Open-Source Implementation of a Digital Radio Mondiale (DRM) Receiver

Dipl.-Ing. Alexander Kurpiers
Dipl.-Ing. Volker Fischer

Darmstadt University of Technology
Institute of Communication Technology

Survey

- Introduction & Motivation
- Channel Estimation
  - 2D Interpolation
  - 1D Wiener Interpolation in Time- and Frequency Direction
- Synchronization
  - Acquisition
    - Frequency: FFT-based Algorithm
    - Time and Robustness Mode Detection: Guard-Interval Correlation
    - DRM Frame Synchronization
  - Tracking
    - Frequency and Sample Rate: Using Phase Increment of Pilots
    - Time: Using Channel Estimation
- Performance Simulation
- Conclusion and Outlook

Motivation

Digital Radio Mondiale (DRM) is a new OFDM-based digital radio standard for the long-, medium- and short-wave ranges

- DRM has a small bandwidth of less than 20 kHz - easy to handle with current PC sound cards
- Real-time software implementation possible
- No publicly available open-source DRM receiver

We looked for a test bed for OFDM algorithm development (channel estimation, ICI compensation, synchronisation)

Aim: complete DRM Receiver under GPL

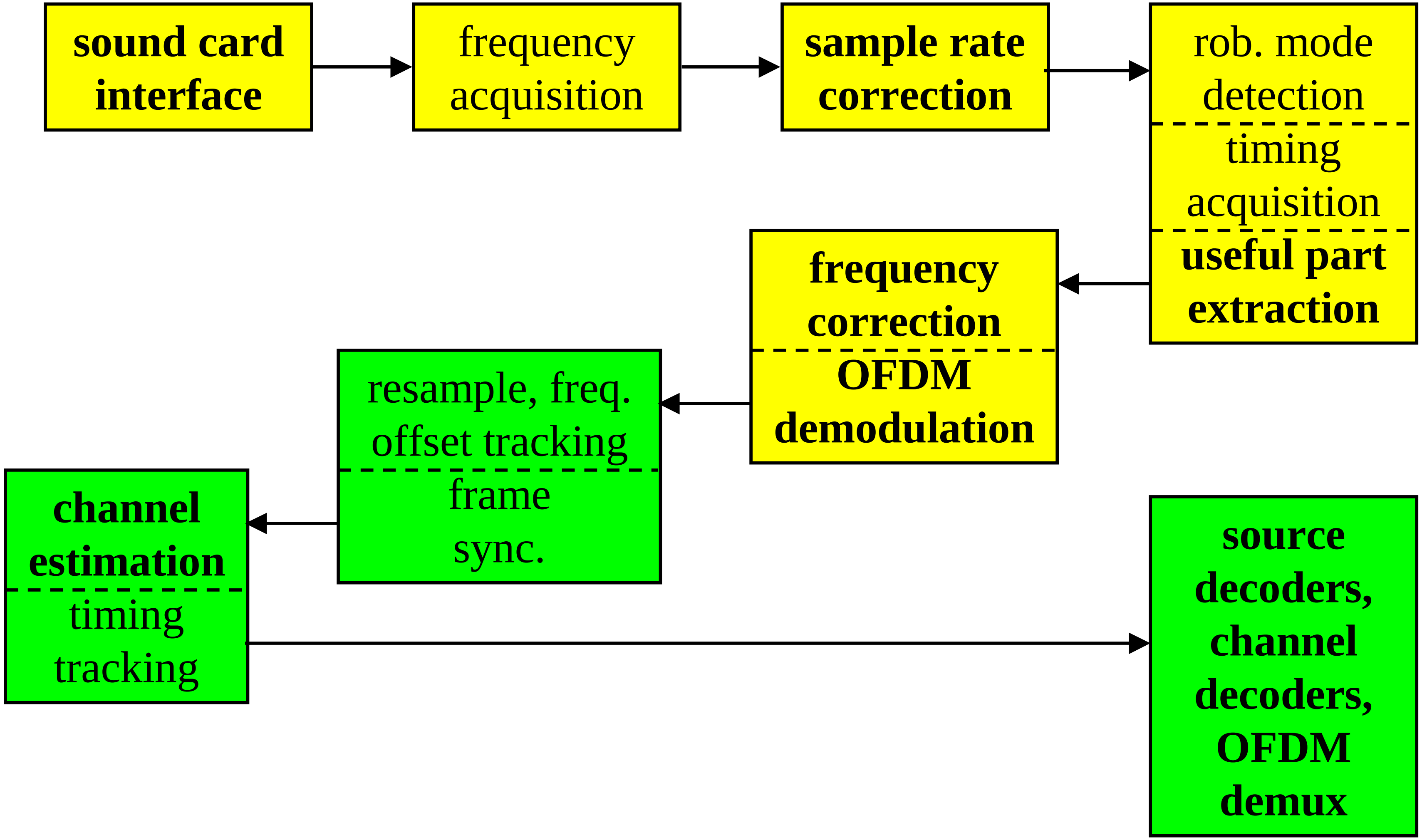
Project started in summer 2001

Introduction

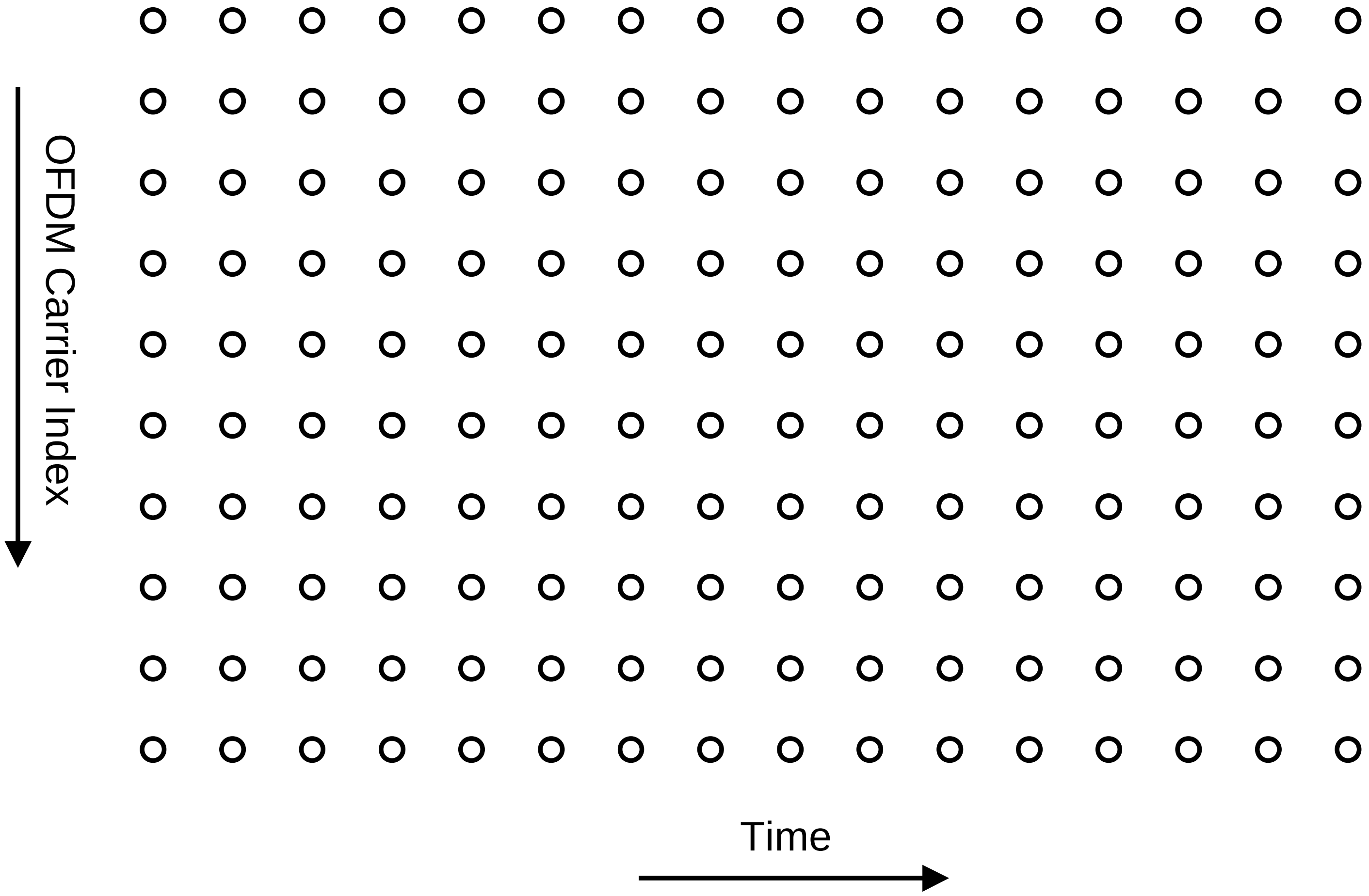
OFDM

- Vulnerable to frequency offsets (causes ICI)
- Timing critical if delay spread is in the range of the guard-interval
- Only one coefficient per carrier and symbol has to be estimated for equalisation

Software Overview

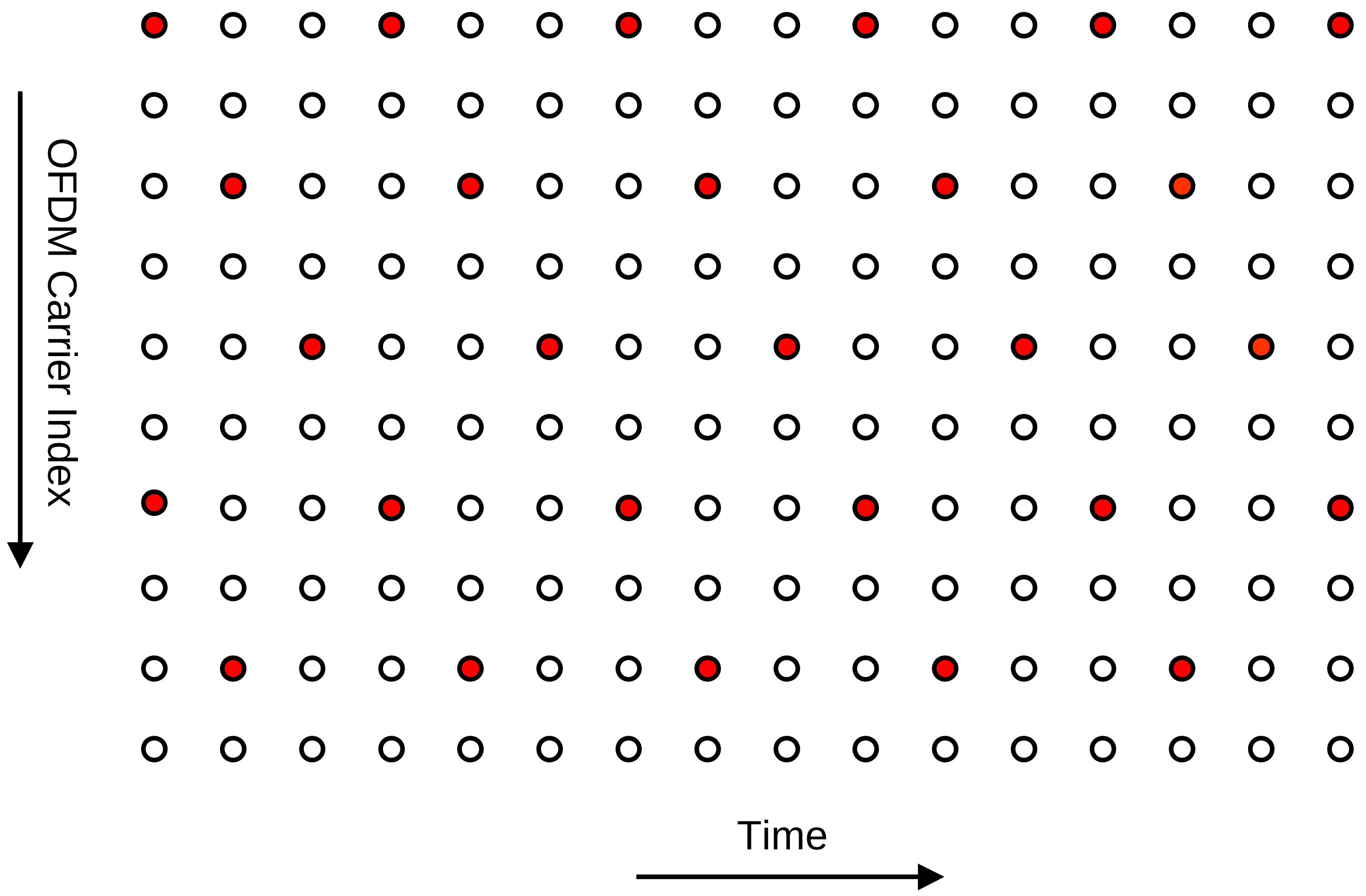


Channel Estimation



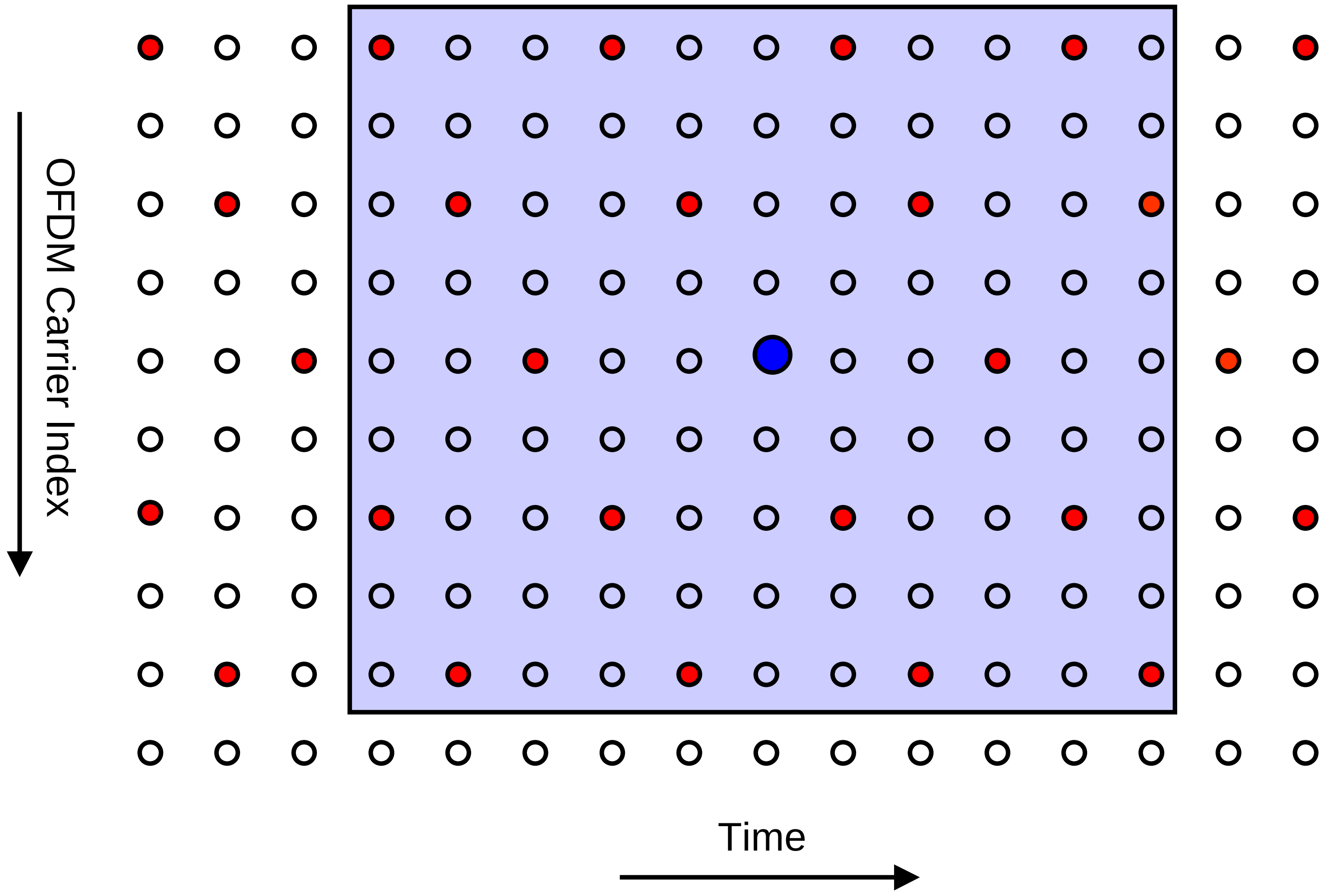
On every OFDM carrier the channel transfer function $H_{k,l}$ has to be estimated

Channel Estimation



Scattered pilots allow for easy channel estimation

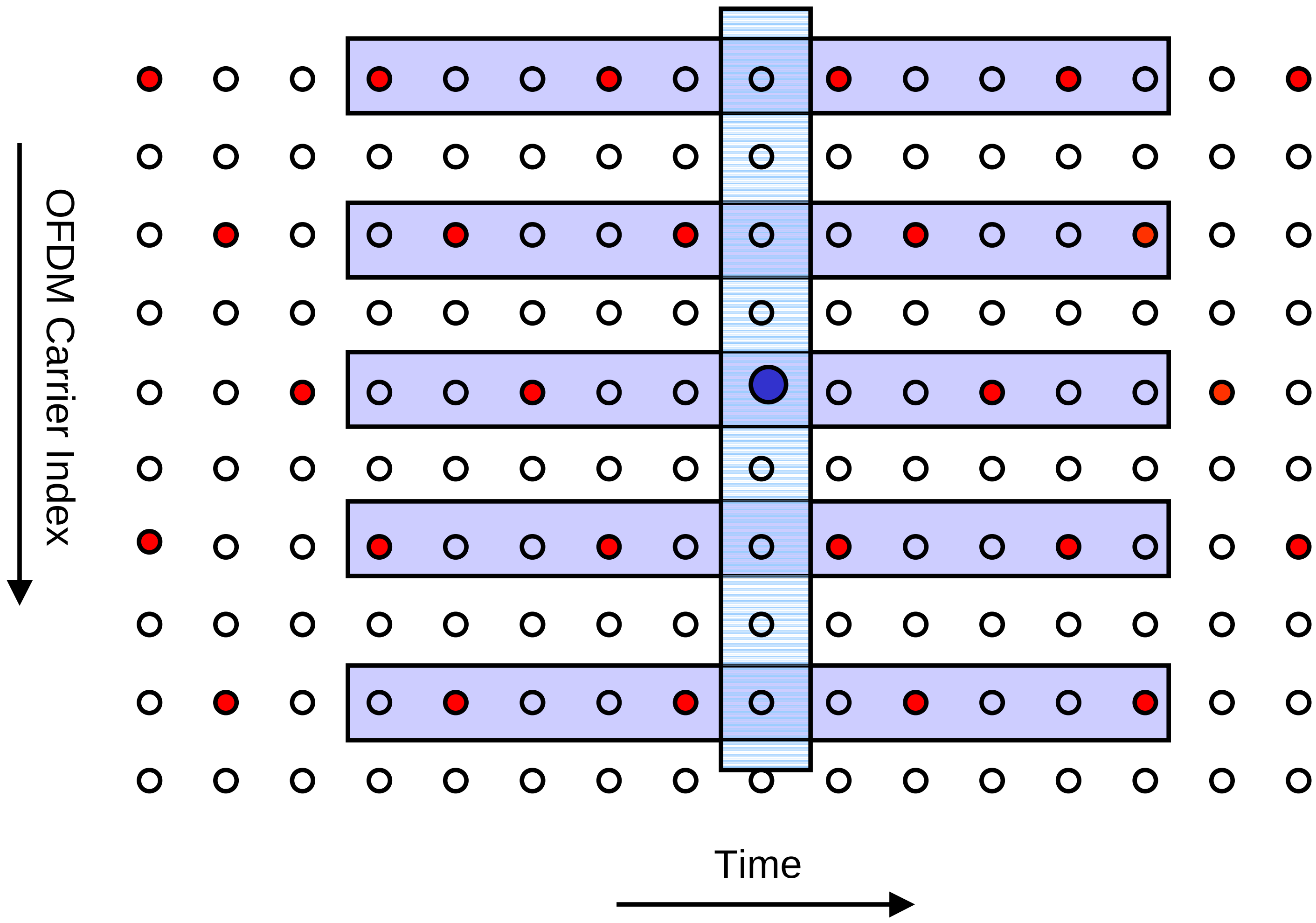
Channel Estimation



two dimensional Wiener Interpolator is optimal

but too complex

Channel Estimation



Wiener Interpolator can be separated into two 1D filters

Channel Estimation: Wiener Interpolation

- **MMSE solution:** $\hat{\mathbf{h}} = \mathbf{R}_{h\hat{p}} \mathbf{R}_{\hat{p}\hat{p}}^{-1} \hat{\mathbf{p}}$ $\qquad \mathbf{R}_{\hat{p}\hat{p}} = \mathbf{R}_{pp} + \frac{1}{\mathrm{SNR}} \mathbf{I}$

  $\mathbf{R}_{h\hat{p}}$: Cross-covariance matrix between $\mathbf{h}$ and the noisy pilot estimates $\hat{\mathbf{p}}$

  $\mathbf{R}_{\hat{p}\hat{p}}$: Auto-covariance matrix of the pilot estimates

- **Doppler profile of a typical shortwave channel:** $|H(f)| = \frac{1}{\sqrt{2\pi\sigma_{\mathrm{d}}^2}} e^{-\frac{f^2}{2\sigma_{\mathrm{d}}^2}}$

  Resulting correlation function: $r_{f_d}(\Delta k) = e^{-2(\sigma_{\mathrm{d}}\, \pi\, NT\Delta k)^2}$

- **Assuming uniform delay power spectrum with the length of the guard-interval:**

$$r_\tau(\Delta l) = \mathrm{sinc}\left(\Delta l \frac{N_{\mathrm{G}}}{N}\right)$$

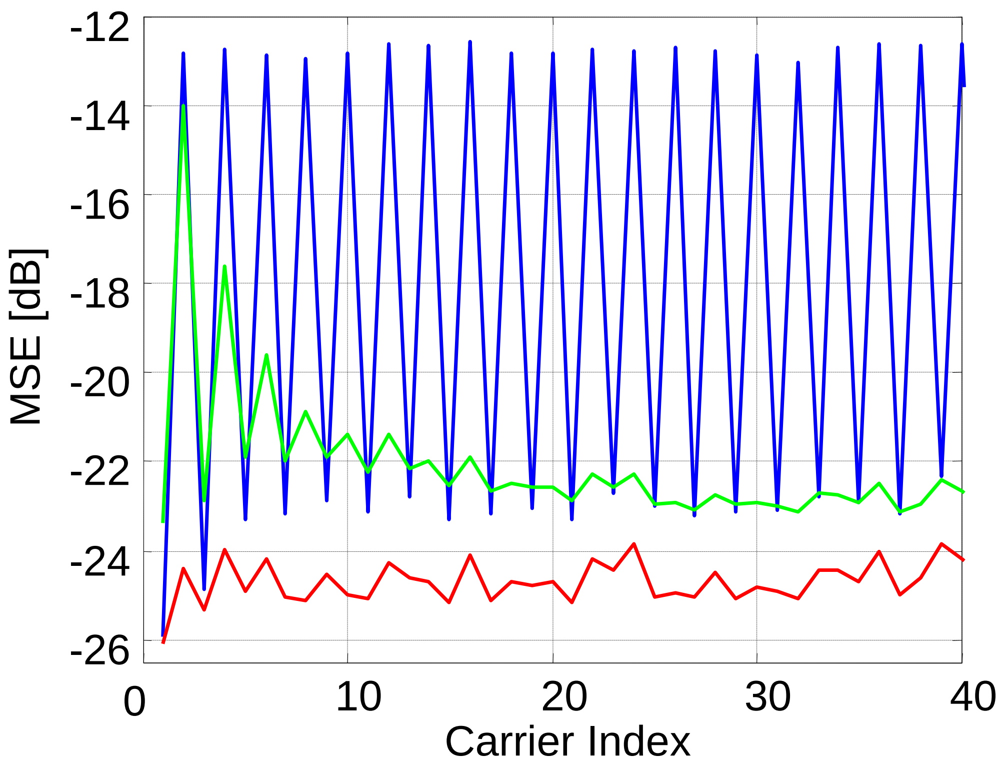
N_{G} : Length of guard-interval

N : Length of useful part

Channel Estimation in Frequency Direction

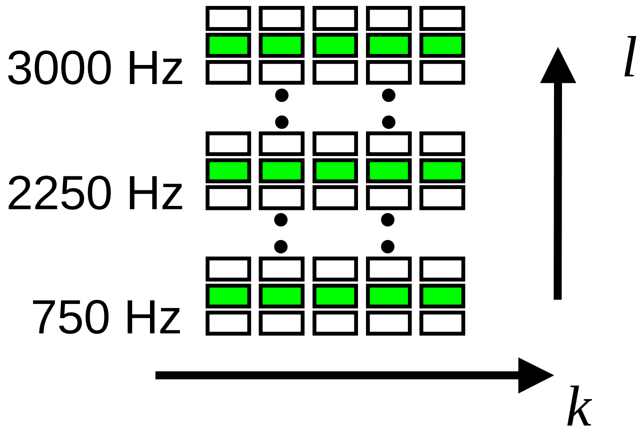
The following parameters were used in this simulation:
Robustness mode B, 10 kHz bandwidth, 20 dB SNR, channel No. 3 (US Consortium)
The mean squared error (MSE) between the estimated channel and an ideal channel estimation is plotted.

- **Blue line**: Linear interpolation
- **Green line**: Windowed DFT algorithm
- **Red line**: Wiener interpolation (using all pilot carriers for each interpolated cell



Frequency Acquisition (I)

- Exploiting the power difference of the three frequency pilot cells and data cells
  - **Pilot cells:** boosted, continuous tones
  - **Data cells:** power spread due to modulation



- FFT- based algorithm
  - Squared norm of FFT calculated over more than one symbol (Estimation of PSD)
  - Correlation with known frequency pilot positions
- **Effects of a large FFT window:**
  - Statistical properties of data cells more distinct, peak detection improved
  - BUT fading channel effects reduce performance

Frequency Acquisition (II)

- Estimation of PSD

$$R_{m,l} = \left| \sum_{n=0}^{N_{\text{ac}}-1} r_{n+l} \; e^{-j\frac{2\pi}{N_{\text{ac}}}nm} \right|^2$$

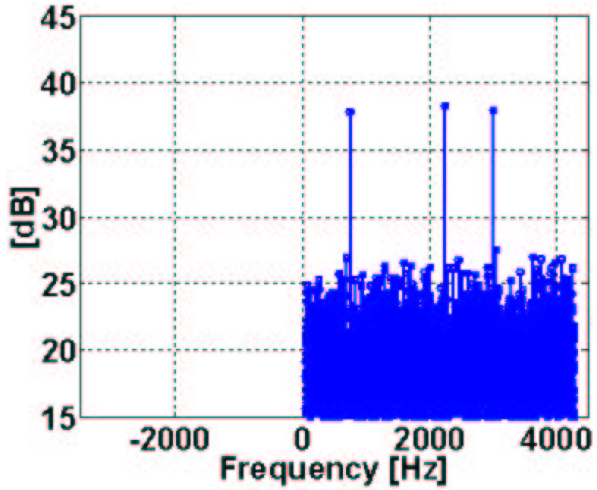
- Correlation with pilot positions

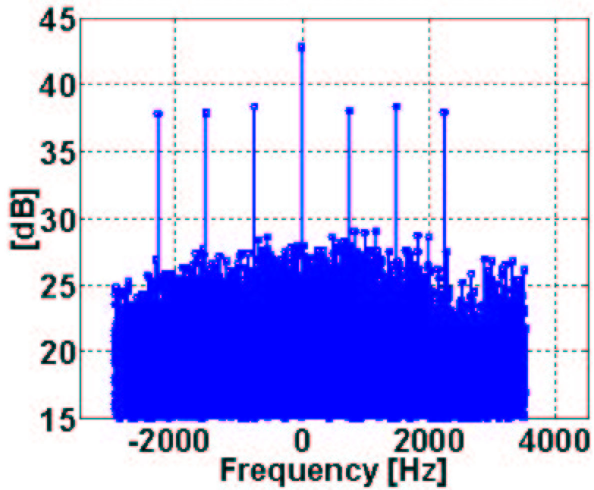
$$\hat{f}_{\text{acq}} = \frac{f_s}{N_{\text{ac}}} \max_m \left\{ \sum_{i=0}^{2} R_{m+p_{f_{\text{ac}}}(i),l} \right\}$$

- Placement of FFT window arbitrary

  → No prior timing information needed

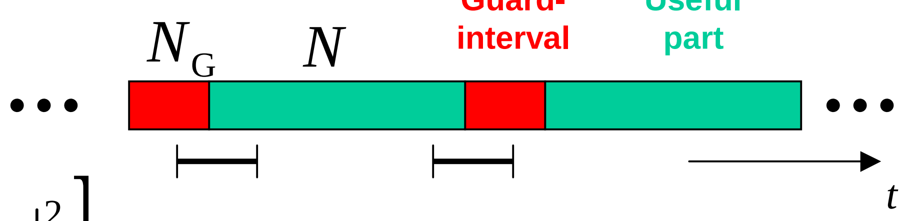
- Average error rate < 10% for all channels and robustness modes





Time Acquisition (I)

- Guard-interval correlation



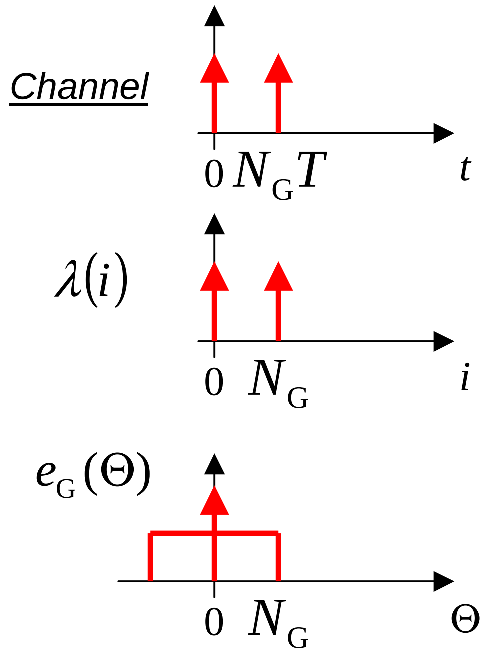
$$\lambda(i) = \left| \sum_{n=i}^{i+Ng-1} r_n r_{n+N}^{*} \right| - \sum_{n=i}^{i+Ng-1} \left[|r_n|^2 + |r_{n+N}|^2 \right]$$

- Using energy in guard-interval
  - For multipath fading channel

$$e_G(\Theta) = \sum_{m=\Theta}^{\Theta+N_G-1} \lambda(m)$$

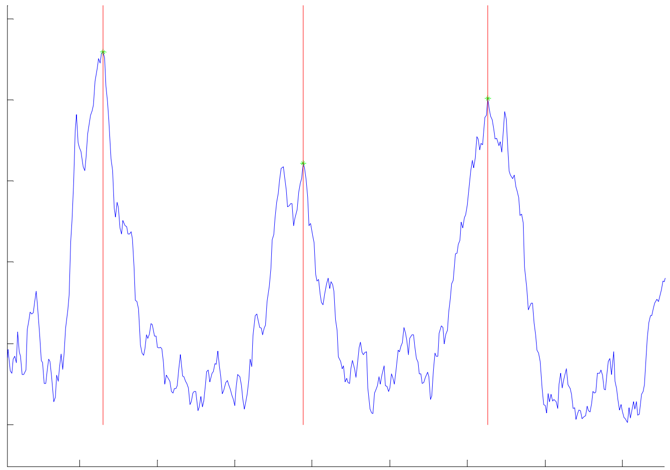
- $\arg\max\{e_G(\Theta)\}$ is the resulting estimated timing position

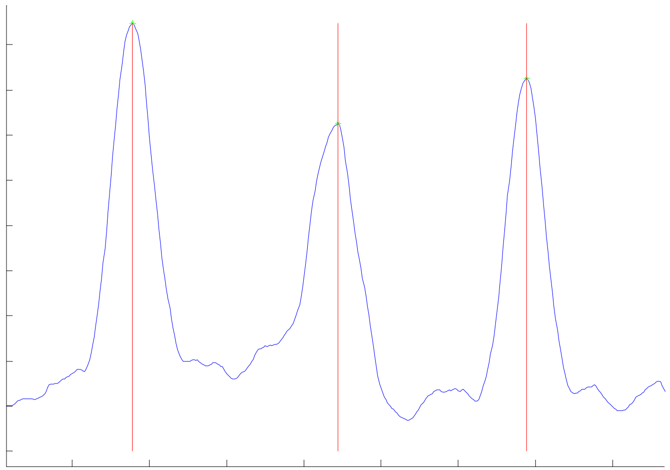
Example (idealised):



Time Acquisition (II)

Influence of Guard Energy Consideration on a two path fading channel:

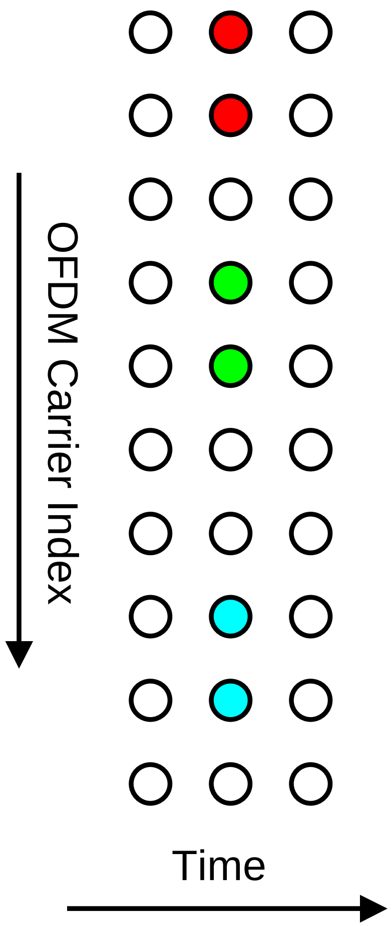

Only correlation


Correlation with guard energy consideration

Robustness modes can be detected by using time difference between peaks (period equals useful part duration NT)

Frame Synchronization

Time pilot pairs



Assumption: channel is identical at adjacent pilot positions:

$$H_{k,p_t(i)} \approx H_{k,p_t(i)+1}$$

With this „channel estimate“ we can calculate the squared distance between received and pilot cells:

$$\gamma(k) = \sum_{i=0}^{L_T-1} \left| z_{k,p_t(i)} \frac{c_{k,p_t(i)+1}}{c_{k,p_t(i)}} e^{-j\frac{2\pi}{N}\frac{N_G}{2}} - z_{k,p_t(i)+1} \right|^2$$

This yields a minimum at the beginning of the frame

Frequency Tracking

- **Frequency offset estimation based on phase increment between two successive symbols at the frequency pilot carriers**
  - Frequency offset causes phase shift

$$\hat{\Omega} T_{\mathrm{s}} = \arg\left\{ \sum_{\mathrm{j}=0}^{2} z_{l+1,p_{\mathrm{f}}(j)}\left(\hat{f}_{\mathrm{acq}}\right) z^{*}_{l,p_{\mathrm{f}}(j)}\left(\hat{f}_{\mathrm{acq}}\right) \right\}$$

$z_{l,k}$: Output of the FFT unit for the *l* -th symbol and the *k*-th sub-carrier

T_{s} : Duration of one symbol

$p_{\mathrm{f}}(j)$: Positions of frequency pilots

Time Tracking

- Using averaged IFFT-transformation of windowed channel estimation $\left(\hat{H}_{k,l}\right)$ for estimation of channel impulse response

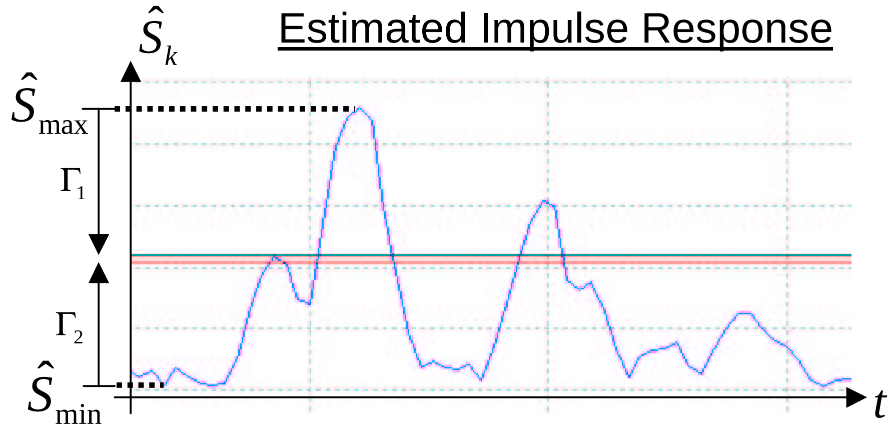
$$\hat{S}_m(k) = \frac{1}{N_{\mathrm{TiTr}}} \sum_{i=0}^{N_{\mathrm{TiTr}}-1} \left| \ \mathrm{IFFT}\left\{ \ \hat{H}_{k-i,l} \right\} \right|^2$$

- Afterwards using peak detection for first path estimation

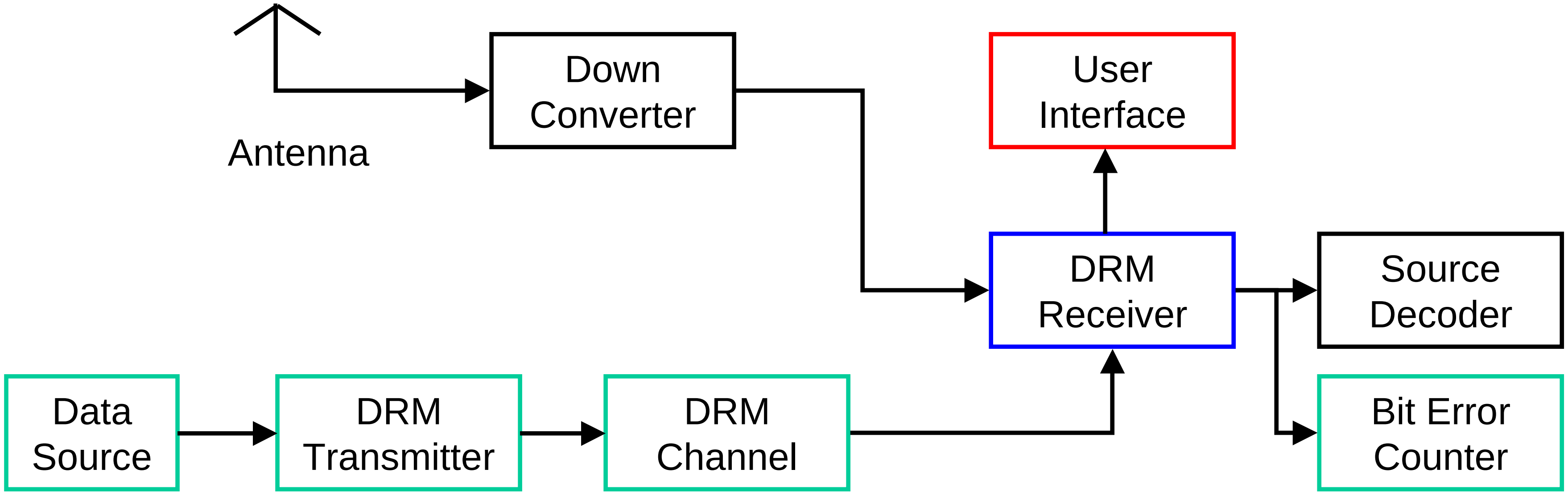
$$-\hat{e} = \frac{1}{2} \min\left\{ \ m \,|\, \hat{S}_m(k) > \Gamma, \text{and } \ \hat{S}_m(k) > \hat{S}_{m+1}(k) \right\}$$

  - Definition of the bound Γ

$$\Gamma = \max\left\{ \hat{S}_{\max} \times 10^{-\frac{\Gamma_1}{10}}, \hat{S}_{\min} \times 10^{-\frac{\Gamma_2}{10}} \right\}$$



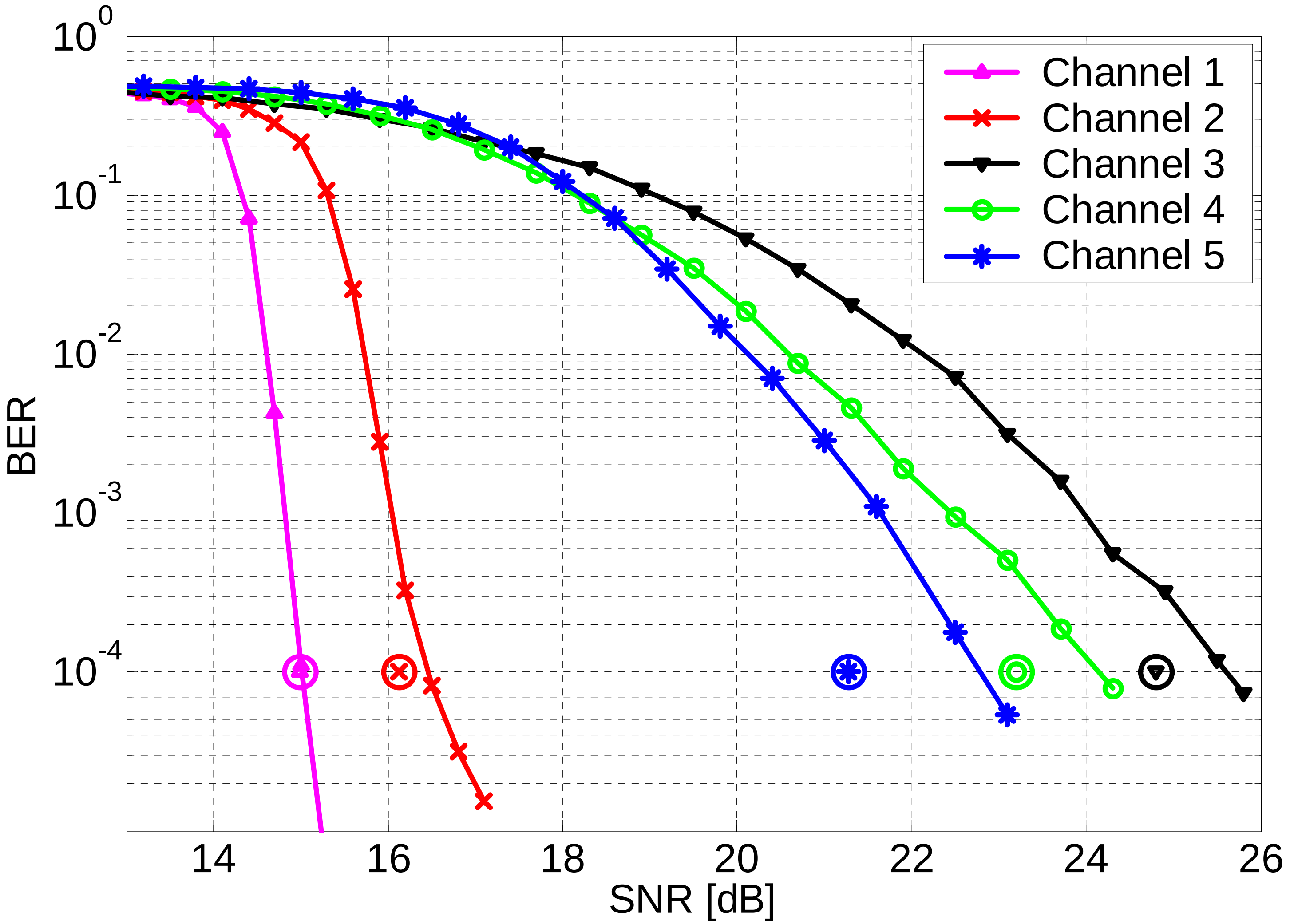
Software Modules



Software can be used:

- together with Down Converter and Source Decoder to receive real-time DRM radio broadcast
    - Source coding currently limited to plain MPEG4 AAC (no SBR, no CELP/HVXC)
- for BER or Channel Estimation Simulations with build in Data Source, Transmitter and Channel Simulator

Simulation Results



Outlook

- **Developed DRM Receiver operates close to the possible limits**
  - Try to close the gap between ideal channel estimation and realisation
    - ICI compensation
    - Decision directed channel estimation
    - Noise cancellation for narrow-band interference
- **Software runs real-time on a 700 MHz Pentium PC**
  - Improve to allow „background“ reception
    - Use SIMD instructions to speed up (MMX, SSE etc.)
    - Improve „pipelining“ of the algorithms to make acquisition phase shorter
- **Source Decoder (faad2) needs additional features (SBR, CELP, HVXC)**

See http://drm.sourceforge.net for details and download

Hard- and Software