gr-dvbs2rx

An overview of the project state and path forward.

Igor Freire

Sept 26 2022 - GRCon 22 - Washington, D.C.

Project Overview

Project Overview

- GNU Radio OOT module containing:
  - DVB-S2 signal processing blocks.
  - Example flowgraphs.
  - Production-ready Python apps: dvbs2-rx and dvbs2-tx.

Project Overview

- GNU Radio OOT module containing:
  - DVB-S2 signal processing blocks.
  - Example flowgraphs.
  - Production-ready Python apps: dvbs2-rx and dvbs2-tx.
- **Previous work** (since 2018): FEC and upper layers.
  - Ron Economos and Ahmet Inan.

Project Overview

- GNU Radio OOT module containing:
  - DVB-S2 signal processing blocks.
  - Example flowgraphs.
  - Production-ready Python apps: dvbs2-rx and dvbs2-tx.
- **Previous work** (since 2018): FEC and upper layers.
  - Ron Economos and Ahmet Inan.
- **New work** (released in 2021)**:** PHY, apps, examples, documentation, and improvements.

Project Overview

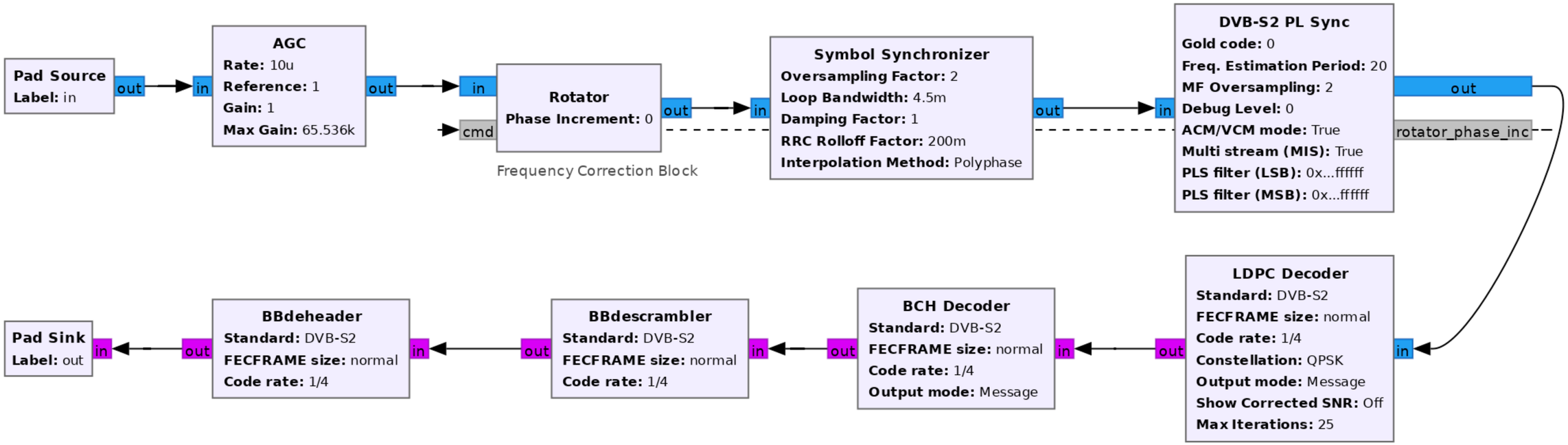
- GNU Radio OOT module containing:
  - DVB-S2 signal processing blocks.
  - Example flowgraphs.
  - Production-ready Python apps: dvbs2-rx and dvbs2-tx.
- **Previous work** (since 2018): FEC and upper layers.
  - Ron Economos and Ahmet Inan.
- **New work** (released in 2021)**:** PHY, apps, examples, documentation, and improvements.
- Fully-functional software-defined DVB-S2 receiver.

Outline

- Project state.
- Path forward for optimization and improvements.
- Examples based on the Blockstream Satellite Ku-band signal.
- CPU performance results.

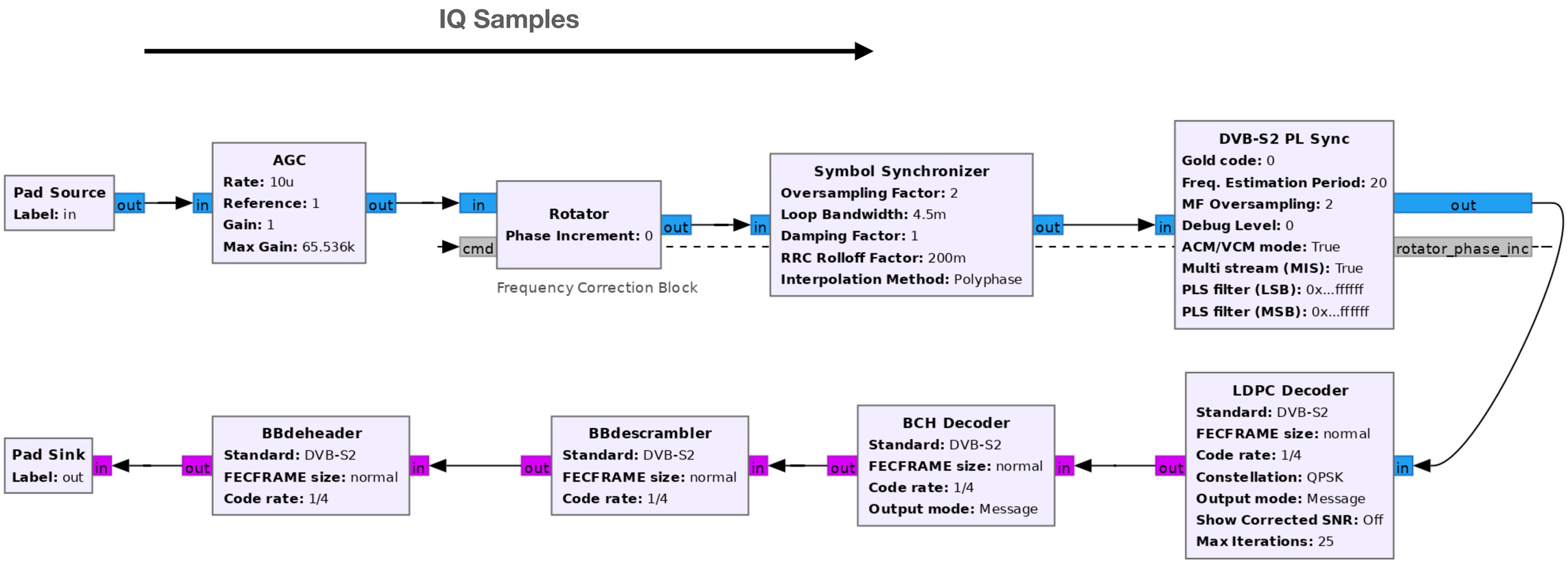
DVB-S2 Rx Architecture

Block Diagram



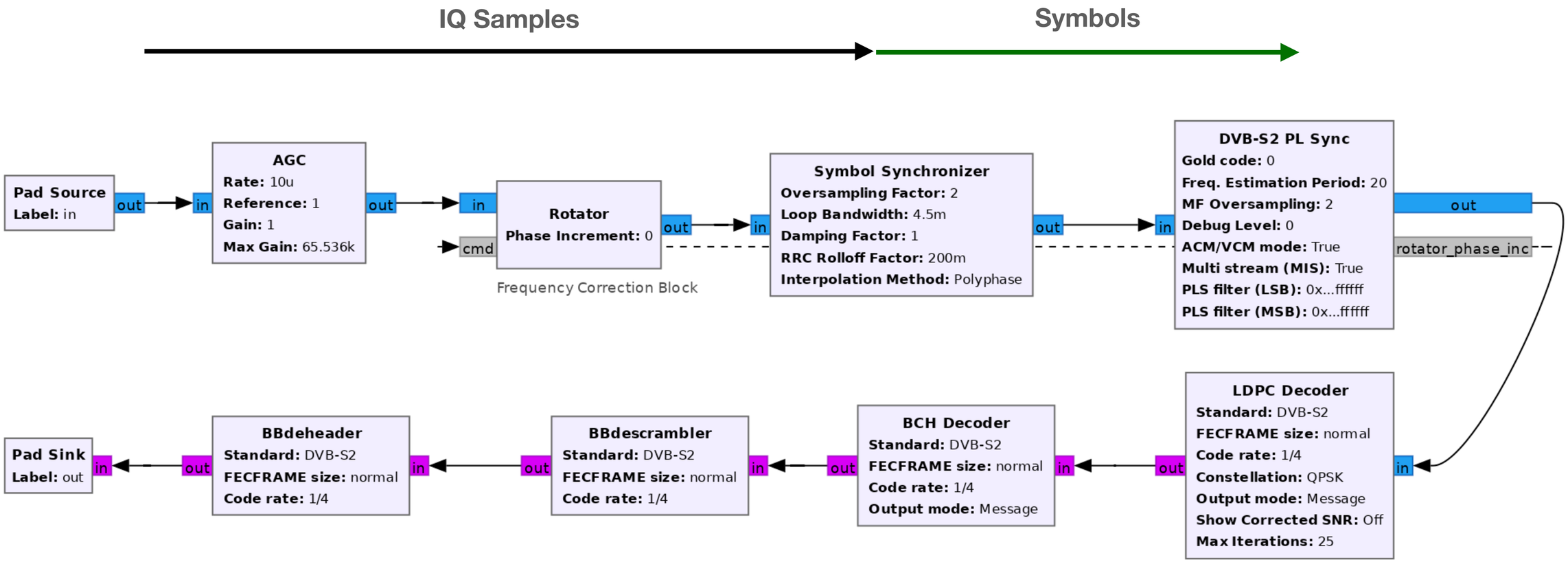
DVB-S2 Rx Architecture

Block Diagram



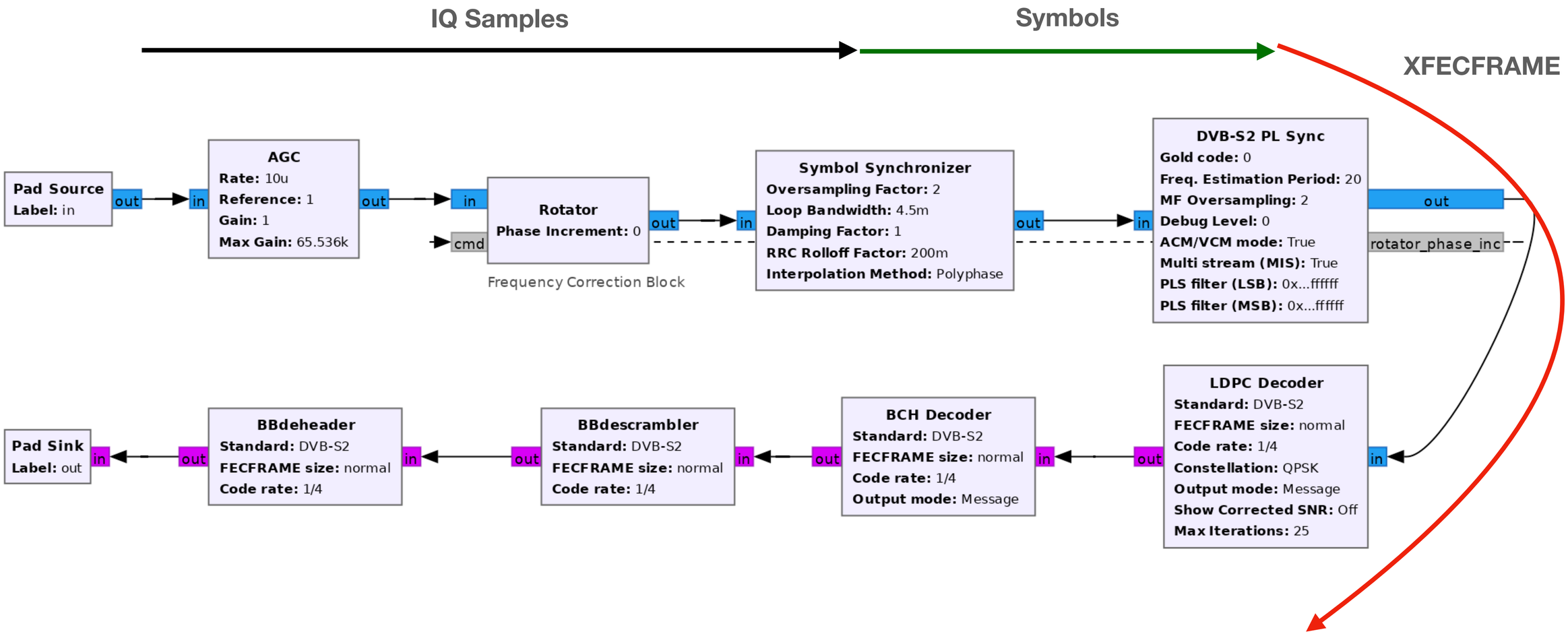
DVB-S2 Rx Architecture

Block Diagram



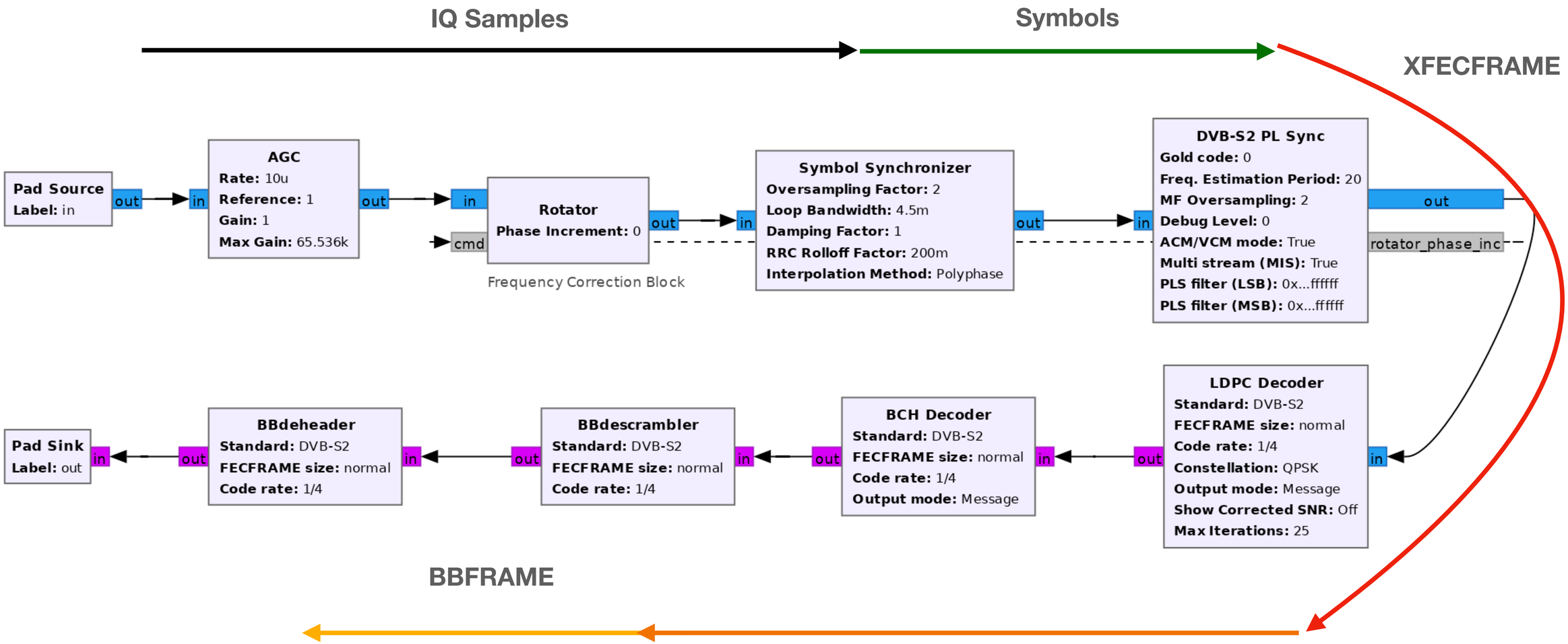
DVB-S2 Rx Architecture

Block Diagram



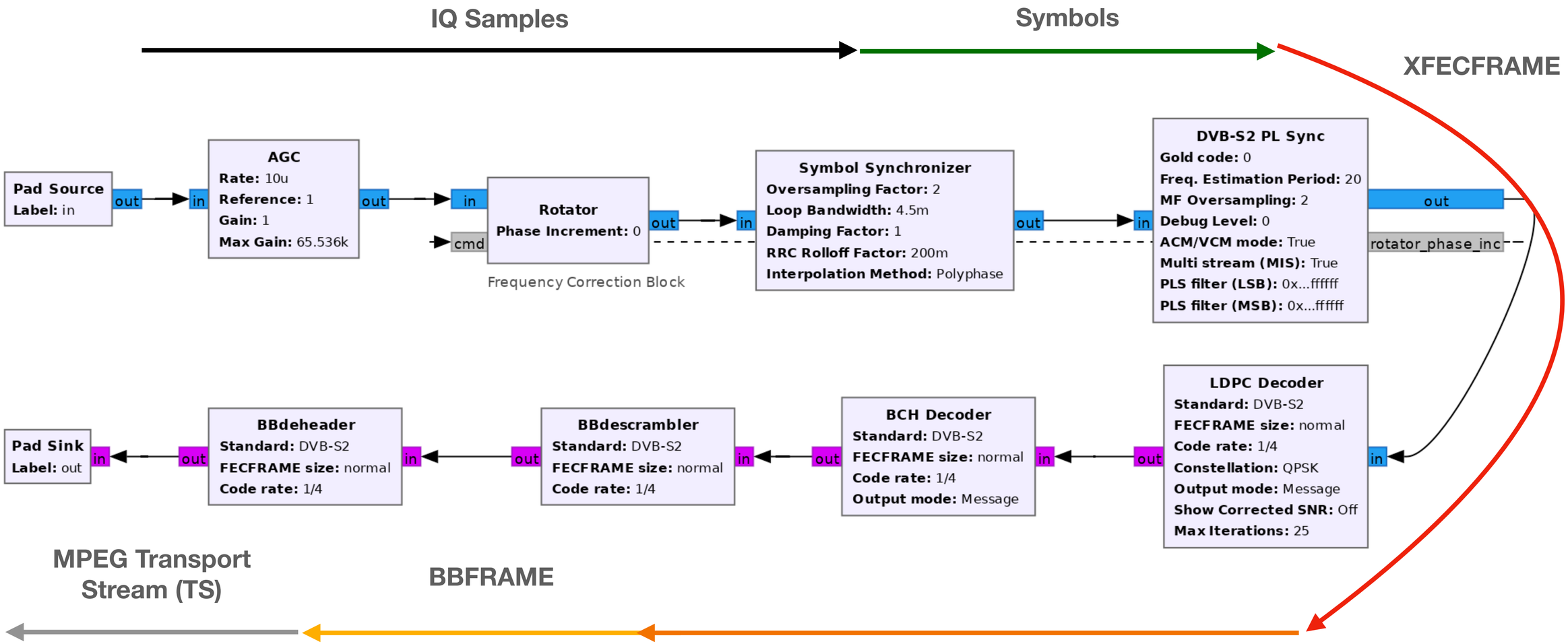
DVB-S2 Rx Architecture

Block Diagram



DVB-S2 Rx Architecture

Block Diagram



Project State and Path Forward

Current Stage

Minimal Functionality

- Constant coding and modulation (CCM).
- Single Input Stream (SIS).
- QPSK and 8PSK constellations.
- Pilot mode operation.
- Support for RTL-SDR and USRP.
- Tested on Ubuntu and Fedora.

Project State and Path Forward

Current Stage

Minimal Functionality → Optimization and Performance

- Constant coding and modulation (CCM).
- Single Input Stream (SIS).
- QPSK and 8PSK constellations.
- Pilot mode operation.
- Support for RTL-SDR and USRP.
- Tested on Ubuntu and Fedora.

- Faster BCH and LDPC decoding.
- More efficient symbol timing sync.
- Support for lower signal-to-noise ratios (SNRs).
- Improved implementation overall.

Project State and Path Forward

Current Stage

Minimal Functionality → **Optimization and Performance** → **Comprehensive DVB-S2 Support**

- Constant coding and modulation (CCM).
- Single Input Stream (SIS).
- QPSK and 8PSK constellations.
- Pilot mode operation.
- Support for RTL-SDR and USRP.
- Tested on Ubuntu and Fedora.

- Faster BCH and LDPC decoding.
- More efficient symbol timing sync.
- Support for lower signal-to-noise ratios (SNRs).
- Improved implementation overall.

- Adaptive/variable coding and modulation (ACM/VCM).
- Multiple Input Stream (MIS).
- 16APSK and 32APSK constellations.
- Pilotless mode.

Project State and Path Forward

Current Stage

Minimal Functionality → **Optimization and Performance** → **Comprehensive DVB-S2 Support** → **Widely Tested, Compatible and Stable**

- Constant coding and modulation (CCM).
- Single Input Stream (SIS).
- QPSK and 8PSK constellations.
- Pilot mode operation.
- Support for RTL-SDR and USRP.
- Tested on Ubuntu and Fedora.

- Faster BCH and LDPC decoding.
- More efficient symbol timing sync.
- Support for lower signal-to-noise ratios (SNRs).
- Improved implementation overall.

- Adaptive/variable coding and modulation (ACM/VCM).
- Multiple Input Stream (MIS).
- 16APSK and 32APSK constellations.
- Pilotless mode.

- Tested with more:
  - SDR boards.
  - DVB-S2 carriers.
  - Linux distributions/versions.
  - Host platforms.

Project State and Path Forward

Current Stage

Minimal Functionality → **Optimization and Performance** → **Comprehensive DVB-S2 Support** → **Widely Tested, Compatible and Stable** → DVB-S2X

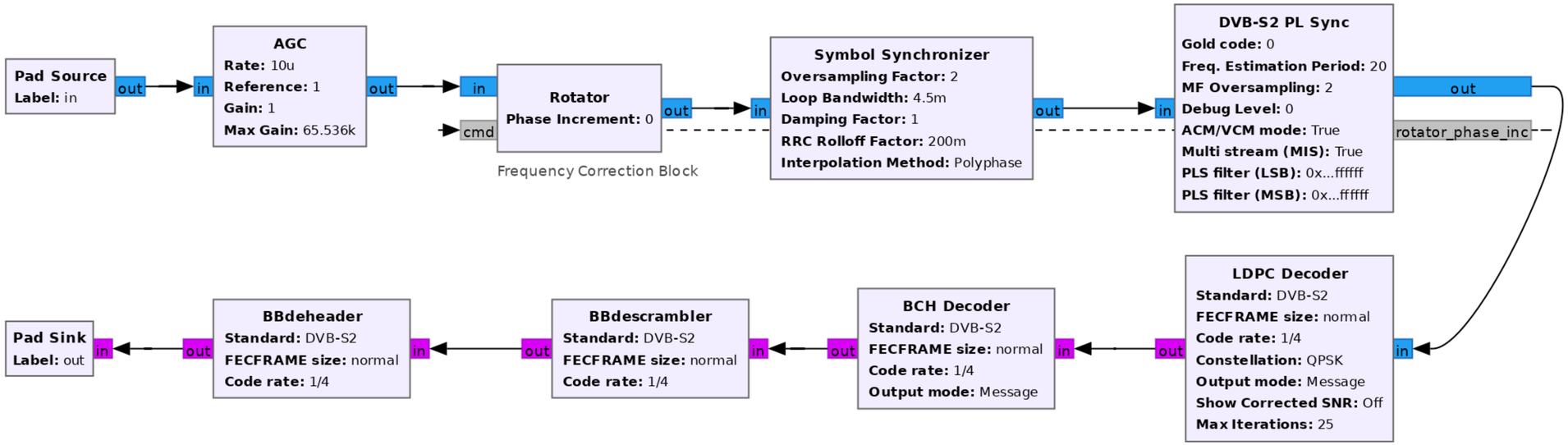
- Constant coding and modulation (CCM).
- Single Input Stream (SIS).
- QPSK and 8PSK constellations.
- Pilot mode operation.
- Support for RTL-SDR and USRP.
- Tested on Ubuntu and Fedora.

- Faster BCH and LDPC decoding.
- More efficient symbol timing sync.
- Support for lower signal-to-noise ratios (SNRs).
- Improved implementation overall.

- Adaptive/variable coding and modulation (ACM/VCM).
- Multiple Input Stream (MIS).
- 16APSK and 32APSK constellations.
- Pilotless mode.

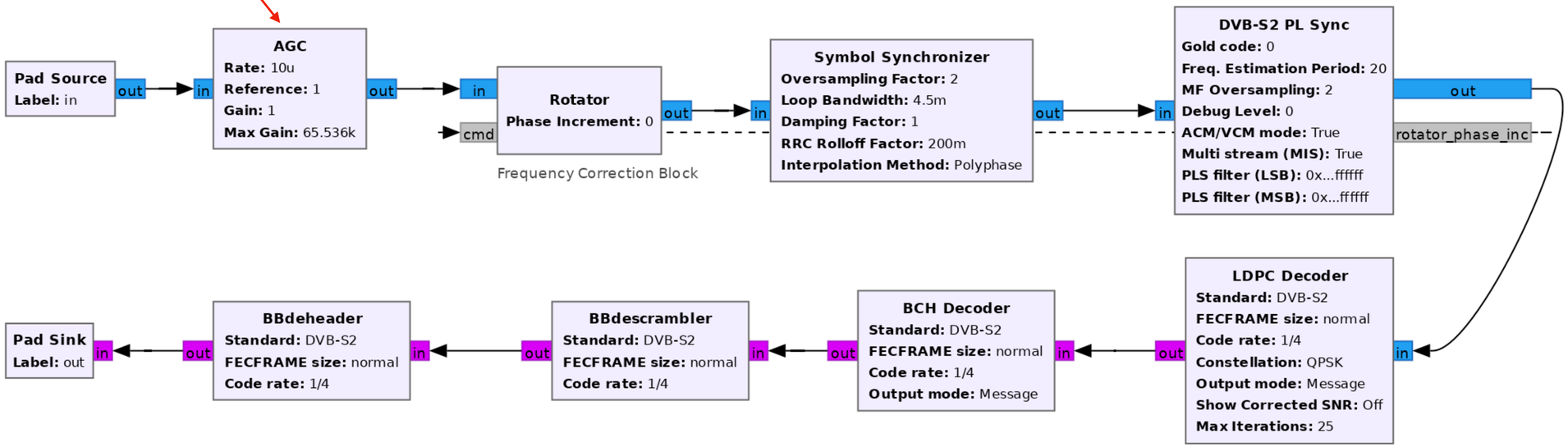
- Tested with more:
  - SDR boards.
  - DVB-S2 carriers.
  - Linux distributions/versions.
  - Host platforms.

Current Limitations



Current Limitations

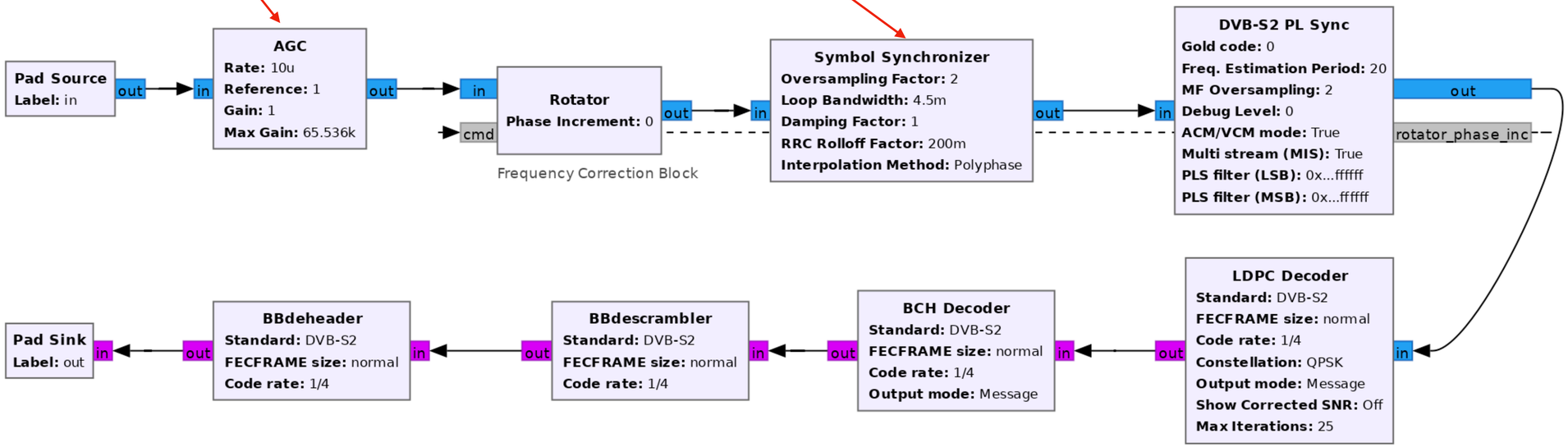
- Disturbed by the oversampling factor at low SNR.
- Not data-aided.



Current Limitations

- Disturbed by the oversampling factor at low SNR.
- Not data-aided.

- Faster than the GR in-tree symbol sync.
- Still too slow.
- Even integer oversampling only.

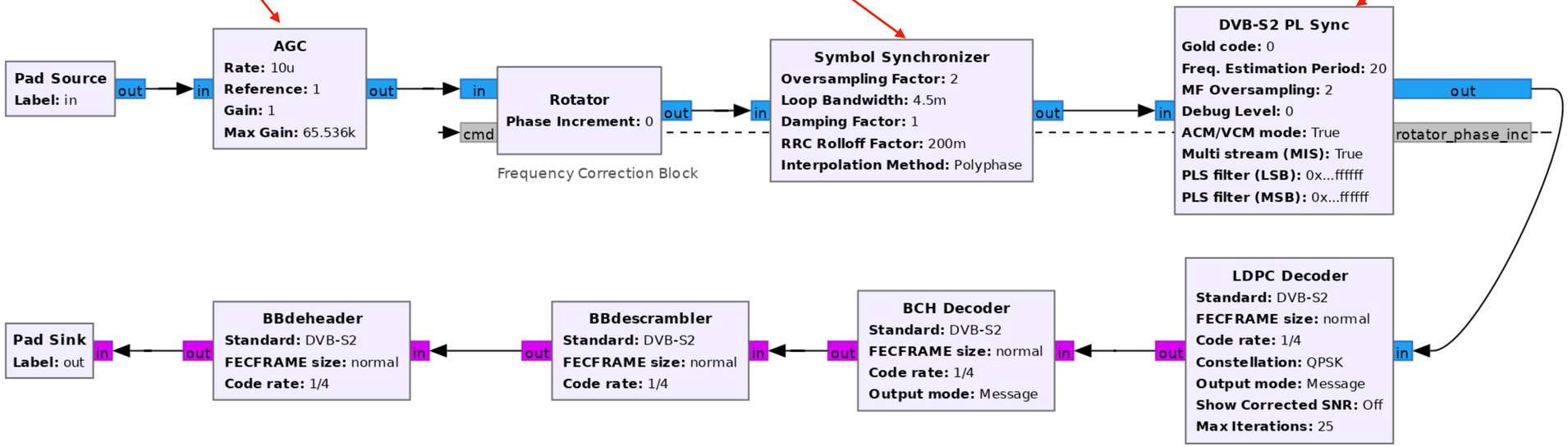


Current Limitations

- Disturbed by the oversampling factor at low SNR.
- Not data-aided.

- Faster than the GR in-tree symbol sync.
- Still too slow.
- Even integer oversampling only.

- Pilotless mode not fully implemented.
- SNR limit around +2 dB in pilot mode.

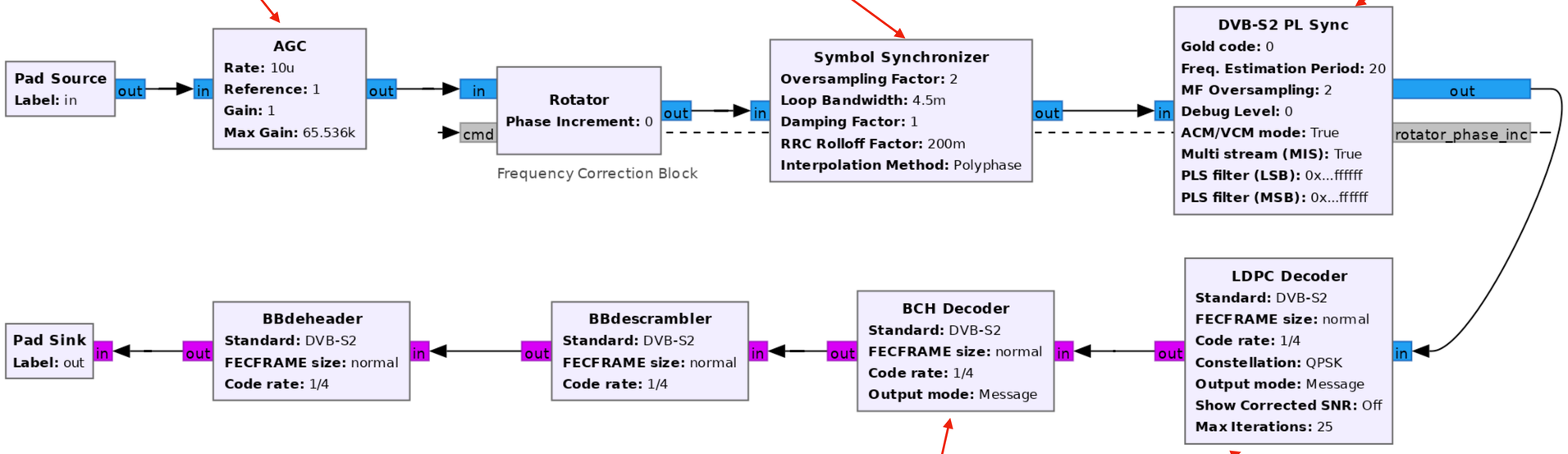


Current Limitations

- Disturbed by the oversampling factor at low SNR.
- Not data-aided.

- Faster than the GR in-tree symbol sync.
- Still too slow.
- Even integer oversampling only.

- Pilotless mode not fully implemented.
- SNR limit around +2 dB in pilot mode.

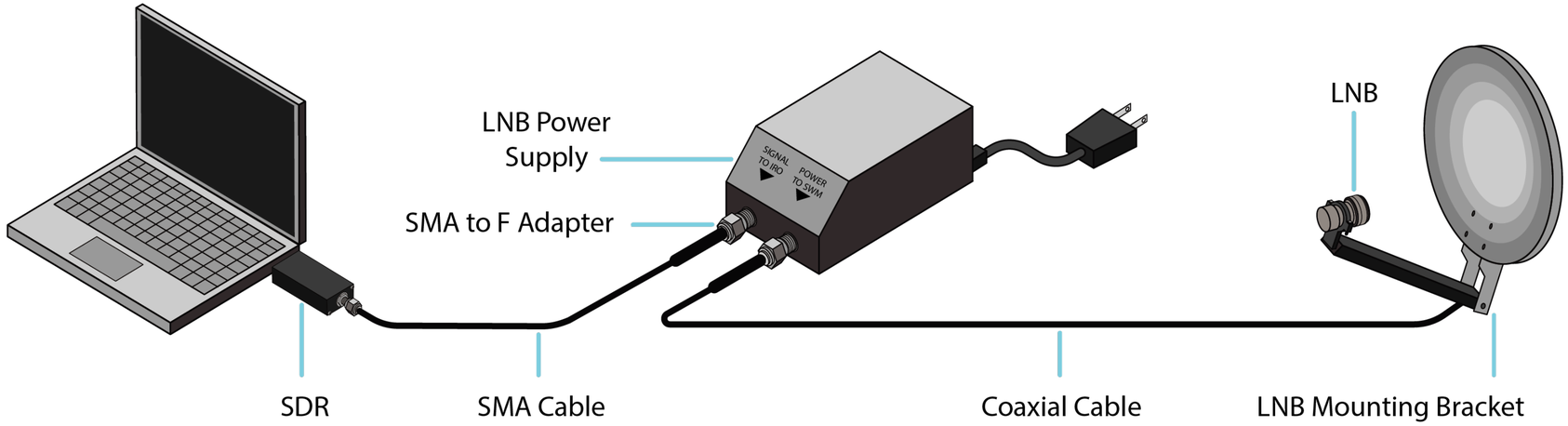


- Could be faster.
- Refactoring and benchmarking.

Example

Blockstream Satellite Reception

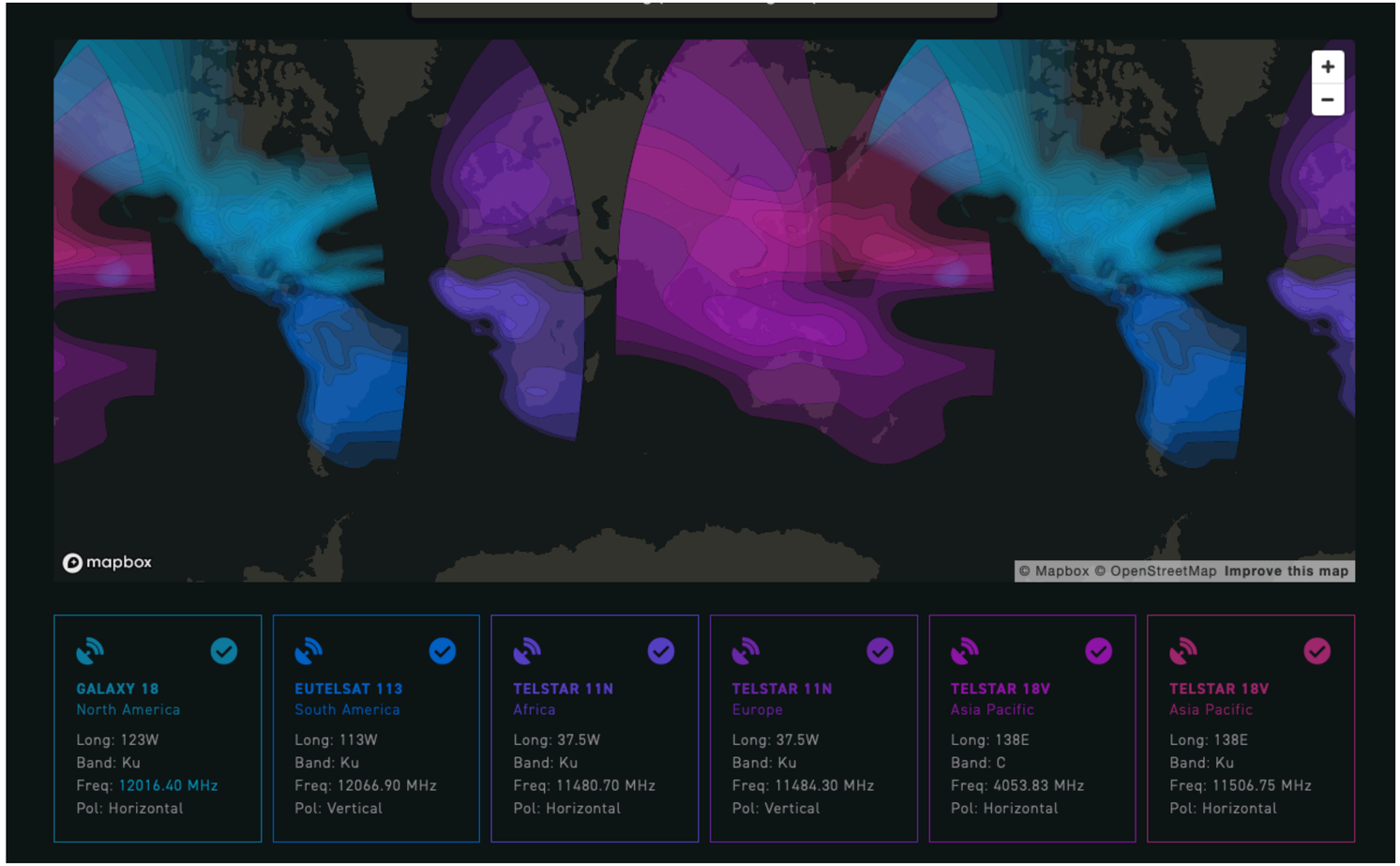
Hardware Setup



Example

Blockstream Satellite Reception

Carrier Info and Frequencies

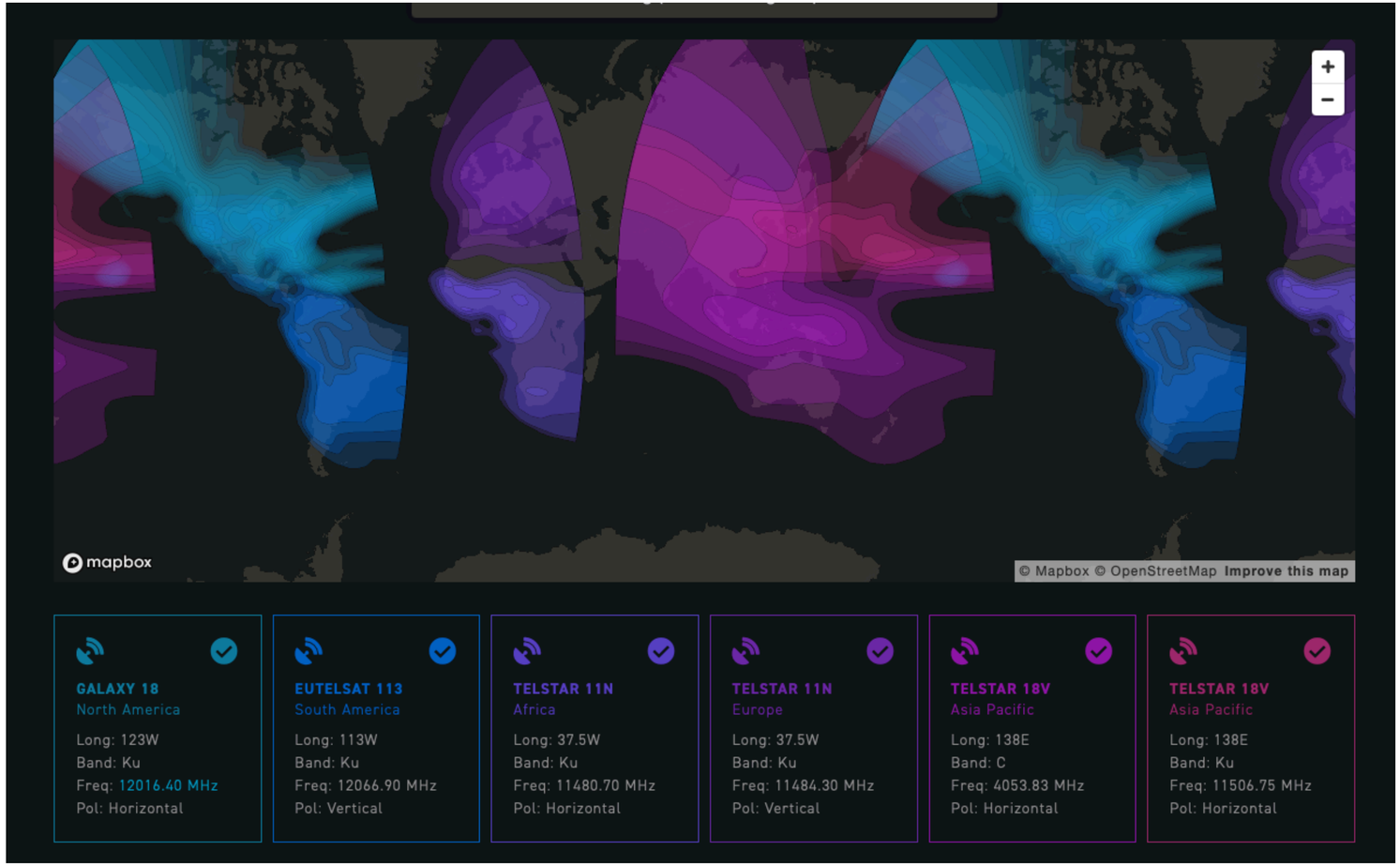


Satellite	Eutelsat 113
Longitude	113° W
DL Frequency	12066.9 MHz
LNB LO	10750 MHz
L-band Frequency	1316.9 MHz
Baud Rate	1 MBaud
MODCOD	QPSK 3/5
FECFRAME	Normal
Pilots	Enabled

Example

Blockstream Satellite Reception

Carrier Info and Frequencies



Satellite	Eutelsat 113
Longitude	113° W
DL Frequency	12066.9 MHz
LNB LO	10750 MHz
L-band Frequency	1316.9 MHz
Baud Rate	1 MBaud
MODCOD	QPSK 3/5
FECFRAME	Normal
Pilots	Enabled

RTL-SDR Supported

Example

Blockstream Satellite Reception

Example Commands

```
dvbs2-rx --source rtl --freq 1316.9e6 --samp-rate 2e6 --sym-rate 1e6 --rolloff 0.2 --modcod qpsk3/5 --frame-size normal --pilots on
```

Example

Blockstream Satellite Reception

Example Commands

```
dvbs2-rx --source rtl --freq 1316.9e6 --samp-rate 2e6 --sym-rate 1e6 --rolloff 0.2 --modcod qpsk3/5 --frame-size normal --pilots on
```

RTL-SDR
Source

Example

Blockstream Satellite Reception

Example Commands

```
dvbs2-rx --source rtl --freq 1316.9e6 --samp-rate 2e6 --sym-rate 1e6 --rolloff 0.2 --modcod qpsk3/5 --frame-size normal --pilots on
```

RTL-SDR Source

L-band Freq.

Example

Blockstream Satellite Reception

Example Commands

```
dvbs2-rx --source rtl --freq 1316.9e6 --samp-rate 2e6 --sym-rate 1e6 --rolloff 0.2 --modcod qpsk3/5 --frame-size normal --pilots on
```

RTL-SDR Source

L-band Freq.

sps=2

Example

Blockstream Satellite Reception

Example Commands

```
dvbs2-rx --source rtl --freq 1316.9e6 --samp-rate 2e6 --sym-rate 1e6 --rolloff 0.2 --modcod qpsk3/5 --frame-size normal --pilots on
```

RTL-SDR Source

L-band Freq.

sps=2

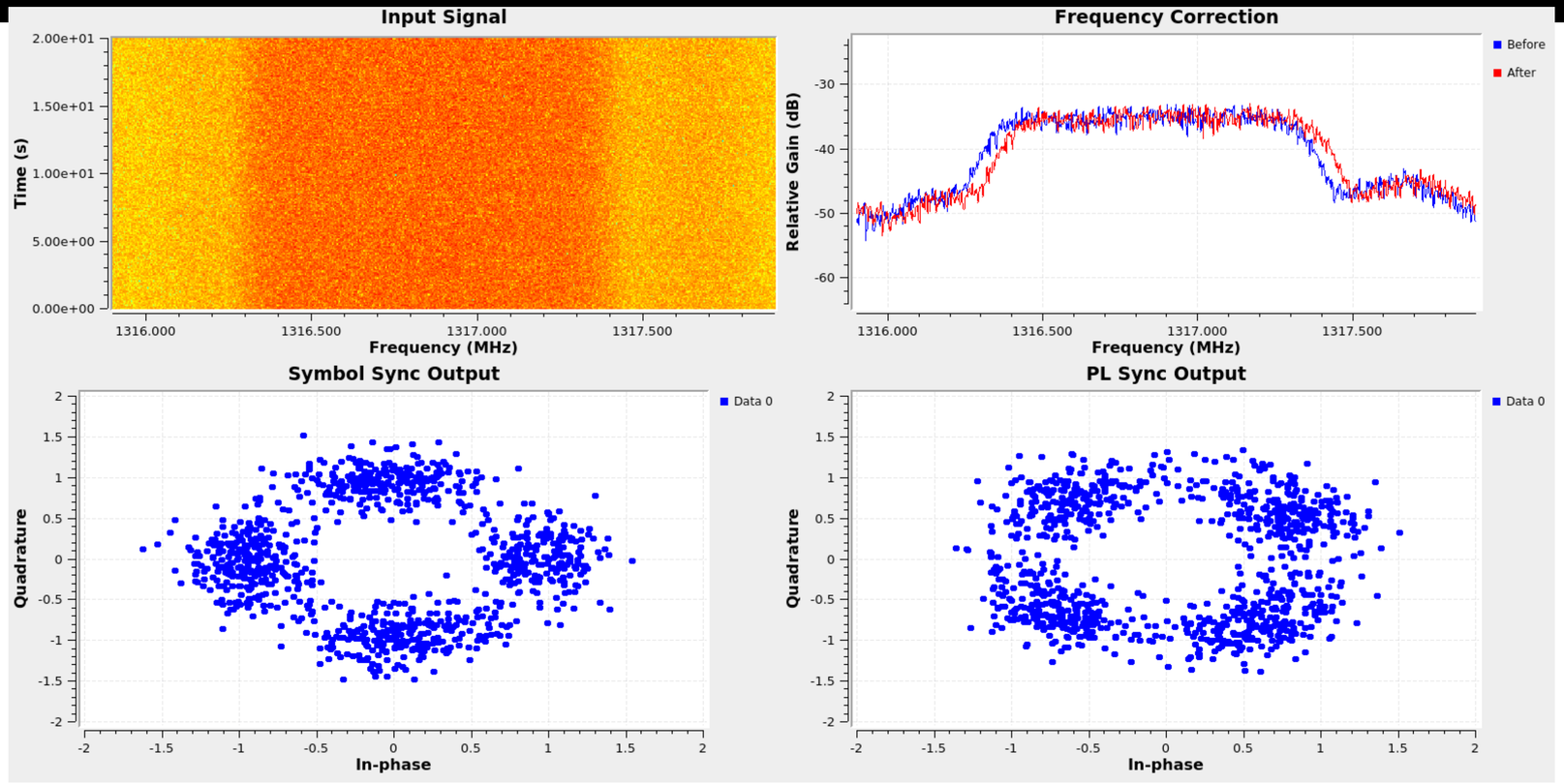
"Auto mode" available but a priori info recommended.

Example

Blockstream Satellite Reception

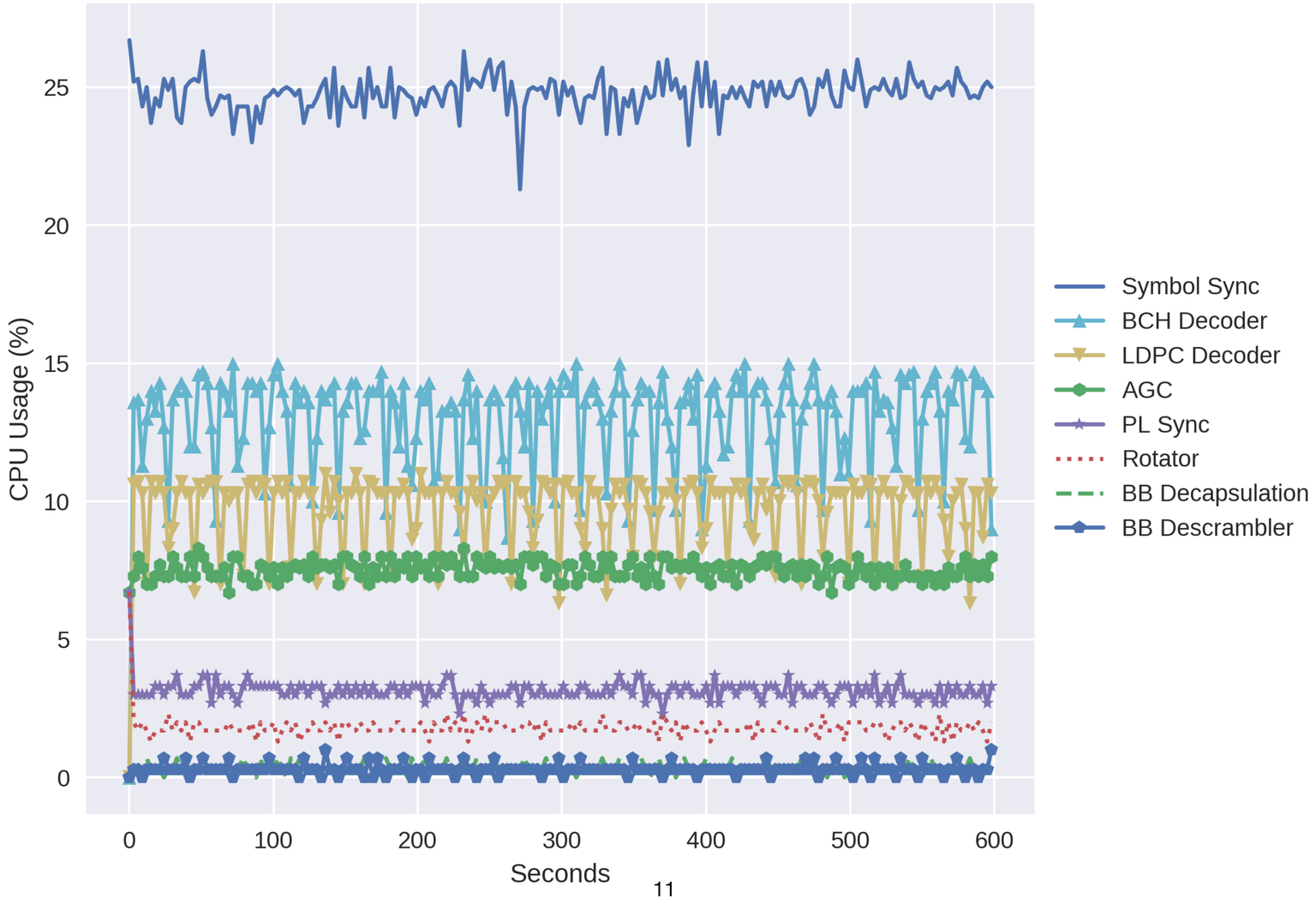
GUI Mode

```
dvbs2-rx --source rtl --freq 1316.9e6 --samp-rate 2e6 --sym-rate 1e6 --rolloff 0.2 --modcod qpsk3/5 --frame-size normal --pilots on --gui
```



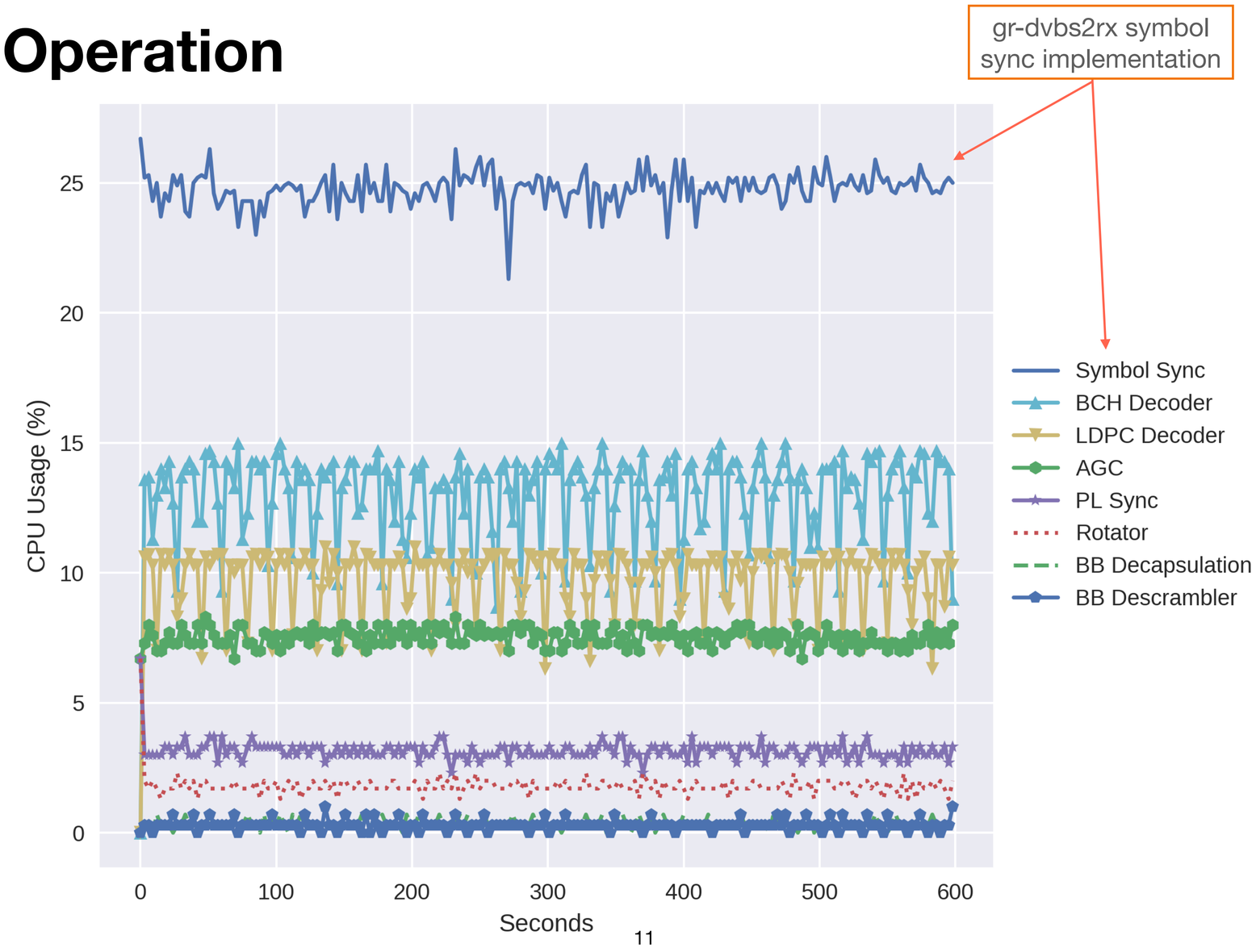
CPU Utilization

High SNR Operation



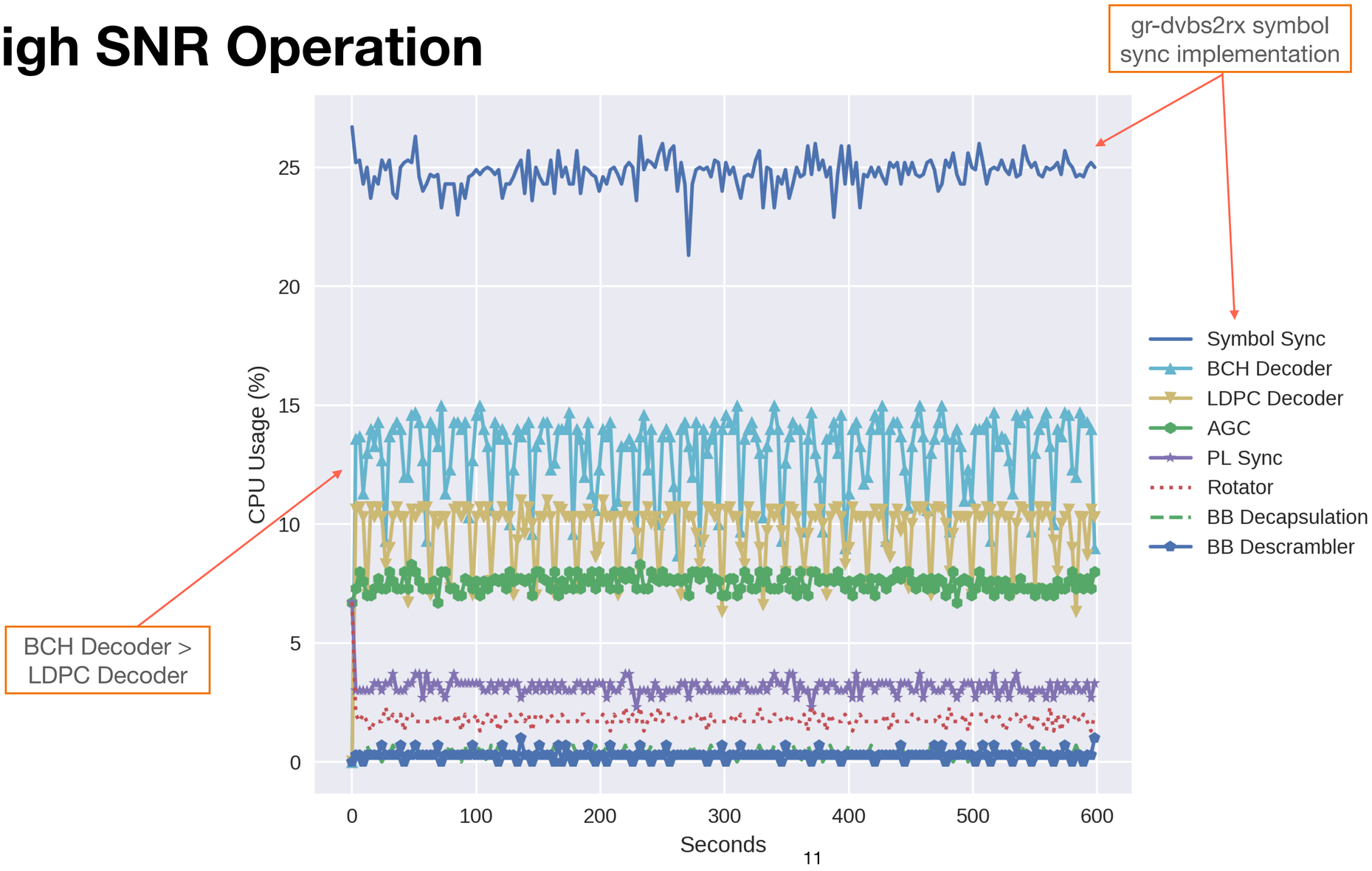
CPU Utilization

High SNR Operation



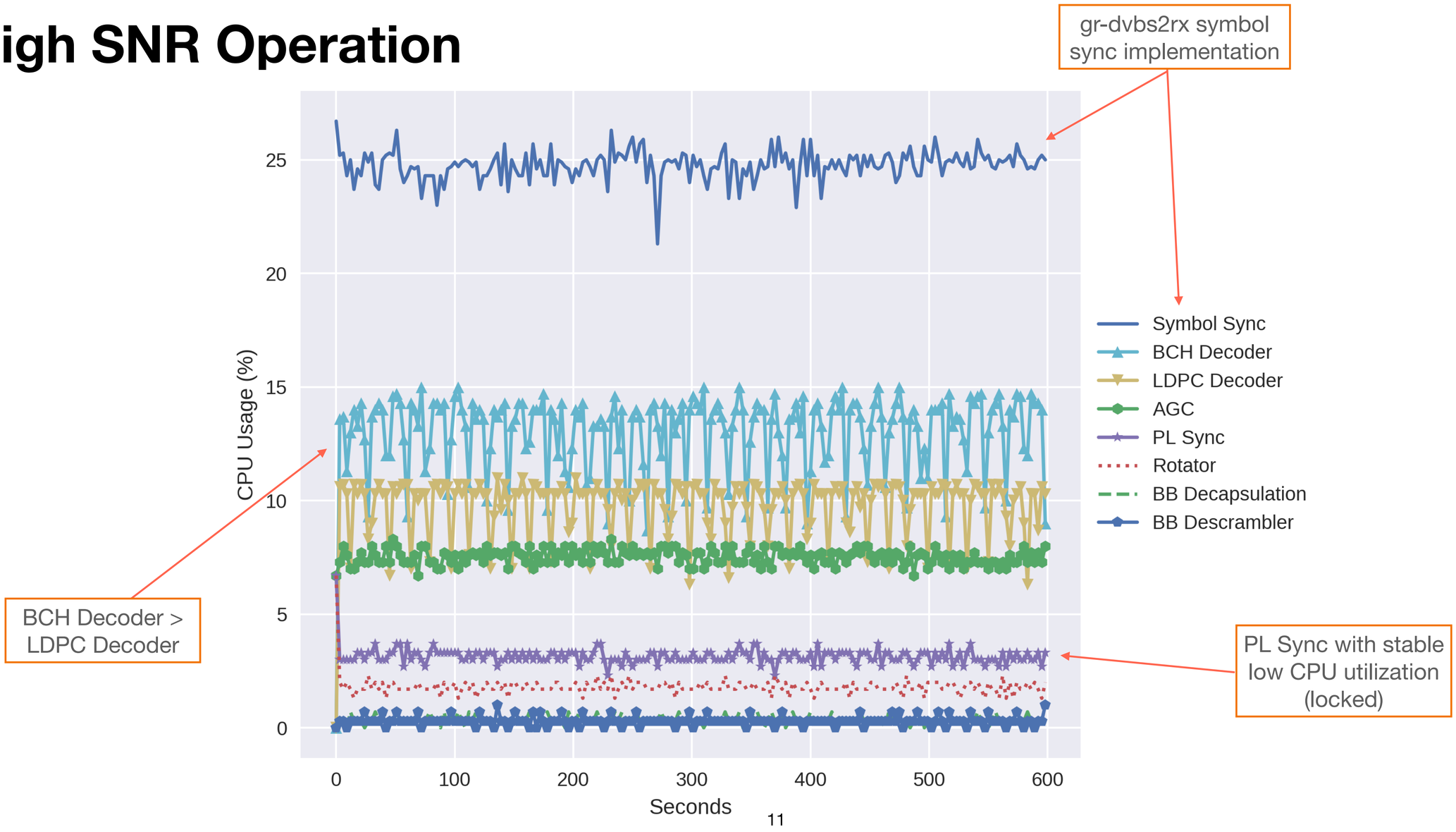
CPU Utilization

High SNR Operation



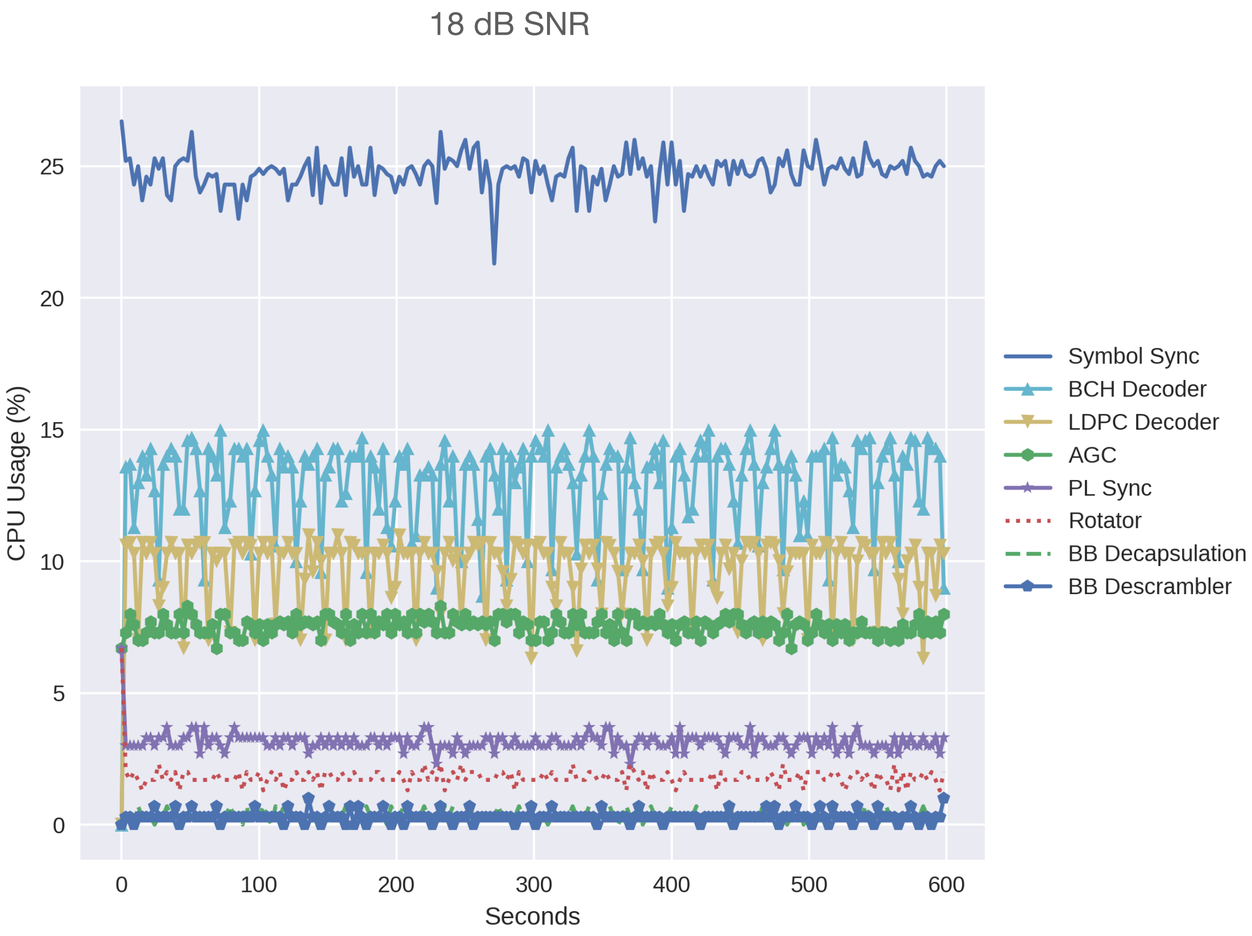
CPU Utilization

High SNR Operation



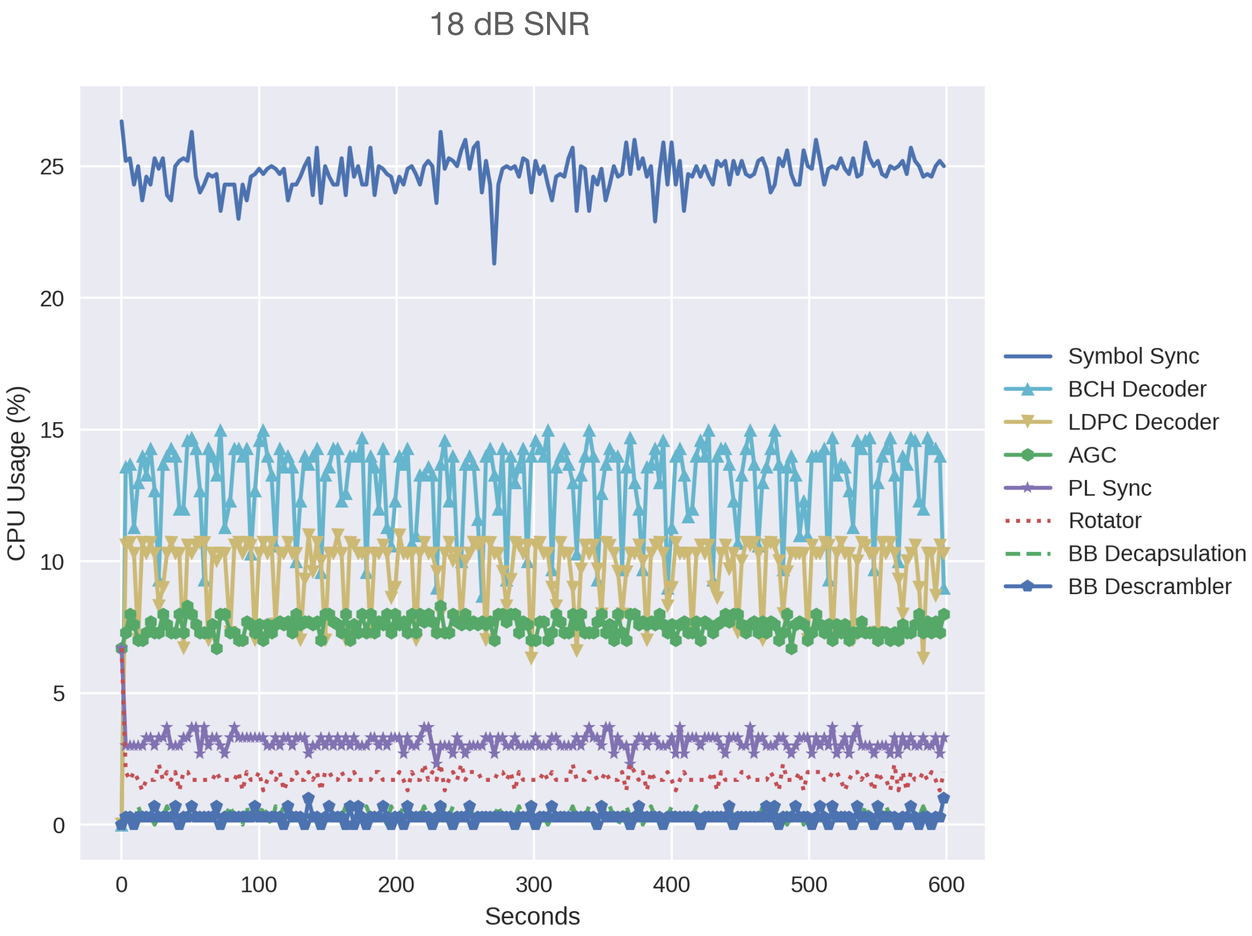
CPU Utilization

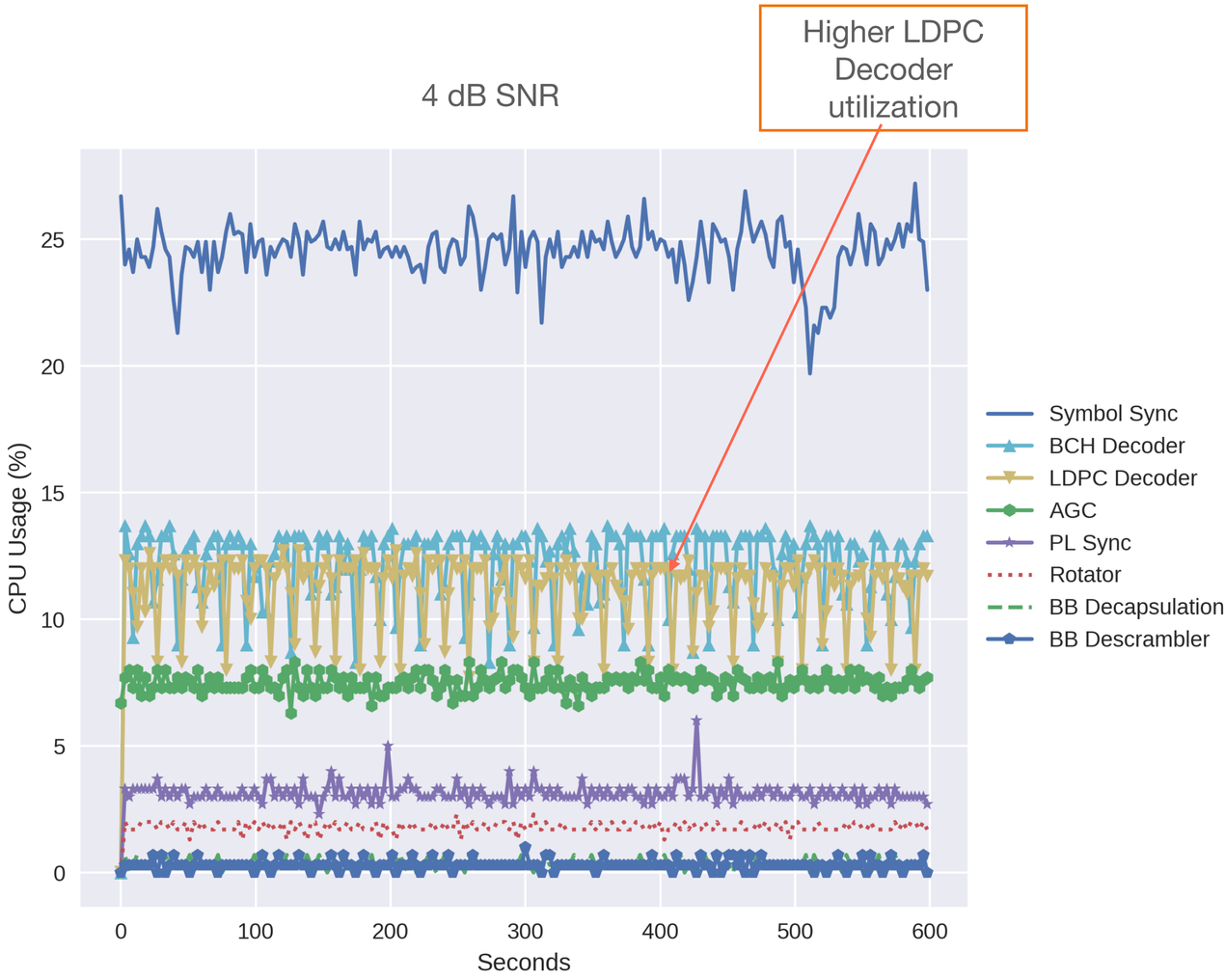
High SNR vs. Low SNR



CPU Utilization

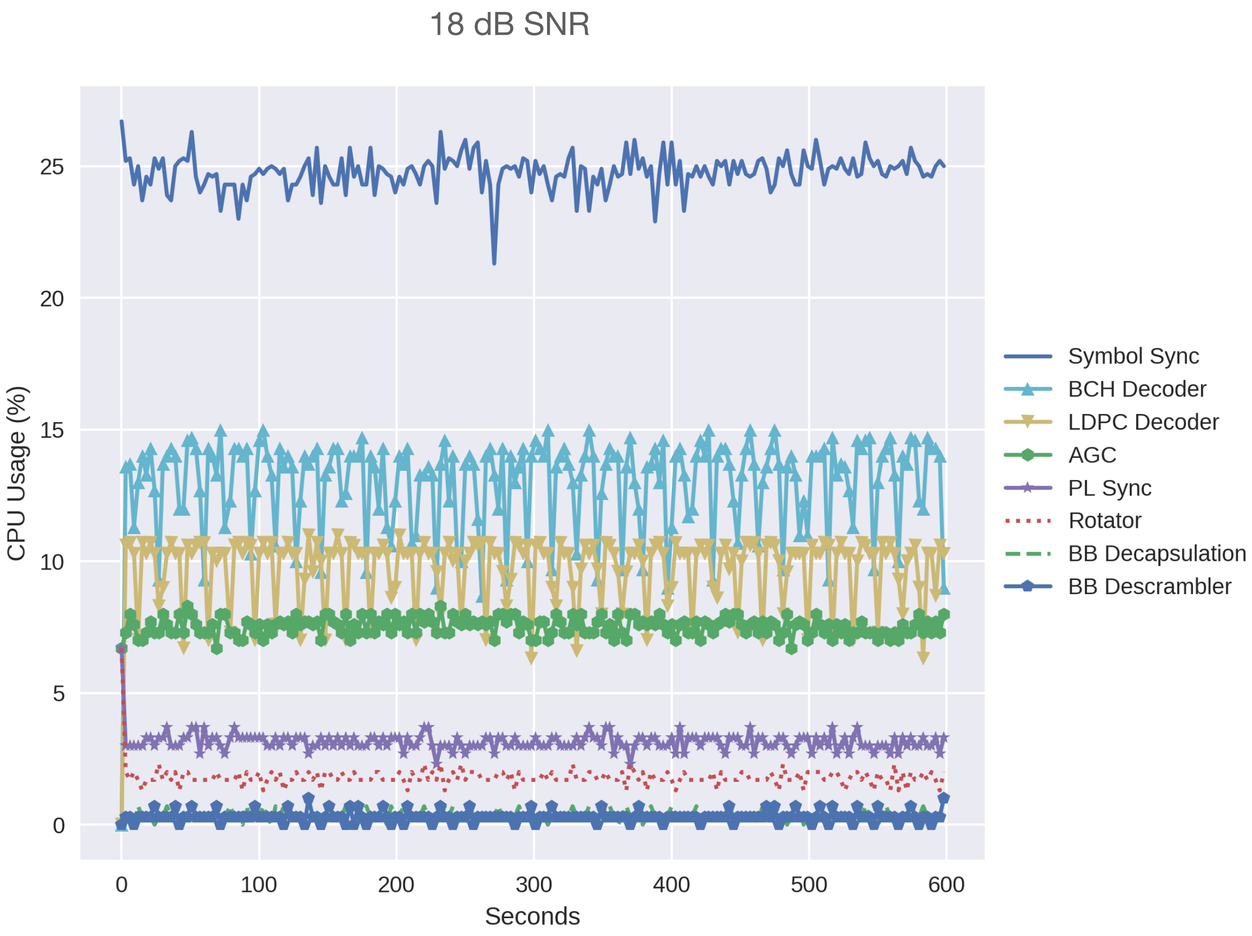
High SNR vs. Low SNR





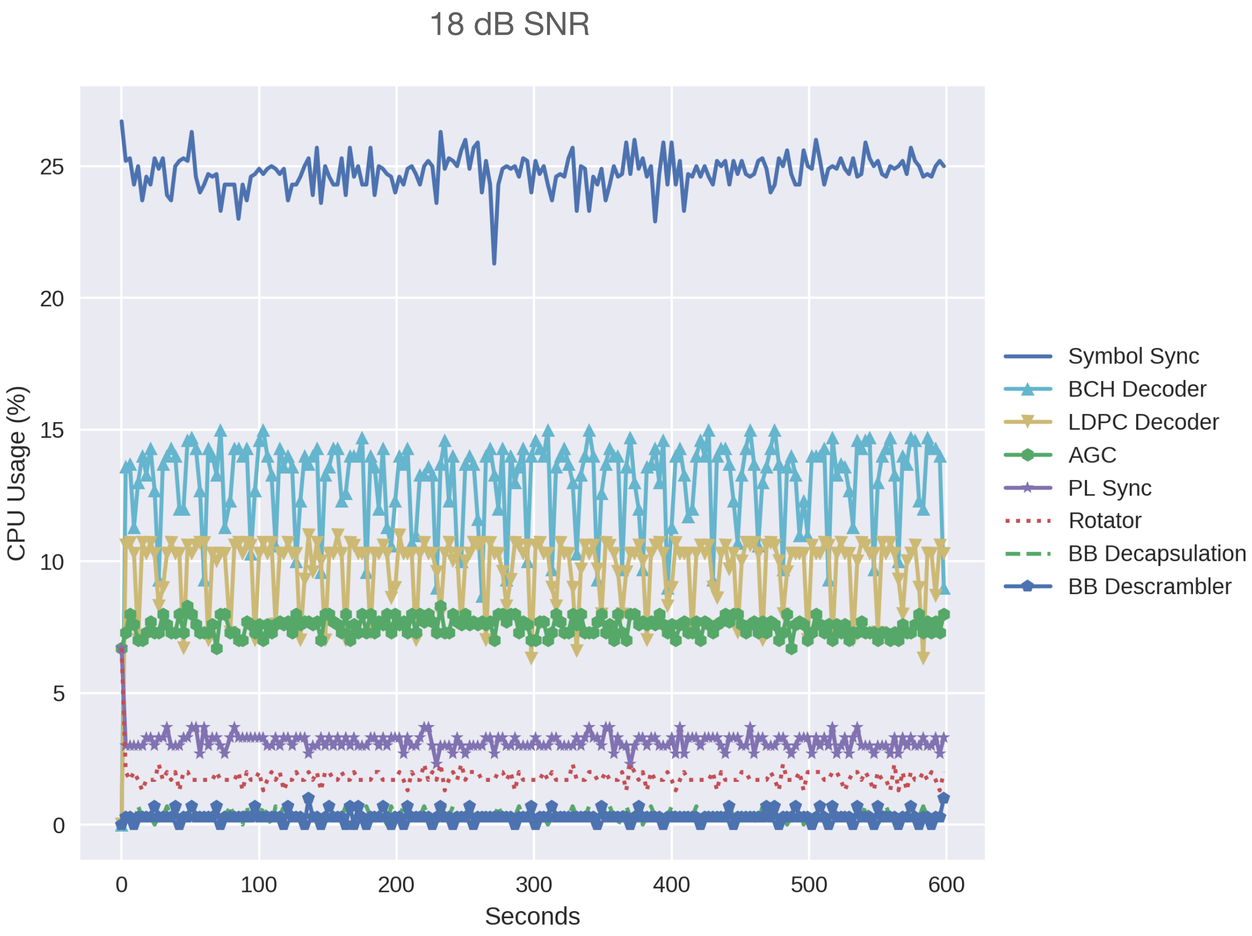
CPU Utilization

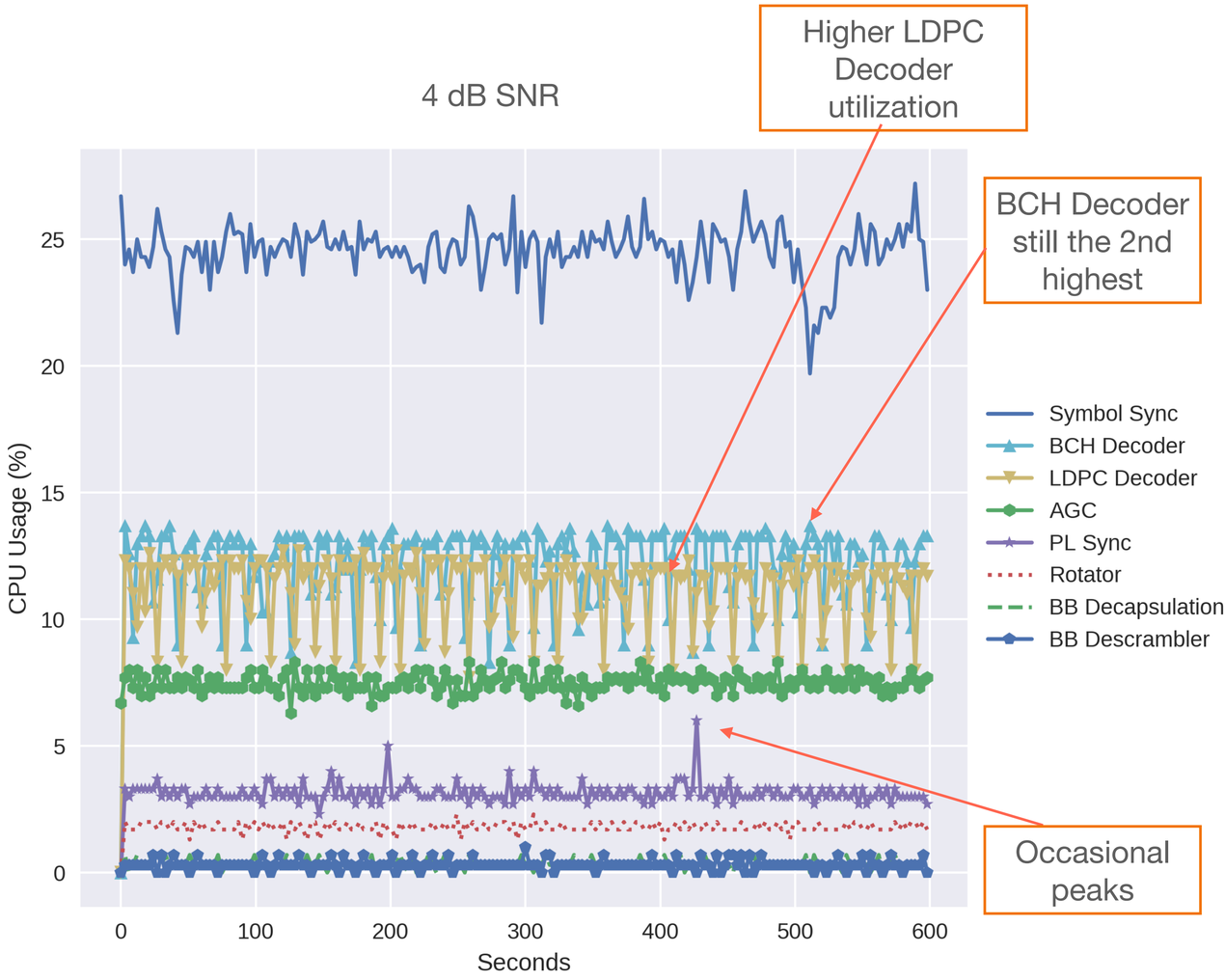
High SNR vs. Low SNR



CPU Utilization

High SNR vs. Low SNR





Questions?

Repository: https://github.com/igorauad/gr-dvbs2rx

Docs: https://igorauad.github.io/gr-dvbs2rx/

Igor Freire
igor@blockstream.com