Implementation and Evaluation of Phase Synchronization of USRP devices in GNU Radio / GRC Environment for Rapid Prototyping

Tomoya Nakahama, Yoji Yamada (NIT, Ishikawa College)
Suguru Kameda, Mizuki Motoyoshi, Noriharu Suematsu (RIEC, Tohoku University)

Motivation

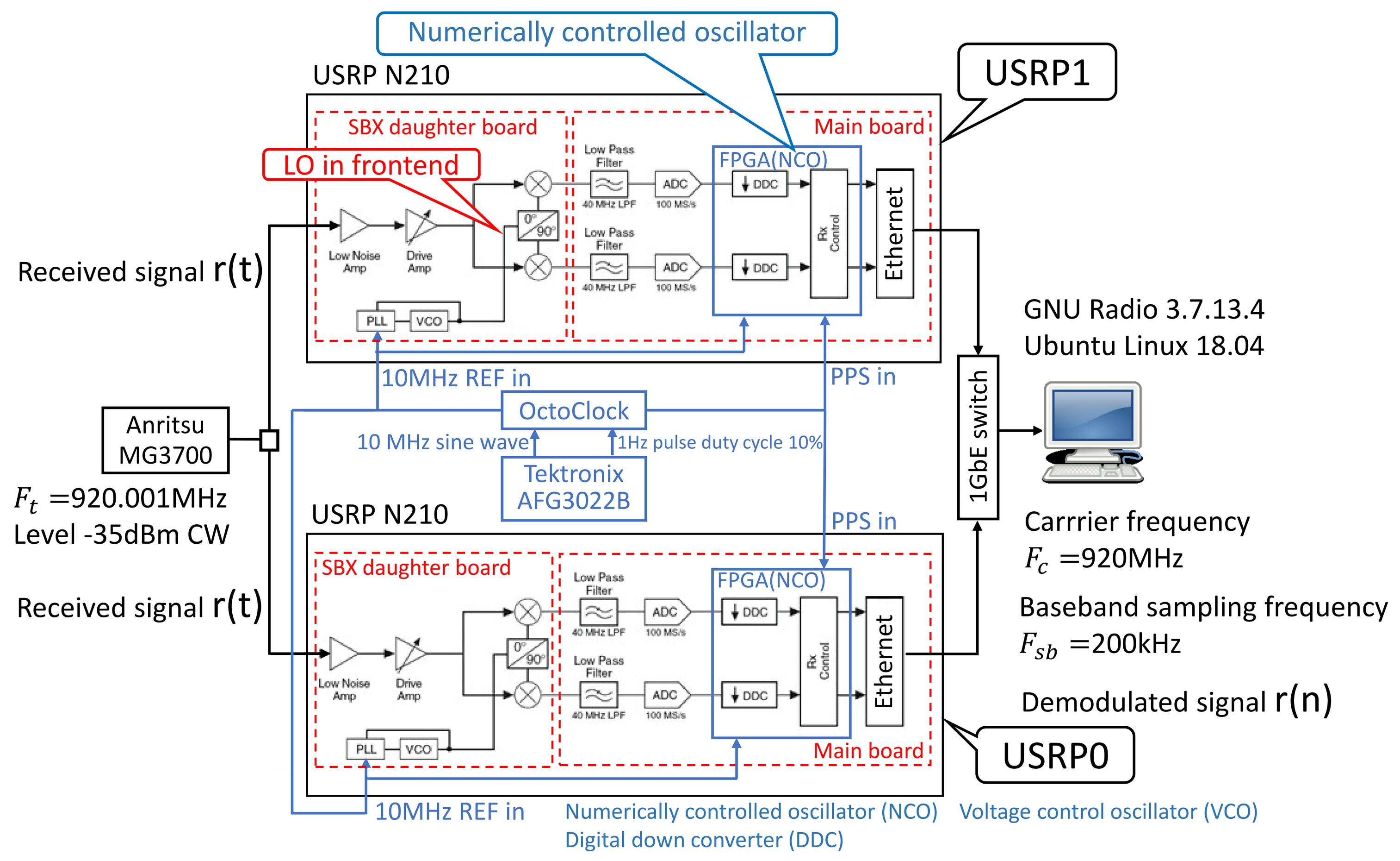
- Rapid prototyping with USRP and GNU Radio requires the following features:
  - ✓ Synchronizing carrier frequency and channel phase
  - ✓ Timing control of transmitting / receiving signals
- By default, two USRP equipped with a daughter board with a local oscillator (LO) cannot achieve channel-to-channel synchronization even when 10MHz Ref and PPS are supplied. This is still a major issue in rapid prototyping using USRPs[1-2].
- To solve this problem, we propose the use of UHD timed-commands[3] and GNU Radio Companion (GRC) to synchronize the two USRPs.

Universal Software Radio Peripheral (USRP)   Pulse-per-second signal (PPS)
USRP Hardware Driver (UHD)

Summary and future plans

- On USRP N200 with SBX daughterboard (0.4 to 4.4GHz) and GNU Radio, we proposed a two-channel coherent receiver implementation. This technique can also be applied to transmitters and transceivers.
- Aligning LOs at the front end is executed only once at startup by using of UHD timed-commands (UHD-Com).
- Because UHD-Com runs on FPGA, timing is controlled according to the accuracy of the FPGA clock, and no customization of FPGA code is required.
- Evaluating the accuracy of inter-channel synchronization remains for future work.
- USRP X300 and SBX120 have not yet achieved inter-channel synchronization.

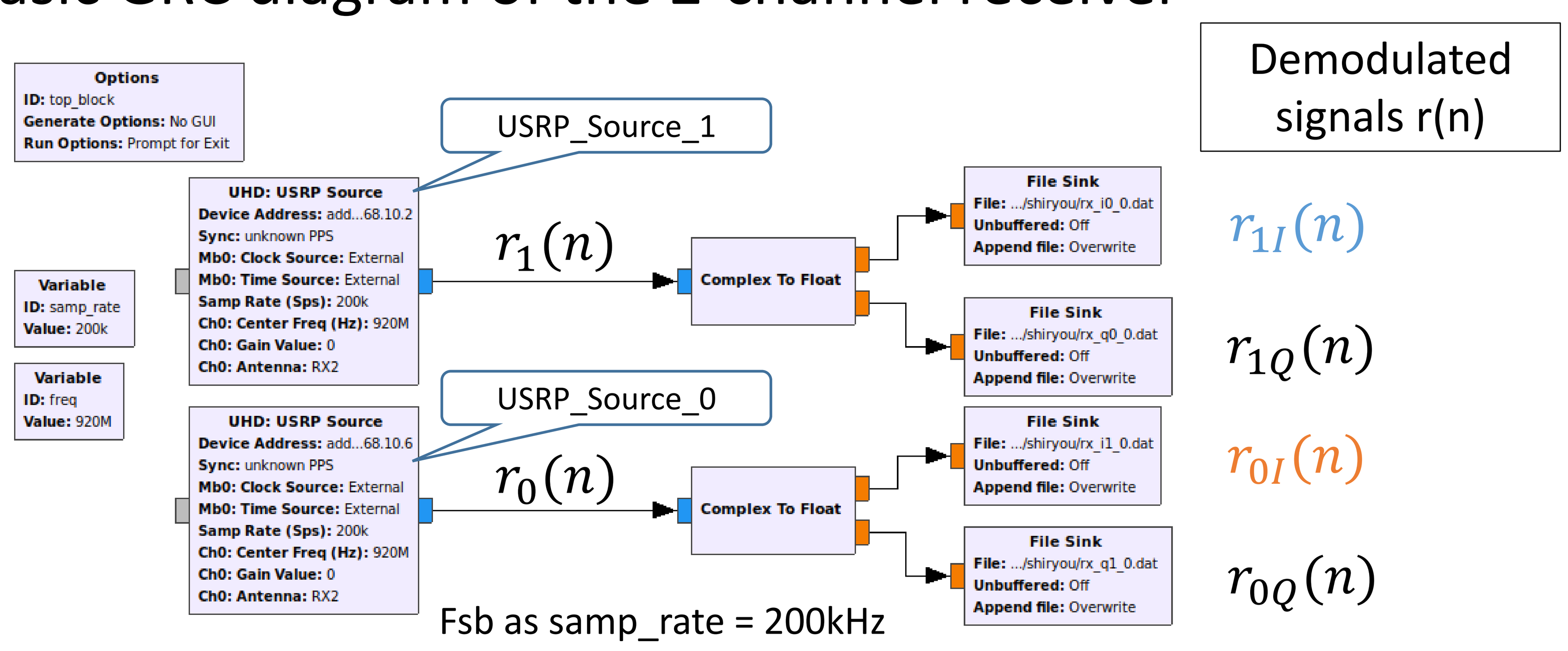
Experimental setup for 2-channel receiver using two USRPs



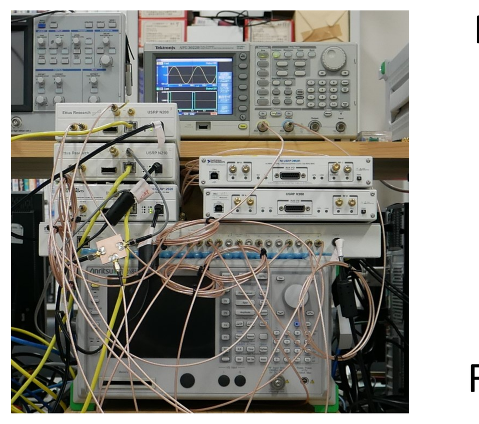
NOTE: Added and reprinted from http://www.ni.com/tutorial/14705/ja/

Fig.1 2-channel receiver block diagram using two USRP N210 with a SBX daughter board.

Basic GRC diagram of the 2-channel receiver



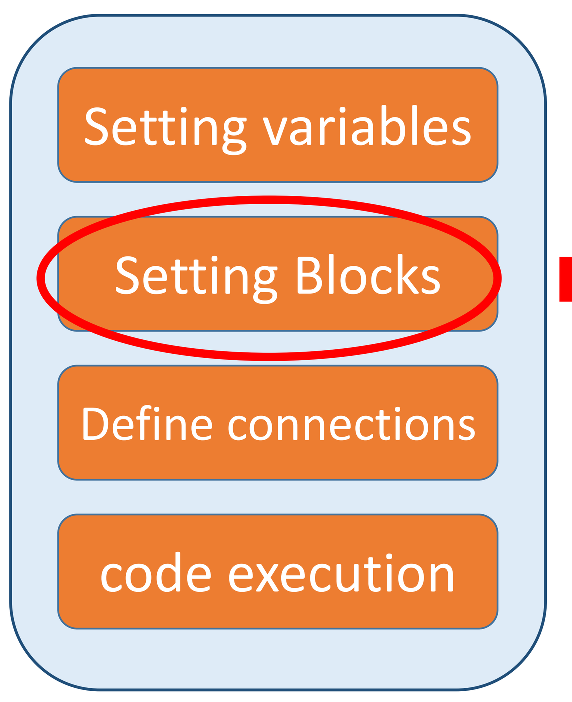
Fsb as samp_rate = 200kHz
Fc as Center_Freq = 920MHz



Problem to be solved:
In spite of 10MHz Ref and PPS are supplied, USRP equipped with a daughter board with a local oscillator (LO) cannot achieve channel-to-channel synchronization of demodulated signals.

Fig.2 GRC block diagram of the two-channel receiver in Figure 1.
GNU Radio Companion (GRC)

Typical Python code structure of 2-channel non-coherent receiver



top_block.py

```
self.uhd_usrp_source_1 = uhd.usrp_source(
    ",".join(("addr=192.168.10.2", "")),
    uhd.stream_args(
        cpu_format="fc32",
        channels=range(1),
    ),
)
self.uhd_usrp_source_1.set_clock_source('external', 0)
self.uhd_usrp_source_1.set_time_source('external', 0)
self.uhd_usrp_source_1.set_samp_rate(samp_rate)
self.uhd_usrp_source_1.set_time_unknown_pps(uhd.time_spec())
self.uhd_usrp_source_1.set_center_freq(freq, 0)
self.uhd_usrp_source_1.set_gain(0, 0)
self.uhd_usrp_source_1.set_antenna('RX2', 0)
self.uhd_usrp_source_0 = uhd.usrp_source(
    ",".join(("addr=192.168.10.6", "")),
    uhd.stream_args(
        cpu_format="fc32",
        channels=range(1),
    ),
)
self.uhd_usrp_source_0.set_clock_source('external', 0)
self.uhd_usrp_source_0.set_time_source('external', 0)
self.uhd_usrp_source_0.set_samp_rate(samp_rate)
self.uhd_usrp_source_0.set_time_unknown_pps(uhd.time_spec())
self.uhd_usrp_source_0.set_center_freq(freq, 0)
self.uhd_usrp_source_0.set_gain(0, 0)
self.uhd_usrp_source_0.set_antenna('RX2', 0)
```

Fig.3 Python code structure generated by the GRC shown in Figure 2.

Adding UHD timed-commands to the Python code

STEP 1 Align the system time of two USRPs
STEP 2 Set the receive start time for each USRP
STEP 3 Align LO phase at a specified time

```
self.uhd_usrp_source_1 = uhd.usrp_source(
    ",".join(("addr=192.168.10.2", "")),
    uhd.stream_args(
        cpu_format="fc32",
        channels=range(1),
    ),
)
self.uhd_usrp_source_1.set_clock_source('external', 0)
self.uhd_usrp_source_1.set_time_source('external', 0)
self.uhd_usrp_source_1.set_samp_rate(samp_rate)
self.uhd_usrp_source_1.set_time_unknown_pps(uhd.time_spec())
self.uhd_usrp_source_1.set_center_freq(freq, 0)
self.uhd_usrp_source_1.set_gain(0, 0)
self.uhd_usrp_source_1.set_antenna('RX2', 0)
self.uhd_usrp_source_0 = uhd.usrp_source(
    ",".join(("addr=192.168.10.6", "")),
    uhd.stream_args(
        cpu_format="fc32",
        channels=range(1),
    ),
)
self.uhd_usrp_source_0.set_clock_source('external', 0)
self.uhd_usrp_source_0.set_time_source('external', 0)
self.uhd_usrp_source_0.set_samp_rate(samp_rate)
self.uhd_usrp_source_0.set_time_unknown_pps(uhd.time_spec())
self.uhd_usrp_source_0.set_center_freq(freq, 0)
self.uhd_usrp_source_0.set_gain(0, 0)
self.uhd_usrp_source_0.set_antenna('RX2', 0)
```

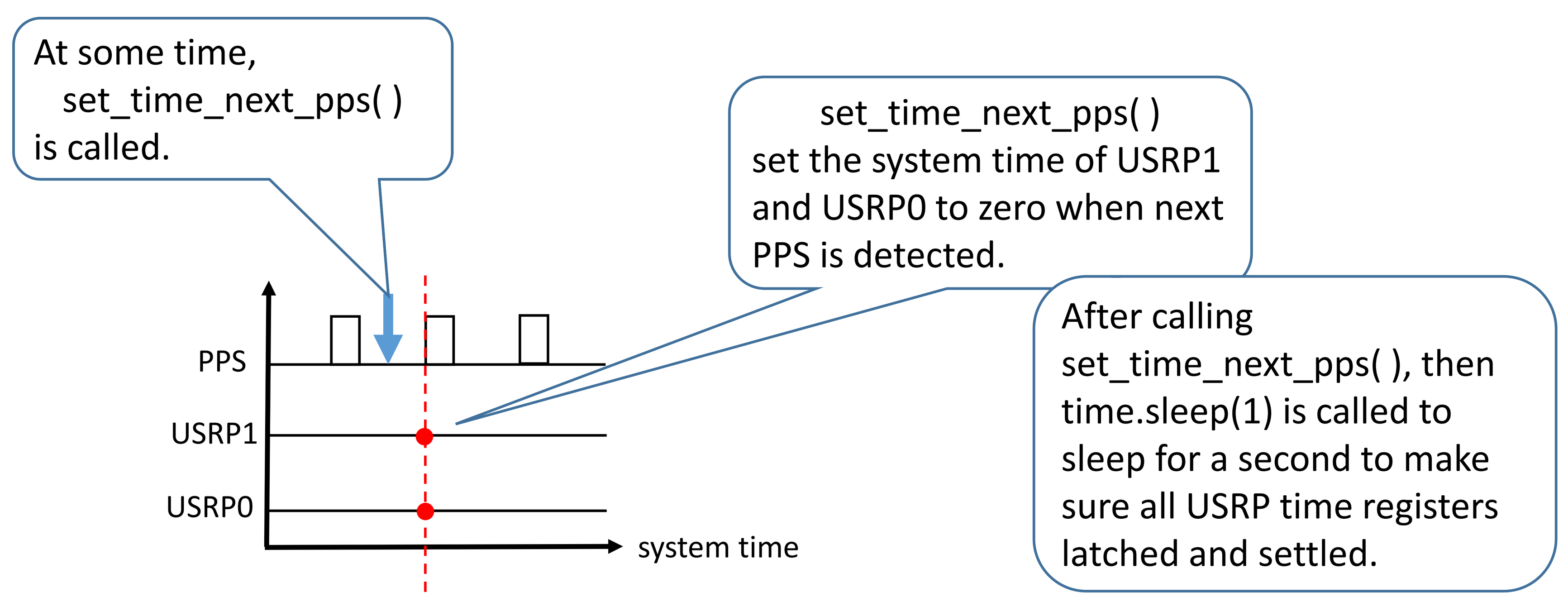
Deleting

```
#setting mb time
self.uhd_usrp_source_1.set_time_next_pps(uhd.time_spec())
self.uhd_usrp_source_0.set_time_next_pps(uhd.time_spec())
time.sleep(1)
#setting start timing
cmd_time = self.uhd_usrp_source_1.get_time_now() + uhd.time_spec(1)
self.uhd_usrp_source_1.set_start_time(cmd_time+uhd.time_spec(2))
self.uhd_usrp_source_0.set_start_time(cmd_time+uhd.time_spec(2))
#Align LO's in the front-end
self.uhd_usrp_source_1.set_command_time(cmd_time)
self.uhd_usrp_source_0.set_command_time(cmd_time)
self.uhd_usrp_source_1.set_center_freq(freq, 0)
self.uhd_usrp_source_0.set_center_freq(freq, 0)
self.uhd_usrp_source_0.clear_command_time()
self.uhd_usrp_source_1.clear_command_time()
```

STEP 1
STEP 2
STEP 3

Adding

Align the system time of two USRPs



STEP 1: Set the same system time for two USRPs.

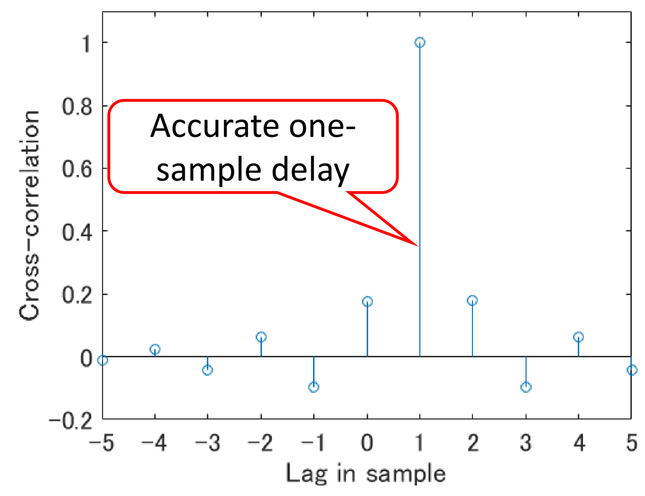
Fig.4 Two USRP time setting operations using the set_time_next_pps ( ) function.

Setting the receive start time for each USRP

STEP 2: By using set_start_time( ), the receiving start time for each USRP can be preprogrammed. This allows you to start receiving after aligning the LOs.

Verification example

```
start_time = self.uhd_usrp_source_1.get_time_now() + uhd.time_spec(2)
self.uhd_usrp_source_1.set_start_time(start_time)
self.uhd_usrp_source_0.set_start_time(start_time+uhd.time_spec(1/200e3))
```



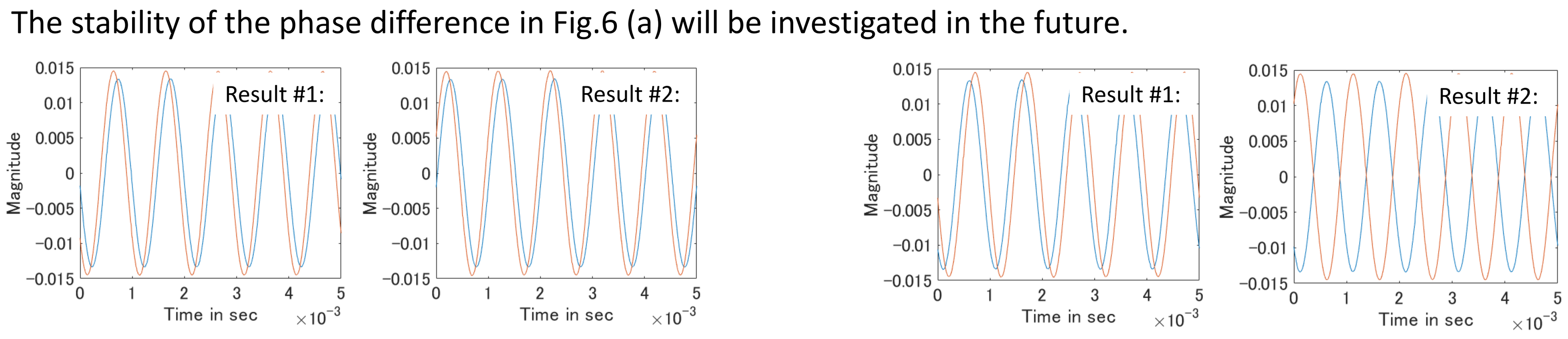
- DSB-SC modulated M-sequence is received by the implemented 2-channel coherent receiver.
- On one side of the USRP, set the receive start time 1/Fsb seconds later than the other USRP.
- Evaluate the cross-correlation between the demodulated signals of each USRP.

Fig.5 Verification test result of set_start_time( ) function.

Evaluation results of 2-channel coherent receiver implementation using UHD timed-commands

- The phase difference between $r_{0I}(n)$ and $r_{1I}(n)$ does not change from run to run as shown in Fig.6 (a).
- When the UHD timed-command is not used, the phase difference between $r_{0I}(n)$ and $r_{1I}(n)$ is not stable in each run, as shown in Fig. 6 (b).
- The inherent delay of the USRP or daughterboard has been suggested as a cause of the remaining phase difference in Fig.6 (a).
- The stability of the phase difference in Fig.6 (a) will be investigated in the future.

STEP 3: By using UHD timed-commands, the phase relationship between two USRPs remains fixed from run to run.



(a) Demodulated signals for each run using UHD timed-commands. (b) Demodulated signals for each run using GRC generated codes.

Fig.6 Evaluation results of phase difference fluctuation of $r_{0I}(n)$ and $r_{1I}(n)$ for each run.

References

[1]Dan Baker, "Phase Synchronization Techniques," GRCon 2019 at the Marriot at the Space & Rocket Center in Huntsville, Alabama, Sept. 2019.
[2]M. Krueckemeier, F. Schwartau, C. Monka-Ewe and J. S. Technische, "Synchronization of Multiple USRP SDRs for Coherent Receiver Applications," Proc. Sixth International Conference on Software Defined Systems, pp.11-16, DOI: 10.1109/SDS.2019.8768634, June 2019.
[3]USRP Hardware Driver and USRP Manual, https://files.ettus.com/manual/index.html