GNURadio GRC Description for Git

25 Jul 2023

chrip_generator

Simple chirp generator made up of a signal source in sawtooth mode feeding into a VCO. Note that the VCO sensitivity is is units of rad/s/v, so keeping signal source at amplitude 1 and feeding it into a VCO with sensitivity 6.28M will result in a 1MHz chirp

lowpass_test

Used to test out the behaviour and functionality of the low pass filter block. Two signals of different frequencies are added together, then a filter is designed to suppress the higher frequency signal. Data is displayed in a GUI sink and saved into a file for plotting.

filesink_test

Outputs real components of a complex sine wave into a file to test analysis in numpy. Useful to test out different dtypes.

fmcw_sim_beat

Produces a chirp and introduces two independent delays, then multiplying of this delayed signal with a reference signal. Performing an FFT in Python should produce well defined peaks correlated to distance (Look up ‘frequency beat’ for more info). Note that passing the output through a low pass filter will greatly improve results.

fmcw_sim_complex

Same as above, but with complex streams and a multiply conjugate block. Unclear what this is supposed to do honestly.

fmcw_sim_matched_filter

Produces a chirp and introduces two independent delays. Passing the signals through a matched filter (FFT, Multiply conjugate, IFFT), then saving the results to be plotted in Python.

fmcw_matched_filter

Implementation of an FMCW radar using a matched filter. To be used with direct loopback testing (TX going into splitter, then into two RX channels. RX cables should be of differing

lengths). Chirp generator feeding into USRP sink, then USRP source (reference and delayed outputs) feeding into matched filter. Data gets saved into a file.

fmcw_beat

Implementation of an FMCW radar using a beat detector. To be used with direct loopback. Reference and delayed outputs are multiplied together

phase_sim

Computes the phase difference between two signal sources using two different methods: One finds the argument of each and finds their difference, the other computes a conjugate multiplication and finds its argument. Both should be equivalent in output.

phase_usrp

Designed to measure the phase offset between two inputs. A signal generator feeds into the USRP sink. The two outputs of the USRP source are then fed into a multiply conjugate block, and the argument of the output is fed into a GUI number sink.