A NEW CLASS OF SPARSE CHANNEL ESTIMATION METHODS BASED ON SUPPORT VECTOR MACHINES

By

DONGHO HAN

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2006

To my family

ACKNOWLEDGMENTS

I would like to sincerely thank my advisor, Dr. José C. Principe, and co-advisor, Dr. Liuqing Yang, for their support, encouragement and patience in guiding the research.

I would also like to thank Dr. John Harris and Dr. Douglas Cenzer, for being on my committee and for their helpful advice.

I gratefully acknowledge Dr. Ignacio Santamaría and Javier Vía for providing numerous suggestions and for helpful comments on several of my papers that are part of this dissertation.

I also appreciate the helpful discussions from people in the Computational Neuroengineering Lab.

Finally, I wish to thank my wife, Hyewon Park, for her support, as well as my dear princesses, Eujin and Jiwon, for providing big motivations to finish my study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

A NEW CLASS OF SPARSE CHANNEL ESTIMATION METHODS BASED ON SUPPORT VECTOR MACHINES

By

Dongho Han

December 2006

Chair: José C. Principe
CoChair: Liuqing Yang
Major Department: Electrical and Computer Engineering

In this dissertation sparse channel estimation is reformulated as support vector regression (SVR) in which the channel coefficients are the Lagrange multipliers of the dual problem. By employing the Vapnik's ϵ-insensitivity loss function, the solution can be expanded in terms of a reduced number of Lagrange multipliers (i.e., the nonzero filter coefficients) and then a sparse solution is found. Furthermore, methods to extend the SVR technique are investigated to derive an iterative algorithm for blind estimation of sparse single-input multiple-output (SIMO) channels. This method can be also used for non-sparse channels, in particular when the channel order has been highly overestimated. In this situation, the structural risk minimization (SRM) principle pushes the small leading and trailing terms of the impulse response to zero. Results show that the SVR approach outperforms other conventional techniques of channel estimation. The main drawback of this approach is the high computational cost of the resulting quadratic programming (QP) solution. To reduce the complexity, we propose a simple and fast iterative algorithm called the Adatron to solve the SVR problem iteratively. Simulation results demonstrate the performance of the method.

CHAPTER 1
INTRODUCTION

In many wireless communication systems, the propagation channels involved exhibit a large delay spread, but a sparse impulse response consisting of a small number of nonzero coefficients. Such sparse channels are encountered in many communication systems. Terrestrial transmission channel of high definition television (HDTV) signals are hundreds of data symbols long but there are only a few nonzero taps [3]. A hilly terrain delay profile has a small number of multipath in the broadband wireless communication [4] and underwater acoustic channels are also known to be sparse [5]. An example of the sparse channel is shown in Fig. 1-1 and estimation of sparse channels will be mainly considered in this dissertation.

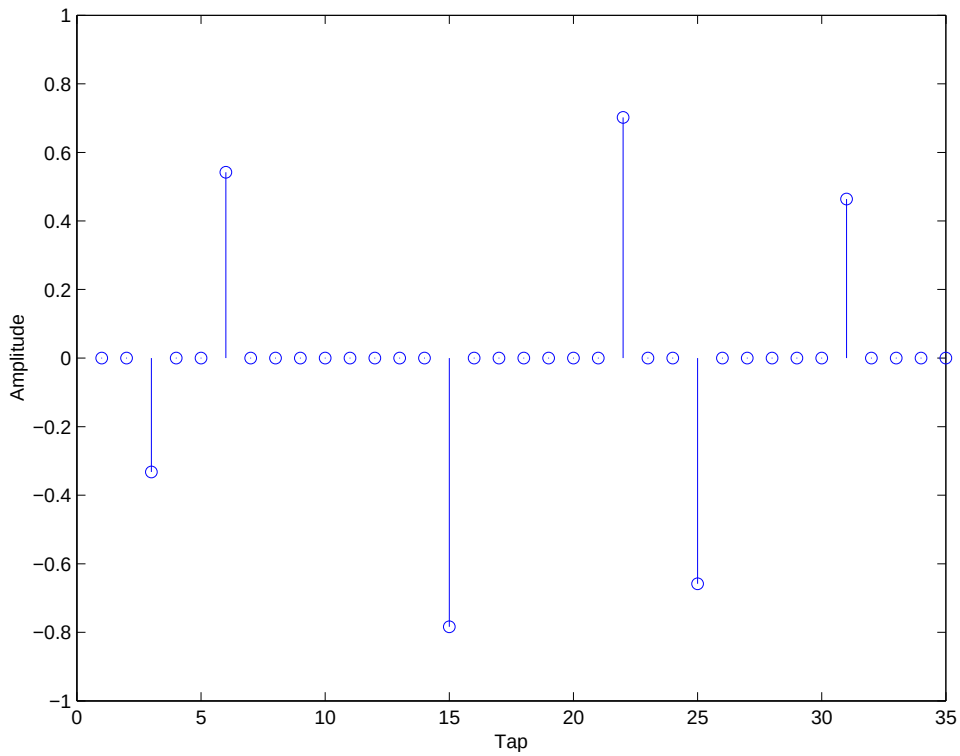


Figure 1-1. A sample sparse channel impulse response (adapted from [1])

1.1 Background

The roots of sparse system identification lie deep in signal representation. Instead of just representing objects as superpositions of sinusoids (the traditional Fourier representation) we have available alternate dictionaries. Wavelet dictionaries, Gabor dictonaries, Wavelet Packets, Cosine Packets, and a wide range of other representation [6], [7]. We should obtain the sparsest possible representation of the object (i.e. the one with the fewest significant coefficients) from a large dictionaries.

Mallat and Zhang [8] have proposed the matching pursuit (MP) algorithm which builds up a sequence of sparse approximations. But the algorithm is greedy, when run for many interations, it might spending most of its time correcting for any mistakes made in the first few terms. Also it is possible to re-select a previously selected vector in the dictionary.

To avoid the limitations of greedy optimization, Chen and Donoho [9] have suggested a method of decomposition based on a true global optimization which is called basis pursuit (BP). They propose to pick one whose coefficients have minimum L_1-norm from the many possible solutions to $\mathbf{Ax} = \mathbf{y}$ as in equation (1–1).

$$\min \| \mathbf{h} \|_1 - \lambda |\mathbf{y} - \mathbf{Ax}|^2 \tag{1–1}$$

Both MP and L_1-norm regularization (BP) can be viewed as trying to solve a combinatorial sparse signal representation problem. MP provides a greedy solution, while L_1-norm based BP replace the original problem with a relaxed version. Sparse signal representation problem has various applications such as time/frequency representations, speech coding, spectral estimation, and image coding [10–13]. Sparse channels are frequently encountered in communication applications and we are going to focus on estimation of sparse channel.

Modern estimation theory can be found at the heart of many electronic signal processing systems designed to extract information. These systems include radar, sonar,

speech, image analysis, biomedicine, communication, control, seismology and so on. All of these systems share the common problem of needing to estimate the values of a group of parameters. This problem has a long and glorious history, dating back to Gauss who in 1975 used least squares data analysis to predict planetary movements [14]. A salient feature of this least squares method is that no probabilistic assumptions are made about the data, only a signal model is assumed. The advantage is its broader range of possible applications. On the negative side, no claims about optimality can be made and furthermore, the statistical performance cannot be assessed without some specific assumptions about the probabilistic structure of the data.

Channel estimation using least-squares does not exploit the sparsity of channels and need long training symbols to produce an accurate estimate. A MP algorithm based sparse channel estimation method is proposed in [1] and this method viewed the estimation problem as a sparse representation problem and exploited the sparse nature of the channel using MP algorithm. It is shown that, the MP based channel estimation is more accurate and outperformed the conventional least square based methods in robustness and low complexity accuracy. A parametric method for selecting the structure of sparse model has been proposed in [15]. This method exploits the information provided by the local behavior of an information criterion to select the structure of sparse models.

An adaptive L_1-norm regularization method (sparse-LMS) using the augmented Lagrangian to find the sparse solution has been proposed in [16]. This method augmented the L_1-norm constraint of the channel impulse response as a penalty term to the MSE criterion. However, determination of the parameters in regularization problems, here L_1-norm of the channel coefficients which play an important role in achieving good estimation, has remained an open problem. In [17], a searching tool for the L_1-norm of channel coefficients using the convolution inequality for entropy is developed to improve the performance [18, 19].

Since Sato [20] first proposed the innovative idea of self-recovering (blind) adaptive identification, blind channel identification and equalization have been studied by many researchers [21–27]. Since the work of [28], it has been well known that second order statistics (SOS) are sufficient for blind identification when the input signal is informative enough and the channels do not share any common roots. Widely used SOS-based methods include the subspace (SS) approach [29], the least squares (LS) [27] technique and the linear prediction (LP) methods. However, a common drawback of SS and LS techniques is their poor performance when the channel order is overestimated. Recently some robust techniques have been proposed to mitigate this problem (see e.g., [30, 31]). Although these methods offer increased robustness, they still fail when the channel order is highly overestimated.

1.2 Contribution of this Thesis

Before this research had started, L_1-norm regularization method (BP) and matching pursuit (MP) algorithms are widely used methods in many sparse applications because it is well known that the concavity of the L_1-norm in the parameter space yields sparse solution. But MP algorithm is greedy and L_1-norm regularization method is non-quadratic optimization problem.

Support vector machines (SVMs) are a powerful learning technique for solving classification and approximation which can be derived from the structural risk minimization (SRM) principle [32]. The SRM principle is a criterion that establishes a trade-off between the complexity of the solution and the closeness to the data. This support vector machine (SVM) technique typically provides sparse solutions in the framework of SRM [33]. Specifically, the SVM solution can be expanded in terms of a reduced set of relevant input data samples (support vectors).

The first contribution of this dissertation is to use support vector machines (SVMs) for sparse channel estimation because SVM is known to build parsimonious models and results in a quadratic problem [34]. In the proposed formulation, the channel coefficients

play the role of the Lagrange multipliers. By using the Vapnik's ϵ-insensitive loss function only those Lagrange multipliers corresponding to support vectors are nonzero and therefore a sparse solution is obtained.

The second contribution is an iterative algorithm for blind estimation of sparse single-input multiple-output (SIMO) channels is derived. The work in [35] was the first attempt to apply an SVM-based approach to the blind identification of SIMO channels. However, the sparsity provided by the SVM solution was not explicitly exploited in [35]. In this direction we present a new blind identification algorithm based on support vector regression and specifically tailored for sparse SIMO channels in [34]. The main idea is that the sparse SIMO channel identification can be reformulated as a set of regression problems in which the channel coefficients play the role of the Lagrange multipliers. Thus we can get a sparse soluton. This method can be also used for non-sparse channels, in particular when the channel order has been highly overestimated. In this situation, the structural risk minimization (SRM) principle pushes the small leading and trailing terms of the impulse response to zero.

The main drawback of applying this SVM technique is the high computational cost of the resulting quadratic programming (QP) problem. As a third contribution, a simple and fast iterative algorithm called the Adatron is used to solve the SVR problem [36, 37].

1.3 Organization of the Dissertation

The organization of this proposal is as follows: In Chapter 2, we briefly summarize the conventional sparse solutions such as MP method and sparse-LMS using the convolution inequality for entropy. Chapter 3 introduces support vector machines including the classification and the regression problem. Then we present a new SVM-based sparse system identification algorithm. SVM meta-parameter selection is also discussed. In Chapter 4, iterative algorithm for blind estimation of sparse SIMO channels will be derived and Adatron algorithm to reduce the complexity of support vector machine based method will be also introduced. In Chapter 5, SVM technique with the exponential basis

expansion model (BEM) is applied to the time-varying channel identification. Finally, Chapter 6 gives conclusions and future work. In appendix, to investigate the feasibility of the FPGA implementation of the proposed algorithm, the core computation of this Adatron is simulated in FPGA using System Generator which is a high-level design tool for Xilinx FPGAs that extends the capabilities of Simulink to include accurate modeling of FPGA circuits. The performance is also compared to the TI TMS320C33 DSP implementation.

CHAPTER 2
CONVENTIONAL SPARSE SOLUTION

2.1 Observation Model

The sparse channel estimation problem can be stated as follows. Let $s(n)$ be the training sequence for $n = 0, \ldots, N-1$, that is transmitted through a stationary channel, $h(0), h(1), \ldots, h(M-1)$ which is sparse. The training sequence symbols $r(n)$ for $n < 0$ can be obtained from the previous estimates or for the first arriving frame they are assumed to be zero. White Gaussian noises $e(0), \ldots, e(N-1)$ are added to this transmitted signal where N is the training symbol number and M is the length of the channel to be identified. Then the receive signal $r(n)$ can be expressed in matrix form as

$$\mathbf{r} = \mathbf{Sh} + \mathbf{e} \tag{2–1}$$

where

$$\mathbf{r} = \begin{pmatrix} r(0) \\ \vdots \\ r(N-1) \end{pmatrix}, \mathbf{h} = \begin{pmatrix} h(0) \\ \vdots \\ h(M-1) \end{pmatrix}, \mathbf{e} = \begin{pmatrix} e(0) \\ \vdots \\ e(N-1) \end{pmatrix}$$

$$\mathbf{S} = \begin{pmatrix} s(0) & \cdots & s(-M+1) \\ s(1) & \cdots & s(-M+2) \\ \vdots & \ddots & \vdots \\ s(M+N-1) & \cdots & s(N-1) \end{pmatrix},$$

Since the channel is sparse, most components of the channel impulse response are zero. Through this knowledge, a sparse solution to $\mathbf{r} \approx \mathbf{Sh}$ is achieved by approximating

r as a linear combination of a small number of columns of **S** and will be explained in the following section.

2.2 Matching Pursuit (MP) Based Method

Mallat and Zhang [8] have proposed the use of a greedy algorithm which builds up a sequence of sparse approximations. The MP algorithm first finds the best fitted column, s_{k_1} in the matrix **S** with the received signal $r_0 = r$. Then the projection of the initial residual r_0 along with the vector s_{k_1} is removed from r_0 and the residual is r_1. Similarly, s_{k_2} is decided that is best aligned to the residual r_1. This algorithm keeps finding the best aligned column to the consecutive residuals until a specified number of taps or small residuals. After p iterations, one has a representation of the form equation (2–1), with residual r_p.

This algorithm is summarized as follows:

$$k_p = \arg\max \|P_{sl} r_{p-1}\| = \arg\max \frac{|s_l^H r_{p-1}|^2}{\|s_l\|^2} \tag{2–2}$$

$$l = 1, ..., N, l \neq k_{p-1},$$

$$r_p = r_{p-1} - P_{sk_p} r_{p-1} = r_{p-1} - \frac{(s_{k_p}^H r_{p-1}) s_{k_p}}{\|s_{k_p}\|^2} \tag{2–3}$$

$$\hat{h}_{k_p} = \frac{(s_{k_p}^H r_{p-1})}{\|s_{k_p}\|^2} \tag{2–4}$$

where the projection onto vector s_l is denoted as

$$P_{s_l} = \frac{s_l s_l^H}{\|s_l\|^2},$$

and the channel coefficient at position k_p is $\hat{h}_{k_p}$.

Possible problem of this algorithm is, the number of nonzero taps should be given which is unrealistic because it is unknown. This algorithm might spend most of its time correcting for any mistakes made in the first few terms because the algorithm is greedy.

One can give examples of dictionaries and signals where the method gives a solution which is badly sub-optimal in terms of sparsity.

2.3 SPARSE-LMS

An adaptive method using the augmented Lagrangian to find the sparse solution has been proposed by Rao et. al [16]. This method augmented the L_1-norm constraint of the channel impulse response as a penalty term to the MSE criterion. The cost function is

$$J(\mathbf{h}) = E(e_k^2) + \lambda \left[\sum_{i=1}^{L} |h_i| - \alpha \right] \tag{2–5}$$

Where the first term is the mean squared error between the received signals and the estimated signals, and the second term is the penalty term. This constraint fixes the L_1-norm of the channel impulse response $\mathbf{h}$ to a constant α. It is well known that the concavity of this L_1 norm function yields the sparse solution [13]. The penalty factor λ can be included as an adaptive parameter by modifying the cost function as,

$$J(\mathbf{h}, \lambda) = E(e_k^2) + \lambda\beta \left[\sum_{i=1}^{L} |h_i| - \alpha \right] - \lambda^2\beta \tag{2–6}$$

where β is a positive stabilization constant that keeps the penalty factor λ bounded. This modified cost function is known as the augmented Lagrangian [38]. The stochastic gradients of the cost function are given by,

$$\frac{\partial J(\mathbf{h}, \lambda)}{\partial h_i} = -2e_k u_{ki} + \lambda\beta[sign(h_i)] \tag{2–7}$$

$$\frac{\partial J(\mathbf{h}, \lambda)}{\partial \lambda} = \beta \left[\sum_{i=1}^{L} |h_i| - \alpha \right] - 2\beta\lambda \tag{2–8}$$

Then the adaptation rule of parameters is given by

$$h_i(k+1) = h_i(k) + \eta_h[2e_k u_{ki} - \lambda_k \beta sign(h_i)] \tag{2–9}$$

$$\lambda_{k+1} = \lambda_k + \eta_\lambda \beta \left[\sum_{i=1}^{L} |h_i| - \alpha - 2\lambda_k \right] \tag{2–10}$$

where η_h and η_λ are step size.

λ_k converges to a value λ^* [16]

$$\lambda^* = \frac{1}{2}\left(\sum_{i=1}^{L} |h_i^*| - \alpha\right) \tag{2–11}$$

where h^* is the asymptotic coefficient vector from equation (2–9).

Note that the constant denoted by α in equation (2–5) is unknown so it has to be searched cautiously to get a good performance. With an inappropriate choice of α, the performance of the adaptive system becomes poor. Therefore we propose a method based on the convolution inequality of entropy to determine α before learning system parameters using the stochastic gradient algorithm in equations (2–7) and (2–8).

2.3.1 Convolution Inequality For Renyi's Entropy

Recall the system model is

$$\mathbf{r} = \mathbf{S}\mathbf{h} + \mathbf{e} \tag{2–12}$$

where $\mathbf{h}$ is sparse channel.

Let $H_r(\cdot)$ denote the Renyi's entropy. The convolution inequality for Renyi's entropy is [18] [19]

$$H_r(\mathbf{r}) \geq H_r(\mathbf{s}) + \log|h_k|, \quad \forall k$$

$$\Updownarrow$$

$$e^{H_r(\mathbf{r}) - H_r(\mathbf{s})} \geq \max|h_k| \tag{2–13}$$

The equality holds if and only if the filter is a pure delay.

Let $h_{\max}$ denote the $\max|h_i|$, then we can start to search the adaptive parameter α in equation (2–5) using the above equation (2–13). If the vector $\mathbf{h}$ is sparse, then the L_1-norm of the channel impulse response will not be much bigger than $h_{\max}$. Instead of

searching α in an arbitrary range, we can simplify searching based on this information. Therefore, we start searching with $\alpha = h_{\max}$, and train the adaptive filter with the algorithm given by equations (2–7) and (2–8). After training, we compute MSE between the received signals $\mathbf{r}$ and the filter outputs. Then, we increase α and repeat learning. We can stop the training when the MSE start to increase. In Fig. 2-1, we can stop at $\alpha = (4 \cdot h_{\max})$.

Since we cannot work directly with the PDF, a nonparametric method is used here to estimate the entropy. Entropy estimate is obtained from Renyi's quadratic entropy estimator which estimates the PDF by Parzen-window method using Gaussian kernel [39], [40].

This proposed method can be summarized as follows:

Algorithm 1 Summary of sparse channel estimation with regularization method

Initial λ, β and h

Calculate input, output entropy

Set $\alpha = h_{max}$ from entropy inequality by Eq. (2–13)

while flag is true **do**

  Train adaptive filter by Eq. (2–9), (2–10)

  Compute MSE and check if it's reached minimum.

  If minimum, set flag=false.

  If not minimum, increase α

end while

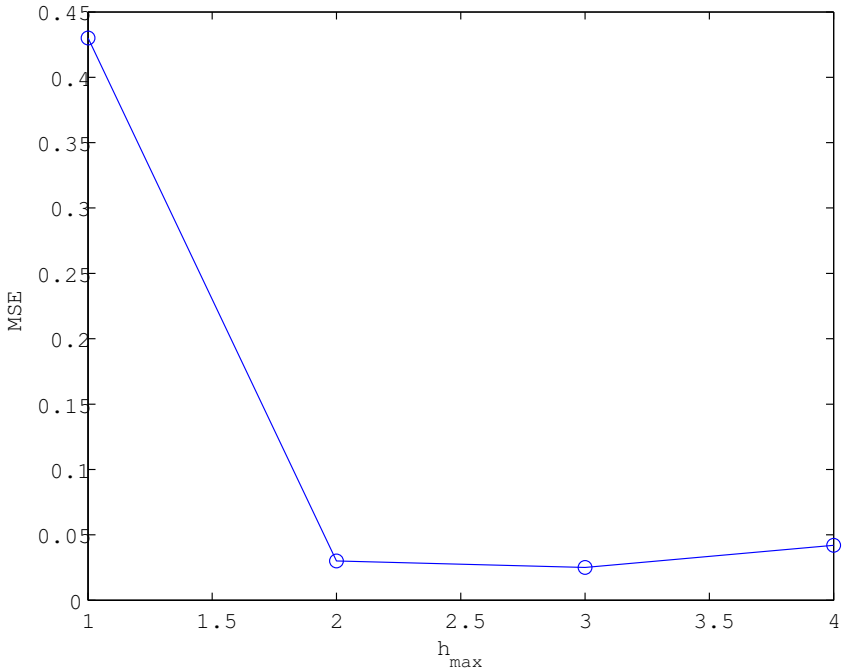


Figure 2-1. Searching α parameter using the proposed method based on convolution inequality for entropy.

2.4 Simulation Results

Nonzero coefficients of the sparse channels are drawn from a uniform distribution on $[-1, -0.2] \cup [0.2, 1]$ and the number of the nonzero coefficients is 8 which is 10% of the channel length, $M = 80$. These nonzero coefficients are positioned randomly over 80 taps. A white Gaussian training data with zero mean and unit variance is transmitted over the sparse channel and a white Gaussian noise with zero mean and variance of σ_n^2 is added to the received data. Variance σ_n^2 is varied to change the SNR from 5dB to 30dB. The number of training data is $N = 200$. 1000 Monte Carlo simulations with different input signals, SNR, and the nonzero channel coefficients are performed.

Table 2-1 shows the comparison of the proposed criterion performance to the Wiener's MSE criterion. We searched the α using the proposed method based on convolution inequality and fix the value of α to $(3 \cdot h_{\max})$ from Fig. 2-1. Here, the estimated α value is 5.5995 which is very close to the value of L_1-norm of true vector $\mathbf{h}$, 5.6242. As can be

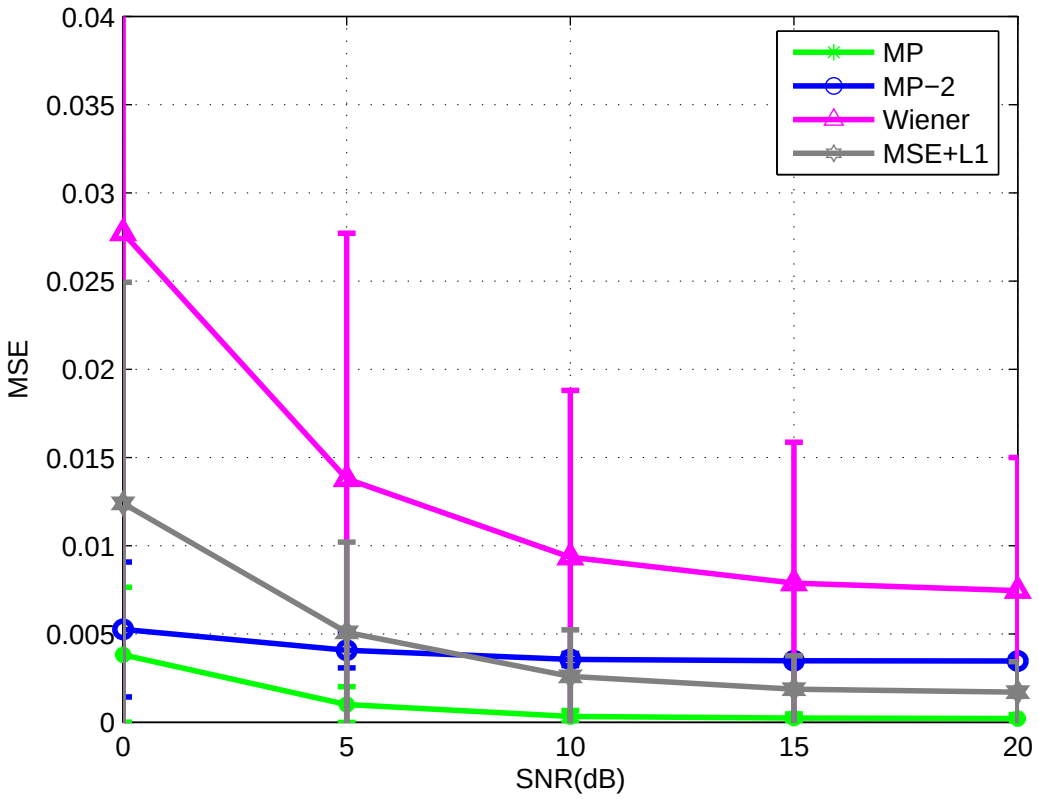


Figure 2-2. Performance comparison of proposed method with MP, MP-2 and Wiener

Table 2-1. Difference between the true weights ($\mathbf{h}$) and estimated weights ($\hat{\mathbf{h}}$) with the parameters ($\mu_h = 0.05, \mu_\lambda = 0.01, \beta = 1$) when $SNR = 10dB$.

	overall tap difference	nonzero tap difference	zero tap difference
$\mid \mathbf{h} - \hat{\mathbf{h}}_{wiener} \mid$	0.6960	0.1746	0.6737
$\mid \mathbf{h} - \hat{\mathbf{h}}_{mse+L_1} \mid$	0.2862	0.2425	0.1520

seen in Table 2-1, the proposed method estimates the zero tap coefficients better than Wiener's MSE solution but there is a penalty for this. MSE is increased in the nonzero tap coefficients estimation which can be explained by the performance loss due to the penalty term.

We also compared the performance of the proposed method to Wiener filter and matching pursuit algorithm which is widely used in sparse application. Fig. 2-2 shows that MP method outperforms other methods in terms of the variance and MSE of channel impulse response estimates. In this simulation, MP method knows the number of nonzero channel but in practical the number of nonzeros taps are unknown. So we run the MP method with incorrect number of nonzero taps (here 6) and denote as 'MP-2'. We can see from Fig. 2-2 that the performance of the proposed method is much better than Wiener and also better than 'MP-2' from 10dB. The MP method outperforms other methods but the MP algorithm is greedy and needs to know the exact number of nonzero channel coefficients while the proposed algorithm is an adaptive method also performs better than MP-2 from 10dB.

CHAPTER 3
SUPPORT VECTOR MACHINE BASED METHOD

3.1 Introduction

Support Vector Machines (SVM) proposed by V. Vapnik are a new method to solve pattern recognition problems which are based on the Structural Risk Minimization (SRM) [32]. While the Empirical Risk Minimization (ERM) principle is used by neural networks to minimize the error on the training data, SRM minimizes a bound on the test error.

In the linear separable case, SVM maximizes the margin of the hyperplane which classifies the input data. The hyperplane location is determined by some points of the input data which are termed support vectors. However, SVMs have also been proposed and applied to a number of different type of problems such as regression problem [41, 42], detection [43] and inverse problems [44]. In regression, the goal of SVM is to construct a hyperplane that lies close to as many of the data points as possible. We must choose a hyperplane with small norm while minimizing the loss using ϵ-insensitive loss function. A remarkable property of SVM is the sparsity of its solution. Typically a small number of support vectors are nonzero. Girosi showed the equivalence between sparse approximation and SVMs [33].

In this chapter, support vector machines are briefly introduced including the classification and the regression problem. Then we present a new SVM-based sparse system identification algorithm. SVM parameter selection is also discussed. For additional material of SVMs one can refer to the works of V. Vapnik, C. Burges, B. Schölkopf and A. Smola [32, 45].

3.2 Support Vector Classification

3.2.1 Linearly Separable Patterns

Consider the training sample $(\mathbf{x}_i, y_i)_{i+1}^N$, where $\mathbf{x}_i$ is the input pattern for the ith example and d_i is the corresponding desired response and which are linearly separable. The equation of a decision surface in the form of a hyperplane that does the separation is

$$\mathbf{w}^T\mathbf{x} + b = 0 \tag{3–1}$$

For a given vector $\mathbf{w}$ and bias b, the separation between the hyperplane defined in equation (3–1) and the closest data point is called the margin of separation. The goal of a support vector machine is to find the particular hyperplane for which the margin of separation is maximized. This can be formulated as follows: suppose that all the training data satisfy the constraints

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq 1, \quad \texttt{for } y_i = 1 \tag{3–2}$$

$$\mathbf{w} \cdot \mathbf{x}_i + b \leq 1, \quad \texttt{for } y_i = -1 \tag{3–3}$$

where $\mathbf{w}$ is normal to the hyperplane, $\frac{|b|}{\|\mathbf{w}\|}$ is the perpendicular distance from the hyperplane to the origin and $\|\mathbf{w}\|$ is the Euclidean norm of $\mathbf{w}$ (Fig. 3-1). These constraints can be combined into

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0 \quad \forall i \tag{3–4}$$

The points for which the equality (3–2) holds lie on the hyperplane $H_1 : \mathbf{w} \cdot \mathbf{x}_i + b = 1$ with normal $\mathbf{w}$ and perpendicular distance from the origin $\frac{|1-b|}{\|\mathbf{w}\|}$. The points for which the equality (3–3) holds lie on the hyperplane $H_2 : \mathbf{w} \cdot \mathbf{x}_i + b = -1$, with normal $\mathbf{w}$ and perpendicular distance from the origin $\frac{|-1-b|}{\|\mathbf{w}\|}$. Therefore the margin is

$$\frac{|1 - b + 1 + b|}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|} \tag{3–5}$$

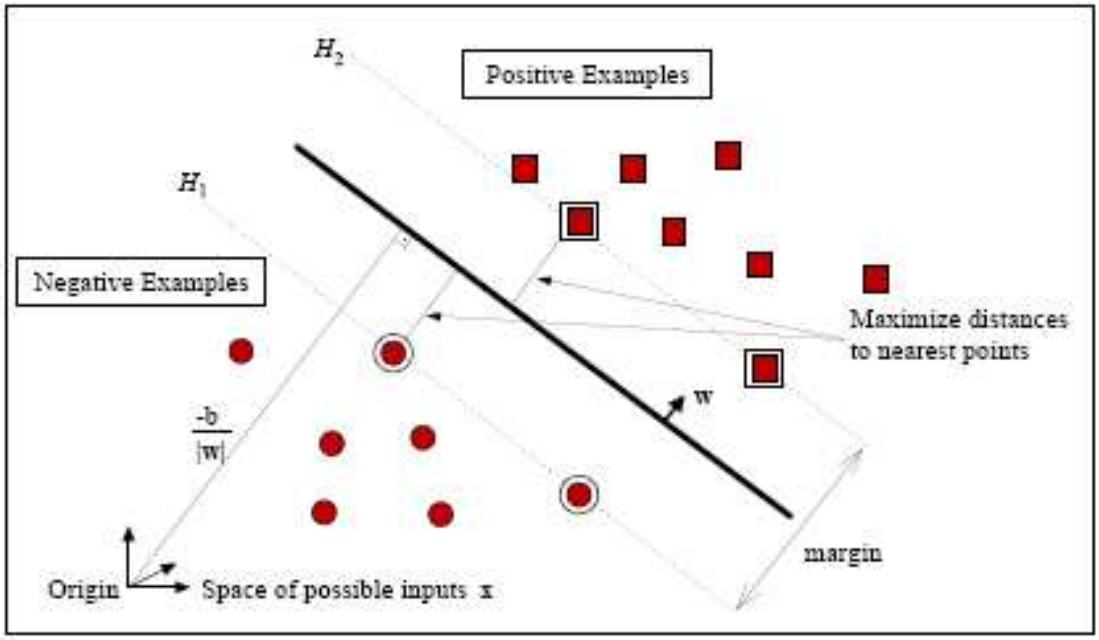


Figure 3-1. Illustration of the idea of an optimal hyperplane for linearly separable patterns. The support vectors are outlined.(adapted from [2])

Equation (3–5) states that maximizing the margin of separation between classes is equivalent to minimizing the Euclidean norm of the weight vector $\mathbf{w}$.

The optimization problem finds the optimal margin hyperplane by

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \tag{3–6}$$

subject to equation (3–4). By introducing positive Lagrange multipliers $\alpha_i,\ i = 1, 2, \ldots, l$, the primal formulation of the problem is

$$L_p \equiv \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i y_i(\mathbf{w} \cdot \mathbf{x}_i + b) + \sum_{i=1}^{l} \alpha_i, \quad \alpha_i \geq 0 \tag{3–7}$$

L_p must be minimized with respect to $\mathbf{w}, b$ subject to the constraints $\alpha_i \geq 0$. This is a convex quadratic programming (QP) problem and we can use the dual which is to

maximize L_p with respect to α_i, subject to the constraints

$$\frac{\partial L_p}{\partial \mathbf{w}} = \mathbf{0} \tag{3–8}$$

$$\frac{\partial L_p}{\partial b} = 0 \tag{3–9}$$

which gives the conditions

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \tag{3–10}$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{3–11}$$

Substituting these conditions into equation (3–7) gives the dual formulation of the Lagrangian

$$L_D \equiv \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{3–12}$$

subject to

$$\sum_{i=1}^{N} \alpha_i y_i = 0$$

$$\alpha_i \geq 0 \quad \texttt{for}\ i = 1, 2, \ldots, N$$

New data $\mathbf{x}$ can be classified using

$$f(x) = \mathbf{w} \cdot \mathbf{x} + b \tag{3–13}$$

$$= \left(\sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i \right) \cdot + b \tag{3–14}$$

$$= \sum_{i=1}^{l} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}) + b \tag{3–15}$$

The solution can be found by minimizing L_p or maximizing L_D and can be efficiently solved by [46].

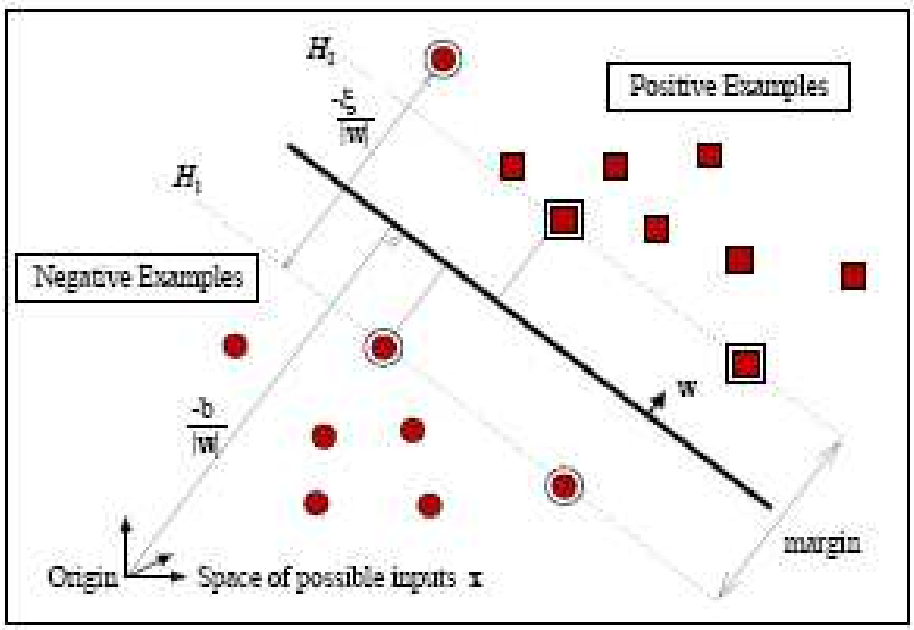


Figure 3-2. Hyperplane for the non-separable case. The slack variable ξ permits margin failure. Data point is wrong side of the decision surface. (adapted from [2])

3.2.2 Nonseparable Patterns

Consider the case where some data points $(\mathbf{x}_i, y_i)$ fall on the wrong side of the decision surface as illustrated in Fig. 3-2. To treat this nonlinearly separable data cases, we need to introduce positive slack variables $\xi_i, i = 1, 2, \ldots, l$ in the constraints. The constraints (3–2) and (3–3) then become:

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq 1 - \xi_i, \quad \texttt{for } y_i = 1 \tag{3–16}$$

$$\mathbf{w} \cdot \mathbf{x}_i + b \leq 1 + \xi_i, \quad \texttt{for } y_i = -1 \tag{3–17}$$

$$\xi_i \geq 0, \quad \forall\, i \tag{3–18}$$

To assign penalties for errors, the objective function (3–6) is modified to

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i \tag{3–19}$$

Minimizing the first term in equation (3–19) is related to minimizing the VC dimension of the support vector machine. As for the second term $\sum_i \xi_i$, it is an upper bound on

the number of test errors. The parameter C controls the tradeoff between complexity of the machine and the number of nonseparable points. It may be viewed as a form of a regularization parameters. A larger C corresponds to assigning a higher penalty to errors. the dual problem for nonseparable patterns becomes:

$$L_D \equiv \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{3–20}$$

subject to

$$\sum_{i=1}^{N} \alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq C$$

The nonseparable case differs from the separable case in that the constraint $\alpha_i \geq 0$ is replaced with the more stringent constraint $0 \leq \alpha_i \leq C$. The solution is again given by

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \tag{3–21}$$

where N is the number of support vectors.

3.2.3 The Karush-Kuhn-Tucker Conditions

For the primal problem equation (3–6) the KKT conditions are [47]

$$\frac{\partial}{\partial w_v} L_p = w_v - \sum_{i=1}^{l} \alpha_i y_i x_{iv} = 0 \quad v = 1, 2, \ldots, d \tag{3–22}$$

$$\frac{\partial}{\partial b} L_p = -\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{3–23}$$

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0 \quad i = 1, 2, \ldots, l \tag{3–24}$$

$$\alpha_i \geq 0 \quad \forall i \tag{3–25}$$

$$\alpha\{y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1\} = 0 \quad \forall i \tag{3–26}$$

where v runs from 1 to the dimension d of the data. The equation (3–22) to equation (3–26) are satisfied at the solution of any constrained optimization problem, provided that the

intersection of the set of feasible directions with the set of descent directions coincides with the intersection of the set of feasible directions for linearized constraints with the set of descent directions [47, 48]. This regularity assumption holds for all support vector machines, since the constraints are always linear. In addition, the problem for SVMs is convex and for convex problems the KKT conditions are necessary and sufficient for $\mathbf{w}, b, \alpha$ to be a solution [47]. Thus, finding a solution to the KKT conditions is equivalent to solving the SVM problem. The threshold b is found by using the KKT condition, equation (3–26), by choosing any i where $\alpha_i \neq 0$ (i.e. support vectors) and computing b. The KKT conditions for the primal problem are also used in the non-separable case. The primal Lagrangian is

$$L_p = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i - \sum_i \alpha_i\{y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i\} - \sum_i \mu_i\xi_i \tag{3–27}$$

where μ_i are the Lagrange multipliers introduced to enforce positivity of the slack variables ξ_i. The KKT conditions for the primal problem are

$$\frac{\partial L_p}{\partial w_v} = w_v - \sum_{i=1}^{l} \alpha_i y_i x_{iv} = 0 \quad v = 1, 2, \ldots, d \tag{3–28}$$

$$\frac{\partial L_p}{\partial b} = -\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{3–29}$$

$$\frac{\partial L_p}{\partial \xi_i} = C - \alpha_i - \mu_i = 0 \tag{3–30}$$

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i \geq 0 \tag{3–31}$$

$$\alpha_i, \xi_i, \mu_i \geq 0 \tag{3–32}$$

$$\alpha\{y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i\} = 0 \tag{3–33}$$

$$\xi_i\mu_i = 0 \tag{3–34}$$

where $i = 1, 2, \ldots, l$ and $v = 1, 2, \ldots, d$.

The KKT complementary conditions equation (3–33) and equation (3–34) can be used to determine the b. Any training point for which $0 < \alpha_i < C$, that is not penalized can be taken to compute b.

3.3 Support Vector Regression

The structural risk minimization (SRM) principle is a criterion that establishes a trade-off between the complexity of the solution and the closeness to the data. We apply the SRM principle to the sparse channel estimation problem. In the proposed formulation, the channel coefficients play the role of the Lagrange multipliers. To solve this sparse channel estimation, we will next reformulate it as a set of support vector regression (SVR) problem.

Consider the problem of approximating the set of data $\{(\mathbf{x}_i, y_i)_{i=1}^{l}\}$ with a linear function $f(\mathbf{x})$.

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b \tag{3–35}$$

w_i, b are parameters to be estimated from the data. The method of SVM regression corresponds to the following minimization:

$$J(\mathbf{w}) = C \sum_{i=1}^{l} |y_i - f(\mathbf{x}_i)|_\epsilon + \frac{1}{2} \parallel \mathbf{w} \parallel^2 \tag{3–36}$$

The parameter C controls the tradeoff between training error and regularization terms in equation (3–36) and the ϵ-insensitive loss function is defined as

$$V(x) = |x|_\epsilon \equiv \begin{cases} 0 & \text{if } |x| < \epsilon \\ |x| - \epsilon & \text{otherwise} \end{cases} \tag{3–37}$$

Since it is difficult to solve equation (3–36) with the ϵ-insensitive loss function $|x|_\epsilon$, the problem is replaced by following minimization problem.

$$C\sum_{i=1}^{l}(\xi_i+\xi_i^*)+\frac{1}{2}\parallel \mathbf{w} \parallel^2 \tag{3–38}$$

subject to

$$\begin{aligned}
f(\mathbf{x}_i)-y_i &\leq \epsilon+\xi_i && i=1,\ldots,l\\
y_i-f(\mathbf{x}_i) &\leq \epsilon+\xi_i^* && i=1,\ldots,l\\
\xi_i &\geq 0 && i=1,\ldots,l\\
\xi_i^* &\geq 0 && i=1,\ldots,l
\end{aligned}$$

Notice that the penalty is paid only when the absolute value of the interpolation error exceeds ϵ. To solve the above constrained minimization problem, we use Lagrange multipliers. The Lagrangian corresponding to equation (3–38) is:

$$\begin{aligned}
L(f,\xi,\xi^*;\alpha,\alpha^*,\gamma,\gamma^*) = C\sum_{i=1}^{l}(\xi_i+\xi_i^*)+\frac{1}{2}\mathbf{w}^T\mathbf{w}+\sum_{i=1}^{l}\alpha_i^*(y_i-f(\mathbf{x}_i)-\epsilon-\xi_i^*)+\\
+\sum_{i=1}^{l}\alpha_i(f(\mathbf{x}_i)-y_i-\epsilon-\xi_i)-\sum_{i=1}^{l}(\gamma_i\xi_i+\gamma_i^*\xi_i^*)
\end{aligned} \tag{3–39}$$

The solution is given by minimizing the Lagrangian equation (3–39) with respect to f(that is w.r.t w,b), ξ, ξ^* and maximizing with respect to $\alpha,\alpha^*,\gamma,\gamma^*$.

$$\frac{\partial L}{\partial w}=0 \quad\Rightarrow\quad \mathbf{w}=\sum_{i=1}^{l}(\alpha_i^*-\alpha_i)\mathbf{x}_i \tag{3–40}$$

$$\frac{\partial L}{\partial b}=0 \quad\Rightarrow\quad \sum_{i=1}^{l}(\alpha_i^*-\alpha_i)\mathbf{x}_i=0 \tag{3–41}$$

$$\frac{\partial L}{\partial \xi_i}=0 \quad\Rightarrow\quad \gamma_i=C-\alpha_i \tag{3–42}$$

$$\frac{\partial L}{\partial \xi_i^*}=0 \quad\Rightarrow\quad \gamma_i^*=C-\alpha_i^* \tag{3–43}$$

Substituting w in equation (3–35) with equation (3–40), we can express the problem (3–38) as belows

$$f(\mathbf{x}, \alpha, \alpha^*) = \sum_{i=1}^{l} (\alpha_i^* - \alpha_i) < \mathbf{x}; \mathbf{x}_i > +b \tag{3–44}$$

In this dual representation equation (3–44), data appears only in the dot product. Substituting equation (3–44) in the Lagrangian eq (3–39), we obtain a maximization problem with respect to $\alpha, \alpha^*, \gamma, \gamma^*$, where α_i^* and α_i are positive Lagrange multipliers which solve the following Quadratic Programming(QP) problem in dual representation:

$$\begin{aligned} \max_{\alpha,\alpha^*} L(\alpha, \alpha^*) = &- \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) \langle \mathbf{x}_i, \mathbf{x}_j \rangle - \epsilon \sum_{i=1}^{l} (\alpha_i^* + \alpha_i) + \\ &+ \sum_{i=1}^{l} y_i (\alpha_i^* - \alpha_i) \end{aligned} \tag{3–45}$$

subject to the constraints

$$0 \leq \alpha^*, \alpha \leq C$$

$$\sum\nolimits_{i=1}^{l} (\alpha_i^* - \alpha_i) = 0$$

This equation (3–45) is the QP problem which is convex and having no local minimum. This has to be solved to compute the SVM. Due to the nature of the QP problem, only a small number of coefficients $\alpha_i^* - \alpha_i$ will be nonzero and the associated input data are called support vectors. Interpolation error of the support vectors is either greater or equal to ϵ so if $\epsilon = 0$ then all the input becomes support vectors.

3.4 Sparse Channel Estimation using SVM

The system model is

$$\mathbf{r} = \mathbf{S}\mathbf{h} \tag{3–46}$$

where $\mathbf{r}$ is received vector, and $\mathbf{S}$ is input matrix. Premultiply $\mathbf{S}^T$ on both sides yields

$$\underbrace{\mathbf{S}^T\mathbf{r}}_{\mathbf{y}} = \mathbf{S}^T \underbrace{\mathbf{S}\mathbf{h}}_{\mathbf{x}}$$
$$\Downarrow$$
$$\mathbf{y} = \mathbf{S}^T\mathbf{x}$$

where $\mathbf{y}$ is a new output vector, and $\mathbf{S}^T$ is a new input matrix.

The SVM minimize the primal problem

$$J(\mathbf{x}) = C\sum_{n=1}^{M} |y_n - \mathbf{x}^T\mathbf{s}_n|_\epsilon + \frac{1}{2}\mathbf{x}^T\mathbf{x}$$

or maximize the dual.

$$\begin{aligned} L = &-\frac{1}{2}\sum_{n=1}^{M}\sum_{m=1}^{M}(\alpha_n^* - \alpha_n)(\alpha_m^* - \alpha_m) < \mathbf{s}_n, \mathbf{s}_m > \\ &- \epsilon\sum_{n=1}^{M}(\alpha_n^* + \alpha_n) + \sum_{n=1}^{M} y_n(\alpha_n^* - \alpha_n) \end{aligned}$$

The solution is

$$\begin{aligned} \mathbf{x} &= \sum_{n=1}^{M}(\alpha_n^* - \alpha_n)\mathbf{s}_n \\ &\triangleq \mathbf{S}\mathbf{h} \end{aligned} \tag{3–47}$$

The Lagrange multipliers $(\alpha_n^* - \alpha_n)$ in the SVM solution of equation (3–47) correspond to channel coefficients $\mathbf{h}$ in equation (3–46), yielding a sparse solution.

3.5 Extension to Complex Channel

To handle the complex-valued channel and data, we use the isomorphism between complex numbers and real-valued matrices. Consider the system model

$$\mathbf{r} = \mathbf{S}\mathbf{h}$$

where $\mathbf{S}$ is a complex-valued matrix of dimension $m \times n$, $\mathbf{h}$ is a complex-valued vector of length n and $\mathbf{r}$ is a complex-valued vector of dimension m. This system can be equivalently expressed as follows:

$$\begin{bmatrix} \bar{\mathbf{r}} \\ \tilde{\mathbf{r}} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{S}} & -\tilde{\mathbf{S}} \\ \bar{\mathbf{S}} & \tilde{\mathbf{S}} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{h}} \\ \tilde{\mathbf{h}} \end{bmatrix} \tag{3–48}$$

where all involved matrices and vectors are real-valued (the notation $\bar{(\cdot)}$ is $\mathrm{Re}(\cdot)$ and $\tilde{(\cdot)}$ is $\mathrm{Im}(\cdot)$). Let us introduce the following notation

$$\acute{\mathbf{r}} = \begin{bmatrix} \bar{\mathbf{r}} \\ \tilde{\mathbf{r}} \end{bmatrix}, \acute{\mathbf{h}} = \begin{bmatrix} \bar{\mathbf{h}} \\ \tilde{\mathbf{h}} \end{bmatrix}$$

$$\acute{\mathbf{S}} = \begin{bmatrix} \bar{\mathbf{S}} & -\tilde{\mathbf{S}} \\ \bar{\mathbf{S}} & \tilde{\mathbf{S}} \end{bmatrix}$$

Then we can write equation (3–48) as:

$$\acute{\mathbf{r}} = \acute{\mathbf{S}}\acute{\mathbf{h}}$$

The representation in equation (3–48) is useful to extend the complex-valued number to real-valued matrices.

3.6 SVM Parameter Selection

Recall that SVM minimizes

$$J(\mathbf{x}) = C \sum_{n=1}^{M} |y_n - \mathbf{x}^T \mathbf{s}_n|_\epsilon + \frac{1}{2}\mathbf{x}^T\mathbf{x}$$

It is well known that SVM performance (estimation accuracy) depends on a good setting of meta-parameters C and ϵ. Optimal selection of these parameters are difficult because SVM model performance depends on all these parameters.

The parameter C determines the trade off between the model complexity and the degree to which deviations larger than ϵ are tolerated. For example, if C is too large, then the objective is to minimize the empirical risk only, not considering the model complexity part in the optimization formulation equation (3–36).

The parameter ϵ controls the width of the ϵ-insensitive zone, used to fit the training data. The value of ϵ can affect the number of support vectors used to construct the regression function. The bigger ϵ, the fewer support vectors are selected. Bigger ϵ values result in more flat estimates. Therefore, both C and ϵ affect model complexity but in different way.

Some practical approaches to the choice of C and ϵ are as follows:

- Selecting parameter C equal to the range of output values [49].
- Choose ϵ so that the percentage of support vectors in the SVM model is around 50%
- Optimal ϵ values are proportional to noise variance [50].

To investigate the effects of parameter C and ϵ via simulation, we checked the channel estimation error, number of selected support vectors, training error and test error with various ϵ and C values at $15dB$ and $25dB$. Sparse channel data for this simulations are the HDTV channel data from ATTC (Advanced Television Test Center) tests [3]. Dependence of MSE (Mean Squared Error) of channel estimation as a function of chosen C and ϵ values for ATTC channel D data set at $15dB$ is shown in Fig. 3-3. Fig. 3-4 also shows the

dependence of MSE of channel estimation error as a function of chosen C and ϵ values for ATTC channel D data set at $25dB$.

One can clearly see that C values above certain threshold have only minor effect on the MSE of channel estimation. According to the simulation results, suggested C is 15 and suggested ϵ value is 1.5 for Fig. 3-3 and Fig. 3-4,

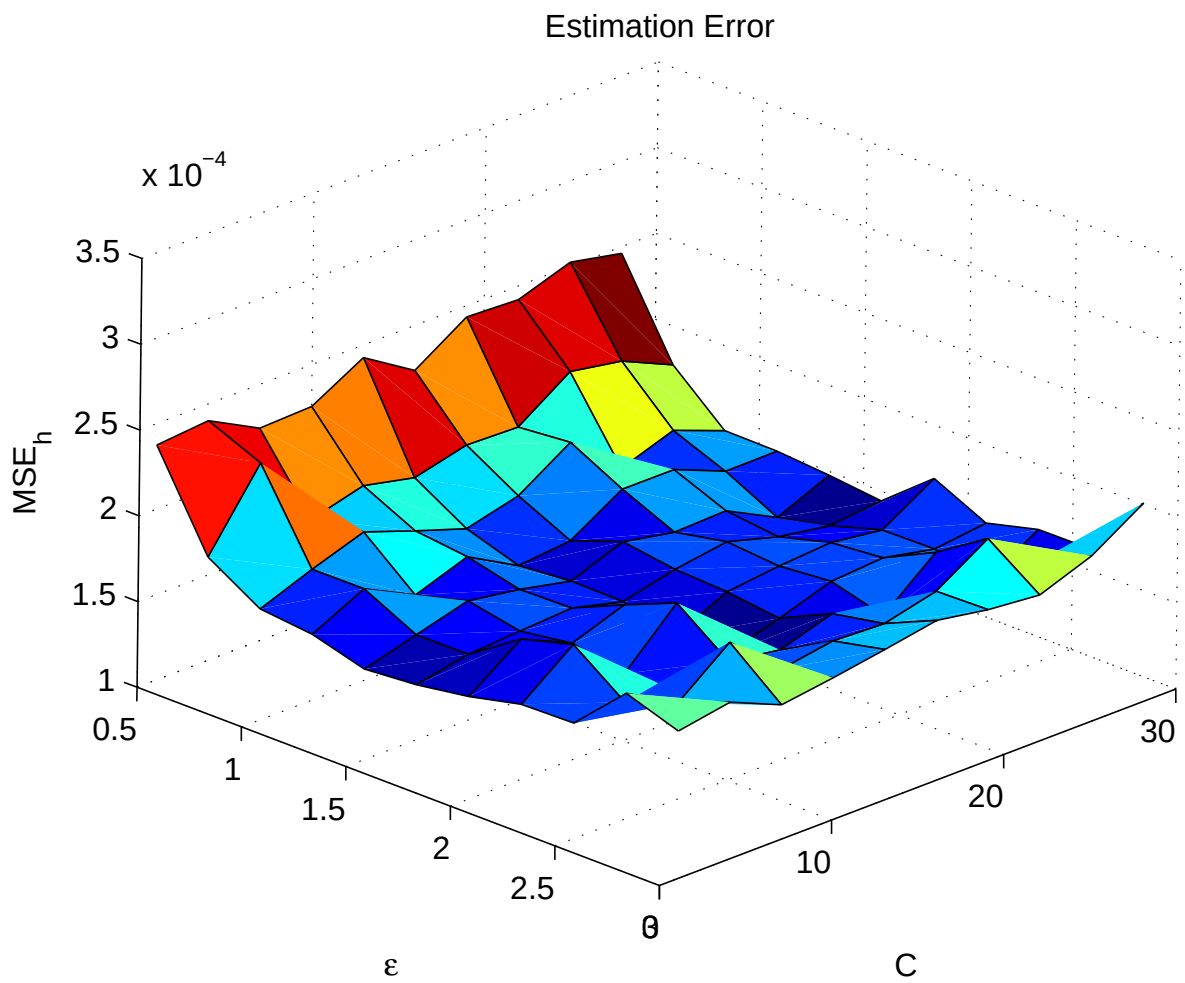


Figure 3-3. MSE of channel estimation as a function of C and ϵ values for HDTV sparse channel D data from ATTC (SNR = 15dB, 200 training samples, suggested values of $\epsilon \approx 1.5$ and $C \approx 15$)

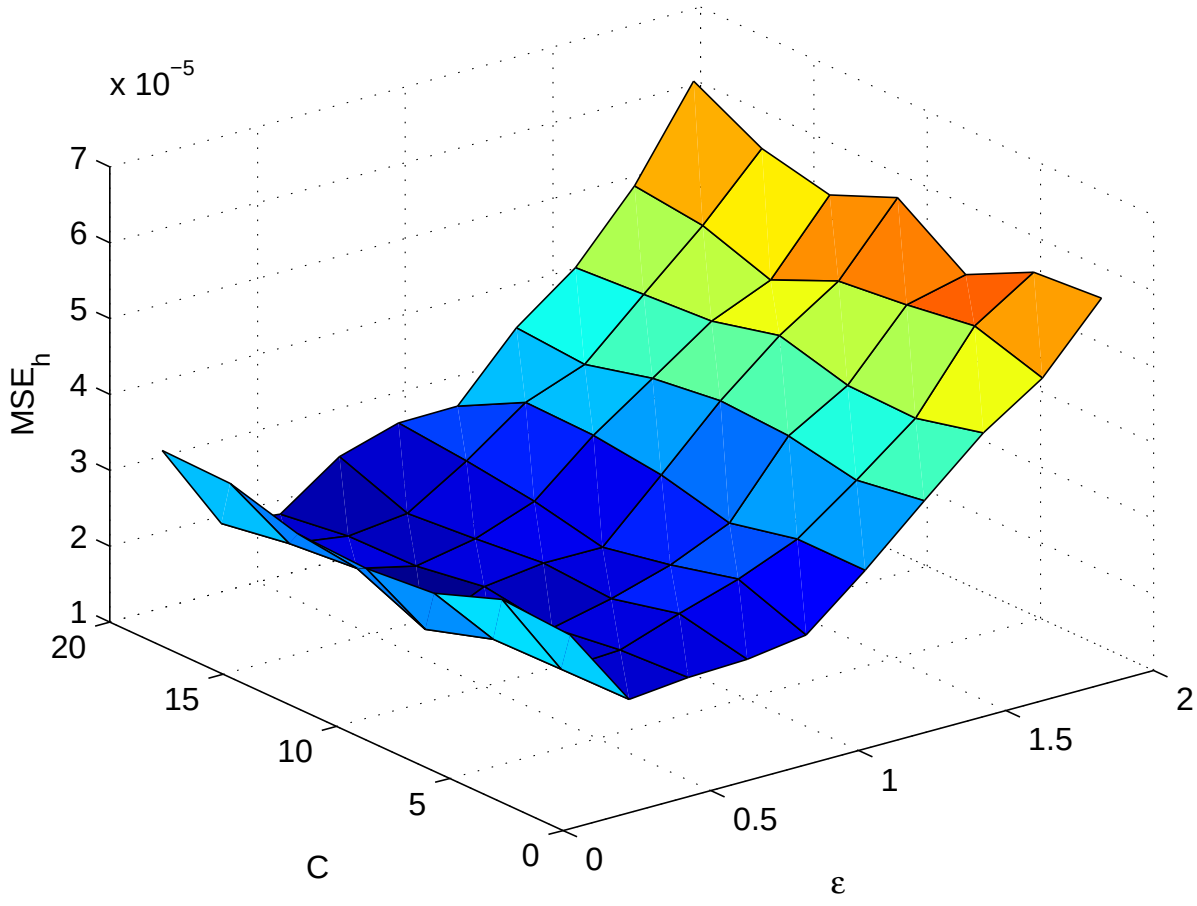


Figure 3-4. MSE of channel estimation as a function of C and ϵ values for HDTV sparse channel D data from ATTC (SNR = 25dB, 200 training samples, suggested values of $\epsilon \approx 0.2$)

Fig. 3-5 shows the number of support vectors selected by chosen C and ϵ values for ATTC channel D data set with 200 training samples at $15dB$ and Fig. 3-6 is for $25dB$. We can see that small ϵ values correspond to higher number of support vectors, whereas parameter C has negligible effect on the number of support vectors. Fig. 3-7 and Fig. 3-7 shows the dependence of MSE as a function of ϵ values more clearly at $15dB$ and $25dB$.

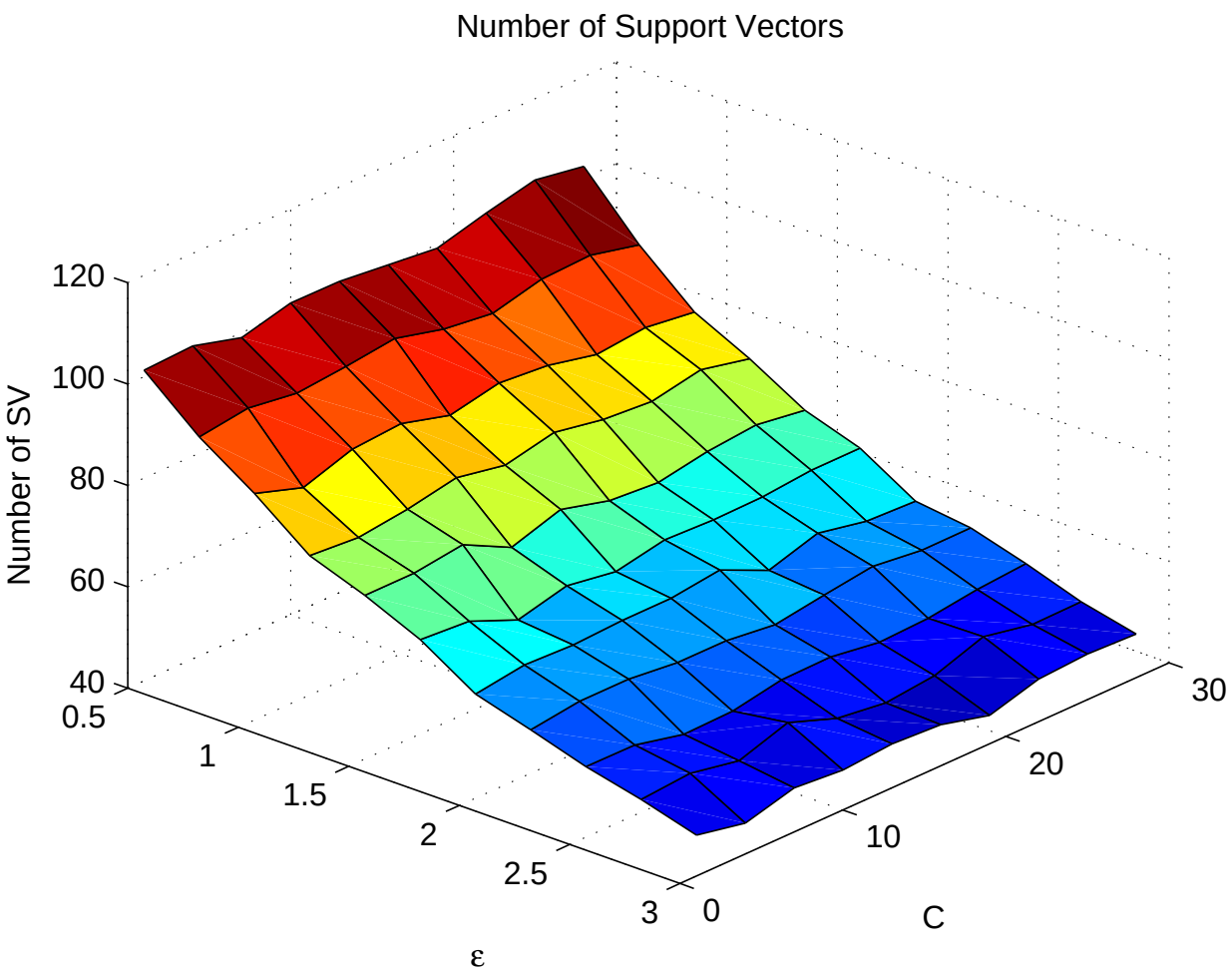


Figure 3-5. Number of support vectors as a function of C and ϵ values for HDTV sparse channel D data from ATTC(SNR=15dB, 200 training samples)

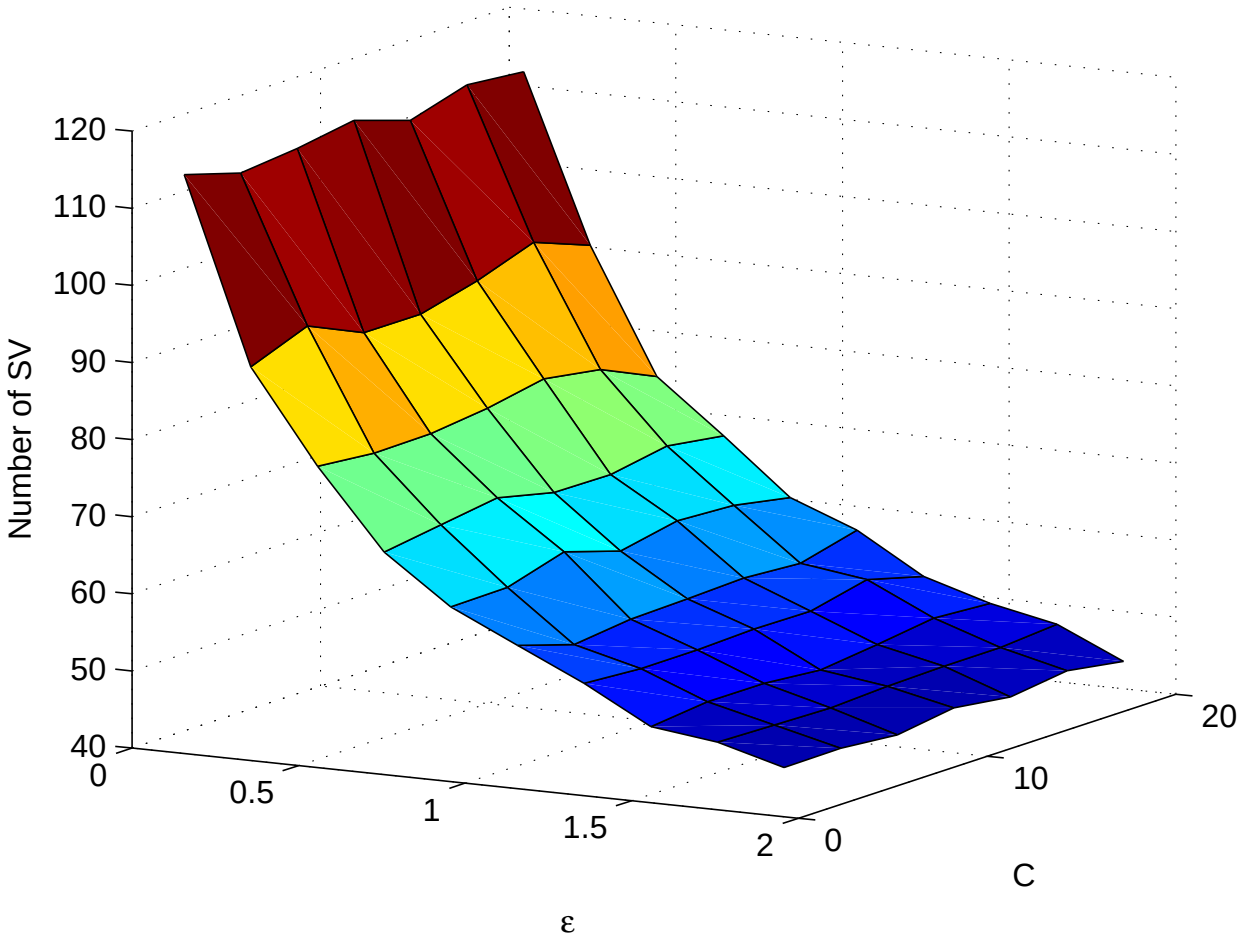
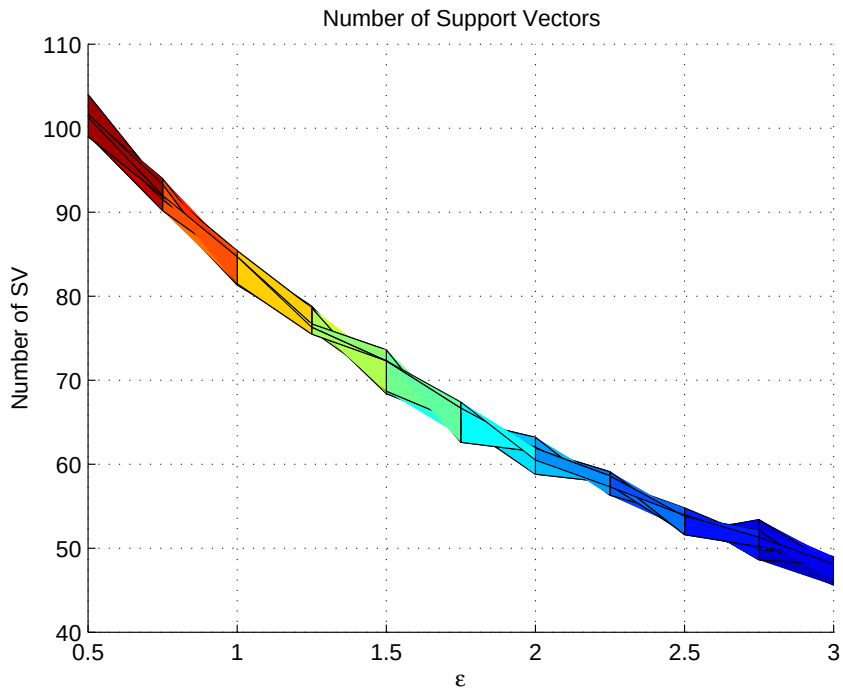
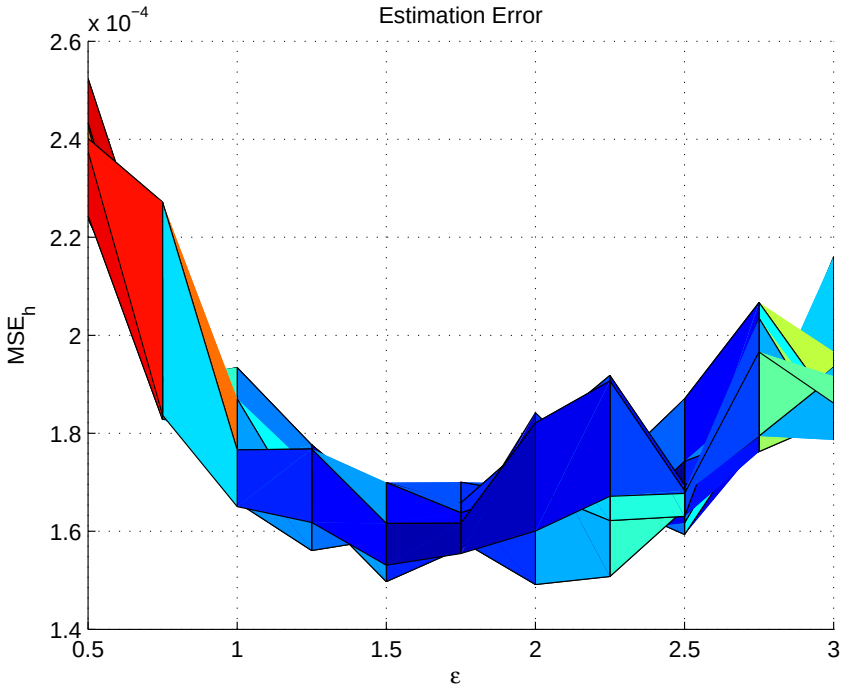


Figure 3-6. Number of support vectors as a function of C and ϵ values for HDTV sparse channel D data from ATTC(SNR=25dB, 200 training samples)

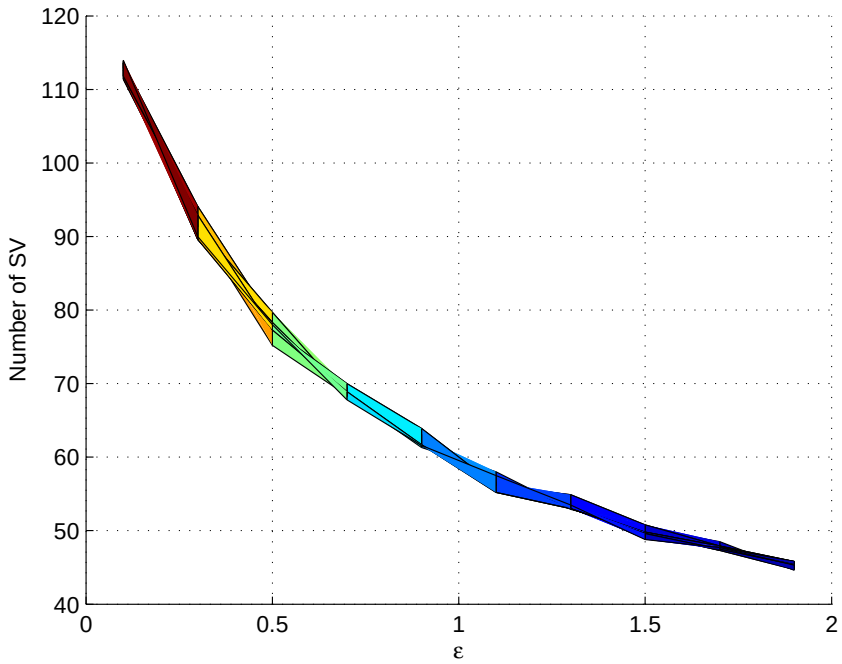


(a) Number of support vectors

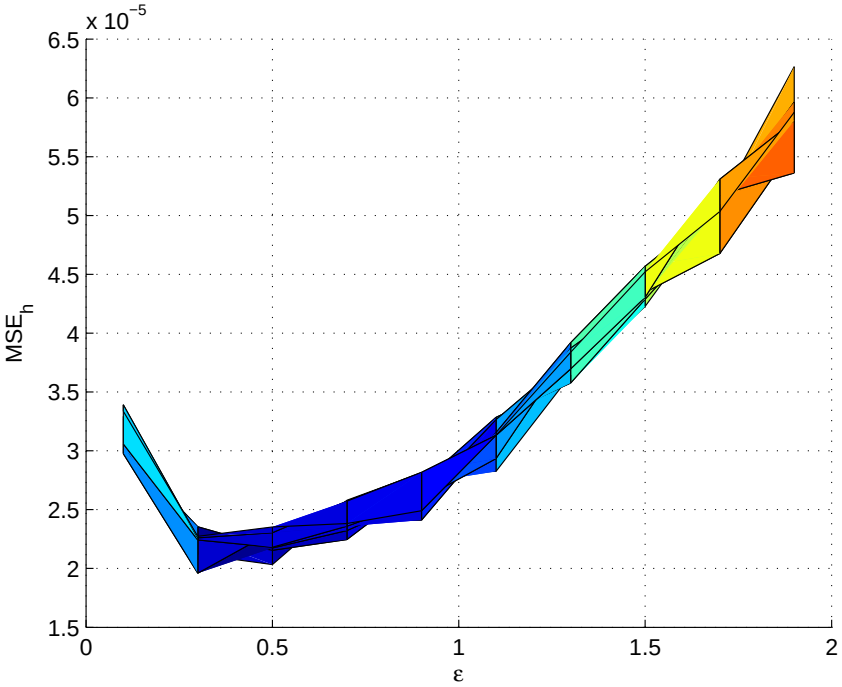


(b) Channel estimation error

Figure 3-7. Selection of ϵ when SNR=15dB



(a) Number of support vectors



(b) Channel estimation error

Figure 3-8. Selection of ϵ when SNR=25dB

Fig. 3-9 and Fig. 3-10 shows the performance in terms of training error as a function of C and ϵ values. We can see that training error increases as ϵ values increases and parameter C has negligible effect on the number of support vectors.

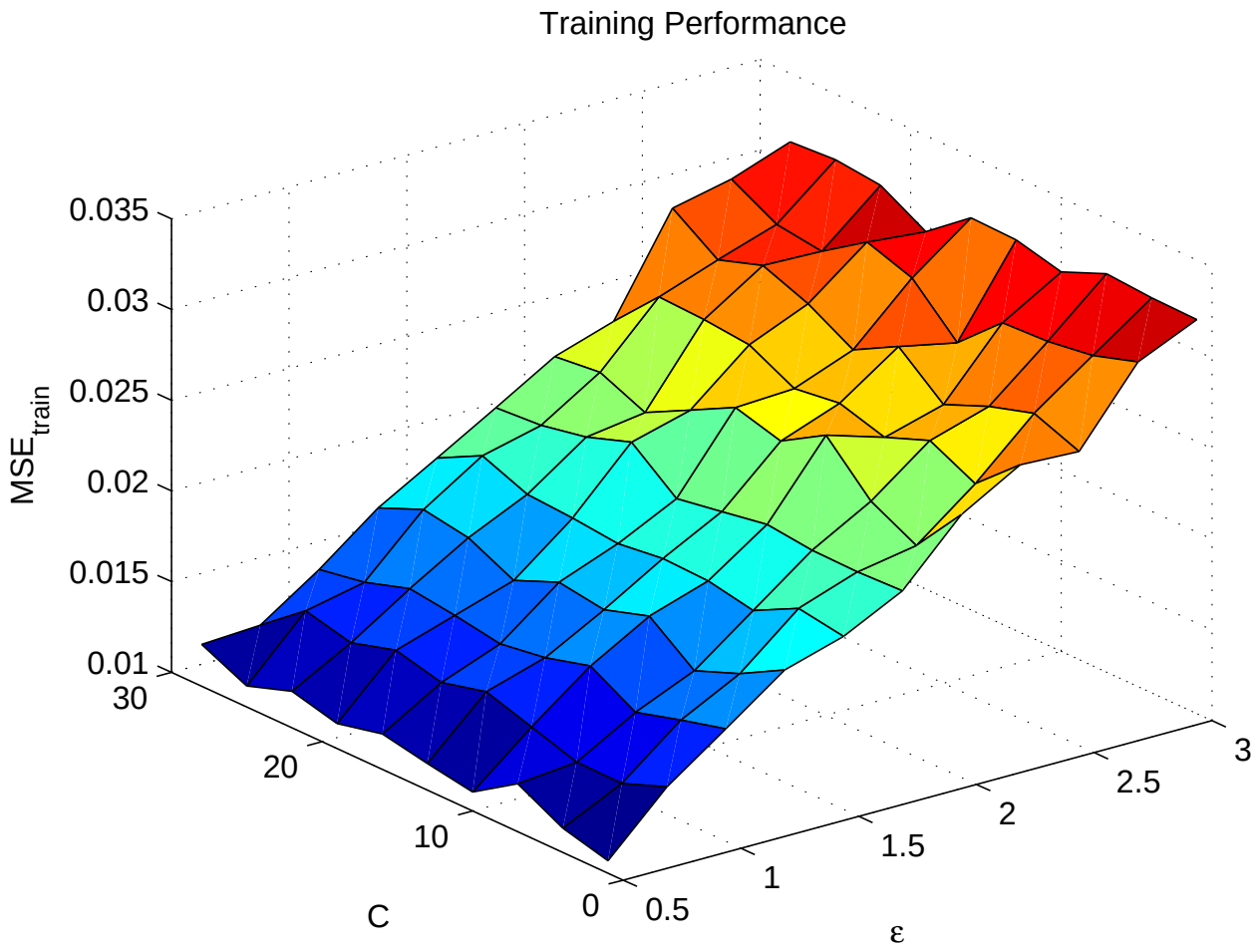


Figure 3-9. Training Performance as a function of C and ϵ values for HDTV sparse channel D data from ATTC (SNR=15dB, 200 training samples)

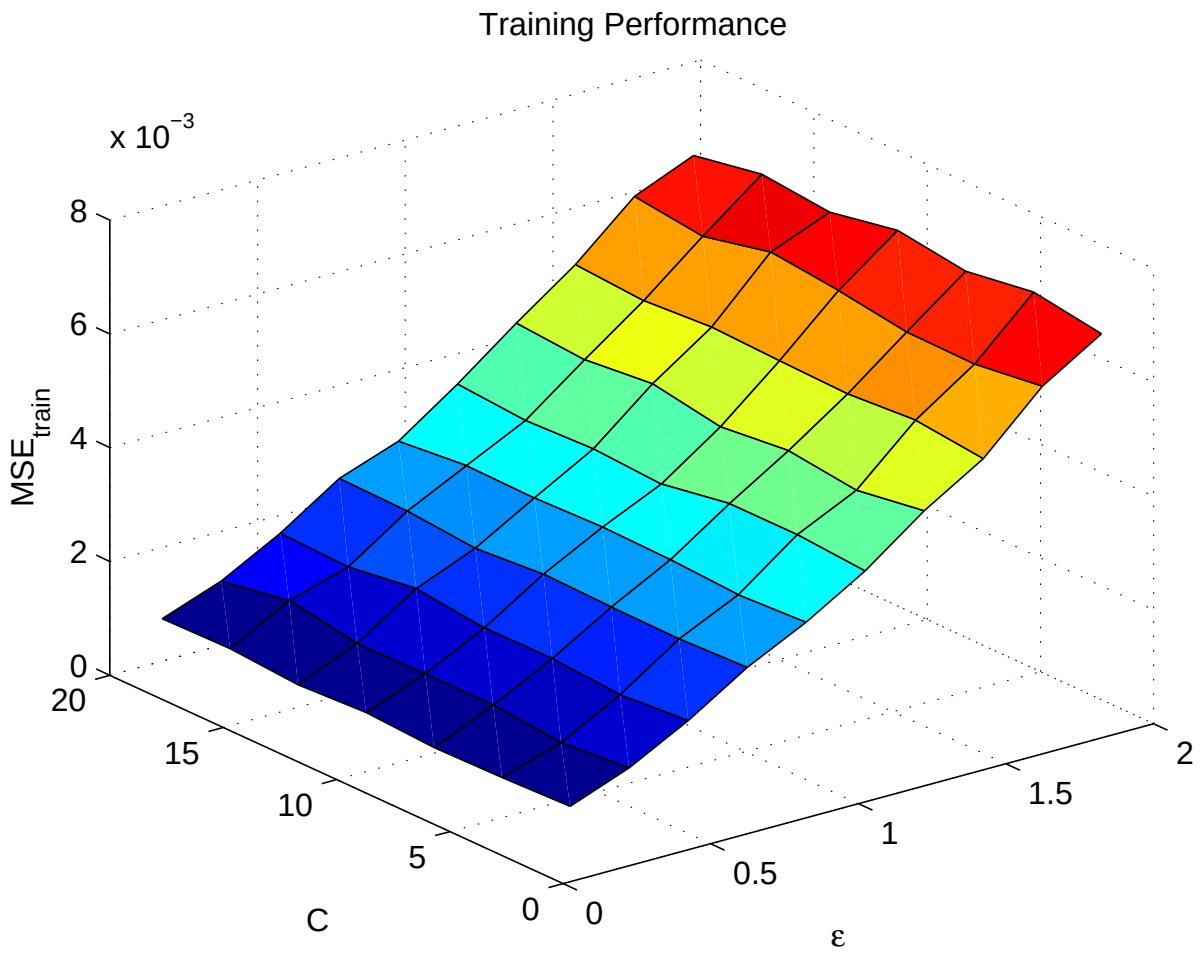


Figure 3-10. Training Performance as a function of C and ϵ values for HDTV sparse channel D data from ATTC (SNR=25dB, 200 training samples)

Fig. 3-11 and Fig. 3-12 shows the generalization performance in terms of test error as a function of C and ϵ values. We can see that the test error increases after an optimal ϵ values which corresponds to the values suggested from Fig. 3-3 and Fig. 3-4.

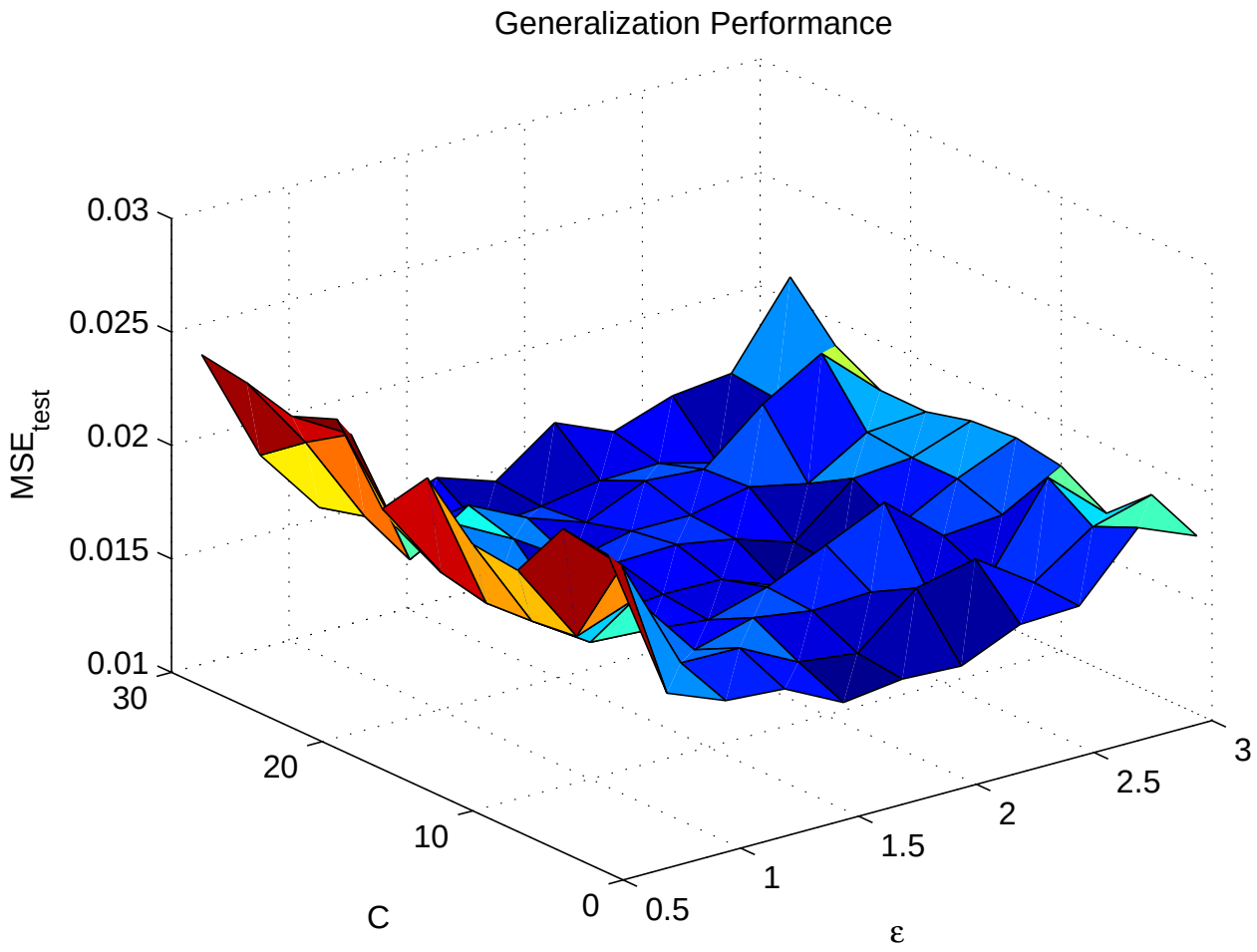


Figure 3-11. Generalization Performance as a function of C and ϵ values for HDTV sparse channel D data from ATTC (SNR=15dB, 200 training samples)

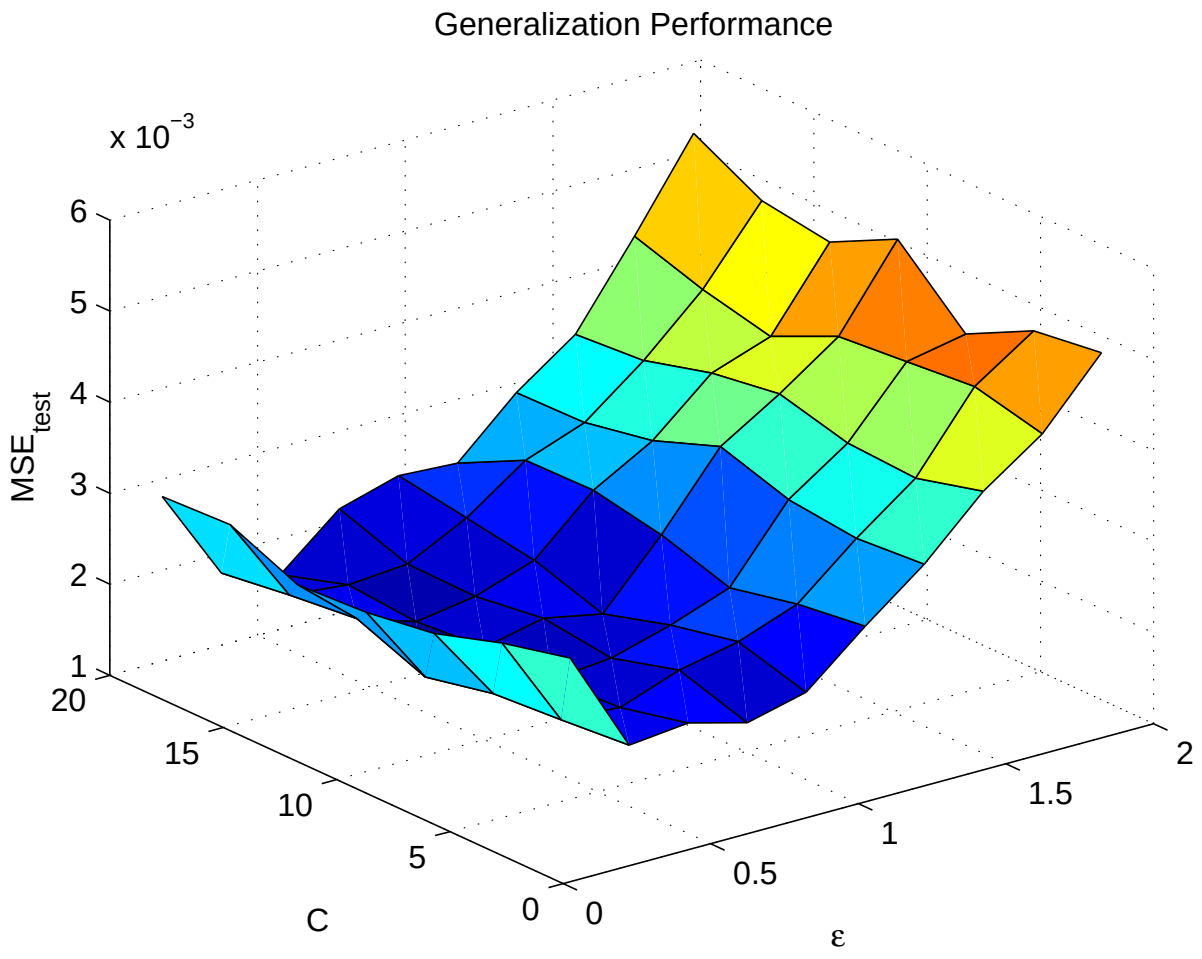


Figure 3-12. Generalization Performance as a function of C and ϵ values for HDTV sparse channel D data from ATTC (SNR=25dB, 200 training samples)

3.7 Simulation Results

To perform the sparse channel estimation, we use the two data set. One data set is the HDTV channel data from ATTC (Advanced Television Test Center) [3]. These data sets are commonly used in literature for sparse channel estimation. Multipath parameters of ATTC channel D are listed in Table 3-1 and channel response shown in Fig. 3-13 which is obtained by pulse shaping filters with 11.5% excess bandwidth and 5.38MHz sampling frequency. The other sparse channel data set has nonzero coefficients which are drawn from a uniform distribution on $[-1, -0.2] \cup [0.2, 1]$ so nonzero taps of this channel are more distinct whereas some coefficients of the HDTV channel are very small.

Table 3-1. ATTC Channel D Path Parameters

path	delay(μs)	delay(T)	phase	atten
1	0.00	0.00	288 deg	20 dB
2	1.80	9.68	180 deg	0 dB
3	1.95	10.49	0 deg	20 dB
4	3.60	19.37	72 deg	18 dB
5	7.50	40.35	144 deg	14 dB
6	19.80	106.52	216 deg	10 dB

Fig. 3-14 shows the SVM solution of the estimation for HDTV channel and Fig. 3-15 for the MP solution. Performance of the HDTV sparse channel estimation is shown in Fig. 3-16. We can see that SVM outperforms other methods in terms of MSE and variance in Fig. 3-16.

Fig. 3-17 shows the performance of the first sparse channel data set which has nonzero coefficients drawn from a uniform distribution on $[-1, -0.2] \cup [0.2, 1]$. We can see that when the sparse channel has distinct nonzero coefficients MP method performs very well. But when the channel taps are small which could be buried under noise, then MP performs poor as in Fig. 3-16.

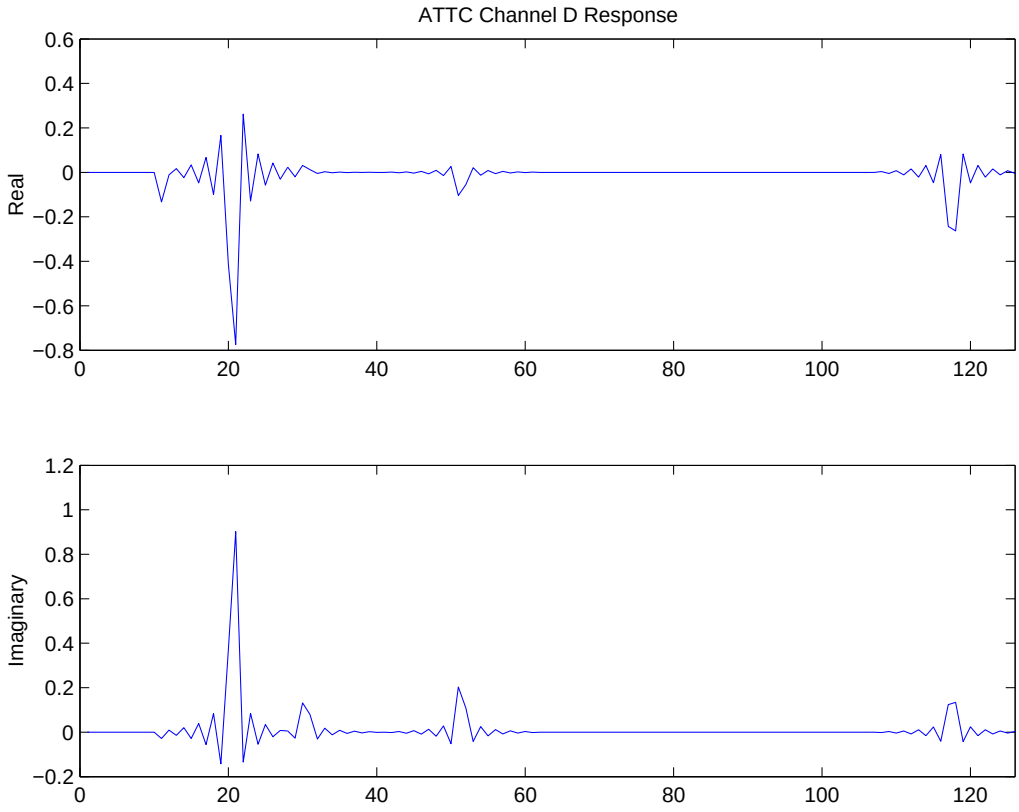
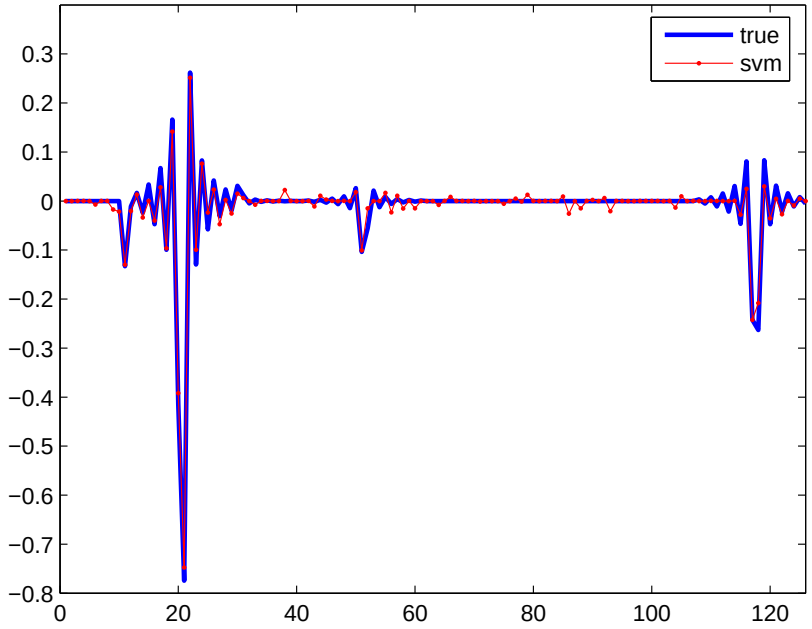
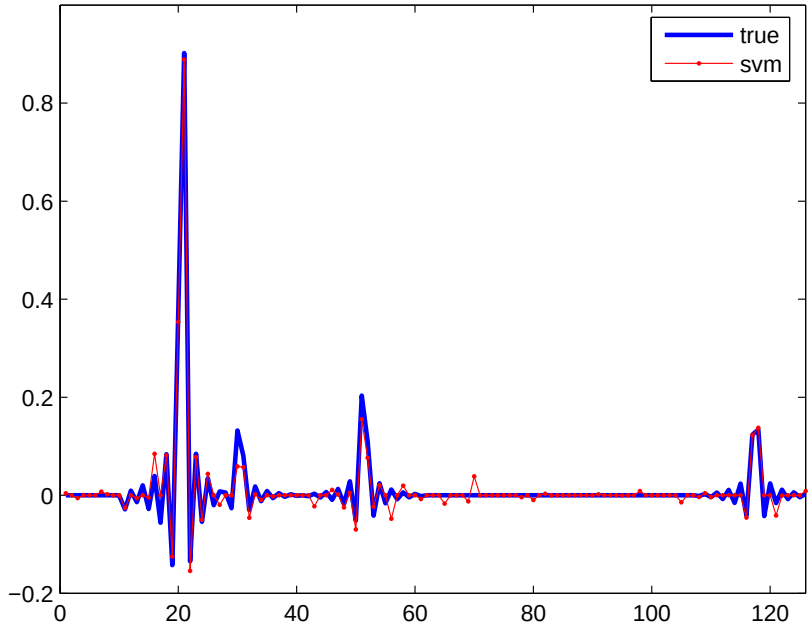


Figure 3-13. Symbol rate sampled response of ATTC channel D

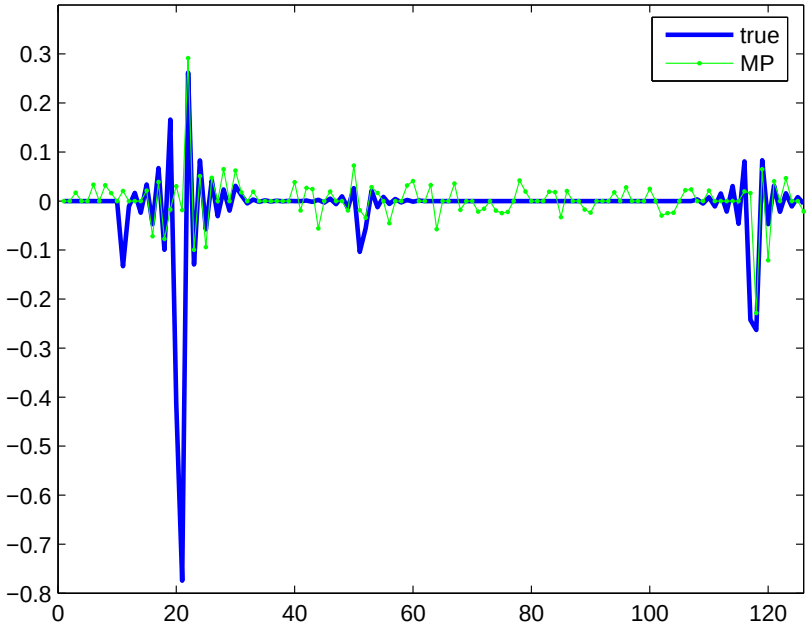


(a) Real

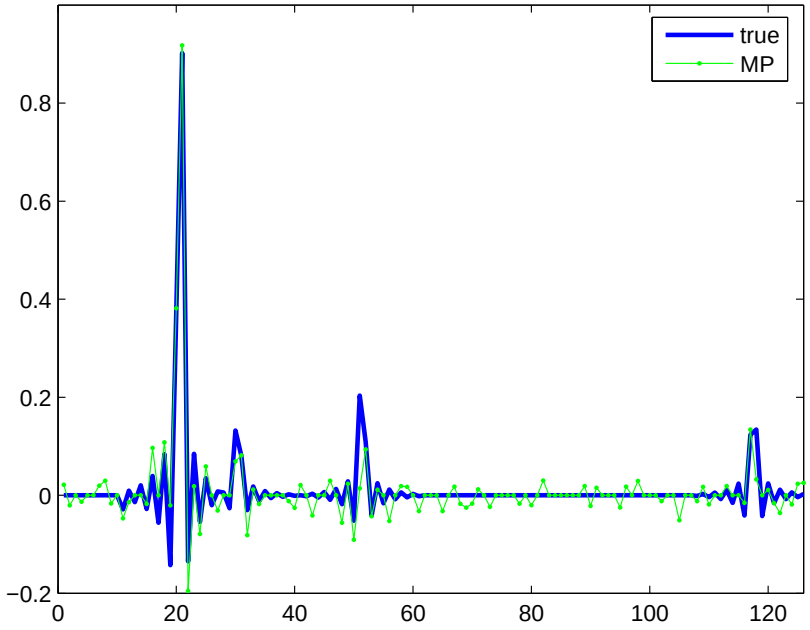


(b) Image

Figure 3-14. SVM solution of sparse channel estimation for HDTV channel.



(a) Real



(b) Image

Figure 3-15. MP solution of sparse channel estimation for HDTV channel.

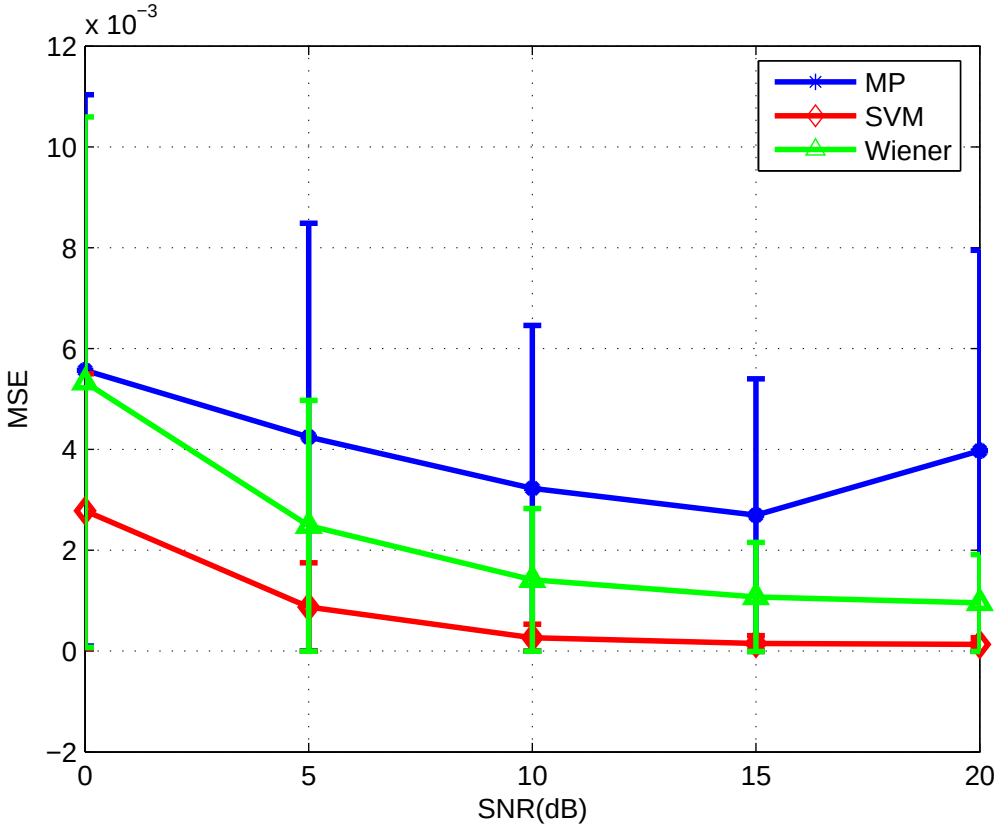


Figure 3-16. Performance of the sparse channel (HDTV channel) estimation. Multipath parameters of ATTC channel D are listed in Table 3-1 and channel response shown is obtained by pulse shaping filters with 11.5% excess bandwidth and 5.38MHz sampling frequency.

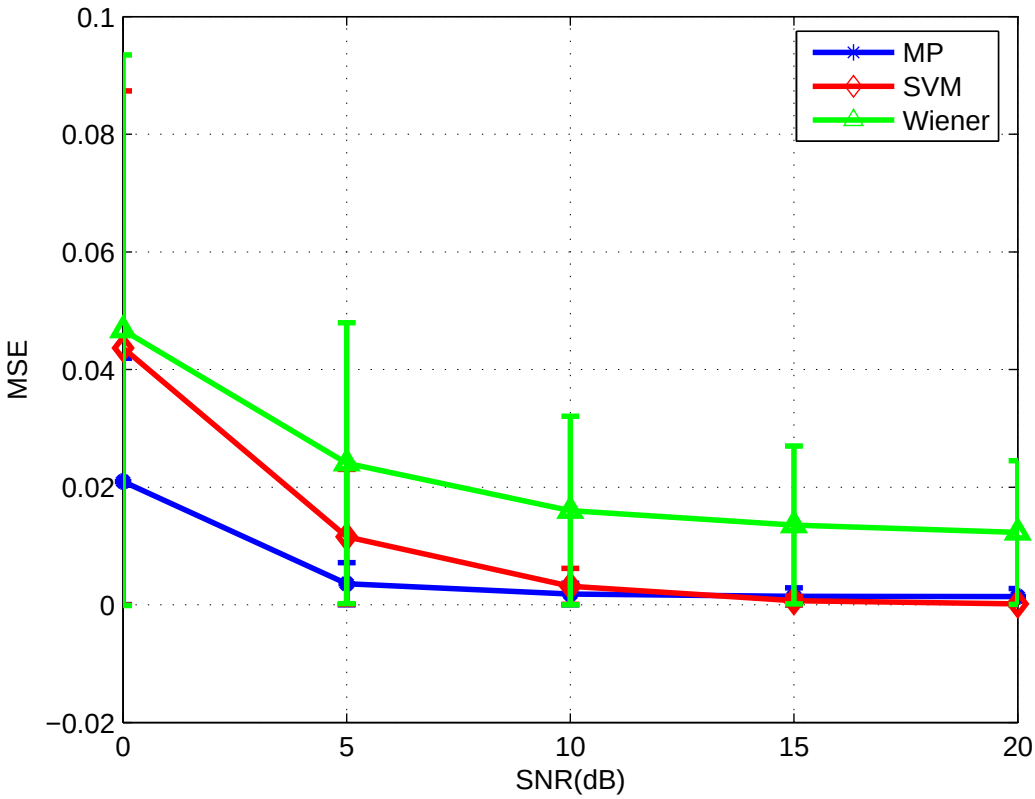


Figure 3-17. Performance of the sparse channel estimation. Nonzero coefficients of the sparse channels are drawn from a uniform distribution on $[-1, -0.2] \cup [0.2, 1]$

In this chapter, support vector machines are briefly introduced including the classification and the regression problem. Then we present a new SVM-based sparse system identification algorithm. SVM parameter selection is also discussed. Simulations demonstrated SVM-based sparse channel estimation method outperforms other methods even when the sparse channel contains very small nonzero taps.

CHAPTER 4
BLIND SPARSE SIMO CHANNEL IDENTIFICATION

4.1 Introduction

Blind estimation of single-input multiple-output (SIMO) channels is a widely studied problem with many signal processing applications. Since the work of [28], it has been well known that second order statistics (SOS) are sufficient for blind identification when the input signal is informative enough and the channels do not share any common roots. Widely used SOS-based methods include the subspace (SS) approach, the least squares (LS) technique and the linear prediction (LP) methods. However, a common drawback of SS and LS techniques is their poor performance when the channel order is overestimated. Recently some robust techniques have been proposed to mitigate this problem (see e.g., [30, 31]). Although these methods offer increased robustness, they still fail when the channel order is highly overestimated.

The structural risk minimization (SRM) principle is a criterion that establishes a trade-off between the complexity of the solution and the closeness to the data. In particular, the support vector machine (SVM) technique, which can be derived from the SRM principle, typically provides a robust solution. The work in [35] was the first attempt to apply an SVM-based approach to the blind identification of SIMO channels [35]. However, the sparsity provided by the SVM solution was not explicitly exploited in [35]. Later work in this direction was presented in [34], in which a new blind identification algorithm based on support vector regression and specifically tailored for sparse SIMO channels was proposed. The main idea of [34] is that the sparse SIMO channel identification can be reformulated as a set of regression problems in which the channel coefficients play the role of the Lagrange multipliers. By using the ϵ-insensitive Vapnik's loss function in the regression problem, a large number of Lagrange multilpliers (and, therefore, a large number of filter coefficients) become zero, thus yielding a sparse filter estimate.

In [37] the previous work [34, 35] is extended in the following direction: first, a robust algorithm for the blind estimation of a non-sparse channel when the channel order has been highly overestimated is derived; secondly, to avoid the high computational cost in solving a QP problem, a fast and simple algorithm called the Adatron [36] is used. We propose to combine iterative regression and SRM principle to solve blind sparse SIMO channel identification problem.

4.2 Observation Model

Without loss of generality, in this work we focus on the one-input, two-output SIMO system shown in Fig. 4-1. In blind channel identification, we need to identify the unknown channel responses, $\mathbf{h}_1, \mathbf{h}_2$, from the received signals only. If the order of the channels is M, then the received signal $\mathbf{r}_i(n)$ from the ith channel is

$$r_i(n) = \sum_{k=0}^{M} h_i(k)s(n-k) + e_i(n), \quad i = 1, 2. \tag{4–1}$$

When we cast $r_i(n), h_i(k), s(n), e_i(n)$ into vectors $\mathbf{r}_i$, $\mathbf{h}_i$, $\mathbf{s}$ and $\mathbf{e}_i$ respectively, equation (4–1) becomes

$$\mathbf{r}_i = \mathbf{h}_i * \mathbf{s} + \mathbf{e}_i, \quad i = 1, 2 \tag{4–2}$$

1 where $*$ denotes convolution. As shown in Fig. 4-1, using the channel outputs $(\mathbf{r}_1, \mathbf{r}_2)$ and the channel estimates $(\hat{\mathbf{h}}_1, \hat{\mathbf{h}}_2)$, one can obtain the following matrix-vector form,:

$$\mathbf{y}_1 = \mathbf{r}_1 * \hat{\mathbf{h}}_2,$$

$$\mathbf{y}_2 = \mathbf{r}_2 * \hat{\mathbf{h}}_1 \ .$$

This relationship can be re-expressed in a matrix-vector form as:

$$\mathbf{y}_1 = \mathbf{R}_1\hat{\mathbf{h}}_2 = \mathbf{R}_2\hat{\mathbf{h}}_1 = \mathbf{y}_2 \ , \tag{4–3}$$

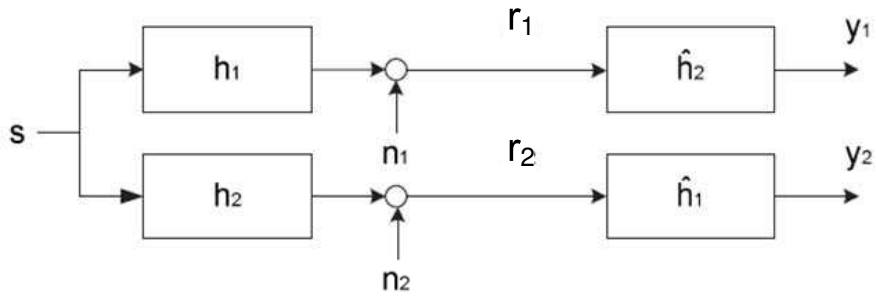


Figure 4-1. Single-input two-output channel.

where $\mathbf{R}_i$'s are Toeplitz matrices defined as

$$\mathbf{R}_i = \begin{pmatrix} r_i(M) & \cdots & r_i(0) \\ r_i(M+1) & \cdots & r_i(1) \\ \vdots & \ddots & \vdots \\ v_i(M+N-1) & \cdots & r_i(N-1) \end{pmatrix} , \tag{4–4}$$

or equivalently,

$$\mathbf{R}\ \hat{\mathbf{h}} = \mathbf{0}\ , \tag{4–5}$$

where

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_2 & -\mathbf{R}_1 \end{bmatrix}, \quad \hat{\mathbf{h}} = \begin{bmatrix} \hat{\mathbf{h}_1} \\ \hat{\mathbf{h}_2} \end{bmatrix} .$$

If we solve (**??**) by minimizing $\hat{\mathbf{h}}^H\mathbf{R}^H\mathbf{R}\hat{\mathbf{h}}$ with the constraint $\| \hat{\mathbf{h}} \| = 1$, then $\hat{\mathbf{y}}$ is the LS solution which is the eigenvector corresponding to the minimum eigenvalue of $\mathbf{R}^H\mathbf{R}$. Based on (**??**), we will next develop a SVM based robust blind identification method even when the channel order is highly overestimated.

4.3 Combined Iterative and SVM Based Approach

4.3.1 Iterative Regression

From equation (4–3), we can formulate the following two coupled regression problems:

$$\mathbf{R}_1\hat{\mathbf{h}}_2 \simeq \mathbf{y}_1, \tag{4–6}$$

$$\mathbf{R}_2\hat{\mathbf{h}}_1 \simeq \mathbf{y}_2, \tag{4–7}$$

Our goal is to make $\mathbf{y}_1 \approx \mathbf{y}_2$. This is an intuitive and simple choice, because it drags the actual outputs $\mathbf{y}_1$ and $\mathbf{y}_2$ closer to each other in order to achieve the equality in (4–3). At each iteration (given $\hat{\mathbf{h}}_1$ and $\hat{\mathbf{h}}_2$), desired output is constructed as

$$\mathbf{y}_d = \frac{\mathbf{y}_1 + \mathbf{y}_2}{2},$$

and we end up with following two new uncoupled regression problems:

$$\mathbf{R}_1\hat{\mathbf{h}}_2 \simeq \mathbf{y}_d, \tag{4–8}$$

$$\mathbf{R}_2\hat{\mathbf{h}}_1 \simeq \mathbf{y}_d, \tag{4–9}$$

This iterative algorithm can be summarized as following

Algorithm 2 Iterative regression [35]

1. Initialize $\hat{\mathbf{h}}_1 = \hat{\mathbf{h}}_2 = \delta[n-d]$.
2. Obtain the outputs $\mathbf{R}_1\hat{\mathbf{h}}_2 \simeq \mathbf{y}_1$ and $\mathbf{R}_2\hat{\mathbf{h}}_1 \simeq \mathbf{y}_2$ and form the desired signal $\mathbf{y}_d = \frac{\mathbf{y}_1+\mathbf{y}_2}{2}$.
3. Solve two new LS regression problems with $\mathbf{y}_d$ as desired output.
4. Normalize the solution and go to step 2.

Note that this algorithm converges to the CCA (Canonical Correlation Analysis) solution

$$\arg\max_{\mathbf{h}_1,\mathbf{h}_2} \rho = \mathbf{h}_1^T \mathbf{R}_{12} \mathbf{h}_2 \quad \texttt{subject to } \mathbf{h}_1^T \mathbf{R}_{11} \mathbf{h}_1 = \mathbf{h}_2^T \mathbf{R}_{22} \mathbf{h}_2 = 1$$

Canonical Correlation Analysis (CCA) is a well-known technique in multivariate statistical analysis to find maximally correlated projections between two data sets. CCA was developed by H. Hotelling [51] and it has been widely used in economics, meteorology and in many modern information processing fields, such as communication problems [52], statistical signal processing [53], independent component analysis [54] and blind source separation [55].

4.3.2 SVM Regression

To fully exploit the sparse approximation characteristics provided by SVMs, each of the regression problems is premultiplied by its conjugate transposed input matrix $\mathbf{R}_i^H$ to yield [c.f. (4–8),(4–9)]:

$$\mathbf{R}_1^H \underbrace{\mathbf{R}_1 \hat{\mathbf{h}}_2}_{\mathbf{x}_1} = \underbrace{\mathbf{R}_1^H \mathbf{y}_d}_{\tilde{\mathbf{y}}_1}, \tag{4–10}$$

$$\mathbf{R}_2^H \underbrace{\mathbf{R}_2 \hat{\mathbf{h}}_1}_{\mathbf{x}_2} = \underbrace{\mathbf{R}_2^H \mathbf{y}_d}_{\tilde{\mathbf{y}}_2}. \tag{4–11}$$

The resultant regression problems have input matrices that are simply the conjugate transposed input matrix $\mathbf{R}_i^H$, and the corresponding desired output vectors become $\mathbf{R}_i^H \mathbf{y}_d$. Moreover, the new regressor $\mathbf{x}_i$ admits an expansion in terms of the filter coefficients, which, in this way, become the Lagrange multipliers of the SVM formulation.

The SVM method minimizes the following cost function

$$J(\mathbf{x}_i) = C \sum_{n=1}^{M} (\xi_n + \xi_n^*) + \frac{1}{2} \parallel \mathbf{x}_i \parallel^2, \quad i = 1, 2 \tag{4–12}$$

subject to

$$\begin{aligned}
\tilde{y}_i(n) - \mathbf{x}_i^H \mathbf{r}_i(n) &\leq \epsilon + \xi_n, && n = 1, \ldots, M \\
\mathbf{x}_i^H \mathbf{r}_i(n) - \tilde{y}_i(n) &\leq \epsilon + \xi_n^*, && n = 1, \ldots, M \\
\xi_n &\geq 0, && n = 1, \ldots, M \\
\xi_n^* &\geq 0, && n = 1, \ldots, M
\end{aligned}$$

where ξ and ξ^* are positive slack variables introduced by SVM procedure and $\mathbf{r}(n)$ denotes the n-th column of $\mathbf{R}_i$ for $i = 1, 2$.

In equation (4–12), the regularization parameter C controls the tradeoff between the training error and the complexity of the solution. On the other hand, ϵ is a parameter that determines the precision of the regression and therefore controls the sparseness of the final solution. Then, the solution is a linear combination of input data

$$\mathbf{x} = \sum_{n=1}^{M} (\alpha_n^* - \alpha_n)\mathbf{r} \tag{4–13}$$

where α_n^*, α_n are two different Lagrange multipliers.

In equation (4–13), only a small number of Lagrange multipliers $(\alpha_n^* - \alpha_n)$, which correspond to the channel coefficients $h(0), h(1), \ldots, h(M-1)$ will be nonzero. Accordingly, the overestimated channel coefficients will be zeros by the SRM principle.

4.3.3 Implementation of Support Vector Regression

The computational cost in solving the QP problem in equation (4–12) is the main drawback of applying the SVM technique to practical estimation problems. Several techniques have been proposed to solve this problem, including the use of iterative reweighted least squares (IRWLS) techniques [56, 57] and the Adatron algorithm [2, 36, 58]. The IRWLS requires a matrix inversion at each iteration so the computational burden could be considerably high even for a moderate number of data. On the other hand, the Adatron algorithm is a much simpler least mean square (LMS)-like adaptive algorithm and

its convergence rate is exponential with the number of iterations. However, it is a memory intensive method because all the kernel products need to be precomputed and saved.

4.3.3.1 The Adatron Algorithm

We can use the Adatron algorithm to solve the re-formulated regression in equations (4–10) and (4–11). In the dual representation, the optimization problem in equation (4–12) can be written as

$$\begin{aligned} L = &-\frac{1}{2}\sum_{n=1}^{M}\sum_{m=1}^{M}(\alpha_n - \alpha_n^*)(\alpha_m - \alpha_m^*)\langle \mathbf{x}_n, \mathbf{x}_m\rangle \\ &- \epsilon\sum_{n=1}^{M}(\alpha_n + \alpha_n^*) + \sum_{n=1}^{M} y_n(\alpha_n - \alpha_n^*) \end{aligned} \tag{4–14}$$

subject to $\alpha_n, \alpha_n^* \in [0, C]$ and $\langle\rangle$ denotes dot product.

The Adatron algorithm maximizes the above dual problem with gradient ascent techniques [58]. Specifically, the Lagrange multipliers are updated according to

$$\delta\alpha_n = \eta\left(-\sum_{m=1}^{N}(\alpha_m - \alpha_m^*)\langle \mathbf{x}_n, \mathbf{x}_m\rangle + y_n - \epsilon - b\right), \tag{4–15}$$

$$\delta\alpha_n^* = \eta\left(\sum_{m=1}^{M}(\alpha_m - \alpha_m^*)\langle \mathbf{x}_n, \mathbf{x}_m\rangle + y_n - \epsilon + b\right) \tag{4–16}$$

followed by updating α_n and α_n^* with $(\alpha_n + \delta\alpha_n)^\dagger$ and $(\alpha_n^* + \delta\alpha_n^*)^\dagger$, respectively, where $a^\dagger \triangleq \max\{a, 0\}$ and η is the learning rate. In addition, the bias b is updated as to $b + \delta\alpha_n - \delta\alpha_n^*$ and the evolution of this bias value can be used to check the convergence of the algorithm.

As we can see in equations (4–15) and (4–16), the Adatron algorithm is very simple to implement especially in DSP/FPGA hardware. To run this algorithm, all one needs is just an adder and a multiplier. Furthermore, the computation time of Adatron increases linearly with the number of data while the conventional QP's increases exponentially. This simplicity is its main advantage. Using this algorithm, we propose a blind sparse channel estimation method as summarized under Algorithm 3.

Algorithm 3 Adatron algorithm

1. Set ϵ, C and initialize $\alpha_n = 0, \alpha_n^* = 0,\ \forall n$; and $b = 0$.

2. For all training samples $n = 1, \cdots, M$, execute

   $\delta\alpha_n = \eta\left(-\sum_{m=1}^{N}(\alpha_m - \alpha_m^*)\langle \mathbf{x}_n, \mathbf{x}_m\rangle + y_n - \epsilon - b\right)$.

   If $(\alpha_n + \delta\alpha_n) < 0$ then set $\alpha_n = 0$, else $\alpha_n \longleftarrow \alpha_n + \delta\alpha_n$.

   $\delta\alpha_n^* = \eta\left(-\sum_{m=1}^{N}(\alpha_m - \alpha_m^*)\langle \mathbf{x}_n, \mathbf{x}_m\rangle + y_n - \epsilon - b\right)$.

   If $(\alpha_n^* + \delta\alpha_n) < 0$ then set $\alpha_n^* = 0$, else $\alpha_n^* \longleftarrow \alpha_n + \delta\alpha_n$.

3. Repeat step 2 until convergence.

4. $h(n) = (\alpha_n - \alpha_n^*)$, for $n = 1, \cdots, M$.

Combining the iterative regression procedure and the Adatron algorithm, overall algorithm is summarized in Algorithm 4.

The computation cost of the proposed method is $niter1 \times niter2 \times O(M)$ where $niter1$ is the iteration number of Adatron algorithm (Algorithm 2) and $niter2$ is the one for iterative regression algorithm (Algorithm 3). Adatron convergence plots are depicted in Fig. 4-2 and it shows larger η values have fast convergence speed. Convergence of the proposed overall algorithm is shown in Fig. 4-3.

Algorithm 4 Overall algorithm

1. Intialize ϵ, C, and $\hat{\mathbf{h}}_1 = \hat{\mathbf{h}}_2 = \delta[n-d]$.
2. Obtain the outputs $\mathbf{X}_1\hat{\mathbf{h}}_2 = \mathbf{y}_1$ and $\mathbf{X}_2\hat{\mathbf{h}}_1 = \mathbf{y}_2$.
3. Form the desired signal $\mathbf{y}_d = \frac{\mathbf{y}_1+\mathbf{y}_2}{2}$.
4. Construct the transformed SV regression problems:

   $\mathbf{X}_1^T\mathbf{w}_1 = \mathbf{X}_1^T\mathbf{y}_d$ and $\mathbf{X}_2^T\mathbf{w}_2 = \mathbf{X}_2^T\mathbf{y}_d$.
5. Solve the QP problem using Adatron.
   $\Rightarrow$ the Lagrange multipliers are the filter coefficients.
6. Normalize the solution and go to step 2.

4.3.3.2 Learning rate for convergence

The change in the Lagrangian (4–14) is:

$$\begin{aligned}
\delta L &= -\sum_{m=1}^{M} \delta\alpha_n(\alpha_m - \alpha_m^*)\langle \mathbf{x}_n, \mathbf{x}_m\rangle + y_n\delta\alpha_n - \epsilon\delta\alpha_n - \frac{1}{2}(\delta\alpha_n)^2\langle \mathbf{x}_n, \mathbf{x}_n\rangle \\
&= \delta\alpha_n\left(-\sum_{m=1}^{N}(\alpha_m - \alpha_m^*)\langle \mathbf{x}_n, \mathbf{x}_m\rangle + y_n - \epsilon - b\right) - \frac{1}{2}(\delta\alpha_n)^2\langle \mathbf{x}_n, \mathbf{x}_n\rangle \\
&= \delta\alpha_n\left(\frac{\delta\alpha_n}{\eta}\right) - \frac{1}{2}(\delta\alpha_n)^2\langle \mathbf{x}_n, \mathbf{x}_n\rangle \\
&= (\delta\alpha_n)^2\left(\frac{1}{\eta} - \frac{\langle \mathbf{x}_n, \mathbf{x}_n\rangle}{2}\right)
\end{aligned} \tag{4–17}$$

For $\delta\alpha_n^*$, it can be shown that

$$\delta L = (\delta\alpha_n^*)^2\left(\frac{1}{\eta} - \frac{\langle \mathbf{x}_n, \mathbf{x}_n\rangle}{2}\right) \tag{4–18}$$

From eqs. (4–17)and (4–18), we can get the relation $0 < \eta\langle \mathbf{x}_n, \mathbf{x}_n\rangle < 2$ for a positive change of δL.

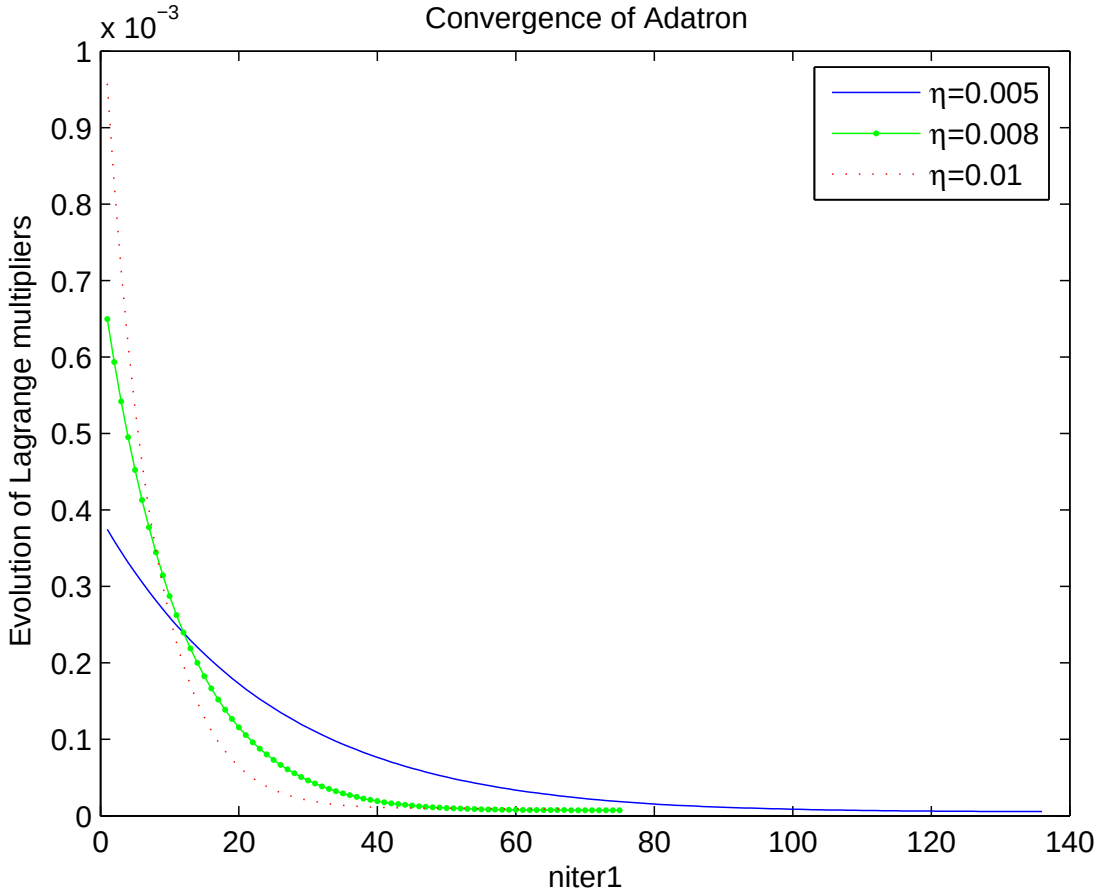


Figure 4-2. Convergence of Adatron

As we can notice from (4–17)and (4–18), bound of the learning rate η is data dependent and can be determined from

$$0 < \eta < \frac{2}{\langle \mathbf{x}_n, \mathbf{x}_n \rangle} \tag{4–19}$$

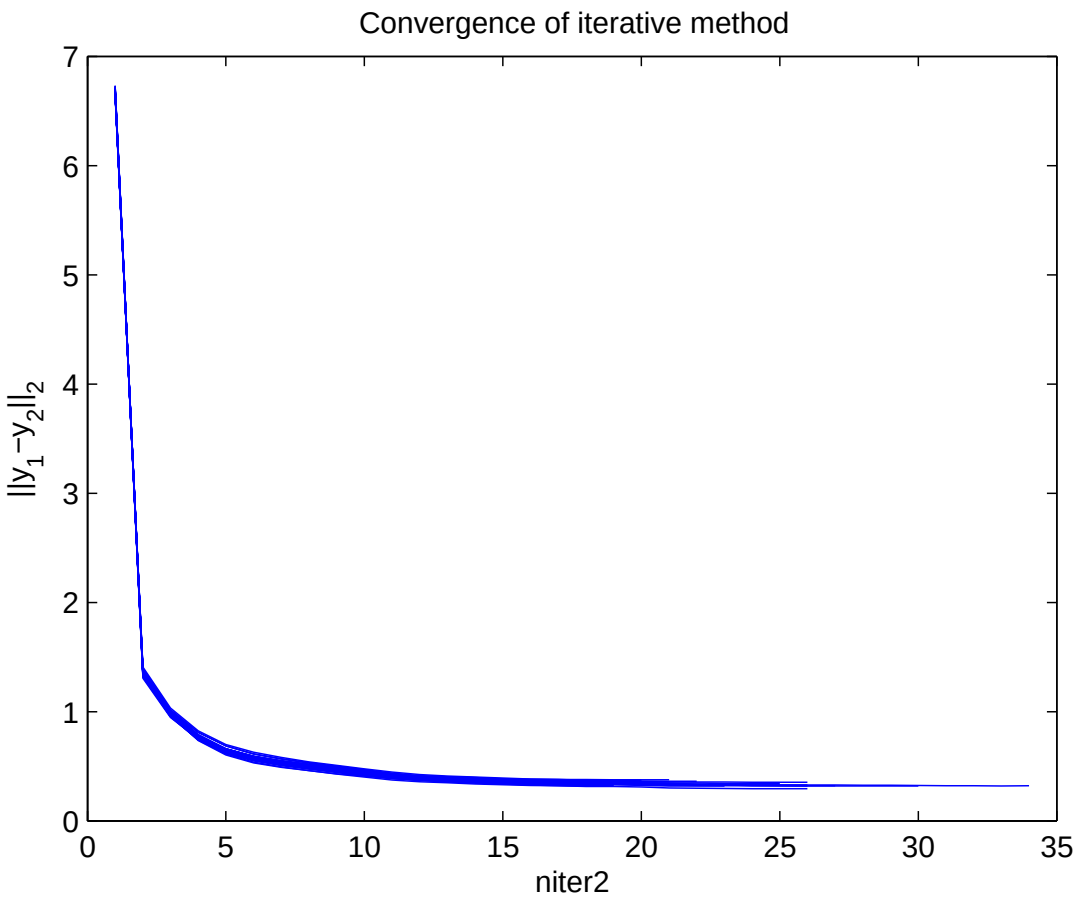


Figure 4-3. Convergence of iterative regression.

4.4 Simulation Results

Several simulations have been conducted to test the performance of our proposed algorithm. The performance is measured in terms of the normalized mean squared error (NMSE) defined as [31]:

$$NMSE = \frac{1}{\| \mathbf{h} \|^2} \min_{\alpha, k \geq 0} \left\| \alpha \hat{\mathbf{h}} - \begin{bmatrix} \mathbf{0}_{k,1} \\ \mathbf{h} \\ \mathbf{0}_{\dot{M}-M-k} \end{bmatrix} \right\|^2$$

where $\dot{M} \geq M$ is the estimated channel order.

In the first simulation we consider a sparse SIMO system which consists of a single transmit antenna and two receive antennas. The two sparse channels are respectively,

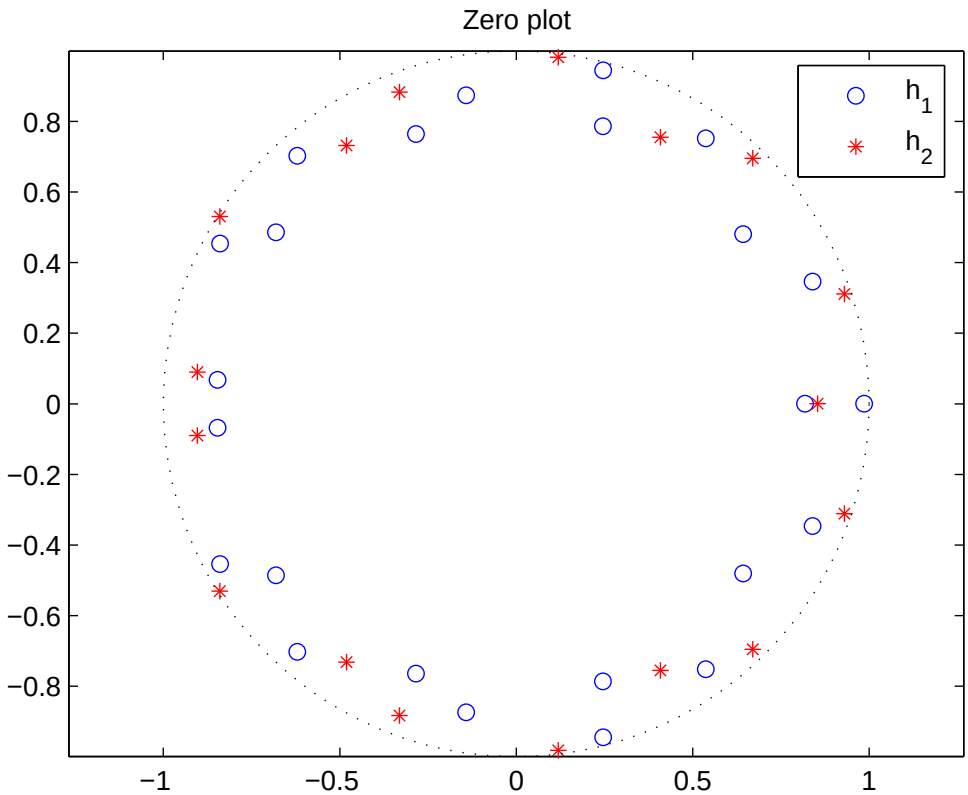
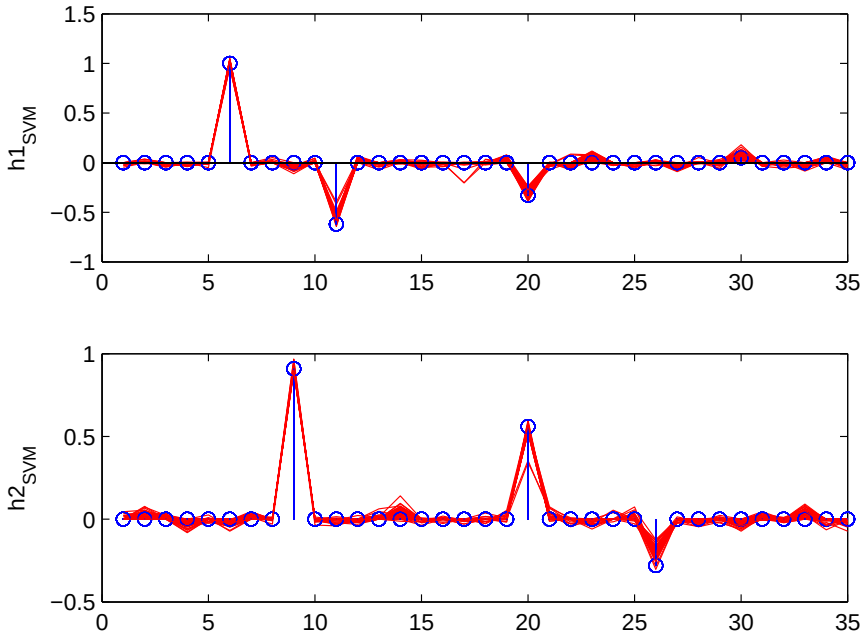


Figure 4-4. Zeros of subchannel $\mathbf{h}_1, \mathbf{h}_2$

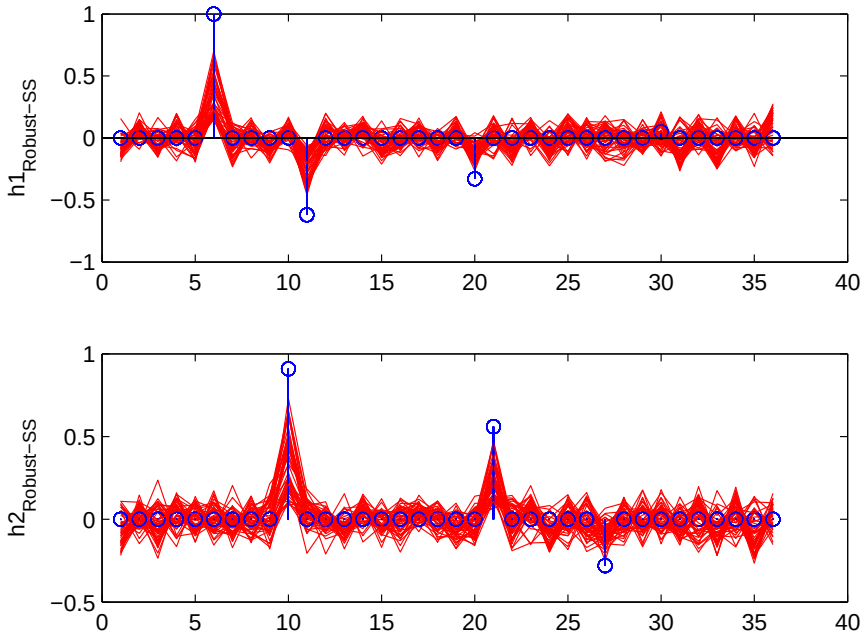
$\mathbf{H}_1(z) = 1 - 0.62z^{-5} - 0.33z^{-14} + 0.08z^{-24}$, $\mathbf{H}_2(z) = 0.91 + 0.56z^{-11} - 0.28z^{-17}$. Input of this system is $N = 100$, i.i.d. BPSK signals. In Fig.4-4, we plot the zeros of $\mathbf{h}_1, \mathbf{h}_2$. Note that there are pairs of close zeros which impair subspace based method because of a badly conditioned input correlation matrix.

Fig.4-5 shows the robustness to order overestimation when SNR is 20dB. It is evident that the proposed method outperforms other methods in highly overestimated channel order estimate. Performance is summarized in Fig.4-6 and shows the proposed method performs much better than Regalia method in identifying the coefficients of zero taps or very small taps.

In the next example we consider a raised-cosine pulse with duration 4T (T is the symbol period) with a roll-off factor 0.1 and the multipath channel is $h(t) = \delta(t) - 0.7\delta(t -$



(a) SVM based method



(b) Robust-SS method

Figure 4-5. 50 trials of the SVM based method and Robust-SS method when overestimated by 10 taps and SNR=30dB

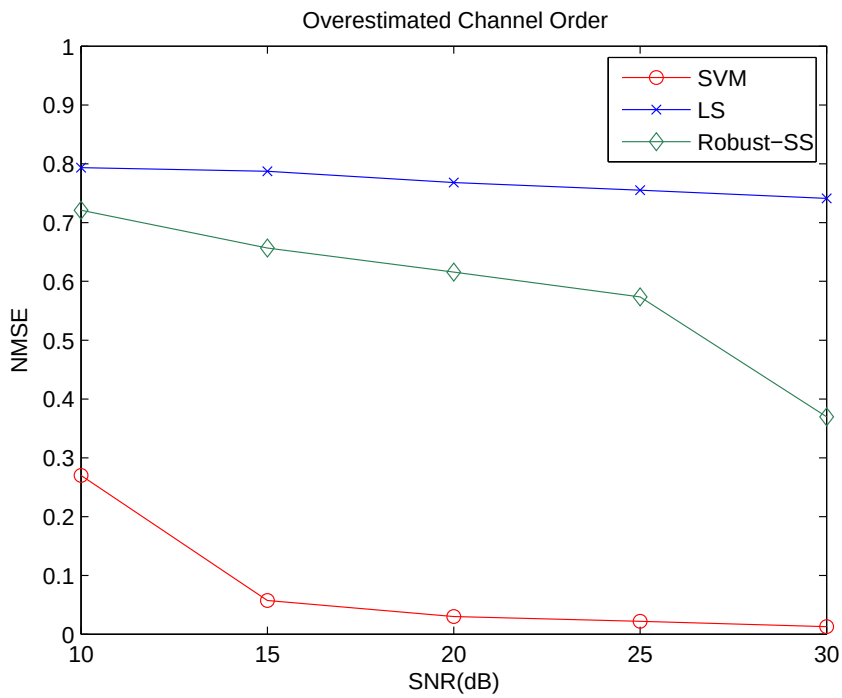
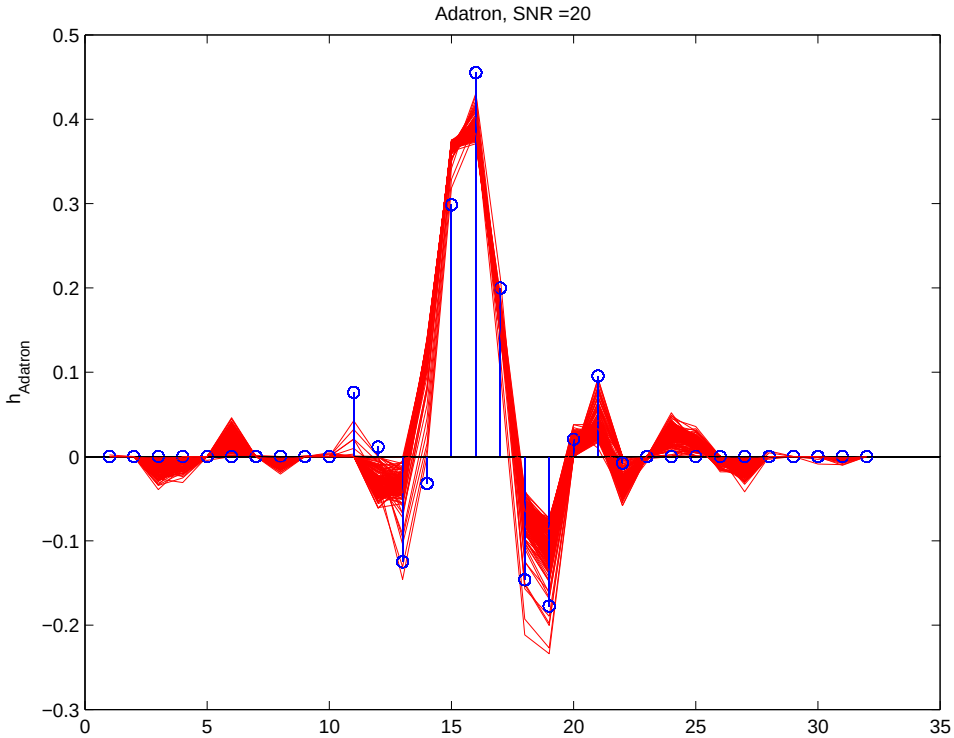


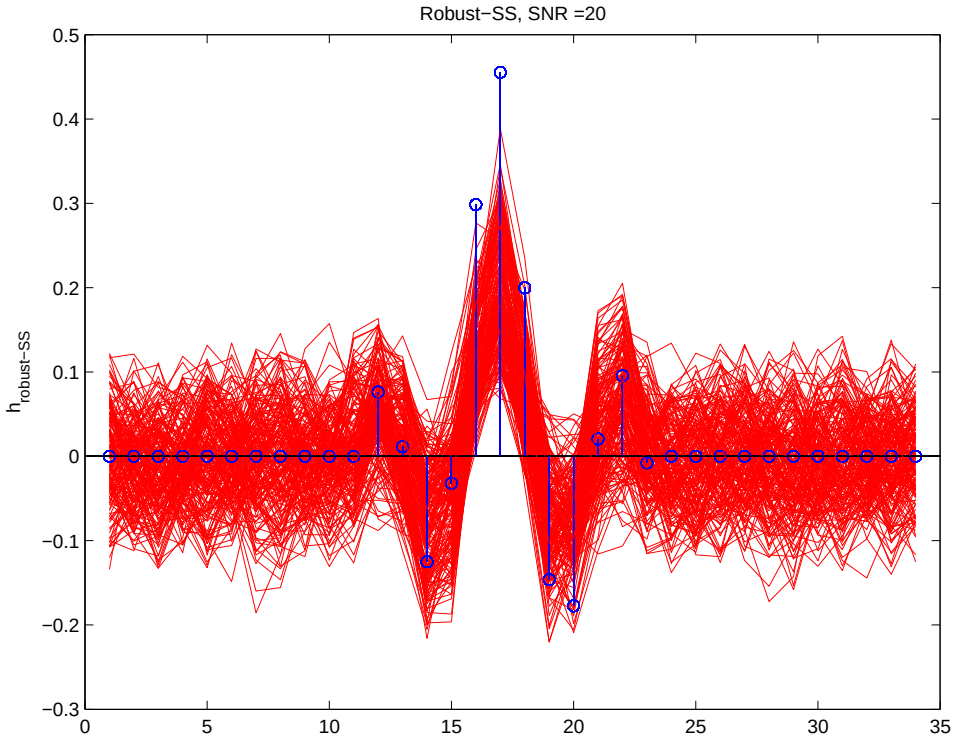
Figure 4-6. Robustness when the channel order is exact and overestimated by 10 taps

$\frac{T}{4}$). The input signal is i.i.d. BPSK signal and the received data is sampled at twice the symbol rate to obtain a SIMO system. Fig. 4-8 depicts the performance at different SNRs. This example shows that proposed SVM based method can be also used for non-sparse channels, in particular when the channel order has been highly overestimated. In this situation, the structural risk minimization (SRM) principle pushes the small leading and trailing terms of the impulse response to zero. Note that the performance of our proposed method is much better than other methods, especially at low SNR. In Fig. 4-7, 50 trials of our proposed algorithm and the robust method proposed in [2], it is clear that the estimation of our proposed method at zero tap coefficients is much better than the robust-SS method.

In this chapter, our SVM based sparse estimation method is extended in the following direction: first, a robust algorithm for the blind estimation of a non-sparse channel when the channel order has been highly overestimated is derived; secondly, to avoid the high computational cost in solving a QP problem, a fast and simple algorithm called the



(a) SVM based method using the Adatron



(b) Robust-SS method

Figure 4-7. 50 trials of the SVM based method using the Adatron and robust-SS method when the channel order is overestimated by 20 taps and SNR=20dB.

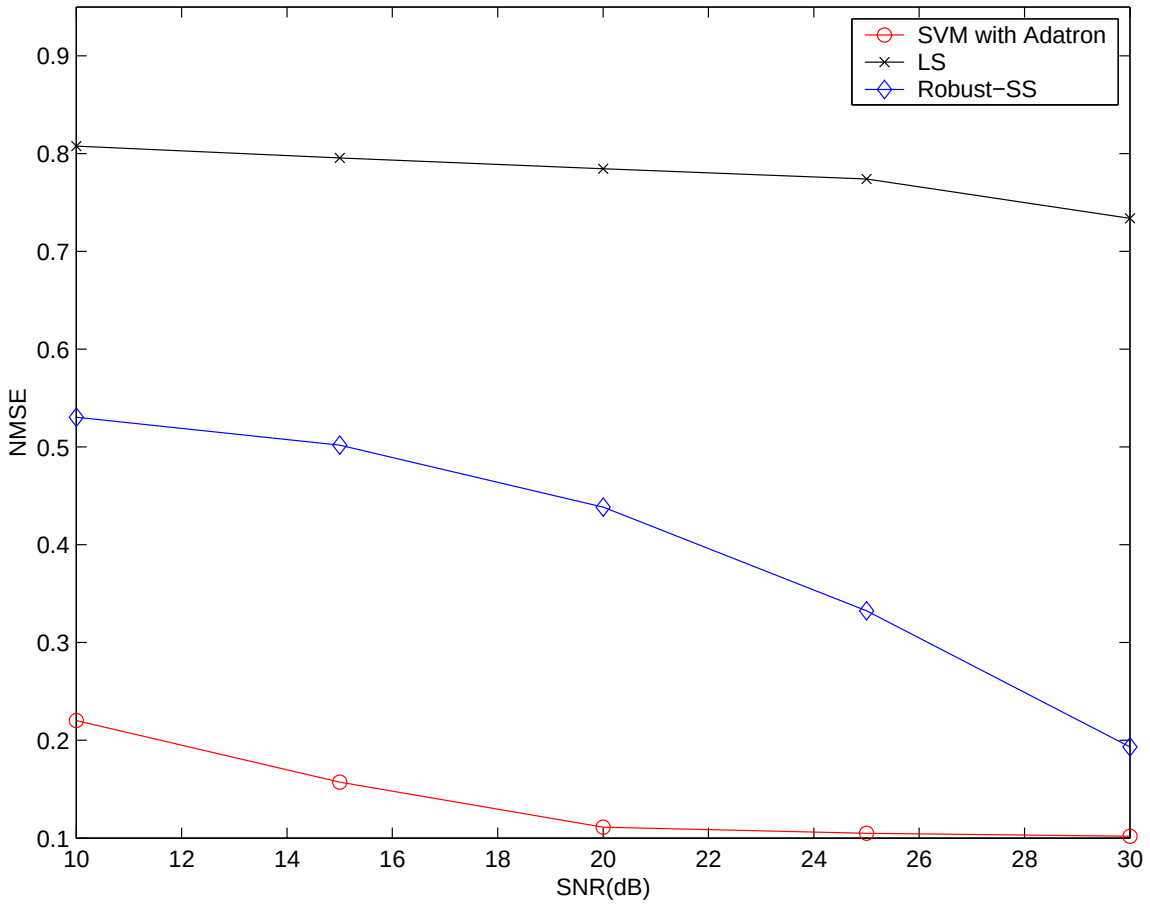


Figure 4-8. Performance comparison when the channel order is overestimated by 20 taps (raised-cosine pulse followed by a multipath channel).

Adatron [36] is used. We proposed to combine iterative regression and SRM principle to solve blind sparse SIMO channel identification problem. Also we shows that proposed SVM based method can be also used for non-sparse channels, in particular when the channel order has been highly overestimated.

CHAPTER 5
IDENTIFICATION OF TIME VARYING & FREQUENCY-SELECTIVE MULTIPATH CHANNEL

5.1 Introduction

Time-varying channel estimation is a major obstacle to increase the capacity and reliability of wireless communication systems. For slow fading channels, adaptive algorithm for time-invariant such as RLS and LMS can provide a reasonable alternative but these adaptive algorithms do not track fast channel variations. For fast fading channels explicit incorporation of the channel's time-varying (TV) characteristics is needed. Recently, a deterministic basis expansion model is widely used for cellular radio applications, especially when the multipath is caused by a few strong reflectors. Receivers that explicitly model the channel variation in time are more successful than the LMS or RLS. The Karhunen-Loève (KL) expansion [59], the polynomial model [60, 61], the exponential model [62, 63], and wavelets [64] have been used for modeling the TV channel. The time-varying taps are expressed as a superposition of time-varying bases with time-invariant coefficients. By assigning time variations to the bases, rapidly fading channels with coherence time as small as a few tens of symbols can be captured.

In this chapter, we propose to use the SVM technique with the exponential basis expansion model to identify the time-varying channel.

5.2 Basis Expansion Models

The TV system is modeled by

$$x(n) = \sum_{l=0}^{L} h(n;l)s(n-l) + v(n) \tag{5–1}$$

where the TV impulse response $h(n;l)$ depends on time n. In this thesis, a deterministic basis expansion is used to model the TV impulse response $h(n;l)$ [62, 63]. TV impulse response of rapidly fading channels is expanded over a basis of complex exponentials that arise due to Doppler effects encountered in the multipath environment. The TV taps are expressed as a superposition of TV bases with time invariant (TI) coefficients. By

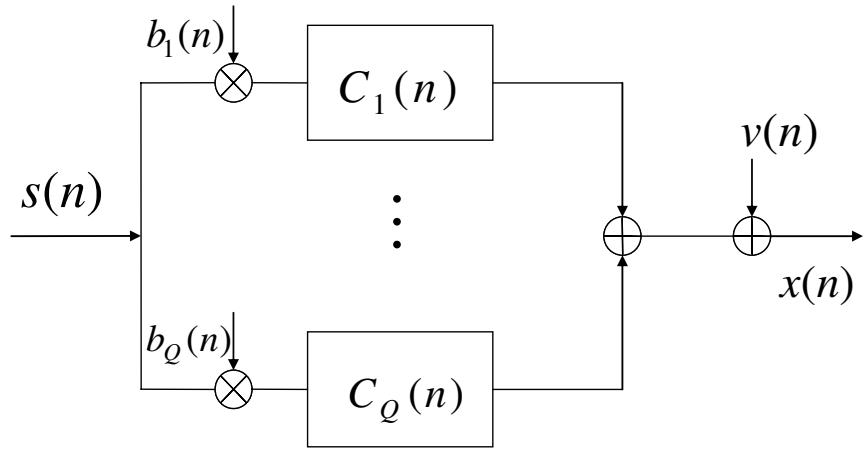


Figure 5-1. Basis expansion model of a time-varying system

assigning time variations to the bases, rapidly fading channels can be captured. Note that complex exponentials model the Doppler effects in (5–2), and by using them, the estimation of time-varying channel system can be cast into a time-invariant estimation problem. The basis expansion model is given by

$$x(n) = \sum_{l=0}^{L} \underbrace{\left[\sum_{q=1}^{Q} c_q(l) e^{j\omega_q n} \right]}_{h(n;l)} s(n-l) + v(n) \tag{5–2}$$

where $s(n)$ is input, $h(n;l)$ is finitely parameterized for each lag l via its expansion coefficients $c_q(l)$ onto known exponential bases $\{1, e^{j\omega_2 n}, \cdots, e^{j\omega_P n}\}$, as depicted in Fig.5-1. To estimate the TI parameters $\{c_q(l)\}$, we assume the knowledge of the base frequencies $\{\omega_i\}_{i=1}^{P}$. This can by estimated using tests for cyclostationarity or adaptive maximum-likelihood methods [65, 66].

From (5–2), we have

$$x(n) = \sum_{q=1}^{Q} \left[\sum_{l=0}^{L} c_q(l) s(n-l) e^{j\omega_q n} \right] + v(n) \tag{5–3}$$

This relationship can be re-expressed in a matrix-vector form as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{S}_1 & \mathbf{S}_2 & \cdots & \mathbf{S}_Q \end{bmatrix} \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_Q \end{bmatrix} + \mathbf{v} \tag{5–4}$$

$$= \mathbf{S}\mathbf{c} + \mathbf{v} \tag{5–5}$$

where

$$\mathbf{S}_q = \begin{bmatrix} s(1)e^{j\omega_q 1} & 0 & \cdots & 0 \\ s(2)e^{j\omega_q 2} & s(1)e^{j\omega_q 2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ s(N)e^{j\omega_q N} & s(N-1)e^{j\omega_q N} & \cdots & s(N-L+1)e^{j\omega_q N} \end{bmatrix}$$

$$\mathbf{c}_q = \begin{bmatrix} c_q(0) & c_q(1) & \cdots & c_q(L-1) \end{bmatrix}^T$$

$$\mathbf{x} = \begin{bmatrix} x(1) & x(2) & \cdots & x(N) \end{bmatrix}^T$$

$$\mathbf{v} = \begin{bmatrix} v(1) & v(2) & \cdots & v(N) \end{bmatrix}^T$$

Using Least Squares, $\mathbf{c}$ in (5–5) can be estimated as

$$\hat{\mathbf{c}} = (\mathbf{S}^H\mathbf{S})^{-1}\mathbf{S}^H\mathbf{x} \tag{5–6}$$

Then TV channel coefficients $h(n;l)$ are estimated using the estimated $\mathbf{c}$. When the number of training symbol is short, the performance of LS estimate (5–6) is bad. In [61], the MP method is used with the polynomial basis model to find the best aligned column to the received signal. In this chapter, we extend the previous work [34, 37] to time-varying environment in a data-aided manner.

5.3 Time Varying Channel Estimation using SVR

From equation (5–5), the system model is

$$\mathbf{x} = \begin{bmatrix} \mathbf{S}_1 & \mathbf{S}_2 & \cdots & \mathbf{S}_Q \end{bmatrix} \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_Q \end{bmatrix} + \mathbf{v}$$

$$= \mathbf{S}\mathbf{c} + \mathbf{v}$$

Then we can estimate time-invariant parameter $\mathbf{c}$ using SVR technique by minimizing

$$J(\mathbf{c}) = C \sum_{n=1}^{M} |x_n - \mathbf{S}^T \mathbf{c}_n|_\epsilon + \frac{1}{2}\mathbf{c}^T\mathbf{c}$$

and the solution is

$$\mathbf{c} = \sum_{n=1}^{M} (\alpha_n^* - \alpha_n)\mathbf{S}_n \tag{5–7}$$

where $\alpha_n^* - \alpha_n$ are the Lagrange multiplier and M is the number of support vectors.

Also SVM-based blind method developed in previous chapter can be easily applied to this TV case.

5.4 Simulation Result

Several simulations have been conducted to test the performance of the proposed algorithm. Normalized mean square error (NMSE) between a $\mathbf{h}$ and its estimate $\hat{\mathbf{h}}$ is computed as follows:

$$NMSE = \frac{1}{R} \sum_{r=1}^{R} \frac{\|\hat{\mathbf{h}} - \mathbf{h}\|^2}{\|\mathbf{h}\|^2} \tag{5–8}$$

where r denotes realization and R is the number of realizations. In Fig. 5-3, we illustrate the estimated TI parameter $\{c_q(l)\}_{q=1}^{Q}$ of basis expansion model using both LS and SVM method. The input signal is i.i.d. BPSK signal and $SNR = 10dB$. $N = 100$ training symbols were used with a channel order $L = 3$ and the $Q = 3$ bases were chosen. All plots

are an average of 100 Monte Carlo runs. Here we can see that SVM performs better than the LS method. Fig. 5-2 illustrates that NMSE of proposed SVM method is better than LS method. Fig. 5-3 and Fig. 5-4 shows the 50 trials of blind estimation of TI parameter $\{c_q(l)\}_{q=1}^{Q}$ of basis expansion model using both LS and SVM method.

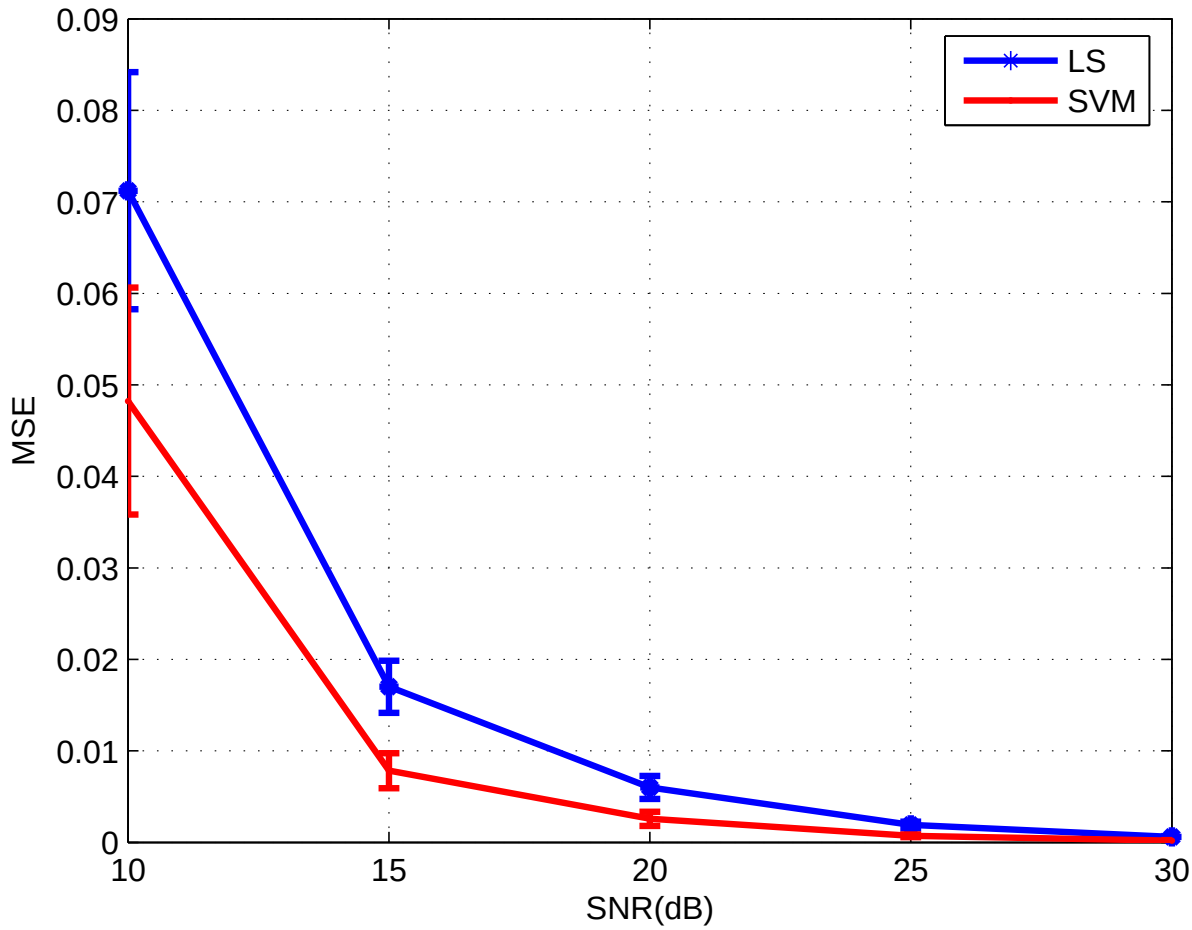


Figure 5-2. MSE Comparison of LS and SVM when $N = 100, L = 3$, and $Q = 3$.

In this chapter, we proposed to use the SVM technique which developed in previous chapter with the exponential basis expansion model to identify the time-varying channel. Simulation demonstrated that BEM-SVM method successfully applied to time varying channel estimation.

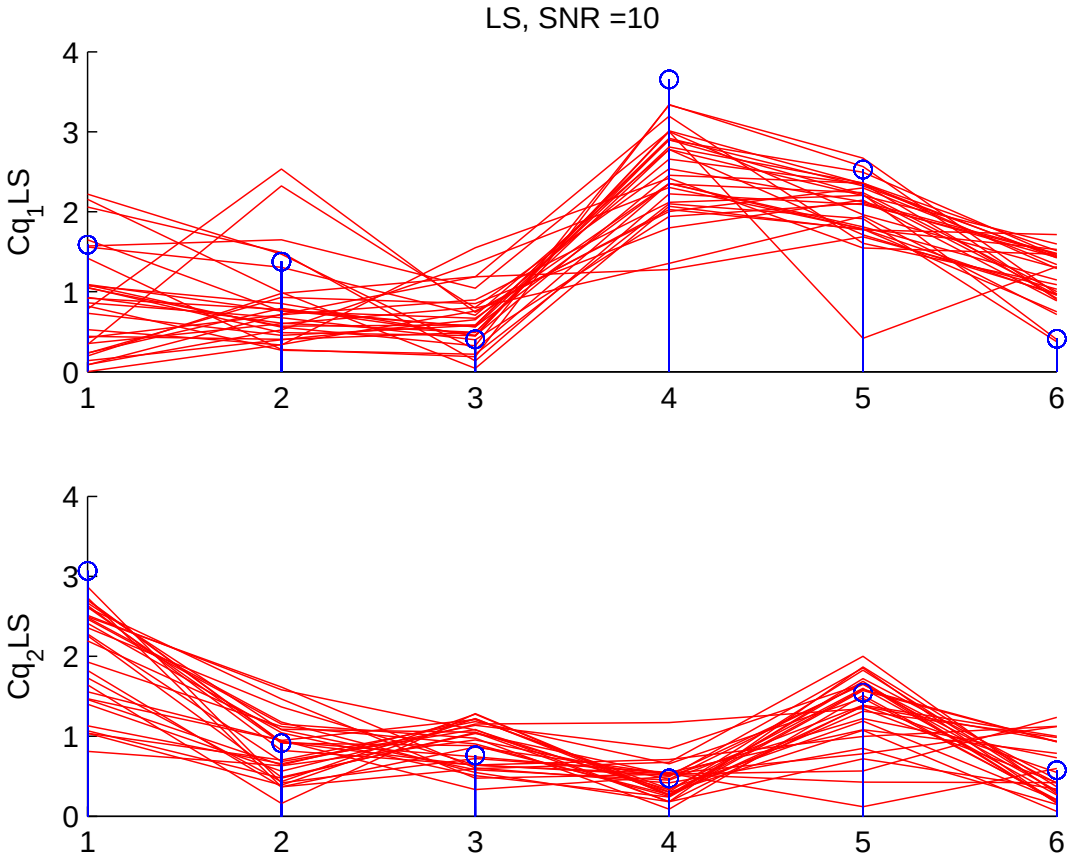


Figure 5-3. Blind estimation of basis coefficients $\{c_q(l)\}_{q=1}^{Q}$ when $N = 80, L = 3$, $Q = 2$, and SNR= $10dB$ (50 trials) using LS.

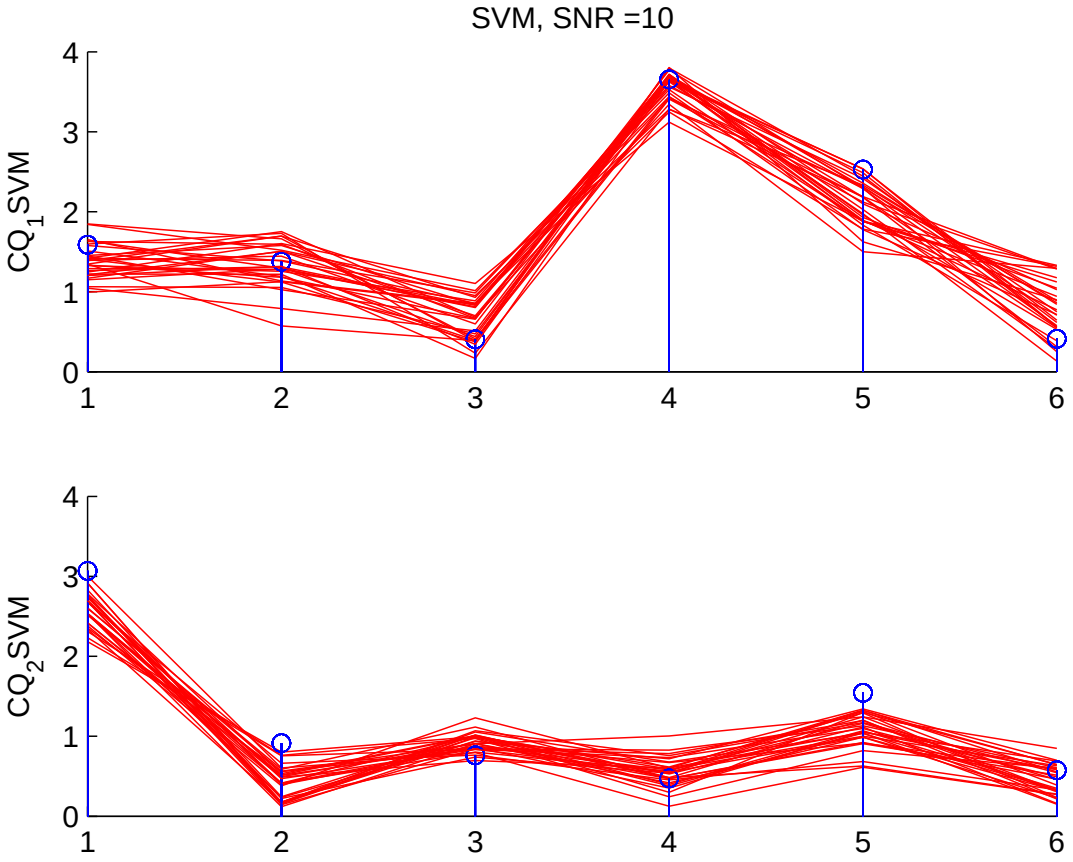


Figure 5-4. Blind estimation of basis coefficients $\{c_q(l)\}_{q=1}^{Q}$ when $N = 80, L = 3$, $Q = 2$, and SNR= $10dB$ (50 trials) using SVM.

CHAPTER 6
CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

This dissertation aims at exploring normal signal processing methods to estimate sparse channels. Before this research had started, L_1-norm regularization method (BP) and matching pursuit (MP) algorithms are widely used methods in many sparse applications. In chapter 2, the MP method and L_1-norm regularization are briefly summarized to understand the conventional sparse solutions.

Previous work in [16] formulates sparse channel estimation as the optimal mean square error (MSE) estimation of the channel impulse response regularized with a L_1-norm constraint. In chapter 3, L_1-norm of the channel impulse response, which play an important role in achieving the sparse solution, is estimated by using the convolution inequality for entropy and this information is exploited to improve the performance of sparse-LMS.

In chapter 4, support vector machines are briefly introduced including the classification and the regression problem. Then we present a new SVM-based sparse system identification algorithm. We reformulate the sparse channel estimation problem as a support vector regression (SVR) problem in which the channel coefficients are the Lagrange multipliers of the dual problem. By employing Vapnik's ϵ-insensitivity loss function, the solution is expanded in terms of a reduced number of Lagrange multipliers (i.e., the nonzero filter coefficients) and then a sparse solution is found. Then this SVR based method is also applied to derive an iterative algorithm for blind estimation of sparse single-input multiple-output (SIMO) channels, in particular when the channel order has been highly overestimated.

In chapter 5, the previous work [34, 35] is extended in the following direction: first, a robust algorithm for the blind estimation of a non-sparse channel when the channel order has been highly overestimated is derived; secondly, to avoid the high computational cost in solving a QP problem, a fast and simple algorithm called the Adatron [36] is used.

Combined iterative regression and SRM principle is proposed to solve blind sparse SIMO channel identification problem.

Time-varying channel estimation is also considered because it is a major obstacle to increase the capacity and reliability of wireless communication systems. For slow fading channels, adaptive algorithm for time-invariant such as recursive least-squares (RLS) and least mean-square (LMS) can give alternative but this adaptive algorithms diverge when channel variations exceed the convergence time of algorithm. In this case additional information of the time-varying channel is needed. Most models of time-varying channels treat the tap coefficients as uncorrelated stationary random processes. By using basis expansion model, estimation of time-varying channel system can be cast into a time-invariant estimation problem. TV impulse response of rapidly fading channels is expanded over a basis of complex exponentials that arise due to Doppler effects encountered with multipath environment. The TV taps are expressed as a superposition of TV bases with TI coefficients. In chapter 6, we propose to use the SVM technique with the exponential basis expansion model to identify the time-varying channel.

In appendix, to investigate the feasibility of the FPGA implementation of the proposed algorithm, the core computation of this Adatron is simulated in FPGA using System Generator which is a high-level design tool for Xilinx FPGAs that extends the capabilities of Simulink to include accurate modeling of FPGA circuits. The performance is also compared to the TI TMS320C33 DSP implementation.

6.2 Future Work

Support vector machine approach was successfully applied to sparse channel estimation problem including blind SIMO channel estimation in this research. High computational cost of the resulting quadratic programming problem is reduced by using Adatron algorithm to solve the SVR problem iteratively. The Adatron is a way of solving a batch problem in a sample-by-sample basis but it's not an on-line technique. Recently

an online learning method in reproducing Hilbert space by considering classical stochastic gradient descent in [67] so we should consider an on-line solution.

It is well known that SVM performance (estimation accuracy) depends on a good setting of meta-parameters C and ϵ. Optimal selection of these parameters are difficult because SVM model performance depends on all these parameters so we presented a SVM meta-parameter setting selection via the simulation. Schölkopf et al [68] suggest to control another parameter ν instead of ϵ so we should investigate the optimal parameter setting of SVM.

In basis expansion model (BEM) for time varying channel estimation, we assumed the knowledge of the base frequencies. This can by estimated using tests for cyclostationarity or adaptive maximum-likelihood methods [65, 66] so we should explore how to decide the basis of BEM.

FPGA implementation of SVM is simulated by mapping Adatron onto FPGA but it was not an efficient implementation because the way of solving is batch. Memory for storing kernel matrix was needed and we should wait until the computations of last sample so we should investigate an efficient FPGA implementation algorithm based on the online solution in [67].

Summarizing the future works,

- Investigate the on-line algorithm
- ν-SVM and optimal selection of SVM meta-parameters
- Estimation of the basis frequencies in BEM model.
- Optimized FPGA implementation of SVM

APPENDIX A
FPGA IMPLEMENTATION OF ADATRON ALGORITHM USING SYSYEM GENERATOR

A.1 Introduction

There has been considerable recent progress in tool development to support DSP applications in FPGAs. System Generator is a high-level design tool for Xilinx FPGAs that extends the capabilities of Simulink to include accurate modeling of FPGA circuits [69]. System Generator provides Simulink libraries for arithmetic and logic functions, memories, and DSP functions. System Generator provides abstractions for these resources. As we saw in chapter 4, calculations for the Adatron algorithm 4–16 can be implemented in below matlab script:

```
for i=1:length(X)
        temp1=alpha-alpha_ast;
        delta_alpha(i)=eta*(-K(i,:)*temp1+y(i)-epsilon-bias_ada);
        if alpha(i)+delta_alpha(i)<0
            alpha(i)=0;
        else
            alpha(i)=alpha(i)+delta_alpha(i);
        end

        delta_alpha_ast(i)=eta*(K(i,:)*temp1-y(i)-epsilon+bias_ada);
        if alpha_ast(i)+delta_alpha_ast(i)<0
            alpha_ast(i)=0;
        else
            alpha_ast(i)=alpha_ast(i)+delta_alpha_ast(i);
        end

        bias_ada=bias_ada+delta_alpha(i)-delta_alpha_ast(i);

 end
```

Figure A-1. Matlab implementation of Adatron algorithm

In this appendix, we use System Generator to map the Adatron algorithm shown above onto a FPGA and investigate the feasibility of FPGA implementation of the algorithm. Also the performance will be compared to TI TMS320C33 DSP implementation.

A.2 Adatron Engine in System Generator

A System Generator model of a Adatron engine block is shown in Fig. A-2. Address control logic block generates the control signals (address, write enable) of the memory for storing the input, kernel evaluations and the Lagrange multipliers. Kernels can be precalculated in software and each row of kernels can be feed through the inp gateway in. The Lagrange multipliers are stored in the single port ram1, while the input update is stored in the workspace. The Adatron engine in Fig. A-2 calculates the update of the Lagrange multipliers using the data available in the shared memory. Two outputs, out1 and out2, from the Ada engine block correspond to α_n and α_n^* in Adatron algorithm, and are restored in workspace1 thru the gateway out.

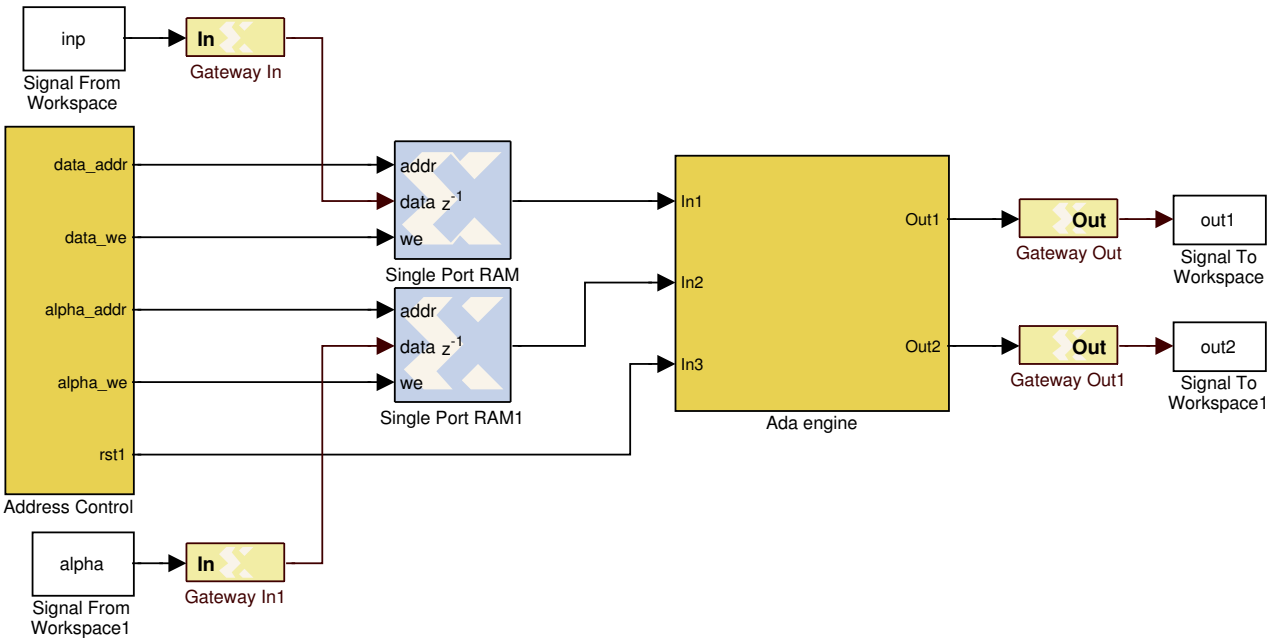


Figure A-2. Top design block of Adatron FPGA implementation using Xilinx System Generator

A.2.1 Address Control Logic

Fig. A-3 shows the detail logics in the address control block. Counters are for generating the address of the memories which stores the input data and the Lagrange multipliers. System Generator provides an interface to the embedded processor into the design so that Software can perform read/write operations to a shared memory through named association. Data write enable and alpha write enable are the shared memories that enables the writing to memory.

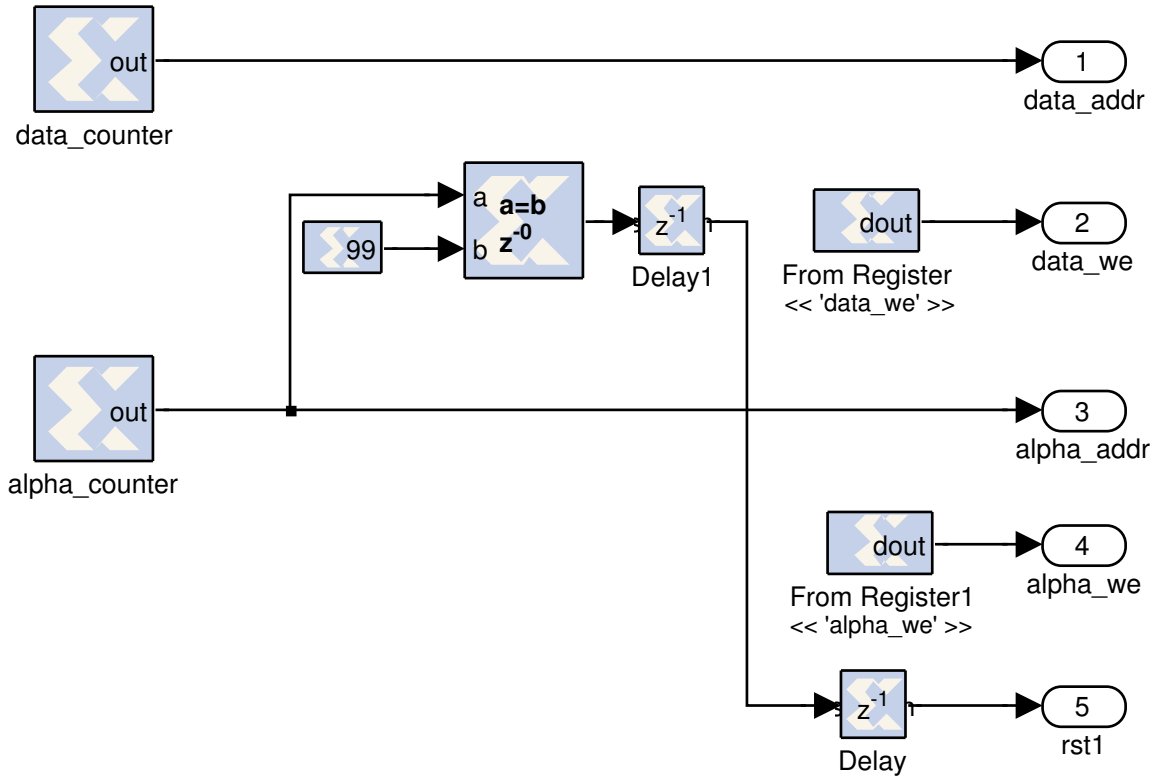


Figure A-3. Address control logic of the memory

A.2.2 Adatron Engine block

Fig. A-5 shows the detail building blocks of the below Adatron algorithm.

$$\delta\alpha_n = \eta\left(-\sum_{m=1}^{N}(\alpha_m - \alpha_m^*)\langle \mathbf{x}_n, \mathbf{x}_m\rangle + y_n - \epsilon - b\right),$$

$$\text{If } (\alpha_n + \delta\alpha_n) < 0 \text{ then set } \alpha_n = 0, \text{else } \alpha_n \longleftarrow \alpha_n + \delta\alpha_n$$

$$\delta\alpha_n^* = \eta\left(\sum_{m=1}^{M}(\alpha_m - \alpha_m^*)\langle \mathbf{x}_n, \mathbf{x}_m\rangle + y_n - \epsilon + b\right),$$

$$\text{If } (\alpha_n^* + \delta\alpha_n^*) < 0 \text{ then set } \alpha_n^* = 0, \text{else } \alpha_n^* \longleftarrow \alpha_n^* + \delta\alpha_n^*$$

Mult and accumulator calculates the sum of multiplications of input and Lagrange multipliers and result is provided to MCode block. Other calculations such as adding, subtracting, and comparing to compute $\delta\alpha$ is executed in calalpha MCode block and is shown in Fig. A-4. We can see that configuring the Xillinx MCode block is an easier solution than building the logic together through Xilinx blockset logic.

```
function alpha = calalpha(palpha, eta, xLag, y, en,
epsilon, b)

delta = eta*(palpha + xLag + y - epsilon - b);
if delta < 0
    alpha = 0;
else
    alpha = palpha + delta;
end
```

Figure A-4. MCode block configuration for updating α

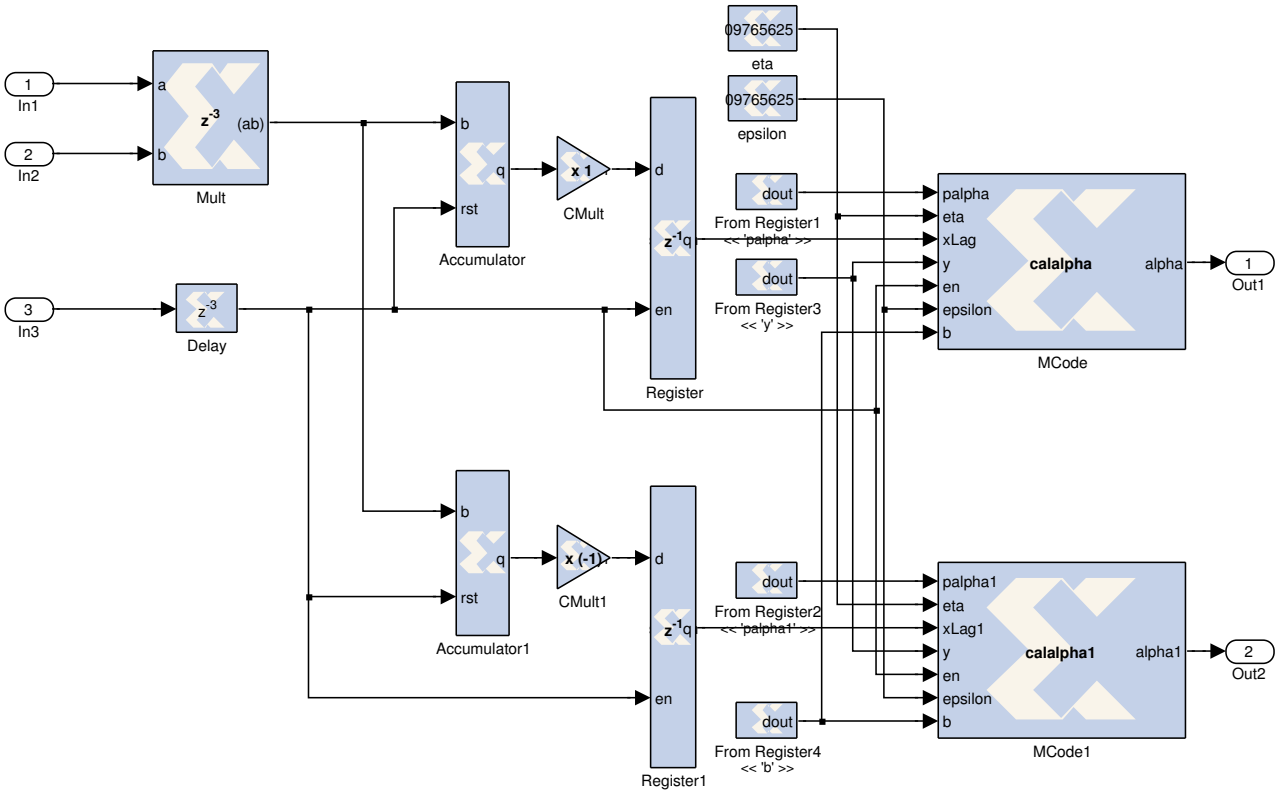


Figure A-5. Adatron Engine block detail

A.3 Implementation Results

All results were obtained using the System Generator V8.1 [70], Xilinx ISE 8.1i [71, 72], and XST synthesis tool to target a Virtex4 xc4vsx35-10 part. All models use dedicated multipliers, block memory for storage, and 16-bit precision. FPGA device implementation results are summarized in table A-1. The basic building block of Xilinx Virtex devices is the logic cell (LC). According to the Xilinx Virtex data sheet [73], an LC includes a 4-input look up table (LUT), carry logic, and a storage element. The Xilinx Virtex-E architecture contains configurable logic blocks (CLBs). Each Virtex-E CLB contains four LCs and a CLB consists of two slices.

If we have an N input data sample, we will have an $N \times N$ kernel matrix and N Lagrange multipliers. Suppose adaptation needs 50 iterations for the Adatron to converge and 100 input data samples. Then it takes $(N+7) \times 50 (= 5350)$ clock cycles to evaluates the updates of the Lagrange multipliers while it takes $(2N+25) \times 50 (= 11250)$ clock cycles with TI TMS320C33 DSP implementation [74, 75]. 5350 cycles corresponds to $26.75\mu s$ at 200MHz clock in the Xilinx ISE 8.1i. . Fig A-6 shows the implementation of the basic computation of Adatron algorithm with TI TMS320C33 assembly language. Note that we can increase the performance of the FPGA implementation by parallelizing the operation.

Table A-1. Device utilization summary

Selected Device	4vsx35ff668-10
Number of Slices	278
Number of Slice Flip Flops	210
Number of 4 input LUTs	445
Number of FIFO16/RAMB16s	2
Number of GCLKs	1
Number of DSP48s	1

In this appendix, we use System Generator to map the Adatron algorithm shown above onto a FPGA resource including address generation logic, memory, Adatron engine, and the logic fabric of Virtex FPGAs. Also the performance is compared to TI TMS320C33 DSP implementation. Results showed that the FPGA implementation of Adatron is feasible and has more design margin than the DSP implementation.

```
* compute K(i,:)*temp1
        ldi     @y, r0
        ldf     Epsilon, r1
        ldf     bias, r2
        ldf     eta, r3
        ldf     0.0, r4
        ldi     @alpha, r5
        ldi     @kaddr, ar1
        ldi     @taddr, ar2

        ldi     N, rc
        rptb    eloop
        mpyf3   *ar1++%,*ar2++,r7
eloop   addf    r7, r4

        addf    r0, r4
        subf    r1, r4
        subf    r2, r4
        mpyf    r3, r4
        addf    r5, r4
        cmpf    0, r4
        bnz     next
        ldf     0, r4
next    stf     r4, @delta
```

Figure A-6. TI DSP (TMS320C33) implementation of the Adatron algorithm

REFERENCES

[1] Shane F. Cotter and Bhaskar D. Rao, "Sparse channel estimation via matching pursuit with application to equalization," *IEEE Tran. on Communications*, vol. 50, no. 3, pp. 374–377, 2002.

[2] Veropoulos K., *Machine Learning Approaches to Medical Decision Making*, Ph.D. dissertation, University of Bristol, Bristol, UK, 2001.

[3] W.F. Schreiber, "Advanced television systems for terrestirial bradcasting: Some problems and some proposed sotutions," in *Proc. IEEE*, June 1995, vol. 83, pp. 958–981.

[4] S. Ariyavisitakul, N.R. Sollenberger, and L.J. Greenstein, "Tap selectable decision-feedback equalization," *IEEE Tran. on Commun.*, vol. 45, pp. 1497–1500, Dec 1997.

[5] M. Kocic et al., "Sparse equalization for real-time digital underwater acoustic communications," in *Proc. OCEANS'95*, San Diego,CA, Oct 1995, pp. 1417–1500.

[6] R. R. Coifman and M. V. Wickerhauser, "Entropy-based algorithms for best-basis selection," *IEEE Tran. on Info. Theory*, vol. 38, pp. 713–718, 1992.

[7] I. Daubechies, *Ten lectures on Wavelets*, SIAM, 1992.

[8] S.G.Mallat and Z.Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Tran. on ASSP*, vol. 41, no. 12, pp. 3397–3415, Dec 1993.

[9] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1999.

[10] R. Gribonval, "Analysis of sound signals with high resolution matching pursuit," 1996.

[11] O. Al-Shaykh, E. Miloslavsky, T. Nomura, R. Neff, and A. Zakhor, "Video compression using matching pursuits," 1999.

[12] B. D. Rao, "Signal processing with the sparseness constraint," in *Proceedings of ICASSP'98*, Seattle, WA, May 1998, vol. III, pp. 1861–1864.

[13] K. Kreutz-Delgado and B. D. Rao, "Sparse basis selection, ica, and majorization: towards a unified perspective," in *Proceedings of ICASSP'99*, March 1999, vol. 2, pp. 1081–1084.

[14] K. G. Gauss, *Theory of Motion of Heavenly Bodies*, Dover, New York, 1963.

[15] Peter Stoica, Yngve Selen, and Jian Li, "On information criteria and the generalized likelihood ratio test of model order selection," *IEEE Signal Processing Letters*, vol. 11, no. 10, Oct 2004.

[16] Yadunandana N. Rao, Sung-Phil Kim, Justin C. Sanchez, Deniz Erdogmus, Jose C. Principe, Jose M. Carmena, Mikhail A. Lebedev, and Miguel A. Nicolelis, "Learning mappings in brain machine interfaces with echo state networks," in *Proceedings of ICASSP'05*, Philadelphia, PA, Mar 2005.

[17] Dongho Han, Sung-Phil Kim, and Jose C. Principe, "Sparse channel estimation with regularization method using convolution inequality for entropy," in *Proceedings of IJCNN'05*, Montreal, Quebec, Canada, 2005.

[18] J.-F. Bercher and C. Vignat, "A renyi entropy convolution inequality with application," in *Proceedings of EUSIPCO'02*, Tolouse, France, 2002.

[19] Deniz Erdogmus, Kenneth E. Hild, Jose C. Principe, Marcelino Lazaro, and Ignacio Santamaria, "Adaptive blind deconvolution of linear channels using renyi's entropy with parzen window estimation," *IEEE Tran. on Signal Processing*, vol. 52, no. 6, pp. 1489–1498, Jun 2004.

[20] Y. Sato, "A method of self-recovering equalization for multilevel amplitude-modulation," *IEEE Tran. on Commun.*, vol. COM-23, no. 6, pp. 679–682, June 1975.

[21] A. Benveniste and M. Goursat, "Blind equalizers," *IEEE Tran. on Commun.*, vol. COM-23, no. 8, pp. 871–883, Aug 1985.

[22] C. Johnson, J. Schniter, T. Endres, J. Behm, R. Casas, V. Brown, and C. Berg, "Blind equalization using the constant modulus criterian: A review," 1997.

[23] H. Liu, G. Xu, L. Tong, and T. Kailath, "Recent developments in blind channel equalization: From cyclostationarity to subspaces," 1996.

[24] E. Serpedin and G. Giannakis, "Blind channel identification and equalization with modulation induced cyclostationarity," 1997.

[25] L. Tong and S. Perreau, "Multichannel blind identification: From subspace to maximum likelihood methods," 1998.

[26] L. Tong, G. Xu, B. Hassibi, and T. Kailath, "Blind channel identification based on second-order statistics: A frequency-domain approach," 1995.

[27] G. Xu, H. Liu, L. Tong, and T. Kailath, "A least-squares approach to blind channel identification," 1995.

[28] Guanghan Xu, Hui Liu, Lang Tong, and Thomas Kailath, "Blind identification and equalization based on second-order statistics: A time domain approach," *IEEE Tran. on Information Theory*, vol. 40, pp. 340–349, Mar 1994.

[29] E. Moulines, P. Duhamel, J. F. Cardoso, and S. Mayrague, "Subspace methods for the blind identification of multichannel fir filters," *IEEE Tran. on Signal Processing*, vol. 43, pp. 516–525, Feb 1995.

[30] Houcem Gazzah, Phillip A. Regalia, Jean-Pierre Delmas, and Kaim Abed-Meraim, "A blind multichanel identification algorithm robust to order overestimation," *IEEE Tran. on Signal Processing*, vol. 50, no. 6, pp. 1449–1458, June 2002.

[31] A. Safavi and K. Abed-Meraim, "Blind channel identification robust to order overestimation: A constant modulus approach," in *Proceedings of ICASSP'03*, Hong Kong, China, Apr 2003, vol. IV, pp. 313–316.

[32] V. Vapnik, *The Nature of Statistical Learninig Theory*, Springer-Verlag, 1995.

[33] Federico Girosi, "An equivalence between sparse approximation and support vector machines," Tech. Rep. AIM-1606, 1997.

[34] Dongho Han, Jose C. Principe, Liuqing Yang, Ignacio Santamaría, and Javier Vía, "Structural risk minimization for robust blind identification of sparse simo channels," in *Proceedings of IEEE SPAWC'05*, New York, NY, June 2005.

[35] Ignacio Santamaría, Javier Vía, and César C. Gaudes, "Robust blind identification of SIMO channels: A support vector regression approach," in *Proceedings of ICASSP'04*, 2004, pp. 673–676.

[36] I. Santamaría, R. González, C. Pantaleón, and J.C. Principe, "Maximum margin equalizers trained with the adatron algorithm," *Signal Processing*, vol. 83, no. 3, pp. 593–602, 2003.

[37] Dongho Han, Jose C. Principe, Liuqing Yang, Ignacio Santamaría, and Javier Vía, "Robust blind simo channel estimation using adatron," in *Proceedings of ICASSP'06*, Toulouse, France, May 2006.

[38] D.G. Luenberger, *Linear and Non-linear Programming*, Addison Wesley, 1989.

[39] J.C. Principe, D. Xu, and J. Fisher, "Information theoretic learning," in *Unsupervised Adaptive Filtering*, Simon Haykin, Ed., pp. 265–319. JOHN WILEY, 2000.

[40] J.C. Principe, N.R. Euliano, and W.C. Lefebvre, *Neural and Adaptive Systems*, JOHN WILEY, 2000.

[41] A.J. Smola and B. Schoelkopf, "A tutorial on support vector regression," 1998.

[42] Tomaso Poggio and Federico Girosi, "A sparse representation for function approximation," *Neural Computation*, vol. 10, no. 6, pp. 1445–1454, 1998.

[43] Edgar Osuna, Robert Freund, and Federico Girosi, "Support vector machines: Training and applications," Tech. Rep. AIM-1602, 1997.

[44] Wang XZ, He Q, and Chen DG, "A genetic algorithm for solving the inverse problem of support vector machines," *NEUROCOMPUTING*, pp. 225–238, Oct 2005.

[45] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," in *Data Mining and Knowledge Discovery*, 1998, pp. 121–167.

[46] S.R.Gunn, "MATLAB svm toolbox available at http://www.isis.ecs.soton.ac.uk/isystems/kernel/," University Southampton, Image Speech and Intelligent Systems Research Group, Southampton, U.K., 1998.

[47] R. Fletcher, *Practical Methods of Optimization*, John Wiley and Sons, 2nd edtion, 1987.

[48] G. P. McCormick, *Nonlinear Programming: Theory, Algorithms and Applications*, John Wiley and Sons, 1983.

[49] D. Mattera and S. Haykin, "Support vector machines for dynamic reconstruction of a chaotic system," in *Advances in Kernel Methods: Support Vector Machine*, B. Sch Ed. MIT Press, 1999.

[50] A. Smola, N. Murata, B. Scholkopf, and K. Muller, "Asymptotically optimal choice of ϵ-loss for support vector machine," in *Proc. ICANN 1998*, 1998.

[51] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, pp. 321–377, 1936.

[52] M. F. Kahn, W. A. Gardner, and M. A. Mow, "Programmable canonical correlation analysis: a flexible framework for blind adaptive spatial filtering," in *Twenty-Ninth Asilomar Conf. Signals, Syst., Comput.*, Oct 1995, pp. 351–356.

[53] L. L. Scharf and J. T. Thomas, "Wiener filters in canonical coordinates for transform coding, filtering, and quantizing," *IEEE Tran. on Signal Processing*, vol. 46, no. 3, pp. 647–654, March 1998.

[54] Francis R. Bach and Michael I. Jordan, "Kernel independent component analysis," *Journal of Machine Learning Research*, vol. 3, pp. 1–48, 2002.

[55] O. Friman, M. Borga, P. Lundberg, and H. Knutsson, "Adaptive analysis of fMRI data," *NeuroImage*, vol. 19, no. 3, pp. 837–845, 2003.

[56] F. Pérez-Cruz, A. Navia-Vázquez, P. Alarcón-Diana, and A. Artés-Rodríguez, "An irwls procedure for svr," in *Proceedings of the EUSIPCO'00*, Tampere, Finland, Sept 2000.

[57] C.C. Gaudes, J. Via, and I. Santamaria, "An irwls procedure for robust beamforming with sidelobe control," in *Proceedings of the Third IEEE Sensor Array and Multichannel Signal Processing Workshop*, Sitges, Spain, July 2004.

[58] J. K. Anlauf and M. Biehl, "The AdaTron: an adaptive perceptron algorithm," *Europhysics Letters*, vol. 10, pp. 687–692, Dec. 1989.

[59] M. Visintin, "Karhunen-loève expansion of a fast rayleigh fading process," *IEE Electronics Letters*, vol. 32, pp. 1712–1713, Aug 1996.

[60] D.K. Borah and B.D. Hart, "Frequency-selective fading channel estimation with a polynomial time-varying channel model," *IEEE Tran. on communications*, vol. 47, pp. 862–873, June 1999.

[61] Y. Liu and D. K. Borah, "Estimation of time-varying frequency-selective channels using a matching pursuit technique," in *Wireless Communications and Networking*, Mar 2003, vol. 2, pp. 941–946.

[62] Georgios B. Giannakis and Cihan Tepedelenlioglu, "Basis expansion models and diversity techniques for blind identification and equalization of time-varying channels," in *Proceedings of the IEEE*, Oct 1998, vol. 86, pp. 1969–1986.

[63] Hui Liu and Georgios B. Giannakis, "Deterministic approaches for blind equalization of time-varying channels with antenna arrays," *IEEE Tran. on Signal Processing*, vol. 46, no. 11, pp. 3003–3013, November 1998.

[64] M. Martone, "Multiresoluton sequence detection in rapidly fading channels based on focused wavelet decomposition," *IEEE Tran. on communications*, vol. 49, pp. 1388–1401, Aug 2001.

[65] S. Barbarossa and A. Scaglione, "Time-varying fading channels," 2000.

[66] Gregory D. Durgin, "Theory of stochastic local area channel modeling for wireless communications," .

[67] Jyrki Kivinen, Alexander J. Smola, and Robert C. Williamson, "Online learning with kernels," *IEEE Tran. on Signal Processing*, vol. 52, no. 8, pp. 2165–2176, August 2004.

[68] B. Schólkopf, P. Bartlett, A. Smola, and R. Williamson, "Support vector regression with automatic accuracy control," in *Proceedings of ICANN98, Perspectives in Neural Computing*, L. Niklasson, M. Bodén, and T. Ziemke, Eds., pp. 111–116. Springer, Berlin, 1998, 1998.

[69] J. Hwang, B. Milne, N. Shirazi, and J. Stroomer, "Lecture notes in computer science," in *System Level Tools for DSP in FPGAs*, pp. 534–543. 2001.

[70] *Xilinx System Generator for DSP Version 8.1*, Xilinx, Inc., 2006.

[71] *Xilinx Intergrated Software Environment (ISE 8.1)*, Xilinx, Inc., 2006.

[72] *Xilinx Embedded Development Kit(EDK)*, Xilinx, Inc., 2006.

[73] *Virtex-4 User Guide*, Xilinx, Inc., 2006.

[74] *TMS320C3X Users guide*, Texas Instrument, 1997.

[75] Dongho Han, Yadunandana N. Rao, Jose C. Principe, and Karl Gugel, "Real-time pca (principal component analysis) implementation on dsp," in *Proceedings of IJCNN'04*, Budapest, Hungary, July 2004, pp. 2159–2162.

BIOGRAPHICAL SKETCH

Dongho Han was born in Seoul, Korea. He received his Bachelor of Science degrees in electrical engineering from the Yonsei University, Seoul, Korea, in 1993. From 1993 to 1998, he was a hardware engineer in Samsung Electronics where he developed several cpu boards for PBX system. He also received his Master of Science degree in electrical and computer engineering from the University of Florida in 2001. He was in the Computational NeuroEngineering Laboratory in the Electrical and Computer Engineering Department at the University of Florida during his Ph.D study. His present research interests are in the areas of signal processing, neural network, communication algorithms and their implementations in FPGA and DSP. While a Ph.D. candidate, Dongho interned with Motorola, where he returned upon graduation for full-time employment as a DSP engineer.