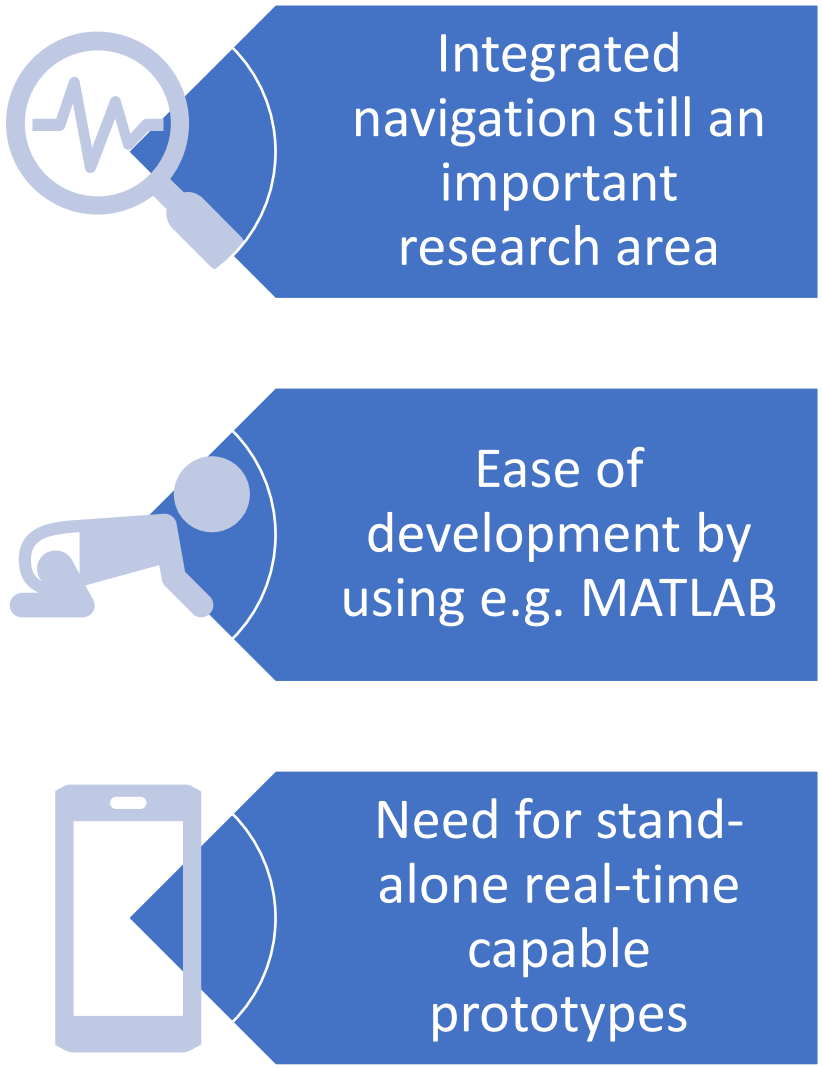
Session C2: Multisensor Integrated System Technologies

Recent Enhancements of the Multi-Sensor Navigation Analysis Tool (MuSNAT)

Markel Arizabaleta (LTE/5G), Jürgen Dampf (IQ/IF), Dominik Dötterböck (AMD Ryzen), Hepzibah Ernest (MP est), Thomas Kraus (Front-End), Thomas Pany (Stand-alone, Deep Coupling), Daniela Sanchez-Morales (LiDAR), Andreas Schütz (PPP), *Universität der Bundeswehr München*

The results presented in this work were partly developed within the projects Galileo FUSION and OPA3L funded by the German Federal Ministry for Economic Affairs and Energy (BMWi) and administered by the Project Management Agency for Aeronautics Research of the German Space Agency (DLR) in Bonn, Germany (grant no. 50NA2001 and 50NA1910)

Motivation and Paper/Presentation Contents



Contents

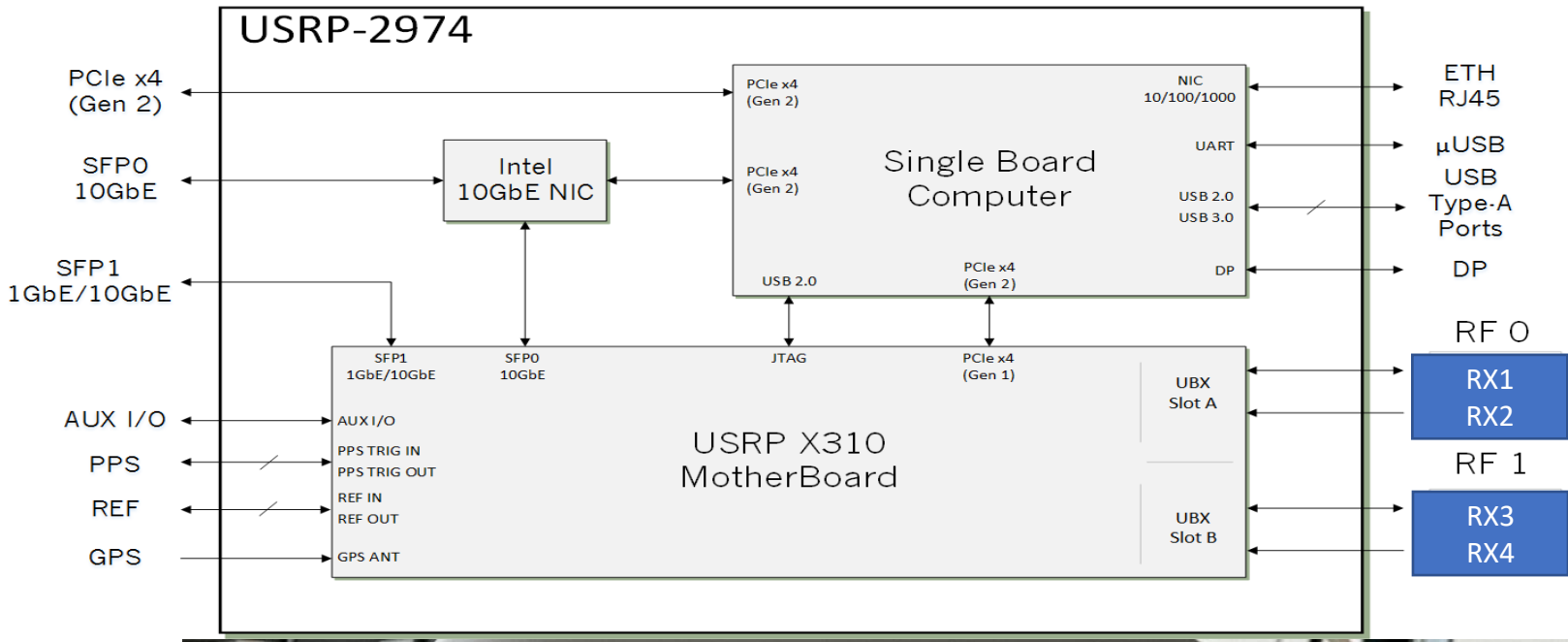
- Stand-Alone GNSS Software Receiver USRP 2974
  - Frontend, GNSS/LTE Processing Performance (real-time)
- C++/MATLAB Interface
  - PPP Module
  - LiDAR Module
  - GNSS/Deep-Coupling Interface
- Other points and future developments

Stand-Alone GNSS Software Receiver

How to use a modified USRP as generic GNSS SDR with LTE/5G and sensor data processing capabilities

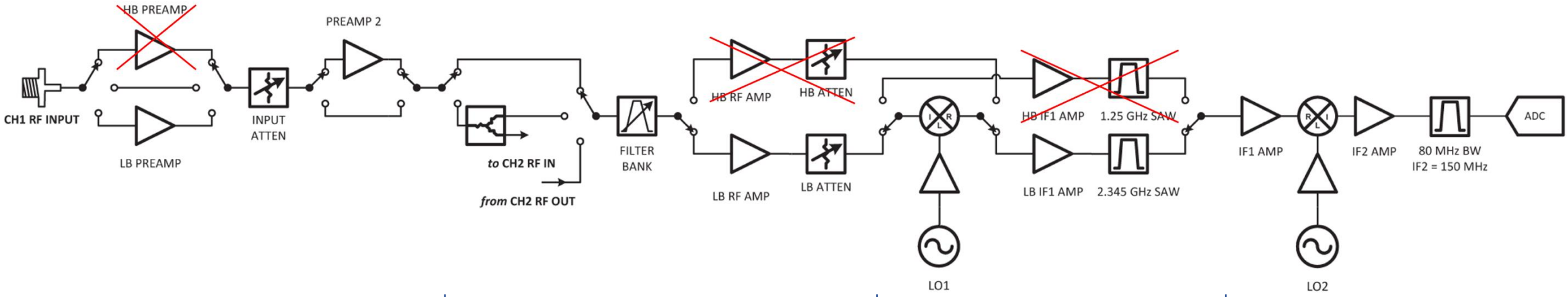
Stand alone GNSS Software Receiver

- “modified” USRP-2974
- 4 RF Channels (next slide)
  - ADC
  - 4 x PCI 2.0, 2GByte/s
- Intel Core i7-6822EQ with a release date Q4/2015
- Windows 10 OS



RF Frontend (with TwinRx daughterboards)

- superheterodyne architecture (G = 0 to 93 dB, N_f < 5 dB)
  - 4 channels, 80 MHz each, independent tuning, RF shielding, LO sharing capability (phase-aligned operation)



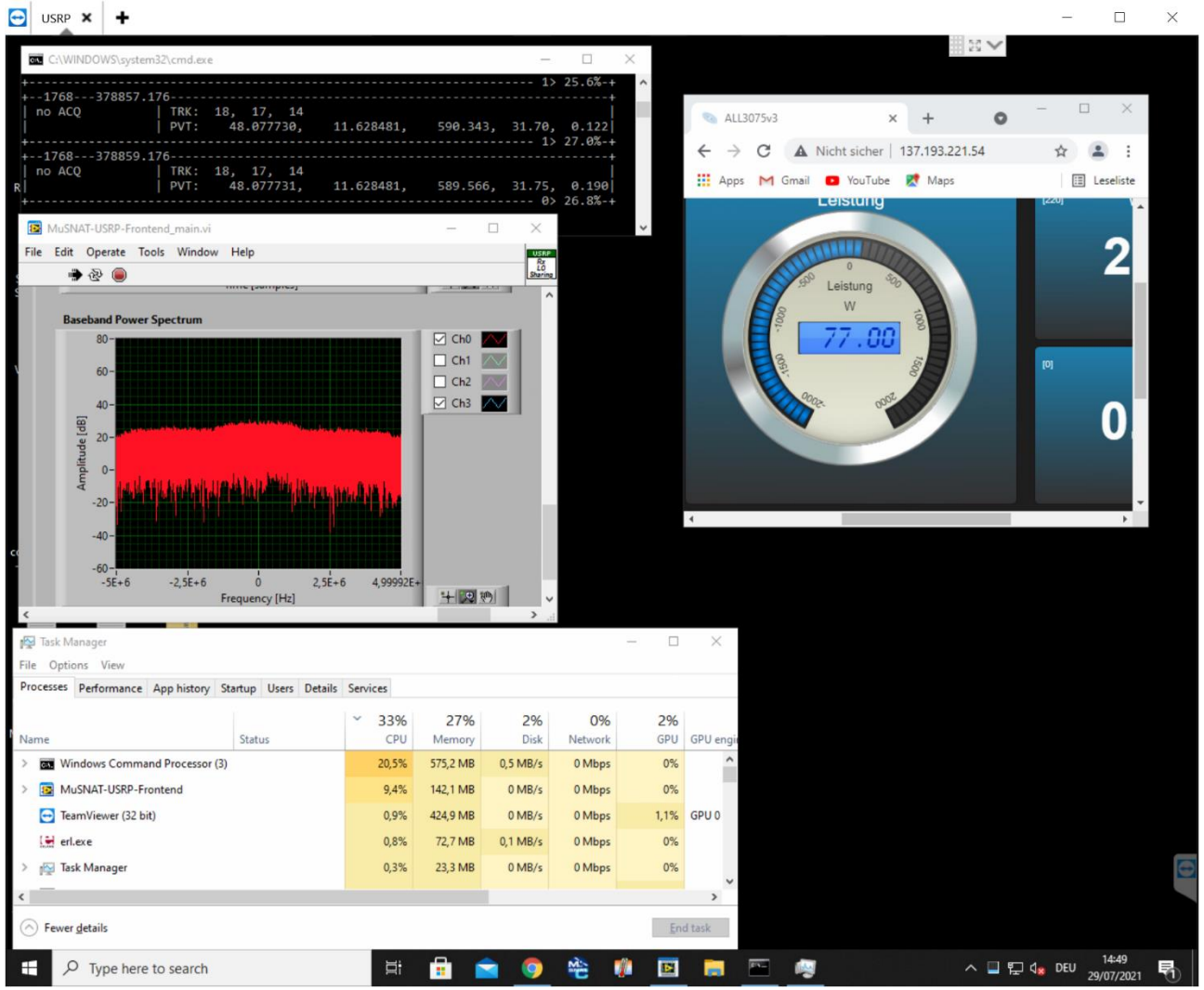
- switchable high-pass filter
- switchable 14-dB amplifier *(2x)*
  - PGA-102+ of Mini-Circuits
  - $N_f \approx 2.4$ dB
- 31-dB programmable attenuator
  - PE43503 of Peregrine Semiconductor
- f = (0.01 or) 0.8 to 6 GHz
- BW = (5.99 or) 5.2 GHz

- filter bank (4 LB configurations)
  - LB 4: f = 1.16 to 1.85 GHz, BW = 680 MHz
    - high-pass ceramic filter (HFCN-1100+)
  - LB 3: f = 0.76 to 1.24 GHz; BW = 480 MHz
    - high-pass ceramic filter (HFCN-740+)
    - low-pass ceramic filter (LFCN-1000+)
- 14-dB amplifier
- 31-dB programmable attenuator
- two high-pass ceramic filter
  - LFCN-1700+ of Mini-Circuits

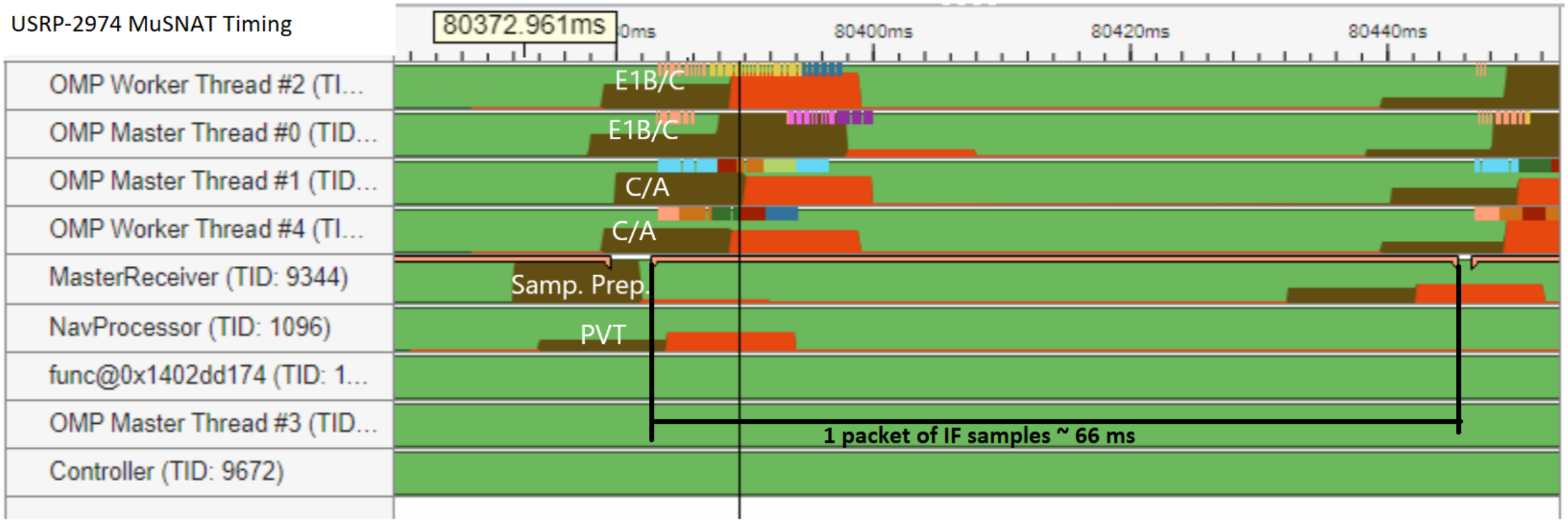
- *wide-band frequency mixer*
  - SIM-722MH+ of Mini-Circuits
- 14-dB amplifier (2x)
- SAW filter
  - TA0581A of TAI-SAW
- f_{IF} = 2.345 GHz, BW = 90 MHz

- *wide-band frequency mixer*
  - LTC 5510 of LLTC/Analog Devices
- 14-dB amplifier
- bandpass filter (NI custom)
- f_{IF} = 125 MHz, BW = 40 MHz, or
- f_{IF} = 150 MHz, BW = 80 MHz
- ADC (complex, f_s = 200 MHz)
  - ADS62P48 of Texas Instruments

Power Consumption and Real-Time Processing

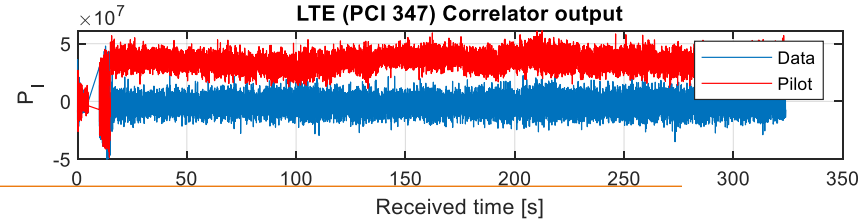


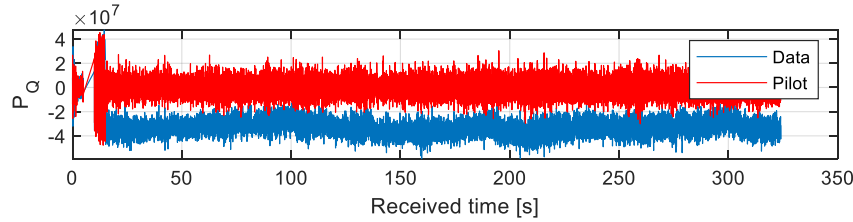
GNSS Real-Time Processing Timing

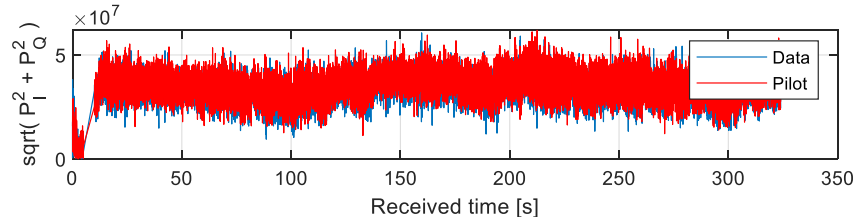


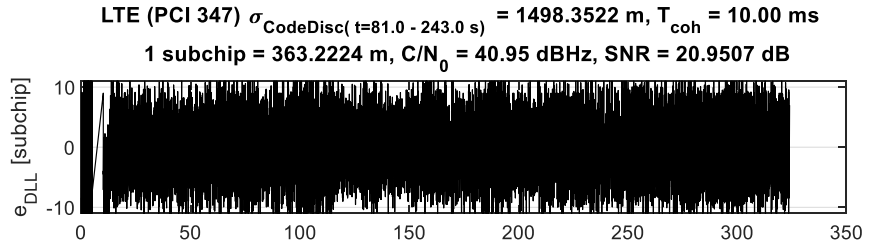
GNSS/LTE Tracking

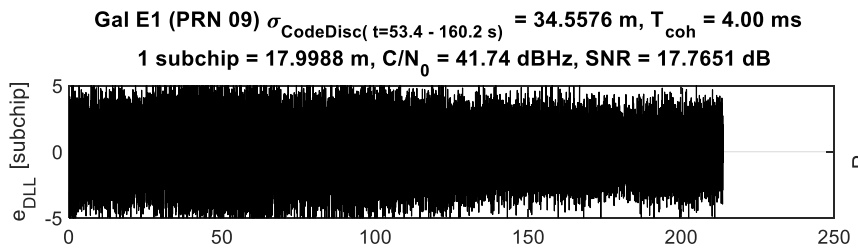
- The LTE acquisition/tracking is based on a combined PSS/SSS sequence (10 ms long)
- The combined GNSS + LTE processing is now possible
- 5G signal acquisition/tracking (and its combination with GNSS) is currently under test

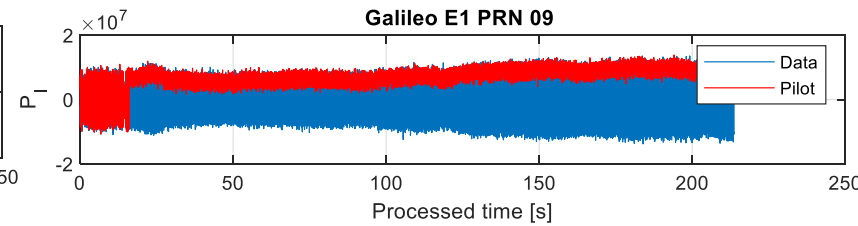


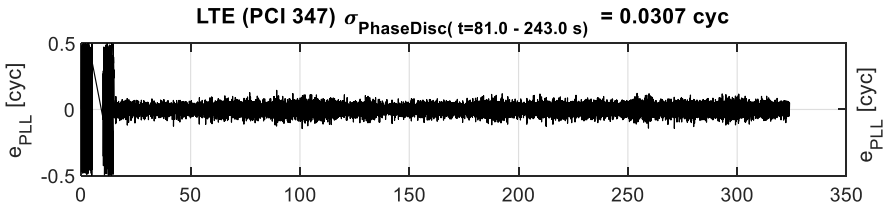


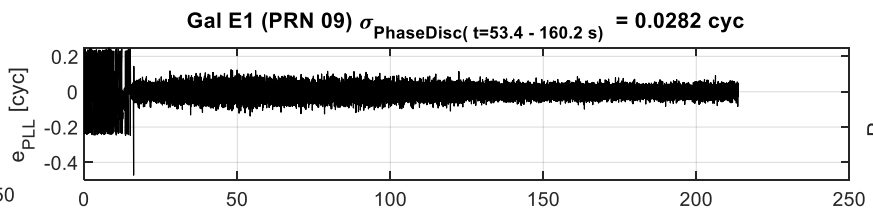


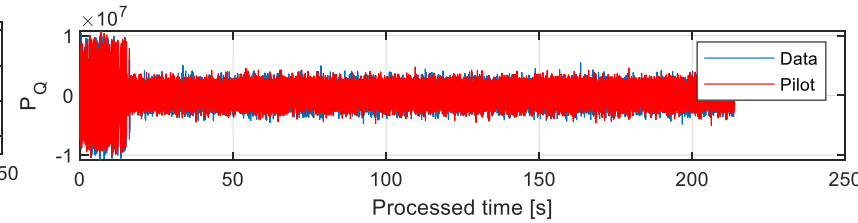


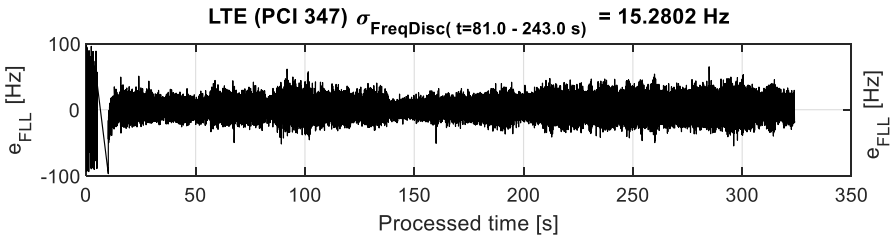


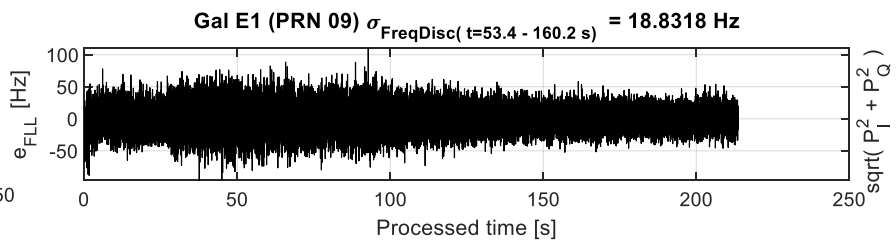


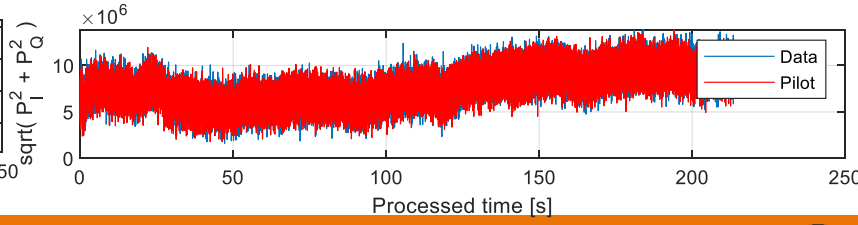












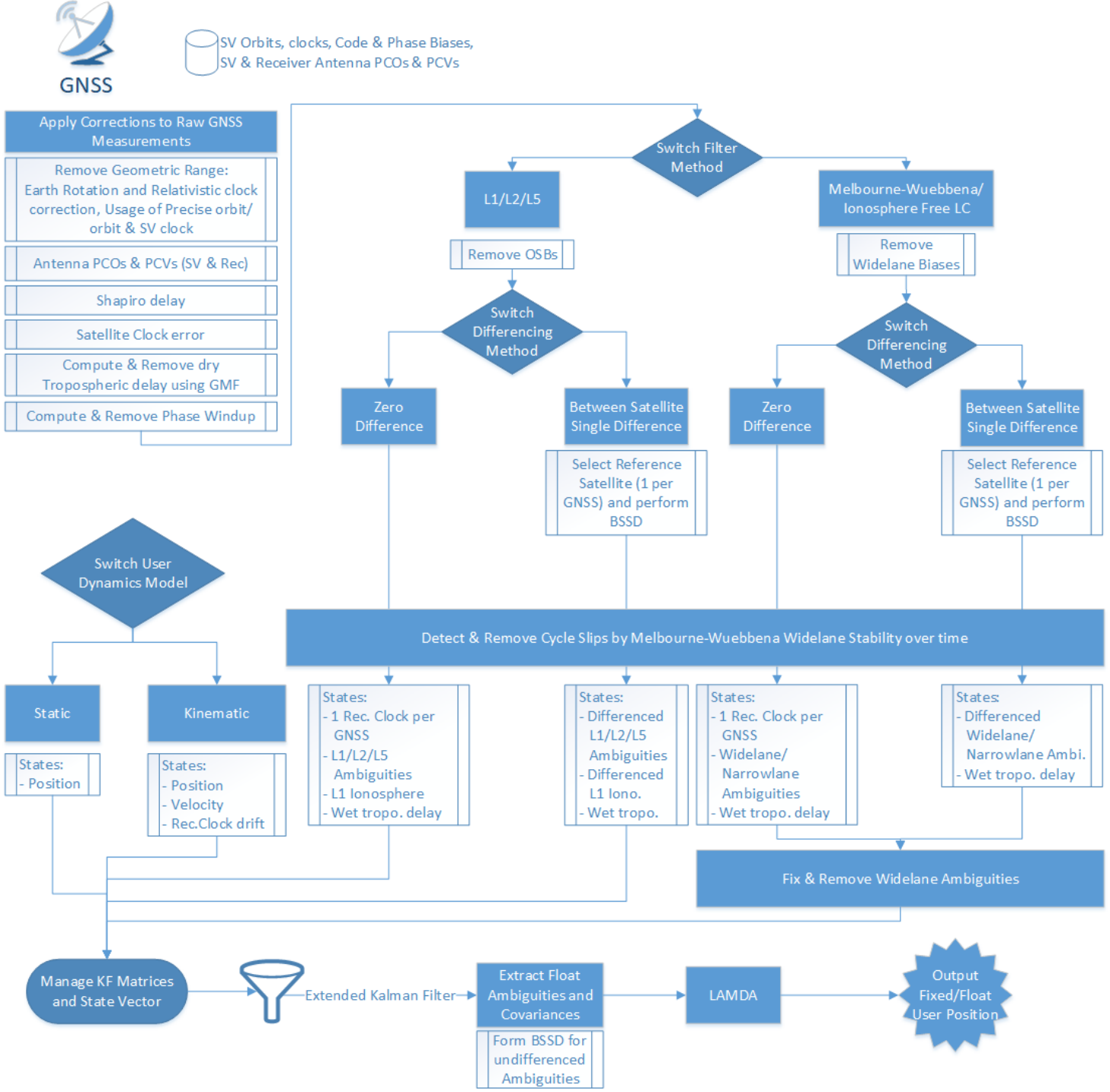
C++/MATLAB Interface

How to integrate MATLAB into the GNSS SDR (MuSNAT) C++ code to ease the development cycle

C++/MATLAB Interface

- MATLAB runs as a separate process under WINDOWS OS
- C++ header files plus libraries provided by MathWorks
- Support of various MATLAB data structures (array, cells, …) in C++
- Execution of MATLAB scripts. (call latency **a few ms**)

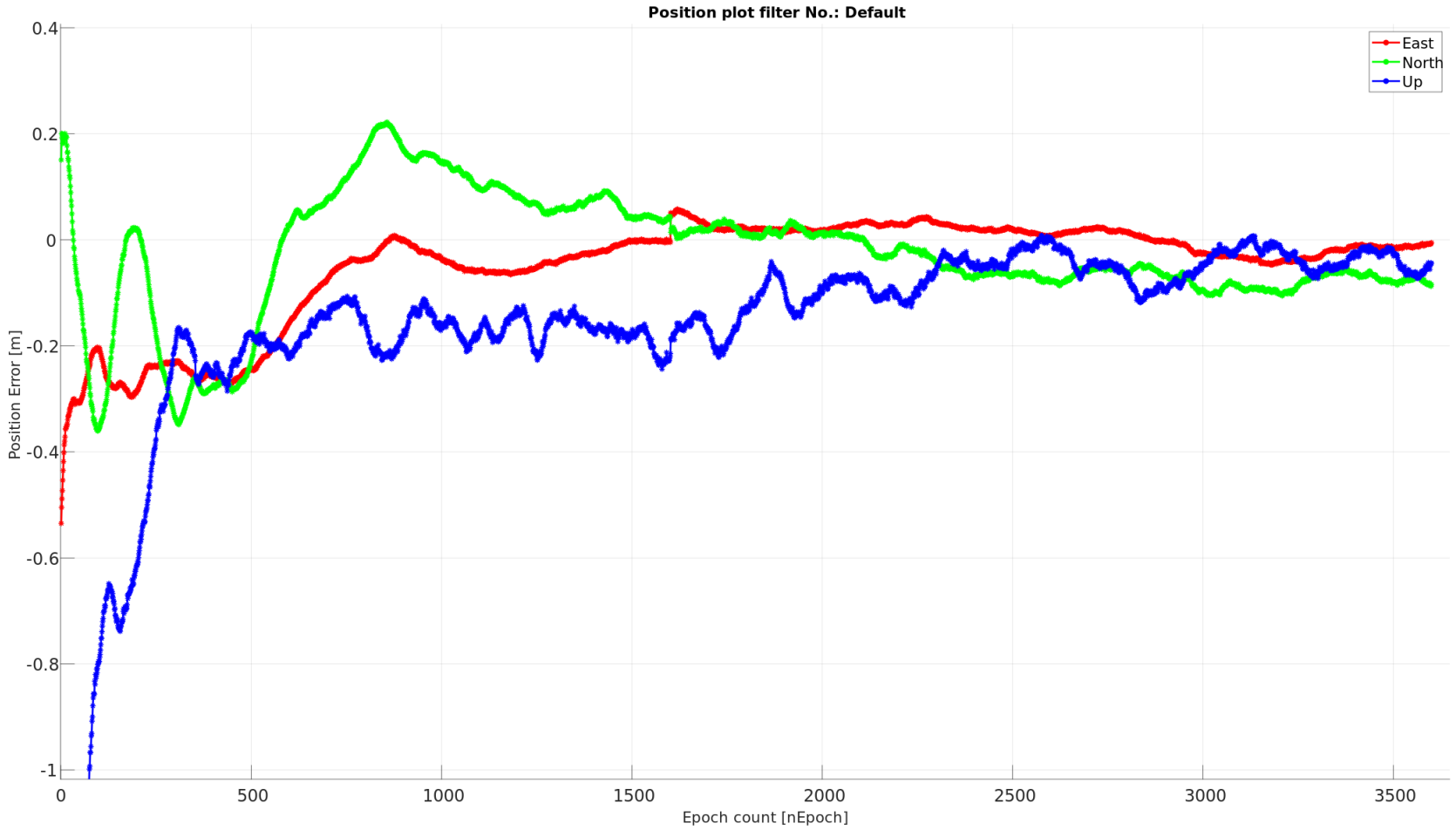
```
254         if (nImuDataCnt)
255         {
256             // create matrix for MATLAB, 0 ... time, 1..3 acc, 4..6 gyro
257             IMU_dat = __dc__factory.createArray<double>({ nImuDataCnt, 8 });
258
259             // iterate through sensor data list
260             nImuDataCnt = 0;
261             for (it = theData.listpAccSensorData.begin(); it != theData.listpAccSensorData.end(); ++it)
262             {
263                 // check if it is a IMU record and from correct IMU id (in case of multiple imus connected)
264                 if ((it->get()->nType == it->get()->MemsImu))
265                 {
266                     DSensorRecordImu* imuData = (DSensorRecordImu*)it->get();
267
268                     IMU_dat[nImuDataCnt][1] = double(imuData->myDat.vfAcc[0]);
269                     IMU_dat[nImuDataCnt][2] = double(imuData->myDat.vfAcc[1]);
270                     IMU_dat[nImuDataCnt][3] = double(imuData->myDat.vfAcc[2]);
271
272                     IMU_dat[nImuDataCnt][4] = double(imuData->myDat.vfGyro[0]);
273                     IMU_dat[nImuDataCnt][5] = double(imuData->myDat.vfGyro[1]);
274                     IMU_dat[nImuDataCnt][6] = double(imuData->myDat.vfGyro[2]);
275
276                     IMU_dat[nImuDataCnt][7] = double(imuData->myDat.nSampleCounter);
277                     imuData->ptRefTime.toGPS(gpsWeek, gpsSec);
278                     IMU_dat[nImuDataCnt][0] = gpsSec;
279
280                     nImuDataCnt++;
281                 }
282             }
283         }
284     }
285
286     // Pass data to Matlab
287     try
288     {
289         if (nImuDataCnt) __dc__matlabPtr->setVariable(u"curImuDat", std::move(IMU_dat));
290         __dc__matlabPtr->setVariable(u"curSatStruct", std::move(satStruct));
291         __dc__matlabPtr->eval(u"dcProcessEpoch;", {}, __dc__error);
292         dcPos = __dc__matlabPtr->getVariable(u"curElemToExtract");
293     }
294     catch (...)
295     {
296         String error_ = matlab::engine::convertUTF16StringToUTF8String((__dc__error.get()->str()));
297         String str;
298         str.format("DeepCouplingMatlabInterface::process -> MATLAB error '%s'", matlab::engine::convertUTF16S
299         throw SoftRecError(SoftRecError::SoftRecErrorMatlabInterface, error_.c_str());
300     }
301
```

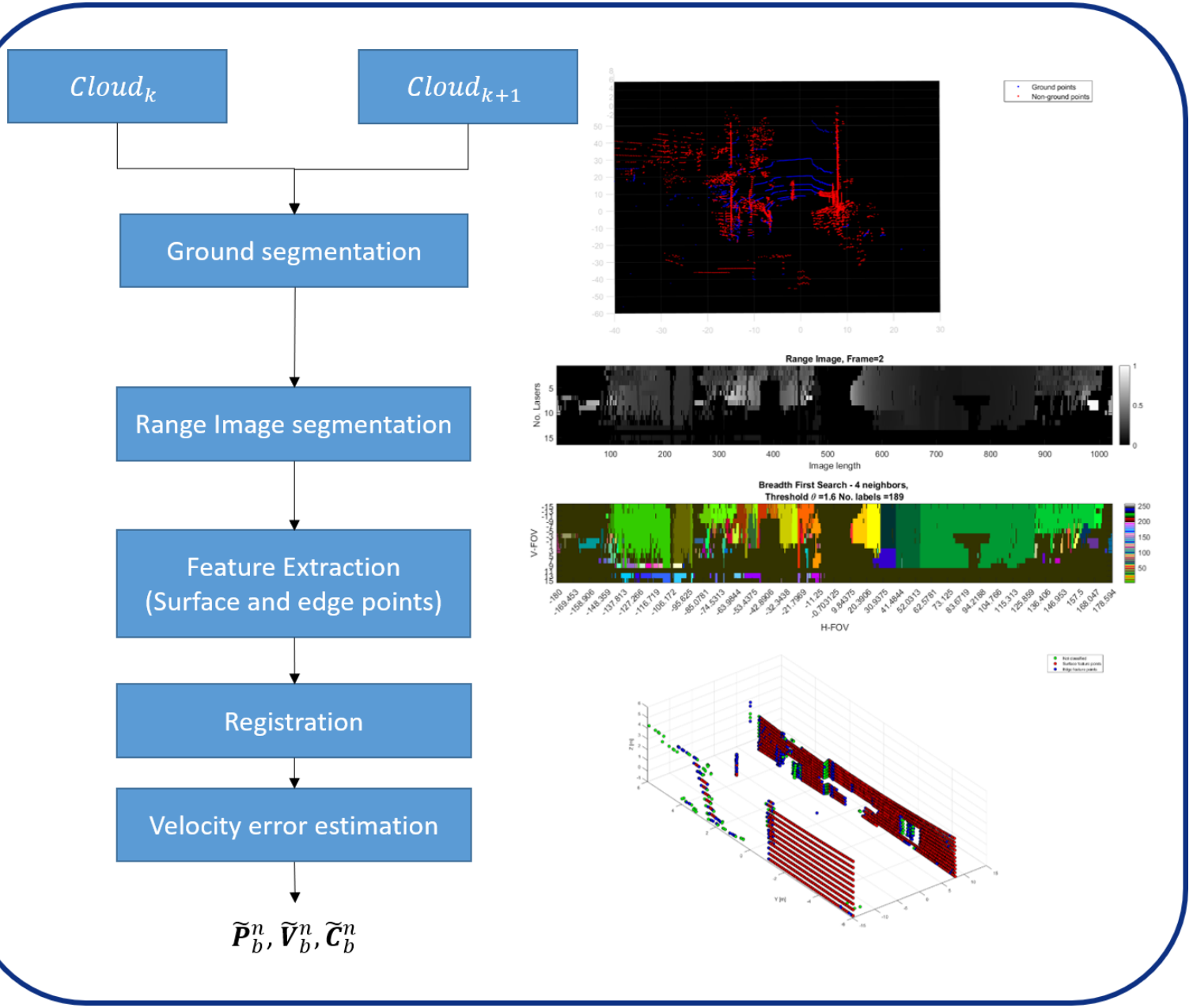


MATLAB PPP Processing

- Support of GPS L1/L2 Gal E1/E5a
- Undifferenced/BSSD
- Uncombined/ Widelane – Narrowlane
- Static/Kinematic Processing
- Cycle Slip detection using MW Measurements
- Realtime capable
- Convenient implementation (KF state handling), testing and development, use of built-in MATLAB functions;
- Processing Speed slower than pure C++

PPP, typical result



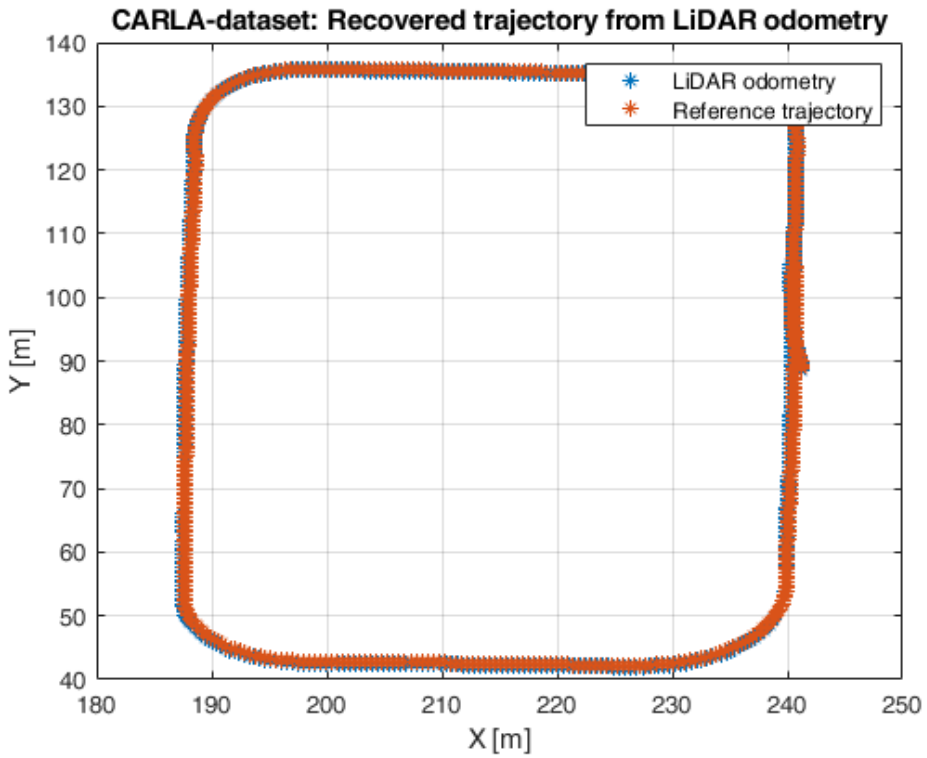


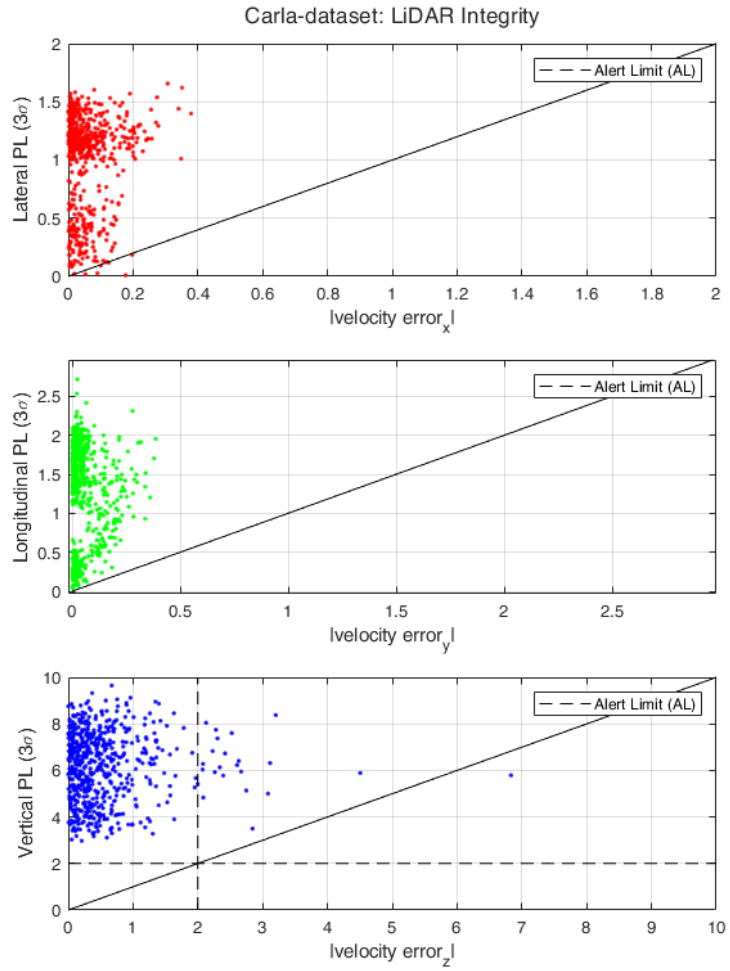
MATLAB LiDAR Processing

- Support of PCAP LIDAR data stream for VLP16 and synthetic data generated with CARLA
- Implementation of state of the art algorithms, specific designed for real time applications (most of them) and that can handle sparse 3D point clouds.
- The LiDAR update consist in the position, velocity and velocity error estimate.

- Facilitates evaluation, testing and visualization of the individual processing modules as it can run independently from the MuSNAT.
- It also accelerates integration with the MuSNAT as the connection with MATLAB is straight forward.
- Integration/testing of opensource libraries may be difficult to include within the matlab environment.

LiDAR, typical results

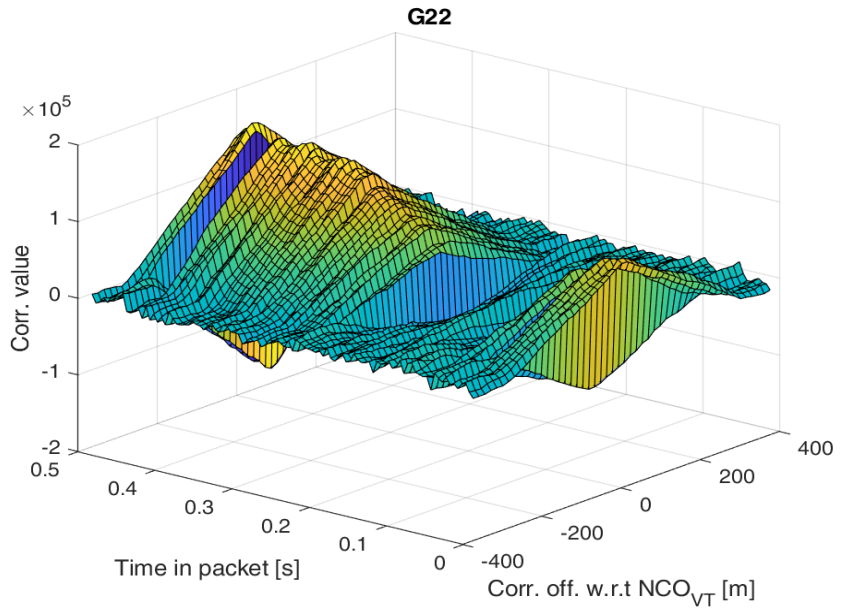




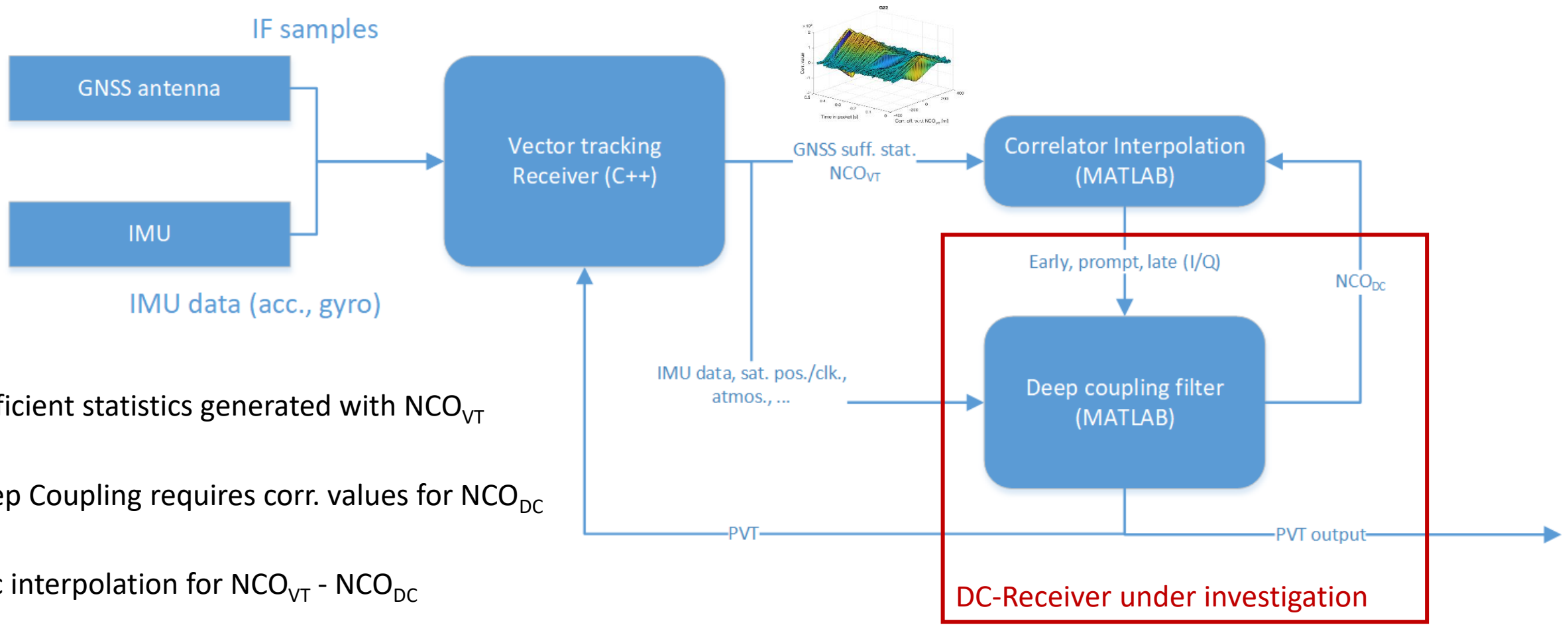
Deep Coupling C++/MATLAB Interface

- GNSS/INS Deep Coupling
  - NCO values, correlator values
  - IMU strapdown and Kalman filter
  - High bidirectional rate
    - 50 Hz … 1 kHz
- Rate incompatible for direct use of C++/MATLAB interface due to call latency
- Another solution required

- Sufficient statistics
  - I.e. multi-correlator value
  - contain full GNSS signal contents
- Batch-wise generation passing to MATLAB for all signals and satellites (e.g. every 0.5) -> latency problem solved
- “sinc”-interpolation to get actual values



Deep Coupling C++/MATLAB Interface

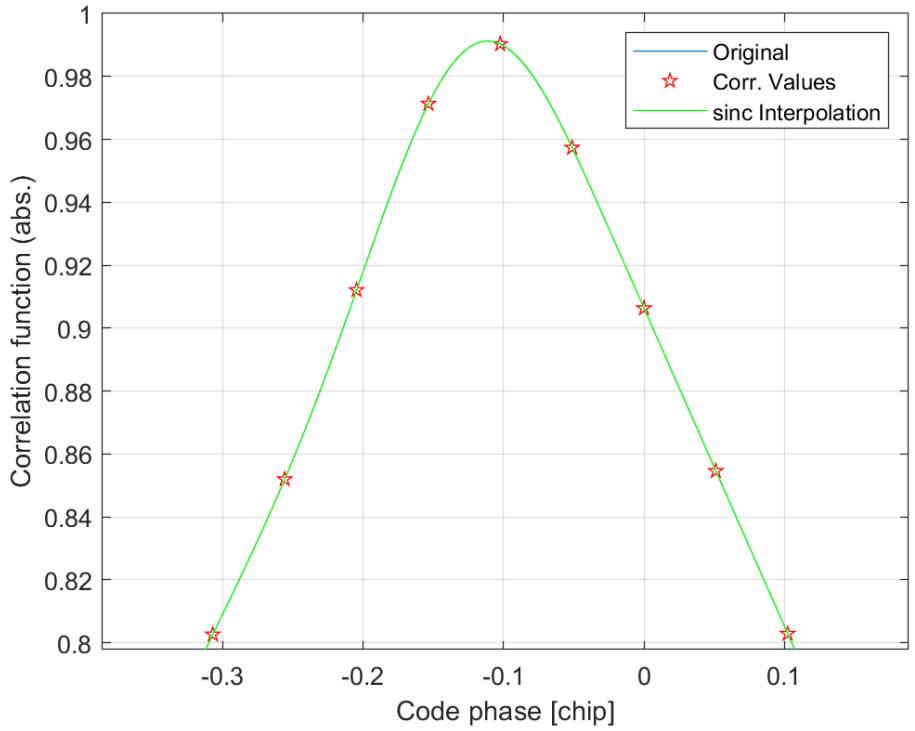


Sufficient statistics generated with NCO_{VT}

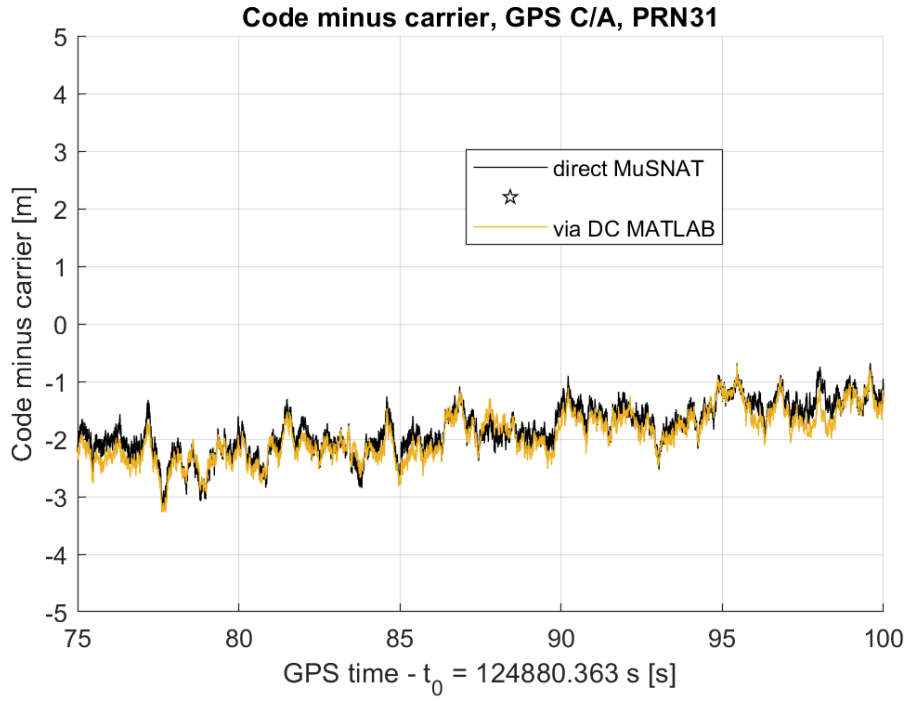
Deep Coupling requires corr. values for NCO_{DC}

sinc interpolation for NCO_{VT} - NCO_{DC}

Deep Coupling C++/MATLAB Interface



Validation of sinc-interpolation



Validation of interface with
standard (scalar) DLL/PLL are real signals

Other points and Outlook

Further points in the paper

- Performance on high-end PCs/laptops
- IQ/IF conversion
  - To support various input formats
  - Loss-less FFT based conversion
  - Benefits for multipath estimating algorithms (and C/N0)

Outlook

- Bit decimation and band selection on USRP2974 FPGA
- Synchronized IMU/GNSS data collection on USRP 2974 FPGA
- Off-loading of GNSS/LTE/5G correlation to GPU within Deep Coupling Interface (low latency)
- Open-Source policy for MATLAB code (TBC)