This README file contains information about the folder rd provided as part of the CTSS Sampling Toolbox written by Michael A Lexa, Mike E Davies, and John S Thompson, University of Edinburgh.

The folder contains 4 MATLAB m-files that simulate the sampling and recovery of a continuous-time spectrally-sparse multitone signal using the Random Demodulator:
rd_demo.m
tonesparse.m
rd_sampling.m
rd_recovery.m

The folder also contains the software package Sparsify 0.4 written by Thomas Blumensath (see http://users.fmrib.ox.ac.uk/%7Etblumens/sparsify/sparsify.html).

Technical information about the Random Demodulator is found in the report "Sampling Sparse Multitone Signals with a Random Demodulator" that is contained in the CTSS Sampling Toolbox Documentation folder.

These scripts are a research tool and by no means represent a finished software product.

To run the simulation execute rd_demo.m from within MATLAB, that is type rd_demo at the MATLAB command prompt. This script, in turn, calls tonesparse.m to generate a (discrete) sparse multitone signal, calls rd_sampling.m to sample it, and then calls rd_recovery.m to recover the original signal from the randomly demodulated samples.

The multitone signal that tonesparse.m generates can be thought of as being the Nyquist samples of at continuous-time multitone signal, i.e. as a signal that results from sampling a continuous-time multitone signal at the Nyquist rate. Hence, the simulation actually subsamples a discrete signal and recovers the Nyquist samples from the randomly demodulated samples.

There are several parameters that characterise the simulated multitone signal and the random demodulator. These include the bound on the largest possible tone (W), the number of tones actually present in the signal (K), and the period of the multitone signal (Tx). These parameters are set in rd_demo.m and tonespare.m. The parameter M specifies the sampling rate: M samples are collected in Tx seconds. The values of these parameters are assigned in rd_demo.m.

The script rd_sampling.m accepts as input the simulated multitone signal and outputs randomly demodulated samples.

The script rd_recovery.m is the algorithm that recovers the Nyquist samples. The algorithm is that of Tropp et al. (see the references listed at http://wwww.see.ed.ac.uk/~mlexa/CTSS.html)

The simulation produces plots comparing the time domain and frequency domain outputs of the original and recovered signals and graphically compares the original and recovered parameters characterising the signals.