Power System Sensitivity Matrix Estimation by Multivariable Least Squares Considering Mitigating Data Saturation

Yingqi Liang
Department of Electrical and Computer Engineering
National University of Singapore
Singapore, Singapore
yingqi.liang@u.nus.edu

Junbo Zhang
School of Electrical Power
South China University of Technology
Guangzhou, China
epjbzhang@scut.edu.cn

Dipti Srinivasan
Department of Electrical and Computer Engineering
National University of Singapore
Singapore, Singapore
dipti@nus.edu.sg

***Abstract*—To data-driven estimate power system sensitivity matrix considering mitigating data saturation, a series of multivariable least squares (MLS) algorithms are proposed and compared, including the ordinary MLS (OMLS), the weighted MLS (WMLS), the memory-limited OMLS (ML-ORMLS), the memory-limited WRMLS (ML-WRMLS), and the memory-fading ML-WRMLS (MF-ML-WRMLS). Considering enhancing computational efficiency and accuracy by mitigating data saturation, the last three of them are specifically derived for sensitivity matrix estimation based on time-varying online-measured data. The effectiveness of the presented algorithms is verified and compared in the Nordic 32 system for voltage sensitivity matrix estimation. The results illustrate the prime algorithm in practice.**

Keywords—power system sensitivity matrix estimation, multivariable regression, data saturation, recursive least squares

I. INTRODUCTION

Power system sensitivity, which is defined as the coefficients in the linear function of multiple system state variables with respect to multiple control variables, is widely applied for power system operation and control [1-8]. Specially, voltage sensitivity is used for reactive power planning and voltage stability control [1-2], power loss sensitivity is for economic operations [3], eigenvalue sensitivity is for small signal stability analysis [4-5], and various kinds of sensitivities are widely employed for control design and system security assessment [6-8].

In conventional deterministic power systems, sensitivities can be obtained using offline models of the systems, such as the calculation by performing inversion of augmented Jacobian matrix [9], by the perturbation method [10], and by direct calculation using the power system offline equations [11]. However, in modern stochastic power systems, the sensitivity changes along with the time-varying operating conditions; therefore, owing to the ensuing model incompatible, the results from conventional methods may no longer be suitable in practice.

To address this issue, sensitivity estimation based on online-measured data of system operations and conditions has been developed. Recent researches in this field have focused on estimation algorithms (e.g., the ordinary least squares in [12], the locally weighted least squares in [13], and neural networks in [14]), considerations on different input data conditions and the corresponding solutions (e.g., solutions for data multicollinearity in [15] and solutions for noise issues in [16]), and applications of estimated sensitivity for various system operations and controls (e.g., the voltage control in [17] and the power loss minimization in [18]). Nevertheless, most existing methods are for sensitivity vector estimation rather than sensitivity matrix estimation. Regarding the applications of sensitivity estimation to real power systems with multiple states variables and multiple control variables, sensitivity matrix estimation is required.

For sensitivity matrix estimation based on online-measured data, data saturation often happens, i.e., when the sample size is large, the most recent samples have slight influence on estimation, which makes the estimator cannot "track" and reflect the near real-time relationships between state variables and control variables. To enhance feasibility to estimate the sensitivity matrix from online-measured data of time-varying power system operations and conditions, data saturation should be mitigated, which can enhance computational efficiency and accuracy.

To fill up the aforementioned research gaps, this paper investigates a series of multivariable least squares (MLS) algorithms for data-driven power system sensitivity matrix estimation, with the specific consideration on mitigating data saturation. Assume that in a linear function of multiple system state variables with respect to multiple control variables, the errors of outputs have zero means and equal variances, thus Gauss-Markov regression model can be used. Our investigation starts with transforming the ordinary least squares from a multiple-input single-output (MISO) version to a multiple-input multiple-output (MIMO) version, which derives the first algorithm named the ordinary multivariable least squares (OMLS). Next, considering the different importance of different data samples, a weighted version of the OMLS is developed, and the second algorithm named the weighted multivariable least squares (WMLS) is proposed. Then, to reduce data saturation when estimate the sensitivity matrix from a large amount of data, a recursive version of the OMLS using sliding window method is advanced, which is the third algorithm called the memory-limited ordinary recursive multivariable least squares (ML-ORMLS). Further, the recursive version of the WMLS is also presented to better mitigate data saturation (i.e., to stress different importance of data samples in each snapshot of recursion), leading to the fourth algorithm called the memory-limited weighted recursive multivariable least squares (ML-WRMLS). Finally, by applying forgetting factor method to the ML-WRMLS, the fifth and the final algorithm called the memory-fading LM-WRMLS (MF-ML-WRMLS) is upgraded. Compared to the fourth algorithm, the fifth algorithm can put higher weights to the most recent data and gradually and automatically slighter weights to the less recent ones, while limiting the sample size for each recursion.

Contributions of the work are to conduct sensitivity matrix estimation and mitigate data saturation by providing a series of solutions to the following issues progressively: 1) the OMLS and the WMLS are to provide mathematical formulations for sensitivity matrix estimation and for our derivations of further algorithms; 2) the ML-ORMLS and the ML-WRMLS are to mitigate data saturation and enhance computation accuracy and efficiency when estimating sensitivity matrices from online-measured data; and 3) the MF-ML-WRMLS is to automatically adjust weights while limiting the sample size in each recursion, which better fits the time-varying characteristic of power systems. To the best of our knowledge, it is the first research work that focuses on sensitivity matrix estimation with specific consideration on mitigating data saturation, aiming to find a fast and accurate algorithm. More generally, we provide feasible computational techniques for data-driven estimation issues in power systems where the time-varying online-measured data usually have large sample sizes and incur data saturation.

This paper proceeds as follows. Section II describes the methodology and mathematical formulations for power system sensitivity matrix estimation. Section III proposes algorithms of the OMLS, the WMLS, the ML-ORMLS, the ML-WRMLS and the MF-ML-WRMLS progressively, including their brief introductions, mathematical formulations and algorithm implementation procedures. Section IV presents three case studies based on the Nordic 32 system for voltage sensitivity estimation using the proposed algorithms compared with the conventional perturbation method and each other. Section V draws the conclusions.

II. Power System Sensitivity Matrix Estimation

The power system operation behavior can be characterized by the following nonlinear equation at each time t [19]:

$$\boldsymbol{s}(t) = f(\boldsymbol{c}(t)) \tag{1}$$

where $\boldsymbol{c}$ is a vector of system control variables, $\boldsymbol{s}$ is a vector of state variables, and f is the function of a MIMO model.

Suppose that the system is stable and operates around an equilibrium operating point, the following increment equation holds:

$$\Delta\boldsymbol{s}(t) = \frac{\partial f(\boldsymbol{c}(t))}{\partial \boldsymbol{c}(t)} \Delta\boldsymbol{c}(t) + \boldsymbol{\alpha}(0) \tag{2}$$

where the differential term $\partial f / \partial \boldsymbol{c}$ is the sensitivity matrix of $\boldsymbol{s}$ with respect to $\boldsymbol{c}$ at time t, and $\boldsymbol{\alpha}$ is the error vector that assumed to be zero mean and independent and identical normally distributed (IID).

Let $\boldsymbol{x}_t^{\mathrm{T}} = [x_{t1} \cdots x_{tm}]^{\mathrm{T}}$ represent $\Delta\boldsymbol{c}(t)$ and $\boldsymbol{y}_t^{\mathrm{T}} = [y_{t1} \cdots y_{tp}]^{\mathrm{T}}$ represent $\Delta\boldsymbol{s}(t)$, where p is the amount of state variables, and m is the amount of control variables. For N measurements, (2) can be transformed to its matrix form shown as the following model:

$$\boldsymbol{Y} = \boldsymbol{XB} + \boldsymbol{\varepsilon} \tag{3}$$

where $\boldsymbol{X}$ is an $N \times m$ matrix containing N sample units of m input control variables, i.e., $\boldsymbol{X} = [\boldsymbol{x}_1 \cdots \boldsymbol{x}_N]^{\mathrm{T}} = [\boldsymbol{x}_{(1)} \cdots \boldsymbol{x}_{(m)}]$, where $\boldsymbol{x}_j = [x_{j1} \cdots x_{jm}]$ is the j-th row vector of $\boldsymbol{X}$, $\boldsymbol{x}_{(i)} = [x_{1i} \cdots x_{Ni}]^{\mathrm{T}}$ is the i-th column vector of $\boldsymbol{X}$, and x_{ji} is the element of $\boldsymbol{X}$ with the coordinate of (j,i), with j=1,2,...,N, i=1,2,…,m; $\boldsymbol{Y}$ is an $N \times p$ matrix containing N sample units of p output state variables , i.e., $\boldsymbol{Y} = [\boldsymbol{y}_1 \cdots \boldsymbol{y}_N]^{\mathrm{T}} = [\boldsymbol{y}_{(1)} \cdots \boldsymbol{y}_{(p)}]$, where $\boldsymbol{y}_j = [\boldsymbol{y}_{j1} \cdots \boldsymbol{y}_{jp}]$ is the j-th row vector of $\boldsymbol{Y}$, $\boldsymbol{y}_{(k)} = [\boldsymbol{y}_{1k} \cdots \boldsymbol{y}_{Nk}]^{\mathrm{T}}$ is the k-th column vector of $\boldsymbol{Y}$, and y_{jk} is the element of $\boldsymbol{Y}$ with the coordinate of (j,k), with k=1,2,…,p; $\boldsymbol{B}$ is an $m \times p$ sensitivity matrix containing the coefficients, i.e., containing elements of the transposed matrix of $\partial f / \partial \boldsymbol{c}$; and $\boldsymbol{\varepsilon}$ is an $N \times p$ matrix of error terms containing elements of the transposed matrix of $\boldsymbol{\alpha}$.

Equation (3) is a standard multivariable linear regression model where N is infinite theoretically. However, for sensitivity matrix estimation, $\boldsymbol{B}$ is estimated based on a finite N. Therefore, Gauss-Markov regression model is adopted:

$$\boldsymbol{Y} = \boldsymbol{X}\hat{\boldsymbol{B}} + \boldsymbol{e} \tag{4}$$

where $\hat{\boldsymbol{B}}$ is the estimator, and $\boldsymbol{e}$ is the residual error matrix defined by

$$\boldsymbol{e} = \boldsymbol{Y} - \hat{\boldsymbol{Y}} \tag{5}$$

The residual error matrix $\boldsymbol{e}$ and error matrix $\boldsymbol{\varepsilon}$ are of identical multivariate normal distributions. Equation (4) can be solved by MLS algorithms.

III. MLS Algorithms for Sensitivity Matrix Estimation Considering Mitigating Data Saturation

A. The OMLS Algorithm

According to Gauss-Markov theorem and the IID assumption of output errors, the best linear unbiased estimator of the coefficients is given by the MLS estimator [20]. Given the sample data matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$, the OMLS minimizes the quadratic sum of elements in the residual error matrix $\boldsymbol{e}$ as follows:

$$\min Q = \min \sum_{j=1}^{N} \sum_{k=1}^{p} e_{jk}{}^2 = \min \sum_{j=1}^{N} \sum_{k=1}^{p} \left(y_{jk} - \hat{y}_{jk}\right)^2 \tag{6}$$

where $\hat{y}_{jk}$ is the element of $\hat{\boldsymbol{Y}}$ with the coordinate of (j,k), and e_{jk} is the element of $\boldsymbol{e}$ with the coordinate of (j,k), which have the same meanings in the WMLS as below.

By matrix transformations, the estimator $\hat{\boldsymbol{B}}$ is given by:

$$\hat{\boldsymbol{B}} = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1} \boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y} \tag{7}$$

The implementation procedure of the OMLS is: build the sample regression model by (4) with N sample units of the online-measured data $\boldsymbol{X}$ and $\boldsymbol{Y}$, and then calculate the estimator $\hat{\boldsymbol{B}}$ by (7).

B. The WMLS Algorithm

Different to the OMLS that treats all data samples with equal weights, to distinguish the different importance of different data samples, the WMLS is developed by minimizing the weighted quadratic sum of elements in $\boldsymbol{e}$ as follows:

$$\min Q = \min \sum_{j=1}^{N} \sum_{k=1}^{p} w_{jk} e_{jk}{}^{2} = \min \sum_{j=1}^{N} \sum_{k=1}^{p} w_{jk} \left(y_{jk} - \hat{y}_{jk} \right)^{2} \tag{8}$$

where w_{jk} is the element of the weighting matrix $\boldsymbol{W}$ with the coordinate of (j,k).

By matrix transformations, the estimator $\hat{\boldsymbol{B}}$ is given by:

$$\hat{\boldsymbol{B}} = \left[\hat{\boldsymbol{\beta}}_{(1)}, \hat{\boldsymbol{\beta}}_{(2)}, \cdots, \hat{\boldsymbol{\beta}}_{(p)} \right] \tag{9-a}$$

$$\hat{\boldsymbol{\beta}}_{(k)} = \left(\boldsymbol{X}^{\mathrm{T}} \boldsymbol{W}_{(k)} \boldsymbol{X} \right)^{-1} \boldsymbol{X}^{\mathrm{T}} \boldsymbol{W}_{(k)} \boldsymbol{y}_{(k)} \tag{9-b}$$

with

$$\boldsymbol{W}_{(k)} = \mathrm{diag}\left(\boldsymbol{w}_{(k)} \right) \tag{10}$$

where $\hat{\boldsymbol{\beta}}_{(k)}$ is the k-th column vector of $\hat{\boldsymbol{B}}$, and $\boldsymbol{w}_{(k)}$ is the k-th column vector of $\boldsymbol{W}$.

In some cases, weights to the residual errors from different variables are the same, i.e.,

$$\boldsymbol{w}_{(k)} = \left[w_{1k} \ w_{2k} \cdots w_{Nk} \right]^{\mathrm{T}} = \boldsymbol{w} \tag{11}$$

where $\boldsymbol{w}$ is the each of column vectors in $\boldsymbol{W}$. Then (9) can be simplified as:

$$\hat{\boldsymbol{B}} = \left(\boldsymbol{X}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{X} \right)^{-1} \boldsymbol{X}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{Y} \tag{12}$$

The implementation procedure of the WMLS is: first, build the sample regression model by (4) with N sample units of the online-measured data $\boldsymbol{X}$ and $\boldsymbol{Y}$; next, set the weighting matrix $\boldsymbol{W}$; then calculate the estimator $\hat{\boldsymbol{B}}$ by (9) or (12).

C. The ML-ORMLS Algorithm

Based on the OMLS, the ML-ORMLS is developed using sliding window method to limit the sample size for each recursion, which can mitigate data saturation. The brief derivation is as follows.

Suppose the total sample size adds one to itself per iterative time unit, and the sensitivity matrix estimators $\hat{\boldsymbol{B}}_{N+n}$ and $\hat{\boldsymbol{B}}_{N+n+1}$ at the time N+n and N+n+1 are

$$\hat{\boldsymbol{B}}_{N+n} = \left(\boldsymbol{X}_{N+n}{}^{\mathrm{T}} \boldsymbol{X}_{N+n} \right)^{-1} \boldsymbol{X}_{N+n}{}^{\mathrm{T}} \boldsymbol{Y}_{N+n} \tag{13-a}$$

$$\hat{\boldsymbol{B}}_{N+n+1} = \left(\boldsymbol{X}_{N+n+1}{}^{\mathrm{T}} \boldsymbol{X}_{N+n+1} \right)^{-1} \boldsymbol{X}_{N+n+1}{}^{\mathrm{T}} \boldsymbol{Y}_{N+n+1} \tag{13-b}$$

respectively, with

$$\boldsymbol{X}_{N+n} = \left[\boldsymbol{x}_{1+n}{}^{\mathrm{T}} \cdots \boldsymbol{x}_{N+n}{}^{\mathrm{T}} \right]^{\mathrm{T}} = \left[\boldsymbol{x}_{n+1}{}^{\mathrm{T}} \ \ \boldsymbol{X}_{R,N+n}{}^{\mathrm{T}} \right]^{\mathrm{T}} \tag{14-a}$$

$$\boldsymbol{X}_{N+n+1} = \left[\boldsymbol{x}_{2+n}{}^{\mathrm{T}} \cdots \boldsymbol{x}_{N+n+1}{}^{\mathrm{T}} \right]^{\mathrm{T}} = \left[\boldsymbol{X}_{R,N+n}{}^{\mathrm{T}} \ \ \boldsymbol{x}_{N+n+1}{}^{\mathrm{T}} \right]^{\mathrm{T}} \tag{14-b}$$

$$\boldsymbol{Y}_{N+n} = \left[\boldsymbol{y}_{1+n}{}^{\mathrm{T}} \cdots \boldsymbol{y}_{N+n}{}^{\mathrm{T}} \right]^{\mathrm{T}} = \left[\boldsymbol{y}_{n+1}{}^{\mathrm{T}} \ \ \boldsymbol{Y}_{R,N+n}{}^{\mathrm{T}} \right]^{\mathrm{T}} \tag{14-c}$$

$$\boldsymbol{Y}_{N+n+1} = \left[\boldsymbol{y}_{2+n}{}^{\mathrm{T}} \cdots \boldsymbol{y}_{N+n+1}{}^{\mathrm{T}} \right]^{\mathrm{T}} = \left[\boldsymbol{Y}_{R,N+n}{}^{\mathrm{T}} \ \ \boldsymbol{y}_{N+n+1}{}^{\mathrm{T}} \right]^{\mathrm{T}} \tag{14-d}$$

where N is the length of the sliding window (and also the sample size for each recursion), n is the recursive time, and $j = 1+n, 2+n, \ldots, N+n$ (and $N+n+1$).

Equations (14) reveal that online-measured data matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ at the time N+n and N+n+1 have common parts of $\boldsymbol{X}_{R,N+n}$ and $\boldsymbol{Y}_{R,N+n}$ respectively. Based on this mathematical relationship, the recursion is built by adding the most recent rows of $\boldsymbol{X}$ and $\boldsymbol{Y}$, and removing their oldest rows, while holding $\boldsymbol{X}_{R,N+n}$ and $\boldsymbol{Y}_{R,N+n}$ in each recursion. Therefore, during the recursion, the sample size for estimation in each recursion is kept constant N and thus the memory is "limited".

To obtain the estimator $\hat{\boldsymbol{B}}_{N+n+1}$ via $\hat{\boldsymbol{B}}_{N+n}$, we partition $\hat{\boldsymbol{B}}_{N+n}$ and turn the crank on the matrices considering the common parts $\boldsymbol{X}_{R,N+n}$ and $\boldsymbol{Y}_{R,N+n}$ in $\hat{\boldsymbol{B}}_{N+n}$, thus have (15) and (16):

$$\boldsymbol{P}_{N+n} = \left(\boldsymbol{X}_{N+n}{}^{\mathrm{T}} \boldsymbol{X}_{N+n} \right)^{-1} = \left(\boldsymbol{x}_{n+1}{}^{\mathrm{T}} \boldsymbol{x}_{n+1} + \boldsymbol{X}_{R,N+n}{}^{\mathrm{T}} \boldsymbol{X}_{R,N+n} \right)^{-1} \tag{15}$$

$$\boldsymbol{Z}_{N+n} = \boldsymbol{X}_{N+n}{}^{\mathrm{T}} \boldsymbol{Y}_{N+n} = \boldsymbol{x}_{n+1}{}^{\mathrm{T}} \boldsymbol{y}_{n+1} + \boldsymbol{X}_{R,N+n} \boldsymbol{Y}_{R,N+n} \tag{16}$$

Then the estimator $\hat{\boldsymbol{B}}_{N+n}$ at the time N+n is expanded as

$$\begin{aligned} \hat{\boldsymbol{B}}_{N+n} &= \boldsymbol{P}_{N+n} \boldsymbol{Z}_{N+n} \\ &= \left(\boldsymbol{x}_{n+1}{}^{\mathrm{T}} \boldsymbol{x}_{n+1} + \boldsymbol{X}_{R,N+n}{}^{\mathrm{T}} \boldsymbol{X}_{R,N+n} \right)^{-1} \left(\boldsymbol{x}_{n+1}{}^{\mathrm{T}} \boldsymbol{y}_{n+1} + \boldsymbol{X}_{R,N+n} \boldsymbol{Y}_{R,N+n} \right) \end{aligned} \tag{17}$$

Based on (15), the expansion of $\boldsymbol{P}_{N+n+1}$ is derived as follows:

$$\begin{aligned} \boldsymbol{P}_{N+n+1} &= \left(\boldsymbol{X}_{N+n+1}{}^{\mathrm{T}} \boldsymbol{X}_{N+n+1} \right)^{-1} = \left(\boldsymbol{P}_{N+n}{}^{-1} + \boldsymbol{U}_{N+n+1} \right)^{-1} \\ &= \boldsymbol{P}_{N+n} - K_{*N+n+1}{}^{-1} \boldsymbol{P}_{N+n} \boldsymbol{U}_{N+n+1} \boldsymbol{P}_{N+n} = \boldsymbol{P}_{N+n} - K_{N+n+1} \boldsymbol{U}_{N+n+1} \boldsymbol{P}_{N+n} \\ &= \boldsymbol{P}_{N+n} - K_{N+n+1} K_{N+n+1}{}^{\mathrm{T}} \left(1 + \boldsymbol{u}_{N+n+1} \boldsymbol{P}_{N+n} \boldsymbol{u}_{N+n+1}{}^{\mathrm{T}} \right) \end{aligned} \tag{18}$$

with (19), (20) and (21) as follows:

$$\boldsymbol{U}_{N+n+1} = -\boldsymbol{x}_{n+1}{}^{\mathrm{T}} \boldsymbol{x}_{n+1} + \boldsymbol{x}_{N+n+1}{}^{\mathrm{T}} \boldsymbol{x}_{N+n+1} = \boldsymbol{u}_{N+n+1}{}^{\mathrm{T}} \boldsymbol{u}_{N+n+1} \tag{19}$$

$$K_{*N+n+1} = 1 + \boldsymbol{u}_{N+n+1} \boldsymbol{P}_{N+n} \boldsymbol{u}_{N+n+1}{}^{\mathrm{T}} \tag{20}$$

$$K_{N+n+1} = \boldsymbol{P}_{N+n} K_{*N+n+1}{}^{-1} = \boldsymbol{P}_{N+n} \Big/ \left(1 + \boldsymbol{u}_{N+n+1} \boldsymbol{P}_{N+n} \boldsymbol{u}_{N+n+1}{}^{\mathrm{T}} \right) \tag{21}$$

Based on (16), the expansion of $\boldsymbol{Z}_{N+n+1}$ is derived as the following (22) and (23) through a series of omitted derivation processes:

$$\boldsymbol{Z}_{N+n+1} = \boldsymbol{X}_{N+n+1}{}^{\mathrm{T}} \boldsymbol{Y}_{N+n+1} = \boldsymbol{Z}_{N+n} + \boldsymbol{V}_{N+n+1} \tag{22}$$

$$\boldsymbol{V}_{N+n+1} = -\boldsymbol{x}_{n+1}{}^{\mathrm{T}} \boldsymbol{y}_{n+1} + \boldsymbol{x}_{N+n+1}{}^{\mathrm{T}} \boldsymbol{y}_{N+n+1} = \boldsymbol{u}_{N+n+1}{}^{\mathrm{T}} \boldsymbol{v}_{N+n+1} \tag{23}$$

Based on (17) and because of (18) to (23), the recursive formulation of $\hat{\boldsymbol{B}}_{N+n+1}$ is derived as follows:

$$\begin{aligned}\hat{\boldsymbol{B}}_{N+n+1} &= \boldsymbol{P}_{N+n+1}\boldsymbol{Z}_{N+n+1} \\ &= \left(\boldsymbol{P}_{N+n} - K_{*N+n+1}{}^{-1}\boldsymbol{P}_{N+n}\boldsymbol{U}_{N+n+1}\boldsymbol{P}_{N+n}\right)\left(\boldsymbol{Z}_{N+n} + \boldsymbol{V}_{N+n+1}\right) \\ &= \hat{\boldsymbol{B}}_{N+n} + \boldsymbol{P}_{N+n}K_{*N+n+1}{}^{-1}\left(\boldsymbol{V}_{N+n+1} - \boldsymbol{U}_{N+n+1}\hat{\boldsymbol{B}}_{N+n}\right) \\ &= \hat{\boldsymbol{B}}_{N+n} + K_{N+n+1}\left(\boldsymbol{V}_{N+n+1} - \boldsymbol{U}_{N+n+1}\hat{\boldsymbol{B}}_{N+n}\right)\end{aligned} \tag{24}$$

To sum up, the sensitivity matrix estimator $\hat{\boldsymbol{B}}_{N+n+1}$ is given by (24), (18), (21), (19) and (23). The implementation procedure of the ML-ORMLS is as follows:

Step 1: build the sample regression model by (4) with N sample units of the online-measured data $\boldsymbol{X}$ and $\boldsymbol{Y}$;

Step 2: set the total recursive time n_r $(n_r > 1)$;

Step 3: calculate initial values by (25) and (26):

$$\hat{\boldsymbol{B}}_{N+0} = \left(\boldsymbol{X}_{N+0}{}^{\mathrm{T}}\boldsymbol{X}_{N+0}\right)^{-1}\boldsymbol{X}_{N+0}{}^{\mathrm{T}}\boldsymbol{Y}_{N+0} \tag{25}$$

$$\boldsymbol{P}_{N+0} = \left(\boldsymbol{X}_{N+0}{}^{\mathrm{T}}\boldsymbol{X}_{N+0}\right)^{-1} \tag{26}$$

Step 4: calculate $\hat{\boldsymbol{B}}_{N+n}$ and $\boldsymbol{P}_{N+n}$ based on $N+n$ units of online-measured data, and get the most recent sample vectors $\boldsymbol{y}_{N+n+1}$ and $\boldsymbol{x}_{N+n+1}$ for the ($N+n+1$) -th measurement;

Step 5: calculate $\hat{\boldsymbol{B}}_{N+n+1}$ by (24), (18), (21), (19) and (23) orderly;

Step 6: remove the oldest data vectors $\boldsymbol{y}_{n+1}$ and $\boldsymbol{x}_{n+1}$; therefore, the online-measured data matrices $\boldsymbol{Y}_{N+n+1}$ and $\boldsymbol{X}_{N+n+1}$ are renewed;

Step 7: if the recursive time $n < n_r - 1$, then let $n = n+1$ and return to Step 2; else, output $\hat{\boldsymbol{B}}_{N+n+1}$.

D. The ML-WRMLS Algorithm

Based on the WMLS and the ML-ORMLS, the ML-WRMLS is advanced on the strength of both recursive and weighted methods, i.e., Similar to the WMLS, the ML-WRMLS can assign different weights to different samples; and similar to the ML-ORMLS, the ML-WRMLS has a recursive definition to mitigate data saturation.

The following derivation of the ML-WRMLS is similar to that of the ML-ORMLS.

At the time $N+n$ and $N+n+1$, the sensitivity matrix estimators $\hat{\boldsymbol{B}}_{N+n}$ and $\hat{\boldsymbol{B}}_{N+n+1}$ are

$$\hat{\boldsymbol{B}}_{N+n} = \left[\hat{\boldsymbol{\beta}}_{(1)N+n}, \hat{\boldsymbol{\beta}}_{(2)N+n}, \cdots, \hat{\boldsymbol{\beta}}_{(p)N+n}\right] \tag{27-a}$$

$$\hat{\boldsymbol{\beta}}_{(k)N+n} = \left(\boldsymbol{X}_{N+n}{}^{\mathrm{T}}\boldsymbol{W}_{(k)N+n}\boldsymbol{X}_{N+n}\right)^{-1}\boldsymbol{X}_{N+n}{}^{\mathrm{T}}\boldsymbol{W}_{(k)N+n}\boldsymbol{y}_{(k)N+n} \tag{27-b}$$

$$\hat{\boldsymbol{B}}_{N+n+1} = \left[\hat{\boldsymbol{\beta}}_{(1)N+n+1}, \hat{\boldsymbol{\beta}}_{(2)N+n+1}, \cdots, \hat{\boldsymbol{\beta}}_{(p)N+n+1}\right] \tag{27-c}$$

$$\hat{\boldsymbol{\beta}}_{(k)N+n+1} = \left(\boldsymbol{X}_{N+n+1}{}^{\mathrm{T}}\boldsymbol{W}_{(k)N+n+1}\boldsymbol{X}_{N+n+1}\right)^{-1}\boldsymbol{X}_{N+n+1}{}^{\mathrm{T}}\boldsymbol{W}_{(k)N+n+1}\boldsymbol{y}_{(k)N+n+1} \tag{27-d}$$

respectively, with (14-a), (14-b) and the following (28):

$$\boldsymbol{Y}_{N+n} = \left[\boldsymbol{y}_{(1)N+n}, \boldsymbol{y}_{(2)N+n}, \cdots, \boldsymbol{y}_{(p)N+n}\right] \tag{28-a}$$

$$\boldsymbol{y}_{(k)N+n} = \left[y_{(k)1+n} \cdots y_{(k)N+n}\right]^{\mathrm{T}} = \left[\, y_{(k)n+1} \;\; \boldsymbol{y}_{R,(k)N+n}{}^{\mathrm{T}}\right]^{\mathrm{T}} \tag{28-b}$$

$$\boldsymbol{Y}_{N+n+1} = \left[\boldsymbol{y}_{(1)N+n+1}, \boldsymbol{y}_{(2)N+n+1}, \cdots, \boldsymbol{y}_{(p)N+n+1}\right] \tag{28-c}$$

$$\begin{aligned}\boldsymbol{y}_{(k)N+n+1} &= \left[y_{(k)2+n} \cdots y_{(k)N+1+n}\right]^{\mathrm{T}} \\ &= \left[\boldsymbol{y}_{R,(k)N+n}{}^{\mathrm{T}} \;\; y_{(k)N+n+1}\right]^{\mathrm{T}}\end{aligned} \tag{28-d}$$

$$\boldsymbol{W}_{(k)N+n} = \mathrm{diag}\left(\boldsymbol{w}_{(k)N+n}\right) = \begin{bmatrix} w_{(k)n+1} & \\ & \boldsymbol{W}_{R,(k)N+n}\end{bmatrix} \tag{28-e}$$

$$\boldsymbol{W}_{(k)N+n+1} = \mathrm{diag}\left(\boldsymbol{w}_{(k)N+n+1}\right) = \begin{bmatrix} \boldsymbol{W}_{R,(k)N+n} & \\ & w_{(k)N+n+1}\end{bmatrix} \tag{28-f}$$

where $\boldsymbol{w}_{(k)N+n} = \left[w_{(k)1+n} \cdots w_{(k)N+n}\right]^{\mathrm{T}}$ and $\boldsymbol{w}_{(k)N+n+1} = \left[w_{(k)2+n} \cdots w_{(k)N+1+n}\right]^{\mathrm{T}}$ are the k-th column vectors of weighting matrices $\boldsymbol{W}_{N+n}$ and $\boldsymbol{W}_{N+n+1}$ at the time $N+n$ and $N+n+1$ respectively.

To obtain the estimator $\hat{\boldsymbol{B}}_{N+n+1}$ via $\hat{\boldsymbol{B}}_{N+n}$, we partition $\hat{\boldsymbol{B}}_{N+n}$ and turn the crank on the matrices considering the common parts $\boldsymbol{X}_{R,N+n}$, $\boldsymbol{y}_{R,(k)N+n}$ and $\boldsymbol{W}_{R,(k)N+n}$ in $\hat{\boldsymbol{B}}_{N+n}$, thus have (29) and (30):

$$\begin{aligned}\boldsymbol{P}_{(k)N+n} &= \left(\boldsymbol{X}_{N+n}{}^{\mathrm{T}}\boldsymbol{W}_{(k)N+n}\boldsymbol{X}_{N+n}\right)^{-1} \\ &= \left(\boldsymbol{x}_{n+1}{}^{\mathrm{T}}w_{(k)n+1}\boldsymbol{x}_{n+1} + \boldsymbol{X}_{R,N+n}{}^{\mathrm{T}}\boldsymbol{W}_{R,(k)N+n}\boldsymbol{X}_{R,N+n}\right)^{-1}\end{aligned} \tag{29}$$

$$\begin{aligned}\boldsymbol{Z}_{(k)N+n} &= \boldsymbol{X}_{N+n}{}^{\mathrm{T}}\boldsymbol{W}_{(k)N+n}\boldsymbol{y}_{(k)N+n} \\ &= \boldsymbol{x}_{n+1}{}^{\mathrm{T}}w_{(k)n+1}\boldsymbol{y}_{(k)N+n} + \boldsymbol{X}_{R,N+n}{}^{\mathrm{T}}\boldsymbol{W}_{R,(k)N+n}y_{(k)n+1}\end{aligned} \tag{30}$$

Derive similarly to the ML-ORMLS, let

$$\begin{aligned}\boldsymbol{U}_{(k)N+n+1} &= -\boldsymbol{x}_{n+1}{}^{\mathrm{T}}w_{(k)n+1}\boldsymbol{x}_{n+1} + \boldsymbol{x}_{n+N+1}{}^{\mathrm{T}}w_{(k)N+n+1}\boldsymbol{x}_{N+n+1} \\ &= \boldsymbol{u}_{(k)N+n+1}{}^{\mathrm{T}}\boldsymbol{u}_{(k)N+n+1}\end{aligned} \tag{31}$$

$$\begin{aligned}\boldsymbol{V}_{(k)N+n+1} &= -\boldsymbol{x}_{n+1}{}^{\mathrm{T}}w_{(k)n+1}\boldsymbol{y}_{(k)N+n} + \boldsymbol{x}_{N+n+1}{}^{\mathrm{T}}w_{(k)N+n+1}y_{(k)N+n+1} \\ &= \boldsymbol{u}_{(k)N+n+1}{}^{\mathrm{T}}\boldsymbol{v}_{(k)N+n+1}\end{aligned} \tag{32}$$

After a series of omitted derivation steps, the sensitivity matrix estimator $\hat{\boldsymbol{B}}_{N+n+1}$ is given by:

$$\hat{\boldsymbol{B}}_{N+n+1} = \left[\hat{\boldsymbol{\beta}}_{(1)N+n+1}, \hat{\boldsymbol{\beta}}_{(2)N+n+1}, \cdots, \hat{\boldsymbol{\beta}}_{(p)N+n+1}\right] \tag{33-a}$$

$$\hat{\boldsymbol{\beta}}_{(k)N+n+1} = \hat{\boldsymbol{\beta}}_{(k)N+n} + K_{(k)N+n+1}\left(\boldsymbol{V}_{(k)N+n+1} - \boldsymbol{U}_{(k)N+n+1}\hat{\boldsymbol{\beta}}_{(k)N+n}\right) \tag{33-b}$$

$$\boldsymbol{P}_{(k)N+n+1} = \boldsymbol{P}_{(k)N+n} - K_{(k)N+n+1}K_{(k)N+n+1}{}^{\mathrm{T}}\left(1+\boldsymbol{u}_{(k)N+n+1}\boldsymbol{P}_{(k)N+n}\boldsymbol{u}_{(k)N+n+1}{}^{\mathrm{T}}\right) \tag{34}$$

$$K_{(k)N+n+1} = \boldsymbol{P}_{(k)N+n}\Big/\left(1+\boldsymbol{u}_{(k)N+n+1}\boldsymbol{P}_{(k)N+n}\boldsymbol{u}_{(k)N+n+1}{}^{\mathrm{T}}\right) \tag{35}$$

$$\begin{aligned}\boldsymbol{U}_{(k)N+n+1} &= -\boldsymbol{x}_{n+1}{}^{\mathrm{T}}w_{(k)n+1}\boldsymbol{x}_{n+1} + \boldsymbol{x}_{N+n+1}{}^{\mathrm{T}}w_{(k)N+n+1}\boldsymbol{x}_{N+n+1} \\ &= \boldsymbol{u}_{(k)N+n+1}{}^{\mathrm{T}}\boldsymbol{u}_{(k)N+n+1}\end{aligned} \tag{36}$$

$$\begin{aligned}\boldsymbol{V}_{(k)N+n+1} &= -\boldsymbol{x}_{n+1}{}^{\mathrm{T}}w_{(k)n+1}\boldsymbol{y}_{(k)n+1} + \boldsymbol{x}_{N+n+1}{}^{\mathrm{T}}w_{(k)N+n+1}y_{(k)N+n+1} \\ &= \boldsymbol{u}_{(k)N+n+1}{}^{\mathrm{T}}\boldsymbol{v}_{(k)N+n+1}\end{aligned} \tag{37}$$

The implementation procedure of the ML-WRMLS is as follows:

Step 1: build the sample regression model by (4) with N sample units of the online-measured data $\boldsymbol{X}$ and $\boldsymbol{Y}$;

Step 2: set the total recursive time $n_r\ (n_r > 1)$;

Step 3: get initial values by (38) and (39):

$$\hat{\boldsymbol{B}}_{N+0} = \left[\hat{\boldsymbol{\beta}}_{(1)N+0}, \hat{\boldsymbol{\beta}}_{(2)N+0}, \cdots, \hat{\boldsymbol{\beta}}_{(p)N+0}\right] \tag{38-a}$$

$$\hat{\boldsymbol{\beta}}_{(k)N+0} = \left(\boldsymbol{X}_{N+0}{}^{\mathrm{T}}\boldsymbol{W}_{(k)N+0}\boldsymbol{X}_{N+0}\right)^{-1}\boldsymbol{X}_{N+0}{}^{\mathrm{T}}\boldsymbol{W}_{(k)N+0}\boldsymbol{y}_{(k)N+0} \tag{38-b}$$

$$\boldsymbol{P}_{N+0} = \left[\boldsymbol{P}_{(1)N+0}, \boldsymbol{P}_{(2)N+0}, \cdots, \boldsymbol{P}_{(p)N+0}\right] \tag{39-a}$$

$$\boldsymbol{P}_{(k)N+0} = \left(\boldsymbol{X}_{N+0}{}^{\mathrm{T}}\boldsymbol{W}_{(k)N+0}\boldsymbol{X}_{N+0}\right)^{-1} \tag{39-b}$$

where

$$\boldsymbol{W}_{(k)N+0} = \mathrm{diag}\left(\boldsymbol{w}_{(k)N+0}\right), \boldsymbol{w}_{(k)N+0} = \left[w_{(k)1+0}\cdots w_{(k)N+0}\right]^{\mathrm{T}} \tag{40}$$

Step 4: calculate $\hat{\boldsymbol{B}}_{N+n}$ and $\boldsymbol{P}_{N+n}$ based on $N+n$ units of online-measured data, get the sample vectors $\boldsymbol{y}_{N+n+1}$ and $\boldsymbol{x}_{N+n+1}$ for the $N+n+1$ measurement, and renew the latest weighting row vector $\boldsymbol{w}_{N+n+1} = \left[w_{(1)N+n+1}, w_{(2)N+n+1}, \ldots, w_{(p)N+n+1}\right]$;

Step 5: calculate $\hat{\boldsymbol{B}}_{N+n+1}$ by (33), (34), (35), (36) and (37) orderly;

Step 6: remove the oldest data vectors $\boldsymbol{y}_{n+1}, \boldsymbol{x}_{n+1}$ and the oldest weighting row vector $\boldsymbol{w}_{n+1} = \left[w_{(1)n+1}, w_{(2)n+1}, \ldots, w_{(p)n+1}\right]$; therefore, the sample matrices $\boldsymbol{Y}_{N+n+1}$, $\boldsymbol{X}_{N+n+1}$ and weighting matrix $\boldsymbol{W}_{N+n+1}$ are renewed;

Step 7: if the recursive time $n < n_r - 1$, then let $n = n+1$ and return to Step 2; else, output $\hat{\boldsymbol{B}}_{N+n+1}$.

E. The FM-LM-WRMLS Algorithm

For estimating the sensitivity matrix that can track the time-varying functional relationship between state variables and control variables in near real-time, it is reasonable to put heavier weights to the most recent data and gradually and automatically lessen weights to the old ones. However, In the WRMLS and the ML-WRMLS, the assignments of weights are unspecified. To overcome the limitations of the WRMLS and ML-WRMLS, the MF-ML-WRMLS is proposed, which can automatically adjust the weights while limiting the sample size for each recursion. In the MF-ML-WRMLS, forgetting factor method is used, which enables the weights to be gradually fading by column, so the memory is “fading”.

The brief derivation of MF-ML-WRMLS is as follows.

First, for the WRMLS, suppose that the weights are assigned by row, and the original weighting matrix $\boldsymbol{W}_{ori}$ is:

$$\boldsymbol{W}_{ori} = \begin{bmatrix} w_1 & w_2 & \cdots w_p \\ \vdots & \vdots & \vdots \\ w_1 & w_2 & \cdots w_p \end{bmatrix} = \left[\boldsymbol{w}_{ori(1)}, \boldsymbol{w}_{ori(2)}, \cdots, \boldsymbol{w}_{ori(p)}\right] \tag{41-a}$$

$$\boldsymbol{w}_{ori(k)} = \left[w_k \cdots w_k\right]^{\mathrm{T}} \tag{41-b}$$

When a forgetting factor $\lambda \in (0,1]$ is applied to $\boldsymbol{w}_{\mathrm{ori}(k)}$, vectors of the weighting matrix is as follows:

$$\boldsymbol{w}_{\lambda,(k)} = \left[\lambda^{N-1}, \lambda^{N-2}, \cdots, \lambda^{N-N}\right]^{\mathrm{T}} w_k \tag{42}$$

Next, drawing an analogy with the WRMLS, vectors of the weighting matrix in the ML-WRMLS are derived as follows:

$$\boldsymbol{w}_{\lambda(k)N+n} = \left[\lambda^{N-1}\ \lambda^{N-2}\cdots\lambda^{N-N}\right]w_k \tag{43-a}$$

$$\boldsymbol{w}_{\lambda(k)N+n+1} = \left[\lambda^{N-1}\ \lambda^{N-2}\cdots\lambda^{N-N}\right]w_k \tag{43-b}$$

Then, the weighting factors in (36) and (37) are accordingly turned to be:

$$w_{(k)n+1} = \lambda^{N-1}w_k \tag{44-a}$$

$$w_{(k)N+n+1} = \lambda^{N-N}w_k = w_k \tag{44-b}$$

By employing a forgetting factor, the MF-ML-WRMLS automatically assigns time-varying weights on N data vectors that are used for estimation in each recursion, so as to put larger weights on recent data than those of the old data, and distinguish the diverse importance of different data to the estimation more precisely. The sensitivity matrix estimator $\hat{\boldsymbol{B}}_{N+n+1}$ is given by (33) to (35) and the following (45) and (46):

$$\begin{aligned}\boldsymbol{U}_{(k)N+n+1} &= w_k\left(\boldsymbol{x}_{N+n+1}{}^{\mathrm{T}}\boldsymbol{x}_{N+n+1} - \lambda^{N-1}\boldsymbol{x}_{n+1}{}^{\mathrm{T}}\boldsymbol{x}_{n+1}\right) \\ &= \boldsymbol{u}_{(k)N+n+1}{}^{\mathrm{T}}\boldsymbol{u}_{(k)N+n+1}\end{aligned} \tag{45}$$

$$\begin{aligned}\boldsymbol{V}_{(k)N+n+1} &= w_k\left(\boldsymbol{x}_{N+n+1}{}^{\mathrm{T}}y_{(k)N+n+1} - \lambda^{N-1}\boldsymbol{x}_{n+1}{}^{\mathrm{T}}\boldsymbol{y}_{k,n+1}\right) \\ &= \boldsymbol{u}_{(k)N+n+1}{}^{\mathrm{T}}\boldsymbol{v}_{(k)N+n+1}\end{aligned} \tag{46}$$

The implementation procedure is similar to that of the ML-WRMLS, just replacing (36) and (37) with (45) and (46) respectively. The weighting matrix in the MF-ML-WRMLS can be automatically renewed, which does not require the steps of adding or removing some weights during recursion manually (i.e., Step 4 and Step 6 in the implementation procedure of the ML-WRMLS), hence is rewarding to speed up the calculation.

IV. Case Study

A. System Introduction

In this study, the Nordic 32 system, which is widely used in power system dynamic analysis and control [21], is employed to emulate online power system operations. The system has 20 generators and 22 loads. Among of all generators, one is a synchronous condenser, another is the generator at slack bus considered as V-θ node, and others are P-V nodes. All loads are considered as P-Q nodes.

B. Data Generation

Real online-measured data recorded by the PJM (Pennsylvania-New Jersey-Maryland) companies in 2013 [22] was used to verify the proposed algorithms.

The procedure of data generation is detailed as follows. First, extract the load fluctuation data in 40 days at different areas from the PJM recoded dataset, and transform them into per-unit values. Next, take the outputs given by the generators and loads in the Nordic 32 system as the initial operating values, and 40 variation trend curves extracted as 40 relevant system parameters (i.e., 18 generators and 22 loads except the generator at slack bus and the synchronous condenser). Then, to improve the simulated sampling rate, run cubic spline interpolation function in MATLAB software; by interpolation, the measurement data is in the same condition of collecting one sample unit of data every 20ms. Afterwards, select of 22 load fluctuations in 6,000 operation modes near an operation point as the control variables. Finally, calculate the voltages of these 6,000 operation modes using DigSILENT / PowerFactory software [23], and take voltages of 54 buses other than those of the generator buses and load buses in each operation mode as the state variables.

By data generation, there are 6,000 sample units in the primary sample dataset. In each sample unit, the 22 reactive powers, which function as control variables, are regarded as input variables; the 54 bus voltages, which function as state variables, are regarded as output variables. After centralizing and standardizing the primary sample data set, the sample data set for sensitivity estimation is formed with a 6000×22 input variable matrix $\boldsymbol{X}$ and a 6000×54 output variable matrix $\boldsymbol{Y}$. The sensitivity matrix $\boldsymbol{B}$ calculated by the perturbation method is regarded as the benchmark, i.e., the real sensitivity matrix.

C. Performance Analysis I: Computational Accuracy

To verify estimation accuracy of the algorithms, dispersion ratio matrix (DRM) is defined as follows:

$$\text{DRM} = \left\{ \left| \frac{\text{estimated value} - \text{real value}}{\text{real value}} \right| \right\} \tag{47}$$

Four indices based on DRM are designed to evaluate the accuracy of $\hat{\boldsymbol{B}}$ as follows: the Frobenius norm of DRM (DRM-F), the mean of DRM (DRM-M), and the variance of DRM (DRM-V). To avoid random results, Monte Carlo method is adopted.

1. *Case I: Verifications on Correctness of Algorithm Derivations*

In this case, the correctness of algorithm derivations and the consistency of algorithm results are addressed. To compare weighted algorithms (i.e., the WMLS, the ML-WRMLS and the MF-ML-WRMLS) with unweighted ones (i.e., the OMLS and the ML-ORMLS) when weights are equal, let all elements in the weighting matrix be 1. To compare recursive algorithms (i.e., the ML-ORMLS, the ML-WRMLS and the MF-ML-WRMLS) with non-recursive ones (i.e., the OMLS and the WMLS) when they are conducted once only, let the recursive time be zero. Let the forgetting factor λ in weighted algorithms be one. Fifty Monte Carlo repetitions are carried out, and the accuracy verification results are shown in Table I.

Table I demonstrates that in all algorithms, values of all estimation indices are similar, with minor differences caused by stochastic errors. Therefore, the accuracy of weighted algorithms and their corresponding unweighted ones is consistent respectively, and the accuracy of recursive algorithms and their corresponding non-recursive ones is consistent respectively too, which proves that all algorithm derivations are correct.

2. *Case II: Verifications on Weighting Effects*

This case focused on weighting effects, i.e., the performances of all algorithms on distinguishing of importance of different samples.

Let the recursive time be zero, and the forgetting factor λ be 0.98. Fifty Monte Carlo repetitions are carried out, and the results are shown in Table II. In Table II, the results of the OMLS and the ML-ORMLS are the same in Case I.

From Table II, after assigning suitable weights by row, the WMLS, the ML-WRMLS and the MF-ML-WRMLS have better overall performances than their corresponding unweighted algorithms respectively. Besides, assigning weights by row in weighted algorithms not only enhance the accuracy of sensitivity estimations, but also have a conspicuous good effect on estimating $\boldsymbol{Y}$, which proves that the weighted algorithms have superior weighting effects on different samples.

Table I. The Results of Case I

Index	**Algorithm**				
	OMLS	***WMLS***	***ML-ORMLS***	***ML-WRMLS***	***MF-ML-WRMLS***
DRM-F	4.683	4.683	4.665	4.661	4.661
DRM-M	0.082	0.082	0.082	0.082	0.082
DRM-V	0.078	0.078	0.082	0.077	0.077

Table II. The Results of Case II

Index	**Algorithm**				
	OMLS	***WMLS***	***ML-ORMLS***	***ML-WRMLS***	***MF-ML-WRMLS***
DRM-F	4.683	4.023	4.694	4.025	4.025
DRM-M	0.082	0.090	0.022	0.020	0.020
DRM-V	0.078	0.072	0.087	0.072	0.072

3. *Case III: Verifications on Data Saturation Mitigation*

In this case, the validities of all algorithms to mitigate data saturation were put under tight scrutiny considering the time-varying characteristic of power systems.

Suppose data saturation was severe, and several most recent sample units were regarded as the most important data to the sensitivity matrix estimation, the amount of these recent data determines the setting of forgetting factor value. Simulations verify that when the forgetting factor is around 0.98 and the amount of the most important recent sample size is about 40, the estimation results can reach their optimums. Therefore, let the forgetting factor be 0.98 and the most important recent sample size be 40.

The weights are assigned both by row, and the way to assign them by row was the same as Case II. The recursive times are set to be ten. Fifty Monte Carlo repetitions are carried out. Estimation results were compared with the most recent data, so estimated values and real values in (47) were the elements of latest 40 rows of data. The accuracy verification results are shown in Table III and are analyzed from two perspectives: 1) all estimation indices in an algorithm; and 2) the performance of all algorithms according to an index.

TABLE III. THE RESULTS OF CASE III

INDEX	**ALGORITHM**			
	OMLS	***WMLS***	***ML-ORMLS***	***ML-WRMLS***
DRM-F	1.176E+3	1.001E+3	27.333	25.346
DRM-M	4.585	4.302	0.122	0.091
DRM-V	48.342	46.723	28.849	26.352

From the first perspective, the results are analyzed from three facets : first, as all estimation indices of the WMLS and the ML-WRMLS are better than that of the OMLS and the ML-ORMLS respectively, indicating that assigning weights by row is feasible in conditions where data saturation is severe; second, as all estimation indices of the MF-ML-WRMLS, ML-WRMLS and ML-ORMLS are better than that of the OMLS and the WMLS, recursive algorithms are superior to their corresponding non-recursive ones on mitigating data saturation.

From the second perspective, it is found that according to B-DRM-F, B-DRM-V, Y-DRM-F and Y-DRM-V of all algorithms, weighted algorithms perform excellent in enhancing the result stabilities of sensitivity matrix estimation and dependent variable matrix estimation; In addition, according to B-DRM-M, B-DRM-Q, Y-DRM-M and Y-DRM-Q of all algorithms, recursive algorithms contribute to improve the result precisions of sensitivity matrix estimation and dependent variable matrix revert.

According to all estimation indices of all algorithms, based on the above findings and analyses, recursive and weighted algorithm with forgetting factor, i.e., the FM-LM-WRMLS, has the best accuracy.

D. *Performance Analysis II: Computational Efficiency*

In this part, performances of presented algorithms were evaluated on the computational efficiency. The CPU used to run the algorithms was Intel (R) Core (TM) i5-3230M CPU @ 2.60GHz. The computational efficiency of all provided algorithms was compared with that of the MISO methods mentioned in Chapter I. The WMLS and the MF-ML-WRMLS algorithms were tested with the forgetting factor value of 0.98, and the recursive time of the ML-ORMLS and the MF-ML-WRMLS algorithms was one. The running time of the algorithms in Case III is shown in Table IV.

From Table IV, the proposed algorithms have higher computational efficiency than conventional MISO methods. What's more, the recursive algorithms have shorter execution time than that of non-recursive ones because of decreasing sample sizes, which manifests that the recursive algorithms have higher computational efficiency.

TABLE IV. COMPUTATIONAL EFFICIENCY

ALGORITHM	**COMPUTING TIME**	
	PROPOSED ALGORITHM	***CONVENTIONAL MISO METHOD***
OMLS	0.088s	1.108s
WMLS	4.643s	5.224s
LM-ORMLS	0.017s	-
LM-WRMLS	1.323s	-
FM-LM-WRMLS	1.472s	-

E. *Further Discussions: The Stability of Estimator*

In this part, further discussions on sensitivity matrix estimation algorithms based on the above results are presented from a more comprehensive view.

In Case I, DRM obtained by the OMLS are shown in Fig. 1. The dispersions of most elements in $\hat{\boldsymbol{B}}$ are approximate to zero and very few of them are extremely large, leading to large values of B-DRM-F, which indicates that $\hat{\boldsymbol{B}}$ estimated based on all data may lose accuracy as for few elements.

In Case III, DRM obtained by the FM-LM-WRMLS is shown in Fig. 2. The values of elements in DRM become smaller conspicuously comparing with those in Fig. 1, and extremely large values are diminished, indicating that the MF-ML-WRMLS has the superiority of overall high accuracy and stability considering severe data saturation.

Comprehensively taking accuracy, efficiency and stability into consideration, the MF-ML-WRMLS will be a preference for power system sensitivity matrix estimation, next the ML-ORMLS, then WMLS, followed by the OMLS, orderly.

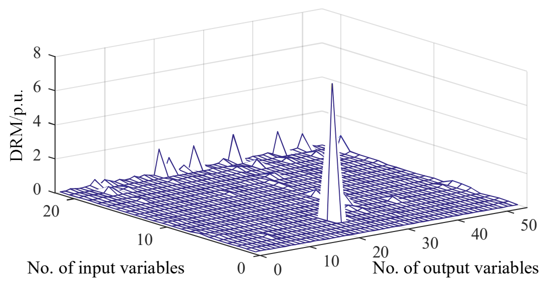

Fig. 1. DRM of estimated sensitivity matrix using the OMLS in Case I.

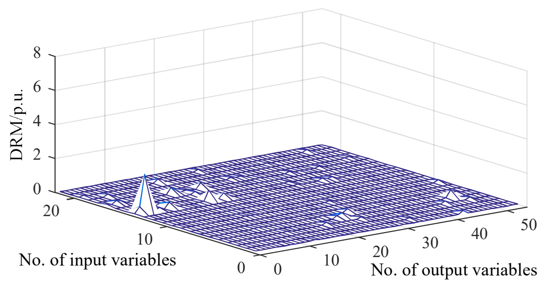

Fig. 2. DRM of estimated sensitivity matrix using the MF-ML-WRMLS in Case III.

V. CONCLUSIONS

This paper presented a series of MLS algorithms for power system multivariable sensitivity matrix estimation progressively, with specific consideration on mitigating data saturation. The proposed algorithms were tested in three cases studies based on the Nordic 32 system for voltage sensitivity estimation compared with the conventional perturbation method and each other. Simulation results prove that 1) All algorithms are derived correctly; 2) The weighted algorithms have good weighting effects to stress different importance of different samples; 3)the recursive algorithms can mitigate data saturation and estimate sensitivity matrix fast and accurately. The presented algorithms are verified to be of high accuracy, efficiency and stability, among of which the MF-ML-WRMLS algorithm performs best.

REFERENCES

[1] J. Dragosavac, Ž. Janda and J. V. Milanović, "Coordinated Reactive Power-Voltage Controller for Multimachine Power Plant," *IEEE Trans. Power Syst.,* vol. 27, no. 3, pp. 1540-1549, Aug. 2012.

[2] M. Paramasivam, A. Salloum, V. Ajjarapu, V. Vittal, N. B.Bhatt and S. Liu, "Dynamic Optimization Based Reactive Power Planning to Mitigate Slow Voltage Recovery and Short Term Voltage Instability", *IEEE Trans. Power Syst.,* vol. 28, no. 4, pp. 3865-3873, Nov. 2013.

[3] S. Deilami, A. S. Masoum, P. S. Moses and A. S. Masoum Mohammad, "Real-Time Coordination of Plug-In Electric Vehicle Charging in Smart Grids to Minimize Power Losses and Improve Voltage Profile," *IEEE Trans. Smart Grid,* vol. 2, no. 3, pp. 456-467, Sept. 2011.

[4] *MANGO – Modal Analysis for Grid Operation: A Method for Damping Improvement through Operating Point Adjustment*, The report number PNNL-19890 prepared by Pacific Northwest National Laboratory, 2010.

[5] J. Zhang, C. Lu, C. Y. Chung, K. Men and L. Tu, "Online Re-Dispatching of Power Systems Based on Modal Sensitivity Identification," *IET Generation, Transmission & Distribution*, vol. 9, no. 12, pp. 1352-1360, Sept. 2015.

[6] C. Li, G. Li, C. Wang and Z. Du, "Eigenvalue Sensitivity and Eigenvalue Tracing of Power Systems With Inclusion of Time Delays," in *IEEE Transactions on Power Systems*, vol. 33, no. 4, pp. 3711-3719, July 2018.

[7] Y. Gu and L. Xie, "Fast Sensitivity Analysis Approach to Assessing Congestion Induced Wind Curtailment," *IEEE Trans. Power Syst.,* vol. 29, no. 1, pp. 101-120, Jan. 2014.

[8] A. Moawwad, E. F. El-Saadany, M. S. El Moursi and M. Albadi, "Critical Loading Characterization for MTDC Converters Using Trajectory Sensitivity Analysis," in IEEE Transactions on Power Delivery, vol. 33, no. 4, pp. 1962-1972, Aug. 2018.

[9] Q. Nguyen, K. Lao, P. Vu and S. Santoso, "Loss Minimization With Optimal Power Dispatch in Multi-Frequency HVac Power Systems," in *IEEE Transactions on Power Systems*, vol. 35, no. 3, pp. 1979-1989, May 2020.

[10] D. Choi and L. Xie, "Data Perturbation-Based Sensitivity Analysis of Real-Time Look-Ahead Economic Dispatch," in *IEEE Transactions on Power Systems*, vol. 32, no. 3, pp. 2072-2082, May 2017.

[11] G. Charlesa, "Power system analysis," *Wiley*, 1986.

[12] M. Mohanpurkar and S. Suryanarayanan, "Regression Modeling for Accommodating Unscheduled Flows in Electric Grids," *IEEE Trans. Power Syst.,* vol. 29, no. 5, pp. 2569-2570, Sept. 2014.

[13] J. Zhang, C. Y. Chung and Y. Han, "Online Damping Ratio Prediction Using Locally Weighted Linear Regression," *IEEE Trans. Power Syst.,* vol. 31, no. 3, pp. 1954-1962, May 2016.

[14] J. Peng, K. Li and G. W. Irwin, "A New Jacobian Matrix for Optimal Learning of Single-Layer Neural Networks," IEEE Trans. on Neural Networks, vol. 19, no. 1, pp. 119-129, Jan. 2008.

[15] J. Zhang, Z. Wang, X. Zheng, L. Guan and C. Y. Chung, "Locally Weighted Ridge Regression for Power System Online Sensitivity Identification Considering Data Collinearity," *IEEE Trans. Power Syst.,* vol. 33, no. 2, pp. 1624-1634, March 2018.

[16] J. Zhao, M. Netto and L. Mili, "A Robust Iterated Extended Kalman Filter for Power System Dynamic State Estimation," *IEEE Trans. Power Syst.,* vol. 32, no. 4, pp. 3205-3216, July 2017.

[17] K. H. Youssef, "A New Method for Online Sensitivity-Based Distributed Voltage Control and Short Circuit Analysis of Unbalanced Distribution Feeders," *IEEE Trans. on Smart Grid*, vol. 6, no. 3, pp. 1253-1260, May 2015.

[18] S. Jung and G. Jang, "A Loss Minimization Method on a Reactive Power Supply Process for Wind Farm," in *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3060-3068, July 2017.

[19] J. Zhang, X. Zheng, Z. Wang, L. Guan and C. Y. Chung, "Power System Sensitivity Identification—Inherent System Properties and Data Quality," *IEEE Trans. Power Syst.,* vol. 32, no. 4, pp. 2756-2766, July 2017.

[20] R. A. Romano and F. Pait, "Matchable-Observable Linear Models and Direct Filter Tuning: An Approach to Multivariable Identification," in *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2180-2193, May 2017.

[21] Y. Xu, M. Yin, Z. Y. Dong, R. Zhang, D. J. Hill and Y. Zhang, "Robust Dispatch of High Wind Power-Penetrated Power Systems Against Transient Instability," *IEEE Trans. Power Syst.,* vol. 33, no. 1, pp. 174-186, Jan. 2018.

[22] [Online]. Available: http://www.pjm.com/markets-and-operations/opsanalysis/historical-load-data.aspx

[23] B. Hoseinzadeh, F. M. Faria da Silva and C. L. Bak, "Adaptive Tuning of Frequency Thresholds Using Voltage Drop Data in Decentralized Load Shedding," *IEEE Trans. Power Syst.,* vol. 30, no. 4, pp. 2055-2062, July 2015.