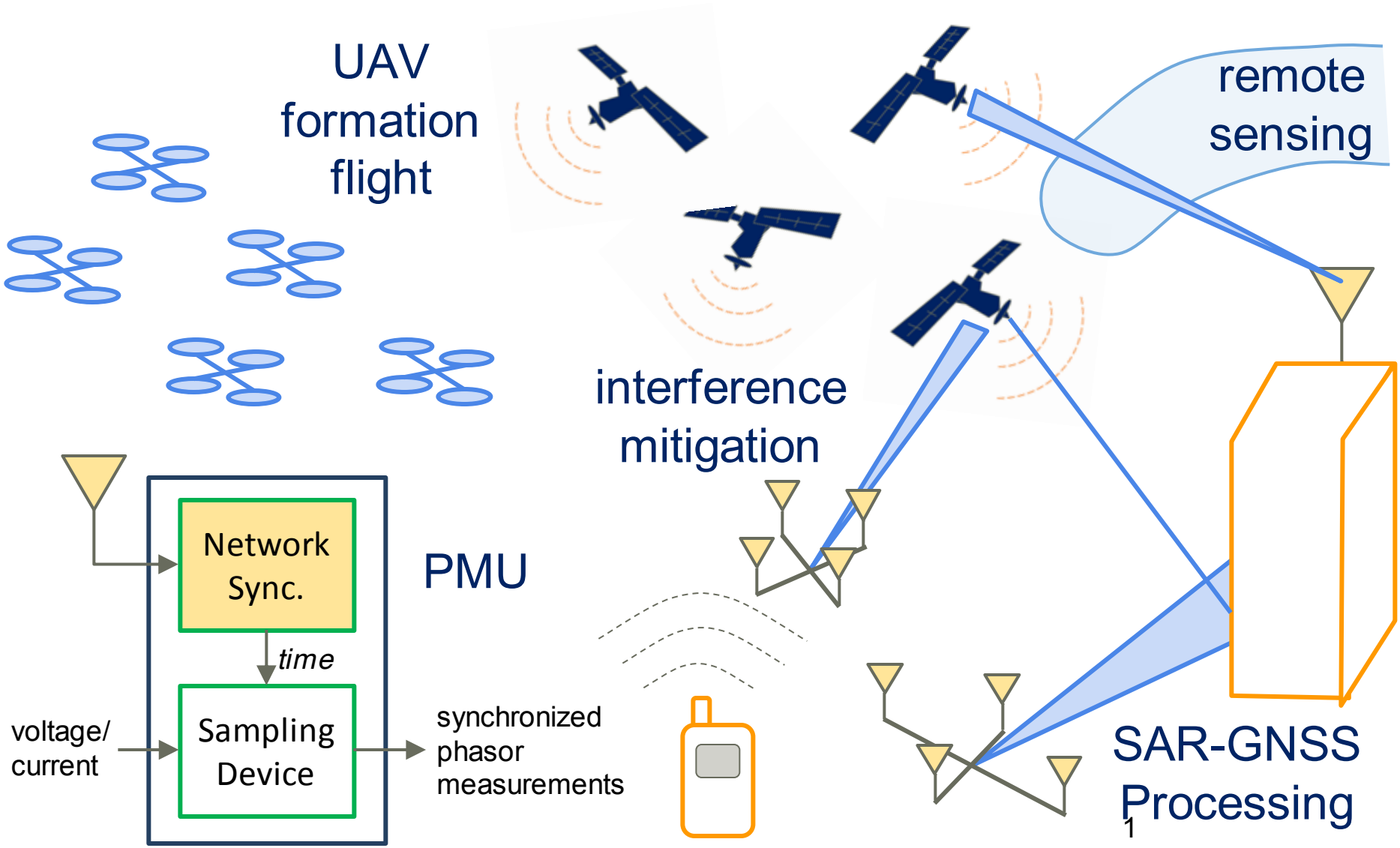
Multi-Receiver Vector Tracking

Yuting Ng and Grace Xingxin Gao

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

please feel free to view the .pptx version for the speaker notes

Cutting-Edge Applications



State-of-the-Art Techniques

increased availability

EKF

VDLL

IMU

VFLL

beamstearing

Signal tracking

clock

Multiple antennas

External aiding

PPP

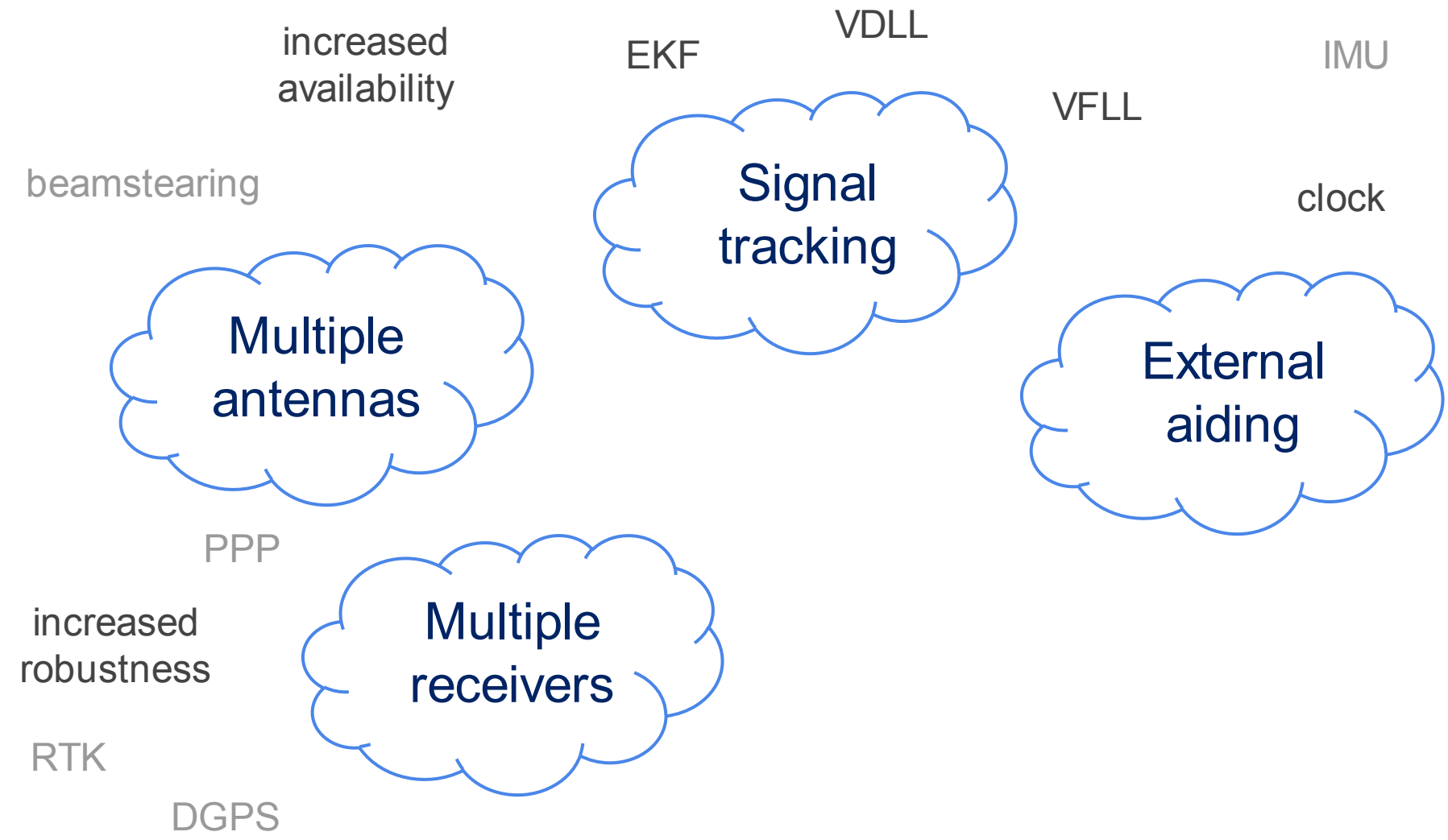
increased robustness

Multiple receivers

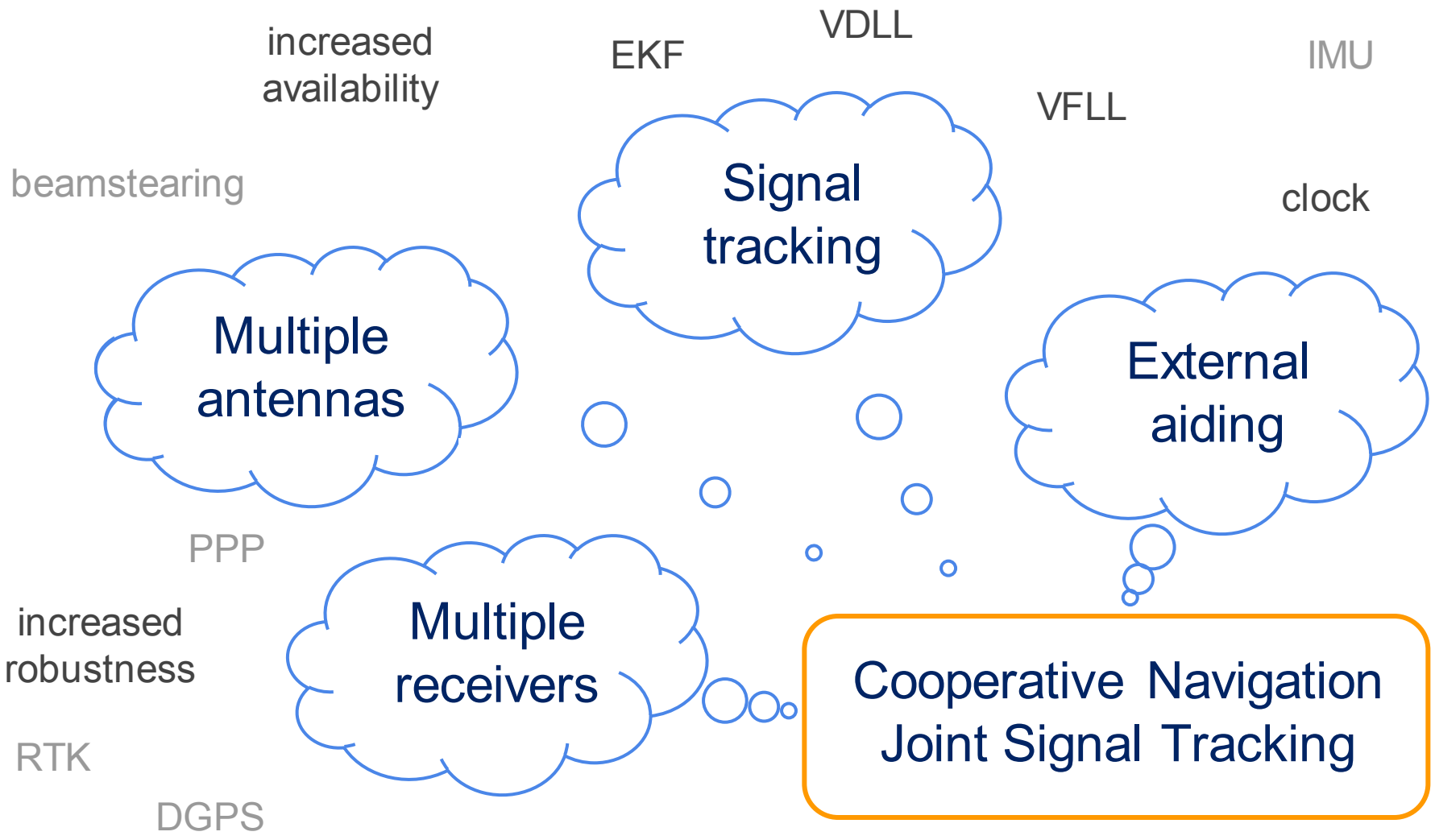
RTK

DGPS

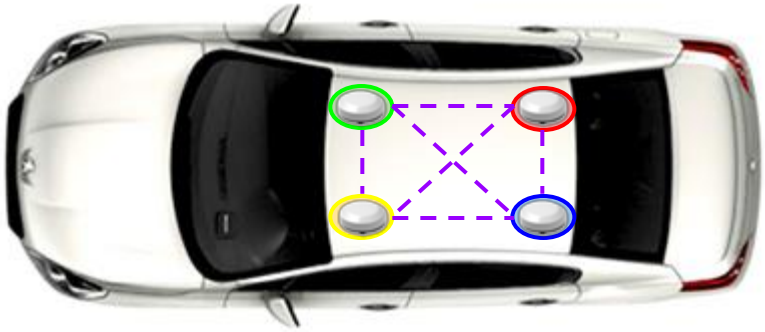
State-of-the-Art Techniques

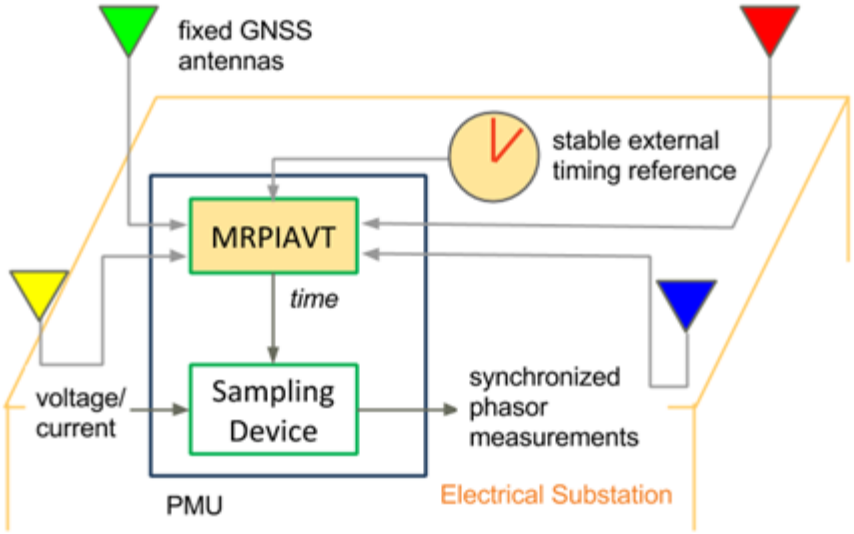


State-of-the-Art Techniques



Multi-Receiver Vector Tracking



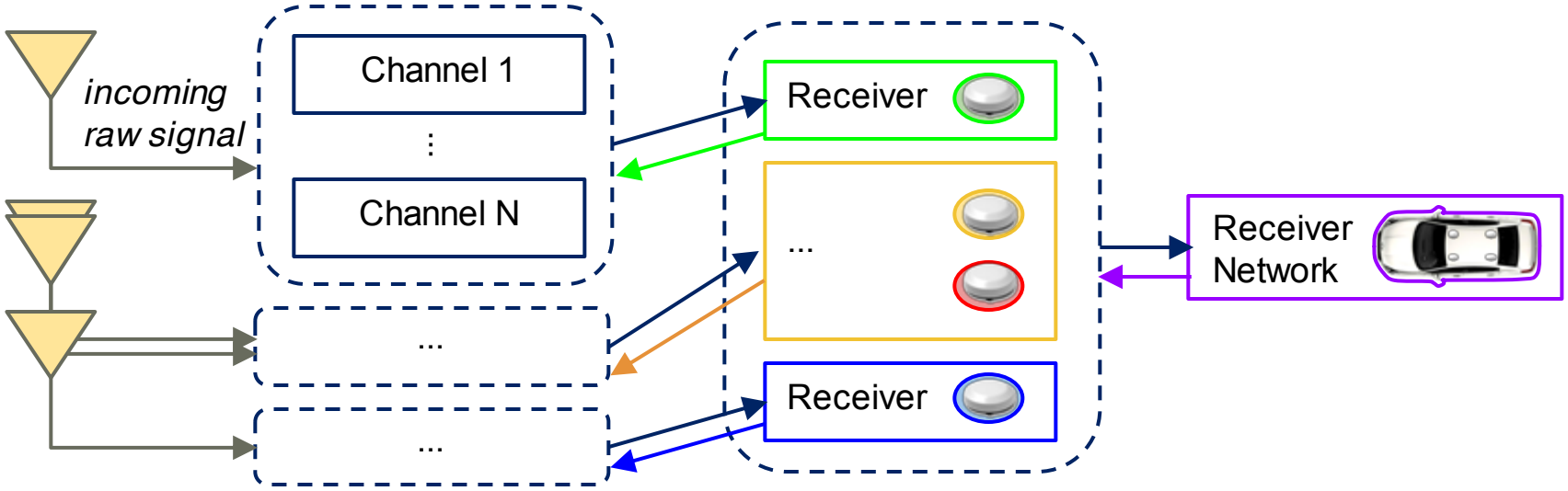


Joint tracking of **multiple** GNSS receivers for increased accuracy, robustness and reliability

Outline

- Motivation | Coop Nav, Joint Signal Tracking
- Our Approach | Multi-Receiver Vector Tracking
- Implementation | Hardware, Python SDR
- Experimental Validation
- Summary

MRVT Receiver Architecture



Scheme	Scalar	SRVT	MRVT
Entities	N x M Channels	M Receivers	1 Network
Unknowns	N x M x 4	M x 8	1 x 14

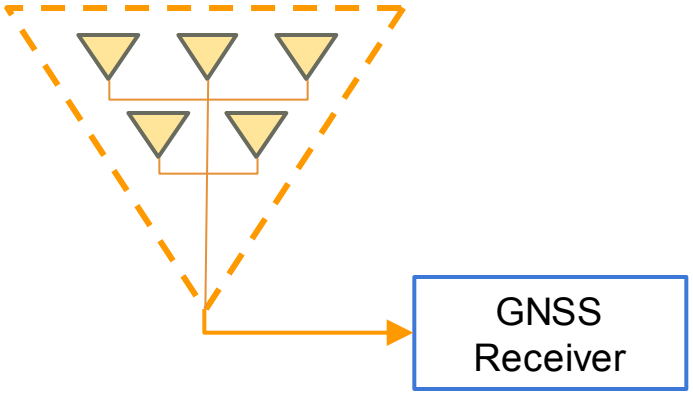
N Channels
M Receivers
4 Unknowns
$[\phi_{carr}\ \phi_{code}\ f_{carr}\ f_{code}]$

M Receivers
8 Unknowns
$[x\ y\ z\ c\delta t\ \dot{x}\ \dot{y}\ \dot{z}\ c\dot{\delta}t]$

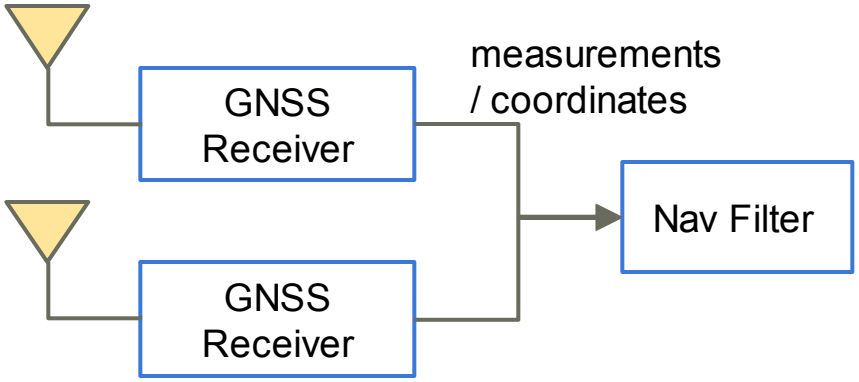
1 Network
14 Unknowns
$[x\ y\ z\ c\delta t\ \dot{x}\ \dot{y}\ \dot{z}\ c\dot{\delta}t]$
$[\psi\ \theta\ \phi\ \dot{\psi}\ \dot{\theta}\ \dot{\phi}]$

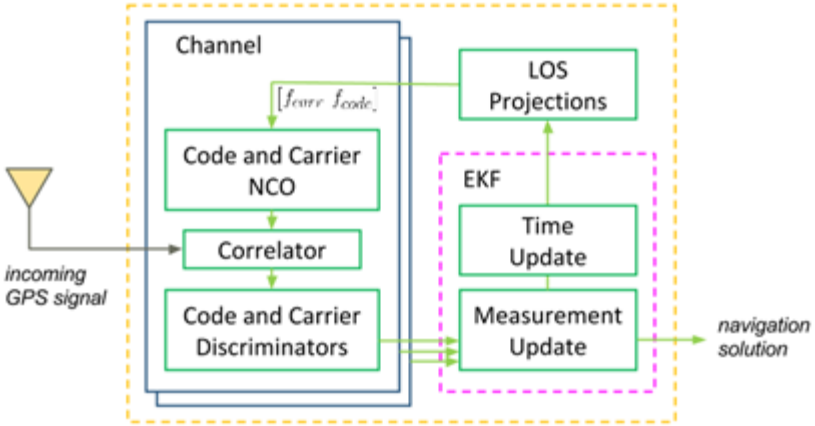
Comparison with other GNSS Processing Implementations

Antenna array

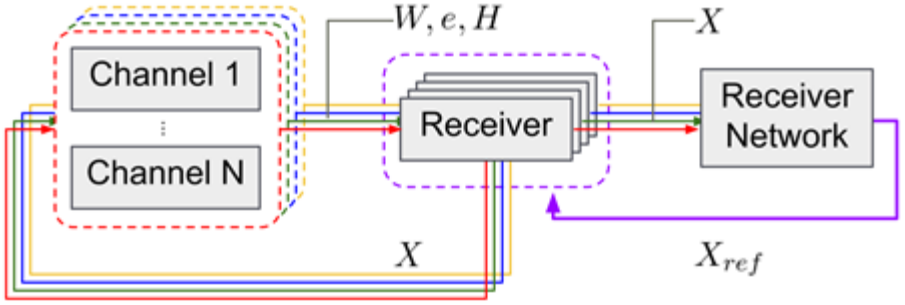


Differential GPS (DGPS)



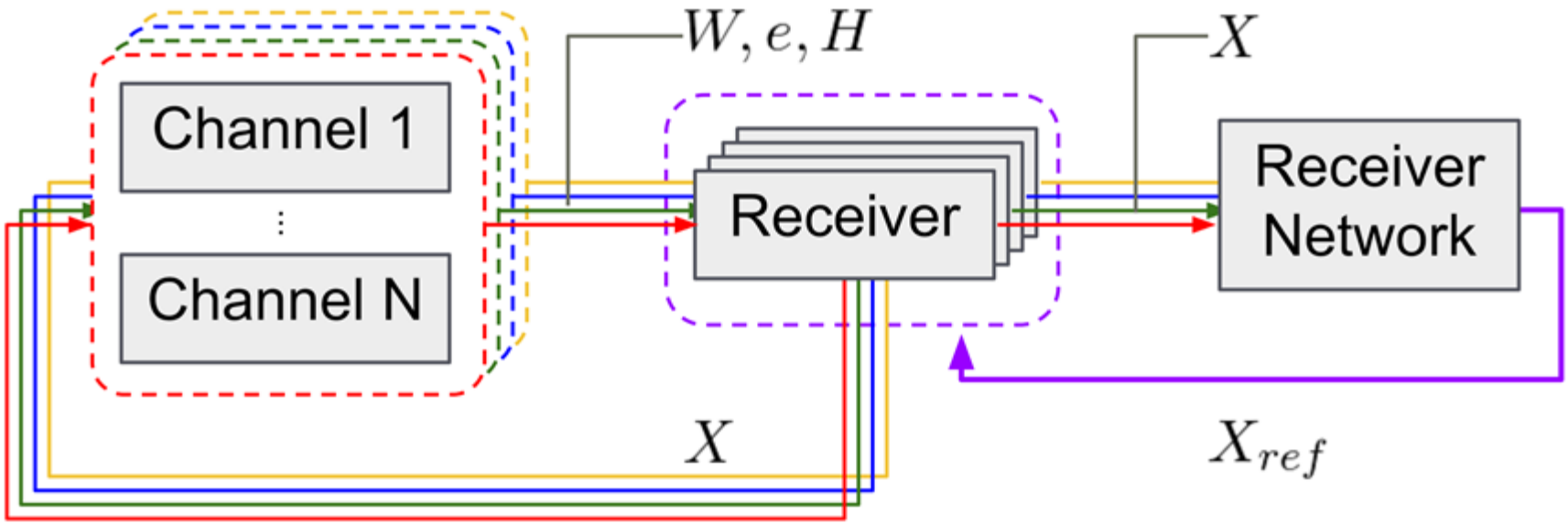


Single-Receiver Vector Tracking (SRVT)

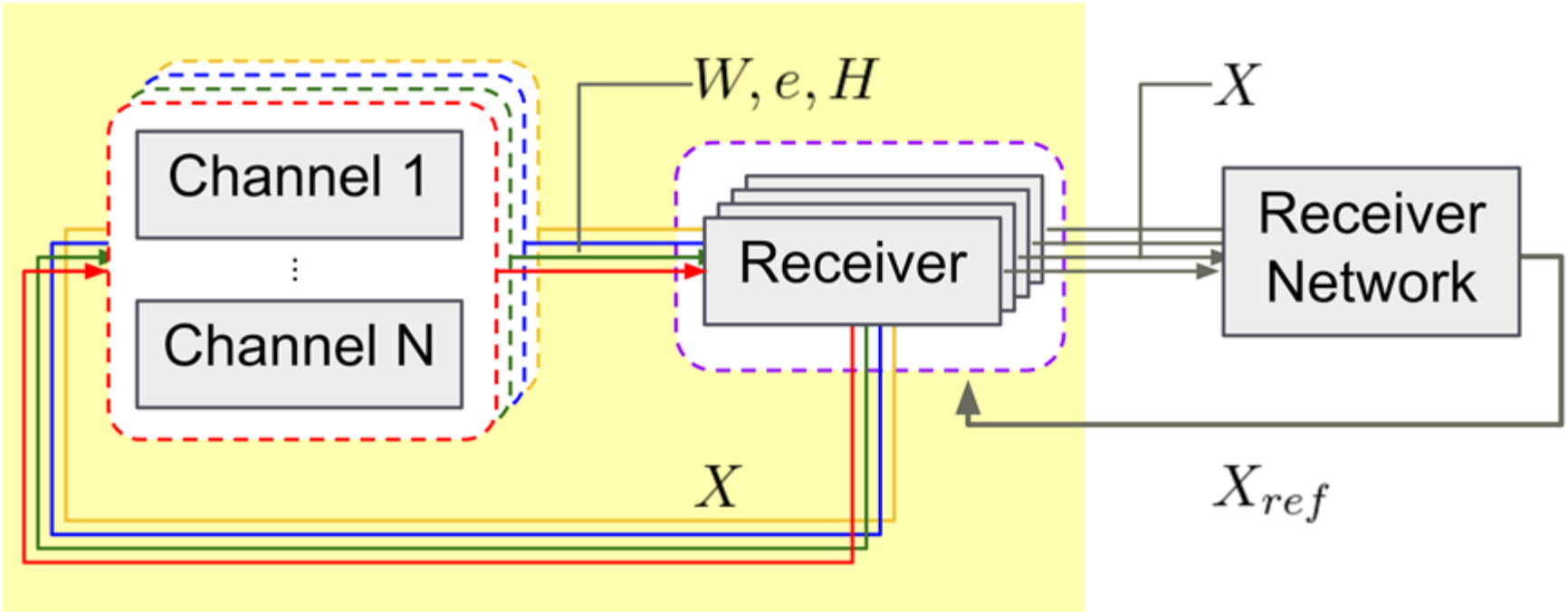


Multi-Receiver Vector Tracking (MRVT)

MRVT Block Diagram

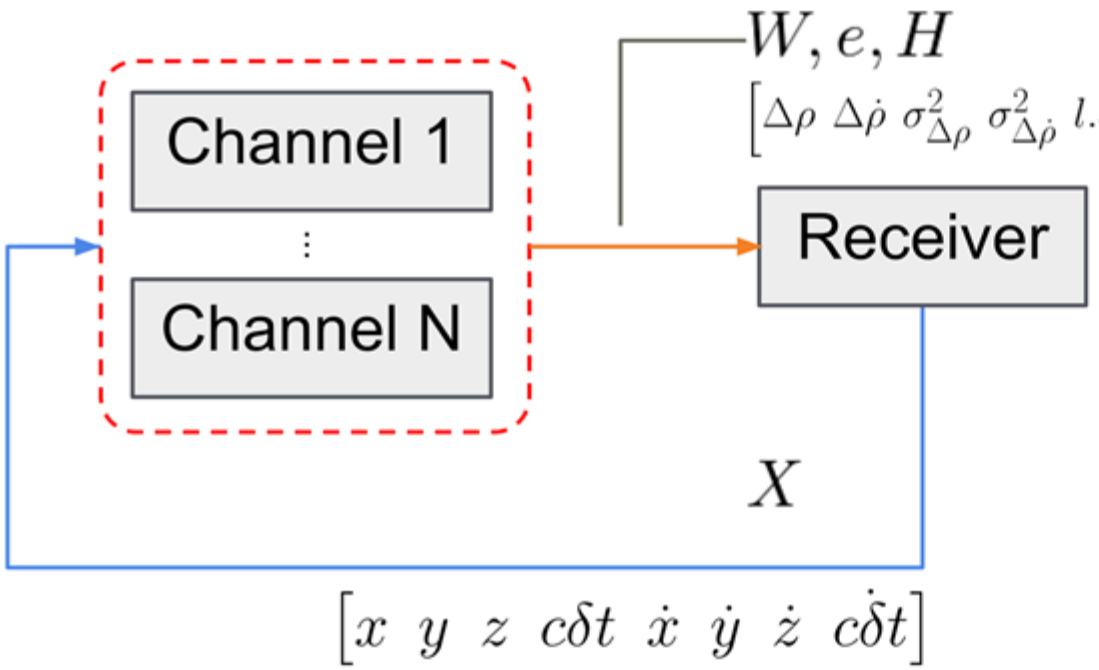


MRVT Block Diagram



SRVT Loops

Single Receiver Vector Tracking Loop



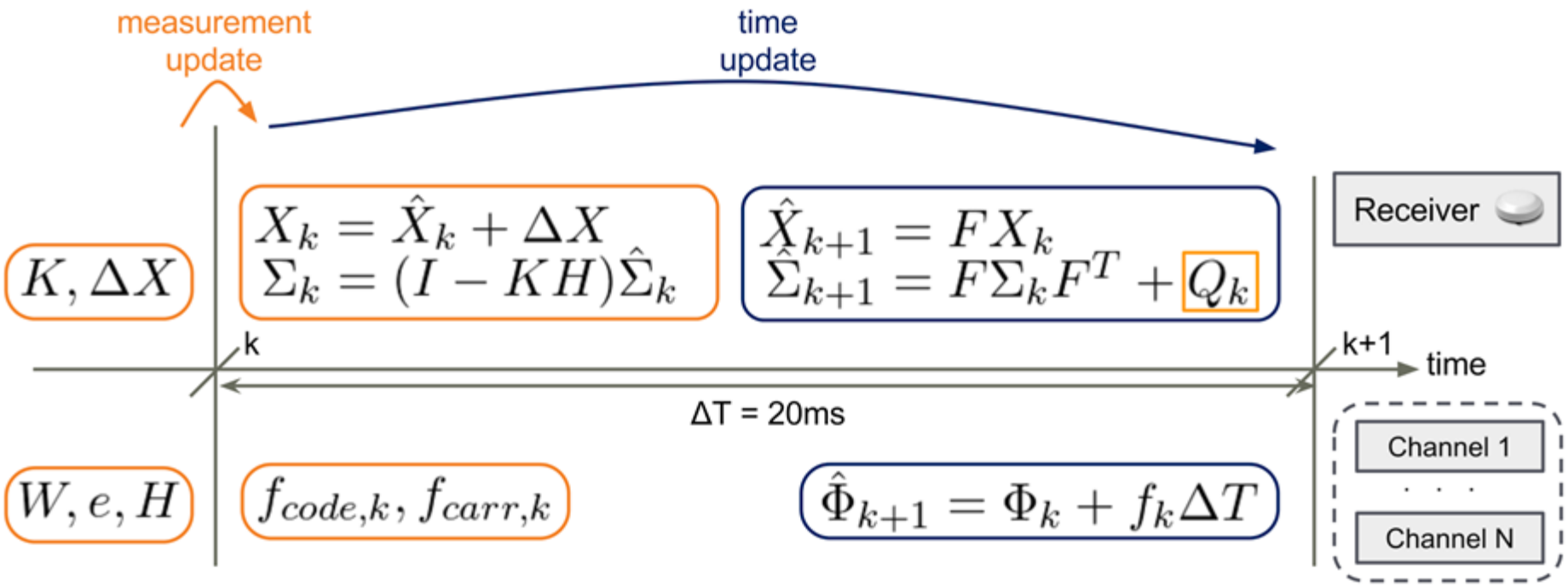
EKF-based Filter

2-Step Update

- Measurement
- Time

Vector Delay Lock Loop - James J. Spilker, Jr.
Global Positioning System-Theory and Applications, Volumes I, II. AIAA, Washington, DC., 1996.

Single Receiver Vector Tracking

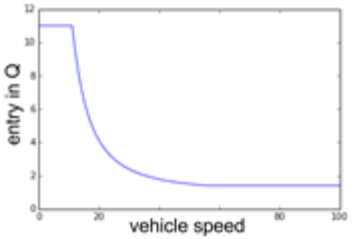


State Representation

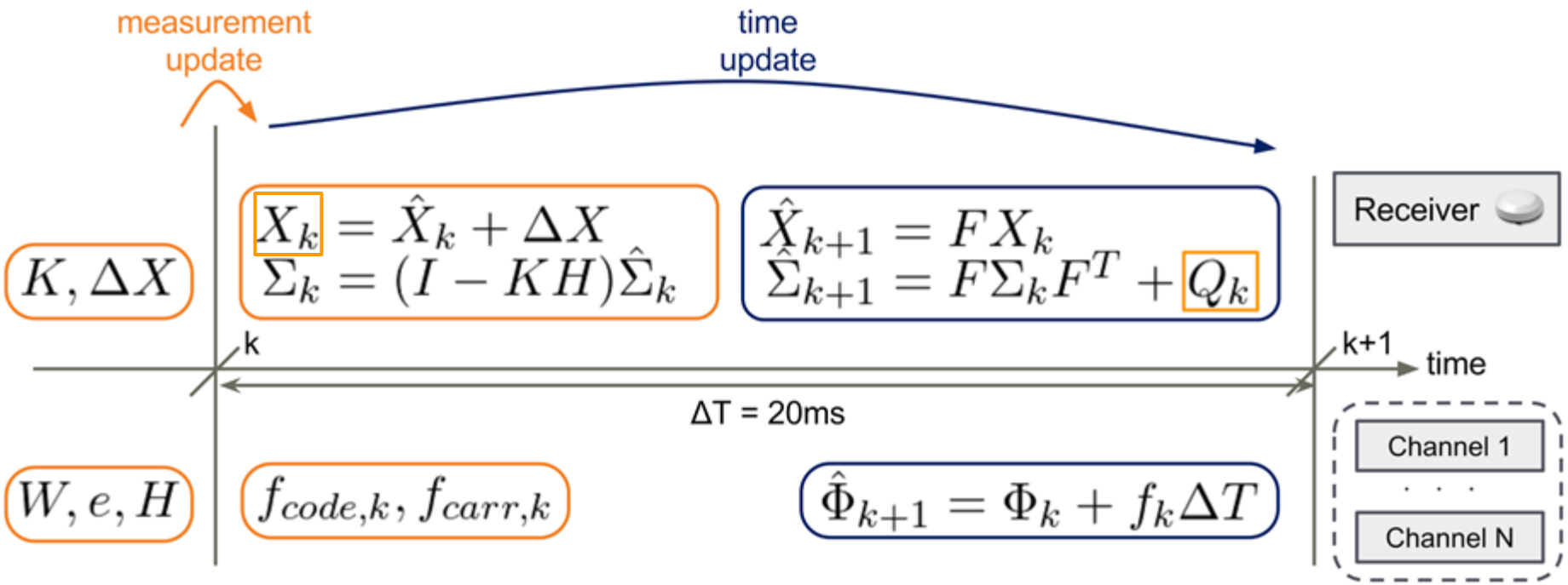
$$X = [x \;\; y \;\; z \;\; c\delta t \;\; \dot{x} \;\; \dot{y} \;\; \dot{z} \;\; c\dot{\delta t}]$$

Process Noise Cov.

$$Q =$$



Single Receiver PIA Vector Tracking



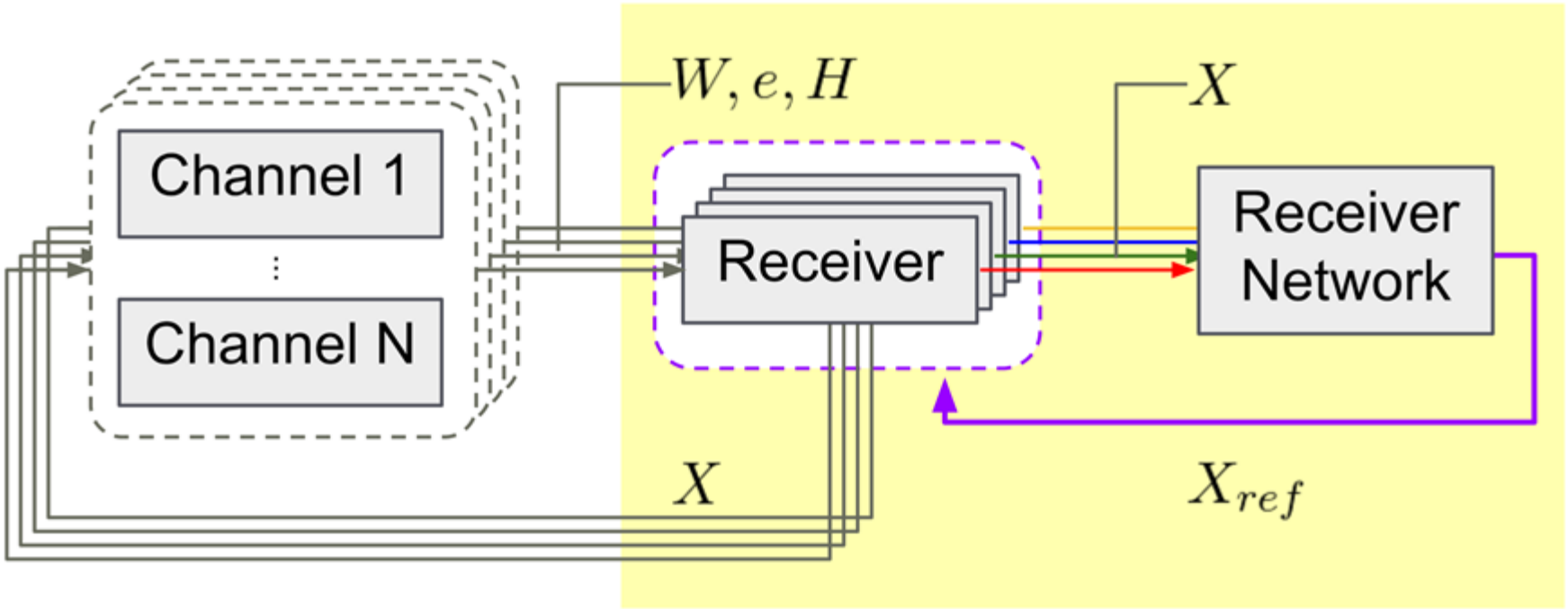
State Representation

$$X = \begin{bmatrix} c\delta t_u \\ c\dot{\delta} t_u \end{bmatrix}$$

Process Noise Covariance Matrix

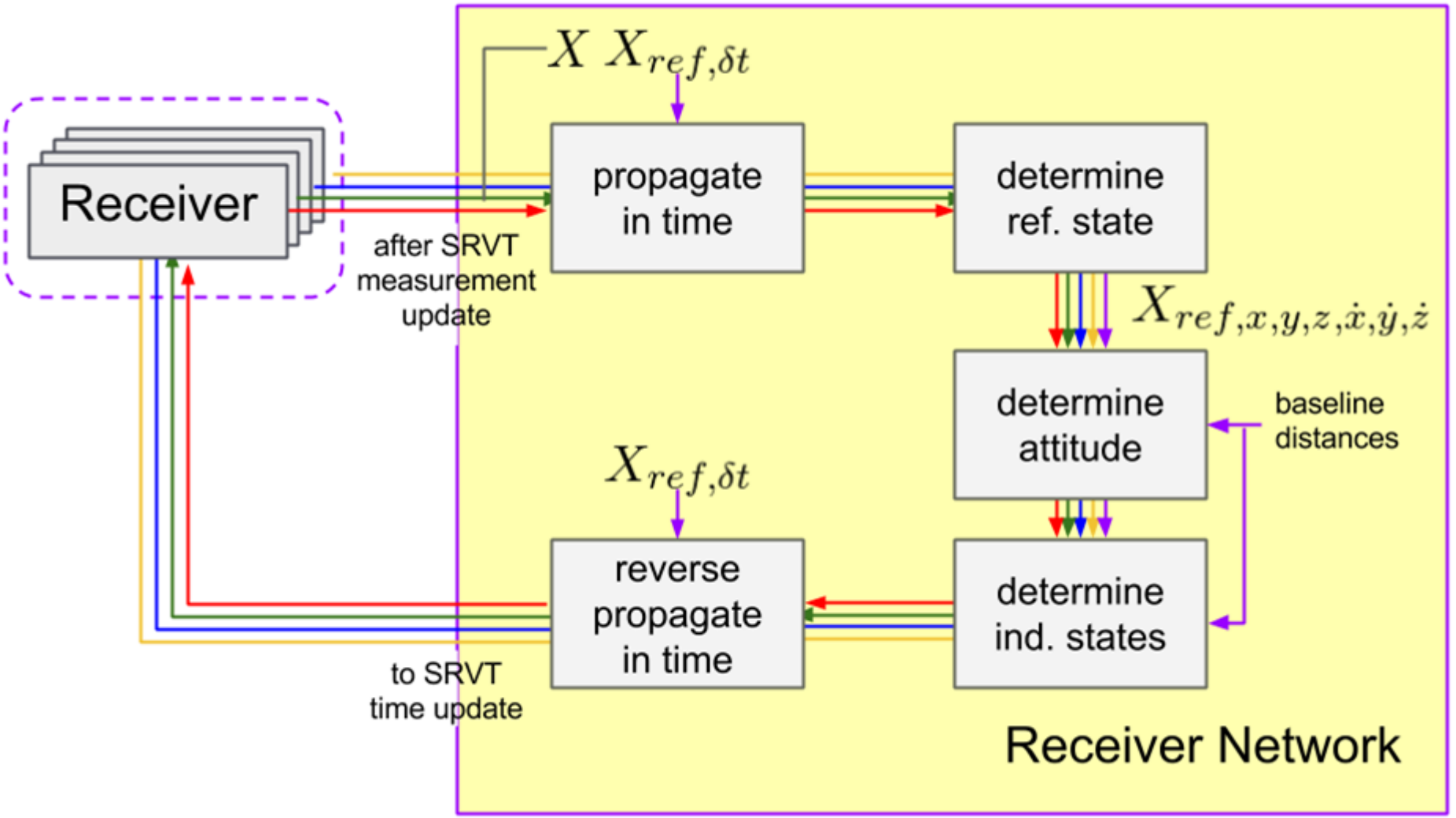
$$Q = \begin{bmatrix} \sigma^2_{c\delta t_u} & 0 \\ 0 & \sigma^2_{c\dot{\delta} t_u} \end{bmatrix}$$

MRVT Block Diagram



MRVT Loop

MRVT Loop


X $X_{ref,\delta t}$
Receiver
after SRVT measurement update
propagate in time
determine ref. state
$X_{ref,x,y,z,\dot{x},\dot{y},\dot{z}}$
determine attitude
baseline distances
$X_{ref,\delta t}$
reverse propagate in time
determine ind. states
to SRVT time update
Receiver Network

MRVT Variations

MRVT

$$X = \begin{bmatrix} x & y & z & c\delta t \\ \dot{x} & \dot{y} & \dot{z} & c\dot{\delta} t \\ \psi & \theta & \phi & \dot{\psi} & \dot{\theta} & \dot{\phi} \end{bmatrix}$$

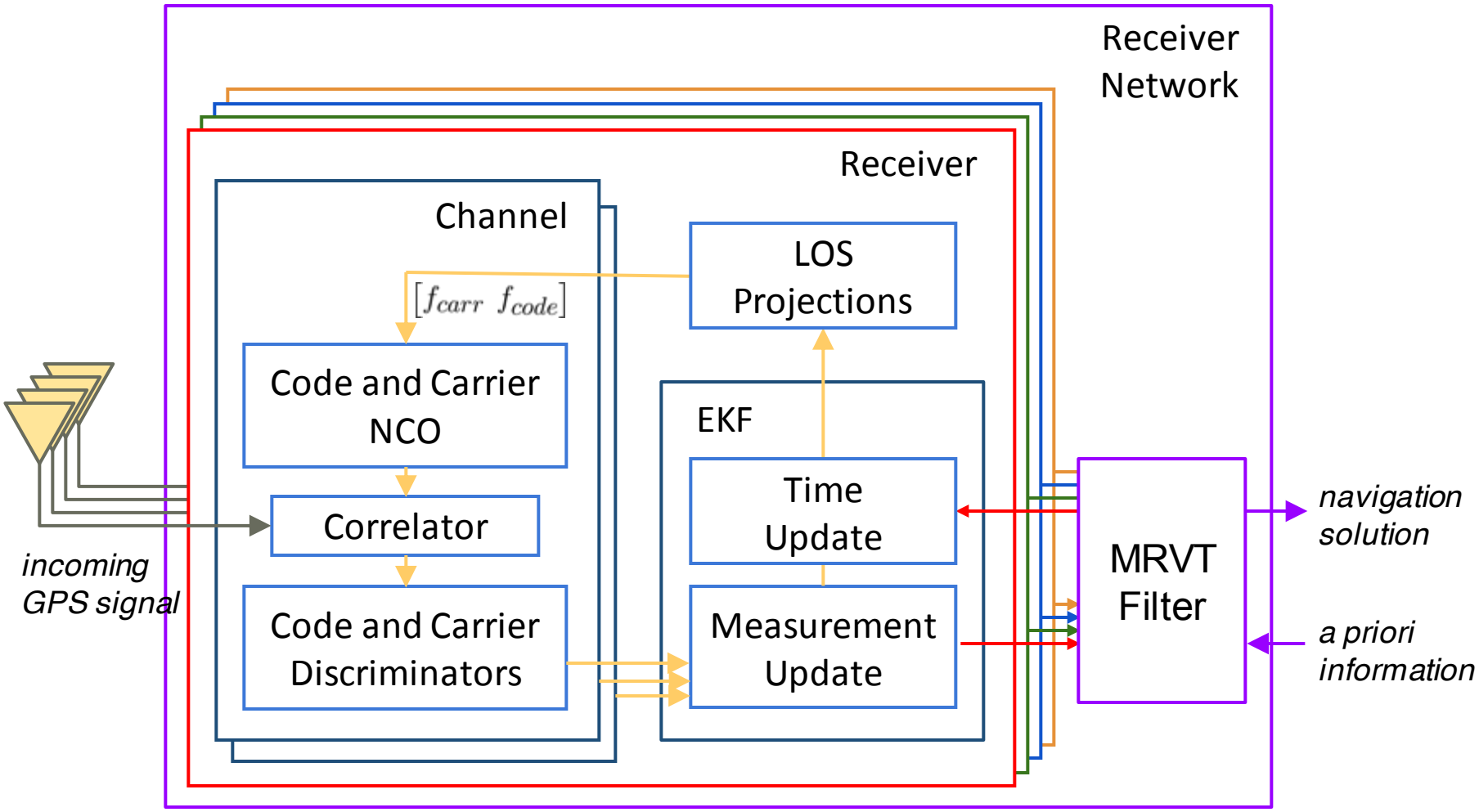
Enforce baseline, attitude and velocity constraints
Enforce timing constraints

MRPIAVT

$$X = \begin{bmatrix} c\delta t_u \\ c\dot{\delta} t_u \end{bmatrix}$$

- Enforce timing constraints
  - →clock bias differ by constant
  - →clock drift same

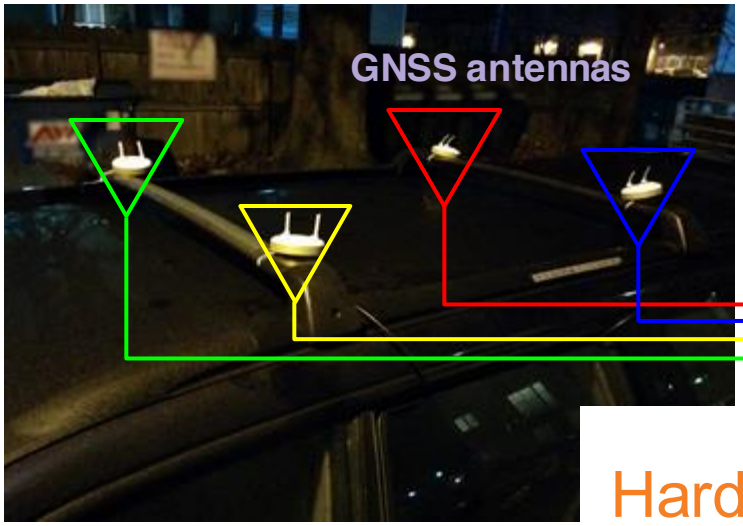
Multi-Receiver Vector Tracking

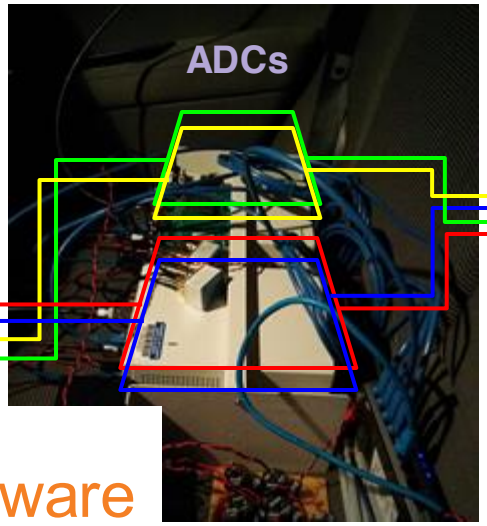


Outline

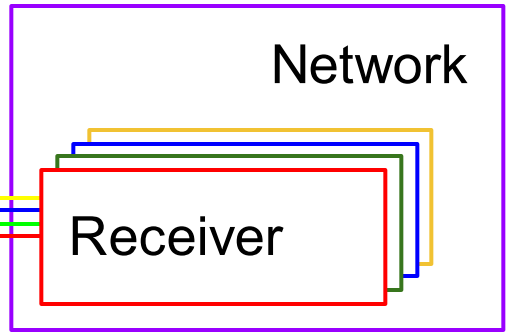
- Motivation | Coop Nav, Joint Signal Tracking
- Our Approach | Multi-Receiver Vector Tracking
- Implementation | Hardware, Python SDR
- Experimental Validation
- Summary

Overall Implementation

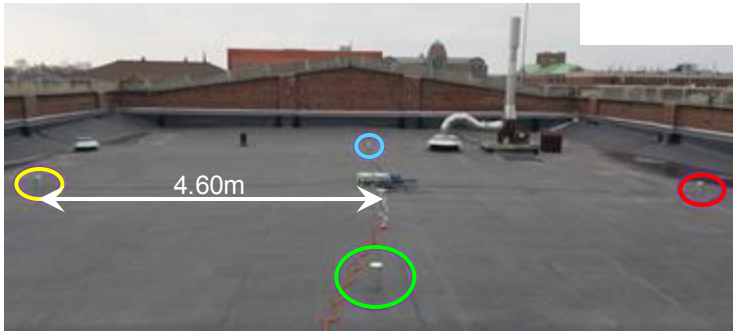


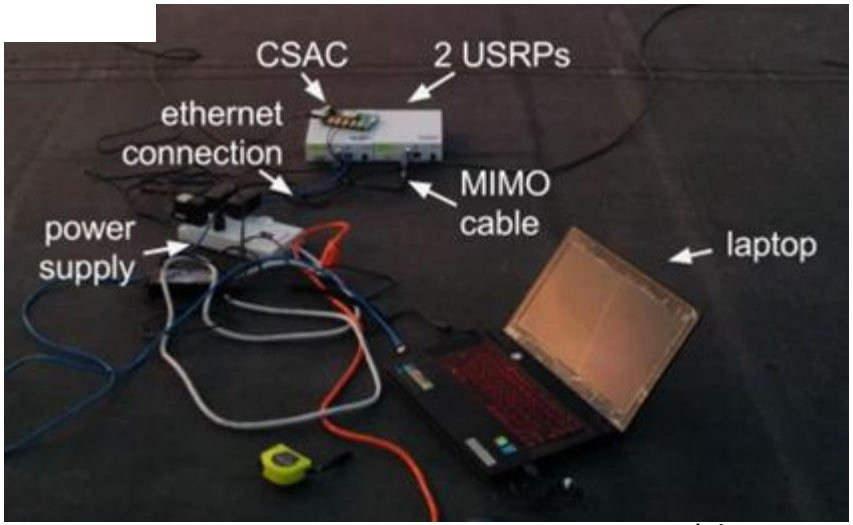


Hardware



Software
(Python SDR)



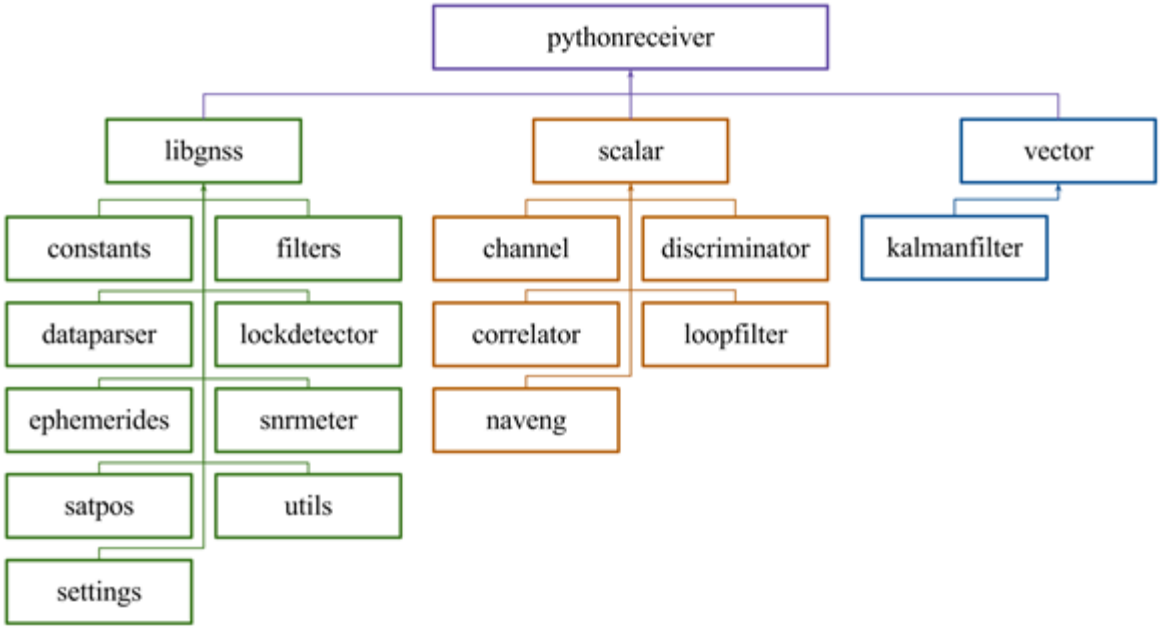


Python SDR : Research Platform

Channel

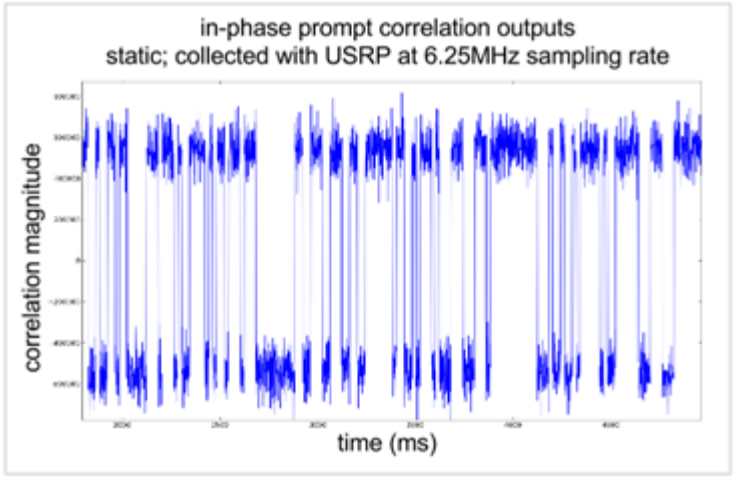
Attributes
- PRN
- Settings
- Correlator
- DLL_Discriminator
- F/PLL_Discriminator
- DLL_LoopFilter
- F/PLL_LoopFilter
- code phase
- code frequency
- carrier phase
- carrier frequency
- ...

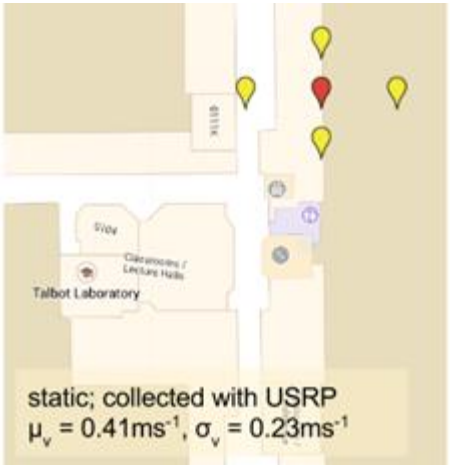
Functions
- __init__()
- cold_start()
- set_integration_period()
- scalar_track()
- update()
- ...

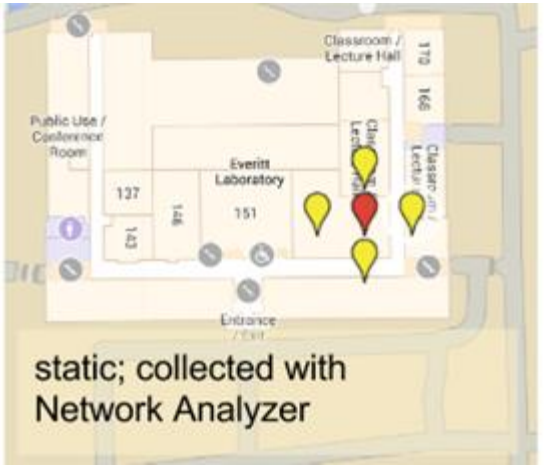


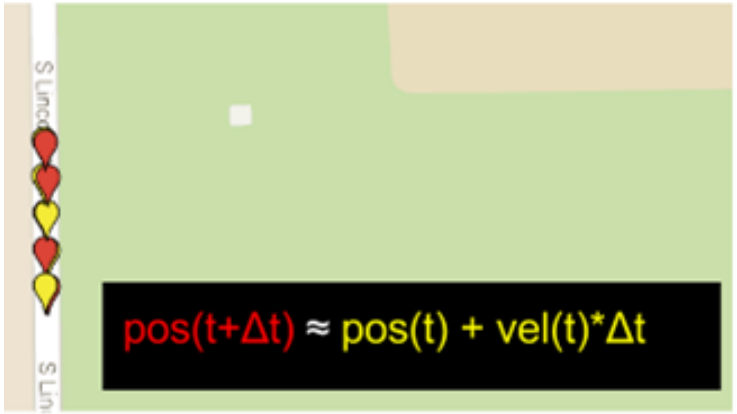
- Share information effectively
- SDR (Software Defined Radio)
- OOP (Object-Oriented Programming)

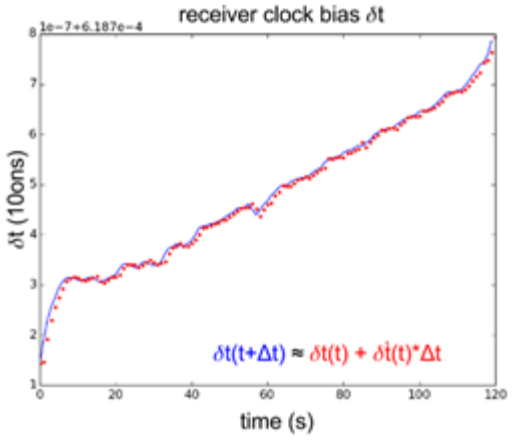
Python SDR : Research Platform



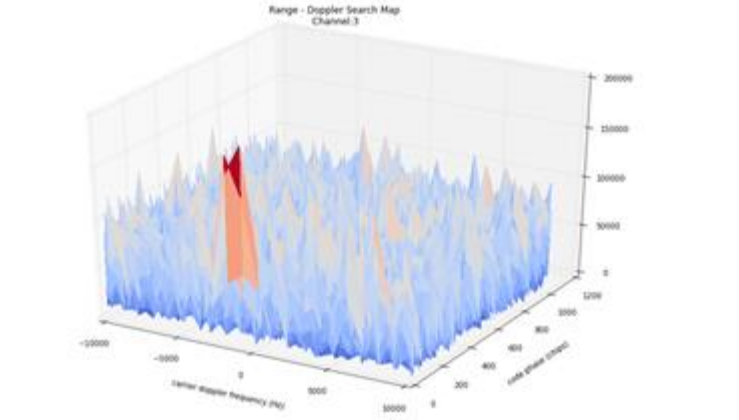








Python SDR : Teaching Platform




```
Doppler_freq_coarse = -1000.000000
code_phase = 27.109500
correlation peak value = 194635.925385
```

The plot is shown above. There is one peak in the plot. The coarse doppler frequency estimation and the code phase corresponding to the peak is listed above, and they match what were obtained in the previous part of the lab.

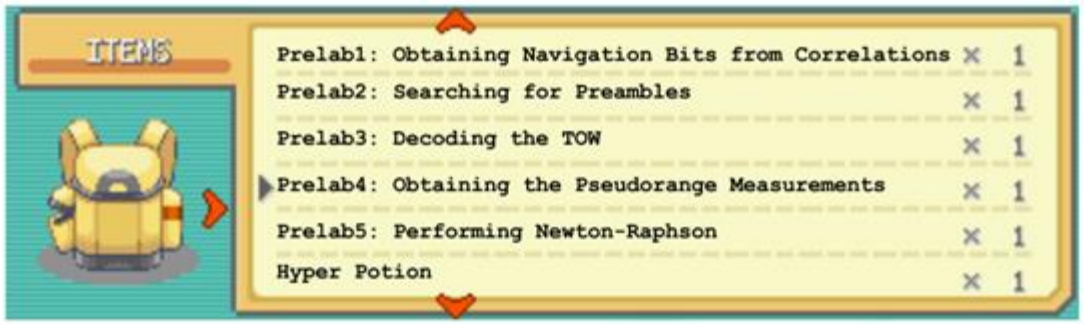
PL2

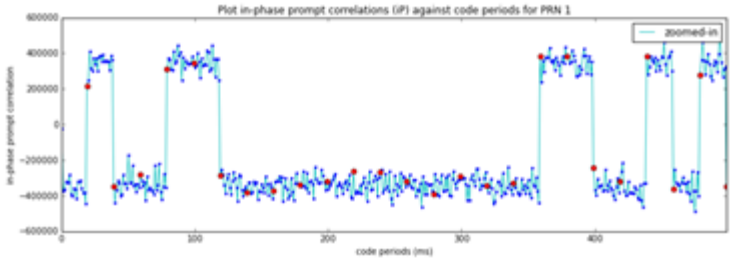
Using what you have learned so far, fill in the table given to you in Q1.

PRN	Code phase lag (chips)	Carrier doppler frequency (Hz)	Correlation Height
3	27.110	-1472.47	194635.93
8	72.123	-1946.88	442370.26
9	278.25	3295.90	221513.77
16	958.55	1666.09	447372.49
23	49.61	2480.71	326821.64
26	169.818	-556.09	438458.43
31	470.069	-2442.96	191263.48
32	805.613	-3288.27	199413.52

```
PRN = 3
Doppler_freq_coarse = -1000.000000
code_phase = 26.086500
correlation peak value = 228704.509143
('medium doppler frequency', -1400.0)
fine doppler frequency = -1517.844179

PRN = 8
Doppler_freq_coarse = -2000.000000
code_phase = 71.098500
correlation peak value = 480295.970509
('medium doppler frequency', -2000.0)
fine doppler frequency = -2081.718039

PRN = 16
Doppler_freq_coarse = 2000.000000
code_phase = 959.574000
correlation peak value = 357613.708767
('medium doppler frequency', 1600.0)
fine doppler frequency = 1759.533461

PRN = 23
Doppler_freq_coarse = 3000.000000
code_phase = 51.150000
correlation peak value = 322131.654385
('medium doppler frequency', 2600.0)
fine doppler frequency = 2502.321317

PRN = 26
Doppler_freq_coarse = -1000.000000
code_phase = 169.306500
correlation peak value = 367505.911670
('medium doppler frequency', -600.0)
fine doppler frequency = -556.684419

PRN = 31
Doppler_freq_coarse = -2000.000000
code_phase = 469.045500
correlation peak value = 193177.080364
('medium doppler frequency', -2400.0)
fine doppler frequency = -2603.364069

PRN = 32
Doppler_freq_coarse = -3000.000000
code_phase = 804.078000
correlation peak value = 213800.183177
('medium doppler frequency', -3200.0)
fine doppler frequency = -3250.842694
```

Python SDR : Teaching Platform





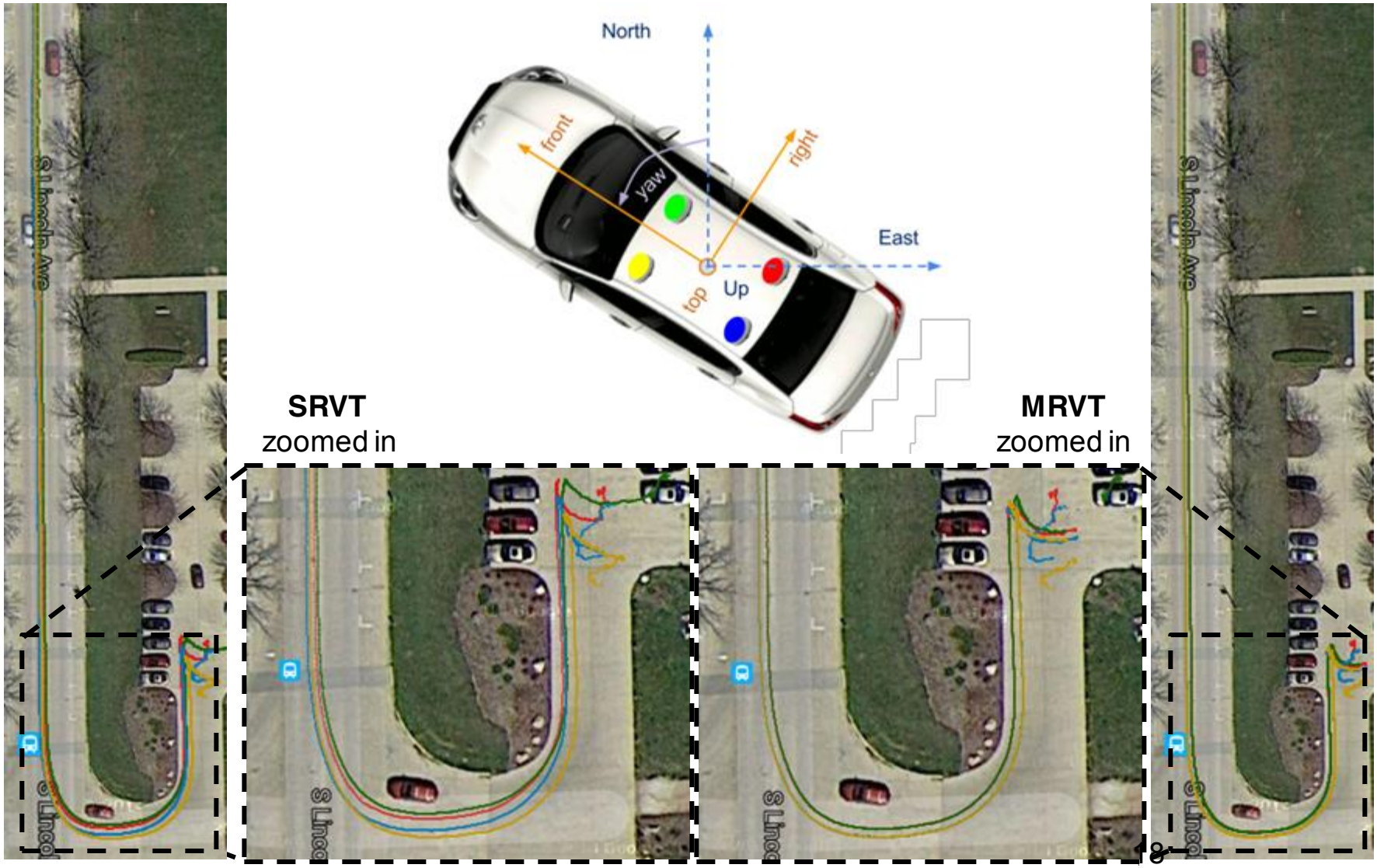
('First 25 Navbits of PRN 1', array([1., -1., -1., 1., 1., -1., -1., -1., -1., -1., -1., -1., -1.,
       -1., -1., -1., -1., 1., 1., -1., -1., 1., -1., 1., -1.]))

```
*** Ephemerides ***
IODE: 79, IODC: 79
weeknumber: 1100110011
accuracy: 0
health: 0
T_GD: 5.58793544769e-09
t_oc: 352800
a_f2: 0.0
a_f1: 4.54747350886e-13
a_f0: -5.89992851019e-06
C_rs: 19.71875
delta_n: 4.90913305653e-09
M_0: 1.8478906143
C_uc: 1.13062560558e-06
e: 0.00416372809559
C_us: 6.8936496973e-06
sqrt_A: 5153.66034698
t_oe: 352800
C_ic: -5.58793544769e-08
OMEGA_0: -1.55220992773
C_is: 1.43423676491e-07
i_0: 0.961708135987
C_rc: 247.78125
omega: 0.374407823429
OMEGADOT: -8.15105380982e-09
IDOT: 3.864446684e-10
timestamp: {'cpcount': 3439, 'timeOfWeek': 348960}
```

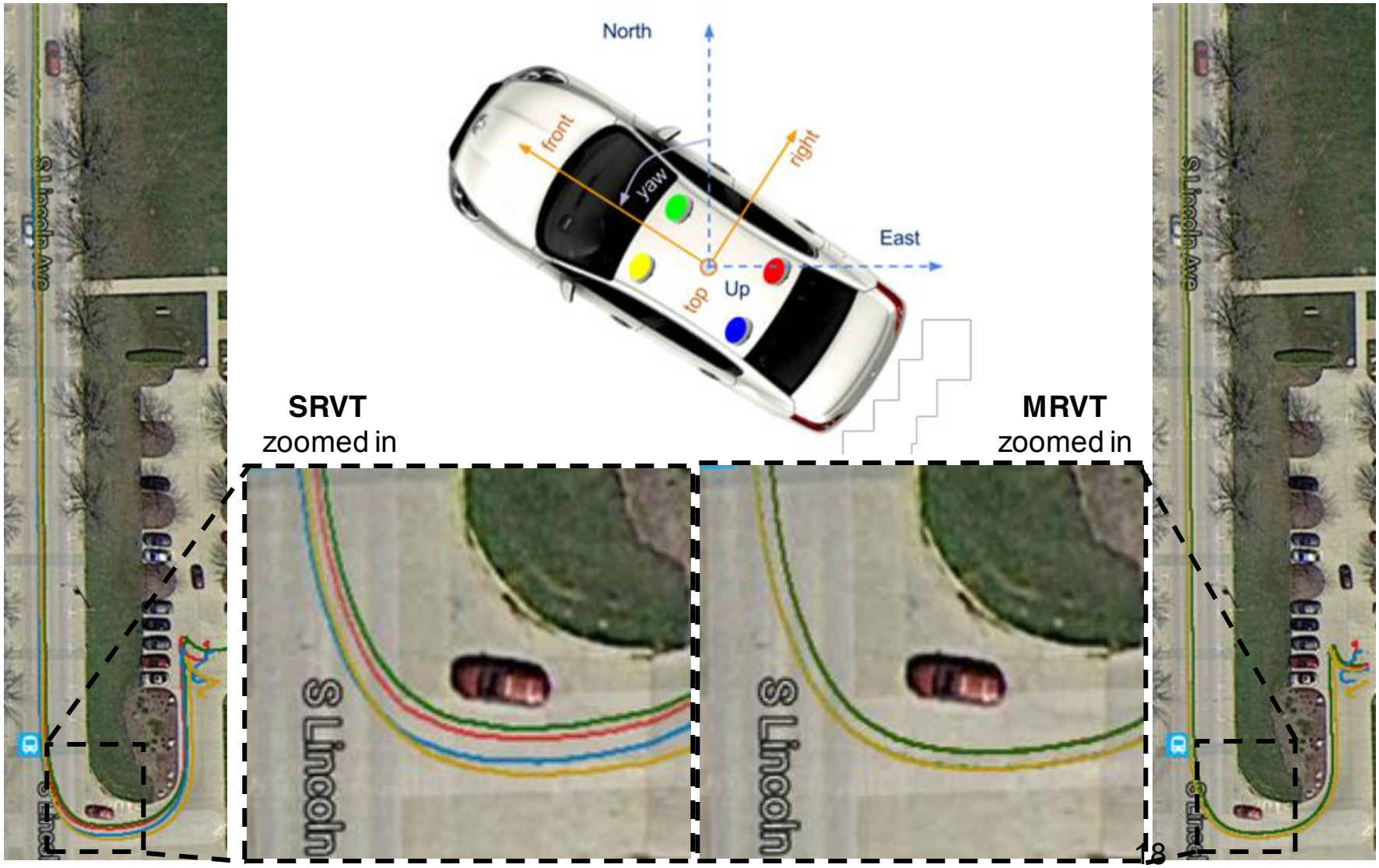
Outline

- Motivation | Coop Nav, Joint Signal Tracking
- Our Approach | Multi-Receiver Vector Tracking
- Implementation | Hardware, Python SDR
- Experimental Validation
- Summary

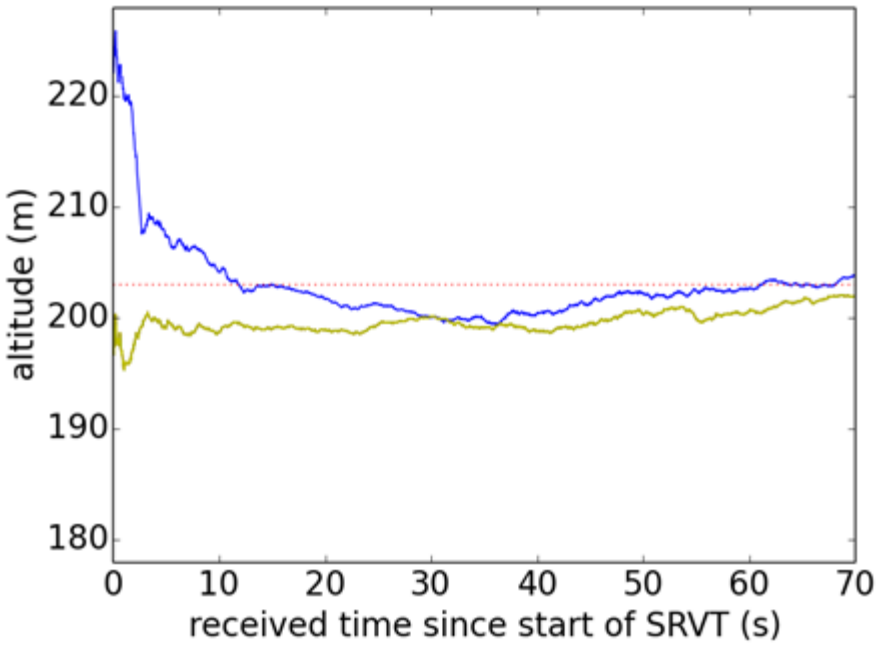
MRVT Positioning Results

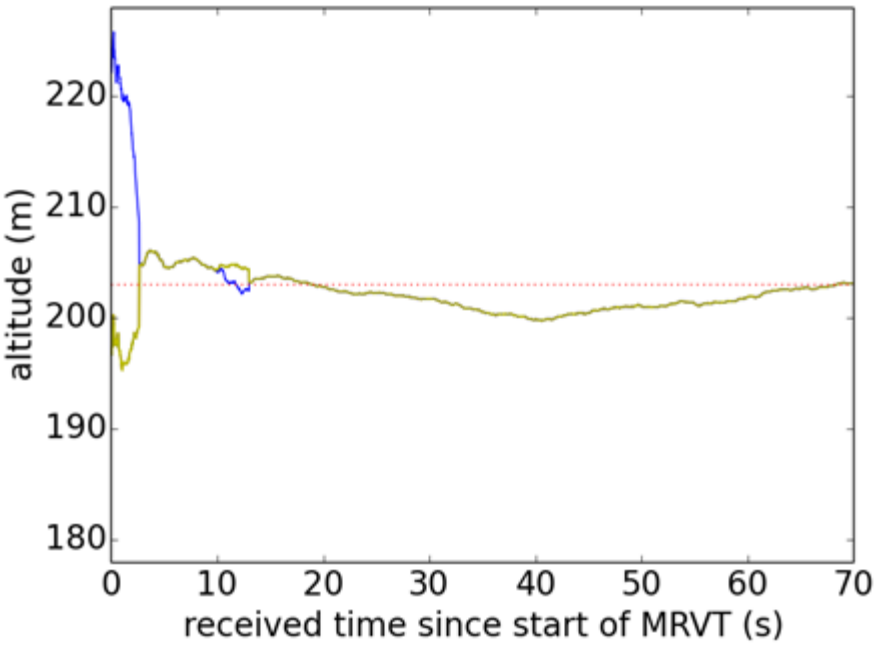


MRVT Positioning Results



MRVT Results: Reliable Altitudes

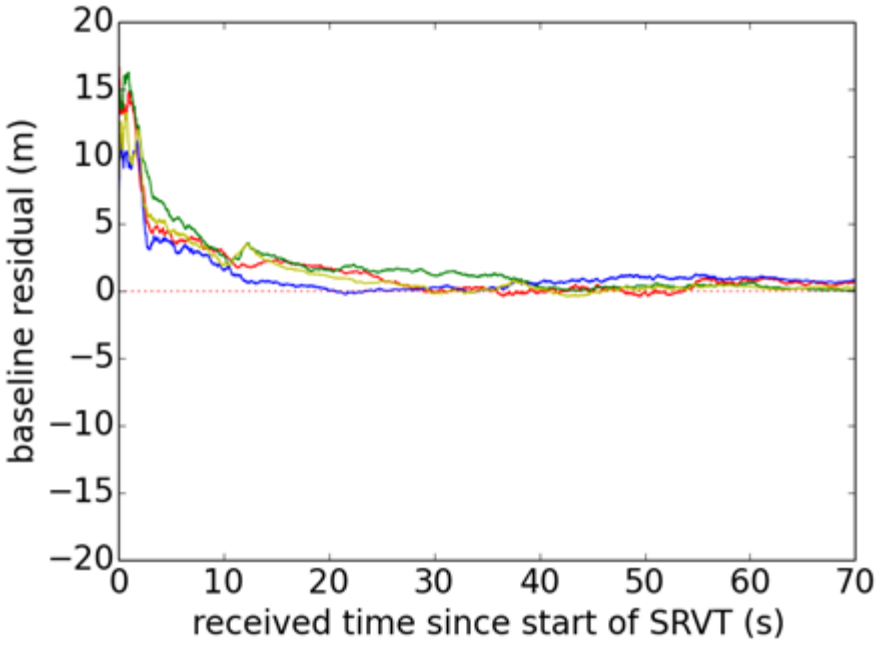


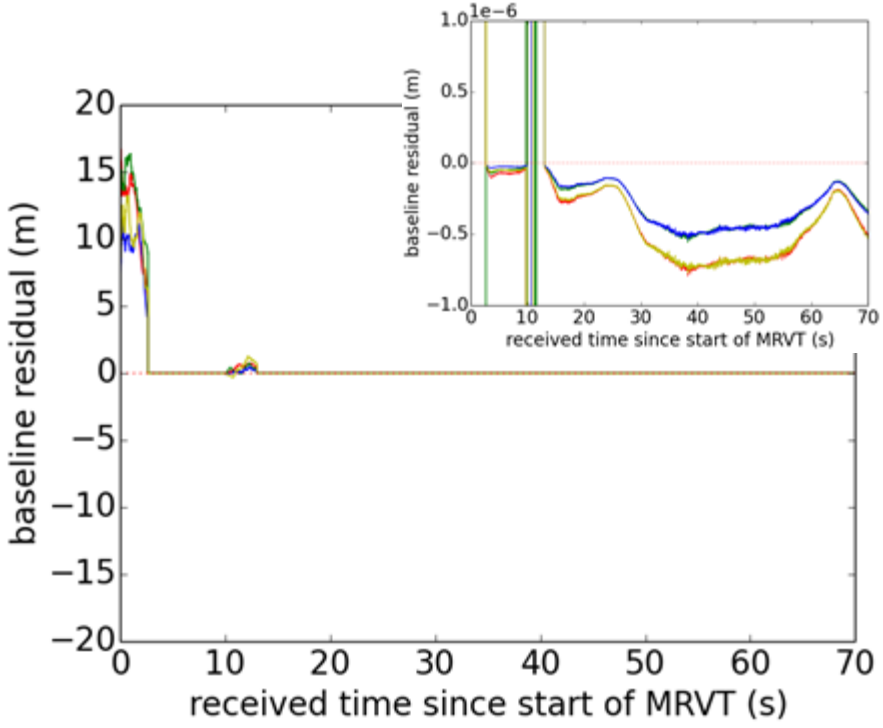


Single-Receiver Vector Tracking (SRVT)

Multi-Receiver Vector Tracking (MRVT)

MRVT Results: Accurate Baselines

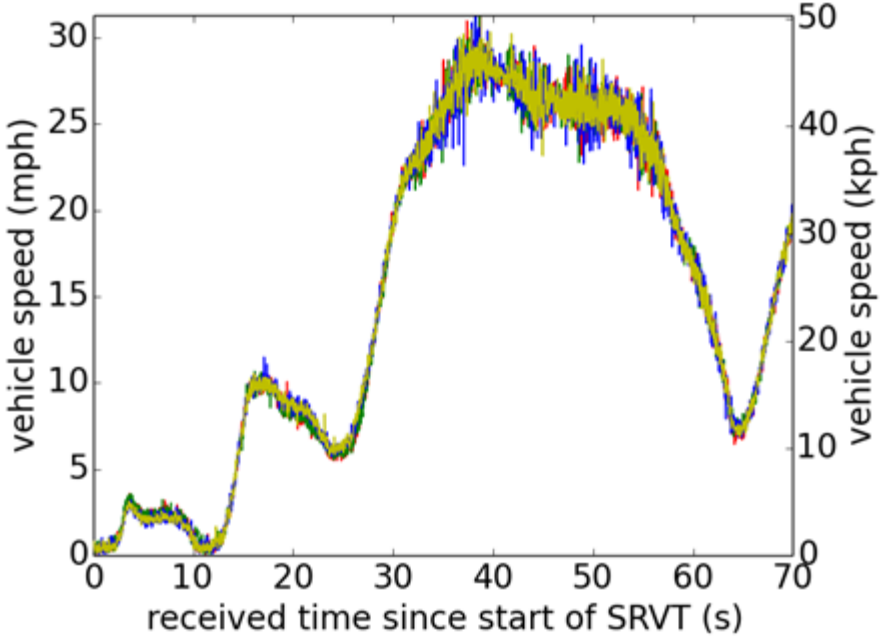




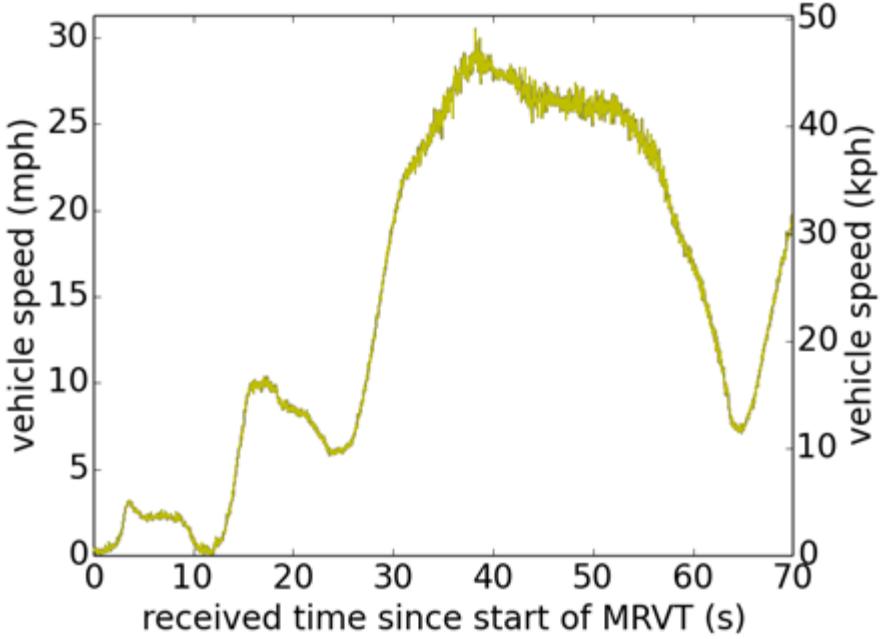
Single-Receiver Vector Tracking (SRVT)

Multi-Receiver Vector Tracking (MRVT)

MRVT Results: Less Noisy Speeds



Single-Receiver Vector Tracking (SRVT)



Multi-Receiver Vector Tracking (MRVT)

Results with Buildings+Accelerations

SRVT

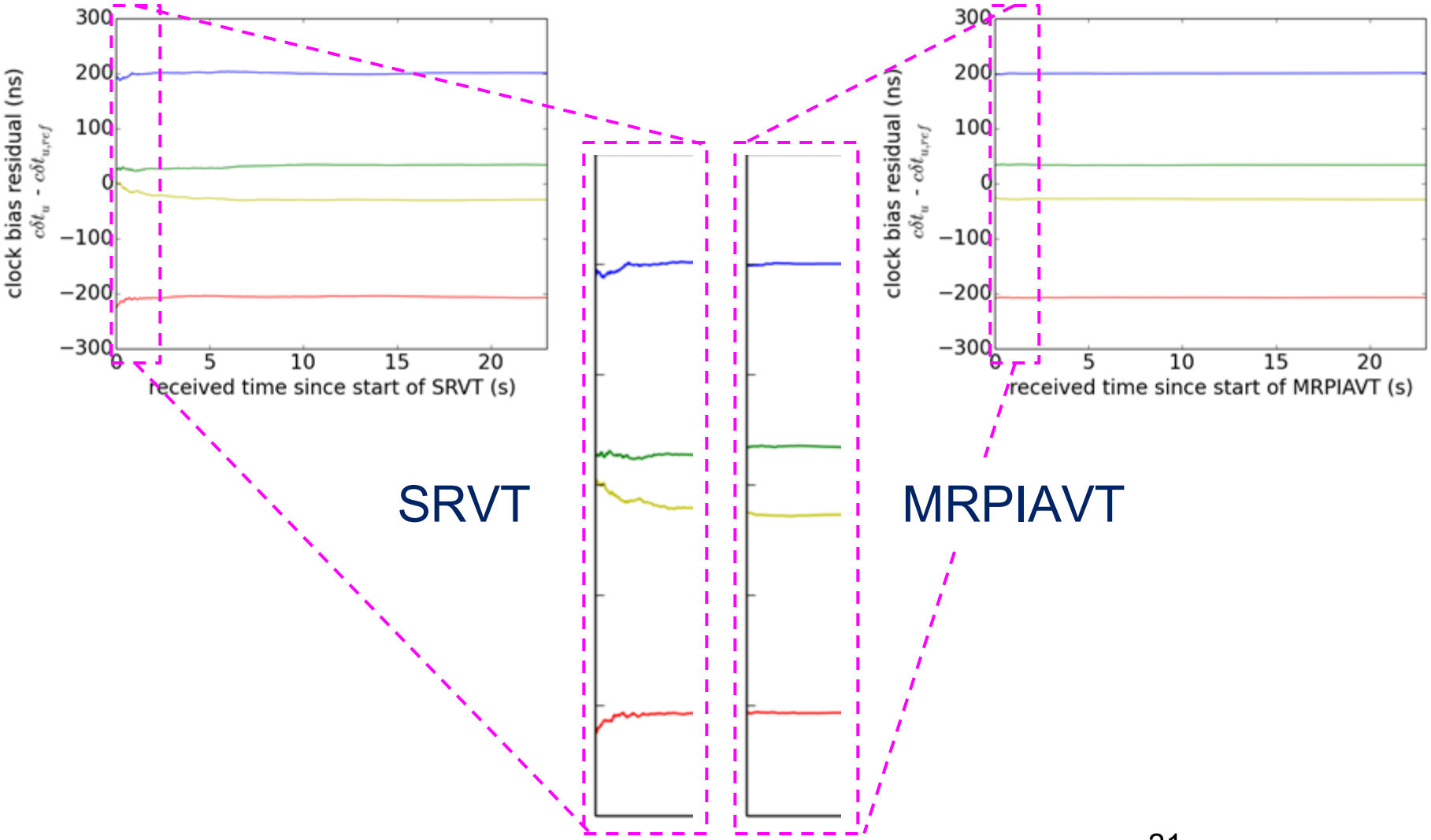
MRVT

Results with Buildings+Accelerations

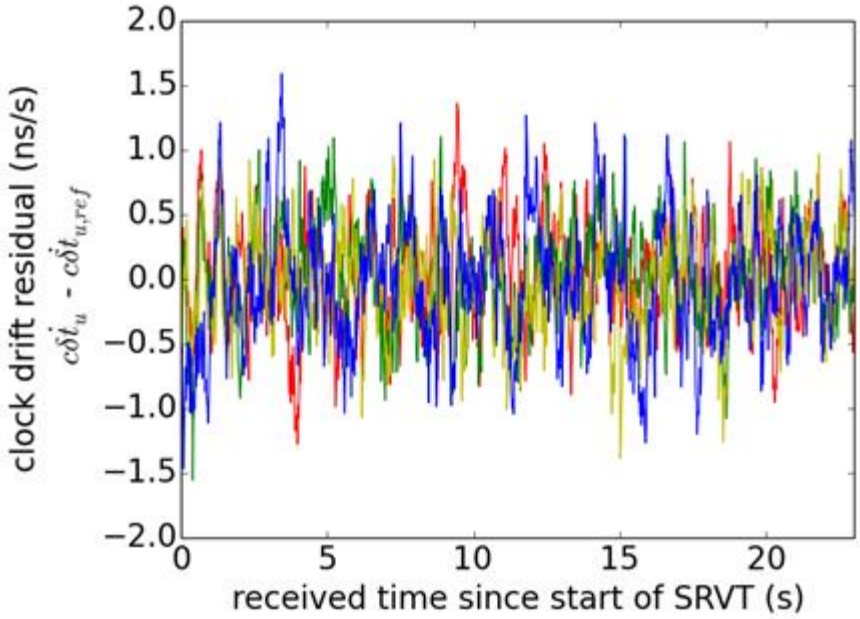
SRVT
zoomed in

MRVT
zoomed in

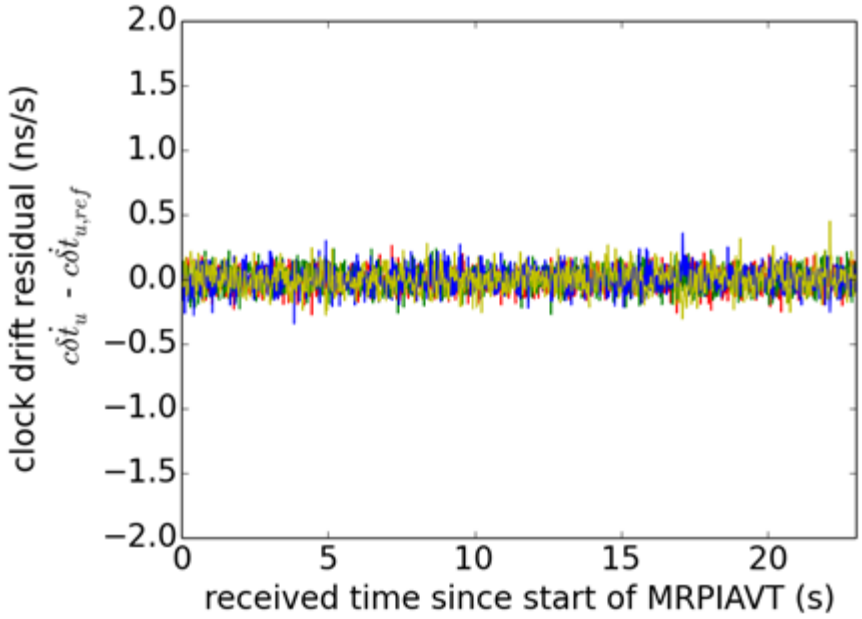
MRPIAVT Results: Improved Timing Accuracy



MRPIAVT Results: Improved Timing Accuracy

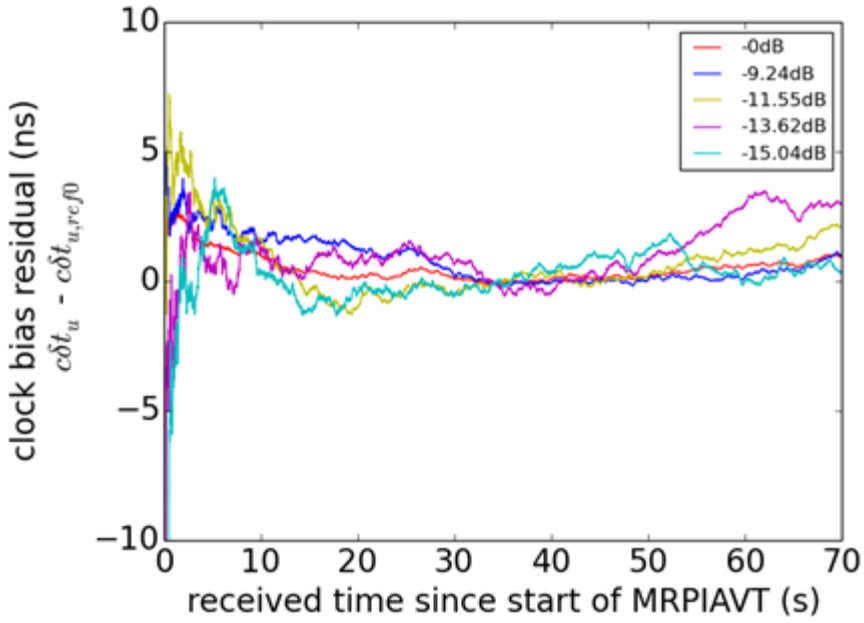


SRVT

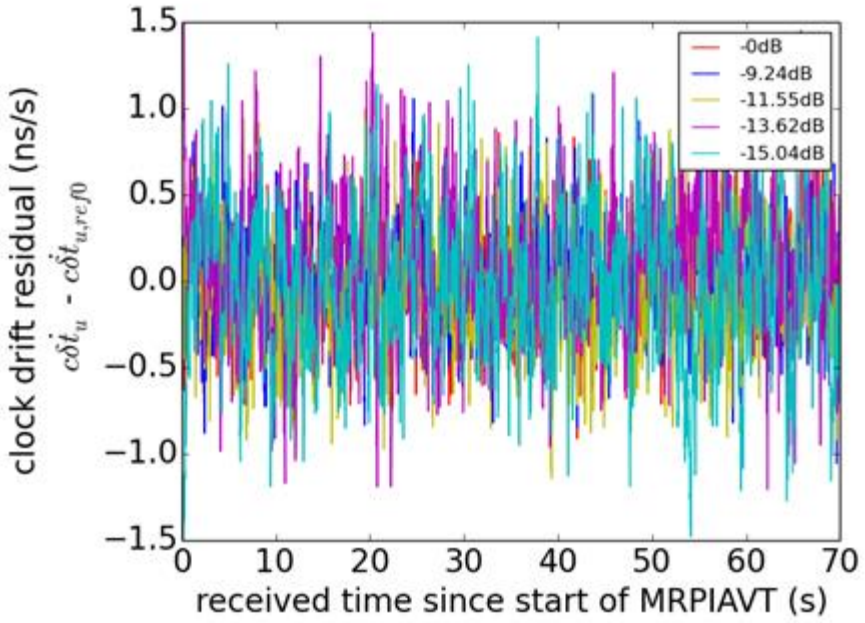


MRPIAVT

MRPIAVT: Resistant to Jamming Added noise of different levels



Clock bias comparison
across different noise power

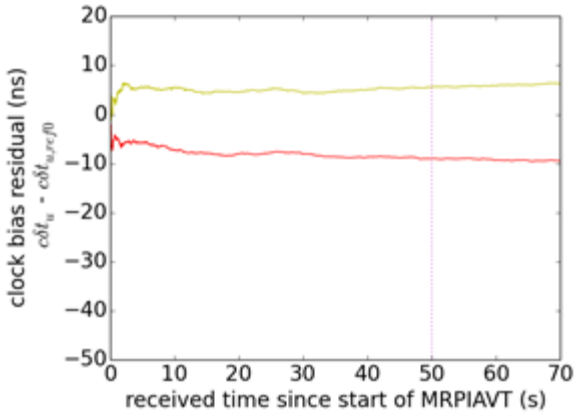


Clock drift comparison
across different noise power

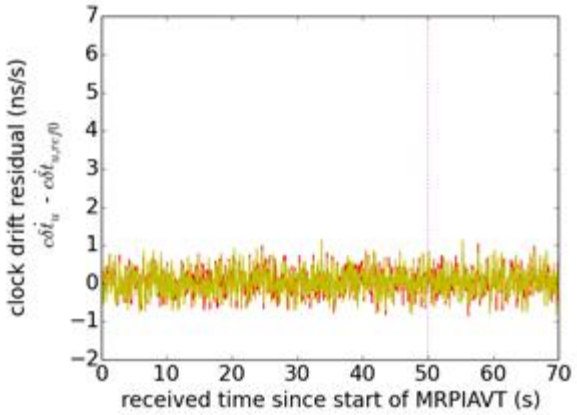
MRPIAVT: Robust to Meaconing

weak meaconing (1:1 power ratio)

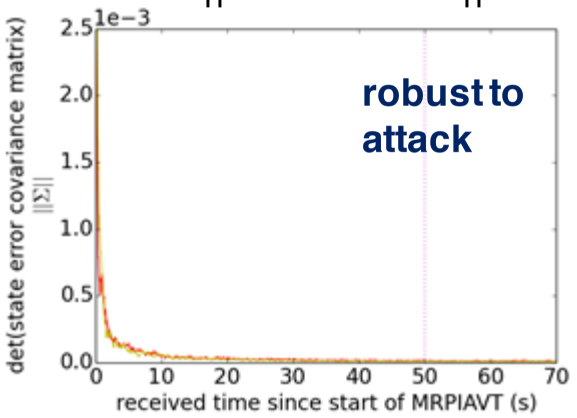
clock bias



clock drift

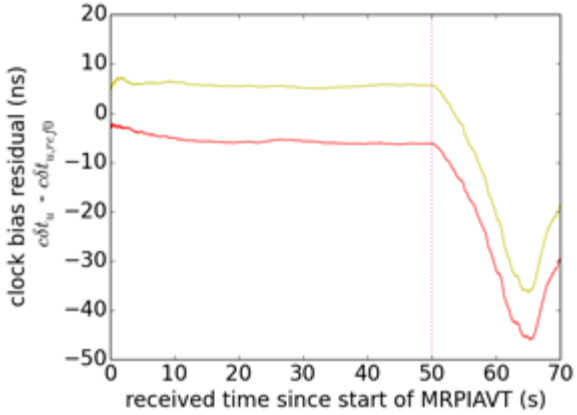


||state error cov||

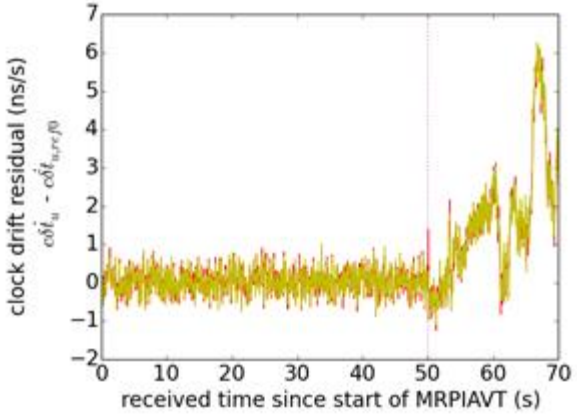


strong meaconing (100:1 power ratio)

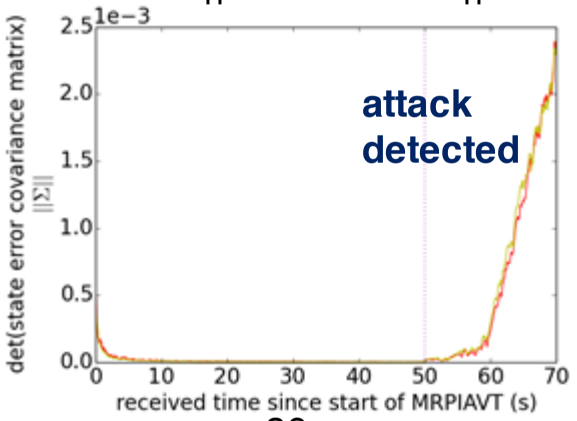
clock bias



clock drift

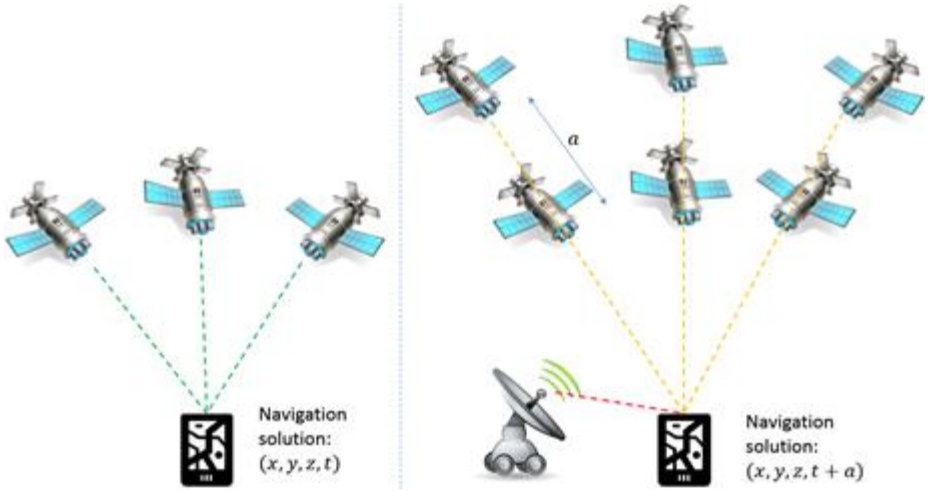


||state error cov||



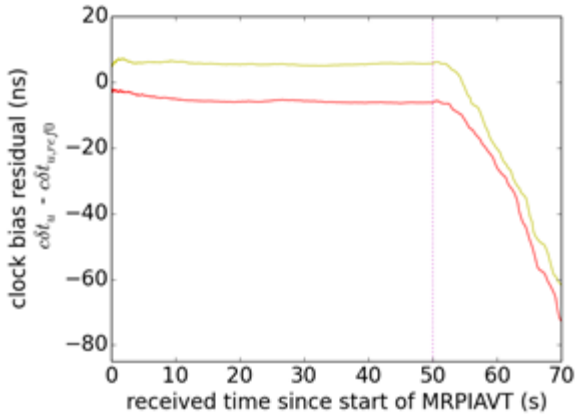
MRPIAVT: Detects Data Spoofing

Attacked af_0 in Subframe 1 of Ephemerides → ephemerides

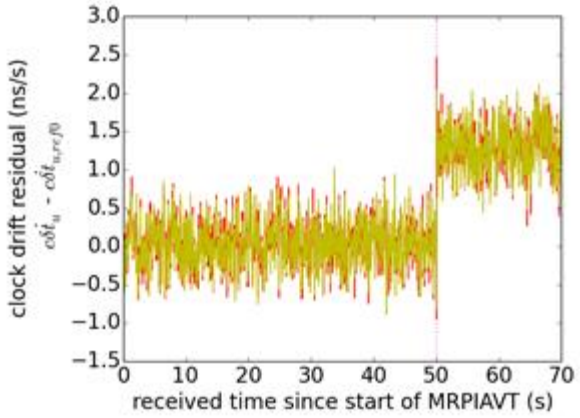


satellite clock biases shifted by 4s

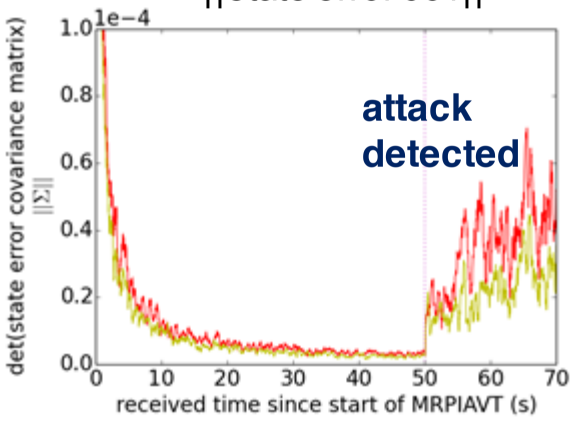
clock bias



clock drift



||state error cov||



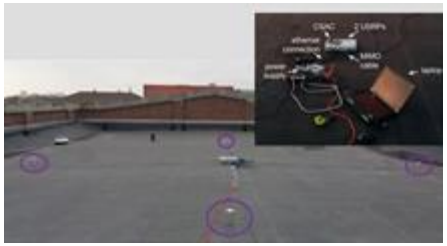
Outline

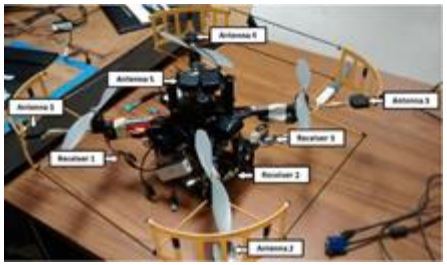
- Motivation | Coop Nav, Joint Signal Tracking
- Our Approach | Multi-Receiver Vector Tracking
- Implementation | Hardware, Python SDR
- Experimental Validation
- Summary

Summary

- Proposed Multi-Receiver Vector Tracking (MRVT)
  - Cooperative navigation
  - Joint signal tracking
- Designed MRVT Architecture
  Implemented in Python SDR
- Conducted MRVT Experiments
- Validated Performance Improvements
  - MRVT : Demonstrated increased accuracy in horizontal heading, altitudes, baselines, speeds, timing
  - MRPIAVT : Demonstrated improved timing accuracy and precision, robustness against timing attacks

Thank you!





From all of us @UIUC