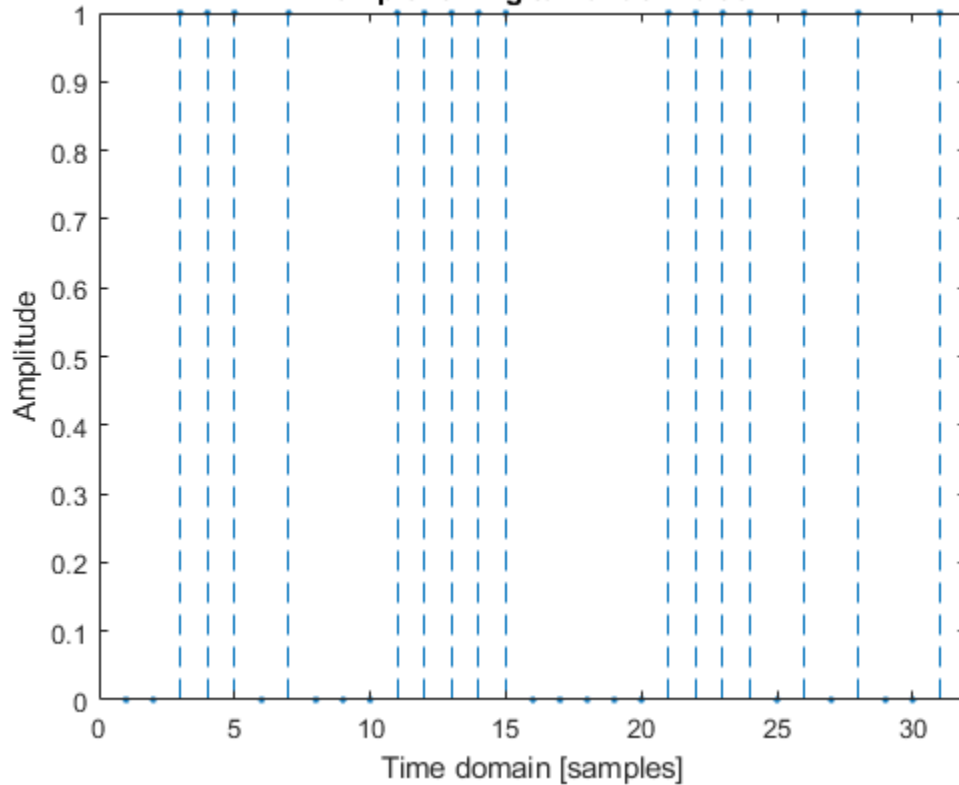# Table of Contents

# Section (A) - Adding new utils

We have implemented new utils using blocks from previous homeworks. Here are the diagram blocks and examples:

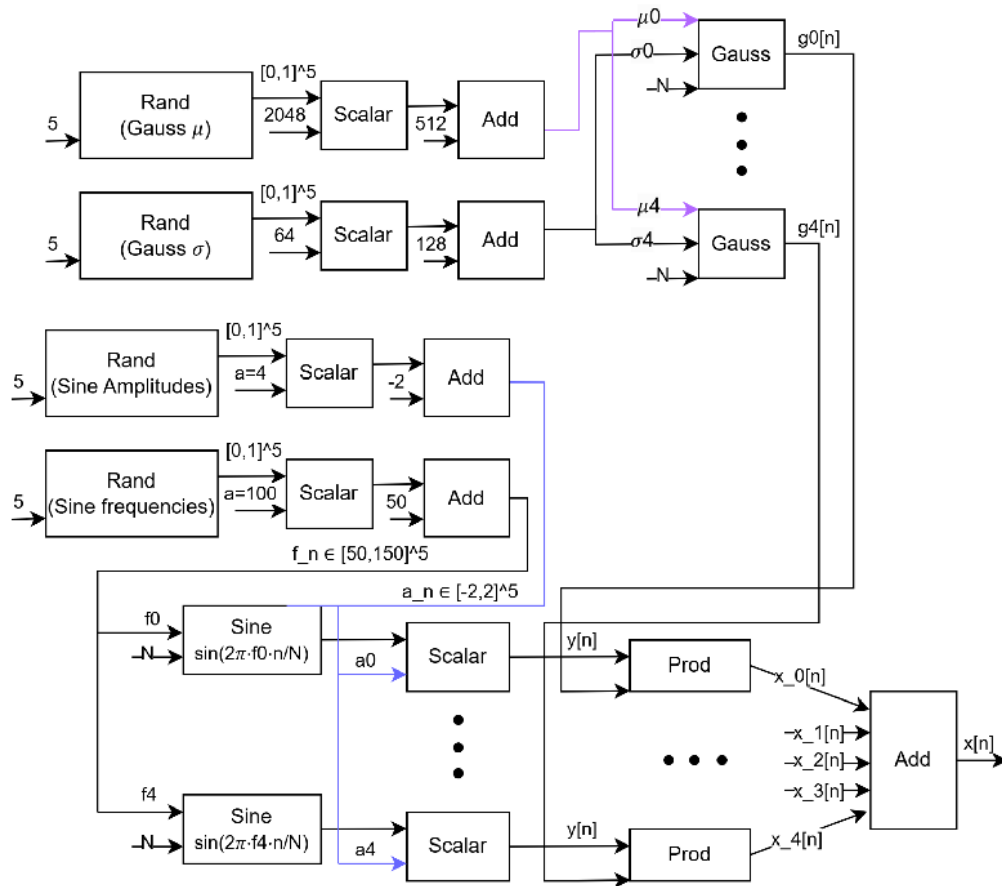**DigitalRandom block diagram**

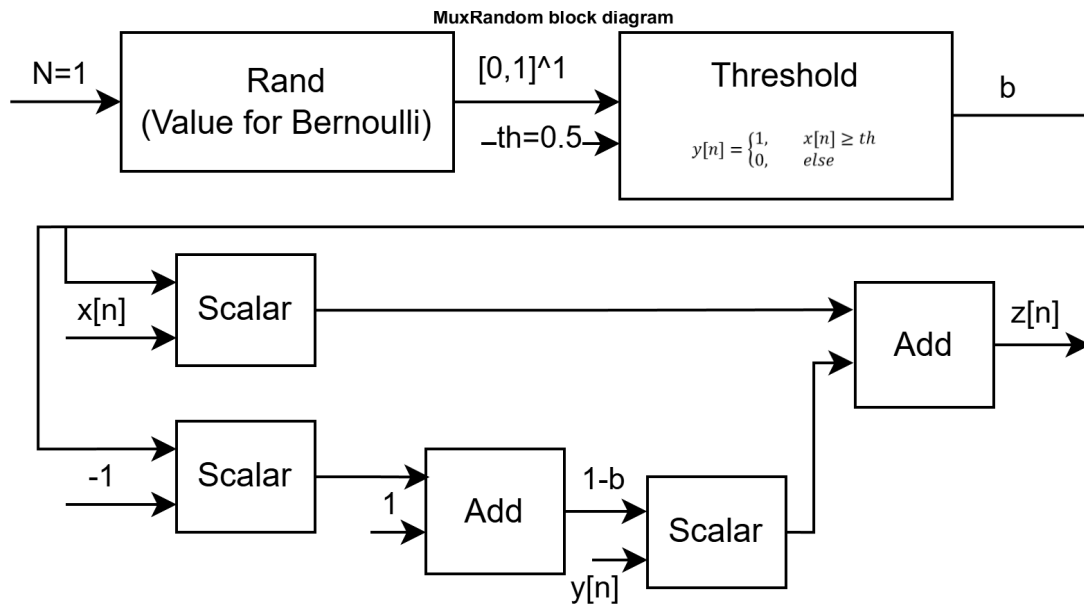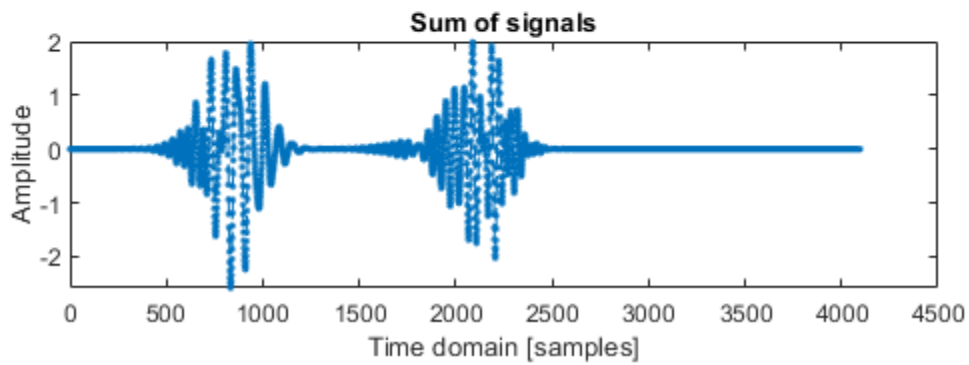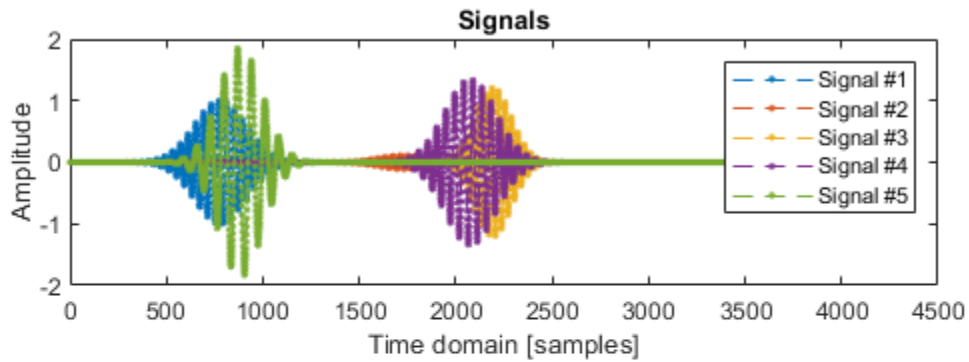N → | Rand | → $y\_n \in [0,1]^N$ → | Threshold $y[n] = \begin{cases} 1, & x[n] \geq th \\ 0, & else \end{cases}$ | → y_n
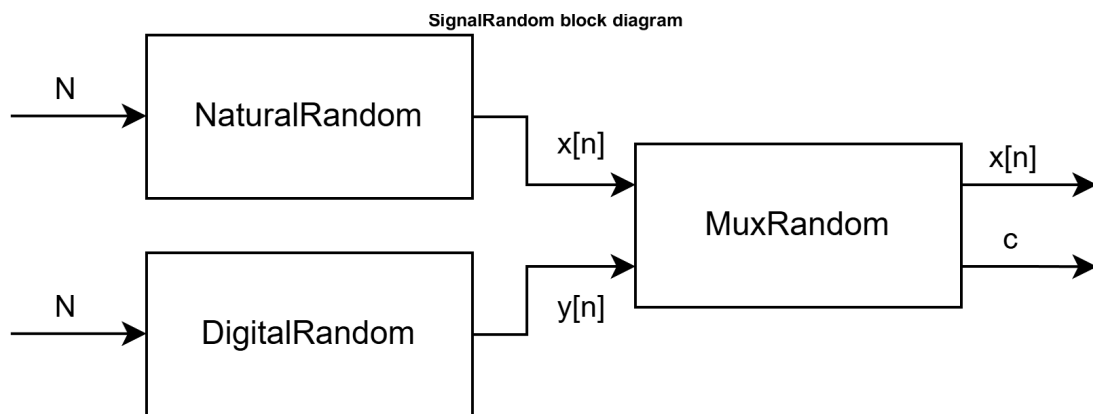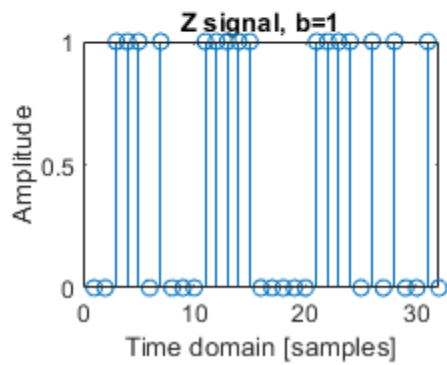
th=0.5
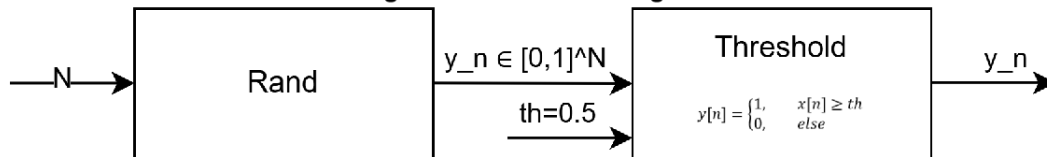

Example for DigitalRandom block

# NaturalRandom block diagram

## Example for NaturalRandom block

### Signals



### Sum of signals



**MuxRandom block diagram**



Rand
(Value for Bernoulli)

N=1

[0,1]^1

th=0.5

Threshold

$$y[n] = \begin{cases} 1, & x[n] \geq th \\ 0, & else \end{cases}$$

b

x[n]

Scalar

-1

Scalar

1

Add

1-b

Scalar

y[n]

Add

z[n]

## Example for MuxRandom block
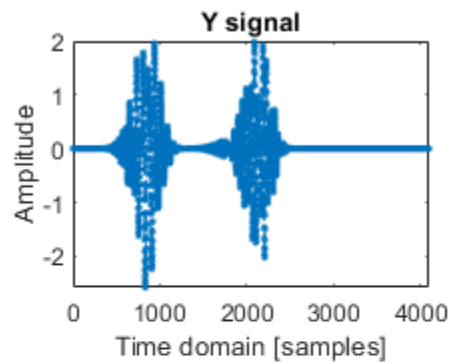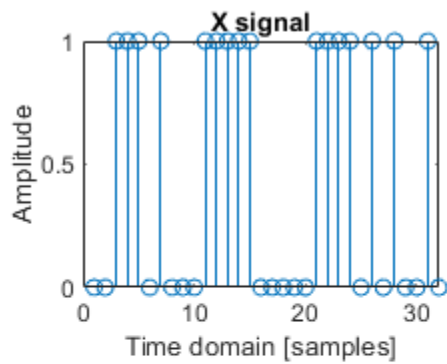
### X signal



### Y signal



### Z signal, b=1



**SignalRandom block diagram**

## Example for SignalRandom block
### c=1
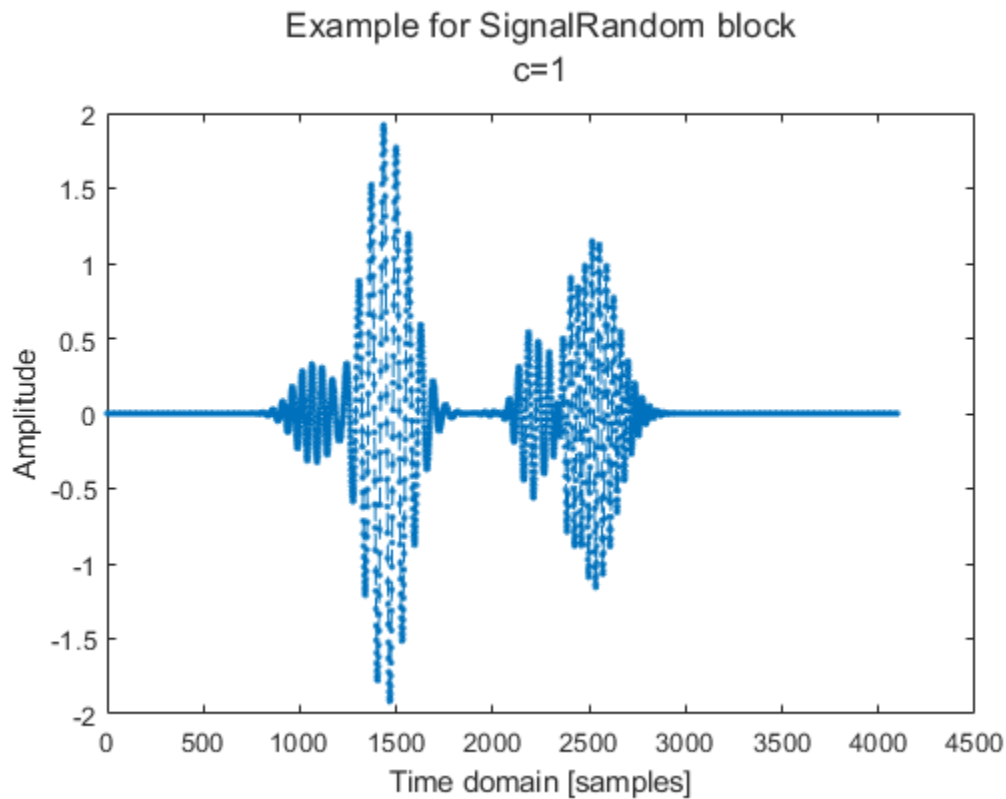


# GreaterThan block

As Example for GreaterThan block, we select 2 signals - Natural signal from Natural signal block and signal of zeros (Using Sine block with DC frequency). We expect that applying GreaterThan block will return only politive values, the negative one will be replaced with zeros.

**GreaterThan block diagram**



$$y[n] = \begin{cases} 1, & x[n] \geq th \\ 0, & else \end{cases}$$
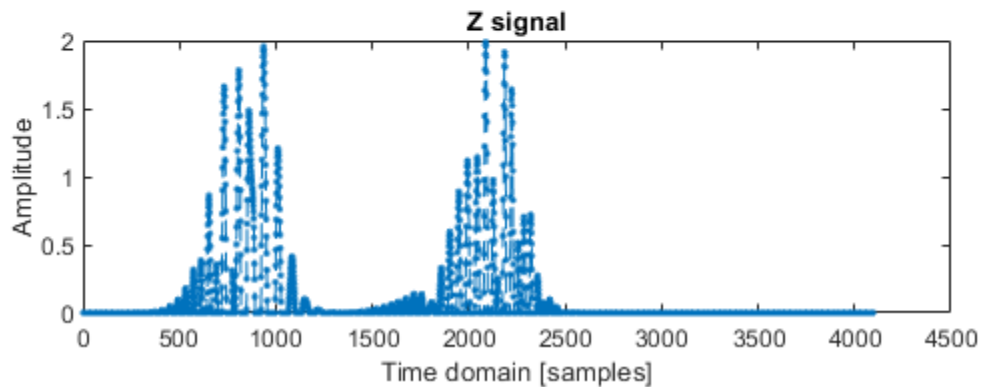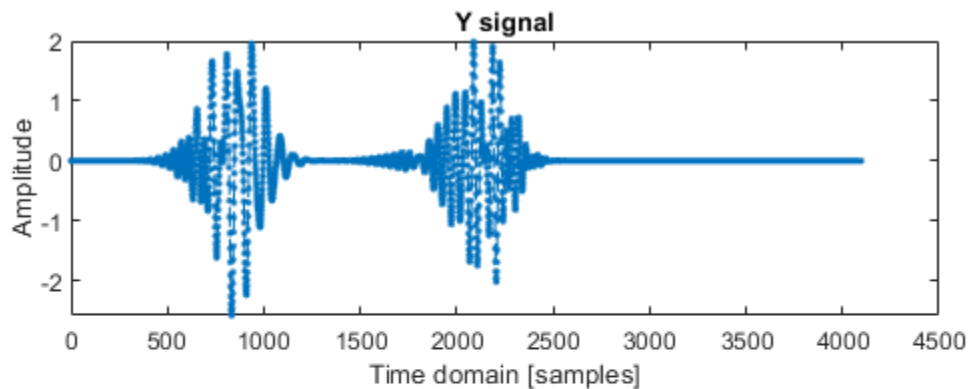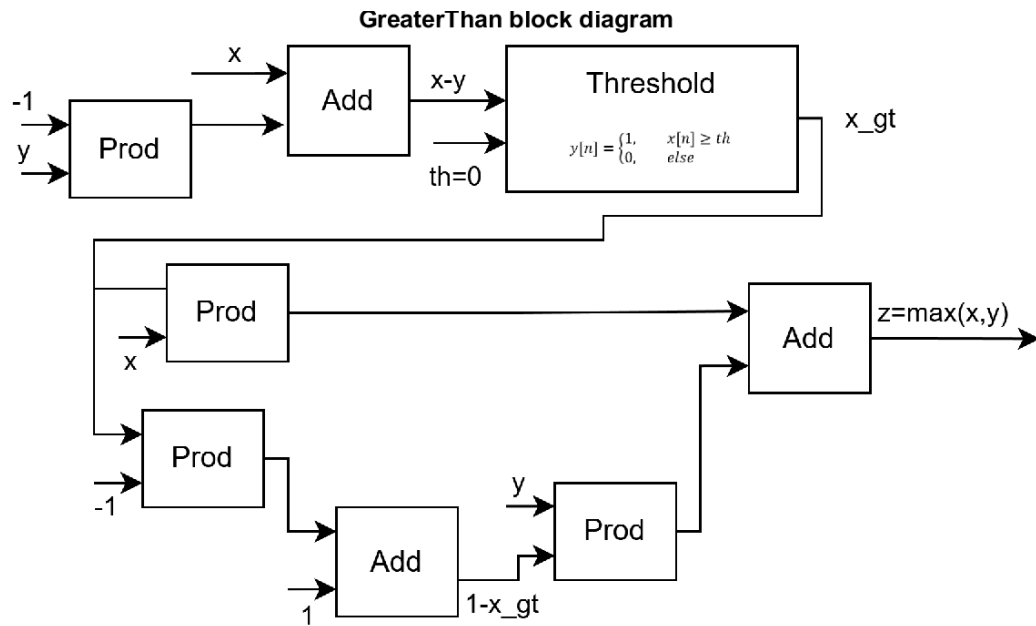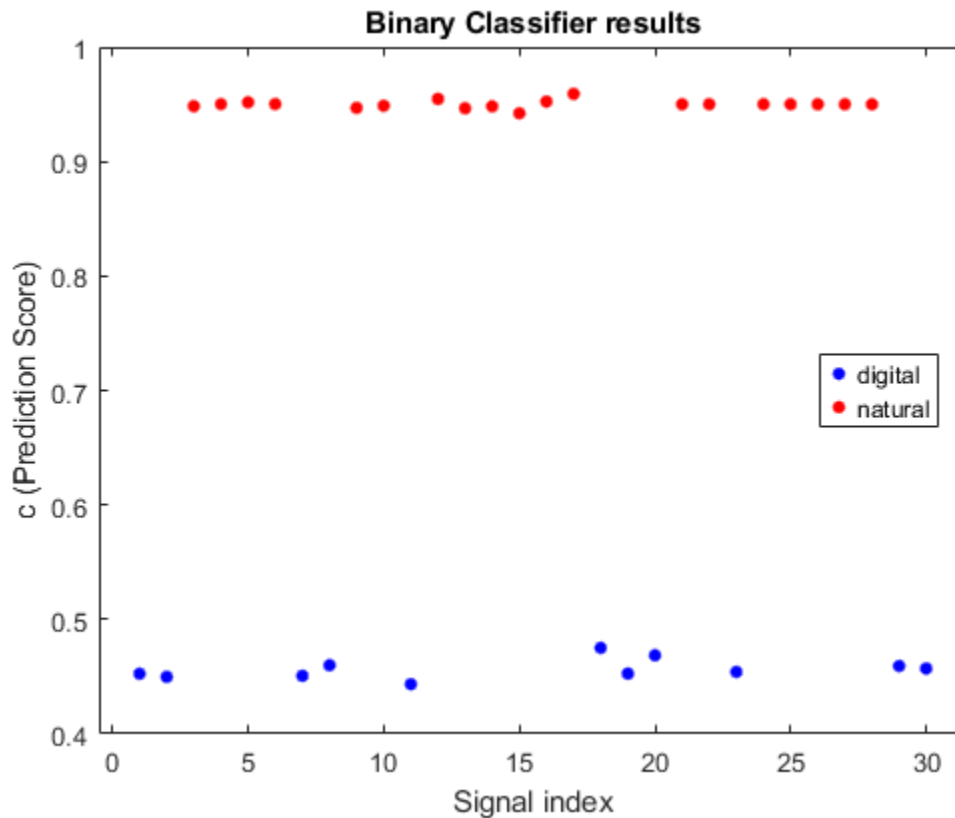


# Section (B) - Implementing BinaryClassifier

As we can see, the goal is to classify two types of signals: digital and natural. Each type of signal has distinct characteristics, some of which can be helpful in the classification process. One key difference between these signal

types is that digital signals contain a DC frequency component, whereas natural signals do not. The DC energy tends to the mean of the digital symbols which equals to 0.5. For the natural signals there is no DC because we choose the sine frequency and gaussian s.t. the mean over time equals to zero (like wavelets). Considering the constraints above, we offer the following classificator - if the signals DC energy is above 0.5 - it's of digital type otherwise it's natural. The mean of the signal can be approximated using Filter function with FIR filter which is defined as $ones(1, N)/N$ where N is the input signal length. The logic is opposite to the desired threshold defined in the question (c >= 0.5 -> natural otherwise digital), so if we look at $1 - DC_{energy}$ we get the desired logic.

**BinaryClassifier block diagram**
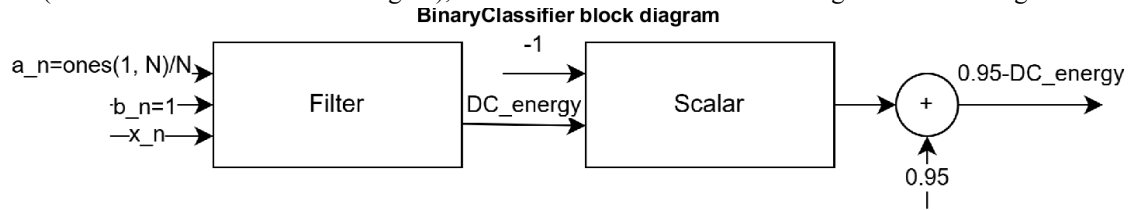


# Section (C)



*Published with MATLAB® R2024b*