JOURNAL OF INTEGRAL EQUATIONS AND APPLICATIONS
Vol. , No. , YEAR
https://doi.org/jie.YEAR..PAGE

FAST MULTISCALE FUNCTIONAL ESTIMATION IN OPTIMAL EMG PLACEMENT FOR ROBOTIC PROSTHESIS CONTROLLERS

JIN REN*, GUOHUI SONG*, LUCIA TABACU*†, AND YUESHENG XU*

Abstract. Electrocardiogram (EMG) signals play a significant role in decoding muscle contraction information for robotic hand prosthesis controllers. Widely applied decoders require large amount of EMG signals sensors, resulting in complicated calculations and unsatisfactory predictions. By the biomechanical process of single degree-of-freedom human hand movements, only several EMG signals are essential for accurate predictions. Recently, a novel predictor of hand movements adopts a multistage Sequential, Adaptive Functional Estimation (SAFE) method based on historical Functional Linear Model (FLM) to select important EMG signals and provide precise projections.

However, SAFE repeatedly performs matrix-vector multiplications with a dense representation matrix of the integral operator for the FLM, which is computational expansive. Noting that with a properly chosen basis, the representation of the integral operator concentrates on a few bands of the basis, the goal of this study is to develop a *fast* Multiscale SAFE (MSAFE) method aiming at reducing computational costs while preserving (or even improving) the accuracy of the original SAFE method. Specifically, a multiscale piecewise polynomial basis is adopted to discretize the integral operator for the FLM, resulting in an *approximately sparse* representation matrix, and then the matrix is truncated to a sparse one. This approach not only accelerates computations but also improves robustness against noises. When applied to real hand movement data, MSAFE saves 85%∼90% computing time compared with SAFE, while producing better sensor selection and comparable accuracy. In a simulation study, MSAFE shows stronger stability in sensor selection and prediction accuracy against correlated noise than SAFE.

1. Introduction

This paper aims at developing a fast computing algorithm for the adaptive functional estimation method for robotic hand prosthesis controllers. Robotic hand prostheses equipped with a prosthesis controller (PC), such as DEKA arm system [1, 2], could emulate the functionality of an intact hand and assist transradial amputees (TRA) in their daily life activities. Electrodes are placed on multiple muscles of the residual limb to collect electromyogram (EMG) signals, which contain the information of muscle contraction magnitude and duration. For an able-bodied person the muscle contractions activate the tendons and the bones to produce the hand movement. For a transradial person the residual muscles can still contract and the prosthesis controller decodes the EMG signals to produce

† Corresponding author.

Supported in part by the National Science Foundation under grants DMS-1521661, DMS-1912958, DMS-1939203, and DMS-220838.

2020 *Mathematics Subject Classification.* 62P10, 62R10, 45A05, 45Q05.

Key words and phrases. Electromyography signal, robotic prosthesis controllers, functional variable selection, group LASSO, multiscale piecewise polynomial basis, sparse matrix, fast algorithm.

the hand prosthesis movements. However, for a transradial person many muscles are not accessible for external EMG sensors placement. The key challenges are deciding on the number/places for the EMG sensors and modeling the decoding of the EMG signals to movement.

A popular approach for prosthesis control is EMG pattern recognition, which usually uses redundant [3] or high-density electrodes [4] to capture sufficient neural information. Classification techniques are then employed to identify the patterns of hand/wrist motions from such abundant data [5]. It has produced promising results and also brought many challenges, including the real-time processing cost of large amount of data and the extra noises introduced by additional sensors. On the other hand, a low-dimentional PC decoder with only 4 EMG signals has been shown to accurately predict wrist/hand movement in [6]. The selection of these EMG signals is based on their prior knowledge of the important muscles in the musculoskeletal structure of able-bodied (AB) subjects. However, it would be much more challenging to select relevant EMG sensors for TRA subjects since the musculoskeletal structure might be changed due to the loss of many muscles.

A functional estimation procedure called Sequential Adaptive Functional Estimation (SAFE) has been proposed recently in [7] to select the EMG sensors and decode the EMG signals into wrist/hand movement for TRA subjects. The statistical model proposed in SAFE uses multiple functional covariates representing recent past behavior EMG signals, whose effects can vary with the recent position (flexion or extension) of fingers and wrist, to predict the velocity or acceleration of a given movement. An adaptive group Least Absolute Shrinkage and Selection Operator (group LASSO) penalty [8, 9] is employed to select the EMG sensors, then a smooth ridge regularization consisting of only the most relevant EMG sensors is used to replace the group LASSO penalty in the decoding process. It could reduce the estimation bias caused by the group LASSO penalty [7]. It selects very few relevant EMG sensors and uses them to decode the finger/wrist movement information without sacrificing prediction accuracy. The adaptive procedure promotes the accuracy performance in sensor selection and movement estimation. However, such adaptive approach involves heavy cross validations. SAFE method uses the single-scale spline basis to estimate the predictors, which generates dense coefficient matrices. Dense matrices could dramatically drag down the speed of the computation. As experiments in [7] illustrate, SAFE costs tens of hours in a personal computer to get final results on every single data set. This is a computational bottleneck for practical applications of the SAFE method.

To overcome this computational burden of the single-scale spline basis SAFE method, we propose in this paper a fast *multiscale* numerical scheme to solve the functional linear model. It has been understood [10] that representation of integral operators could be numerically sparse under the so-called *multiscale method*. Different from single-scale basis, multiscale basis extracts information of the integral operator from different scale, which naturally leads us to coefficient matrices concentrating at 0. After a proper truncation, the resulting model remains precise while having sparse representation. This sparsity could help accelerating the computational process. Also by the truncated multiscale representation, we have noise partly filtered out from input data, which also boosters the multiscale method against noise.

Specially, following the idea of multiscale methods for integral equations [11, 12, 13], here we apply the multiscale piecewise polynomial basis in discretization of the integral operators in the FLM model, to obtain the proposed Multiscale SAFE (MSAFE) method. Such a multiscale basis has vanishing

moments and shrinking supports, which results in a coefficient matrix with decaying entries. This property enables us to approximate the coefficient matrices by a sparse one and therefore accelerates the calculation. After a proper truncation of the multiscale coefficient matrices, the computational costs can be reduced significantly comparing to the single-scale spline basis SAFE method. Also, the multiscale method is more robust against noise due to the truncation. As experiments on real data reveal, MSAFE saves 85%∼90% of computational time as SAFE method, while providing even better sensor selection and prediction errors. In simulation study, we test SAFE and MSAFE methods with correlated data, where MSAFE outperforms single-scale method. Such idea could be extensively applied in other integral models with ease.

For a fair comparison, in this paper we make several compromises on the proposed multiscale method. On one hand, to solve the group LASSO model, SAFE applies the popular method introduced in [14] by solving a smoothing model instead. Such method gives fast solution estimations of group LASSO models with set of parameters, but could have significant deviation from the real global solutions. Actually the group LASSO model is a special case of non-smooth convex optimization problems, which were extensively studied in past decades, and many algorithms were developed to solve such kind of problems with solid convergence analysis. For this type of non-smooth convex optimization problems, one may refer to fixed-point proximity algorithms [15, 16], primal-dual algorithms [17, 18] and alternating direction methods of multipliers [19, 20]. On the other hand, the fast comprehensive method to solve integral equations is considered to be the multiscale collocation method [21, 22], which, in addition to multiscale basis, utilizes the multiscale collocation functionals in model discretization. Both multiscale basis and multiscale collocation functionals contribute together to an even more sparse coefficient matrix. These new concepts however will cause considerable changes to the SAFE method, and make it difficult to distinguish contributions to the final improvement among all modifications. To demonstrate the advantage of the use of the multiscale basis alone, in this paper we keep most of the original SAFE method intact, and only apply the multiscale basis to the corresponding part of the SAFE method.

To summarize, this paper contributes to SAFE method in the following aspects. We substitute the single-scale basis of the SAFE method with multiscale piecewise polynomial basis in the FLM model, which systematically generates sparse coefficient matrices after proper truncation. Also, since the widely-used `R` [23] package `gglasso` [24] for the group LASSO model is not accepting sparse matrices, we modify corresponding functions in the package for maximum acceleration and fair comparison. These together are combined to be the proposed Multiscale SAFE (MSAFE) method and lead to the final improvement to the SAFE method.

The rest of this paper is organized as follows. We describe in Section 2 the functional linear model and the original SAFE method for EMG-based hand-movement predictors. In Section 3, we introduce the fast multiscale method, MSAFE, to solve integral models for sensor selection and movement estimation in SAFE method. Section 4 contains the application of the proposed method on the real data sets studied in [7]. To address the robustness of the proposed method against correlation, Section 5 shows the results on simulated data with correlated noises. Conclusions of this study are drawn in Section 6. For simplicity of the presentation, we provide the mathematical derivations and technical proofs of the multiscale analysis in Appendix.

2. EMG-Based Predictor for Hand Movements

We briefly describe in this section the operating principles of the prosthesis controller with EMG signals, and review the cutting-edge SAFE method which models the wrist/finger movement with historical FLM involving corresponding EMG signals.

For an AB subject, the intended hand movement originates from active potentials in motor neurons in cerebral cortex. Those neural signals conduct along motor neural pathway and infuse into corresponding muscle cells, which then cause muscle contractions to accomplish the intended movement. The terminal action potential measured from the muscle fibers is defined as motor unit action potential (MUAP), which is positively correlated to the magnitude and duration of muscle contractions. The EMG sensors placed on subject's forearm could measure the sum of MUAPs across muscles, and therefore serve as effective indicators for predicting hand movements, for both AB subjects and TRAs. The predicted hand movements are fingers/wrist flexions and extensions in different arm postures. Due to passive forces triggered by muscle relaxation, [7] showed graphically that finger movement can happen when no active EMG signal is recorded. [7] also noticed that there are significant correlations among all EMG signals across the 30-seconds time window the data was collected. It is possible to have multiple active EMG signals when performing one instance of finger flexion and extension. Based on these findings, SAFE predicts finger/wrist movement based on the recent past EMG signals and current finger/wrist position.

We point out that there are two important issues in designing the prosthesis controller with EMG signals: the selection of most relevant EMG sensors and the decoding of EMG signals to wrist/finger movement. There are 20 muscles of the forearm controlling various movements of wrist and fingers of the hand [25]. For a more accurate and interpretable prosthesis controller with a rapid real-time response, we have to select the most important EMG signals and have an accurate and efficient algorithm to decode them into the wrist/finger movement. Both problems would rely on an accurate quantitative model of the relation between EMG signals and wrist/finger movement.

We will employ the flexible statistical model introduced in SAFE. In particular, the functional linear model (FLM) [26, 27, 28, 29, 30, 31], specially historical FLM [32] would be used to describe the velocity/acceleration of wrist/finger based on recent past EMG signals. We remark that historical FLM has received many successful applications in functional regression problems.

In [32] the authors use the finite element method to estimate the historical functional model. This model considers a sample of curves $y_i(t)$ that can be predicted by covariate curves $x_i(s)$ with $s \in [t-\delta, t]$ and $\delta > 0$ was estimated from the data. A speech production experiment is used to show the performance of the historical functional model. The data on different groups of muscles involved in the anatomy and physiology of speech was collected by EMG sensors. The curves $y_i(t)$ represent the accelerations of the center of the lower lip and the covariates $x_i(t)$ represent the EMG signal associated with the depressor labii inferior muscle. In [33] the authors proposed a new method of estimating the historical functional model of [32]. Their procedure combines regularization with L^1- and L^2-norm penalties of the coefficients of the neighboring basis functions. The model is then fit to a data set collected on a sample of boilermaker workers that studies the relationship between occupational particulate matter and heart rate variability.

In [34] the recent functional linear model for sparse longitudinal data is studied. The longitudinal predictor defined only in a sliding window into the recent past $[t-\delta_1, t-\delta_2]$ for $0<\delta_2<\delta_1<T$ has an effect on the longitudinal response. This model is then applied to a primary biliary liver cirrhosis longitudinal data where the relationship between serum albumin concentration and prothrombin time is investigated. Historical functional models with a large number of functional and scalar covariates as in [35] and models with factor-specific random historical effects as in [36] are estimated by a component-wise gradient boosting algorithm which is suitable for complex models. A fully Bayesian estimation approach based on the discrete wavelet-packet transformation was employed in [37].

We next briefly introduce the statistical model applied in SAFE method. Suppose we have K measured and processed EMG signals and N instances of measurement at different time $\{t_i\}_{i=1}^N$. For $1\le i\le N$ and $1\le k\le K$, we use X_{ik} to denote the k-th historical EMG signal at i-th instance. We would like to use them to predict $y_i := y(t_i)$, the response (velocity or acceleration) of the movement at time t_i along with position z_i. The historical FLM employed in SAFE is

$$\mathbb{E}[y_i|X_{i1},X_{i2},\dots,X_{iK},z_i] = \sum_{k=1}^{K}\int_{\mathscr{T}} X_{ik}(\tau)\gamma_k(\tau,z_i)d\tau, \qquad 1\le i\le N, \tag{1}$$

where $\mathscr{T} := [-\delta,0]$ with $\delta>0$ defines the length of historical time window, $X_{ik}(\tau) := X_k(t_i+\tau)$ is the historical EMG signal at time t_i and γ_k are the unknown bivariate kernels defined on $\mathscr{T}\times\mathscr{Z}$ with $\gamma_k(\cdot,z)\in L^2(\mathscr{T})$ for any $z\in\mathscr{Z}$ and $1\le k\le K$. Here $\mathscr{Z}\subset\mathbb{R}$ is the range of position.

We will rely on the above model to select the most important EMG sensors and predict the velocity/acceleration of movement. To select EMG sensors, a group LASSO regularization model [9] could be applied in Eq. (1), resulting in the following model

$$\min_{\gamma_k\in H^2}\left\{\sum_{i=1}^{N}\left\|y_i-\sum_{k=1}^{K}\int_{\mathscr{T}}X_{ik}(\tau)\gamma_k(\tau,z_i)d\tau\right\|^2 + \lambda\sum_{k=1}^{K}\sqrt{f_k\|\gamma_k\|^2+\phi_t g_k\left\|\gamma''_{k,t}\right\|^2+\phi_z h_k\left\|\gamma''_{k,z}\right\|^2}\right\}, \tag{2}$$

where $H^2 := W^{2,2}(\mathscr{T}\times\mathscr{Z})$ is the Sobolev space of all functions possessing L^2 derivatives at least order 2 on $\mathscr{T}\times\mathscr{Z}$, $\|f\|^2 := \iint_{\mathscr{T}\times\mathscr{Z}} f^2(t,z)dtdz$, $f''_t := \partial^2 f/\partial t^2$, and $f''_z := \partial^2 f/\partial z^2$ for any $f\in H^2$. The non-negative constants f_k, g_k, h_k for $1\le k\le K$ control the penalty weights, and non-negative constants $\phi_t,\phi_z\ge 0$ serve as global controllers of penalty weights on the norms of derivatives $\{\|\gamma''_{k,t}\|^2\}_{k=1}^K$ and $\{\|\gamma''_{k,z}\|^2\}_{k=1}^K$, respectively. We remark that the magnitude of estimated γ_k's in Eq. (2) measures the importance of the corresponding EMG signal. If the kernel γ_k is estimated to be 0, then the corresponding k-th EMG signal will be considered insignificant to the movement of interest.

This idea is applied to select the most important EMG sensors. In particular, we will follow the multistage procedure proposed in SAFE method. Let $\mathscr{K}^0 := \{1,2,\dots,K\}$ and we first solve Eq. (2) with $f_k=g_k=h_k=1$ for $1\le k\le K$ to get estimators $\{\hat\gamma_k^1\}_{k\in\mathscr{K}^0}$. We then define the active variable set $\mathscr{K}^1 := \{k\in\mathscr{K}^0 : \hat\gamma_k^1\neq 0\}$ and update the weights

$$f_k^1 = \left\|\hat\gamma_k^1\right\|^{-1}, \quad g_k^1 = \left\|(\hat\gamma_k^1)''_t\right\|^{-1}, \quad h_k^1 = \left\|(\hat\gamma_k^1)''_z\right\|^{-1}, \qquad \text{for all } k\in\mathscr{K}^1. \tag{3}$$

Then we find the new estimators $\{\hat{\gamma}_k^2\}_{k\in\mathscr{K}^1}$ by solving Eq. (2) with the active variable set $\mathscr{K}^1$ and the updated weights:

$$\min_{\gamma_k\in H^2, k\in\mathscr{K}^1}\left\{\sum_{i=1}^{N}\left\|y_i-\sum_{k\in\mathscr{K}^1}\int_{\mathscr{T}}X_{ik}(\tau)\gamma_k(\tau,z_i)d\tau\right\|^2 + \lambda\sum_{k\in\mathscr{K}^1}\sqrt{f_k^1\|\gamma_k\|^2+\phi_t g_k^1\left\|\gamma_{k,t}''\right\|^2+\phi_z h_k^1\left\|\gamma_{k,z}''\right\|^2}\right\}. \tag{4}$$

The new active variable set could be defined in a similar way: $\mathscr{K}^2 := \{k\in\mathscr{K}^1 : \hat{\gamma}_k^2\neq 0\}$. Suppose R stages are repeated till certain stop criteria is met, then we arrive at the selected set of EMG sensors $\mathscr{K}^R$. After the most relevant sensors $\mathscr{K}^R$ are determined, SAFE method suggests using the smooth ridge regression model to get the final estimation of the kernels. That is, we will find the estimator of γ_k for each $k\in\mathscr{K}^R$ by solving

$$\min_{\gamma_k\in H^2, k\in\mathscr{K}^R}\left\{\sum_{i=1}^{N}\left\|y_i-\sum_{k\in\mathscr{K}^R}\int_{\mathscr{T}}X_{ik}(\tau)\gamma_k(\tau,z_i)d\tau\right\|^2 + \sum_{k\in\mathscr{K}^R}\left(\phi\|\gamma_k\|^2+\phi_t\left\|\gamma_{k,t}''\right\|^2+\phi_z\left\|\gamma_{k,z}''\right\|^2\right)\right\}. \tag{5}$$

The regularization parameters $\phi,\phi_t,\phi_z\geq 0$ are usually chosen by cross-validation method over certain candidacies. We point out that since the most relevant sensors have been chosen via Eqs. (2) and (4) in the first stage, we should not apply any sparse regularization in the final stage of functional estimation. The smooth regularization in Eq. (5) could reduce the estimation bias caused by sparse penalty [38, 39].

It is direct to observe that the major computation cost of the method comes from solving Eq. (2). We need to sequentially solve the same model with different parameters f_k, g_k, h_k at each stage of the EMG signal selection. Moreover, cross validation is applied to choose the optimal regularization parameters λ,ϕ_t,ϕ_z among candidates. These above mean that we need to solve Eq. (2) with different constants repeatedly for a large number of times. It is necessary to develop a fast and efficient algorithm to numerically solve Eq. (2).

3. Multiscale SAFE Method

In this section we present the proposed fast multiscale SAFE (MSAFE) method to solve the models Eqs. (2) and (5). Specially, we will employ the multiscale basis functions introduced in [11, 21, 22], which is widely used in developing fast algorithms for solving integral equations efficiently.

We remark that Eqs. (2) and (5) are minimization problems over infinite-dimensional spaces of functions, which requires discretization to solve them numerically. The original SAFE method uses tensor products of orthogonal cubic spline bases to represent the unknown kernels γ_k's. Such full-supported single-scale basis suffers from high computational cost in two aspects. On one hand, such full-supported basis requires integral in the full domain in every calculation, which causes heavy computation when assembling coefficient matrices. On the other hand, each of the basis functions

only extracts information from different part of the kernel function under the same scale. This results in coefficients of flatten distribution and therefore leads to dense matrices, which slow down the computation at later-stages.

To overcome disadvantages of the single-scale basis, here we employ a multiscale piecewise polynomial basis to discretize the integral operators in Eqs. (2) and (5). As shown in the following sections, the multiscale basis would systematically generate sparse coefficient matrices, which therefore significantly improve the computational speed.

3.1. ***Multiscale Piecewise Polynomial Basis.*** We in this section briefly introduce the multiscale piecewise polynomial basis and its properties, then show in a simple numerical case that such basis could systematically yield sparse coefficient matrices. For simplicity of presentation, we leave the technical constructions and mathematical proofs of the multiscale piecewise polynomial basis in A.

To overcome the disadvantages of full-supported single-scale basis, the multiscale basis improves in both the speed of generating coefficient matrices and the sparsity of the resulting matrices. Multiscale piecewise polynomial basis functions, analogous to wavelets, have vanishing moment and are divided into different levels. Such basis functions are orthogonal between different levels, and have exponentially shrinking support as level increases. The shrinking support of the basis functions accelerates calculations of high level coefficients and, together with the vanishing moment and orthogonality, enables the multiscale basis to capture information of the kernel in different levels. These together result in a sparse coefficient matrix.

Specifically, we let $\mathbb{M}_n^p$ be the space of all piecewise polynomials of degree no more than p on $[0,1]$, with nodes at $\{i/2^n\}_{i=0}^{2^n}$. Such a multiscale piecewise polynomial space approaches space H^2 as level n increases, and has the property that $\mathbb{M}_n^p \subset \mathbb{M}_{n+1}^p$, which could lead to a multilevel structured basis, as briefly described below. At the first level W_0, we choose $p+1$ basis polynomials of $\mathbb{M}_0^p$. At next level W_1, we choose $p+2$ piecewise polynomials on $\mathbb{M}_1^p$ that are perpendicular to $\mathbb{M}_0^p$. For higher level $l \geq 2$, W_l could be generated by recursively scaling and shifting functions of W_{l-1}:

$$W_l = \{\mathscr{T}_i w : w \in W_{l-1}, i \in \{0,1\}\},$$

where

$$(\mathscr{T}_0 f)(x) \coloneqq f(2x), \qquad (\mathscr{T}_1 f)(x) \coloneqq f(2x-1).$$

It is direct to observe that every function in W_l is perpendicular to $\mathbb{M}_0^p$ and has a support on an interval with length no more than 2^{-l+1}.

We point out that the introduced multiscale piecewise polynomial basis could be used to design fast algorithms for solving integral equations, such as the Fredholm integral equation of the first kind

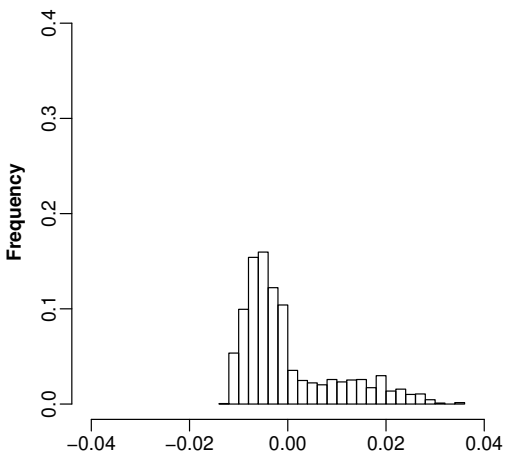
$$y(t) = \int_0^1 K(t,\tau)\gamma(\tau)d\tau, \tag{6}$$

when the unknown function γ is represented by the basis functions in W_l. We provide a brief explanation below:

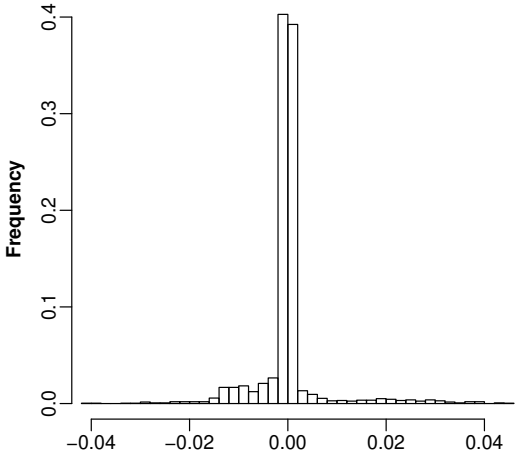
(I) The basis functions in W_l have small supports and we only need to calculate integrals on small intervals rather than $[0,1]$ when creating coefficient matrices of Eq. (6).

(II) The basis W_l for $l \geq 1$ are all perpendicular to $\mathbb{M}_0^p$, which means they have vanishing moment $p+1$. The magnitude of coefficient matrix entries will decay as l increases. In particular, if

FIGURE 1. Frequency histograms of coefficient matrix of integral equation Eq. (6) with spline basis and multiscale basis, on FC1 data set from [7]. Height of each bar indicates the frequency of values failing into the corresponding x-interval. The multiscale coefficients highly concentrate around 0, while single-scale spline coefficients do not.



(a) Frequency histogram of spline coefficients.



(b) Frequency histogram of multiscale coefficients.

kernel K in Eq. (6) is smooth enough, then entries of the corresponding coefficient matrix would have an exponential decay. Therefore it could be approximated by a sparse matrix. Proposition 2 in A provides a theoretical justification of this property.

Here we verify those properties with a numerical example. Suppose we would like to represent the unknown function γ by n level multiscale basis $\{w_j\}_{j=1}^{2^n(p+1)} := \bigcup_{l=0}^{n} W_l$ and discretize Eq. (6) at different sampling times $\{t_i\}_{i=1}^{N}$ as follows

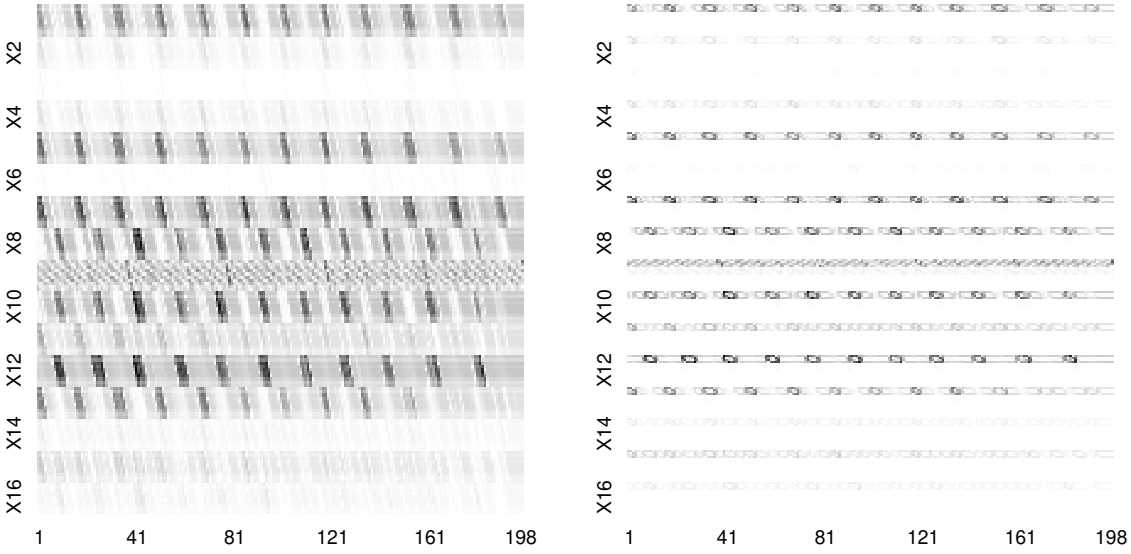
$$y(t_i) \approx \sum_{j=1}^{2^n(p+1)} c_j \int_0^1 K(t_i,\tau) w_j(\tau) d\tau, \qquad 1 \le i \le N. \tag{7}$$

We point out that the coefficient matrix

$$A_n := \left[\int_0^1 K(t_i,\tau) w_j(\tau) d\tau : 1 \le i \le N, 1 \le j \le 2^n(p+1) \right] \tag{8}$$

determines the computational cost of finding the coefficients $c := (c_j)_j$. In other words, if the coefficient matrix A_n is sparse, it would be much more efficient to find c. We consider $K(t,\tau) := X_k(t-\delta\tau)$ with $1 \le k \le 16$ and $\{t_i\}_{i=1}^{198}$ in Eq. (7), where X_k for $1 \le k \le 16$ are the real EMG signals of data set FC1 from [7]. We compare in Figures 1 and 2 the sparsity of the coefficient matrix generated by the spline basis used in [7] and the multiscale basis defined in A. It is direct to observe from Figures 1 and 2 that the coefficient matrices generated by the multiscale basis concentrates around 0, decays rapidly as the level increases and is much more sparse than those generated by the single-scale spline basis.

FIGURE 2. Magnitude plots of coefficient matrices Eq. (6) for spline basis and multiscale basis. Columns correspond to the sampling time $\{t_i\}_{i=1}^{198}$, while rows mean different basis functions. Notice that matrices $\{X_k\}_{k=1}^{16}$ are combined in row for each basis respectively. There is a notable pattern of repeating in matrix of single-scale spline basis, which therefore has no sparse structure. For multiscale basis, most information of the kernel is extracted by the first few levels of basis functions, leading to a sparse coefficient matrix.



(a) Coefficient matrix from the spline basis.

(b) Coefficient matrix from the multiscale basis.

Based on the explanation provided in ((II)) and Proposition 2 in A, we introduce the following multiscale truncation strategy for Eq. (2). Theoretically the higher level n we consider, the more precise solutions we will get, but higher level brings heavy computation. Fortunately, the following theorem implies that with the multiscale basis, relatively high levels will have insignificant effect to the coefficient matrix. This allows us to truncate the coefficient matrix, reducing computational costs while keeping coefficient matrices precise. The truncated $\tilde{A}_n^m$ of matrices A_n for level $1 \le m \le n$ is defined by

$$[\tilde{A}_n^m]_{ij} := \begin{cases} [A_n]_{ij}, & 1 \le j \le 2^m(p+1), \\ 0, & 2^m(p+1) < j \le 2^n(p+1). \end{cases} \tag{9}$$

We notice that A_m is exactly the first $2^m(p+1)$ columns of A_l for all $l \ge m$. To measure the difference caused by truncation, we denote by $\|A\|_2$ the 2-norm for matrix A. Let $C^p[0,1]$ denote the space of all functions on $[0,1]$ possessing p-order continuous derivatives for $p \in \mathbb{N}$.

Theorem 1. *If $f \in C^p[0,1]$ and $m \in \mathbb{N}$, then for $n > m$ there holds*

$$\left\|A_n - \tilde{A}_n^m\right\|_2 \le c \cdot 2^{-mp},$$

where $c > 0$ is independent of n.

Theorem 1 implies that with multiscale basis, the truncation would not cause great impact to the coefficient matrix, as long as m is sufficiently large. With such level m, we truncate the coefficient matrix A_n and work with the sparse $\tilde{A}_n^m$ in computation afterwards.

3.2. *Multiscale SAFE Method.* We thoroughly present the Multiscale SAFE method with multiscale basis to discretize Eqs. (2) and (5). It will produce sparse matrices in the discretized problems and provide a much faster way to select the sensors and estimate the kernels than SAFE method.

In MSAFE, we represent the unknown function γ_k by a basis in the Cartesian product space $\mathscr{S} := \mathbb{M}_n^p \otimes \mathbb{S}$, where $\mathbb{M}_n^p$ has a multiscale basis $\{w_j : 1 \le j \le (p+1)2^n\}$ and $\mathbb{S}$ is a cubic spline space on $[0,1]$ with basis $\{s_l : 1 \le l \le q\}$. That is, we consider

$$\gamma_k(t,z) := \sum_{j=1}^{(p+1)2^n} \sum_{l=1}^{q} b_{jlk} w_j(t) s_l(z), \qquad 1 \le k \le K.$$

For $1 \le i \le N$ and $1 \le k \le K$, define

$$[A_{ik}]_{jl} := s_l(z_i) \int_0^1 \mathscr{X}_{ik}(\tau) w_j(\tau) d\tau, \quad 1 \le j \le (p+1)2^m,\ 1 \le l \le q, \tag{10}$$

where $\mathscr{X}_{ik} := X_{ik}(t_i - \delta\cdot)$ and $m \le n$ is the selected truncation level discussed in previous paragraph. That means the matrix A_{ik} is already truncated according to strategy Eq. (9). For $1 \le k \le K$, we set $[\mathsf{A}_k]_{i,\cdot} = \mathscr{V}(A_{ik})^\mathsf{T}$ for $1 \le i \le N$ and

$$\beta_k := \mathscr{V}(\mathsf{B}_k) \quad \text{with} \quad [\mathsf{B}_k]_{jl} := b_{jlk} \quad \text{for} \quad 1 \le j \le (p+1)2^n,\ 1 \le l \le q,$$

where $\mathscr{V}$ denotes the operator that stacks the columns of a matrix into a column vector. We then discretize the FLM Eq. (2) with multiscale basis as

$$\min_{\beta_k \in \mathbb{R}^{2^n q(p+1)}, k \in \mathscr{K}} \left\{ \left\| \sum_{k=1}^{K} \mathsf{A}_k \beta_k - y \right\|^2 + \lambda \sum_{k=1}^{K} \sqrt{\beta_k^\mathsf{T} \mathsf{G}_k \beta_k} \right\}, \tag{11}$$

where $\mathsf{G}_k := f_k \mathscr{G} + \phi_t g_k \mathscr{G}_w + \phi_z h_k \mathscr{G}_s$, $\mathscr{G} := G_s \otimes G_w$, $\mathscr{G}_w := G_s \otimes D_w$ and $\mathscr{G}_s := D_s \otimes G_w$ with '$\otimes$' denoting the Kronecker product of two matrices. The gram matrices $G_w, D_w \in \mathbb{R}^{(p+1)2^n \times (p+1)2^n}$ and $G_s, D_s \in \mathbb{R}^{q \times q}$ are given by

$$[G_w]_{jj'} := (w_j, w_{j'}), \quad [D_w]_{jj'} := (w_j'', w_{j'}''), \quad [G_s]_{ll'} := (s_l, s_{l'}), \quad [D_s]_{ll'} := (s_l'', s_{l'}'').$$

Since G_k's are symmetric positive-definite, Eq. (11) could be easily reformulated as standard group LASSO model by variable substitution. We will then use the above Eq. (11) to select the most important sensors through the multistage approach described in Section 2.

Once the most important sensors $\mathscr{K}^R$ are selected after R stages, we will use the following ridge regression model to estimate the corresponding coefficient of kernels β_k, for $k \in \mathscr{K}^R$:

$$\min_{\beta_k \in \mathbb{R}^{2^n q(p+1)},\ k \in \mathscr{K}^R} \left\{ \left\| \sum_{k \in \mathscr{K}^R} \mathsf{A}_k \beta_k - y \right\|^2 + \sum_{k \in \mathscr{K}^R} \beta_k^\mathsf{T} \mathsf{G} \beta_k \right\}, \tag{12}$$

where $\mathsf{G} := \phi\mathscr{G} + \phi_t\mathscr{G}_w + \phi_z\mathscr{G}_s$ with parameters $\phi, \phi_t, \phi_z \geq 0$. To solve Eq. (12), notice that the objective function is a quadratic function, which means that Eq. (12) could be easily solved as a linear system.

4. Numerical Experiments

We will implement the proposed MSAFE method on the real data sets [7] consisting of EMG and movement data from an AB subject's right limb. We will also compare its performance with the original SAFE method proposed in [7]. Numerical results demonstrate that the proposed MSAFE method could achieve better or similar performance in sensor selection and prediction error with a significantly less computation time than SAFE. All experiments in this section are executed on `R` [23], within Windows 10 on an Intel Core i7 CPU @ 3.60 GHz and 16GB RAM.

4.1. ***Data Description and Preprocessing.*** We first briefly describe the data sets. We consider two different patterns of movements, constant and random, and each of them contains 3 different movement data of finger and wrist, respectively. These give us 12 different data sets in total. There are 15 EMG sensors placed on the subject's limb. An external EMG signal unrelated to movement is also added to address the validity of sensor selection. Therefore in each data set, we will have 16 EMG signals $\{X_k(t) : t \in T\}_{k=1}^{16}$ at 198 different sampling time $T := \{t_i\}_{i=1}^{198}$, where X_9 is the unrelated one. Moreover, the displacement of finger flexion/extension or wrist flexion/extension $\{z(t) : t \in T\}$ at the corresponding times T are also collected.

We next describe the preprocessing of the raw data of EMG signals and displacement in the convenience of numerical implementation. The displacement data $\{z_i := z(t_i)\}_{i=1}^{198}$ and historical EMG signals $S_{ik} := \{X_k(t) : t \in [t_i - \delta, t_i] \cap T\}$ with window size $\delta = 1/3$ for $1 \leq k \leq 16$ and $1 \leq i \leq 198$ are extracted from the raw data at 198 sampling time $\{t_i\}_{i=1}^{198} \subset T$. To get the corresponding movement velocity $\{z'(t_i)\}_{i=1}^{198}$, six-order spline basis with third-order regularization are used to get a fit $\hat{z}(t)$ out of data $\{z(t) : t \in T\}$, then $\{y_i := \hat{z}'(t_i)\}_{i=1}^{198}$ could be computed explicitly. As for the integral Eq. (1), MSAFE uses continuous piecewise linear functions to interpolate the discretely sampled EMG data $\{X_k(t) : t \in [t_i - \delta, t_i] \cap T\}$, and gets approximations of the continuous signals $\{\mathscr{X}_{ik}(t) : t \in [0,1]\}$ for $1 \leq k \leq 16$ and $1 \leq i \leq 198$ with explicit formulas. The integral Eq. (1) and the matrices A_k for $1 \leq k \leq 16$ in Eqs. (11) and (12) can then be approximated.

4.2. ***Experiment Setups.*** In MSAFE, we will represent the kernels $\gamma_k(t,z)$ for $1 \leq k \leq 16$ in the space $\mathscr{S} = \mathbb{M}_2^3 \otimes \mathbb{S}_{10}$, where $\mathbb{M}_2^3$ denotes the multiscale piecewise cubic polynomial space of level 2 (see more details in A) and $\mathbb{S}_{10}$ is cubic spline space with dimension 10. We point out that the resulting space $\mathscr{S}$ for MSAFE is of dimension 160, which is larger than the one of SAFE method. Actually SAFE method adopts the space $\mathbb{S}_{10} \otimes \mathbb{S}_{10}$.

We start with the sensor selection. The tuning parameters λ, ϕ_t, ϕ_z in the sensor selection Eq. (11) are set in such a way that $\log\lambda$ takes values from -20 to 0 with step 0.25 and $\log\phi_t, \log\phi_z$ ranges from -10 to 0 with step 2.5. We will use 5-fold cross validation to select them in each of the following stages. The sequentially updated parameters $\{f_k, g_k, h_k\}_{k=1}^{16}$ in Eq. (11) are initialized to be 1. We use the `R` package `gglasso` to solve Eq. (11) with the initial values at the first stage. We then get an active variable set $\mathscr{K}^1$ and update the values of $\{f_k, g_k, h_k\}_{k=1}^{16}$ according to Eq. (3). The second stage model Eq. (11) will be solved with the updated values of $\{f_k^1, g_k^1, h_k^1\}_{k=1}^{16}$ and the active variable

set $\mathscr{K}^1$. We repeat this process for 5 stages and obtain the final active variable set $\mathscr{K}^5$ with both methods.

We next continue with the kernel estimation. Specifically, we will solve Eq. (12) with the active variable set $\mathscr{K}^5$. For SAFE method, $\phi = 0$ as claimed in [7] and for MSAFE, we set candidates ϕ such that $\log_{10}\phi \in \{-1,-2,-3,-4,-5\}$ for cross validation. In both methods, candidates of ϕ_t, ϕ_z for cross validation and other setups are kept identical to the ones in sensor selection part, respectively.

We remark that the SAFE method in [7] represents the kernels γ_k in $\mathbb{S}_{10} \otimes \mathbb{S}_{10}$. We already showed in Figure 2 that the coefficient matrices generated by the multiscale basis are much more sparse than those generated by the spline basis. In other words, the coefficient matrices in the proposed MSAFE method have a majority of entries close to 0. Furthermore, Figure 2 indicates that the additional sparsity that is not modeled by truncation strategy Eq. (9), and the spline term in Eq. (10) could contribute to some extra sparsity as well. In regard of these, we further truncate those small entries in A_k's, only keep 10% of entries to be nonzero in those coefficient matrices A_k's.

During this study, the latest version of `gglasso` package does not take sparse matrices. To have a fair comparison with SAFE and fully demonstrate advantages of multiscale basis, we delve into the `Fortran` codes of the package, implement the sparse matrices multiplication (see e.g. [40]) and incorporate it into functions from the package `gglasso`.

4.3. ***Experiment Results.*** We will compare the performance of the proposed MSAFE method with original SAFE in the following aspects: the selected sensors, the prediction error with the estimated kernels, and the total computational time in sensor selection and kernel estimation. More precisely, the prediction error for a specified method and data set is defined by

$$\mathrm{MSE} = \frac{1}{5}\sum_{i=1}^{5}\sum_{j\in F_i}\frac{(\hat{y}_j - y_j)^2}{|F_i|},$$

where F_i identifies the i-th test fold of the 5-fold cross validation, $\{y_j\}_{j\in F_i}$ are the actual responses for testing, and $\{\hat{y}_j\}_{j\in F_i}$ are the predicted values of the kernels estimated on corresponding training folds.

Anatomy of hand movements provides us the most important sensors' placement related to the finger/wrist flexion/extension movements. Let $\mathscr{K} := \mathscr{K}_F \cup \mathscr{K}_E$ denote the index set of those most relevant sensors to the movement of interest, where $\mathscr{K}_F$ and $\mathscr{K}_E$ split $\mathscr{K}$ into groups corresponding to flexion and extension. It was claimed in [7] that for finger movement, $\mathscr{K}_F = \{12\}$ and $\mathscr{K}_E = \{5,7\}$; for wrist movement, $\mathscr{K}_F = \{8,10,11,14\}$ and $\mathscr{K}_E = \{2,7,13,15\}$. Notice that sensors $\{1,3,4,6,9,16\}$ are irrelevant to the movements of interest.

Table 1 shows the performance of SAFE and MSAFE in sensor selection, cross validation mean square error, and the computational time for each data set. It is direct to observe that the proposed MSAFE method selects the same sensors as SAFE in most of the data sets. For finger movement data sets, both methods select exactly the same sensors. For wrist movement data sets, MSAFE method tends to select fewer but more important sensors. For example, for Constant #3 data of wrist movement, sensor 9 is incorrectly chosen by SAFE and is successfully filtered by MSAFE.

Moreover, the cross validation mean square errors of MSAFE are almost the same as those of SAFE in every data set. However, as Figure 3 reveals, the overall computational time of MSAFE is

TABLE 1. Performance metrics for sensor selection at the final stage for constant (top three rows) and random (bottom three rows) finger and wrist movement patterns, and the CV MSE means and time costs for each of the data sets.

Data set	Method	Finger Movement			Wrist Movement		
		Selected Sensor	CV MSE	Time (min)	Selected Sensor	CV MSE	Time (min)
Const #1	SAFE	7, 12	7.794e−2	504.77	2, 11, 15	3.931e−2	527.56
	MSAFE	7, 12	7.161e−2	47.15	8, 15	4.389e−2	68.94
Const #2	SAFE	7, 12	9.937e−2	525.92	11, 15	4.807e−2	734.18
	MSAFE	7, 12	8.298e−2	43.72	11, 15	4.857e−2	88.16
Const #3	SAFE	7, 12	1.012e−1	669.36	2, 9, 11, 15	5.066e−2	886.07
	MSAFE	7, 12	9.279e−2	76.38	11, 15	6.015e−2	113.03
Rand #1	SAFE	7, 12	2.021e−1	358.54	8, 11, 15	1.136e−1	699.47
	MSAFE	7, 12	2.048e−1	54.80	8, 15	1.181e−1	120.06
Rand #2	SAFE	5, 7, 12	1.725e−1	593.23	2, 8, 11, 15	1.095e−1	1345.79
	MSAFE	5, 7, 12	1.434e−1	103.17	11, 15	1.204e−1	201.87
Rand #3	SAFE	7, 12	1.749e−1	725.37	11, 15	8.222e−2	615.27
	MSAFE	7, 12	1.749e−1	122.47	11, 15	8.473e−2	65.81

remarkably less than that of SAFE in every data set; MSAFE costs only about 10%∼15% time of that in the SAFE method. These results confirm that the multiscale polynomial basis used in MSAFE have brought a huge advantage in computational cost, while maintaining the prediction accuracy.

5. Simulation Study

This section tests the robustness of SAFE and MSAFE in sensor selection against the impact of covariance misspecification, based on simulated data with correlated noises studied in [7]. We use the data set FC3 and the corresponding estimated kernels $\hat{\gamma}_7$ and $\hat{\gamma}_{12}$ from SAFE method to generate the responses. Specially, we generate data by

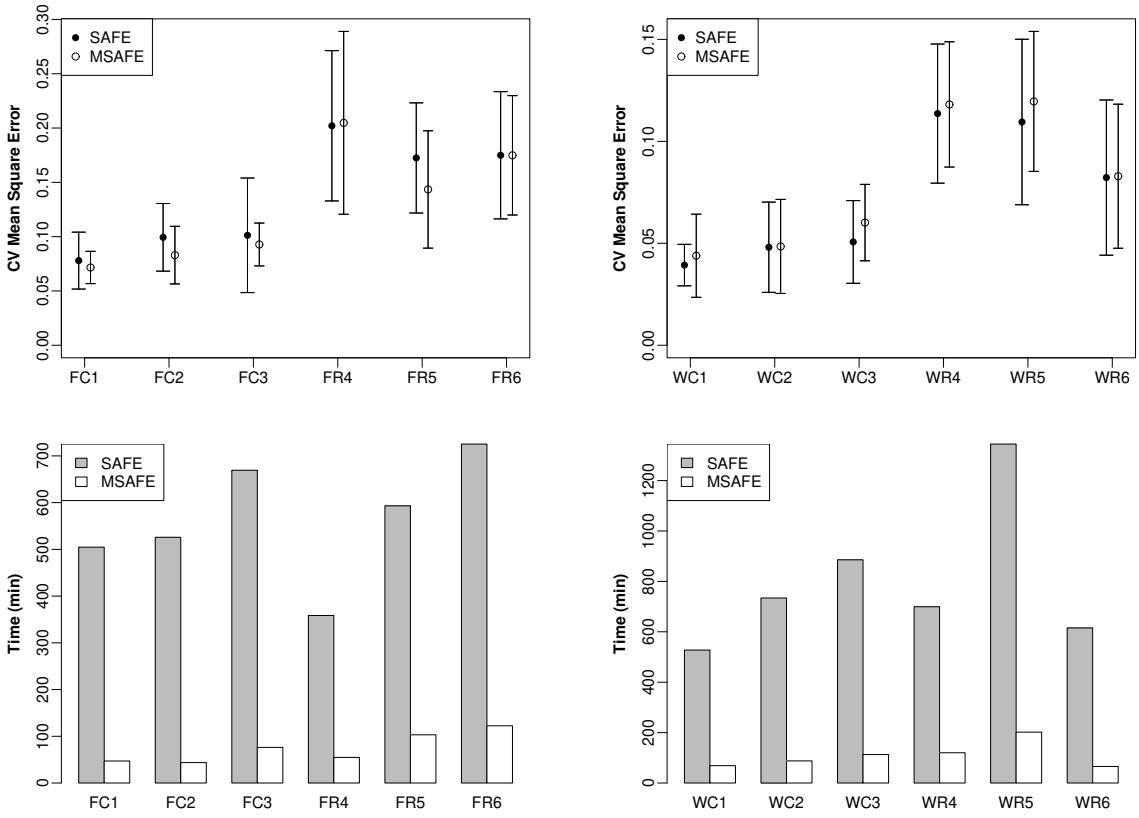
$$y_i = \sum_{k=7,12} \int_0^1 X_k(t_i - \delta\tau)\hat{\gamma}_k(\tau, z_i)d\tau + \varepsilon_i, \qquad 1 \le i \le N, \tag{13}$$

where X_k's and t_i's are from data set FC3 in [7], $\hat{\gamma}_7$ and $\hat{\gamma}_{12}$ are estimated kernels of FC3 by SAFE method, $\{\varepsilon_i\}_{i=1}^N$ are zero-mean multivariate Gaussian noises with covariance matrix $\Sigma \in \mathbb{R}^{N\times N}$ such that

$$\mathrm{Cov}[\varepsilon_i, \varepsilon_j] = \sigma_h^2\left[\delta_{ij} + \theta\exp\left(-(i-j)^2/\eta^2\right)\right].$$

Here $\theta > 0$ is related to the dominant sources of dependence, $\eta > 0$ controls the correlation decay where larger values imply slower correlation decay, and $\sigma_h > 0$ is chosen such that $\Sigma_{ii} = \sigma_h^2(1+\theta)$ equals the standard deviations of CV MSE means from the SAFE method on FC3 data set. The simulation experiment runs over factors $\theta \in \{0.25, 10, 100\}$ and $\eta \in \{10, 100\}$, and $J = 100$ individual sets of noises on each scenario. We use the same setting as [7] and will compare the performance of MSAFE with SAFE. We set the number of stages $R = 2$ for both methods.

FIGURE 3. CV MSE means and standard deviations for the optimal tuning parameters of the last selection stage (top panel) and the time cost (bottom panel). MSAFE has comparable prediction error than SAFE, while time plots illustrate the advanced efficiency brought by multiscale basis.



For the generated J different sets of noises, we will report the following quantities in Table 2.

- **Mean Size**: $\sum_{j=1}^{J}|\mathscr{K}_j^2|/J$, where $\mathscr{K}_j^2$ is the set of selected sensors after 2 stages for each $1 \le j \le J$. We note that the ideal value is 2 with $\{7,12\}$ as the ground-truth sensors for all data sets.
- **Mean False Positive**: $\sum_{j=1}^{J}|\mathscr{K}_j^2 \setminus \{5,7,12\}|/J$. The ideal value is 0.
- **Mean CV MSE**: $\sum_{j=1}^{J}\text{MSE}_j/J$, where MSE_j is the CV MSE for j-th data set.
- **Mean Time**: $\sum_{j=1}^{J}T_j/J$, where T_j is the time cost for j-th data set.

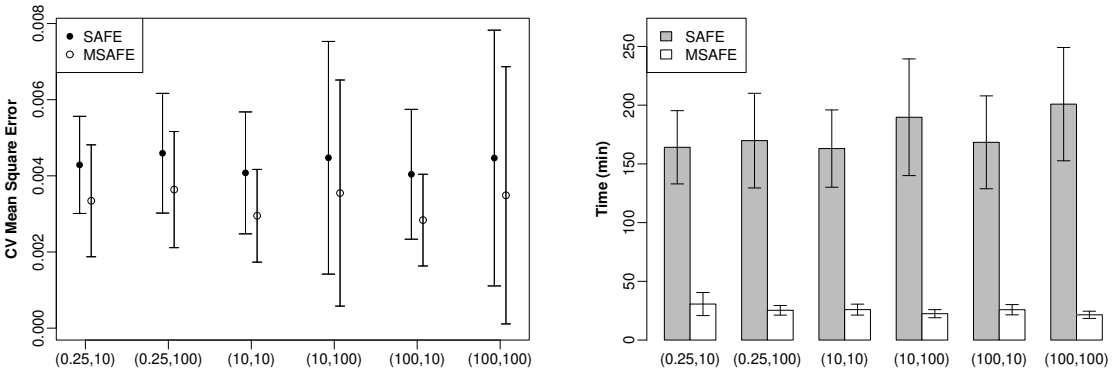
We also display in Figure 4 the mean and standard deviation of the CV MSE of both SAFE and MSAFE methods with each scenario of θ and η.

We observe that both methods select the correct sensors 7 and 12 on all data sets. However in every setting of θ and η, MSAFE selects fewer sensors and less incorrect selections than SAFE. MSAFE method merely selects 45%∼65% extra sensors as SAFE does; specially MSAFE selects only 8%∼23%

TABLE 2. Performance metrics across 100 data sets for various covariance settings with simulation model Eq. (13).

θ	η	Mean Size		Mean False Positive		Mean CV MSE		Mean Time (min)	
		SAFE	MSAFE	SAFE	MSAFE	SAFE	MSAFE	SAFE	MSAFE
0.25	10	3.61	2.36	1.59	0.36	4.287e−3	3.344e−3	164.14	30.66
	100	3.90	2.42	1.87	0.42	4.593e−3	3.639e−3	169.82	25.34
10	10	3.60	2.13	1.55	0.13	4.078e−3	2.951e−3	163.06	25.93
	100	4.86	2.38	2.72	0.38	4.474e−3	3.548e−3	189.71	22.46
100	10	3.76	2.20	1.68	0.20	4.040e−3	2.837e−3	168.38	25.85
	100	5.19	2.39	2.98	0.39	4.467e−3	3.489e−3	200.89	21.50

FIGURE 4. Plots of CV MSE and time of SAFE and MSAFE methods across 100 data sets for various covariance settings with simulation model Eq. (13). The x-axis labels denote the pair (θ, η).



(a) CV MSE on simulated data.

(b) Time on simulated data.

misspecified sensors than SAFE does. Moreover, MSAFE tends to have less prediction error in every scenario; specially, MSAFE has 70%∼80% mean square error as the one of SAFE. Overall, MSAFE method is more robust against the covariance misspecification than SAFE. Finally, the computational time of MSAFE on those simulation data sets is always about 10%∼18% of that of the SAFE method, which once again corroborates the stability and speed advantage of multiscale piecewise polynomial basis.

6. Conclusions

To perform fast and precise algorithm for robotic prosthesis controllers, we propose MSAFE method based on SAFE, with the multiscale piecewise polynomial basis to discretize the integral operator in FLM. Multiscale basis systematically generates sparse coefficient matrices, accelerating the calculation and improving the stability of original SAFE method. Compared to single-scale spline basis SAFE method, MSAFE with multiscale basis costs only 10%∼15% computational time on the hand movement

data, while performing better sensor selection and comparable prediction accuracy. We also test the robustness of sensor selection for multi- and single-scale basis against correlation noise. Experiments on simulated data shows that with various patterns of correlated noise, MSAFE always has slighter misspecification and prediction error than SAFE. Both real-data experiments and simulation studies corroborate the efficiency and stability of the proposed MSAFE method.

Acknowledgement

The authors are grateful to Professors Ana-Maria Staicu and Jonathan Stallrich of North Carolina State University for their generosity in providing the hand/wrist movement data and fruitful discussions.

Appendix A. Multiscale Piecewise Polynomial Basis

In this appendix, we present the multiscale piecewise cubic polynomial basis on $\Omega := [0,1]$ and its properties. For $n \geq 0$, we denote by $\mathbb{M}_n^p$ the linear space of all piecewise polynomials of degree less than or equal to p, supported on Ω with nodes $\{i/2^n\}_{i=0}^{2^n}$. Abbreviation $\mathbb{M}_n$ is used when the degree p is clear from the context. By definition, we have the nestedness $\mathbb{M}_n \subset \mathbb{M}_{n+1}$. This enables us to define wavelet subspace $\mathbb{W}_{n+1} \subset \mathbb{M}_{n+1}$ such that $\mathbb{W}_{n+1} \perp \mathbb{M}_n$ in $L^2(\Omega)$ sense, and $\mathbb{M}_{n+1} = \mathbb{M}_n \oplus^\perp \mathbb{W}_{n+1}$, where '$\oplus^\perp$' denotes the direct sum of two perpendicular spaces. With these spaces, the multiscale decomposition of function space $\mathbb{M}_n$ could be represented by

$$\mathbb{M}_n = \mathbb{M}_0 \oplus^\perp \mathbb{W}_1 \oplus^\perp \mathbb{W}_2 \oplus^\perp \cdots \oplus^\perp \mathbb{W}_n. \tag{14}$$

Such a decomposition has a spectacular property that $\mathbb{W}_n$ can be constructed based on $\mathbb{W}_1$. To see this, define $\Phi := \{\phi_0, \phi_1\}$ where $\phi_0 := \cdot/2$, $\phi_1 := (1+\cdot)/2$, and transformations of functions $f \in L^\infty(\Omega)$ such that

$$\mathscr{T}_0 f := f \circ \phi_0^{-1}, \qquad \mathscr{T}_1 f := f \circ \phi_1^{-1}.$$

One can learn from [11] that

$$\mathbb{W}_{n+1} = \mathscr{T}_0 \mathbb{W}_n \oplus^\perp \mathscr{T}_1 \mathbb{W}_n, \qquad \text{for all } n \in \mathbb{N}.$$

Moreover, if W_n is a basis of $\mathbb{W}_n$, then

$$W_{n+1} := \{\mathscr{T}_i w : w \in W_n, i \in \{0,1\}\}$$

is a basis of $\mathbb{W}_{n+1}$. This result reveals the relation between wavelet space $\mathbb{W}_n$ and $\mathbb{W}_1$, and a systematic way to generate basis W_n of $\mathbb{W}_n$ out from basis W_1 of $\mathbb{W}_1$.

$$\mathscr{T}_e := \mathscr{T}_{e_1} \circ \mathscr{T}_{e_2} \circ \cdots \circ \mathscr{T}_{e_n}.$$

Then for $n > 1$ we have that

$$\mathbb{W}_n = \bigcup_{e \in \{0,1\}^{n-1}} \mathscr{T}_e \mathbb{W}_1, \qquad W_n = \bigcup_{e \in \{0,1\}^{n-1}} \mathscr{T}_e W_1, \tag{15}$$

and we would have W_n as a basis of $\mathbb{W}_n$.

The relation above not only serves as a systematical generator of basis functions for high levels, but also leads us to the estimation of coefficient matrix entries. The proposition following claims that as level n increases, the entries of coefficient matrix A in Eq. (8) will decay exponentially. Define

$C^p[0,1]$ as the space of all functions on $[0,1]$ possessing p-order continuous derivatives for $p \in \mathbb{N}$, and for $f \in C[0,1]$, define $\|f\|_\infty := \max_{t\in[0,1]} |f(t)|$.

Proposition 2. *Suppose that $\{W_n\}_{n=1}^\infty$ is a sequence of multiscale piecewise polynomial bases of degree $p \geq 0$ generated by Eq. (15). If $f \in C^p[0,1]$, then for any $w \in W_n$ with $n \geq 1$, there holds*

$$\left| \int_0^1 f(\tau) w(\tau) d\tau \right| \leq c_p \cdot 2^{-(p+1)n} \left\| f^{(p)} \right\|_\infty, \tag{16}$$

with positive constant

$$c_p := \frac{\sqrt{2p+3}}{(p+1)!2^{p+1}} \max_{v\in W_1} \|v\|_{L^2[0,1]}.$$

Proof. By Eq. (15), for any $w \in W_{n+1}$ with $n \geq 0$ we have $w = v(2^n \cdot -i)$ for some $v \in W_1$ and $0 \leq i \leq 2^n - 1$. Then

$$\int_0^1 f(\tau) w(\tau) d\tau = \int_0^1 f(\tau) v(2^n\tau - i) d\tau = 2^{-n} \int_0^1 f\big(2^{-n}(\tau+i)\big) v(\tau) d\tau.$$

On the other hand, notice that by Taylor expansion, we have

$$f\big(2^{-n}(\tau+i)\big) = \sum_{l=0}^{p-1} \frac{f^{(l)}(i2^{-n})}{l!2^{nl}} \tau^l + \int_0^\tau \frac{t^p}{p!2^{pn}} f^{(p)}\big(2^{-n}(t+i)\big) dt.$$

Therefore we have

$$\begin{aligned}
\left| \int_0^1 f(\tau) w(\tau) d\tau \right| &= \frac{1}{p!2^{(p+1)n}} \left| \int_0^1 \int_0^\tau t^p f^{(p)}\big(2^{-n}(t+i)\big) v(\tau) dt d\tau \right| \\
&\leq \frac{\left\| f^{(p)} \right\|_\infty}{p!2^{(p+1)n}} \left| \int_0^1 \int_0^\tau t^p v(\tau) dt d\tau \right| \\
&= \frac{\left\| f^{(p)} \right\|_\infty}{(p+1)!2^{(p+1)n}} \left| \int_0^1 \tau^{p+1} v(\tau) d\tau \right|.
\end{aligned}$$

Next we estimate the last integral. Notice the fact that monic Legendre polynomials L_{p+1} of degree $p+1$ has the smallest L^2-norm among monic polynomials with the same degree on $[-1,1]$. Then since $v \in W_1$ has vanishing moment $p+1$, we have

$$\begin{aligned}
\left| \int_0^1 \tau^{p+1} v(\tau) d\tau \right| &= \frac{1}{2^{p+2}} \left| \int_{-1}^1 (t+1)^{p+1} v\left(\frac{t+1}{2}\right) dt \right| \\
&= \frac{1}{2^{p+2}} \left| \int_{-1}^1 L_{p+1}(t) v\left(\frac{t+1}{2}\right) dt \right|.
\end{aligned}$$

Then by Cauchy inequality,

$$\begin{aligned}\left|\int_0^1 \tau^{p+1}v(\tau)d\tau\right| &\le \frac{\sqrt{2}}{2^{p+2}}\left\|L_{p+1}\right\|_{L^2[-1,1]}\|v\|_{L^2[0,1]}\\ &= \frac{\sqrt{2}}{2^{p+2}}\frac{2^{p+1}((p+1)!)^2}{(2p+2)!}\sqrt{\frac{2}{2p+3}}\|v\|_{L^2[0,1]}\\ &= \frac{\|v\|_{L^2[0,1]}}{\binom{2p+2}{p+1}\sqrt{2p+3}} \le \frac{\sqrt{2p+3}}{2^{2(p+1)}}\|v\|_{L^2[0,1]},\end{aligned}$$

where the L^2-norm of L_{p+1} and the estimation of central binomial coefficient $\binom{2n}{n} \ge 4^n/(2n+1)$ are applied. Then combining the estimations above gives the final bound

$$\left|\int_0^1 f(\tau)w(\tau)d\tau\right| \le \frac{\sqrt{2p+3}}{(p+1)!2^{(p+1)(n+2)}}\left\|f^{(p)}\right\|_\infty \max_{v\in W_1}\|v\|_{L^2[0,1]},$$

which is exactly the desired. □

Based on Proposition 2 now we are capable to prove Theorem 1, which could be used to determine the multiscale level needed for a certain accuracy level. Recall that the multiscale coefficient matrix for n level is A_n, and the truncated $\tilde{A}_n^m$ of matrices A_n for level $1 \le m \le n$ is

$$\left[\tilde{A}_n^m\right]_{ij} := \begin{cases}[A_n]_{ij}, & 1 \le j \le 2^m(p+1),\\ 0, & 2^m(p+1) < j \le 2^n(p+1).\end{cases}$$

Notice that for $n \ge m$, A_m is exactly the first $2^m(p+1)$ columns of $\tilde{A}_n^m$. For matrix A, define $\|A\|_2$ and $\|A\|_F$ as the matrix 2- and Frobenius-norm of A respectively.

Proof of Theorem 1. By Proposition 2, for all $n > m$ we have

$$\begin{aligned}\left\|A_n - \tilde{A}_n^m\right\|_F^2 &\le c_p^2\left\|f^{(p)}\right\|_\infty^2 (p+1)N\sum_{i=m+1}^{n} 2^{i-1}2^{-2(p+1)i}\\ &\le c_p^2\left\|f^{(p)}\right\|_\infty^2 (p+1)N\sum_{i=m+1}^{\infty} 2^{i-1}2^{-2(p+1)i}\\ &= c_p^2\left\|f^{(p)}\right\|_\infty^2 \frac{N(p+1)}{(2^{2p+2}-2)2^m}2^{-2pm}.\end{aligned}$$

Then notice for matrix A there holds $\|A\|_2 \le \|A\|_F$, we have the desired inequality. Worthy to notice that with large m, another inequality $\|A\|_2 \le \sqrt{\|A\|_1\|A\|_\infty}$ yields a better bound. □

Now we focus on constructing multiscale basis of $\mathbb{M}_2^3$. Inspired by [13, 21], here we define points $T_0 := \{t_i := (i+1)/5 : i = 0,1,2,3\}$, which satisfies $T_0 \subset \Phi(T_0)$. Then we construct the basis of $\mathbb{M}_0^3$ and $\mathbb{W}_1^3$ as follows. Define $W_0 := \{w_{0i}\}_{i=0}^3 \subset \mathbb{M}_0^3$ such that $w_{0i}(t_j) = \delta_{ij}$ for $i,j = 0,1,2,3$, which

results in

$$
\begin{aligned}
w_{00}(x) &= -\frac{125}{6}x^3 + \frac{75}{2}x^2 - \frac{65}{3}x + 4,\\
w_{01}(x) &= \frac{125}{2}x^3 - 100x^2 + \frac{95}{2}x - 6,\\
w_{02}(x) &= -\frac{125}{2}x^3 + \frac{175}{2}x^2 - 35x + 4,\\
w_{03}(x) &= \frac{125}{6}x^3 - 25x^2 + \frac{55}{6}x - 1.
\end{aligned}
$$

Clear that W_0 forms a basis of $\mathbb{M}_0^3$. For wavelet space $\mathbb{W}_1^3$, we require the basis $W_1 := \{w_{1i} : i = 0,1,2,3\} \subset \mathbb{M}_1^3$ consisting of functions with vanishing moment 4, that is, $(w_{1i}, w_{0j}) = 0$ for $i, j = 0,1,2,3$. One possible basis could be

$$
\begin{aligned}
w_{10}(x) &= \begin{cases} \frac{1}{48}\left(-920x^3 + 1080x^2 - 320x + 19\right), & 0 \le x < \frac{1}{2},\\ \frac{1}{48}\left(7080x^3 - 15720x^2 + 11360x - 2669\right), & \frac{1}{2} \le x \le 1, \end{cases}\\
w_{11}(x) &= \begin{cases} \frac{1}{48}\left(-23480x^3 + 15720x^2 - 2700x + 91\right), & 0 \le x < \frac{1}{2},\\ \frac{1}{48}\left(520x^3 - 1080x^2 + 660x - 101\right), & \frac{1}{2} \le x \le 1, \end{cases}\\
w_{12}(x) &= \begin{cases} \frac{1}{48}\left(-520x^3 + 480x^2 - 60x - 1\right), & 0 \le x < \frac{1}{2},\\ \frac{1}{48}\left(23480x^3 - 54720x^2 + 41700x - 10369\right), & \frac{1}{2} \le x \le 1, \end{cases}\\
w_{13}(x) &= \begin{cases} \frac{1}{48}\left(-7080x^3 + 5520x^2 - 1160x + 51\right), & 0 \le x < \frac{1}{2},\\ \frac{1}{48}\left(920x^3 - 1680x^2 + 920x - 141\right), & \frac{1}{2} \le x \le 1, \end{cases}
\end{aligned}
$$

therefore we have W_1 as a basis of $\mathbb{W}_1^3$. Then basis W_n for level $n \ge 2$ could be obtained via relation Eq. (15).

References

[1] L. Resnik, Development and testing of new upper-limb prosthetic devices: Research designs for usability testing, Journal of Rehabilitation Research & Development 48 (6) (2011).

[2] L. Resnik, K. Etter, S. L. Klinger, C. Kambe, Using virtual reality environment to facilitate training with advanced upper-limb prosthesis, Journal of Rehabilitation Research & Development 48 (6) (2011).

[3] H. Huang, F. Zhang, Y. L. Sun, H. He, Design of a robust EMG sensing interface for pattern classification, J Neural Eng 7 (5) (2010) 056005. doi:10.1088/1741-2560/7/5/056005.

[4] L. Resnik, H. H. Huang, A. Winslow, D. L. Crouch, F. Zhang, N. Wolk, Evaluation of EMG pattern recognition for upper limb prosthesis control: A case study in comparison with direct myoelectric control, Journal of neuroengineering and rehabilitation 15 (1) (2018) 1–13.

[5] E. Scheme, K. Englehart, Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use, Journal of Rehabilitation Research & Development 48 (6) (2011).

[6] D. L. Crouch, H. Huang, Lumped-parameter electromyogram-driven musculoskeletal hand model: A potential platform for real-time prosthesis control, J Biomech 49 (16) (2016) 3901–3907. doi:10.1016/j.jbiomech.2016.10.035.

[7] J. Stallrich, M. N. Islam, A.-M. Staicu, D. Crouch, L. Pan, H. Huang, Optimal EMG placement for a robotic prosthesis controller with sequential, adaptive functional estimation (SAFE), The Annals of Applied Statistics 14 (3) (2020) 1164–1181. doi:10.1214/20-AOAS1324.

[8] R. Tibshirani, Regression shrinkage and selection via the lasso, J. Roy. Statist. Soc. Ser. B 58 (1) (1996) 267–288. URL http://links.jstor.org/sici?sici=0035-9246(1996)58:1<267:RSASVT>2.0.CO;2-G&origin=MSN

[9] M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables, J. R. Stat. Soc. Ser. B Stat. Methodol. 68 (1) (2006) 49–67. doi:10.1111/j.1467-9868.2005.00532.x.
URL https://doi.org/10.1111/j.1467-9868.2005.00532.x

[10] G. Beylkin, R. Coifman, V. Rokhlin, Fast wavelet transforms and numerical algorithms. I, Comm. Pure Appl. Math. 44 (2) (1991) 141–183. doi:10.1002/cpa.3160440202.
URL https://doi.org/10.1002/cpa.3160440202

[11] C. A. Micchelli, Y. Xu, Using the matrix refinement equation for the construction of wavelets on invariant sets, Appl. Comput. Harmon. Anal. 1 (4) (1994) 391–401. doi:10.1006/acha.1994.1024.
URL https://doi.org/10.1006/acha.1994.1024

[12] C. A. Micchelli, Y. Xu, Reconstruction and decomposition algorithms for biorthogonal multiwavelets, Multidimens. Systems Signal Process. 8 (1-2) (1997) 31–69. doi:10.1023/A:1008264805830.
URL https://doi.org/10.1023/A:1008264805830

[13] Z. Chen, C. A. Micchelli, Y. Xu, Fast collocation methods for second kind integral equations, SIAM J. Numer. Anal. 40 (1) (2002) 344–375. doi:10.1137/S0036142901389372.
URL https://doi.org/10.1137/S0036142901389372

[14] Y. Yang, H. Zou, A fast unified algorithm for solving group-lasso penalize learning problems, Stat. Comput. 25 (6) (2015) 1129–1141. doi:10.1007/s11222-014-9498-5.
URL https://doi.org/10.1007/s11222-014-9498-5

[15] C. A. Micchelli, L. Shen, Y. Xu, X. Zeng, Proximity algorithms for the L1/TV image denoising model, Advances in Computational Mathematics 38 (2) (2013) 401–426. doi:10.1007/s10444-011-9243-y.

[16] Q. Li, L. Shen, Y. Xu, N. Zhang, Multi-step fixed-point proximity algorithms for solving a class of optimization problems arising from image processing, Advances in Computational Mathematics 41 (2) (2015) 387–422. doi:10.1007/s10444-014-9363-2.

[17] E. Esser, X. Zhang, T. F. Chan, A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science, SIAM J. Imaging Sci. 3 (4) (2010) 1015–1046. doi:10.1137/09076934X.

[18] A. Chambolle, T. Pock, A first-order primal-dual algorithm for convex problems with applications to imaging, Journal of Mathematical Imaging and Vision 40 (1) (2011) 120–145. doi:10.1007/s10851-010-0251-1.

[19] E. Esser, Applications of lagrangian-based alternating direction methods and connections to split bregman, CAM Report 9 (2009) 31.

[20] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends in Machine Learning 3 (1) (2011) 1–122.

[21] Z. Chen, C. A. Micchelli, Y. Xu, A construction of interpolating wavelets on invariant sets, Math. Comp. 68 (228) (1999) 1569–1587. doi:10.1090/S0025-5718-99-01110-2.
URL https://doi.org/10.1090/S0025-5718-99-01110-2

[22] Z. Chen, C. A. Micchelli, Y. Xu, Multiscale methods for Fredholm integral equations, Vol. 28 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2015. doi:10.1017/CBO9781316216637.
URL https://doi.org/10.1017/CBO9781316216637

[23] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria (2021).

[24] Y. Yang, H. Zou, S. Bhatnagar, Gglasso: Group Lasso Penalized Learning Using a Unified BMD Algorithm (2020).

[25] B. Mitchell, L. Whited, Anatomy, Shoulder and Upper Limb, Forearm Muscles, StatPearls, 2019.

[26] H. Cardot, F. Ferraty, P. Sarda, Functional linear model, Statist. Probab. Lett. 45 (1) (1999) 11–22. doi:10.1016/S0167-7152(99)00036-X.
URL https://doi.org/10.1016/S0167-7152(99)00036-X
[27] H. Cardot, F. Ferraty, P. Sarda, Spline estimators for the functional linear model, Statist. Sinica 13 (3) (2003) 571–591.
[28] F. Ferraty, W. González-Manteiga, A. Martínez-Calvo, P. Vieu, Presmoothing in functional linear regression, Statist. Sinica 22 (1) (2012) 69–94. doi:10.5705/ss.2010.085.
URL https://doi.org/10.5705/ss.2010.085
[29] J. Goldsmith, C. M. Crainiceanu, B. S. Caffo, D. S. Reich, Penalized functional regression analysis of white-matter tract profiles in multiple sclerosis, NeuroImage 57 (2) (2011) 431–439.
[30] M. W. McLean, G. Hooker, A.-M. Staicu, F. Scheipl, D. Ruppert, Functional generalized additive models, J. Comput. Graph. Statist. 23 (1) (2014) 249–269. doi:10.1080/10618600.2012.729985.
URL https://doi.org/10.1080/10618600.2012.729985
[31] Y. Wu, J. Fan, H.-G. Müller, Varying-coefficient functional linear regression, Bernoulli 16 (3) (2010) 730–758. doi:10.3150/09-BEJ231.
URL https://doi.org/10.3150/09-BEJ231
[32] N. Malfait, J. O. Ramsay, The historical functional linear model, Canad. J. Statist. 31 (2) (2003) 115–128. doi:10.2307/3316063.
URL https://doi.org/10.2307/3316063
[33] J. Harezlak, B. A. Coull, N. M. Laird, S. R. Magari, D. C. Christiani, Penalized solutions to functional regression problems, Comput. Statist. Data Anal. 51 (10) (2007) 4911–4925. doi:10.1016/j.csda.2006.09.034.
URL https://doi.org/10.1016/j.csda.2006.09.034
[34] K. Kim, D. Şentürk, R. Li, Recent history functional linear models for sparse longitudinal data, J. Statist. Plann. Inference 141 (4) (2011) 1554–1566. doi:10.1016/j.jspi.2010.11.003.
URL https://doi.org/10.1016/j.jspi.2010.11.003
[35] S. Brockhaus, M. Melcher, F. Leisch, S. Greven, Boosting flexible functional regression models with a high number of functional historical effects, Stat. Comput. 27 (4) (2017) 913–926. doi:10.1007/s11222-016-9662-1.
URL https://doi.org/10.1007/s11222-016-9662-1
[36] D. Rügamer, S. Brockhaus, K. Gentsch, K. Scherer, S. Greven, Boosting factor-specific functional historical models for the detection of synchronization in bioelectrical signals, J. R. Stat. Soc. Ser. C. Appl. Stat. 67 (3) (2018) 621–642. doi:10.1111/rssc.12241.
URL https://doi.org/10.1111/rssc.12241
[37] M. J. Meyer, E. J. Malloy, B. A. Coull, Bayesian wavelet-packet historical functional linear models, Stat. Comput. 31 (2) (2021) Paper No. 14, 13. doi:10.1007/s11222-020-09981-3.
URL https://doi.org/10.1007/s11222-020-09981-3
[38] H. Leeb, B. M. Pötscher, K. Ewald, On Various Confidence Intervals Post-Model-Selection, Statistical Science 30 (2) (2015) 216–227. doi:10.1214/14-STS507.
[39] S. Zhao, D. Witten, A. Shojaie, In Defense of the Indefensible: A Very Naïve Approach to High-Dimensional Inference, Statist. Sci. 36 (4) (2021) 562–577. doi:10.1214/20-sts815.
URL https://doi.org/10.1214/20-sts815
[40] Y. Saad, SPARSKIT: A Basic Tool Kit for Sparse Matrix Computations (1990).

Department of Mathematics & Statistics, Old Dominion University, Old Dominion University, USA
Email address: jren@odu.edu

Department of Mathematics & Statistics, Old Dominion University, Old Dominion University, USA
Email address: gsong@odu.edu

Department of Mathematics & Statistics, Old Dominion University, Old Dominion University, USA
Email address: ltabacu@odu.edu

Department of Mathematics & Statistics, Old Dominion University, Old Dominion University, USA
Email address: y1xu@odu.edu