Data Fusion-Based Distributed Prony Analysis

Lingling Fan∗

∗Department of Electrical Engineering, University of South Florida, Tampa FL USA 33620.
Phone: 1(813)974-2031, Fax: 1(813)974-5250, Email: linglingfan@usf.edu.

Abstract
This paper presents a distributed Prony analysis algorithm using data fusion approach. This classic approach can be found in Kalman filter’s measurement update. Distributed optimization algorithms, e.g., alternating direction method of multipliers (ADMM), suitable for constrained optimization problems, have been proposed in the previous literature to develop distributed architecture. In this article, we show that Prony analysis, a non-constrained least square estimation (LSE) problem, can be solved using the classic data fusion approach. Compared to the iterative distributed optimization algorithms (e.g., ADMM and subgradient methods), data fusion takes only one step. There is no need for iteration and there is no issue related to convergence. This approach leads to a distributed Prony analysis architecture which requires a much less demanding communication system (the bandwidth can be less than 0.1 Hz) compared to the conventional centralized Prony analysis for multiple channels which requires a bandwidth of 30 Hz. The application discussed in this paper is to identify oscillation modes from real-world Phasor Measurement Unit (PMU) data and further reconstruct signals. A key technical challenge to implement Prony analysis for signals from multiple channels is the difficulty to identify the noise characteristics of each channel. In this paper, a method is proposed to identify the noise covariances, which leads to the construction of a weighted least square estimation (WLSE) problem. This problem is solved through a distributed architecture. The effectiveness of the proposed distributed Prony analysis is demonstrated through case study results. The accuracy of the estimation is improved in one order compared with the centralized Prony analysis.

Keywords: Prony analysis, data fusion, Kalman filter, distributed computing

1. Introduction
A power system is a massive system that can be perturbed by load changes, generator trips, faults or networks changes. Power system oscillations are common issues. To mitigate oscillations, oscillations should be identified and studied in a timely manner. There are two separate approaches to identify power system
oscillations. The first approach is based on detailed dynamic models of the system. State-space modeling and the eigenvalue analysis can give the system’s oscillation modes [1]. Detailed modeling of a complicated power system is challenging and prone to errors. The second approach is based on measurements to identify oscillation modes. Measurement-based approach has been adopted by control engineers in practice. For example, equivalent system models will be constructed based on the measurements and further control strategies will be developed based on the identified system models.

With phasor measurement unit (PMU) data collected, electromechanical oscillation modes (< 2 Hz) can be identified from these measurements at 30 Hz sampling rate. Several measurement-based system identification have been proposed for PMU data-based estimation, such as Kalman filters [2, 3, 4], least square estimation [5], and subspace algorithm [6]. Prony analysis is one of the most common measurement-based identification approaches to identify oscillatory modes. Prony analysis has been introduced by Hauer et al in power systems in 1990 [7, 8]. The main idea is to directly estimate the frequency, damping and phase of modal components of a measured signal. An extension to Prony analysis is then introduced which allowed multiple signals to be analyzed at the same time resulting [9].

Application of distributed optimization techniques has recently been introduced in system modes identification [10, 11, 12]. For example, in [10], distributed Prony analysis using alternating direction method of multipliers (ADMM) has been combined with centralized Prony method to estimate the slow frequency eigenvalues. Simulation data generated by PST [13] toolbox of IEEE 39-bus system is used to conduct Prony analysis. In the author’s previous paper [14], another distributed Prony analysis algorithm using consensus and subgradient update is developed. Distributed Prony analysis presented in the aforementioned papers can be applied to multiple signals from multiple locations collected at the same period of time. These algorithms can handle a large-dimension of PMU data by solving least square estimation (LSE) problems with small sizes in parallel and iteratively.

This paper serves as a rebuttal of the above distributed optimization approaches: iterations are not necessary. Indeed, Prony analysis is essentially an LSE problem without any constraints. Prony analysis of multi-channel signals is a multi-objective LSE problem. LSE problems were introduced by Gauss in 1790s. In 1960s, R. Kalman designed an iterative approach for LSE. See [15] for a detailed description. In Kalman filter, measurement update takes one step to find the best estimate given the prior information and current measurement [16]. There is no need of iteration. Compared to Kalman filter-based approach, distributed optimization approaches are not efficient. Kalman filter-based approach has been used in multi-sensor data fusion [17]. In this paper, the approach is named as data fusion approach.
In this paper, the philosophy of data fusion is examined in detail and applied to develop an effective algorithm for distributed Prony analysis. A key technical challenge to implement Prony analysis for signals from multiple channels is the difficulty to identify the noise characteristics of each channel. In this paper, a method is proposed to identify the noise covariances, which leads to the construction of a weighted least square estimation (WLSE) problem. This problem is solved through a distributed architecture.

In a nutshell, the contribution of the paper is to implement Kalman filter-based data fusion approach in Prony analysis with multiple channels. This approach has not been seen in Prony analysis. Compared to the other approaches where constrained optimization problems are formulated and further been solved by iterative distributed algorithms, e.g., [14, 12], the proposed approach does not require iterations and has advantages in computation.

The rest of the paper is as follows. Section II describes the fundamentals of Prony analysis. Section III describes the centralized multi-channel Prony analysis. Section IV presents data fusion and distributed Prony analysis. Section V further examines the relationship of data fusion based Prony analysis and multi-channel Prony analysis. Section VI presents the case study results. Section VII concludes this paper.

2. Fundamentals of Prony Analysis

Consider a Linear-Time Invariant (LTI) system with the initial state of \( x(t_0) = x_0 \) at the time \( t_0 \), if the input is removed from the system, the dynamic system model can be represented as [18]:

\[
\dot{x}(t) = Ax(t) \tag{1}
\]
\[
y(t) = Cx(t) \tag{2}
\]

where \( y \in \mathbb{R} \) is defined as the output of the system, \( x \in \mathbb{R}^n \) is the state of the system, \( A \in \mathbb{R}^{n \times n} \) and \( C \in \mathbb{R}^{1 \times n} \) are system matrices. The order of the system is defined by \( n \). If the \( \lambda_i \), \( p_i \), and \( q_i \) are the \( i \)-th eigenvalue, the corresponding right eigenvector, and left eigenvectors of \( A \) respectively, [1] can be expressed as:

\[
x(t) = \sum_{i=1}^{n} (q_i^T x_0)p_i e^{\lambda_i t} = \sum_{i=1}^{n} R_i x_0 e^{\lambda_i t} \tag{3}
\]
where $x_0$ is the initial state and $R_i = p_i q_i^T$ is a residue matrix. Based on (2), the $y(t)$ can be expressed as:

$$y(t) = \sum_{i=1}^{n} CR_i x_0 e^{\lambda_i t}.$$  \hspace{1cm} (4)

The observed or measured $y(t)$ consists of $N$ samples which are equally spaced by $\Delta t$ as:

$$y(t_k) = y(k), k = 1, ..., N - 1.$$  \hspace{1cm} (5)

Due to the fact that $k = 1, ..., N$, (5) can be expressed in matrix form as:

$$ZB = Y$$  \hspace{1cm} (9)

where $B_i = CR_i$, $N$ is the number of samples, $z_i$ are the eigenvalues of the system in discrete time domain, and $B_i$ is the residue of $z_i$. $z_i$ can be expressed as:

$$z_i = e^{\lambda_i \Delta t}$$  \hspace{1cm} (7)

As the $z_i$ are the roots of the characteristic polynomial function of the system, in order to find the $z_i$,
the coefficients of the polynomial need to be found first. The polynomial is formed as:

\[ z^n - (a_1 z^{n-1} + a_2 z^{n-2} + \ldots + a_n z^0) = 0. \]  
(10)

While the roots \( z_i \) might be complex numbers, the system polynomial coefficients \( a_i \) are real numbers. This feature helps develop algorithms since real numbers will be handled by computer algorithms while complex numbers cannot be directly handled.

From (10), we have

\[ z^n = a_1 z^{n-1} + a_2 z^{n-2} + \ldots + a_n z^0. \]  
(11)

Further, a linear prediction model (12) can be formulated since \( y(k) \) is the linear combination of \( z_i(k) \) based on (6). Therefore,

\[ y(n) = a_1 y(n - 1) + a_2 y(n - 2) + \ldots + a_n y(0). \]  
(12)

Enumerating the signal samples from step \( n \) to step \( N \), we have (13): \( Y = Da \).

The best estimate of \( a \) is found from the following normal equation.

\[ \hat{a} = (D^T D)^{-1} D^T Y. \]  
(14)

In the computing implementation, direct matrix inversion may result in numerical inaccuracy when the matrix \( D^T D \) approaches singular or its conditional number is too large. Instead, QR decomposition is used for computation.

Consider \( D = QR \) (\( Q \in \mathbb{R}^{(N-n+1)\times(N-n+1)} \) is an orthogonal matrix \( Q^T Q = I \) and \( R \in \mathbb{R}^{(N-n+1)\times n} \) is...
an upper triangular matrix), the estimate will be derived as:

\[ \hat{a} = (D^T D)^{-1} D^T Y \]
\[ = (R^T Q R)^{-1} R^T Q^T Y \]
\[ = (R^T R)^{-1} R^T Q^T Y \]

\[ \Rightarrow R^T R \hat{a} = R^T Q^T Y \]
\[ \Rightarrow R \hat{a} = Q^T Y \]

(15)

Since \( R \) is an upper triangular matrix, we can find \( \hat{a} \) by solving the last row of the equation set and moving backward.

\[ R_{n,n} \hat{a}_n = (Q^T Y)_n \Rightarrow \hat{a}_n = \frac{(Q^T Y)_n}{R_{n,n}} \]  
(16a)

\[ R_{n-1,n-1} \hat{a}_{n-1} + R_{n-1,n} \hat{a}_n = (Q^T Y)_{n-1} \Rightarrow \hat{a}_{n-1} = \frac{(Q^T Y)_{n-1} - R_{n-1,n} \hat{a}_n}{R_{n-1, n-1}} \]  
(16b)

\[ \cdots \]

(16) indicates that matrix inversion is avoided. This paper uses CVX [19] toolbox in Matlab to implement state estimation, where QR decomposition based solving is embedded. The CVX codes are shown as follows.

```matlab
cvx_begin
variable a(n)
minimize (Y-D*a)'*(Y-D*a)
cvx_end
```

Remarks: The dimension of \( D \) matrix is \((N - n + 1) \times n\). If \( n < \text{floor}(N/2) \), this is an over-determined linear equation and will be solved by the least square estimation (LSE). If \( n > N/2 \), the linear equations are under-determined and there are multiple solutions for \( a \). When the \( D \) matrix is square, there is a unique solution of \( a \) and the match will be the best. That is the reason that \( n \) is selected to be close to \( \text{floor}(N/2) \) [7].

\( D \) matrix is based on measurements. If the measurements contain noise, \( D \) matrix is not accurate and the resulting estimation is not accurate. Instead, if we have an over-determined case, the estimation will be...
more accurate. This is one reason that Prony analysis with multiple signals can perform better according to [20].

Briefly, \( Da = Y \) will be obtained, where \( a \) contains the coefficients of the characteristic polynomial (10) of the system and \( D \) and \( Y \) are constructed from the measured signals. Solution of (13) will provide the coefficients of (10). From (10), the roots \( z_i \) \((i = 1, \cdots, n)\) will be found.

The next step of the Prony analysis is to find the residues \( B_i \) in (6). As a result, \( B_i \) can be found by solving a set of overdetermined linear equations (9).

2.1. Signal reconstruction

Equation (9) can also be used to reconstruct the estimated signals. For a given set of residues \( B_i \) and system roots \( z_i \), signals at every sampling step \( \hat{y}(k) \) can be found.

3. Centralized Multi-Channel Prony Analysis

The conventional way to find the vector \( a \) from multiple signals has been documented in [9]. A brief description is offered as follows. Suppose that there are \( m \) channels of PMU data taken from the same period of time. For each channel of the PMU data, it is possible to formulate the \( D \) matrix and \( Y \) vector. They will be notated as \( D_i \) and \( Y_i \) for the \( i \)-th channel. \( a \) can be found by the following estimation problem.

\[
\begin{bmatrix}
D_1 \\
D_2 \\
\vdots \\
D_m
\end{bmatrix}
\begin{bmatrix}
a
\end{bmatrix}
= 
\begin{bmatrix}
Y_1 \\
Y_2 \\
\vdots \\
Y_m
\end{bmatrix}
\]

(17)

The \( m \) channels can be from one single PMU or from multiple PMUs. For example, one PMU can have multiple channels for measurements such as frequency, voltage, real and reactive power. The data do not need to be of same type.

When the number of the signal channels increases, the least square estimation has to handle a large dimension of matrix. The size of the matrix will be tremendous if a long period of time with signals collected from thousands of locations is considered. Therefore, to achieve scalability, distributed Prony analysis algorithms are sought.
4. Distributed Prony Analysis through Data Fusion

4.1. Data Fusion

Data fusion can be explained by the following simple example. Assume the state $x$ to be estimated has prior information $x_1 \in \mathbb{R}^n$ from the first measurement and $x_2 \in \mathbb{R}^n$ from the second measurement. These two measurements have different accuracy and the covariance matrices are assumed to be $\Sigma_1$ and $\Sigma_2$. In order to find the best estimation, we conduct the following multi-objective weighted least square estimation (LSE).

$$
\min_x (x - x_1)^T \Sigma_1^{-1} (x - x_1) + (x - x_2)^T \Sigma_2^{-1} (x - x_2)
$$  \hspace{1cm} (18)

Suppose $x \in \mathbb{R}^2$, we can then plot $x_1$ and $x_2$ in two-dimension as shown in Fig. 1.

The minimization problem is equivalent to find a $x$ in the entire two dimensional space that can minimize the weighted sum of the distances from this $x$ to the two prior information $x_1$ and $x_2$.

The best estimate $\hat{x}$ must located on the line between $x_1$ and $x_2$. This is a very important observation.

It can be illustrated intuitively. Suppose there is a best estimate $x'$ located not on the line of between $x_1$ and $x_2$, we can always project this point to the line and the projection is called $x''$. Then the distance between $x'$ to $x_i$ must be greater than the distance from $x''$ to $x_i$, $i = 1, 2$.

With this knowledge, we can then write the expression of any $x$ which locates between $x_1$ and $x_2$ as

$$
x = x_1 + d_1
$$  \hspace{1cm} (19)

where

$$
d_1 = K(x_2 - x_1)
$$  \hspace{1cm} (20)

$$
0 \leq K \leq 1
$$  \hspace{1cm} (21)

Note $K$ is in fact the Kalman filter gain. Fig. 1 is the geometric meaning of Kalman filter’s measurement update step.

In Fig. 1 it can be observed that $d_1 + d_2 = x_2 - x_1$, where $d_1$ and $d_2$ are the Euclidean distances from $x$ to $x_1$ and $x_2$ respectively.
Using $d_1$ and $d_2$, the original multi-LSE becomes the following minimization problem.

$$\min_{d_1, d_2} d_1^T \Sigma_1^{-1} d_1 + d_2^T \Sigma_2^{-1} d_2$$

subject to $d_1 + d_2 = x_2 - x_1$  \hspace{1cm} (22)

Substitute $d_2$ by an expression of $d_1$.

$$\min d_1^T \Sigma_1^{-1} d_1 + (D - d_1)^T \Sigma_2^{-1} (D - d_1)$$

where $D \triangleq x_2 - x_1$. \hspace{1cm} (24)

After a few steps of algebraic manipulation, the above objective function can be found to be the same as the following objective function.

$$\min (d_1 - \hat{d}_1)^T (\Sigma_1^{-1} + \Sigma_2^{-1}) (d_1 - \hat{d}_1) + D^T (\Sigma_1 + \Sigma_2)^{-1} D$$

where $\hat{d}_1 = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1} \Sigma_2^{-1} D$. \hspace{1cm} (25)

The above LSE component clearly shows that the best estimate of $d_1$ should be

$$\hat{d}_1 = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1} \Sigma_2^{-1} (x_2 - x_1)$$

$$= \Sigma_1 (\Sigma_1 + \Sigma_2)^{-1} (x_2 - x_1)$$  \hspace{1cm} (26,27)

Therefore, the best estimate of $x$ is as follows.

$$\hat{x} = x_1 + \Sigma_1 (\Sigma_1 + \Sigma_2)^{-1} (x_2 - x_1)$$

where $K$ is the Kalman filter gain. Note Kalman filter’s measurement correction step uses the same approach. \hspace{1cm} (28)

What is more important, the single-LSE in (25) shows the resulting covariance matrix is

$$\Sigma = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}$$

$$= \Sigma_1 - \Sigma_1 (\Sigma_1 + \Sigma_2)^{-1} \Sigma_1$$  \hspace{1cm} (29,30)

$$= \Sigma_2 - \Sigma_2 (\Sigma_1 + \Sigma_2)^{-1} \Sigma_2$$  \hspace{1cm} (31)
The above equation indicates that the resulting covariance after data fusion will be smaller than either one of the covariance matrices.

The above data fusion procedure can be extended to fuse more data. The fused estimate will be more accurate. In the following subsection, data fusion based distributed Prony analysis is introduced.

4.2. Distributed Prony Analysis

This subsection discusses the application of data fusion technique in distributed Prony analysis. For every channel, Prony analysis will be carried out. The output of Prony analysis is the parameter vector \( a \).

If there are \( m \) channels, \( a_i, i = 1, \ldots, m \) will be generated at the local level.

The minimization problem is as follows.

\[
\min_a \sum_i (a - a_i)^T \Sigma_i^{-1} (a - a_i)
\] (32)

To carry out data fusion, every two channels with direct communication links will exchange the parameter vectors and fuse the data. So we will have a new set of parameter vectors. For the new set, fusion will be carried out for every two vectors. This procedure will be carried on until there is only one parameter vector left.

Suppose two parameter vectors \( a_1 \) and \( a_2 \) are to be fused, each with a known covariance matrix, the fused data and the resulting covariance can be expressed as follows.

\[
a_{12} = a_1 + \Sigma_1 (\Sigma_1 + \Sigma_2)^{-1} (a_2 - a_1)
\] (33)

\[
\Sigma_{12} = \Sigma_1 - \Sigma_1 (\Sigma_1 + \Sigma_2)^{-1} \Sigma_1
\] (34)

If \( m \) is an even number, the number of fused parameter vectors is now \( m/2 \). This fusion step can be kept carrying out until we have only one parameter vector. Fig. 2 gives an illustrative computing procedure of fusion. For 8 channels of data, three steps are needed to find the fused data. The approximate number of steps to be carried out is \( \log_2 m \) where \( m \) is the number of channels.

Fig. 2 also indicates the communication links for information exchange.

Remarks on communication requirements: It is easy to observe that for multi-channel Prony analysis, all PMU measurements with 30 Hz interval should be sent to a central controller to compute the parameter vector. On the other hand, for data fusion based Prony analysis, a much slower communication bandwidth can be used since the data to be exchanged are the parameter vector and its covariance matrix.
The information is calibrated at local level approximately every ten seconds. Information exchange also happens every tens of seconds. Therefore, neither the speed requirement nor the capacity requirement of the communication links is demanding for data fusion based Prony analysis.

5. Comparison of multi-channel Prony analysis and data fusion Prony analysis

In this section, the relationship between the two methods is examined. First, the objective functions of the least square estimation problems are examined.

For multi-channel Prony analysis, the objective function can be written as follows.

\[
\text{multi-channel Prony: } \min_a \sum_{i=1}^{m} \| D_i a - Y_i \|_2^2 (35) 
\]

As an LSE problem, the underlying assumption of the above objective function is that the noises of all channels are with the same covariance.

Solving the above minimization problem is the same as solving the following minimization problem.

\[
\min_a \sum_i (a - a_i)^T D_i^T D_i (a - a_i) \\
\text{where } a_i = (D_i^T D_i)^{-1} D_i^T Y_i (36) 
\]

\( a_i \) is the projection of \( Y_i \) on the range of \( D_i \), or the parameter vector found through individual Prony analysis.

The above relationship shows that if the covariance matrix \( \Sigma_i \) for each channel is selected as

\[
\Sigma_i = (D_i^T D_i)^{-1}, (37) 
\]

the resulting parameter vector from the data fusion approach should be exactly same as that from the multi-channel Prony analysis.

In reality, since noises at different channels may have different covariances, weighted LSE is used to achieve maximum likelihood for the measurement to happen. Therefore, the objective function should be as follows.

\[
\min_a \sum_i (D_i a - Y_i)^T \Sigma_{vi} (D_i a - Y_i) (38) 
\]
where $\Sigma_{vi}$ is the covariance matrix of the noise of each channel.

The above minimization problem is equivalent to the following minimization problem:

$$\min_a \sum_i (a - a_i)^T D_i^T \Sigma_{vi}^{-1} D_i (a - a_i)$$

where $a_i = (D_i^T \Sigma_{vi}^{-1} D_i)^{-1} D_i^T \Sigma_{vi}^{-1} Y_i$ \hspace{1cm} (39)

If $\Sigma_i$ used in data fusion approach is exactly as $(D_i^T \Sigma_{vi}^{-1} D_i)^{-1}$, then the data fusion approach is exactly same as solving the weighted LSE problem in (38).

6. Case Study

In this section, a case study of real-world PMU data is carried out. 17 frequency deviation signals from PMU measurements are given in Fig. 3. Three scenarios are studied:

1. Centralized multi-channel Prony analysis using 17. The reconstructed signals are shown in Fig. 5. The reconstructed signals match poorly with the PMU measurements presented in Fig. 3.

2. Data fusion assuming the same covariance. Reconstructed signals are shown in Fig. 6. The reconstructed signals have an overall better matching degree with the PMU measurements compared to those constructed from multi-channel Prony analysis.

3. Data fusion with adequate covariance assumption. The reconstructed signals are shown in Fig. 7. The reconstructed signals have the best matching degree with the the PMU measurements.

For each channel of time series data, 855 samples are obtained. With the sampling rate as 30 Hz, data of 28.5 seconds are obtained. The data are resampled to have a sampling rate of 10 Hz to achieve an accurate estimation from single-channel Prony analysis. The above procedure is necessary for single-channel Prony analysis since Prony analysis has its numerical limitation. For example, if the dominant oscillation mode is 0.44 Hz, then the best sampling rate about 6 Hz. If the sampling rate is 30 Hz, Prony analysis is not able to give accurate estimation due to numerical reasons. This shortcoming has been addressed in the author’s previous paper [14]. This issue can be solved by reducing the sampling rate or including more channels of signals.

With a reduced sampling rate, 285 samples are used for each channel. Take the floor of 285/2, we end up with a parameter vector $a$ with a dimension of 142. For each channel, Prony analysis is carried out to find the parameter vector. For 17 channels, the single-channel Prony analysis results in 17 parameter vectors. Multi-Prony analysis and data fusion will each give a parameter vector.
Table 1 lists the corresponding LES objective function \( \| Da - Y \|_2^2 \) for every channel using the estimated 19 parameter vectors. The bold numbers are the related sum of error squares for channel \( i \) using the parameter vector estimated based on the data of channel \( i \). Compare the bolded numbers with the sum of error squares from multi-channel Prony analysis and data fusion, we can find that single-channel Prony analysis gives the minimum value for this channel. However, when the parameter vector based on channel \( i \) is applied to another channel \( j \), the resulting sum of error squares can be significant. For example, the parameter vector estimated based on Channel 1 data will lead to large sum of error squares for Channel 3, 4, 9, 13, 14, 15, and 17.

6.1. How to select \( \Sigma_i \)

The key technical part in data fusion is to select adequate covariance matrices. Since there is no prior knowledge on \( \Sigma_{vi} \), we use the following approach to find covariance matrix \( \Sigma_i \) used in data fusion.

First, for every channel, Prony analysis is carried out and the signal is reconstructed. The reconstructed signal is compared with the original measurements to find the total squared error. This value will give us an indication on how to put weight on each parameter vectors. If the squared error for Channel \( i \) is very large compared to other channels, then the corresponding parameter vector \( a_i \) should have a small weight. If the squared error for Channel \( j \) is very small, then the corresponding parameter vector \( a_j \) will be penalized much less. The fused parameter vector should be closer to \( a_j \) than \( a_i \).

For the 17 signals, 17 individual Prony analysis is carried out. The reconstructed signals are plotted in Fig. 4. The error square for each channel \( e_i^2 \) is listed in Table 2. The covariance matrices are selected as the identity matrices multiplied by those error square values \( \Sigma_i = e_i^2 I \).

As for data fusion computing, the following steps are required.

1. Find \( m \) set of the parameter vector \( a_i \) based on every channel’s measurement and the normal equation in (14).

2. For the \( m \) parameter vectors and their covariance matrices, conduct data fusion by fusing every two vectors. This results in \( \lfloor \frac{m}{2} \rfloor \) parameter vectors.

3. Carry out the next layer data fusion and continue until there is only 1 parameter vector.

For the reconstructed signals generated from single-channel Prony analysis and the three test scenarios, Table 3 lists the error square between the reconstructed signals and the original signals. It can be observed that the accuracy achieved by the data fusion with adequate covariance is one order above the multi-channel Prony analysis. This example demonstrates the effectiveness of data fusion-based Prony analysis.
The reconstructed signals are shown in Figs. 4-7. Fig. 4 shows the reconstructed signals for each channel based on its own estimated parameter vector. Note that the parameter vector employed for each plot line is different. Ideally, it is best to present 17 reconstructed signals for each parameter vector. Due to space limit, for each parameter vector, only one signal is presented. It can be seen that the reconstructed signals do not match the original measurements very well. Fig. 5 presents the 17 reconstructed signals from a parameter vector generated from multi-channel Prony analysis. The total error squares is 2.081 (Table 3). The matching with the original signals is particularly poor from 12 seconds to 18 seconds. Fig. 6 and Fig. 7 are reconstructed signals based on parameters vectors from data fusion approach. The only difference is that in Fig. 6 same covariances are assumed for all channels. Therefore, the resulting parameter vector is the average of 17 parameter vectors obtained from single-channel Prony analysis. In Fig. 7 the covariances are selected based on the matching degree of each channel after single-channel Prony analysis (shown in Table 2). Numerical values from Table 3 indicate that the sum of error squares reduces by half after using the adequate covariances.

For comparison, Channel 3 original measurements from PMU, the reconstructed signals based on single-channel Prony analysis, multi-channel Prony analysis and two data fusion approaches are plotted in Fig. 8. It can be clearly seen that data fusion with adequate covariances results in the best match with the original measurements. The next best is data fusion assuming the same covariances.

Remarks: The case study clearly shows that the proposed data fusion approach can give the most accurate estimation of oscillation modes.

7. Conclusion

This paper presents distributed Prony analysis based on classic Kalman filter data fusion approach. The approach is much more efficient compared to many distributed Prony analysis algorithms in the literature. The paper presents the in-depth analysis of data fusion and the relationship of multi-channel Prony analysis and distributed Prony analysis. Real-world PMU measurements are used in case studies to demonstrate the effectiveness of the distributed Prony analysis method. The proposed distributed Prony analysis is efficient in computing and communication with superior matching degree.


List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$|Da - Y|_2^2 \times 10^{-5}$ for each channel and each estimated a vector</td>
<td>17</td>
</tr>
</tbody>
</table>
2 Error square from each channel used as the inverse of the covariance for each channel (unit: $10^{-5}$ pu)

3 The sum of the error squares (unit $10^{-5}$) between the reconstructed signal and the original signal.

List of Figures

1 Geometric explanation of weighted LSE. It is similar as to find a best \( x \) that will make the sum of the weighted distances to \( x_1 \) and \( x_2 \) the minimum. \( x \) must located on the line of \( x_1 - x_2 \).

2 Data fusion procedure and required communication links (in red). For example, PMU2 sends data to PMU1 and PMU1 carries out the fusion procedure.

3 Frequency deviation measurements.

4 Reconstructed signals after individual Prony analysis.

5 Reconstructed signals using the parameter vector from multiple channel Prony analysis. The reconstructed signals match poorly with the PMU measurements presented in Fig. 3.

6 Reconstructed signals using the averaged parameter vector.

7 Reconstructed signals after data fusion.

8 Comparison of measurements and reconstructed signals of Channel 3.

Figure 1: Geometric explanation of weighted LSE. It is similar as to find a best \( x \) that will make the sum of the weighted distances to \( x_1 \) and \( x_2 \) the minimum. \( x \) must located on the line of \( x_1 - x_2 \).
|       | chn 1 | chn 2 | chn 3 | chn 4 | chn 5 | chn 6 | chn 7 | chn 8 | chn 9 | chn 10 | chn 11 | chn 12 | chn 13 | chn 14 | chn 15 | chn 16 | chn 17 | all chns | fusion |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|--------|--------|--------|--------|--------|---------|---------|
| chn 1 | 0.0000| 0.0890| 0.4299| 0.5877| 0.4794| 0.1962| 0.1997| 0.1244| 0.1146| 0.3113  | 0.1666 | 0.3191 | 1.0257 | 0.8342 | 1.3452 | 2.3000  | 0.9037  | 0.0042  | 0.0027  |
| chn 2 | 0.0430| 0.0001| 0.1842| 0.1296| 0.0234| 0.1190| 0.0719| 0.0586| 0.0363| 0.0575  | 0.0774 | 0.0605 | 0.0792 | 0.1190 | 0.0373 | 0.0562 | 0.0310  | 0.0018  | 0.0163  |
| chn 3 | 0.1227| 0.0122| 0.0000| 0.0371| 0.0030| 0.0074| 0.0281| 0.0344| 0.0198| 0.0418  | 0.0666 | 0.1475 | 0.0045 | 0.0806 | 0.0080 | 0.0273 | 0.0147  | 0.0003  | 0.0052  |
| chn 4 | 0.1215| 0.0724| 0.2353| 0.0001| 0.6517| 0.4136| 0.1777| 0.1601| 0.0945| 0.0950  | 0.1585 | 0.1551 | 0.0885 | 0.4663 | 0.0959 | 0.2827 | 0.0962  | 0.0023  | 0.0100  |
| chn 5 | 0.0825| 0.0123| 0.0805| 0.0001| 0.0499| 0.0000| 0.0237| 0.0706| 0.0403| 0.0081  | 0.0184 | 0.0627 | 0.0058 | 0.0529 | 0.0137 | 0.0229 | 0.0347  | 0.0005  | 0.0017  |
| chn 6 | 0.1225| 0.0115| 0.0018| 0.0635| 0.0026| 0.0000| 0.0271| 0.0333| 0.0191| 0.0439  | 0.0664 | 0.1412 | 0.0038 | 0.0804 | 0.0278 | 0.0242 | 0.0154  | 0.0003  | 0.0056  |
| chn 7 | 0.0631| 0.0273| 0.0884| 0.0816| 0.0174| 0.0602| 0.0000| 0.0246| 0.0376| 0.0286  | 0.0358 | 0.0486 | 0.0570 | 0.0823 | 0.0227 | 0.0483 | 0.0271  | 0.0016  | 0.0073  |
| chn 8 | 0.0692| 0.0124| 0.0203| 0.0446| 0.0039| 0.0179| 0.0041| 0.0000| 0.0129| 0.0199  | 0.0040 | 0.0480 | 0.0073 | 0.0247 | 0.0063 | 0.0104 | 0.0235  | 0.0004  | 0.0060  |
| chn 9 | 0.1618| 0.1519| 0.3743| 0.0959| 0.0973| 0.8343| 0.1786| 0.1990| 0.0000| 0.2530  | 0.2583 | 0.4129 | 0.3544 | 0.3337 | 0.0801 | 0.2266 | 0.1025  | 0.0031  | 0.0461  |
| chn 10| 0.0388| 0.0114| 0.0147| 0.0479| 0.0025| 0.0210| 0.0245| 0.0393| 0.0337| 0.0000  | 0.0092 | 0.0194 | 0.0089 | 0.0323 | 0.0098 | 0.0154 | 0.0362  | 0.0004  | 0.0032  |
| chn 11| 0.0225| 0.0153| 0.0361| 0.0621| 0.0233| 0.0453| 0.0261| 0.0223| 0.0294| 0.0109  | 0.0000| 0.0171 | 0.0218 | 0.0649 | 0.0170 | 0.1933 | 0.2000  | 0.0008  | 0.0052  |
| chn 12| 0.0466| 0.0156| 0.0533| 0.0490| 0.0089| 0.0493| 0.0229| 0.0399| 0.0266| 0.0115  | 0.0243| 0.0000| 0.0311| 0.0414| 0.0148| 0.0159| 0.0163| 0.0008  | 0.0038  |
| chn 13| 0.1449| 0.0323| 0.0563| 0.1560| 0.0371| 0.0631| 0.0852| 0.0521| 0.0157| 0.1488  | 0.0220| 0.2170| 0.0000| 0.0493| 0.0064| 1.4360| 0.0736| 0.0012  | 0.0356  |
| chn 14| 0.1615| 0.0373| 0.0411| 0.1989| 0.0191| 0.0809| 0.0939| 0.0539| 0.0236| 0.1768  | 0.0144| 2.8633| 0.0039| 0.0000| 0.0036| 0.1028| 0.0901| 0.0009  | 0.0435  |
| chn 15| 0.1441| 0.0284| 0.0322| 0.1531| 0.0174| 0.0467| 0.0697| 0.0413| 0.0149| 0.1524  | 0.0120| 0.2334| 0.0031| 0.0143| 0.0000| 0.0867| 0.0710| 0.0007  | 0.0367  |
| chn 16| 0.0762| 0.0113| 0.0624| 0.0660| 0.0120| 0.0423| 0.0351| 0.0132| 0.0292| 0.0301  | 0.0143| 0.0291| 0.0274| 0.0461| 0.0144| 0.0000| 0.2853| 0.0009  | 0.0056  |
| chn 17| 2.8666| 0.3542| 0.5717| 0.8129| 0.1832| 0.7548| 0.3287| 0.2925| 0.2491| 0.2695  | 4.4834| 2.0434| 0.4798| 0.6543| 0.2524| 0.3775| 0.0005  | 0.0067  | 0.0597  |
Table 2: Error square from each channel used as the inverse of the covariance for each channel (unit: \((10^{-5})\) pu)

<table>
<thead>
<tr>
<th>chn</th>
<th>value</th>
<th>chn</th>
<th>value</th>
<th>chn</th>
<th>value</th>
<th>chn</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0550</td>
<td>6</td>
<td>0.1614</td>
<td>11</td>
<td>0.0881</td>
<td>16</td>
<td>0.1231</td>
</tr>
<tr>
<td>2</td>
<td>0.0665</td>
<td>7</td>
<td>0.0744</td>
<td>12</td>
<td>0.0967</td>
<td>17</td>
<td>0.1165</td>
</tr>
<tr>
<td>3</td>
<td>0.1852</td>
<td>8</td>
<td>0.1851</td>
<td>13</td>
<td>0.0935</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.0230</td>
<td>9</td>
<td>0.2335</td>
<td>14</td>
<td>0.0431</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.0538</td>
<td>10</td>
<td>0.1151</td>
<td>15</td>
<td>0.1516</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: The sum of the error squares (unit \((10^{-5})\)) between the reconstructed signal and the original signal.

<table>
<thead>
<tr>
<th>chn 1</th>
<th>Single-channel</th>
<th>0.0550</th>
<th>0.1225</th>
<th>0.1225</th>
</tr>
</thead>
<tbody>
<tr>
<td>chn 2</td>
<td></td>
<td>0.0665</td>
<td>0.1230</td>
<td>0.1230</td>
</tr>
<tr>
<td>chn 3</td>
<td></td>
<td>0.1852</td>
<td>0.1008</td>
<td>0.1008</td>
</tr>
<tr>
<td>chn 4</td>
<td></td>
<td>0.0230</td>
<td>0.1548</td>
<td>0.1548</td>
</tr>
<tr>
<td>chn 5</td>
<td></td>
<td>0.0538</td>
<td>0.1179</td>
<td>0.1179</td>
</tr>
<tr>
<td>chn 6</td>
<td></td>
<td>0.1614</td>
<td>0.1012</td>
<td>0.1012</td>
</tr>
<tr>
<td>chn 7</td>
<td></td>
<td>0.0744</td>
<td>0.1398</td>
<td>0.1398</td>
</tr>
<tr>
<td>chn 8</td>
<td></td>
<td>0.1851</td>
<td>0.1400</td>
<td>0.1400</td>
</tr>
<tr>
<td>chn 9</td>
<td></td>
<td>0.2335</td>
<td>0.1041</td>
<td>0.1041</td>
</tr>
<tr>
<td>chn 10</td>
<td></td>
<td>0.1151</td>
<td>0.1225</td>
<td>0.1225</td>
</tr>
<tr>
<td>chn 11</td>
<td></td>
<td>0.0881</td>
<td>0.1229</td>
<td>0.1229</td>
</tr>
<tr>
<td>chn 12</td>
<td></td>
<td>0.0967</td>
<td>0.1199</td>
<td>0.1199</td>
</tr>
<tr>
<td>chn 13</td>
<td></td>
<td>0.0935</td>
<td>0.1119</td>
<td>0.1119</td>
</tr>
<tr>
<td>chn 14</td>
<td></td>
<td>0.0431</td>
<td>0.1184</td>
<td>0.1184</td>
</tr>
<tr>
<td>chn 15</td>
<td></td>
<td>0.1516</td>
<td>0.1128</td>
<td>0.1128</td>
</tr>
<tr>
<td>chn 16</td>
<td></td>
<td>0.1231</td>
<td>0.1300</td>
<td>0.1300</td>
</tr>
<tr>
<td>chn 17</td>
<td></td>
<td>0.1165</td>
<td>0.1385</td>
<td>0.1385</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>chn 1</th>
<th>multi-channel</th>
<th>0.1225</th>
<th>0.1225</th>
<th>0.0419</th>
<th>0.0161</th>
</tr>
</thead>
<tbody>
<tr>
<td>chn 2</td>
<td></td>
<td>0.1230</td>
<td>0.1230</td>
<td>0.0360</td>
<td>0.0113</td>
</tr>
<tr>
<td>chn 3</td>
<td></td>
<td>0.1008</td>
<td>0.1008</td>
<td>0.0380</td>
<td>0.0219</td>
</tr>
<tr>
<td>chn 4</td>
<td></td>
<td>0.1548</td>
<td>0.1548</td>
<td>0.0644</td>
<td>0.0371</td>
</tr>
<tr>
<td>chn 5</td>
<td></td>
<td>0.1179</td>
<td>0.1179</td>
<td>0.0420</td>
<td>0.0176</td>
</tr>
<tr>
<td>chn 6</td>
<td></td>
<td>0.1012</td>
<td>0.1012</td>
<td>0.0379</td>
<td>0.0219</td>
</tr>
<tr>
<td>chn 7</td>
<td></td>
<td>0.1398</td>
<td>0.1398</td>
<td>0.0435</td>
<td>0.0181</td>
</tr>
<tr>
<td>chn 8</td>
<td></td>
<td>0.1400</td>
<td>0.1400</td>
<td>0.0405</td>
<td>0.0151</td>
</tr>
<tr>
<td>chn 9</td>
<td></td>
<td>0.1041</td>
<td>0.1041</td>
<td>0.0422</td>
<td>0.0297</td>
</tr>
<tr>
<td>chn 10</td>
<td></td>
<td>0.1225</td>
<td>0.1225</td>
<td>0.0448</td>
<td>0.0181</td>
</tr>
<tr>
<td>chn 11</td>
<td></td>
<td>0.1229</td>
<td>0.1229</td>
<td>0.0385</td>
<td>0.0120</td>
</tr>
<tr>
<td>chn 12</td>
<td></td>
<td>0.1199</td>
<td>0.1199</td>
<td>0.0386</td>
<td>0.0120</td>
</tr>
<tr>
<td>chn 13</td>
<td></td>
<td>0.1119</td>
<td>0.1119</td>
<td>0.0479</td>
<td>0.0318</td>
</tr>
<tr>
<td>chn 14</td>
<td></td>
<td>0.1184</td>
<td>0.1184</td>
<td>0.0562</td>
<td>0.0378</td>
</tr>
<tr>
<td>chn 15</td>
<td></td>
<td>0.1128</td>
<td>0.1128</td>
<td>0.0495</td>
<td>0.0329</td>
</tr>
<tr>
<td>chn 16</td>
<td></td>
<td>0.1300</td>
<td>0.1300</td>
<td>0.0419</td>
<td>0.0169</td>
</tr>
<tr>
<td>chn 17</td>
<td></td>
<td>0.1385</td>
<td>0.1385</td>
<td>0.0689</td>
<td>0.0352</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>chn 1</th>
<th>data fusion (same covariance)</th>
<th>0.0419</th>
<th>0.0161</th>
</tr>
</thead>
<tbody>
<tr>
<td>chn 2</td>
<td></td>
<td>0.0360</td>
<td>0.0113</td>
</tr>
<tr>
<td>chn 3</td>
<td></td>
<td>0.0380</td>
<td>0.0219</td>
</tr>
<tr>
<td>chn 4</td>
<td></td>
<td>0.0644</td>
<td>0.0371</td>
</tr>
<tr>
<td>chn 5</td>
<td></td>
<td>0.0420</td>
<td>0.0176</td>
</tr>
<tr>
<td>chn 6</td>
<td></td>
<td>0.0379</td>
<td>0.0219</td>
</tr>
<tr>
<td>chn 7</td>
<td></td>
<td>0.0435</td>
<td>0.0181</td>
</tr>
<tr>
<td>chn 8</td>
<td></td>
<td>0.0405</td>
<td>0.0151</td>
</tr>
<tr>
<td>chn 9</td>
<td></td>
<td>0.0422</td>
<td>0.0297</td>
</tr>
<tr>
<td>chn 10</td>
<td></td>
<td>0.0448</td>
<td>0.0181</td>
</tr>
<tr>
<td>chn 11</td>
<td></td>
<td>0.0385</td>
<td>0.0120</td>
</tr>
<tr>
<td>chn 12</td>
<td></td>
<td>0.0386</td>
<td>0.0120</td>
</tr>
<tr>
<td>chn 13</td>
<td></td>
<td>0.0479</td>
<td>0.0318</td>
</tr>
<tr>
<td>chn 14</td>
<td></td>
<td>0.0562</td>
<td>0.0378</td>
</tr>
<tr>
<td>chn 15</td>
<td></td>
<td>0.0495</td>
<td>0.0329</td>
</tr>
<tr>
<td>chn 16</td>
<td></td>
<td>0.0419</td>
<td>0.0169</td>
</tr>
<tr>
<td>chn 17</td>
<td></td>
<td>0.0689</td>
<td>0.0352</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>chn 1</th>
<th>data fusion (different covariances)</th>
<th>0.0161</th>
</tr>
</thead>
<tbody>
<tr>
<td>chn 2</td>
<td></td>
<td>0.0113</td>
</tr>
<tr>
<td>chn 3</td>
<td></td>
<td>0.0219</td>
</tr>
<tr>
<td>chn 4</td>
<td></td>
<td>0.0371</td>
</tr>
<tr>
<td>chn 5</td>
<td></td>
<td>0.0176</td>
</tr>
<tr>
<td>chn 6</td>
<td></td>
<td>0.0219</td>
</tr>
<tr>
<td>chn 7</td>
<td></td>
<td>0.0181</td>
</tr>
<tr>
<td>chn 8</td>
<td></td>
<td>0.0151</td>
</tr>
<tr>
<td>chn 9</td>
<td></td>
<td>0.0297</td>
</tr>
<tr>
<td>chn 10</td>
<td></td>
<td>0.0181</td>
</tr>
<tr>
<td>chn 11</td>
<td></td>
<td>0.0120</td>
</tr>
<tr>
<td>chn 12</td>
<td></td>
<td>0.0120</td>
</tr>
<tr>
<td>chn 13</td>
<td></td>
<td>0.0318</td>
</tr>
<tr>
<td>chn 14</td>
<td></td>
<td>0.0378</td>
</tr>
<tr>
<td>chn 15</td>
<td></td>
<td>0.0329</td>
</tr>
<tr>
<td>chn 16</td>
<td></td>
<td>0.0169</td>
</tr>
<tr>
<td>chn 17</td>
<td></td>
<td>0.0352</td>
</tr>
</tbody>
</table>

\[\sum\text{ n.a.} \quad 2.081 \quad 0.773 \quad 0.385\]

Figure 2: Data fusion procedure and required communication links (in red). For example, PMU2 sends data to PMU1 and PMU1 carries out the fusion procedure.
Figure 3: Frequency deviation measurements.

Figure 4: Reconstructed signals after individual Prony analysis.

Figure 5: Reconstructed signals using the parameter vector from multiple channel Prony analysis. The reconstructed signals match poorly with the PMU measurements presented in Fig. 3.
Data fusion with the assumption of same covariance for every channel

Figure 6: Reconstructed signals using the averaged parameter vector.

Figure 7: Reconstructed signals after data fusion.

Figure 8: Comparison of measurements and reconstructed signals of Channel 3.