Documentation for Python code for the paper: "Compressive Sensing for Spread Spectrum Receivers"

Karsten Fyhn - kfn@es.aau.dk

June 28, 2012

This is the documentation for the Python code for the paper: "Compressive Sensing for Spread Spectrum Receivers". The code relies on the PyComm module, which must be included in your Python path. Both the paper and code are available from www.sparsesampling.com/css.

The code in this package is separated into four folders:

- Maleki_plot - The code to reproduce a plot from the paper "Optimally tuned iterative reconstruction algorithms for compressed sensing" by Maleki and Donoho
- donoho_tanner_phase_transition - The code to reproduce Donoho-Tanner phase transition diagrams, the Subspace Pursuit iteration count plot and the computational cost plots
- discreteAnalysis - The code to reproduce the simple, discrete numerical experiments with BER versus SNR
- rfAnalysis - The code to reproduce the more complex, RF numerical experiments with BER versus EbN0 and with/without quantization

Each folder contains four important files:

- experiments.py - A range of ready-made simulation setups that will generate the plots from our paper
- exp_funcs.py - The implementations of our receiver structures and test setups
- main.py - The main test function that calls our receiver structure setups and manages the simulation results
- main_DISPLAY_RESULTS.py - Displays the results of the simulations

Our simulation framework is designed for use on supercomputers using the Message Passing Interface (MPI)[1]. To start a simulation, run the following command from a terminal:

```
mpirun python experiments.py EXPERIMENT
```

where EXPERIMENT is one of the function names in experiment.py. The result is saved to a Pickled file, which may be loaded using the main_DISPLAY_RESULTS.py file to display the results.

In the folders discreteAnalysis and rfAnalysis, there is also a Maple script, which we use to generate the theoretical curve for non-coherent MFSK. This needed to be a Maple script to obtain the high number of digits needed to generate the correct result.

I hope you find this software usable. If you find any bugs or have comments, you are welcome to contact me through e-mail.

[1] See http://en.wikipedia.org/wiki/Message_Passing_Interface and http://mpi4py.scipy.org/.