

Evaluating channel estimation methods for 802.11p systems

Master of Science Thesis in Communication Engineering

MATTIAS HERMANSSON
VIKTOR SKODA

Department of Signals and Systems
Division of Communication Systems and Information Theory
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2011
Report No. EX085/2010

REPORT NO. EX085/2010

Evaluating channel estimation methods in 802.11p systems.

MATTIAS HERMANSSON
VIKTOR SKODA

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2011

Evaluating channel estimation methods in 802.11p systems.

MATTIAS HERMANSSON
VIKTOR SKODA

© MATTIAS HERMANSSON, VIKTOR SKODA, 2011

Technical report no EX085/2010
Department of Signals and Systems
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Göteborg, Sweden 2011

Abstract

In this thesis work a number of channel estimation methods are implemented and evaluated in an existing 802.11p physical layer simulator. Conventional channel estimators like Least Squares, Minimum Mean Square Error and interpolation methods as well as more complex estimators such as decision feedback methods have been analyzed. The purpose of 802.11p is to standardize wireless access in vehicular environments. It is shown that conventional estimation methods do not perform sufficiently well because of factors like the high relative speed of vehicles and the delay spread of the channel. In this work the feedback estimators are shown to perform well, in some scenarios, but are sensitive to high relative speeds.

Acknowledgements

We would like to thank our examiner Prof. Erik Ström for giving us the opportunity to do this thesis work and for his valuable advice and insight during meetings. Furthermore we would also like to thank our supervisor Dr. Stylianos Papanastasiou for his work on the simulator and for his valuable support during this thesis work.

Contents

1	Introduction	4
2	Preliminaries	4
2.1	Description of 802.11p	4
2.2	802.11p vs 802.11a	5
2.3	System model	6
3	Channel estimation methods and symbol timing	9
3.1	LS estimator	10
3.2	MMSE estimator	10
3.3	Interpolation estimators	11
3.4	Decision Feedback	13
3.5	Symbol timing	15
4	Wireless Channel models	17
4.1	Vehicular LOS	17
4.2	Vehicular NLOS	17
5	Simulation	17
5.1	Simulation setup	19
5.2	Scenarios	20
6	Results and Discussion	21
6.1	Perfect estimator	21
6.2	Simulation results and discussion	21
6.3	Symbol timing simulations	30
6.4	Complexity	30
7	Conclusions and Future Work	31

List of Figures

1	The subcarriers in an 802.11p OFDM symbol.	4
2	Structure of an 802.11p OFDM frame.	5
3	Structure of the pilot symbols in an 802.11p OFDM frame.	6
4	Simplified base-band structure of an 802.11p transmitter and receiver.	6
5	The system model.	8
6	A comparison of the four interpolation estimators.	12
7	The sinc interpolation estimator.	13
8	Block diagram for the decision feedback estimator.	14
9	Coarse timing metrics.	16
10	$ h_t(\theta, l) $ given different values of θ and l	16
11	LOS, Base Case.	22
12	LOS, variable relative speed, method-wise.	23
13	LOS, variable relative speed, relative speed-wise.	23
14	LOS, variable relative speed, LS, MLS and MMSE.	24
15	LOS, variable frame length.	25
16	NLOS, Base Case.	25
17	NLOS, variable relative speed.	26
18	NLOS, variable frame length	26
19	LOS vs NLOS, LS, MLS, MMSE.	27
20	LOS vs NLOS, feedback.	28
21	NLOS, variable N_{iter}	28
22	NLOS, variable N_{prev} , w_e , $v = 0$ m/s.	29
23	NLOS, variable N_{prev} , w_e , $v = 25$ m/s.	29
24	Symbol timing algorithm	30
25	Complexity analysis.	31

List of Tables

1	Data rates supported by 802.11p.	7
2	The channel estimation methods and their respective training and updating properties.	10
3	Parameters for Vehicular LOS.	17
4	Parameters for Vehicular NLOS.	18
5	The parameter values used in Scenarios 1-2.	20
6	The parameter values used in Scenarios 3-4.	20
7	The parameter values used in Scenario 5.	20
8	The parameter values used in Scenarios 6-7.	21

List of Abbreviations

AWGN	Additive White Gaussian Noise
BPSK	Binary Phase Shift Keying
DFT	Discrete Fourier Transform
FER	Frame Error Rate
FFT	Fast Fourier Transform
GI	Guard Interval
IFFT	Inverse Fast Fourier Transform
ISI	InterSymbol Interference
LOS	Line Of Sight
LS	Least Squares
LT	Long Training OFDM symbols
MMSE	Minimum Mean Square Error
MMSEpil	Minimum Mean Square Error Pilot
MLS	Modified Least Squares
NLOS	Non Line Of Sight
OFDM	Orthogonal Frequency Division Multiplexing
PS	Pilot Subcarrier
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying

Generally, an 802.11p OFDM frame consists of several concatenated OFDM symbols. In Figure 2 the general structure of an OFDM frame is presented. The duration of an OFDM symbol is $T_{FFT} = 6.4 \mu s$ with a cyclic prefix denoted as GI of duration $T_{GI} = 1.6 \mu s$ which is added for each OFDM symbol. GI2 denotes a cyclic prefix with duration twice that of a GI, i.e. $T_{GI2} = 1.6 \cdot 2 = 3.2 \mu s$.

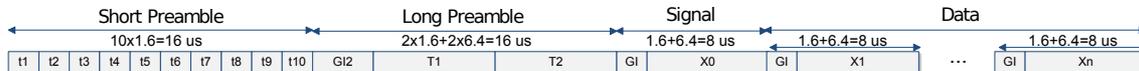


Figure 2: Structure of an 802.11p OFDM frame.

An OFDM frame begins with a short preamble that consists of 10 identical short training symbols (t_1, t_2, \dots, t_{10}), each with duration $1.6 \mu s$. The short training symbols are mainly used for signal detection, coarse frequency offset estimation and timing synchronization. A long preamble follows, consisting of two identical long training symbols (T_1, T_2), each with duration $6.4 \mu s$. The long training symbols are mainly used for fine frequency offset estimation and channel estimation. It should be noted that since the 10 short training symbols in the short preamble are identical they also function as cyclic prefixes for each other, i.e., t_1 is a cyclic prefix for t_2 , which in turn is a cyclic prefix for t_3 and so on. This is also true for the T_1 and T_2 symbols in the long preamble.

Following the training sequences in an OFDM frame is a SIGNAL OFDM symbol which contains information about the length of the OFDM frame and the type of modulation and coding rate used for the remainder of the OFDM frame. Then, the next OFDM symbol contains a scrambling sequence as well as actual data; subsequent symbols only contain data.

Figure 3 shows the pilot symbols and data symbols. Note that no distinction is made between the SIGNAL OFDM symbol and the actual data OFDM symbols since there is no difference between them from a channel estimation point of view. The symbols at frequency indexes $-26 \rightarrow -1$ in Figure 3 correspond to indexes $38 \rightarrow 63$ in Figure 1 and, similarly, frequency indexes $1 \rightarrow 26$ in Figure 3 correspond to index $1 \rightarrow 26$ in Figure 1. The two OFDM symbols in Figure 3 with symbol indexes (T_1, T_2) correspond to the long preamble and each consists of 52 pilot subcarriers. The SIGNAL OFDM symbol corresponds to symbol index 0 and the Data symbols correspond to symbol indexes $1, \dots, N_{frame}$. As mentioned, the SIGNAL and data OFDM symbols, each have 4 pilot subcarriers — the remaining 48 subcarriers consist of actual data.

The 802.11p amendment supports 4 different modulation techniques, namely BPSK, QPSK, 16-QAM, 64-QAM. It also makes use of error-correction and interleaving. With respect to error-correction, convolutional coding is used with the coding rates $R = 1/2, 2/3$ or $3/4$. Combining different coding rates with different modulation techniques results in 8 different data rates, as summarized in Table 1. Only the data rates of 3, 6 and 12 Mbit/s are mandatory in 802.11p.

Finally, the channel spacing is 10 MHz, subcarrier spacing is $10/64$ MHz, and the difference between the lowest and highest subcarrier is $(53/64) \cdot 10$ MHz. Figure 4 describes a simplified block diagram for a transmitter and receiver that utilize 802.11p.

2.2 802.11p vs 802.11a

Although, Section 17 in the 802.11 standard allows operations in 5, 10 or 20 MHz mode, the p amendment defines operation in 10 MHz mode only. As 802.11p uses a 10 MHz

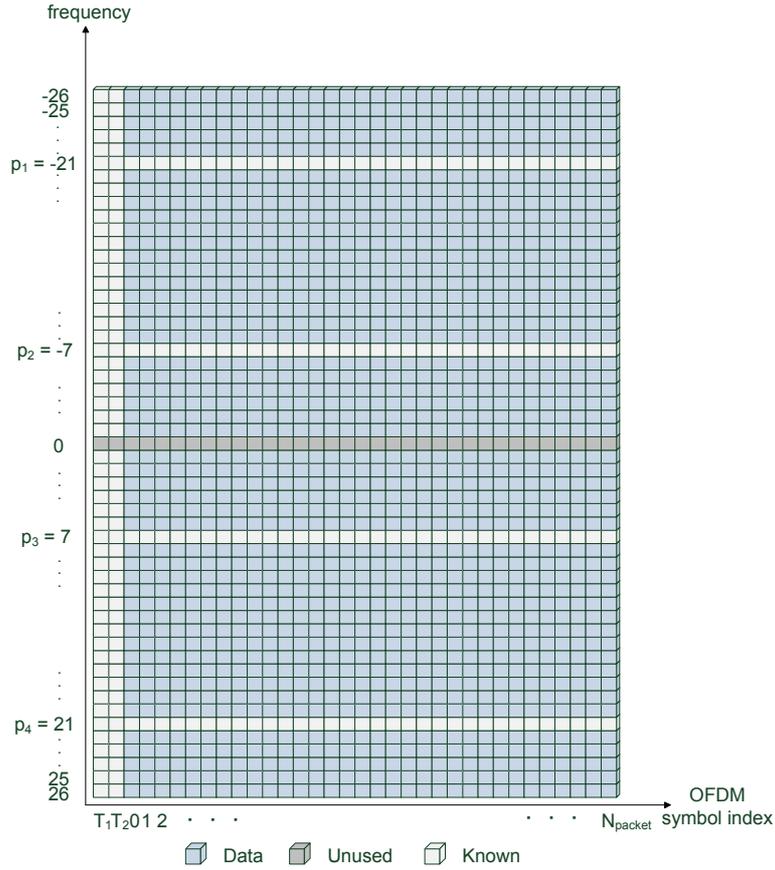


Figure 3: Structure of the pilot symbols in an 802.11p OFDM frame.

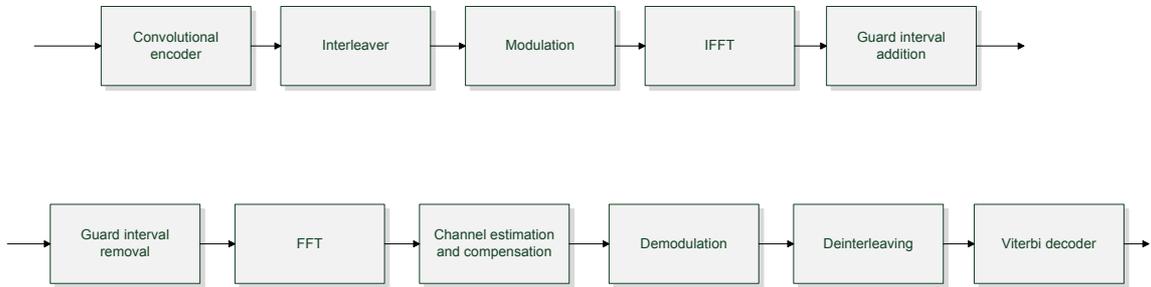


Figure 4: Simplified base-band structure of an 802.11p transmitter and receiver.

channel bandwidth, the symbol times in the time domain get doubled compared to the 802.11a in 20 MHz mode and the cyclic prefix becomes $1.6 \mu\text{s}$ instead of $0.8 \mu\text{s}$ — this provides additional robustness against delay spread. The carrier frequency for a 802.11a system is 5 GHz, while the carrier frequency for a 802.11p system is 5.9 GHz [2], [3].

2.3 System model

We consider a system as shown in Figure 5. The input to the system is an OFDM frame, \mathbf{x}_{frame} , that is a sequence of OFDM symbols. In turn, the OFDM symbols contain modulated symbols such that $\mathbf{x}_n = [x_n(-26) \dots x_n(26)]^T$ where \mathbf{x}_n is the n th OFDM symbol. The OFDM frame consists of a preamble, $n \in \{t1, \dots, t10, T1, T2\}$, and data, $n \in \{0, \dots, N_{frame}\}$. The data has been encoded and modulated as described in Section 2.1. The OFDM symbols are transformed to time domain by a 64-point IFFT operation.

Modulation	Coding rate	Data rate (Mbit/s)
BPSK	1/2	3
BPSK	3/4	4.5
QPSK	1/2	6
QPSK	3/4	9
16-QAM	1/2	12
16-QAM	3/4	18
64-QAM	2/3	24
64-QAM	3/4	27

Table 1: Data rates supported by 802.11p.

The input to the IFFT operator is defined as

$$\mathbf{x}_{64,n} = [x_{64,n}(0) \dots x_{64,n}(63)]^T = [0 \ x_n(1) \dots x_n(26) \ 0 \dots 0 \ x_n(-26) \dots x_n(-1)]^T.$$

The transformed time domain samples are concatenated and GIs are inserted to form $\mathbf{x}_{t,frame} = [x_t(0) \dots x_t(N_t - 1)]$, where N_t is the total number of time samples in the OFDM frame. The transmitted signal is distorted by a multitap channel with impulse response $\mathbf{h}_t(t) = [h_t(t, 0) \dots h_t(t, L - 1)]^T$ at discrete time $t = d \cdot T_s$ where $d \in \mathbb{Z}$ (i.e. d is an integer), $T_s = 0.1 \mu\text{s}$ is the sampling time and L is the number of taps. Note that the subscript t denotes time domain samples and that the parenthesized t denotes a specific point in time. Here, we make the following two assumptions:

1. The channel impulse response is approximately constant during one OFDM-symbol. This assumption is valid if $f_D \cdot T_{SYM} \leq 0.01$ [4] where f_D is the maximum Doppler frequency. Since $f_D = f_c \cdot v/c$ where v is the relative speed between the transmitter and the receiver and c is the speed of light, the allowed relative speed under this assumption is

$$v \leq \frac{0.01 \cdot c}{T_{SYM} f_c} = \frac{0.01 \cdot 3 \cdot 10^8}{8 \cdot 10^{-6} \cdot 5.9 \cdot 10^9} \approx 64 \text{ m/s}.$$

2. The channel delay spread is shorter than the GI duration, T_{GI} . When this assumption is met there is no ISI.

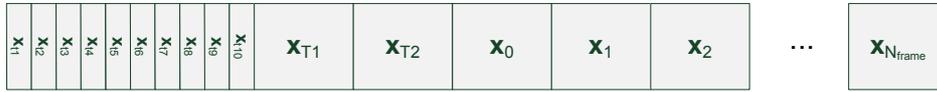
Under these assumptions the received signal is

$$y_{t,frame}(t) = \sum_{l=0}^{L-1} h_t(t, l) x_{t,frame}(t - l) + w_t(t)$$

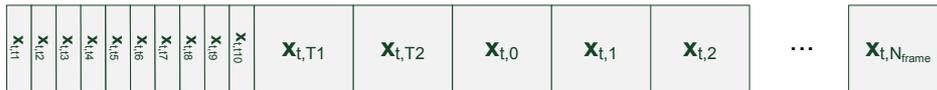
where $w_t(t)$ is AWGN with variance σ_w^2 . Since the channel is approximately constant during the transmission of a single OFDM-symbol the time index of the channel can be dropped and the notation introduced to \mathbf{x} can be applied to the channel as well. Hence $\mathbf{h}_{t,n} = [h_{t,n}(0) \dots h_{t,n}(L - 1)]^T$ is the channel impulse response during transmission of $\mathbf{x}_{t,n}$ and the received signal is then

$$y_{t,n}(t) = \sum_{l=0}^{L-1} h_{t,n}(l) x_{t,n}(t - l) + w_{t,n}(t)$$

$\mathbf{x}_{\text{frame}}$:



$\mathbf{x}_{a,\text{frame}}$:



$\mathbf{x}_{t,\text{frame}}$:

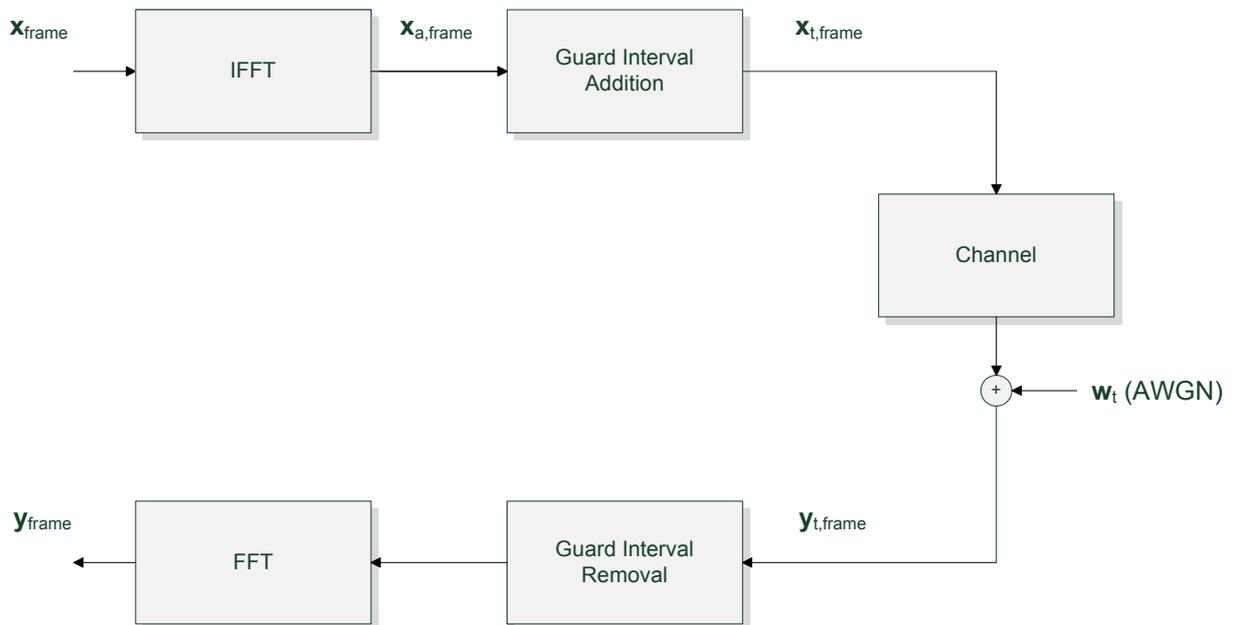
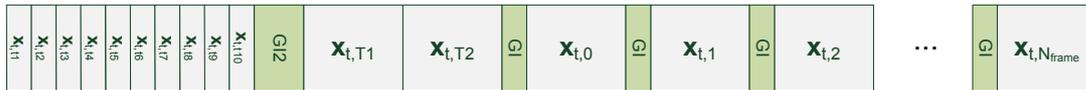


Figure 5: The system model.

and can be expressed in the frequency domain, for the discrete frequency indexes $f = 0, \dots, 63$, as

$$\begin{aligned} y_{64,n}(f) &= \left(\sum_{l=0}^{L-1} h_n(l) e^{-j2\pi \frac{fl}{64}} \right) x_{64,n}(f) + w_n(f) \\ &= h_{64,n}(f) x_{64,n}(f) + w_n(f) \end{aligned}$$

or rewritten

$$y_n(f) = h_n(f) x_n(f) + w_n(f)$$

where $w_n(f)$ is the 64-point discrete Fourier transform of $w_n(t)$. The channel frequency response \mathbf{h}_n (or $\mathbf{h}_{64,n}$) and the received signal \mathbf{y}_n (or $\mathbf{y}_{64,n}$) have the same structure as the transmitted signal \mathbf{x}_n (or $\mathbf{x}_{64,n}$).

Moreover, the pilot structure of the OFDM frame is shown in Figure 3. There are two long training OFDM-symbols, \mathbf{x}_{T1} and \mathbf{x}_{T2} , in the preamble of the OFDM frame and four pilot subcarriers, $x_n(f)$ for $f \in \{p_1 = -21, p_2 = -7, p_3 = 7, p_4 = 21\}$.

Finally, some of the estimation methods presented use the DFT-matrix defined as

$$\mathbf{F} = \begin{bmatrix} W_N^{00} & \dots & W_N^{0(N-1)} \\ \vdots & \ddots & \vdots \\ W_N^{(N-1)0} & \dots & W_N^{(N-1)(N-1)} \end{bmatrix} \quad (1)$$

in an N-point DFT where

$$W_N^{k_1 k_2} = \frac{1}{\sqrt{N}} e^{-j2\pi \frac{k_1 k_2}{N}}.$$

Note that, \mathbf{F}_a is a submatrix of \mathbf{F} consisting of the first a columns.

3 Channel estimation methods and symbol timing

This section presents the ten channel estimation methods (estimators) and the symbol synchronization method evaluated in this work. The base concept behind the estimators is to attempt to characterize the channel by using the training part of the OFDM frame, i.e. the long training OFDM symbols ($T1$, $T2$) and/or the pilot subcarriers (p_1 , p_2 , p_3 and p_4). To increase robustness against AWGN, the mean of the two received symbols $\mathbf{y}_{64,T12} = \frac{1}{2}(\mathbf{y}_{64,T1} + \mathbf{y}_{64,T2})$ is used by the estimators. For notation purposes \mathbf{X} is defined as a diagonal matrix containing $\mathbf{x}_{64,T1}$ (i.e. $\mathbf{X} = \text{diag}(\mathbf{x}_{64,T1})$).

Some estimators use the long training OFDM symbols only to produce a single channel estimate, $\hat{\mathbf{h}}_{64}$, which is used for equalization throughout the OFDM frame. Intuitively, these estimators are efficient when the channel is slowly fading and the frame is short. Some estimators use the pilot subcarriers of the current OFDM symbol, n , to give the current channel estimate, $\hat{\mathbf{h}}_n$. Since the channel estimate is updated for every OFDM symbol these estimators are a better choice when the channel is fading faster, however, their performance is not satisfactory for channels with great delay spreads. There are also two feedback based estimators that are initialized with the long training OFDM symbols and then updated with an iterative feedback algorithm.

Overall, the estimators can be categorized by two properties:

Training: refers to the type of training used: either LT = Long training symbols or PS = Pilot subcarriers, or both.

S	Name	Training	Update
1	Least squares (LS)	LT	Not updated
2	Modified Least squares (MLS)	LT	Not updated
3	Minimum mean square error (MMSE)	LT	Not updated
4	Minimum mean square error pilot (MMSE pilot)	PS	Updated
5	Linear interpolation	PS	Updated
6	Third order polynomial interpolation	PS	Updated
7	Natural cubic spline interpolation	PS	Updated
8	Sinc interpolation	PS	Updated
9	Feedback LS	LT	Updated
10	Feedback MLS	LT and PS	Updated

Table 2: The channel estimation methods and their respective training and updating properties.

Update: refers to whether the estimators provide a single channel estimate from the long training symbols or the channel estimate is updated for each OFDM symbol.

A list of the estimators considered here can be found in Table 2 specifying their training and updating properties. Finally, a symbol timing method is presented that combines an auto-correlation method with a path delay estimator.

3.1 LS estimator

The Least Squares (LS) estimator is given by [5]

$$\hat{\mathbf{h}}_{64} = \mathbf{X}^{-1} \mathbf{y}_{64, T_{12}}. \quad (2)$$

As there are zeros on some subcarriers (see Section 2.1) \mathbf{X} is not invertible and $\hat{\mathbf{h}}_{64}$ is obtained by dividing $y_{64, T_{12}}$ by the nonzero elements of x_{64, T_1} element-wise.

3.1.1 Modified LS estimator

It is assumed that the channel impulse response is shorter than the GI duration (see Section 2.3). This assumption can be used to increase the performance of the LS estimator. The Modified LS estimator takes into account only the first T_{GI}/T_s ($= 16$) samples of the channel impulse response. It is given by [5]

$$\hat{\mathbf{h}}_{64} = \mathbf{F}_{16} (\mathbf{F}_{16}^H \mathbf{X}^H \mathbf{X} \mathbf{F}_{16})^{-1} \mathbf{F}_{16}^H \mathbf{X}^H \mathbf{y}_{64, T_{12}}. \quad (3)$$

3.2 MMSE estimator

The Minimum Mean Square Error (MMSE) estimator is given by [5]

$$\hat{\mathbf{h}}_{64} = \mathbf{F} \mathbf{R}_{\mathbf{h}\mathbf{y}} \mathbf{R}_{\mathbf{y}\mathbf{y}}^{-1} \mathbf{y}_{64, T_{12}} \quad (4)$$

where

$$\mathbf{R}_{\mathbf{h}\mathbf{y}} = E\{\mathbf{h}_t(t) \mathbf{y}_t^H\} = \mathbf{R}_{\mathbf{h}\mathbf{h}} \mathbf{F}^H \mathbf{X}^H \quad (5)$$

$$\mathbf{R}_{\mathbf{y}\mathbf{y}} = E\{\mathbf{y}_t \mathbf{y}_t^H\} = \mathbf{X} \mathbf{F} \mathbf{R}_{\mathbf{h}\mathbf{h}} \mathbf{F}^H \mathbf{X}^H + \sigma_w^2 \mathbf{I}_{64} \quad (6)$$

are the cross covariance matrix between $\mathbf{h}_t(t)$ and \mathbf{y}_t and the auto-covariance matrix of \mathbf{y}_t . $\mathbf{R}_{\mathbf{h}\mathbf{h}}$ is the auto-covariance matrix of $\mathbf{h}_t(t)$, σ_w^2 is the noise variance $E\{|w_t(t)|^2\}$ and \mathbf{I}_{64} is the identity matrix of size 64. To be able to implement the MMSE estimator the noise variance, σ_w^2 , and the channel auto-covariance matrix, $\mathbf{R}_{\mathbf{h}\mathbf{h}}$, need to be estimated.

3.2.1 Noise variance estimation

According to [6], the noise variance can be estimated by utilizing the long training symbols in the preamble, as follows:

$$\hat{\sigma}_w^2 = \frac{1}{2 \cdot 64} \sum_{f=0}^{63} |y_{T1}(f) - y_{T2}(f)|^2 \quad (7)$$

3.2.2 Channel autocorrelation matrix estimation

To estimate the channel autocorrelation matrix some older channel estimates are needed. If the channel estimates are stored during ongoing communication, they can then be used for this computation in the following manner:

$$\hat{\mathbf{R}}_{\mathbf{h}\mathbf{h}} = E\{\hat{\mathbf{h}}_t(t)\hat{\mathbf{h}}_t(t)^H\} \quad (8)$$

The channel estimate $\hat{\mathbf{h}}(t)$ can be obtained by e.g. LS estimation.

3.2.3 MMSE pilot estimator

The MMSE pilot estimator is based on the MMSE estimator but the pilot subcarriers are used instead of the long training symbols. \mathbf{X}_4 has the same structure as \mathbf{X} but with the elements corresponding to data-subcarriers set to zero. \mathbf{y}_4 and \mathbf{y}_{64} have the same relation as \mathbf{X}_4 and \mathbf{X} .

$$\hat{\mathbf{h}}_{64} = \mathbf{F}\mathbf{R}_{\mathbf{h}\mathbf{y}}\mathbf{R}_{\mathbf{y}\mathbf{y}}^{-1}\mathbf{y}_4 \quad (9)$$

where

$$\mathbf{R}_{\mathbf{h}\mathbf{y}} = \mathbf{R}_{\mathbf{h}\mathbf{h}}\mathbf{F}^H\mathbf{X}_4^H \quad (10)$$

$$\mathbf{R}_{\mathbf{y}\mathbf{y}} = \mathbf{X}_4\mathbf{F}\mathbf{R}_{\mathbf{h}\mathbf{h}}\mathbf{F}^H\mathbf{X}_4^H + \sigma_w^2\mathbf{I}_{64} \quad (11)$$

3.3 Interpolation estimators

Four interpolation estimators are considered. The real part and the imaginary part of $h_n(f)$ are interpolated separately. Figure 6 shows the absolute value curves of the interpolation estimators given the pilot sequence, p_i , where $i \in \{1, 2, 3, 4\}$. The underlying scheme for channel estimation at the pilot subcarriers is LS estimation.

3.3.1 Linear interpolation estimator

A linear interpolation estimate is defined as:

$$\hat{h}_n(f) = \begin{cases} \hat{h}_n(p_1) + (\hat{h}_n(p_2) - \hat{h}_n(p_1)) \cdot \frac{f-p_1}{p_2-p_1}, & f < p_1 \\ \hat{h}_n(p_i) + (\hat{h}_n(p_{i+1}) - \hat{h}_n(p_i)) \cdot \frac{f-p_i}{p_{i+1}-p_i}, & f \in [p_i, p_{i+1}), i = 1, 2, 3 \\ \hat{h}_n(p_3) + (\hat{h}_n(p_4) - \hat{h}_n(p_3)) \cdot \frac{f-p_3}{p_4-p_3}, & f \geq p_4 \end{cases} \quad (12)$$

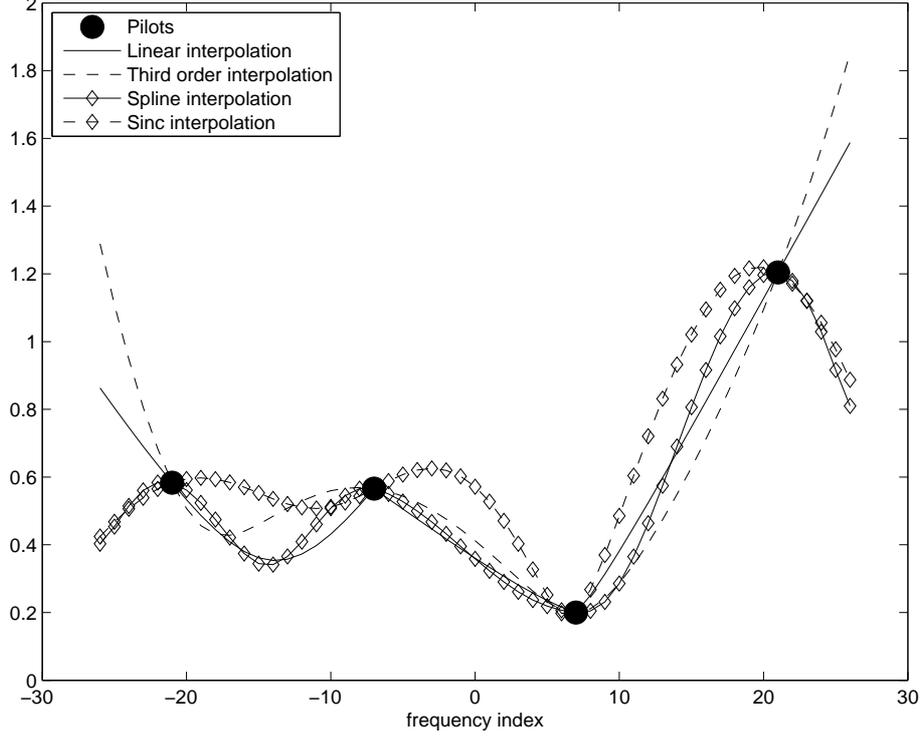


Figure 6: A comparison of the four interpolation estimators. Their absolute values are shown. Pilots, p_i , are marked with filled circles.

3.3.2 Third order polynomial interpolation estimator

A third order polynomial estimate that passes through all the pilot subcarriers is $\hat{h}_n(f) = a_{n,0} + a_{n,1} \cdot f + a_{n,2} \cdot f^2 + a_{n,3} \cdot f^3$ where the coefficients $a_{n,i}$ can be obtained by [7]

$$\begin{bmatrix} a_{n,0} \\ \vdots \\ a_{n,3} \end{bmatrix} = \begin{bmatrix} 1 & p_1 & p_1^2 & p_1^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & p_4 & p_4^2 & p_4^3 \end{bmatrix}^{-1} \begin{bmatrix} \hat{h}_n(p_1) \\ \vdots \\ \hat{h}_n(p_4) \end{bmatrix} \quad (13)$$

The polynomial is used for all f .

3.3.3 Natural cubic spline interpolation estimator

The frequency range is divided into three intervals $i \in \{1, 2, 3\}$, each represented by a frequency normalized third order polynomial $\hat{h}_{n,i}(f_{norm}) = a_{n,i} + b_{n,i} \cdot f_{norm} + c_{n,i} \cdot f_{norm}^2 + d_{n,i} \cdot f_{norm}^3$. The frequency is normalized such that $f \in [p_i, p_{i+1})$ is mapped to $f_{norm} \in [0, 1)$. The coefficients are given by [8]

$$a_{i,n} = \hat{h}_n(p_i) \quad (14)$$

$$b_{i,n} = \hat{h}_n'(p_i) \quad (15)$$

$$c_{i,n} = 3(\hat{h}_n(p_{i+1}) - \hat{h}_n(p_i)) - 2\hat{h}_n'(p_i) - \hat{h}_n'(p_{i+1}) \quad (16)$$

$$d_{i,n} = 2(\hat{h}_n(p_i) - \hat{h}_n(p_{i+1})) + \hat{h}_n'(p_i) + \hat{h}_n'(p_{i+1}) \quad (17)$$

where

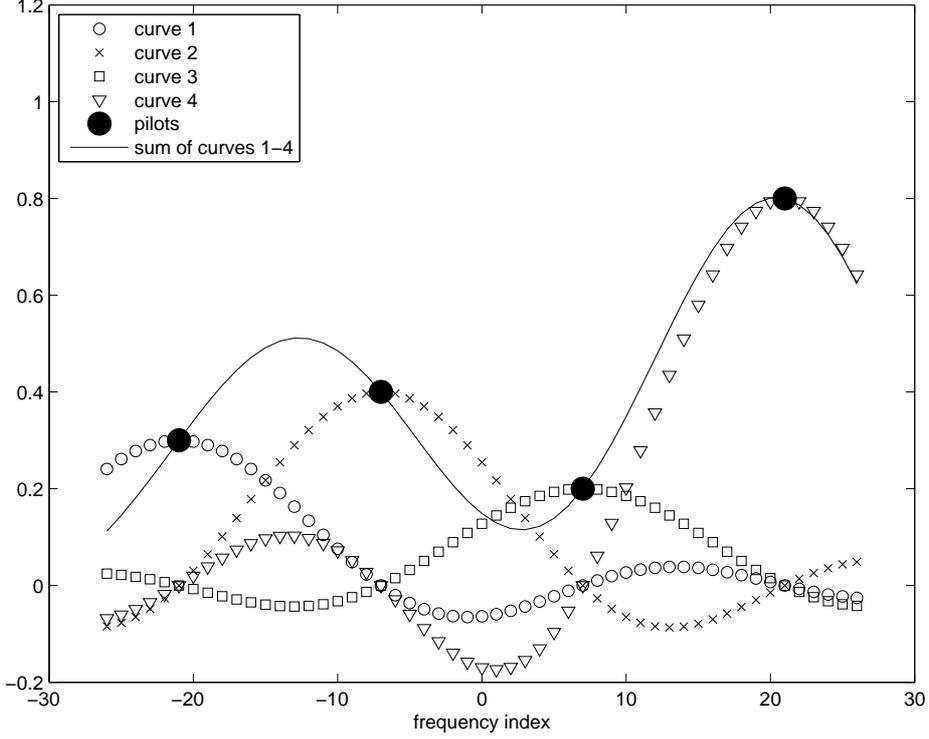


Figure 7: The four sinc curves centered at the pilot subcarriers and their sum (20), i.e. the channel estimate.

$$\begin{bmatrix} \hat{h}_n'(p_1) \\ \hat{h}_n'(p_2) \\ \hat{h}_n'(p_3) \\ \hat{h}_n'(p_4) \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 4 & 1 & 1 \\ 1 & 1 & 4 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix}^{-1} \begin{bmatrix} \hat{h}_n(p_1) \\ \hat{h}_n(p_2) \\ \hat{h}_n(p_3) \\ \hat{h}_n(p_4) \end{bmatrix} \quad (18)$$

Then

$$\hat{h}_n(f) = \begin{cases} \hat{h}_{n,1} \left(\frac{f-p_1}{p_2-p_1} \right), & f < p_1 \\ \hat{h}_{n,i} \left(\frac{f-p_i}{p_2-p_1} \right), & f \in [p_i, p_{i+1}), \quad i = 1, 2, 3 \\ \hat{h}_{n,3} \left(\frac{f-p_3}{p_2-p_1} \right), & f \geq p_4 \end{cases} \quad (19)$$

3.3.4 Sinc interpolation estimator

A sinc function is centered at each of the pilot subcarriers and have zero-crossings at the other pilot subcarriers, see Figure 7. The channel estimate is then the sum of all sinc functions. It can be expressed mathematically as [9]

$$\hat{h}_n(f) = \sum_{i=1}^{N_{SP}} \hat{h}_n(p_i) \cdot \text{sinc} \left(\frac{f - p_i}{p_2 - p_1} \right) \quad (20)$$

for all f .

3.4 Decision Feedback

The iterative decision feedback equalizer considered here is mainly based on the algorithm from [4] with some modifications. Figure 8 shows the general block diagram for the

decision feedback estimator. As the feedback equalizer makes use of some previous channel estimates, let N_{prev} denote the number of previous channel estimates. Let N_{iter} denote the number of iterations and w_e the weighting method to be used. As mentioned in Figure 3, n is the OFDM symbol index and $i \in \{0, \dots, N_{iter} - 1\}$ is the iteration index. The channel estimation methods considered are LS and MLS.

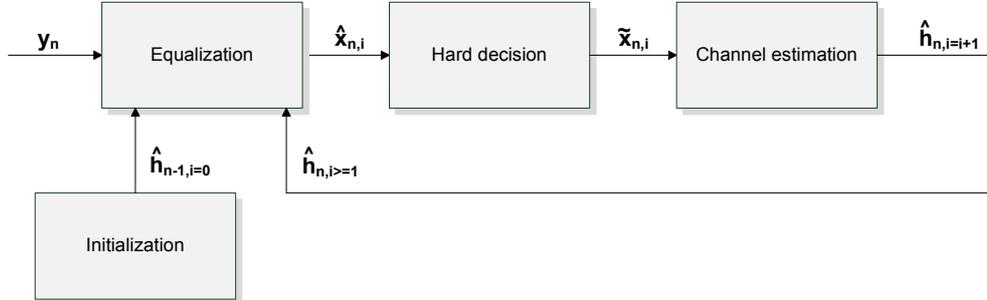


Figure 8: Block diagram for the decision feedback estimator.

3.4.1 Algorithm

The iterative algorithm may be outlined as follows:

Initialization Let $\hat{\mathbf{h}}_{n,i}$ denote the channel estimate for the n -th OFDM symbol in the OFDM packet and the i -th iteration. In the initialization step the channel estimate $\hat{\mathbf{h}}_{n,i=0}$ is obtained based on the long training symbols T_1, T_2 if $n = 0$. Otherwise if $n > 0$ the channel estimate $\hat{\mathbf{h}}_{n,i=0}$ is obtained based on channel estimates from previous OFDM symbols.

1. If $n > 0$ set $\hat{\mathbf{h}}_{n,i=0}$ to be the final channel estimates from N_{prev} previous OFDM symbols weighted according to some weighting method, see Section 3.4.2.
2. If $n = 0$ set $\hat{\mathbf{h}}_{n,i=0}$ to be the channel estimate calculated based on the long training symbols T_1, T_2 and a channel estimation method.

Update The updating step consists of two main steps: the equalization step and the channel estimation step. The output from the equalization step is an estimate for the symbols $\hat{\mathbf{x}}_{n,i}$. A hard decision is performed on $\hat{\mathbf{x}}_{n,i}$ to obtain $\tilde{\mathbf{x}}_{n,i}$. The output of the channel estimation step is an updated channel estimate $\hat{\mathbf{h}}_{n,i=i+1}$. The process is outlined below in more detail:

1. Equalization: If $i = 0$ the equalization step consists of equalizing the current OFDM symbol y_n with the channel estimate from the initialization step to obtain $\hat{\mathbf{x}}_{n,i}$. If $i \geq 1$ the equalization step consists of equalizing the current OFDM symbol y_n with the output from the channel estimation step of the previous iteration to obtain $\hat{\mathbf{x}}_{n,i}$. Hard decision is performed on $\hat{\mathbf{x}}_{n,i}$ to obtain the estimate $\tilde{\mathbf{x}}_{n,i}$.
2. Channel estimation: Given the hard decision estimate $\tilde{\mathbf{x}}_{n,i}$, replace the pilot subcarriers with their known values. Calculate the updated channel estimate based on a channel estimation method as a function of $\tilde{\mathbf{x}}_{n,i}$ and \mathbf{y}_n .
3. If $i \geq N_{iter}$ exit and continue on to process the next OFDM symbol \mathbf{y}_{n+1} and start at step 1, otherwise go to step 2 and set $i = i + 1$.

3.4.2 Weighting methods:

As mentioned above, a number of previous channel estimates is used for calculating the current estimate if $i = 0$ and $n > 0$. This introduces the problem that somehow the previous channel estimates must be assigned weights. This can be described mathematically as:

$$\hat{\mathbf{h}}_{n,i=0} = \sum_{j=1}^{\eta} w_j \hat{\mathbf{h}}_{n-j} \quad (21)$$

where $\eta = \min(n, N_{prev})$.

The constraint that $\sum_{j=1}^{\eta} w_j = 1$ is also true for the different weighting methods. Two weighting methods, w_e for $e \in \{0, 1\}$, are considered:

Averaging ($w_e = 0$) A straightforward way to assign the weights is to weight each previous channel estimate with the same weight, that is $w_j = \frac{1}{\eta}$, for $j = 1$ to η .

Linear ($w_e = 1$) The weights should be calculated as $w_{j+1} = -\frac{2}{(\eta+1)\eta}j + \frac{2}{\eta+1}$, for $j \in \{0, \dots, \eta - 1\}$.

3.5 Symbol timing

Symbol timing is an issue that needs to be considered in an 802.11p system. According to [10] this issue can be solved by combining an auto-correlation synchronization algorithm for coarse timing followed by a path delay estimator for fine timing. The coarse timing is based on an auto-correlation synchronization algorithm that makes use of two timing metrics $M_1(\theta)$, $M_2(\theta)$ and the short training symbols in the preamble. If $N_s = 16$ is the number of samples for a short training symbol, the timing metrics are defined as:

$$M_1(\theta) = \frac{\sum_{m=0}^{N_s-1} y_t(\theta + m)y_t^*(\theta + m + N_s)}{\sum_{m=0}^{N_s-1} |y_t(\theta + m)|^2} \quad (22)$$

$$M_2(\theta) = \frac{\sum_{m=0}^{N_s-1} y_t(\theta + m)y_t^*(\theta + m + 2N_s)}{\sum_{m=0}^{N_s-1} |y_t(\theta + m)|^2} \quad (23)$$

where * denotes the complex conjugate. The coarse timing estimate $\hat{\theta}$ is given by:

$$\hat{\theta} = \arg \max_{\theta} (M_1(\theta) - M_2(\theta)) \quad (24)$$

Typical plots for (22), (23) and (24) are shown in Figure 9 from which one can observe that $\hat{\theta}$ gives a coarse timing estimate for the sample where the beginning of the 9th short training symbol is located. The fine timing is obtained by finding the channel impulse response $h_t(\theta, l)$ of the window of $l = N_w$ samples that contains the maximum energy. The channel impulse response is obtained by utilizing the long training symbol (T_1) and performing an LS estimate in the frequency domain and transforming it back to the time domain. Let $\hat{h}_{\theta}(f)$ denote the LS estimate for a given subcarrier with frequency index f and given timing θ .

$$h_t(\theta, l) = \sum_{f=0}^{63} \hat{h}_{\theta}(f) e^{j\frac{2\pi fl}{64}} \quad , l = 0, 1, \dots, 63 \quad (25)$$

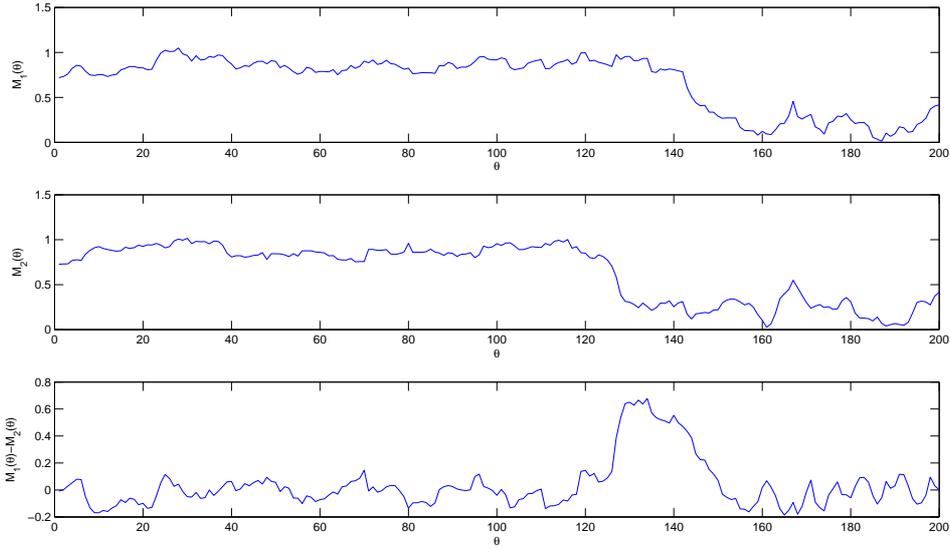


Figure 9: Coarse timing metrics.

N_w denotes the length of the channel impulse response (in number of samples). Obtaining the windows of N_w samples that contain maximum energy can be mathematically expressed as:

$$\hat{\theta}_e = \arg \max_{\theta} \sum_{l=0}^{N_w-1} |h_t(\theta, l)|^2 \quad (26)$$

The parameter N_w is unknown at the receiver and according to the system model the duration of the channel impulse response can not exceed the duration of the cyclic prefix. In Figure 10 one can observe the absolute value of the estimated channel impulse response $|h_t(\theta, l)|$ for different values of θ and l . Utilizing the knowledge of $|h_t(\theta, l)|$ and setting

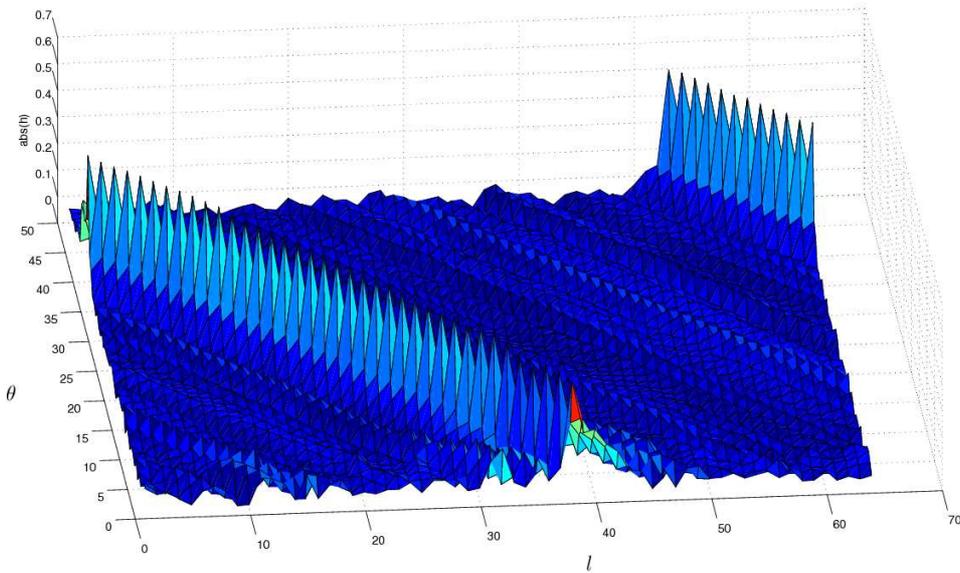


Figure 10: $|h_t(\theta, l)|$ given different values of θ and l .

Tap number	Tap delay (ns)	Tap power (dB)	Tap distribution	Doppler spectrum
1	0	0	Rician	Jakes
2	100	-6.3	Rayleigh	Jakes
3	200	-25.1	Rayleigh	Jakes
4	300	-22.7	Rayleigh	Jakes

Table 3: Parameters for Vehicular LOS.

$N_w = L$, equation (26) can be extended to (27), which introduces linear weighting.

$$\hat{\theta}_f = \arg \max_{\theta} \sum_{L=1}^{16} \sum_{l=0}^{L-1} |h_t(\theta, l)|^2 \quad (27)$$

4 Wireless Channel models

The channel estimation methods were evaluated using two different channel models, a Vehicular Line Of Sight channel model (Vehicular LOS) and a Vehicular Non Line Of Sight channel model (Vehicular NLOS). The two channel models are based on real-world measurements. Specifically, Vehicular LOS is based on [11] and describes a channel where there exists a strong line of sight component between the transmitter and receiver. Conversely, Vehicular NLOS describes a channel where such a strong line of sight component does not exist. It is a modified version of [12] received from Paul Gray at Cohda Wireless.

4.1 Vehicular LOS

The parameters for Vehicular LOS are summarized in Table 3. The channel parameters outline a multitap Rician channel where the path between transmitter and receiver is not blocked and so a strong line of sight component exists.

4.2 Vehicular NLOS

The parameters for Vehicular NLOS are summarized in Table 4. Vehicular NLOS describes a multitap Rayleigh channel where the path between the transmitter and receiver is blocked and there does not exist a strong line of sight component between the transmitter and receiver.

5 Simulation

The PhysLayerSim library [1] has been used as a basis for the simulator. PhysLayerSim contains helper functions and methods for 802.11p simulation; it operates in baseband, on a time-sample level and is written in C++ using the IT++ library [13]. In turn, IT++ is a library of mathematical, signal processing and communication classes and functions. Some minor modifications has been made to PhysLayerSim to ease the evaluation of the simulation results and, significantly, the channel estimation methods outlined in this report have been added.

Tap number	Tap delay (ns)	Tap power (dB)	Tap distribution	Doppler spectrum
1	0	-3.3	Rayleigh	Jakes
2	10	-3.6	Rayleigh	Jakes
3	20	-3.9	Rayleigh	Jakes
4	30	-4.2	Rayleigh	Jakes
5	50	0.0	Rayleigh	Jakes
6	80	-0.9	Rayleigh	Jakes
7	110	-1.7	Rayleigh	Jakes
8	140	-2.6	Rayleigh	Jakes
9	180	-1.5	Rayleigh	Jakes
10	230	-3.0	Rayleigh	Jakes
11	280	-4.4	Rayleigh	Jakes
12	330	-5.9	Rayleigh	Jakes
13	400	-5.3	Rayleigh	Jakes
14	490	-7.9	Rayleigh	Jakes
15	600	-9.4	Rayleigh	Jakes
16	730	-13.2	Rayleigh	Jakes
17	880	-16.3	Rayleigh	Jakes
18	1050	-21.2	Rayleigh	Jakes

Table 4: Parameters for Vehicular NLOS.

Design of a complete 802.11p system simulator can be distinguished into three parts: the transmitter, the channel and the receiver. The transmitter and receiver’s internal structure is shown in Figure 4. The output from the transmitter is an OFDM frame formed by time samples in baseband. The samples are sent through a channel (see Section 4) and AWGN is added. The receiver considers frame reception in three separate stages as reflected in the following three events:

Event 1: Synchronization The first event is concerned with finding the beginning of an OFDM frame. This is also where the channel is estimated (initially) in some of the channel estimation methods, using the long training OFDM symbols.

Event 2: SIGNAL decoding The SIGNAL OFDM symbol contains information about length of the OFDM frame and modulation format and is needed to decode the following symbols. This event is considered successful if the parity check is valid.

Event 3: Data decoding This is where the data is extracted and the transmitted data is compared with the received data to detect if there are any errors present. If no errors are detected, after hard-decision channel decoding, this event is considered successful.

The above events are considered sequentially and if any of the events fail, the OFDM frame is discarded immediately (i.e. the next events are not considered) and the OFDM frames reception is considered a failure. Reception is considered successful only if all events succeed. Note that the distinction of reception into separate events of interest per frame is, intuitively, completely compatible with the use of Frame Error Rate (FER) as a performance measure.

Since the channel bandwidth is 10 MHz the sampling time is set to $T_s = 0.1 \mu\text{s}$. When the signal is transmitted through the channel it is worth noting that IT++ merges the channel taps to the closest integer multiplier of the sampling time. All taps within $((i-0.5)T_s, (i+0.5)T_s]$ belong to the i th discrete tap (in the IT++ channel representation). As an example in the Vehicular NLOS channel model, specified in Section 4.2, tap number 1-5 are merged and represented by the first tap and tap number 6-8 are merged and represented by the second tap.

It is also of interest to outline how AWGN is calculated and added to the system. As the simulation results are presented in FER vs E_b/N_0 plots, AWGN is added to obtain a desired value of the E_b/N_0 ratio. In more detail N_0 is calculated as:

$$N_0 = \frac{E_{OFDM}}{N_{ST} \cdot R_c \cdot N_{BPSC} \cdot \left(\frac{E_b}{N_0}\right)} \quad (28)$$

where

$$E_{OFDM} = \text{E} \left\{ \sum_{n=0}^{N_{FFT}-1} |\mathbf{x}_t(n)|^2 \right\} \quad (29)$$

$N_{ST} = 52$ is the number of subcarriers that carry symbols. R_c is the coding rate. N_{BPSC} is the number of coded bits per subcarrier. E_{OFDM} is obtained from the transmitted OFDM frame before it is sent through the channel and the noise that is calculated according to (28) is added to the OFDM frame upon reception at the receiver.

5.1 Simulation setup

A number of different scenarios are simulated, characterized by the following three main parameters:

Channel model (C) Two channel models are implemented, a Vehicular LOS channel model and a Vehicular NLOS channel model, as discussed in Section 4.

Relative speed (v) The relative speed between the transmitter and the receiver, as discussed in Section 2.3.

Frame length (L_f) The number of bytes of information transmitted in an OFDM Frame.

There are also some parameters which are common for all scenarios:

AWGN The AWGN is added to get the desired $\frac{E_b}{N_0}$ -ratio, see Section 2.3 and Section 5.

Seven levels of AWGN have been applied: $\frac{E_b}{N_0} = 0, 5, \dots, 30$ dB.

Number of Trials The simulations are performed 10 000 times.

Data Rate 6 Mbit/s (see Table 1)

The following parameters are only utilized for the Feedback estimator:

Number of iterations (N_{iter}) Specifies the number of iterations used in the feedback estimator. The parameter can only be specified for the Feedback MLS estimator. For the Feedback LS estimator the parameter is always one.

Number of previous channel estimates (N_{prev}) The number of previous channel estimates utilized in the Initialization step of the decision feedback estimator.

Weighting method (w_e) The weighting method utilized in the Initialization step of the decision feedback estimator.

Parameter	Scenario 1	Scenario 2
C	Vehicular LOS	Vehicular LOS
v	0, 10, 25, 50 m/s	25 m/s
L_f	300 bytes	300, 800 bytes

Table 5: The parameter values used in Scenarios 1-2.

Parameter	Scenario 3	Scenario 4
C	Vehicular NLOS	Vehicular NLOS
v	0, 10, 25, 50 m/s	25 m/s
L_f	300 bytes	300, 800 bytes

Table 6: The parameter values used in Scenarios 3-4.

5.2 Scenarios

The scenarios are designed such that the effect of the three main parameters can be evaluated one at a time.

5.2.1 Scenarios 1-2

The three main parameters for Scenarios 1-2 are shown in Table 5. The Vehicular LOS channel model is used in both scenarios, the relative speed is varied in Scenario 1 and the frame length is varied in Scenario 2.

5.2.2 Scenarios 3-4

The parameters for Scenarios 3-4 are shown in Table 6. The Vehicular NLOS channel model is used in both scenarios, the relative speed is varied in Scenario 3 and the frame length is varied in Scenario 4.

5.2.3 Scenario 5

In Scenario 5 the two channel models are analyzed, fixing the other parameters, as given in Table 7. Note also that there is no mobility.

5.2.4 Scenario 6

In Scenario 6 the effect of varying the parameter N_{iter} is investigated. The relative speed is varied and the frame length is constant. The other parameters in this scenario are given in Table 8.

Parameter	Scenario 5
C	Vehicular LOS, Vehicular NLOS
v	0 m/s
L_f	300 bytes

Table 7: The parameter values used in Scenario 5.

Parameter	Scenario 6	Scenario 7
C	Vehicular NLOS	Vehicular NLOS
v	0, 25 m/s	0, 25 m/s
L_f	300 bytes	300 bytes
w_e	0	0, 1
N_{iter}	1, 2, 3	1
N_{prev}	1	1, 2, 5

Table 8: The parameter values used in Scenarios 6-7.

5.2.5 Scenario 7

In Scenario 7 the effect of varying the parameters N_{prev} and w_e are investigated. The relative speed is varied and the frame length is constant. The other parameters in this scenario are given in Table 8.

6 Results and Discussion

In this section the simulation results presented in Frame Error Rate (FER) vs E_b/N_0 plots are discussed. The FER is calculated at the receiver as the ratio of discarded OFDM frames (see Section 5 for a more thorough description) over the total number of transmitted OFDM frames. The FERs obtained when using the different estimators are compared against what we term a Perfect estimator. The Perfect estimator is described in more detail below. This section also presents some results regarding the symbol timing method. There is also a comparison of the complexity of the different estimation methods. There is also an appraisal of the time complexity of the different estimation methods.

6.1 Perfect estimator

The results given by the estimators are contrasted against those obtained if the channel was perfectly known. We call this a Perfect estimator since it uses a perfect estimate of the channel for equalization. The Perfect estimator is implemented in the same manner as the other estimators, i.e. the channel estimate is given OFDM symbol-wise. This can be justified by assumption 1 in the system model (Section 2.3). It is, however, interesting to point out that since the channel is not strictly constant (just approximately) during an OFDM symbol, even in the absence of AWGN, $\hat{\mathbf{x}}_n \neq \mathbf{x}_n$. But according to the system model the difference between the transmitted symbols and their estimates is negligible, i.e. $\hat{\mathbf{x}}_n \approx \mathbf{x}_n$ (again, when there is no AWGN present in the system).

6.2 Simulation results and discussion

The simulation results are presented scenario-wise, according to the scenarios described in Section 5.2.

6.2.1 LOS, Base Case

The case where $v = 25$ m/s in Scenario 1 is termed the LOS, Base Case. All estimators except the LS, MLS and MMSE estimators are presented in Figure 11. These will

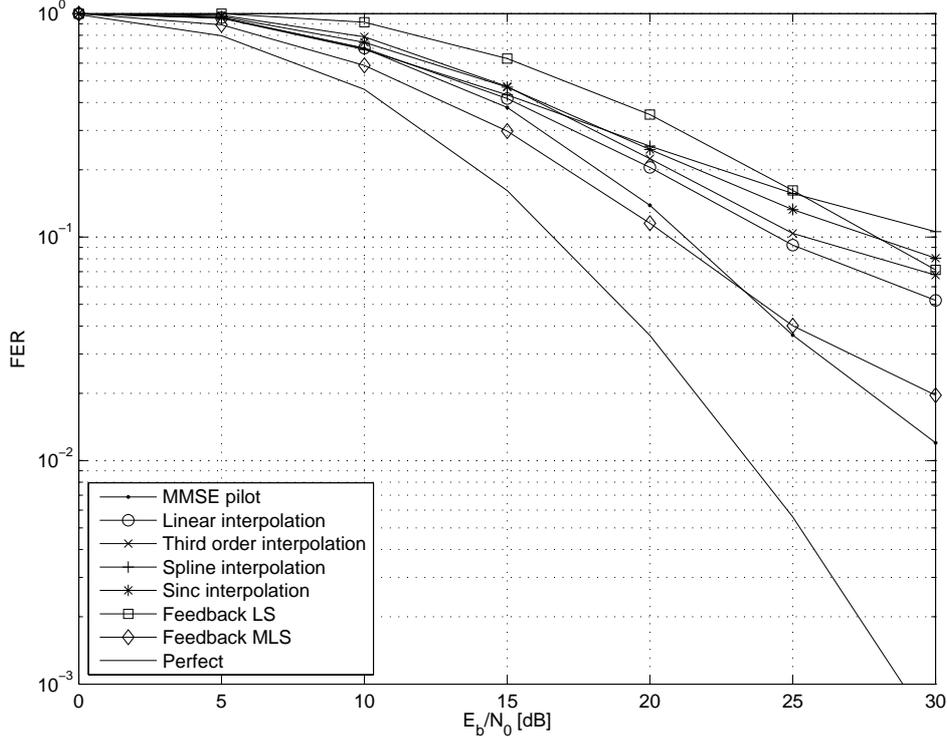


Figure 11: LOS, Base Case.

be presented later for lower relative speeds. Note that the MMSE pilot and interpolation estimators perform fairly well; this is mainly because the channel model has an LOS-component. As seen in Figure 11 the four interpolation estimators perform similarly, therefore, not all of them are evaluated in cases to follow. Typically the Linear interpolation estimator is shown in the plots to aid clarity, i.e. make the plots easier to examine.

6.2.2 LOS, variable relative speed

In Scenario 1 the relative speed parameter is varied and the effect on the FER is analyzed. In Figure 12 and Figure 13 the MMSE pilot, Linear interpolation, Feedback LS and Feedback MLS estimators are plotted for four different relative speeds. It can be seen that the relative speed affects the feedback LS and feedback MLS estimators more than the MMSE pilot and linear interpolation estimators. This can be explained by the dependence of previous OFDM symbols for the feedback estimators while the MMSE pilot and linear interpolation estimators depend only on the current OFDM symbol. This (dependence on previous OFDM symbols) might be more obvious in Figure 14 where the LS, MLS and MMSE estimators are plotted. These estimators rely on the long training OFDM symbols only and the results show that for a relative speed of 10 m/s the FERs are well above 0.1. This limit is important as a FER of less than 0.1 is needed for efficient communication [14].

6.2.3 LOS, variable frame length

The frame length is varied in Scenario 2. The effect of this is shown in Figure 15. The FER is generally higher for longer frames, however, the feedback estimators seem to be slightly

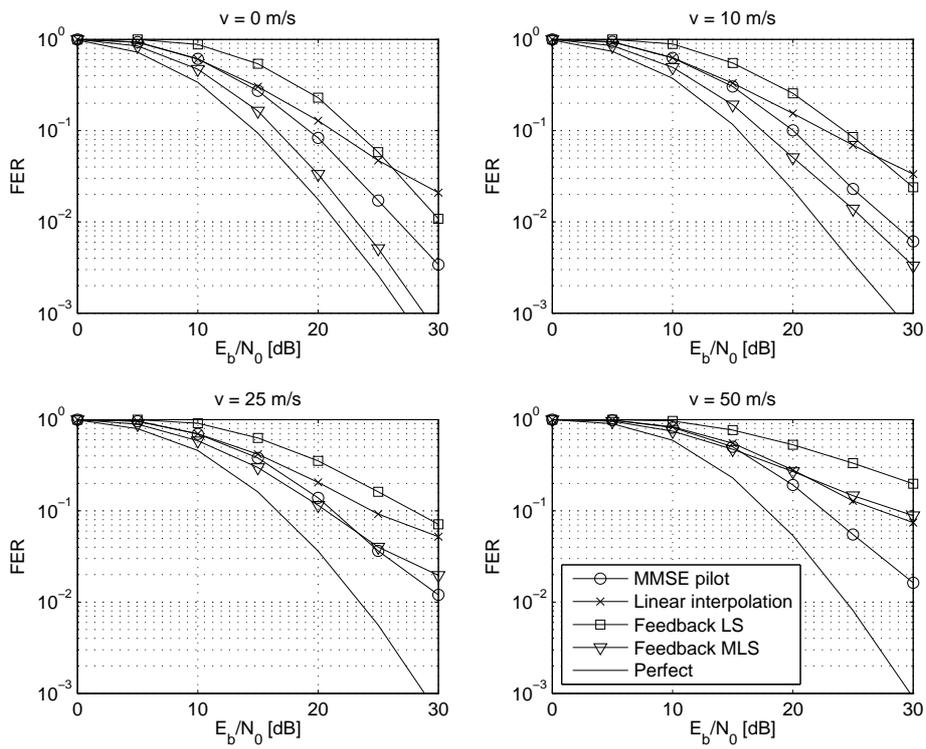


Figure 12: LOS, variable relative speed, method-wise.

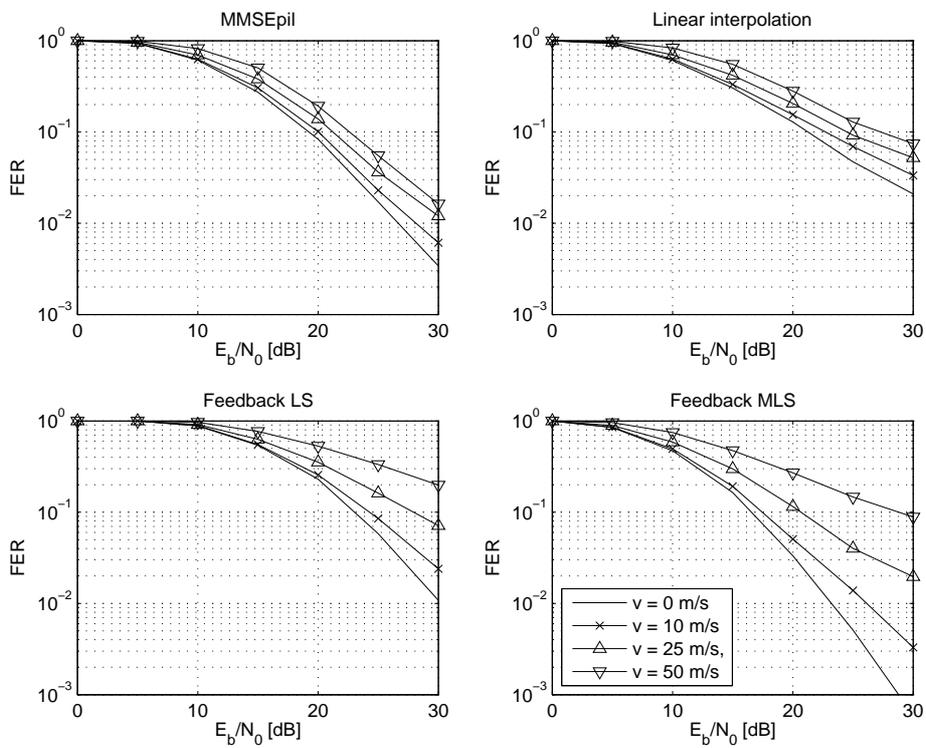


Figure 13: LOS, variable relative speed, relative speed-wise.

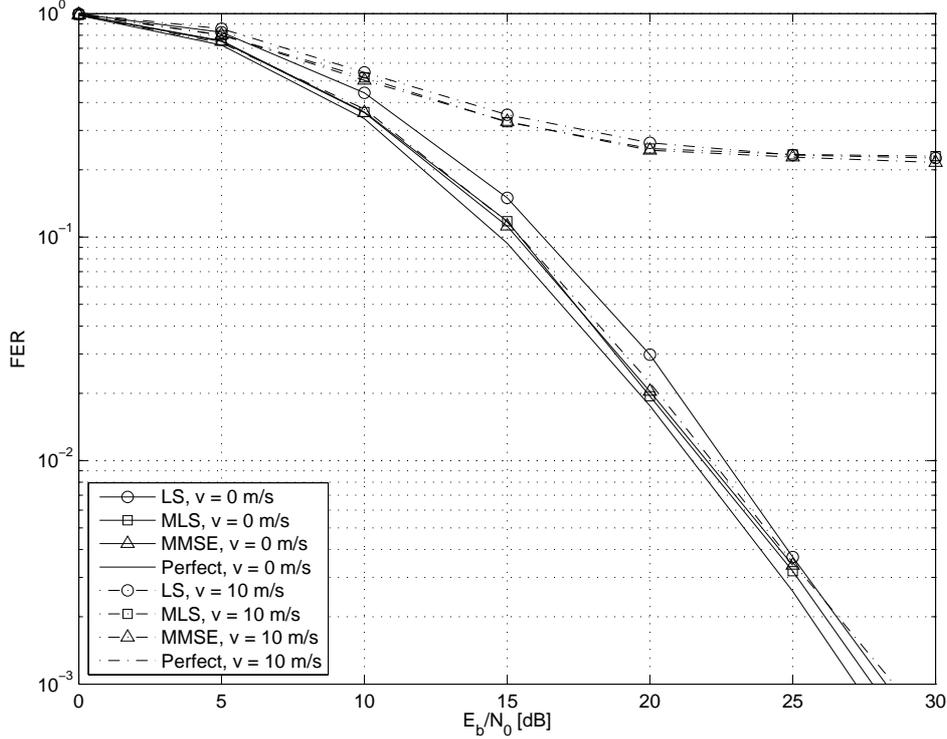


Figure 14: LOS, variable relative speed, LS, MLS and MMSE.

more sensitive to frame length than the Linear interpolation estimator. The feedback estimators rely on previous channel estimates and if a symbol error occurs it is reflected in the subsequent "previous channel estimates", thus one symbol error is likely to lead to more symbol errors and so on. As the frame length is increased more symbol errors are likely to occur, eventually to the extent where the channel coding is not capable of correcting all the erroneous bits and the OFDM frame is discarded.

6.2.4 NLOS, Base Case

The case where $v = 25$ m/s in Scenario 3 is termed the NLOS, Base Case. As can be seen in Figure 16 the MMSE pilot and interpolation estimators do not perform well, probably due to the small coherence bandwidth of the channel.

6.2.5 NLOS, variable relative speed

In Scenario 3 the relative speed parameter is varied and the effect on the FER is analyzed. Figure 17 shows that the impact on the performance of the Feedback MLS estimator is greater than that of the Perfect estimator, this is probably related to the property discussed in the LOS, variable relative speed case.

6.2.6 NLOS, variable frame length

The frame length is varied in Scenario 4. As in the LOS, variable frame length case, Figure 18 shows that the FER is increased when the frames are longer. The same discussion why the FER is increased as in the LOS, variable frame length case can be applied here.

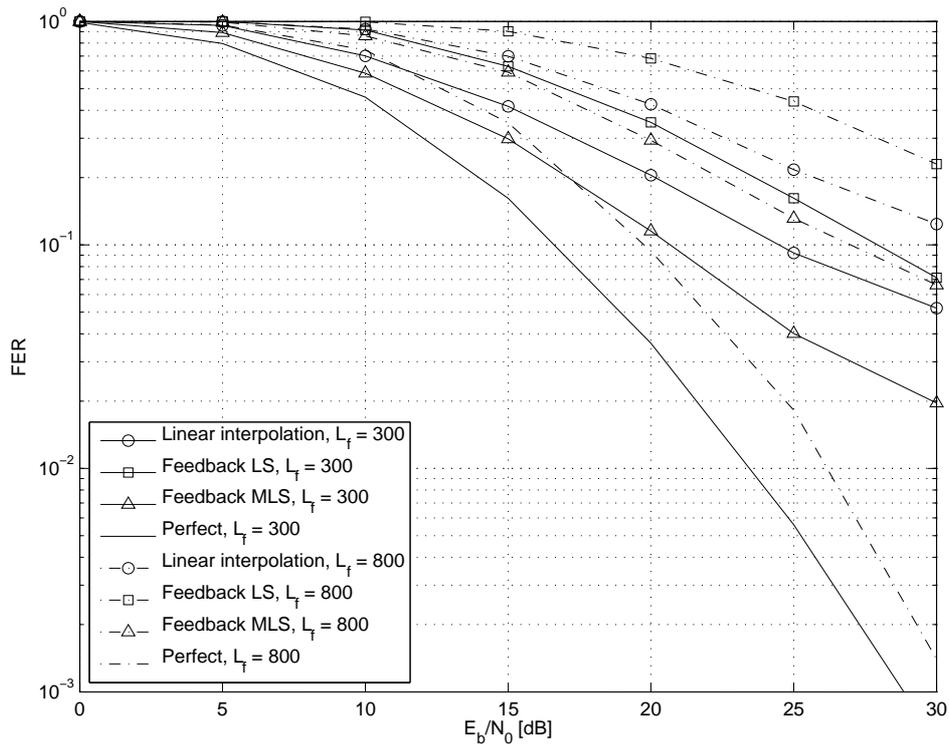


Figure 15: LOS, variable frame length.

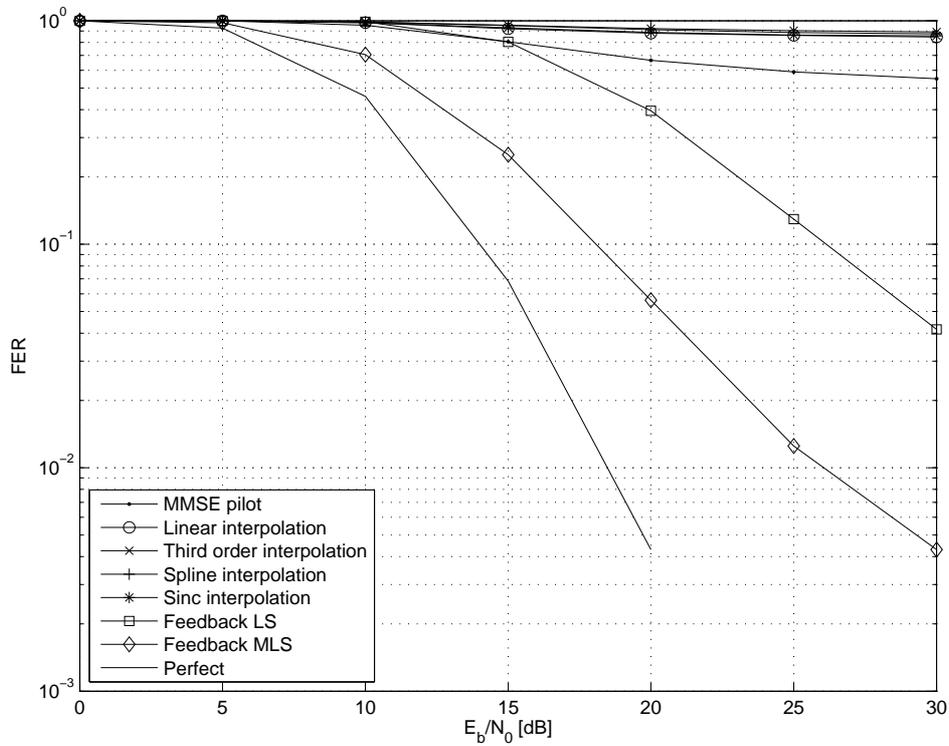


Figure 16: NLOS, Base Case.

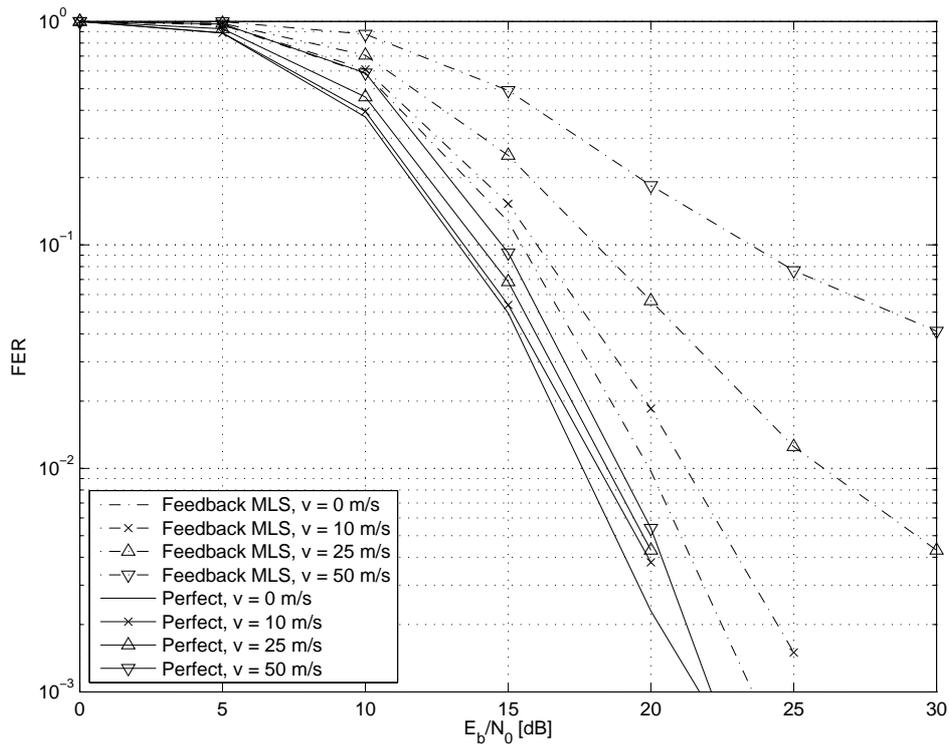


Figure 17: NLOS, variable relative speed.

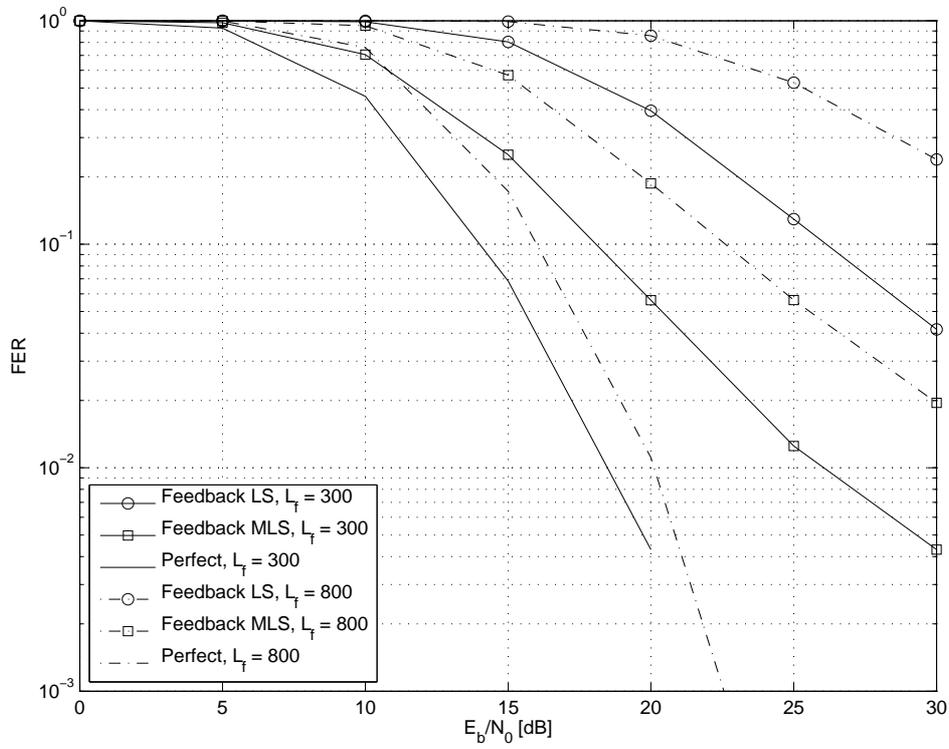


Figure 18: NLOS, variable frame length

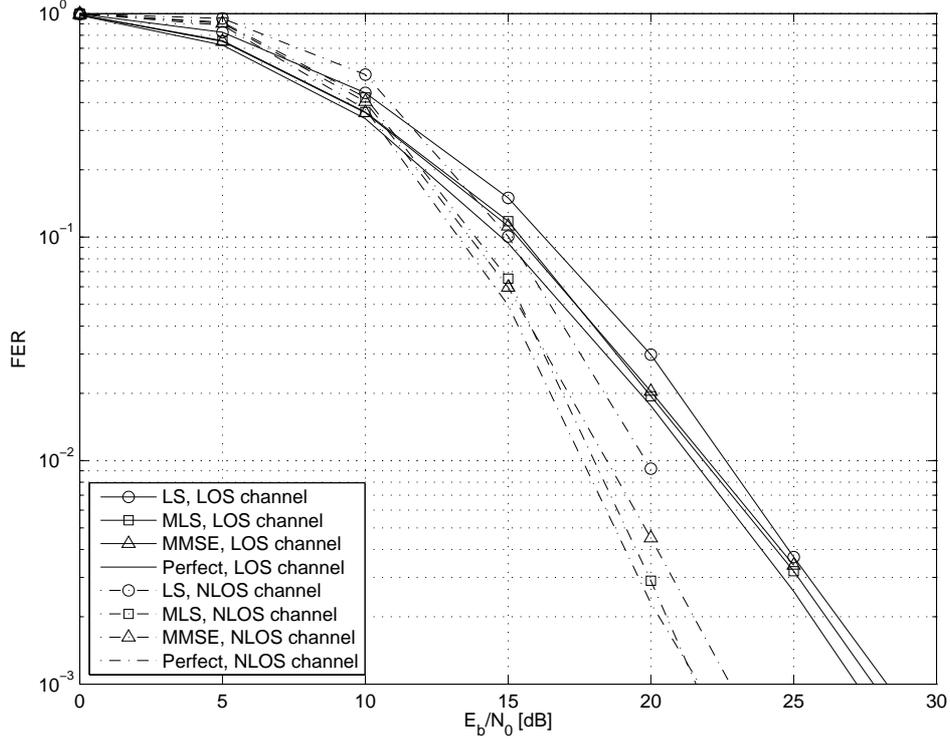


Figure 19: LOS vs NLOS, LS, MLS, MMSE.

6.2.7 LOS vs NLOS

The two channel models are investigated in Scenario 5. As seen in previous plots the interpolation estimators suffer from the frequency variations in the Vehicular NLOS channel model, however, the other estimators combined with the channel coding and interleaving seem to make use of the extra diversity. The LS, MLS and MMSE estimators are shown in Figure 19 and the feedback estimators are shown in Figure 20.

6.2.8 NLOS, variable N_{iter}

In Scenario 6 the Feedback-parameter N_{iter} is varied. From Figure 21 one can observe that increasing the number of iterations N_{iter} for $v = 25$ m/s leads to a lower FER. But for $v = 0$ m/s increasing the number of iterations N_{iter} actually leads to a slightly higher FER.

6.2.9 NLOS, variable N_{prev} , w_e

In Scenario 7 the Feedback-parameters N_{prev} and w_e are varied. From Figure 22 one can observe that increasing the parameter N_{prev} leads to a lower FER. This is to be expected since $v = 0$ m/s and the channel is constant over the whole OFDM frame. The use of weighting method seems to have little effect on the FER. Figure 23 one can observe that increasing the parameter N_{prev} leads to a higher FER. Since $v = 25$ m/s the channel is varying over the whole OFDM frame (and also over several OFDM symbols) and using too many previous channel estimates leads to a higher FER. The value $N_{prev} = 1$ give the lowest FER. The weighting method $w_e = 1$ seems to give slightly better performance than using the weighting method $w_e = 0$.

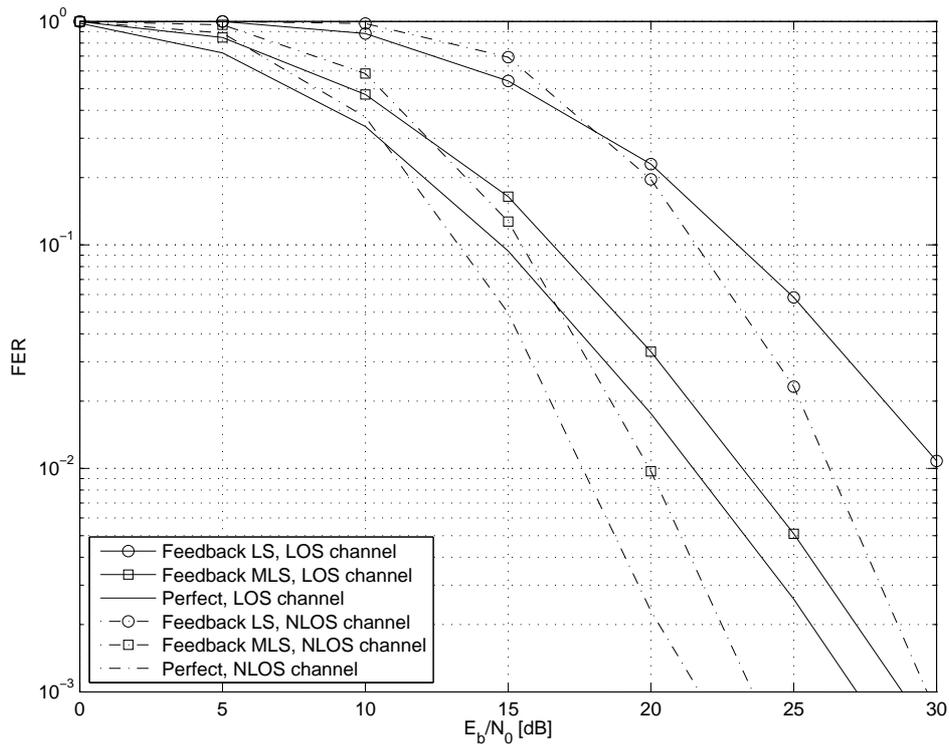


Figure 20: LOS vs NLOS, feedback.

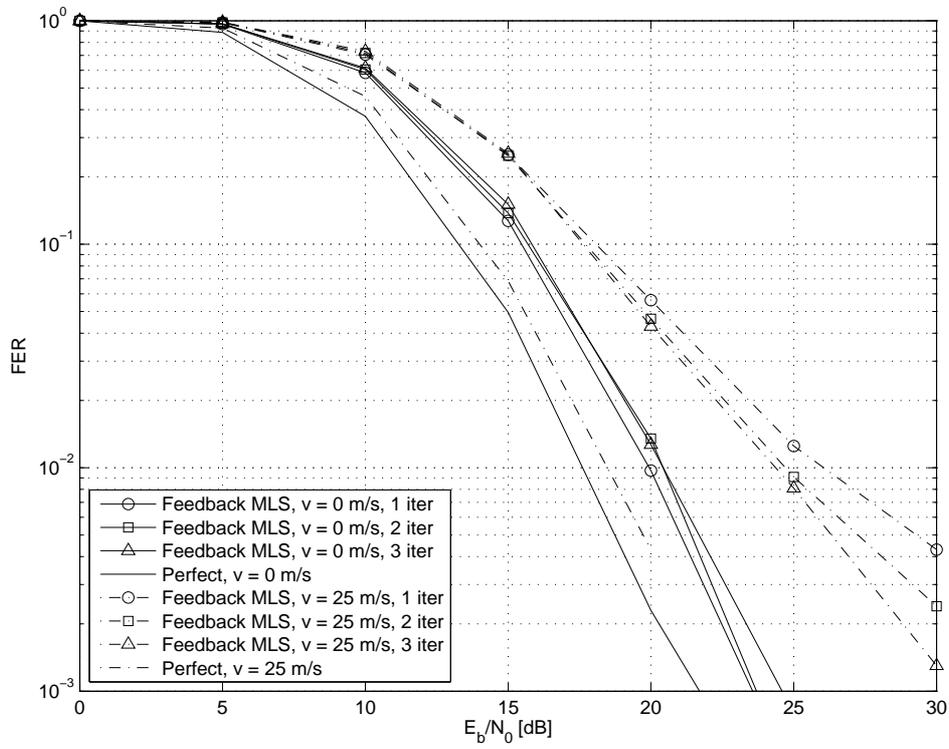


Figure 21: NLOS, variable N_{iter} .

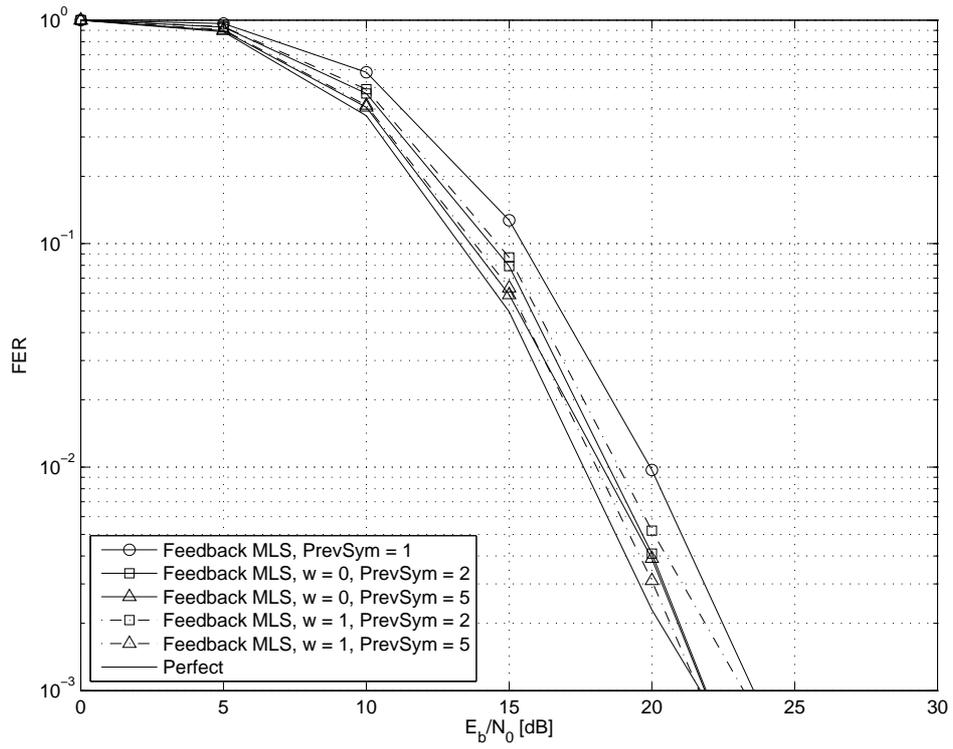


Figure 22: NLOS, variable N_{prev} , w_e , $v = 0$ m/s.

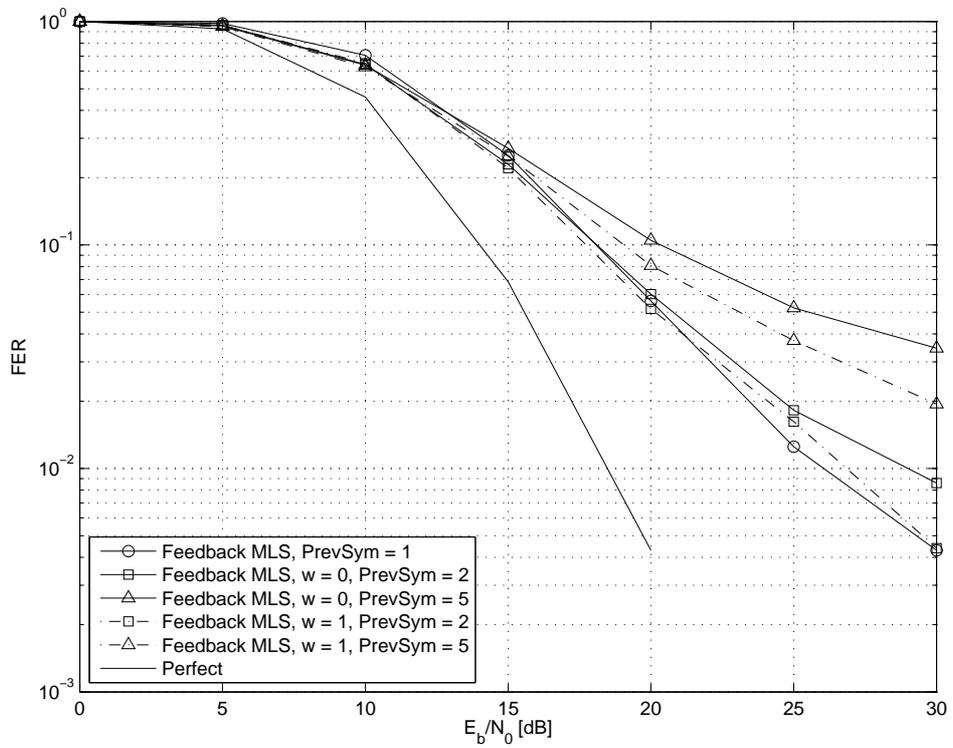


Figure 23: NLOS, variable N_{prev} , w_e , $v = 25$ m/s.

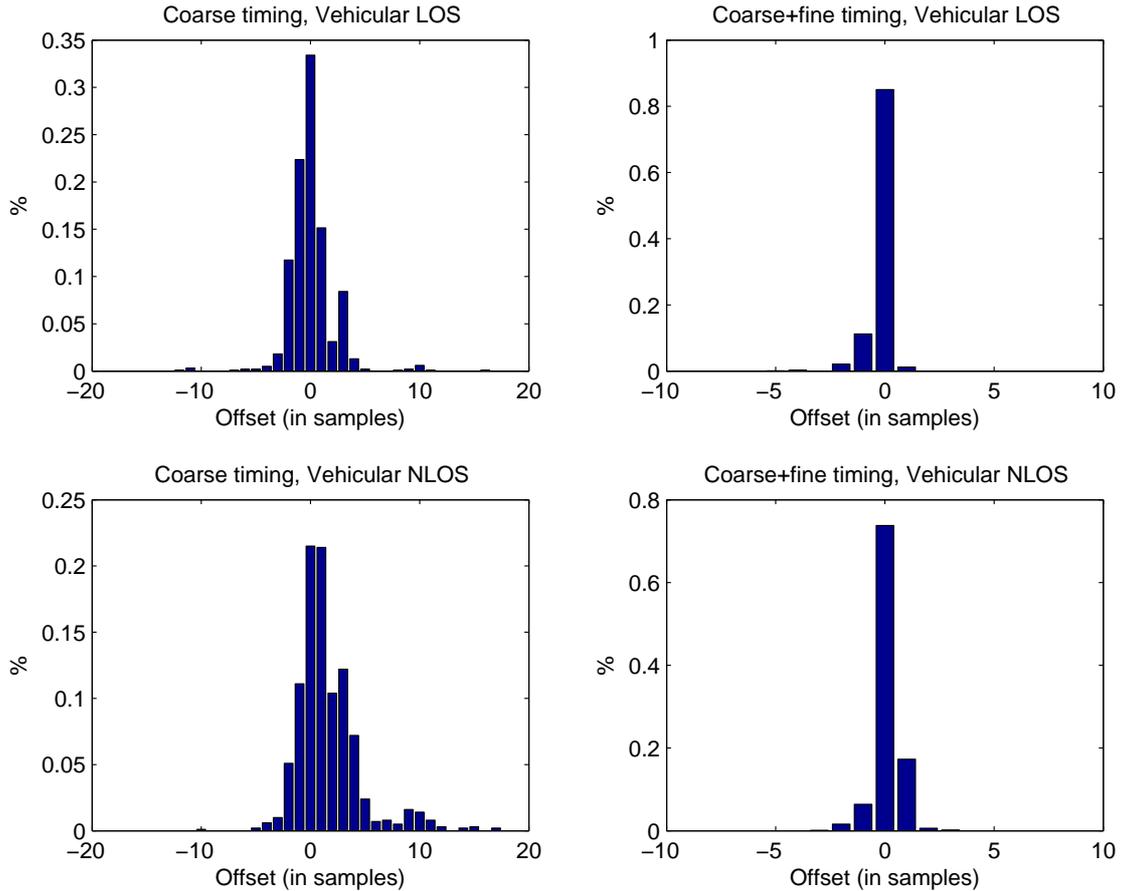


Figure 24: Symbol timing algorithm evaluated utilizing the Vehicular LOS and Vehicular NLOS channel models. The parameters were set to $E_b/N_0 = 15$ dB and $v = 25$ m/s.

6.3 Symbol timing simulations

The symbol timing algorithm was evaluated with the same parameters as in LOS, Base Case and NLOS, Base case keeping $E_b/N_0 = 15$ dB fixed by simulating 1000 times.

As can be seen in Figure 24 the coarse timing algorithm alone does not perform very well. Combining the coarse and fine timing algorithm yields much better results as expected.

6.4 Complexity

The time complexity of each algorithm is measured by running the simulator and recording the amount of time spent estimating the channel and equalizing. The parameters used are those in the LOS, Base Case. Figure 25 shows the time spent on each estimator. The numbers were obtained by running the simulator 1000 times and noting the mean.

This method of measuring the complexity might have some drawbacks, e.g. the results are highly dependent on the specific implementation and optimization potential might not be consistent for all estimators. The results may also depend on the specific test platform used. Overall, due to the coarse granularity of the measurement method, there is little value in comparing estimators with similar numbers but some tendencies can be observed. The MMSE pilot estimator is the most complex estimator and as the number of iterations increase the Feedback MLS estimator complexity increases significantly. In Figure 25 the

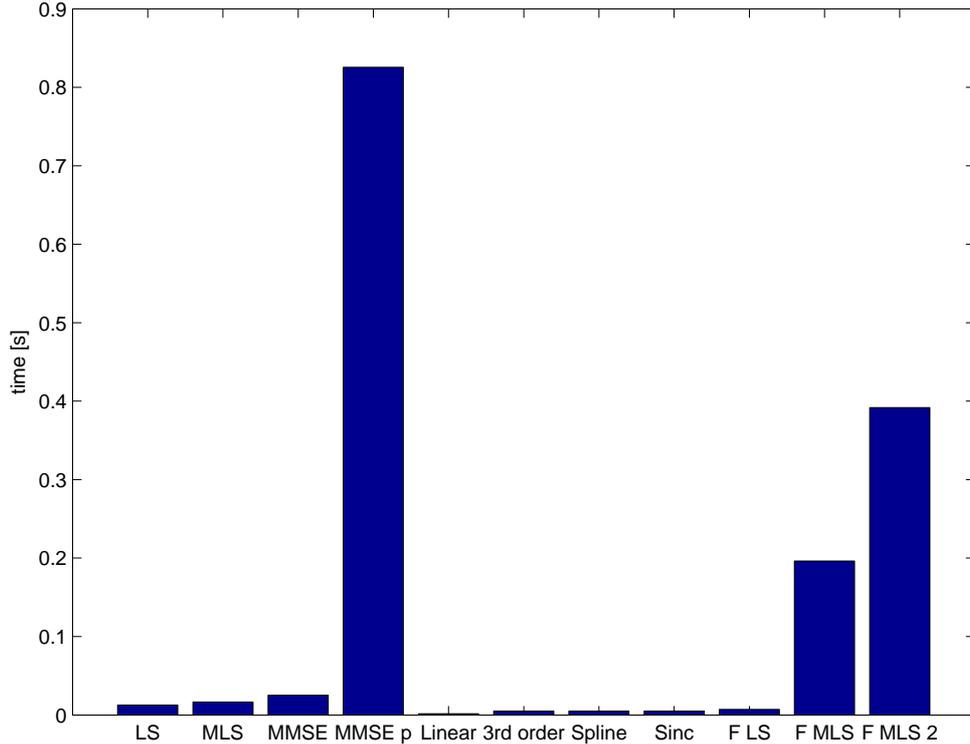


Figure 25: Complexity analysis.

Feedback MLS with $N_{iter} = 1$ is denoted F MLS and with $N_{iter} = 2$ is denoted F MLS 2. It is also important to mention that the MMSE and MMSE pilot estimator need some information of the channel beforehand which has not been taken into account in this complexity measure.

7 Conclusions and Future Work

In this thesis a number of channel estimation methods have been implemented and evaluated in an existing 802.11p simulator. From the results it can be concluded that conventional channel estimation methods (interpolation methods, LS, MLS, MMSE) do not perform adequately in vehicular environments because of the relative speed between sender and receiver and the delay spread induced by the channel. The Feedback estimator seems to be a more promising choice in vehicular environments, but at the cost of an increase in complexity. The interpolation estimators are fairly robust to relative speed changes, while the Feedback estimator is more sensitive to relative speed changes. The interpolation methods are also slightly more robust as the frame length is increased, while the Feedback estimator is slightly less robust as the frame length is increased. It should also be mentioned that neither of the estimators can perform adequately when the given scenario involve long frame lengths combined with high relative speeds.

In the future we suggest the following improvements to be made to the simulator. The estimators can be optimized to allow a more fair complexity comparison. In the current implementation the channel estimators have not been implemented in such a way that optimality can be claimed with respect to complexity. Note also that the simulator can be improved to support not just IEEE 802.11p, specifically it would be interesting to see how MIMO technology would affect the performance. Furthermore, in the current

implementation of the simulator the received OFDM frame is processed OFDM symbol-wise. This can be changed such that the OFDM frame is processed as time samples, this would also give the opportunity to implement a new kind of estimators. Finally, more channel models can be implemented.

References

- [1] PhysLayerSim, 802.11 Physical Layer Simulator. Retrieved at 2010-04-01 from <http://sourceforge.net/projects/physlayersim>.
- [2] Telecommunications and information exchange between systems– Local and metropolitan area networks– Specific requirements–Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Standard 802.11-2007.
- [3] Telecommunications and information exchange between systems– Local and metropolitan area networks– Specific requirements–Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Amendment 6: Wireless Access in Vehicular Environments, IEEE Standard 802.11p-2010.
- [4] Pei Chen and Hisashi Kobayashi. Maximum Likelihood Channel Estimation and Signal Detection for OFDM Systems. In *Proc. IEEE Int. Conf. Commun.*, volume 3, pages 1640–1645, 2002.
- [5] Jan-Jaap van de Beek, Ove Edfors, Magnus Sandell, Sarah Kate Wilson, and Per Ola Börjesson. On Channel Estimation in OFDM Systems. *Proc. IEEE 45th Vehicular Technology Conf.*, pages 815–819, 1995.
- [6] Ren GuangLiang, Zhang HuiNing, and Chang YiLin. SNR estimation algorithm based on the preamble for wireless OFDM systems in Frequency Selective Channels. In *IEEE Transactions on Communications*, volume 57, pages 2230–2234, 2009.
- [7] Saeed V. Vaseghi. *Advanced Digital Signal Processing and Noise Reduction, Second Edition*. John Wiley & Sons Ltd, 2000.
- [8] Richard H. Bartels, John C. Beatty, and Brian A. Barsky. *An introduction to Splines for use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann Publishers, Inc, 1987.
- [9] Thomas Schanze. Sinc Interpolation of Discrete Periodic Signals. *IEEE Trans. Signal Processing*, 43(6):1502–1503, 1995.
- [10] K Wang, M Faulkner, J Singh, and I Tolochko. Timing Synchronization for 802.11a WLANs under Multipath Channels. In *Australian Telecommunications, Networks and Applications Conference*, 2003.
- [11] Guillermo Acosta-Marum and Mary Ann Ingram. Six Time- and Frequency-Selective Empirical Channel Models for Vehicular Wireless LANs. In *IEEE Vehicular Technology Conference*, pages 2134–2138, 2009.
- [12] ETSI EP BRAN WG3 PHY, "Criteria for Comparison." Report 30701F, May 1998.

- [13] ITPP, C++ signal processing library. Available at <http://sourceforge.net/projects/itpp>.
- [14] C. Skupin, F. Hofmann, A. Garcia, and J. Nuckelt. Adaptive channel estimation for VANETs. In *7th International Workshop on Intelligent Transportation (WIT 2010)*, Hamburg,, pages 91–96, March 2010.