BPSK system

October 19, 2016

This document describes the simulation of a Key Distribution in back-to-back configuration. The signal is generated by the system described in the *MQAM_system.pdf* document, as such the blocks described in that document will not be described here.

1 System to analyse

In this step on our work we are attempting to simulate a *back-to-back* QKD setup, in which the signal output by the emitter is directly fed into the receiver.

For simplicity, in this early stage we have chosen to use a BPSK coding. This is a simplified setup but still allows us to study the effect of the relative phase between the local oscillator and the signal.

A simplified diagram of the system being simulated is presented in the Figure 1. In broad terms, our QKD simulation is required to be able to send a random bit sequence (the key) by means of an optical signal, accomplish a full key reconstruction from the optical signal and subsequently perform a key security check through the evaluation of the Bit Error Rate (BER).

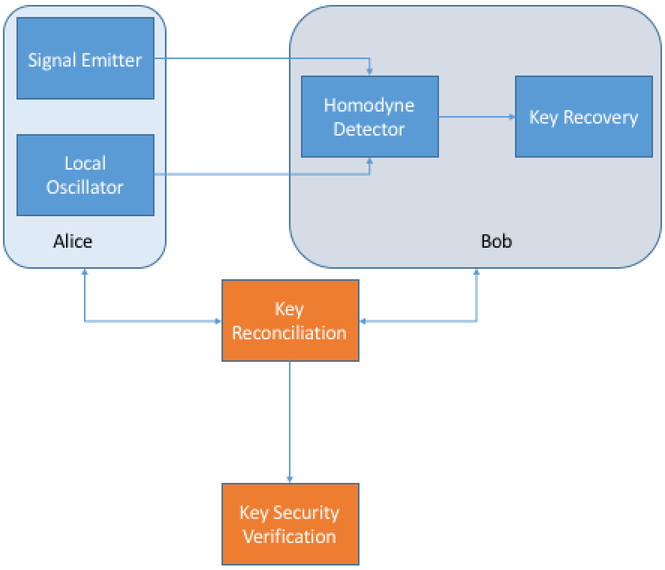

Figure 1: Overview of the QKD system being simulated.

In the next section we will show that our simulation can currently accomplish all of the proposed objectives.

2 Numerical model

The current simulation can be visually interpreted by the block diagram presented in Figure 2. In this section we will go into some detail on the workings of the blocks responsible for the Signal

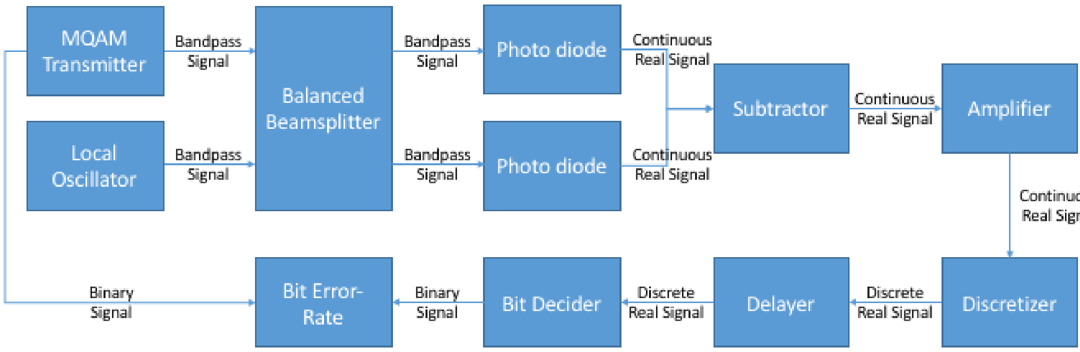

Figure 2: Visual representation of the simulated system.

detection and Key reconstruction (i.e. all blocks other than **MQAM Transmitter** and **Local Oscillator**).

Along the description of each block, we include the crucial coding (the code responsible for setting the signal parameters is omitted) and a figure showing the output of each block.

2.1 Local Oscillator

Input Parameters

- LocalOscillatorPhase
- LocalOscillatorOpticalPower
- `LocalOscillatorOpticalPower_dBm`

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.

Input Signals

Number: 1
Type: Complex signal (ContinuousTimeContiousAmplitude)

Output Signals

Number: 2
Type: Complex signal (ContinuousTimeContiousAmplitude)

2.2 Balanced Beamsplitter

Input Parameters

- setTransferMatrix

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 (ContinuousTimeContiousAmplitude)

Output Signals

Number: 2
Type: Complex signal (ContinuousTimeContiousAmplitude)

2.3 Photodiode

Input Parameters

- setResponsivity

Functional Description

This block accepts two complex signals and outputs two real signals built from an evaluation of the power of the input signals.

Input Signals

Number: 2
Type: Complex signal (ContinuousTimeContiousAmplitude)

Output Signals

Number: 2
Type: Real signal (ContinuousTimeContiousAmplitude)

2.4 Subtractor

Input Parameters

This block has no input parameters.

Functional Description

This block accepts two real signals and outputs one real signals built from subtracting the value of the input signals.

Input Signals

Number: 2
Type: Real signal (ContinuousTimeContiousAmplitude)

Output Signals

Number: 1
Type: Real signal (ContinuousTimeContiousAmplitude)

2.5 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 (ContinuousTimeContiousAmplitude)

Output Signals

Number: 1
Type: Real signal (ContinuousTimeContiousAmplitude)

2.6 Discretizer

Input Parameters

- setSamplingRate

Functional Description

This block accepts one real continuous signal and outputs one real discrete signal built from a sampling of the input signals with a predetermined sampling rate. The sampling rate is defined by the value *SamplingRate*.

Input Signals

Number: 1
Type: Real signal (ContinuousTimeContiousAmplitude)

Output Signals

Number: 1
Type: Real signal (DiscreteTimeContiousAmplitude)

2.7 Delayer

Input Parameters

- setDelay

Functional Description

This block accepts one real discrete signal and outputs one real discrete signal discarding a predetermined number of samples of the beginning of the input signal. The number of samples discarded rate is defined by the value *Delay*.

Input Signals

Number: 1
Type: Real signal (DiscreteTimeContiousAmplitude)

Output Signals

Number: 1
Type: Real signal (DiscreteTimeContiousAmplitude)

2.8 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 (DiscreteTimeContiousAmplitude)

Output Signals

Number: 1
Type: Binary (DiscreteTimeDiscreteAmplitude)

2.9 Bit Error-Rate

Input Parameters

- setZ

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 a *.txt* file with a report of the calculate BER as well as the estimated Confidence bounds for a given probability $1 - \alpha$. The probability for the confidence bounds is determined by the $z_{1-\frac{\alpha}{2}}$ percentile of a standard gaussian distribution, value which has to be fed into the block.

Input Signals

Number: 1
Type: Binary (DiscreteTimeDiscreteAmplitude)

Output Signals

Number: 1
Type: Binary (DiscreteTimeDiscreteAmplitude)