Optimization Methods and Software
Vol. 00, No. 00, Month 2006, 1–16

Real time input subset selection for linear time-variant MIMO systems

S.-P. KIM*†, J. C. SANCHEZ‡ and J. C. PRINCIPE†

†The Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, U.S.A.
‡The Department of Pediatrics, Division of Neurology, University of Florida, Gainesville, FL 32611, U.S.A.

(Received 15 May 2005; in final form 24 February 2006)

In this paper we propose an approach for multi-input multi-output (MIMO) system identification when the statistical relationship between input and output varies in input space as well as in time; i.e. nonstationary in space and time. An on-line variable selection algorithm, which has been recently developed for selecting a subset of input variables in real time by modifying least angle regression (LAR) with recursive estimators, is extensively applied to the linear time-variant MIMO systems. In our approach, a subset of input channels relevant with output is selected at every time instance based on the correlation between the filtering outcome of individual input channels and desired output. The on-line variable selection algorithm performs channel selection with weights using this real-time correlation. The proposed model is compared with a typical linear model in which only the least mean squares (LMS) is used to update system parameters. Tracking performances of these two models are demonstrated in a computer simulation and in a real-world application for tracking a linear relationship between neural firing rates of a primate and synchronously recorded hand kinematics. In both cases, our model demonstrates superior tracking performance.

Keywords: Time-variant multi-input multi-output systems; On-line channel selection; Least angle regression

1. Introduction

Tracking time-variant systems has been of great interest in many fields, including automatic control, communication systems, biological systems and signal processing. Adaptive filtering approaches have been widely used to track time-variant systems by adjusting filter coefficients with given input–output data in real time [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 mostly used for training linear adaptive systems, and their tracking abilities in nonstationary environment have been

*Corresponding author. Email: spkim@cs.brown.edu

Optimization Methods and Software
ISSN 1055-6788 print/ISSN 1029-4937 online
http://www.tandf.co.uk/journals
DOI: 10.1080/10556780600881886

extensively studied for the single input single output (SISO) cases [2–5]. However, the performance of these methods may be limited in the case of multi-input multi-output (MIMO) systems when both spatial (across inputs) and temporal statistics of the input change with time. These statistical changes often cause different subsets of the inputs to be involved in the mapping structures of the unknown system. In such cases, variable (or feature) selection techniques may play an important role in building better tracking systems.

A number of variable selection methods, including subspace projection, neural networks, support vector machine, and clustering [6], have been designed to operate in a stationary environment. However, these methods mostly utilize the entire training data by assuming stationarity. A recently developed on-line variable selection algorithm is capable of selecting input variables at every time instance [7]. This algorithm implements the on-line version of least angle regression (LAR), which has been proposed to perform forwards stagewise variable selection in a linear regression framework with the L_1-norm constraint on the coefficient vector [8]. It has been demonstrated that the on-line variable selection algorithm enhanced tracking performance for multivariate linear time-variant systems.

This on-line variable selection algorithm, however, can only be utilized in tracking systems when output is related to instantaneous multivariate input. If output is a function of the past history of input as well as current input, i.e. the linear MIMO system, then the on-line variable selection cannot be directly applied due to the temporal correlation of input (unless input is white noise); since the independent assumption between variables underlies the LAR based selection procedure, selecting variables in the temporal history of input may yield incorrect results. However, if we impose the independence assumption on not every single temporal history of input but only input channels, we may be able to utilize the on-line variable selection algorithm effectively. In this case, the on-line variable selection algorithm will perform 'on-line channel selection' to enhance tracking performance. But, applying on-line variable selection to input channels requires a set of new hyper variables. Theses variables should represent individual channels' statistics such that the correlation between each variable and desired output is coupled with the correlation of the input temporal history at the corresponding channel with desired output. In the linear MIMO system, which can be considered as a set of linear finite impulse response (FIR) filters placed at each channel, the output of each FIR filter may be a reasonable hyper variable since each filter is adapted to maximize the correlation between its output and desired output. Then, the subsequent on-line variable selection algorithm determines which channels are more correlated with current desired output. Figure 1 illustrates a conceptual diagram of our modelling scheme.

Our approach will be very useful in many biological applications in which a linear model is used to find a temporal relationship between two multidimensional factors. Especially, if one factor is coupled not only with instantaneous quantities but also with past history of the other factor, then the linear MIMO system with on-line channel selection will be well suited for modelling the temporal relationship. In addition, the selection properties of our model may help to gain insights of the relationship much better. We will demonstrate our model with a specific biological application in which a real time relationship between neural firing rates in the motor cortex of a primate and its hand kinematics is sought. The data are collected from a brain–machine interface (BMI) experiment where a monkey performs a three-dimensional reaching task using an arm [9]. The hand positions and neural activities are synchronously recorded during performance. The linear MIMO system is adapted to predict the hand position using neural firing rates in real time. A preliminary approach used the LMS algorithm to update parameters of the linear MIMO system [10]. In our approach, we preserve the topology of the linear MIMO system but update parameters using on-line channel selection as well as the LMS algorithm. We will demonstrate how this on-line channel selection can enhance tracking performance over the preliminary model.

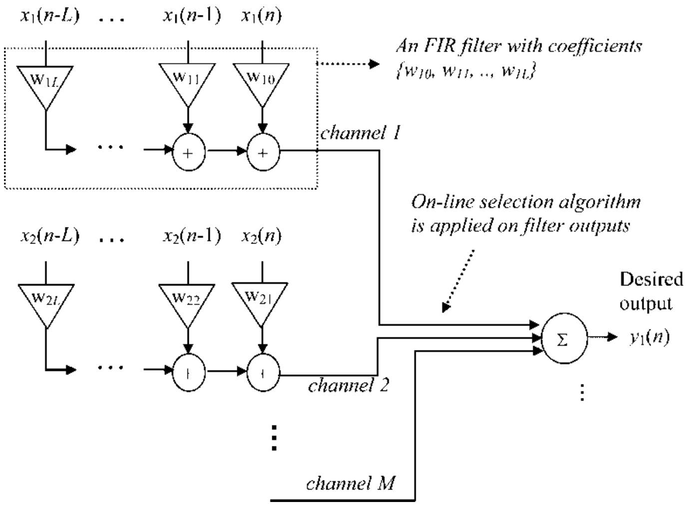

Figure 1. An illustration of the linear MIMO system with M input channels where up to L past inputs as well as an instantaneous input are related with current output. $x_i(n-l)$ denotes an input of the ith channel at (discrete) time instance $n-l$, and $y_j(n)$ is the jth output at time n. The output of each channel is a weighted sum of $L+1$ inputs.

In the next section, we will review the on-line variable selection algorithm with the LAR procedure. Then, the architecture and procedure of an on-line channel selection method will be presented. The following experimental results will demonstrate comparison of our approach with a conventional model in both a computer simulation and a specific biological application.

2. On-line variable selection based on least angle regression

In this section, we briefly review the on-line variable selection algorithm. This algorithm has been devised by modifying a forward selection procedure, LAR, using the recursive estimation of correlations between model variables. With these recursions, a subset of input variables that are relatively more correlated with output variables can be determined at every time instance.

2.1 *Least angle regression*

The LAR algorithm has been recently developed to accelerate computation and improve performance of forward model selection methods. It has been shown in Efron *et al.* [8] that simple modifications to LAR can implement the least absolute shrinkage and selection operator (LASSO) [11] and the forward stagewise linear regression [12]. However, the LAR algorithm only requires the same order of magnitude of computational complexity as ordinary least squares (OLS) while the LASSO requires a substantial higher complexity.

The LAR 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 starts 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. In fact, the constraint is imposed on the coefficients such that the sum of the absolute values of coefficients is kept to be less than the threshold. Since each input variable joins the selection at successive stages, the constraint can lead some coefficients to remain zeros. 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.

Now we describe the LAR procedure mathematically. Consider an $N \times M$ input matrix $\mathbf{X}$ (each row being M-dimensional sample vector) with x_{ij} being an element at ith row and jth column, and an $N \times 1$ desired response vector $\mathbf{y} = [y_1, \ldots, y_N]^T$. The model coefficient is initialized as $\beta_i = 0$, for $i = 1, \ldots, M$, and let $\beta = [\beta_1, \ldots, \beta_M]^T$. $\mathbf{X}$ and $\mathbf{y}$ are standardized such that $1/N \sum_{i=1}^{N} x_{ij} = 0$, $1/N \sum_{i=1}^{N} x_{ij}^2 = 1$ and $1/N \sum_{i=1}^{N} y_i = 0$, for $j = 1, \ldots, M$. Then, the initial LAR estimate becomes $\hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\beta} = \mathbf{0}$. At each stage, the correlation between input and current residual can be represented by

$$\mathbf{c} = \mathbf{X}^T(\mathbf{y} - \hat{\mathbf{y}}). \tag{1}$$

Each element c_j of $\mathbf{c}$ measures a correlation between the jth input dimension and residual. Let $C_{\max}$ be the maximum value of $|c_j|$. The joint selection set A is then defined as

$$A \equiv \{j : |c_j| = C_{\max}\}. \tag{2}$$

Consider a matrix $\mathbf{X}_a$ consisting of the jth columns of $\mathbf{X}$ for $j \in A$, multiplied by the sign of correlation as

$$\mathbf{X}_a \equiv \{\ldots, \mathrm{sign}(c_j)\mathbf{x}_j, \ldots\} \quad \text{for} \;\; j \in A. \tag{3}$$

where $\mathbf{x}_j = [x_{1j}, x_{2j}, \ldots, x_{Nj}]^T$. Then, $\mathbf{X}_a$ has a dimension of $N \times |A|$ where $|A|$ denotes a size of A. In order to find the equiangular vector between the joint variables, we first define a matrix $\boldsymbol{\Phi}$ and a scalar value α such that

$$\begin{aligned} \boldsymbol{\Phi} &\equiv \mathbf{X}_a^T\mathbf{X}_a \\ \alpha &\equiv (\mathbf{1}_a^T\boldsymbol{\Phi}^{-1}\mathbf{1}_a)^{-1} \end{aligned} \tag{4}$$

where $\mathbf{1}_a$ is a vector of ‘one’s with a length equal to $|A|$. The equiangular vector $\boldsymbol{\mu}$ is then determined as

$$\boldsymbol{\mu} = \mathbf{X}_a(\alpha\boldsymbol{\Phi}^{-1}\mathbf{1}_a). \tag{5}$$

Note that $\|\boldsymbol{\mu}\| = 1$ and $\mathbf{X}_a\boldsymbol{\mu} = \alpha\mathbf{1}_a$ (angles between all $\mathbf{x}_j$, $j \in A$ and $\boldsymbol{\mu}$ are equal to each other). Now, a step size by which we move in the direction of $\boldsymbol{\mu}$ is computed by

$$\gamma = \min{}^{+}_{j \in A^C} \left\{ \frac{C_{\max} - c_j}{\alpha - \theta_j}, \frac{C_{\max} + c_j}{\alpha + \theta_j} \right\} \tag{6}$$

where $\min^+$ indicates considering only positive minimum values over j and θ_j is defined as

$$\theta_j = \mathbf{x}_j^T\boldsymbol{\mu}, \quad \text{for} \;\; j = 1, \ldots, M \tag{7}$$

Or, $\theta = \mathbf{X}^T\boldsymbol{\mu}$ if we define a vector $\theta = [\theta_1, \ldots, \theta_M]^T$. Then, quantities in the parenthesis of the right-hand side of equation (6) are evaluated only for $j \in A^c$ to determine γ. Finally, we update the regression output as

$$\hat{\mathbf{y}}_+ = \hat{\mathbf{y}} + \gamma\boldsymbol{\mu} \tag{8}$$

A procedure in equations (1) to (8) is repeated until all inputs join A, or $\sum_j |\beta_j|$ exceeds a given threshold, denoted by $\xi > 0$. Therefore, $\sum_j |\beta_j|$ is evaluated at every step and if $\sum_j |\beta_j| > \xi$ then the procedure stops even before including all inputs in A.

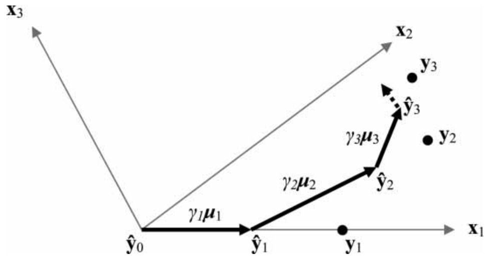

Figure 2. An illustration of LAR procedure.

The above LAR procedure is illustrated in figure 2 (modified from [8]). In this figure, we start to move on the first selected input variable $\mathbf{x}_1$ and step forward as successive variables $\mathbf{x}_j$ for $j > 1$ join the selection set. $\boldsymbol{\mu}_j$ is the equiangular unit vector as described in equation (5). The amount of movement along $\boldsymbol{\mu}_j$ is denoted by γ_j. $\mathbf{y}_j$ is the OLS estimate of desired response $\mathbf{y}$ with input $\mathbf{x}_1$. Note that the estimate by LAR ($\hat{\mathbf{y}}_1$) moves toward $\mathbf{y}_1$, but does not reach it.

Therefore, the LAR estimate is in the exactly same direction as the OLS estimate at every step. In fact, if the procedure progresses until all input variables join the selection set, the LAR estimate becomes the OLS estimate. On the other hand, if the procedure stops before including all input variables, the LAR estimate is in the same direction as the OLS estimate obtained only by the selected variables, with a smaller magnitude.

Determination of the threshold ξ belongs to the model selection problem (bias-variance dilemma) [13]: if ξ is too small, bias will be increased while variance is decreased; if it is too large, variance will be increased while bias is decreased. A choice of ξ will be greatly dependent on a specific problem. A detail discussion of this issue is beyond the scope of the present paper, but this problem will be briefly revisited in section 3.2 for the linear MIMO system case.

2.2 *On-line variable selection*

The LAR algorithm has been applied to variable selection in multivariate data analysis (static problems), but if properly modified, LAR can effectively provide a tool for time varying model selection with the L_1-norm penalty. However, since LAR selects input variables by computation of correlation using the entire data set, it requires the assumption of stationary statistics. Therefore, a modified version of LAR, which selects a subset of input variables locally in time without the stationary assumption, has been proposed in Kim *et al.* [7]. This algorithm is able to implement real-time variable selection for time-variant system identification problems.

Correlation between inputs and the desired response can be estimated in real time by recursively updating the correlation vector. The input covariance matrix can also be estimated recursively. By decoupling the variable selection part from the model update part in LAR, the input variables can be selected locally with the recursive estimates of correlations. The modification to the LAR procedure using these recursions is described as follows.

First, the correlation in equation (1) at a given step can be simply updated without computing residuals if we plug equation (8) into equation (1), given by

$$c_j(k) = c_j(k-1) - \gamma\theta_j \tag{9}$$

for the kth step of variable selection. Hence, the update procedure of equation (8) can be removed.

Next, instead of computing the correlation with entire data samples, we can recursively estimate the correlation using a forgetting factor $0 < \rho < 1$, given by

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

where $\mathbf{x}(n)$ is an $1 \times M$ input vector at time instance n ($\mathbf{x}(n)$ can also be considered as the nth row of $\mathbf{X}$ if $\mathbf{X}$ is a temporal data matrix). $\mathbf{p}(n)$ is utilized within the LAR routine as $\mathbf{c}(0) = \mathbf{p}(n)$. The forgetting factor ρ is data dependent and pre-determined by taking the nonstationarity of the data into account: ρ is set small (e.g. $\rho \approx 0.8$) to include only recent past samples in estimation if the statistics of the data change rapidly and large (e.g. $\rho \approx 0.99$) to include more past samples if the statistics change slowly.

For the computation of the covariance matrix $\mathbf{\Phi}$ in equation (4), we also estimate the input covariance matrix as

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

This matrix is not directly used for determining $\mathbf{\Phi}$ since $\mathbf{\Phi}$ is the covariance of only a subset of input variables. Also, the input vectors are multiplied by the sign of correlation before computing $\mathbf{\Phi}$. Therefore we need to introduce a diagonal matrix $\mathbf{S}$ whose elements are signs of $c_j(k)$ for $j \in A$. In addition, another matrix $\mathbf{R}_a$ is composed by the elements $r_{ij}(n)$ of $\mathbf{R}(n)$ for $i, j \in A$. Then $\mathbf{\Phi}$ can be computed using $\mathbf{R}_a$ and $\mathbf{S}$ as

$$\mathbf{\Phi} = \mathbf{S}\mathbf{R}_a\mathbf{S} \tag{12}$$

Note that a $|A| \times |A|$ matrix $\mathbf{R}_a$ represents the covariance between the joint input variables in A. A scalar value α can also be computed in the same way as in equation (4) using $\mathbf{\Phi}$ obtained from this equation.

To remove the computation of the equiangular vector $\boldsymbol{\mu}$ that requires a batch operation, while making it possible to obtain the step size γ, we plug equations (5) into (7) as

$$\theta = \mathbf{X}^T\boldsymbol{\mu} = \mathbf{X}^T\mathbf{X}_a(\alpha\mathbf{\Phi}^{-1}\mathbf{1}_a) = \alpha\mathbf{X}^T\mathbf{X}_a\mathbf{\Phi}^{-1}\mathbf{1}_a. \tag{13}$$

By noticing that $\mathbf{X}^T\mathbf{X}_a$ results in a matrix identical with the one with the jth columns of $\mathbf{R}(n)$ (for $j \in A$) multiplied by $\mathbf{S}$, we define a matrix $\mathbf{R}_{acol}$ consisting of the jth columns of $\mathbf{R}(n)$ for $j \in A$. Then, equation (13) can be rewritten as

$$\theta = \alpha\,\mathbf{R}_{acol}\mathbf{S}\mathbf{\Phi}^{-1}\mathbf{1}_a \tag{14}$$

This modification removes the computation of the equiangular vector in equation (5) which is not directly required for computing θ and γ.

With the procedures described so far, we can complete the modified LAR routine for on-line variable selection. In summary, the model is given by input $\mathbf{x}(n)$ and output $y(n)$ at time instance n. From these samples, $\mathbf{R}(n)$ and $\mathbf{p}(n)$ are estimated and fed into the LAR routine modified by equations (9) to (14). This routine is executed iteratively until $\sum_j |\beta_j(n)| > \xi$ where $\beta_j(n)$ a coefficient for the jth input variable estimated at n, and ξ is a pre-determined threshold. The procedure of this on-line variable selection is described in table 1.

An illustrative example of a linear adaptive system with the on-line variable selection algorithm is shown in figure 3. In this architecture, we seek to identify the linear time-variant target system which can be described as

$$y(n) = \mathbf{h}(n)^T\mathbf{x}(n) + v(n) \tag{15}$$

where $\mathbf{x}(n)$ is an input vector, $\mathbf{h}(n)$ is a system parameter vector, $v(n)$ is a measurement noise and $y(n)$ is the system output. We assume that $\mathbf{h}(n)$ is sparse such that some of its elements

Table 1. On-line variable selection algorithm (taken from [7]).

Given an $N \times M$ input matrix $\boldsymbol{X}$ (each row being M-dimensional sample vector), and an $N \times 1$ desired response matrix $\boldsymbol{Y}$, initialize

$$\boldsymbol{p}(0) = \mathbf{0} \text{ and } \boldsymbol{R}(0) = \mathbf{0}$$

Transform $\boldsymbol{X}$ and $\boldsymbol{Y}$ such that

$$\frac{1}{N}\sum_{i=1}^{N} x_{ij} = 0,\ \frac{1}{N}\sum_{i=1}^{N} x_{ij}^2 = 1,\ \text{and } \frac{1}{N}\sum_{i=1}^{N} y_i = 0 \text{ for } j = 1, \ldots, M.$$

Update the correlation: $\boldsymbol{p}(n) = \rho \boldsymbol{p}(n-1) + d(n)\boldsymbol{x}(n)$

Update the input covariance: $\boldsymbol{R}(n) = \rho \boldsymbol{R}(n-1) + \boldsymbol{x}(n)^T \boldsymbol{x}(n)$

Let $\boldsymbol{c}(0) = \boldsymbol{p}(n)$.

For $k = 0, \ldots, M-1$

Let $C_{\max} = \max_j\{|c_j|\}$, *and* $A = \{j : |c_j(k)| = C_{\max}\}$.

Compute a diagonal matrix $\boldsymbol{S}$ with elements of sign of $c_j(k)$ for $j \in A$.

$$\boldsymbol{\Phi} = \boldsymbol{S}\boldsymbol{R}_a\boldsymbol{S},$$

where $\boldsymbol{R}_a$ is submatrix of $\boldsymbol{R}(n)$ with jth rows and jth columns for $j \in A$.

$$\alpha = (\boldsymbol{1}_a^T \boldsymbol{\Phi}^{-1} \boldsymbol{1}_a)^{-1}$$

$$\theta_j = \alpha \boldsymbol{R}_{acol} \boldsymbol{S} \boldsymbol{\Phi}^{-1} \boldsymbol{1}_a,$$

where $\boldsymbol{R}_{acol}$ is a matrix consisting of j^{th} columns of $\boldsymbol{R}(n)$ for $j \in A$.

Compute the step size, $\gamma = \min^{+}_{j \in A^C} \left\{ \frac{C_{\max} - c_j}{\alpha - \theta_j}, \frac{C_{\max} + c_j}{\alpha + \theta_j} \right\}$.

Update correlation: $c_j(k) = c_j(k) - \gamma\theta_j$.

are zero. Also, $\mathbf{h}(n)$ is time varying, which implies that its nonzero elements change over time and space; e.g. one group of elements have nonzero coefficients in some period and a different group of elements have nonzero elements in the following period. Typically, the target system can be approximated by a linear adaptive system based on the mean squared error (MSE) criterion. This linear system is adapted continuously by updating the parameters $\mathbf{w}(n)$ with a stochastic gradient such that

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \Delta\mathbf{w}(n) \tag{16}$$

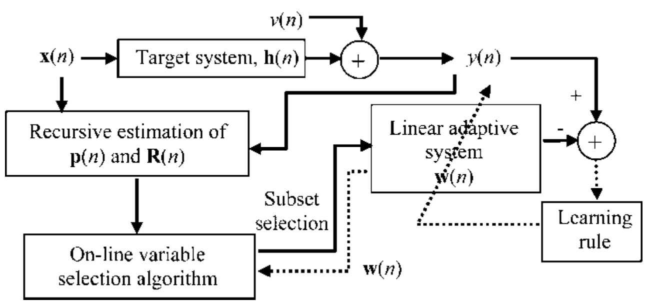

Figure 3. An example of the linear adaptive system to track a time-variant sparse system. The on-line variable selection algorithm is utilized along with the typical linear adaptive system to select an instantaneously correlated input subset.

where $\Delta\mathbf{w}(n)$ represents the stochastic update of $\mathbf{w}(n)$. For this example, we assume that the gradient is used to minimize $E[\|\mathbf{h}(n) - \mathbf{w}(n)\|^2]$.

Since the target system is sparse and time-variant, the on-line variable selection algorithm can be added to train the linear adaptive system, determining which inputs in $\mathbf{x}(n)$ are more correlated with the target output $y(n)$. Then, only the selected input weights are updated, reducing estimation noise caused by irrelevant inputs. To enable on-line variable selection, $\mathbf{R}(n)$ and $\mathbf{p}(n)$ are recursively estimated from the given $\mathbf{x}(n)$ and $y(n)$ and passed to the modified LAR algorithm. $\mathbf{w}(n)$ is also used to determine when to stop the search for correlated inputs with a criterion as $\sum_j |w_j(n)| > \xi$. The final weight update only utilizes this subset of selected inputs j

$$w_j(n+1) = w_j(n) + \Delta w_j(n), \quad j \in A \tag{17}$$

where A is a set of indices for the selected inputs. This procedure is repeated at every time instance n.

3. Real-time input selection for linear time-variant MIMO systems

3.1 *Time-variant linear MIMO system*

In the linear MIMO system considered in the present paper, the output is assumed to be linearly and causally correlated with the spatio-temporal pattern of input channels. The temporal pattern of each input channel is represented by embedding the input time series using a time delay line. In our representation, only a discrete time series is considered. Hence, an input sample is delayed by a discrete time delay operator, denoted by z^{-1} in the Z-domain. Let $x_j(n)$ be an input sample for the jth channel at time instance n. The temporal pattern of the jth channel is represented in the embedded vector space as $\mathbf{x}_j(n) \in \Re^L$, where L is the number of taps in the delay line. The linear approximation of the time-varying input–output mapping between all input channels and desired output, denoted by $y(n)$, is given by

$$y(n) = \sum_{j=1}^{M} \mathbf{w}_j(n)^T \mathbf{x}_j(n) + b(n) \tag{18}$$

where $\mathbf{w}_j(n) \in \Re^L$ is the coefficient vector for the jth channel at time instance n for $j = 1, \ldots, M$ and $b(n)$ is a bias term. Notice that the operation of $\mathbf{w}_j(n)^T \mathbf{x}_j(n)$ can be considered as filtering the jth channel input signal by a linear time-variant finite impulse response (FIR) filter. Therefore, $\sum_j \mathbf{w}_j(n)^T \mathbf{x}_j(n)$ can be the sum of the FIR filter outputs from all the channels. We can also remove the bias term $b(n)$ by zeroing the mean of the input and output. Note that $y(n)$ is a one-dimensional time series but it can be easily extended to a multi-dimensional signal just by augmenting equation (18) multiple times. Therefore, we will present our approach for single output case in this section.

The coefficient vector $\mathbf{w}_j(n)$ can be adapted by a number of methods, among which the LMS algorithm plays a central role due to its computational simplicity and tracking capability [5]. With the LMS algorithm, the coefficient vector is updated as

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) + \eta e(n) \mathbf{x}_j(n) \tag{19}$$

for $j = 1, \ldots, M \cdot \eta$ is a learning rate, controlling convergence speed and the misadjustment, and $e(n)$ is the instantaneous error such that

$$e(n) = y(n) - \hat{y}(n) = y(n) - \sum_{j=1}^{M} \mathbf{w}_j(n)^T \mathbf{x}_j(n). \tag{20}$$

See [4] for the review of the tracking linear time-variant systems using the LMS.

3.2 *On-line input channel selection for linear MIMO systems*

Direct application of the on-line variable selection algorithm to the embedded input space of a MIMO system may not be feasible owing to correlation over time lags, which prevents the LAR algorithm from finding appropriate subsets. In order to correct this shortcoming we developed the variable selection scheme explained above.

Assuming linear independence among input channels in MIMO systems, we may be able to apply on-line variable selection to channels instead of every element in $x_j(n)$ for $j = 1, \ldots, M$. To that end, we selected the weighted sum of each channel as a hyper variable which represents the trajectory in the subspace. These filtered outputs can indicate the relationship between desired outputs and the input temporal patterns at each channel. Since on-line variable selection is based on correlation, the filter output is hypothesized to be a sufficient variable to provide the correlation information between desired outputs and input temporal patterns. We propose to utilize the LMS update rule for the individual FIR filter coefficients.

Figure 4 depicts the overall architecture of the proposed on-line input channel selection approach. The embedded input vector $\mathbf{x}_j(n)$ at the jth channel is filtered by an FIR filter with a coefficient vector of $\mathbf{w}_j(n) = [w_{j0}(n), \ldots, w_{jL-1}(n)]^T$, yielding the filter output vector $\mathbf{z}(n) = [z_1(n), \ldots, z_M(n)]^T$. $w_{jl}(n)$ is weight at time n for the tap of l delay(s) of the jth channel. The auto-covariance matrix $\mathbf{R}(n)$ of $\mathbf{z}(n)$ and the cross-correlation vector $\mathbf{p}(n)$ between $\mathbf{z}(n)$ and $y(n)$ are recursively estimated by equations (10) and (11), respectively. Then, the on-line variable selection algorithm receives $\mathbf{R}(n)$ and $\mathbf{p}(n)$ to yield a LAR coefficient vector $\mathbf{g}(n) = [g_1(n), \ldots, g_M(n)]^T$ based on the routine described in table 1. Note that some of the elements in $\mathbf{g}(n)$ can be zero due to the L_1-norm constraint in the LAR procedure. Since the estimate of desired output is the weighted sum of the channel filter outputs, the instantaneous

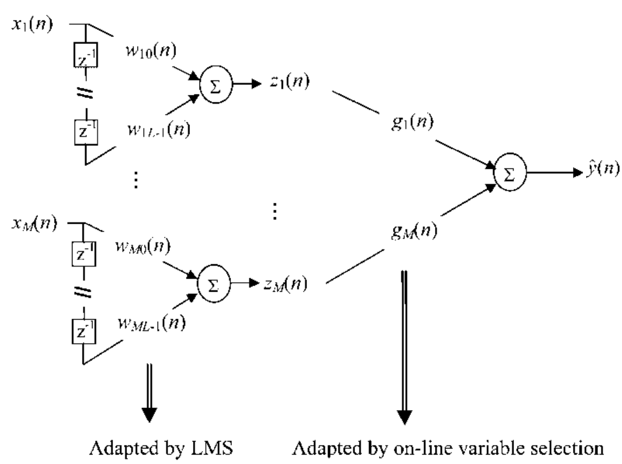

Figure 4. A diagram of the architecture of on-line channel selection method.

error becomes

$$e(n) = y(n) - \hat{y}(n) = y(n) - \mathbf{g}(n)^T \mathbf{z}(n). \tag{21}$$

The update of $\mathbf{w}_j(n)$ is then given by

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) + \eta e(n) g_j(n) \mathbf{x}_j(n). \tag{22}$$

Notice the difference between this update rule and the one in equation (19). This leads to two-stage learning since $\mathbf{g}(n)$ is first updated and then back-propagated to the FIR filters for updating $\mathbf{w}(n)$. Hence, the topology of this system can be regarded as a two-layer network similar to the multilayer perceptrons (MLP) [14], but linear.

Since the coefficient vector $\mathbf{g}(n)$ is determined from the LAR routine, the adaptation of this variable can be regarded as a soft selection. Also, $g_i(n)$ can take a negative value since it is nothing but a regression coefficient from the LAR procedure: regression of $y(n)$ on $\mathbf{z}(n)$ using LAR. Where this procedure differs from ordinary least squares regression in on the constraint $\sum_i |g_i(n)|$ and the recursive in time stepwise variable selection of the LAR. Note that it is critical to choose an appropriate threshold for $\sum_i |g_i(n)|$ if there is no information about the parameters of the target system. For on-line channel selection, one possible way to determine the threshold is empirically choosing the minimum threshold such that recursive average tracking error ($E[e(n)]$) can be made sufficiently small.

With the proposed model, we hope to enhance the tracking ability of linear MIMO models for systems whose statistics change in space as well as in time. Especially when not every channel is related with the desired output at every time instance and different subsets of input channels are correlated in different time periods, our model will be greatly useful for tracking time-variant MIMO systems. The following experimental examples will demonstrate how much our model can improve tracking performance over convectional models.

4. Experiments

4.1 *Computer simulations*

We first demonstrate the variable selection performance of our model with synthetic data. In our computer simulations, we aim to track a linear time-variant MIMO system.

A multivariate discrete time series generated from the Gaussian distribution is used as input to this system, with M variables (channels) and N samples. The time series at each channel is embedded through an L-tap time delay line. The target output is obtained from equation (18), that is a linear weighted sum of all tap inputs (from every channel) corrupted by measurement noise. This system model comprises multiple input channels with temporal embedding and single output. It is straightforward to extend this model to a multiple output case in which the linear combinations of inputs can be individually constructed multiple times. Hence, this model constitutes a basic building block of linear MIMO system identification. The weights for this linear combination are assumed to evolve randomly in time, described in the following equation given by

$$w_{ij}^o(n+1) = w_{ij}^o(n) + \omega(n) \tag{23}$$

where, $w_{ij}^o(n)$ is a weight for the ith tap of the jth channel at time instance n and $\omega(n)$ is a Gaussian random noise with zero mean and variance of σ^2. A variance σ^2 is determined for this model such that $w_{ij}^o(n)$ does not change too rapidly. The N-sample data are equally divided into K regimes in which different subsets of input channels are linearly related with desired outputs; i.e. it is assumed that only a subset of input channels has nonzero weights during a

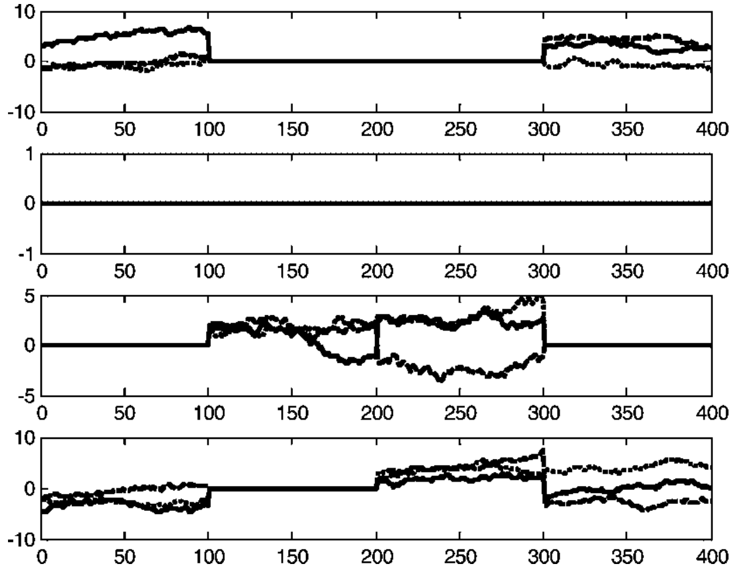

Figure 5. An example of the change of weights for each channel in time.

given regime. The subset size for each regime is randomly chosen from a uniform distribution ranging between $M/2 - 2$ to $M/2 + 2$. Figure 5 depicts an example of weight time evolution, with $M = 4$, $N = 400$, $L = 3$, $K = 4$ and $\sigma = 0.2$. In this figure, each subplot displays the temporal change of $\mathbf{w}_j^o(n) = [w_{0j}^o(n), \ldots, w_{L-1j}^o(n)]^T$ for $j = 1, \ldots, M$.

Two linear models are designed to track this MIMO system; one trained only by the LMS algorithm and the other by the described on-line channel selection algorithm. Both systems have exactly the same topology as the target system to simplify tracking simulations. However, if the target system order is unknown, the number of taps in the adaptive MIMO system can be set large enough to take into account the intrinsic dynamics of the input data, as specified by order selection methods. In such a case, significant weights can be imposed on the taps which represent the dynamics. This will give rise to parsimonious linear models, which can be implemented by sparse channel estimation methods (e.g. see Cotter and Rao [15] for a matching pursuit based channel estimation approach), in order to avoid over-fitting of the model.

The tap weights in each model are trained using equations (19) for the LMS and (22) for on-line channel selection, respectively. These tracking models do not have *a priori* knowledge about which channels have nonzero weights in each regime.

Common measures of tracking assessment, which are the steady-state mean squared deviation and the steady-state excess mean squared error, are utilized to evaluate each tracking model [16]. The mean squared deviation between the true weight vector $\mathbf{w}_j^o(n)$ and the learned weight vector $\mathbf{w}_j(n)$ for every j is defined as

$$D = \lim_{n\to\infty} E\left[\sum_{j=1}^{K} \|\mathbf{w}_j^o(n) - \mathbf{w}_j(n)\|^2\right]. \qquad (24)$$

The excess MSE is defined as

$$\xi = \lim_{n\to\infty} E[e(n)^2] - \varepsilon^2 \qquad (25)$$

Table 2. Mean ratios of D and ξ in each regime.

M	4		8		12	
σ	0.01	0.1	0.01	0.1	0.01	0.1
$\frac{D^{\text{selection}}}{D^{\text{LMS}}}$	0.92	0.89	0.75	0.79	0.79	0.82
	0.75	0.83	0.75	0.79	0.68	0.68
	0.85	0.84	0.76	0.79	0.56	0.59
	0.78	0.79	0.77	0.80	0.52	0.51
$\frac{\xi^{\text{selection}}}{\xi^{\text{LMS}}}$	0.28	0.43	0.32	0.33	0.35	0.30
	0.51	0.42	0.48	0.48	0.38	0.31
	0.61	0.50	0.44	0.33	0.22	0.23
	0.62	0.49	0.55	0.44	0.19	0.29

where ε^2 is the variance of white noise added to desired output. In the case of the on-line channel selection model, D is computed slightly differently such that

$$D = \lim_{n\to\infty} E\left[\sum_{j\in A(n)} \|\mathbf{w}_j^o(n) - \mathbf{w}_j(n)\|^2\right] \tag{26}$$

where $A(n)$ is the set of selected channels at time instance n. Therefore, the deviation between true weight vectors and the estimated ones is considered only for the selected channels. Since there are K independent regimes in which different subsets of input channels are correlated with desired output, 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 a learning rate (η) for the LMS, 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 range of η, and pick the one for which $(D + \xi)/2$ is minimum. The settings of model parameters used in this simulation are $\{N, K, L, \eta, \rho = 400, 4, 10, 0.01, 0.95\}$.

The evaluation results of D and ξ based on the above setting of parameters are summarized in table 2. In this table, the ratios of D and ξ for the on-line channel section model to those for the LMS-only model are presented for different values of M. These results show the superior tracking performance of the on-line channel selection model. We can also observe a tendency that the improvement of tracking performance due to on-line channel selection increases for larger number of channels.

Figure 6 shows the comparison of learning dynamics between the LMS-only model and on-line channel selection model. It presents the evolution of $D^{\text{selection}}/D^{\text{LMS}}$ and $\xi^{\text{selection}}/\xi^{\text{LMS}}$ over time for different M. The results show that the trajectories of these ratios are less than unity most of the time, which again indicates the superior tracking performance of the on-line channel selection model.

In figure 7, the selection error rate, which is defined as the ratio of the false selection to the total number of channels, is averaged over 100 Monte Carlo runs. This rate decreases with time in a given regime since the recursive estimates of $\mathbf{R}(n)$ and $\mathbf{p}(n)$ become closer to the true ones with more samples. It empirically shows that the selection errors are minimized around $\rho = 0.95$ on average. The choice of ρ, however, will depend upon the data.

The computational cost of our approach can be considered as a sum of two parts due to two-stage learning scheme; the LMS update and the on-line variable selection algorithm. The computation in the LMS update is very effective with complexity $O(N)$ where N is the

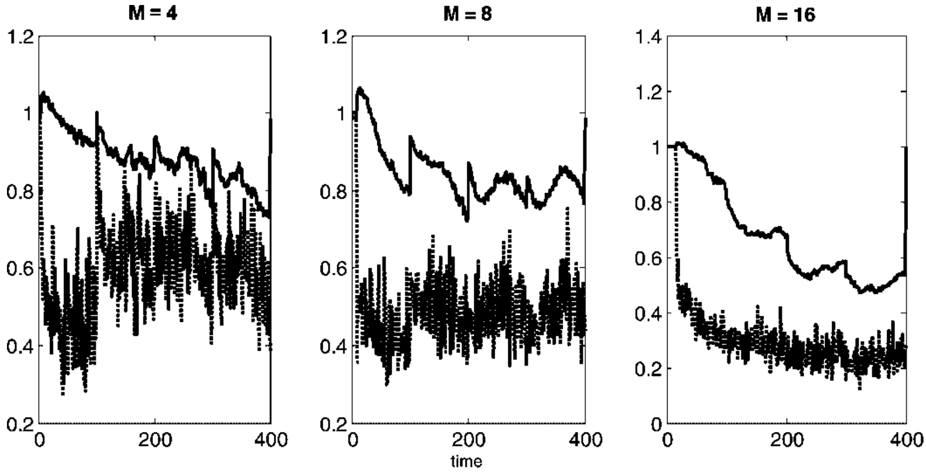

Figure 6. Evolutions of the ratio of $D^{\text{selection}}/D^{\text{LMS}}$ (solid lines) and $\xi^{\text{selection}}/\xi^{\text{LMS}}$ (dotted lines) during tracking over 400 samples ($\sigma = 0.1$).

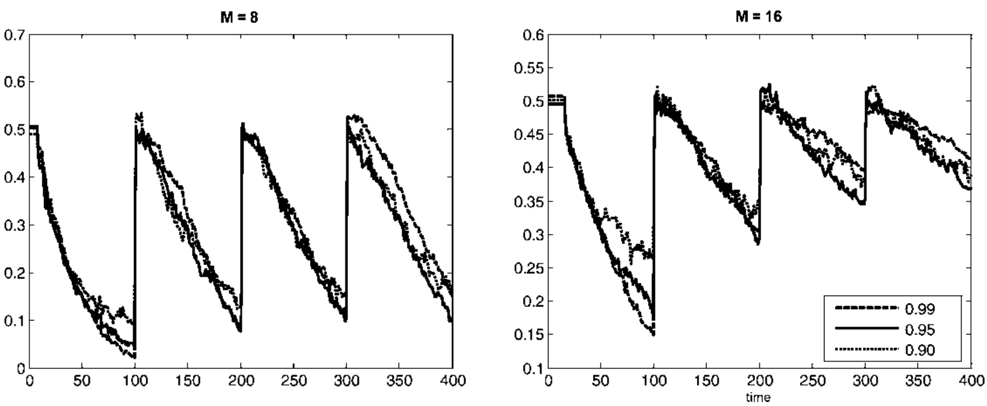

Figure 7. Evolution of the selection error percentage averaged over 100 Monte Carlo runs during tracking.

number of parameters. The computation in the variable selection portion of the algorithm includes updating $\mathbf{R}(n)$ and $\mathbf{p}(n)$, and repeating the modified LAR routine until the threshold of the L_1-norm of parameter vector is reached. Since the algorithm deals with input channels that are as few as $1/L$ of the number of parameters in the LMS (L is the number of taps), the recursive update of $\mathbf{R}(n)$ and $\mathbf{p}(n)$ without matrix inversion does not significantly increase computational cost. The matrix inversion in the modified LAR routine is not computationally expensive since the dimension of a matrix $\mathbf{\Phi}$ starts from 1 and increases linearly as the routine proceeds. The number of iteration of the modified LAR routine is also typically less than the number of channels. In fact, the simulation shows that the computational complexity of the on-line variable selection algorithm is approximately same order of the RLS, which has been widely applied as a real time tracking algorithm.

4.2 *Experiments with neural activity data*

In this section, we demonstrate the performance of our model in biological applications: a linear MIMO system is designed to approximate the functional mapping from the neural activity of neuronal ensemble in the primary cortex of a monkey to arm movements. Two linear MIMO systems are learned using either the LMS or the on-line channel selection algorithm. The tracking performances of two systems are compared.

4.2.1 Data description. The firings of 104 cells were collected by microwire arrays implanted in several cortical areas – posterior parietal cortex (PP), left and right primary motor cortex (M1),and dorsal premotor cortex (PMd) – while an owl monkey (*Aotus trivirgatus*) performed the reaching task. In this task, the monkey was trained to reach for food randomly placed in a tray. This movement consisted of four phases: reaching for food, grasping food, taking food to mouth, and returning to the resting position. Both three-dimensional (3D) hand positions and neural activity were recorded simultaneously during the task. Further the details of experimental setup can be found in [9].

The collected signals were then spike sorted to individualize the firings from each neuron. The cells firings were counted in non-overlapping 100 milliseconds time windows, and a 1 s time window was selected as an appropriate memory depth to derive the best linear projector [9]. The primate's hand position, used as the model desired signal, was also recorded (with a time shared clock) and digitized with 200 Hz sampling rate. The desired hand position signal was then downsampled to 10 Hz to be aligned synchronously with firing count data for linear models. The data were collected by Dr Nicolelis group at Duke University.

4.2.2 Results. A linear MIMO system is designed to track the trajectory of hand positions in 3D space from input of multiple neuronal firing rates. Since the desired response lies in the 3D space, three different linear systems described as figure 3 and equation (18) are constructed in parallel. In these data, the dimension of input $\mathrm{x}(n)$ is set as $M = 104$ and the number of taps at each neuronal channel is $L = 10$ accounting for 1 s time window. The output of the system $y(n)$ is one of three Cartesian coordinate of hand position.

For comparison, two learning algorithms are applied to adapt the linear MIMO system (with same topology) to track the hand position. The first system, denoted as SYS1, is trained only by the LMS algorithm and the second, denoted as SYS2, is trained by the on-line channel selection algorithm that we propose. System SYS1 has been previously used for predicting the position of the hand position, resulting in reasonable performance [10]. Both algorithms are run over $N = 3000$ samples (300 s). The optimal learning rate of LMS is determined to maximize the tracking performance of SYS1. The same rate is employed in SYS2 to update the filter tap weights. The forgetting factor in the recursive estimates in equations (10) and (11) is empirically determined as 0.8.

In figure 8, a typical tracking performance of two systems is compared. The sample outputs of both SYS1 and SYS2 are displayed on top of the actual trajectory in the z-coordinate of hand position. Although more systematic performance measurements in terms of the mean squared deviation and misadjustment must be conducted in future studies, we can clearly see from this figure that SYS2 identifies the peaks of hand trajectory a lot better than SYS1. This superior tracking performance of SYS2 may result from the fact that additional spatial filtering by the on-line channel selection algorithm optimally combines filter outputs to reduce the instantaneous error between desired response and the estimation. In other words, the sparse set of coefficients in spatial filtering $\mathbf{g}(n)$ reduces noise from irrelevant channels locally in time.

The neuronal channels selected on-line through a series of reaching movements reflect groups of neurons that are most correlated with movements. This group of neurons was compared with the result from a preliminary sensitivity analysis on the same data in which neurons

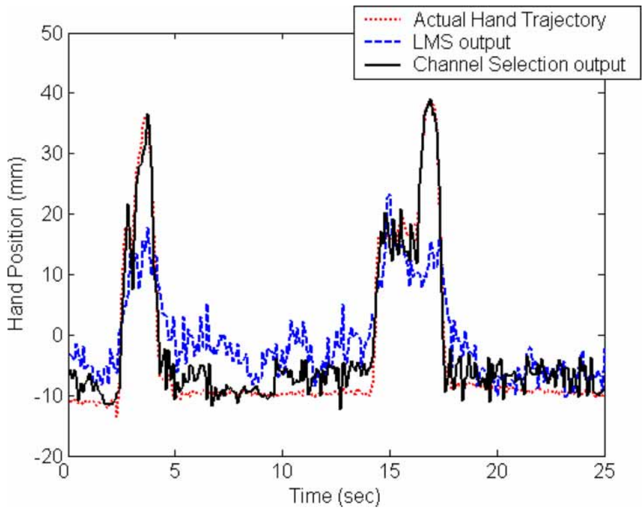

Figure 8. An example of the outputs of two tracking systems; SYS1 (dashed line) and SYS2 (solid line), on top of actual hand trajectory (dotted line).

are sorted by evaluating their sensitivity to the output of models [17]. It turns out that our neuronal subset matches well the neurons exhibiting the highest sensitivities. More comprehensive analyses of neural activity using our model will be explored in a follow-up study.

5. Conclusions

An on-line channel section method for identification of linear time-variant MIMO systems has been proposed to enhance tracking performance. This on-line channel selection algorithm is based on the LAR procedure which selects correlated variables in a stagewise fashion with the constraints on the L_1-norm of coefficient vector. The on-line variable selection algorithm that was developed by simplifying the LAR procedure using recursive estimators is applied to the outputs of the FIR filters at every channel. The evaluations of the proposed method compared with a conventional tracking model using only the LMS have demonstrated superior performance in computer simulations. Besides the enhancement of MIMO system tracking performance, we anticipate that this on-line channel selection method will be of great importance to the analysis of many real applications of nonstationary MIMO models. Indeed, we demonstrate our method in an on-line channel selection for tracking a time-variant relationship between neural firing rates and hand kinematics of a monkey. An obvious improvement of tracking performance using our model compared to a preliminary linear MIMO system was found. Besides tracking improvement, it would be more interesting to analyze the sequence of neuronal subsets and corresponding kinematic properties to uncover a temporal relationship from individual areas of motor cortex and continuous behaviour in real time. We will pursue this topic in future studies. One of the issues that need to be addressed in this methodology is the requirement for a desired response to apply the LAR procedure. For analysis applications as in the BMI, or in tracking applications the desired response is assumed available, so this requirement is not a limiting factor. However, there are a lot of important applications in system identification and learning where during testing no desired signal is available.

Acknowledgements

This work was partially supported by grants NSF CNS-0540304, ECS-0422718, ONR-450595112. The authors extend special thanks to Dr Miguel A. L. Nicolelis for sharing the data collected in his laboratory.

References

[1] Nishiyama, K., 2004, An H_∞ optimization and its fast algorithm for time-variant system identification. *IEEE Transactions on Signal Processing*, **52**, 1335–1342.

[2] Macchi, O.M. and Bershad, N.J., 1991, Adaptive recovery of a chirped sinusoid in noise, part1: performance of the RLS algorithm. *IEEE Transactions on Signal Processing*, **39**, 583–594.

[3] Hajivandi, M. and Gardner, W.A., 1990, Measuring of tracking performance for the LMS algorithm. *IEEE Transactions on Acoustic Speech and Signal Processing*, **38**, 1953–1958.

[4] Haykin, S., Sayed, A.H., Zeidler, J., Yee, P. and Wie, P., 1995, Tracking of linear time-variant systems. *IEEE MILCOM'95*, **2**, 602–606.

[5] Haykin, S., 1996, *Adaptive Filter Theory* (New Jersey: Prentice Hall).

[6] Guyon, I. and Elisseeff, A., 2003, An introduction to variable and feature selection. *JMLR S. I. Var. and Fea. Sel.*, **3**, 1157–1182.

[7] Kim, S-.P., Rao, Y.N., Erdogmus, D. and Principe, J.C., 2004, Paper presented at the *2004 IEEE Workshop on Machine Learning for Signal Processing*, Sao Luis, Brazil, 29 September 29–1 October 2004.

[8] Efron, B., Johnstone, I., Hastie, T. and Tibshirani, R., 2004, Least angle regression. *Annals of Statistics*, **32**, 407–499.

[9] Wessberg, J., Stambaugh, R., Kralik, J.F., Beck, P.D., Laubach, M., Chapin, J.K., Kim, J., Biggs, J., Srinivasan, M.A. and Nicolelis, M.A.L., 2000, Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature*, **408**, 361–365.

[10] Sanchez, J.C., Kim, S-.P., Erdogmus, D., Rao, Y.N., Principe, J.C., Wessberg, J. and Nicolelis, M.A.L., 2002, Input–output mapping performance of linear and nonlinear models for estimating hand trajectories from cortical neuronal firing patterns. *Proceedings of the IEEE International Workshop on Neural Networks for Signal Processing*, pp. 139–148, Valais, Switzerland.

[11] Tibshirani, R., 1996, Regression shrinkage and selection via the lasso. *Journal of Royal Statistics Society B*, **58**, 267–288.

[12] Weisberg, S., 1980, *Applied Linear Regression* (New Jersey: Wiley).

[13] Hastie, T., Tibshirani, R. and Friedman, J., 2004, *Elements of Statistical Learning: Data Mining, Inference and Prediction* (New York: Springer-Verlag).

[14] Haykin, S., 1996, *Neural Networks: A Comprehensive Foundations* (New York, NY: McMillan).

[15] Cotter, S.F., and Rao, B.D., 2002, Sparse channel estimation via matching pursuit with application to equalization. *IEEE Transactions on Communications*, **50**, 374–377.

[16] Eweda, E., 1994, Comparison of RLS, LMS, and sign algorithms for tracking randomly time-varying channels. *IEEE Transactions of Signal Processing*, **42**, 2937–2944.

[17] Sanchez, J.C., Erdogmus, D., Rao, Y.N., Principe, J.C., Nicolelis, M.A.L. and Wessberg, J., 2003, Learning the contributions of the motor, premotor, and posterior parietal cortices for hand trajectory reconstruction in a brain-machine interface. Paper presented at the *25th International Conference IEEE EMBS*, Cancun, Mexico, 17–21 September.