CCE V1.0b

A MATLAB® Toolbox for Simulating Compressive MIMO Channel Estimation and/or Tracking

© Daniel Eiwen, NuHAG, Faculty of Mathematics, University of Vienna
Georg Tauböck, Institute of Telecommunications, Vienna University of Technology
2008-2015
E-mail: daniel.eiwen@univie.ac.at

July, 2015

Installation

Download *cce_v1.0b.zip* and extract all the files. The password for unzipping is freely available via email from daniel.eiwen@univie.ac.at, if personal and institutional information is provided.

To run CCE V1.0b as it is, the following software package has to be installed:

- cvx http://cvxr.com/
- spgl http://www.cs.ubc.ca/~mpf/spgl1/

Instead of *spgl*, any other software package including a solver for the Group Basis Pursuit Denoising (G-BPDN) problem can be used. Furthermore, channels can be simulated using the toolbox

- IlmProp http://www2.tu-ilmenau.de/nt/en/ilmprop/

Of course, channels can be calculated using any other software package as well, but then some changes have to be made in the code.

Quick Start Guide

The files of the toolbox *spgl* (or an alternative G-BPDN-solver) have to be stored in the folder *cce_v1.0b*. If you want to use *IlmProp* for channel simulation, the IlmProp-files have to be stored in the folder *cce_v1.0b\IlmProp*, and the files *IPG_CCE_G.m*, *IPG_CCE_Tx.m*, *IPG_CCE_Rx.m* and *IPG_CCE_Env.m* have to be copied into the folder *cce_v1.0b\IlmProp\geometry_scripts*.

The main file is *CompChaEst.m*, the first few lines of which are dedicated to setting the parameters for the compressed sensing CS methods. These methods are divided into the ones used channel-per-channel, the multichannel methods, and the ones used for channel tracking. All the methods can take group sparsity into account. If you want to perform channl tracking you have to set the variable *Track* to one. The MIMO OFDM parameters are also defined in this file (for simulating SISO systems just set *NR* and *NT* to one), whereas the geometrical setting of transmitter, receiver and scatterers is defined in the files *IPG_CCE_G.m*, *IPG_CCE_Tx.m*, *IPG_CCE_Rx.m* as well as *IPG_CCE_Env.m*.

Executing *CompChaEst.m* first calculates optimized bases for the channel-per-channel as well as the mutlichannel estimation methods (which can already take up to a few hours, depending on your machine), then generates an $NR \times NT$ MIMO channel using IlmProp (which can take more than one hour, again), and finally performs compressive channel estimation and/or tracking using the algorithms specified in

the beginning of the file for both the DFT-basis and the optimized basis/bases. The results are stored in the file *results_CompChaEst.mat* and can be displayed using the file *display_results.m*.

Notes

To speed up the program, precalculate the optimized bases and the channels and store them. The code in *CompChaEst.m* can be adapted to this case easily (the relevant lines are marked by comments).

CCE V1.0b aims to serve as a proof of concept that compressive channel estimation can produce better channel estimates than conventional least-squares channel estimation. We use conventional compressed sensing, group-sparse CS and multichannel CS methods for compressive channel estimation, and modified CS methods for compressive channel tracking. The performance of all these CS methods strongly depends on some parameters, and there is no estimate for a good choice of these parameters. Therefore, bad results are likely to be due to mismatched parameter settings. Furthermore, for some simulation scenarios, the number of iterations required to average over a sufficiently large number of transmitted symbols and channel realizations can be quite large.
The code is not optimized in any way, and many improvements, most importantly in running time, are definitely possible and desirable. Although the code documentation is poor, support will be very limited in general. There is no guarantee that questions via email will be answered, although it is probable.

For algorithmic details, see [1]. The code implemented in CompChaEst.m and display_results.m was essentially used to generate the MSE plots in [1].

Academic use of the toolbox is permitted, commercial use prohibited, cf. License_Agreement_CCE.pdf. For commercial use, please contact daniel.eiwen@univie.ac.at.

Finally note that the code was developed in MATLAB® 7.8.0.

[1] D. Eiwen, "*Compressive Channel Estimation - Compressed Sensing Methods for Estimating Doubly Selective Channels in Multicarrier Systems*," PhD thesis, University of Vienna, May 2012.