Pulse Shaper

October 13, 2016

This block applies a raised-cosine filter to the signal. It accepts one input signal of that is a sequence of Dirac delta functions and it produces one output signal continuous in time and in amplitude.

Input Parameters

- filterType{RaisedCosine}

- impulseResponseTimeLength{16}
  (int)
- rollOfFactor{0.9}
  (real $\in$ [0,1])

Methods

PulseShaper(vector<Signal *> &InputSig, vector<Signal *> OutputSig) :FIR_Filter(InputSig, OutputSig){};

void initialize(void);

void setImpulseResponseTimeLength(int impResponseTimeLength){ impulseResponseTimeLength = impResponseTimeLength; };

int const getImpulseResponseTimeLength(void) { return impulseResponseTimeLength; };

void setFilterType(PulseShaperFilter fType){ filterType = fType; };

PulseShaperFilter const getFilterType(void){ return filterType; };

void setRollOffFactor(double rOffFactor){ rollOffFactor = rOffFactor; };

double const getRollOffFactor(){ return rollOffFactor; };

Functional Description

The filter's transfer function is defined by the vector *impulseResponse*.

Input Signals

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

Output Signals

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

Example

Sugestions for future improvement

Introduce other types of filters.