BPSK system

1 Introduction

This document describes the simulation BPSK system in back-to-back configuration.

2 Functional Description

A simplified diagram of the system being simulated is presented in the Figure 1. The system simulated takes a random binary string and encodes it in an optical bandpass signal, signal that afterwards is decoded in order to re-obtain the original binary string.

The decoding of the optical signal is accomplished by an homodyne receiver, which combines the signal with a local oscillator with a user-determined phase. The homodyne receiver block output is then fed into a block that compares it with the original binary string and computes the Bit Error Rate (BER) along with it's upper and lower bounds for a certain user defined confidence level.

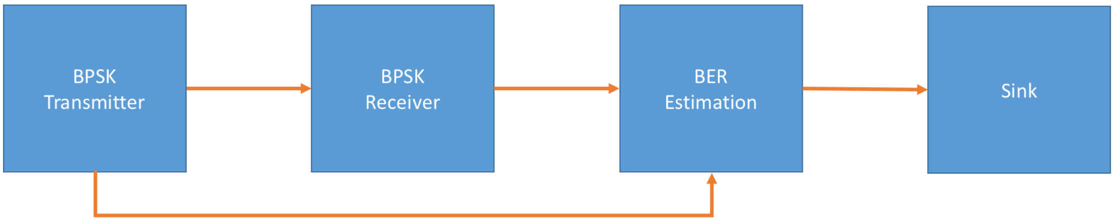


Figure 1: Overview of the BPSK system being simulated.

System Blocks	netxpto Blocks
BPSK Transmitter	MQamTransmitter
BPSK Receiver	HomodyneReceiver
BER Estimator	BitErrorRate

3 Required files

Header Files

File	Description
netxpto.h	Generic purpose simulator definitions.
m_qam_transmitter.h	Generates the signal with coded constellation.
homodyne_reciever.h	Performs coherent detection on the input signal.
bit_error_rate.h	Calculates the bit error rate of the decoded string.
sink.h	Closes any unused signals.

Source Files

File	Description
netxpto.cpp	Generic purpose simulator implementations.
m_qam_transmitter.cpp	Generates the signal with coded constellation.
homodyne_reciever.cpp	Performs coherent detection on the input signal.
bit_error_rate.cpp	Calculates the bit error rate of the decoded string.
sink.cpp	Closes any unused signals.

4 System Input Parameters

This system takes into account the following input parameters:

System Parameters	Description
numberOfBits	Gives the number of bits to be simulated
bitPeriod	Sets the time between adjacent bits
samplesPerSymbol	Establishes the number of samples each bit in the string is given [1]
pLength	PRBS pattern length
iqAmplitudesValues	Sets the state constellation
outOpticalPower_dBm	Sets the optical power, in units of dBm, at the transmitter output
loOutOpticalPower_dBm	Sets the optical power, in units of dBm, of the local oscillator used in the homodyne detector
localOscillatorPhase	Sets the initial phase of the local oscillator used in the homodyne detector
transferMatrix	Sets the transfer matrix of the beam splitter used in the homodyne detector
responsivity	Sets the responsivity of the photodiodes used in the homodyne detector
amplification	Sets the amplification of the trans-impedance amplifier used in the homodyne detector
noiseAmplitude	Sets the amplitude of the gaussian thermal noise added in the homodyne detector
delay	Sets the delay factor of the homodyne detector
posReferenceValue	Set the positive and negative reference values for the bit decision block
negReferenceValue	
confidence	Sets the confidence interval for the calculated QBER
midReportSize	Sets the number of bits between generated QBER mid-reports

5 Inputs

This system takes no inputs.

6 Outputs

This system outputs the following objects:

- Signals:
  - Initial Binary String; (S_0)
  - Optical Signal with coded Binary String; (S_1)

[1] Simulation time resolution = $\frac{\text{BitPeriod}}{\text{SamplesPerSymbol}}$

  - Decoded Binary String; (S_2)
  - BER result String; (S_3)

- Other:
  - Bit Error Rate report in the form of a .txt file. (BER.txt)

6.1 BER evolution

The following results show the dependence of the error rate with the signal power assuming a constant Local Oscillator power of -20 dBm. For reference, the eye diagram at 3 different power levels are also presented. The full line represents the expected results, note that it has been computed assuming a gaussian distribution of the thermal noise, which is not exact given the effect of the matched filter applied before the decoding of the bits, this explains the deviation between the simulation results and the expected values.

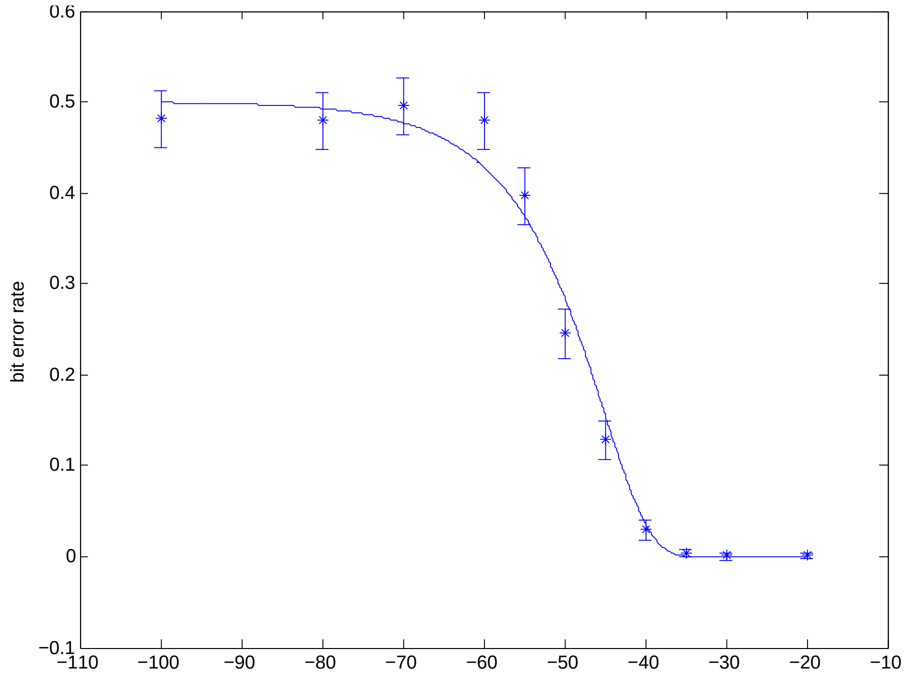
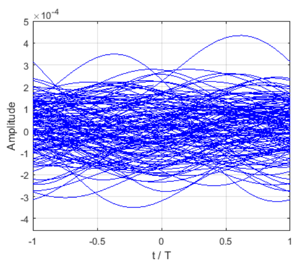
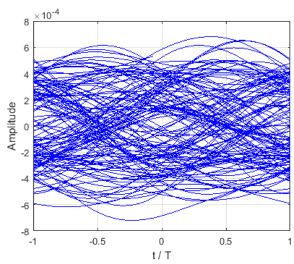


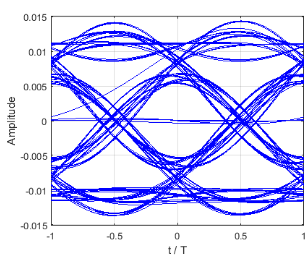
Figure 2: Bit Error Rate in function of the signal power in dBm.



(a) Power=-180dBm



(b) Power=-130dBm



(c) Power=-100dBm

Figure 3: Eye diagrams at different signal powers.

7 Simulation Results

We consider the following scenarios:

- 7.1 Basic BPSK back to back with thermal noise.

7.1 BPSK with thermal noise

The following results were obtained from the simulation using the following input parameters:

numberOfBits=	1000
samplesPerSymbol=	16
pLength=	5
iqAmplitudesValues=	{ { 1, 0 }, { -1, 0 } }
outOpticalPower_dBm=	-20
loOutOpticalPower_dBm=	-10
localOscillatorPhase=	0
transferMatrix=	{ { 1/sqrt(2), 1/sqrt(2), 1/sqrt(2), -1/sqrt(2) } }
responsivity=	1
amplification=	1e6
noiseAmplitude=	15.397586549153788
delay=	9

The system took the binary string presented in Figure 4 and encoded it into the optical signal in Figure 5. Notice the BPSK constelation of the signal, presented in Figure 6.

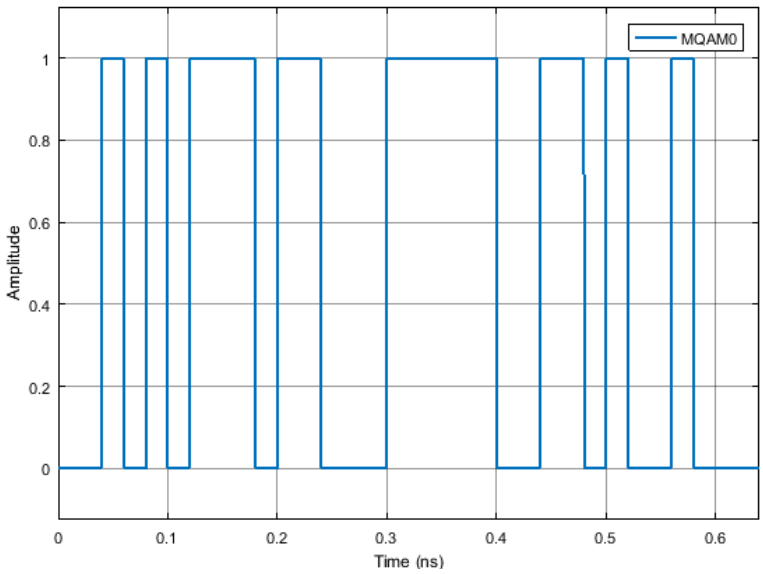


Figure 4: Sent binary key.

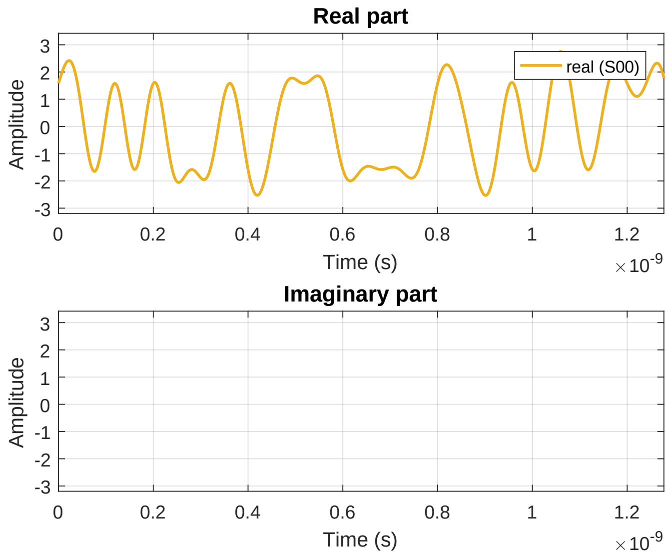


Figure 5: Sent signal.

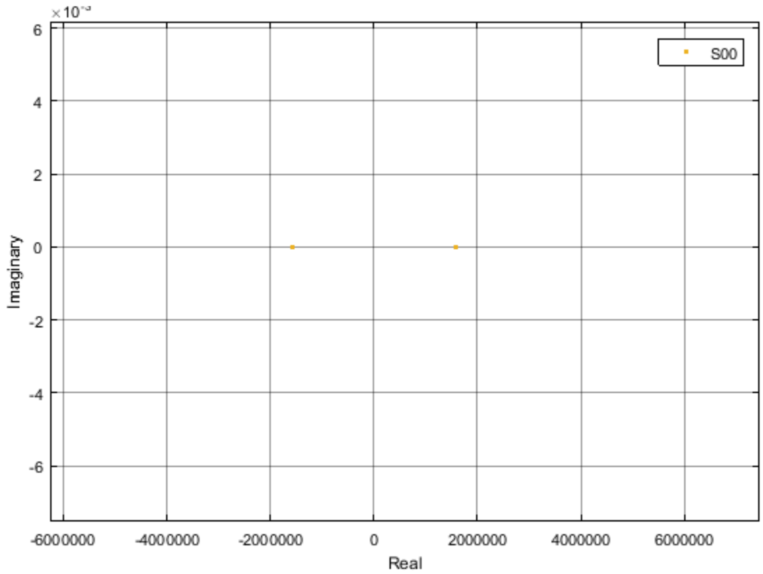


Figure 6: Constellation of the sent signal.

Homodyne detection is then performed, using to that effect the local oscillator signal presented in Figure 7. Figures 8 and 9 show the addition of noise to the signal.

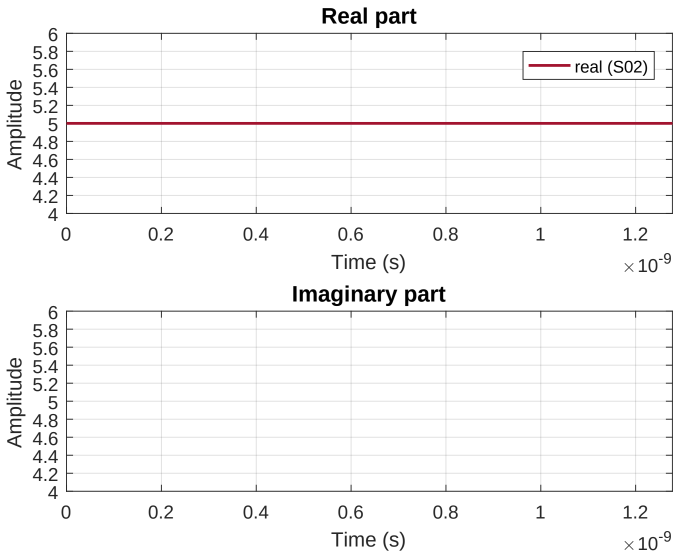


Figure 7: Homodyne receiver internal signal: local oscillator used for Homodyne detection.

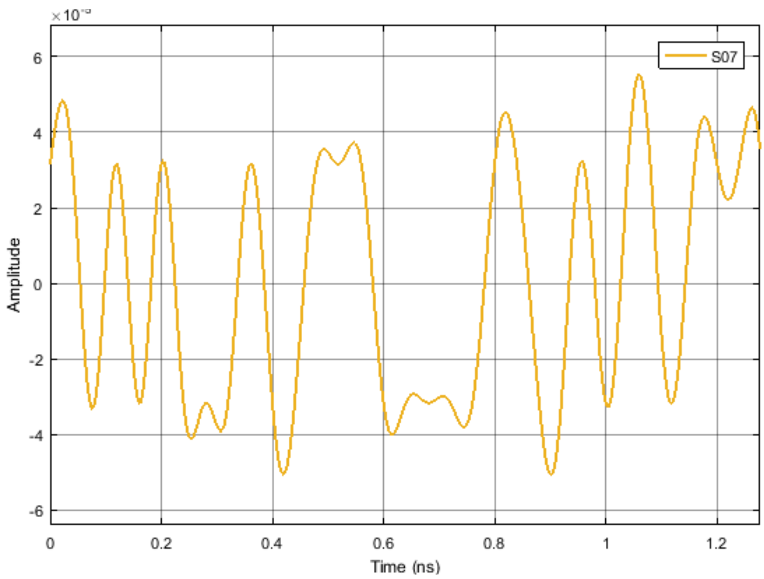


Figure 8: Homodyne receiver internal signal: subtraction of the signals outputted by the photodiodes.

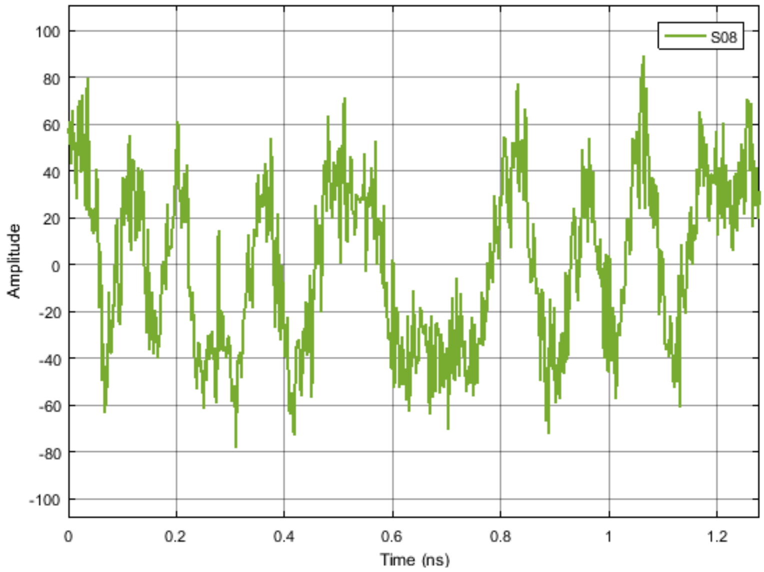


Figure 9: Homodyne receiver internal signal: amplification of the signal in Figure 8 with added noise.

The result of the homodyne detection is the binary string presented in 10, which is then compared to the original binary string by the BER block, which outputs the report presented in Figure 11.

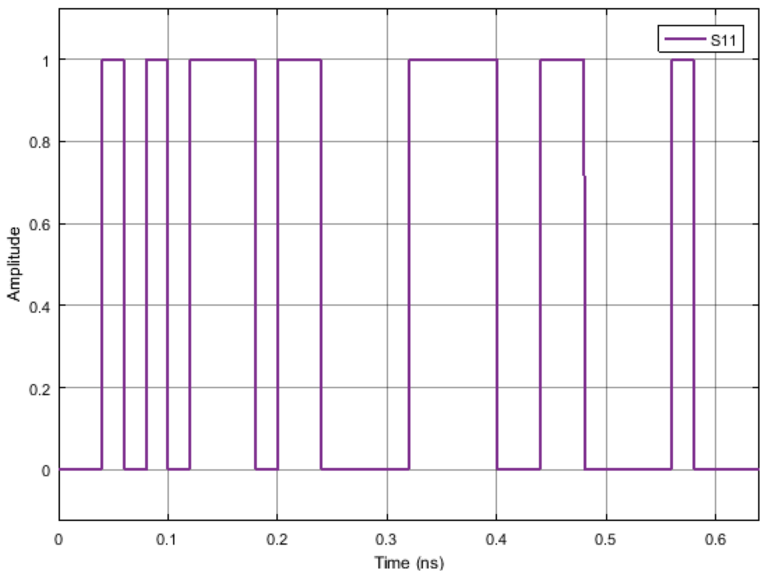


Figure 10: Decoded binary string, output of the Homodyne receiver block.

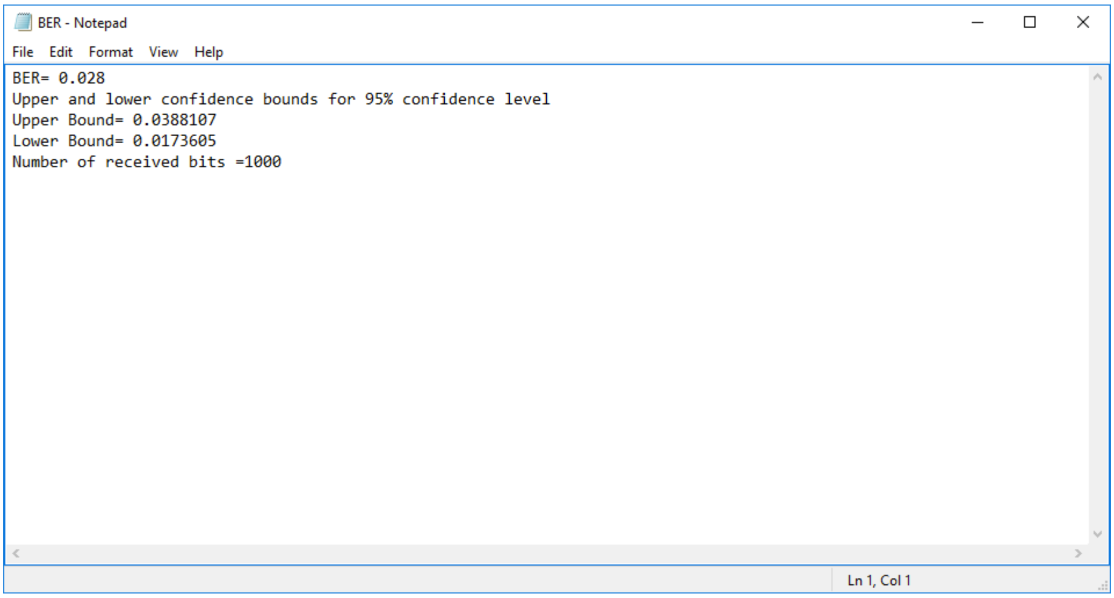


Figure 11: Bit-Error-Rate report.

8 Block Description

8.1 MQAM Transmitter

8.2 Homodyne Receiver

Introduction

This super-block compresses the function of the following blocks:

- Local Oscillator;
- Balanced Beamsplitter;
- Photodiode;
- Electrical Filter;
- Amplifier;
- Discretizer;
- Delayer
- Bit Decider;

This compression allows for a cleaner code.

Input Parameters

- LocalOscillatorOpticalPower
- LocalOscillatorOpticalPower_dBm
- LocalOscillatorPhase
- TransferMatrix
- Responsivity
- Amplification

- NoiseAmplitude
- SamplingRate
- Delay
- ReferenceValue

Functional Description

The input signal is evaluated and a binary string is generated from this evaluation.
A diagram of the blocks that constitute this super-block, with the corresponding relations is presented in Figure 12.

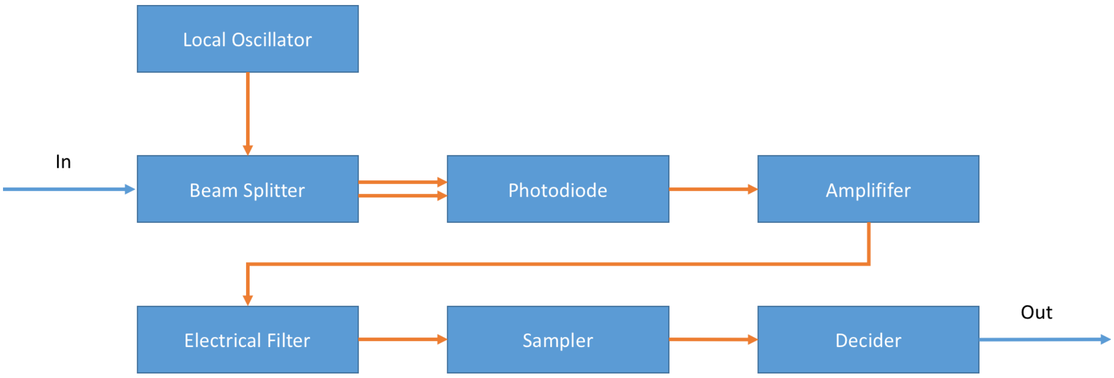


Figure 12: Homodyne Receiver Block Diagram.

Inputs

Number: 1
Type: Sequence of impulses modulated by the filter (OpticalSignal)

Outputs

Number: 1
Type: Binary String (Binary)

8.3 Bit Error Rate

Input Parameters

- setConfidence
- setMidReportSize

Functional Description

This block accepts two binary strings and outputs a binary string, outputting a 1 if the two input samples are equal to each other and 0 if not. This block also outputs *.txt* files with a report of the calculated BER as well as the estimated Confidence bounds for a given probability P. The block allows for mid-reports to be generated, the number of bits between reports is customizable, if it is set to 0 then the block will only output the final report.

Input Signals

Number: 1
Type: Binary (DiscreteTimeDiscreteAmplitude)

Output Signals

Number: 1
Type: Binary (DiscreteTimeDiscreteAmplitude)

8.4 Local Oscillator

Input Parameters

- LocalOscillatorPhase
- LocalOscillatorOpticalPower
- `LocalOscillatorOpticalPower_dBm`
- useShotNoise

Functional Description

This blocks accepts a complex signal (either with XY polarization or a simple Band Pass signal) and outputs a phase constant complex signal with the same length as the input signal. The phase and optical power are defined by the values of *LocalOscillatorPhase* and *LocalOscillatorOpticalPower* respectively. This block also adds random gaussian distributed shot noise with an amplitude defined by the power of the inputs. The shot noise is activated by a boolean check fed into the block by the *useShotNoise* parameters.

Input Signals

Number: 1
Type: Complex signal (ContinuousTimeContinuousAmplitude)

Output Signals

Number: 2
Type: Complex signal (ContinuousTimeContinuous Amplitude)

8.5 Beam Splitter

Input Parameters

- setTransferMatrix

For simplicity, the input of the transfer Matrix is in the form of a 4x1 array, with the following relation between the array, A, and matrix, M, elements:

$$A = \{\ \{\ \alpha,\ \beta,\ \gamma,\ \delta\ \}\ \} \Rightarrow M = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix} \tag{1}$$

Functional Description

This block accepts two complex signals and outputs two complex signals built from a mixture of the two inputs according to a pre-determined and user defined transfer matrix.

Input Signals

Number: 2
Type: Complex signal (ContinuousTimeContinuousAmplitude)

Output Signals

Number: 2
Type: Complex signal (ContinuousTimeContinuousAmplitude)

8.6 Photodiode

Input Parameters

- setResponsivity

Functional Description

This block accepts two complex signals and outputs one real signal built from an evaluation of the power of the input signals and their subsequent subtraction. The responsivity is defined by the value of *Responsivity*.

Input Signals

Number: 2
Type: Complex signal (ContinuousTimeContinuousAmplitude)

Output Signals

Number: 1
Type: Real signal (ContinuousTimeContinuousAmplitude)

8.7 Amplifier

Input Parameters

- setAmplification
- setNoiseAmplitude

Functional Description

This block accepts one real signal and outputs one real signal built from multiplying the input signals by a predetermined value. This block also adds random gaussian distributed noise with a user defined amplitude. The multiplying factor and noise amplitude are defined by the values of *Amplification* and *NoiseAmplitude* respectively.

Input Signals

Number: 1
Type: Real signal (ContinuousTimeContinuousAmplitude)

Output Signals

Number: 1
Type: Real signal (ContinuousTimeContinuousAmplitude)

8.8 Electrical Filter

Input Parameters

- setFilterType

Functional Description

This block accepts one real signal and outputs one real signal built from a finite convolution of the input with a filter function. The filter function is defined by *setFilterType*.

Input Signals

Number: 1
Type: Real signal (ContinuousTimeContinuousAmplitude)

Output Signals

Number: 1
Type: Real signal (ContinuousTimeContinuousAmplitude)

8.9 Sampler

Input Parameters

- setSamplingRate
- setDelay

Functional Description

This block accepts one real continuous signal and outputs one real discrete signal built from a sampling of the input signal with a predetermined sampling rate. The sampling rate is defined by the value *SamplingRate*. This block also allows for a controlled adjustment of the starting point of the output signal, defined by the value *Delay*

Input Signals

Number: 1
Type: Real signal (ContinuousTimeContinuousAmplitude)

Output Signals

Number: 1
Type: Real signal (DiscreteTimeContinuousAmplitude)

8.10 Bit Decider

Input Parameters

- setPosReferenceValue
- setNegReferenceValue

Functional Description

This block accepts one real discrete signal and outputs a binary string, outputting a 1 if the input sample is above the predetermined reference level and 0 if it is below another reference value The reference values are defined by the values of *PosReferenceValue* and *NegReferenceValue*.

Input Signals

Number: 1
Type: Real signal (DiscreteTimeContinuousAmplitude)

Output Signals

Number: 1
Type: Binary (DiscreteTimeDiscreteAmplitude)

8.11 Bit Error Rate

Input Parameters

- setConfidence
- setMidReportSize

Functional Description

This block accepts two binary strings and outputs a binary string, outputting a 1 if the two input samples are equal to each other and 0 if not. This block also outputs *.txt* files with a report of the calculated BER as well as the estimated Confidence bounds for a given probability P. The block allows for mid-reports to be generated, the number of bits between reports is customizable, if it is set to 0 then the block will only output the final report.

Input Signals

Number: 1
Type: Binary (DiscreteTimeDiscreteAmplitude)

Output Signals

Number: 1
Type: Binary (DiscreteTimeDiscreteAmplitude)

9 Known Problems

- Homodyne Super-Block not functioning
- MQAM Transmitter PDF needs to be written
- 8 bits being lost of every signal
- If the bit string length is larger than 512, this first 512 bits are lost