cases:
  a. The 'reference satellite' at t_{k-1} goes down between t_{k-1} & t_k.
  b. The 'reference satellite' at t_{k-1} remains at t_k.

- **Dual frequency receivers**

  Some receivers can measure carrier phase and code for both L1 and L2 carrier signals.

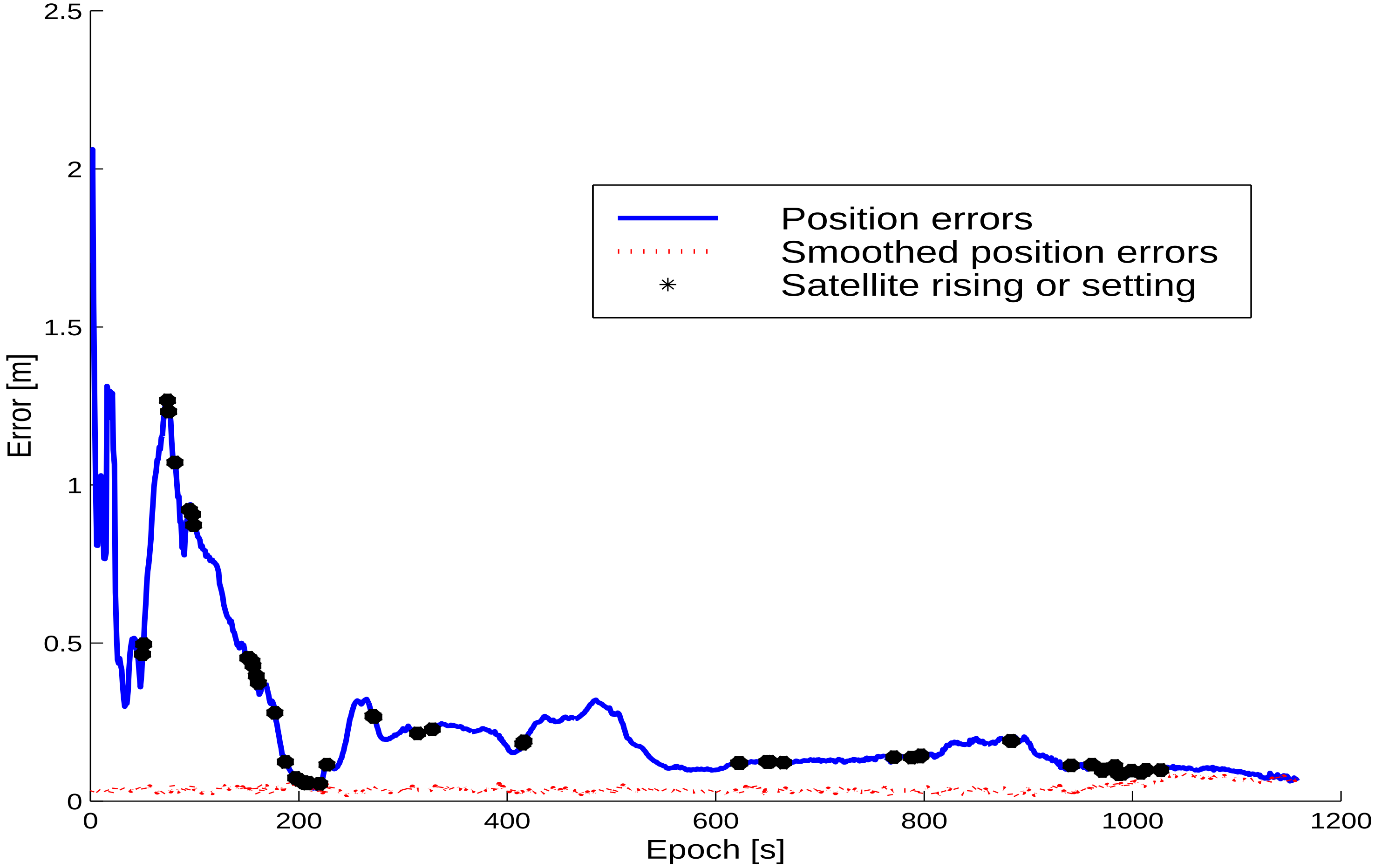
  It is easy to modify our approach to include more measurement equations.

- **Kalman filtering**

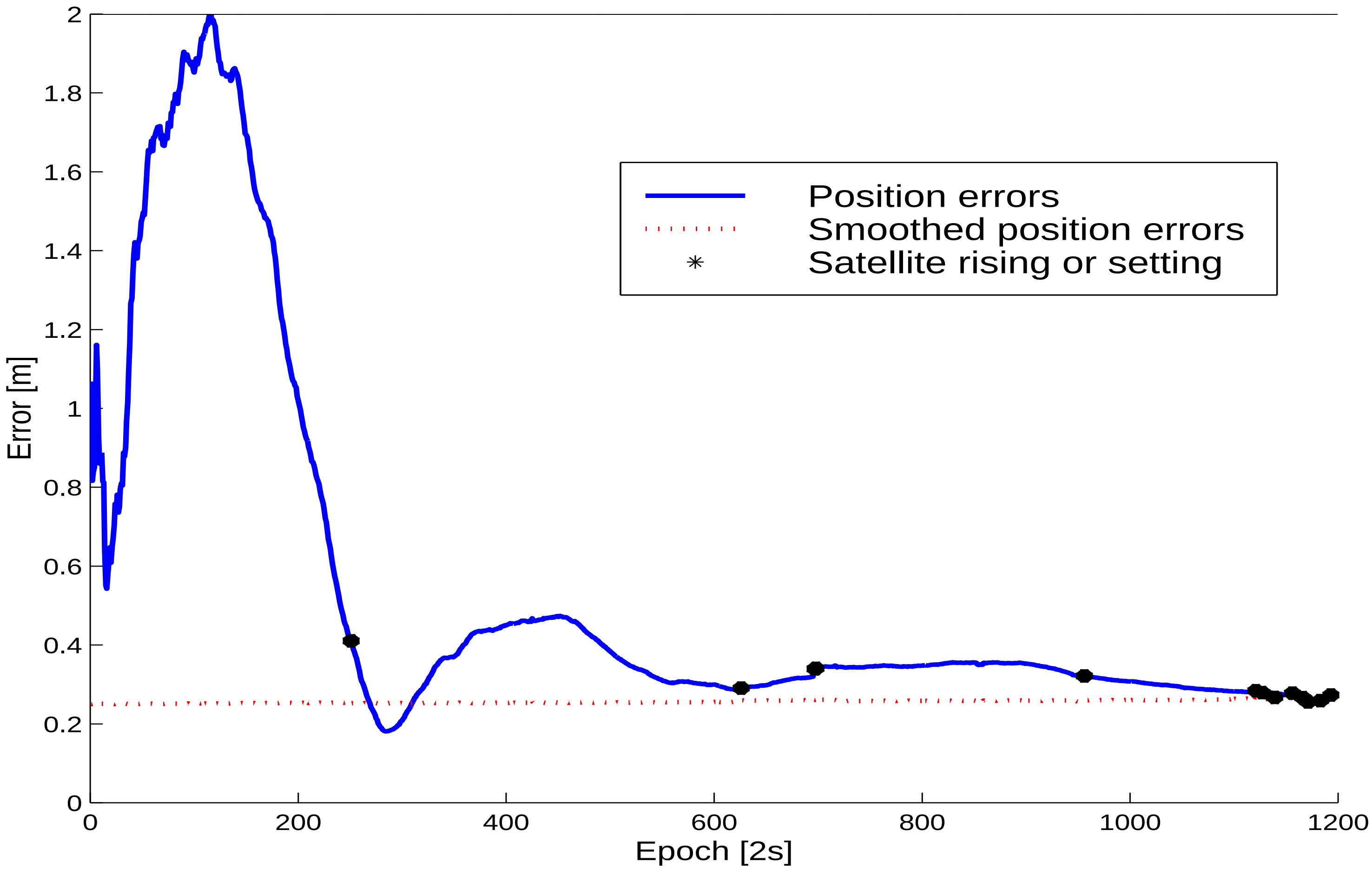
  When a dynamic model for the roving receiver is available, we can modify our approach to handle it.

Real data tests (3)

- Two data sets were provided by VIASAT Geo-Technology Inc.
- The receivers were made by Canadian Marconi Company.
- Data set 1: The user was walking;
  Data set 2: The user was riding a four wheel trail bike;
  Both were in an open sky environment.
- The time interval between two consecutive epochs:
  Data set 1: 1 second, Data set 2: 2 seconds.
- We took $\sigma = \sigma_\phi / \sigma_\rho = 10^{-3}$.
- We used the position estimates obtained by VIASAT software as the "true" positions. The software used a complex positioning algorithm. It is believed the errors are about a few centimeters.



Position errors versus run time for **data set 1**



Position errors versus run time for **data set 2**

Part II:

Computation of a Test Statistic in Data Quality Control

Collaborator: Christian Tiberius, Delft University of Technology.

Data Quality Control

Motivations

- GPS signals may be corrupted.
- Using corrupted data may lead big errors in the position estimates — too dangerous for some applications, such as aircraft landing.
- The data need to be carefully validated by statistical testing.
- Data quality control is useful not only in GPS, but also in many other applications.

Goal:
Provide a numerically stable method to compute the commonly used generalized likelihood ratio test statistic.

Example (2)

Suppose at some epoch, there are 5 visible satellites.

In a normal situation, the code measurements satisfy

$$\mathrm{H}_0 : \quad \underbrace{\begin{bmatrix} \rho^1 \\ \rho^2 \\ \rho^3 \\ \rho^4 \\ \rho^5 \end{bmatrix}}_{\text{meas}} = \begin{bmatrix} \lambda^{-1}(\boldsymbol{e}^1)^T & 1 \\ \lambda^{-1}(\boldsymbol{e}^2)^T & 1 \\ \lambda^{-1}(\boldsymbol{e}^3)^T & 1 \\ \lambda^{-1}(\boldsymbol{e}^4)^T & 1 \\ \lambda^{-1}(\boldsymbol{e}^5)^T & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \beta^\rho \end{bmatrix} + \underbrace{\begin{bmatrix} \nu^1 \\ \nu^2 \\ \nu^3 \\ \nu^4 \\ \nu^5 \end{bmatrix}}_{\text{noise}}.$$

If the signal from satellite 2 has a blunder (outlier), then

$$
\mathrm{H}_a : \quad \begin{bmatrix} \rho^1 \\ \rho^2 \\ \rho^3 \\ \rho^4 \\ \rho^5 \end{bmatrix} = \begin{bmatrix} \lambda^{-1}(\boldsymbol{e}^1)^T & 1 \\ \lambda^{-1}(\boldsymbol{e}^2)^T & 1 \\ \lambda^{-1}(\boldsymbol{e}^3)^T & 1 \\ \lambda^{-1}(\boldsymbol{e}^4)^T & 1 \\ \lambda^{-1}(\boldsymbol{e}^5)^T & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \beta^\rho \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \nabla + \begin{bmatrix} \nu^1 \\ \nu^2 \\ \nu^3 \\ \nu^4 \\ \nu^5 \end{bmatrix},
$$

$\boldsymbol{c} \equiv [0, 1, 0, 0, 0]^T$ specifies the type of model error, ∇ is unknown.

Do a statistic test to determine whether the data **supports** or **rejects** H_0 on the basis of H_a .

Test Statistic (4)

Consider a very general case.

Null hypothesis $\mathbf{H}_0$: The measurements satisfy the model

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{v}, \qquad \boldsymbol{v} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{V}).$$

$\boldsymbol{A} \in \mathcal{R}^{m\times n}$ has full column rank, $\boldsymbol{V}$ is symmetric positive definite.

Alternative hypothesis $\mathbf{H}_a$: The corrupted measurements satisfy the model

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{C}\nabla + \boldsymbol{v}, \qquad \boldsymbol{v} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{V}),$$

$\boldsymbol{C} \in \mathcal{R}^{m\times q}$ is known, $[\boldsymbol{A}, \boldsymbol{C}]$ has full column rank, and ∇ is an unknown vector.

Maximum likelihood estimator (MLE)

The **density functions** of $\boldsymbol{y}$:

under H_0:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{m}{2}}\sigma|\boldsymbol{V}|^{\frac{1}{2}}} \exp\left[-\frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x})^T\boldsymbol{V}^{-1}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x})\right]$$

under H_a:

$$p(\boldsymbol{y}|\boldsymbol{x},\nabla) = \frac{1}{(2\pi)^{\frac{m}{2}}\sigma|\boldsymbol{V}|^{\frac{1}{2}}} \exp\left[-\frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}-\boldsymbol{C}\nabla)^T\boldsymbol{V}^{-1}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}-\boldsymbol{C}\nabla)\right]$$

The MLE of x under H_0:

$$\boldsymbol{x}_0 = \arg\max_{\boldsymbol{x}} p(\boldsymbol{y}|\boldsymbol{x})$$

The MLE of $\{\boldsymbol{x},\nabla\}$ under H_0:

$$\{\boldsymbol{x}_a,\nabla_a\} = \arg\max_{\boldsymbol{x},\nabla} p(\boldsymbol{y}|\boldsymbol{x},\nabla).$$

Obviously x_0 and $\{x_a, \nabla_a\}$ are respectively the solutions of the **generalized linear least squares (GLLS)** problems:

$$\begin{aligned} \mathrm{GLLS}_0: &\quad \min(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x})^T\boldsymbol{V}^{-1}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}); \\ \mathrm{GLLS}_a: &\quad \min(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}-\boldsymbol{C}\nabla)^T\boldsymbol{V}^{-1}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}-\boldsymbol{C}\nabla). \end{aligned}$$

They are also the **best linear unbiased estimators (BLUE)**.

Define the residuals

$$\boldsymbol{r}_0 \equiv \boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}_0, \qquad \boldsymbol{r}_a \equiv \boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}_a-\boldsymbol{C}\nabla_a.$$

The generalized likelihood ratio is given by

$$l(\boldsymbol{y}) \equiv \frac{p(\boldsymbol{y}|\boldsymbol{x}_0)}{p(\boldsymbol{y}|\boldsymbol{x}_a, \nabla_a)} = \exp\left[-\frac{1}{2\sigma^2}\left(\boldsymbol{r}_0^T\boldsymbol{V}^{-1}\boldsymbol{r}_0-\boldsymbol{r}_a^T\boldsymbol{V}^{-1}\boldsymbol{r}_a\right)\right].$$

The test statistic is defined by

$$\boxed{\delta_{\mathrm{TS}} \equiv -2\log l(\boldsymbol{y}) = \sigma^{-2}(\boldsymbol{r}_0^T\boldsymbol{V}^{-1}\boldsymbol{r}_0-\boldsymbol{r}_a^T\boldsymbol{V}^{-1}\boldsymbol{r}_a)}$$

Distribution of δ_{TS}

$$\delta_{\text{TS}} \sim \chi^2(q, 0), \quad \text{under } \mathrm{H}_0;$$

$$\delta_{\text{TS}} \sim \chi^2(q, \lambda), \quad \lambda = \nabla^T \boldsymbol{C}^T \boldsymbol{V}^{-1} \mathrm{cov}\{\boldsymbol{r}_0\} \boldsymbol{V}^{-1} \boldsymbol{C} \nabla, \quad \text{under } \mathrm{H}_a.$$

where $\quad \mathrm{cov}\{\boldsymbol{r}_0\} = \sigma^2[\boldsymbol{V} - \boldsymbol{A}(\boldsymbol{A}^T \boldsymbol{V}^{-1} \boldsymbol{A})^{-1} \boldsymbol{A}^T].$

Statistic testing

When δ_{TS} is larger than a given threshold, reject H_0 in favor of H_a. Otherwise accept H_0.

The threshold is usually determined by the requirement of a specific application.

An Obvious Method to Compute δ_{TS} (2)

Let $\boldsymbol{V}$ have the factorization: $\boldsymbol{V} = \boldsymbol{B}\boldsymbol{B}^T$.

Nonsingular $\boldsymbol{B}$ is given or obtained by the Cholesky factorization.

Define $\bar{\boldsymbol{y}} = \boldsymbol{B}^{-1}\boldsymbol{y}$, $\bar{\boldsymbol{A}} = \boldsymbol{B}^{-1}\boldsymbol{A}$, $\bar{\boldsymbol{C}} = \boldsymbol{B}^{-1}\boldsymbol{C}$.

Transform problems GLLS_0 & GLLS_a to the ordinary LS problems:

$$\boldsymbol{x}_0 = \arg\min(\bar{\boldsymbol{y}} - \bar{\boldsymbol{A}}\boldsymbol{x})^T(\bar{\boldsymbol{y}} - \bar{\boldsymbol{A}}\boldsymbol{x});$$

$$\{\boldsymbol{x}_a, \nabla_a\} = \arg\min(\bar{\boldsymbol{y}} - \bar{\boldsymbol{A}}\boldsymbol{x} - \bar{\boldsymbol{C}}\nabla)^T(\bar{\boldsymbol{y}} - \bar{\boldsymbol{A}}\boldsymbol{x} - \bar{\boldsymbol{C}}\nabla)$$

Compute the QR factn: $[\bar{\boldsymbol{A}},\ \bar{\boldsymbol{C}}] = [\boldsymbol{Q}_A, \boldsymbol{Q}_C, \boldsymbol{Q}_3]\begin{bmatrix} \bar{\boldsymbol{R}}_A & \bar{\boldsymbol{R}}_{AC} \\ \mathbf{0} & \bar{\boldsymbol{R}}_C \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$.

We can show

$$\boxed{\delta_{\mathrm{TS}} = \sigma^{-2}\|\boldsymbol{Q}_C^T\bar{\boldsymbol{y}}\|_2^2}$$

Two problems with this method:

- The method is not numerically stable.

  When $\boldsymbol{B}$ is ill-conditioned, accuracy may unnecessarily be lost.

- Recall $\boldsymbol{V} = \boldsymbol{B}\boldsymbol{B}^T$. If $\boldsymbol{V}$ is singular, then $\boldsymbol{B}^{-1}$ does not exist. So the method will not work.

  Note: The original formula $\delta_{\text{TS}} = \sigma^{-2}(\boldsymbol{r}_0^T \boldsymbol{V}^{-1} \boldsymbol{r}_0 - \boldsymbol{r}_a^T \boldsymbol{V}^{-1} \boldsymbol{r}_a)$ is not defined.

A Backward Stable Method for Computing δ_{TS} (5)

Idea: Use Paige's approach (1978) to solving GLLS problems.

Recall $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{V})$ and $\boldsymbol{V} = \boldsymbol{B}\boldsymbol{B}^T$. Write

$$\boldsymbol{v} = \boldsymbol{B}\boldsymbol{u}, \qquad \boldsymbol{u} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}).$$

Reformulate the GLLS problems:

$$\begin{aligned} &\mathrm{GLLS}_0: \quad \min \|\boldsymbol{u}\|_2^2, \quad \text{s.t. } \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}, && \text{under } \mathrm{H}_0 \\ &\mathrm{GLLS}_a: \quad \min \|\boldsymbol{u}\|_2^2, \quad \text{s.t. } \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{C}\nabla + \boldsymbol{B}\boldsymbol{u}, && \text{under } \mathrm{H}_a \end{aligned}$$

For simplicity, we still assume $\boldsymbol{B}$ is nonsingular.

Let $[\boldsymbol{A}, \boldsymbol{C}]$ and $\boldsymbol{B}$ have the following generalized QR factorization

$$\boldsymbol{P}^T[\boldsymbol{A}, \boldsymbol{C}] = \begin{bmatrix} \boldsymbol{U}_A & \boldsymbol{U}_{AC} \\ \boldsymbol{0} & \boldsymbol{U}_C \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \quad \boldsymbol{P}^T \boldsymbol{B} \boldsymbol{Q} = \begin{bmatrix} \boldsymbol{R}_A & \boldsymbol{R}_{AC} & \boldsymbol{R}_{A3} \\ \boldsymbol{0} & \boldsymbol{R}_C & \boldsymbol{R}_{C3} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{R}_3 \end{bmatrix}$$

Solving Problem GLLS$_a$

Transform problem GLLS$_a$:

$$\boxed{\min \|\boldsymbol{u}\|_2^2, \quad \text{s.t.} \quad \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{C}\nabla + \boldsymbol{B}\boldsymbol{u}}$$

$$\Downarrow$$

$$\boxed{\min \|\boldsymbol{u}\|_2^2, \quad \text{s.t.} \quad \underbrace{\boldsymbol{P}^T\boldsymbol{y}}_{\boldsymbol{z}} = \boldsymbol{P}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{P}^T\boldsymbol{C}\nabla + \boldsymbol{P}^T\boldsymbol{B}\boldsymbol{Q}\underbrace{\boldsymbol{Q}^T\boldsymbol{u}}_{\boldsymbol{w}}}$$

$$\Downarrow$$

$$\boxed{\begin{gathered}\min \|\boldsymbol{w}_A\|_2^2 + \|\boldsymbol{w}_C\|_2^2 + \|\boldsymbol{w}_3\|_2^2 \\ \text{s.t.} \quad \begin{bmatrix} \boldsymbol{z}_A \\ \boldsymbol{z}_C \\ \boldsymbol{z}_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{U}_A \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \boldsymbol{U}_{AC} \\ \boldsymbol{U}_C \\ \boldsymbol{0} \end{bmatrix} \nabla + \begin{bmatrix} \boldsymbol{R}_A & \boldsymbol{R}_{AC} & \boldsymbol{R}_{A3} \\ \boldsymbol{0} & \boldsymbol{R}_C & \boldsymbol{R}_{C3} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{R}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_A \\ \boldsymbol{w}_C \\ \boldsymbol{w}_3 \end{bmatrix}\end{gathered}}$$

Solving Problem GLLS$_a$, cont.

$$\min \|\boldsymbol{w}_A\|_2^2 + \|\boldsymbol{w}_C\|_2^2 + \|\boldsymbol{w}_3\|_2^2$$
$$\text{s.t.} \quad \begin{bmatrix} \boldsymbol{z}_A \\ \boldsymbol{z}_C \\ \boldsymbol{z}_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{U}_A \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \boldsymbol{U}_{AC} \\ \boldsymbol{U}_C \\ \boldsymbol{0} \end{bmatrix} \nabla + \begin{bmatrix} \boldsymbol{R}_A & \boldsymbol{R}_{AC} & \boldsymbol{R}_{A3} \\ \boldsymbol{0} & \boldsymbol{R}_C & \boldsymbol{R}_{C3} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{R}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_A \\ \boldsymbol{w}_C \\ \boldsymbol{w}_3 \end{bmatrix}$$

The optimal soln $\boldsymbol{x}_a$, ∇_a, $\boldsymbol{w}_a \equiv [(\boldsymbol{w}_A^a)^T, (\boldsymbol{w}_C^a)^T, (\boldsymbol{w}_3^a)^T]^T$ satisfies

$$\boldsymbol{w}_A^a = 0, \qquad \boldsymbol{w}_C^a = 0, \qquad \begin{bmatrix} \boldsymbol{U}_A & \boldsymbol{U}_{AC} & \boldsymbol{R}_{A3} \\ \boldsymbol{0} & \boldsymbol{U}_C & \boldsymbol{R}_{C3} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{R}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_a \\ \nabla_a \\ \boldsymbol{w}_3^a \end{bmatrix} = \begin{bmatrix} \boldsymbol{z}_A \\ \boldsymbol{z}_C \\ \boldsymbol{z}_3 \end{bmatrix}.$$

The GLS residual or the optimal $\boldsymbol{u}$: $\quad \boldsymbol{u}_a = \boldsymbol{Q}\boldsymbol{w}_a$.

Solving Problem GLLS$_0$

$$\boxed{\min \|\boldsymbol{u}\|_2^2, \quad \text{s.t.} \quad \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}}$$

$$\Downarrow$$

$$\boxed{\begin{array}{c} \min(\|\boldsymbol{w}_A\|_2^2 + \|\boldsymbol{w}_C\|_2^2 + \|\boldsymbol{w}_3\|_2^2), \\ \text{s.t.} \quad \begin{bmatrix} \boldsymbol{z}_A \\ \boldsymbol{z}_C \\ \boldsymbol{z}_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{U}_A \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \boldsymbol{R}_A & \boldsymbol{R}_{AC} & \boldsymbol{R}_{A3} \\ \boldsymbol{0} & \boldsymbol{R}_C & \boldsymbol{R}_{C3} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{R}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_A \\ \boldsymbol{w}_C \\ \boldsymbol{w}_3 \end{bmatrix} \end{array}}$$

The optimal soln $\boldsymbol{x}_0$, $\boldsymbol{w}_0 \equiv [(\boldsymbol{w}_A^0)^T, (\boldsymbol{w}_C^0)^T, (\boldsymbol{w}_3^0)^T]^T$ satisfies

$$\boldsymbol{w}_A^0 = 0, \qquad \begin{bmatrix} \boldsymbol{U}_A & \boldsymbol{R}_{AC} & \boldsymbol{R}_{A3} \\ \boldsymbol{0} & \boldsymbol{R}_C & \boldsymbol{R}_{C3} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{R}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{w}_C^0 \\ \boldsymbol{w}_3^0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{z}_A \\ \boldsymbol{z}_C \\ \boldsymbol{z}_3 \end{bmatrix}$$

The GLS residual or the optimal $\boldsymbol{u}$: $\quad \boldsymbol{u}_0 = \boldsymbol{Q}\boldsymbol{w}_0$.

Computing the Test Statistic δ_{TS}

The GLS residual under H_a: $\quad \boldsymbol{u}_a = \boldsymbol{Q}\,[(\boldsymbol{w}_A^a)^T, (\boldsymbol{w}_C^a)^T, (\boldsymbol{w}_3^a)^T]^T$

$$\boldsymbol{w}_A^a = 0, \quad \boldsymbol{w}_C^a = 0, \quad \begin{bmatrix} \boldsymbol{U}_A & \boldsymbol{U}_{AC} & \boldsymbol{R}_{A3} \\ \mathbf{0} & \boldsymbol{U}_C & \boldsymbol{R}_{C3} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{R}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_a \\ \nabla_a \\ \boldsymbol{w}_3^a \end{bmatrix} = \begin{bmatrix} \boldsymbol{z}_A \\ \boldsymbol{z}_C \\ \boldsymbol{z}_3 \end{bmatrix}$$

The GLS residual under H_0: $\quad \boldsymbol{u}_0 = \boldsymbol{Q}\,[(\boldsymbol{w}_A^0)^T, (\boldsymbol{w}_C^0)^T, (\boldsymbol{w}_3^0)^T]^T$

$$\boldsymbol{w}_A^0 = 0, \quad \begin{bmatrix} \boldsymbol{U}_A & \boldsymbol{R}_{AC} & \boldsymbol{R}_{A3} \\ \mathbf{0} & \boldsymbol{R}_C & \boldsymbol{R}_{C3} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{R}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{w}_C^0 \\ \boldsymbol{w}_3^0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{z}_A \\ \boldsymbol{z}_C \\ \boldsymbol{z}_3 \end{bmatrix}$$

Thus we can easily show

$$\boxed{\delta_{\mathrm{TS}} = \sigma^{-2}(\|\boldsymbol{u}_0\|_2^2 - \|\boldsymbol{u}_a\|_2^2) = \sigma^{-2}\|\boldsymbol{w}_C^0\|_2^2}$$

Remarks

- The test statistic δ_{TS} can be computed in parallel with the the optimal estimates $\boldsymbol{x}_0$ under H_0 and $\boldsymbol{x}_a$ under H_a.
- Can easily show our method is numerically stable.
- Can find the covariance matrices of $\boldsymbol{x}_0$ and $\boldsymbol{x}_a$.
- Can extend the method to deal with the non-square $\boldsymbol{B}$ case.
- Can easily handle linear equality constraints.

Summary

- Computing GPS relative position estimates
  - Present a recursive LS approach for relative positioning based on carrier phase and code measurements for L1 carrier signal.
  - The algorithm is efficient—makes full use of structure, & numerically reliable—uses orthogonal transformations.
  - The approach allows: satellite rising/setting, more measurement equations, dynamic equations.
- Computing a test statistic for data quality control
  - Present a backward stable method
  - Can handle the singular noise covariance matrix

Numerical Linear Algebra Is Important and Useful in GPS !!!