Interdisciplinary Division of Aeronautical and Aviation Engineering
航空工程跨領域學部
THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

State Estimation: Factor Graph for Integrated Navigation II

AAE4203 – Guidance and Navigation

Dr Weisong Wen
Research Assistant Professor
Department of Aeronautical and Aviation Engineering
The Hong Kong Polytechnic University

Week 11, 30 Mar 2022

About the Assignment 2

Answer to AAE4203 Guidance and Navigation
Assignment 2

Submission Portal

Please scan your hand-written work and submit it in PDF format to my email (welson.wen@polyu.edu.hk).

Deadline

Please submit the assignment on or before **10th April 2022**.

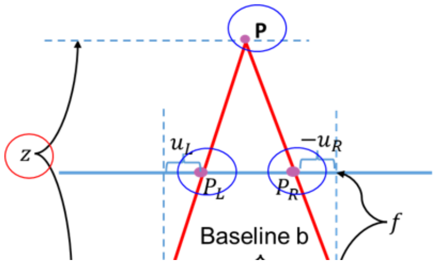
Question 1 (20 marks)

Given the stereo camera model as Fig. 1. The baseline distance, b, between the two cameras is 0.01 meters. The focal length f is 460.

(1) Given a 3D feature that is captured by both the left and right camera, the pixel locations of the features in the left image are (30, 100) and (30, 300). What is the depth of the 3D feature? (10 marks)

(2) Given the image size of the left and right camera as 460×460, what is the maximum depth the stereo camera can measure? (5 marks)

(3) Describe the key drawbacks of the stereo camera for depth calculation. (5 marks)



Question 2 (30 marks)

Given the pinhole model of the monocular camera as Fig. 2. Given a feature with pixel location (200,300),

(1) Describe the reason behind the image distortion, the radial and tangential distortion (3 marks)

(2) The parameters for the radial distortion are $k1 = -0.046, k2 = 0.013, k3 = 0.012$. The parameters for the tangential distortion are $p1 = -0.03, p2 = 0.02$. What is the undistorted location of the pixel? (10 marks)

(3) The depth of the feature is 10 meters. The intrinsic parameters of the camera are $f_u = 264.9, f_v = 264.7, \Delta u = 334.39, \Delta v = 183.39$. What is the intrinsic matrix of the camera? Based on the undistorted location in (2), what is the 3D location of the feature in the camera frame? (10 marks)

(4) Based on the 3D location of the feature calculated in (3), what is the 3D location of the feature after applying the rotation (**R**) and translation (**t**) matrix as below? (7 marks)

$$\mathbf{R} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } \mathbf{t} = \begin{bmatrix} 2 \\ 5 \\ 7 \end{bmatrix}$$

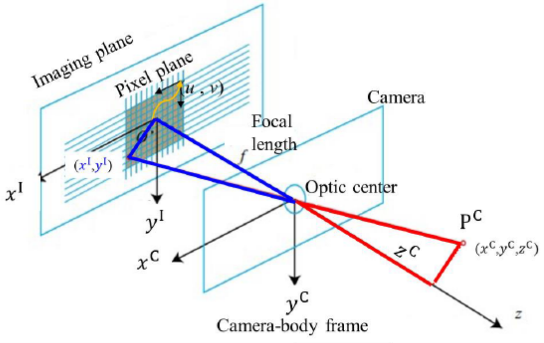


Figure 2. Illustration of the pinhole camera model.

Group Presentation

> https://docs.google.com/document/d/1RujUcD1sNyUs31-2Bucn20h99OA8EtTBhY2J-XJfeDw/edit

3	Wen, Weisong, Tim Pfeifer, Xiwei Bai, and Li-Ta Hsu. "Factor graph optimization for GNSS/INS integration: A comparison with the extended Kalman filter." NAVIGATION, Journal of the Institute of Navigation 68, no. 2 (2021): 315-331.	LI Zhengdao (18081447D) SU Meiling (19083997D)

9	GNSS Real-time Kinematic Positioning for Autonomous Driving	Kwok Yiu (20096834D) Lam Cho Yi(20099376D)

11	B. Xu, Q. Jia and L. -T. Hsu, "Vector Tracking Loop-Based GNSS NLOS Detection and Correction: Algorithm Design and Performance Analysis," in IEEE Transactions on Instrumentation and Measurement, vol. 69, no. 7, pp. 4604-4619, July 2020, doi: 10.1109/TIM.2019.2950578.	Marcelino Jason (18079168d) ZHANG Cheng chen (19078955d) FANG Jingxiaotao (18081477d)

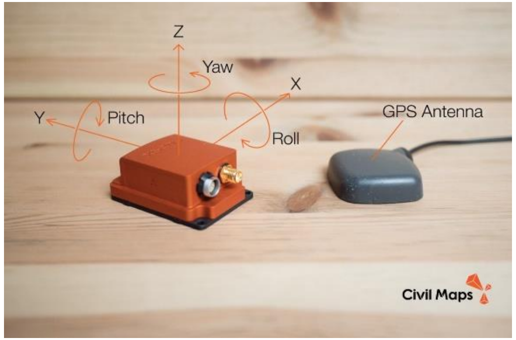
Outline

> Factor Graph Optimization
- From MAP to **Factor Graph Optimization**
- Solving the Factor Graph Optimization
  - Gradient Descent
  - Newton Method
  - Gauss-Newton
  - Levenberg-Marquardt Method
- GNSS Positioning with Factor Graph Optimization

> Supplementary: GNSS/INS Integration Using Kalman Filtering and Factor Graph Optimization

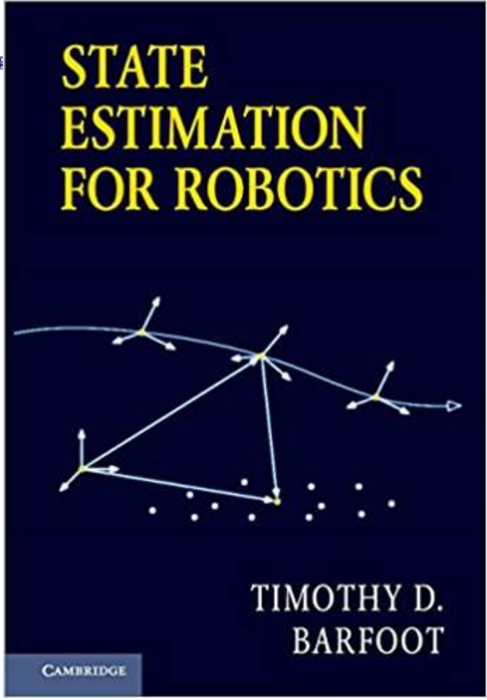
How the Sensor Fusion Problem Looks like…

- GPS provide the position in x, y, z
- IMU provide the linear angular velocity in x, y, z
- GPS Doppler provide velocity in x, y, z
- Visual positioning provide relative motion $\Delta x, \Delta y, \Delta z$



How to achieve this sensor fusion by combining the positioning from different sources?

State Estimation Methods



- First Order-Markov Assumption
- Gaussian noise assumption

Maximum A Posterior (MAP)

Filtering-based State Estimation

- First Order-Markov Assumption

Optimization-based State Estimation

- Gaussian noise assumption

Kalman Filtering

Extended Kalman Filtering

Unscented Kalman Filtering

Factor Graph Optimization

Basics for Probabilistic

Event A and B. $P(\mathrm{A})$ denotes the probability that the event A happens.

$$P(\mathrm{B}|\mathrm{A}) = \frac{P(\mathrm{AB})}{P(\mathrm{A})} \Rightarrow P(\mathrm{AB}) = P(\mathrm{B}|\mathrm{A})P(\mathrm{A}) \Rightarrow P(\mathrm{B}|\mathrm{A}) = \frac{P(\mathrm{A}|\mathrm{B})P(\mathrm{B})}{P(\mathrm{A})}$$

Event z denotes the measurements. $P(\mathrm{z})$ denotes the probability that the event A happens.

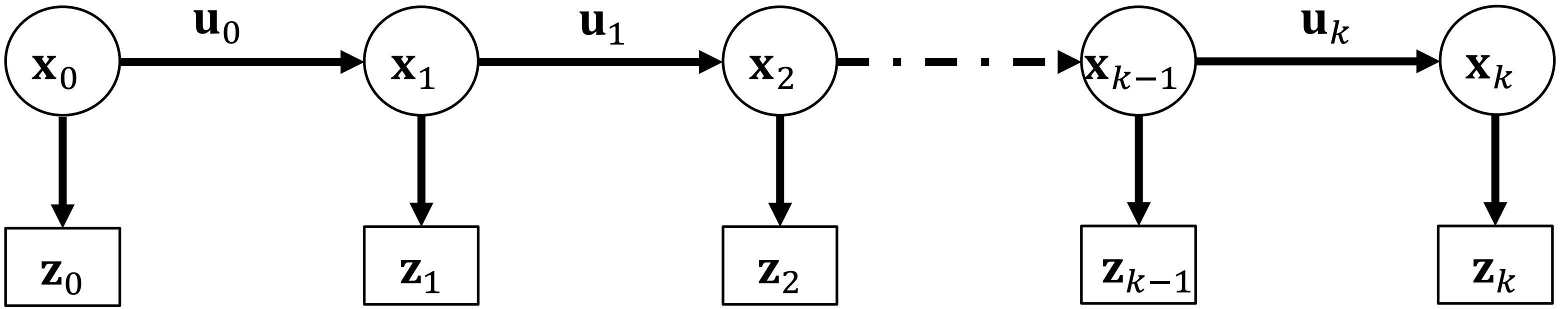
$$P(\mathbf{x}_k|\mathbf{z}_1, \quad \dots, \quad \mathbf{z}_k) = \frac{P(\mathbf{z}_0, \quad \dots, \quad \mathbf{z}_k|\mathbf{x}_k)P(\mathbf{x}_k)}{P(\mathbf{z}_0, \quad \dots, \quad \mathbf{z}_k)}$$

The probabilistic view of the state estimation is: given a set of measurements $(\mathbf{z}_0, \quad \dots, \quad \mathbf{z}_k)$, can we find a best state $\mathbf{x}_k$ to maximize the conditional probabilistic $P(\mathbf{x}_k|\mathbf{z}_1, \quad \dots, \quad \mathbf{z}_k)$?

Estimation Formulation

$\mathbf{u}_i$: IMU Measurement

$\mathbf{z}_i$: GNSS measurement



States set

$$\chi = \{\mathbf{x}_o, \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k\}$$

The **maximum a posteriori (MAP)** estimate is given by

Optimal State set

$$\hat{\chi} = arg \max_{\chi}(P(\chi|\mathbf{Z},\mathbf{U})) \quad P(\chi|\mathbf{Z},\mathbf{U}) = \prod_k P(\mathbf{z}_k|\mathbf{x}_k)\, P(\mathbf{x}_0) \prod_k P(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{u}_{k-1})$$

Bayesian theory

How to understand the $P(\mathbf{z}_0, \ldots, \mathbf{z}_k|\mathbf{x}_k)$

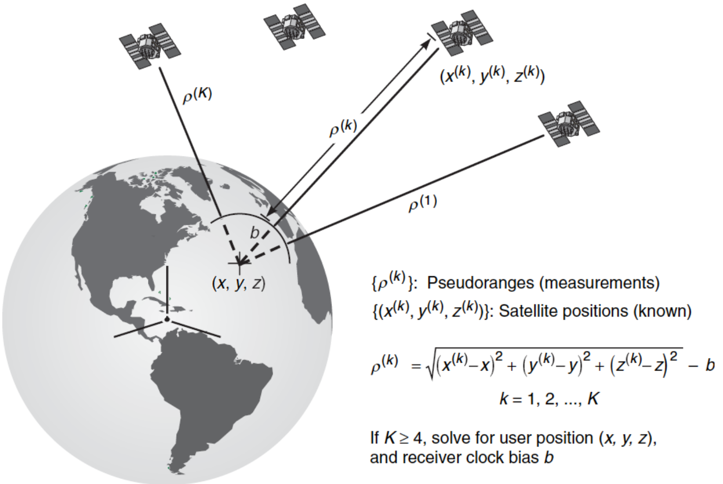
Observation function for pseudorange (code) measurement

$$\underbrace{\rho^s_{r,t}}_{\text{Pseudorange}} = \underbrace{r^s_{r,t}}_{\substack{\text{Range distance} \\ ||\mathbf{p}^{G,s}_t - \mathbf{p}^G_{r,t}||}} + c\big(\underbrace{\delta_{r,t}}_{\substack{\text{Receiver clock} \\ \text{Bias (1\textasciitilde 2m)}}} - \underbrace{\delta^s_{r,t}}_{\substack{\text{Satellite clock} \\ \text{bias}}}\big) + \underbrace{I^s_{r,t}}_{\substack{\text{ionospheric delay} \\ \text{Distance (1\textasciitilde 2m)}}} + \underbrace{T^s_{r,t}}_{\substack{\text{tropospheric delay} \\ \text{Distance (1\textasciitilde 2m)}}} + \varepsilon^s_{r,t}$$

$\varepsilon^s_{r,t}$: multipath effects, NLOS receptions, receiver noise, antenna phase-related noise (0~100m)

$\rho^s_{r,t} \Rightarrow z^s_{r,t}$ $\qquad$ $\mathbf{p}^G_{r,t} \Rightarrow \mathbf{x}^G_{r,t}$

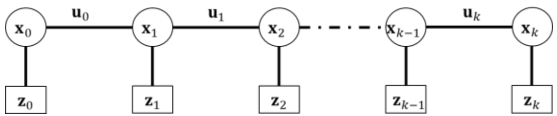
$$P(\mathbf{x}_k|\mathbf{z}_1, \ldots, \mathbf{z}_k) = \frac{P(\mathbf{z}_0, \ldots, \mathbf{z}_k|\mathbf{x}_k)P(\mathbf{x}_k)}{P(\mathbf{z}_0, \ldots, \mathbf{z}_k)}$$



Formulate the $P(\mathbf{x}_k|\mathbf{z}_1, \ldots, \mathbf{z}_k)$ for the GNSS pseudorange measurements!

From MAP Estimate to Factor Graph Optimization (FGO)

Maximum a posteriori (MAP) estimate is given by



$$P(\boldsymbol{\chi}|\mathbf{Z},\mathbf{U}) = \prod_k P(\mathbf{z}_k|\mathbf{x}_k)\, P(\mathbf{x}_0) \prod_k P(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{u}_{k-1})$$

Assumption
Gaussian
noise model

$$\mathbf{x}_j = \mathbf{f}_j(\mathbf{x}_{j-1},\mathbf{u}_j) + \mathbf{w}_j$$

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k$$

Factorization

$F_{propagation}$

$$P(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{u}_{k-1}) \doteq \exp(||\mathbf{f}_k(\mathbf{x}_{k-1},\mathbf{u}_k) - \mathbf{x}_k||^2_{\boldsymbol{\Sigma}_{\mathbf{u},k}})$$

F_{update}

$$P(\mathbf{z}_k|\mathbf{x}_k) \doteq \exp(||\mathbf{h}_k(\mathbf{x}_k) - \mathbf{z}_k||^2_{\boldsymbol{\Sigma}_{\mathbf{z},k}})$$

Graph construction

Factor in graph

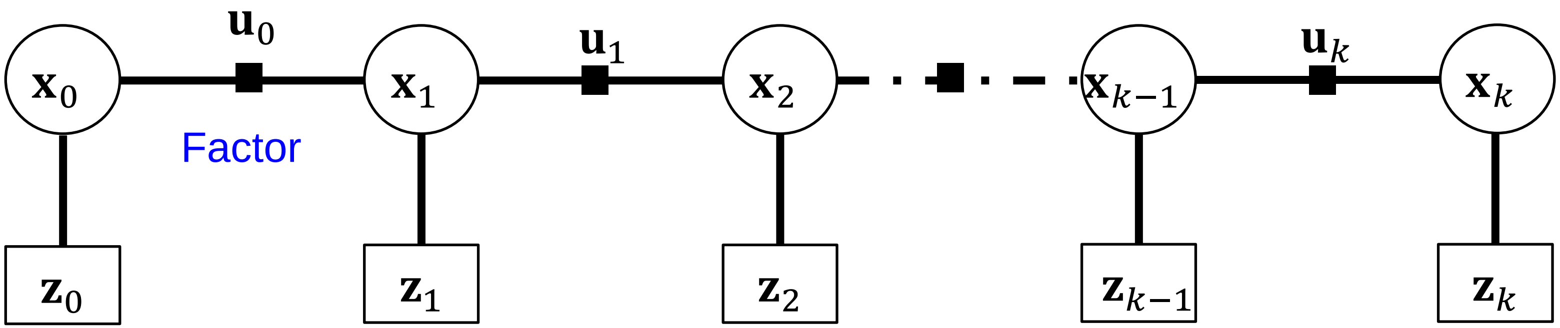
$$P(\boldsymbol{\chi}|\mathbf{Z},\mathbf{U}) \propto \prod_k F_k^{Pro}(\mathbf{x}_k) F_k^{upd}(\mathbf{x}_k)$$

Find the maximum
likelihood-> optimization

$$\hat{\boldsymbol{\chi}} = \arg\max_{\chi} \prod_k F_k(\mathbf{x}_k) = \underset{\chi}{\operatorname{argmin}}\, e(\chi)$$

$$e(\chi) \doteq \sum_k ||\mathbf{h}_k(\mathbf{x}_k) - \mathbf{z}_k||^2_{\Sigma_k} + \sum_k ||\mathbf{f}_k(\mathbf{x}_{k-1},\mathbf{u}_k) - \mathbf{x}_k||^2_{\Sigma_k}$$

From MAP Estimate to Factor Graph Optimization (FGO)



Factor in graph

$$P(\boldsymbol{\chi}|\mathbf{Z},\mathbf{U}) \propto \prod_k F_k^{Pro}(\mathbf{x}_k) F_k^{upd}(\mathbf{x}_k)$$

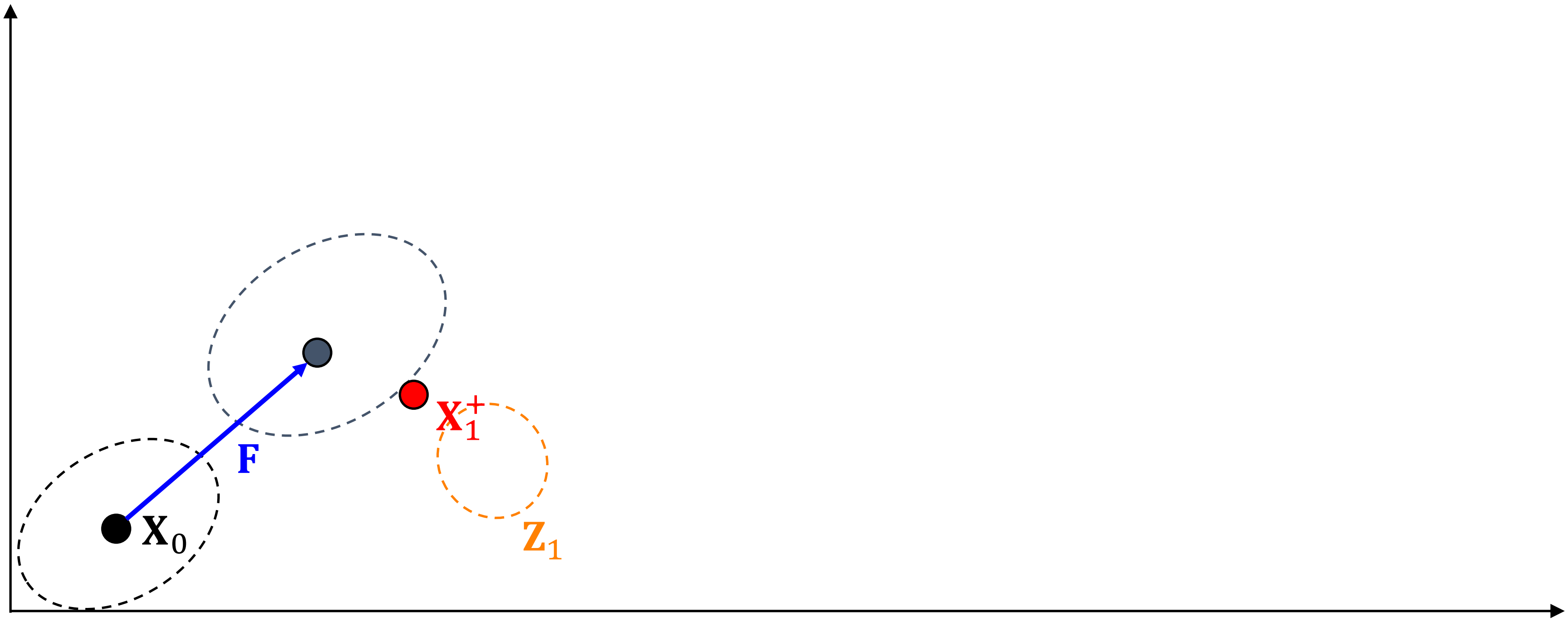
Find the maximum likelihood-> optimization

$$e(\chi) \doteq \sum_k ||\mathbf{h}_k(\mathbf{x}_k) - \mathbf{z}_k||^2_{\Sigma_k} + \sum_k ||\mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{u}_k) - \mathbf{x}_k||^2_{\Sigma_k}$$

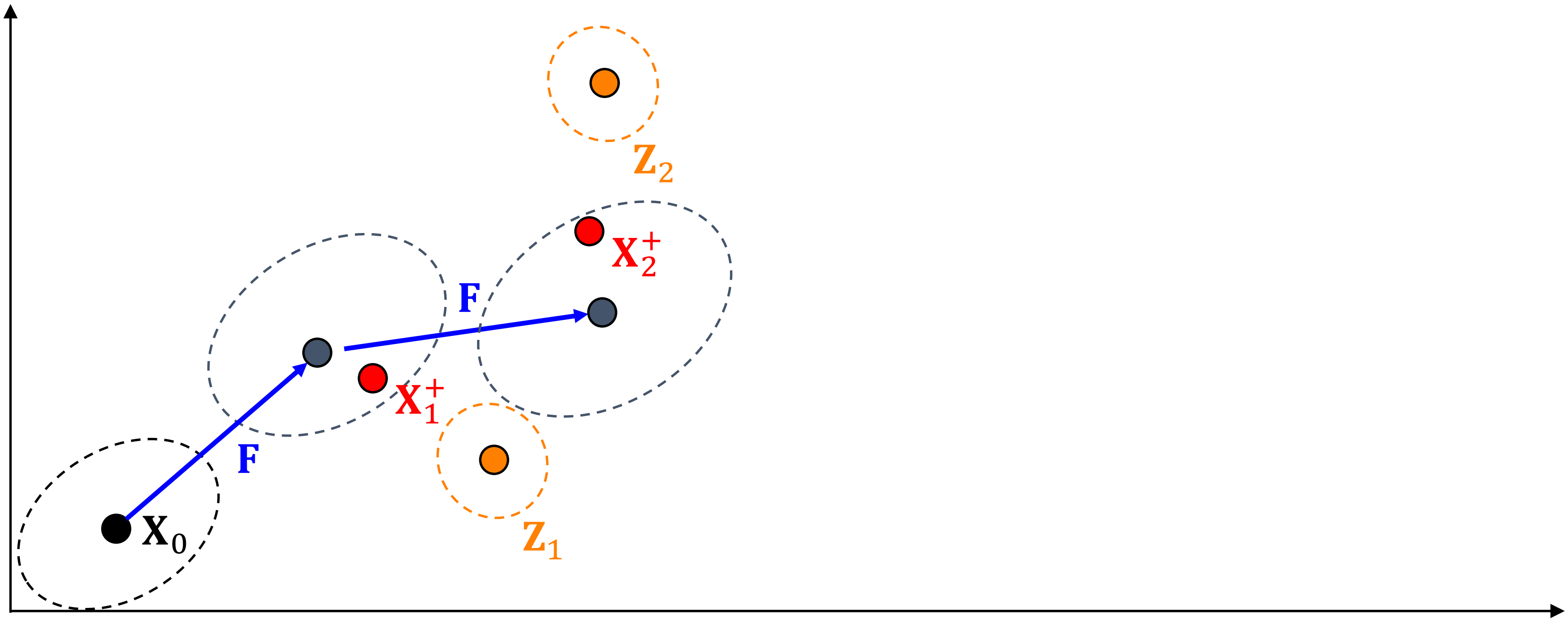
Factor Graph Optimization in GNSS

- **Example of the GNSS loosely-coupled pseudorange/Doppler integration using Kalman filtering and factor graph optimization**
- Example of the GNSS tightly-coupled pseudorange/Doppler integration using factor graph optimization

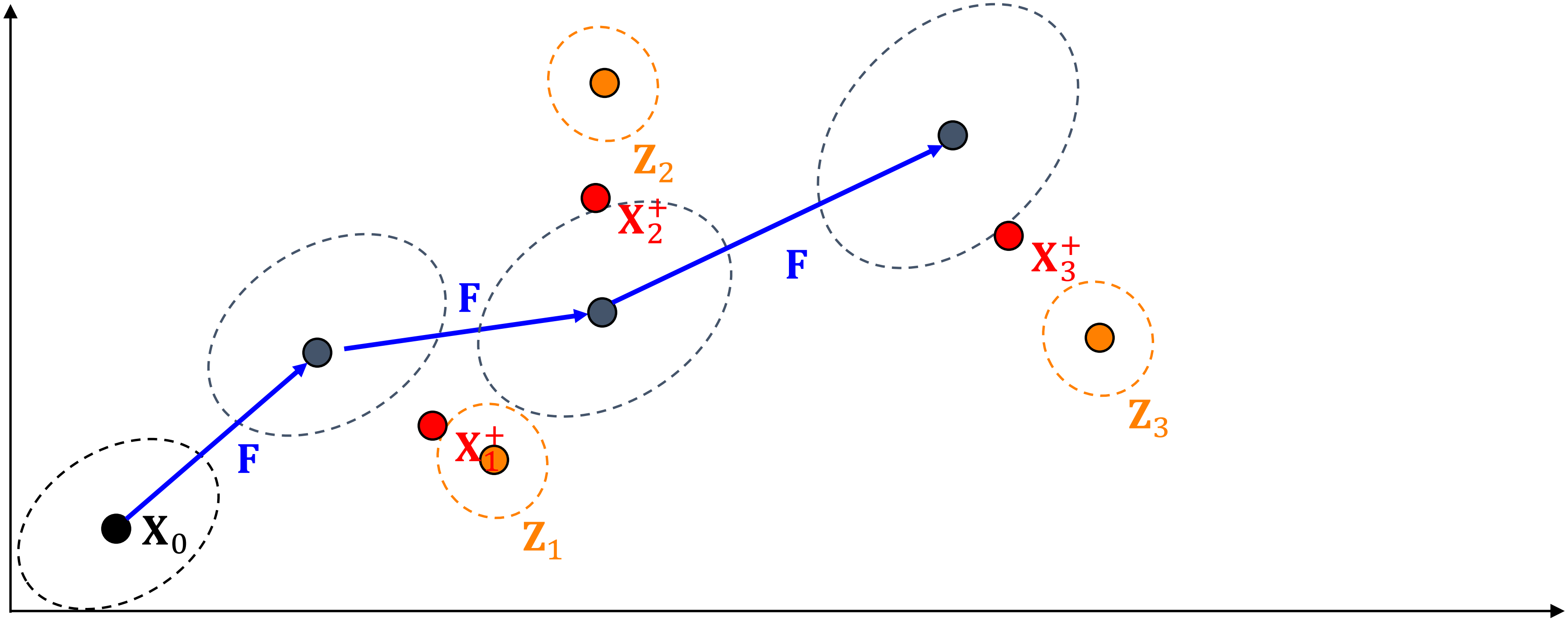
Batch (including all the data in the past) optimization



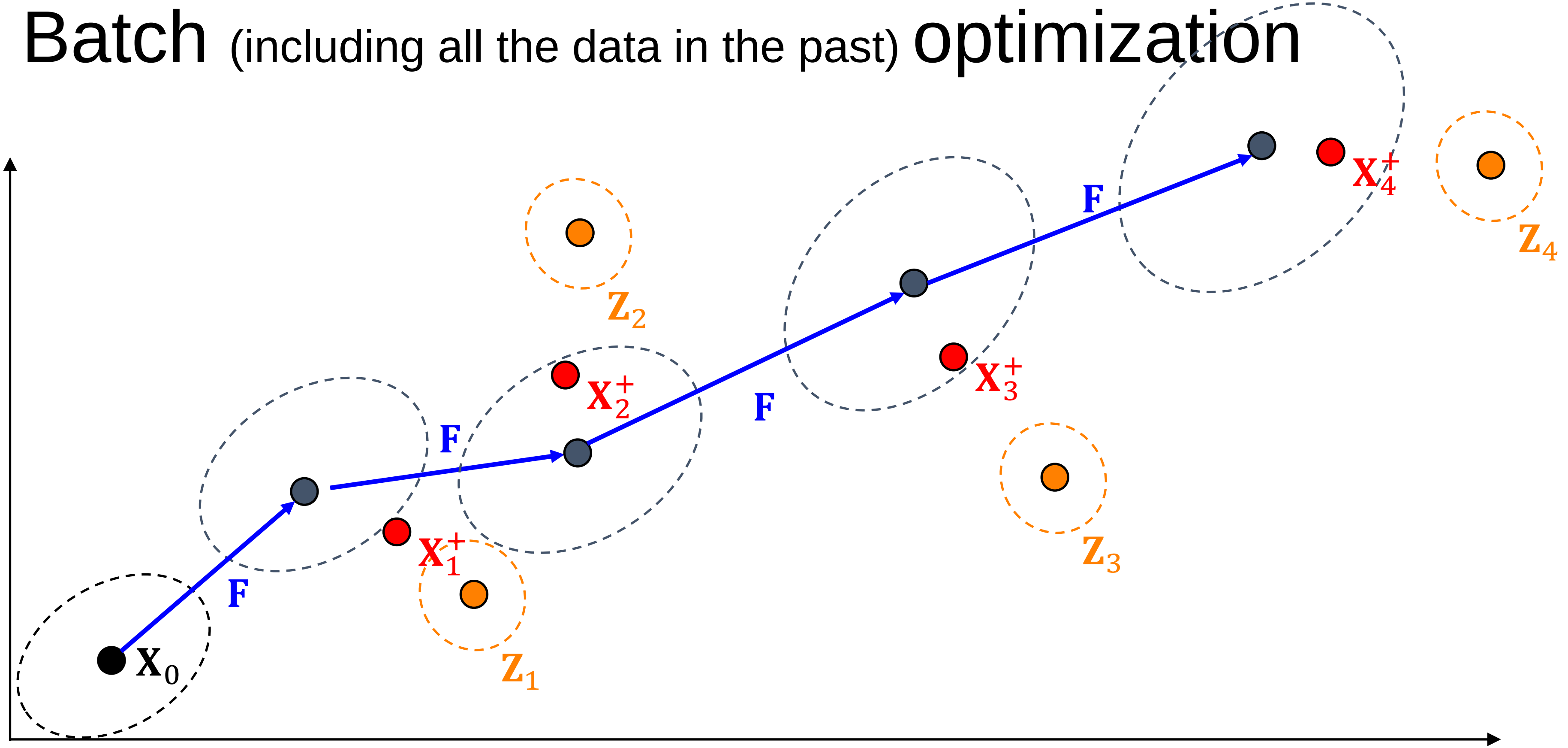
Batch (including all the data in the past) optimization



Batch (including all the data in the past) optimization

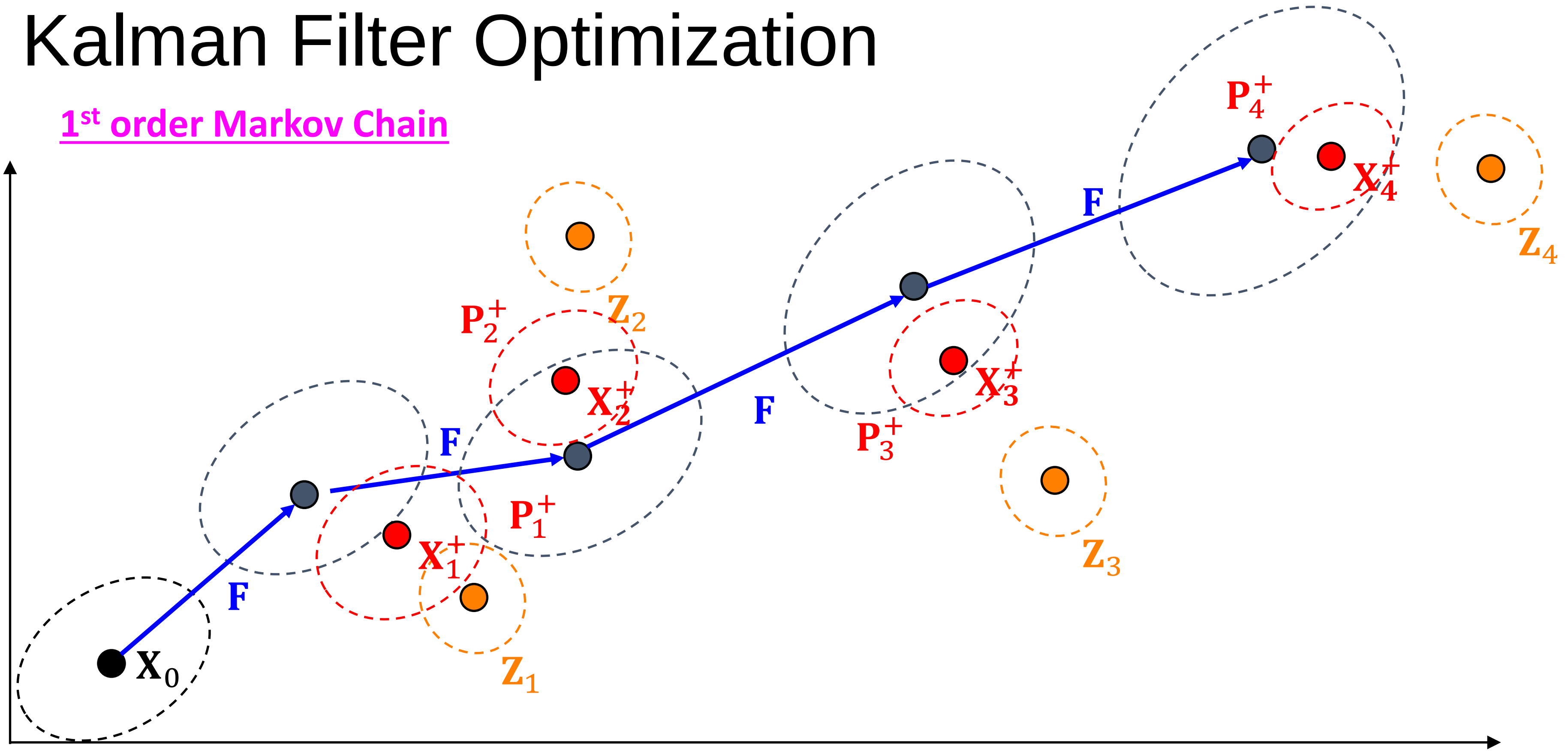


Batch (including all the data in the past) optimization

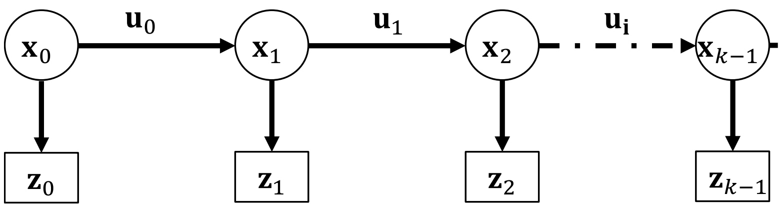


Kalman Filter Optimization

1st order Markov Chain



How to solve the optimization problem?



$$P(\boldsymbol{\chi}|\mathbf{Z},\mathbf{U}) \propto \prod_k F_k^{upd}(\mathbf{x}_k) F_k^{Pro}(\mathbf{x}_k)$$

Find the maximum likelihood-> optimization

$$\hat{\boldsymbol{\chi}} = \arg\max_{\boldsymbol{\chi}} \prod_k F_k(\mathbf{x}_k) = \underset{\boldsymbol{\chi}}{\operatorname{argmin}}\, e(\boldsymbol{\chi})$$

$$e(\boldsymbol{\chi}) \doteq \sum_k \|\mathbf{h}_k(\mathbf{x}_k) - \mathbf{z}_k\|^2_{\Sigma_k} + \sum_k \|\mathbf{f}_k(\mathbf{x}_{k-1},\mathbf{u}_k) - \mathbf{x}_k\|^2_{\Sigma_k}$$

Non-linear optimization iteratively obtain the optimal solution

$$\boldsymbol{\chi} = \boldsymbol{\chi}^{(0)}$$

$$\boldsymbol{\chi}^{(1)} = \boldsymbol{\chi}^{(0)} + \Delta\boldsymbol{\chi} \qquad \mathbf{J}_R = \frac{\partial(\mathbf{H}(\boldsymbol{\chi}) - \mathbf{Z})}{\partial\boldsymbol{\chi}} + \frac{\partial(\mathbf{F}(\boldsymbol{\chi},\mathbf{U}) - \boldsymbol{\chi})}{\partial\boldsymbol{\chi}}$$

Using Levenberg-Marquardt algorithm (LM)

$$\Delta\boldsymbol{\chi} = (\mathbf{J}_R^{\mathrm{T}}\mathbf{J}_R + \mu\mathbf{I})^{-1}\mathbf{J}_R^{\mathrm{T}}\left((\mathbf{H}(\boldsymbol{\chi}^{(0)}) - \mathbf{Z})\right)$$

$$\hat{\boldsymbol{\chi}} = \boldsymbol{\chi}^{(n)},\ \text{if } e(\boldsymbol{\chi}^{(n+1)}) - e(\boldsymbol{\chi}^{(n)}) \le \varepsilon$$

Dellaert, Frank, and Michael Kaess. "Factor graphs for robot perception." Foundations and Trends® in Robotics 6.1-2 (2017): 1-139.

Non-Statistical Method – Least Square

Least Square

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|h(\mathbf{x}) - \mathbf{z}\|^2$$

Linear Least Square

$$\hat{\mathbf{x}} = h^{-1}(\mathbf{z}) \cong \mathbf{Hz} = \left(\mathbf{H}^{\mathrm{T}}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{z}$$

$$\boxed{\hat{\mathbf{x}} = \frac{1}{2}\arg\min_{\mathbf{x}} \boxed{\|h(\mathbf{x}) - \mathbf{z}\|^2} = \frac{1}{2}\arg\min_{\mathbf{x}} e(\mathbf{x})}$$

$$\frac{de(\mathbf{x})}{d\mathbf{x}} = \frac{d\frac{1}{2}(\mathbf{Hx} - \mathbf{z})^2}{d\mathbf{x}} = \frac{d\frac{1}{2}(\mathbf{Hx} - \mathbf{z})^2}{d(\mathbf{Hx} - \mathbf{z})}\frac{d(\mathbf{Hx} - \mathbf{z})}{d\mathbf{x}} = 0$$

$$(\mathbf{Hx} - \mathbf{z})\left(\mathbf{H}^{\mathrm{T}}\right) = 0$$

$$\mathbf{H}^{\mathrm{T}}\mathbf{Hx} = \mathbf{H}^{\mathrm{T}}\mathbf{z}$$

$$\boxed{\mathbf{x} = \left(\mathbf{H}^{\mathrm{T}}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{z}}$$

Non-Linear Least Square

- **Gradient Descent Method**
- **Gauss-Newton Method**
- **Levenberg-Marquardt Method**

NLS – Gradient Descent

Non-Linear Least Square

Gradient Descent Method → **Jacobian Matrix**

Residual function

error function

$$\hat{\mathbf{x}} = \frac{1}{2}\arg\min_{\mathbf{x}} \|\boxed{h(\mathbf{x}) - \mathbf{z}}\|^2 = \frac{1}{2}\arg\min_{\mathbf{x}} e\boxed{(\mathbf{x})}$$

Initialize

$$\mathbf{x} = \mathbf{x}^{(0)}$$

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \alpha\mathbf{J}^{(0)}\mathbf{d}\,,\,\mathbf{J}^{(0)}\mathbf{d} < 0$$

Iterate by descending the state

α: step size

$\mathbf{d}$: unit vector of descends

$$\mathbf{J} = \begin{bmatrix} \frac{\partial e_1}{\partial x_1^{(0)}} & \cdots & \frac{\partial e_1}{\partial x_m^{(0)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial e_n}{\partial x_1^{(0)}} & \cdots & \frac{\partial e_n}{\partial x_m^{(0)}} \end{bmatrix}$$

$$e\left(\mathbf{x}^{(1)}\right) = \frac{1}{2}\left\|h\left(\mathbf{x}^{(1)}\right) - \mathbf{z}\right\|^2$$

$$\hat{\mathbf{x}} = \mathbf{x}^{(n)},\ \text{if } e\left(\mathbf{x}^{(n+1)}\right) - e\left(\mathbf{x}^{(n)}\right) \leq \varepsilon$$

ε: threshold for iteration

the matrix of all its first-order partial derivatives, the gradient of model.

NLS – Newton Method

Non-Linear Least Square → **Newton Method** → **Hessian Matrix**

$$e(\mathbf{x}) \approx \|h(\mathbf{x}) - \mathbf{z}\|_2$$

$$\mathcal{H} = \begin{bmatrix} \dfrac{\partial^2 e}{\partial x_1 \partial x_1} & \cdots & \dfrac{\partial^2 e}{\partial x_1 \partial x_m} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial^2 e}{\partial x_1 \partial x_m} & \cdots & \dfrac{\partial^2 e}{\partial x_m \partial x_m} \end{bmatrix}$$

The square matrix of second-order partial derivatives of a scalar-valued function, the gradient of the gradient model, the acceleration.

Expand the error function of Taylor series and take the second-order

$$e(\mathbf{x}) \cong e(\mathbf{x}^{(0)}) + \frac{\partial e(\mathbf{x}^{(0)})}{\partial \mathbf{x}^{(0)}}(\mathbf{x} - \mathbf{x}^{(0)}) + \frac{1}{2}\frac{\partial^2 e(\mathbf{x}^{(0)})}{\partial^2 \mathbf{x}^{(0)}}(\mathbf{x} - \mathbf{x}^{(0)})^2$$

$$e(\mathbf{x}) \cong e(\mathbf{x}^{(0)}) + \mathbf{J}(\mathbf{x} - \mathbf{x}^{(0)}) + \frac{1}{2}\mathcal{H}(\mathbf{x} - \mathbf{x}^{(0)})^2$$

$$e(\mathbf{x}) \cong e(\mathbf{x}^{(0)}) + \mathbf{J}\Delta\mathbf{x} + \frac{1}{2}\mathcal{H}\Delta\mathbf{x}^2$$

Find a $\Delta\mathbf{x}$ to achieve least square of error function

$$\frac{\partial e(\mathbf{x})}{\partial \Delta\mathbf{x}} = \mathbf{J}^{\mathrm{T}} + \mathcal{H}\Delta\mathbf{x} = 0 \longrightarrow \Delta\mathbf{x} = -\frac{\mathbf{J}^{\mathrm{T}}}{\mathcal{H}} = -\mathcal{H}^{-1}\mathbf{J}^{\mathrm{T}}$$

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \Delta\mathbf{x} = \mathbf{x}^{(0)} - \mathcal{H}^{-1}\mathbf{J}^{\mathrm{T}}$$

NLS – Gauss-Newton Method (approximation of Newton method to reduce computation load of Hessian matrix)

Non-Linear Least Square

Gauss-Newton Method

First order on the residual function

Hessian Matrix

Expand the error function of Taylor series from $\mathbf{x}^{(0)} + \Delta\mathbf{x}$ and take the first-order so that we found the best $\mathbf{x}$

$$h\left(\mathbf{x}^{(0)} + \Delta\mathbf{x}\right) \cong h\left(\mathbf{x}^{(0)}\right) + \frac{\partial h\left(\mathbf{x}^{(0)}\right)}{\partial \mathbf{x}^{(0)}}(\Delta\mathbf{x}) = h\left(\mathbf{x}^{(0)}\right) + \mathbf{J}_R\Delta\mathbf{x}$$

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|h(\mathbf{x}) - \mathbf{z}\|^2 = \arg\min_{\mathbf{x}} \frac{1}{2}\left\|\left(h\left(\mathbf{x}^{(0)}\right) + \mathbf{J}_R\Delta\mathbf{x}\right) - \mathbf{z}\right\|^2$$

$$\frac{1}{2}\|h(\mathbf{x}) - \mathbf{z}\|^2 = \frac{1}{2}\left(e\left(\mathbf{x}^{(0)}\right)^{\mathrm{T}} e\left(\mathbf{x}^{(0)}\right) + 2e\left(\mathbf{x}^{(0)}\right)^{\mathrm{T}}\mathbf{J}_R\Delta\mathbf{x} + \Delta\mathbf{x}^{\mathrm{T}}\mathbf{J}_R^{\mathrm{T}}\mathbf{J}_R\Delta\mathbf{x}\right) = e'(\mathbf{x} + \Delta$$

Find a $\Delta\mathbf{x}$ to achieve least square of error function

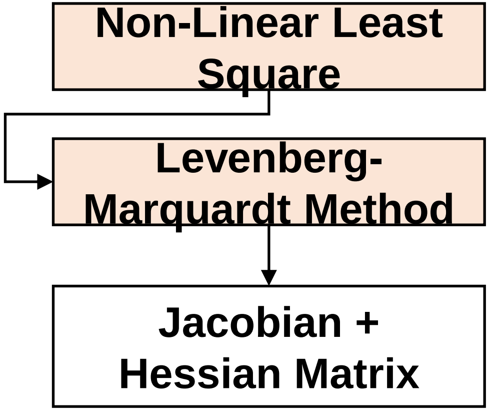
$$\frac{\partial e(\mathbf{x} + \Delta\mathbf{x})}{\partial \Delta\mathbf{x}} = 0 \longrightarrow \mathbf{J}_R^{\mathrm{T}}\mathbf{J}_R\Delta\mathbf{x} = -\mathbf{J}_R^{\mathrm{T}}\left(h\left(\mathbf{x}^{(0)}\right) - \boldsymbol{z}\right)$$

$$\Delta\mathbf{x} = -\left(\mathbf{J}_R^{\mathrm{T}}\mathbf{J}_R\right)^{-1}\mathbf{J}_R^{\mathrm{T}}\left(h\left(\mathbf{x}^{(0)}\right) - \boldsymbol{z}\right) \cong -\,\boldsymbol{\mathcal{H}}^{-1}\,\mathbf{J}^{\mathrm{T}}$$ From Newton method

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \Delta\mathbf{x} = \mathbf{x}^{(0)} - \left(\mathbf{J}_R^{\mathrm{T}}\mathbf{J}_R\right)^{-1}\mathbf{J}_R^{\mathrm{T}}\left(h\left(\mathbf{x}^{(0)}\right) - \boldsymbol{z}\right) \qquad e\left(\mathbf{x}^{(1)}\right) = \frac{1}{2}\left\|h\left(\mathbf{x}^{(1)}\right) - \mathbf{z}\right\|^2$$

NLS – Levenberg-Marquardt Method

Jacobian Matrix: slow but accurate (Gradient decent)
Hessian Matrix: fast but less accurate (Gauss-Newton)



$$\Delta\mathbf{x} = -\left(\mathbf{J}_R^{\mathrm{T}}\mathbf{J}_R\right)^{-1}\mathbf{J}_R^{\mathrm{T}}\left(h\left(\mathbf{x}^{(0)}\right) - \mathbf{z}\right)$$ By Gauss-Newton

$$\left(\mathbf{J}_R^{\mathrm{T}}\mathbf{J}_R\right)\Delta\mathbf{x}_{\mathrm{GN}} = \mathbf{J}_R^{\mathrm{T}}h\left(h\left(\mathbf{x}^{(0)}\right) - \mathbf{z}\right)$$

$$\left(\mathbf{J}_R^{\mathrm{T}}\mathbf{J}_R + \mu\mathbf{I}\right)\Delta\mathbf{x}_{\mathrm{LM}} = \mathbf{J}_R^{\mathrm{T}}\left(h\left(\mathbf{x}^{(0)}\right) - \mathbf{z}\right), \mu > 0$$

μ: damping factor, to benefit between Jacobian + Hessian Matrix

If μ is very large than it becomes Gradient decent

If μ is very small than it becomes Gauss-Newton

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \Delta\mathbf{x}_{\mathrm{LM}} = \mathbf{x}^{(0)} - \left(\mathbf{J}_R^{\mathrm{T}}\mathbf{J}_R + \mu\mathbf{I}\right)^{-1}\mathbf{J}_R^{\mathrm{T}}\left(\left(h\left(\mathbf{x}^{(0)}\right) - \mathbf{z}\right)\right)$$

Open-source solvers:
Ceres (by Google) and *GSTAM (by Carnegie Mellon)*

Factor Graph Optimization in GNSS

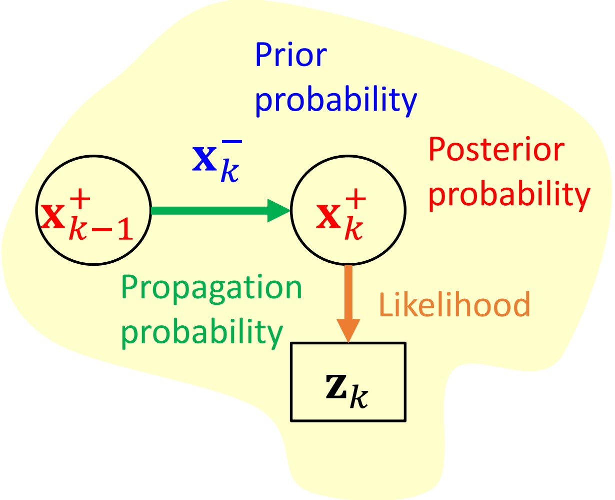
- **Example of solving the GNSS loosely-coupled pseudorange/Doppler integration using factor graph optimization using the Levenberg-Marquardt Method**
- Example of the GNSS tightly-coupled pseudorange/Doppler integration using factor graph optimization

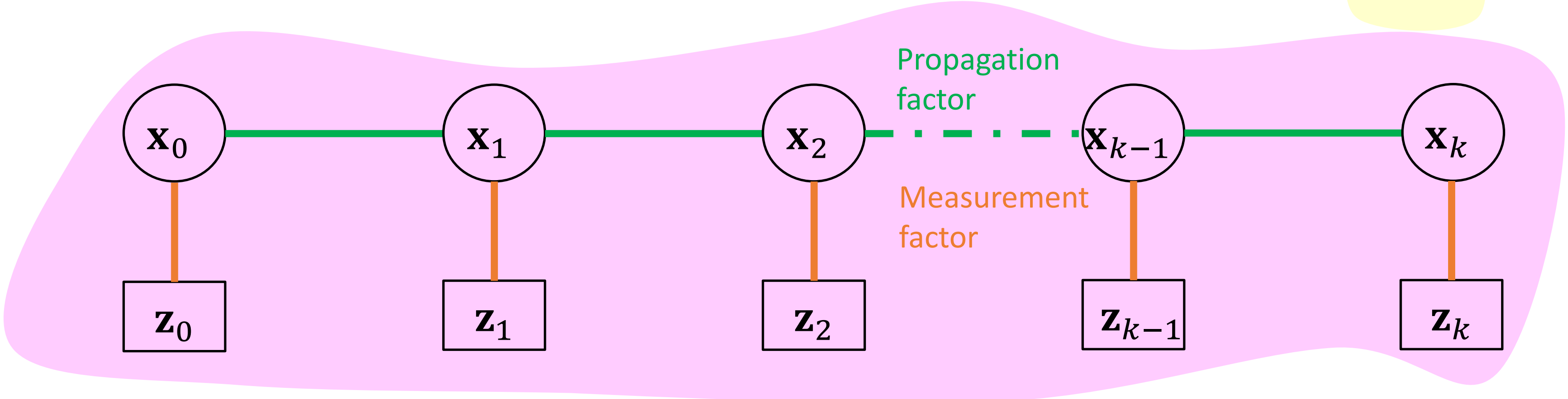
EKF vs FGO

> Both EKF and FGO are MAP

> EKF simplified MAP based on two assumptions

- 1st order Markov chain
- Gaussian random noise





Theoretical Comparison

	EKF	FGO
Assumption in Gaussian Noise	√	√*
Assumption in 1^{st} Order Markov Chain	√	
Solved by Iterative Non-linear Optimization		√

** Not strict. Only if we wish to solve FGO by NLS.*

Main differences are:

1. FGO uses batch data

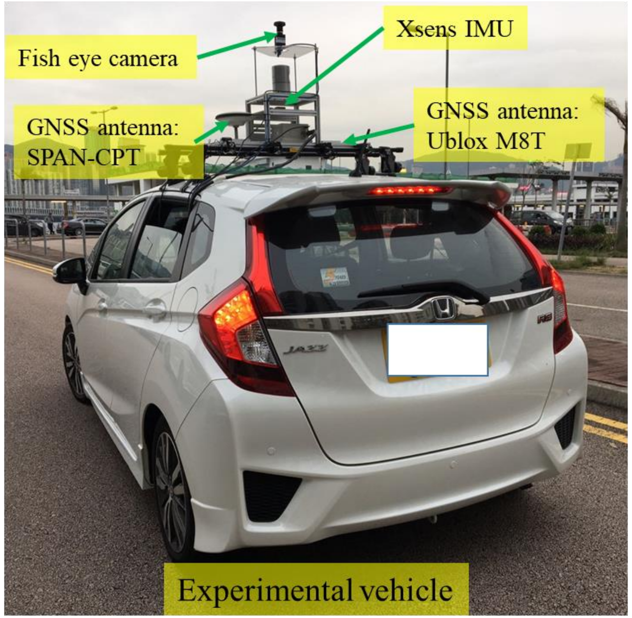
2. FGO applies iterative optimization

<u>Cons:</u>
Computational expansive

<u>Pros:</u>
Robust when two assumptions are violated

W. Wen, T. Pfeifer, X. Bai, L-T. Hsu, Factor graph optimization for GNSS/INS integration: A comparison with the extended Kalman filter. *NAVIGATION*, 2021; 68(2): 315– 331.

Experimental Setup



GNSS: ublox M8T (1Hz)
GPS L1 and Beidou B1

IMU: Xsens MTi 10 (100Hz)

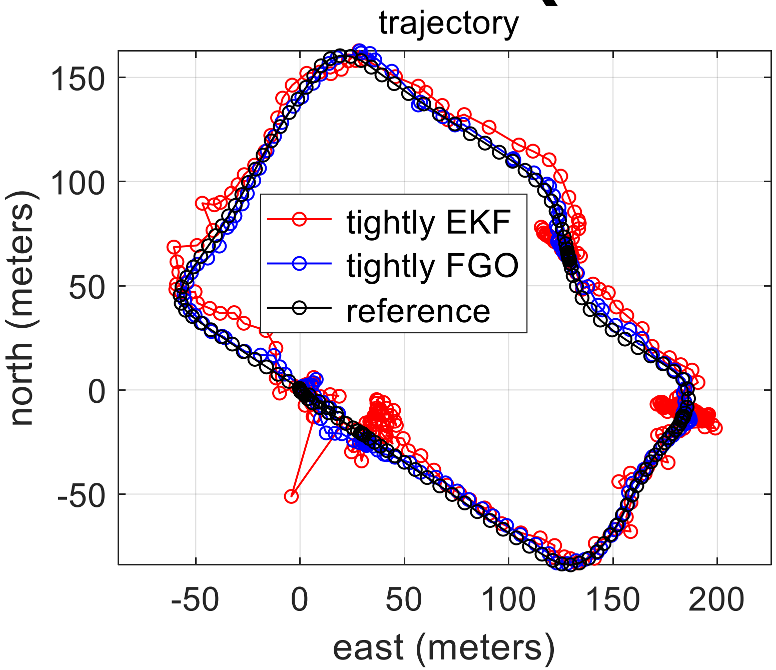
Reference: NovAtel SPAN-CPT(1Hz)

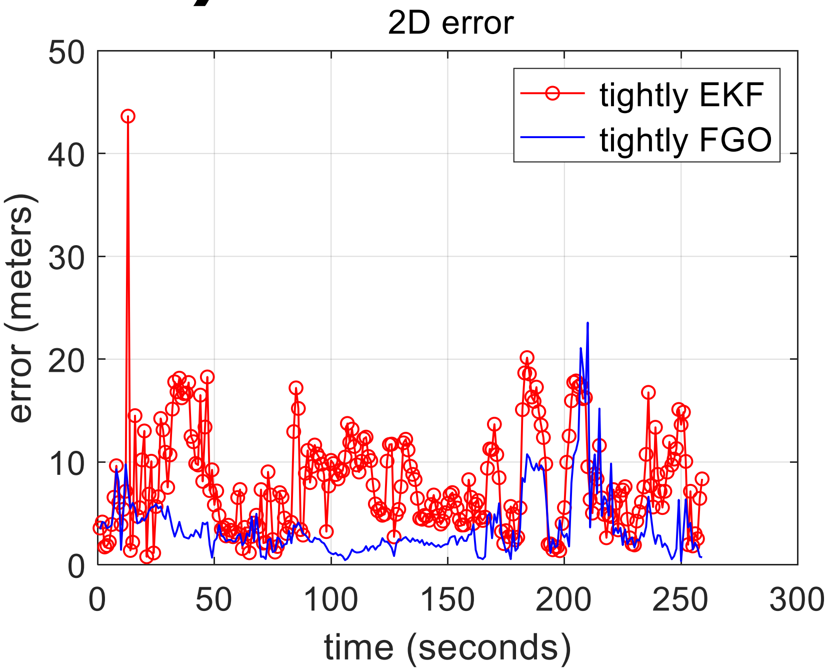
FGO solver: Linux ROS, C++, GTSAM.

EKF solver: Linux ROS, C++, Eigen.

Both EKF and FGO compared in
- **Tightly coupled (TC) GNSS/INS**

EKF vs FGO (TC GNSS-INS)



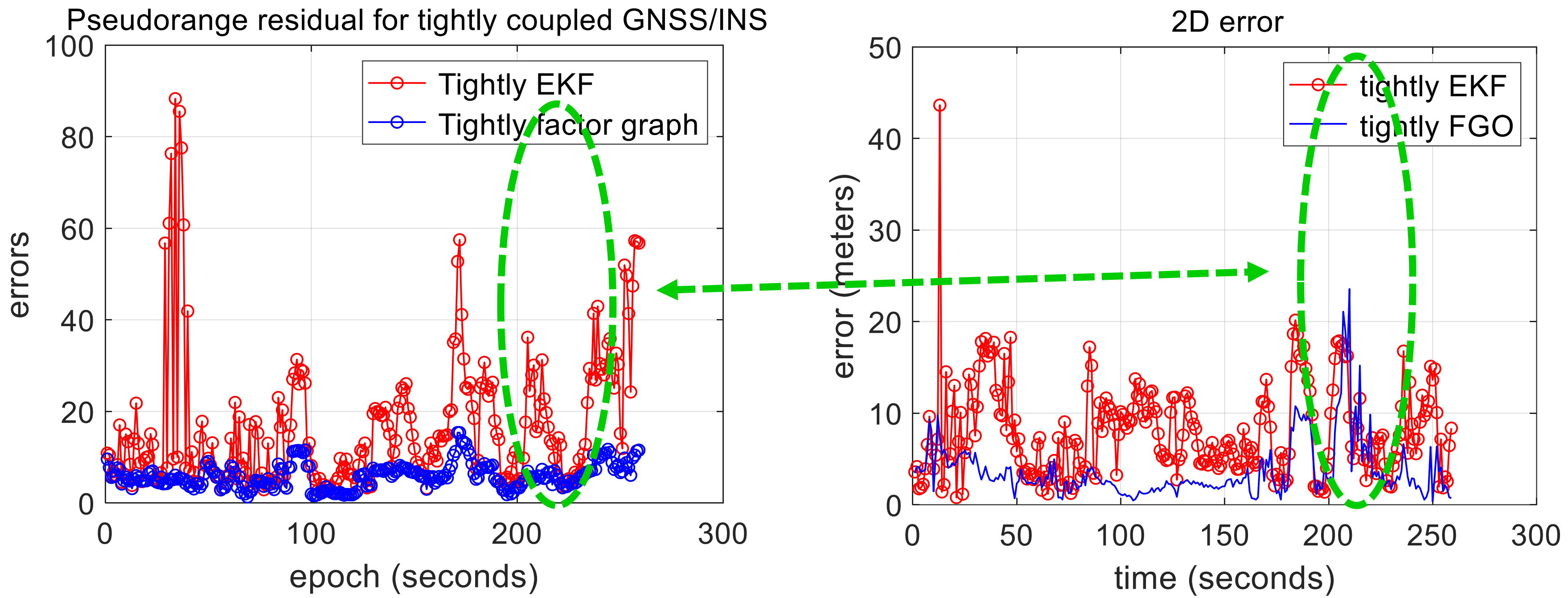


2D error: Tightly EKF vs. FGO

Mean: 8.03→ 3.64 meters

Std: 7.60 → 3.19 meters

Tightly EKF vs FGO in terms of Residual



Residual calculation

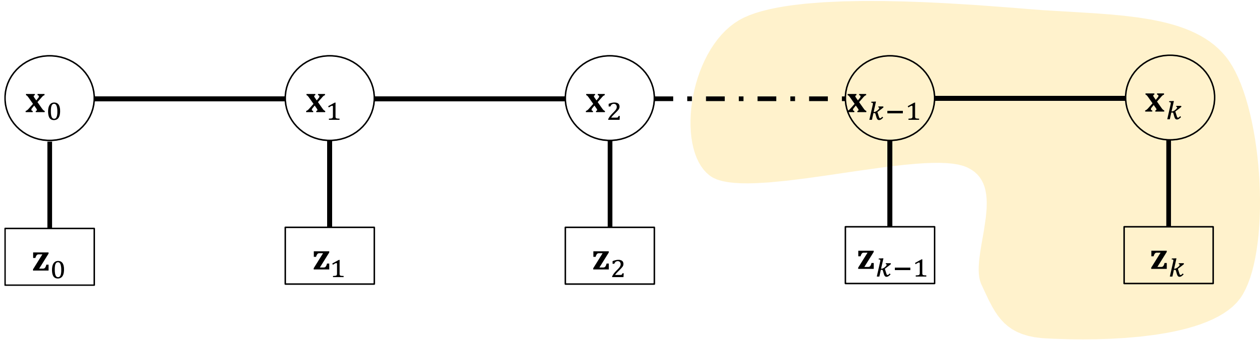
$$\rho_{r,TC} = 1/N_{sat} \sum_{i=1}^{N_{sat}} ||(\rho_{k,i}^{GNSS} - h^{GNSS,TC}(\mathbf{SV}_{k,i}, \mathbf{X}_{k,r}^{ecef}, \delta_{k,r}^{clock}))||_1$$

Residual: Tightly EKF vs. **FGO**

Mean: 16.72 → 5.98 meters

Std: 14.98 → 2.71 meters

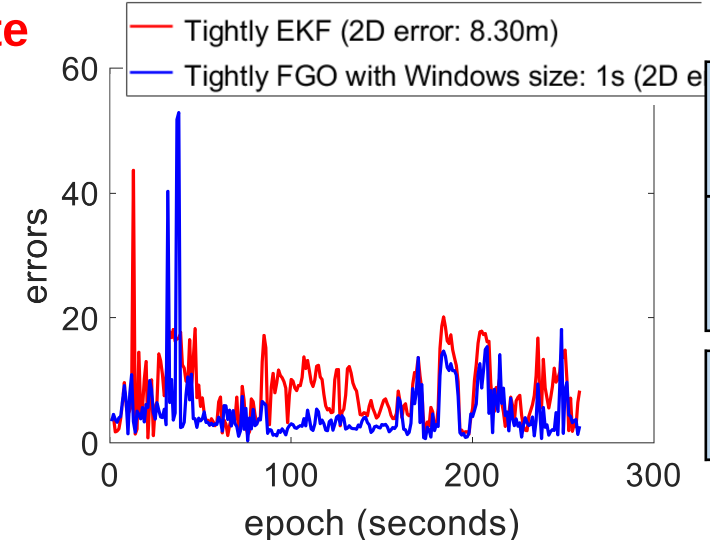
Iteration using a single epoch of data in FGO



Main differences are:

1. ~~FGO uses batch data~~
2. FGO applies iterative optimization

while EKF recursively estimate



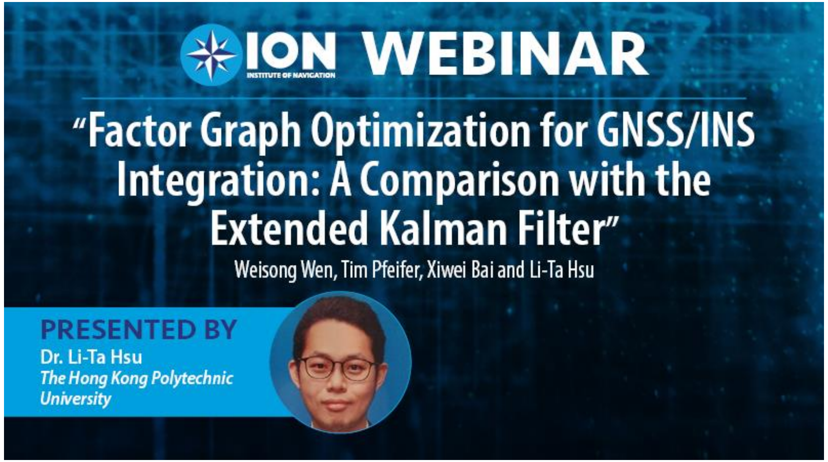
If we considering batch data

Window size	1	5	10	30	batch
FGO	5.18m	4.53m	3.95m	3.74m	3.65m

EKF	8.30 m

Theoretical Comparison

	EKF	FGO
Assumption in Gaussian Noise	√	√ (not strict)
Assumption in 1st Order Markov Chain	√	
Solved by Iterative Non-linear Optimization		√



Details (derivations and implementation)

Video can be found at ION YouTube Channel

Main differences are:

1. FGO uses batch data

2. FGO applies iterative optimization

Cons:
Computational expansive

Pros:
Robust when two assumptions are violated

W. Wen, T. Pfeifer, X. Bai, L-T. Hsu, Factor graph optimization for GNSS/INS integration: A comparison with the extended Kalman filter. *NAVIGATION*, 2021; 68(2): 315– 331.

GraphGNSSLib

2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021. Xi'an, China

Towards Robust GNSS Positioning and Real-time Kinematic Using Factor Graph Optimization

Weisong Wen and Li-Ta Hsu*

2021 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9077-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRA48506.2021.9562037

Abstract— **Global navigation satellite systems (GNSS) are one of the utterly popular sources for providing globally referenced positioning for autonomous systems. However, the performance of the GNSS positioning is significantly challenged in urban canyons, due to the signal reflection and blockage from buildings. Given the fact that the GNSS measurements are highly environmentally dependent and time-correlated, the conventional filtering-based method for GNSS positioning cannot simultaneously explore the time-correlation among historical measurements. As a result, the filtering-based estimator is sensitive to unexpected outlier measurements. In this paper, we present a factor graph-based formulation for GNSS positioning and real-time kinematic (RTK). The formulated factor graph framework effectively explores the time-correlation of pseudorange, carrier-phase, and doppler measurements, and leads to the non-minimal state estimation of the GNSS receiver. The feasibility of the proposed method is evaluated using datasets collected in challenging urban canyons of Hong Kong and significantly improved positioning accuracy is obtained, compared with the filtering-based estimator.**

I. Introduction

Global navigation satellite system (GNSS) [1] is currently one of the major sources for providing globally referenced positioning for autonomous systems with navigation requirements, such as the unmanned aerial vehicle (UAV) [2], autonomous driving vehicles (ADV) [3]. With the increased availability of multi-constellations, the GNSS solution becomes even more popular. In general, the major positioning methods involve GNSS positioning and GNSS real-time kinematic (RTK) positioning.

The popular GNSS positioning method is to use the extended Kalman filter (EKF) [4] to estimate the position, velocity, and time (PVT) of the GNSS receiver simultaneously based on the available GNSS measurements. General positioning accuracy (5~10 meters) [5] can be obtained in open sky areas. The remaining error is mainly caused by the ionosphere error, troposphere error and satellite clock/orbit biases, etc. To increase the accuracy of the GNSS positioning, RTK is proposed to perform GNSS positioning which can deliver centimeter-level positioning accuracy. The GNSS-RTK removes the errors (including the errors mentioned above and the receiver clock offset) using the double-difference technique based on the observations (e.g. pseudorange and carrier-phase measurements) received from a reference station. The GNSS-RTK positioning algorithm mainly includes two steps, the float solution estimation, and carrier-phase integer ambiguity resolution. A common approach [4] is to use an extended Kalman filter (EKF) [6] to estimate the float solution and the double-differenced (DD) carrier-phase ambiguity bias based on the DD pseudorange and carrier-phase measurements. Meanwhile, the LAMBDA algorithm [7] is employed to resolve the integer ambiguity to further achieve a fixed solution leading to centimeter-level accuracy. In short, the EKF dominates both the GNSS positioning and the GNSS-RTK positioning, due to the maturity and efficiency of the EKF estimator. Satisfactory performance can be obtained for GNSS-RTK (~5 centimeters) in open-sky areas where the error sources can be dealt with by differential techniques. Unfortunately, the performances of both the GNSS positioning and GNSS-RTK are significantly degraded in urban canyons [8] which are mainly due to the outlier measurements, arising from the multipath effects and none-line-of-sight (NLOS) [9] receptions caused by the building reflection and blockage. To mitigate the effects of GNSS outliers from NLOS receptions and multipath effects, numerous methods are studied, such as the 3D mapping aided GNSS (3DMA GNSS) [10-12], the 3D LiDAR aided GNSS positioning [13-16], and the camera aided GNSS positioning [1, 17]. However, these methods rely heavily on the availability of 3D mapping information or additional sensors.

Interestingly, instead of estimating the state of the GNSS receiver mainly based on the observation at the current epoch recursively via the EKF estimator, the recent researches [18-21] propose the factor graph-based formulation to process the GNSS pseudorange measurements and significantly improved performance is achieved, compared with the conventional EKF. The work [22] by a team from the Chemnitz University of Technology was the first paper utilizing factor graph optimization (FGO) in GNSS positioning. However, only the pseudorange measurements were considered. Then their continuous works focused on developing a robust model [23-25] for mitigating the effects of the potential NLOS receptions. Interestingly, a team from West Virginia University carried out similar researches [20, 26, 27], applying FGO models to GNSS precise point positioning (PPP) and obtaining significantly improved results. Inspired by the significant improvement arising from FGO, our previous work [28] extensively evaluated the performance of the integration of GNSS pseudorange and inertial measurement unit (IMU) using EKF and FGO. Our finding showed that the FGO could explore the time-correlation among the environment dependant GNSS pseudorange measurements simultaneously, leading to improved robustness against outliers, compared with the EKF-based estimator. However, the potential of FGO in

Weisong Wen and Li-Ta Hsu are with the Hong Kong Polytechnic University, Hong Kong. (corresponding author to provide e-mail: lt.hsu@polyu.edu.hk).

$$\rho^s_{r,t} = r^s_{r,t} + c(\delta_{r,t} - \delta^s_{r,t}) + I^s_{r,t} + T^s_{r,t} + \varepsilon^s_{r,t} \tag{1}$$

where $r^s_{r,t}$ is the geometric range between the satellite and the GNSS receiver. $I^s_{r,t}$ represents the ionospheric delay distance; $T^s_{r,t}$ indicates the tropospheric delay distance. $\varepsilon^s_{r,t}$ represents the errors caused by the multipath effects, NLOS receptions, receiver noise, antenna phase-related noise. In this paper, the satellite systems that we used include the global positioning system (GPS) and BeiDou. Besides, we follow the methods used in RTKLIB [4] to model the atmosphere effects ($I^s_{r,t}$ and $T^s_{r,t}$).

Given the Doppler measurement $(d^1_{r,t}, d^2_{r,t}, \ldots)$ of each satellite at an epoch t, the velocity $(\mathbf{v}_{r,t})$ of the GNSS receiver can be calculated using the LS method [30]. Giving that the state of the velocity, $\mathbf{x}^d_t$, is as follows:

$$\mathbf{x}^d_t = (\mathbf{v}_{r,t}, \delta_{r,t})^T \tag{2}$$

where the $\mathbf{v}_{r,t}$ represents the velocity of the GNSS receiver. The variable $\delta_{r,t}$ stands for the receiver clock drift. The range rate measurement vector $(\mathbf{y}^d_{r,t})$ at an epoch t is expressed as follows:

$$\mathbf{y}^d_{r,t} = (\lambda d^1_{r,t}, \lambda d^2_{r,t}, \lambda d^3_{r,t}, \ldots)^T \tag{3}$$

where the λ denotes the carrier wavelength of the satellite signal, the $d^s_{r,t}$ represents the Doppler measurement. The observation function $h^d(*)$ which connects the state and the Doppler measurements are expressed as follows:

$$h^d(\mathbf{x}^d_t) = \begin{bmatrix} rr^1_{r,t} + \delta_{r,t} - \delta^1_{r,t} \\ rr^2_{r,t} + \delta_{r,t} - \delta^2_{r,t} \\ rr^3_{r,t} + \delta_{r,t} - \delta^3_{r,t} \\ \cdots \\ rr^m_{r,t} + \delta_{r,t} - \delta^m_{r,t} \end{bmatrix} \tag{4}$$

$$\text{With } \mathbf{y}^d_{r,t} = h^d(\mathbf{x}^d_t) + \boldsymbol{\omega}^d_{r,t}$$

where the superscript m denotes the total number of satellites and the variable $\omega^d_{r,t}$ stands for the noise associated with the $\mathbf{y}^d_{r,t}$. The variable $rr^m_{r,t}$ denotes the expected range rate. The variables $\delta_{r,t}$ and $\delta^m_{r,t}$ represent the receiver and satellite clock bias drift. The Jacobian matrix $\mathbf{H}^d_t$ for the observation function $h^d(*)$ is denoted as follows:

$$\mathbf{H}^d_t = \begin{bmatrix} \frac{p^1_{t,x}-p_{r,t,x}}{\|\mathbf{p}^1_t-\mathbf{p}_{r,t}\|} & \frac{p^1_{t,y}-p_{r,t,y}}{\|\mathbf{p}^1_t-\mathbf{p}_{r,t}\|} & \frac{p^1_{t,z}-p_{r,t,z}}{\|\mathbf{p}^1_t-\mathbf{p}_{r,t}\|} & 1 \\ \frac{p^2_{t,x}-p_{r,t,x}}{\|\mathbf{p}^2_t-\mathbf{p}_{r,t}\|} & \frac{p^2_{t,y}-p_{r,t,y}}{\|\mathbf{p}^2_t-\mathbf{p}_{r,t}\|} & \frac{p^2_{t,z}-p_{r,t,z}}{\|\mathbf{p}^2_t-\mathbf{p}_{r,t}\|} & 1 \\ \frac{p^3_{t,x}-p_{r,t,x}}{\|\mathbf{p}^3_t-\mathbf{p}_{r,t}\|} & \frac{p^3_{t,y}-p_{r,t,y}}{\|\mathbf{p}^3_t-\mathbf{p}_{r,t}\|} & \frac{p^3_{t,z}-p_{r,t,z}}{\|\mathbf{p}^3_t-\mathbf{p}_{r,t}\|} & 1 \\ \cdots & \cdots & \cdots & \cdots \\ \frac{p^m_{t,x}-p_{r,t,x}}{\|\mathbf{p}^m_t-\mathbf{p}_{r,t}\|} & \frac{p^m_{t,y}-p_{r,t,y}}{\|\mathbf{p}^m_t-\mathbf{p}_{r,t}\|} & \frac{p^m_{t,z}-p_{r,t,z}}{\|\mathbf{p}^m_t-\mathbf{p}_{r,t}\|} & 1 \end{bmatrix} \tag{5}$$

where the operator $\|*\|$ is employed to calculate the range distance between the given satellite and the GNSS receiver. The expected range rate $rr^s_{r,t}$ for satellite s can also be calculated as follows:

$$rr^s_{r,t} = \mathbf{e}^{s,LOS}_{r,t}(\mathbf{v}^s_t - \mathbf{v}_{r,t}) + \frac{\omega_{earth}}{c_L}(v^s_{t,y}p_{r,t,x} + p^s_{t,y}v_{r,t,x} - p^s_{t,x}v_{r,t,y} - v^s_{t,x}p_{r,t,y}) \tag{6}$$

where the variable ω_{earth} denotes the angular velocity of the earth rotation [4]. The variable c_L denotes the speed of the light. The variable $\mathbf{e}^{s,LOS}_{r,t}$ denotes the line-of-sight vector connecting the GNSS receiver and the satellite (See equation (5)). Therefore, the velocity $(\mathbf{v}_{r,t})$ of the GNSS receiver can be estimated via LS [4] based on equations (4) and (5).

The graph structure of the proposed factor graph for solving the GNSS positioning is shown in Fig. 2. The subscript n denotes the total epochs of measurements considered in the FGO. Each state in the factor graph is connected using the Doppler velocity factor. The state of the GNSS receiver is represented as follows:

$$\boldsymbol{\chi} = [\mathbf{x}_{r,1}, \mathbf{x}_{r,2}, \ldots, \mathbf{x}_{r,n}] \tag{7}$$

$$\mathbf{x}_{r,t} = (\mathbf{p}_{r,t}, \mathbf{v}_{r,t}, \delta_{r,t})^T \tag{8}$$

where the variable $\boldsymbol{\chi}$ denotes the set states of the GNSS receiver from the first epoch to the current n. The $\mathbf{x}_{r,t}$ denotes the state of the GNSS receiver at epoch t which involves the position $(\mathbf{p}_{r,t})$, velocity $(\mathbf{v}_{r,t})$ and receiver clock bias $(\delta_{r,t})$.

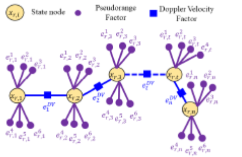


Fig. 2 The purple circle denotes the pseudorange factor (e.g $e^s_{r,t}$). The blue shaded rectangle represents the Doppler velocity factor (e.g e^{DV}_t). The yellow shaded circle stands for the state of the GNSS receiver.

The observation model for GNSS pseudorange measurement from a given satellite s is represented as follows:

$$\rho^s_{r,t} = h^s_{r,t}(\mathbf{p}_{r,t}, \mathbf{p}^s_t, \delta_{r,t}) + \omega^s_{r,t} \tag{9}$$

$$\text{with } h^s_{r,t}(\mathbf{p}_{r,t}, \mathbf{p}^s_t, \delta_{r,t}) = \|\mathbf{p}^s_t - \mathbf{p}_{r,t}\| + \delta_{r,t}$$

where the variable $\omega^s_{r,t}$ stands for the noise associated with the $\rho^s_{r,t}$. Therefore, we can get the error function $(\mathbf{e}^s_{r,t})$ for a given satellite measurement $\rho^s_{r,t}$ as follows:

$$\|\mathbf{e}^s_{r,t}\|^2_{\boldsymbol{\Sigma}^s_{r,t}} = \|\rho^s_{r,t} - h^s_{r,t}(\mathbf{p}_{r,t}, \mathbf{p}^s_t, \delta_{r,t})\|^2_{\boldsymbol{\Sigma}^s_{r,t}} \tag{10}$$

where $\boldsymbol{\Sigma}^s_{r,t}$ denotes the covariance matrix. We calculate the $\boldsymbol{\Sigma}^s_{r,t}$ based on the satellite elevation angle, signal, and noise ratio (SNR) following the work in [31]. The observation model for the velocity $(\mathbf{v}_{r,t})$ is expressed as follows:

$$\mathbf{v}^{DV}_{r,t} = h^{DV}_{r,t}(\mathbf{x}_{r,t+1}, \mathbf{x}_{r,t}) + \boldsymbol{\omega}^{DV}_{r,t} \tag{11}$$

mean error decreases to only 9.45 meters after applying the FGO with a significantly decreased STD of 8.06 meters. Meanwhile, the maximum error decreases to only 31.94 meters. The significantly improved positioning accuracy shows the effectiveness of the proposed framework based on FGO.

Fig. 5 shows the trajectories using three different methods and ground truth. The black curve denotes the ground truth trajectory provided by the SPAN-CPT. The smoother trajectory is achieved using EKF with the help of velocity measurements, compared with the WLS. However, the trajectory can still deviate significantly from the ground truth trajectory in some epochs. With the help of the proposed framework, a smoother trajectory is obtained almost throughout the test.

Table.1 GNSS positioning performance using the three listed methods

All data	WLS	EKF	FGO
MEAN (m)	17.39	13.61	9.45
STD (m)	16.01	15.19	8.06
MAX (m)	94.43	88.97	31.94
Availability	100%	100%	100%

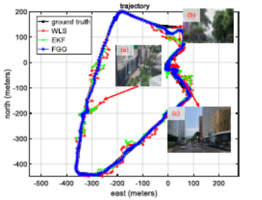


Fig. 5 Trajectories of three methods WLS (red), EKF (green), and FGO (blue). The x-axis and y-axis denote the east and north directions, respectively.

B. Evaluation of the Proposed GNSS-RTK

During the static test in urban canyon 2, the Doppler velocity measurements are employed to connects the consecutive states. The positioning accuracy of GNSS-RTK in the evaluated dataset is shown in the following Table 2. Be noted that the float solution is recorded when the fixed solution is not available. A mean of 2.01 meters is obtained using RTK-EKF with an STD of 0.67 meters. Meanwhile, the maximum error reaches 3.33 meters. The mean error decreases to only 0.64 meters after applying the RTK-FGO with a slightly decreased STD of 0.40 meters. Meanwhile, the maximum error decreases to only 1.70 meters. The improved positioning accuracy shows that the proposed RTK-FGO method can effectively mitigate the effects of the GNSS outlier measurements, leading to improved accuracy.

methods and ground truth. The more accurate trajectory is achieved using RTK-FGO with the help of velocity measurements, compared with the RTK-EKF. We can see from Table 2 that the RTK-EKF gets a fixed rate of 4.4% in the evaluated urban canyon 2. Interestingly, the fixed rate of the proposed RTK-FGO is slightly decreased to 3.8%. The reason is that the proposed RTK-FGO did not consider the cycle slip detection [33] and the ambiguity is solved independently in each epoch.

Table.2 Positioning performance of the GNSS-RTK

All data	RTK-EKF	RTK-FGO
MEAN (m)	2.01	0.64
STD (m)	0.67	0.40
MAX (m)	3.33	1.70
Availability	100%	100%
Fixed Rate	4.4%	3.8%

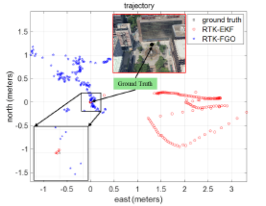


Fig. 6 Positioning results of two methods RTK-EKF (red dots) and RTK-FGO (blue dots).

V. Conclusion and Future Work

This paper developed a factor graph-based formulation, that enables the capability of the two most popular positioning methods, the GNSS positioning, and GNSS-RTK. We evaluate the proposed framework using the dataset collected in the urban canyons of Hong Kong. The results show that the proposed method can effectively help to mitigate the effects of GNSS outlier measurements, deriving improved accuracy both in GNSS positioning and GNSS-RTK positioning. The cycle slip detection will be applied to the integer ambiguity resolution to improve the fixed rate in the future. Moreover, achieving a fixed solution for positioning autonomous systems in the urban canyon is still a challenging problem to solve, we will also explore adding more sensors to the proposed framework to increase the fixed rate of GNSS-RTK.

Acknowledgment

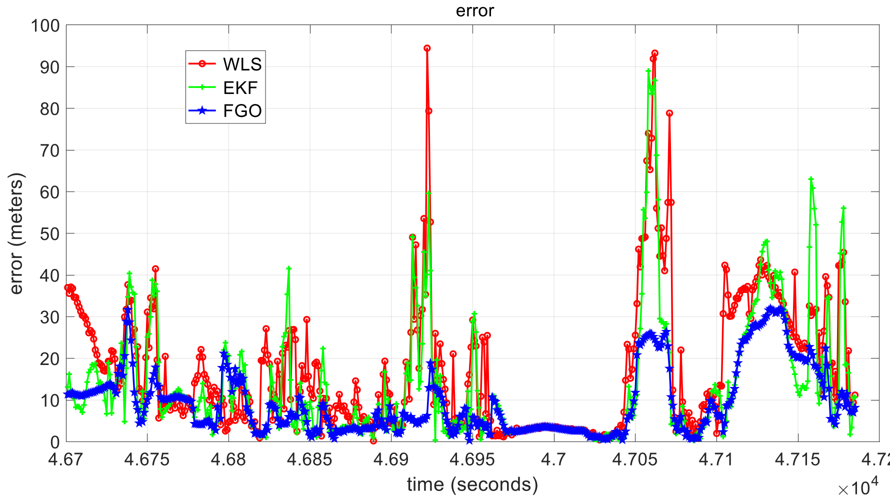
The authors acknowledge the ROS, RTKLIB, and the provider of OpenStreetMap.

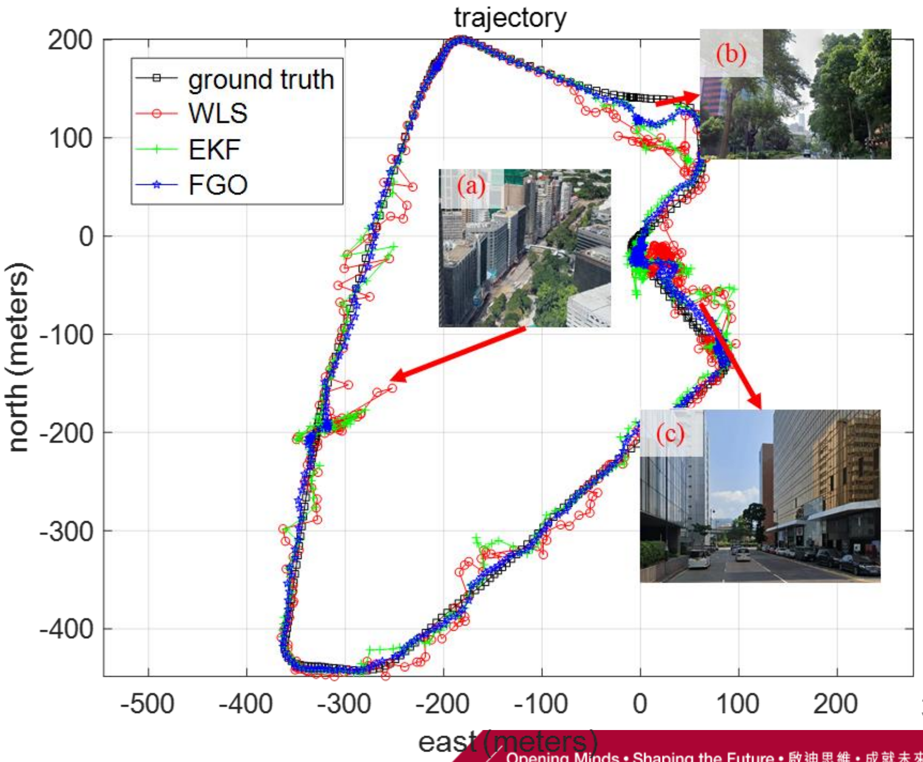
Evaluation of GNSS Positioning

WLS*: weighted least square with pseudorange
EKF*: Pseudorange/Doppler fusion with extended Kalman filter
FGO*: Pseudorange/Doppler fusion with factor graph optimization

GNSS positioning performance using the three listed methods

All data	WLS	EKF	FGO
MEAN (m)	17.39	13.61	9.45
STD (m)	16.01	15.19	8.06
MAX (m)	94.43	88.97	31.94
Availability	100%	100%	100%



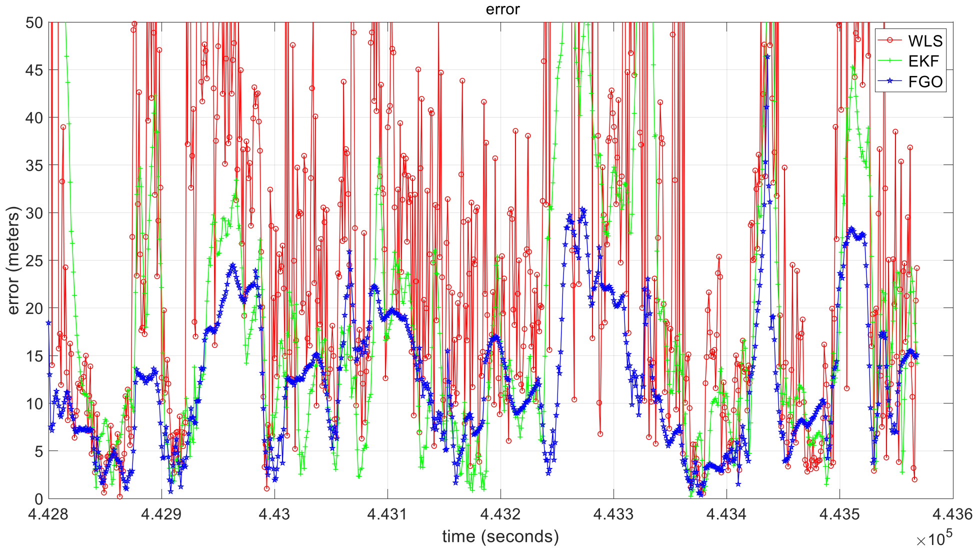


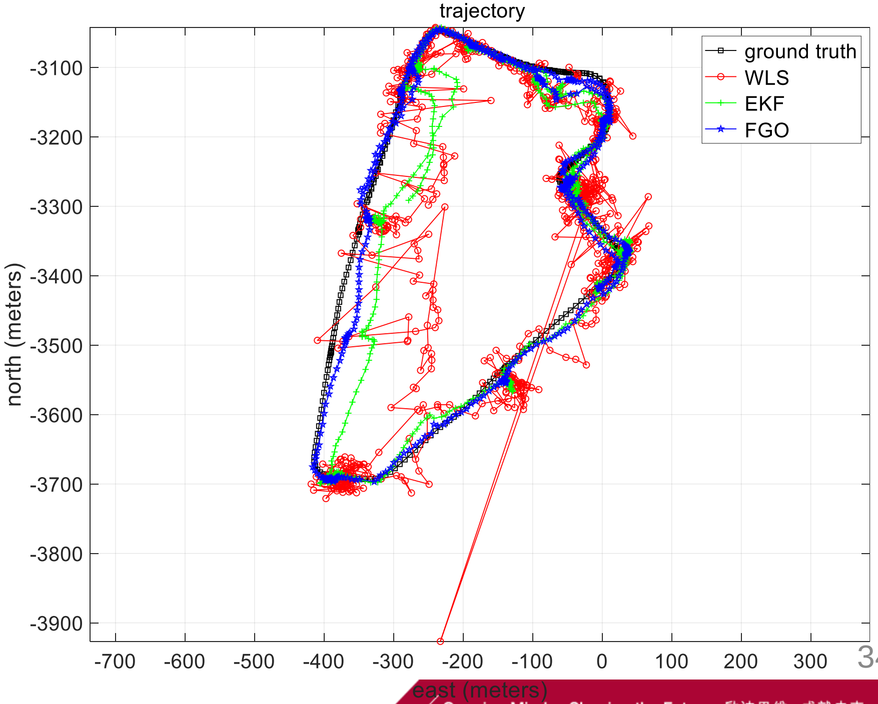
Evaluation with Huawei P40 Pro

Huawei P40 Pro Phone

WLS*: weighted least square with pseudorange
EKF*: Pseudorange/Doppler fusion with extended Kalman filter
FGO*: Pseudorange/Doppler fusion with factor graph optimization

All data	WLS	EKF	FGO
MEAN (m)	31.98	19.84	12.541
STD (m)	38.22	15.78	7.48
MAX (m)	701.7	77.28	46.36





Q&A

Thank you for your attention ☺

Q&A

Dr. Weisong Wen

If you have any questions or inquiries, please feel free to contact me.

Email: welson.wen@polyu.edu.hk

Supplementary: GNSS/INS Integration Using **Kalman Filtering** and **Factor Graph Optimization**

Inertial navigation system

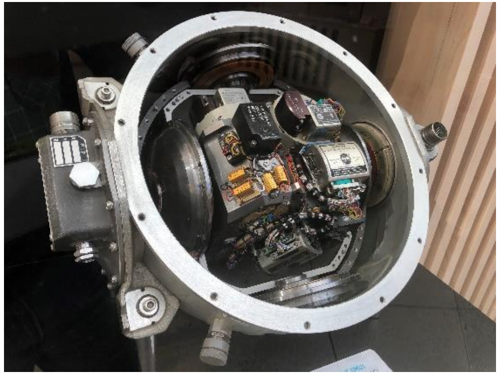
$\hat{\mathbf{a}}_t$, $\hat{\boldsymbol{\omega}}_t$ are the raw accelerometer and gyroscope measurements in the body frame
$\mathbf{a}_t$, $\boldsymbol{\omega}_t$ are expected measurements

The cap ^ denotes the noisy measurement or estimation of a certain quantity

$$\hat{\mathbf{a}}_t = \mathbf{a}_t + \mathbf{R}_w^t \mathbf{g}^w + \mathbf{b}_{a_t} + \mathbf{n}_a \quad (1)$$

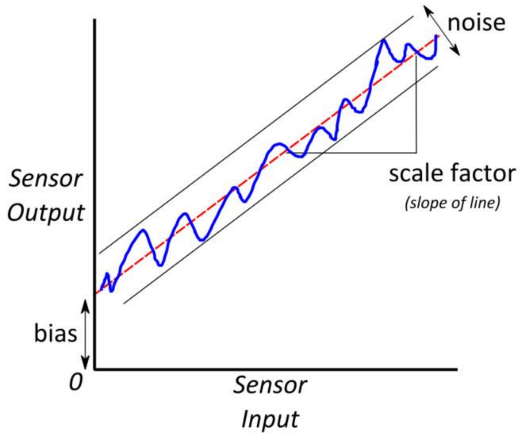
$$\hat{\boldsymbol{\omega}}_t = \boldsymbol{\omega}_t + \mathbf{b}_{\omega_t} + \mathbf{n}_\omega \quad (2)$$

$$\mathbf{n}_a \sim \mathcal{N}(0, \sigma_a^2), \mathbf{n}_\omega \sim \mathcal{N}(0, \sigma_\omega^2)$$



Error analysis of inertial navigation system

- The errors of **accelerometer and gyroscope** can be divided into: **deterministic error & random error.**
- Deterministic errors can be calibrated in advance including bias, scale...
- Random error usually assumes that noise obeys Gaussian distribution, including Gaussian white noise, bias random walk...



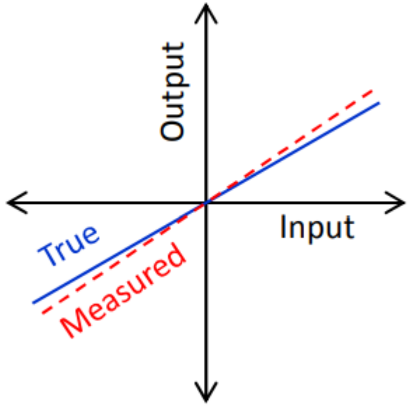
Common systematic errors in IMU
bias
noise
scale factor

Deterministic error
(sourcing imperfectness of electrical/mechanical components)

> Bias: In theory, the output of the IMU sensor should be 0 when there is no external action. However, there is a bias $\mathbf{b}$ to the international data. Influence of accelerometer bias on orientation estimation:

$$\mathbf{V}_{\mathbf{error}} = \mathbf{b}_{\mathbf{a}} t, \mathbf{P}_{\mathbf{error}} = \frac{1}{2} \mathbf{b}_{\mathbf{a}} t^2$$

> Scale: The ratio between the actual value and the sensor output value.



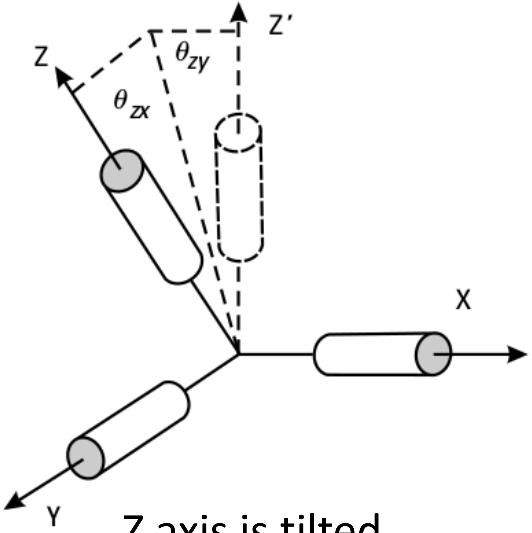
Deterministic error (sourcing imperfectness of installation)

> Nonorthogonality/Misalignment Errors: When manufacturing multi-axis IMU sensors, due to the manufacturing process, the xyz axis may not be vertical.

Scale + Misalignment

$$\begin{bmatrix} l_{ax} \\ l_{ay} \\ l_{az} \end{bmatrix} = \begin{bmatrix} s_{xx} & m_{xy} & m_{xz} \\ m_{yz} & s_{yy} & m_{yz} \\ m_{zx} & m_{zy} & s_{zz} \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$$

Measured Acc — True Acc



Z axis is tilted

Deterministic error calibration method—Accelerometer

> The six-sided method means that the three axes of the accelerometer are placed horizontally up or down for a period of time, and data on the six sides are collected to complete the calibration.

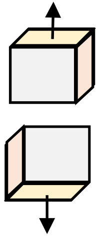
If the axes are orthogonal, it is easy to get bias and scale:

$$\mathbf{b_a} = \frac{\mathbf{l_f^{up}} + \mathbf{l_f^{down}}}{2}$$

$$\mathbf{s_a} = \frac{\mathbf{l_f^{up}} - \mathbf{l_f^{down}}}{2 \cdot \mathbf{g}} = \begin{bmatrix} s_{a,xx} \\ s_{a,yy} \\ s_{a,zz} \end{bmatrix}$$

$\mathbf{l_f^{up}}$

$\mathbf{l_f^{dowb}}$



$\mathbf{l}$ is the measured value of a certain axis of the accelerometer, $\mathbf{g}$ is the local gravity acceleration

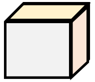
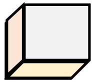
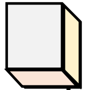
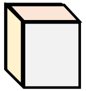
Deterministic error calibration method—Accelerometer

> When considering the inter-axis error, the relationship between the actual acceleration and the measured value is:

$$\begin{bmatrix} l_{ax} \\ l_{ay} \\ l_{az} \end{bmatrix} = \begin{bmatrix} s_{xx} & m_{xy} & m_{xz} \\ m_{yz} & s_{yy} & m_{yz} \\ m_{zx} & m_{zy} & s_{zz} \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} + \begin{bmatrix} b_{ax} \\ b_{ay} \\ b_{az} \end{bmatrix}$$

> When placed horizontally and statically on 6 sides, the theoretical value of acceleration is

$$\mathbf{a_1} = \begin{bmatrix} g \\ 0 \\ 0 \end{bmatrix}, \mathbf{a_2} = \begin{bmatrix} -g \\ 0 \\ 0 \end{bmatrix}, \mathbf{a_3} = \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix}, \mathbf{a_4} = \begin{bmatrix} 0 \\ -g \\ 0 \end{bmatrix}, \mathbf{a_5} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}, \mathbf{a_6} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}$$

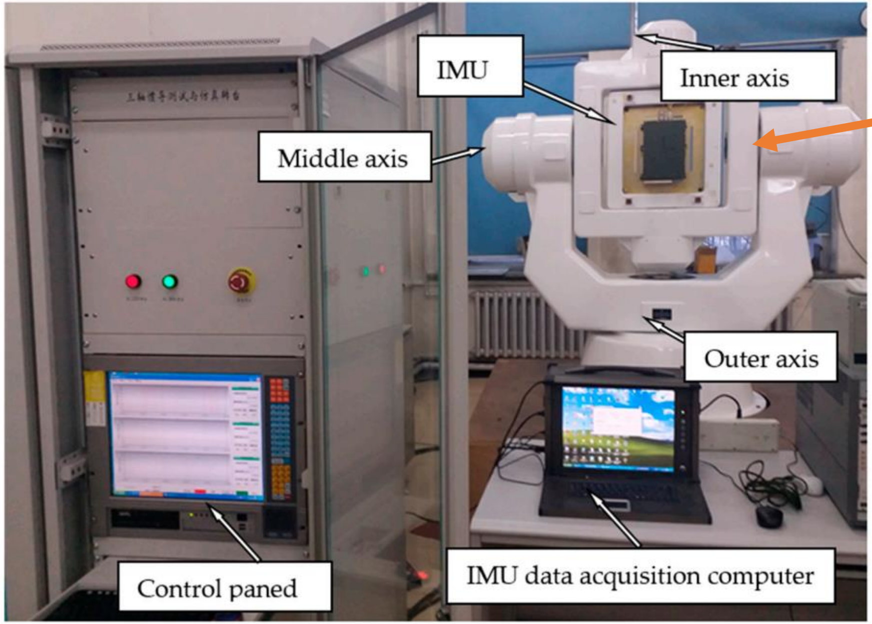
> Corresponding measurement value matrix $\mathbf{L}$

$$\mathbf{L} = [\boldsymbol{l_1} \quad \boldsymbol{l_2} \quad \boldsymbol{l_3} \quad \boldsymbol{l_4} \quad \boldsymbol{l_5} \quad \boldsymbol{l_6}]$$

> 12 variables can be obtained by using least squares.

Deterministic error calibration method——Gyroscope

> Unlike the six-sided method of accelerometer, the true value of the gyroscope is provided by a high-precision turntable. The 6 faces in this refer to the clockwise and counterclockwise rotation of each axis



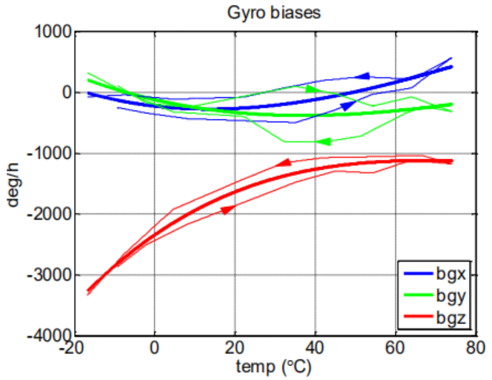
high-precision three-axis turntable

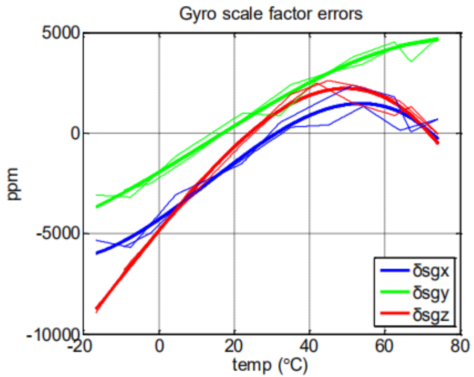
Random error – Unstableness of electrical and mechanical component due to temperature

> We can calibrate to do temperature compensation on the bias and scale estimated by the sensor, and to obtain the values of bias and scale at different temperatures and draw them into a curve.

> Soak method: control the temperature value of **the constant temperature room**, and then read the sensor value for calibration.

The thin solid lines are the results under separated heating and cooling processes The thick lines are the final curve fitted result .

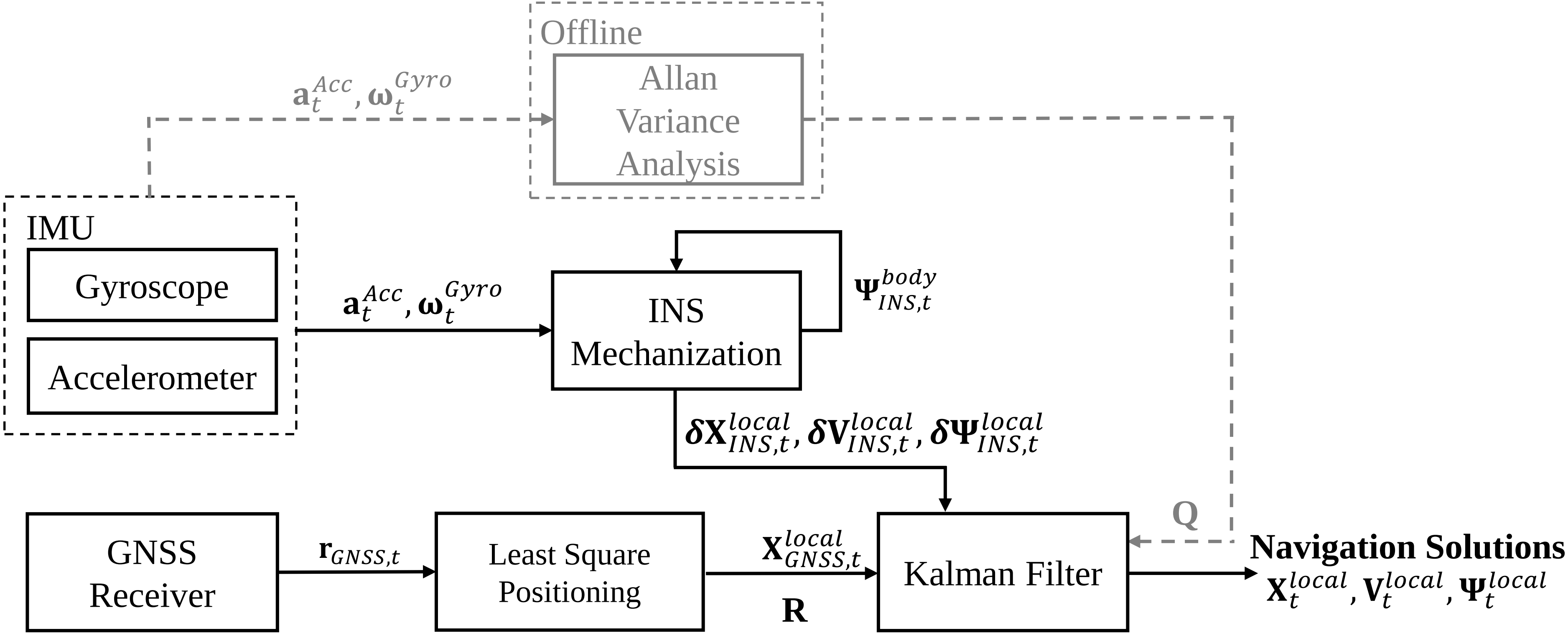




Nomenclature

- $\mathbf{a}_t^{Acc}$:measured 3-axis accelerations by the accelerometers at epoch t
- $\boldsymbol{\omega}_t^{Gyro}$:measured 3-axis rotation by the gyroscopes at epoch t
- $\mathbf{X}_{INS,t}^{body}$:estimated 3-axis position in body frame by INS at epoch t
- $\mathbf{V}_{INS,t}^{body}$:estimated 3-axis velocity in body frame by INS at epoch t
- $\boldsymbol{\Psi}_{INS,t}^{body}$:estimated 3-axis orientation in body frame (Euler angles) by INS at epoch t
- $\mathbf{B}_{a,t}^{body}$:estimated 3−axis biaes of accelerometers in body frame at epoch t
- $\mathbf{B}_{\omega,t}^{body}$:estimated 3−axis biaes of gyroscopes in body frame at epoch t
- $\boldsymbol{W}_{b_a}$:estimated 3−axis random walk noise of accelerometers in body frame
- $\boldsymbol{W}_{b_a}$:estimated 3−axis random walk noise of gyroscopes in body frame

Open Loop


Offline
Allan Variance Analysis
$\mathbf{a}_t^{Acc}, \boldsymbol{\omega}_t^{Gyro}$
IMU
Gyroscope
Accelerometer
$\mathbf{a}_t^{Acc}, \boldsymbol{\omega}_t^{Gyro}$
INS Mechanization
$\boldsymbol{\Psi}_{INS,t}^{body}$
$\boldsymbol{\delta}\mathbf{X}_{INS,t}^{local}, \boldsymbol{\delta}\mathbf{V}_{INS,t}^{local}, \boldsymbol{\delta}\boldsymbol{\Psi}_{INS,t}^{local}$
Q
GNSS Receiver
$\mathbf{r}_{GNSS,t}$
Least Square Positioning
$\mathbf{X}_{GNSS,t}^{local}$
R
Kalman Filter
Navigation Solutions
$\mathbf{X}_t^{local}, \mathbf{V}_t^{local}, \boldsymbol{\Psi}_t^{local}$

INS Mechanization

Sola, Joan. "Quaternion kinematics for the error-state Kalman filter." *arXiv preprint arXiv:1711.02508* (2017).

The Euler angle rates obtained by angular velocity:

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \dot{\boldsymbol{\Psi}}_{INS,t}^{body} = \begin{bmatrix}1 & sin(\phi)\,tan(\theta) & cos(\phi)\,tan(\theta)\\ 0 & cos(\phi) & -sin(\phi)\\ 0 & sin(\phi)\,sec(\theta) & cos(\phi)\,sec(\theta)\end{bmatrix}\boldsymbol{\omega}_t^{Gyro} = \begin{bmatrix}p\\ q\\ r\end{bmatrix}$$

$$\boldsymbol{\delta\Psi}_{INS,t}^{body} = \dot{\boldsymbol{\Psi}}_{INS,t}^{body}\Delta t = \begin{bmatrix}\delta\phi\\ \delta\theta\\ \delta\psi\end{bmatrix}$$

Rotate the Euler angles from Body to Local

$$\boldsymbol{\Psi}_{INS,t}^{local} = \mathbf{R}_{body}^{local}\boldsymbol{\Psi}_{INS,t}^{body}$$

Update the current Euler angles

$$\boldsymbol{\Psi}_{INS,t}^{body} = \boldsymbol{\Psi}_{INS,t-1}^{body} + \boldsymbol{\delta\Psi}_{INS,t}^{body}$$

$$\begin{aligned}\mathbf{R}_{body}^{local} &= \mathbf{R}(X,\phi)\mathbf{R}(Y,\theta)\mathbf{R}(Z,\psi)\\ &= \begin{bmatrix}1 & 0 & 0\\ 0 & cos(\phi) & -sin(\phi)\\ 0 & sin(\phi) & cos(\phi)\end{bmatrix}\begin{bmatrix}cos(\theta) & 0 & sin(\theta)\\ 0 & 1 & 0\\ -sin(\theta) & 0 & cos(\theta)\end{bmatrix}\begin{bmatrix}cos(\psi) & -sin(\psi) & 0\\ sin(\psi) & cos(\psi) & 0\\ 0 & 0 & 1\end{bmatrix}\end{aligned}$$

INS Mechanization

To remove the gravity from the acceleration

$$\mathbf{a}_{INS,t}^{local} = \mathbf{R}_{body}^{local}\mathbf{a}_{INS,t}^{body} - \mathbf{g}$$

To obtain the change of aircraft in terms of position, velocity and orientation

$$\boldsymbol{\delta}\mathbf{X}_{INS,t}^{local} = \mathbf{X}_{INS,t}^{local} - \mathbf{X}_{INS,t-1}^{local} = \mathbf{V}_{INS,t-1}^{local}\Delta t + \frac{\boldsymbol{a}_{INS,t}^{local}\Delta t^2}{2}$$

$$\boldsymbol{\delta}\mathbf{V}_{INS,t}^{local} = \mathbf{V}_{INS,t}^{local} - \mathbf{V}_{INS,t-1}^{local} = \boldsymbol{a}_{INS,t}^{local}\Delta t$$

$$\boldsymbol{\delta}\boldsymbol{\Psi}_{INS,t}^{local} = \boldsymbol{\Psi}_{INS,t}^{local} - \boldsymbol{\Psi}_{INS,t-1}^{local}$$

Kalman Filter——GNSS/INS(Open Loop)

System States:

$$\mathbf{X}_t = \left(\mathbf{X}_t^{local}, \mathbf{V}_t^{local}, \mathbf{\Psi}_t^{local}\right)$$
$$\mathbf{X}_t^{local} = \left(x_t^{local}, y_t^{local}, z_t^{local}\right)$$
$$\mathbf{V}_t^{local} = \left(vx_t^{local}, vy_t^{local}, vz_t^{local}\right)$$
$$\mathbf{\Psi}_t^{local} = \left(\phi_{roll}, \theta_{pitch}, \psi_{yaw}\right)$$

Propagation model:

$$\mathbf{X}_t^{-} = \mathbf{F}\mathbf{X}_{t-1}^{+} + \mathbf{B}\mathbf{U}_t$$

$$\mathbf{F} = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} \quad \mathbf{U}_t = \begin{bmatrix} \delta\mathbf{X}_{INS,t}^{local} \\ \delta\mathbf{V}_{INS,t}^{local} \\ \delta\mathbf{\Psi}_{INS,t}^{local} \end{bmatrix}$$

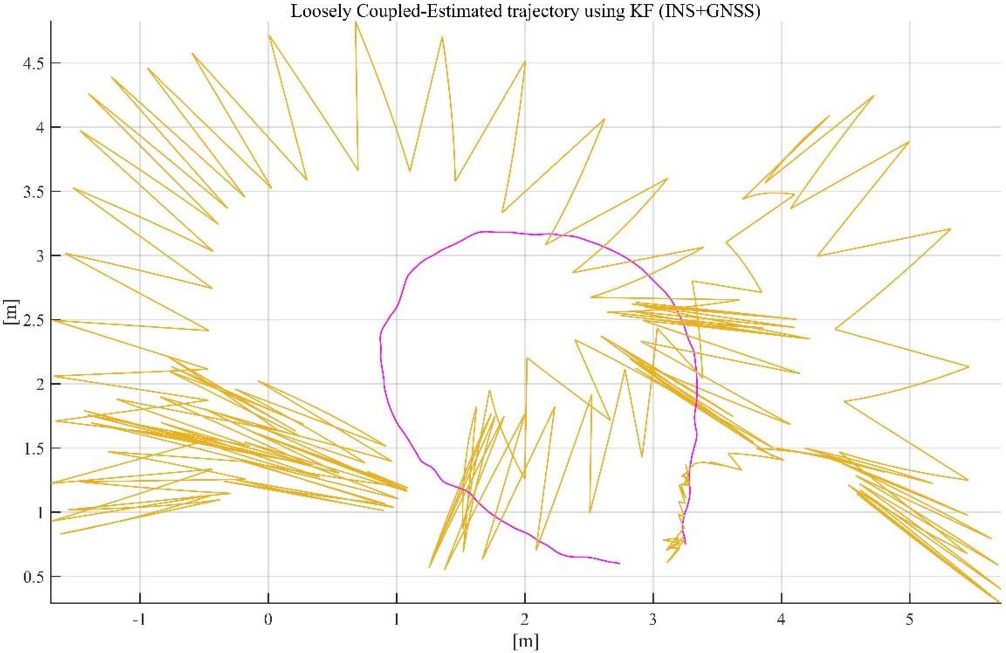
Measurement model:

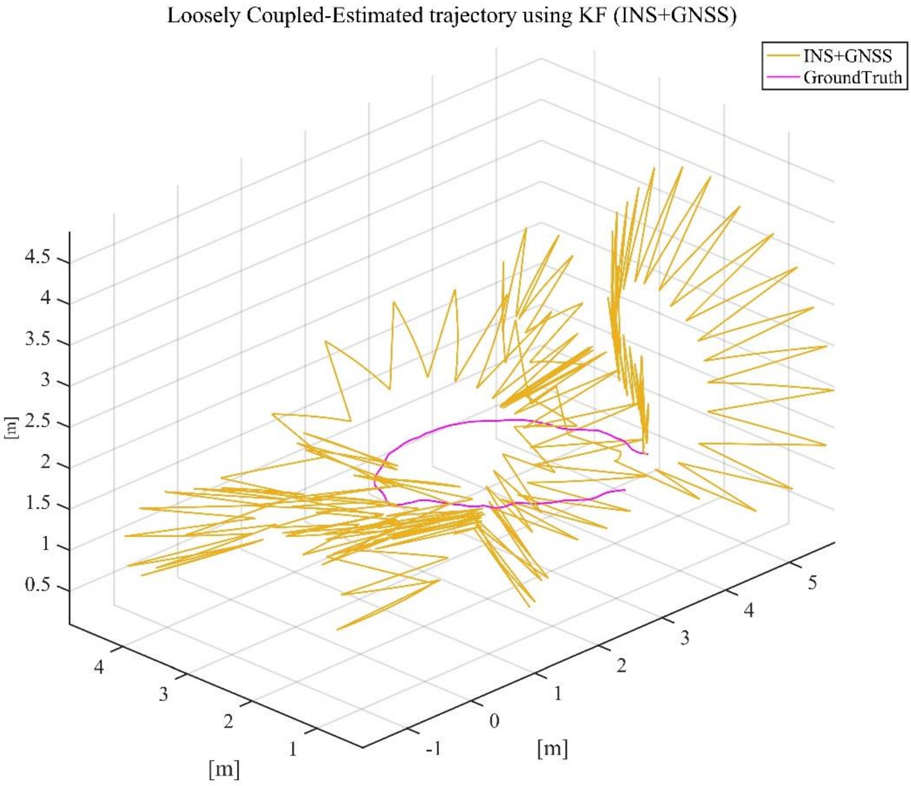
$$\Delta\mathbf{Z}_t = \mathbf{Z}_t - \mathbf{H}\mathbf{X}_t^{-}$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_{3\times3} & \cdots & \mathbf{0} \\ \vdots & \mathbf{0} & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}$$

$$\mathbf{Z}_t = \mathbf{X}_{GNSS,t}^{local} = \begin{bmatrix} x_{GNSS,t}^{local} \\ y_{GNSS,t}^{local} \\ z_{GNSS,t}^{local} \end{bmatrix}$$

Open Loop


Loosely Coupled-Estimated trajectory using KF (INS+GNSS)
[m]
[m]


Loosely Coupled-Estimated trajectory using KF (INS+GNSS)
INS+GNSS
GroundTruth
[m]
[m]
[m]

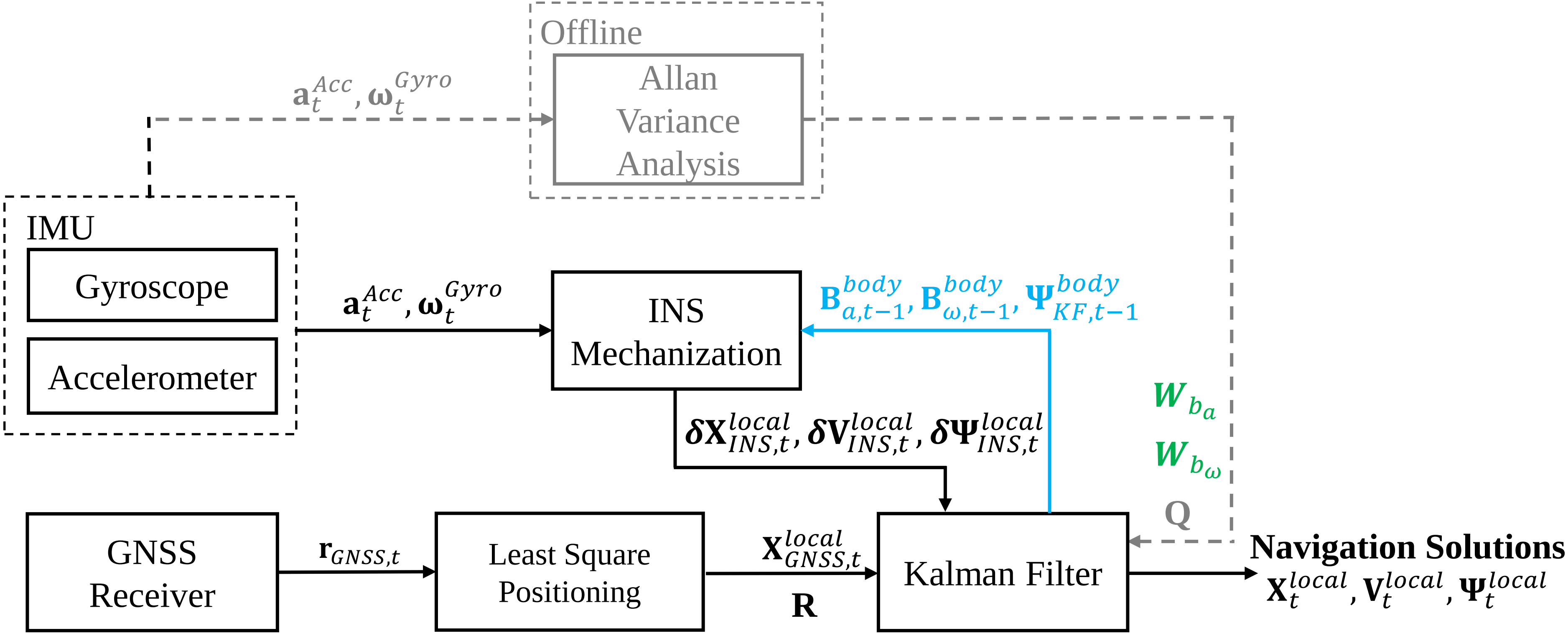
Closed-loop Correction

> The estimated position, velocity, and attitude errors are fed back to the inertial navigation processor, where they are used to correct the inertial navigation solution itself.

> any accelerometer and gyro errors estimated by the Kalman filter are fed back to correct the IMU measurements, as they are input to the inertial navigation equations.

> Unlike the position, velocity, and attitude corrections, the accelerometer and gyro corrections must be applied on every iteration

Closed Loop


Offline
Allan Variance Analysis
$\mathbf{a}_t^{Acc}, \boldsymbol{\omega}_t^{Gyro}$
IMU
Gyroscope
Accelerometer
$\mathbf{a}_t^{Acc}, \boldsymbol{\omega}_t^{Gyro}$
INS Mechanization
$\mathbf{B}_{a,t-1}^{body}, \mathbf{B}_{\omega,t-1}^{body}, \boldsymbol{\Psi}_{KF,t-1}^{body}$
$\boldsymbol{\delta X}_{INS,t}^{local}, \boldsymbol{\delta V}_{INS,t}^{local}, \boldsymbol{\delta \Psi}_{INS,t}^{local}$
W_{ba}
$W_{b\omega}$
Q
GNSS Receiver
$\mathbf{r}_{GNSS,t}$
Least Square Positioning
$\mathbf{X}_{GNSS,t}^{local}$
R
Kalman Filter
Navigation Solutions
$\mathbf{X}_t^{local}, \mathbf{V}_t^{local}, \boldsymbol{\Psi}_t^{local}$

INS Mechanization

The Euler angle rates obtained by angular velocity:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \dot{\mathbf{\Psi}}_{INS,t}^{body} = \begin{bmatrix} 1 & sin(\phi)\,tan(\theta) & cos(\phi)\,tan(\theta) \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi)\,sec(\theta) & cos(\phi)\,sec(\theta) \end{bmatrix} \left(\boldsymbol{\omega}_t^{Gyro} - \mathbf{B}_{\omega,t-1}^{body}\right)$$

$$\boldsymbol{\delta\Psi}_{INS,t}^{body} = \dot{\mathbf{\Psi}}_{INS,t}^{body} \Delta t = \begin{bmatrix} \delta\phi \\ \delta\theta \\ \delta\psi \end{bmatrix}$$

Update the current Euler angles

$$\mathbf{\Psi}_{INS,t}^{body} = \mathbf{\Psi}_{KF,t-1}^{body} + \boldsymbol{\delta\Psi}_{INS,t}^{body}$$

$$\mathbf{\Psi}_{KF,t-1}^{local} = \mathbf{R}_{body}^{local} \mathbf{\Psi}_{KF,t-1}^{body}$$

Rotate the Euler angles from Body to Local

$$\mathbf{\Psi}_{INS,t}^{local} = \mathbf{R}_{body}^{local} \mathbf{\Psi}_{INS,t}^{body}$$

$$\begin{aligned} \mathbf{R}_{body}^{local} &= \mathbf{R}(X,\phi)\mathbf{R}(Y,\theta)\mathbf{R}(Z,\psi) \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi) & cos(\phi) \end{bmatrix} \begin{bmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & 1 & 0 \\ -sin(\theta) & 0 & cos(\theta) \end{bmatrix} \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

INS Mechanization

To remove the gravity from the acceleration

$$\mathbf{a}_{INS,t}^{local} = \mathbf{R}_{body}^{local}\left(\mathbf{a}_{INS,t}^{body} - \mathbf{B}_{a,t-1}^{body}\right) - \mathbf{g}$$

To obtain the change of aircraft in terms of position, velocity and orientation

$$\boldsymbol{\delta}\mathbf{X}_{INS,t}^{body} = \mathbf{V}_{INS,t-1}^{local}\Delta t + \frac{\boldsymbol{a}_{INS,t}^{local}\Delta t^2}{2}$$

$$\boldsymbol{\delta}\mathbf{V}_{INS,t}^{local} = \boldsymbol{a}_{INS,t}^{local}\Delta t$$

$$\boldsymbol{\delta\Psi}_{INS,t}^{local} = \boldsymbol{\Psi}_{KF,t-1}^{local} - \boldsymbol{\Psi}_{INS,t-1}^{local}$$

Kalman Filter——GNSS/INS (Closed Loop)

System States:

$$\mathbf{X}_t = \left(\mathbf{X}_t^{local}, \mathbf{V}_t^{local}, \mathbf{\Psi}_t^{local}, \mathbf{B}_{a,t}^{body}, \mathbf{B}_{\omega,t}^{body}\right)$$

$$\mathbf{X}_t^{local} = \left(x_t^{local}, y_t^{local}, z_t^{local}\right)$$

$$\mathbf{V}_t^{local} = \left(vx_t^{local}, vy_t^{local}, vz_t^{local}\right)$$

$$\mathbf{\Psi}_t^{local} = \left(\phi_{roll}, \theta_{pitch}, \psi_{yaw}\right)$$

$$\mathbf{B}_{a,t}^{body} = \left(b_{ax,t}^{body}, b_{ay,t}^{body}, b_{az,t}^{body}\right)$$

$$\mathbf{B}_{\omega,t}^{body} = \left(b_{\omega x,t}^{body}, b_{\omega y,t}^{body}, b_{\omega z,t}^{body}\right)$$

Measurement model:

$$\Delta\mathbf{Z}_t = \mathbf{Z}_t - \mathbf{H}\mathbf{X}_t^-$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_{3\times3} & \cdots & \mathbf{0} \\ \vdots & \mathbf{0} & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}$$

$$\mathbf{Z}_t = \mathbf{X}_{GNSS,t}^{local} = \begin{bmatrix} x_{GNSS,t}^{local} \\ y_{GNSS,t}^{local} \\ z_{GNSS,t}^{local} \end{bmatrix}$$

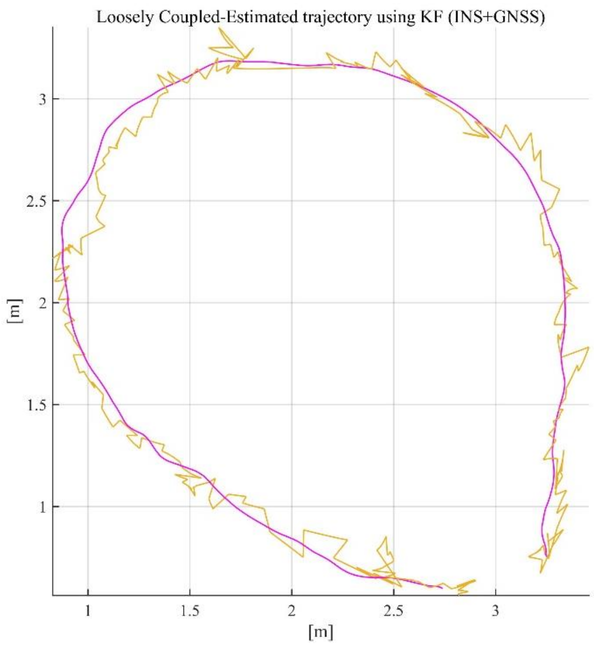
Propagation model:

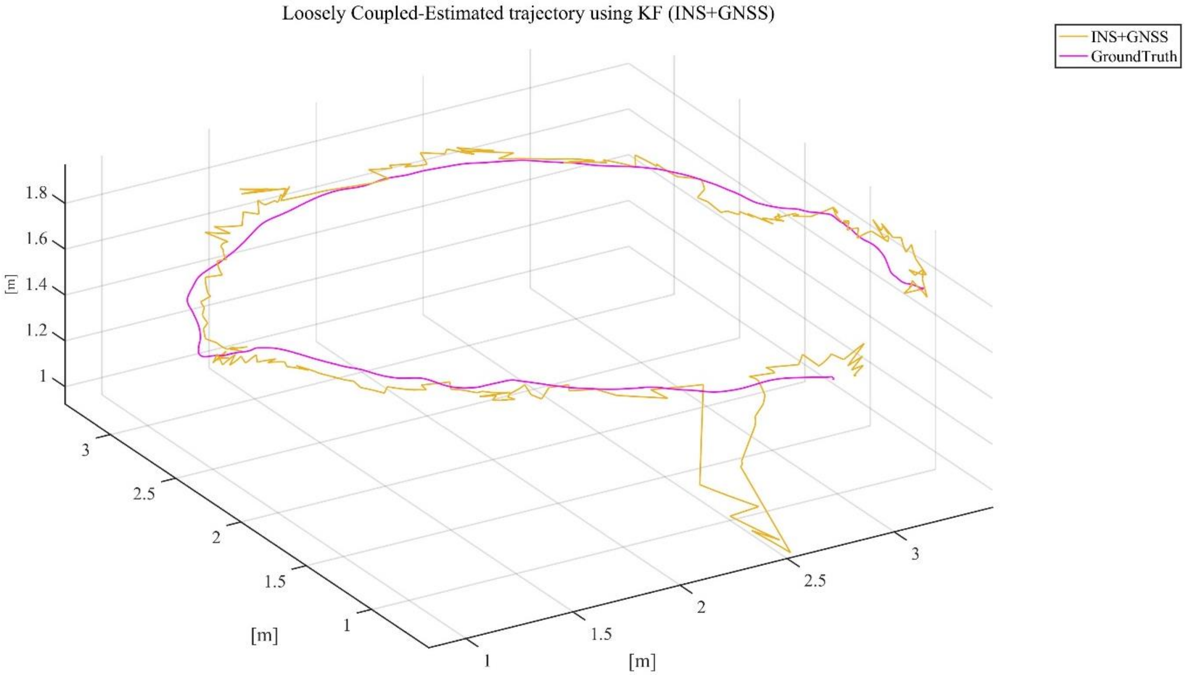
$$\mathbf{X}_t^- = \mathbf{F}\mathbf{X}_{t-1}^+ + \mathbf{B}\mathbf{U}_t$$

$$\mathbf{F} = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}$$

$$\mathbf{U}_t = \begin{bmatrix} \delta\mathbf{X}_{INS,t}^{local} \\ \delta\mathbf{V}_{INS,t}^{local} \\ \delta\mathbf{\Psi}_{INS,t}^{local} \\ W_{b_\omega} \\ W_{b_a} \end{bmatrix}$$

Closed Loop


Loosely Coupled-Estimated trajectory using KF (INS+GNSS)
3
2.5
2
1.5
1
[m]
1
1.5
2
2.5
3
[m]


Loosely Coupled-Estimated trajectory using KF (INS+GNSS)
INS+GNSS
GroundTruth
1.8
1.6
1.4
1.2
1
[m]
3
2.5
2
1.5
1
[m]
1
1.5
2
2.5
3
[m]

Non-Statistical Method of Estimation

Non-Statistical Method

- **Least Square**
  $$\hat{\mathbf{x}} = \underset{\mathbf{x}}{arg\,\min}\|h(\mathbf{x}) - \mathbf{z}\|^2$$
  - **Linear Least Square**
    $$\hat{\mathbf{x}} = h^{-1}(\mathbf{z}) \cong \mathbf{Hz} = \left(\mathbf{H}^{\mathrm{T}}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{z}$$
  - **Non-Linear Least Square**
- **Best Linear Unbiased Estimation (BLU)**

Robust Estimator

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{arg\,\min}\|h(\mathbf{x}) - \mathbf{z}\|^2$$

Residual function of estimator

$$\mathbf{r}(\mathbf{x}) = h(\mathbf{x}) - \mathbf{z}$$

Cost function based on residuals

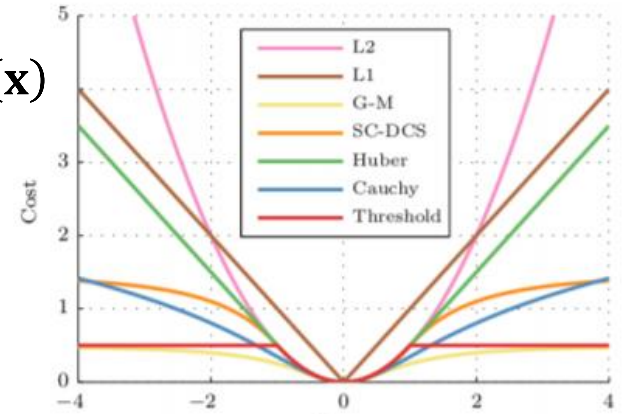
$$\mathrm{u}(\mathbf{x}) = \sqrt{\mathbf{r}(\mathbf{x})^{\mathrm{T}}\mathbf{r}(\mathbf{x})}$$

Error function of estimator

$$\mathrm{e}(\mathrm{u}(\mathbf{x}))$$

L_2	$\frac{1}{2}\|\mathrm{u}(\mathbf{x})\|^2$
L_1	$\lvert\mathrm{u}(\mathbf{x})\rvert$
G-M	$\frac{1}{2}\frac{\|\mathrm{u}(\mathbf{x})\|^2}{1+\|\mathrm{u}(\mathbf{x})\|^2}$
Cauchy	$\frac{1}{2}\ln(1+\|\mathrm{u}(\mathbf{x})\|^2)$
Huber	$\frac{1}{2}\|\mathrm{u}(\mathbf{x})\|^2$ for $\lvert\mathrm{u}(\mathbf{x})\rvert \le \delta$ $\delta(\lvert\mathrm{u}(\mathbf{x})\rvert - \frac{1}{2}\delta)$, otherwise

$\mathrm{u}(\mathbf{x})$



$\mathrm{e}(\mathrm{u}(\mathbf{x}))$

Statistical Methods of Estimation and Optimization

Statistical method

States set

$$\boldsymbol{\chi} = \{\mathbf{x}_o, \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k\}$$

$$\mathbf{x} = \begin{bmatrix} x_0 \\ \vdots \\ x_m \end{bmatrix}$$

Measurement set

$$\mathbf{Z} = \{\mathbf{z}_o, \mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_k\}$$

$$\mathbf{z} = \begin{bmatrix} z_0 \\ \vdots \\ z_n \end{bmatrix}$$

Frequentist → **Maximum Likelihood Estimation (MLE)**

$$P(\mathbf{z}|\mathbf{x})$$

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}\in\mathbf{X}} P_{(\mathbf{Z}|\chi)}(\mathbf{z}|\mathbf{x})$$

event-centric estimation, fully based on data. Should be used in Big Data application.

Bayesians → **Maximum a Posterior Estimation (MAP)**

$$P(\mathbf{x}|\mathbf{z}) = \frac{P(\mathbf{z}|\mathbf{x})P(\mathbf{x})}{P(\mathbf{z})}$$

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}\in\mathbf{X}} P_{(\mathbf{Z}|\chi)}(\mathbf{x}|\mathbf{z})$$

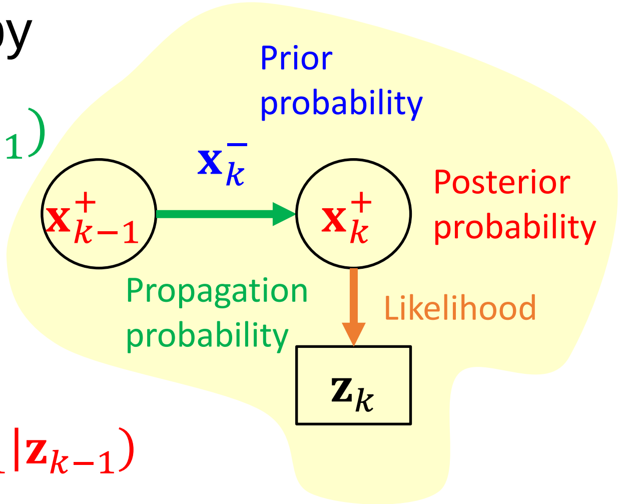
observer-centric estimation, required a knowledgeful observer (good prior-information)

Optimization, i.e., Kalman filter, factor graph optimization, etc.

From MAP Estimate to Kalman Filter

Maximum a posteriori (MAP) estimate is given by

$$P(\chi|\mathbf{Z},\mathbf{U}) = \prod_k P(\mathbf{z}_k|\mathbf{x}_k)\, P(\mathbf{x}_0) \prod_k P(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{u}_{k-1})$$



Assumption 1: First-order Markov

Recursive filter

Propagation: $P(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{z}_{k-1},\mathbf{u}_k) \propto P(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{u}_k)P(\mathbf{x}_{k-1}|\mathbf{z}_{k-1})$

Update: $P(\mathbf{x}_k|\mathbf{z}_k) \propto P(\mathbf{z}_k|\mathbf{x}_k)P(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{z}_{k-1},\mathbf{u}_k)$

Assumption 2: System Modelling & Gaussian Noise

Propagation: $(\mathbf{x}_{k|k-1}, \mathbf{P}_{k|k-1})$

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{f}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k)$$
$$\mathbf{P}_{k|k-1} = \mathbf{F}_k\mathbf{P}_{k-1|k-1}\mathbf{F}_k^{\mathrm{T}} + \mathbf{Q}_k$$

Update: $(\mathbf{x}_k, \mathbf{P}_k)$

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1})$$
$$\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k$$
$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^{\mathrm{T}}\mathbf{S}_k^{-1}$$
$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\tilde{\mathbf{y}}_k$$
$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_\mathbf{k})\mathbf{P}_{k|k-1}$$

Thrun, Sebastian. "Probabilistic algorithms in robotics." Ai Magazine 21.4 (2000): 93