CORRIDOR measurement campaign

This document describes some background information on the tools used for the post-processing of the CORRIDOR measurement data. All the code that is referenced in this document can be found in the OpenAirInterface SVN repository in openair4G/trunk/targets/PROJECTS/CORRIDOR.

# Design of the sounding signal

The sounding signal will occupy 3 channels

* 5MHz at 771.5 MHz (channel 1)
* 10MHz at 2.605GHz (channel 2a)
* 20MHz at 2.590GHz (channel 2b)

The sounding signal is an OFDM signal with specifications taken from the LTE standard. The following table summarizes the main parameters.

|  |  |
| --- | --- |
| Frame length | 10 ms |
| Symbol duration | 66 μs |
| Prefix length | 16 μs |
| OFDM size | 512/1024/2048 |
| Useful carriers | 300/600/1200 |

The first symbol of each frame contains the LTE primary synchronization sequence (PSS) and the rest of the signal is filled with OFDM modulated random QPSK symbols. In order to minimize inter carrier interference (ICI) in high mobility scenarios, we only use ever second subcarrier. In case multiple transmit antennas are used, we use the following pattern:

Figure 1: Pilot patterns for 2 (left) and 4 (right) TX antennas

The sounding signal can be generated with the script generation\_ca.m, which generates the file ofdm\_pilots\_sync\_30MHz.mat. A pre-computed version of this file is also on the SVN server, which is the one used for the measurement campaign (since the generation of the signals uses a random number generator, the signals might be different on different machines).

## Notes on signal design

*If we assume a maximum speed of v=300km/h and the two carrier frequencies f1=800MHz and f2=2.6GHz, then we get Doppler shifts of fd1=222Hz and fd2=723Hz. Using the OFDM symbol time of LTE Ts=66.7 μs (subcarrier spacing of 15kHz), we can compute the ICI as [1, eqn (5.15)]  
  
P\_ICI = 1-sinc^2(fd Ts)   
  
and get P1=-31dB and P2=-21dB. In my opinion this is already quite low and for most of the cases (unless we are very close to the base station) even below the measurement SNR. To be on the safe side we could use every 2nd spacing (subcarrier spacing 30kHz) for the higher frequency, which would give P2=-27dB, but I would not go much lower than that.*

# Data collection

We save the raw IQ data of all antennas in real-time. The data of the 5MHz channel at 771.5 MHz and the 30MHz channel at 2.6 GHz are stored independently. In order to get good resolution for the Doppler profile estimation (see below), we should at least store a continuous chunk of 1 sec. Due to the enormous amount of data and limited write speed of the data (even with a RAID 0 system), for the 30MHz channel at 2.6GHz we only save 1 second out of 2. For the 5MHz channel at 771.5MHz, we can save all the data continuously.

# Post processing

There are two scripts for post processing, one for channel 1 (emos\_read.m) and one for channel 2 (emos\_read\_ca.m). The filename and the number of antennas has the be adjusted. The different algorithms used in the script are explained in more detail below.

## Synchronization

Synchronization it is the most important part of the Post processing. In OFDM systems, there exist three different problems related to synchronization: The first one is the time offset (or symbol synchronization), the solve of this problem allows the receiver to determine the start point in the received OFDM symbol. The second one is the frequency offset (or frequency synchronization), which tries to eliminate the carrier frequency offset caused by the mismatch from the radio frequency local oscillators and the Doppler shift.

If we ignore the channel dispersion, isolating these two offset problems. The model for the received signal becomes [van de Beek reference]

where is the unknown valued delay and is the unknown carrier frequency offset, is the AWGN and is the number of samples.

Finally, the last issue is the sampling clock synchronization, which manages to synchronize the sampling frequency between transmitter and receiver, because both of them work with different physical clocks.

All of the above problems we will solve it with use of known symbols (synchronization sequence) by the transmitter. At the literature there are a lot of works which they don’t use synchronization sequence for achieve higher throughput but they use the knowledge, that there is cycle prefix in the symbol. This way is less accurate and more complex.

### Time offset

To define the start of the frame we make cross correlation between our received data and the (known) synchronization sequence (pilots) which is in the beginning of every frame.

We suppose that we have line of sight, or generally, we don’t have any weird geometry (at not- line of sight case) which is not give us maximum of the cross correlation at the beginning of every frame.

Where is the received samples and is the synchronization sequence, “” is the convolution operator and is our estimate for the initial sample.

An example result of the cross correlation is seams in Figure 2 (All of the plots below are for the simple case which our channel between and is just a cable).



Figure 2: Cross Correlation

So at this point we have to choose as the beginning of the frame the sample which maximize the cross correlation. The problem is that we load a large amount of frames so we don’t want only one max but the max of every frame. We put a threshold at half of the max (global max) but unfortunately they do not “survive” only the samples at the beginning of each frame but and some samples around them. So we have to grouped them and find the max of each group, Figure 3.



Figure 3: Closer look at Cross Correlation

We create groups of successive samples, and we pick the max of each group as the beginning of each frame (on the case of large plateau which leads to some uncertainty as to the start of the frame we should use more complex methods from just taking the max [3]). We have two possible sources of error, the first is to have some cross correlated values over the threshold at the “body” of frame, or to have dip under the threshold of one sample around the maximum so the next will be considered as second group. At the simple channel which we tested we don’t have any of these problems. We discussed more sophisticated solutions, with using the knowledge of the expected distance between two maximum (frame length) or taking advantage from the knowledge of the number of frames. We will implement these solutions if the channel requires.

### Frequency offset

At this subsection the goal is to eliminate the frequency offset which offset caused by the mismatch from the radio frequency local oscillators and the Doppler shift. The most important effect of a frequency offset between transmitter and receiver is a loss of orthogonality between the subcarriers resulting in ICI. The characteristics of this ICI are similar to white Gaussian noise and lead to a degradation of the SNR. For both AWGN and fading channels, this degradation increases with the square of the number of subcarriers.

Taking as granted the beginning of the frame , from the time offset solution we calculate the frequency offset by the angle between the OFDM symbols. From [4] we take

is the number of consecutive symbols. So the offset is estimated by

### Sampling clock synchronization

The different physical clocks of the transmitter and receiver they cause a drift in time. The reason is simple, if the duration of one sample in the oscillator of transmitter is and the duration at the oscillator of the receiver is . Even if we have perfect estimation for the beginning time, If the after samples the receiver it has drift one, Figure 3.



Figure 4: time drift

To avoid that problem we can make least squares fit at the beginning of every frame to find the for each sample, after that we can make Lanczos resampling [5] at the correct time (transmitter and receiver must have the same time constant).

At Figure 4 we see that we have fix the problem of time drift, but this procedure makes a lot of time

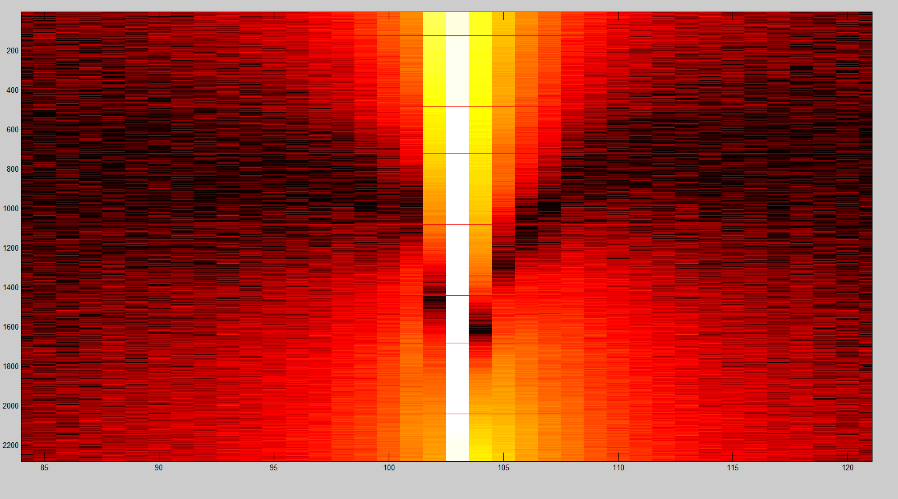


Figure 5: Lanczos resampling

## Power delay profile estimation

## Doppler profile estimation

# References

1. Sesia, Toufik, Baker (eds.), "LTE - The UMTS Long Term Evolution", Wiley 2011 (2nd ed.)
2. Paier, A., Karedal, J., Czink, N., Hofstetter, H., Dumard, C., Zemen, T., ... & Mecklenbrauker, C. F. (2007, October). Car-to-car radio channel measurements at 5 GHz: Pathloss, power-delay profile, and delay-Doppler spectrum. In *Wireless Communication Systems, 2007. ISWCS 2007. 4th International Symposium on* (pp. 224-228). IEEE.
3. Minn, Hlaing, Mao Zeng, and Vijay K. Bhargava. “On Timing Offset Estimation for OFDM Systems.” *Communications Letters, IEEE* 4, no. 7 (2000): 242–44.
4. J.J. van de Beek, M. Sandell, and P.O. BÄorjesson, `ML estimation of time and frequency offsets in OFDM systems', *IEEE Transactions on Signal Processing*, vol. 45, no. 7, pp. 1800{1805, July 1997.
5. http://en.wikipedia.org/wiki/Lanczos\_resampling