Abstract—A simple, yet powerful, learning method is presented by combining the famed kernel trick and the least-mean-square (LMS) algorithm, called the KLMS. General properties of the KLMS algorithm are demonstrated regarding its well-posedness in very high dimensional spaces using Tikhonov regularization theory. An experiment is studied to support our conclusion that the KLMS algorithm can be readily used in high dimensional spaces and particularly in reproducing kernel Hilbert spaces (RKHS) to derive nonlinear self-regularized, stable algorithms.

Index Terms—Least Mean Square, kernel methods, Tikhonov regularization.

I. INTRODUCTION

The solid mathematical foundation and considerable successes in practical problems of the support vector machines (SVM) [1] sparked a widespread interest in kernel methods as a means of solving classification and regression problems. However, almost all the kernel methods (SVM, regularization networks [2], kernel principal component analysis [3], etc.) are derived in batch mode and usually $O(N^3)$ operations along with $O(N^2)$ memory storage are required to perform the inversion or the singular value decomposition of the Gram matrices, which is very demanding for many real world problems. Recent work [4]-[7] is addressing this shortcoming and it is theoretically and practically interesting to exploit the computational-efficient online extensions.

Introduced in 1960 by Widrow and Hoff, the least-mean-square (LMS) is the workhorse of adaptive signal processing due to its simplicity and elegance [8]. Simple manipulation of its weight update equation shows that this algorithm can be expressed solely in term of inner products, which by the famed kernel trick, allows us readily to construct different nonlinear versions. We call this algorithm the kernel least mean
square (kernel LMS or KLMS) [9].

An issue of well-posedness follows immediately since the algorithm is formulated in very high dimensional spaces with finite training data available. The concept of well-posedness is proposed by Hadamard [10]. Regularization as a remedy for ill-posedness became widely known due to the work of Tikhonov [11] and also from a Bayesian learning perspective [12]. In solving least squares (LS) problems, the Tikhonov regularization is essentially a trade-off between fitting training data and reducing solution norms. Consequently it reduces the sensitivity of the solution to small changes in the training data and imposes stability on the ill-posed problem. Moreover, the significance of well-posedness related to generalization ability has been also revealed recently in statistical learning theory [1], [13]. Following this line of reasoning, an algorithm similar to the KLMS (a leaky version) has been introduced in [4] where a stochastic gradient approach is adopted to solve the regularized risk functional in a reproducing kernel Hilbert space (RKHS). The analysis in [4] only focuses on the probabilistic upper bound for convergence in a statistical learning framework, and concludes that the regularization (or the leaky) term appears essential for its well-posedness.

In this paper, we employ the conventional LMS theory to prove that the kernel LMS algorithm is well-posed and does not need the extra regularization as proposed in [4]. With this message at the core, the asymptotic behavior of the KLMS is examined in the framework of the small-step-size theory [8], which shows that it always seeks a solution in the data subspace (normally of small dimension) and more importantly it has a self-regularization mechanism due to its different convergence speeds along different eigen-directions. Furthermore a bound on the solution norm is established on the basis of the $H^\infty$ stability theory [8], which concludes directly the well-posedness of the KLMS algorithm with finite data training.

The organization of the paper is as follows. In section II, a formulation of the KLMS algorithm is presented. Next in section III, the asymptotic convergence behaviors are studied and important insights into its self-regularization mechanism are gained. The bound on the solution norm is established and its
implication is elaborated in section IV. An experiment is studied in section V to support our theory. Finally section VI summarizes the conclusions and future lines of research.

II. FORMULATION OF THE KERNEL LEAST MEAN SQUARE ALGORITHM

A. Learning problem setting
Suppose the goal is to learn a function $f : U \rightarrow R$ based on a sequence $((u_i, y_i), \ldots, (u_N, y_N)) \in Z^N$ of examples $(u_i, y_i) \in Z = U \times Y$. $Y$ is a compact subset of $R$ and $U$ is assumed as a compact subspace of $R^M$. A quadratic loss is chosen here. The problem statement is to find a function $f$ in a pre-specified hypothesis space $H$, such that the following empirical risk is minimized

$$R_{\text{emp}}[f \in H, Z^N] = \sum_{i=1}^{N} (y_i - f(u_i))^2 \quad (1)$$

There are several ways to solve this problem. If $H$ is the space of all the linear operators on $U$, the linear LS solution applies [14]. Correspondingly, the steepest descent or LMS algorithms can be employed for on-line (i.e. sample by sample) estimation. If $H$ is some RKHS, (1) can be solved by the radial basis function networks (RBF) [15], or kernel regression [16]. The appeal of this alternative is that linear solutions in $H$ normally correspond to nonlinear solutions in $U$. Only recently there has been interest in pursuing on-line algorithms in RKHS [4], [7]. The two most prominent papers suggest utilizing regularized cost functions instead of (1) as will be fully discussed in section IV. The motivation for this paper is to derive an appropriate LMS algorithm for (1) in RKHS, and show that the norm of the solution is always bounded, meaning that the algorithm is always stable and does not need regularization.

B. Least Mean Square (LMS) algorithm
In the LMS algorithm the hypothesis space $H_1$ consists of all the linear operators on $U$, denoted by $w : U \rightarrow R$. Since $U$ is assumed to be an $M$ dimensional Hilbert space, the linear operator becomes the standard inner product and $H_1$ is the $M$ dimensional dual space of $U$.

The LMS algorithm starts to minimize the following empirical risk:
\[
\min_{w} R_{\text{emp}}[w \in H_1, Z_N] = \sum_{i=1}^{N} (y_i - w(u_i))^2
\]

Using the stochastic gradient, the well-known LMS algorithm is

\[
w_0 = 0 \\
e_n^* = y_n - w_{n-1}(u_n) \\
w_n = w_{n-1} + \eta e_n^* u_n
\]

where \(e_n^*\) is called the apriori error and \(\eta\) is the step size.

The repeated application of the weight-update equation yields

\[
w_n = \eta \sum_{i=1}^{n} e_i^* u_i
\]

Therefore, after \(n\)-step training, the weight is expressed as the linear combination of the previous and present input data weighted by the apriori errors. More importantly, the input-output of this learning system can be solely expressed in term of inner products

\[
y = w_n(\tilde{u}) = \eta \sum_{i=1}^{n} e_i^* < u_i, \tilde{u} >_U
\]

\[
e_n^* = y_n - \eta \sum_{i=1}^{n-1} e_i^* < u_i, u_n >_U
\]

Therefore, by the kernel trick, the LMS algorithm can be readily extended to nonlinear versions [9].

C. Reproducing Kernel Hilbert Space

A kernel [17] is a continuous, symmetric, positive-definite function \(\kappa : U \times U \to R\). The commonly used kernels include

1) Radial basis kernel

\[
\kappa(u_i, u_j) = \exp(-a \|u_i - u_j\|^2)
\]

2) Polynomial kernel

\[
\kappa(u_i, u_j) = (u_i^T u_j + 1)^p
\]

By the Mercer’s theorem [17] there exists a mapping \(\Phi\) such that

\[
\kappa(u_i, u_j) = \Phi(u_i)^T \Phi(u_j), \quad \forall u_i, u_j \in U
\]

Usually \(\Phi\) is treated as a nonlinear mapping and \(\Phi(u)\) is the transformed feature vector lying in the feature space \(F\) (which is a Hilbert space). Equation (9) can be simply interpreted as the usual dot product in the
vector space. \( F \) is isometric-isomorphic to the RKHS induced by the kernel. Denote this RKHS as \( H_3 \), which is a function space satisfying the following properties [17]:

1) \( H_3 \) is the closure of the span of all \( \kappa(u,\cdot) \) with \( u \in U \). In other words,

\[
f = \sum_{u \in U} \alpha_u \kappa(u,\cdot), \quad \forall f \in H_3
\]

where \( \{\kappa(u,\cdot) \mid u \in U\} \) is the kernel basis and \( \alpha_u \) is the real coefficients.

2) \( \kappa \) has the reproducing property

\[
\langle f, \kappa(u,\cdot) \rangle_{H_3} = f(u), \quad \forall f \in H_3, \forall u \in U
\]

3) An interesting special case follows as

\[
\langle \kappa(u_1,\cdot), \kappa(u_2,\cdot) \rangle_{H_3} = \kappa(u_1, u_2), \quad \forall u_1, u_2 \in U
\]

4) The inner product induces the standard norm

\[
\| f \|^2_{H_3} = \langle f, f \rangle_{H_3}, \quad \forall f \in H_3
\]

It is easy to check that \( H_3 \) is essentially the same as \( F \) by identifying \( \Phi = \kappa(u,\cdot) \), which are the bases of the two spaces respectively and preserve the inner product. By identifying \( F \) and \( H_3 \) respectively, we can better understand the role of the kernel LMS as a link between the LMS algorithm and the RBF as demonstrated below.

\[D. \text{ Kernel LMS Algorithm}\]

Now we are ready to derive the KLMS algorithm. We utilize the Mercer’s theorem to transform the data \( u_i \) into the feature space \( F \) as \( \Phi(u_i) \). Let the hypothesis space \( H_2 \) be the dual space of \( F \). Then the KLMS is nothing but the LMS performed on the data sequence \( \{(\Phi(u_1), y_1), \ldots, (\Phi(u_N), y_N)\} \) which is summarized as follows:

\[
\Omega_0 = 0 \\
e_n = y_n - \Omega_n(\Phi(u_n)) \\
\Omega_n = \Omega_{n-1} + \eta e_n \Phi(u_n)
\]

It may be computationally difficult to have direct access to the weight and the transformed data in the feature space. Using (5), (6) and (9), we have
The final input-output relation (after $N$ step training) of the learning algorithm is

$$\Omega_N = \eta \sum_{i=1}^{N} e_i \Phi(u_i)$$

and

$$\tilde{y} = \eta \sum_{i=1}^{N} e_i \kappa(u_i, \tilde{u})$$

This formulation bears resemblance to the RBF networks. First of all, this algorithm is a stochastic gradient approximation to minimize the following risk

$$\min_{\Omega \in \mathcal{H}^2} \sum_{i=1}^{N} (y_i - \Omega(\Phi(u_i)))^2$$

which is equivalent to

$$\min_{f} R_{\text{emp}}[f \in \mathcal{H}^2, Z^N] = \sum_{i=1}^{N} (y_i - f(u_i))^2$$

The equivalence is easily recognized by identifying

$$f = \sum_{u \in \mathcal{U}} \alpha \kappa(u, \cdot) = \sum_{a \in \mathcal{A}} \alpha \Phi(u_a) = \Omega$$

Notice the equation in (21) really means the isometric isomorphism between $\mathcal{H}^2$ and $\mathcal{H}_f$ (and $f = \Omega \circ \Phi$ will be more rigorous). The optimal solution provided by the KLMS is linear in $\mathbf{F}$ but nonlinear in $\mathbf{U}$.

A question follows immediately: does the KLMS converge in such a high-dimensional space? The answer is actually two fold: 1) Is the solution of the KLMS close to the least squares (LS) solution when (19) is well-posed? 2) Does the KLMS give reasonable approximation when (19) itself is ill-posed? Before we answer these questions for the KLMS algorithm, it is necessary to briefly review the least squares method and Tikhonov regularization theory applied to LS, as well as to establish the notation used throughout the paper.

### E. Least Squares and Regularization Theory

In LS the multiple linear regression model is hypothesized as

$$y_n = \Omega^\top (\Phi_n) + \nu(n)$$
From now on, $\Phi$ is used for $\Phi(u_x)$ unless the role of $u_x$ is emphasized and $v(n)$ is the modeling uncertainty.

Denote the transformed data matrix as $D_\Phi^{T} = [\Phi(u_1), \Phi(u_2), \ldots, \Phi(u_x)]$, the correlation matrix

$$R_\Phi = D_\Phi^{T}D_\Phi / N$$

(23)

the cross correlation vector $p_\Phi = \sum_{i=1}^{N} \Phi(u_i) y_i / N = D_\Phi^{T}y / N$ and the Gram matrix

$$G_\Phi = D_\Phi D_\Phi^{T} = [\kappa(u_i, u_j)]_{N \times N}$$

(24)

Then the LS solution is known to satisfy the following normal equations [8]

$$R_\Phi \hat{\Omega} = p_\Phi$$

(25)

Unfortunately the LS problem is not always well-posed. According to Hadamard [10], a problem is well-posed if the solution exists, is unique, and smoothly depends on data and parameters. More specifically the following theorems demonstrate how to get a well-posed LS algorithm.

**Theorem 2.1**: (uniqueness theorem of LS [18]) The least-squares estimate $\hat{\Omega}$ is unique if and only if the correlation matrix $R_\Phi$ is non-singular and the solution is given by

$$\hat{\Omega} = R_\Phi^{-1} p_\Phi$$

(26)

**Theorem 2.2**: (pseudo-inverse [18]) If $R_\Phi$ is singular, there are infinitely many solutions to (19). Out of them there is a unique minimum-norm solution given by

$$\hat{\Omega} = D_\Phi^{+} \tilde{y}$$

(27)

Here $D_\Phi^{+}$ is the general pseudo-inverse of $D_\Phi$ given by

$$D_\Phi^{+} = P \begin{bmatrix} S^{-1} & 0 \\ 0 & 0 \end{bmatrix} Q^T$$

(28)

and matrices $P$, $Q$, $S$ are given by the singular value decomposition (SVD) of $D_\Phi$ [14]

$$D_\Phi = Q \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} P^T$$

(29)
where \( P, Q \) are orthogonal matrices and \( S = \text{diag}\{s_1, s_2, ..., s_r\} \) assuming the rank of \( D_\phi \) is \( r \). \( s_i \) are the singular values of \( D_\phi \) and assumed \( s_1 \geq s_2 \geq ... \geq s_r \geq 0 \).

Remarks: the solution (27) includes (26) as a special case and together can be expressed as

\[
\hat{\Omega} = R_\phi^+ P_\phi
\]  

(30)

Equation (30) will be very important to understand the regularization of the KLMS.

The designer's job is finished if we are in the ideal world, but in the real world more caution should be taken. The fundamental issues are the finite-precision of our computers and the noise—universal thermal noise, measurement errors, modeling uncertainty, etc. For example, when the modeling uncertainty \( \nu(n) \) in (22) is white with zero mean and constant variance \( \sigma^2 \), the covariance matrix of the LS estimate by (27) is

\[
\text{cov}[\hat{\Omega}] = \sigma^2 P \begin{bmatrix} S^{-2} & 0 \\ 0 & 0 \end{bmatrix} P^T
\]  

(31)

If some singular values are too small, the variance of the estimate is pessimistically large. On the other hand, too small singular values also cause catastrophic numerical errors due to the finite-precision [14]. The typical symptoms of this type of ill-posedness are large norm solutions and extreme sensitivity to small changes in training data.

The Tikhonov regularization is widely used to address this problem. Since small norm solutions are desirable, a regularization term is introduced in the LS cost function which penalizes the solution norm

\[
\min_{\Omega} R_{\text{reg}}[\Omega \in H_2, Z^N] = \| \tilde{y} - D_\phi \Omega \|_F^2 + \lambda \| \Omega \|_F^2
\]  

(32)

In the Bayesian interpretation, the error square term is the likelihood and the regularized term is some priori knowledge on the norm of the solution [12].

**Theorem 2.3**: (Tikhonov regularization [11]) The Tikhonov regularized LS solution is

\[
\hat{\Omega}_{\text{TR}} = (D_\phi^T D_\phi + \lambda I)^{-1} D_\phi^T \tilde{y}
\]  

(33)

Remarks: More insights can be obtained through the SVD,
\[
\hat{\Omega}_{TR} = \text{Pdiag}\left(\frac{s_j}{s_j^2 + \lambda}, \ldots, \frac{s_r}{s_r^2 + \lambda}, 0, \ldots, 0\right)Q^T \hat{y} \tag{34}
\]

Compare it with the pseudo-inverse solution

\[
\hat{\Omega} = \text{Pdiag}(s_1^{-1}, \ldots, s_r^{-1}, 0, \ldots, 0)Q^T \hat{y} \tag{35}
\]

We see that the Tikhonov regularization modifies the singular values through the following regularization function (reg-function):

\[
H_{TR}(x) = \frac{x^2}{x^2 + \lambda} \tag{36}
\]

Notice that \(H_{TR}(x) \to 1\) when \(x\) large and \(H_{TR}(x) \to 0\) when \(x\) small. In that sense, the Tikhonov regularization smoothly filters out the singular components that are small (relative to \(\lambda\)). This viewpoint actually brings new understanding into all kinds of regularization techniques.

**Theorem 2.4**: (Truncated pseudo-inverse regularization) If the following hard cut-off reg-function

\[
H_{HC}(x) = \begin{cases} 
1 & \text{if } x > a \\
0 & \text{if } x \leq a
\end{cases} \tag{37}
\]

is applied, where \(a\) is the cut-off threshold and satisfies \(s_m > a \geq s_{m+1}\), then the solution becomes

\[
\hat{\Omega}_{HC} = \text{Pdiag}(s_1^{-1}, \ldots, s_m^{-1}, 0, \ldots, 0)Q^T \hat{y} \tag{38}
\]

Remarks: This kind of regularization is equivalent to applying principal component analysis to the data and using the first \(m\) principal components to represent the original data. Under reasonable signal-noise-ratio, the small singular components are purely associated with the noise, i.e. the spurious features. The so-called ‘over-learning’ phenomenon means the LS solution adapts itself to the noise. Therefore, discarding these spurious features can effectively prevent over-learning—another typical symptom of ill-posedness. This viewpoint will be the key to understand the well-posedness of the KLMS algorithm. Similar idea can be found in [19].

Sometimes, we do not have direct access to the transformed data or the computation in the feature space is prohibitive. Then the RKHS theory comes to the rescue. The optimization problem (32) is equivalent to

\[
\min_{f \in H} R_{emp}(f) = \sum_{i=1}^{N} (y_i - f(u_i))^2 + \lambda \| f \|_{H}^2 \tag{39}
\]
The solution to (39) is given by the RBF networks [15]

$$f = \sum_{i=1}^{N} \alpha_i \kappa(u, \cdot)$$  \hspace{1cm} (40)

$$\tilde{\alpha} = (G_{\alpha} + \lambda I)^{-1} \tilde{y}$$  \hspace{1cm} (41)

where $\tilde{\alpha} = [\alpha_1, \alpha_2, ..., \alpha_N]^T$. Some relevant properties are listed below.

**Theorem 2.5:** Assume the optimal $f$ is found according to (40) and (41). The empirical error defined as the first term in (39) monotonically increases as $\lambda$ increases. The norm of $f$ defined as the second term in (39) monotonically decreases as $\lambda$ increases.

**Theorem 2.6:** For a fixed positive $\lambda$, the norm of the solution is bounded in the following way

$$\| \tilde{\alpha} \|^2_{\alpha} \leq \frac{\| \tilde{y} \|^2}{\lambda^2}$$  \hspace{1cm} (42)

$$\| f \|^2_{\alpha} \leq \frac{\| \tilde{y} \|^2}{4\lambda}$$  \hspace{1cm} (43)

**Theorem 2.7:** The optimization problem (39) is equivalent to the following problem

$$\min_{f} R_{emp}[f \in H, z^N] = \sum_{i=1}^{N} (y_i - f(u_i))^2$$  \hspace{1cm} (44)

subject to $\| f \|^2_{\alpha} \leq C$

in the sense that for any positive number $C$ there exists a unique $\lambda_C$ such that (39) and (44) lead to the same solution. Moreover, $\lambda_C$ monotonically decreases as $C$ increases.

The conclusions in Theorem 2.5, 2.6 and 2.7 also apply to linear LS. Adding the regularization term is equivalent to constraining the norm of the solution. The proofs of these theorems are included in the appendix.

To conclude, the key of regularization is to examine how the small singular components are taken care of and how the norm of the solution is constrained.
III. CONVERGENCE ANALYSIS OF THE KERNEL LMS

In this section, we will examine some important properties of the KLMS algorithm in light of the results obtained in section II-E. We will show that the KLMS algorithm is well-posed under usual conditions in the sense of Hadamard, using Tikhonov regularization theory. We assume the dimensionality of $F$ is $d$. Our discussion is based on the finite dimension assumption for simplicity. Since $F$ is a bounded Hilbert space, all the discussion extends to infinite dimension gracefully except for different terminology (operator theory) [20].

**Proposition 3.1** [8]: If $R_\phi$ is non-singular, the KLMS algorithm (14) converges asymptotically in the mean sense to the optimal solution (26) in Theorem 2.1 under the ‘small-step-size’ condition.

Remarks: Asymptotically, i.e. as $N$ approaches infinity, the LS solution converges to the optimal Wiener solution very quickly. Therefore, we use the LS estimator as the optimal solution in the asymptotic analysis.

**Proposition 3.2**: If $R_\phi$ is singular, the KLMS algorithm converges asymptotically in the mean sense to the optimal solution (30) in Theorem 2.2 under the ‘small-step-size’ condition.

Proof: From the data SVD (29) and (23), we know that $P$ is the eigen-vector matrix of $R_\phi$ and the corresponding eigenvalues are $\xi_i = \frac{s_i^2}{N}$.

Express the weight error in the following way

$$\Omega_n - \Omega^* = \sum_{i=1}^{d} \epsilon_i(n)P_i$$

where $P_i$ is the $i$-th column of $P$. $\epsilon_i(n)$ denotes the distance between $\Omega_n$ and $\Omega^*$ in the eigen-vector direction. It has been shown [8] that

$$E[\epsilon_i(n)] = (1 - \eta\xi_i)\epsilon_i(0)$$

$$E[|\epsilon_i(n)|^2] = \frac{\eta J_{\min}}{2 - \eta\xi_i} + (1 - \eta\xi_i)\epsilon_i(0)^2 - \frac{\eta J_{\min}}{2 - \eta\xi_i}$$

where $J_{\min} = E[y(n)]$ is the irreducible mean square error.
In the case of \( \xi_i = 0 \), (46) and (47) yield
\[
E[\xi_i(n)] = \xi_i(0) \tag{48}
\]
\[
E[|\xi_i(n)|^2] = |\xi_i(0)|^2 \tag{49}
\]
That means \( \xi_i(n) \) will stay in the neighborhood of the initial value forever. Therefore if the initialization is zero and the weight is decomposed as
\[
\hat{\Omega} = \sum_{i=1}^{d} \hat{\Omega}_i P_i \tag{50}
\]
the KLMS converges asymptotically (in the mean sense) to
\[
\hat{\Omega}_i = \begin{cases} 
\Omega_i^* & i = 1, \ldots, r \\
0 & i = r + 1, \ldots, N
\end{cases} \tag{51}
\]
where in the asymptotic case the optimal solution can be replaced by the LS solution.

On the other hand, substituting (50) into (27), we have the LS pseudo-inverse solution
\[
\hat{\Omega}_{i,\text{LSPL}} = \begin{cases} 
\Omega_i^{-1}(P^T p) & i = 1, \ldots, r \\
0 & i = r + 1, \ldots, N
\end{cases} \tag{52}
\]
which exactly states that the KLMS algorithm converges asymptotically (in the mean sense) to the LS pseudo-inverse solution in Theorem 2.2. This is expected because the LMS algorithm only seeks solution in the data subspace and is not affected by the null subspace. Though for small eigen-values, the speed is very slow, it does converge asymptotically.

Notice that the above analysis is based on the asymptotic assumption, i.e. \( N \) approaches to infinity. However, in practice the KLMS as formulated in (14) only has finite data and convergence would be never attained for small eigen-value components. For example, assuming the step size \( \eta = 0.1 \), the eigenvalue \( \xi_i = 0.001 \) and data point \( N = 500 \), we have \( (1 - \eta \xi_i)^N = 0.95 \), which means the corresponding component is attenuated by 0.05. More specifically, we have

**Proposition 3.3**: If the KLMS algorithm is trained with \( N \) data point, under the ‘small-step-size’ condition (in the mean sense) the solution is
\[
\hat{\Omega}_i = [1 - (1 - \eta \xi_i)^N] \xi_i^{-1}(P^T p) \tag{53}
\]
where \( \hat{\Omega}_i \) is defined in (50).
Proof: The initialization is zero so

\[
\varepsilon_i(0) = -\Omega^\circ
\]  

(54)

By (46),

\[
E[\hat{\Omega}_i(n)] = E[\Omega^\circ + \varepsilon_i(n)] = \Omega^\circ + E[\varepsilon_i(n)] = [1 - (1 - \eta \varepsilon_i^2)^\circ] \Omega^\circ
\]  

(55)

then using (52) as the approximate to the optimal solution, we complete the proof.

In other words, compared with pseudo-inverse solution (52), the KLMS algorithm modifies the singular values by the following reg-function (using the relation between the eigen-values and the singular values)

\[
H_{LMS}(x) = \sqrt{1 - (1 - \eta x^2 / N)^\circ}
\]  

(56)

The shape of the reg-function depends on the step size and number of data point. Indeed the KLMS (as well as the LMS) also filters out the small eigen-value components as prescribed by the Tikhonov regularization (Theorem 2.3 and 2.4) (see Fig.1). This understanding constitutes the core concept of this paper, although the stochastic nature of the LMS algorithm makes it dependent upon experimental parameters (the choice of step-size and the number of samples). In fact, this result indicates that the KLMS of (14) provides well-posed solutions even when the empirical risk cost (19) is ill-posed.
In this section, we established KLMS algorithm properties in parallel with the three LS theorems. The three propositions are based on the small-step-size theory and the conclusions are in the mean sense. The analysis reveals the convergence behavior of the KLMS and elucidates its self-regularization mechanism.

IV. STABILITY ANALYSIS OF THE KERNEL LMS

In this section, we will give a rigorous analysis of the norm of the solution obtained from the KLMS, thus complementing the insightful analysis of section III. We will show that the KLMS algorithm should be regarded as the stochastic gradient approximation of a constrained least square optimization

\[ \min_{\Omega} J(\Omega) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \Omega \Phi_i)^2 \]

subject to \( \| \Omega \|^2 \leq C \) (57)

instead of the conventional view of a stochastic approximation to the un-regularized LS solution. Moreover we will explicitly estimate \( C \).
Theorem 4.1: \((H^\infty \text{ stable [21]})\) Given training data \(\{\Phi_i, y_i\}_{i=1}^N\) that satisfy the linear regression model (22) for any unknown vector \(\Omega^o\) and finite energy noise sequence \(\{v(i)\}\) without any statistical assumption. Use the following LMS algorithm to estimate \(s(i) = \Omega^o(\Phi_i)\).

Start with \(\Omega_0 = 0\) and iterate for \(i = 1, 2, \ldots, N:\)

\[
\hat{s}(i | i-1) = \Omega_{i-1}(\Phi_i) \\
\Omega_i = \Omega_{i-1} + \eta [y_i - \Omega_{i-1}(\Phi_i)]
\]

The solution satisfies the robustness condition

\[
\sum_{j=1}^i |\hat{s}(j | j-1) - s(j)|^2 < \eta^{-1} ||\Omega^o||^2 + \sum_{j=1}^{i-1} |v(j)|^2 < 1 \quad \text{for all } i = 1, 2, \ldots, N
\]

if and only if the matrices \(\{\eta^{-1}I - \Phi_i\Phi_i^T\}\) are positive-definite for \(i = 1, 2, \ldots, N.\)

We use this theorem to prove the following theorems.

Theorem 4.2: Under the \(H^\infty\) stable condition, the apriori error defined in (14) satisfies the following inequality:

\[
||\hat{\eta}||^2 < \eta^{-1} ||\Omega^o||^2 + 2 ||\hat{\nu}||^2
\]

where \(\hat{\eta} = [e_1^o, e_2^o, \ldots, e_N^o]^T\) and \(\hat{\nu} = [v(1), v(1), \ldots, v(N)]^T.\)

Proof: First notice that

\[
e_i^o - v(i) = s(i) - \hat{s}(i | i-1)
\]

Substituting (61) into (59), we have

\[
\sum_{j=1}^i |e_j^o - v(j)|^2 < \eta^{-1} ||\Omega^o||^2 + \sum_{j=1}^{i-1} |v(j)|^2 < 1 \quad \text{for all } i = 1, 2, \ldots, N
\]

or equivalently,

\[
\sum_{j=1}^i |e_j^o - v(j)|^2 < \eta^{-1} ||\Omega^o||^2 + \sum_{j=1}^{i-1} |v(j)|^2 \quad \text{for all } i = 1, 2, \ldots, N
\]

By the triangle inequality

\[
\sum_{j=1}^i |e_j^o| \leq \sum_{j=1}^i |e_j^o - v(j)| + \sum_{j=1}^i |v(j)|^2 < \eta^{-1} ||\Omega^o||^2 + \sum_{j=1}^{i-1} |v(j)|^2 \quad \text{for all } i = 1, 2, \ldots, N
\]
In terms of norms,

\[ \| \tilde{e}^x \|^2 < \eta^{-1} \| \tilde{\Omega}^x \|^2 + 2 \| \tilde{v} \|^2 \]  \hspace{1cm} (65)

□

Theorem 4.3: Under the \( H^\infty \) stable condition, the a priori error further satisfies the following inequality:

\[ \| \tilde{e}^x \|^2 < 2 \| \tilde{y} \|^2 \]  \hspace{1cm} (66)

□

The proof is in the appendix.

Theorem 4.4: Under the \( H^\infty \) stable condition, the norm of the solution \( \tilde{\Omega}^x \) given by (17) is upper-bounded:

\[ \| \tilde{\Omega}^x \|^2 < 2\eta^2 \sigma_1 \| \tilde{y} \|^2 \]  \hspace{1cm} (67)

where \( \sigma_1 \) is the largest eigen-value of \( G_\Phi \).

Proof: By (17) and (66)

\[ \| \tilde{\Omega}^x \|^2 = \eta^2 \tilde{x}^\top G_\Phi \tilde{x} \leq \sigma_1 \eta^2 \| \tilde{e}^x \|^2 < 2\eta^2 \sigma_1 \| \tilde{y} \|^2 \]  \hspace{1cm} (68)

□

Comparing with (42), (43) in Theorem 2.6, (66) and (67) give very nice bounds. All the quantities depend on the input data very smoothly and never go to infinity provided finite energy of the training data.

In light of this fact, instead of viewing the kernel LMS as the stochastic gradient approximation of the unconstrained least square optimization

\[ \min_{\tilde{\Omega}} J(\Omega) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \Omega \Phi_i)^2 \]  \hspace{1cm} (69)

it is more precise to regard the kernel LMS as the stochastic gradient approximation of the following constrained least square optimization

\[ \min_{\tilde{\Omega}} J(\Omega) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \Omega \Phi_i)^2 \]

subject to \[ \| \tilde{\Omega} \|^2 \leq 2\eta^2 \sigma_1 \| \tilde{y} \|^2 \]  \hspace{1cm} (70)

which according to Theorem 2.7 is equivalent to Tikhonov regularization.
To conclude, the KLMS (or the LMS) is a self-regularized learning process thus preventing the possibility of over-fitting. Notice once again the dependence of the norm of the solution on the step size.

V. SIMULATIONS

The example is the short-term prediction of the Mackey-Glass chaotic time series with parameter $\tau = 30$ and the sampling period 6s. We compare the performance of the LMS, kernel LMS and RBF networks. The time embedding is 10 for all the systems and a segment of 500 samples is used as the training data and another 100 as the test data. All the data is corrupted by the Gaussian noise with zero mean and 0.04 variance. A Gaussian kernel with kernel width 1 is chosen for both RBF and kernel LMS. In RBF, every input point is used as the center of the RBF network. One hundred Monte Carlo simulations are run with different realizations of noise. The results are summarized in the following tables. Fig. 2 is the learning curves for the LMS and kernel LMS respectively and the learning rate is 0.2 for both algorithms. As expected, the KLMS converges to a smaller value of MSE because it is a nonlinear system in the input space and the system is nonlinear. Surprisingly, the rate of decay of both learning curves is basically the same, which implies that the eigenvalue spread of both systems is rather similar.

As we can see in table I, the performance of the KLMS is 5 times better than the linear LMS as can be expected (the MG is a nonlinear system) and is comparable to the RBF with suitable regularization. All the results in the tables are in the form of ‘average $\pm$ standard deviation’. It is interesting to compare the design and performance of KLMS with different stepsizes and the RBF with different regularizations. First, the KLMS performance in the training and test sets is statistically the same. Therefore, with KLMS the design ends with the proper performance in the training set. The regularized RBF is capable of a slightly better performance in the test set with the proper regularization parameter ($\lambda = 1$), but at the expense of a more complex solution (see Table II) and with a careful selection of the regularization parameter, which involves cross validation and it is non trivial. Table II summarizes the computational complexity of three algorithms. The KLMS effectively reduces the computational complexity and memory storage when compared with the
RBF. And if we have direct access to the feature space, the complexity can be reduced further to the dimensionality of the feature space if it is not prohibitively large. Table III simply supports our theory that the norm of the KLMS solution is naturally well-bounded (In this example, the norm of the desired signal is about 6). As we see, increasing the step size in the KLMS increases the norm of the solution but fails to increase the performance because of the gradient noise in the estimation (misadjustment).

![Figure 2: The learning curves of the LMS and kernel LMS (learning rate 0.2 for both).](image)

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Linear LMS</th>
<th>KLMS ($\eta=0.1$)</th>
<th>KLMS ($\eta=0.2$)</th>
<th>KLMS ($\eta=0.6$)</th>
<th>RBF ($\lambda=0$)</th>
<th>RBF ($\lambda=1$)</th>
<th>RBF ($\lambda=10$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training MSE</td>
<td>0.021±0.002</td>
<td>0.0074±0.0003</td>
<td>0.0054±0.0004</td>
<td>0.0062±0.0012</td>
<td>0±0</td>
<td>0.0038±0.0002</td>
<td>0.011±0.0001</td>
</tr>
<tr>
<td>Testing MSE</td>
<td>0.026±0.007</td>
<td>0.0069±0.0008</td>
<td>0.0056±0.0008</td>
<td>0.0058±0.0017</td>
<td>0.012±0.004</td>
<td>0.0039±0.0008</td>
<td>0.010±0.0003</td>
</tr>
</tbody>
</table>

**TABLE I**

**PERFORMANCE COMPARISON**
TABLE II
COMPLEXITY COMPARISON

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Linear LMS</th>
<th>KLMS</th>
<th>RBF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computation (training)</td>
<td>O(N)</td>
<td>O(N^2)</td>
<td>O(N^3)</td>
</tr>
<tr>
<td>Memory (training)</td>
<td>O(M)</td>
<td>O(N)</td>
<td>O(N^2)</td>
</tr>
<tr>
<td>Computation (test)</td>
<td>O(M)</td>
<td>O(N)</td>
<td>O(N)</td>
</tr>
<tr>
<td>Memory (test)</td>
<td>O(M)</td>
<td>O(N)</td>
<td>O(N)</td>
</tr>
</tbody>
</table>

TABLE III
SOLUTION NORM COMPARISON

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>KLMS(η=0.1)</th>
<th>KLMS(η=0.2)</th>
<th>KLMS(η=0.6)</th>
<th>RBF (λ=0)</th>
<th>RBF (λ=1)</th>
<th>RBF (λ=10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>solution norm</td>
<td>0.84±0.02</td>
<td>1.14±0.02</td>
<td>1.73±0.06</td>
<td>3375±639</td>
<td>1.47±0.03</td>
<td>0.55±0.01</td>
</tr>
</tbody>
</table>

VI. DISCUSSION AND CONCLUSION

This paper proposes the KLMS algorithm which is a straight application of the stochastic gradient methodology to solve the LS problem in RKHS. Since the update equation of the KLMS can be written as inner products, KLMS can be efficiently computed in the input space. The good approximation ability of the KLMS stems from the fact that the transformed data \( \phi(u_n) \) includes possibly infinite different features of the original data. In the framework of stochastic projection, the space spanned by \( \{\phi(u_i)\} \) is so large that the projection error of the desired signal \( y_n \) could be very small [22], as is well known from Cover’s theorem [15]. This capability includes modeling of nonlinear systems, which is the main reason why the kernel LMS can achieve good performance in the Mackey-Glass system prediction.

However, this projection to potentially infinite dimensional spaces brings two types of concerns: First, since one works in practice with finite data sets, is the solution well-posed in the feature space? Second, since gradient descent algorithms depend upon the eigenvalue spread of the data correlation matrix, and the misadjustment is also a function of the trace of this matrix, does the KLMS provide practical solutions?

This paper provides a conclusive answer to the first question that is rather surprising, in light of the results and methods in the kernel methods literature. In fact, the KLMS provides well-posed solutions, even when the original problem is ill-posed because the KLMS is actually solving a regularized learning problem due
to the stochastic gradient approximation. We show this in two ways for clarity. First, we use the idea of singular value filtering derived from Tikhonov regularization to show that the KLMS “filters” the small singular values of the data matrix dependent upon the value of the step size that the user selects (and also as a function of the data size). More formally we show that the norm of the KLMS solution is always bounded by the product of the power of the input and desired responses multiplied again by the step size (a tighter bound is shown in the text). Therefore, the KLMS is self-regularized by the step size which effectively works as the regularization parameter.

These results seem to imply that the KLMS provides a “free lunch”. Unfortunately this is not the case, but there are important implications for the design and implementation of adaptive solutions in high dimensional spaces. In terms of design, there is no need to regularize the LS cost function when using the KLMS algorithm, which means that the design stage ends with the verification that the performance in the training set is adequate. In terms of implementation, the KLMS is far simpler to implement, both in terms of computational complexity and memory storage, and it is intrinsically an on-line method. The reason there is no free lunch, is that the generalization performance is dependent upon the step size which also affects two fundamental characteristics of the KLMS algorithm: the speed of convergence and the misadjustment. The requirements for regularization, convergence speed and misadjustment do indeed conflict among themselves, and it is important to understand how regularization changes the well known speed/misadjustment compromise. When seen from the point of view of generalization, the goal is to find a small step-size (see (56), (67)) that will keep the small eigenvalues close to zero, but allow the large eigenvalues to converge. Of course this depends on the eigenvalue spread of the problem, but the user can still select the stopping criterion and the stepsize. Problems for which there is an abrupt jump of eigenvalues (and where the large ones contain the information to solve the problem) are the ones for which a reasonable stepsize can be found that guarantees good generalization. But intrinsically, the selection of an appropriate stepsize for generalization collides with the requirement for fast convergence, and corroborates solutions with small misadjustment. More specifically, the misadjustment is
\[ M_t = \frac{J(\infty) - J_{\text{min}}}{J_{\text{min}}} = \frac{\eta}{2} \text{tr}[R_p] = \frac{\eta}{2N} \text{tr}[G_p]. \]  
(71)

And for shift-invariant kernel i.e.,
\[ \langle \Phi(u_i), \Phi(u_j) \rangle = \Phi = \| \Phi(u_i) \|^2 = g_o \]  
(72)

the misadjustment of the KLMS simply equals \( \eta g_o / 2 \) which is data-independent but is proportional to the step size. Therefore, the convergence speed of the gradient estimate limits effectively the amount of regularization, and KLMS may be useful only in a subset of problems of interest. However, notice that now we are talking about a nonlinear solution to the problem at hand, and so the performance as demonstrated is almost identical to the regularized RBF network.

Regarding the second concern, further work should be conducted to elucidate the eigen-structure of the data in feature space, which as we know limits the speed of convergence of the algorithm. As demonstrated in Figure 2, at least for this problem the eigen-structure must be similar because the two curves converge at the same rate, but it may be a special case for this data set. For a complete discussion of convergence, a rigorous eigen-spread analysis after the nonlinear transformation is essential. However as showed in most of the kernel principal component analysis applications, the transformed data in the feature space are usually well clustered in a small subspace (say 95% power) [3], [23] such that fast convergence is attained for the kernel LMS algorithm, which is also corroborated in our experiment. Another interesting problem to be further investigated is the application of KLMS to nonstationary environments.

APPENDIX

Proof of theorem 2.5: By using (41) and denoting \( \tilde{y}_q = Q\tilde{y} \), we obtain
\[
\| \tilde{c}_q \|^2 = \tilde{y}^T[I - G(G + \lambda I)^{-1}]^T \tilde{y} \\
= \tilde{y}_q^T[\text{diag}(\frac{\sigma_1}{\sigma_1 + \lambda}, ..., \frac{\sigma_N}{\sigma_N + \lambda})] \tilde{y}_q \\
= \sum_{i=1}^{N} \tilde{y}_q^T \frac{\sigma_i}{\sigma_i + \lambda}^2
\]  
(73)
Observing the quadratic terms in (73), if \( \sigma_i = 0 \), the corresponding term will not be affected by \( \lambda \); otherwise it monotonically increases as \( \lambda \) increases. Overall, the empirical error monotonically increases as \( \lambda \) increases. Similarly, the norm of \( f \) monotonically decreases as \( \lambda \) increases. This observation also coincides with the optimization setting (39) where \( \lambda \) serves as a trade-off parameter between two terms. ■

Proof of theorem 2.6: By , for a fixed positive \( \lambda \),

\[
\| \tilde{\alpha} \|_2^2 = \tilde{y}^T (G + \lambda I)^{-1} \tilde{y} \\
= \tilde{y}_0^T \left[ \text{diag} \left( \frac{\sigma_1}{(\sigma_1 + \lambda)^2}, \ldots, \frac{\sigma_N}{(\sigma_N + \lambda)^2} \right) \right] \tilde{y}_0 \\
= \sum_{i=1}^N \tilde{y}_0^2 \frac{\sigma_i}{(\sigma_i + \lambda)^2} \\
\leq \frac{\| \tilde{y} \|_2^2}{\lambda^2} = \frac{\| \tilde{y} \|_2^2}{\lambda^2}
\]  

by the fact that \( Q \) is orthogonal.

By (74), for a fixed positive \( \lambda \), the function \( g : R^+ \cup \{0\} \rightarrow R \) given by

\[
g(r) = \frac{r}{(r + \lambda)^2}
\]

has a maximum at \( r = \lambda \), which can be easily verified by taking the derivatives. Therefore

\[
\| f \|_2^2 = \tilde{y}^T \left[ \text{diag} \left( \frac{\sigma_1}{(\sigma_1 + \lambda)^2}, \ldots, \frac{\sigma_N}{(\sigma_N + \lambda)^2} \right) \right] \tilde{y} \\
\leq \frac{\| \tilde{y} \|_2^2}{4\lambda} = \frac{\| \tilde{y} \|_2^2}{4\lambda}
\]

Proof of theorem 2.7: This constrained optimization problem (44) can be solved by the method of Lagrange multipliers.

\[
\min_{f \in \mathbb{R}^N} R_{\text{reg}}[f, z^N] = \sum_{i=1}^N (y_i - f(u_i))^2 + \lambda (\| f \|_2^2 - C)
\]
where the auxiliary nonnegative variable $\lambda$ is called the Lagrange multiplier. For a fixed $\lambda$, (78) is equivalent to (39) except for a constant. Therefore, the solution is given by (40) and (41) with $\lambda$ as a parameter. According to the first part of Theorem 2.6, it is now equivalent to find the smallest $\lambda$ (such that the empirical error is minimized) which satisfies the constraint. Denote

$$C_0 := \| f_{\lambda=0} \|_H^2 = \sum_{i=1}^{N} \frac{y_i^2}{\sigma_i} \leq \infty$$

(79)

Notice $C_0$ can be infinity. Moreover, by (74) $\| f_{\lambda=\infty} \|_H^2$ goes to 0. By Theorem 2.6, $\| f_{\lambda} \|_H^2$ monotonically depends on $\lambda$, so for any $C$ which lies between 0 and $C_0$, there is a unique $\lambda_c$ satisfies

$$\| f_{\lambda=\lambda_c} \|_H^2 = C$$

(80)

If $C_0 \leq C < \infty$, the constraint is ineffective and $\lambda = 0$, which is also clear from the Kuhn-Tucker conditions.

To conclude

$$\lambda_c = \begin{cases} 0 & \text{if } C > C_0 \\ \arg\{\| f_{\lambda} \|_H^2 = C \} & \text{if } C \leq C_0 \end{cases}$$

(81)

It is clear that $\lambda_c$ monotonically depends upon $C$.

Proof of theorem 4.3: Notice (65) is valid for any unknown vector $\Omega^e$ and finite energy noise sequence $\{v(i)\}$ and a trade-off is there to achieve the tightest bound. A trivial $\Omega^e$ can be chosen with large noise term in (22). On the other hand, a ‘precise’ $\Omega^e$ can be chosen such that $\| \bar{v} \| = 0$, while the price is $\| \Omega^e \|$ may be very large. For this problem, $\Omega^e = 0$ can be chosen and $v(i) = y_i$ follows. Substituting them into (65) yields

$$\| \tilde{a}^e \|^2 < 2 \| \bar{y} \|^2$$

(82)

We can utilize the results from the regularization networks to show that this is actually a very tight bound. Using the results from the RBF networks, we have

$$\tilde{a} = (G_a + \lambda I)^{-1} \bar{y}$$

(83)

$$\Omega^e = \sum_{i=1}^{N} \alpha_i \Phi_i$$

(84)

$$\bar{v} = \bar{y} - G_a \tilde{a}$$

(85)
where $\bar{\alpha} = [\alpha_1, \ldots, \alpha_N]^T$. Therefore

$$\| \Omega^\alpha \|^2 = \bar{\alpha}^T G_\alpha \bar{\alpha} = \bar{y}^T (G_\alpha + \lambda I)^{-1} G_\alpha (G_\alpha + \lambda I)^{-1} \bar{y}$$

(86)

$$\| \bar{y} \|^2 = \| y \|^2 - 2 \bar{y}^T G_\alpha \bar{\alpha} + \bar{\alpha}^T G_\alpha G_\alpha \bar{\alpha}$$

(87)

Substituting (86) and (87) into (65) yields

$$\| \bar{e}^\alpha \|^2 < \eta^{-1} \bar{\alpha}^T G_\alpha \bar{\alpha} + 2 \| \bar{y} \|^2 - 4 \bar{y}^T G_\alpha \bar{\alpha} + 2 \bar{\alpha}^T G_\alpha G_\alpha \bar{\alpha}$$

$$= \bar{\alpha}^T (2G_\alpha + \eta^{-1}G_\alpha) \bar{\alpha} + 2 \| \bar{y} \|^2 - 4 \bar{y}^T G_\alpha \bar{\alpha}$$

(88)

$$= \bar{y}^T (G_\alpha + \lambda I)^{-1} (2G_\alpha + \eta^{-1})G_\alpha (G_\alpha + \lambda I)^{-1} \bar{y} + 2 \| \bar{y} \|^2 - 4 \bar{y}^T G_\alpha (G_\alpha + \lambda I)^{-1} \bar{y}$$

$$= \bar{y}^T [(G_\alpha + \lambda I)^{-1} (2G_\alpha + \eta^{-1})G_\alpha (G_\alpha + \lambda I)^{-1} + 2 - 4G_\alpha (G_\alpha + \lambda I)^{-1}] \bar{y}$$

Define

$$M(\lambda) = [(G_\alpha + \lambda I)^{-1} (2G_\alpha + \eta^{-1})G_\alpha (G_\alpha + \lambda I)^{-1} + 2 - 4G_\alpha (G_\alpha + \lambda I)^{-1}]$$

(89)

Suppose $G_\alpha = Q \sum Q^\alpha$ and $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_N)$, where $\sigma_1 \geq \ldots \geq \sigma_N \geq 0$

$$M(\lambda) = Q[(\Sigma + \lambda I)^{-1} (2 \Sigma + \eta^{-1}) \Sigma (\Sigma + \lambda I)^{-1} + 2 - 4 \Sigma (\Sigma + \lambda I)^{-1}] Q^T$$

$$= Q\text{diag}\left[\frac{(2\sigma_1 + \eta^{-1})\sigma_1}{\sigma_1 + \lambda}, \ldots, \frac{(2\sigma_N + \eta^{-1})\sigma_N}{\sigma_N + \lambda}, 2 - \frac{4\sigma_N}{\sigma_N + \lambda}\right] Q^T$$

(90)

$$= Q\left[\frac{\sigma_1 \eta^{-1} + 2 \lambda^2}{(\sigma_1 + \lambda)^2}, \ldots, \frac{\sigma_N \eta^{-1} + 2 \lambda^2}{(\sigma_N + \lambda)^2}\right] Q^T$$

Since (65) is valid for any $\lambda$, we try to find the tightest upper-bound.

Let us analyze the following function

$$h_\lambda(\lambda) = \frac{\sigma_\lambda \eta^{-1} + 2 \lambda^2}{(\sigma_\lambda + \lambda)^2}$$

(91)

Notice that $\lambda = \infty$ corresponds to the trivial solution $\Omega^\alpha = 0$. It is easy to check that the minimum is at $\lambda = 1/2 \eta$. Therefore specifically,

$$\| \bar{e}^\alpha \|^2 < \bar{y}^T M(\lambda) \bar{y}$$

$$= (Q^T \bar{y})^T \text{diag}\left[\frac{1}{\sigma_\eta + 1/2}, \ldots, \frac{1}{\sigma_N \eta + 1/2}\right] (Q^T \bar{y})$$

(92)

$$< 2 \| Q^T \bar{y} \|^2 / (1 + 2 \sigma_N \eta)$$

$$< 2 \| Q^T \bar{y} \|^2 = 2 \| \bar{y} \|^2$$
ACKNOWLEDGMENT

This work was partially supported by NSF grant ECS-0300340 and ECS-0601271.

REFERENCES