OPEN-SOURCE IMPLEMENTATION OF A DIGITAL RADIO MONDIALE (DRM) RECEIVER

A F Kurpiers and V Fischer

Darmstadt University of Technology, Germany

Abstract

The characteristics of a DRM pass-band signal allow to implement a real-time software receiver on a conventional personal computer. In this paper we propose a possible structure for such an implementation. Algorithms for channel estimation and equalization as well as methods for synchronization are shortly described together with their relevance for the entire system. The performance of the system is then evaluated by simulation.

1 Introduction

DRM is a new OFDM-based digital radio standard for the long-, medium- and short-wave ranges which was formed by an international consortium [1]. It is designed to use the same frequency allocation as the current analogue systems with the aim to replace it owing to the advantages of the new digital system. The perceived audio quality is much better and additional digital information can be transmitted. Four transmission modes (so called robustness modes A-D) and different system bandwidths are defined to cope with different allocation schemes and channel conditions. The bandwidth of a DRM pass-band signal is less than 20 kHz and the number of carriers used in the OFDM-modulation is relatively small (max. 460). These features motivate a real-time software implementation of a DRM receiver on a conventional personal computer (PC) using the sound card as the input and output device. An object oriented approach using C++ is advantageous to handle the complexity of the DRM standard.

To cope with the time variant transmission channel, channel estimation and equalization is required. Very good performance was achieved using separate Wiener interpolation in time and frequency direction [6]. These results are compared to simpler structures.

Additionally, to receive a digital data stream, synchronization is mandatory [2], [3], [4]. The frequency offset needs to be estimated and corrected and also the symbol timing has to be adjusted. We distinguish between two operation modes, the so called tracking mode and acquisition mode. First, in an acquisition step the initial offsets are estimated roughly so that as a second step the tracking algorithm can be used to achieve low residual errors.

Finally, we evaluate the performance of the resulting receiver. The DRM standard specifies test channels as the environment to use for simulation. We will report on bit error rate simulations on these channels.

The paper is organized as follows: in Sec. 2 the system model considered is presented. Channel estimation is covered in Sec. 3. The different synchronization algorithms are presented in Sec. 4. Sec. 5 deals with the implementation of the algorithms covered in the previous sections. The performance of the complete receiver is analyzed by computer simulations in Sec. 6.

2 System Model

We assume an OFDM signal $s(t)$, which is transmitted over a time variant channel and disturbed by white Gaussian noise. Transmit and receive filters and channel together yield the impulse response $h(\tau,t)$. At the receiver this signal is sampled with the sample rate $f_s = \frac{1}{T}$ and yields the received signal samples:

$$r_n = e^{j2\pi f_o nT} \sum_{m=0}^{M} h_m(n)s((n-m)T) + n_n, \qquad (1)$$

where the channel is assumed to have limited support of M samples (shorter than the length of the guard interval N_g), $h_m(n) = h(mT,nT)$ and n_n are the white Gaussian noise samples. The frequency offset between transmitter and receiver is f_o. Assuming a correctly placed DFT in the intersymbol-interference (ISI) free region of the received signal r_n and only a small residual frequency error Δf, we demodulate the OFDM symbol k and get for sub-carrier index l:

$$z_{k,l} = x_{k,l}H_{k,l}e^{j2\pi\phi_l k\frac{N_s}{N}} + \tilde{n}_{k,l}, \qquad (2)$$

where N is the length of the DFT, $N_s = N + N_g$ is the length of one OFDM symbol in samples, $H_{k,l}$ is the channel transfer function for sub-carrier l including time-invariant phase offsets due to frequency offsets, $x_{k,l}$ is the transmitted data symbol and $\tilde{n}_{k,l}$ is the white Gaussian noise plus the intercarrier-interference (ICI) caused by the loss of orthogonality due to time variations of the channel and the frequency offset. The frequency error Δf and the sample rate offset $\zeta = \frac{f_s - f_s'}{f_s'}$ are incorporated into the phase $\phi_l = \Delta f NT + \zeta l$.

3 Channel Estimation and Equalization

Due to the time variant and frequency selective nature of the channel encountered on DRM transmissions, the channel has to be estimated continuously and its influence on the data transmission eliminated. This is done by using the known scattered pilots c transmitted in the OFDM signal. These pilots represent noisy estimates of the channel transfer function at their respective positions. After transmission over the channel and assuming correct synchronization we get (see Equ. 2): $z_{k',l'} = c_{k',l'} H_{k',l'} + n_{k',l'}$, where k' and l' are the pilot positions in time and frequency. For equalization these estimates have to be filtered to remove the noise and interpolated to get the channel transfer function on all OFDM carriers.

3.1 Wiener Interpolation

The optimal solution to this problem is the two dimensional Wiener interpolator [6] that can be computed if the channel statistics are known:

$$\hat{\mathbf{H}} = R_{\mathbf{H}\hat{\mathbf{c}}} R_{\hat{\mathbf{c}}\hat{\mathbf{c}}}^{-1} \hat{\mathbf{c}} = \mathbf{w}\hat{\mathbf{c}}, \tag{3}$$

where $R_{\mathbf{H}\hat{\mathbf{c}}}$ is the cross-covariance matrix with elements $r_{\mathbf{H}\hat{\mathbf{c}}_{k,l}} = E\{H_{k,l}\hat{c}^*_{k',l'}\} = r(k-k', l-l')$ and $R_{\hat{\mathbf{c}}\hat{\mathbf{c}}}$ is the auto-covariance matrix of the pilots with elements $r_{\hat{\mathbf{c}}\hat{\mathbf{c}}_{k'',l''}} = E\{\hat{c}_{k'',l''}\hat{c}^*_{k',l'}\} = r(k''-k', l''-l')$ and $r(\Delta k, \Delta l)$ the two-dimensional auto-correlation function of the channel [2]. The coefficients of the Wiener interpolator are $\mathbf{w}$. Unfortunately, the channel statistics are in general unknown at least at the beginning of reception so that only a worst case assumption on the channel is possible. The second problem is that two dimensional filtering requires a prohibitively high effort. Thus, the 2D Wiener interpolator is separated into two 1D filters. This can be done without noticeable performance penalty [6] because the auto-correlation function of the fading channel process is separable:

$$r(\Delta k, \Delta l) = r_{f_d}(\Delta k) r_\tau(\Delta l) \tag{4}$$

where r_τ is the correlation in frequency direction (due to the delay spread) and r_{f_d} is the correlation in time direction (due to the Doppler spread).
For the ionospheric channel the Doppler spectral density is usually modeled as Gaussian with bandwidth $2\sigma_d$ and the delay profile consists of a number of distinct paths that are independently fading (WSSUS model). We assume worst case channel conditions with the highest Doppler spread $\sigma_{d_{\max}}$ possible for the given pilot density and a uniform delay power profile with a maximum delay spread equal to the OFDM guard interval:

$$\begin{aligned} r_{f_{d\text{wc}}}(\Delta k) &= \exp\left(-2(\sigma_{d_{\max}} \pi N T \Delta k)^2\right), & (5) \\ r_{\tau_{\text{wc}}}(\Delta l) &= \operatorname{sinc}\left(\Delta l \frac{N_g}{N}\right). & (6) \end{aligned}$$

We first interpolate the pilots in time direction and afterwards in frequency direction.
The channel statistics, the signal to noise ratio (SNR) and the number of taps of the Wiener interpolator determine the minimum mean squared error (MMSE) of the estimate. Furthermore, if there is a model mismatch the mean squared error (MSE) will be higher. We analyzed the MSE and found only a weak dependence on the SNR and a noticeable dependence on the actual channel correlation, but the loss was tolerable if the bandwidth of the filter was chosen according to the worst case conditions given above.
During channel estimation it is possible to estimate the actual time and frequency correlation of the channel and adapt the Wiener interpolators to achieve an even lower MSE.

3.2 Sub-optimal Interpolation

The Wiener interpolators require a fairly high processing power so we looked for alternatives to reduce this load. In time direction for a lot of channels linear interpolation seems sufficient if in frequency direction a Wiener interpolator is employed to improve the SNR of the estimate. For the interpolation and noise reduction in frequency direction some papers like [7] propose DFT based algorithms. The main drawback despite their simplicity is the large estimation error on the edges of the OFDM signal due to the DFT used in these algorithms that assumes the channel to be periodic in frequency, which it is of course not. The suggested use of a window function does not improve the MSE much, it is still a lot higher than with the Wiener interpolator so we decided not to use a DFT based interpolation.

4 Synchronization

The synchronization of a DRM stream is divided into different procedures. The most common operations are the frequency synchronization and the OFDM symbol timing. However, for DRM additional synchronization is needed. The OFDM symbols are bundled up in frames. These frames are always 400 ms long. The first OFDM symbol of each frame contains additional pilots for frame synchronization. To be able to perform the channel estimation and de-map the OFDM multiplex, the receiver must know the beginning of a frame.
Another problem is the existence of the different robustness modes which differ in the main parameters like carrier spacing $\frac{1}{NT}$ or guard interval length N_g. Except of the frequency acquisition all other units need the mode to be detected first.
The proposed synchronization strategy is as follows: the first step is to acquire a coarse frequency offset estimation. This is possible without having knowledge about the current robustness mode because it is based on three frequency pilots which are common for all robustness modes. The next step is to detect the robustness mode and perform timing acquisition. Both algorithms utilize a correlation based on the received signal. Since the DRM standard defines different possible bandwidths and we do not know the correct one at this point, we have to assume the smallest one possible to make sure no other signal interferes our estimation process. Thus, the received signal is filtered with a pass-band filter having a bandwidth of the narrowest DRM mode and having

a center frequency defined by the frequency acquisition before calculating the correlation. Since we have a band-pass signal at this point, we design the filter as a Hilbert filter to get the analytic complex base-band signal.
After determining the right robustness mode and the timing, the useful part of the OFDM signal can be extracted and demodulated. At this time the frequency offset tracking can be activated since it also uses the independent frequency pilots. Now the beginning of a DRM frame needs to be detected. Having the information about the frame position, the channel estimation and timing tracking can be started.
An additional sample rate correction is needed since the PC sound cards can have large sample rate offsets which degrade the receiver performance [3].
The resulting structure of the receiver is shown in Fig. 1.

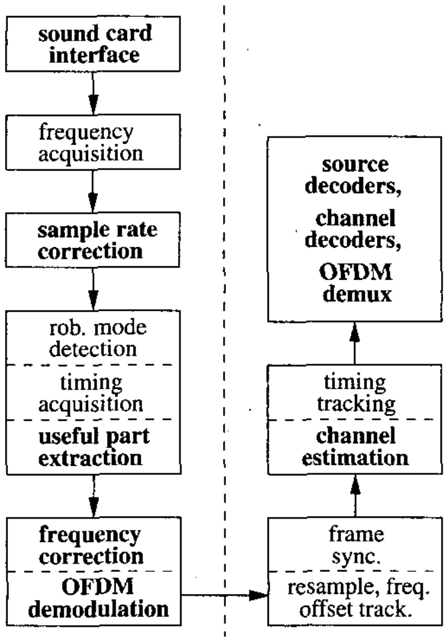

Figure 1: Structure of the receiver.

In the following sections all synchronization algorithms are explained in detail.

4.1 Frequency Acquisition

The frequency acquisition is based on [3] but improved for fading channels. First, we average the estimated power spectral density of the received signal:

$$\widetilde{R}_{k,l} = \frac{1}{N_{\mathrm{FrAc}}} \sum_{i=0}^{N_{\mathrm{FrAc}}-1} \left| \sum_{n=0}^{4N_s-1} r_{n+(k-i)N_s} e^{-j\frac{2\pi}{4N_s}nl} \right|^2 , \quad (7)$$

where r_n is the received signal at time index n and N_{FrAc} is the number of spectra used for averaging. The correlation of the known pilot positions with $\widetilde{R}_{k,l}$ results in

$$\alpha_{k,l} = \sum_{i=0}^{2} \widetilde{R}_{k,l+p_{\mathrm{fac}}(i)} , \quad (8)$$

with

$$p_{\mathrm{fac}}(i) = 4\, p_{\mathrm{f}}(i) \left(\frac{N_g}{N} + 1 \right) , \quad (9)$$

where $p_{\mathrm{f}}(i)$ are the carrier indices of the frequency pilots tabulated in [1].
Considering the attenuations in the spectrum of frequency selective channels, the peak detection can be improved by normalizing the correlation result $\alpha_{k,l}$ to the signal power. The power can be approximated by low-pass filtering the signal $\alpha_{k,l}$ along the frequency axis resulting in a signal $\sigma^2_{k,l}$. The final frequency acquisition estimate is given by:

$$\hat{f}_{\mathrm{acq}}(k) = \frac{f_s}{4N_s} \arg\max_l \left\{ \frac{\alpha_{k,l}}{\sigma^2_{k,l}} \right\} . \quad (10)$$

4.2 Timing Acquisition

The cyclic prefix of an OFDM signal can be utilized in a timing acquisition algorithm [8]. Based on the ML criterion the following algorithm is derived:

$$\lambda(i) = \sum_{n=i}^{i+N_g-1} \left[|r_n r^*_{n+N}| - \left(|r_n|^2 + |r_{n+N}|^2 \right) \right] . \quad (11)$$

The ML estimate of the timing phase is then $i_{ML} = \arg\max_i \{\lambda(i)\}$. For a time dispersive channel we expect several maxima for all the distinctive paths, so we have modified this estimate to take this into account and collect the energy over the guard interval time [2]:

$$\Theta_{ML_{mod}} = \arg\max_{\Theta} \left\{ \sum_{m=\Theta}^{\Theta+N_g-1} \lambda(m) \right\} . \quad (12)$$

4.3 Robustness Mode Detection

The robustness mode detection is based on the guard interval correlation which is also utilized in the timing acquisition in Equ. 11. We calculate the correlation using the system parameters for each robustness mode separately. The idea is that distinct peaks being one symbol interval apart will show up in the correlation result if the correct parameters were chosen. To detect this periodic signal, we propose to correlate the result with a cosine function of the frequency $f_c = \frac{1}{N_s T}$:

$$\beta = \sum_{i=0}^{N_{\mathrm{RoDe}}N_s-1} \lambda(i) \cos\left(\frac{2\pi}{N_s} i \right) , \quad (13)$$

where N_{RoDe} is the number of symbols used for detection. The robustness mode whose system parameters maximize the value β is chosen.

4.4 Frame Synchronization

The frame synchronization utilizes the time pilots inserted at the beginning of each DRM frame. These pilots are located in groups so that we can divide them into pairs of pilots being direct neighbors. We now assume that the channel is identical at adjacent pilot positions, i.e. $H_{k,p_t(i)} \approx H_{k,p_t(i)+1}$, where $p_t(i)$ denotes the position of the first pilot of a pair. With this assumption we can eliminate the channel and calculate the squared distance between the received cells and the pilot cells:

$$\gamma(k) = \sum_{i=0}^{L_T-1} \left| z_{k,p_t(i)} \frac{c_{k,p_t(i)+1}}{c_{k,p_t(i)}} e^{-j\frac{2\pi}{N}\frac{N_g}{2}} - z_{k,p_t(i)+1} \right|^2, \tag{14}$$

where L_T is the number of pilot pairs. The exponential term compensates for the target timing position of the timing acquisition which is in the middle of the guard interval. The beginning of a frame is found if the total squared error $\{\gamma(k), k = 0, \cdots, N_{\text{fra}}\}$ is minimum, where N_{fra} is the number of symbols in a DRM frame.

4.5 Frequency and Sample Rate Offset Tracking

In [9], a frequency estimator is derived which is suitable for frequency tracking in frequency selective channels:

$$\hat{\Omega} T_s = \arg\left\{ \sum_{j=0}^{L_F-1} \left(z_{k+1,p_f(j)}(\hat{f}_{\text{acq}}) z^*_{k,p_f(j)}(\hat{f}_{\text{acq}}) \right) \right\}, \tag{15}$$

where $\hat{\Omega} = 2\pi\Delta\hat{f}$ and $p_f(j)$ are the positions of the frequency pilots. This estimator is utilized to track the three ($L_F = 3$) frequency correction pilots in the DRM signal. The estimation is based on the frequency error dependent phase increment between successive symbols. The parameter $\hat{f}_{\text{acq}}$ shall mean that the output of the FFT unit is based on the initial frequency offset estimate done by the acquisition unit (see Equ. 10).
The sample rate offset estimator uses the frequency offset estimation of each of the three frequency pilots. It calculates the difference between two pilot frequencies and relates it to the desired one. This can be written as:

$$\hat{\zeta} = \frac{N}{4\pi N_s} \sum_{i=1}^{2} \frac{\hat{\Omega}_i - \hat{\Omega}_0}{p_f(i) - p_f(0)}, \tag{16}$$

$$\hat{\Omega}_i T_s = \arg\left\{ z_{k+1,p_f(i)}(\hat{f}_{acq}) z^*_{k,p_f(i)}(\hat{f}_{acq}) \right\}. \tag{17}$$

The variance of the sample rate estimation is much higher than the frequency offset estimation since very small frequency differences are analyzed. This drawback can be compensated by a closed loop structure.

4.6 Timing Tracking

The timing estimation obtained from the guard interval correlation has a high variance and is not very reliable since only a fractional part of the useful signal is utilized. A reliable timing is very important for the performance of the receiver since timing offsets can cause inter-symbol-interference (ISI) or even cause the synchronization to get lost. To improve the timing, we use a method described in [4] which is based on the results from the channel estimation. If we transform the estimated transfer function $\hat{H}_{k,l}$ at symbol index k multiplied with a Hamming window function $\{w_l | l = 0, \cdots, N-1\}$ into the time domain and do an averaging over N_{TiTr} frames, we get an estimation of the channel power delay profile:

$$\hat{S}_m(k) = \frac{1}{N_{\text{TiTr}}} \sum_{i=0}^{N_{\text{TiTr}}-1} \left| \frac{1}{N} \sum_{l=0}^{N-1} \hat{H}_{k-i,l}\, w_l\, e^{j\frac{2\pi}{N}ml} \right|^2. \tag{18}$$

From $\{\hat{S}_m(k), m = 0, \cdots, N-1\}$, an estimate of the offset of the first path delay can be derived by detecting the first peak which is above a certain bound. This bound is introduced to make the estimation less sensitive to SNR and to reduce the probability of selecting a noisy or noise-only path. Hence, the estimated first path time delay is

$$-\hat{e}(k) = \frac{1}{2} \min\left\{ m | \hat{S}_m(k) > \Gamma, \text{ and } \hat{S}_m(k) > \hat{S}_{m+1}(k) \right\}, \tag{19}$$

where the bound Γ is defined as

$$\Gamma = \max\left\{ \hat{S}_{\max} \cdot 10^{-\frac{\Gamma_1}{10\text{dB}}}, \hat{S}_{\min} \cdot 10^{\frac{\Gamma_2}{10\text{dB}}} \right\} \tag{20}$$

and $\hat{S}_{\max}$ and $\hat{S}_{\min}$ are the maximum and minimum values of $\hat{S}_m(k)$, respectively. Good results were achieved by setting both Γ_1 and Γ_2 to 25 dB.

5 Implementation Issues

A DRM receiver consists of different logical units, e.g. OFDM-demodulation, channel decoder or demultiplexer [1], which we will call modules in the following. Each module is defined to work frame-wise where the size of the frames need not be the same in all modules. Therefore an intermediate buffer must be provided for managing the data transfer between different modules. To separate the specific implementation of a module from the standard task of organizing the data transfer between the modules, an object-oriented implementation was chosen. Furthermore, this type of implementation increases the clearness and provides an easier maintenance of the resulting code.
Using the C++ language it is possible to implement the data transfer in a base class and derive all modules from that class inheriting its functionality. The task for the base class (we call it *CModule*) is to check the input intermediate buffer whether enough data is available or not. If data is available, the processing routine of the derived class is called. It produces an output which again is stored in the intermediate buffer for the following module. This buffer is implemented in a separate class called *CBuffer*. This buffer can be a cyclic buffer if the frame sizes of the different modules are not the same or if the frame sizes change with time, e.g., in the sample rate correction module. If both modules have the same frame size the *CBuffer* object is simply a vector. The resulting framework is illustrated in Fig. 2.
The software runs in real-time on a 700 MHz Pentium III PC. Currently the software is not optimized to utilize the SIMD instructions modern processors offer like MMX

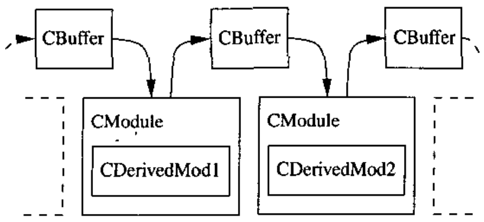

Figure 2: Illustration of the dependency between *CModule, CBuffer* and derived modules.

or SSE. A profiling analysis of the software indicated that the run time behavior is mostly determined by the Viterbi algorithm running in the multilevel decoder.

6 Simulation Results

The simulations are made in compliance with the parameters[1] used in Annex A of the DRM standard [1]. The channels are WSSUS and defined in this standard, where channel 1 is AWGN and channel 2 is a very slow fading channel. The channels 3-5 are fast fading channels. The encircled marks in Fig. 3 indicate the required SNR to

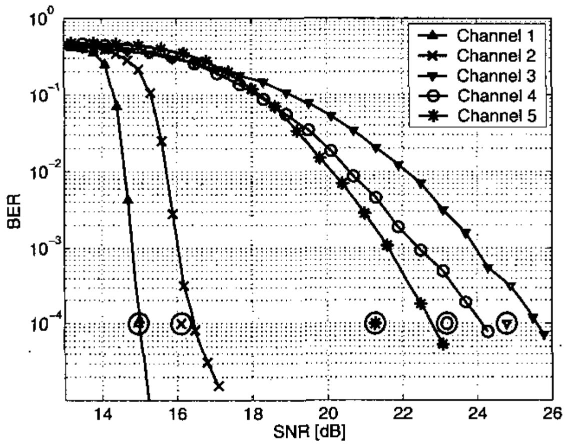

Figure 3: Overall receiver performance. Robustness mode A is used with the channels 1 and 2, mode B for the channels 3-5. The markers at an SNR of 10^{-4} are the results for ideal channel estimation and synchronization.

achieve a BER of 10^{-4} for these channels assuming ideal channel estimation and synchronization.
We simulated the overall tracking performance of our receiver. As seen in Fig. 3, the curves for the slow fading channels are very close to the values given for ideal channel estimation whereas on fast fading channels we loose approx. 1 dB. This loss is due to the fast fading that causes more noise in the channel estimation and additional noise owing to ICI on the demodulated symbols.

[1]The parameters are 64-QAM modulation, code rate $R = 0.6$, long cell interleaving (approx. 2 s) and a bandwidth of 10 kHz. The signal power includes pilots and the guard interval, the noise bandwidth equals the nominal signal bandwidth. Robustness mode A is used for the channels 1 and 2, mode B for the channels 3-5.

7 Conclusion

In this paper we presented algorithms for synchronization and channel estimation needed to decode a DRM stream. Additionally, we addressed some implementation issues.
An open source project implementing a software DRM receiver [10] successfully utilizes the described system proposal. Several tests with live DRM transmissions showed that the acquisition and tracking performance as well as the channel estimation are sufficient for decoding DRM.

References

[1] European Telecommunications Standards Institute, Digital Radio Mondiale (DRM), 2001, System Specification ETSI TS 101980

[2] R. Van Nee and R. Prasad, OFDM for wireless multimedia communications, 2000, Artech House, Boston

[3] V. Fischer and A. Kurpiers, Frequency Synchronization Strategy for a PC-based DRM Receiver, 2002, 7th Int. OFDM-Workshop (InOWo'02), Hamburg

[4] B. G. Yang, K. B. Letaief, R. S. Cheng and Z. Cao: Timing Recovery for OFDM Transmission, 2000, IEEE J-SAC, Vol. 18(11), 2278–2291

[5] F. Classen et al, Channel Estimation Unit for an OFDM System suitable for Mobile Communication, 1995, ITG Fachtagung Mobile Kommunikation, ITG Fachbericht 135, 457–466

[6] P. Hoeher et al, Two-Dimensional Pilot-Symbol-Aided Channel Estimation By Wiener Filtering, 1997, IEEE Conf. Acoustics, Speech and Signal Proc., ICASSP-97, 1845–1848

[7] Yang B. G. et al, Windowed DFT based pilot-symbol-aided channel estimation for OFDM systems in multipath fading channels, 2000, IEEE Int. Conf. on Vehicular Technology, VTC'00-Spring, 1480–1484

[8] J. van de Beek, M. Sandell, P. Borjesson, ML Estimation of Time and Frequency Offset in OFDM Systems, 1997, IEEE Trans. Signal Processing, Vol. 45(7), 1800–1805

[9] F. Claßen and H. Meyr, Synchronization Algorithms for an OFDM System for Mobile Communication, 1994, 1. ITG Fachtagung Codierung für Quelle, Kanal und Übertragung, ITG Fachbericht 130, 105–113

[10] http://drm.sourceforge.net