A Machine Learning Approach to Link Adaptation for SC-FDE System

Zrinka Puljiz, Mijung Park, and Robert Heath, Jr.
Department of Electrical and Computer Engineering
The University of Texas at Austin, USA
{zrinka,mjpark,rheath}@mail.utexas.edu

***Abstract*—Single carrier frequency domain equalization (SC-FDE) uses cyclically prefixed quadrature amplitude modulation to permit simple frequency domain equalization at the receiver. Link adaptation for SC-FDE systems, where the modulation and coding rate are adapted based on the current channel state, is straightforward with perfect channel state information due to the simple analytical form of the post-processing signal-to-noise ratio (SNR). Imperfect channel state information, however, introduces adaptation errors. This paper proposes a machine learning-based approach for link adaptation in bit interleaved convolutionally encoded SC-FDE systems. To improve performance in the presence of channel uncertainty, principal component analysis is used to reduce the feature space dimensionality consisting of the channel coefficients, noise variance, and post-processing SNR. The reduced dimension feature set improves performance of the link adaptation classifier and leads to higher performance versus just the post-processing SNR estimate. Simulation results indicate that the proposed algorithm increases the goodput while maintaining the target packet error rate, achieving optimal adaptation in 95% of the tested cases.**

I. Introduction

Single carrier frequency domain equalization (SC-FDE) is a digital transmission technique for sending quadrature amplitude modulated (QAM) signals over a wideband frequency selective channel. SC-FDE uses a cyclic prefix, like in orthogonal frequency division multiplexing (OFDM), to enable low complexity equalization in the frequency domain [1]. Compared with OFDM, SC-FDE has a lower peak-to-average-power ratio, allowing more power efficient hardware architectures [1]. For example, SC-FDE is used in 60 GHz systems to enable lower cost power amplifiers and improve performance in the presence of nonlinearities [2]. A variation of SC-FDE is used for the uplink in Third Generation Partnership Project Long-Term Evolution cellular systems [3].

Link adaptation is used in wireless systems to increase spectrum efficiency by adapting the modulation and coding parameters at the transmitter to the current channel state. In most systems, link adaptation maximizes a capacity proxy known as the goodput, which accounts for packet errors. The idea is to use a previous estimate of the channel to choose the transmit parameters for use in the next channel. Link adaptation is challenging in practical systems because (i) error control techniques like bit interleaved convolutional coding make it hard to predict analytically the error for a given (possibly time-varying) channel response as [4] (ii) estimation error reduces the accuracy of link adaptation predictions [5], and (iii) there may be hardware-dependent impairments like nonlinearities and phase noise that are not included when designing the adaptation algorithm [6]. There is some prior work on link adaptation for SC-FDE systems, e.g. stochastic learning based link adaptation for single user and single carrier systems [7] and SC-FDE system link adaptation in [8]. Prior work, however, imperfect channel state information in the link adaptation algorithm.

In this paper, we propose a robust approach for link adaptation in coded SC-FDE systems based on machine learning. Our data driven algorithm predicts the optimum rate for the next transmission based on rules extracted from the previously observed channel realizations and rates. To increase performance beyond prior work that uses the post-processing SNR for adaptation, our approach expands the feature space to include the channel coefficients, noise variance and post-processing SNR. The noise variance provides information about the reliability of the channel estimates. Our first contribution is to apply principal component analysis (PCA) to reduce the dimensionality of the proposed feature set. Reducing the dimensionality improves classification-based link adaptation. We show that three dimensions are sufficient for good performance, compared with the one dimension used in prior work. Our second contribution is to evaluate the performance of two different classification techniques: a threshold method and k nearest neighbors (k-NN). Both methods are evaluated in terms of achievable goodput and packet error rate. They show favorable performance versus the one dimensional approach in [8]. Compared with related work on machine learning for link adaptation in OFDM systems [4] [9], our system model is that of SC-FDE. Compared with OFDM-based approaches, in SC-FDE the feature set, the dimensionality reduction techniques, and the adaptation algorithms are different. For example, in OFDM systems the ordered post-processing SNR in the frequency domain is important while in our case the ordering is not relevant since coding is performed in the time-domain. While our algorithm uses k-NN for classification (also considered in [4]), it could be extended to online adaptation with support vector machines along the lines of [6] [9].

II. System Model

Consider a SC-FDE system as illustrated in Figure 1. Let $x[n]$ denote the QAM symbol transmitted in time period n, $\{h[\ell]\}_{\ell=0}^{L}$ denote the channel coefficients with order L, and

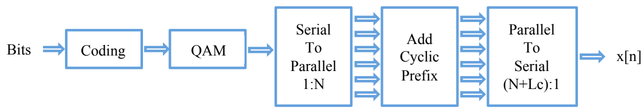

Fig. 1. Block diagram of the transmitter

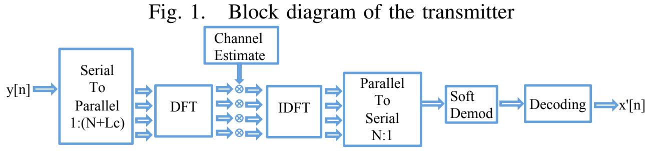

Fig. 2. Block diagram of the receiver

$v[n]$ is complex Gaussian additive white noise. Assuming perfect frame and carrier synchronization, the received complex baseband discrete-time input-output relationship for a frequency selective channel [4] is

$$y[n] \;=\; \sum_{\ell=0}^{L} h[\ell]x[n-\ell] + v[n]. \tag{1}$$

Supposes that every N symbols are prepended with a cyclic prefix of length $L_c \geq L$ at the transmitter. FDE is considered to have lower complexity than time domain equalization when $L \geq 5$ [1]. Discarding the first L_c samples at the receiver, the channel effect takes the form of a circular convolution. SC-FDE exploits the fact that the discrete Fourier transformation (DFT) performs circular convolution. Thus a typical SC-FDE receiver equalizes the channel by dividing the DFT of the received signal with DFT of channel estimate (Figure 2). This can be written using the circulant matrix $\mathbf{H}$, diagonalized by the DFT matrix $\mathbf{F}$ to create $\mathbf{D}$ containing the DFT coefficients of the zero padded channel

$$\mathbf{y} \;=\; \mathbf{Hx} + \mathbf{v} \;=\; \mathbf{F}^{*}\mathbf{DFx} + \mathbf{v}. \tag{2}$$

Assuming perfect channel knowledge, the signal estimate is

$$\begin{aligned} \hat{\mathbf{x}} := \mathbf{D}^{-1}\mathbf{Fy} &= \mathbf{Fx} + \mathbf{D}^{-1}\mathbf{Fv} \\ &= \mathbf{x} + \mathbf{F}^{*}\mathbf{D}^{-1}\mathbf{Fv} \end{aligned} \tag{3}$$

where $\mathbf{F}^{*}\mathbf{D}^{-1}\mathbf{Fv}$ is spatially correlated noise. In real systems, perfect channel state information is not available, thus an estimate is used in its place.

As in several commercial wireless systems, we consider QAM modulated signals, bit interleaving, and convolutional coding. The outputs from (3) are converted to bit likelihoods, using the max-log map approximation [10], neglecting any coloring of the noise and fed into a soft-input Viterbi decoder.

To facilitate link adaptation, a finite number of possible modulation and coding schemes (MCS) are considered, defined as the pair consisting of modulation parameter M_i (corresponding to M-QAM) and code rate $(k/n)_i$. With symbol period T, the maximum throughput of MCS class i is given by

$$T_i = \frac{N}{N+L_c}\frac{\log_2(M_i)(k/n)_i}{T}. \tag{4}$$

Note that (4) omits channel training overhead. Given the maximal throughput T_i and packet error rate PER_i of a class i, the goodput of class i in the channel is defined as

$$R_i = T_i(1 - PER_i). \tag{5}$$

Goodput focuses on the useful bits that were successfully decoded at the receiver. It takes into account decoding errors that happen with finite block lengths (compared with capacity), and other sources of loss in the channel. Therefore it is a practical measure of useful bit transmissions and is often used in the literature as the link adaptation measure of performance [4].

In this paper we consider the link adaptation problem where the MCS with the highest goodput is chosen subject to the packet error rate (PER) constraint giving the selected MCS

$$arg\max_{i} R_i : PER_i < P. \tag{6}$$

It is commonly assumed that P is 10% [11] [4].

III. Proposed Approach to Link Adaptation

We view the link adaptation problem through the lens of classification in a machine learning framework. The modulation and coding scheme that results in the highest goodput for given channel conditions corresponds to the best classification of the multidimensional data. The classification method is based on supervised learning, where a set of training data with optimum classifications is processed off-line to extract the classification rule. The quality of the extracted rules is evaluated on a set of testing data, which is independent of the training data. The amount of data required for efficient classification depends on the dimensionality of the data. High dimensionality retains the important information in the data but requires an exponential increase in the training data size (dimensionality curse) for nearest neighbor classification. Lower dimensionality reduces the required amount of training data for good classification. Thus reducing dimensionality of the raw data is an essential pre-curser to classification. The dimensions of the resulting data in a machine learning framework are called features. We propose to determine the dimensionality of the feature set offline, though it could be done online as well [6] [9]. In this section we present our approach for dimensionality reduction followed by our approach for classification-based link adaptation.

A. Dimensionality Reduction

For uncoded systems with perfect equalization, [8] shows that the post-processing SNR for a SC-FDE system is sufficient for choosing the best modulation parameters. The post-processing SNR for a SC-FDE system is derived in [8] as

$$\text{ppSNR} = \left(\sigma^2 \frac{1}{N}\sum_{n=0}^{N-1} |H[n]|^{-2}\right)^{-1} \tag{7}$$

where $\{H[n]\}$ are channel coefficients in the frequency domain. Though the post-processing SNR is a function of the other parameters, the relationship between them is nonlinear and simple estimates of other coefficients do not suffice for extracting its mean. Therefore, we estimate the post-processing SNR for each packet sent and average it over a burst of packets sent using the same channel. Consequently we use the *estimated* ppSNR, the estimated channel coefficients $\{h[n]\}$, and the noise variance σ^2 as a starting point for our feature vectors. We employ both types of dimensionality reduction techniques, namely, feature selection and feature extraction. Both of the dimensionality reduction techniques used result in a linear transformation of the original features to reduced feature space, meaning that the computational complexity will be fixed to at most $(L+2)^2$.

1) Simple Projection: This method is a simple dimensionality reduction technique. Given a data point in multidimensional space $x = [x_1; x_2; \ldots x_n]$, simple projection is the linear operator P_j that keeps first j components of x. Link adaptation based on $ppSNR$ therefore can be viewed as classification of the data projected to one component. This approach will be used as a baseline for comparison since it corresponds to previous work [8].

2) Principal Component Analysis (PCA): Feature extraction relies on the more automated methods of identifying key features. PCA is a data centric method broadly used for various applications such as dimensionality reduction, data compression, feature extraction, and data visualization [12]. PCA is an orthogonal transformation that maps the data from a higher dimensional space into a lower dimensional space. The first component has the highest variance possible; the following components maximize the variance under constraint of orthogonality. This in turn results in reducing the data to few directions that contain most of the information of the original data.

We adapt the PCA description from [13] to reduce the dimensionality of the channel realization as follows. In the set of the training data of length N, let $\mathbf{w^{(i)}}$ represent the i^{th} channel realization. The channel realization $\mathbf{w^{(i)}}$ consists of the estimated channel coefficients $h[n]$, the noise variance σ^2 and the estimated $ppSNR$. Furthermore let k be the dimensionality of the targeted space.

Based on the training vectors $\mathbf{w^{(i)}}$, we find the empirical covariance matrix $\mathbf{R}$

$$\mathbf{R} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{w^{(i)}} - \bar{\mathbf{w}})(\mathbf{w^{(i)}} - \bar{\mathbf{w}})^T, \qquad (8)$$

where $\bar{\mathbf{w}}$ is the mean value of the $\mathbf{w^{(i)}}$. We use $\mathbf{w}$ instead of $\mathbf{w^{(i)}}$ when the specific index is not important. Next, find eigenvalues and eigenvectors of $\mathbf{R}$. The size of the matrix in our case is given by $L+2$, where L is the channel length, so for most practical systems this is a cost effective operation that needs to be done only once. We order the eigenvalues from highest to lowest ($\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{n+1}$) and form matrix $\mathbf{P_k} = [\mathbf{v}_1\mathbf{v}_2 \cdots \mathbf{v}_k]$, where $\mathbf{v}_j$ is the eigenvector associated with eigenvalue λ_j. The PCA matrix $\mathbf{P_k}$ is completely defined using the training data. To reduce the dimensionality of the testing data $\mathbf{u}$ we use

$$\mathbf{z}(\mathbf{u}) = \mathbf{P_k}^T(\mathbf{u} - \bar{\mathbf{w}}) \qquad (9)$$

where $\bar{\mathbf{w}}$ is the average of the training data.

B. Classification Algorithms

The classification algorithm is the heart of supervised learning. Based on the set of training data with the explicit values of the output function, the classification algorithm derives rules for mapping new data to the output function. We use a classification algorithm to find the map from our reduced dimension space of channel realizations to the MCS class that leads to highest goodput. The performance of classification algorithm is evaluated using a set of testing data. In the testing data, the best MCS is known; we use the classification algorithm to hypothesize the best MCS. When the hypothesis corresponds to the best MCS, we say that the classifier achieved perfect link adaptation. The goodness of the classifier is therefore evaluated based on how often it achieves perfect adaptation. We consider two different classification algorithms, threshold extraction and k-nearest neighbor algorithm.

Threshold extraction is a classification algorithm for one dimensional space. Two neighboring classes are separated with a threshold value, such that the number of misclassifications in the training data set is minimized. The optimization is performed over a finite set of possible thresholds.

k-NN algorithm is a classification algorithm algorithm that classifies the new realization to the same class as found by a majority of its k-nearest neighbors. The measure of closeness between a query and its k nearest neighbors is determined by their Euclidean distance $d(\cdot)$, though other distance functions may improve performance. The dimensionality reduction operator $\mathbf{P}$ maps training data $\mathbf{w}$ to elements of the feature space $\mathbf{z}(\mathbf{w}) = \mathbf{P}^T\mathbf{w}$. Let $i(\mathbf{w})$ be a function that associates an AMC class to a channel realization, so $i(\mathbf{w}) \in \{MC_1, MC_2, ..., MC_m\}$. Given the training data $\mathbf{z}(\mathbf{w})$, its associated classes $i(\mathbf{w})$, and a new channel realization $\mathbf{u}$ the class of a new data point $\mathbf{z}(\mathbf{u})$ is found using the k-NN algorithm: find $d(\mathbf{z}(\mathbf{w^{(i)}}), \mathbf{z}(\mathbf{u}))$ for all point $\mathbf{w^{(i)}}$ in the training set, order the distances, and finally, choose the k points $\mathbf{z}(\mathbf{w^{(i_j)}})$, $j \in \{1, 2, .., k\}$ that are nearest to $\mathbf{z}(\mathbf{u})$ (example in the Figure 3). We assign $i(\mathbf{u})$ to be equal to the most frequent value of $i(\mathbf{w^{(j)}})$, with ties broken by choosing the lower value of i.

IV. Simulation Results and Discussion

In this section we evaluate the performance of our algorithms. First we introduce the details of the simulation setup. Second we present the simulation results and discussion of the observed system behavior.

A. Channel Model

We consider a SC-FDE system with 8 channel taps and bandwidth of $50MHz$. The RMS delay spread of the channel is 50μs which makes the coherence bandwidth on the order of $4MHz$. We test our method on model A channels [14]. The symbol period is 20μs and the Rayleigh distributed channel taps have power delay profile in table I, obtained from [15].

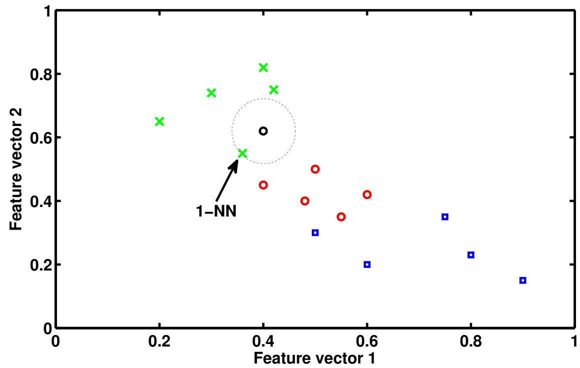

Fig. 3. 1-NN example

TABLE I
MEAN VALUE OF THE CHANNEL TAPS IN dB

Channel tap #	0	1	2	3	4	5	6	7
Power [dB]	0	-1.7	-3.5	-5.2	-6.9	-9.0	- ∞	-11.1

One channel realization corresponds to one sampling of appropriately parameterized Rayleigh distribution. Noise is AWGN and for each channel realization there are six different noise levels: -30dB, -27dB, -24dB, -21dB, -18dB and -15dB.

B. System Assumptions

In our system we assume the length of all packets is fixed at 500 bits of data per packet, since the packet length can change the performance of rate adaptation. While our approach is general enough to include synchronization errors, we assume perfect synchronization in the simulations. The channel coefficients are estimated at the receiver using the an MMSE estimator with Baker sequence as channel training data.

C. System Parameters

We use goodput to measure the performance while keeping PER below the threshold. Packets are sent in bursts of 1000 packets and the PER corresponds to the percentage of discarded packets. A packet is discarded if it is not perfectly decoded. We use 6 different MCSs which are distinguished by the combination of QAM parameter M and convolutional code rate (n/k). The DFT size, N, is fixed to be 64. A larger DFT size induces less overhead due to the cyclic prefix but increases the computational complexity. The values of parameters M and (n/k) as well as corresponding throughput are given in table II. The exact values of generator polynomials for convolutional codes and interleaving scheme are specified in [10].

D. Results

First we consider the case with perfect channel knowledge available at the receiver since this results represent a reference point for later experiments. For a system with perfect channel knowledge, goodput is a function of the post-processing SNR, with only minor discrepancies that we contribute to the impact of coding and interleaving on the data. This matches the results from [8].

TABLE II
MODULATION AND CODING SCHEMES

Class #	M	R	Thr(Mbit/s)
1	4	1/2	44
2	4	3/4	59
3	16	1/2	89
4	16	3/4	118
5	64	1/2	133
6	64	3/4	178

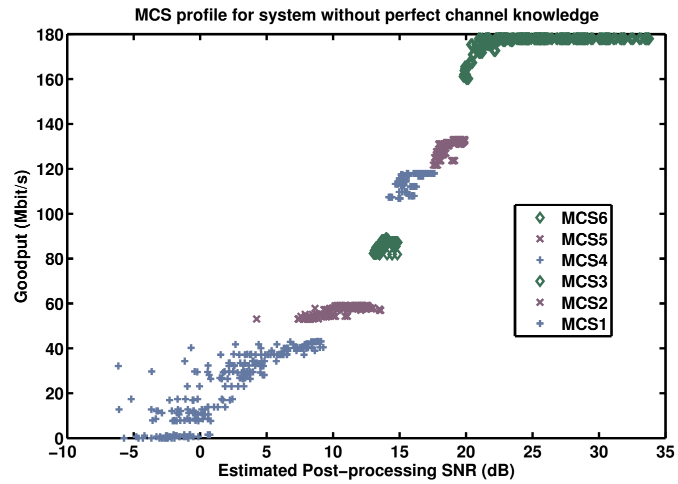

Fig. 4. MCS profile without perfect channel knowledge

To apply the machine learning approach, we divide the generated data into training data and testing data. Each group has a different number of channel realizations. For each combination of channel coefficients and noise power we send the data in a LabVIEW simulation setup using all six different modulation and coding schemes. The training data consists of 80 different channel coefficients sampled from Rayleigh distribution, and for each set of channel coefficients we have 6 different noise levels and use the transmission with all 6 different modulation and coding schemes resulting in 2880 samples. For testing data the number of different channel realizations is 50 leading to total of 1800 samples. Training data is used for extracting the rules while testing data showed in results validates the extracted rules.

For the receiver that estimates the channel conditions, MCS profile is given in the figure 4. To make the differences more visible, we plot the goodput vs. SNR for each of the resulting MCS classes used in the training data. On the graph if we pick a vertical line and look at how good it separates MCS classes from one another based on the estimated $ppSNR$ we observe that the boundaries are not hard. This means that ppSNR estimate does not contain enough information to enable perfect rate adaptation, justifying the inclusion of additional information.

Figure 5 compares different link adaptation methods in terms of the overall goodput and packet error rate. This figure shows how the cumulative goodput and the packet error rate of the testing data changes with different values of post-processing SNR. The curve is obtained by creating different bins for different values of observed SNR and averaging the goodput of all the data that falls into the same bin. For comparison the graph includes the curves for goodput and PER for each single MCS class, which helps us observe when the MCS class should switch due to the higher goodput given the fixed PER. We compare three different adaptation methods to the ideal adaptation. Ideal adaptation is obtained explicitly by sending the data with all the different modulation and coding schemes and then picking the one that achieves the maximal goodput subject to the PER constraint. The worst performance is observed when thresholds obtained from the system with the perfect channel knowledge are applied to a system with channel estimation - the traditional approach. An easy way to improve performance with just the $ppSNR$ is by learning the thresholds in the system with imperfect channel knowledge. This adaptation method leads to 82 misclassifications. The correct classification is therefore given in more than 95% of the cases. The performance of link adaptation is further improved with the PCA dimensionality reduction, in which case three dimensional PCA with 25-NN algorithm gives 71 misclassifications, leading to a 10% further improvement.

Observe that the packet error rate constraint can not be removed by simply optimizing the goodput. With the adaptation based on ideal channel knowledge and post-processing SNR value at 20dB the goodput achieved is actually greater than the rate of the ideal adaptation, but the PER is above the threshold value of 10%. The adaptation gives good performance by picking the MCS class that is near-optimal.

In summary, our results indicate that the machine learning approach improves the link adaptation for SC-FDE system with imperfect channel knowledge.

V. Conclusion

In this paper we proposed a robust data-driven algorithm for link adaptation in SC-FDE systems. Our approach leverages the data centric nature of machine to provides an algorithm that is flexible and can naturally incorporate uncertainty. In our case, the data consist of the post-processing SNR along with the estimated channel coefficients and the noise variance, while uncertainty comes from channel estimation. We proposed two approaches for link adaptation that consist of dimensionality reduction based on principle component analysis followed by classification. We showed that post-processing SNR based link adaptation in coded SC-FDE systems can be improved when adaptation thresholds account for channel estimation error; more accuracy can be achieved with a slightly larger reduced dimensionality feature set. Our classifier is designed using off-line training data; creating an on-line algorithm is a topic of future work.

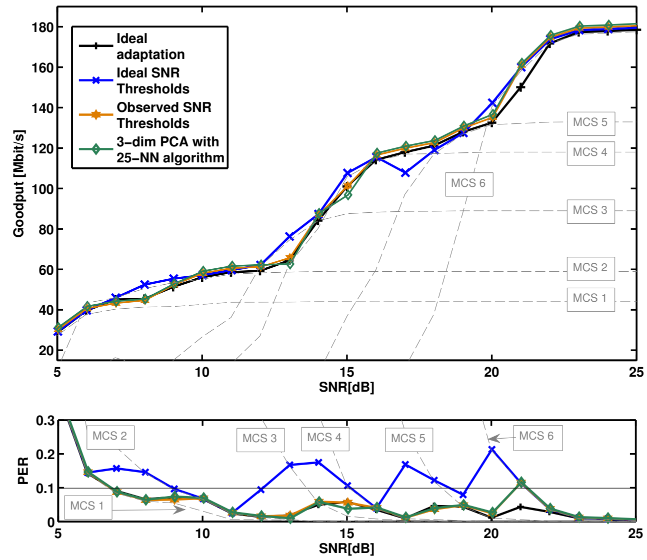

Fig. 5. Goodput and PER for different adaptation schemes

References

[1] D. Falconer, S. L. Ariyavisitakul, A. Benyamin-Seeyar, and B. Eidson, "Frequency domain equalization for single-carrier broadband wireless systems," *IEEE Communications Magazine*, Apr. 2002.

[2] M. Lei, C.-S. Choi, R. Funada, H. Harada, and S. Kato, "Throughput comparison of multi-gbps wpan (ieee 802.15.3c) phy layer designs under non-linear 60-ghz power amplifier," in *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*, 2007, pp. 1 –5.

[3] H. G. Myung, J. Lim, and D. J. Goodman, "Single carrier FDMA for uplink wireless transmission," *IEEE Vehicular Technology Magazine*, Sept. 2006.

[4] R. C. Daniels, C. M. Caramanis, and R. W. Heath, Jr., "A supervised learning approach to adaptation in practical MIMO-OFDM wireless systems," in *Proceedings of the IEEE Global Communications Conference*. IEEE, 2008.

[5] M. Lampe, H. Rohling, and W. Zirwas, "Misunderstandings about link adaptation for frequency selective fading channels," in *Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on*, vol. 2, 2002, pp. 710 – 714 vol.2.

[6] R. Daniels and R. Heath, "Online adaptive modulation and coding with support vector machines," in *Wireless Conference (EW), 2010 European*, 2010, pp. 718 –724.

[7] A. Misra, V. Krishnamurthy, and S. Schober, "Stochastic learning algorithms for adaptive modulation," in *Signal Processing Advances in Wireless Communications, 2005 IEEE 6th Workshop on*.

[8] T. Junyi Wang Baykas, R. Funada, A. Chin-Sean Sum Rahman, H. Zhou Lan Harada, and S. Kato, "A SNR mapping scheme for ZF/MMSE based SC-FDE structured WPANs," Apr. 2009.

[9] S. Yun and C. Caramanis, "Multiclass support vector machines for adaptation in MIMO-OFDM wireless systems," in *Allerton 2009.*

[10] S. B. Wicker, *Error control systems for digital communication and storage*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1995.

[11] S. Kant and T. L. Jensen, *Fast link adaptation for IEEE 802.11n*. M.S. thesis, Aalborg University, 2007.

[12] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, August 2006.

[13] I. Jolliffe, *Principal Component Analysis*. John Wiley & Sons, Ltd, 2005.

[14] A. Doufexi, *A Comparison of the HIPERLAN/2 and IEEE 802.11a Wireless LAN Standards*, May, 2002.

[15] V. Erceg and at al., *IEEE P802.11 Wireless LANs, TGn Channel Models*, May, 2004.