DAT116 (Mixed-Signal System Design)
MATLAB functions used in lab series
Version 6.1

Lars Svensson
lars.svensson@chalmers.se

November 1, 2019

Signal viewing and post-processing

`sigview(args...)` displays a signal in a new MATLAB plot window. Several signals may be listed in the argument list of the function, and each of these are then plotted in a different color. Two special arguments are recognized anywhere in the argument list: `'hold'` uses the active plot window for the plot and adds the new waveform to the existing plot, if any. The second special argument, `'nolegend'`, suppresses the creation of a legend for the plot.

`sigspectrum(args...)` works like `sigview`, but plots a power spectrum estimate instead. In addition to the special arguments available for `sigview`, `'linf'` causes the frequency scale to be linear (default is logarithmic); a single integer specifies the number of signal points to use for the FFT (must be a power of two); and a single character from the set used as markers in MATLAB's plot function (`'.'`, `'+'`, `'x'`, `'o'`, `'*'`, `'s'`, `'d'`, `'^'`, `'v'`, `'>'`, `'<'`, `'p'`, `'h'`) sets the marker for the plot (the default is `'.'`).

The statement `bar = sigspectrum(args...)` assigns to `bar` a vector of the powers of each spectral component. (If several signal spectra are being plotted, a matrix will be returned where the column vectors contain the spectral components of each of the signals in the order listed in the plot legend). The special argument `'noplot'` can be used to turn off plot generation if only the spectral values are of interest.

In case the number of signal points available is not a power of two and the point count is not given explicitly in the argument list, the largest possible

power-of-two vector is selected from the *trailing* end of the signal.

`sigharms(specarray, fullcycles, paramarray)` picks out and plots harmonic power values as functions of some swept parameter value. It takes three parameters: `specarray` is an array of spectrum values such as generated by `sigspectrum` for several signals; `fullcycles` is the number of full cycles of the signal included in the spectrum calculation; and `paramarray` is the array of values for the swept parameter.

`issignal(arg)` checks whether the argument conforms to the signal conventions (the **Structure With Time** option must be used when saving the data). This function is used by the other functions and is not very useful on its own.

Lookup tables

`lutix(arg)` takes one integer argument and returns a vector suitable as the `Vector of input values` parameter to the `Lookup Table` block. The argument specfies the number of quantization levels.

`lutdata(arg)` works like `lutix`, but is intended for the `Table data` parameter instead.

`r2r(n, sigma)` generates a vector of output levels for an `n`-bit D/A converter which uses an R2R ladder to determine the values. A non-zero value for `sigma` sets the standard deviation for the relative errors of the resistance values. Keep the `sigma` values below 0.1 to avoid unphysical negative resistance values.

Cadence interface

`cds2sig(cdswave, interval, points)` converts a signal waveform, generated by simulation in Cadence Spectre and imported into MATLAB with `cds_srr`, into the format used by `sigspectrum` etc. `cdswave` is the original waveform, `interval` is the time interval to be converted, and `points` is the number of equidistant points to be produced in the signal array. Since the original waveform is typically not uniformly sampled, the conversion involves interpolation to find new values.