Utilizing Batch Processing for GNSS Signal Tracking

Andrey Soloviev

Avionics Engineering Center, Ohio University

Presented to:

ION Alberta Section,

Calgary, Canada

February 27, 2007

Outline

- **Motivation:**
  - ***Clean up the record*: resolve confusion between different definitions used for processing of digitized GNSS signals.**
  - ***Deliver new material*: explore new applications utilizing batch processing techniques.**
- **Introduction to sequential and batch processing (filtering example).**
- **Generalized architectures of sequential and batch processing of GNSS signals.**
- **Complementary features of batch and sequential processing.**
- **Receiver design implementations that utilize a combination of sequential and batch processing techniques.**
- **Implementation examples:**
  - **Deep GPS/IMU integration (urban tests and flight tests).**

Motivation

- *Existing terminology* **for processing of digitized GNSS signals**

Block processing

Software receiver

FLL-aided PLL

Sequential processing

Batch processing

Massive parallel correlator banks

Deep integration

No loop

Phase lock loop

High-sensitivity receivers

Software-defined radio

Parallel FFT-based frequency search

DLL

Deep GPS/INS integration

ed receiver design

Parallel FFT-based code search

Ultra-tight integration

e processing

Sequential processing *recognized*

Batch processing *overlooked*

Frequency lock loop

Joint time-frequency tracking

Space time adaptive processing

- **Usage of** ***batch processing*** **techniques to** ***overcome limitations*** **of** ***sequential processing*** **for** ***practical application areas*****, e.g.:**

CONFUSING…

- Indoor navi
- Tracking un
- Navigation i

Framework for GNSS receiver design

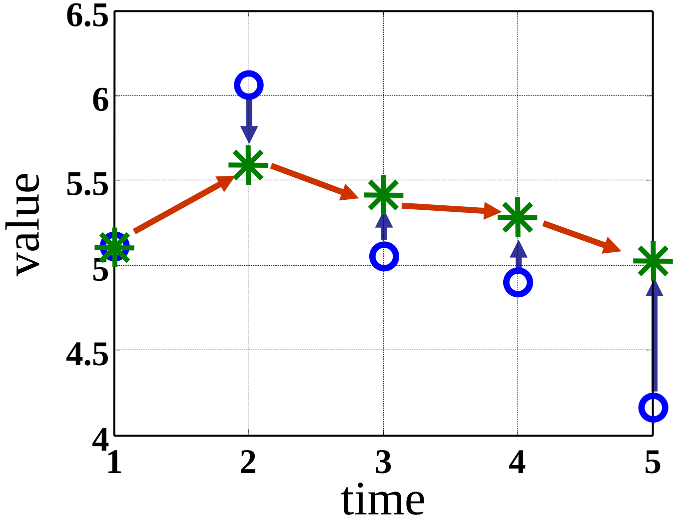
Case study: *Deep GPS/INS integration*

Introduction to Sequential and Batch Processing

❑ Filtering example: smoothing of 5 measurements

- ○ measurements (constant value plus noise) - $\tilde{x}$
- ✱ filter estimates - $\hat{x}$

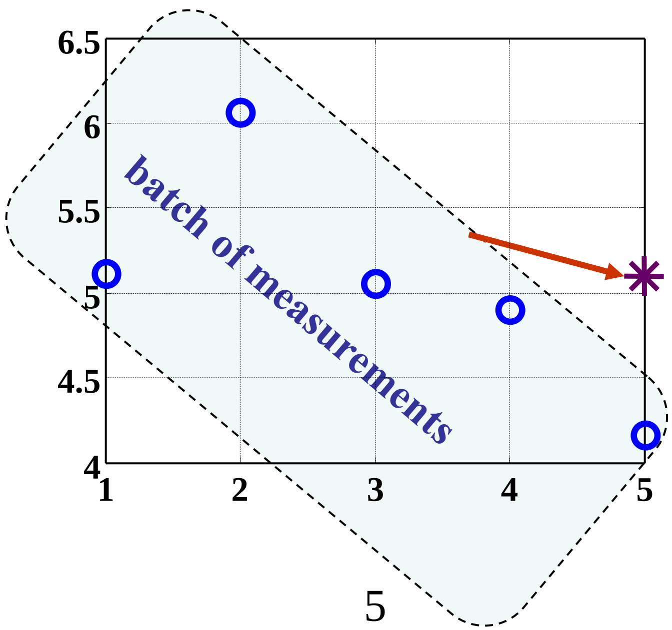
Sequential approach



$$\hat{x}_n = (1-\alpha_n)\cdot\hat{x}_{n-1} + \alpha_n \cdot \tilde{x}_{n-1}$$

• Perform recursive computations: estimates are updated with every new measurement.

Batch approach



$$\hat{x}_5 = \sum_{k=1}^{5} \alpha_k \cdot \tilde{x}_k$$

• Wait till all measurements have arrived and then do all the computations at once.

Introduction to Sequential and Batch Processing (cont.)

❑ Sequential vs. batch: first glance

Computational aspects

sequential

$$\hat{x}_n = (1-\alpha_n)\cdot\hat{x}_{n-1} + \alpha_n \cdot \tilde{x}_{n-1}$$

vs.

$$\hat{x}_n = \sum_{k=1}^{n} \alpha_k \cdot \tilde{x}_k$$

batch

- **3 elements in memory**
- **4 operations per update (20 operations total)**

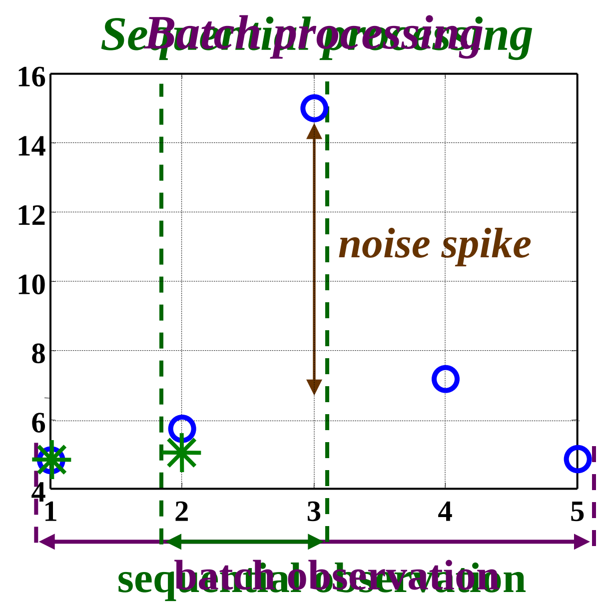
- **10 elements in memory**
- **10 operations**

➢ **Sequential processing requires less memory.**

➢ **Sequential processing spreads computations over time – fewer operations are therefore required to compute an estimate.**

Signal observability

Example: measurement noise spike



Batch processing delivers better observability.

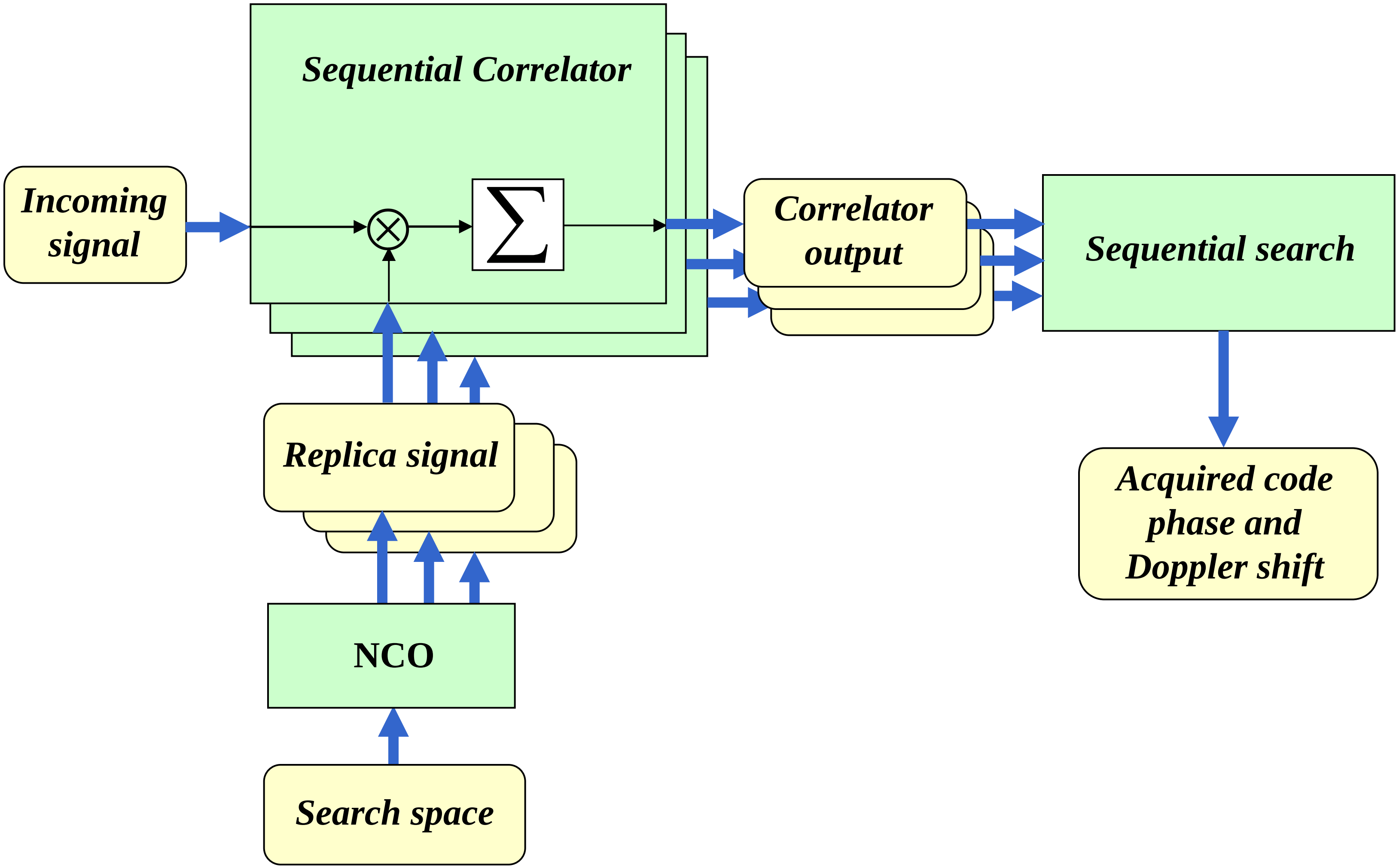
Next Step

- **Next step is to consider applications of sequential and batch approaches for processing of GNSS signals.**

- **Consideration provided is *independent of the receiver implementation platform* (ASIC, software receiver, software radio, DSP, FPGA)**

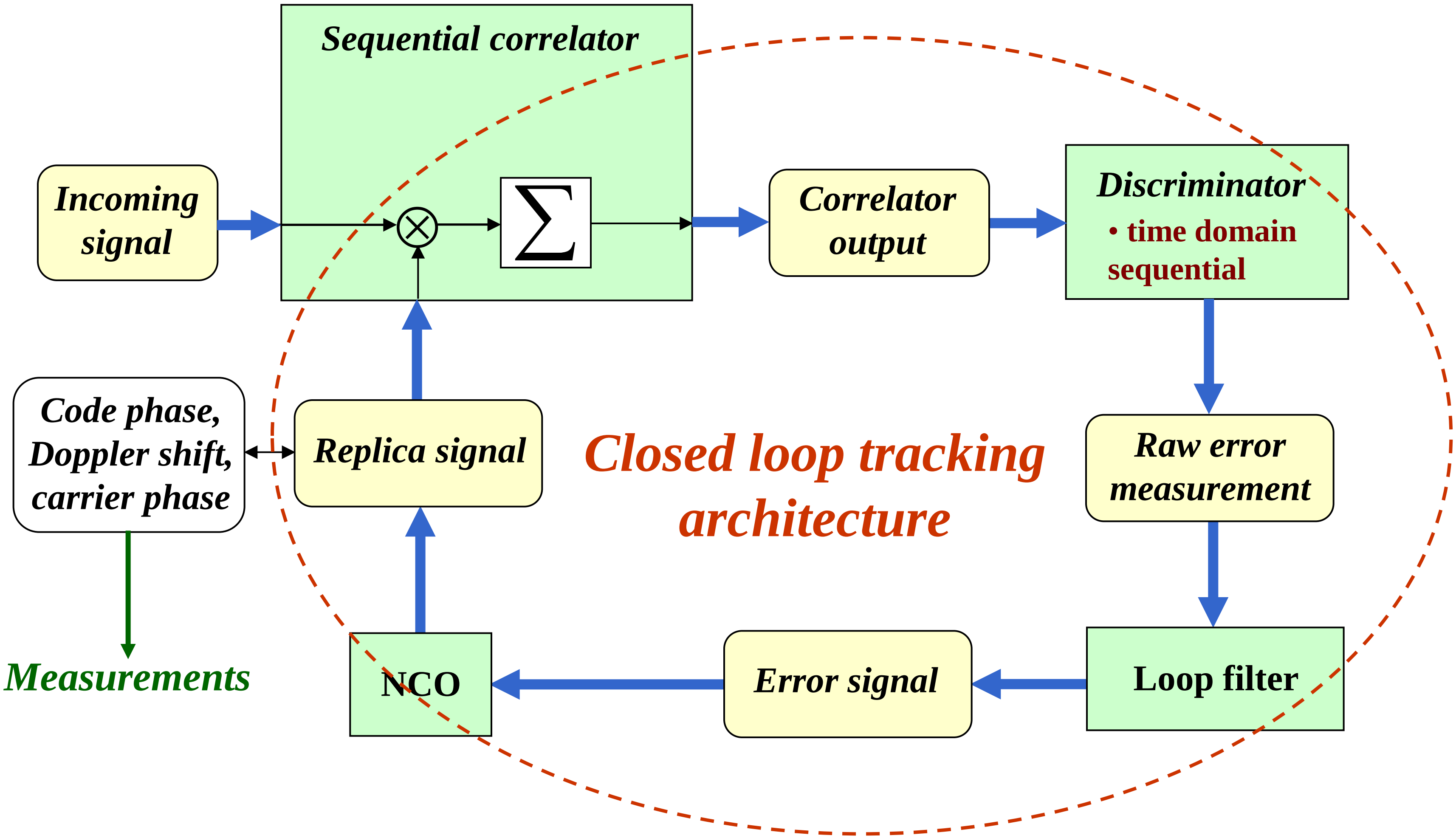
Sequential Acquisition of GNSS Signals

Generalized Architecture



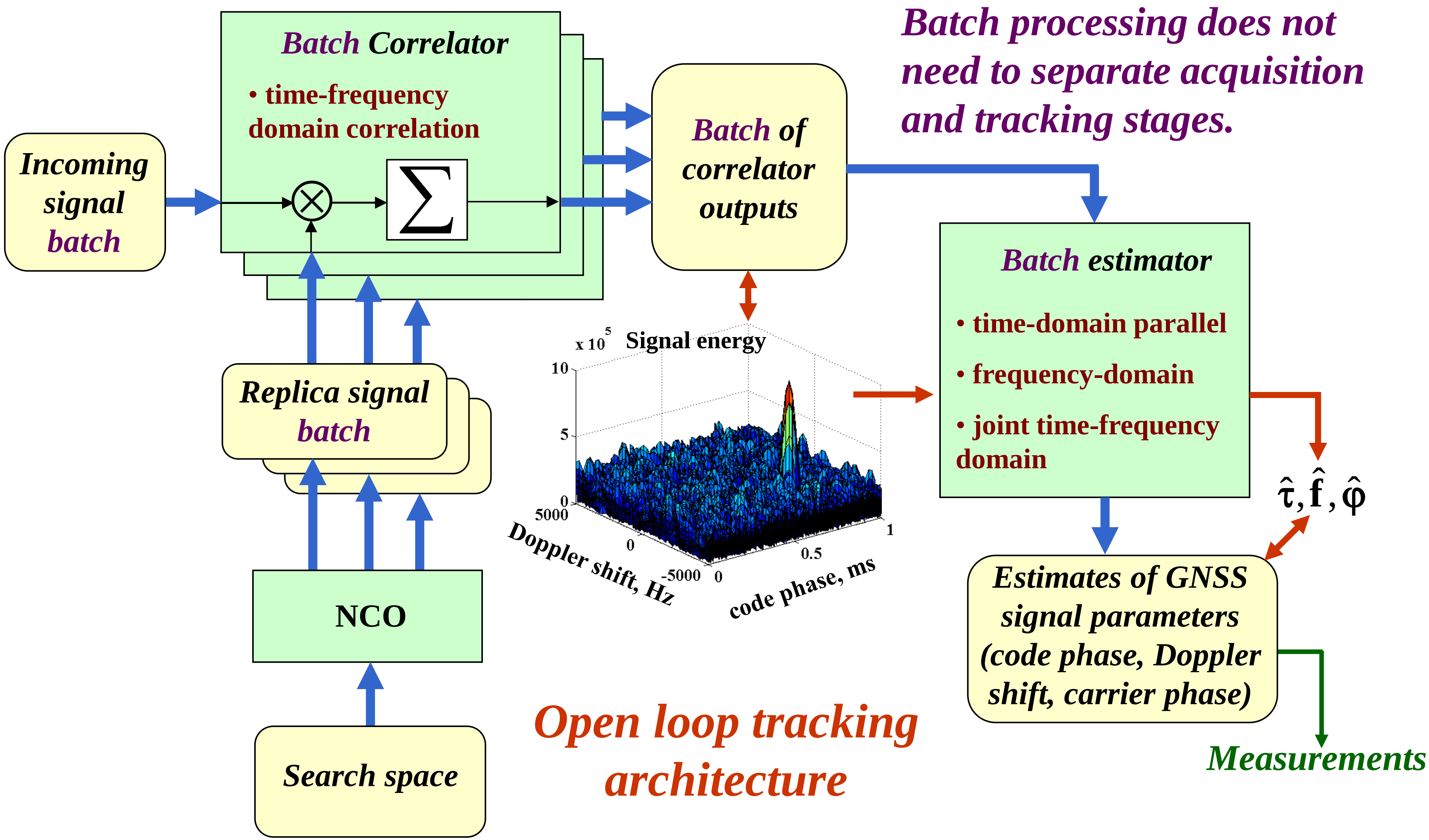
Sequential Tracking of GNSS Signals

Generalized Architecture



Batch Processing of GNSS Signals

Generalized Architecture



Measurement Exploitation for the Batch Processing Approach

- **Full search: no measurement exploitation.**

- **Local search: signal parameter *estimates* from the *previous measurement batch* can be used to *reduce the computational load* by *narrowing the replica search space* (*local search instead of full search*).**

- **Either way: signal parameter *estimation* for the *current batch* is *independent* from *previous batches*.**

Measurement Exploitation for the Batch Processing Approach (cont.)

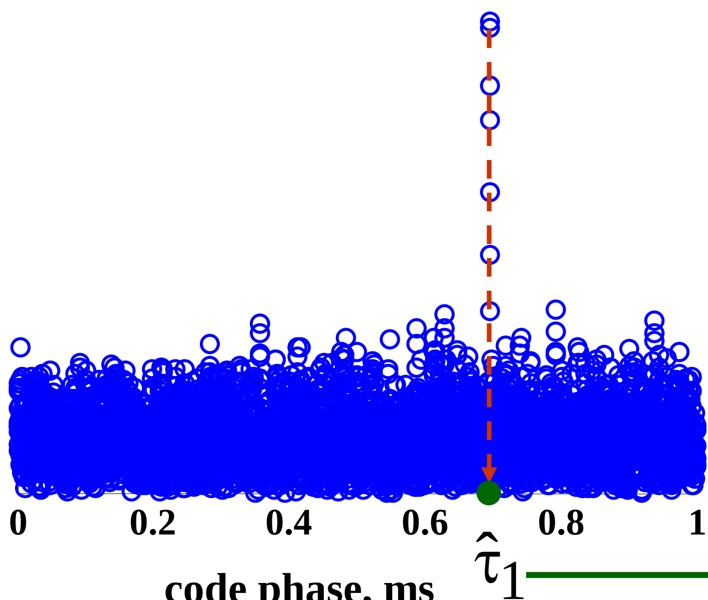
Example: CA-code phase search

1st estimation (1st batch)

No *a priori* knowledge of the code phase

Full search

Code autocorrelation



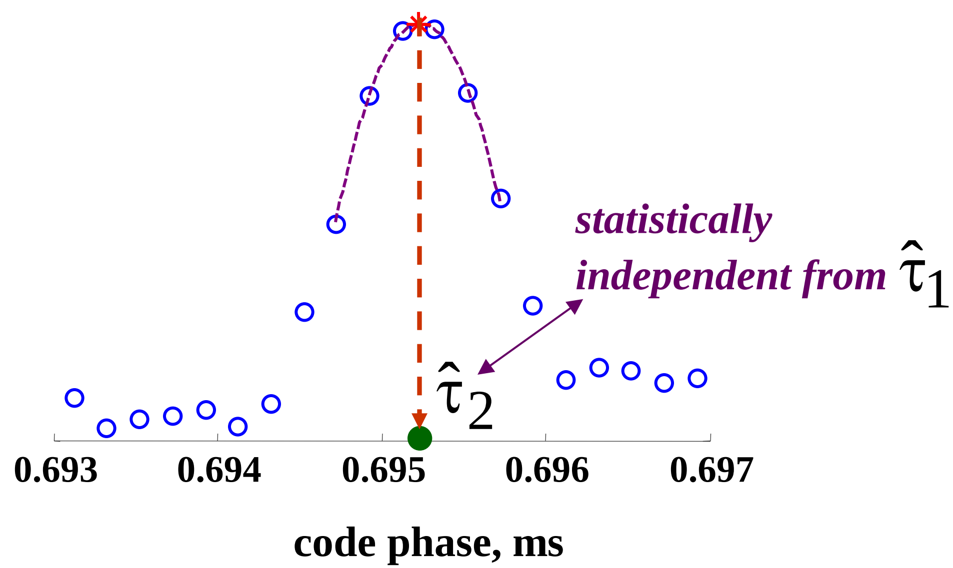
reducing the search space

2nd estimation (2nd batch)

Code phase is searched within $[\hat{\tau}_1 - 2T_{chip}, \hat{\tau}_1 + 2T_{chip}]$

Local search

Code autocorrelation



$\hat{\tau}_1$ only *helps* to *reduce the search space*, but *does not influence* the *estimation computation*.

Main Features of the Batch-Based Approach for GNSS Signal Processing

- ***Improved* signal *observability***
  - **Enhancement of observability of signal parameters utilizing parallel time domain, frequency domain and joint time-frequency domain techniques:**
    - **E.g.: Short-Time Fourier Transform and median filtering for high dynamic frequency tracking.**

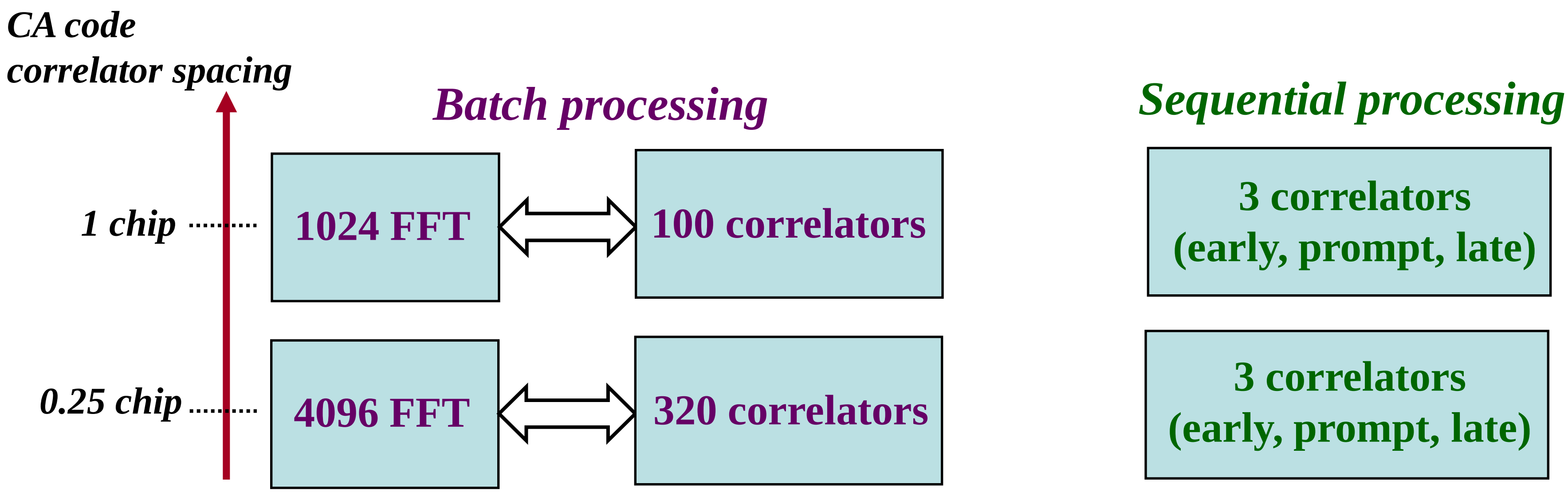
- ***Parallel* signal *processing* capabilities:**
    - **E.g. parallel code search via FFT.**

- ***Open loop tracking* architecture:**
  - **Immediate tracking recovery after a temporary signal loss;**
  - **Minimization of tracking dynamic sensitivity.**

Why Still Do Sequential?

To *minimize memory* and *computational resources*

- **Increasing signal parameter resolution does not require additional resources;**



- **Sequential processing is computationally cheaper for high-accuracy, wide-bandwidth tracking.**

Combining Complementary Features of Sequential and Batch Processing

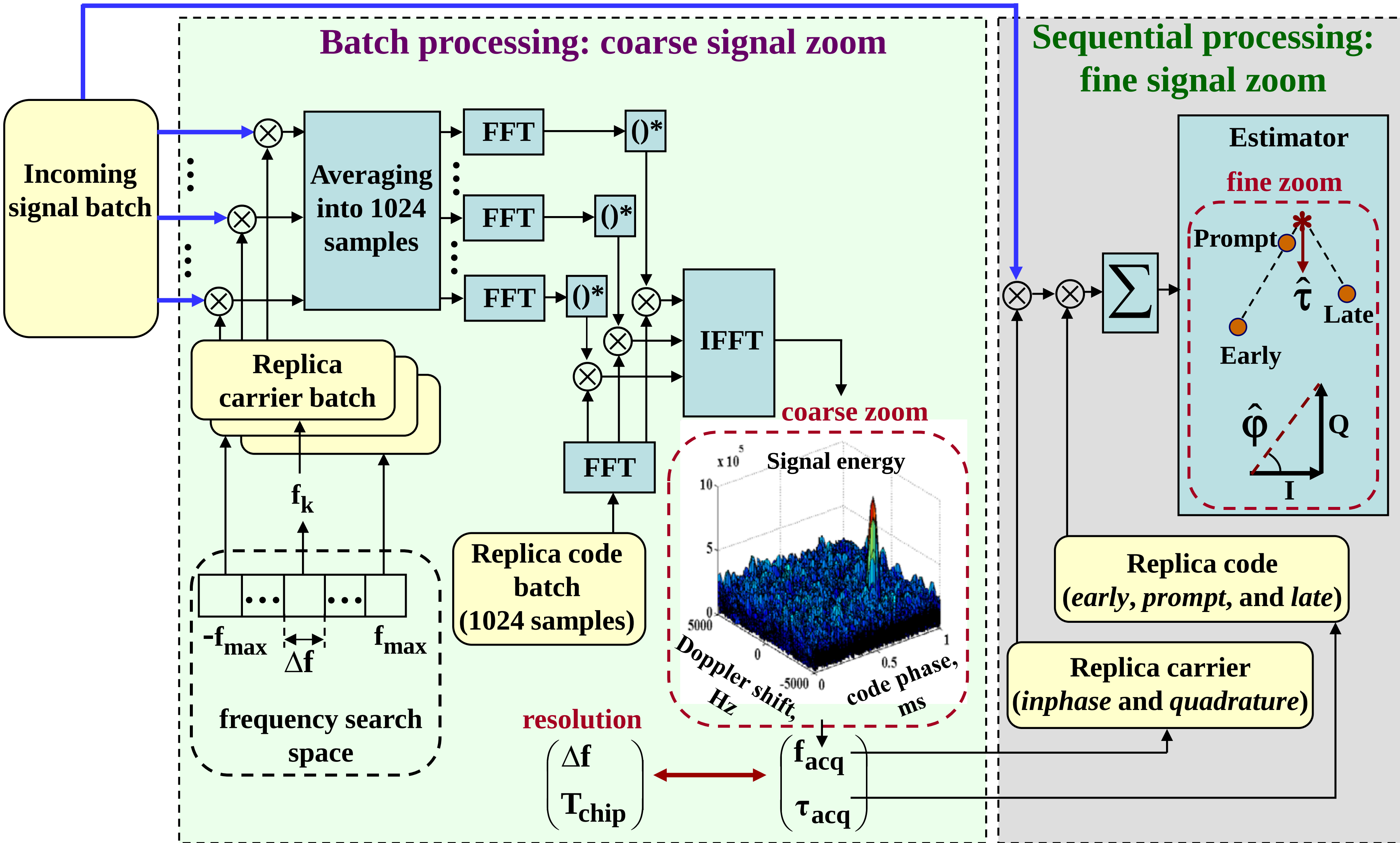
- **Consider *receiver designs* that *utilize both sequential and batch processing techniques*.**

- ***Key processing components* include:**
  - ***correlation*;**
  - ***measurement computation*;**
  - ***measurement exploitation*.**

Receiver Design Implementations

component \ processing type	Sequential processing	Batch processing	Combination of sequential/batch
Correlation	• Sequential correlation (time-domain).	• Batch parallel correlation (joint time-frequency domain).	• Batch-based parallel correlation for coarse signal localization combined with sequential correlation for fine signal zoom.
Measurement computation	• Time-domain sequential.	• Time-domain parallel; • Frequency-domain; • Joint time frequency domain.	• Combination of time-domain sequential estimation of signal parameters with batch-based parallel estimation of signal parameters. E.g.: time-frequency-domain estimation of the Doppler shift with sequential estimation of the code phase.
Measurement exploitation	• Measurement feedback via a closed tracking loop.	• No feedback; • Define local search space.	• Batch-based measurements aid sequential correlators via: *a) multiple passes through the same data batch* or *b) feed-forward aiding*.

Example of Batch/Sequential Combination Processing



Case Study: Deep GPS/IMU Integration

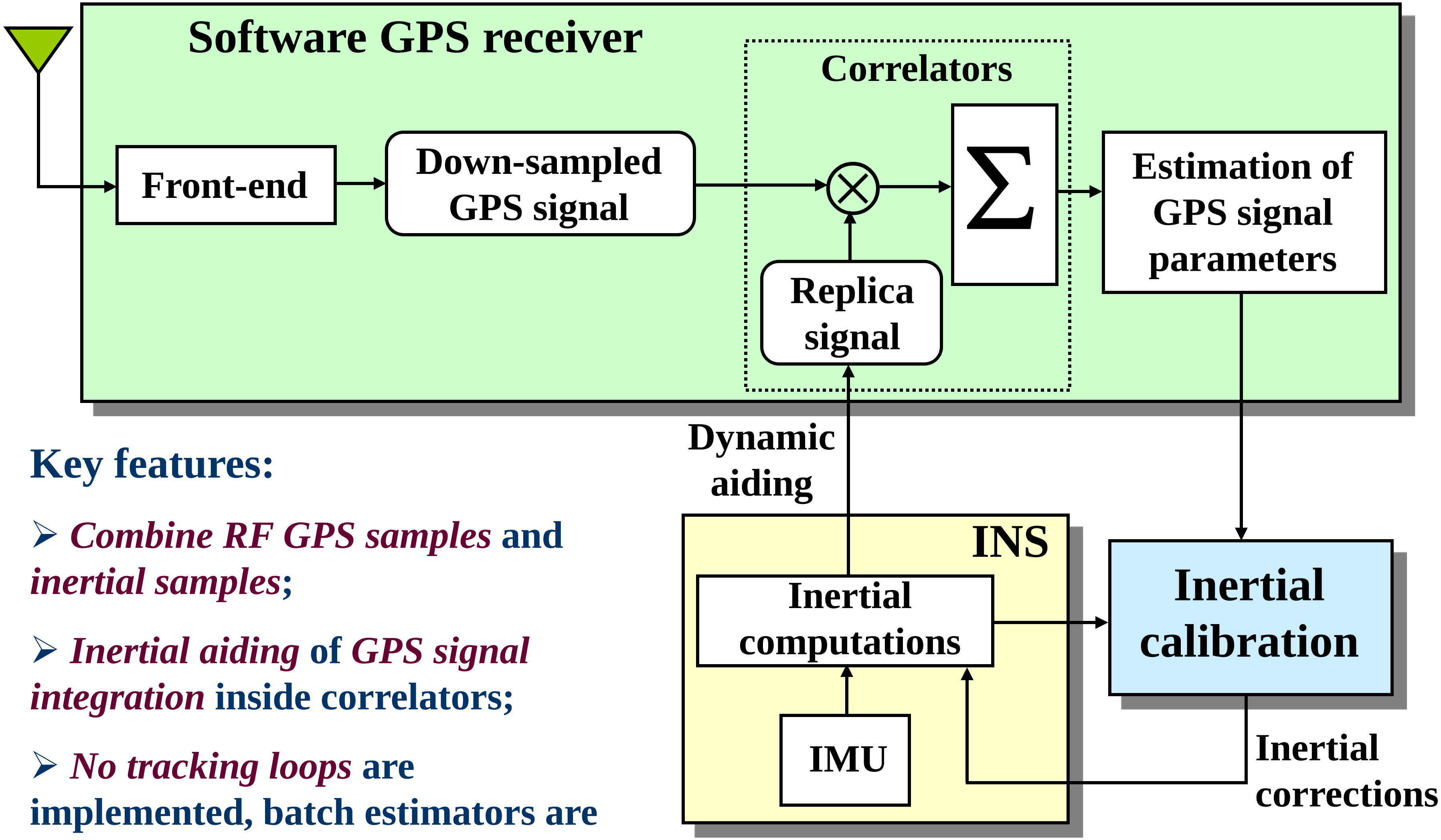
***Deep integration*: fusion of GPS signal samples and IMU measurements.**

***Deep integration goal*: to increase the signal integration time (e.g. to *improve* the *GPS tracking margin* by ~17 dB).**

***Deep integration implementation* is *determined* by the *integration mode*:**

- **Deep integration for the *code tracking only*;**
- **Deep integration for the *code and frequency tracking* with *known nav data bits*;**
- **Deep integration for the *code and carrier phase tracking* with *known nav data bits*;**
- **Deep integration for the *code and carrier phase tracking* with *data bit recovery*.**

Deeply Integrated GPS/IMU[1]: General Structure



Key features:

- ➢ *Combine RF GPS samples* and *inertial samples*;
- ➢ *Inertial aiding* of *GPS signal integration* inside correlators;
- ➢ *No tracking loops* are implemented, batch estimators are used instead.

[1] R. E. Phillips, G. T. Schmidt, GPS/INS Integration, AGARD Lecture Series, 1996.

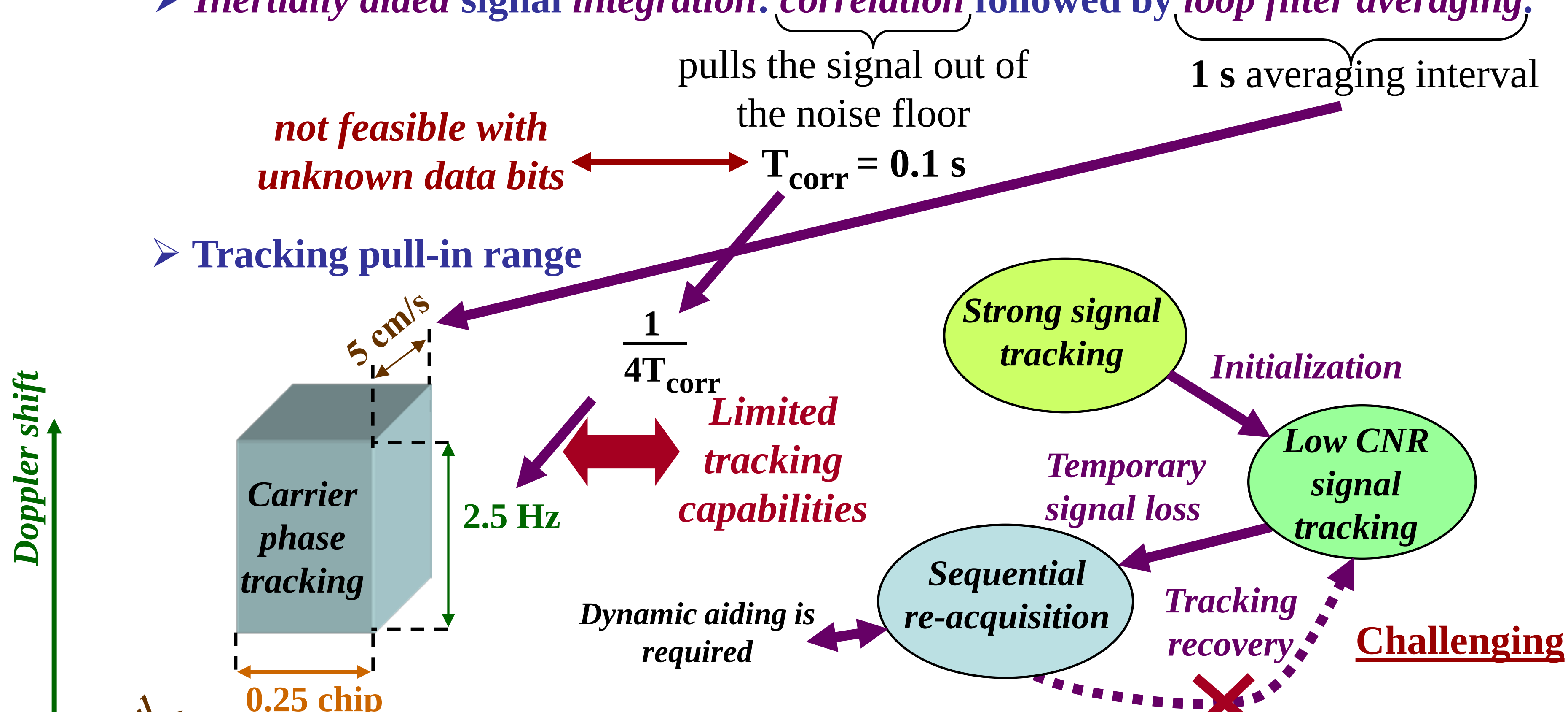
Motivation for Deep Integration

- **Conventional unaided GPS receiver:**
  - **GPS signal *integration time*: *10 – 20 ms*;**
  - ***CNR* required: *> 32 dB-Hz*;**
  - **Limited usage for a number of applications, e.g.:**
    - **Navigation in a presence of a wideband interference source;**
    - **Urban application;**
    - **Navigation under dense canopy, etc.**

- **Deeply integrated GPS/IMU:**
  - ***Inertial aiding* of *GPS signal integration* is implemented to significantly *increase* the *<u>coherent</u> integration time*;**
  - **Low CNR GPS signals (*CNR << 32 dB-Hz*) can be *acquired* and *tracked*.**

SEQUENTIAL Deep GPS/IMU Integration for Additional 17 dB Tracking Margin

- **Example case:** ***Deep integration for the code and carrier phase tracking with/without data bit recovery***
  - ***Inertially aided* signal *integration*: *correlation* followed by *loop filter averaging*.**
  - **Tracking pull-in range**



Example 1: Indoor tracking scenario where code phase jumps due to switching between multipath and direct signal.

Example 2: Frequency walk due to a low-cost inertial drift during the sequential frequency search.

BATCH Deep GPS/IMU Integration for Additional 17 dB Tracking Margin

❑ Example case: *Deep integration* for the *code and carrier phase tracking with/without data bit recovery*

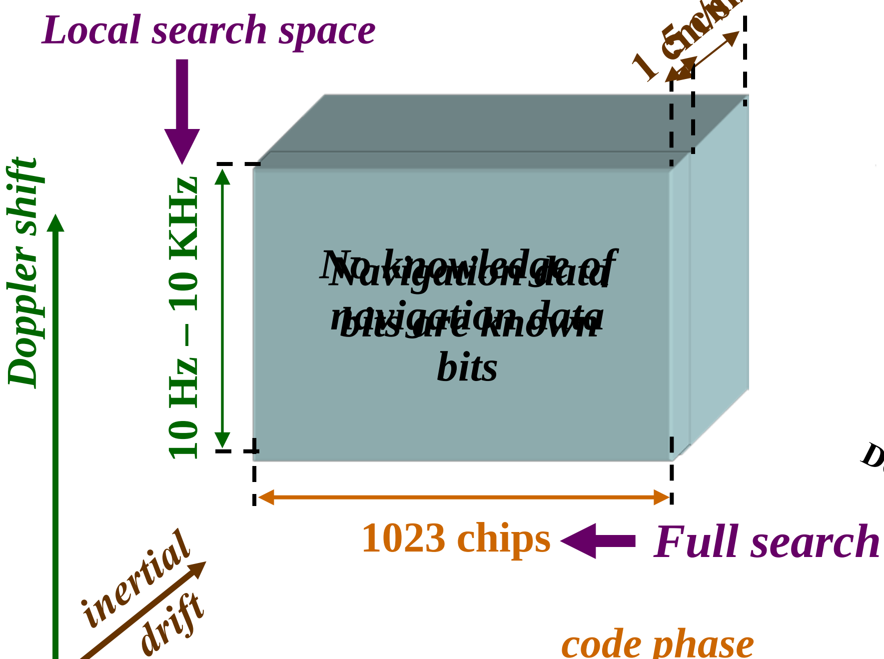
➢ *Inertially aided* signal *integration*: $\underbrace{\textit{correlation}}$ integration only.

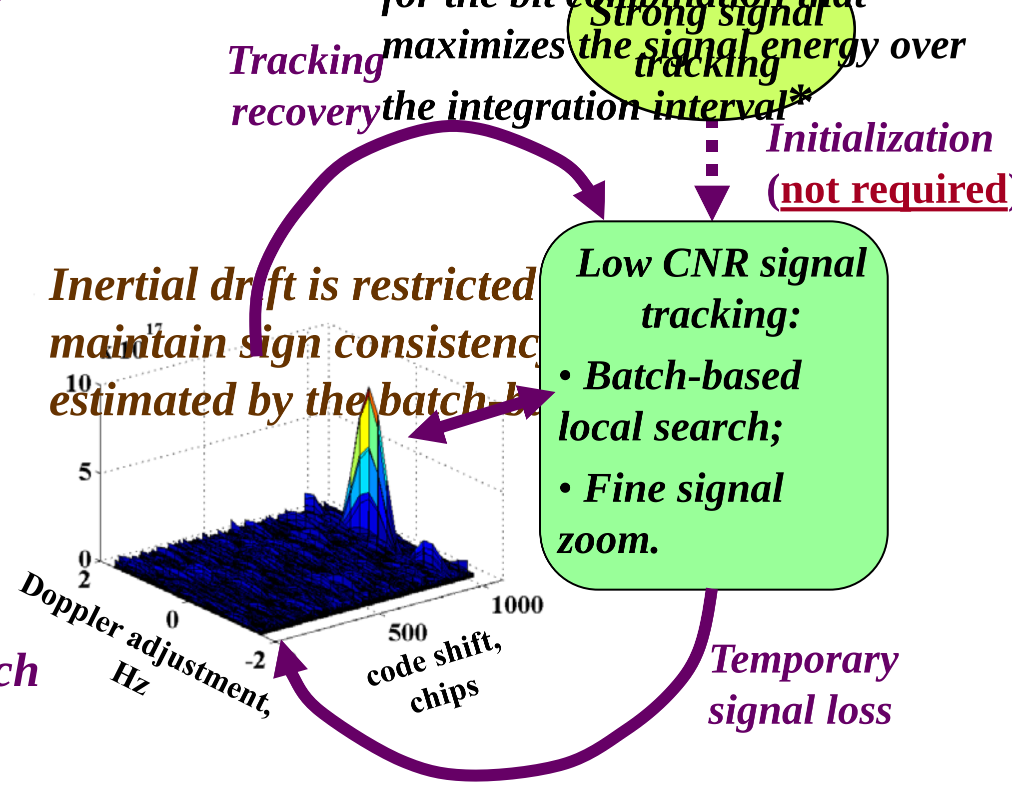
No knowledge of nav data bits is required ⟷ 1 s integration + batch-based bit estimation

*time-domain batch-based search for the bit combination that maximizes the signal energy over the integration interval**

➢ Tracking pull-in range

Defined by the local search space



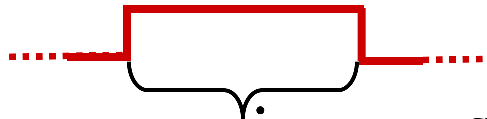


Batch-Based Wipe-Off of Navigation Data Bits

- **Energy-Based Bit Guessing Approach**
  - **I and Q are computed for all possible bit combinations for the tracking integration interval (0.1 s).**
  - ***Bit combination* that *maximizes* the signal *energy* (I^2+Q^2) is chosen.**
  - ***No additional corellators* are required to compute energy for possible bit combinations.**

Batch-Based Wipe-Off of Navigation Data Bits

- **Energy estimation for possible bit combinations**
  - **Accumulation of I and Q over intervals with no bit transitions**



$i_{(20ms)m}$, $q_{(20ms)m}$, $m=1,\ldots,5$

  - **Computation of possible I and Q values for the 0.1-s tracking integration interval**

$$\begin{bmatrix} \tilde{I}_1 \\ \cdots \\ \tilde{I}_{16} \end{bmatrix} = \mathbf{H} \cdot \begin{bmatrix} i_{(20ms)1} \\ \cdots \\ i_{(20ms)5} \end{bmatrix} \qquad \begin{bmatrix} \tilde{Q}_1 \\ \cdots \\ \tilde{Q}_{16} \end{bmatrix} = \mathbf{H} \cdot \begin{bmatrix} q_{(20ms)1} \\ \cdots \\ q_{(20ms)5} \end{bmatrix}$$

, where **H** contains possible **bit combinations** (*sign polarity* of bit combinations is resolved at a later stage)

Example:

$$\tilde{I}_2 = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & -1 \end{bmatrix}}_{\text{bit combination 2}} \cdot \begin{bmatrix} i_{(20ms)1} \\ \cdots \\ i_{(20ms)5} \end{bmatrix} = (1) \cdot i_{(20ms)1} + \ldots + (-1) \cdot i_{(20ms)5}$$

  - **Energy computation**

$$\begin{bmatrix} \tilde{E}_1 \\ \cdots \\ \tilde{E}_{16} \end{bmatrix} = \begin{bmatrix} \tilde{I}_1^2 + \tilde{Q}_1^2 \\ \cdots \\ \tilde{I}_{16}^2 + \tilde{Q}_{16}^2 \end{bmatrix} \begin{matrix} 1 \\ \cdots \\ 16 \end{matrix}$$

bit combination index

$$\boxed{\tilde{E}_{max} = \max(\tilde{E}_1, \ldots, \tilde{E}_{16}) \Rightarrow I^{\pm} = \tilde{I}_{max}, Q^{\pm} = \tilde{Q}_{max}}$$

Sign polarity of the energy-based bit detection

Deep Integration Case Studies

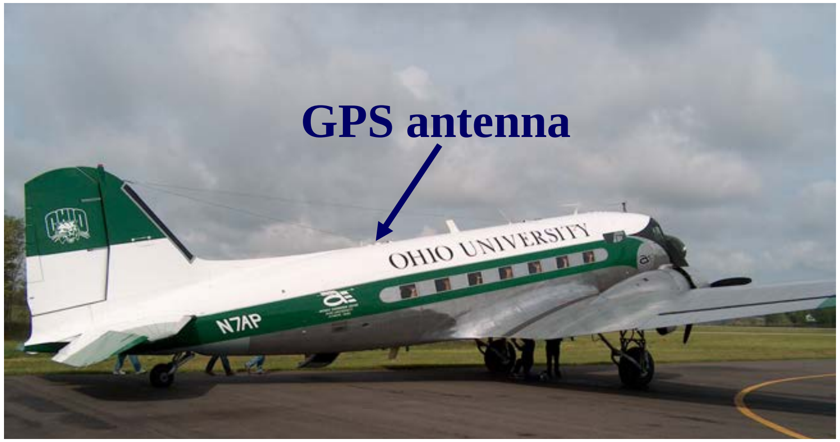
- Flight tests
- Assessment of GPS signal quality in urban environments

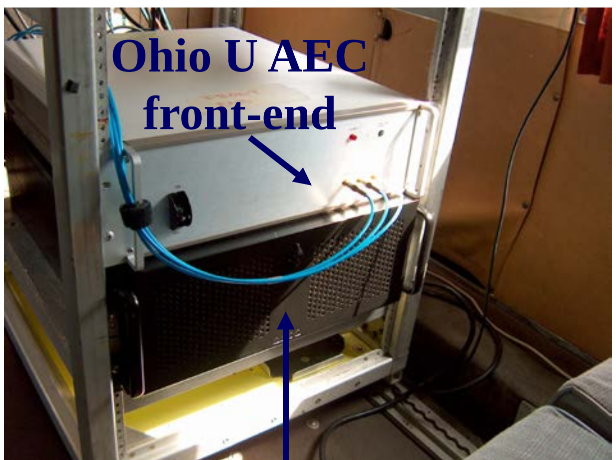
System Hardware Components

- **Software GPS receiver**
  - **Front-end developed at Ohio University Avionics Engineering Center**
    - **Downconverted carrier frequency: $f_{IF} = 1.27$ MHz;**
    - **Sample rate: 5 Msamples/s.**
- **Low-cost MEMS IMU**
  - **American GNC coremicro®**
    - **Sensor specs**

Parameter	Typical value
Gyro drift	0.1 deg/s
Gyro noise	$0.07\ dg/\sqrt{s}$ (sigma)
Accelerometer bias	2 mg
Accelerometer noise	1 mg
Axis misalignment	1 deg

Case Study 1: Flight Test



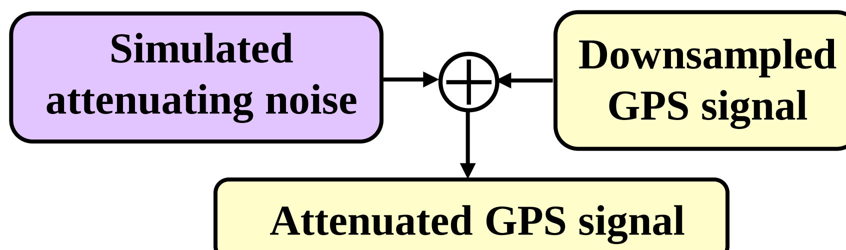


Data collection computer (AGNC IMU mounted inside)

- ❑ **Test scenarios:**
  - ➢ **Straight flight**
  - ➢ **90-deg turn**

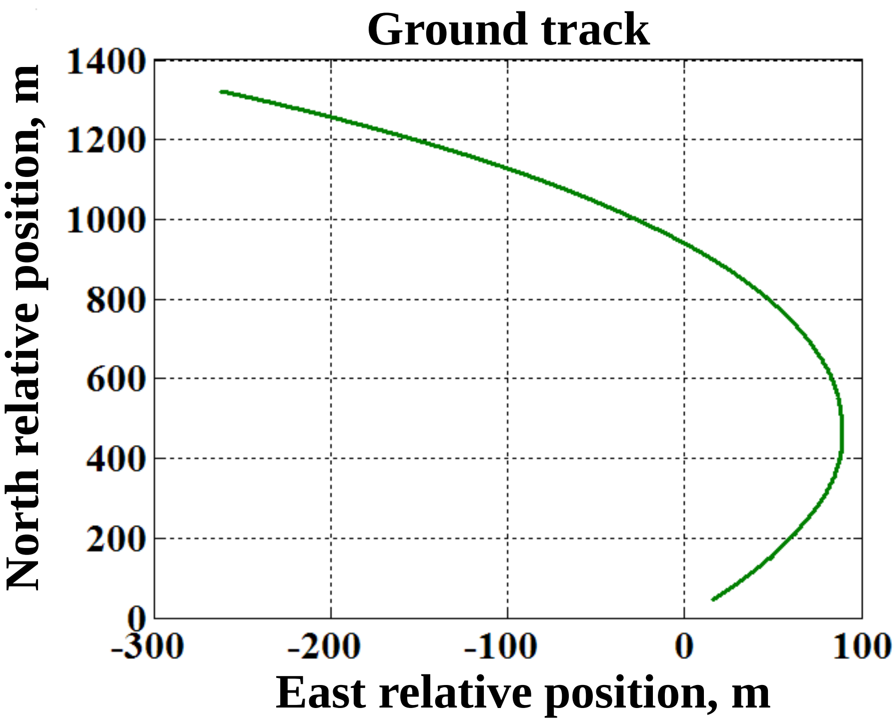
- ❑ **Processing durations**
  - ➢ **Initialization phase: 5 s;**
  - ➢ **Weak signal processing: 20 s.**

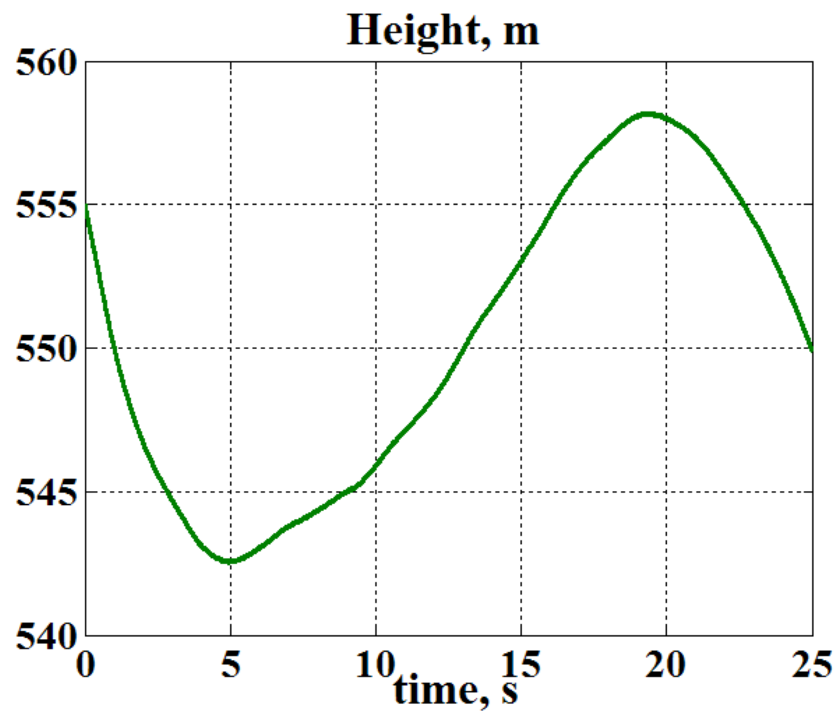
- ❑ **Signal attenuation**



90-deg Turn: Trajectory

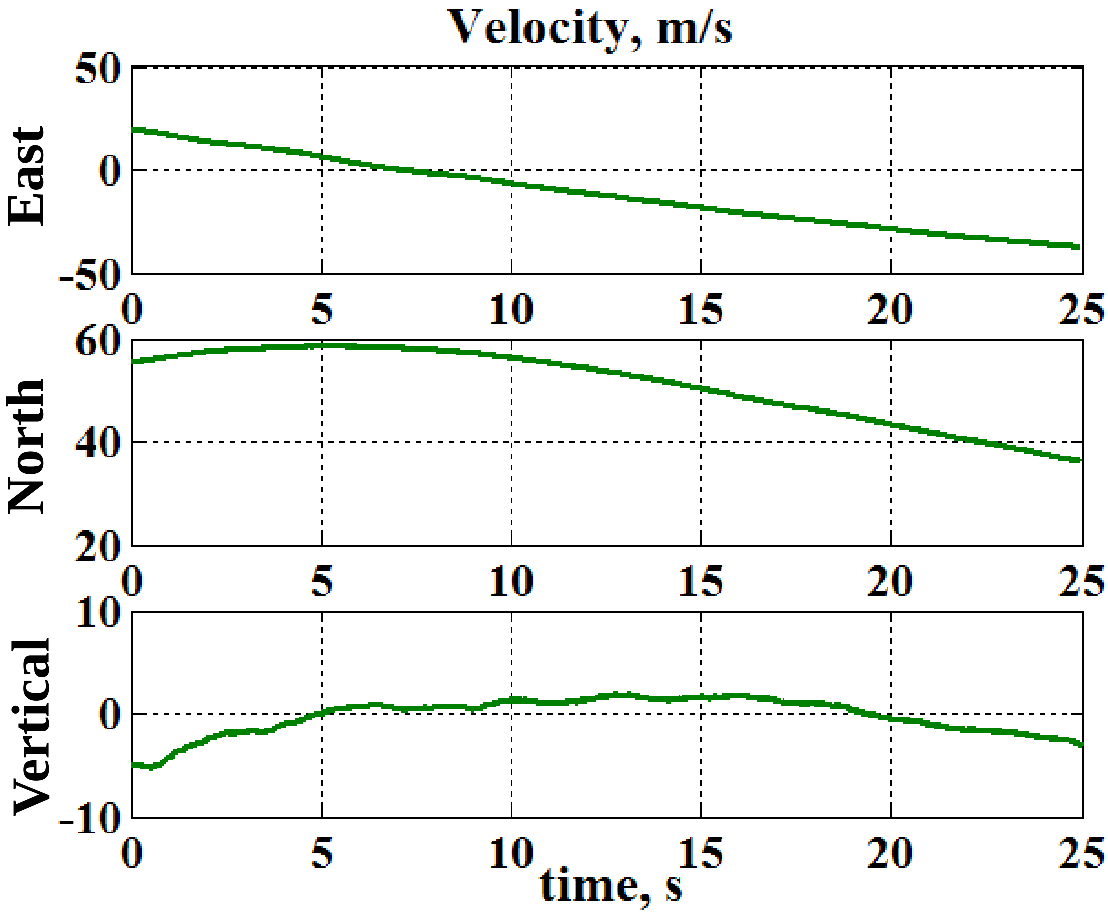
Flight trajectory **is defined by** ***relative position*** **derived from** ***strong signal accumulated Doppler measurements*****.**





90-deg Turn: Velocity Profile

Velocity profile **is derived from** ***strong signal accumulated Doppler measurements*****.**

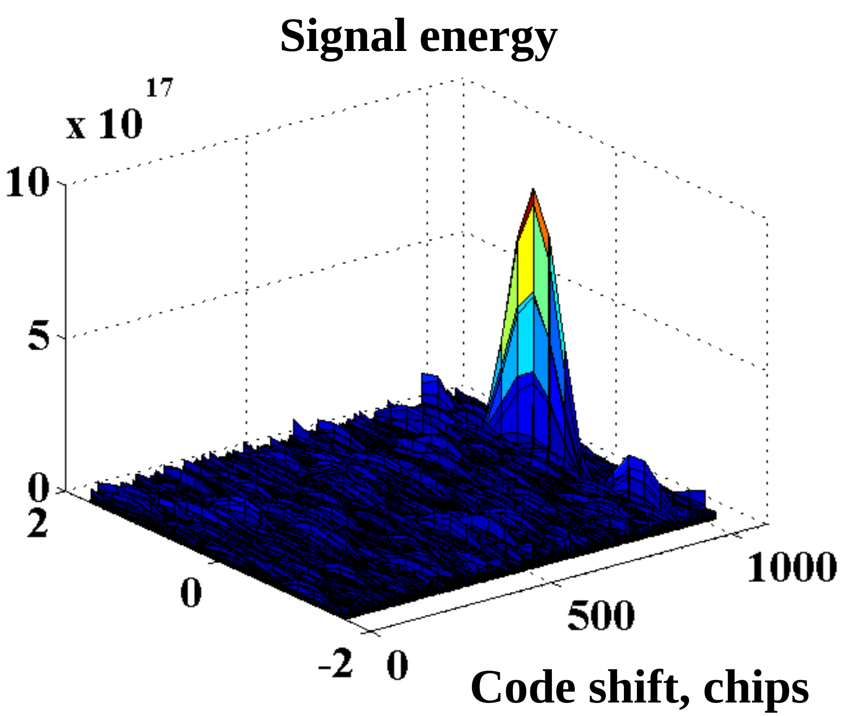


Test Results (90-deg Turn)

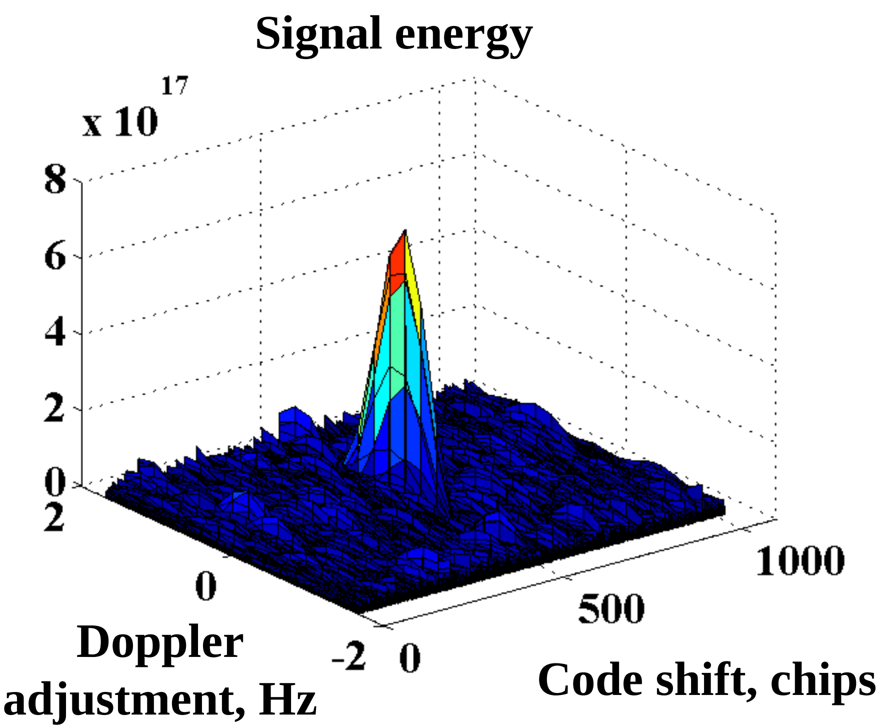
IMU-aided GPS Signal Acquisition (1.2-s signal integration)

- Examples of Acquisition Plots

SV 4 (CNR ~ 17 dB-Hz)



SV 30 (CNR ~ 15 dB-Hz)

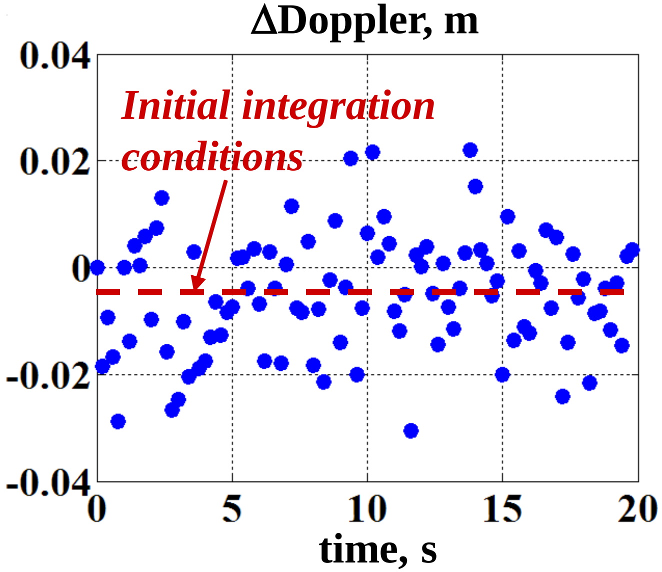


Test Results (90-deg Turn)

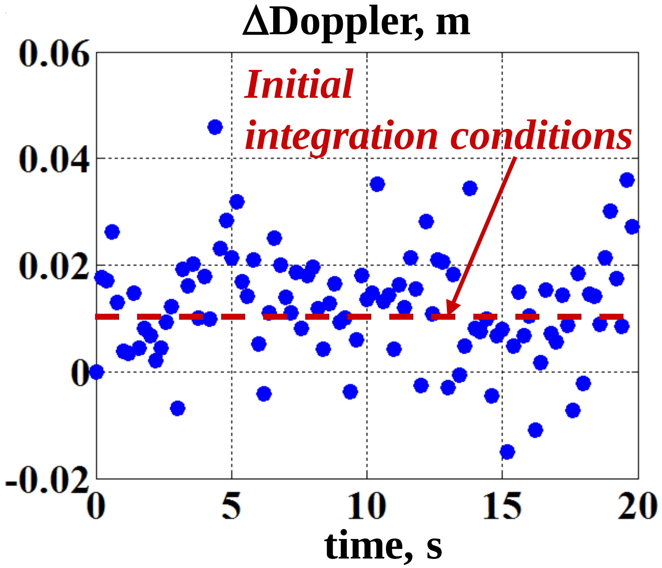
IMU-aided GPS Signal Tracking (0.4-s signal integration)

- ΔPhase = (Carrier phase)$_{\text{strong signal}}$ - (Carrier phase)$_{\text{low CNR signal}}$

SV 10 (CNR range is [17,18] dB-Hz)



SV 30 (CNR range is [15,18] dB-Hz)



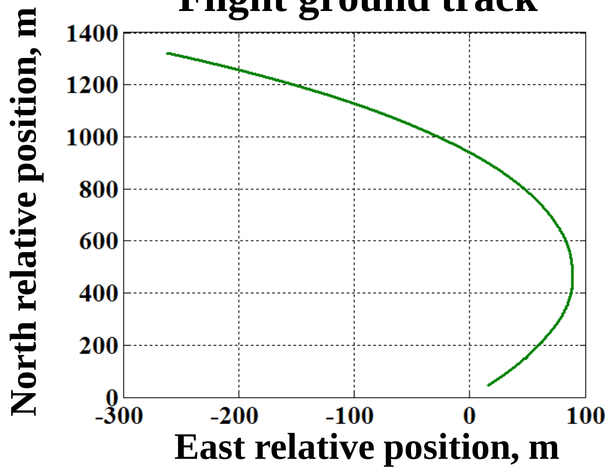
SVID	CNR range (dB-Hz)	ΔPhase std (cm)
4	[18,20]	0.97
5	[18,20]	0.77
7	[17,19]	1.09
10	[17,18]	0.90
30	[15,18]	1.04

Deep GPS/IMU Integration: Flight Test Results for Batch and Sequential Processing Strategies

Deep GPS/low-cost IMU integration for *code and carrier phase tracking* with *data bit recovery*

Stressful, but practical example

Flight ground track



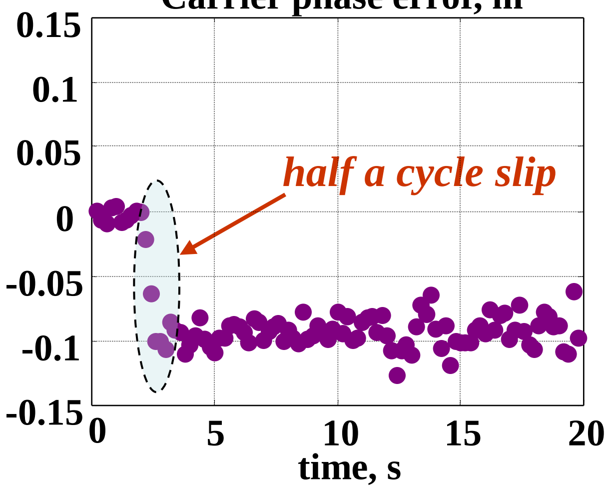
***Flight test results demonstrate* that *batch processing provides* at least a *8 dB increase* in the *tracking margin* as *compared* to the *sequential approach*.**

6 dB lower

Sequential processing

SV 30 (CNR range is [21, 24] dB-Hz)

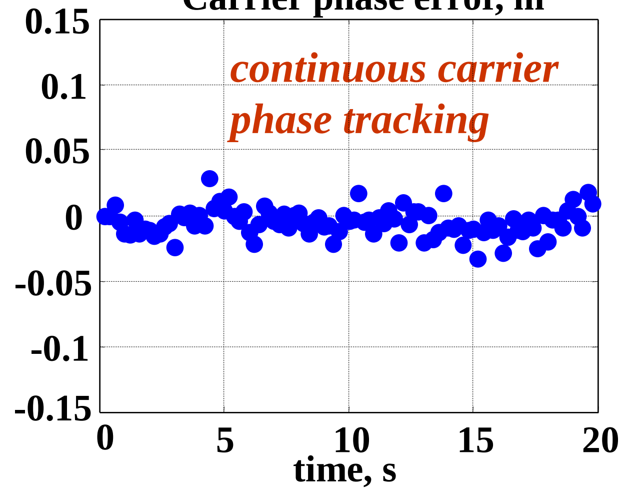
Carrier phase error, m



Batch processing

SV 30 (CNR range is [15,18] dB-Hz)

Carrier phase error, m

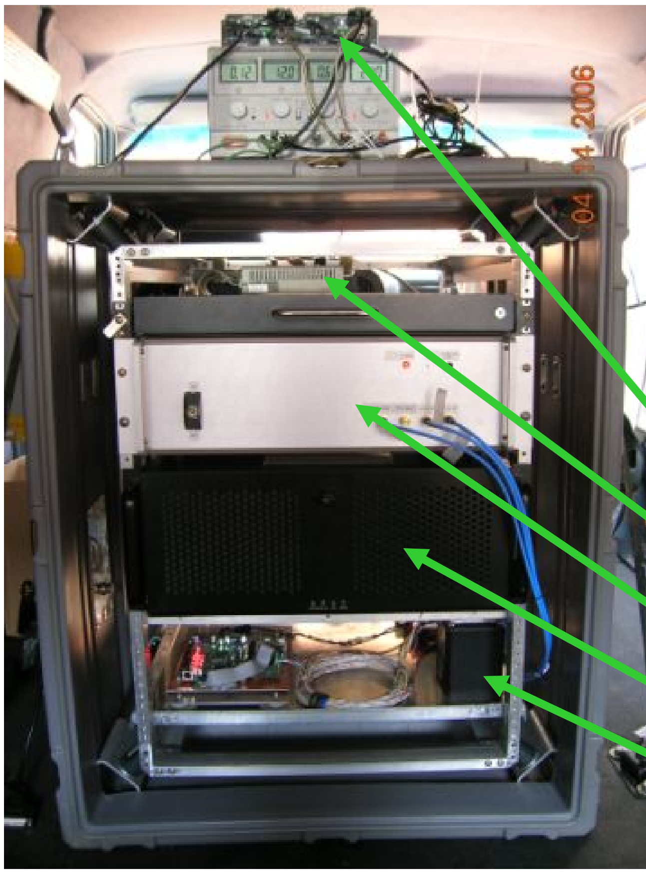


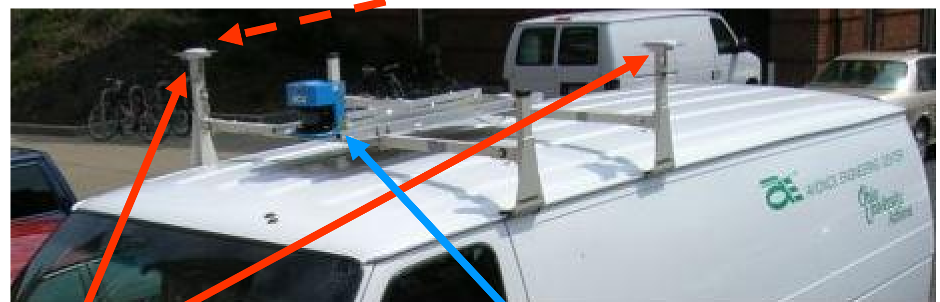
Case Study 2: Urban Navigation

- **Study GPS signals in urban environments:**
  - **Use batch processing techniques to observe the entire image of the signal**
  - **Apply long signal integration (~ 1 s) to improve the signal availability**

Equipment Setup

NovAtel L1/L2 pinwheel antenna substituted here





GPS antennas

Laser sensor — *potential augmentation to GPS*

NovAtel OEM-4 GPS receivers
Controller for laser sensor
— *for sequential processing*

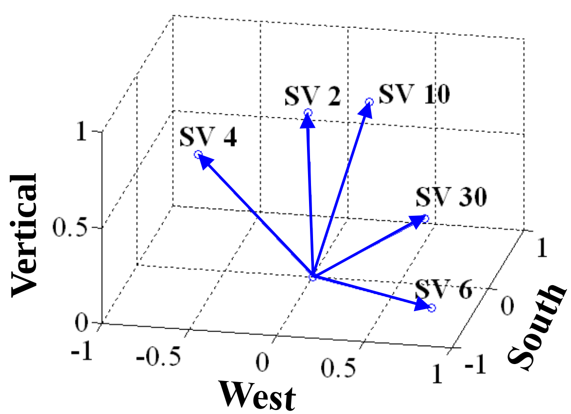
Software radio RF components
Software radio digital components
— *for batch processing*

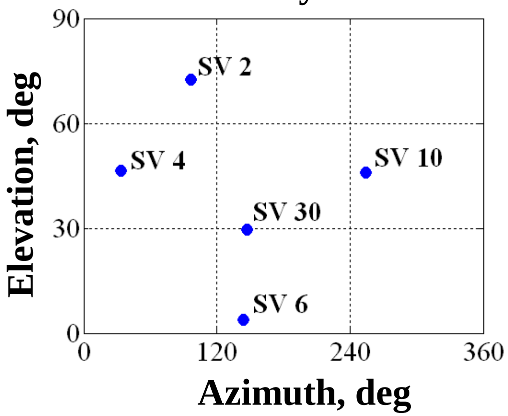
Inertial Measurement Unit and circuitry

Example Stationary Test Results

Data collection scenery

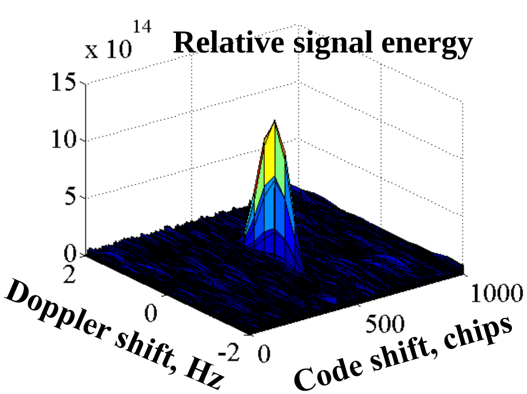
5 satellites were acquired and tracked.



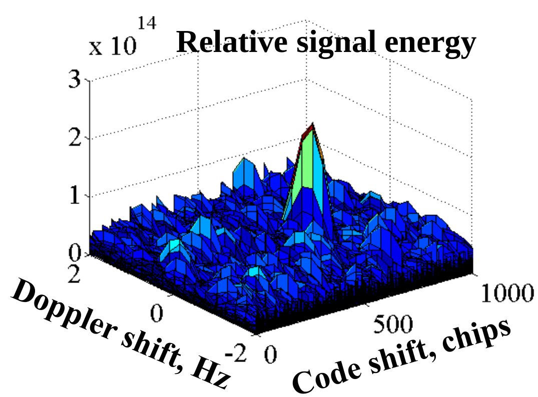


Examples of 3D signal images

SV 2 (CNR ~ 19 dB-Hz)



SV 6 (CNR ~ 12 dB-Hz)

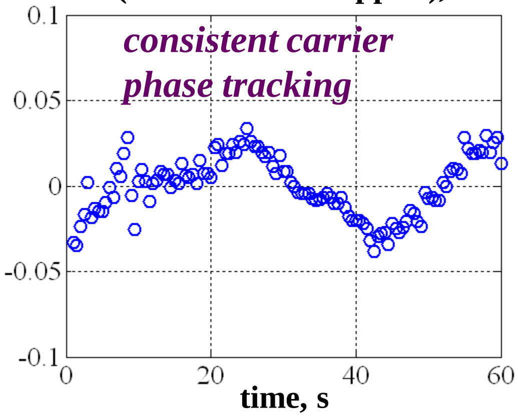


Example Stationary Test Results

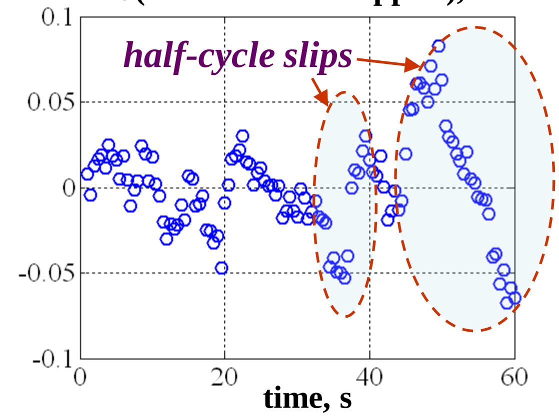
- **GPS signal tracking quality: tracking consistency**
  - **Consistency of carrier phase measurements for individual SV channels**

Accumulated Doppler measurements
compensated for SV motion; and receiver clock, iono and tropo first-order drifts components

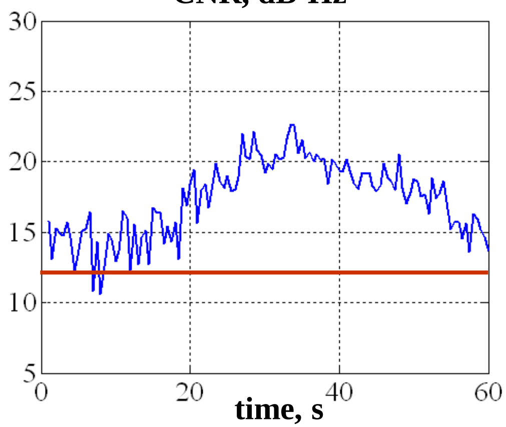
SV 2
δ(Accumulated Doppler), m



SV 4
δ(Accumulated Doppler), m

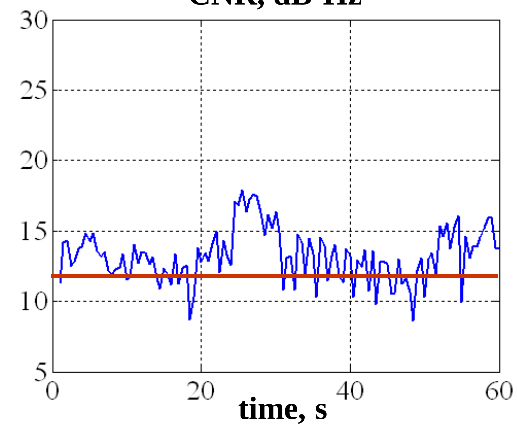


CNR, dB-Hz



Carrier phase tracking threshold ~ 12 dB-Hz

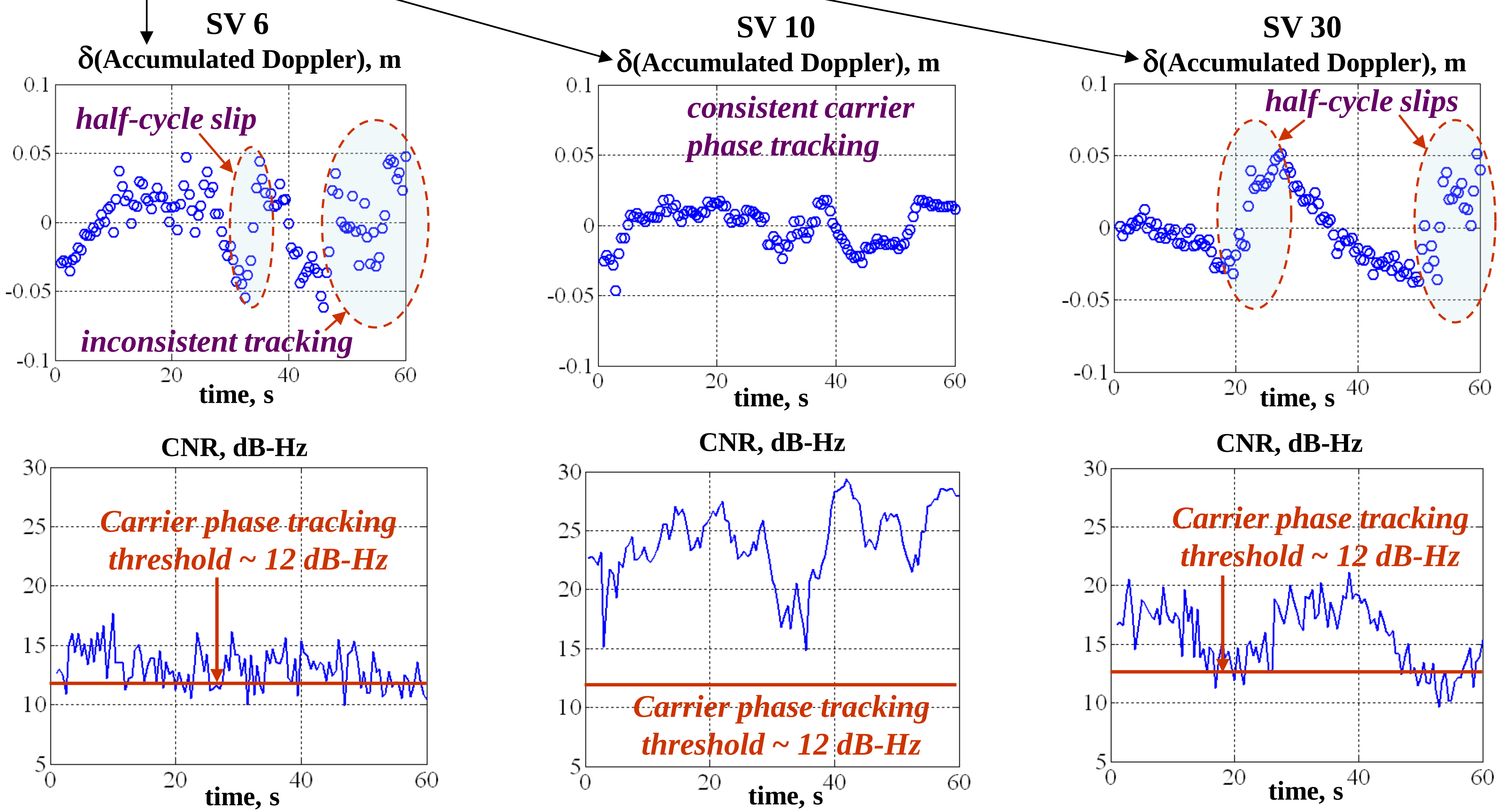
CNR, dB-Hz



Example Stationary Test Results

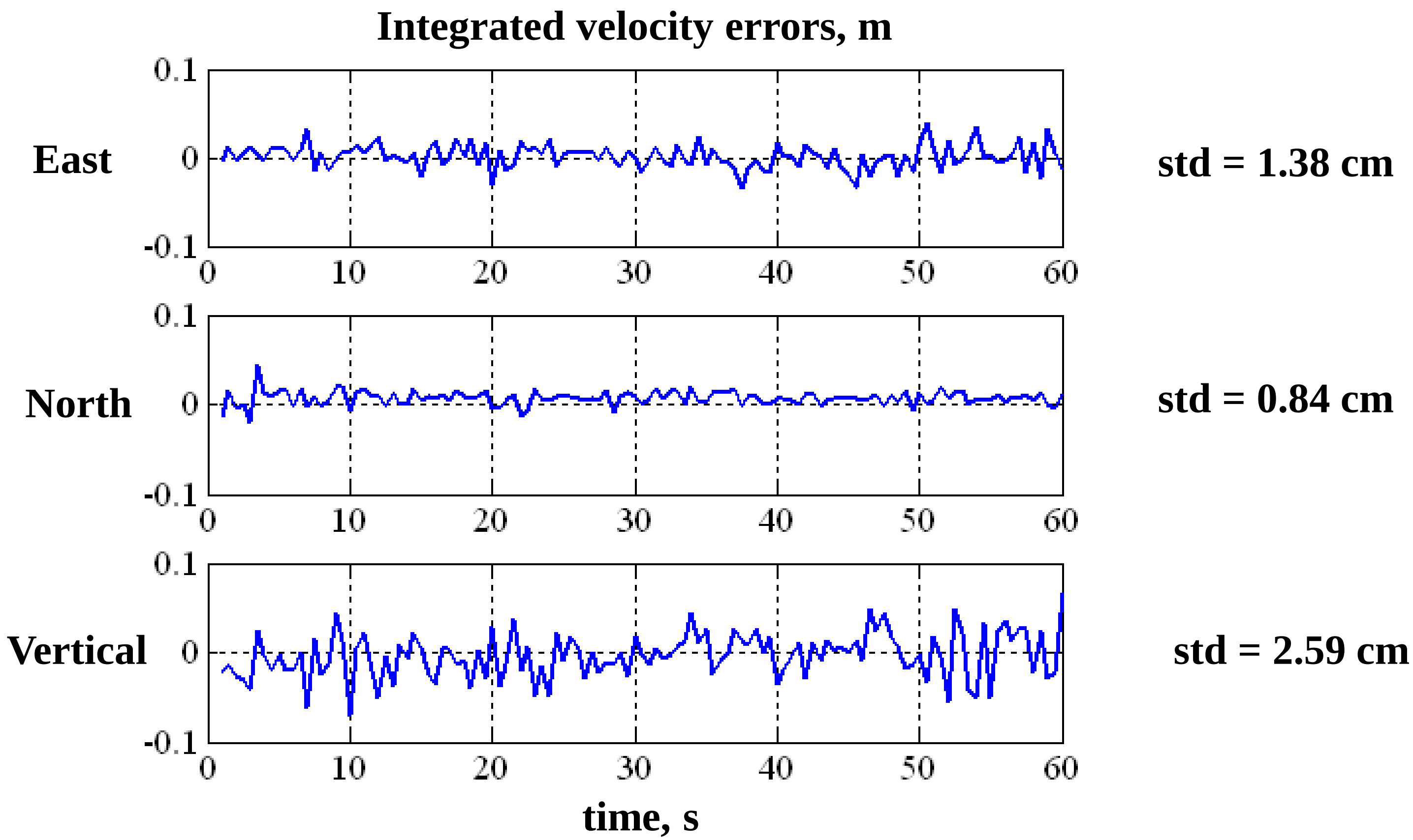
- ❑ **GPS signal tracking quality: tracking consistency**
  - ➢ **Consistency of carrier phase measurements for individual SV channels**

Accumulated Doppler measurements
compensated for SV motion; and receiver clock, iono and tropo first-order drifts components

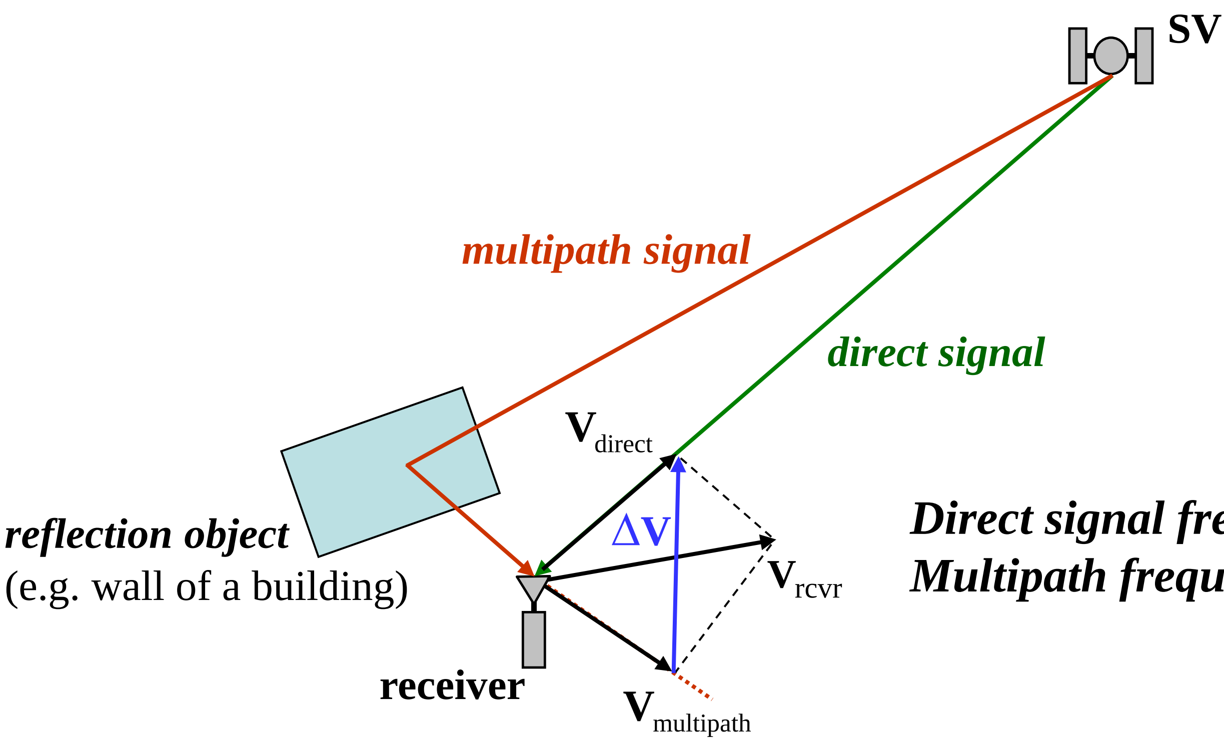


Example Stationary Test Results

- **GPS signal tracking quality: accuracy performance**
  - **Integrated velocity derived from carrier phase measurements**



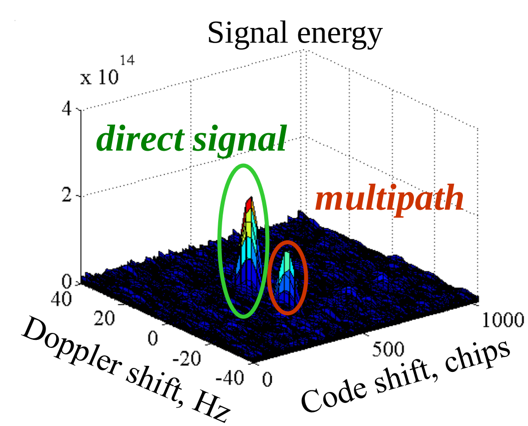
Dynamic Test: Effect of Frequency Multipath



Direct signal frequency – Multipath frequency ~ ΔV

***Multipath frequency* can *differ significantly* from the *direct signal frequency* due to:**

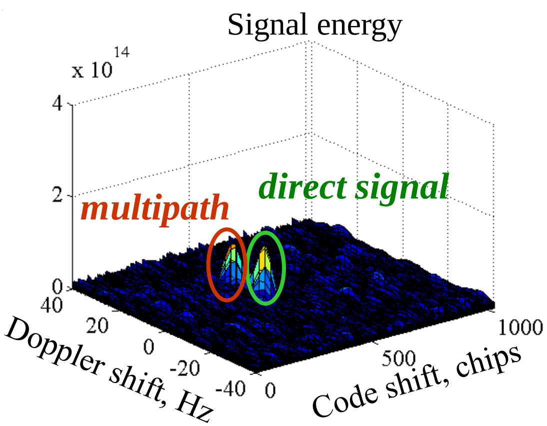
- ***Non-zero receiver velocity*;**
- ***Difference* between *SV - receiver Line-of-Sight* (LOS) *vector* and *reflecting object – receiver LOS vector*.**



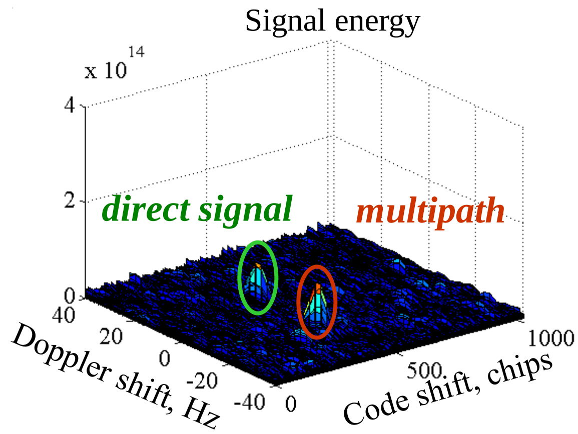
Dynamic Test: Illustration of Frequency Multipath

- 3D signal images: signal integration over 0.1 s intervals

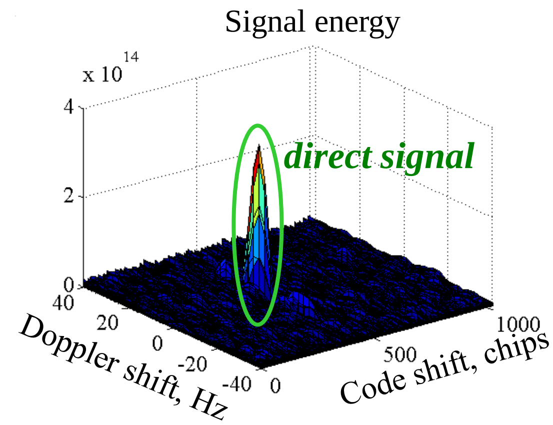
Integration interval: [0, 0.1] s



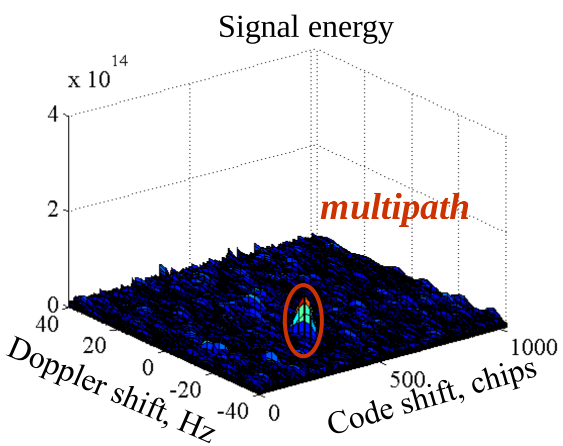
Integration interval: [0.1, 0.2] s



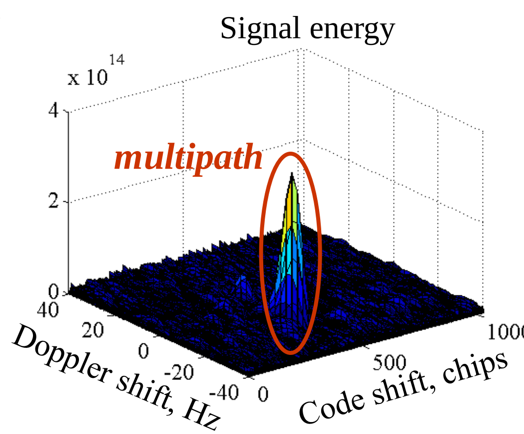
Integration interval: [0.2, 0.3] s



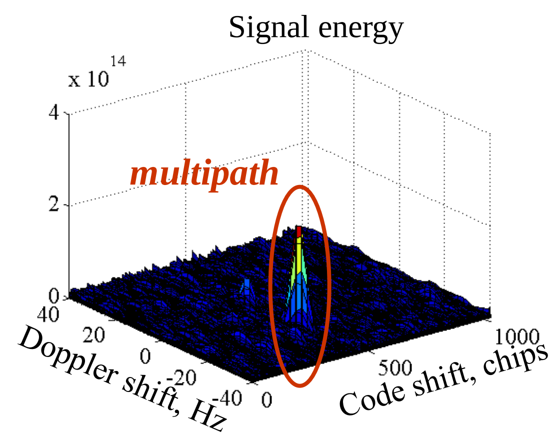
Integration interval: [0.3, 0.4] s



Integration interval: [0.4, 0.5] s



Integration interval: [0.5, 0.6] s



Batch processing is instrumental to observe these signals!

Conclusion

For improved tracking performance, receiver design needs to be considered in terms of *BATCH vs. SEQUENTIAL processing <u>not</u> in terms of the implementation platform* (ASIC, software receiver, software radio, DSP, FPGA)