Discrete to Continuous Time

October 13, 2016

This block converts a signal from a discrete time signal to a continuous time signal. It accepts one input signal of binary type and it produces one output signal that is a sequence of Dirac delta functions.

Input Parameters

- numberOfSamplesPerSymbol{8}
  (int)

Methods

DiscreteToContinuousTime(vector<Signal *> &inputSignals, vector<Signal *> &outputSignals) :Block(inputSignals, outputSignals){};

void initialize(void);

bool runBlock(void);

void setNumberOfSamplesPerSymbol(int nSamplesPerSymbol){ numberOfSamplesPerSymbol = nSamplesPerSymbol; };

int const getNumberOfSamplesPerSymbol(void){ return numberOfSamplesPerSymbol; };

Functional Description

This block reads the input signal buffer value, puts it in the output signal buffer and it fills the rest of the space available for that symbol with zeros. The space available in the buffer for each symbol is given by the parameter *numberOfSamplesPerSymbol*.

Input Signals

Number: 1
   Type: Sequence of 1's and -1's. (DiscreteTimeDiscreteAmplitude)

Output Signals

Number: 1
   Type: Sequence of Dirac delta functions (ContinuousTimeDiscreteAmplitude)

Example

Sugestions for future improvement