

TRACKING OF MULTIVARIATE TIME-VARIANT SYSTEMS BASED ON ON-LINE VARIABLE SELECTION

Sung-Phil Kim, Yadunandana N. Rao, Deniz Erdogmus, Jose C. Principe
Computational NeuroEngineering Laboratory,
University of Florida, Gainesville, FL 32611
{phil, yadu, deniz, principe}@cnel.ufl.edu

Abstract. Tracking time-variant systems has been of great interest in many engineering fields. Specifically, when system statistics change both in space (multivariate) and time with a short stationary regime, conventional adaptive algorithms suffer from the trade-off between convergence rate and accuracy. In this paper, we propose a tracking system consisting of a linear adaptive system accompanied by an on-line variable selection algorithm that is based on the least angle regression algorithm. This algorithm explicitly employs local (in time) correlation between the input and the output of an unknown system to select a subset of input variables at every time step. Therefore, it enables the multivariate adaptive filter to track the temporal changes of correlated variables. Simulations involving tracking of multi-channel time-variant systems demonstrate superior performance of the proposed approach when compared with the conventional methods.

INTRODUCTION

Identification of the time-variant systems has been one of the most important problems in many areas, including automatic control, communication systems, biological systems and signal processing. Adaptive filtering algorithms have been widely used to model any unknown system by adjusting a set of parameters with the given input-output data [1, 2]. In a nonstationary environment the adaptive filter must track the time-varying statistics of the unknown system. The least mean squares (LMS) and the recursive least squares (RLS) algorithms have been the most popular methods for training linear adaptive systems, and their tracking abilities in nonstationary environment has been well studied for the single input single output (SISO) case [2,3,4,5]. However, the performance of these methods may be limited when both spatial (across inputs) and temporal statistics of the input change with time. These changes in space and time often cause different subsets of the inputs to be involved in the mapping properties of the unknown system. In such cases, variable (or feature) selection techniques that select a subset of input variables by certain criteria may play an important role in enhancing the identification performance.

Numerous methods for variable selection have been developed including subspace

projection, neural networks, support vector machine, and clustering [6]. Most of them, however, have been designed to work in a stationary environment. Therefore, the application of the variable selection methods to identification of time-variant systems requires a novel approach that is able to track the spatio-temporal change of the underlying system.

The purpose of this paper is two-fold: Firstly, we develop an on-line variable selection algorithm for system identification by simplifying the least angle regression algorithm proposed by Efron *et al.* [7]. This algorithm is capable of selecting most correlated input variables that are unknown *a priori* which also change over time. Secondly, we demonstrate enhanced tracking performance by combining the on-line variable selection algorithm with a linear filtering algorithm (LMS). Simulations demonstrate the superior tracking capability of the proposed approach compared with the case when only the LMS algorithm is used.

THE LEAST ANGLE REGRESSION ALGORITHM

The least angle regression (abbreviated LARS) algorithm has been recently developed to accelerate computation and improve performance of forward model selection methods [7]. It has been shown in Efron *et al.* that simple modifications to LARS can implement the LASSO (Least Absolute Shrinkage and Selection Operator) [8] and the forward stagewise linear regression [9]. The LARS algorithm requires the same order of magnitude of computational complexity as ordinary least squares.

The selection property of LARS, which leads to zeroing coefficients, is preferable for identification of sparse systems when compared to methods with regularization based on L_2 -norm penalty [10] and popularized in the weight decay techniques of neural networks [11]. Also, an analysis of the selection process often provides better insights into the unknown system than shrinkage methods based on the L_2 -norm.

The LARS procedure starts with an all zero coefficients initial condition. The input variable having the most correlation with the desired response is selected. We proceed in the direction of the selected input with a step size which is determined such that some other input variable gets to have as much correlation with the current residual as the first input. Then, we move in the equiangular direction between these two inputs until the third input has the same correlation. This procedure is repeated until either all input variables join the selection, or the sum of coefficients meets a preset threshold (constraint). Note that the maximum correlation between inputs and the residual over each selection step decreases in order to de-correlate the residual with the inputs. We summarize the details of the LARS procedure [7] in table 1.

These techniques have been applied to variable selection in multivariate data analysis (static problems), but due to its computational efficiency, LARS can effectively provide a tool for time varying model selection with the L_1 -norm penalty similar to the LASSO. However, since LARS selects input variables by computation of correlation using the entire data set, it assumes stationary statistics.

In this paper, we will present a modified version of LARS which selects the most correlated input variables locally in time, thereby removing the stationarity limitation, which is conducive for real-time system identification problems.

<p>Given an $N \times M$ input matrix \mathbf{X} (each row being M-dimensional sample vector), and an $N \times 1$ desired response matrix \mathbf{Y}, initialize the model coefficient $\beta_i = 0$, for $i=1, \dots, M$, and let $\boldsymbol{\beta} = [\beta_1, \dots, \beta_M]^T$</p> <p>Then the initial LARS estimate becomes, $\hat{\mathbf{Y}} = \mathbf{X}\boldsymbol{\beta} = \mathbf{0}$.</p> <p>Transform \mathbf{X} and \mathbf{Y} such that $\frac{1}{N} \sum_{i=1}^N x_{ij} = 0$, $\frac{1}{N} \sum_{i=1}^N x_{ij}^2 = 1$, and $\frac{1}{N} \sum_{i=1}^N y_i = 0$ for $j = 1, \dots, M$.</p>	
<p>The current correlation $c = \mathbf{X}^T(\mathbf{Y} - \hat{\mathbf{Y}})$</p>	(1)
<p>Let $C_{max} = \max_j \{c_j\}$, and $A = \{j: c_j = C_{max}\}$.</p>	(2)
<p>Let $\mathbf{X}_a = \{\dots, \text{sign}(c_j)\mathbf{x}_j, \dots\}$ for $j \in A$.</p>	(3)
<p>Let $\boldsymbol{\Phi} = \mathbf{X}_a^T \mathbf{X}_a$, and $\alpha = (\mathbf{1}_a^T \boldsymbol{\Phi}^{-1} \mathbf{1}_a)^{-1}$</p>	(4)
<p>where $\mathbf{1}_a$ is a vector of one's with a length equal to size of A.</p> <p>The equiangular vector $\boldsymbol{\mu} = \mathbf{X}_a(\alpha \boldsymbol{\Phi}^{-1} \mathbf{1}_a)$</p>	(5)
<p>has the unit length and the property such that $\mathbf{X}_a \boldsymbol{\mu} = \alpha \mathbf{1}_a$ (angles between all inputs in A and $\boldsymbol{\mu}$ are equal).</p>	
<p>Compute the step size, $\gamma = \min_{j \in A}^+ \left\{ \frac{C_{max} - c_j}{\alpha - \theta_j}, \frac{C_{max} + c_j}{\alpha + \theta_j} \right\}$</p>	(6)
<p>where \min^+ indicates considering only positive minimum values over possible j.</p> <p>Also θ_j is defined as the inner product between all inputs and $\boldsymbol{\mu}$ such as,</p>	
<p>$\theta_j = \mathbf{X}_j^T \boldsymbol{\mu}$</p>	(7)
<p>Update $\hat{\mathbf{Y}}_+ = \hat{\mathbf{Y}} + \gamma \boldsymbol{\mu}$.</p>	(8)
<p>Repeat (1)–(8) until all inputs join the active set A, or $\sum_j \beta_j$ exceeds the given threshold.</p>	

Table 1. Outline of the LARS algorithm.

LOCAL IN TIME VARIABLE SELECTION BASED ON LARS

Correlation between inputs and the desired response can be accomplished by recursively updating the correlation vector. The input covariance matrix can also be estimated recursively. If one decouples the variable selection part from the model update part in LARS, we can select the input variables locally with recursive estimates of correlations. The modified version of LARS for on-line variable selection is as follows.

As stated in [7], the time correlation (step 1 in table 1) at a given step can be

simply updated without computing residuals,

$$c_j(k) = c_j(k) - \gamma\theta_j \quad (9)$$

for the k^{th} step of variable selection. Hence, the update procedure of (8) in table 1 can be removed. The initial correlation $c_j(0)$, which represents the correlation between inputs and desired response can be estimated outside of the LARS routine. Instead of computing the correlation with entire data, we can recursively estimate the correlation using a forgetting factor (leaky integrator), given by

$$\mathbf{p}(n) = (1-\rho)\mathbf{p}(n-1) + \rho(d(n)\mathbf{x}(n)) \quad (10)$$

where ρ is the parameter controlling memory depth, and $\mathbf{x}(n)$ is an $1 \times M$ input vector at time instance n . This estimate of the correlation vector at a given time instance, $\mathbf{p}(n)$ is utilized by the LARS routine such that $c_j(0) = p_j(n)$.

For the computation of the covariance matrix Φ in (4), we also estimate the input covariance matrix using the leaky integrator in the same way as (10),

$$\mathbf{R}(n) = (1-\delta)\mathbf{R}(n-1) + \delta\mathbf{x}(n)^T\mathbf{x}(n) \quad (11)$$

where δ is another step size for covariance estimation. This matrix is not directly used in (4) since Φ is the covariance of only subset of inputs. Also, the input vectors are multiplied by the sign of correlations before computing Φ . Therefore we need to introduce a diagonal matrix \mathbf{S} whose elements are signs of $c_j(k)$ for $j \in A$. Then Φ can be computed using $\mathbf{R}(n)$ and \mathbf{S} as,

$$\Phi = \mathbf{S}\mathbf{R}_a\mathbf{S} \quad (12)$$

where \mathbf{R}_a is $L_a \times L_a$ (L_a is the length of A) matrix representing covariance among the selected input variables. \mathbf{R}_a can be given by elements of $\mathbf{R}(n)$, i.e., r_{ij} for $i, j \in A$.

To remove the computation of the equiangular vector μ that requires a batch operation, we incorporate equation (5) into (7) such that,

$$\theta_j = \mathbf{X}^T\mu = \mathbf{X}^T\mathbf{X}_a(\alpha\Phi^{-1}\mathbf{1}_a) = \alpha\mathbf{X}^T\mathbf{X}_a\Phi^{-1}\mathbf{1}_a \quad (13)$$

However, $\mathbf{X}^T\mathbf{X}_a$ is nothing but the j^{th} columns of $\mathbf{R}(n)$, for $j \in A$ followed by multiplication with \mathbf{S} . So if we let \mathbf{R}_{acol} be a submatrix of $\mathbf{R}(n)$ consisting of the j^{th} columns for $j \in A$, then

$$\theta_j = \alpha\mathbf{R}_{acol}\mathbf{S}\Phi^{-1}\mathbf{1}_a \quad (14)$$

Hence, using Φ obtained by $\mathbf{R}(n)$ and \mathbf{S} , we can compute α and consecutively θ_j for $j \in A$. This modification removes the computation of the equiangular vector in (5) which is not directly required for computing θ_j and γ in (6). Table 2 summarizes this modified version of the LARS algorithm.

LINEAR ADAPTIVE SYSTEM WITH ON-LINE VARIABLE SELECTION

In our architecture, we combine a linear adaptive system with the on-line variable selection algorithm described in the previous section. Accordingly, the on-line

<p>Given an $N \times M$ input matrix \mathbf{X} (each row being M-dimensional sample vector), and an $N \times 1$ desired response matrix \mathbf{Y}, initialize $\mathbf{p}(0) = \mathbf{0}$ and $\mathbf{R}(0) = \mathbf{0}$</p> <p>Transform \mathbf{X} and \mathbf{Y} such that</p> $\frac{1}{N} \sum_{i=1}^N x_{ij} = 0, \quad \frac{1}{N} \sum_{i=1}^N x_{ij}^2 = 1, \quad \text{and} \quad \frac{1}{N} \sum_{i=1}^N y_i = 0 \quad \text{for } j = 1, \dots, M.$ <p>Update the correlation: $\mathbf{p}(n) = (1-\rho)\mathbf{p}(n-1) + \rho(d(n)\mathbf{x}(n))$</p> <p>Update the input covariance: $\mathbf{R}(n) = (1-\delta)\mathbf{R}(n-1) + \delta\mathbf{x}(n)\mathbf{x}(n)^T$</p>
<p>Let $\mathbf{c}(0) = \mathbf{p}(n)$.</p> <p>For $k = 0, \dots, M-1$</p> <p>Let $C_{max} = \max_j \{c_j\}$, and $A = \{j: c_j(k) = C_{max}\}$.</p> <p>Compute a diagonal matrix \mathbf{S} with elements of sign of $c_j(k)$ for $j \in A$.</p> <p>$\Phi = \mathbf{S}\mathbf{R}_a\mathbf{S}$,</p> <p>where \mathbf{R}_a is submatrix of $\mathbf{R}(n)$ with j^{th} rows and j^{th} columns for $j \in A$.</p> <p>$\alpha = (\mathbf{1}_a^T \Phi^{-1} \mathbf{1}_a)^{-1}$</p> <p>$\theta_j = \alpha \mathbf{R}_{acol} \mathbf{S} \Phi^{-1} \mathbf{1}_a$</p> <p>where \mathbf{R}_{acol} is a matrix consisting of j^{th} columns of $\mathbf{R}(n)$ for $j \in A$.</p> <p>Compute the step size, $\gamma = \min_{j \in A}^+ \left\{ \frac{C_{max} - c_j}{\alpha - \theta_j}, \frac{C_{max} + c_j}{\alpha + \theta_j} \right\}$.</p> <p>Update correlation: $c_j(k) = c_j(k) - \gamma\theta_j$.</p>

Table 2. The modified LARS algorithm for on-line variable selection.

selection algorithm selects input variables *before* updating the weights at any given time sample. Then, only the weights corresponding to the selected inputs are updated by the least mean squares algorithm (LMS) [5]. This instantaneous preprocessing of inputs enables the adaptive system to selectively update weights

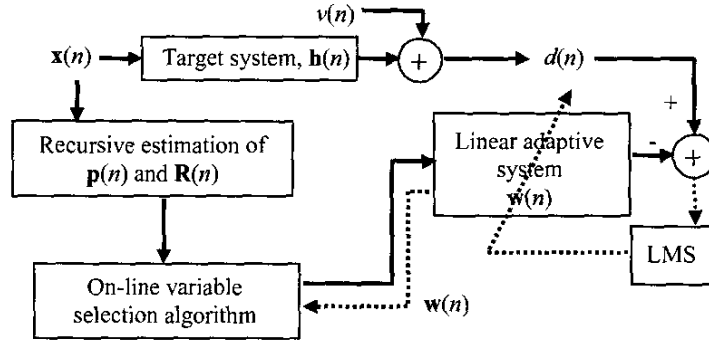


Figure 1. A block diagram of the linear adaptive system with on-line variable selection. $\mathbf{h}(n)$ and $\mathbf{w}(n)$ denote the target and adaptive system weight vectors, respectively, and $v(n)$ is the measurement noise added to the desired response.

based on a new least angle regression criterion. Fig. 1 illustrates the architecture of the linear adaptive system with the on-line variable selection algorithm. Note that the current weight vector is fed to the selection algorithm. This weight vector is used by the algorithm to stop the iteration when the absolute sum of weights reaches a given threshold.

The benefits of on-line variable selection will become apparent in the multivariate input case where only a subset of inputs is primarily involved to generate the system outputs. Two major benefits of using this online variable selection are: it can reduce the degrees of freedom of the model to enhance the generalization performance, and it can allow faster learning of the subsequent model, which improves tracking performance. Hence we will apply the linear system with the on-line variable selection algorithm for tracking a time-variant system in which only a subset of inputs are correlated with outputs. The comparison with the standard LMS will demonstrate the improvement in tracking performance.

COMPUTER EXPERIMENTS

In our experiment, we assume a static (i.e. constant value) multi-channel system that changes its parameters at certain time instants. For each period between changes, only a subset of parameters has non-zero values. We present results of two adaptive systems trained by the LMS with variable selection and LMS only. The weights of the system are updated with the straight LMS for all the weights, while with variable selection only weights corresponding to the selected channels are updated as

$$w_j(n+1) = w_j(n) + \eta e(n)x_j(n), \text{ for } j \in A \quad (15)$$

where A is a set of input channel indices which are selected. The first simulation tracks a time-variant system with M channels and sparse constant weights that change every 50 samples. The change occurs three times during the simulation so that there are four stationary regions with duration of 50 samples (total number of samples is $N = 200$). During each stationary regime, only K of M weights, which are generated from a Gaussian distribution, have nonzero values. The setup of K is randomly chosen from a uniform distribution ranging between $M/2-2$ to $M/2+2$. The tracking system does not have a priori knowledge about which weights have nonzero values for each regime. Further, the sum of magnitudes of weights for each period is set to unity. Input variables are also generated independently from a Gaussian distribution with zero mean and unit variance. The desired output of the system is further corrupted by white noise whose power is determined by the signal-to-noise ratio (SNR). Therefore, the system parameters include the number of input channels (M) and the SNR of the desired response.

The second experiment tracks a nonlinear time-variant system. The setup is similar but the desired signal is generated as

$$d(n) = \tanh(\mathbf{h}(n)\mathbf{x}(n)^T) - 0.5 \cos(\mathbf{h}(n)\mathbf{x}(n)^T) \quad (16)$$

where $\mathbf{h}(n)$ is an M -dimensional constant weight vector generated in the same way

as above. Commonly used measures of tracking assessment such as the steady-state mean squared deviation and the steady-state excess mean squared error are utilized here [12]. The mean squared deviation between the true weight vector $\mathbf{h}(n)$ and the learned weight vector $\mathbf{w}(n)$ is defined as

$$D = \lim_{n \rightarrow \infty} E \left[\|\mathbf{h}(n) - \mathbf{w}(n)\|^2 \right] \quad (17)$$

The excess mean squared error is defined as

$$\xi = \lim_{n \rightarrow \infty} E[e(n)^2] - \sigma^2 \quad (18)$$

where σ^2 is the variance of white noise added to the desired response. Since there are four stationary regimes that have different weight vectors $\mathbf{h}(n)$, D and ξ are evaluated in each regime. To evaluate the mean performance, we conducted 100 Monte Carlo runs for each experiment.

In order to determine the learning rate (η) for the LMS (with or without variable selection), we perform an exhaustive search and choose the learning rate that yields the best tracking performance for the given data. Since there is a tradeoff between the mean squared deviation and the excess mean squared error [5], the learning rate is chosen by imposing equal importance to both performance measures, D and ξ . We experimentally evaluate D and ξ (averaged over four stationary regimes) by scanning through a possible range of η , and pick the one for which $(D + \xi)/2$ is minimum. This searching procedure is conducted for both systems with and without variable selection, and also for every case of M and SNR. Fig. 2 illustrates this procedure by showing plots of performance measures evaluated for different learning rates, with the straight LMS and the LMS with variable selection respectively. Other parameters for variable selection are set as $\{\rho, \delta\} = \{0.1, 0.1\}$.

The evaluation results of D and ξ based on the above setting of parameters are summarized in table 3. In the table, the ratio of D and ξ for the LMS with on-line variable selection versus D and ξ for the straight LMS is presented for different

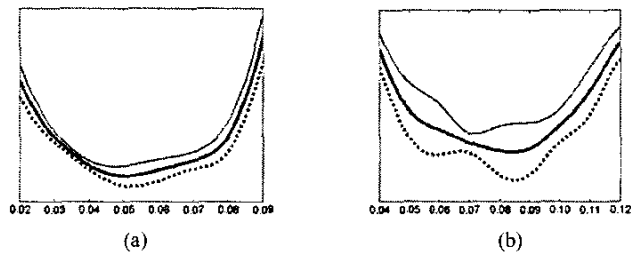


Figure 2. Illustration of the selection of the learning rate: the experimental results of D (thin solid line), ξ (dotted line), and $(D+\xi)/2$ (thick solid line) for a range of learning rates for (a) the straight LMS, and (b) the LMS with variable selection. In this example, we set $M=6$, and $\text{SNR}=30\text{dB}$. The selected learning rates are 0.052 in (a), and 0.083 in (b).

M	8		16		24	
	30dB	0dB	30dB	0dB	30dB	0dB
SNR						
$\frac{D^{selection}}{D^{LMS}}$	0.43	0.78	0.26	0.77	0.50	0.76
	0.40	0.81	0.18	0.72	0.37	0.65
	0.59	0.75	0.16	0.71	0.33	0.60
	0.49	0.78	0.15	0.75	0.30	0.64
$\frac{\xi^{selection}}{\xi^{LMS}}$	0.41	0.77	0.26	0.75	0.49	0.75
	0.37	0.82	0.18	0.71	0.36	0.66
	0.60	0.73	0.15	0.72	0.31	0.61
	0.49	0.78	0.15	0.74	0.29	0.65

Table 3. Mean ratios of D and ξ for different M and SNR in each stationary regime.

values of M and SNR.

Fig. 3 shows the comparison of learning dynamics between the LMS with on-line variable selection and the straight LMS. It presents the average ratio of $E[\|\mathbf{h}(n) - \mathbf{w}(n)\|^2]$ for the LMS with on-line variable selection to that of the straight LMS in different cases of M and SNR in linear system tracking. The results show that the trajectory of the ratio is less than unity most of the time, which indicates the superior performance of the LMS with on-line variable selection. Note that variable selection becomes less effective with decreasing SNR as the correlation estimates are noisy.

More details on variable selection are described in Fig. 4. The selection error rate, which is the ratio of the false selection to the total number of variables, is averaged over 100 Monte Carlo simulations. This rate decreases with time in a given stationary region since the algorithm is utilizing more samples to estimate correlation. Fig. 4a shows the effect of the feedback parameter for the leaky integrator (ρ in eq.10) used to estimate correlation. Smaller values of ρ indicate larger memory depth. It empirically shows that the parameter ρ must be set in the middle of the convergence range $[0,1]$ while the selection error is minimized around $\rho = 0.1$. The choice of ρ , however, may vary depending upon the data. Fig. 4b shows the effect of the SNR by comparing error rates with different levels of the SNR. The error rate increases approximately by 20% as the SNR drops by 30dB near convergence in each stationary segment.

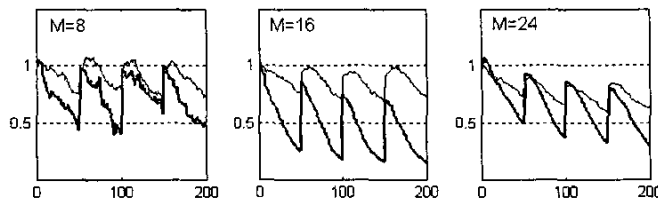


Figure 3. Evolution of the ratio of $D^{selection}/D^{LMS}$ during linear system tracking over 200 samples (four 50-sample stationary segments). The thick line denotes the ratio for SNR = 30dB and the thin line denotes for SNR = 0dB.

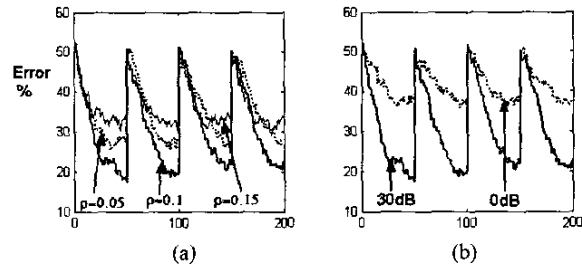


Figure 4. Evolution of the selection error percentage averaged over 100 Monte Carlo simulations ($M = 16$) during tracking: (a) Effect of different feedback parameters in the leaky integrator, and (b) Effect of the SNR on the selection error percentage.

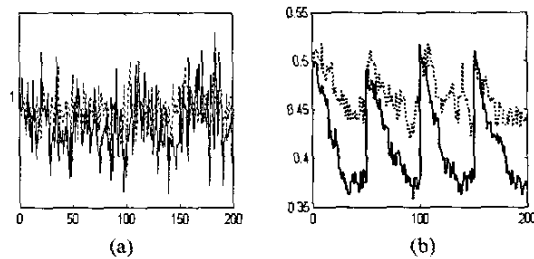


Figure 5. (a) Evolution of the ratio $\frac{p_{\text{selection}}}{p_{\text{LMS}}}$ for tracking a nonlinear system. (solid line: SNR = 30dB, dotted line: SNR = 0dB). (b) Comparison of the selection error percentage between linear and nonlinear system tracking ($M = 16$ and SNR = 0dB). The solid line presents the ratio for the linear system, and the dotted line represents the nonlinear system.

Fig. 5a shows trajectories of the excess mean squared error ratio computed for the nonlinear system ($M = 16$). The SNR is kept constant at 30dB. It is obvious that the improvement with variable selection for a nonlinear system is much lesser than the linear case, which reflects the fact that the selection algorithm is based on the linear correlation measures. Hence the algorithm has more chances of incorrect selection for a nonlinear system than for a linear one. This is presented in Fig. 5b in which the selection error rates for linear and nonlinear systems are compared.

CONCLUSIONS

We have proposed an on-line variable selection algorithm that simplifies the LARS forward model selection algorithm. With the experimental design involving a system with time-variant sparse input-output mapping, the proposed architecture combining the linear system with the on-line variable selection algorithm has shown superior tracking capability. We hope that the demonstration in this paper may provide many possibilities of enhancement for a variety of practical fields

where time-varying selection of variables are of great importance to infer the target system. In particular, this algorithm may be useful for the identification of time varying multi-input multi-output (MIMO) systems, which is basically an unexplored area. Further studies of the application of on-line variable selection to MIMO systems will be presented in a follow-up paper.

The marginal improvement in identification of the nonlinear time-variant system has motivated us to consider more general criteria to select variables. One approach would be to utilize distance metrics such as mutual information [13] between the inputs and the residuals. With recently developed methods that stochastically estimate Renyi's entropy [13], we hope to perform time-variant variable selection based on information theoretic criteria.

ACKNOWLEDGEMENTS: This work was supported by DARPA sponsored grant # ONR-450595112.

REFERENCES

- [1] K. Nishiyama, "An H_∞ Optimization and Its Fast Algorithm for Time-Variant System Identification," *IEEE Trans. Sig. Proc.*, 52(5): 1335-1342, 2004.
- [2] O.M. Macchi and N.J. Bershad, "Adaptive Recovery of a Chirped Sinusoid in Noise, Part1: Performance of the RLS Algorithm," *IEEE Trans. Sig. Proc.*, 39(3):583-594, 1991.
- [3] M. Hajivandi and W.A. Gardner, "Measuring of Tracking Performance for the LMS Algorithm," *IEEE Trans. Acou. Speech and Sig. Proc.*, 38(11):1953-1958, 1990.
- [4] S. Haykin, A.H. Sayed, J. Zeidler, P. Yee and P. Wie, "Tracking of Linear Time-Variant Systems," *IEEE MILCOM'95*, 2:602-606, 1995.
- [5] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, 1996.
- [6] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *JMLR S. I. Var. and Fea. Sel.*, 3:1157-1182, 2003.
- [7] B. Efron, I. Johnstone, T. Hastie, and R. Tibshirani, "Least Angle Regression," to appear in *Annals. of Statistics.*, 2004.
- [8] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *J. Royal Stat. Soc. B*, 58(1): 267-288, 1996.
- [9] S. Weisberg, *Applied Linear Regression*, Wiley, 1980.
- [10] T. Hastie, R. Tibshirani, and J. Friedman, *Elements of Statistical Learning: Data Mining, Inference and Prediction*, Springer, 2001.
- [11] A. Krogh and J.A. Hertz, "A Simple Weight Decay Can Improve Generalization," *In Advances in NIPS*, 4: 950-957, 1995.
- [12] E. Eweda, "Comparison of RLS, LMS, and Sign Algorithms for Tracking Randomly Time-Varying Channels," *IEEE Trans. Sig. Proc.*, 42(11): 2937-2944, 1994.
- [13] D. Erdogmus, "Information Theoretic Learning: Renyi's Entropy and Its Applications to Adaptive System Training," *Ph. D. Dissertation, Univ. of Florida*, 2002.