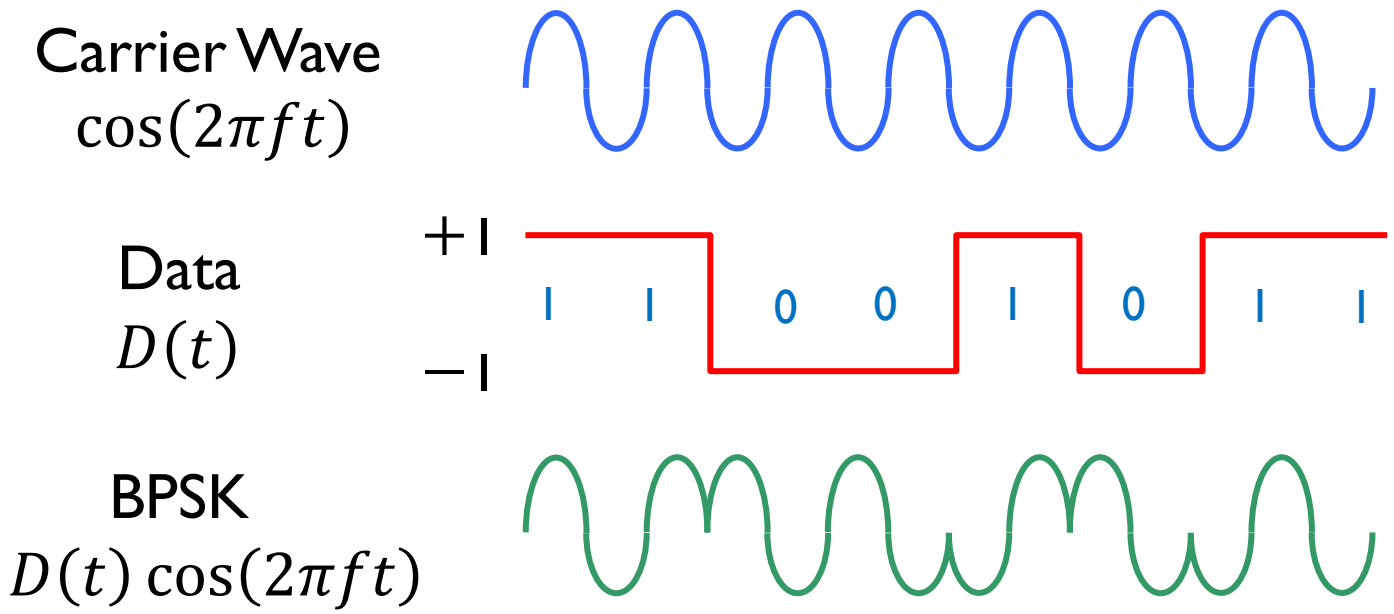
06: GNSS Signals

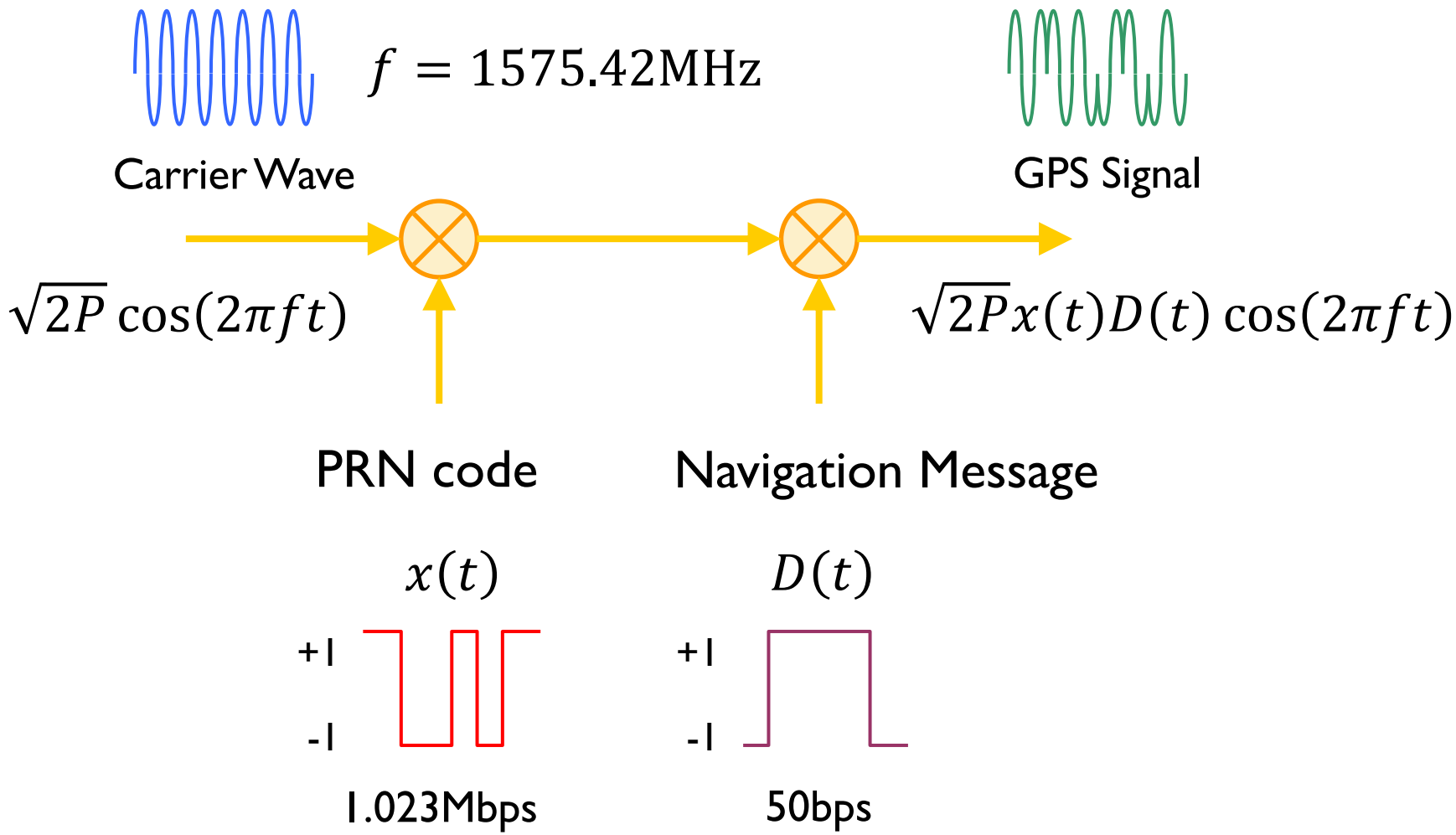
Taro Suzuki

GPS Signal Configuration

- Carrier wave + Data (PRN code + Navigation message)
  - BPSK modulation
- Tracking BPSK modulated signal and decode navigation message

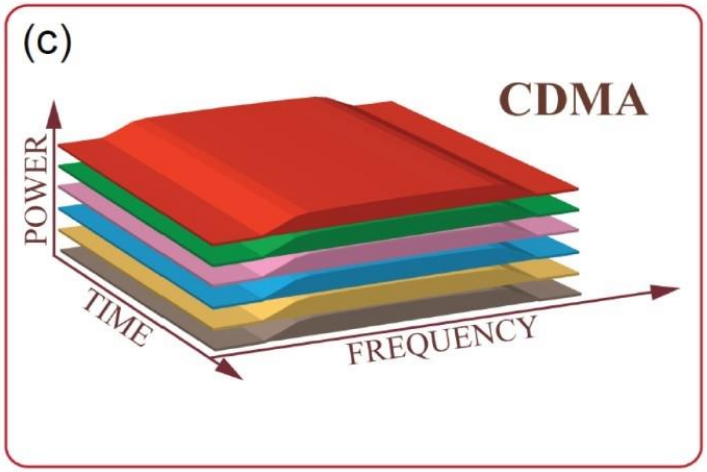


GPS Signal Architecture



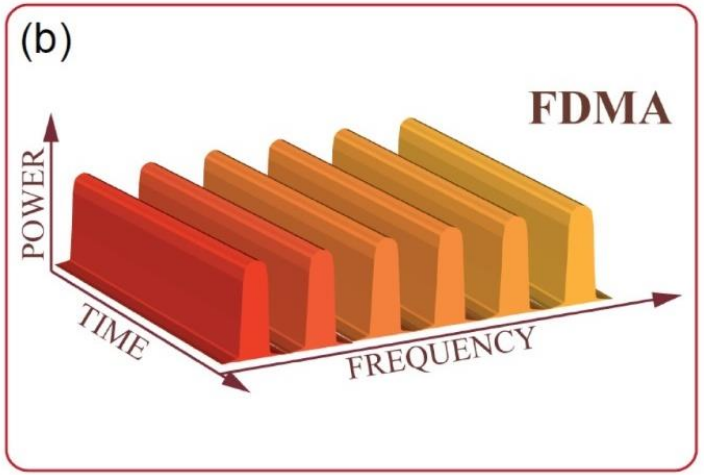
CDMA vs FDMA

CDMA
(Code Division Multiple Access)



- Each satellite uses the same frequency, but mixed with different distinguishing code pattern
- A channel is unique code pattern

FDMA
(Frequency Division Multiple Access)

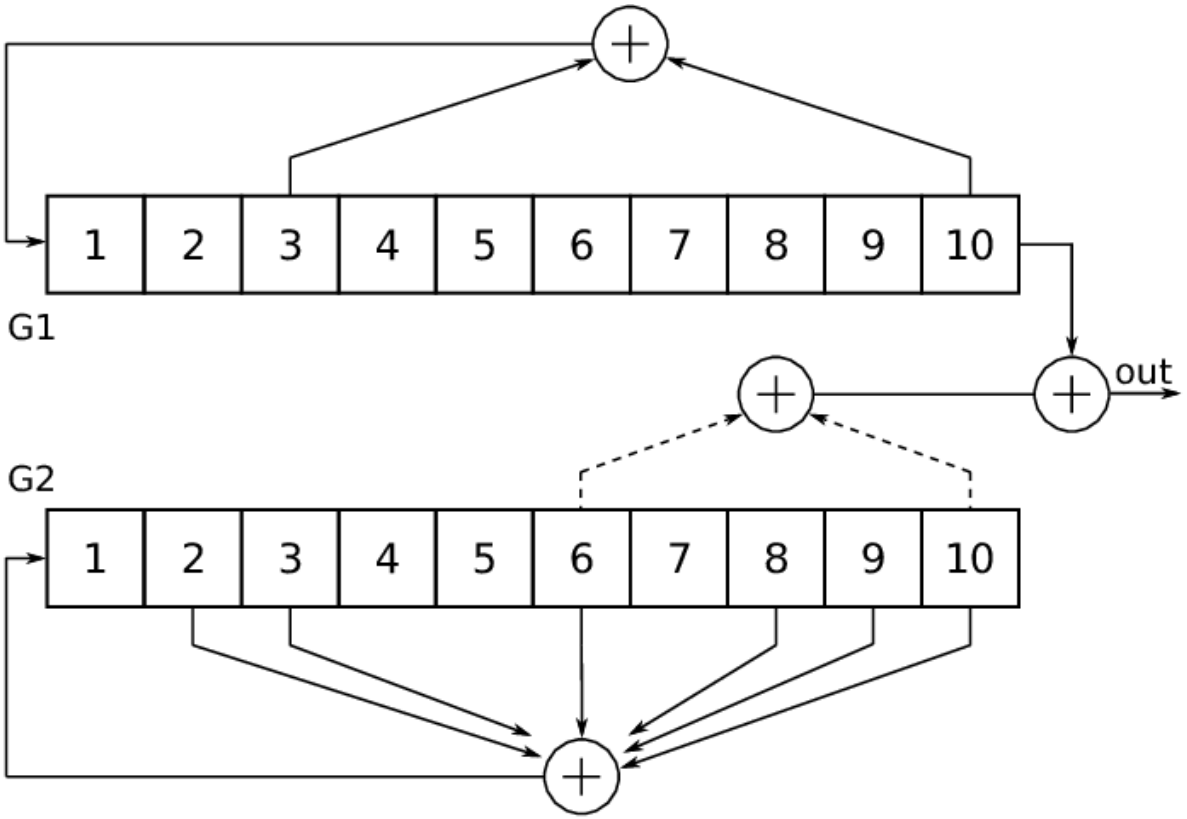


- Each satellite signal on different frequency
- A channel is frequency

Zhang, J., Liu, Yx., Özdemir, Ş. *et al.* Quantum internet using code division multiple access. *Sci Rep* **3,** 2211 (2013).

L1 C/A Code Generation

- Based on Gold Codes
- Two 10 bit linear feed shift register



Modulo 2 Operation

- Exclusive OR

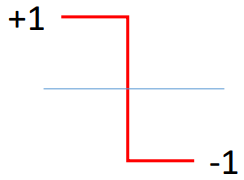
Represent all signals by “+1” or “-1” and use multiplication for Software Approach

0	⊕	0	=	0
0	⊕	1	=	1
1	⊕	0	=	1
1	⊕	1	=	0

1	X	1	=	1
1	X	-1	=	-1
-1	X	1	=	-1
-1	X	-1	=	1

- Either use “0” and “1” or “1” and “-1”
- We need the code representation in “1” and “-1” format

Example:
PRN 1 : 1 1 0 0 1 0 0 0 0 0….. Shall be
-1 -1 1 1 -1 1 1 1 1 1 ……

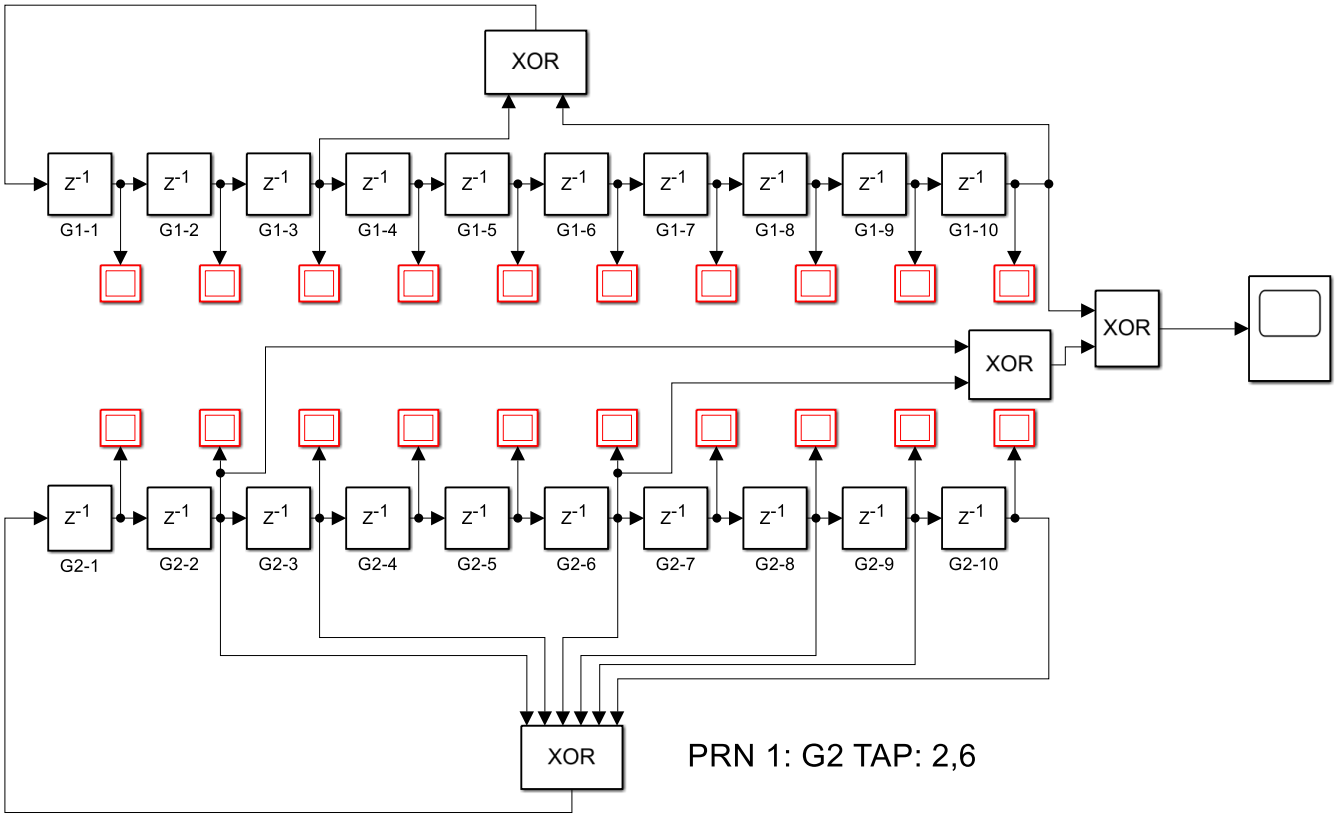


- Represent “0” by “1” and “1” by “-1”
- Then multiply for Modulo - 2 operation
  - Simpler for programming
- Digital signal can have only “+1” or “-1”
- “1” or “0” means “ON” and “OFF”

Exercise 1: C/A Code Generation

- Shift register model
  - **/06_06_GNSS_Signals+SDR_Overview/Simulink/Ex1/generate_L1CA.slx**

- **Blocks**
  - Logical Operator
  - Delay
  - Display



Exercise 2: C/A Code Generation

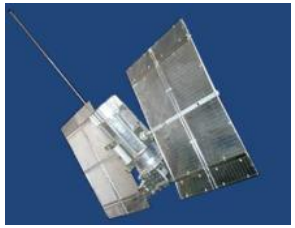
- MATLAB
  - **/06_06_GNSS_Signals+SDR_Overview/matlab/Ex2_generateCAcode.m**
  - phase selector

```
function [code,clen,crate] = generateCAcode(PRN)
clen = 1023;
crate = 1.023e6;

%--- Make the code shift array. The shift depends on the PRN number -------
% The g2s vector holds the appropriate shift of the g2 code to generate
% the C/A code (ex. for SV#19 - use a G2 shift of g2s(19) = 471)
g2s = [   5,   6,   7,   8,  17,  18, 139, 140, 141, 251,...
        252, 254, 255, 256, 257, 258, 469, 470, 471, 472,...
        473, 474, 509, 512, 513, 514, 515, 516, 859, 860,...
        861, 862, 863, 950, 947, 948, 950,  67, 103,  91,...
         19, 679, 225, 625, 946, 638, 161,1001, 554, 280,...
        710, 709, 775, 864, 558, 220, 397,  55, 898, 759,...
        367, 299,1018, 729, 695, 780, 801, 788, 732,  34,...
        320, 327, 389, 407, 525, 405, 221, 761, 260, 326,...
        955, 653, 699, 422, 188, 438, 959, 539, 879, 677,...
        586, 153, 792, 814, 446, 264,1015, 278, 536, 819,...
        156, 957, 159, 712, 885, 461, 248, 713, 126, 807,...
        279, 122, 197, 693, 632, 771, 467, 647, 203, 145,...
        175,  52,  21, 237, 235, 886, 657, 634, 762, 355,...
       1012, 176, 603, 130, 359, 595,  68, 386, 797, 456,...
        499, 883, 307, 127, 211, 121, 118, 163, 628, 853,...
        484, 289, 811, 202,1021, 463, 568, 904, 670, 230,...
        911, 684, 309, 644, 932,  12, 314, 891, 212, 185,...
        675, 503, 150, 395, 345, 846, 798, 992, 357, 995,...
        877, 112, 144, 476, 193, 109, 445, 291,  87, 399,...
        292, 901, 339, 208, 711, 189, 263, 537, 663, 942,...
        173, 900,  30, 500, 935, 556, 373,  85, 652, 310];
```

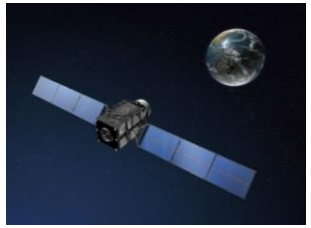
Multi-GNSS Era


GPS


GLONASS


Galileo

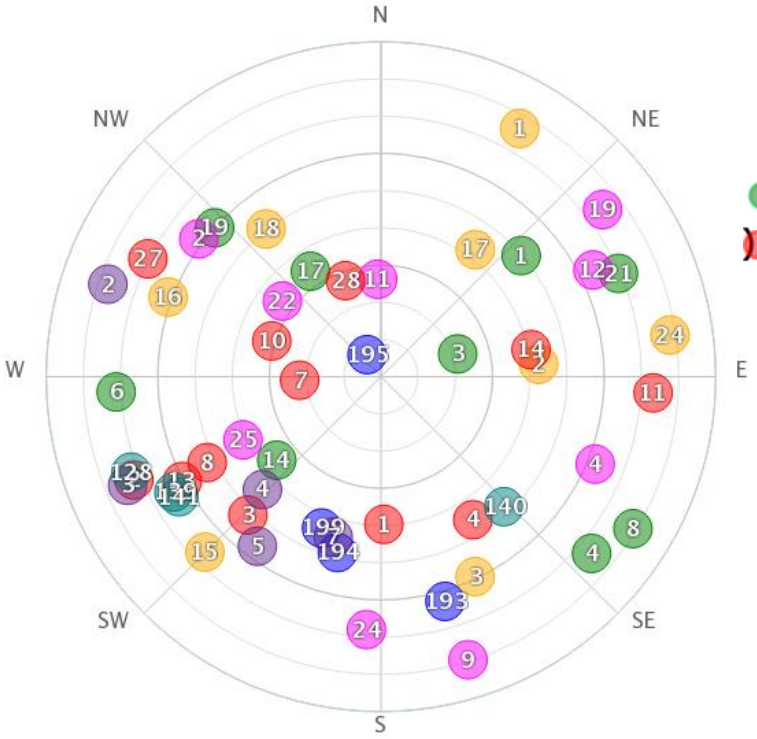

BeiDou

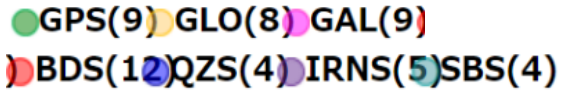

QZSS


IRNSS

2022/7/19
at Tokyo

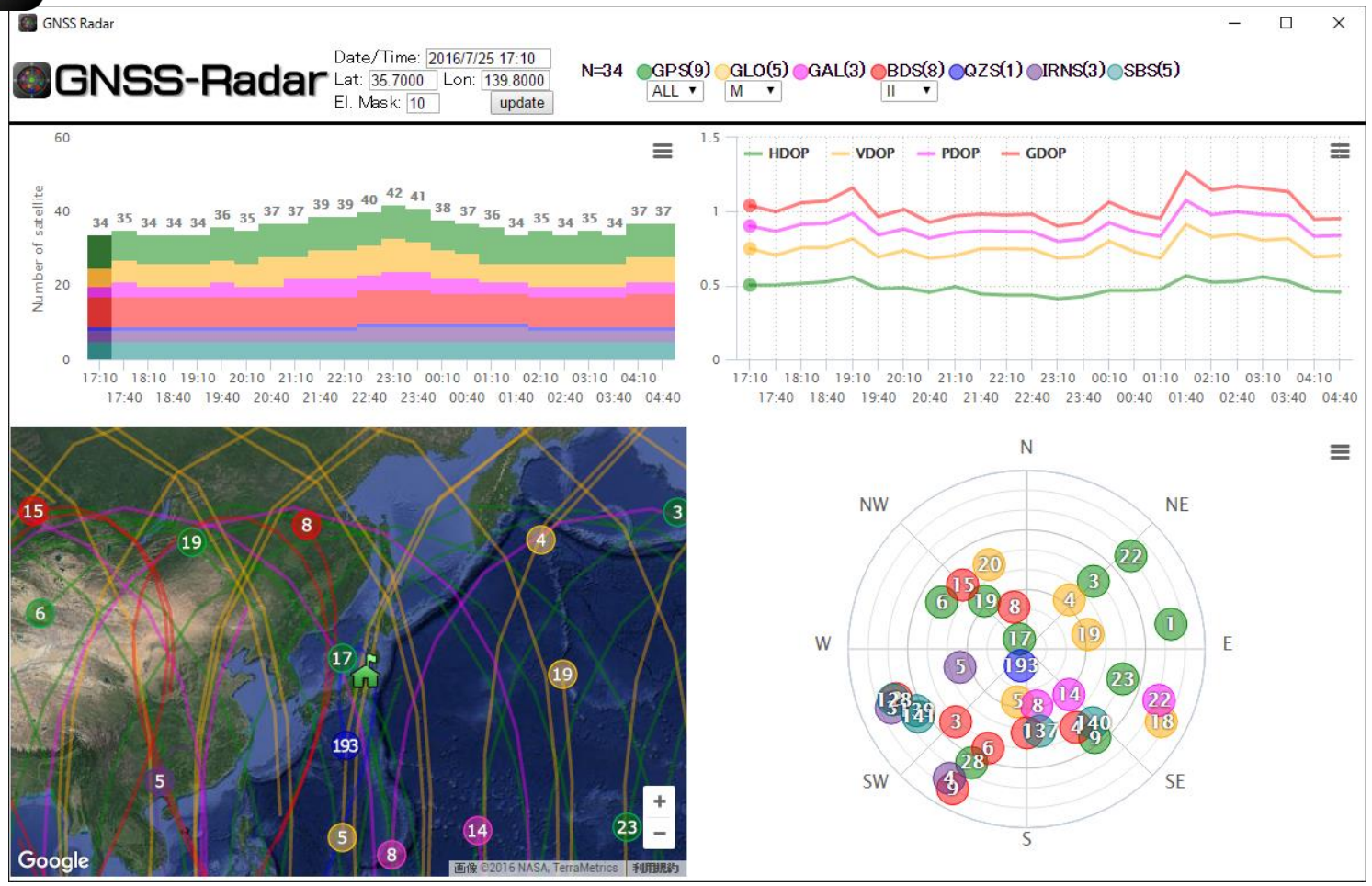
51 satellites!





GNSS Constellation

GNSS-Radar: http://www.taroz.net/GNSS-Radar.html



Exercise 3: GNSS-Radar

Source Code: https://github.com/taroz/GNSS-Radar

Options:

Set the observer location by latitude and longitude (the unit is degree)

ULR+?lat=xxx&lon=xxx (default: lat=35.7&lon=139.8 (Tokyo))

e.g. http://www.taroz.net/GNSS-Radar.html?lat=-37.8&lon=145

Set the elevation mask angle when computing the sky plot (the unit is degree)

ULR+?elemask=xxx (default: elemask=10)

e.g. http://www.taroz.net/GNSS-Radar.html?elemask=45

Set the time offset when computing the sky plot (the unit is hour)

ULR+?offhr=xxx (default: offhr=0)

e.g. http://www.taroz.net/GNSS-Radar.html?offhr=12

Set the time interval when computing the sky plot (the unit is minutes)

ULR+?tint=xxx (default: tint=30)

e.g. http://www.taroz.net/GNSS-Radar.html?tint=5

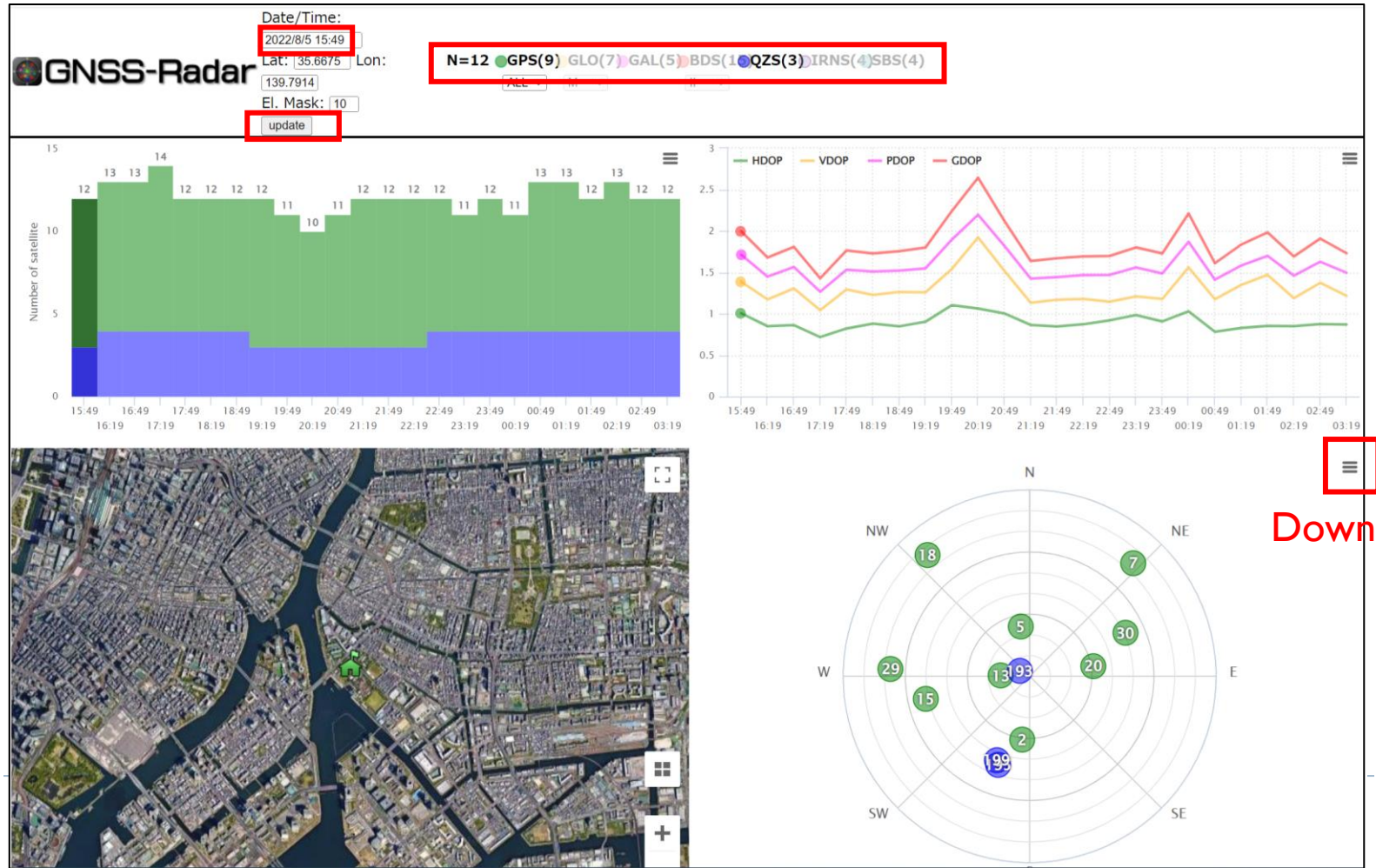
Set the number of times when computing the sky plot

ULR+?ntimes=xxx (default: tint=24, 24*30min=12hour)

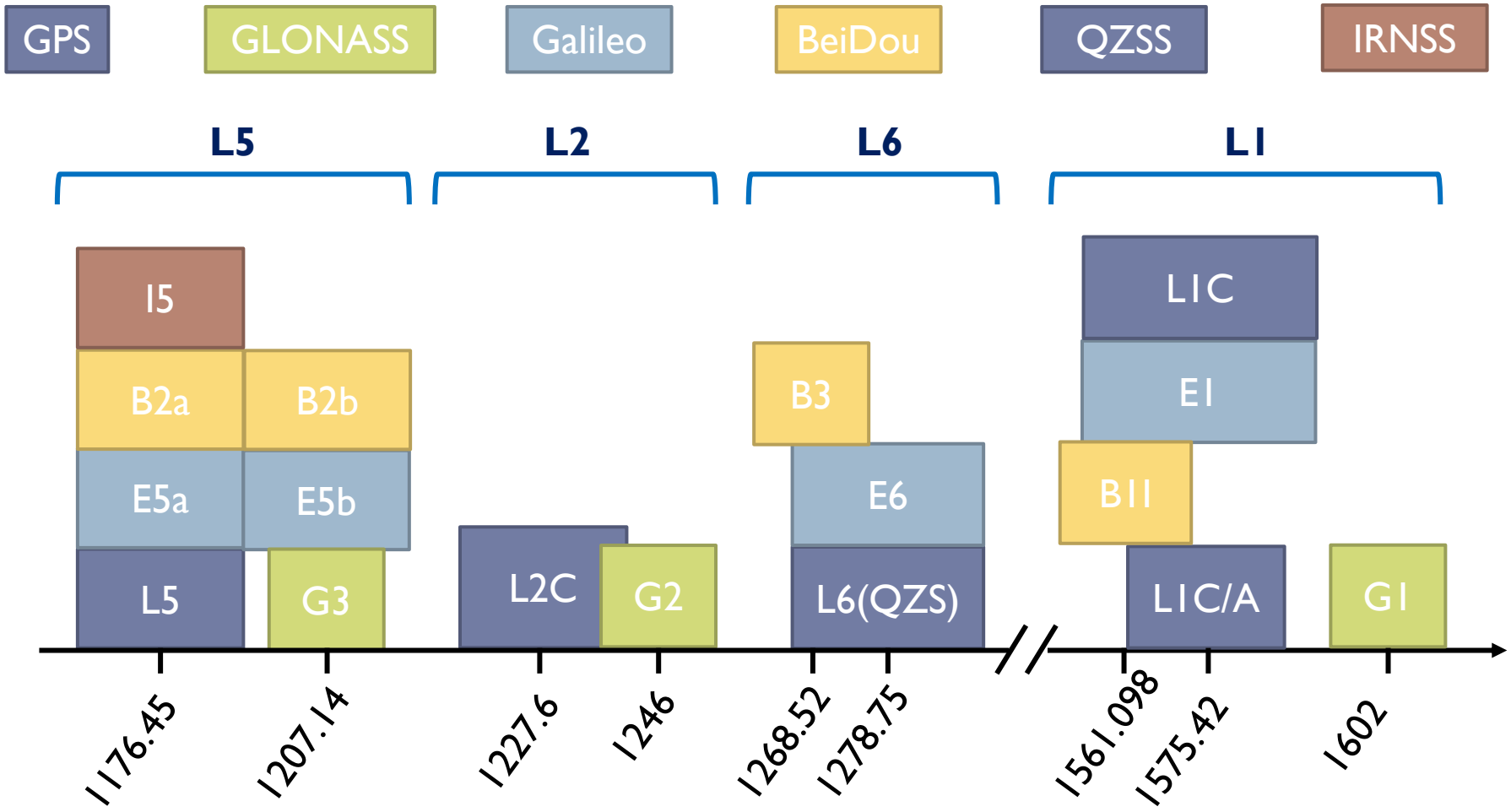
e.g. http://www.taroz.net/GNSS-Radar.html?ntimes=48

Exercise 3: GNSS-Radar

- Save satellite constellation at the time of RF data acquisition
  - 2022/08/05 15:49, At TUMSAT



GNSS Signal Band

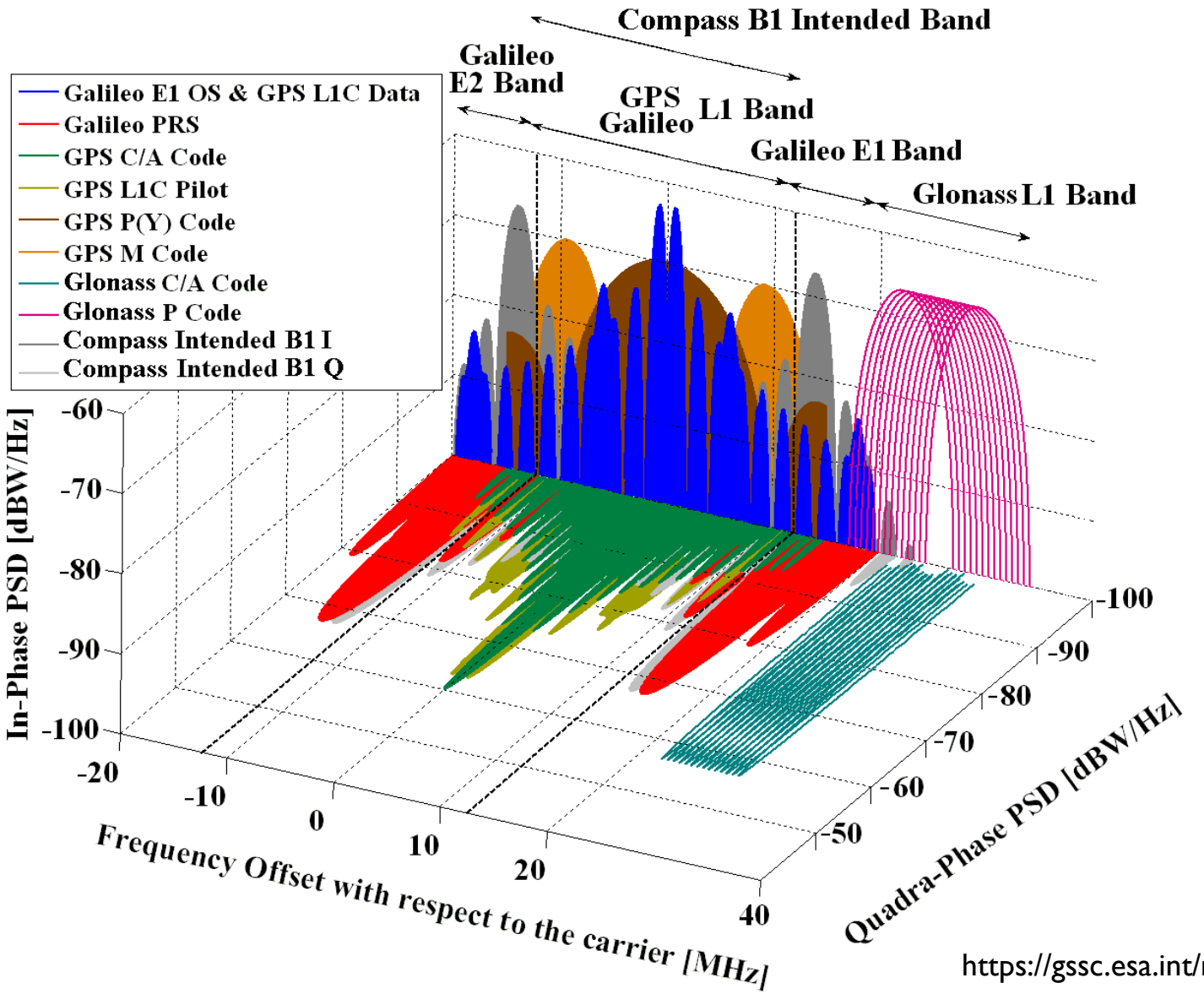


GNSS L1 Band

Around **L1 frequency** (1575.42 MHz)

GNSS	GPS/QZSS	GPS/QZSS		GALILEO		GLONASS	BeiDou
Service Name	C/A	L1C		E1		C/A (G1)	B1I
Center Freq.	1575.42 MHz	1575.42 MHz		1575.42 MHz		1602+ 0.5625K MHz	1561.098 MHz
Signal Component	Data	L1CD Data	L1CP Pilot	E1B Data	E1C Pilot	Data	Data
I/Q	Q	I	Q	I	Q	I	I
Band Width	2.046 MHz	4.096 MHz		24.552 MHz		1.002 MHz	2.046 MHz
Modulation	BPSK(1)	BOC(1,1)		CBOC(6,1,1/11)		BPSK	BPSK
Code Freq.	1.023 MHz	1.023 MHz		1.023 MHz		0.511 MHz	2.046 MHz
Code Chips	1023	10230		4092		511	2046
Code Length	1ms	10 ms	10 ms	4 ms	4 ms	1 ms	1 ms
Nav. Data	NAV	CNAV-2	-	I/NAV	-	NAV	D1/D2 NAV
Min. Received Power	-158.5 dBW	-163.0 dBW	-158.25 dBW	-163.0 dBW	-158.25 dBW	-161.0 dBW	-163.0 dBW

GNSS L1 Band PSD



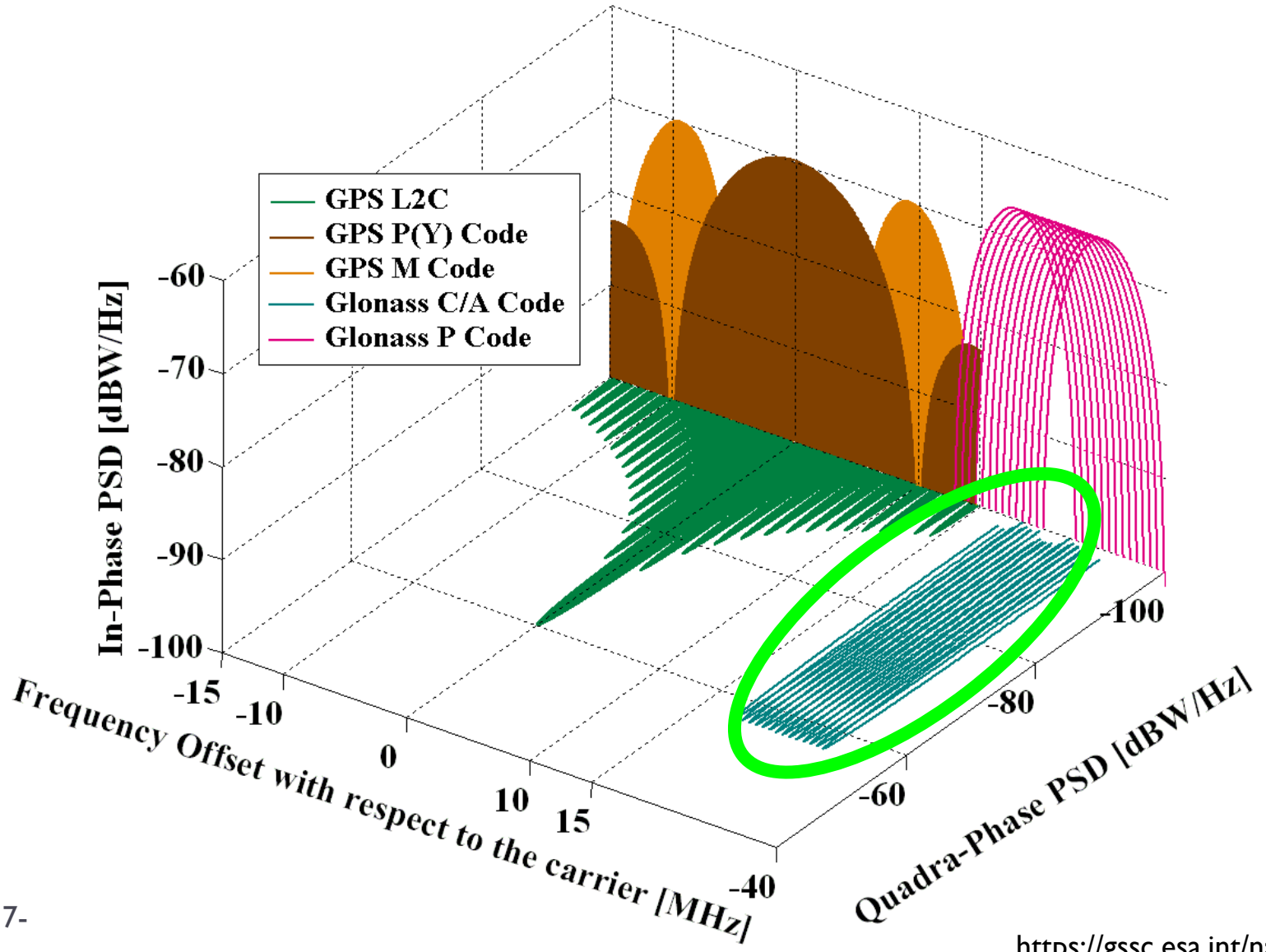
https://gssc.esa.int/navipedia

GNSS L2 Band

Around **L2 frequency** (1227.60 MHz)

GNSS	GPS/QZSS		GLONASS
Service Name	L2C		C/A (G2)
Center Freq.	1227.60 MHz		1246+ 0.4375K MHz
Signal Component	L2CM Data	L2CL Pilot	Data
I/Q	I		I
Band Width	2.046 MHz		1.022 MHz
Modulation	BPSK		BPSK
Code Freq.	0.5115 MHz		0.511 MHz
Code Chips	10230	767250	511
Code Length	20 ms	1.5 s	1 ms
Nav. Data	CNAV	-	NAV
Min. Received Power	-160.0 dBW		-167.0 dBW

GNSS L2 Band PSD



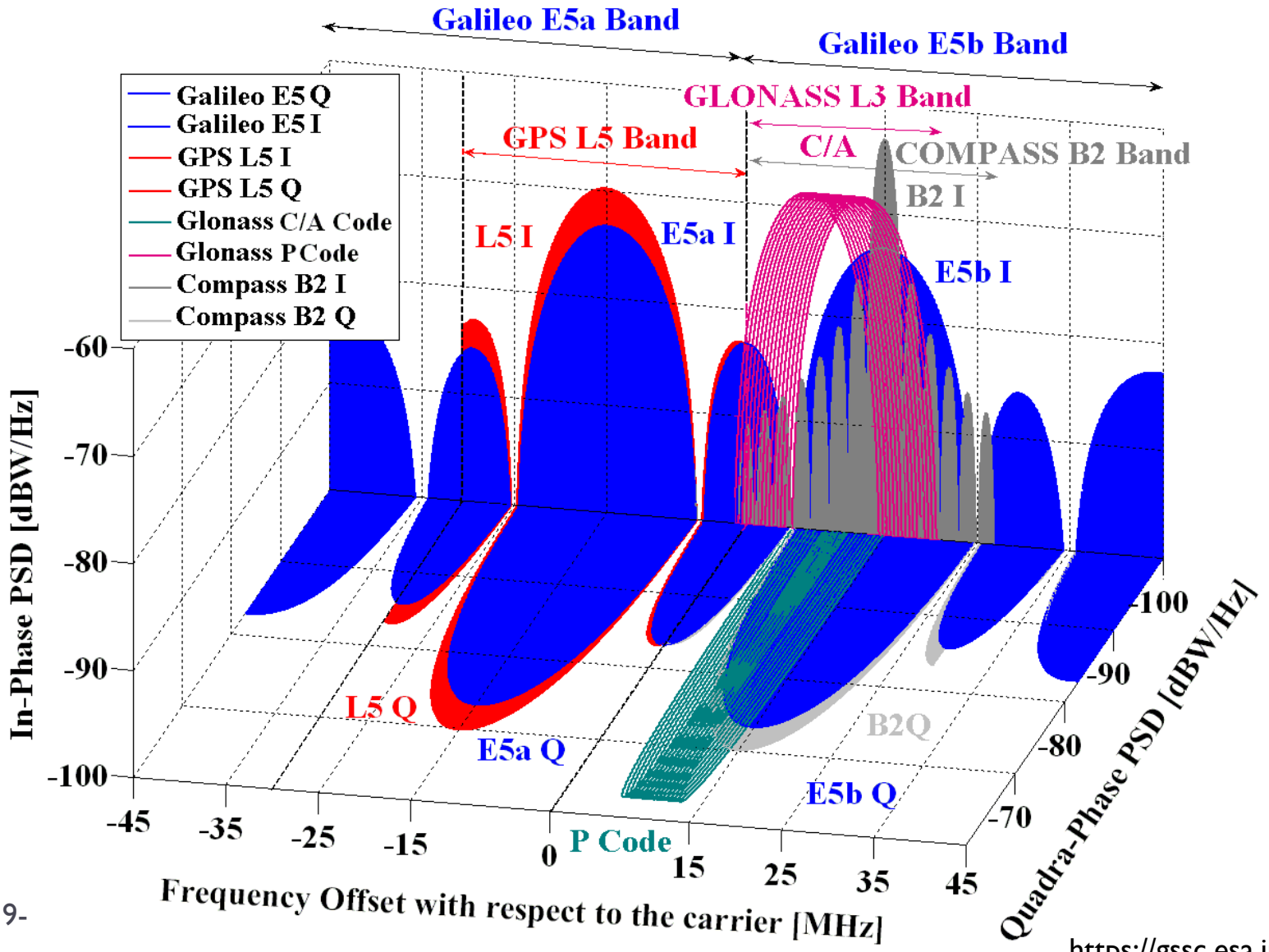
https://gssc.esa.int/navipedia

GNSS L5 Band

Around **L5 frequency** (1176.45 MHz)

GNSS	GPS/QZSS		GALILEO				BeiDou
Service Name	L5		E5a		E5b		B2I
Center Freq.	1176.45MHz		1176.45MHz		1207.14MHz		1207.14 MHz
Signal Component	L5I Data	L5Q Pilot	E5aI Data	E5aQ Pilot	E5bI Data	E5bQ Pilot	B2I Data
I/Q	I	Q	I	Q	I	Q	I
Band Width	20.46 MHz		20.46 MHz		20.46 MHz		24.0 MHz
Modulation	BPSK(10)		BPSK(10)		BPSK(10)		BPSK(10)
Code Freq.	10.23 MHz		10.23 MHz		10.23 MHz		10.23 MHz
Code Chips	10230		10230		10230		10230
Code Length	1 ms	1 ms	1 ms	1 ms	1 ms	1 ms	1 ms
Nav. Data	CNAV	-	F/NAV	-	I/NAV	-	D1/D2 NAV
Min. Received Power	-157.9 dBW	-157.9 dBW	-155.0 dBW	-155.0 dBW	-155.0 dBW	-155.0 dBW	-163 dBW

GNSS L5 Band PSD



https://gssc.esa.int/navipedia

Navigation Message

Band	System	Signal	Nav. Type	Rate	Error Detection / Correction	Preamble bits	Secondary Code
L1	GPS/ QZS	L1CA	NAV	50 bps, 300 bits, 6 sec.	Hamming Code	8bit	-
		L1C	CNAV-2	100 bps, 1800 bits, 18 sec.	BCH+LDPC+Interleving	None	1800 bits
	GALILEO	E1	I/NAV	125 bps, 250 bits, 2 sec.	½Convolusion+Interleaving+CRC	10bit	25 bits (E1C)
	GLONASS	G1	NAV	50 bps, 100 bits, 2 sec.	Hamming Code	30bit	-
	BeiDou (MEO)	B1I	D1 NAV	50 bps, 300 bits, 6 sec.	BCH+Interleaving	11bit	NH20
	BeiDou (GEO)	B1I	D2 NAV	500 bps, 300 bits, 0.6 sec.	BCH+Interleaving	11bit	-
	SBAS	L1	SBAS	250 bps, 250 bits, 1 sec.	½Convolusion	(8x3) bit Encoded	-
L2	GPS/QZS	L2C	CNAV	25 bps, 300 bits, 12 sec.	½Convolusion	8bit	-
	GLONASS	G2	NAV	50 bps, 100 bits, 2 sec.	Hamming Code	30bit	-
L5	GPS/ QZS	L5	CNAV	50 bps, 300 bits, 6 sec.	½Convolusion	8bit	NH10 (L5I), NH20 (L5Q)
	GALILEO	E5a	F/NAV	25 bps, 250 bits, 10 sec.	½Convolusion+Interleaving+CRC	10bit	20 bits (E5aI) 100 bits (E5aQ)
	GALILEO	E5b	I/NAV	125 bps, 250 bits, 2 sec.	½Convolusion+Interleaving+CRC	10bit	4 bits (E5bI) 100 bits (E5aQ)
	BeiDou (MEO)	B1I	D1 NAV	50 bps, 300 bits, 6 sec.	BCH+Interleaving	11bit	NH20
	BeiDou (GEO)	B1I	D2 NAV	500 bps, 300 bits, 0.6 sec.	BCH+Interleaving	11bit	-

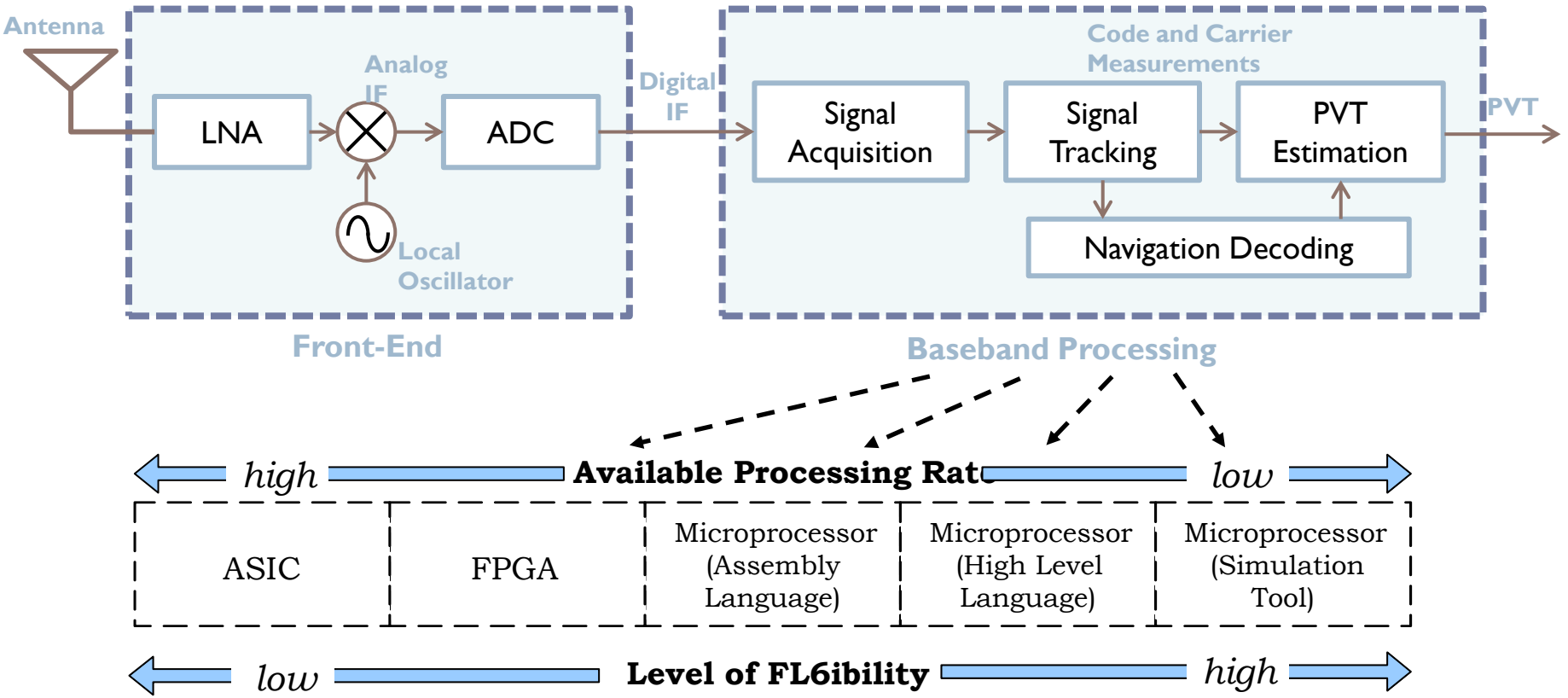
06: SDR Overview

Taro Suzuki

SDR (Software Defined Radio)

The software radio concept is built upon two basic principles

1. Move the analog-to-digital converter (ADC) as close to the antenna as possible
2. Process the resulting samples using a programmable processor



Why Software Receiver?

Features	ASIC (Hardware)	SDR (Software)
Upgradability	◆ A fixed platform ◆ Dictate the potential capabilities of the receiver	◆ Re-programmable ◆ Re-configurable
Acquisition	◆ Serial search acquisition ◆ Convolution in the time domain	◆ Parallel search acquisition ◆ FFT, Multiplication in the frequency domain
Tracking	◆ More efficient ◆ Cost effective	◆ Depends on the processor MIPS availability
Power consumption	◆ Less power consumption	◆ More power consumption
Cost effectiveness	◆ More hardware, More cost	◆ Less hardware, Less cost

Open-Source GNSS Software Receivers

SoftGNSS
http://ccar.colorado.edu/gnss/
MATLAB source codes, only for GPS L1 and post processing.

Fast GPS
http://sourceforge.net/projects/fastgps/ C++, only for GPS L1 and post processing

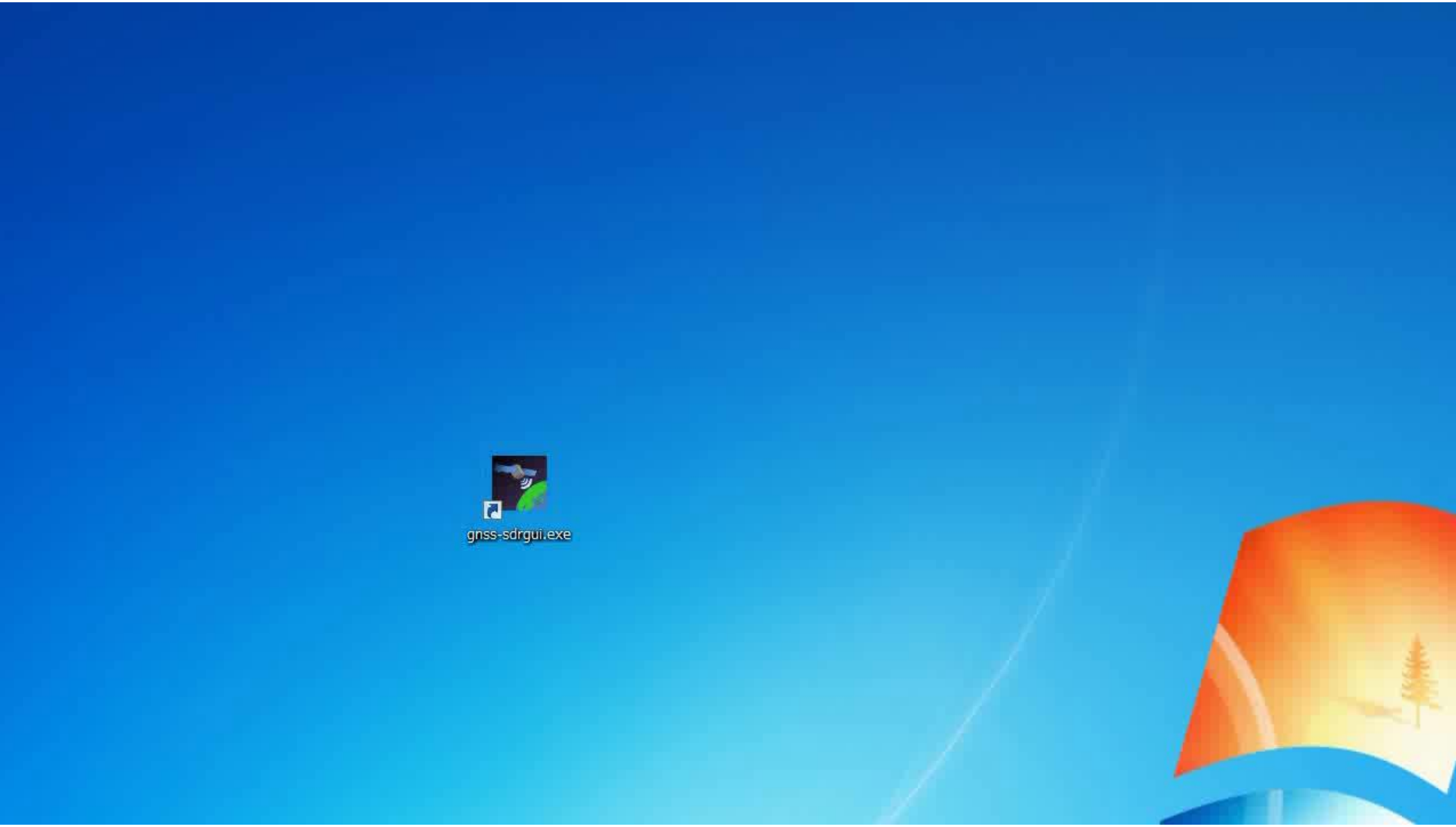
OpenSourceGPS
http://sourceforge.net/projects/osgps/ C++, only for GPS L1, real-time processing

GPS-SDR
https://github.com/gps-sdr C++, only for GPS L1, real-time processing

GNSS-SDR
http://gnss-sdr.org/ C++, Real-time Processing and Multi-GNSS support

PocketSDR
http:// github.com/gps-sdr Python, C, Multi-GNSS support

GNSS-SDRLIB



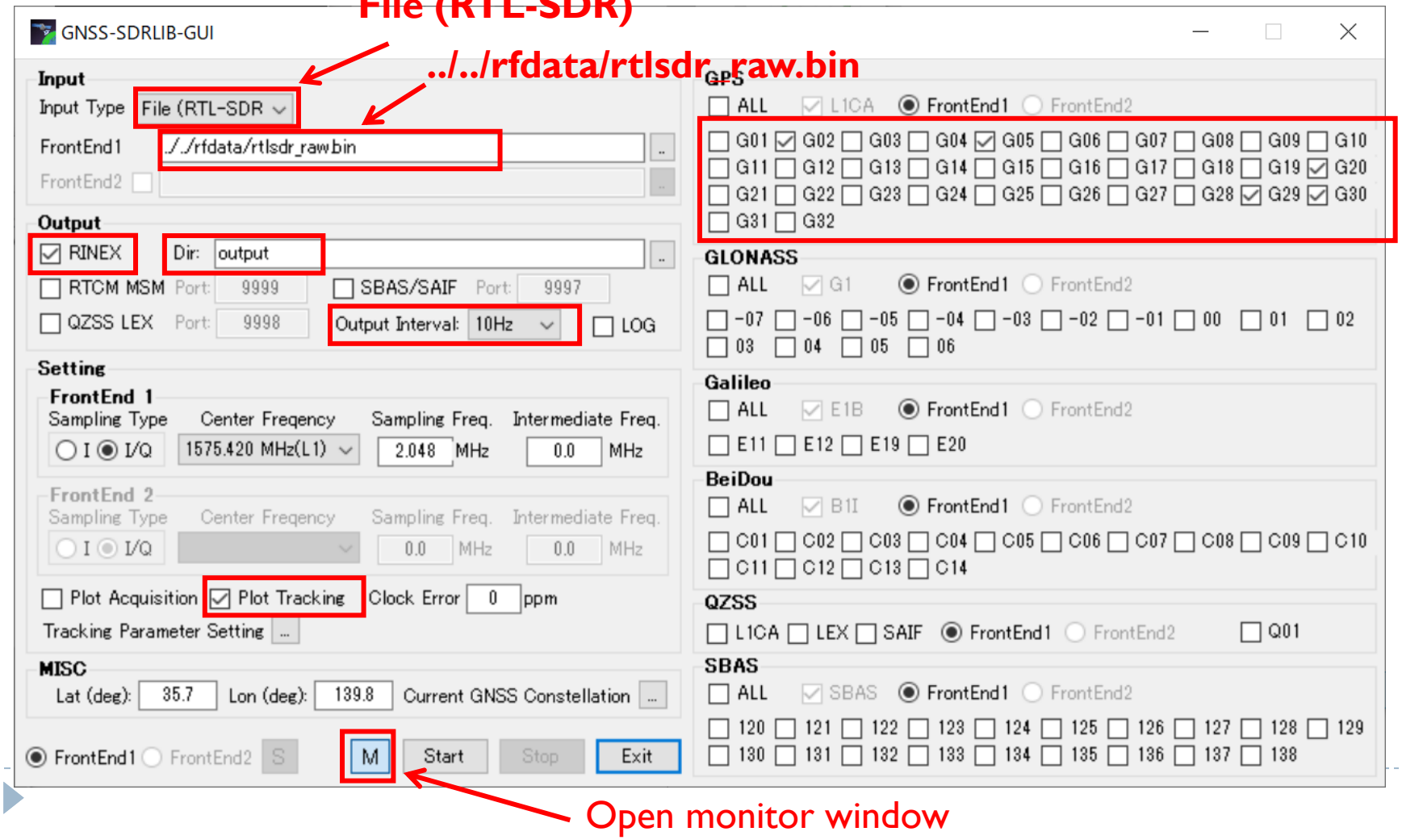
GNSS-SDRLIB

GNSS-SDRLIB https://github.com/taroz/GNSS-SDRLIB

- GNSS signal processing functions written in C
  - Code generation of all existing satellites
  - Signal acquisition / tracking functions
  - Decoding navigation messages
  - Pseudo-range / carrier phase measurements
- GUI application (AP) written in C++/CLI
- Real-time positioning with RTKLIB
- Observation data can be outputted in RINEX or RTCM format
  - Support following signals (tracking and decoding navigation message)
  - GPS, GLONASS, Galileo, BeiDou, QZSS L1 signals
  - Decoding QZSS SAIF/L6 message and SBAS message
- Support commercial front-ends for real-time positioning
- Support RF binary file for post processing

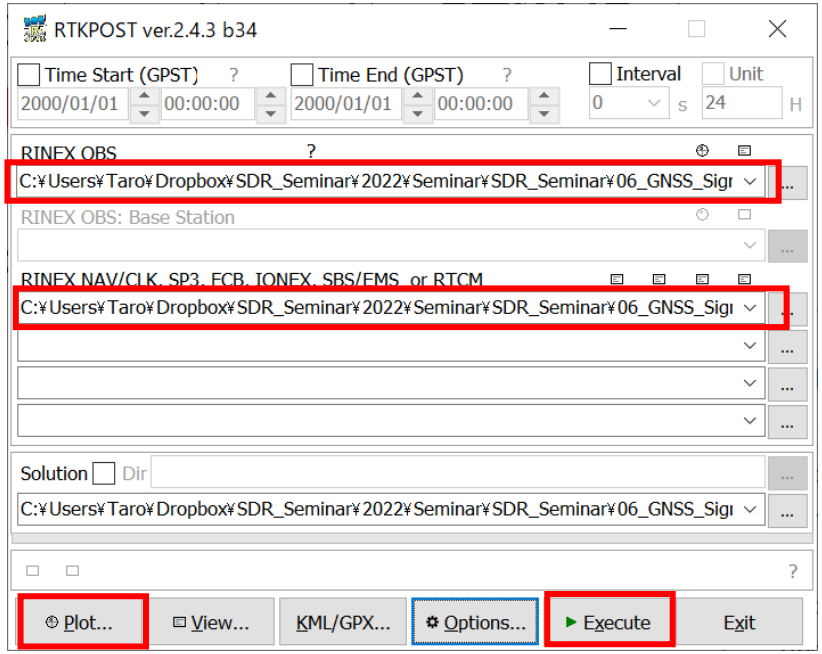
Exercise 4: Post-Processing (1)

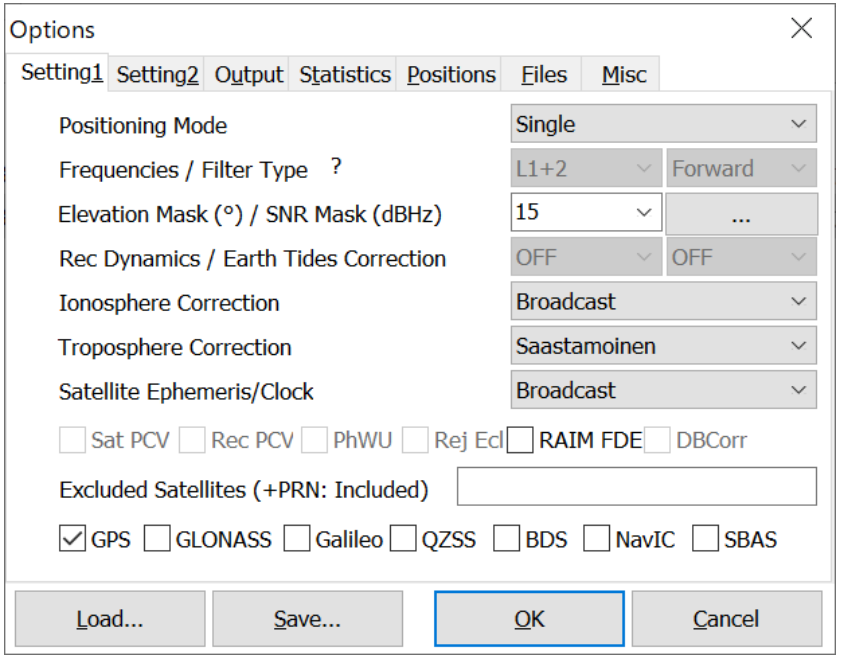
- **/06_GNSS_Signals+SDR_Overview/gnss-sdrlib/ gnss-sdrgui.exe**



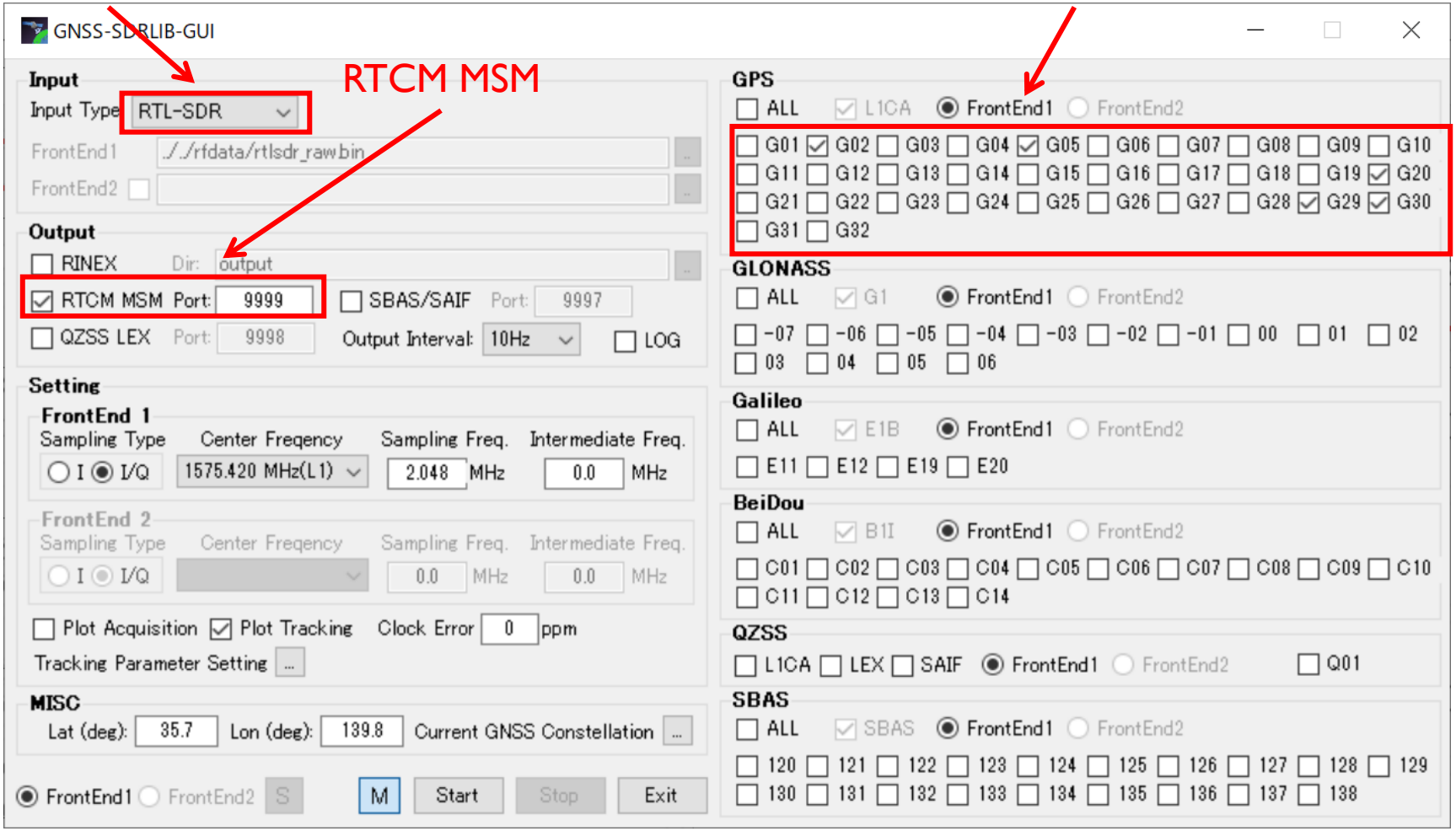
Exercise 4: Post-Processing (2)

- Post-processing using RTKLIB
  - **/rtklib/rtkpost.exe**
    - **output/sdr_2022++++++.obs**
    - **output/ sdr_2022++++++.nav**



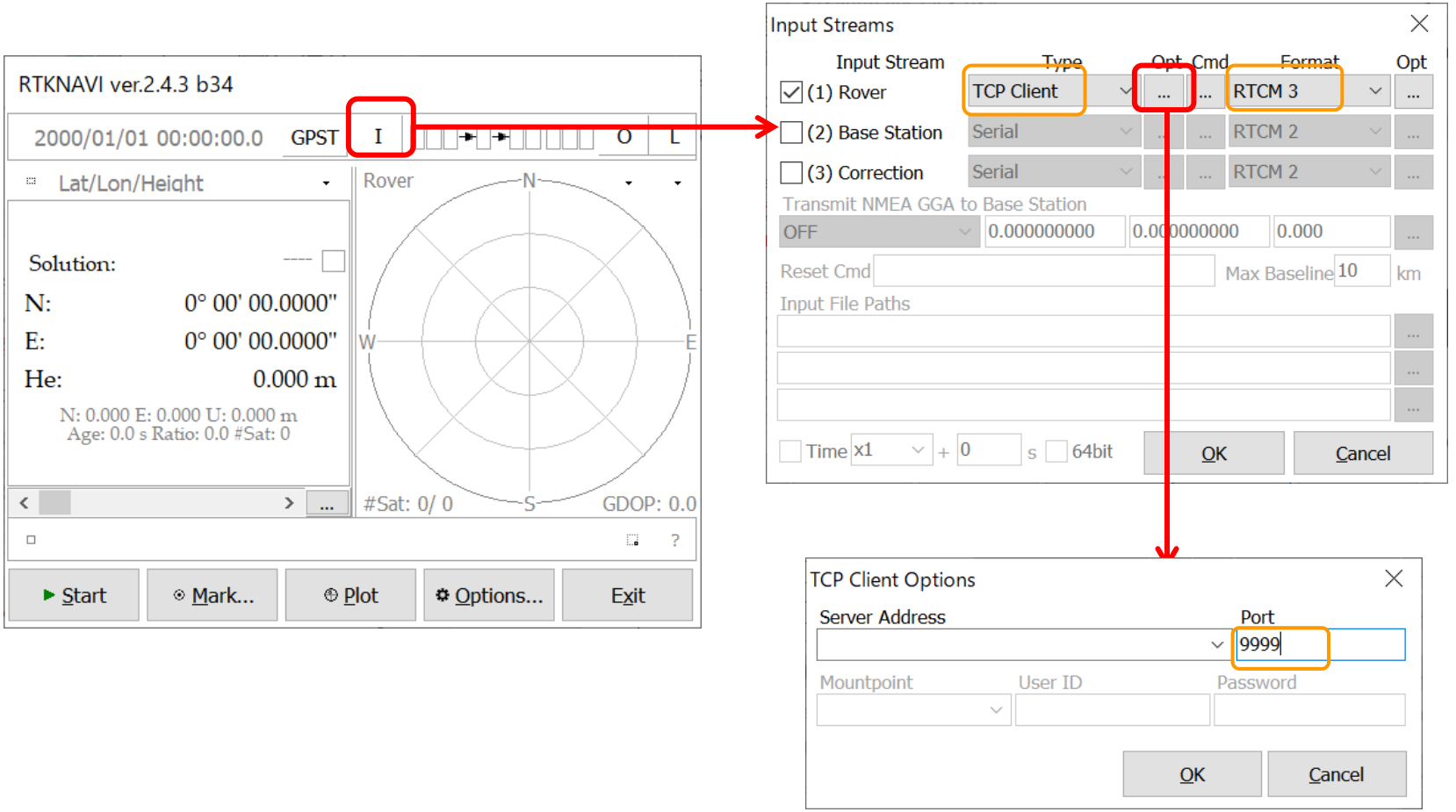


Exercise 5: Real-Time Processing (1)



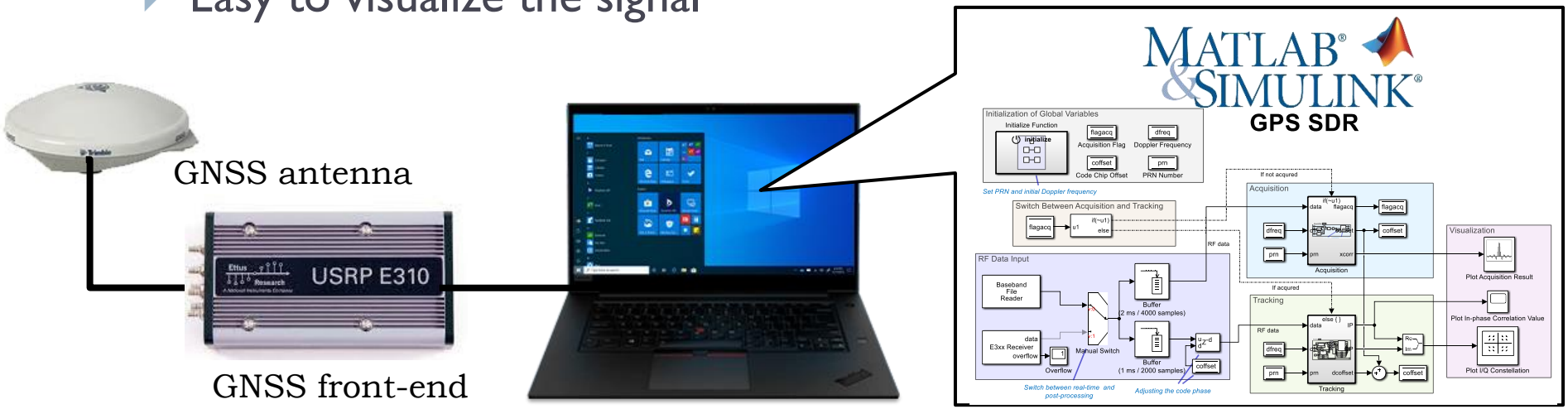
Exercise 5: Real-Time Processing (2)

- Real-time processing using /**rtklib**/**rtknavi.exe**

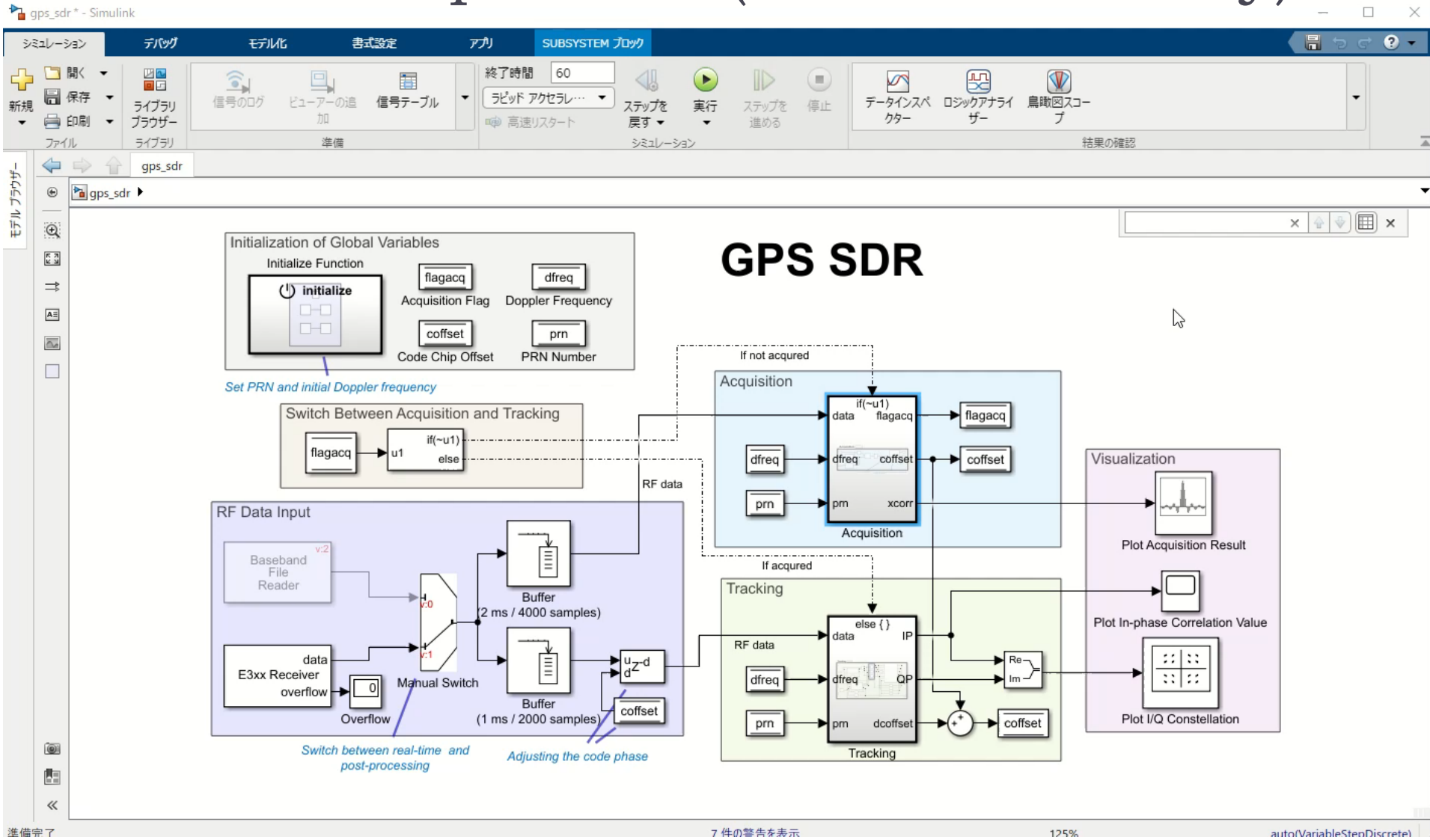


Simulink based SDR

- **Model-based Development** of software GNSS receiver
- Using **Matlab/Simlink**
  - Describes GNSS signal processing with a combination of models (blocks)
  - Easy to understand the signal processing flow
  - Easy to understand and low initial hurdle for **educational use**
  - Easy to visualize the signal



Real-time Operation (one satellite only)



Real-time Operation (6 satellites)