Master Thesis

Adaptive Channel Estimation Techniques for Massive MIMO Systems

적응 신호처리 기법을 이용한 Massive MIMO 시스템에서의 채널 추정 기법

By

Chung Jinjoo

February 2015

Department of Electrical Engineering and Computer Science

College of Engineering

Seoul National University

적응 신호처리 기법을 이용한 Massive MIMO 시스템에서의 채널 추정 기법

Adaptive Channel Estimation Techniques for Massive MIMO Systems

지도 교수 이 정 우

이 논문을 공학석사 학위논문으로 제출함

2015 년 2 월

서울대학교 대학원

전기 컴퓨터 공학부

정 진 주

정진주의 공학석사 학위논문을 인준함

2015 년 2 월

위 원 장 ______________________

부위원장 ______________________

위 원 ______________________

Abstract

Wireless communication systems have been required to transmit a large volume of data more rapidly, and thus wireless communication systems require the utilization of more efficient bandwidth and larger channel capacity. Recently, extending multiple input multiple output (MIMO) system on a very large scale by using a myriad number of antennas at the base station was introduced. This method was so that the frequency efficiency can be greatly increased. Massive MIMO system can maximize frequency efficiency by applying a precoding scheme using the downlink channel state information (CSI) to the transmission data at the base station through the usage of large number of antennas. Time-division duplex (TDD) systems have been mainly studied as they can easily obtain the CSI by using the channel reciprocity between uplink and downlink. In frequency-division duplex (FDD) system, the computational complexity of the downlink channel estimation is proportional to the number of antennas at the base station as the channel reciprocity cannot be used. Therefore, effective channel estimation techniques may have to be studied. In this thesis, novel channel estimation algorithms using some of the adaptive techniques are proposed in a channel model with temporal and spatial correlations. The Kalman filter, known as the optimal channel estimation technique, is impossible to estimate the real-time channel due to matrix operations. When consecutive training signals are transmitted, we proposed time division operation of Kalman filter and normalized least mean square (nLMS) filter to enable channel estimation in real-time. Furthermore, we propose decision feedback nLMS filter which updates the CSI by using correctly decoded data as a training signal during data transmission period. With this approach, the performance can be greatly improved without much increase of the hardware complexity compared to the conventional nLMS filter. Simulation results show the performance of proposed algorithms compared to conventional algorithms in terms of mean square error (MSE) and bit error rate (BER).

Key words: massive MIMO, adaptive channel estimation, Kalman, nLMS, decision feedback

Student Number: 2013–20882

Contents

List of Figures

List of Tables

Chapter 1.

Introduction

Wireless communication systems have been required to transmit a large volume of data more rapidly due to the explosive growth of different types of radio communication devices in recent years. To meet such a high demand, wireless communication systems need to use bandwidth more efficiently with larger channel capacity. Since a multiple input multiple output (MIMO) system can increase the reliability and the channel capacity without additional frequency allocation or transmission power increment compared to a single input single output (SISO) system, this has been widely studied in the recent years and continues to be applied to many physical devices [1], [2]. Over the past few years, extending the MIMO system on a very large scale by using a myriad number of antennas at the base station was introduced [3], [4]. Such system is called a massive MIMO or a large-scale MIMO.

Massive MIMO system has emerged as a key technology for the 5G mobile communication system by placing hundreds of antennas at the base station to increase the channel capacity and to reduce the transmit power drastically.

Conventional MIMO systems typically transmit signals to the entire service area. However, massive MIMO system may selectively pass a signal to the limited service area by sending a signal with a very narrow beam. As the number of antennas at the base station increases, the beam width can be narrower. Therefore, it is possible to further increase the channel capacity as well as use very low power at the base station.

To reap the aforementioned gain, the base station needs to apply a precoding scheme which relies on estimated channel state information (CSI). Considering that the size of the channel matrix is determined according to the number of antennas of the transmitter and the receiver, the size of the channel matrix that is to be estimated becomes very large in the massive MIMO system due to the hundreds of antennas deployed at the base station. In order to solve the problem of estimating the channel matrix, previous studies have mainly considered time-division duplex (TDD) systems where the base station can easily obtain the CSI by using the channel reciprocity between uplink and downlink [3]–[5]. For a TDD system, the overhead of the training signal for channel estimation is proportional to the number of users because the CSI of the downlink can be obtained over the uplink channel estimation. If the number of antennas at the base station is assumed to be far larger than the total number of users, the downlink channel estimation via uplink can reduce the overhead of the training signals than the direct downlink channel estimation.

In frequency-division duplex (FDD) systems, the base station can obtain the estimated downlink CSI at user devices through uplink channel feedback. The overhead of training signals for channel estimation of the downlink in FDD systems which cannot use the channel reciprocity is proportional to the number of antennas at the base station. This overhead will disturb the full increase in spectral efficiency expected by introducing massive MIMO systems. However, since the currently used mobile communication systems are mostly based on FDD, the effective channel estimation techniques are needed for the expansion of the massive MIMO system.

The channel estimation techniques can roughly be categorized into two classes – adaptive and non-adaptive schemes. The non-adaptive channel estimation schemes are appropriate to use in the case of independent channel environment, and there are some algorithms such as least square (LS) or minimum mean square error (MMSE) [6]. When the channel is correlated temporally and spatially, the performance of channel estimation can be improved through adaptive schemes. Kalman filter, least mean square (LMS), and recursive least square (RLS) algorithms are corresponding to this adaptive schemes [7].

In this thesis, novel channel estimation algorithms using some adaptive techniques are proposed in a channel model with temporal and spatial correlations. When the channel has such characteristics, the optimal channel estimation scheme is known to use a Kalman filter [8]. However, because the Kalman filter has to perform matrix operations in proportion to the size of the channel matrix, the computational complexity is large, and thus the processing delay of channel estimation is increased. Two types of algorithms are proposed to address this impractical issue. First, we propose a time division operation of Kalman filter and normalized LMS (nLMS) filter. Using this technique to efficiently handle the processing delay occurring in the Kalman filter, it is possible to estimate the channel in real-time even though the size of hardware is increases slightly compared to the Kalman filter as nLMS filter is added. Because the computational complexity of nLMS filter is small compared to that of Kalman filter, the increased portion of the hardware size is very low. Second, decision feedback algorithm with nLMS filter is proposed to improve the performance of channel estimation during the data transmission period. Typically, it is difficult to estimate the channel since the random data is transmitted during the data transmission period. However, it is possible to use correctly decoded data as a training signal for additional channel estimation. Computational complexity can be reduced with decision feedback nLMS scheme because it does not contain any matrix operations, and the performance is improved compared to the traditional nLMS algorithms. The performance of proposed channel estimation algorithm is compared with conventional channel estimation algorithms by the simulations.

The rest of the thesis is organized as follows. In Chapter 2, the system model, background, and the precoding techniques of massive MIMO systems are introduced. Chapter 3 presents the conventional adaptive and non-adaptive channel estimation techniques. Chapter 4 describes the proposed adaptive channel estimation techniques. Simulation results are provided in Chapter 5, followed by conclusions in Chapter 6.

Chapter 2.

Massive MIMO Systems

In this chapter, the background of this thesis, the principles and characteristics of the massive MIMO system will be introduced.

The massive MIMO system is designed to apply the advantages of the conventional MIMO system in a much larger scale. Hundreds of antennas are installed at the base station to support dozens of users at the same time and frequency resources. With this large-scale scheme, some benefits for wireless communication can be obtained [9]. First, since the number of data streams can be increased as the number of base station antennas increases, the user devices can be supported more. Second, because the extra antennas can form the transmit beam pattern to a specific direction, the energy efficiency is improved. Third, the transmitted beam has a direction, so that it is possible to prevent the transmission in the direction of that the interference should not occur.

However, there are also some limiting factors of massive MIMO systems [9]. For TDD operation, channel reciprocity is assumed between uplink and downlink channels. This assumption may seem like a reasonable consensus, but since the practical hardware chains between the base station and user devices may not be reciprocal, the calibration of the hardware chains is required in order to use the channel reciprocity. For FDD operation, channel reciprocity cannot be used because of the carrier frequency difference. Therefore, to obtain the downlink CSI at the base station, user devices need to feedback the estimated CSI to the base station. In this case, the training signal overhead for channel estimation at the user device is proportional to the number of antennas at the base station. As the number of base station antennas increases, the spectral efficiency that can be obtained by using massive MIMO system is reduced.

2.1. Overview

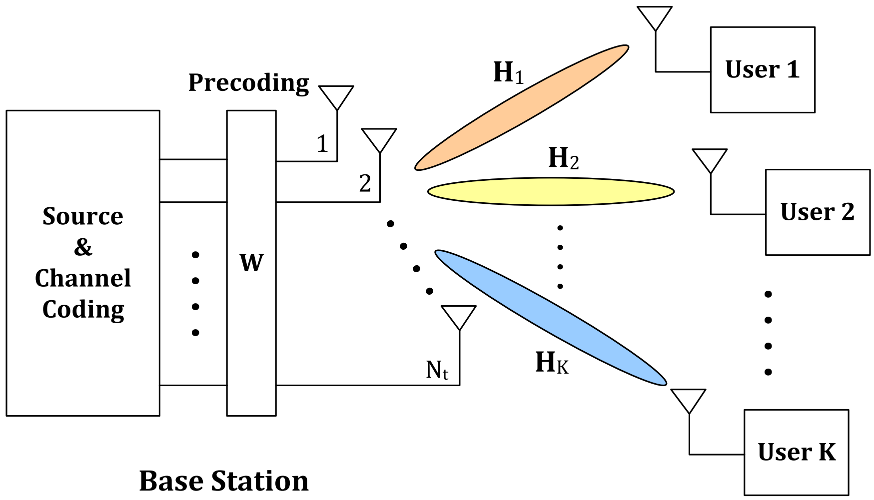

Fig. 2.1 Massive MIMO System Model

The massive MIMO system model is shown in Fig. 2.1. The base station has N_t transmit antennas and each user has N_r receive antennas. The received signal at user N is given by

$$\mathbf{y}_N = \mathbf{H}_N^H \mathbf{x}_N + \mathbf{n}_N \quad , \tag{2.1}$$

where $\mathbf{y}_N$ is an $N_r \times 1$ received signal vector, $\mathbf{x}_N$ is an $N_t \times 1$ transmitted signal vector, $\mathbf{H}_N$ is an $N_r \times N_t$ channel matrix, $\mathbf{n}_N$ is an $N_r \times 1$ independent and identically distributed (i.i.d.) complex Gaussian noise vector with zero mean and unit variance, and superscript H denotes the Hermitian transpose. Total transmit power is unity, i.e. $\mathrm{E}\{||\mathbf{x}||^2\} = 1$.

As shown in Fig. 2.1, the transmitted beam has to be narrow in order to maximize the efficiency obtained by the massive MIMO system. The precoding scheme is one of the solutions of the narrow beam forming, and thus the downlink channel information is needed at the base station to generate the precoding matrix.

2.2. Achievable rate

2.2.1. Point-to-Point MIMO

With i.i.d. complex Gaussian inputs and under the assumption that the receiver has perfect knowledge of the channel matrix, the mutual information of the point-to-point MIMO system is

$$C = I(\mathbf{y};\mathbf{x}) = \log_2 \det\left(\mathbf{I}_{N_r} + \frac{\rho}{N_t}\mathbf{H}\mathbf{H}^H\right) \quad , \tag{2.2}$$

where $I(\mathbf{y};\mathbf{x})$ denotes the mutual information, $\mathbf{I}_{Nr}$ denotes an $N_r \times N_r$ identity matrix, and ρ is the SNR of the link [4]. $\mathbf{H}_N$ for convenience is referred to as $\mathbf{H}$.

In order to express the achievable rate (2.2) in terms of the singular values, singular value decomposition (SVD) of the channel matrix $\mathbf{H}$ needs to be operated which is

$$\mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^H \quad . \tag{2.3}$$

(2.3) shows the SVD of $\mathbf{H}$, where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices of dimension $N_r \times N_r$ and $N_t \times N_t$ respectively, and $\mathbf{\Lambda}$ is an $N_r \times N_t$ diagonal matrix whose diagonal elements are the singular values, $\{\lambda_1, \lambda_2, \cdots, \lambda_{\min(Nt,Nr)}\}$. With (2.3), the achievable rate (2.2) can be expressed as

$$C = \sum_{i=1}^{\min(N_t,N_r)} \log_2\left(1 + \frac{\rho\lambda_i^2}{N_t}\right), \tag{2.4}$$

with constraint

$$\sum_{i=1}^{\min(N_t,N_r)} \lambda_i^2 = \mathrm{Tr}\left(\mathbf{H}\mathbf{H}^H\right) \quad , \tag{2.5}$$

where "Tr" denotes trace operation. (2.4) is equivalent to the achievable rate by a combination of parallel links where the l-th link has an SNR of $\rho\lambda_l^2/N_t$ [4]. The bound of the achievable rate (2.4) is

$$\log_2\left(1+\frac{\rho\mathrm{Tr}(\mathbf{HH}^H)}{N_t}\right) \le C \le \min(N_t,\ N_r)\cdot\log_2\left(1+\frac{\rho\mathrm{Tr}(\mathbf{HH}^H)}{N_t\min(N_t,N_r)}\right). \tag{2.6}$$

The lower bound is when all but one of the singular values are equal to zero, that is rank-1, and the upper bound is when all of the min(N_t, N_r) singular values are equal. The rank-1 (worst) case occurs under the Line-of-Sight (LOS) circumstances and the equal singular value (best) case is accomplished when the entries of the channel matrix are i.i.d. random variables [4]. Since normalization is generally carried out for practical purposes, the total magnitude of a channel matrix is typically equal to one. By using this, Tr($\mathbf{HH}^H$) is possible to approximate to the N_tN_r, and (2.6**)** can be rewritten as

$$\log_2(1+\rho N_r) \le C \le \min(N_t,N_r)\cdot\log_2\left(1+\frac{\rho\max(N_t,N_r)}{N_t}\right). \tag{2.7}$$

2.2.2. Multi-User MIMO

In this section, we only consider the achievable rate of the downlink that is from the base station to the user devices. We assume TDD operation using channel reciprocity property that the downlink channel matrix is the transpose of the uplink channel matrix. The channel matrix in (2.1) is the product of an $N_t \times N_r$ matrix $\mathbf{G}$ and an $N_r \times N_r$ diagonal matrix $\mathbf{D}_\beta$ with elements β which account for small scale fading and large scale fading, respectively, and that is

$$\mathbf{H} = \mathbf{G}\mathbf{D}_{\beta}^{1/2} \quad . \tag{2.8}$$

Since the downlink channel matrix can be obtained from the uplink channel matrix, $\mathbf{H}$ stands for the uplink channel matrix and the downlink channel matrix is $\mathbf{H}^H$. Under the condition of $N_t \gg N_r$ and the favorable propagation[1] [10], the column vectors of the channel matrix $\mathbf{H}$ are orthogonal, so that

$$\begin{aligned} \left(\frac{\mathbf{H}^H\mathbf{H}}{N_t}\right)_{N_t \gg N_r} &= \mathbf{D}_{\beta}^{1/2}\left(\frac{\mathbf{G}^H\mathbf{G}}{N_t}\right)_{N_t \gg N_r}\mathbf{D}_{\beta}^{1/2} \\ &\approx \mathbf{D}_{\beta} \quad . \end{aligned} \tag{2.9}$$

The collection of received signal at the users is

$$\mathbf{y} = \mathbf{H}^H\mathbf{x} + \mathbf{n} \quad , \tag{2.10}$$

where $\mathbf{y}$ is an $N_r \times 1$ received vector, $\mathbf{x}$ is an $N_t \times 1$ transmitted vector with unit power, and $\mathbf{n}$ is an $N_r \times 1$ noise vector whose elements are i.i.d. complex Gaussian noise vector with zero mean unit variance. Using the known result of capacity for this channel, the sum-capacity of downlink with constraint is [4]

$$C_{\text{sum}} = \max_{\{\gamma_k\}} \log_2 \det\left(\mathbf{I}_{N_t} + \rho\mathbf{H}\mathbf{D}_{\gamma}\mathbf{H}^H\right)$$

[1] When all fading coefficients are i.i.d. and zero mean, this is said to be favorable propagation.

$$\text{subject to } \sum_{k=1}^{N_t} \gamma_k = 1, \quad \gamma_k \geq 0, \quad \forall k, \tag{2.11}$$

where ρ is the SNR of the downlink, and $\mathbf{D}_\gamma$ is an $N_r \times N_r$ diagonal matrix whose diagonal elements are large scale fading coefficients γ. With (2.9), the asymptotic form of the sum capacity is

$$\begin{aligned} C_{\text{sum } N_t \gg N_r} &= \max_{\{\gamma_k\}} \log_2 \det\left(\mathbf{I}_{N_k} + \rho \mathbf{D}_\gamma^{1/2} \mathbf{H}^H \mathbf{H} \mathbf{D}_\gamma^{1/2}\right) \\ &\approx \max_{\{\gamma_k\}} \log_2 \det\left(\mathbf{I}_{N_k} + N_t \rho \mathbf{D}_\gamma \mathbf{D}_\beta\right) \\ &= \max_{\{\gamma_k\}} \sum_{k=1}^{N_t} \log_2\left(1 + N_t \rho \gamma_k \beta_k\right), \end{aligned} \tag{2.12}$$

where β is the diagonal element of $\mathbf{D}_\beta$ in (2.8), and γ_k can be used to allocate power to different users at the base station. The columns of the channel matrix $\mathbf{H}$ are asymptotically orthogonal as the number of transmit antennas grows, and then the result of the sum capacity can be represented as (2.12).

2.3. Zero-forcing Precoding Techniques

In massive MIMO systems, precoding scheme is needed to improve the performance of the overall system. An intuitive approach to generate precoding matrix is to invert the channel matrix, which is referred to as zero-forcing (ZF) precoding [11], [12]. Transmitted signal is multiplied by the inverse of the channel matrix, and as a result the signal is received through equalized channel.

The ZF precoding matrix is generated with the pseudo-inverse of channel matrix **H** and can thus be written as

$$\mathbf{H}^{\dagger} = \mathbf{H}(\mathbf{H}^{H}\mathbf{H})^{-1} \ , \tag{2.13}$$

where superscript † denotes pseudo-inverse. With this precoding matrix, the transmitted signal is

$$\mathbf{x} = \mathbf{W}\mathbf{s} = \mathbf{H}^{H}(\mathbf{H}\mathbf{H}^{H})^{-1}\mathbf{P}^{1/2}\mathbf{s} \ , \tag{2.14}$$

where **W** is a precoding matrix, **s** is an $N_t \times 1$ data symbol vector, and **P** is an $N_t \times N_t$ diagonal matrix which is used for power normalization. The diagonal matrix **P** is defined as

$$\mathbf{P} = \begin{bmatrix} p_1 & 0 & \cdots & 0 \\ 0 & p_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & p_{N_t} \end{bmatrix} , \tag{2.15}$$

where p_k is

$$p_k = \frac{1}{\left[\left(\mathbf{H}\mathbf{H}^{H}\right)^{-1}\right]_{kk}} \quad . \tag{2.16}$$

$[(\mathbf{H}\mathbf{H}^{H})^{-1}]_{kk}$ denotes the (k,k)-th element of the matrix $(\mathbf{H}\mathbf{H}^{H})^{-1}$. Recall that to guarantee the existence of pseudo-inverse matrix of $\mathbf{H}$, the number of columns must be larger than the number of rows, i.e. $N_t \geq N_r$. This means that the number of transmit antennas is larger than the number of receive antennas and is also a very reasonable condition for massive MIMO systems.

The received signal (2.1) using ZF precoding at the transmitter can be rewritten as

$$\mathbf{y} = \mathbf{HWs} + \mathbf{n} = \mathbf{HH}^H(\mathbf{HH}^H)^{-1}\mathbf{Ws} + \mathbf{n} = \mathbf{Ws} + \mathbf{n} \ , \tag{2.17}$$

Since $\mathbf{W}$ is a diagonal matrix, the channel can be decomposed into N_r parallel channels.

Chapter 3.

Channel Estimation Schemes

In this chapter, several channel estimation schemes which are adaptive and non-adaptive schemes will be reviewed. The obtained CSI through channel estimation is used to compensate the distorted channel for the coherent detection at the receiver, and also at the same time it is used for precoding at the transmitter. The more accurate the channel estimation result is the more correct symbol detection is possible, because the channel compensation capability at the receiver can be increased or more elaborate precoding matrix at the transmitter can be generated.

One of the most popular and widely used approaches to the MIMO channel estimation is to employ pilot symbols (also referred to as training sequences), and then to estimate the channel based on the received data and the knowledge of pilot symbols [13]. The channel estimation scheme using this kind of method has already been widely studied. Conventional researches are based on the different types of cost functions, however, we classified channel estimation schemes into two categories which are adaptive scheme and non-adaptive scheme. These two categories are used in the different channel environments. For adaptive channel estimation, there are some algorithms called LMS, RLS, and Kalman filter. These schemes automatically adapts to time-varying channels and are usually used when the channel is correlated temporally or spatially or both. On the other hand, if the channel is independent and does not need any of the statistics of the channel model, the non-adaptive channel estimation scheme is generally used. The LS algorithm will be introduced for one of these methods.

3.1. Adaptive Channel Estimation

3.1.1. Least Mean Square (LMS) Algorithm

We now start with the Wiener filter. The optimal tap-weight vector in Fig. 3.1 is

$$\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p} \quad , \tag{3.1}$$

where $\mathbf{R}$ denotes an $M \times M$ correlation matrix of the tap inputs $x(n)$, $x(n-1)$, … , $x(n-M+1)$; that is $\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^H(n)]$, where $\mathbf{x}(n)$ is a transmitted signal vector referred to as $[x(n), x(n-1), \dots, x(n-M+1)]^T$ in (2.1). Correspondingly, $\mathbf{p}$ denotes an $M \times 1$ cross-correlation vector between the tap inputs of the filter and the desired response $y(n)$; that is $\mathbf{p} = E[\mathbf{x}(n)y^*(n)]$.

By using $e(n)$ which is the estimation error in Fig. 3.1, we can derive the cost function J. Refer to Fig. 3.1, the estimation error $e(n)$ is

$$e(n) = y(n) - \sum_{k=0}^{M-1} w_k^* x(n-k) \quad , \tag{3.2}$$

where $y(n)$ is the desired response which is equivalent to the received signal in (2.1), and w_k is the filter coefficients.

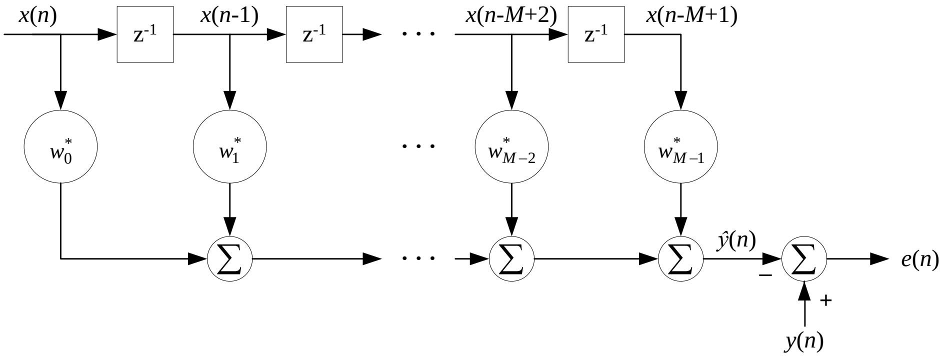

Fig. 3.1 Transversal Filter

Accordingly, the cost function can be written as

$$\begin{aligned} J &= E[e(n)e^*(n)] \\ &= \sigma_y^2 - \sum_{k=0}^{M-1} w_k^* p(-k) + \sum_{k=0}^{M-1} w_k p^*(-k) + \sum_{k=0}^{M-1}\sum_{i=0}^{M-1} w_k^* w_i r(i-k) \quad , \end{aligned} \tag{3.3}$$

where $\sigma_y{}^2$ is the variance of the desired response $y(n)$, and $r(i-k) = E[x(n-k)x^*(n-i)]$. By substituting (3.2) into $e(n)$, we can get the second line of (3.3). We may rewrite this equation in matrix form by using the definitions of **R** and **p** given in (3.1), and it is shown as

$$J(\mathbf{w}) = \sigma_y^2 - \mathbf{w}^H\mathbf{p} - \mathbf{p}^H\mathbf{w} + \mathbf{w}^H\mathbf{R}\mathbf{w} \quad , \tag{3.4}$$

where the mean square error (MSE) is written as $J(\mathbf{w})$ to emphasize its dependence on the tap-weight vector $\mathbf{w}$.

The LMS algorithm is an important member of the family of stochastic gradient algorithms, which uses a deterministic gradient in a recursive computation of the Wiener filter for stochastic inputs [7]. This algorithm is very popular for practical implementations because it does not require measurements of any covariance matrices or matrix inversion for its low computational complexity. Since the stochastic gradient is a recursive way of updating filter weights, its actual implementation is represented by a feedback system. Starting from an arbitrary initial value of the filter weight vector, the solution improves with the increased number of iterations. Under the appropriate conditions, the solution converges to the Wiener solution [7].

Because the cost function $J(\mathbf{w})$ is reduced after each iteration, the condition $J(\mathbf{w}(n)) > J(\mathbf{w}(n+1))$ must be satisfied. Where $\mathbf{w}(n)$ is the old value and $\mathbf{w}(n+1)$ is its updated value. With this time index n, the MSE or cost function (3.4) can be rewritten as

$$J(n) = \sigma_y^2 - \mathbf{w}^H(n)\mathbf{p} - \mathbf{p}^H\mathbf{w}(n) + \mathbf{w}^H(n)\mathbf{R}\mathbf{w}(n) \quad . \tag{3.5}$$

$J(\mathbf{w}(n))$ for convenience is referred to as $J(n)$.

In a simple form of iterative descent known as the method of steepest descent, the successive adjustments applied to the weight vector $\mathbf{w}$ are in the direction of steepest descent – that is, in a direction opposite to the gradient vector of the cost function $J(n)$, which is denoted by $\nabla J(n)$ [7]. The gradient vector $\nabla J(n)$ is

$$\nabla J(n) = \partial J(n)/\partial \mathbf{w}(n) = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}(n) \quad , \tag{3.6}$$

As shown in (3.6), for an exact measurement of the gradient vector, the correlation matrix $\mathbf{R}$ of the tap inputs and the cross-correlation vector $\mathbf{p}$ between the tap inputs and the desired response are needed, however, these conditions are not practical. To overcome this problem, the simplest choice is to use instantaneous estimates for $\mathbf{R}$ and $\mathbf{p}$, which are defined by

$$\begin{aligned} \widehat{\mathbf{R}}(n) &= \mathbf{x}(n)\mathbf{x}^H(n) \quad , \\ \widehat{\mathbf{p}} &= \mathbf{x}(n)y^*(n) \quad . \end{aligned} \tag{3.7}$$

Then, the instantaneous estimate of the gradient vector becomes

$$\widehat{\nabla} J(n) = -2\mathbf{x}(n)y^*(n) + 2\mathbf{x}(n)\mathbf{x}^H(n)\widehat{\mathbf{w}}(n) \quad . \tag{3.8}$$

Recall that the steepest descent method operates with the direction opposite to the gradient vector to find the minimum value of the cost function $J(n)$. Therefore, by using this method, the updated tap-weight vector is

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{1}{2}\mu\big(\nabla J(n)\big) \quad , \tag{3.9}$$

where μ is a positive constant called step-size parameter, and the factor 1/2 is introduced for mathematical convenience. Substituting the estimate of (3.8) for the gradient vector $\nabla J(n)$ in (3.9), the new recursive relation for updated tap-weight vector is

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu\mathbf{x}(n)[y^*(n) - \mathbf{x}^H(n)\hat{\mathbf{w}}(n)] \quad . \tag{3.10}$$

The formula inside the square bracket is equivalent to the estimation error $e^*(n)$.

The block diagram of LMS algorithm is described in Fig. 2.1

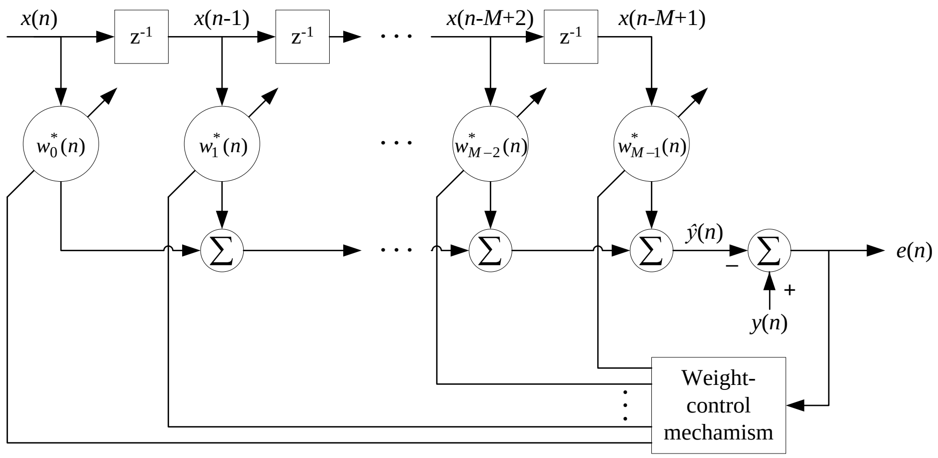

Fig. 3.2 Adaptive Transversal Filter

The summary of LMS algorithm is as follows:

Step 1: Filter output

$$\hat{y}(n) = \hat{\mathbf{w}}^H(n)\mathbf{x}(n)$$

Step 2: Estimation error

$$e(n) = y(n) - \hat{y}(n)$$

Step 3: Tap-weight update

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu e^*(n)\mathbf{x}(n)$$

One variation of LMS algorithm is normalized LMS (nLMS) algorithm. Even if the average power of the transmitted signal $\mathbf{x}(n)$ is the same, the instantaneous power can be different. Since $\mathbf{x}(n)$ is included in the update section as shown in (3.10), the update rate depends on the magnitude of the transmitted signal. By taking this into consideration, nLMS algorithm is introduced which normalizes the update section with the Euclidean norm of the transmitted signal $\mathbf{x}(n)$. The difference between LMS and nLMS algorithm is the step 3 (tap–weight update) of the summary of LMS algorithm, and can be replaced as

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu e^*(n)\mathbf{x}(n)/\|\mathbf{x}(n)\|^2 \quad , \tag{3.11}$$

where $\|\cdot\|^2$ stands for the Euclidean norm.

3.1.2. Kalman Filter

Theoretically the Kalman filter is an estimator for what is called the linear-quadratic problem, which is the problem of estimating the instantaneous state of a linear dynamic system perturbed by white noise – by using measurements linearly related to the state but corrupted by white noise [14]. The Kalman filter is known to be an optimal estimator in the sense of all noise is a Gaussian, and in this case the Kalman filter minimizes the MSE of the estimated parameters. However, if the noise is not Gaussian, non-linear estimators may be better. A distinctive feature of the Kalman filter is that its mathematical formulation is described in terms of state-space model. Another novel feature of the Kalman filter is that its solution is computed recursively [7].

The algorithm consists of two steps: prediction step and update step. In the prediction step, the Kalman filter generates an estimate of the current state variables by using the state estimated from the previous time. Although the predicted state estimate is the estimated state at the current time, it is known as *a priori* state because they do not contain observation information on the current time. In the update step, the current *a priori* prediction is combined with current observation information to refine the state estimate. This updated estimate is referred to as *a posteriori* state estimate.

The state, denoted by $\mathbf{w}(n)$, comprises the fewest data on the past behavior of the system that are needed to predict its future behavior. To estimate $\mathbf{w}(n)$, the observation $\mathbf{y}(n)$ is used. With state–space model, the process for next state is

$$\mathbf{w}(n+1) = \mathbf{F}\big(n+1,\, n\big)\mathbf{w}(n) + \boldsymbol{\nu}_1(n) \;\;, \tag{3.12}$$

and the observation is

$$\mathbf{y}(n) = \mathbf{C}(n)\mathbf{w}(n) + \boldsymbol{\nu}_2(n) \;\;, \tag{3.13}$$

where $\mathbf{F}(n+1,\, n)$ is the state transition model, $\mathbf{C}(n)$ is the observation model which maps the true state space into the observed space, $\mathbf{v}_1(n)$ is white noise with correlation matrix $\mathbf{Q}_1(n)$, and $\mathbf{v}_2(n)$ is process noise with zero mean and correlation matrix $\mathbf{Q}_2(n)$. In general, the state transition model $\mathbf{F}(n+1, n)$ can be a constant if the system is stationary. Our final goal is to find $\mathbf{w}(n)$ with using recursive way like RLS algorithm.

Once we get the observation, it is possible to obtain the MMSE estimate of $\hat{\mathbf{w}}(\mathrm{n}+1)$ and $\hat{\mathbf{y}}(n)$. The notation $\hat{\mathbf{w}}(n|n-1)$ is the MMSE estimate of $\mathbf{w}(n)$ given $\mathbf{y}(0)$, $\mathbf{y}(1)$, . . . , $\mathbf{y}(n-1)$, and $\hat{\mathbf{y}}(n|n-1)$ can be similarly defined. At update step, *a posteriori* (updated) state estimate is

$$\hat{\mathbf{w}}(n+1|n) = \mathbf{F}(n+1, n)\hat{\mathbf{w}}(n|n-1) + \mathbf{K}(n)\boldsymbol{\alpha}(n) \ , \tag{3.14}$$

where $\mathbf{K}(n)$ is the Kalman gain matrix, and $\boldsymbol{\alpha}(n)$ is named innovation process which is an added information by the incoming new observation to the past accumulated information. (3.14) can be obtained from the RLS algorithm which can be expressed as $\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mathbf{k}(n)\xi(n)$, where $\mathbf{k}(n)$ is the gain vector and $\xi(n)$ is a priori estimation error. Detailed explanation of (3.14) will be skipped. $\boldsymbol{\alpha}(n)$ is also referred to as one-step prediction error defined by

$$\boldsymbol{\alpha}(n) = \mathbf{y}(n) - \hat{\mathbf{y}}(n|n-1) \ . \tag{3.15}$$

The MMSE estimate of $\mathbf{y}(n)$ of the observation vector using (3.13) is

$$\hat{\mathbf{y}}(n|n-1) = \mathbf{C}(n)\hat{\mathbf{w}}(n|n-1) + \mathbf{v}_2(n|n-1)$$

$$= \mathbf{C}(n)\hat{\mathbf{w}}(n|n-1) \ . \tag{3.16}$$

The second row of (3.16) is because the noise vector is orthogonal to the past observations. Using (3.13), (3.15) and (3.16), the innovation process can be expressed as

$$\begin{aligned} \boldsymbol{\alpha}(n) &= \mathbf{C}(n)\mathbf{w}(n) + \mathbf{v}_2(n) - \mathbf{C}(n)\hat{\mathbf{w}}(n|n-1) \\ &= \mathbf{C}(n)[\mathbf{w}(n) - \hat{\mathbf{w}}(n|n-1)] + \mathbf{v}_2(n) \\ &= \mathbf{C}(n)\boldsymbol{\varepsilon}(n|n-1) + \mathbf{v}_2(n) \ , \end{aligned} \tag{3.17}$$

where $\boldsymbol{\varepsilon}(n|n-1)$ is the predicted state error vector at time n, using data up to time $n-1$. That is, $\boldsymbol{\varepsilon}(n|n-1)$ is the difference between the state $\mathbf{w}(n)$ and the one-step prediction $\hat{\mathbf{w}}(n+1)$. The correlation matrix of the innovation process $\boldsymbol{\alpha}(n)$ is

$$\mathbf{R}_{\boldsymbol{\alpha}} = E[\boldsymbol{\alpha}(n)\boldsymbol{\alpha}^H(n)] \ . \tag{3.18}$$

By substituting (3.17) into (3.18), $\mathbf{R}_{\boldsymbol{\alpha}}$ can be rewritten as

$$\begin{aligned} \mathbf{R}_{\boldsymbol{\alpha}} &= \mathbf{C}(n)E[\boldsymbol{\varepsilon}(n|n-1)\boldsymbol{\varepsilon}^H(n|n-1)]\mathbf{C}^H(n) + \mathbf{Q}_2(n) \\ &= \mathbf{C}(n)\mathbf{P}(n|n-1)\mathbf{C}^H(n) + \mathbf{Q}_2(n) \ , \end{aligned} \tag{3.19}$$

where $\mathbf{P}(n|n-1)$ is the predicted state error correlation matrix and is used as the statistical description of the error in the predicted estimate $\hat{\mathbf{w}}(n|n-1)$.

Define the Kalman gain matrix $\mathbf{K}(n)$ as

$$
\begin{aligned}
\mathbf{K}(n) &= E[\mathbf{w}(n+1)\boldsymbol{\alpha}^H(n)]\mathbf{R}_{\boldsymbol{\alpha}}^{-1}(n) \\
&\overset{(a)}{=} \mathbf{F}(n+1,n)\mathrm{E}[\mathbf{w}(n)\boldsymbol{\varepsilon}^H(n|n-1)\mathbf{C}^H(n)]\mathbf{R}_{\boldsymbol{\alpha}}^{-1}(n) \\
&\overset{(b)}{=} \mathbf{F}(n+1,n)E[\boldsymbol{\varepsilon}(n|n-1)\boldsymbol{\varepsilon}^H(n|n-1)]\mathbf{C}^H(n)\mathbf{R}_{\boldsymbol{\alpha}}^{-1}(n) \\
&\overset{(c)}{=} \mathbf{F}(n+1,\, n)\mathbf{P}(n|n-1)\mathbf{C}^H(n)\mathbf{R}_{\boldsymbol{\alpha}}^{-1}(n) \ . \qquad (3.20)
\end{aligned}
$$

(a) is established by substituting (3.12) and (3.17) into the first row of (3.20), and using the orthogonality between $\mathbf{w}(n)$ and $\mathbf{v}_2(n)$. In a similar way, by using the definition of $\boldsymbol{\varepsilon}(n|n-1)$ and the orthogonality between $\hat{\mathbf{w}}(n|n-1)$ and $\boldsymbol{\varepsilon}(n|n-1)$, (b) can be obtained. Finally, (c) can be easily established by the definition of $\mathbf{P}(n|n-1)$.

Next, what do we need to update the error correlation matrix $\mathbf{P}(n|n-1)$? By using the definition of $\mathbf{P}(n+1|n)$ and $\boldsymbol{\varepsilon}(n|n-1)$, it can be rewritten as

$$
\begin{aligned}
\mathbf{P}(n+1|n) &= \mathbf{F}(n+1,n)\mathbf{w}(n) + \mathbf{v}_1(n) - \mathbf{F}(n+1,n)\hat{\mathbf{w}}(n|n-1) - \mathbf{K}(n)\boldsymbol{\alpha}(n) \\
&= \mathbf{F}(n+1,n)\mathbf{P}(n)\mathbf{F}^H(n+1,n) + \mathbf{Q}_1(n) \ . \qquad (3.21)
\end{aligned}
$$

The second row is obtained by using the Riccati equation that is not discussed in this thesis. The new matrix $\mathbf{P}(n)$ is the correlation matrix of the error in $\hat{\mathbf{w}}(n|n)$ and can be described by the recursion

$$
\mathbf{P}(n) = \mathbf{P}(n|n-1) - \mathbf{F}(n,n+1)\mathbf{K}(n)\mathbf{C}(n)\mathbf{P}(n|n-1) \ , \qquad (3.22)
$$

where $\mathbf{F}(n, n+1)$ is the inverse of $\mathbf{F}(n+1, n)$.

The summary of Kalman filter is as follows:

Step 1: Kalman gain

$$\mathbf{K}(n) = \mathbf{F}(n+1,n)\mathbf{P}(n|n-1)\mathbf{C}^H(n)[\mathbf{C}(n)\mathbf{P}(n|n-1)\mathbf{C}^H(n) + \mathbf{Q}_2(n)]^{-1}$$

Step 2: Predicted estimate of the state

$$\boldsymbol{\alpha}(n) = \mathbf{y}(n) - \mathbf{C}(n)\hat{\mathbf{w}}(n|n-1)$$

$$\hat{\mathbf{w}}(n+1|n) = \mathbf{F}(n+1,n)\hat{\mathbf{w}}(n|n-1) + \mathbf{K}(n)\boldsymbol{\alpha}(n)$$

Step 3: Error correlation matrix

$$\mathbf{P}(n) = \mathbf{P}(n|n-1) - \mathbf{F}(n,n+1)\mathbf{K}(n)\mathbf{C}(n)\mathbf{P}(n|n-1)$$

$$\mathbf{P}(n+1|n) = \mathbf{F}(n+1,n)\mathbf{P}(n)\mathbf{F}^H(n+1,n) + \mathbf{Q}_1(n)$$

Initial Condition:

$$\hat{\mathbf{w}}(1|0) = E[\mathbf{w}(1)]$$

$$\mathbf{P}(1|0) = E[(\mathbf{w}(1) - E[\mathbf{w}(1)])(\mathbf{w}(1) - E[\mathbf{w}(1)])^H]$$

3.2. Non-Adaptive Channel Estimation

3.2.1. Least Square (LS) Algorithm

The method of LS is a standard approach to the approximate solution of the overdetermined[2] system, and the goal of the LS estimator is to minimize the square distance between the received signal and the original signal [15]. In other words, the object of the LS algorithm is to find the filter coefficients such that the

[2] A system of linear equations is called overdetermined if there are more equations than unknowns.

squared norm of the error **e** is minimized. The error **e** can be defined as

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\mathbf{w} \ , \tag{3.23}$$

where **y** is a received signal in (2.1), **X** is a transmitted signal matrix which is defined in (3.24), and **w** is the filter coefficients.

$$\begin{aligned} \mathbf{X}^H &= [\mathbf{x}(M), \quad \mathbf{x}(M+1), \quad \cdots, \quad \mathbf{x}(N)] \\ &= \begin{bmatrix} x(M) & x(M+1) & \cdots & x(N) \\ x(M-1) & x(M) & \cdots & x(N-1) \\ \vdots & \vdots & \cdots & \vdots \\ x(1) & x(2) & \cdots & x(N-M+1) \end{bmatrix} . \end{aligned} \tag{3.24}$$

Then, the cost function *J* can be expressed as

$$\begin{aligned} J &= (\mathbf{y} - \mathbf{X}\mathbf{w})^H(\mathbf{y} - \mathbf{X}\mathbf{w}) \\ &= \mathbf{y}^H\mathbf{y} - \mathbf{w}^H\mathbf{X}^H\mathbf{y} - \mathbf{y}^H\mathbf{X}\mathbf{w} + \mathbf{w}^H\mathbf{X}^H\mathbf{X}\mathbf{w} \ . \end{aligned} \tag{3.25}$$

To find the minimum value of the cost function *J*, derivative of *J* for **w** will be performed and the solution for **w** is

$$\mathbf{w} = (\mathbf{X}^H\mathbf{X})^{-1}\mathbf{X}^H\mathbf{y} \ . \tag{3.26}$$

If the number of unknown variables is same as the number of equations, the exact solution for **w** will then be expressed as $\mathbf{w} = \mathbf{X}^{-1}\mathbf{y}$. However, the existence of the exact solution does not occur with high probability in many cases. Therefore, if it is unable to find the solution of **w** which satisfies $\mathbf{y} = \mathbf{X}\mathbf{w}$, the solution of **w** has to be approximated by minimizing the distance between $\mathbf{y}$ and $\mathbf{Xw}$.

In practice, the transmitted matrix $\mathbf{X}$ and the received vector $\mathbf{y}$ cannot be the entire data stream due to the implement limitation, so $\mathbf{X}$ and $\mathbf{y}$ are configured to have only a few recent data. $\mathbf{X}$ and $\mathbf{y}$ are in the case of which may have the entire data stream, the solution (3.26) converges to Wiener filter (3.1). This is because $E[\mathbf{X}^H\mathbf{X}]$ is getting closer to the correlation matrix of $\mathbf{X}$ as the number of samples in $\mathbf{X}$ increases. In the same way, $E[\mathbf{X}^H\mathbf{y}]$ becomes the cross-correlation of $\mathbf{X}$ and $\mathbf{y}$. Combining previously mentioned conditions together, $E[\mathbf{w}]$ becomes $\mathbf{R}^{-1}\mathbf{p}$ which is the Wiener filter solution.

Chapter 4.

Practical Channel Estimation for Massive MIMO Systems

In this chapter, adaptive channel estimation techniques using Kalman filter and nLMS filter are proposed for efficient channel estimation in massive MIMO systems. Two types of channel estimation schemes are proposed, one is time division operation of Kalman filter and nLMS filter, and the other is decision feedback nLMS filter.

4.1. System Model

In this proposed algorithm we consider a massive MIMO system with N_t transmit antennas and a single receive antenna (single user), referred to as a multiple input single output (MISO) system. Also we assume that the number of transmit antennas is much larger than the total number of receive antennas, i.e. $N_t \gg 1$. We mainly consider about the FDD case, because the downlink channel information can be easily obtained by uplink channel estimation in TDD systems due to the channel reciprocity. With this system model, the transmitter and the receiver are the base station and user devices.

The received signal at the symbol time k is

$$y_k = \mathbf{h}_k^H \mathbf{x}_k + n_k, \qquad k = 1, 2, \ldots, \tag{4.1}$$

where $\mathbf{x}_k$ is an $N_t \times 1$ transmit symbol vector at time k, $\mathbf{h}_k$ is an $N_t \times 1$ MISO channel vector at time k, and n_k is zero-mean i.i.d. complex Gaussian noise at time k with covariance σ_n^2. The channel is time-varying and Rayleigh-faded under a state-space model, and is correlated temporally and spatially. The temporal correlation is given by the first-order stationary Gauss-Markov process [16]

$$\mathbf{h}_{k+1} = a\mathbf{h}_k + \sqrt{1-a^2}\mathbf{b}_k \ , \tag{4.2}$$

where a is temporal fading correlation coefficients, and $\mathbf{b}_k$ is a zero-mean and temporally independent Gaussian vector. Under Jake's model, $a = J_0(2\pi f_D T_s)$, where $J_0(\cdot)$ is the zero-order Bessel function, f_D is the Doppler frequency shift, and T_s is the transmit symbol interval. The spatial correlation is defined as

$$\mathbf{R}_\mathbf{h} = E[\mathbf{h}_k \mathbf{h}_k^H] \ , \tag{4.3}$$

where $\mathbf{R}_\mathbf{h}$ is the covariance matrix of the channel vector $\mathbf{h}_k$, satisfies $\mathbf{R}_\mathbf{h} = \mathbf{R}_\mathbf{b} = E[\mathbf{b}_k \mathbf{b}_k^H]$ for all k. We assume that the transmitter and the receiver both know the channel covariance matrix $\mathbf{R}_\mathbf{h}$.

The training and data symbols are transmitted on a block-by-block basis. Each block is composed of a total of M consecutive symbols with M_p training symbols and M_d data symbols, that is $M = M_p + M_d$ [16]. This kind of block type transmission is described in Fig. 4.1. In this figure, for example, the total number of transmitted symbols in one block is ten, which is composed of four pilot symbols and six data symbols. During training periods, pilot symbols, that are predetermined signals between transmitter and receiver, are transmitted. When the receiver receives pilot symbols, it performs channel estimation. The estimated channel is feedback to the transmitter by uplink transmission and is used to generate transmit precoding vectors. During data transmission periods, unknown data is transmitted by using these precoding vectors. We assume that the feedback CSI is unlimited, that it is not quantized information.

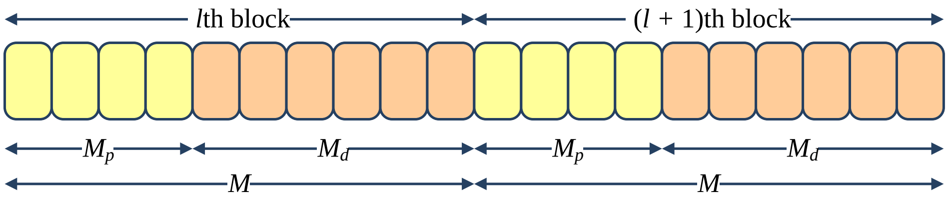

Fig. 4.1 Training and Data Symbol Block Configuration

4.2. Practical Channel Estimation Techniques

4.2.1. Hybrid Channel Estimation

The received signal (4.1) can be rewritten as

$$y_k = \mathbf{x}_k^H \mathbf{h}_k + n_k \ , \tag{4.4}$$

Then, (4.2) and (4.4) form a state-space model as described in (3.12) and (3.13) and the optimal channel estimation scheme is given by Kalman filtering [8]. By using the summary of the Kalman filter as described in the last part of section 3.1.2, the overall channel estimation scheme for (4.4) is as follows:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{x}_k\left(\mathbf{x}_k^H \mathbf{P}_{k|k-1}\mathbf{x}_k + \sigma_n^2\right)^{-1} \ , \tag{4.5}$$

$$\hat{\mathbf{h}}_{k|k} = \hat{\mathbf{h}}_{k|k-1} + \mathbf{K}_k\left(y_k - \mathbf{x}_k^H \hat{\mathbf{h}}_{k|k-1}\right) \ , \tag{4.6}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{x}_k^H \mathbf{P}_{k|k-1} \ , \tag{4.7}$$

$$\mathbf{P}_{k+1|k} = a^2 \mathbf{P}_{k|k} + (1 - a^2)\mathbf{R}_{\mathbf{h}} \ , \tag{4.8}$$

where $\hat{\mathbf{h}}_{k|k}$ is an $N_t \times 1$ estimated channel, $\mathbf{P}_{k|k}$ and $\mathbf{P}_{k+1|k}$ are $N_t \times N_t$ estimation and prediction error covariance matrices, respectively, and $\mathbf{K}_k$ is the Kalman gain. The state transition model $\mathbf{F}(n+1, n)$ in (3.12) is constant, $\mathbf{w}(n)$ and $\hat{\mathbf{w}}(n)$ in (3.13) and (3.14) are $\mathbf{h}_k$ and $\hat{\mathbf{h}}_k$, respectively, and $\mathbf{C}(n)$ in (3.13) is $\mathbf{x}_k$. The initial values are defined as $\hat{\mathbf{h}}_{1|0} = \mathbf{0}$, and $\mathbf{P}_{1|0} = \mathbf{R}_{\mathbf{h}}$.

As you can see in (4.6), $\hat{\mathbf{h}}_{k|k-1}$ is required to calculate $\hat{\mathbf{h}}_{k|k}$, but the whole process from (4.5) to (4.8) is consists of matrix operations such as **P** and $\mathbf{R_h}$. Since the transmitter (base station) will have hundreds of antennas in massive MIMO systems, the size of these matrices will become tremendously large. If the training symbols are continuously received as shown in Fig. 4.1, the channel estimation of the current training symbol cannot be completed before the next training symbol is received because of large computational complexity due to the matrix operations. For real-time channel estimation, we propose a method using the nLMS filter introduced in (3.11) additionally.

The training symbols can be separated into two parts, the first training symbol in a block and the rest of training symbols in same block. The first channel estimation of each block which is the yellow part of Fig. 4.2 is using the channel estimation result of the previous block. From the second to the last channel estimation of each block, the green part of Fig. 4.2, are using the channel estimation result of the previous training symbols of the same block.

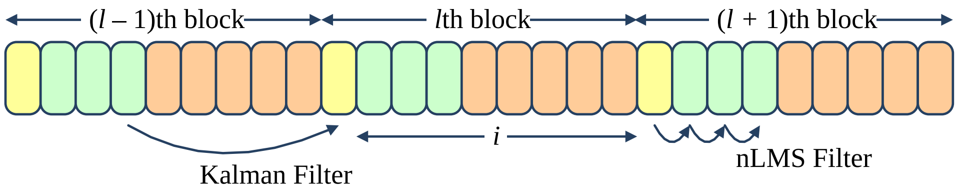

Fig. 4.2 Block Configuration of Hybrid Channel Estimation

The first channel estimation for each block, there is a margin of time that is from the last channel estimation to the data transmission of the previous block and is expressed as the orange part of Fig. 4.2.

In view of this time margin, we propose a novel channel estimation scheme that is using the Kalman filter for the first training symbol and using the nLMS filter for the rest of the training symbols in each block as described in Fig. 4.2. Since the receiver knows which training symbol will be transmitted at the next block, (4.7), (4.8) and with the exception of σ_n^2 in (4.5) can be calculated during the data transmission period of the current block. Then, only vector operations remain when the first training symbol is arrived at the next block. Because the last channel estimation from the second one uses the nLMS filter, the calculation is just the vector operations during the entire channel estimation period and real-time channel estimation is possible. Simple block diagram of this whole process is described in Fig. 4.3.

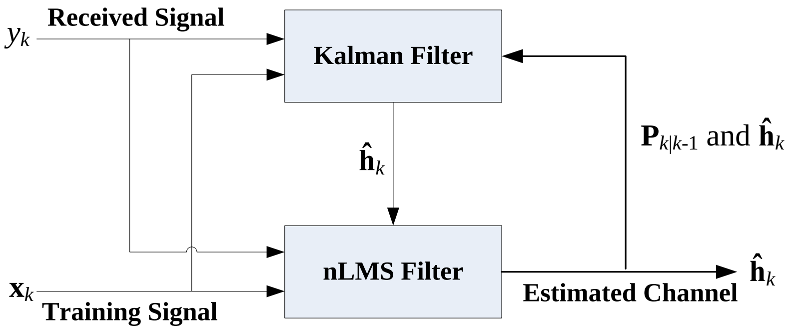

Fig. 4.3 Block Diagram of Hybrid Channel Estimation

After complete channel estimation with nLMS filter, the error covariance matrix $\mathbf{P}_{k|k-1}$ is needed that cannot be obtained from nLMS filter for the calculation of

Kalman filter. Since $\mathbf{P}_{k|k-1}$ has to be predicted without the measurement update step in (3.21), this matrix can be obtained by

$$\mathbf{P}_{k|k-i} = a^{2i}\mathbf{P}_{k-i|k-i} + \left(1 - a^{2i}\right)\mathbf{R}_{\mathbf{h}} \ , \tag{4.9}$$

where *i* is the number of symbols from the Kalman filter calculation of the previous block to that of the current block as described in Fig. 4.2.

The outline of the proposed algorithm is as follows:

Hybrid Kalman – nLMS Channel Estimation Algorithm

Step 1: Initialize

$\hat{\mathbf{h}}_{1|0} = \mathbf{0}$, and $\mathbf{P}_{1|0} = \mathbf{R}_{\mathbf{h}}$

Step 2: Kalman filtering for the first training symbol

$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{x}_k\left(\mathbf{x}_k^H\mathbf{P}_{k|k-1}\mathbf{x}_k + \sigma_n^2\right)^{-1}$

$\hat{\mathbf{h}}_{k|k} = \hat{\mathbf{h}}_{k|k-1} + \mathbf{K}_k\left(y_k - \mathbf{x}_k^H\hat{\mathbf{h}}_{k|k-1}\right)$

$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{x}_k^H\mathbf{P}_{k|k-1}$

Step 3: nLMS filtering for the rest of the training symbols

$\hat{y}_k = \mathbf{x}_k^H\hat{\mathbf{h}}_{k-1}$, where $\hat{\mathbf{h}}_{k-1}$ is $\hat{\mathbf{h}}_{k|k}$ from step 1

$e_k = y_k - \hat{y}_k$

$\hat{\mathbf{h}}_k = \hat{\mathbf{h}}_{k-1} + \mu e_k\mathbf{x}_k/\|\mathbf{x}_k\|^2$

Step 4: Predict error covariance matrix

$\mathbf{P}_{k|k-i} = a^{2i}\mathbf{P}_{k-i|k-i} + \left(1 - a^{2i}\right)\mathbf{R}_{\mathbf{h}}$, where $\mathbf{P}_{k-i|k-i}$ is $\mathbf{P}_{k|k}$ from step 1

When the first training symbol of the next block is received, go to step 2 and repeat the entire process.

4.2.2. Decision Feedback nLMS Channel Estimation

Decision feedback algorithm is a nonlinear filter which uses previous detected symbols as training symbols, in addition to conventional channel estimation filters. Usually this algorithm is used for the equalizer which compensates the distorted signal due to the multipath propagation and Doppler spreading.

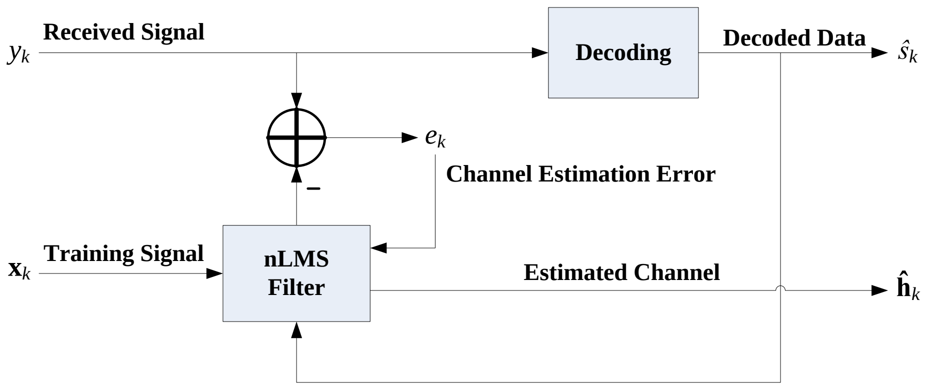

Fig. 4.4 Block Diagram of Decision Feedback nLMS Channel Estimator

The block diagram of channel estimation using decision feedback algorithm is in Fig. 4.4. As shown in this figure, decoded data symbols are used to estimate the channel in the same manner as the training symbols during data receiving period described in Fig. 4.1. Through this process, the estimated channel could be updated again so the precoding vector at the transmitter can also be updated. When decision feedback algorithm is used for equalizing the channel, there is an additional feedback filter that is not described in Fig. 4.4. In the proposed algorithm, an additional feedback filter is not required because the estimated channel using data symbols is used to update the already existing nLMS filter coefficients.

One of the problems that may occur is the propagation of channel estimation error caused by the decision feedback process. The cyclic redundancy check (CRC) code can be used to avoid this problem. CRC code is an error detecting code commonly used in communication or storage systems to detect accidental changes to raw data. This code is popular because it is simple to implement in hardware and particularly good at detecting errors caused by noise. After decoding the received data, it is determined by the CRC code whether the received data is correctly decoded or not. If the decoded data is correct, the data is used for the additional channel estimation using decision feedback algorithm, but if the decoded data is wrong, previously estimated channel is maintained. Accordingly, at the beginning of the channel estimation with the inaccurate CSI, the decision feedback process hardly works.

Another problem is the number of CSI feedback link increases due to the frequently updated channel information. By transmitting the CSI periodically without doing at every symbol, e.g. once in a block, can solve this kind of feedback link overhead problem. Even though the performance will be degraded with this method, the number of feedback link can be reduced. The trade-offs will be shown in chapter 5 with some bit error rate (BER) curves.

The outline of the proposed algorithm is as follows:

Decision Feedback nLMS Channel Estimation Algorithm

Step 1: Initialize

$\hat{\mathbf{h}}_0 = \mathbf{0}$

Step 2: nLMS filtering for training symbols

$\hat{y}_k = \hat{\mathbf{h}}_{k-1}^H \mathbf{x}_k$

$e_k = y_k - \hat{y}_k$

$\hat{\mathbf{h}}_k = \hat{\mathbf{h}}_{k-1} + \mu e_k^* \mathbf{x}_k / \|\mathbf{x}_k\|^2$

Step 3: Received data decoding and check CRC code

If the decoded data $\hat{s}_k$ is correct, multiply with the zero-forcing precoding vector $\mathbf{w}_k$ in (2.13) and generate the transmitted signal.

$\hat{\mathbf{x}}_k = \mathbf{w}_k \hat{s}_k$

Step 4: Decision Feedback nLMS filtering for data symbols

Go to step 2, and substitute $\hat{\mathbf{x}}_k$ into $\mathbf{x}_k$.

4.3. Complexity Evaluation

In this section, we evaluate the computational complexity of the proposed algorithms and the conventional algorithms. The number of multipliers and adders are used to compare the computational complexity.

With N_t transmit antennas and a single receive antenna, the size of the channel matrix that needs to be estimated is $N_t \times 1$, so the size of all vectors in (4.5) – (4.8) and (3.11) are $N_t \times 1$. Also, all matrices in (4.5) – (4.8), which are $\mathbf{R_h}$, $\mathbf{P}$, and $\mathbf{K}$, have a size of $N_t \times N_t$. By taking this into consideration, we will count the number of multipliers and adders. The only difference between nLMS and decision feedback nLMS is the *step 3* of the summary in section 4.2.2.

	Multiplier		**Adder**	
Kalman	$3(N_t^2 + N_t)$	100%	$3N_t^2$	100%
nLMS	$3N_t$	9%	$3N_t - 1$	10%
Hybrid	$3(N_t^2 + 2N_t)$	109%	$3(N_t^2 + N_t) - 1$	110%
DFB nLMS	$4N_t$	12%	$4N_t - 2$	13%

Table 1. Number of Operations for proposed algorithms and convectional algorithms

The number of multipliers and adders are listed in Table 1. Multipliers are actually complex multipliers although they are expressed as multipliers. Multipliers in (4.8) are omitted because they are constant multipliers with predetermined temporal coefficient a. As shown in Table 1, computational complexity of hybrid algorithms is increased about 10% compared to the Kalman filter, and that of decision feedback nLMS is increased about 3% compared to the nLMS filter. Therefore, with slight increase of computational complexity, real time channel estimation and improvement of performance can be achieved.

Chapter 5.

Simulation Results

In this chapter, simulation results are shown to illustrate the performance of the proposed channel estimation algorithms. The MSE and the BER are simulated of the channel realizations by taking the average of 1000 runs.

5.1. Simulation Environments

Since massive MIMO systems assume $N_t \gg 1$, we consider 32 transmit antennas and a single user with a single receive antenna, that is N_t = 32, N_r = 1. Each block is composed of 10 consecutive symbols with 4 training symbols and 6 data symbols, that is M = 10, M_p = 4, M_d = 6. The carrier frequency is 2.5GHz and the symbol duration is 100μs. We adopt two velocities; 3km/h and 130km/h which represent a man walking speed and a transportation movement speed, respectively. The temporal fading correlation coefficients a in (4.2) are 0.9999 and 0.99 with these two velocities. The spatial correlation coefficients of the channel is the quadratic exponential correlation model [16], given by

$$[\mathbf{R_h}]_{i,j} = r^{|i-j|^2} \quad , \tag{5.1}$$

where $|r| < 1$ and normalized so that $\mathrm{tr}(\mathbf{R_h}) = 1$, and the value is set to 0.8. The SNR is set to 20dB for MSE performance evaluation, and binary phase shift keying (BPSK) modulation is used for BER performance evaluation. For nLMS channel estimation scheme, the step-size μ is 0.2.

For training signal generation, we use two types of training signals – orthogonal and eigenvalue training signals. Orthogonal training signal is generated by the classical Gram-Schmidt orthogonalization algorithm which generates an orthonormal basis. Eigen training signal is a scheme which uses M_p dominant eigenvalues of the channel covariance matrix. By performing eigen decomposition of $\mathbf{R_h}$, indices of M_p dominant eigenvalues are picked and column vectors of the unitary matrix of $\mathbf{R_h}$ corresponding to these indices are used as training signals [16].

Zero-forcing precoding technique is used described in section 2.3. Since we only consider a single user with a single receive antenna, the channel matrix becomes a vector accordingly. Therefore, the precoding vector at transmitter becomes

$$\mathbf{w}_k = \frac{\hat{\mathbf{h}}_k}{\left\|\hat{\mathbf{h}}_k\right\|_2} \quad , \tag{5.2}$$

where ||·||2 stands for L2-norm.

5.2. MSE Performance Evaluation

First, we compare the MSE of the proposed channel estimation algorithms with conventional Kalman and nLMS algorithm. The convergence rate and residual error is a measure of the MSE performance evaluation. Whether using orthogonal training sequences or eigenvalue training sequences, the MSE will eventually be converged after passing thousands of symbols. The residual error is similar in any algorithms mentioned above after convergence, so the convergence rate is the only measure of the performance.

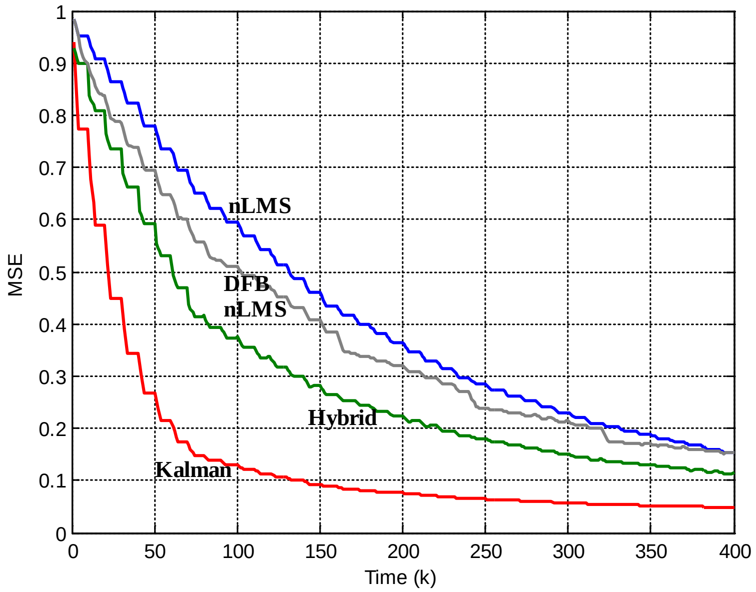

Fig. 5.1 MSE comparison of proposed algorithms with conventional algorithms using orthogonal training sequence and *v*=3*km/h*

The MSE performance of the proposed algorithms with orthogonal training sequences is shown in Fig. 5.1. In terms of MSE, the proposed hybrid algorithm outperforms compared to the conventional nLMS algorithm even if the performance falls off compared to the Kalman filter. The performance of decision feedback nLMS algorithm is slightly better than the performance of conventional nLMS algorithm. However, because the computational complexity of the decision feedback is not substantially higher than in the original algorithm, this small improvement is significant.

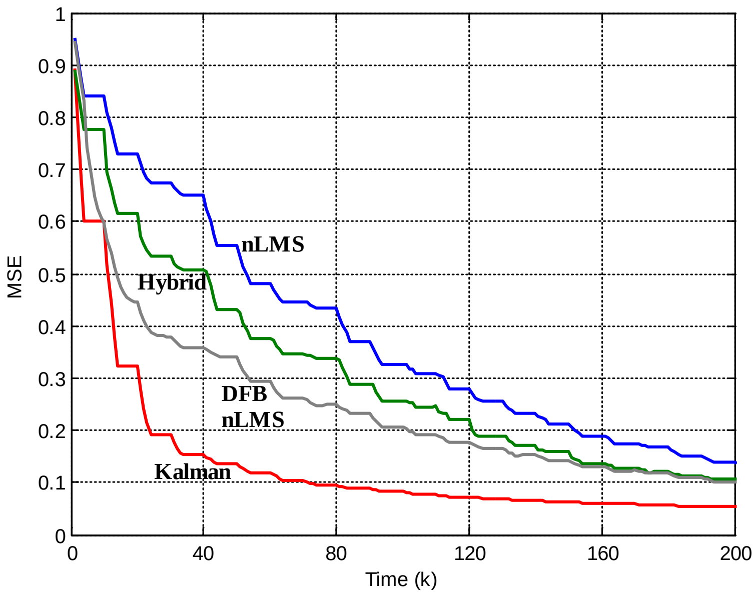

Fig. 5.2 MSE comparison of proposed algorithms with conventional algorithms using eigen training sequence and *v*=3*km/h*

In Fig. 5.2, the MSE performance using eigenvalue training sequences is shown. Both hybrid and decision feedback nLMS algorithms outperform than nLMS algorithm. One difference with Fig. 5.1 is the mutual performance change between hybrid algorithm and decision feedback nLMS algorithm. This is because as will be seen in section 5.3, the BER performance is improved by using eigenvalue training sequences, so the opportunity to apply a decision feedback scheme increases. In other words, there are many decoding errors at the beginning of the channel estimation with orthogonal training sequences, so the decision feedback scheme is hardly applied and it is difficult to see the effect. This also means that if the base station uses eigenvalue training sequences, the MSE performance of the channel estimation can be greatly improved without much increase in the hardware.

5.3. BER Performance Evaluation

Now, we compare the BER of the proposed channel estimation algorithms with conventional Kalman and nLMS algorithms. Because in actual wireless communications, how much the desired data is transmitted without error is an important issue, we evaluated BER performance of proposed algorithms. All figures in this section indicate the BER after channel estimation has converged to some extent.

5.3.1. Large Temporal Correlation Channel

In this section, we compare the BER performance of several algorithms when the temporal correlation coefficient a = 0.9999 (v = 3km/h). This condition indicates a person walking speed, so the correlation of the channel is very large in the time domain.

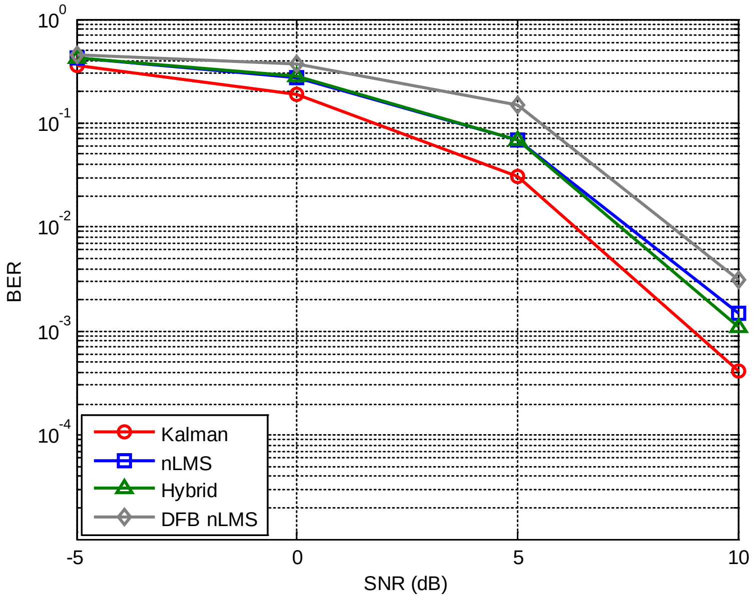

Fig. 5.3 BER comparison of proposed algorithms with conventional algorithms using orthogonal training sequence and ***v*****=3*****km/h***

Fig. 5.3 shows the BER by using orthogonal training sequences. Unlike the MSE case in Fig. 5.1, the performance difference between conventional nLMS and hybrid algorithm is hardly recognizable. Also, the performance of decision feedback nLMS is even worse than the nLMS because the decision feedback algorithm can mislead the CSI in a low SNR region. However, if the SNR is higher than 10dB, the BER is zero at any algorithm.

Fig. 5.4 shows the BER with using eigenvalue training sequences. In this case, BER performance of all algorithms is attached closer than in Fig. 5.3. Also the performance of decision feedback nLMS is the worst for the same reason as mentioned earlier.

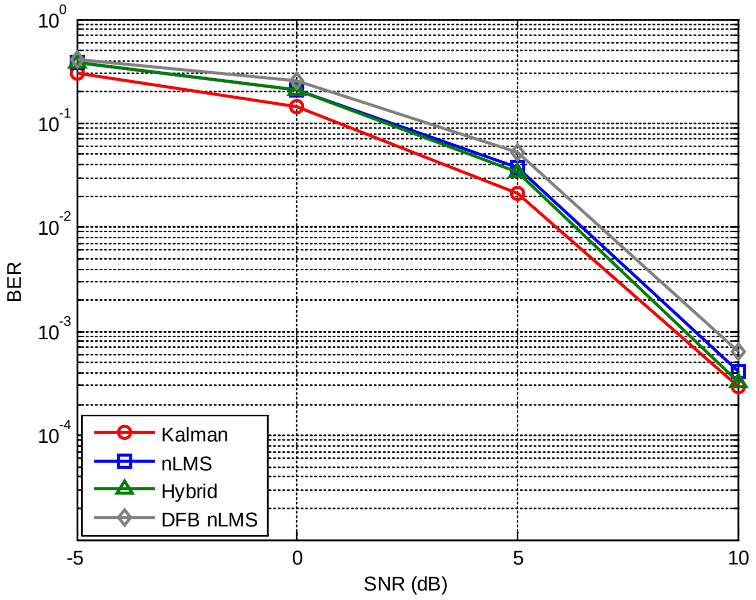

Fig. 5.4 BER comparison of proposed algorithms with conventional algorithms using eigenvalue training sequence and *v*=3*km/h*

In these two simulations, we can find out that the performance of BER even when using any channel estimation algorithm is similar in a large temporal correlation channel environment.

5.3.2. Small Temporal Correlation Channel

In this section, we compare the BER performance when the temporal correlation coefficient a = 0.99 (v = 130km/h). This condition indicates a transportation movement speed which means small correlation of the channel in the time domain.

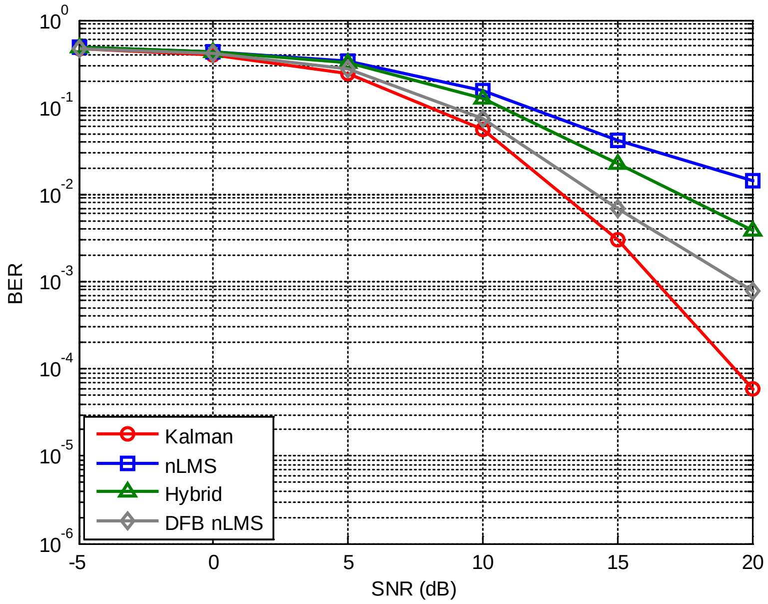

Fig. 5.5 BER comparison of proposed algorithms with conventional algorithms using orthogonal training sequence and *v*=130*km/h*

With small temporal correlation, two proposed algorithms improve the BER performance as shown in Fig. 5.5. About 3.5dB and 6.5dB at BER 10^{-2} is improved by using the hybrid algorithm and the decision feedback nLMS algorithm, respectively, compared to the nLMS algorithm. In this case, the decision feedback algorithm assumes the CSI feedback for every symbol during the data transmission period, so the BER performance will be degraded if the feedback is once in a block. Later in this chapter, we will show the performance in the case of reducing the number of feedback.

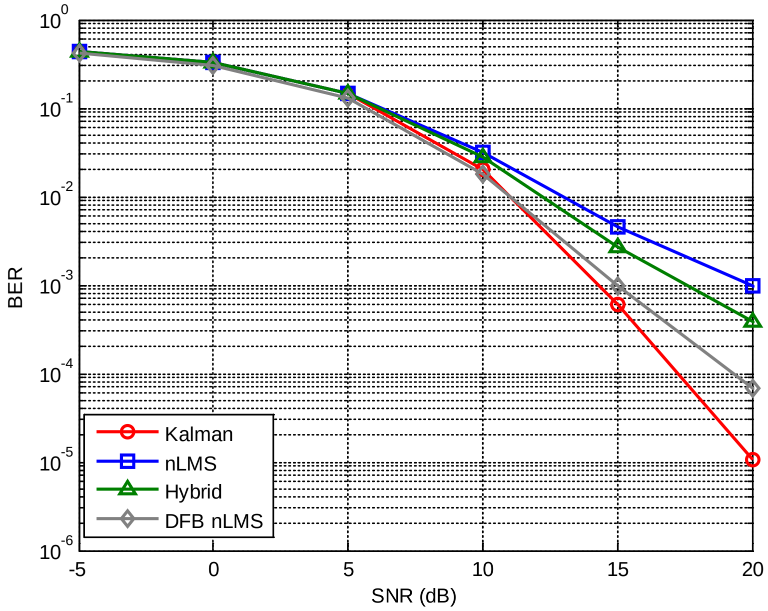

Fig. 5.6 BER comparison of proposed algorithms with conventional algorithms using eigenvalue training sequence and *v*=130*km/*

The BER performance trend in Fig. 5.6 is similar to Fig. 5.5. About 2.5dB and 5dB at BER 10^{-3} are improved by using the hybrid algorithm and the decision feedback nLMS algorithm, respectively, compared to the nLMS algorithm. With eigenvalue training sequences, the overall BER performance is improved compared to using the orthogonal training sequences. However, since the computational complexity of generating eigenvalue training sequence is higher than generating orthogonal training sequences at the base station, there is a problem of the choice between the trade-offs.

5.3.3. Delayed Channel Feedback

In this section, we first compare the BER performance when the number of CSI feedback to be reduced to one within the block.

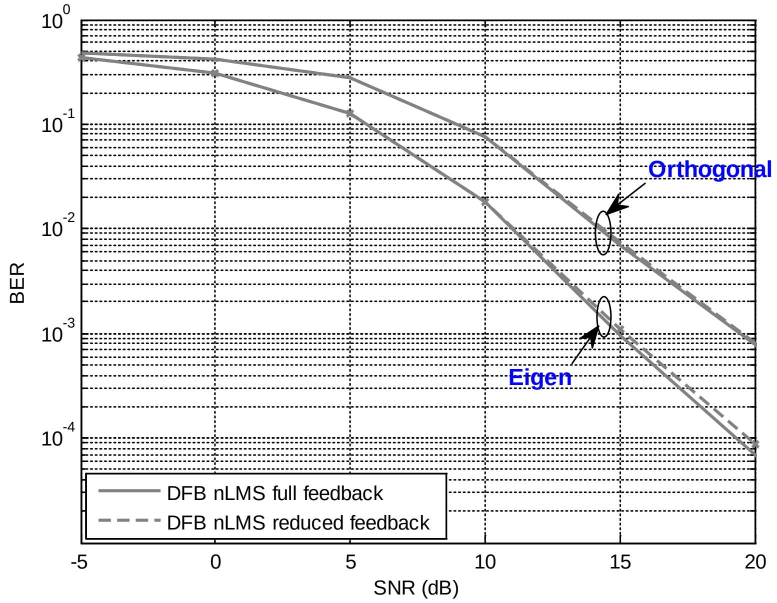

Fig. 5.7 BER comparison of proposed algorithms with conventional algorithms with reduced channel information feedback using eigenvalue training sequence and *v*=130*km/h*

In the decision feedback nLMS algorithm, the estimated channel is fed back once in a block at the end of pilot sequence transmission period. Fig. 5.7 shows the performance difference between the full feedback case and the reduced feedback case. When using orthogonal training sequences, the performance difference cannot be recognized between two feedback schemes. In the case of eigenvalue training sequences, about 0.5dB loss of performance at BER 10^{-3} is occurred than full feedback scheme. Since this is also a very narrow margin, almost no performance degradation is shown.

Next, we compare the BER performance when the transmitter has delayed CSI, so the precoding vector of current data symbols are generated by the CSI of the previous block.

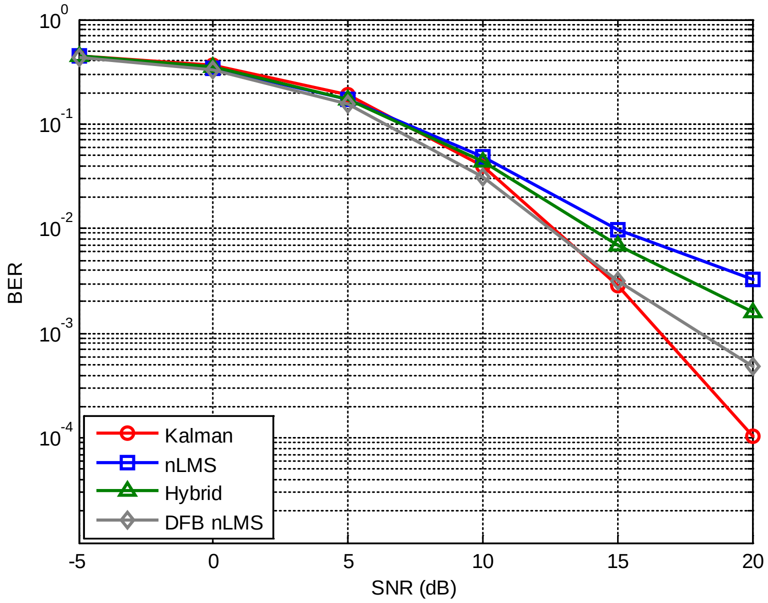

Fig. 5.8 BER comparison of proposed algorithms with conventional algorithms with delayed channel information feedback using eigenvalue training sequence and *v*=130*km/*

The BER performance with delayed CSI is shown in Fig. 5.8. Graphs are shifted to the right, and the error floor appears as SNR increases although not represented in Fig. 5.8. If we use orthogonal training sequences, graphs will be shifted far more to the right and the error floor becomes a more serious problem because the performance is never improved no matter how much the SNR increases. With orthogonal training sequences, the BER does not fall below 10^{-2} until the SNR goes up to 25dB with using the nLMS algorithm. With reduced feedback for decision feedback nLMS and delayed CSI, about 3dB and 5dB loss of performance at BER 10^{-3} is occurred by using decision feedback nLMS algorithm and hybrid algorithm, respectively.

Chapter 6.

Conclusions

Wireless communication systems have been required to effectively use the bandwidth and to have larger channel capacity in recent years, so the massive MIMO system was introduced as a response to these requirements. Since the size of a channel matrix in massive MIMO system is very large due the large number of antennas at the base station, acquisition of the channel information requires large computational complexity. Many studies so far consider the TDD based system to use channel reciprocity between uplink and downlink, however, the currently used mobile communication systems are mostly based on the FDD system which cannot use the channel reciprocity.

In this thesis, two types of adaptive channel estimation algorithms in a channel model with temporal and spatial correlations are proposed. First we propose time division operation of Kalman and normalized LMS (nLMS) filter which we call hybrid channel estimation algorithm, to overcome the processing delay caused by the matrix operation in Kalman filter. With this algorithm, it is possible to estimate the channel in real-time even though the size of hardware is slightly increased compared to the Kalman filter, and we call it the hybrid channel estimation algorithm. Second, decision feedback algorithm with nLMS filter has been proposed to improve the result of channel estimation during the data transmission period. By using correctly decoded data as a training signal for additional channel estimation, the channel estimation performance can be greatly improved without much increase of the computational complexity.

Simulation results are shown that the proposed channel estimation algorithms outperform than the conventional nLMS channel estimation algorithm in terms of both MSE and BER. Also, in the case of generating the precoding matrix with delayed channel information, both of the proposed algorithms showed the better performance than nLMS algorithm.

References

[1] E. Telatar, “Capacity of multi-antenna gaussian channels,” *AT&T-Bell Labs, Tech. Rep.*, June 1995.

[2] G. J. Foschini, and M. J. Gans, “On Limits of Wireless Communication in a Fading Environment when using Multiple Antennas”, *Wireless Personal Commun*., Vol. 6, no.3, pp 311-335, 1998.

[3] T. L. Marzetta, “Noncooperative cellular wireless with unlimited numbers of base station antennas,” *IEEE Trans. Wireless Commun*., vol. 9, no. 11, pp. 3590 - 3600, Nov. 2010.

[4] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, and T. L. Marzetta, “Scaling up MIMO: opportunities and challenges with very large arrays,” *IEEE Signal Process. Mag*., vol. 30, no. 1, pp. 40 - 60, Jan. 2013.

[5] J. Hoydis, S. ten Brink, and M. Debbah, “Massive MIMO in the UL/DL of cellular networks: How many antennas do we need?,” IEEE J. Sel. Areas Commun., vol. 31, no. 2, pp. 160 - 171, Feb. 2013.

[6] John G. Proakis, *Digital Communications*, McGrawHill, 2001

[7] S. Haykin, *Adaptive Filter Theory*, PrenticeHall, Upper Saddle River, New Jersey, 2002.

[8] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*, PrenticeHall, Upper Saddle River, New Jersey, 2000.

[9] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, “Massive MIMO for next generation wireless systems,” *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, 2013.

[10] H. Q. Ngo, E. G. Larsson, and T.L. Marzetta, “Energy and spectral efficiency of very large multiuser MIMO systems,”, IEEE Trans. Commun. vol. 61, pp. 1436-1449, Apr. 2013.

[11] C. Peel, B. M. Hochwald, and A. L. Swindlehurst, “A vector-perturbation technique for near-capacity multiantenna multiuser communication – part I: channel inversion and regularization,” *IEEE Trans. Commun.*, vol. 53, pp. 195–202, Jan. 2005.

[12] Q. H. Spencer, A. L. Swindlehurst, and M. Haardt, “Zero-forcing methods for downlink spatial multiplexing in multiuser MIMO channels,” *IEEE Trans. Sig. Proc.*, vol. 52, pp. 461–471, Feb. 2004.

[13] M. Biguesh, and A. B. Gershman, “Training-based MIMO channel estimation: a study of estimator tradeoffs and optimal training signals,” *IEEE Trans. Sig. Proc*., vol. 54, pp. 884-893, Mar. 2006.

[14] M. S. Grewal, and A. P. Andrews, *Kalman Filtering,* Wiley-Interscience, New York, New York, 2001.

[15] A.Khlifi, and R. Bouallegue, “Performance analysis of LS and LMMSE

channel estimation techniques for LTE downlink systems," International Journal of Wireless & Mobile Networks (IJWMN), vol. 3, No. 5, pp. 141-149, Oct. 2011.

[16] S. Noh, M.D. Zoltowski, Y. Sung, and D. J. Love, "Optimal pilot beam pattern design for massive MIMO systems," in *Proc. IEEE ASILOMAR*, Pacific Grove, CA, Nov. 2013.

초록

무선 통신 시스템은 더 많은 양의 데이터를 더 빠르게 전송하도록 요구되고 있고, 이에 무선 통신 시스템들은 더 효율적인 대역폭의 활용과 더 큰 채널 용량을 필요로 하게 되었다. 최근 이런 multiple input multiple output (MIMO) 시스템을 매우 큰 규모로 확장하여 무수히 많은 개수의 안테나를 사용하는 massive MIMO 시스템이 소개되었고, 주파수 효율을 매우 높일 수 있게 되었다. Massive MIMO 시스템은 많은 개수의 안테나를 이용하여 기지국에서 송신 데이터에 하향링크 채널 정보를 이용한 프리코딩 기법을 적용 함으로써 효율을 극대화할 수 있다. Time-division duplex (TDD) 시스템은 상향링크와 하향링크 채널 사이의 상호성을 이용하여 하향링크의 채널을 추정할 수 있어 주로 연구되어 왔다. 채널 상호성을 이용할 수 없는 frequency-division duplex (FDD) 시스템에서 하향 링크 채널 추정의 복잡도는 기지국의 안테나 수에 비례하므로 효율적인 채널 추정 기법이 연구될 필요가 있다. 본 논문에서는 채널이 시간, 공간적으로 서로 상관관계가 있는 경우, 적응 신호처리 기법을 이용한 채널 추정 알고리즘을 제안한다. 최적 채널 추정 기법이라고 알려진 칼만 필터는 행렬 연산이 주를 이루므로 실시간으로 채널 추정이 불가능 하다. 트레이닝 신호가 연속적으로 전송되는 경우, 첫 번째 트레이닝 신호는 칼만 필터를 이용하고 두 번째부터 마지막 트레이닝 신호는 nLMS 필터를 이용하도록 하는 시간분할방식을 제안하여 실시간으로 채널 추정이 가능하도록 한다. 또한 올바르게 디코딩 된 데이터를 트레이닝 신호로 이용하여 데이터를

전송하는 구간 동안 추가로 채널 정보를 갱신하도록 하는 decision feedback nLMS 필터를 제안한다. 이 방식을 이용하면 기존의 nLMS 필터를 이용하는 것과 비교하여 큰 하드웨어의 복잡도 증가 없이 성능을 매우 향상시킬 수 있다. 모의 실험을 통하여 mean square error (MSE)와 bit error rate (BER) 측면에서 기존의 알고리즘과 제안된 알고리즘 간의 성능을 비교하고 분석한다.

주요어 : massive MIMO, 채널 추정, 칼만 필터, nLMS, decision feedback

학번 : 2013-20882