Implementation of GNSS Signal Processing into the GNSS-SDR platform during GSoC 2017

Damian Miralles
University of Colorado Boulder
Outline

Overview of Google Summer of Code (GSoC)
  • GSoC 2017 edition

GNSS-SDR

VOLK
  • Intrinsic development

GNSS-SDR GLONASS
  • Acquisition
  • Tracking
  • Decoding
  • Navigation Solution

Summary / Future Work

References

11/10/17
Google Summer of Code (GSoC) is a global program focused on bringing more student developers into open source software development*

Students work with an open source organization on a 3 month programming project during their summer

13,000+ students, 104 countries, 12 years, 607 open source organizations

11/10/17
Google Summer of Code 2017

Coding Dates: May 30th - August 29th

1,318 students accepted from 72 countries

1,647 mentors with active projects from 69 countries

439 registered mentors

198 open source organizations

86.2% overall success rate
Outline

Overview of Google Summer of Code (GSoC)
  • GSoC 2017 edition

GNSS-SDR

VOLK
  • Intrinsic development

GNSS-SDR GLONASS
  • Acquisition
  • Tracking
  • Decoding
  • Navigation Solution

Summary / Future Work

References
GNSS-SDR is an open source project that implements a global navigation satellite system software defined receiver in C++*.

GNSS-SDR proposes a software architecture that builds upon the GNU Radio framework in order to implement its receiver.

GNSS-SDR is one of the member organizations of GSoC. Each year defines set of ideas list to be completed during the summer:

- Updated some of the VOLK intrinsic for faster processing
- Added GLONASS processing to the platform

11/10/17
*Signal Source* block that injects a continuous stream of raw samples of GNSS signal to the processing flow graph.

*Signal Conditioner* block is in charge of adapting the sample bit depth to a data type tractable at the host computer running the software receiver.

Data Type Adapter block performs a conversion of the data type in the incoming sample stream.

Input Filter block filters the incoming signal.

Resampler block resamples the signal and to deliver it to the NN parallel processing channels.
Channel* encapsulates blocks for signal acquisition, tracking and data decoding.

Observables* block: collects the synchronization data coming from all the processing Channels, and computes from them the GNSS basic measurements.

PVT* block: computes navigation solutions and deliver information in adequate formats for further processing or data representation.
The code defines interfaces for the receiver’s processing blocks. Interfaces allow to define several algorithm and implementations for each processing block. Hierarchy models offers maximum flexibility and reconfiguration.
Outline

Overview of Google Summer of Code (GSoC)
  • GSoC 2017 edition

GNSS-SDR

VOLK
  • Intrinsic development

GNSS-SDR GLONASS
  • Acquisition
  • Tracking
  • Decoding
  • Navigation Solution

Summary

References
VOLK Intrinsic

The Vector-Optimized Library of Kernels (VOLK)* is a free library, that contains kernels of hand-written SIMD code for different mathematical operations.

How does VOLK work?

• For each architecture or platform that a developer wishes to vectorize for, a new proto-kernel is added to VOLK.
• At runtime, VOLK will select the correct and most efficient proto-kernel.
• The users of VOLK call a kernel for performing the operation that is platform/architecture agnostic.

GNSS-SDR created a VOLK module that targets common operations in GNSS-SDR technologies, i.e. correlations.
VOLK Intrinsic

Intrinsics Instructions are C style functions that provide access to many Intel instructions without the need of assembly code.

VOLK libraries were in need of support for the new extensions like AVX2
- Adds 256 bit register support for integer operations
- Fused Multiply Add (FMA) operations

Work focused on the addition of new protokernels supporting the new extensions.
- Both GNSS-SDR and GNURadio community will benefit from the latest addition.

<table>
<thead>
<tr>
<th>Technology</th>
<th>Register</th>
<th>Integers</th>
<th>Floats</th>
<th>ParAl</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSEn</td>
<td>128</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>AVX</td>
<td>256</td>
<td>Yes (128)</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>AVX2</td>
<td>256</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>AVX512</td>
<td>512</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>
VOLK Intrinsic

More than 15 protokernels where updated

- Changed mainly focused in AVX and AVX2 addition

Cumulative time reduction for updated kernels of 125 ms

11/10/17
Outline

Overview of Google Summer of Code (GSoC)
  • GSoC 2017 edition

GNSS-SDR

VOLK
  • Intrinsic development

GNSS-SDR GLONASS
  • Acquisition
  • Tracking
  • Decoding
  • Navigation Solution

Summary

References
LONASS Constellation

Three orbital planes at 64.8 degree inclination (GPS uses six planes at 55 degrees)
- 24 satellites, 8 per each plane
- MEO, ~19100 km altitude
- The inclination of the orbits is ideal to ensure good coverage of polar latitudes
- Eccentricity of ~0

Orbital period = 11 hours 15 min and 28 sec
- Ground track repeats every 7 days, 23 hours, 27 minutes and 28 seconds

As of 11-Sept-2017:
- 23 satellites in operation
- 1 satellite (k=-5) in flight test
- 1 satellite under check by contractor
GLONASS Time, Coordinate System & Freq. Plan

Uses PZ-90 (Parametry Zemli 1990 Goda or Parameters Of the Earth Year)
- Originally expressed in the Soviet Geodetic System 1985 (SGS-85)
- GPS employs WGS84

GLONASS Time is relative to UTC(SU) time
- Accounts for leap seconds
- Fixed offset from UTC

GLONASS uses FDMA for its ranging code
- \( f_{L1} = 1602 + k \times 0.5625 \, MHz \)
- \( f_{L2} = 1246 + k \times 0.4375 \, MHz \)

4 satellites and only 14 frequency slots allocated per signal
- Satellites sharing same frequency values are in antipodal positions
## GLONASS Signal Plan

<table>
<thead>
<tr>
<th>Parameter</th>
<th>GLONASS L1</th>
<th>GLONASS L2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Name</td>
<td>C/A Code</td>
<td>C/A Code</td>
</tr>
<tr>
<td>Central Frequency</td>
<td>1602 MHz + k*562.5 kHz</td>
<td>1246 MHz + k*437.5 kHz</td>
</tr>
<tr>
<td>Access Technique</td>
<td>FDMA</td>
<td>FDMA</td>
</tr>
<tr>
<td>Spreading Mod.</td>
<td>BPSK</td>
<td>BPSK</td>
</tr>
<tr>
<td>Code Freq.</td>
<td>0.511 MHz</td>
<td>0.511 MHz</td>
</tr>
<tr>
<td>Code Length</td>
<td>511</td>
<td>511</td>
</tr>
<tr>
<td>Meander Sequence</td>
<td>100 Hz</td>
<td>100 Hz</td>
</tr>
<tr>
<td>Data Rate</td>
<td>50 bps</td>
<td>50 bps</td>
</tr>
<tr>
<td>Navigation Message</td>
<td>GNAV</td>
<td>GNAV</td>
</tr>
</tbody>
</table>

11/10/17
GLONASS L1 C/A Acquisition

PCPS Acquisition by means of FFT implemented

GLONASS Acquisition loops through frequency channels

- Same ranging code spread through 14 frequency channels

11/10/17
GLONASS L1 C/A Tracking

Diagram of GLONASS L1 C/A Tracking system.
GLONASS L1 C/A Tracking

Discrete-Time Scatter Plot

Bits of the navigation message

Correlation results

Raw PLL discriminator

Filtered PLL discriminator

Raw DLL discriminator

Filtered DLL discriminator

Carrier to Noise Ratio

Carrier Freq

Code Freq
GLONASS L1 C/A Telemetry Decoding

Symbol to bit conversion
- Relative data rate happens at 100 Hz (meander sequence)
- Transformation from bi-binary to relative code
- Transformation from relative to delayed code
- Transition from frequency channel (K) to satellite slot number
- In cold start applications user does not know slot number of transmitting satellite

Decoded GLONASS navigational data or GNAV is shared among L1 and L2 carrier
- Single decoder will suffice for both implementations, L1 and L2
GLONASS L1 C/A Observables

Decoded GNAV parameters allow for GLONASS to GPST conversion

GLONASS Observables time tagged in GPST
- Time standard translation enables integration with other constellations
GLONASS L1 C/A PVT

Position Solution using GLONASS L1 only measurements

Position Solution in ‘Single’ mode of operation. See RTKLib details

- Average errors ~ 10 m
Outline

Overview of Google Summer of Code (GSoC)
  • GSoC 2017 edition

GNSS-SDR

VOLK
  • Intrinsic development

GNSS-SDR GLONASS
  • Acquisition
  • Tracking
  • Decoding
  • Navigation Solution

Summary / Future Work

References
Summary/ Future Work

Add support for Intel Intrinsic AVX512 to VOLK and its GNSS-SDR modules

- AVX512 register size of 512 bits offer tremendous possibilities for speed up
- Processors will AVX512 will have more cores per CPU increasing chance of functional channels

Combine GLONASS L1 C/A position solution with existent signals in the platform like GPS L1, Galileo E1B, etc

Add GLONASS L2 C/A to GNSS-SDR

- Minimal changes in the addition since GNAV message is shared in both signals
References


Questions?

Thank You