



Università
di Genova

hotinemarussi2026
XI Hotine-Marussi Symposium

Optimizing GNSS Sequential Filtering Efficiency: A Variable-Dimension Update Strategy Based on Sparse Coupling Characteristics



Hailin Zhong^{1,2}, Pengyu Hou¹, Baocheng Zhang^{1*}

¹Innovation Academy for Precision Measurement Science and Technology, CAS

²College of Earth and Planetary Sciences, University of Chinese Academy of Sciences, China

Sep 8, 2026, Genova, Italy

A vertical banner image on the left side of the slide. The top half shows a close-up of a space shuttle launch with a large plume of fire and smoke. The bottom half shows a satellite in orbit above the Earth, with the planet's curvature and clouds visible.

CONTENTS ▶

1

Background

2

Methodology

2

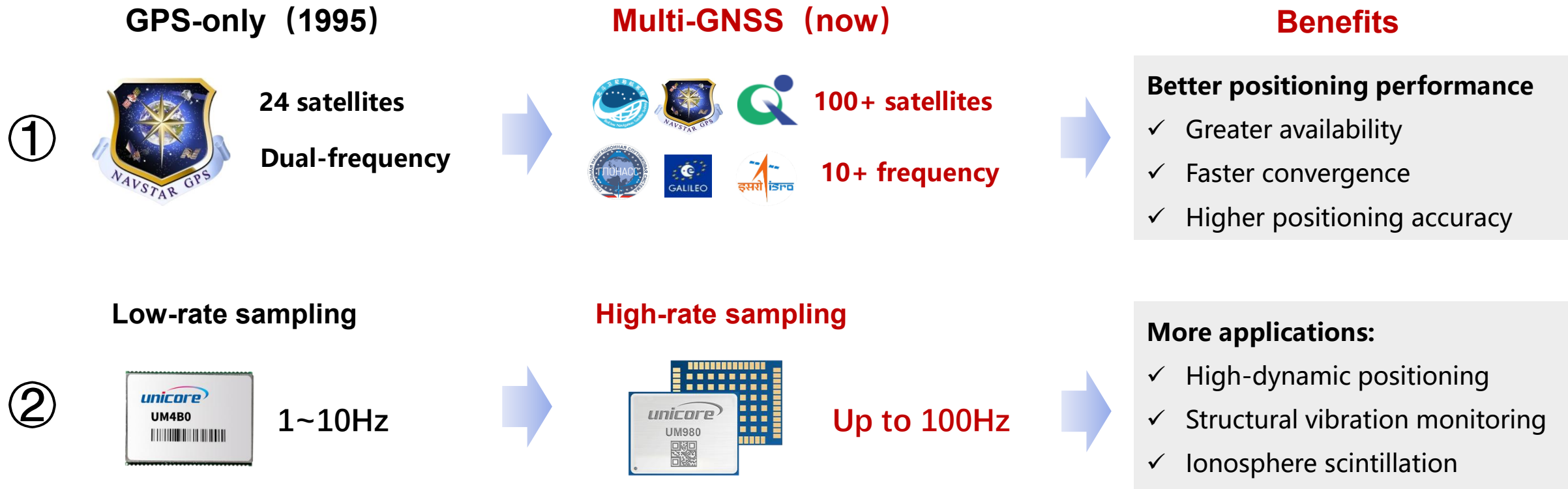
Results

3

Summary

1.1 Research Background

Challenges & Significance

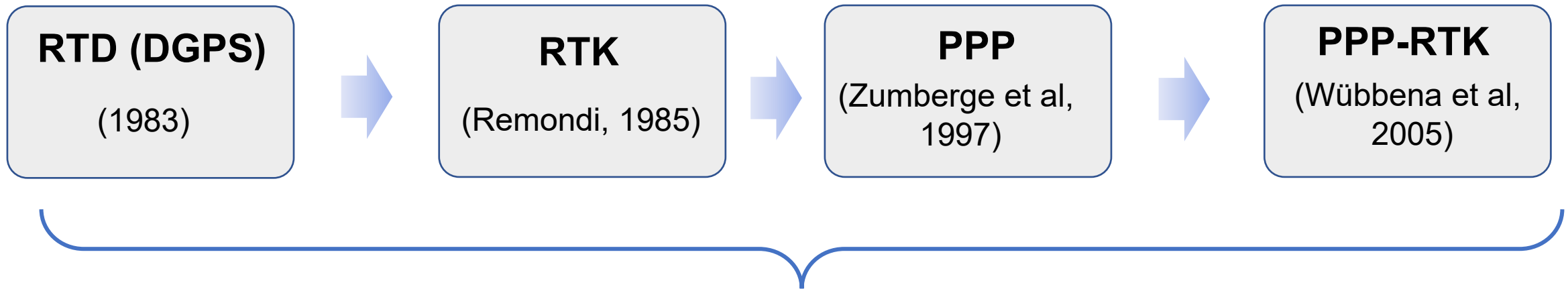


However, **mass-market GNSS chips/modules** have **limited computing resources** and therefore struggle to support high-rate positioning in multi-GNSS scenarios. Therefore, improving the data-processing efficiency is very important.

1.2 Review

The developments of GNSS models and Least Square estimation theory

GNSS models



Least Square (LS)

Minimum Variance Criterion

Linear model

$$y = Ax + v$$

$$\text{where } E(v) = 0, \text{Cov}(v) = Q$$

Objective function

$$\hat{x} = \arg \min_x J(x)$$

$$J(x) = (y - Ax)^T Q^{-1} (y - Ax)$$

LS solutions

$$\hat{x} = (A^T Q^{-1} A)^{-1} A^T Q^{-1} y$$

$$P_{\hat{x}} = (A^T Q^{-1} A)^{-1}$$

1.2 Review

From Batch Least Squares to Kalman Filtering

Batch processing of LS

$$y = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_t \end{pmatrix} \quad A = \begin{pmatrix} A_0 \\ A_1 \\ \vdots \\ A_t \end{pmatrix} \quad Q = \begin{pmatrix} Q_0 & & \\ & Q_1 & \\ & & \ddots \\ & & & Q_t \end{pmatrix}$$
$$\hat{x} = (A^T Q^{-1} A)^{-1} A^T Q^{-1} y$$

Processing all epochs simultaneously is computationally expensive.

Assumptions

1. Observation noise is uncorrelated in time;
2. Process noise is uncorrelated in time;
3. Process noise and observation noise are uncorrelated;
4. Initial state error is uncorrelated with the observation noise.

Standard Kalman filter (KF) (Kalman, 1960)

Step 1: Initialization

$$\hat{x}_{0|0} = (A_0^T Q_0^{-1} A_0)^{-1} A_0^T Q_0^{-1} y_0$$

$$P_{0|0} = (A_0^T Q_0^{-1} A_0)^{-1}$$

Step 2: Time update (TU)

$$\begin{cases} \hat{x}_{k|k-1} = F_{k|k-1} \hat{x}_{k-1} + B_{k|k-1} u_{k-1} \\ P_{k|k-1} = F_{k|k-1} P_{k-1} F_{k|k-1}^T + Q_{pro} \end{cases}$$
$$k = 1, 2, \dots, t$$

Step 3: Measurement update (MU)

$$\begin{cases} K_k = P_{k|k-1} A_k^T (A_k P_{k|k-1} A_k^T + Q_k)^{-1} \\ \hat{x}_k = \hat{x}_{k|k-1} + K_k (y_k - A_k \hat{x}_{k|k-1}) \\ P_k = P_{k|k-1} - K_k A_k P_{k|k-1} \end{cases}$$

1.2 Review

Sequential Kalman filter (SKF)

At epoch k , we assume that Q_k (VCM of observations) is a **diagonal matrix**

$$A_k = \begin{pmatrix} a_k^1 \\ a_k^2 \\ \vdots \\ a_k^m \end{pmatrix}_{m \times n} \quad y_k = \begin{pmatrix} y_k^1 \\ y_k^2 \\ \vdots \\ y_k^m \end{pmatrix}_{m \times 1} \quad Q_k = \begin{pmatrix} q_k^1 & & \\ & q_k^2 & \\ & & \ddots \\ & & & q_k^m \end{pmatrix}_{m \times m}$$

a_k^i — Row vector
 y_k^i — scalar
 q_k^i — scalar
($i = 1, 2, \dots, m$)

Standard MU

One-step matrix inversion

$$\begin{cases} K_k = P_{k|k-1} A_k^T (A_k P_{k|k-1} A_k^T + Q_k)^{-1} \in \mathbb{R}^{m \times m} \\ \hat{x}_k = \hat{x}_{k|k-1} + K_k (y_k - A_k \hat{x}_{k|k-1}) \\ P_k = P_{k|k-1} - K_k A_k P_{k|k-1} \end{cases}$$



Sequential processing of MU

m-step scalar division

$$\begin{cases} K_k^i = P_k^{i-1} (a_k^i)^T (a_k^i P_k^{i-1} (a_k^i)^T + q_k^i)^{-1} \in \mathbb{R} \\ \hat{x}_k^i = \hat{x}_k^{i-1} + K_k^i (y_k^i - a_k^i \hat{x}_k^{i-1}) \\ P_k^i = P_k^{i-1} - K_k^i a_k^i P_k^{i-1} \end{cases}$$

1.3 Discussion

Applications of SKF in GNSS

Undifferenced

$$Q_r = \begin{pmatrix} q_r^1 & & & \\ & q_r^2 & & \\ & & \ddots & \\ & & & q_r^m \end{pmatrix}$$

Inter-station differenced

Station 1

Station 2

$$Q_1 + Q_2 = \begin{pmatrix} q_{12}^1 & & & \\ & q_{12}^2 & & \\ & & \ddots & \\ & & & q_{12}^m \end{pmatrix}$$

Inter-satellite differenced

$$ZQ_rZ^T = \begin{pmatrix} q_r^1 + q_r^2 & q_r^1 & & q_r^1 \\ q_r^1 & q_r^1 + q_r^3 & & q_r^1 \\ & & \ddots & \\ q_r^1 & q_r^1 & & q_r^1 + q_r^m \end{pmatrix}$$

We usually can avoid the correlations introduced by inter-satellite difference or other linear combinations!

Whitening

Decorrelation

$$Q_k = \Delta\Delta^T \quad \text{Cholesky}$$

$$\bar{v}_k = \Delta^{-1}v_k$$

$$\Delta^{-1}Q_k\Delta^{-T} = E$$

Equivalent transformation

$$\underbrace{\Delta^{-1}y_k}_{\bar{y}_k} = \underbrace{(\Delta^{-1}A_k)}_{\bar{A}_k}x_k + \underbrace{\Delta^{-1}v_k}_{\bar{v}_k}$$

$$\bar{y}_k = \bar{A}_kx_k + \bar{v}_k \quad \text{Cov}(\bar{v}_k) = E$$

Identity matrix

What should we do if Q_k is not a diagonal matrix?

A vertical banner on the left side of the slide. The top half shows a close-up of a space shuttle launch with a large plume of fire and smoke. The bottom half shows a satellite in orbit above the Earth, with a bright light source creating a lens flare effect.

CONTENTS ▶

1

Background

2

Methodology

3

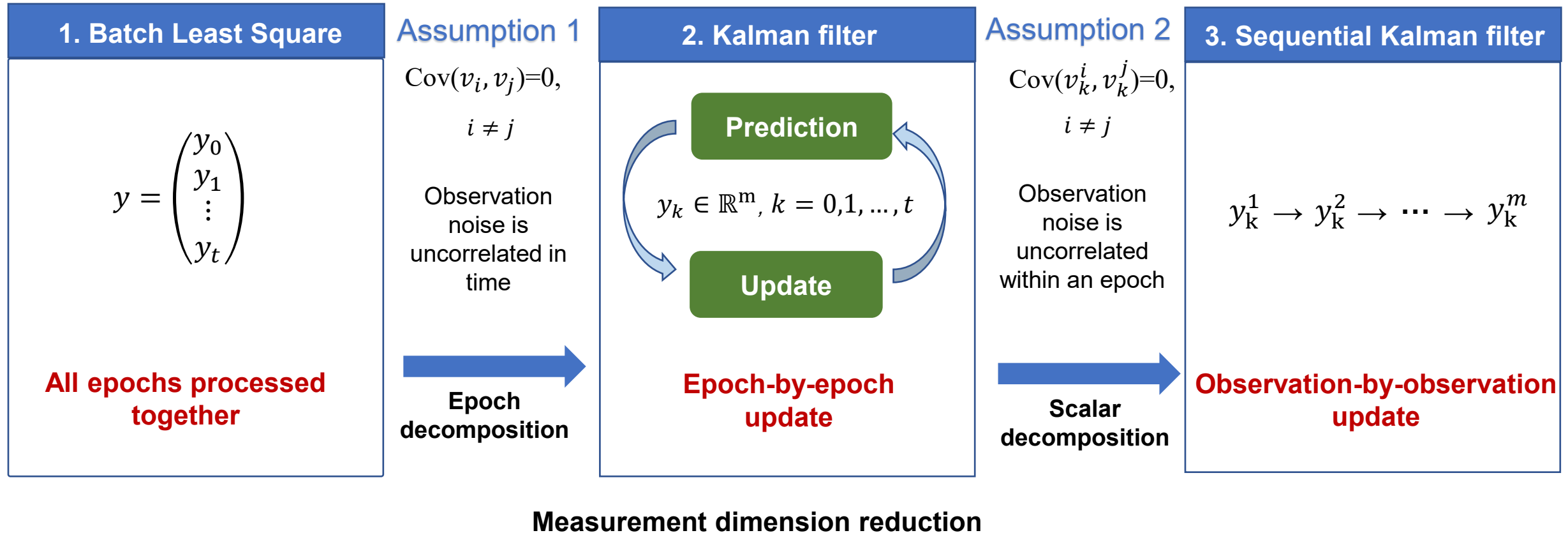
Results

4

Summary

2 Methodology

Review of LS, KF and SKF



Can we further improve the computational efficiency based on the SKF?

2 Methodology

Sparse Coupling characteristic in GNSS Observation Models

Two Classes of Parameters



Common Parameters

- ✓ Position
- ✓ Receiver clock
- ✓ ZTD

Coupled with all observations



Non-common Parameters

- ✓ Code/phase biases
- ✓ Ionosphere
- ✓ Ambiguities

Type

Satellite

Frequency

Coupled only with specific observations

Design matrix segmentation and reduction

Echelon Design Matrix

$$A_k \in \mathbb{R}^{m \times n}$$

split into Row vectors in different dimension

Code

Phase



$$a_k^1 \in \mathbb{R}^4$$

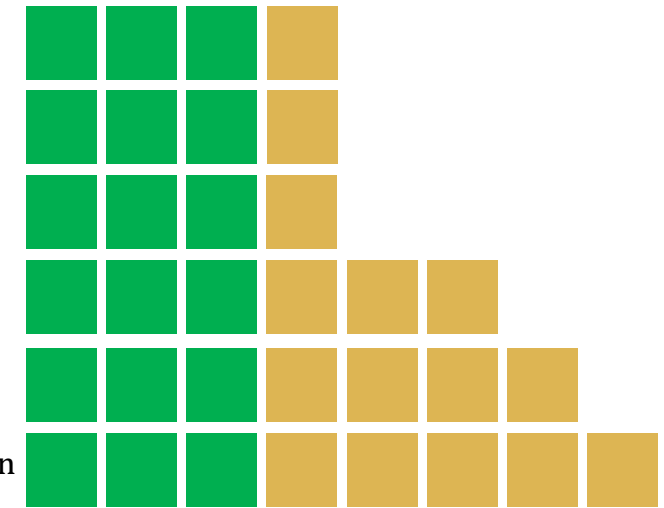
$$a_k^2 \in \mathbb{R}^4$$

$$a_k^3 \in \mathbb{R}^4$$

$$a_k^4 \in \mathbb{R}^6$$

⋮

$$a_k^m \in \mathbb{R}^n$$



Common parameter



Non-common parameter



Zero element

Can we conduct variable-dimension (VD) update based on such a echelon structure to avoid redundant computation and further improve the efficiency?

2 Methodology

A Single Example for Variable-Dimension (VD) State Vector Update

1. Assumption

$$A_k \begin{cases} a_1 = \begin{bmatrix} a_{11} & 0 \end{bmatrix} & a_{11} \in \mathbb{R}^{1 \times n_c}, \\ a_2 = \begin{bmatrix} a_{21} & a_{22} \end{bmatrix} & a_{22} \in \mathbb{R}^{n_t \times 1} \end{cases} \quad y_k = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

Echelon structure

$$\begin{cases} x_{k|k-1} = \begin{bmatrix} x_c^0 & x_t^0 \end{bmatrix}^T & x_c^0 \in \mathbb{R}^{n_c \times 1}, \quad x_t^0 \in \mathbb{R}^{n_t \times 1} \\ P_{k|k-1} = \begin{bmatrix} P_c^0 & 0 \\ 0 & P_t^0 \end{bmatrix} & P_c \in \mathbb{R}^{n_c \times n_c}, \quad P_t \in \mathbb{R}^{n_t \times n_t} \end{cases}$$

Diagonal or block-diagonal

Common parameters

Non-common

2. Partial gain

When process the first observation

$$K^1 = \frac{P_{k|k-1} a_1^T}{a_1 P_{k|k-1} a_1^T + q_1}$$

where $P_{k|k-1} a_1^T = \begin{bmatrix} P_c^0 a_{11}^T \\ 0 \end{bmatrix}$

$$K^1 = \begin{bmatrix} K_{c,1} \\ 0 \end{bmatrix} \quad K_c^1 = \frac{P_c^0 a_{11}^T}{a_{11} P_c^0 a_{11}^T + q_1}$$

Gain for the non-common parameters is zero

3. State update

The gain from the y_1

$$\begin{aligned} \hat{x}_k^1 &= x_{k|k-1} + K^1 (y_1 - a_1 x_{k|k-1}) \\ &= \begin{bmatrix} x_c^1 \\ x_t^1 \end{bmatrix} = \begin{bmatrix} x_c + K_c^1 (y_1 - a_{11} x_c^0) \\ x_t^0 \end{bmatrix} \end{aligned}$$

No change

Only the parameter x_c related to y_1 are changed. The x_t **keep the same** as the prediction

Conclusion

When $P_{k|k-1}$ is **diagonal**, the Kalman gain of parameters unrelated to the current observation is zero.

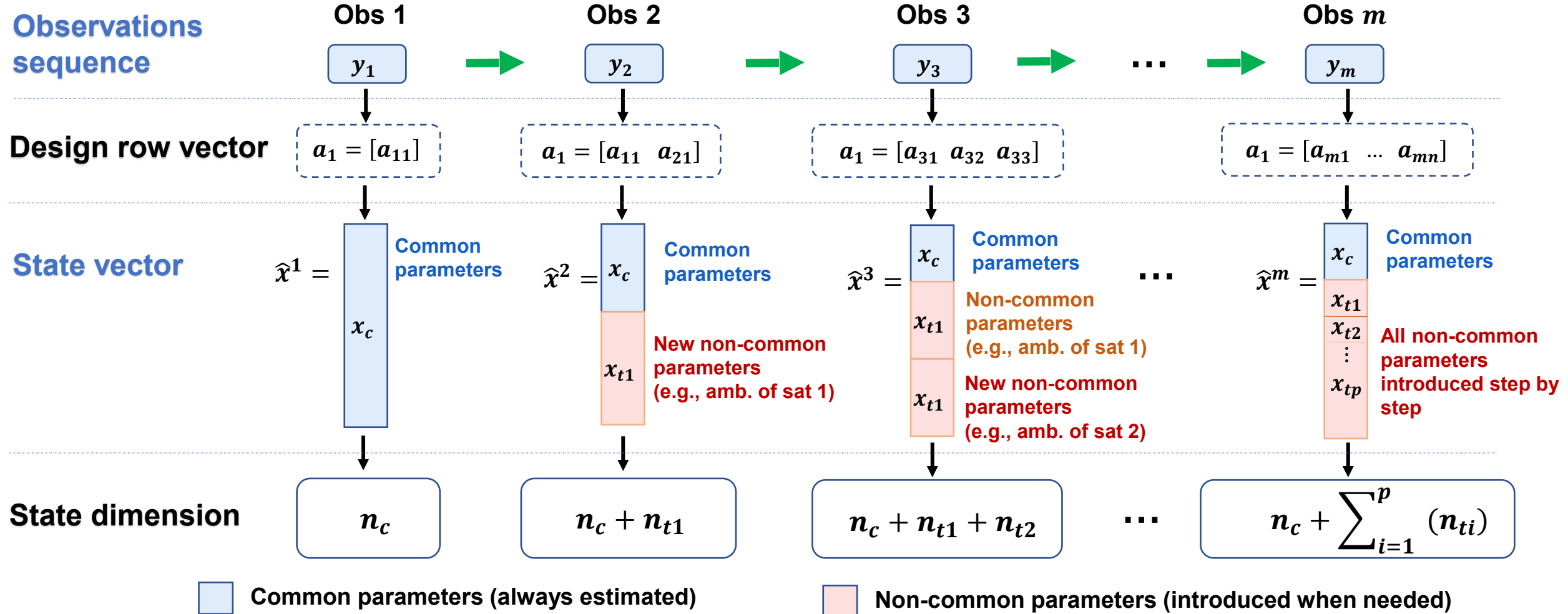
⇒ These parameters are not updated and don't need to be involved in the computation.

⇒ **Full-dimension update could be replaced by variable-dimension update, avoiding redundant computations.**

2 Methodology

Variable-Dimension (VD) State Vector Update

In VD update processing, only the parameters related to the current observation are estimated. **New parameters** are introduced only when they appear in the design matrix



2 Methodology

VD Update in Single-Epoch Processing

The VD update requires the predicted **VCM** ($P_{k|k-1}$) to be **diagonal**, which can be achieved in single-epoch processing.

No prior information

$$P_{k|k-1} = \begin{pmatrix} +\infty & & & \\ & +\infty & & \\ & & \ddots & \\ & & & +\infty \end{pmatrix}$$

$$K_k = P_{k|k-1} A_k^T (A_k P_{k|k-1} A_k^T + Q_k)^{-1}$$

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k (y_k - A_k \hat{x}_{k|k-1}) = \hat{x}_k = (A_k^T Q_k^{-1} A_k)^{-1} A_k^T Q_k^{-1} y_k$$

KF

LS

In the limiting case, the Kalman filter solution is equivalent to the LS solution.

However, an infinite value cannot be represented in a computer. To meet the requirements of Kalman filtering, we can instead assign an rough initial value to each parameter with a very large variance.

Table 2 The determination of initial values and prior variance

Parameter	Initial value	Prior variance
$\Delta x_r, dt_r$	Provided by Standard Point Positioning (SPP) or previous epoch	$10^6 m^2$
τ_r	Saastamoinen (Saastamoinen 1972)	$10^6 m^2$
dt^s	Provided by broadcast ephemeris	$10^6 m^2$
$\delta_{r,j}, \delta_{,j}^s, b_{r,j}, b_{,j}^s$	Set to 0	$10^6 m^2$
l_r^s	Klobuchar model	$10^6 m^2$
$z_{r,j}^s$	$(p_{r,j}^s - \varphi_{r,j}^s) / \lambda_j$	$10^6 cycles$

Assign a very large variance

2 Methodology

VD Update in Multi-Epoch Processing

However, in multi-epoch processing, predicted $P_{k|k-1}$ is usually **not a diagonal matrix** after time update.

1. TU (prediction)

$$\begin{cases} \hat{x}_{k|k-1} = F_{k|k-1} \hat{x}_{k-1} + B_{k|k-1} u_{k-1} + w_{k|k-1} \\ P_{k|k-1} = F_{k|k-1} P_{k-1} F_{k|k-1}^T + Q_{pro} \end{cases}$$

Predicted parameters are correlated

2. Decorrelation

$$\text{LDLT decomposition: } P_{k|k-1} = LDL^T$$

$$\begin{aligned} \text{New parameter space} \quad D &= L^{-1} P_{k|k-1} (L^{-1})^T \\ \hat{z} &= L^{-1} \hat{x}_{k|k-1} \end{aligned}$$

Transformed parameters are uncorrelated

3. Transform the observation function

$$y_k = A_k x_k + v_k$$

$$y_k = \underbrace{(A_k L)}_H \underbrace{(L^{-1} x_k)}_z + v_k$$

$$y_k = H z + v_k$$

4. Sequential MU

$$\begin{cases} K_k^i = D^{i-1} a_k^i (a_k^i D^{i-1} a_k^i + q_k^i)^{-1} \\ \hat{z}^i = \hat{z}^{i-1} + K_k^i (y_k - a_k^i \hat{z}^{i-1}) \\ D^i = D^{i-1} - K_k^i a_k^{i-1} D^{i-1} \end{cases}$$

Variable-Dimension update

5. Back-transform

$$\begin{aligned} \hat{x}_k &= L \hat{z} \\ P_k &= L \bar{D} L^T \end{aligned}$$

Recover state vector and covariance

2 Methodology

Discussion

1. What does the VD update change?

The VD update does not change the measurement update procedure. It only keeps the state vector and its covariance matrix (VCM) at the **minimum required dimension** during each measurement update, instead of using the full-dimension update. Therefore, **redundant computations can be avoided** and the computational efficiency can be improved.

2. Can the VD strategy be extended to other filters?

The VD update is a general strategy for sequential processing. Therefore, it can also be integrated into other advanced sequential Kalman filter variants, such as the **Square root covariance filter (SRCF)** and ***U-D* factorization filter (UDF)**.

In this study, we integrate the VD strategy into the UDF and develop the **VD-UDF** method. The UDF is selected because of its good numerical stability and high computational efficiency. In the following experiments, VD-UDF is compared with the **standard UDF**, (KF in information form) **KF-I**, and (Square Root Information Filter) **SRIF** to evaluate its computational efficiency.



CONTENTS ►

1

Background

2

Methodology

3

Results

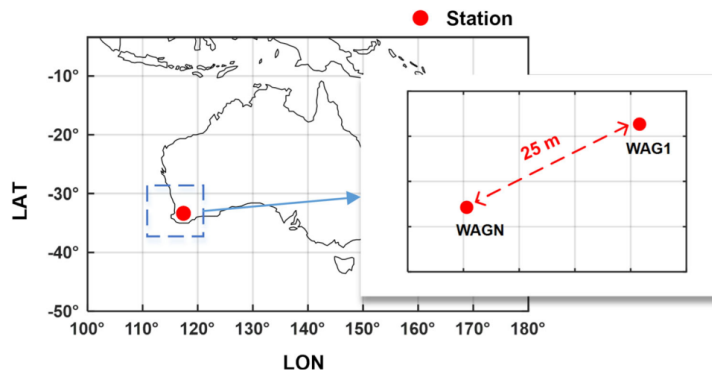
4

Summary

3.1 Experiment Setup

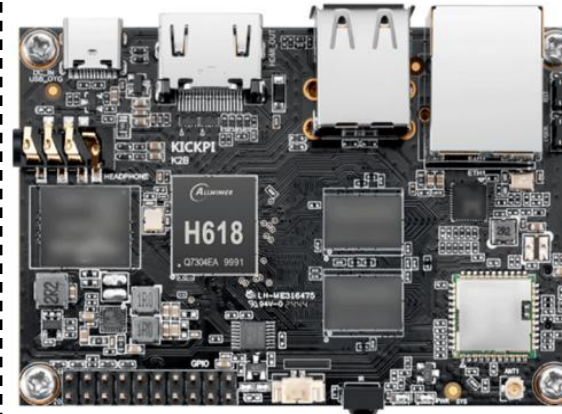
Setup

Data



Two stations from Australian CORS

Low-cost hardware platform

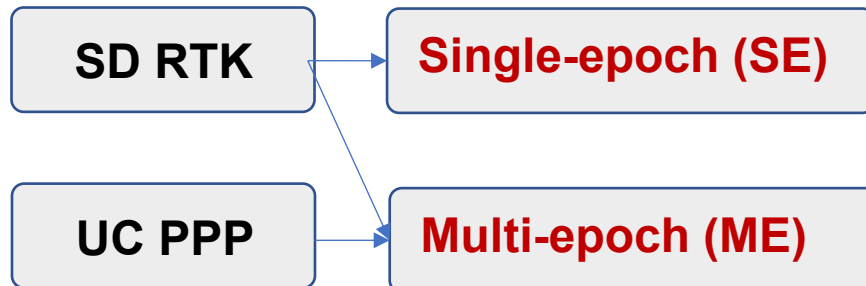


Type	Specification
Main Controller	Allwinner H618
CPU	ARM Cortex-A53 (64-bit) (up to 1.5 GHz)
RAM	2 GB
ROM	16 GB eMMC
OS	Ubuntu 22.04 LTS

Manufacture:
Allwinner

Price:
30~40 US dollar

GNSS model and processing modes



Efficiency Comparison

Traditional methods: standard UDF, KF-I, and SRIF

Widely used in industrial GNSS application

Modified method: VD-UDF

3.2 Comparison of Computational Efficiency

Theoretical Analysis

If we know the **number of observations (m)**, the **number of estimable parameters (n)**, and the structure of the design matrix, we can count the required operations. Then, we can theoretically evaluate their **expected time efficiency**.

Method	Mode	Addition	Multiplication	Division	Square root
KF-I	Single-epoch	$mn^2 + \frac{n^3}{2} + mn - \frac{5n}{2}$	$mn^2 + \frac{n^3}{2} + 2mn + n^2 + \frac{n}{2}$	n^2	n
	Multi-epoch	$2mn^2 + n^3 + mn - n$	$2mn^2 + n^3 + 3mn + n$	$2n^2$	$2n$
SRIF	Single-epoch	$mn^2 - \frac{5n}{6} + \frac{n^2}{2}$	$mn^2 + mn + n^2 + m + \frac{2n}{3}$	$\frac{n^2}{2} + \frac{3n}{2}$	n
	Multi-epoch	$mn^2 + \frac{4n^3}{3} + n^2 - \frac{5n}{3}$	$mn^2 + \frac{4n^3}{3} + mn + \frac{3n^2}{2} + m + \frac{5n}{6}$	$\frac{3n^2}{2} + mn + \frac{3n}{2}$	$2n$
UDF	Single-epoch	$\frac{3n^2m}{2} + \frac{n^3}{6} + \frac{7nm}{2} - \frac{n}{6}$	$\frac{3n^2m}{2} + \frac{n^3}{6} + \frac{5nm}{2} + n^2 + \frac{5n}{6}$	$3nm$	0
	Multi-epoch	$\frac{3n^2m}{2} + \frac{n^3}{3} + \frac{7nm}{2} + \frac{n^2}{2} + \frac{n}{6}$	$\frac{3n^2m}{2} + \frac{n^3}{3} + \frac{5nm}{2} + 2n^2 - \frac{n}{3}$	$n + 3nm$	0
VD-UDF	Single-epoch	$\sum_{i=1}^m \frac{3n_i^2 + 7n_i}{2} + \frac{n^3}{6} + 2mn - \frac{n}{6}$	$\sum_{i=1}^m \frac{3n_i^2 + 5n_i}{2} + \frac{n^3}{6} + n^2 + 2mn + \frac{5n}{6}$	$3nm$	0
	Multi-epoch	$\sum_{i=1}^m \frac{3n_i^2 + 7n_i}{2} + \frac{mn^2}{2} + \frac{n^3}{2} + \frac{3n^2}{2} + \frac{3mn}{2} - n$	$\sum_{i=1}^m \frac{3n_i^2 + 5n_i}{2} + \frac{mn^2}{2} + \frac{n^3}{2} + \frac{5mn}{2} + \frac{7n^2}{2}$	$2n + 3nm$	0

3.2 Comparison of Computational Efficiency

Theoretical Analysis

The improvement in time-efficiency of VD-UDF compared to UDF/KF-I/SRIF:

$$Pt = \frac{T_1 - T_2}{T_1} \times 100\%$$

Time consumption of VD-UDF

Time consumption of UDF/KF-I/SRIF

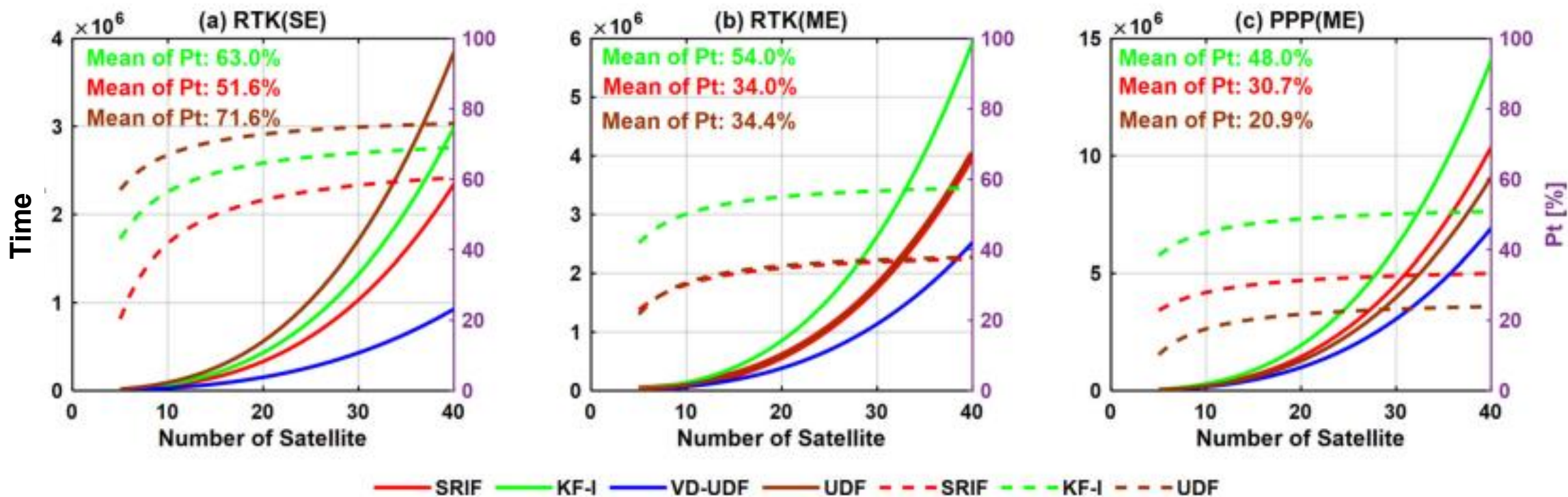


Fig. 7 Theoretical time cost and time-efficiency improvement of VD-UDF relative to KF-I and SRIF under single-epoch (SE) and multi-epoch (ME) modes. The solid lines represent time cost, while brown dashed lines represent the improvement (Pt)

3.2 Comparison of Computational Efficiency

Theoretical Analysis

The improvement in time-efficiency of VD-UDF compared to UDF/KF-I/SRIF:

	RTK(SE)		RTK(ME)		PPP(ME)
UDF	71.60%	>	34.40%	>	20.90%
KF-I	63.00%	>	54.00%	>	48.00%
SRIF	51.60%	>	34.40%	>	20.90%

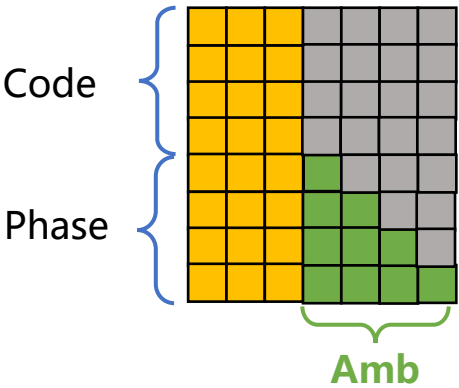
Phenomenon 1: For RTK processing, VD-UDF achieves a greater efficiency gain in single-epoch (SE) than in multi-epoch (ME).

Reason: Multi-epoch processing requires additional operations, including decorrelation of predicted covariance matrix $P_{k|k-1}$ and the transformation of state vector $\hat{x}_{k|k-1}$, whereas these operations are not required in single-epoch processing.

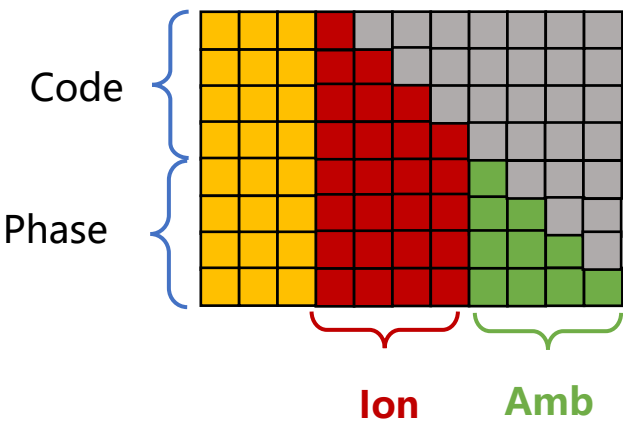
Phenomenon 2: For multi-epoch processing, VD-UDF achieves a greater efficiency gain in RTK than in PPP.

Reason:

Structure of SD-RTK design matrix



Structure of UC-PPP design matrix



SD RTK don't estimate ionosphere parameters, whereas UC PPP does. Consequently, the PPP design matrix is denser than the RTK design matrix.

3.2 Comparison of Computational Efficiency

Evaluation Based on Real GNSS Data

In this experiment, we processed **real GNSS data** on a **low-cost hardware platform** using four different methods.

GNSS system/frequency combination

System/Frequency	Average number of satellites
E/(E1+E5a)	7.3
G/(L1+L2)	8.9
C/(B1I+B3I)	12.4
GE/(L1+L2)+ (E1+E5a)	16.1
EC/ (E1+E5a)+ (B1I+B3I)	19.7
GC/(L1+L2)+ (B1I+B3I)	21.3
GEC/(L1+L2)+ (E1+E5a)+ (B1I+B3I)	28.5

Time consumption of VD-UDF, UDF, KF-I and SRIF

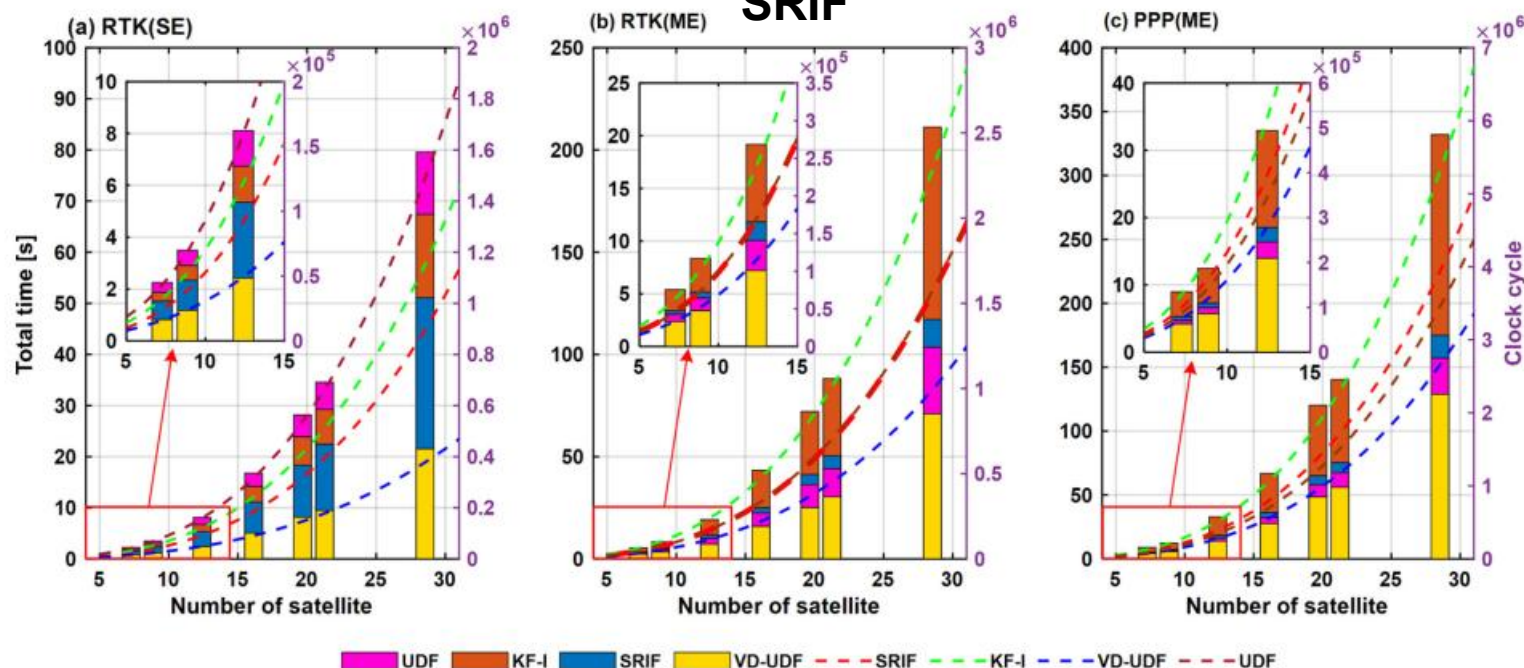


Fig. 10 The comparison of computation time for VD-UDF, KF-I, SRIF, and UDF methods based on real GNSS data. The bar graph represents the processing time for the real data, measured in seconds (s), while the

curve indicates the theoretical time cost, expressed in clock cycles. The corresponding system combinations for each satellite number refer to Table 6

3.2 Comparison of Computational Efficiency

Evaluation Based on Real GNSS Data

The efficiency improvement of VD-UDF compared with UDF, KF-I and SRIF

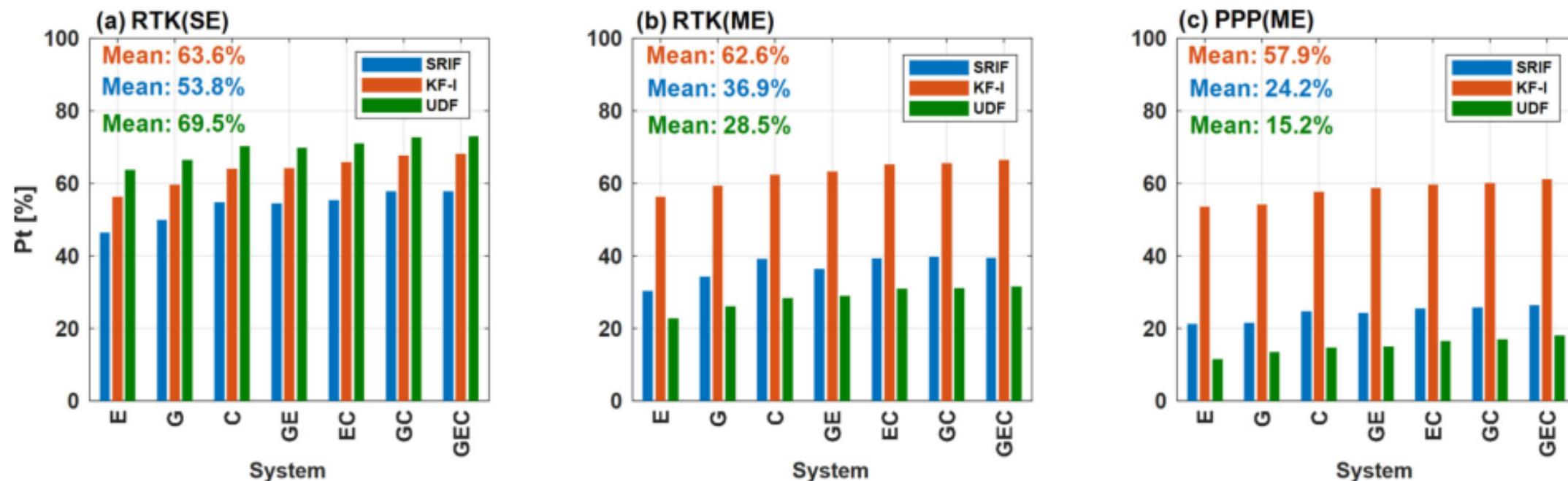


Fig. 11 The time-efficiency improvement of the VD-UDF method for different system combinations under various scenarios

The efficiency improvement with real data is also consistent with our expectations.



CONTENTS ▶

1

Background

2

Methodology

3

Results

4

Summary

4 Summary

Summary

- To summarize, the proposed VD-UDF significantly improves GNSS data-processing efficiency while maintaining the same numerical results as traditional methods.
- Therefore, this method has great potential for real-time, high-rate sampling, and multi-GNSS data processing, especially for low-cost devices with limited computing resources.

Reference

Zhong, H., Hou, P. & Zhang, B. A variable-dimension update of sequential Kalman filter for enhancing GNSS data processing efficiency. *J Geod* 100, 57 (2026).
<https://doi.org/10.1007/s00190-026-02078-1>



中国科学院精密测量科学与技术创新研究院

Innovation Academy for Precision Measurement Science and Technology, CAS

Thank you!

