

Full-Duplex Experiment Using ZedBoard with AD9361 SDR

Kelvin Kuang-Chi Lee
Tamkang University

Overview of the Experiment

- Objective
 - Instrument a full-duplex (FDX) communication system with one half-duplex (HDX) transmitter (Tx), one FDX relay transceiver (Relay), and one FDX receiver (Rx)
- Key Elements:
 - Zedboard Zynq-7000 with AD9361
 - Omni-directional and directional antennas
 - Zynq QPSK transmitter/receiver blocks
 - Zynq QPSK Relay self-interference cancellation (SIC) algorithms
 - SGD-based vs. **RLS** based SIC
 - **Zadoff-Chu (ZC) sequence generation and synchronization**

System Setup (1)

- SDR
 - HDX TX (ZedBoard + AD9361 + omni-directional antenna)
 - FDX Relay (ZedBoard + AD9361 + directional antennas)
 - HDX RX (ZedBoard + AD9361 + omni-directional antenna)
- SIC @ Relay SDR
- ZC reference sequence for synchronization

Lab-3: IB FDX Communications

- Tx SDR Setup
 - Setup ZedBoard + AD9361 as the HDX QPSK Tx SDR
 - Configure "AD9361 Transmitter" (i.e., IP address, **carrier frequency = 5G Hz, antenna selection**, baseband sample rate, gain, ...)
 - Convert the "Hello World ####" message to baseband QPSK modulated signal including Baker/**ZC** sequence; transmit data to "Relay SDR" via the omni-directional antenna

QPSK Transmitter SDR (1)

QPSK Transmitter Using Analog Devices AD9361/AD9364

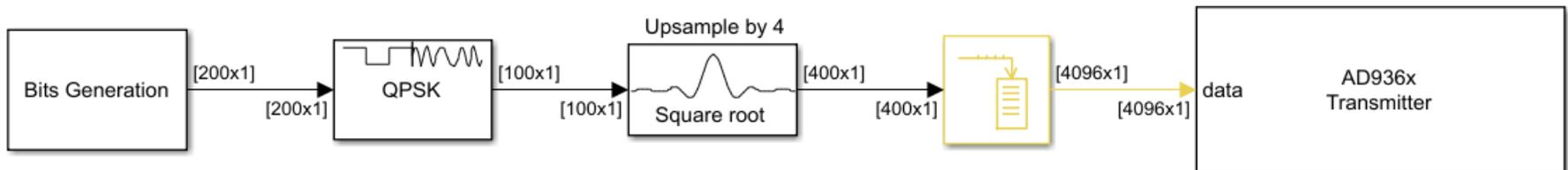
Note: Before running the QPSK models, first run the companion models for frequency offset calibration.

Open the companion `zynqRadioFrequencyCalibrationTxAD9361AD9364SLmodel`

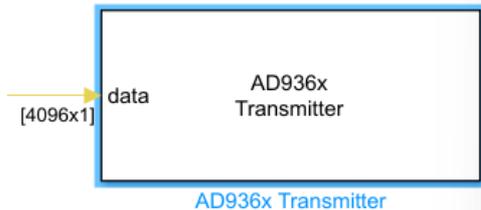
Open the companion `zynqRadioFrequencyCalibrationRxAD9361AD9364SLmodel`

Open the companion `zynqRadioQPSKRxAD9361AD9364SLmodel`

Info



QPSK Transmitter SDR (2)



Block Parameters: AD936x Transmitter

AD936x Transmitter
Transmit data using an AD936x.

Main Filter Advanced

Radio Connection

Radio IP address: 192.168.3.2

Info

Radio Properties

Source of center frequency: Dialog

Center frequency (Hz): 5e9+15500 ← Post Calibration result

Source of gain: Dialog

Gain (dB): -10

Channel mapping: 1

Baseband sample rate (Hz): sdrqpsktx.RadioFrontEndSampleRate 520841

Data

Enable output port for underflow indicator

OK Cancel Help Apply

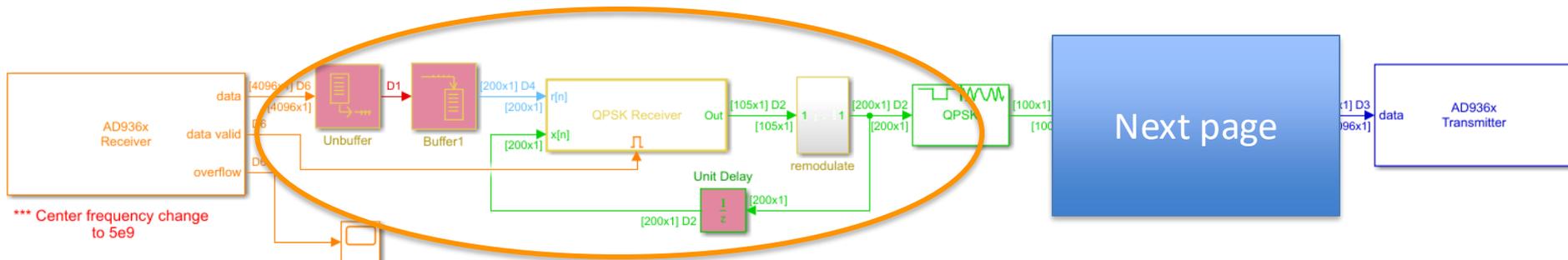
Full-Duplex (FDX) Relay SDR Setup

- Configure AD9361 Tx SDR block
 - Center Frequency: **5 GHz**
 - Baseband Sample Rate (i.e., data rate) = 520841 Hz
 - Set the Input Data Type: **Complex Input**
 - Enable Transmit Continuously (optional)

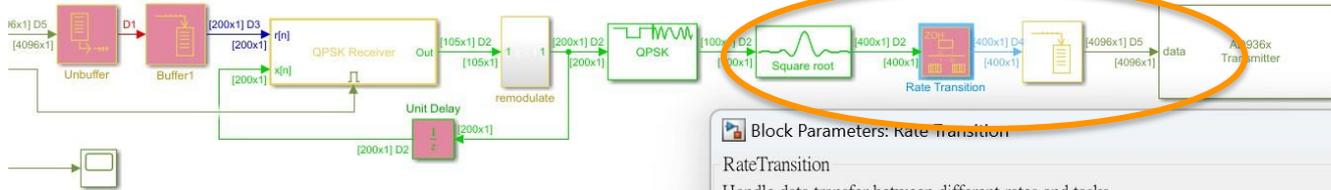
- Configure AD9361 Rx SDR block
 - Center Frequency: **5 GHz**
 - Baseband Sample Rate (i.e., data rate) = 520841 MHz
 - Set the Output Data Type: **Complex Output**

QPSK Relay SDR (0)

- FDX QPSK Relay SDR
 - **QPSK Receiver**: demodulates the received signal
 - **Unbuffer**: frame to sample conversion
 - **Buffer**: sample to frame conversion
 - **remodulate**: processes and forwards the received bits
 - **QPSK modulator**: convert bits to constellation
- Enable Sample Time Color Coding
 - Debug > Information Overlays > Text and Colors



QPSK Relay SDR (1)



Block Parameters: Rate Transition

RateTransition
Handle data transfer between different rates and tasks.

Parameters

- Ensure data integrity during data transfer
- Ensure deterministic data transfer (maximum delay)

Initial conditions:

0

Output port sample time options: Specify

Output port sample time:
0.0003839943476032033*2 0.00076799

OK Cancel Help Apply

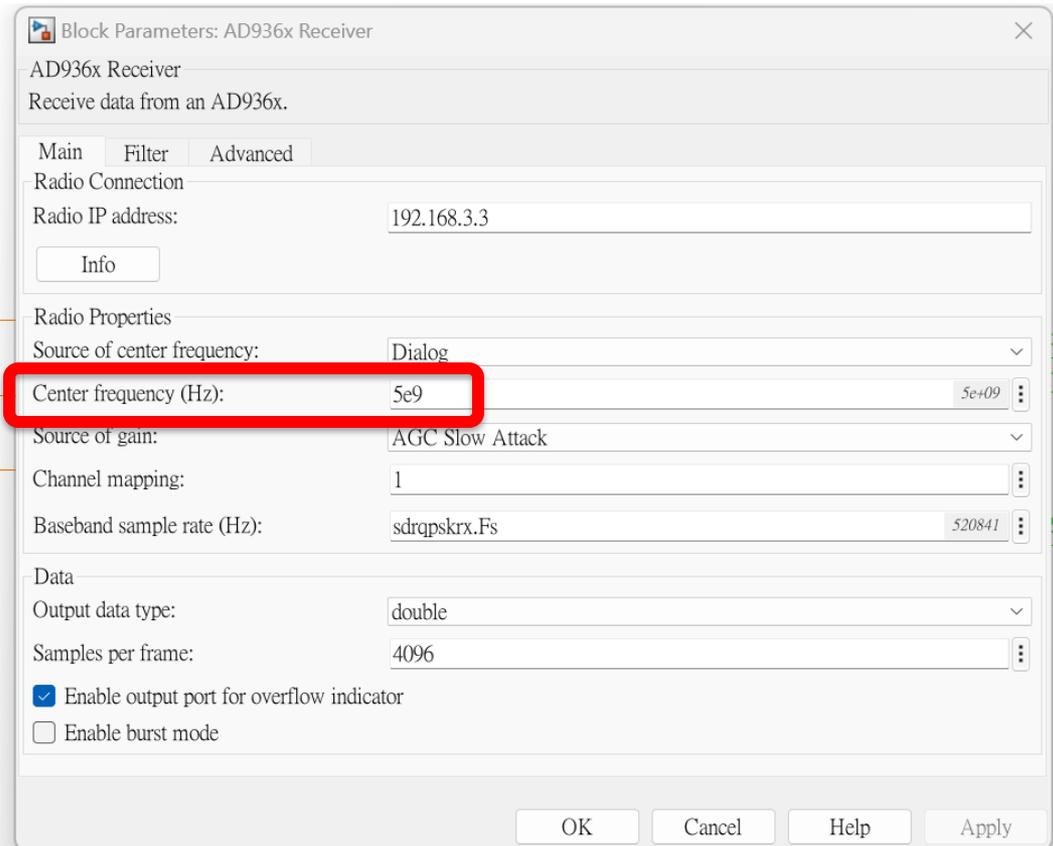
D1	1.9200e-06
D2	3.8399e-06
D3	3.8399e-04
D4	7.6799e-04
D5	7.8642e-03
Other	
Inf	Constant
T1	Triggered, Source: D2
M	Multirate

QPSK Relay SDR (2)

- AD9361 Receiver
 - Bandpass (RF) to baseband conversion

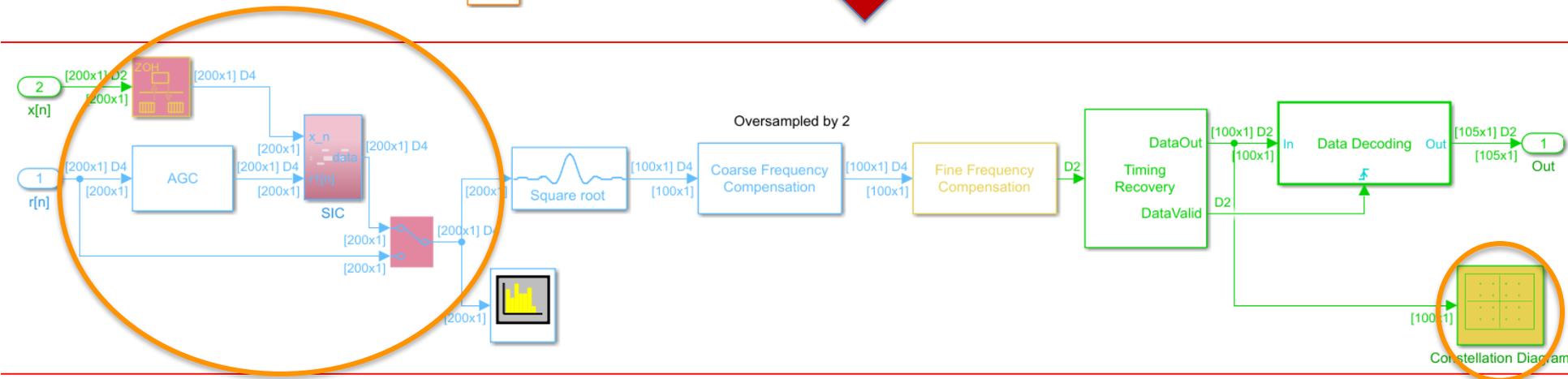
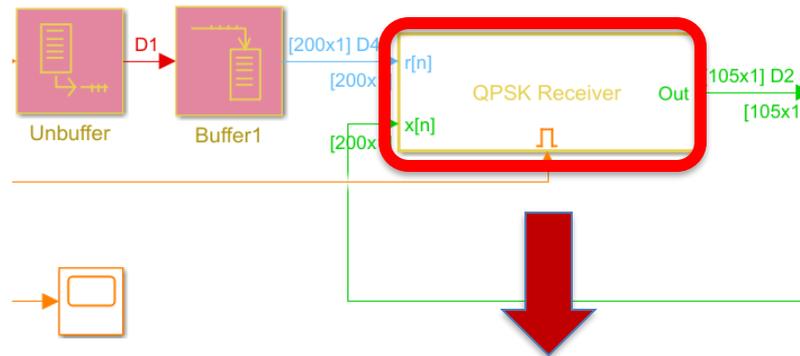


*** Center frequency change to 5e9



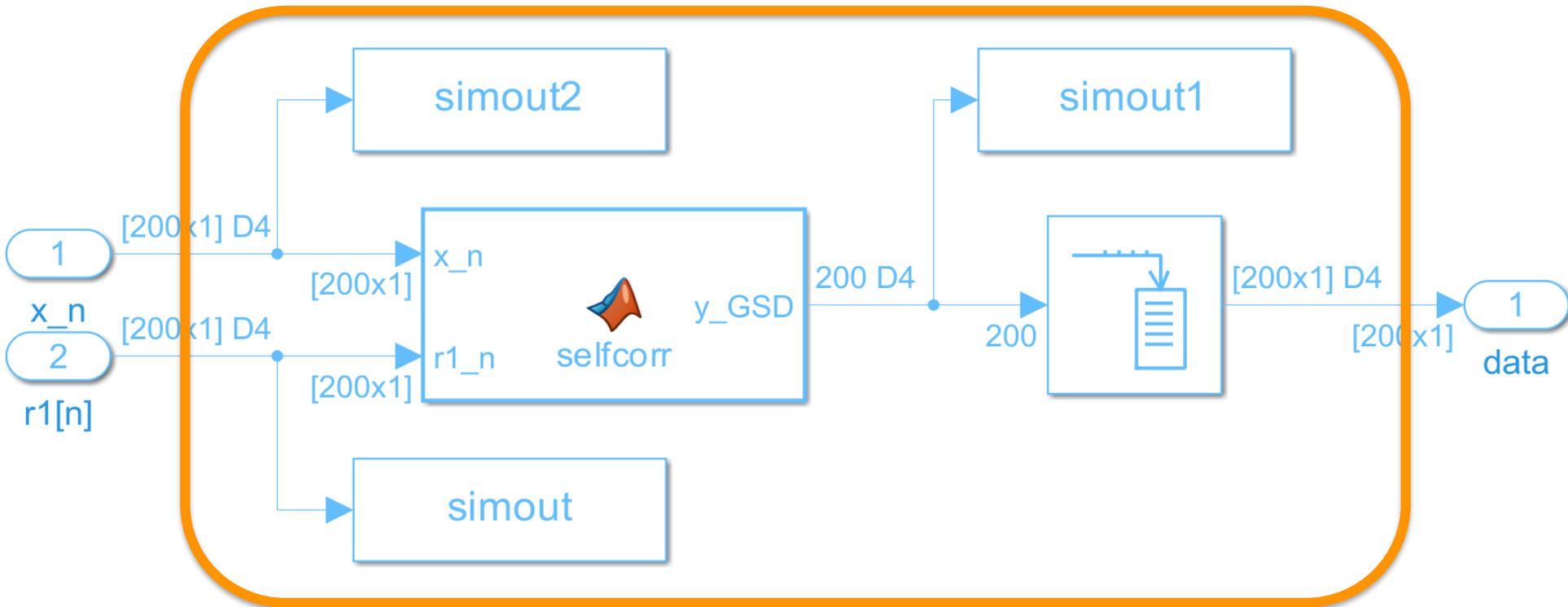
QPSK Relay SDR (4)

- QPSK Receiver
 - Perform QPSK demodulation



QPSK Relay SDR (6)

- QPSK Receiver
 - SIC (Self-Interference Cancellation)



Side note: Digital SIC Algorithms

- Stochastic Gradient Descent (SGD)
- Recursive Least Squares (RLS)

Table 1 SGD Algorithm for Iteratively Updating \mathbf{a}_n .

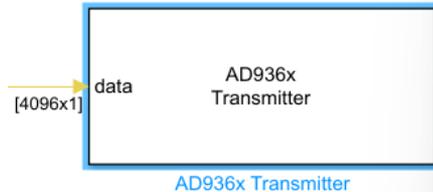
-
- 1: Initialization:
 - 2: $\mathbf{a}_n = \mathbf{0}$;
 - 3: Iteration loop:
 - 4: $\mathbf{a}_n = \mathbf{a}_{n-1} + \mu(r[n] - \mathbf{a}_{n-1}^* \bar{\mathbf{x}}_n) \bar{\mathbf{x}}_n$.
-

Table 2 RLS Algorithm for Iteratively Updating \mathbf{a}_n .

-
- 1: Initialization:
 - 2: $\mathbf{a}_n = \mathbf{0}$, $\mathbf{P}_0 = \varepsilon^{-1} \mathbf{I}_{L_u}$, $0 \leq \varepsilon \leq 1$;
 - 3: Iteration loop:
 - 4: $\xi_n = (1 + \lambda^{-1} \bar{\mathbf{x}}_n^* \mathbf{P}_{n-1} \bar{\mathbf{x}}_n)^{-1} \lambda^{-1} \mathbf{P}_{n-1} \bar{\mathbf{x}}_n$;
 - 5: $\mathbf{a}_n = \mathbf{a}_{n-1} + \xi_n (r[n] - \mathbf{a}_{n-1}^* \bar{\mathbf{x}}_n)^*$;
 - 6: $\mathbf{P}_n = \frac{1}{\lambda} (\mathbf{P}_{n-1} - \xi_n \bar{\mathbf{x}}_n^* \mathbf{P}_{n-1})$.
-

QPSK Relay SDR (9)

- AD9361 Transmitter



Block Parameters: AD936x Transmitter

AD936x Transmitter
Transmit data using an AD936x.

Main Filter Advanced

Radio Connection

Radio IP address: 192.168.3.3

Info

Radio Properties

Source of center frequency: Dialog

Center frequency (Hz): 5e9 5e+09

Source of gain: Dialog

Gain (dB): -10

Channel mapping: 2

Baseband sample rate (Hz): 520841

Data

Enable output port for underflow indicator

OK Cancel Help Apply

QPSK Receiver SDR (1)

- QPSK Receiver simulink model (HDX)

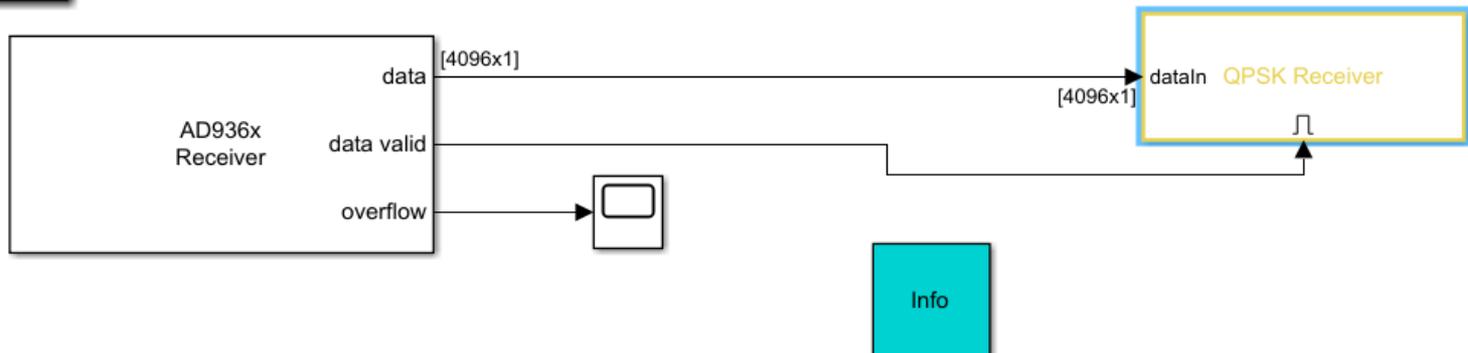
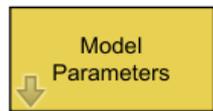
QPSK Receiver Using Analog Devices AD9361/AD9364

Note: Before running the QPSK models, first run the companion models for frequency offset calibration.

Open the companion `zynqRadioFrequencyCalibrationTxAD9361AD9364SLmodel`

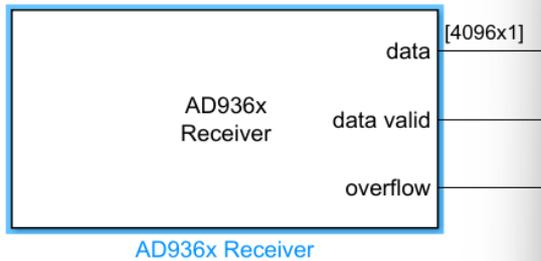
Open the companion `zynqRadioFrequencyCalibrationRxAD9361AD9364SL model`

Open the companion `zynqRadioQPSKTxAD9361AD9364SL model`



QPSK Receiver SDR (2)

- QPSK Receiver simulink model (HDX)



Block Parameters: AD936x Receiver

AD936x Receiver
Receive data from an AD936x.

Main Filter Advanced

Radio Connection

Radio IP address: 192.168.3.4

Info

Radio Properties

Source of center frequency: Dialog

Center frequency (Hz): 5e9-15500 5e+09

Source of gain: AGC Slow Attack

Channel mapping: 1

Baseband sample rate (Hz): sdrqpskrx.Fs 520841

Data

Output data type: double

Samples per frame: 4096

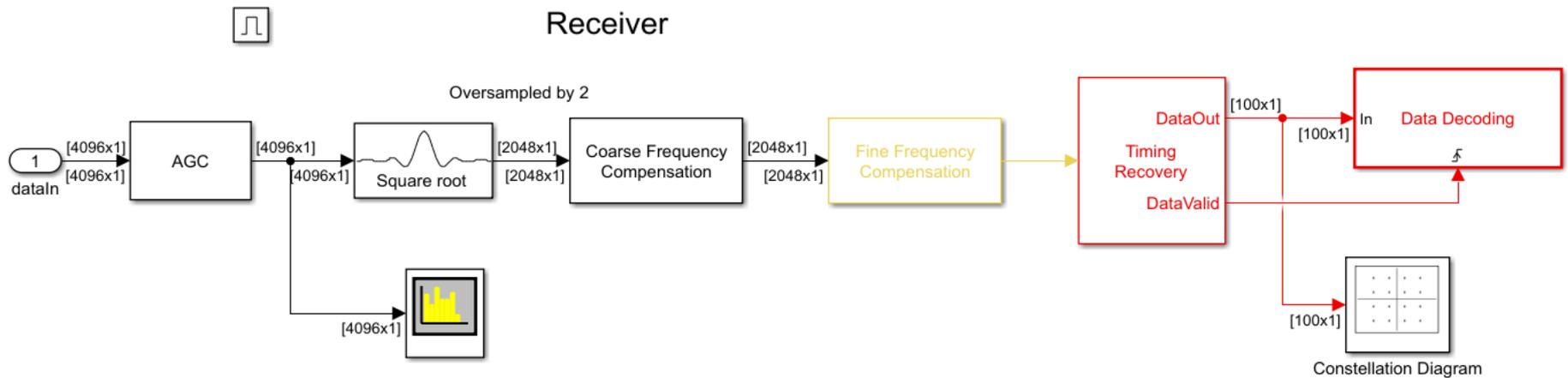
Enable output port for overflow indicator

Enable burst mode

OK Cancel Help Apply

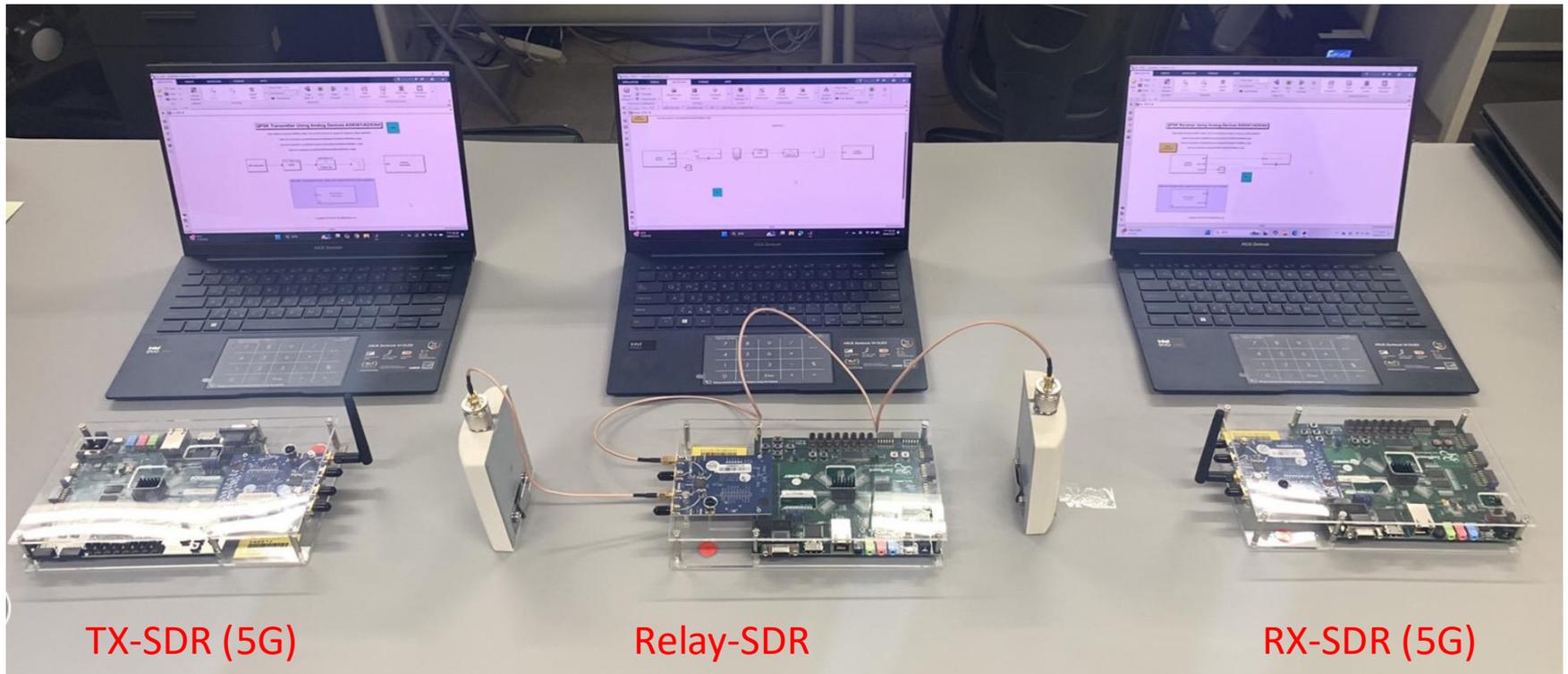
QPSK Receiver SDR (3)

- QPSK Receiver simulink model (HDX)



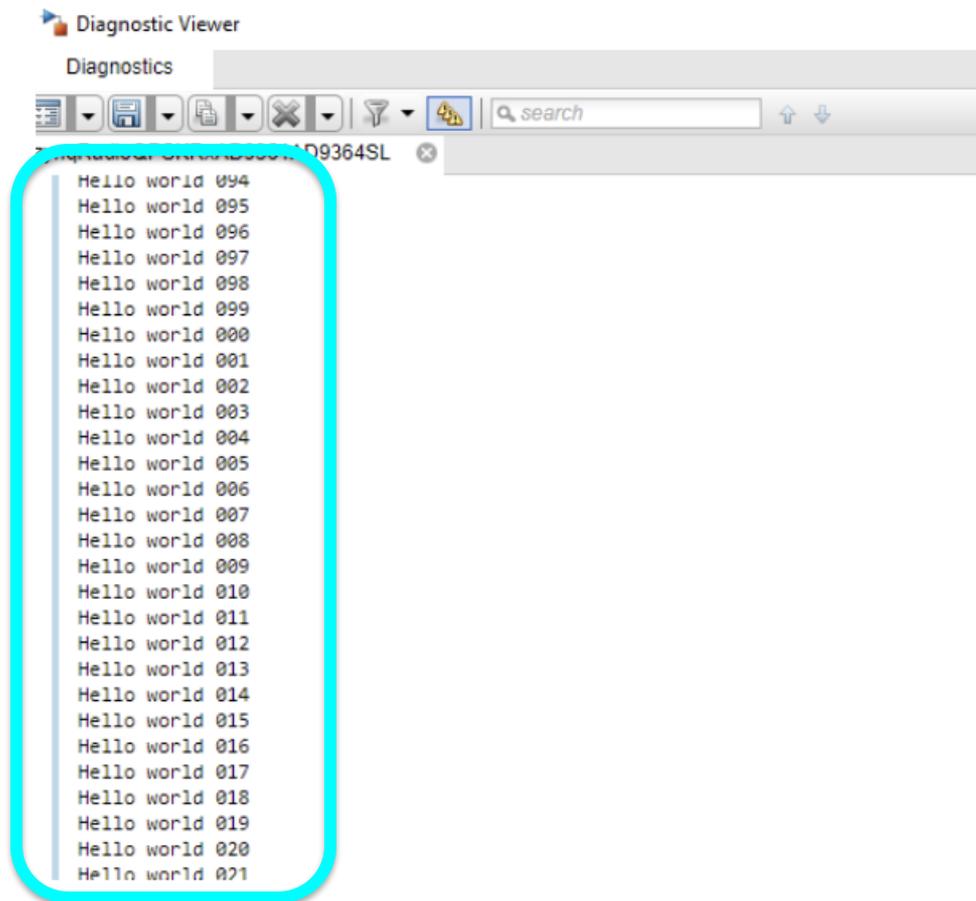
IB FDX QPSK OTA Test (1)

- Execute Tx SDR , Relay SDR and Rx SDR simultaneously.
 - using ‘sim’ mode to display the message in the **command window**
 - in simulink: **Debug > Diagnostics > Diagnostic Viewer**



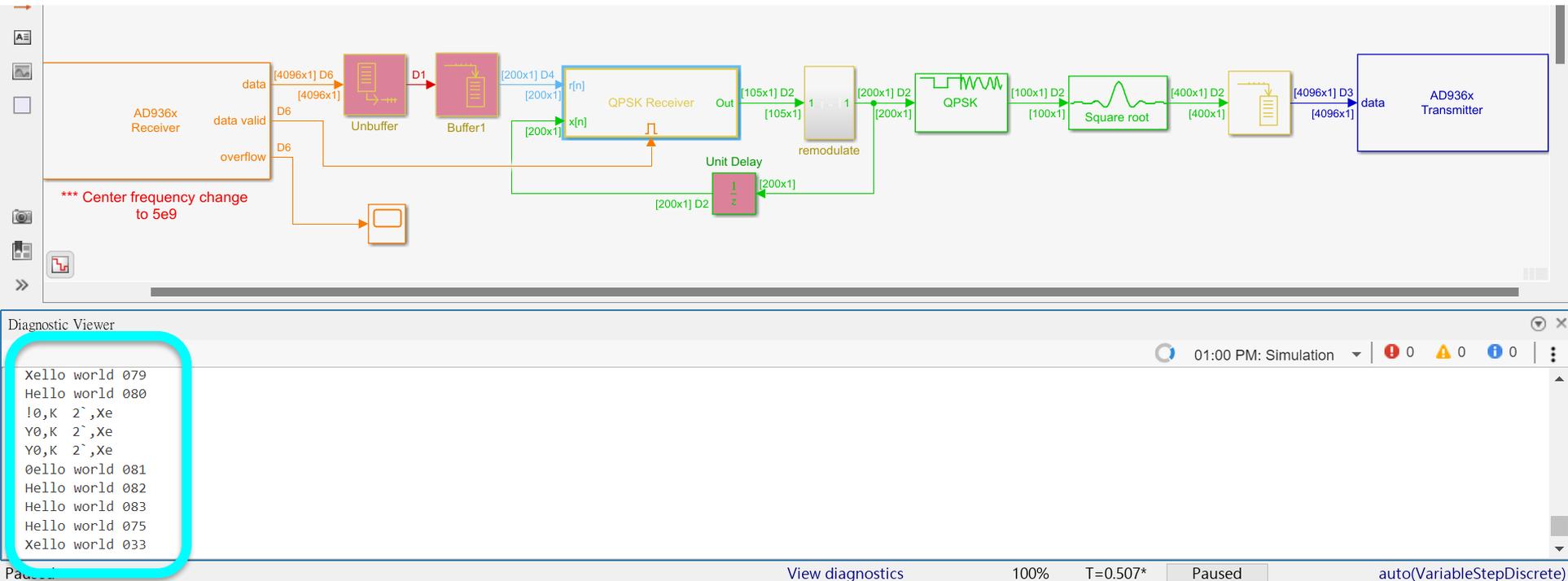
IB FDX QPSK OTA Test (2)

- OTA results appeared at Tx SDR



IB FDX QPSK OTA Test (3)

- OTA results appeared at Relay SDR



Paused

View diagnostics

100%

T=0.507*

Paused

auto(VariableStepDiscrete)

IB FDX QPSK OTA Test (4)

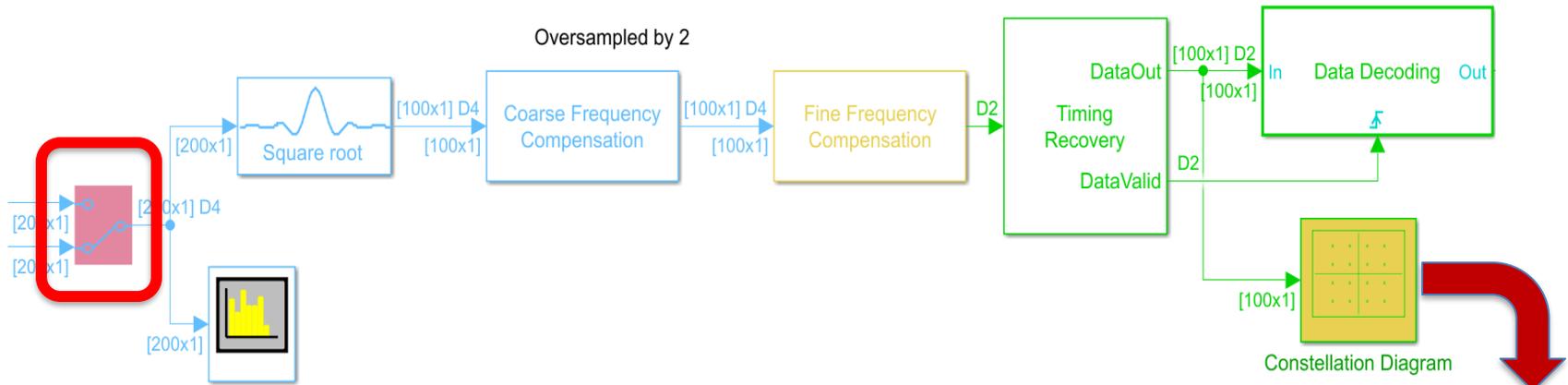
The screenshot displays the Simulink environment for a QPSK receiver test. The main workspace shows a block diagram with an 'AD936x Receiver' block connected to a 'QPSK Receiver' block. A 'Model Parameters' block is also visible. The diagnostic viewer at the bottom left shows the following output:

```
zEVQ5T-5-"k(Zj  
kTu ] (j :@.  
55-"k(ZjN'jAr  
ello world 070  
ello world 071  
oNSII'+Qjh3NP-  
xello world 014  
Hello world 015  
Hello world 016  
Hello world 017  
Hello world 018  
g  
hrXPVE
```

The diagnostic output is highlighted with a red circle. The status bar at the bottom indicates the simulation is paused at 01:44 PM, with a time of T=42.164* and a zoom level of 80%.

IB FDX QPSK OTA Test (5)

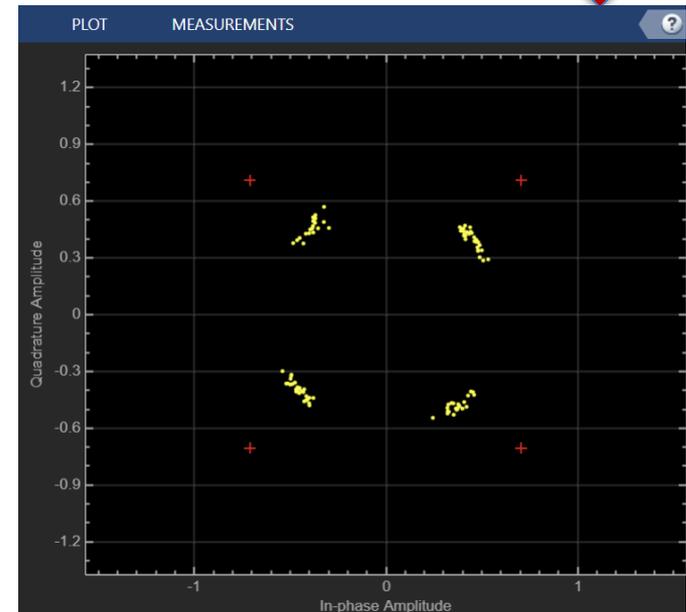
- Without SIC



- Diagnostic Viewer @ Relay-SDR

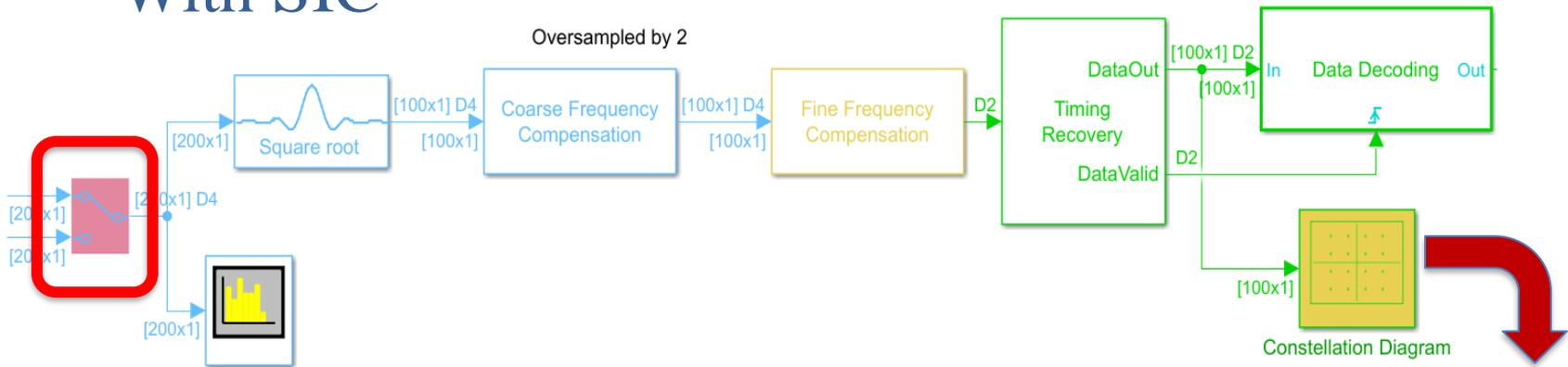
```
*li&fIYR6
M3$li&fIYR6
M3$li&fIYR6
MM3$li&fIYR6
MM3$liCm3 :7
```

```
hello world 085
Hello world 086
Hello wo_saE(1
~}C_0wI{>aoX
V}C 0wl{>aoX
```



IB FDX QPSK OTA Test (6)

- With SIC



- Diagnostic Viewer @ Relay-SDR

```
Zmt [=o%[i6z-^K
```

```
v"+F%mdEu
```

```
Xello world 051
```

```
Hello world 052
```

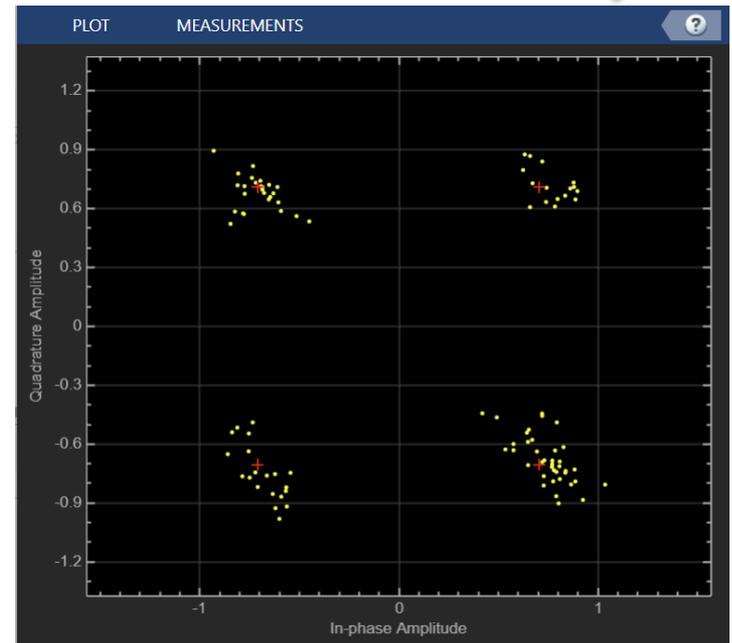
```
Hello world 053
```

```
Hello world 054
```

```
Hello world 055
```

```
Hello world 056
```

```
Hello world 057
```



IB FDX QPSK OTA Test (7)

```
1 function y_GSD = selfcorr(x_n, r1_n)
2     if isempty(x_n)
3         x_n = 0;
4     end
5     N = 200;
6     La = 5;
7     mu = 0.01;
8     a_GSD = complex(zeros(La, 1));
9     y_GSD = zeros(N, 1);
10    for index = 1:N
11        if index < La
12            y_GSD(index) = r1_n(index);
13        else
14            disp('a_GSD:');
15            disp(a_GSD);
16            y_GSD(index) = r1_n(index) - a_GSD' * x_n(index:-1:index-La+1);
17            a_GSD = a_GSD + mu * conj(y_GSD(index)) * x_n(index:-1:index-La+1);
18        end
19    end
20 end
```

Diagnostic Viewer

```
0.0103 + 0.1122i
0.0098 + 0.1084i
0.0058 + 0.1060i
a_GSD:
0.0133 + 0.1091i
0.0129 + 0.1099i
0.0139 + 0.1118i
0.0134 + 0.1080i
0.0094 + 0.1056i
```