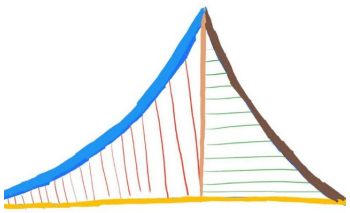
NAVIC L5 POC

Through C

G. V. V. Sharma



Contents

List of Figures

Chapter 1

Introduction

NavIC (an acronym for 'Navigation with Indian Constellation') is the operational name for Indian Regional Navigation Satellite System (IRNSS), developed independently and indigenously by Indian Space Research Organization (ISRO). The objective of this autonomous regional satellite navigation system is to provide accurate real-time positioning and timing services to users in India and a region extending upto $1,500$ km (930 mi) around it.

NavIC is designed with a constellation of 7 satellites and a network of ground stations operating 24 x 7. Three satellites of the constellation are placed in geostationary orbit and four satellites are placed in inclined geosynchronous orbit. The ground network consists of control centre, precise timing facility, range and integrity monitoring stations, two-way ranging stations, etc.

NavIC provides two levels of service, the "standard positioning service", which is open for civilian use, and a "restricted service" (an encrypted one) for authorised users (including the military). NavIC has a theoritical positional accuracy of 5m - 20m for general users and 0.5m for military purposes.

This book describes the Real time NavIC standards simulation using C code. Chapter 2 provides the information about the NavIC transmitter and how it is implemented in real time. Chapter 3 describes how the NavIC receiver is implemented in realtime and Chapter 4 provides the requirements and complete real time implementation of NavIC Transmitter and Receiver and Chapter 6 details out key results from the simulation.

1.1. Scope of simulation

The scope of the simulation is limited to

1. Generating the Real time NavIC Navigation data corresponds to the receiver location that contain the information of position of satellites in the orbit.

2. Generating the NavIC baseband samples from the navigation data.

3. Transmit the basband samples by mixing with carrier signal with L5 frequency to air.

4. Receive the L5 signals from air and down convert to baseband and feed it to the receiver module for computing the location of the receiver.

5. only SPS services signal (RS signal is out of scope)

Chapter 2

Transmitter Module

The NavIC transmitter module is simulated to generate the real time navigation data and generating the real time baseband samples for the NavIC L5 constellation.

2.1. Inputs to the Transmitter module

1. Rinex file that contains the navigation data of all the satellites in the NavIC L5 constellation.

   (a) The user specifies the NavIC satellite constellation through a NavIC broadcast ephemeris file. The daily NavIC broadcast ephemeris file (brdc) is a merge of the individual site navigation files into one.

   (b) The archive for the daily file can be downloaded from below site. Access to this site requires registration, which is free.

https://cddis.nasa.gov/archive/gnss/data/daily/

(c) These files are then used to generate the simulated pseudorange and Doppler for the NavIC satellites in view. This simulated range data is then used to generate the digitized I/Q samples for the NavIC signal.

2. Sampling frequency of the generated baseband samples.

3. Time duration in seconds.

4. size of each sample.(There is a flexibility of 1 bit, 8 bits and 16 bits)

   (a) The output of the transmiter module is a bin file that contain the IQ samples with the given sampling frequency and given size of samples.

   (b) If the bit size is 8 I samples will take 8 bits and Q samples will take 8 bits.

   (c) There is a flexibility of size of samples to 1bit,8bits and 16 bits.

5. Receiver location in terms of latitude,longitude,altitude.

2.2. Output of the Transmitter module

1. The bin file containg the NavIC L5 IQ samples corresponds to the given inputs.

(a) The bin file will be generated as output.The bin file contains the NavIC baseband IQ samples corresponds to the input receiver location.

(b) The transmitter module will give the bin file that contains the navigation data of satellites that are visible to the input receiver location with the real time doppler frequency and the real time codephase and carrier phase.

2.3. Up conversion of Baseband samples to the L5 frequency

1. The samples from the transmitter module is given to the frontend.The frontend will upconvert the basband signals to the L5 frequency and send the signals to the air through antenna.
2. The samples from the transmitter is in the form of I + jQ
3. In the NavIC L5 constellation In phase part is SPS service. and Quadrature part contains the RS service.
4. In the front end I sample is multiplied with cosine signal and Q sample is multiplied with sine signal with L5 frequency.

$$x(t) = I\cos(2\pi f_c t) + Q\sin(2\pi f_c t) \tag{2.1}$$

Chapter 3

Receiver Frontend

The signal transmitted from the transmiited frontend is received by the receiver frontend and down convert the samples to baseband and feed it to the receiver module.

3.1. Down conversion of received signal to baseband

1. The transmitted signal was received by the antenna at the receiver and processed by the receiver front end.

2. Let the received signal be x(t)

$$x(t) = I\cos(2\pi f_c t) + Q\sin(2\pi f_c t) \tag{3.1}$$

   where f_c is L5 frequency i.e 1176.45 MHz.

3. The signal will be downconverted to baseband when we multiply the

signal with the frequency of same signal.

$$y(t) = x(t)e^{-j2\pi f_c t} \tag{3.2}$$

$$y(t) = (I\cos(2\pi f_c t) + Q\sin(2\pi f_c t))(\cos(2\pi f_c t) - j\sin(2\pi f_c t)) \tag{3.3}$$

$$y(t) = (I\cos^2(2\pi f_c t) + Q\sin(2\pi f_c t)\cos(2\pi f_c t)) + j(Q\sin^2(2\pi f_c t) - I\sin(2\pi f_c t)\cos(2\pi f_c t)) \tag{3.4}$$

4. Apply low pass filter to the above signal so high frequency components are removed at the output of lowpass filter.

5. The out of low pass filter is

$$y(t) = I + jQ \tag{3.5}$$

6. The received samples are now down converted to the baseband and feed it to the receiver module.

Chapter 4

Receiver Module

The baseband samples optained from the receiver frontend is given to the receiver module.

4.1. Receiver operation

1. The samples from the frontend contains the doppler frequency and code phase and carrier phase.
2. The doppler frequency and codephase is removed by the receiver module and obtain the navigation data transmitted.
3. From the navigation data we find the position of satellites in the orbit.
4. From the position of satellites we find the position of receiver on the earth.
5. The detailed explanation of receiver is given in

gvv/navic_python/main.pdf

Chapter 5

Real time Implementation of Transmiter and Receiver

5.1. Transmitter implementaion

5.1.1. Installations for Transmitter module

```
sudo apt update
sudo apt install build−essential
sudo apt install make
```

5.1.2. Generating the real time samples using Tranmitter module

1. Clone the transmitter module from below

```
gvv/navic_L5_POC/codes/navic_transmiter
```

2. Build the project

```
make
```

3. The executable file will be generated in the same folder so run the executable file with the specifications.

```
./navic−sdr−sim −s 2500000 −e brdc1380.23n −b 16 −d
    300 −l 30,120,100
```

4. The above command will run and give the bin file that contain the navic L5 baseband samples for the duration of 300 seconds with 2.5MHz sampling frequency, 16 bits size with the location of 30,120,100 (lat,long,alt).

```
mannava@mannava-Inspiron-3476:~/gvv/navic_L5_POC/codes/navic_transmiter$ ./navic-sdr-sim -e brdc1380.23n  -s 2500000 -d 300 -l 30,120,100
Using static location mode.
xyz =  -2764171.6,   4787685.7,   3170423.7
llh =   30.000000,  120.000000,       100.0
Start time = 2023/05/18,00:00:00 (2262:345600)
Duration = 300.0 [sec]
02  181.3  17.4  23934787.2   9.3
03   79.2  47.5  21737003.4   5.8
04  214.7  25.0  23159097.6   7.7
05  316.2  41.5  22191545.6   5.9
06  335.7  75.3  20169966.0   4.3
07  251.1  28.2  22909990.4   7.1
09   67.3  33.2  22526843.4   7.6
Time into run = 300.0
Done!
Process time = 24.4 [sec]
mannava@mannava-Inspiron-3476:~/gvv/navic_L5_POC/codes/navic_transmiter$
```

Figure 5.1: Generating the bin file

5. The file generated as output is

```
gvv/navic_L5_POC/codes/navic_transmiter/navicsim.bin
```

6. The bin file contains the IQ samples with 16 bits size.

5.2. Transmitter frontend implementations

5.2.1. Requirements

1. The front end used at the transmitter is **USRP SDR** (Software Defined Radio).

2. The OS used is ubuntu 22.04 version.

Figure 5.2: USRP SDR

Instalations for USRP SDR

1. Install the basic requirements from the below commands.

```
sudo apt-get install libuhd-dev uhd-host
sudo add-apt-repository ppa:ettusresearch/uhd
sudo apt-get update
sudo apt-get install libuhd-dev uhd-host
sudo apt-get install autoconf automake build-essential ccache
    cmake cpufrequtils doxygen ethtool \
g++ git inetutils-tools libboost-all-dev libncurses5 libncurses5-
    dev libusb-1.0-0 libusb-1.0-0-dev \
libusb-dev python3-dev python3-mako python3-numpy python3
    -requests python3-scipy python3-setuptools \
python3-ruamel.yaml
```

2. Clone the below repo for using the USRP and build the project.

```
git clone https://github.com/EttusResearch/uhd.git
cd uhd/host
mkdir build
cd build
cmake ../
cmake -DCMAKE_INSTALL_PREFIX=/opt/uhd ../
make
sudo ldconfig
#run the python code
```

Linux: /usr/local/lib/uhd/utils/uhd_images_downloader.py

5.2.2. Transmit the NavIC L5 real time samples using the Transmitter module and Transmitter frontend

1. The generated bin file in transmitter module has to be give input to the USRP so it will up convert the samples to L5 frequency and transmit the signals to air through antenna.

2. Before feeding the bin file first we need to setup the usrp.

3. Connect the antenna to the USRP SDR in RF A slot in Tx/Rx port.

4. Connect the USRP to the PC with the USB cable so that the USRP will on.

5. Go the to uhd directory which was cloned before.

6. In order to start the USRP use the command below.

```
uhd_usrp_probe
```

7. Go to the below directory in uhd copy the executable file and paste

Figure 5.3: Connecting antenna to USRP SDR

in the folder that contain the bin file generated by the transmitter module.

```
cd /home/mannava/uhd/host/build/examples
cp /home/mannava/uhd/host/build/examples/
    tx_samples_from_file /home/mannava/gvv/
    navic_L5_POC/codes/navic_transmiter
```

8. Run the executable file with the below command so that the bin file will feed to the USRP and USRP start transmit the signal through antenna.

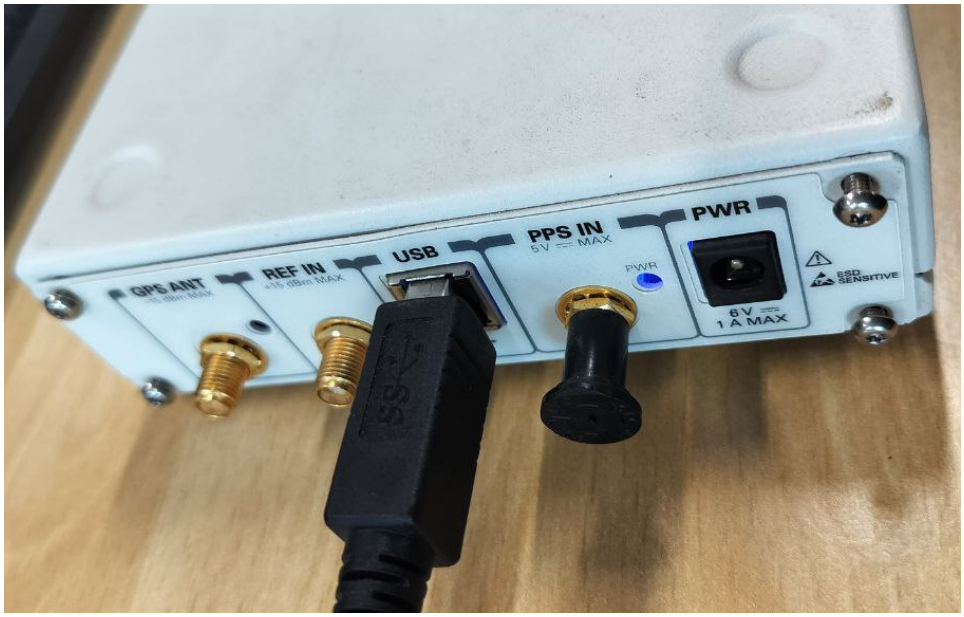


Figure 5.4: Connecting USRP SDR to PC

```
./tx_samples_from_file --file navicsim.bin --type short --
rate 2500000 --freq 1176450000 --gain 75
```

9. By running the command the USRP will start the transmit the signals to air.

10. While transmiting the red led will on in usrp beside antenna port that indicate the signal is being transmiting to air through antenna.

```
mannava@mannava-Inspiron-3476:~$ cd uhd
mannava@mannava-Inspiron-3476:~/uhd$ uhd_usrp_probe
[INFO] [UHD] linux; GNU C++ version 11.4.0; Boost_107400; UHD_4.5.0.0-0ubuntu1~jammy1
[INFO] [B200] Loading firmware image: /usr/share/uhd/images/usrp_b200_fw.hex...
[INFO] [B200] Detected Device: B210
[INFO] [B200] Loading FPGA image: /usr/share/uhd/images/usrp_b210_fpga.bin...
[INFO] [B200] Operating over USB 2.
[INFO] [B200] Detecting internal GPSDO....
[INFO] [GPS] No GPSDO found
[INFO] [B200] Initialize CODEC control...
[INFO] [B200] Initialize Radio control...
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Setting master clock rate selection to 'automatic'.
[INFO] [B200] Asking for clock rate 16.000000 MHz...
[INFO] [B200] Actually got clock rate 16.000000 MHz.
```

Figure 5.5: Setting up the USRP SDR

Figure 5.6: Signal is transmitting to air

5.3. Receiver implementaion

5.3.1. Receiver frontend

1. The frontend used at receiver for receiving the transmitted samples and convert transmitted samples to the basband samples is **blade rf**.

2. The OS used for implementing the receiver module is ubuntu 22.04.

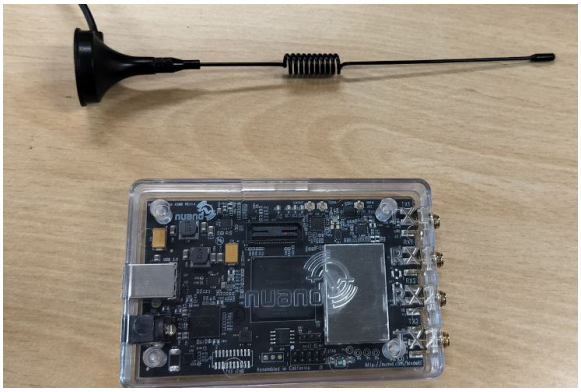


Figure 5.7: Bladerf

5.3.2. Installation for blade rf

1. Install the basic installations by the below commands.

```
sudo add−apt−repository ppa:nuandllc/bladerf
sudo apt−get update
sudo apt−get install bladerf
sudo apt−get install libbladerf−dev
sudo apt−get install bladerf−firmware−fx3
sudo apt−get install bladerf−fpga−hostedx40
sudo apt−get install bladerf−fpga−hostedx115
sudo apt−get install bladerf−fpga−hostedxa4
sudo apt−get install bladerf−fpga−hostedxa9

sudo apt−get install libusb−1.0−0−dev libusb−1.0−0 build−
```

```
    essential cmake libncurses5-dev libtecla1 libtecla-dev pkg-
    config git wget


dpkg -s libusb-1.0-0 libusb-1.0-0-dev
sudo apt-get install doxygen help2man pandoc
```

2. Clone the below repo and build the repo for setting up the bladerrf.

```
git clone https://github.com/Nuand/bladeRF.git ./bladeRF
cd ./bladeRF
cd host/
mkdir build
cd build
cmake -DCMAKE_BUILD_TYPE=Release -
    DCMAKE_INSTALL_PREFIX=/usr/local -
    DINSTALL_UDEV_RULES=ON ../
groups
sudo groupadd bladerf
sudo usermod -a -G bladerf jon
make && sudo make install && sudo ldconfig
```

5.3.3. Setting up the NavIC receiver

1. The transmitted signals are received to the bladerf.The bladerf will convert the signals to baseband by removing the carrier as we discussed

above chapter.These baseband samples are I and Q samples.

2. The IQ samples are feeded to the receiver module.

3. The receiver module perform

   (a) Acquisition

   (b) Tracking

   (c) bit synchronisation

   (d) BPSK demodulation

   (e) Frame synchronisation

   (f) Convolutional decoding

   (g) Frame decoding.

4. The receiver module will perform all the above operations and gives the output as .nav file and .obs file.

5. The nav file contains the orbital parameters of all visible satellites. From the nav file we can find the position of satellite in orbit.

6. The obs file contains the code and carrier properties.

7. With the nav file and obs file we can find out the position of receiver.

8. In order to compute the location of the receiver from the nav file and obs file we use the library called RTKLIB.

9. Clone the RTKLIB from the below repo.

```
git clone git clone https://github.com/tomojitakasu/
    RTKLIB.git
```

10. Build the RTKLIB library.

```
cd RTKLIB/app/rnx2rtkp/gcc
make
```

11. For setting the receiver clone the below repo.

```
gvv/navic_L5_POC/codes/navic_receiver
```

The souce codes for all the receiver module is there in below directory.

```
gvv/navic_L5_POC/codes/navic_receiver/src
```

12. Build the project by using the make in the below directory.

```
cd gvv/navic_L5_POC/codes/navic_receiver/cli/linux
make
```

13. By building the project the executable file of the receiver module will be generated.

14. The executable generated in RTKLIB is copied and paste in the below directory.

```
cp /home/mannava/RTKLIB/app/rnx2rtkp/gcc/rnx2rtkp /
    home/mannava/gvv/navic_L5_POC/codes/navic_receiver
    /bin/rinex
```

5.3.4. Receive the signals and compute the position of receiver

1. After all the installations of bladerf and setting up the receiver module Connect the antenna to the bladerf in RX1 port.

Figure 5.8: Connect antenna to Bladerf

2. Connect the bladerf to the PC through the USB cable.

3. configure the bladerf with the center frequency,sampling frequency gain and other properties by below comands.

Figure 5.9: Connect Bladerf to PC

```
bladeRF−cli −i <<EOF
set frequency rx 1176.45e6
set frequency tx 47e6
set agc off
set samplerate 2.048e6
set bandwidth 3.0e6
set gain rx 60
set gain tx 0
set clock_sel onboard
set biastee rx off
```

4. After all the configurations run the receiver module with the executable file as we generated earlier with the below command.By running the command the receiver module starts and perform all the operations of receiver and give the nav and obs files.

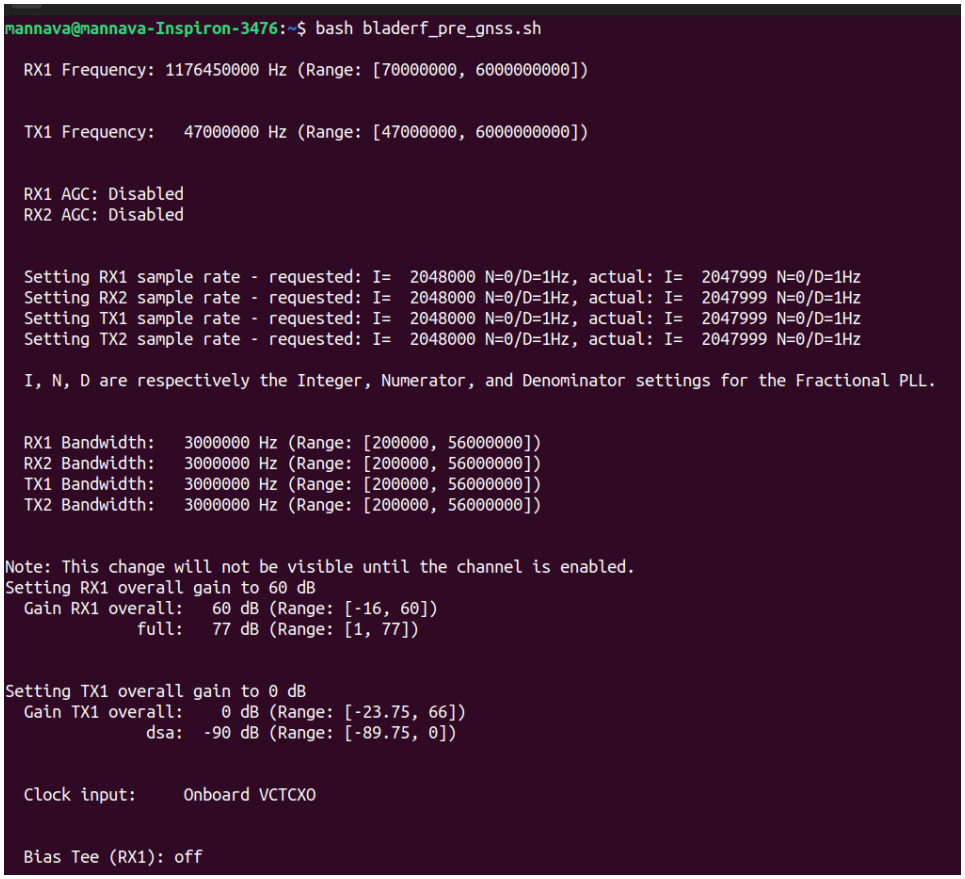


Figure 5.10: Configuring the bladerf

```
cd /home/mannava/gvv/navic_L5_POC/codes/
    navic_receiver/bin
./ignss−sdrcli
```

5. After running the receiver module the obs file and nav file are the output and stored in rinex directory.For computing the position of receiver use the following command.

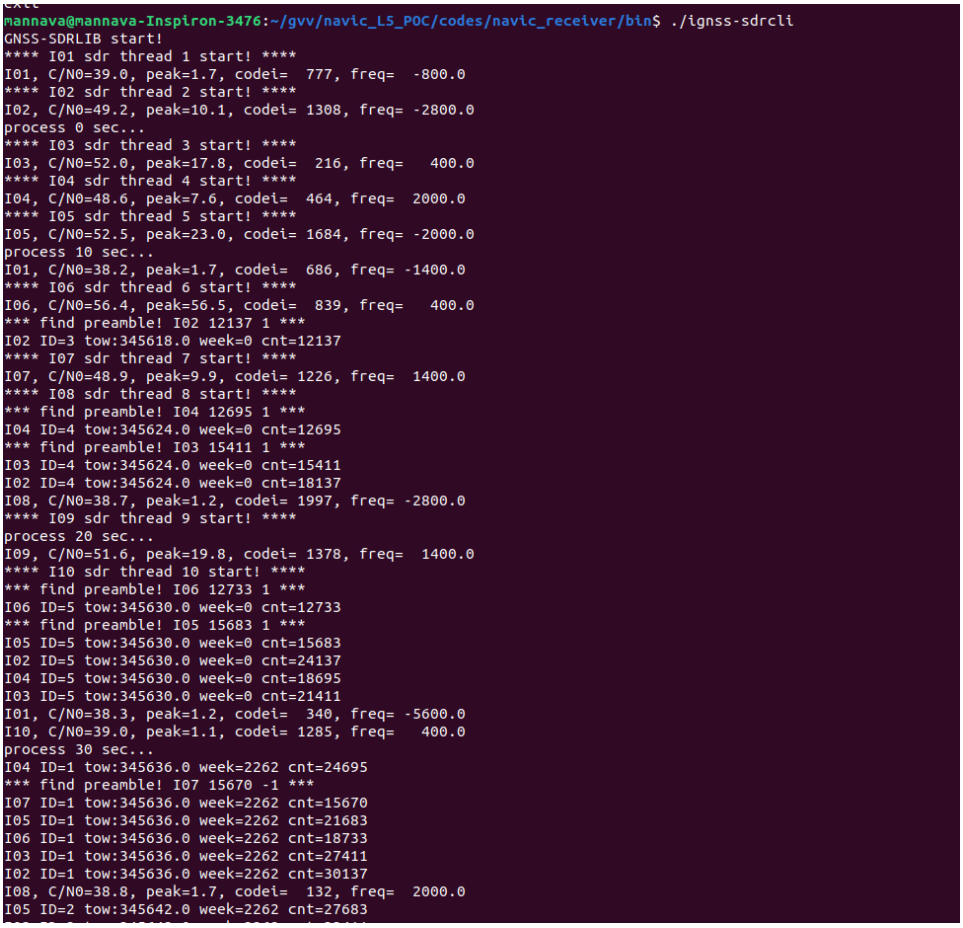
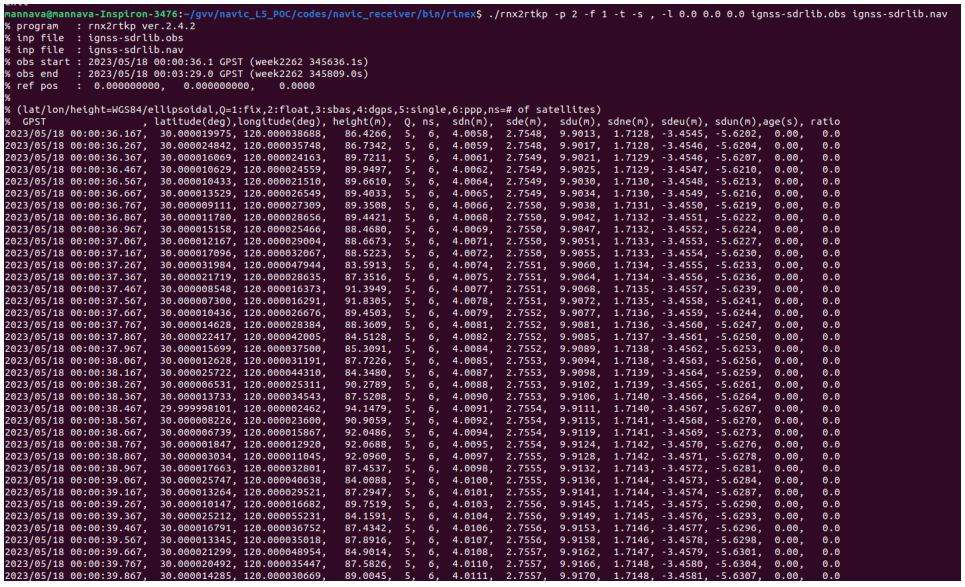


Figure 5.11: Output of receiver module

cd /home/mannava/gvv/navic_L5_POC/codes/
navic_receiver/bin/rinex
./rnx2rtkp −p 2 −f 1 −t −s , −l 0.0 0.0 0.0 ignss−sdrlib
.obs ignss−sdrlib.nav

6. By running the above command we get the PVT of the receiver.



```
nannava@nannava-Inspiron-3476:~/gvv/navic_L5_POC/codes/navic_receiver/bin/rinex$ ./rnx2rtkp -p 2 -f 1 -t -s , -l 0.0 0.0 0.0 ignss-sdrlib.obs ignss-sdrlib.nav
% program   : rnx2rtkp ver.2.4.2
% inp file  : ignss-sdrlib.obs
% inp file  : ignss-sdrlib.nav
% obs start : 2023/05/18 00:00:36.1 GPST (week2262 345636.1s)
% obs end   : 2023/05/18 00:03:29.0 GPST (week2262 345809.0s)
% ref pos   :  0.000000000,   0.000000000,    0.0000
%
% (lat/lon/height=WGS84/ellipsoidal,Q=1:fix,2:float,3:sbas,4:dgps,5:single,6:ppp,ns=# of satellites)
%  GPST                ,  latitude(deg),longitude(deg), height(m),  Q, ns,  sdn(m),  sde(m),  sdu(m), sdne(m), sdeu(m), sdun(m),age(s), ratio
2023/05/18 00:00:36.167,  30.000019975, 120.000038688,   86.4266,  5,  6,  4.0058,  2.7548,  9.9013,  1.7128, -3.4545, -5.6202,  0.00,   0.0
2023/05/18 00:00:36.267,  30.000024842, 120.000035748,   86.7342,  5,  6,  4.0059,  2.7548,  9.9017,  1.7128, -3.4546, -5.6204,  0.00,   0.0
2023/05/18 00:00:36.367,  30.000016069, 120.000024163,   89.7211,  5,  6,  4.0061,  2.7549,  9.9021,  1.7129, -3.4546, -5.6207,  0.00,   0.0
2023/05/18 00:00:36.467,  30.000010629, 120.000024559,   89.9497,  5,  6,  4.0062,  2.7549,  9.9025,  1.7129, -3.4547, -5.6210,  0.00,   0.0
2023/05/18 00:00:36.567,  30.000010433, 120.000021510,   89.6610,  5,  6,  4.0064,  2.7549,  9.9030,  1.7130, -3.4548, -5.6213,  0.00,   0.0
2023/05/18 00:00:36.667,  30.000013529, 120.000026549,   89.4033,  5,  6,  4.0065,  2.7549,  9.9034,  1.7130, -3.4549, -5.6216,  0.00,   0.0
2023/05/18 00:00:36.767,  30.000009111, 120.000027309,   89.3508,  5,  6,  4.0066,  2.7550,  9.9038,  1.7131, -3.4550, -5.6219,  0.00,   0.0
2023/05/18 00:00:36.867,  30.000011780, 120.000028656,   89.4421,  5,  6,  4.0068,  2.7550,  9.9042,  1.7132, -3.4551, -5.6222,  0.00,   0.0
2023/05/18 00:00:36.967,  30.000015158, 120.000025466,   88.4680,  5,  6,  4.0069,  2.7550,  9.9047,  1.7132, -3.4552, -5.6224,  0.00,   0.0
2023/05/18 00:00:37.067,  30.000012167, 120.000029004,   88.6673,  5,  6,  4.0071,  2.7550,  9.9051,  1.7133, -3.4553, -5.6227,  0.00,   0.0
2023/05/18 00:00:37.167,  30.000017096, 120.000032067,   88.5223,  5,  6,  4.0072,  2.7550,  9.9055,  1.7133, -3.4554, -5.6230,  0.00,   0.0
2023/05/18 00:00:37.267,  30.000031984, 120.000047944,   83.5913,  5,  6,  4.0074,  2.7551,  9.9060,  1.7134, -3.4555, -5.6233,  0.00,   0.0
2023/05/18 00:00:37.367,  30.000021719, 120.000028635,   87.3516,  5,  6,  4.0075,  2.7551,  9.9064,  1.7134, -3.4556, -5.6236,  0.00,   0.0
2023/05/18 00:00:37.467,  30.000008548, 120.000016373,   91.3949,  5,  6,  4.0077,  2.7551,  9.9068,  1.7135, -3.4557, -5.6239,  0.00,   0.0
2023/05/18 00:00:37.567,  30.000007300, 120.000016291,   91.8305,  5,  6,  4.0078,  2.7551,  9.9072,  1.7135, -3.4558, -5.6241,  0.00,   0.0
2023/05/18 00:00:37.667,  30.000010436, 120.000026676,   89.4503,  5,  6,  4.0079,  2.7552,  9.9077,  1.7136, -3.4559, -5.6244,  0.00,   0.0
2023/05/18 00:00:37.767,  30.000014628, 120.000028384,   88.3609,  5,  6,  4.0081,  2.7552,  9.9081,  1.7136, -3.4560, -5.6247,  0.00,   0.0
2023/05/18 00:00:37.867,  30.000022417, 120.000042005,   84.5128,  5,  6,  4.0082,  2.7552,  9.9085,  1.7137, -3.4561, -5.6250,  0.00,   0.0
2023/05/18 00:00:37.967,  30.000015699, 120.000037500,   85.3091,  5,  6,  4.0084,  2.7552,  9.9089,  1.7138, -3.4562, -5.6253,  0.00,   0.0
2023/05/18 00:00:38.067,  30.000012628, 120.000031191,   87.7226,  5,  6,  4.0085,  2.7553,  9.9094,  1.7138, -3.4563, -5.6256,  0.00,   0.0
2023/05/18 00:00:38.167,  30.000025722, 120.000044310,   84.3480,  5,  6,  4.0087,  2.7553,  9.9098,  1.7139, -3.4564, -5.6259,  0.00,   0.0
2023/05/18 00:00:38.267,  30.000006531, 120.000025311,   90.2789,  5,  6,  4.0088,  2.7553,  9.9102,  1.7139, -3.4565, -5.6261,  0.00,   0.0
2023/05/18 00:00:38.367,  30.000013733, 120.000034543,   87.5208,  5,  6,  4.0090,  2.7553,  9.9106,  1.7140, -3.4566, -5.6264,  0.00,   0.0
2023/05/18 00:00:38.467,  29.999998101, 120.000002462,   94.1479,  5,  6,  4.0091,  2.7554,  9.9111,  1.7140, -3.4567, -5.6267,  0.00,   0.0
2023/05/18 00:00:38.567,  30.000008226, 120.000023600,   90.9059,  5,  6,  4.0092,  2.7554,  9.9115,  1.7141, -3.4568, -5.6270,  0.00,   0.0
2023/05/18 00:00:38.667,  30.000006739, 120.000015867,   92.0486,  5,  6,  4.0094,  2.7554,  9.9119,  1.7141, -3.4569, -5.6273,  0.00,   0.0
2023/05/18 00:00:38.767,  30.000001847, 120.000012920,   92.0688,  5,  6,  4.0095,  2.7554,  9.9124,  1.7142, -3.4570, -5.6276,  0.00,   0.0
2023/05/18 00:00:38.867,  30.000003034, 120.000011045,   92.0960,  5,  6,  4.0097,  2.7555,  9.9128,  1.7142, -3.4571, -5.6278,  0.00,   0.0
2023/05/18 00:00:38.967,  30.000017663, 120.000032801,   87.4537,  5,  6,  4.0098,  2.7555,  9.9132,  1.7143, -3.4572, -5.6281,  0.00,   0.0
2023/05/18 00:00:39.067,  30.000025747, 120.000040638,   84.0088,  5,  6,  4.0100,  2.7555,  9.9136,  1.7144, -3.4573, -5.6284,  0.00,   0.0
2023/05/18 00:00:39.167,  30.000013264, 120.000029521,   87.2947,  5,  6,  4.0101,  2.7555,  9.9141,  1.7144, -3.4574, -5.6287,  0.00,   0.0
2023/05/18 00:00:39.267,  30.000010147, 120.000016682,   89.7519,  5,  6,  4.0103,  2.7556,  9.9145,  1.7145, -3.4575, -5.6290,  0.00,   0.0
2023/05/18 00:00:39.367,  30.000025212, 120.000055231,   84.1591,  5,  6,  4.0104,  2.7556,  9.9149,  1.7145, -3.4576, -5.6293,  0.00,   0.0
2023/05/18 00:00:39.467,  30.000016791, 120.000036752,   87.4342,  5,  6,  4.0106,  2.7556,  9.9153,  1.7146, -3.4577, -5.6296,  0.00,   0.0
2023/05/18 00:00:39.567,  30.000013345, 120.000035018,   87.8916,  5,  6,  4.0107,  2.7556,  9.9158,  1.7146, -3.4578, -5.6298,  0.00,   0.0
2023/05/18 00:00:39.667,  30.000021299, 120.000048954,   84.9014,  5,  6,  4.0108,  2.7557,  9.9162,  1.7147, -3.4579, -5.6301,  0.00,   0.0
2023/05/18 00:00:39.767,  30.000020492, 120.000035447,   87.5826,  5,  6,  4.0110,  2.7557,  9.9166,  1.7148, -3.4580, -5.6304,  0.00,   0.0
2023/05/18 00:00:39.867,  30.000014285, 120.000030669,   89.0045,  5,  6,  4.0111,  2.7557,  9.9170,  1.7148, -3.4581, -5.6307,  0.00,   0.0
```

Figure 5.12: PVT of the receiver

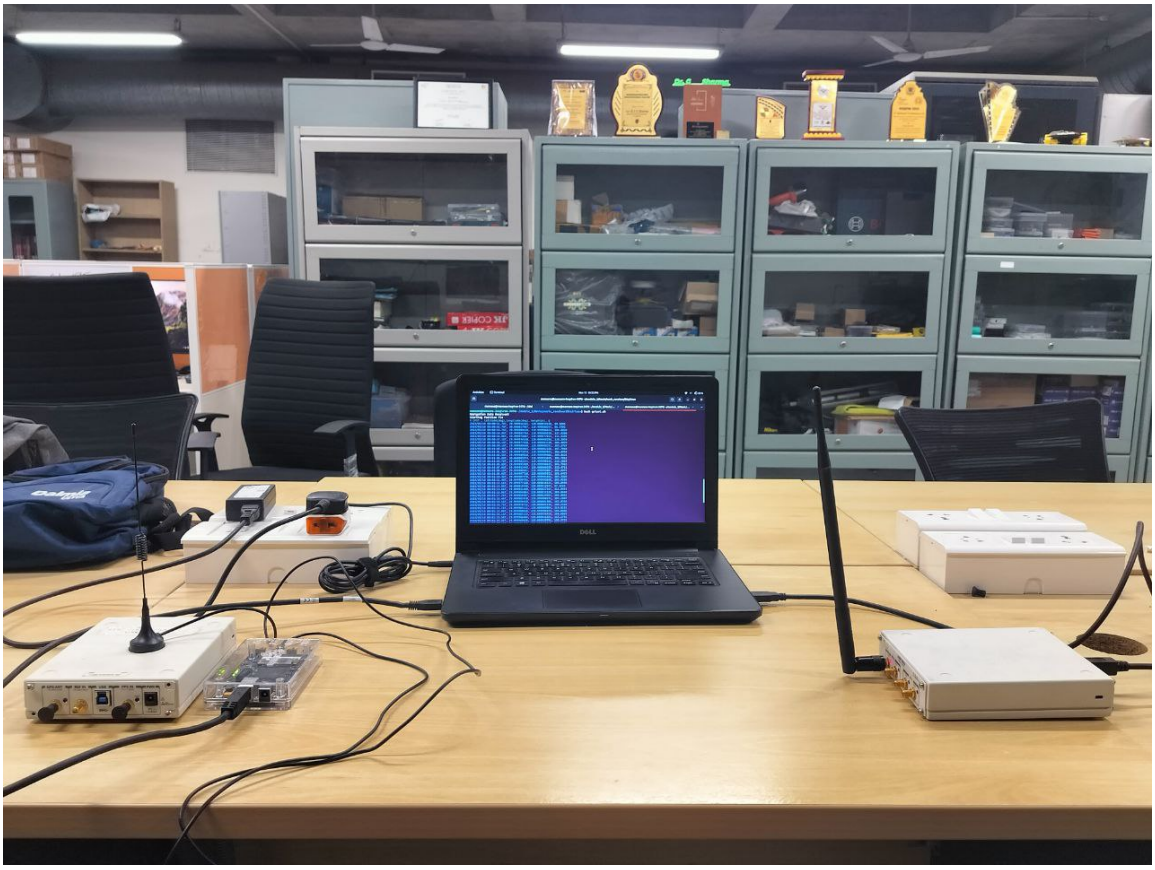


Figure 5.13: final setup

Appendix A

References

1. https://www.isro.gov.in/media_isro/pdf/Publications/Vispdf/Pdf2017/1a_messgingicd_receiver_incois_approved_ver_1.2.pdf

2. https://gnss-sdr.org/docs/sp-blocks/acquisition/

3. https://gnss-sdr.org/docs/sp-blocks/tracking/

4. Elliott D. Kaplan and Christopher J. Hegarty, Understanding GPS Principles and Applications, 3^{rd} edition

5. https://github.com/osqzss/gps-sdr-sim

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

7. https://github.com/tomojitakasu/RTKLIB