

Decoder

April 9, 2017

This block accepts a complex electrical signal and outputs a sequence of binary values (0's and 1's). Each point of the input signal corresponds to a pair of bits.

Input Parameters

- `t_integer m{ 4 }`
- `vector<t_complex> iqAmplitudes{ { 1.0, 1.0 }, { -1.0, 1.0 }, { -1.0, -1.0 }, { 1.0, -1.0 } };`

Methods

`Decoder()`

`Decoder(vector<Signal *> &InputSig, vector<Signal *> &OutputSig) :Block(InputSig, OutputSig)`

`void initialize(void)`

`bool runBlock(void)`

`void setM(int mValue)`

`void getM()`

`void setIqAmplitudes(vector<t_iqValues> iqAmplitudesValues)`

`vector<t_iqValues>getIqAmplitudes()`

Functional description

This block makes the correspondence between a complex electrical signal and pair of binary values using a predetermined constellation.

To do so it computes the distance in the complex plane between each value of the input signal and each value of the *iqAmplitudes* vector selecting only the shortest one. It then converts the point in the IQ plane to a pair of bits making the correspondence between the input signal and a pair of bits.

Input Signals

Number: 1

Type: Electrical complex (TimeContinuousAmplitudeContinuousReal)

Output Signals

Number: 1

Type: Binary

Examples

As an example take an input signal with positive real and imaginary parts. It would correspond to the first point of the *iqAmplitudes* vector and therefore it would be associated to the pair of bits 00.

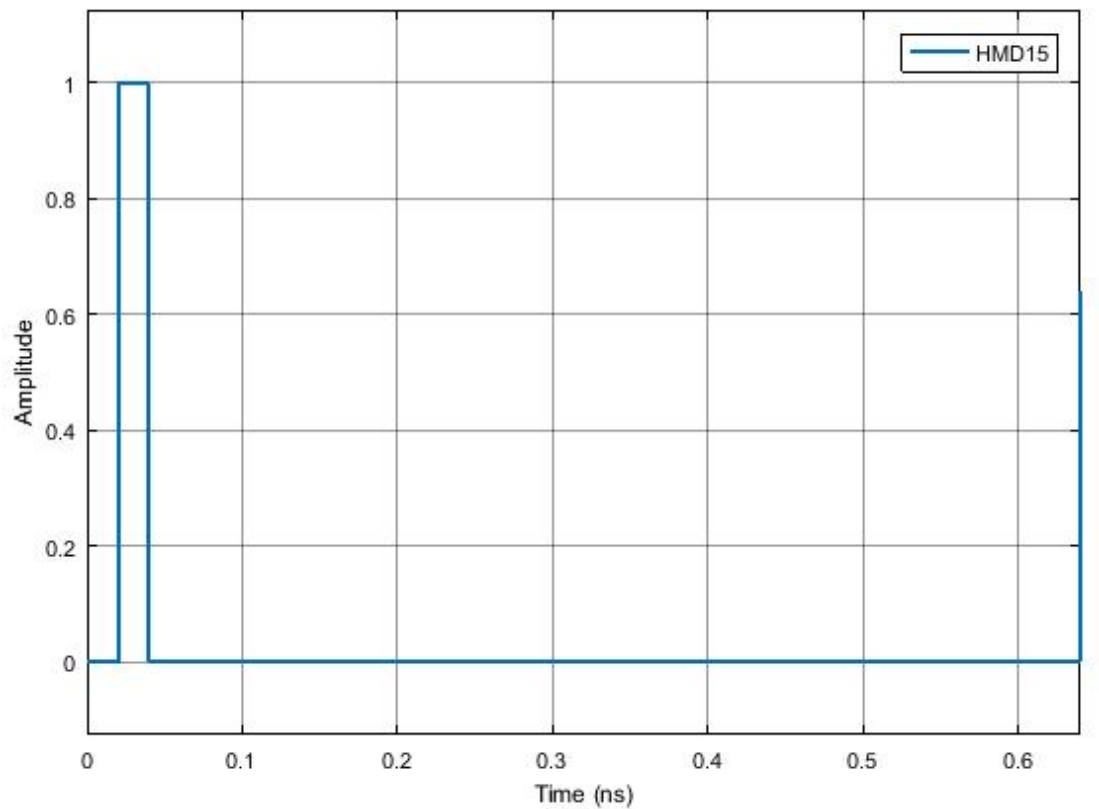


Figure 1: Example of the output signal of the decoder for a binary sequence 01. As expected it reproduces the initial bit stream

Suggestions for future improvement