International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.8, No.8 (2015), pp.135-142
http://dx.doi.org/10.14257/ijsip.2015.8.8.15

Implementation of the Parallel Algorithm of the Cross Power Spectral Density of Random Signals

Qi Xiong[1], Nanhui Chen[2] and Meisen Pan[1]

1. National Linux Technology Training & Development Center, Hunan University of Arts and Science, Changde, China
2. Kunming Institute of Zoology, the Chinese Academy of Sciences, Kunming, China
abcxq@126.com

Abstract

Power spectral density (PSD) is an effective way to analyze random signals. However, it is a time-consuming calculation when it deals with a rapidly growing massive data by using the serial Matlab circumstances. Though a parallel computing toolbox is involved in Matlab 2004, it is too expensive to be wildly used. According to an algorithm of PSD, a parallel algorithm of cross power spectral density (PACPSD), which was implemented using the Master-Slave parallel programming model with the support of Linux clusters and MPI, was proposed here based on the theory of Welch algorithm. The experiment results reveal that PACPSD can achieve the same accuracy as the function cpsd in Matlab do, and greatly reduce the operation cost.

Keywords: *cross power spectral density; Welch algorithm; message passing interface*

1. Introduction

Power spectral density (PSD) analysis, as an effective method for tackling random signals, is widely used in the areas of communication, geological survey, radar, sonar and biomedical engineering, etc. Since the Fourier transform of random signals does not exist, frequency spectrum analysis on a random signal is usually conducted by solving its power spectrum. Because the frequency spectrum can represent the distribution of power of signals and identify the frequencies of useful signals, it is often employed to indicate effective information such as implicit periodicity in the signal and spectral peaks close to each other [1].

The power spectrum that represents the statistical distribution of energy of frequency components of a signal is called its auto-power spectrum, while the power spectrum density that represents the statistical distribution of energy of two signals is their cross power spectrum density (CPSD). Literature[2-3] illustrate the application of CPSD in practice. There are several types of classification for methods of spectrum estimate. According to the history of development of power spectrums, methods of spectral estimate can be classified into classic spectral estimate methods and modern spectral estimate methods.

In general, classic spectral estimate methods based on traditional Fourier transform include periodogram method and BT method. As modern spectrum estimate methods based on signal models, they include AR, MA and ARMA model methods. The periodogram method, one of classic spectral estimate methods, is widely used for spectral estimates since it has a clear physical concepts, ease to implement and high computation efficiency. However, because it does not satisfy the consistency condition for estimate, the conflict between resolution and variance makes it difficult for the periodogram method to perform satisfactorily. To improve its performance, lots of upgraded algorithms have been proposed upon the periodogram method. Among them, Welch algorithm is a famous and commonly-used one. Through segmentation and windowing on data, Welch algorithm becomes an effective spectral estimate method by effectively reducing the variance of the spectral estimate while avoiding severe damage of the resolution [4]. This paper reports the investigation of the implementation of PACPSD - a parallel algorithm of CPSD based on Welch algorithm in the parallel programming environment of MPI.

2. CPSD Algorithm and its Serial Implementation

2.1 CPSD Algorithm

Welch's spectral estimate algorithm improves the periodogram from two aspects. On one hand, neighboring segments of data are allowed to overlap partly when the input sequence of signal is split-up. On the other hand, the window function applied to each data segment can be chosen from various options and is not necessarily the conventional rectangular window [5]. The flow chart for Welch's spectral estimate algorithm is shown in Figure 1.

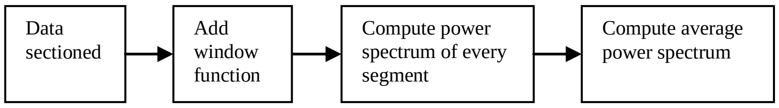

Data sectioned
Add window function
Compute power spectrum of every segment
Compute average power spectrum

Figure 1. Flow Chart for Welch's Spectral Estimate Algorithm

Major steps of the CPSD algorithm based on Welch algorithm are as follows:

(1) The sampling sequence $X_N(n)$, $Y_N(n)$, n=0,1,…, N-1 are split up into overlapping segments and let the length of each data segment be L. Beginning from the second (i=1) segment, the first D sample points of the ith segment are overlapped with the last D points of the (i-1)th segment. For example, the ith segment of the sequence $X_N(n)$ is mathematically expressed as:

$$x_i = \begin{cases} X_N[i(L-D)+n] & n=0,1,\cdots,L-1; i=0,1,\cdots,K-1 \\ 0 & others \end{cases}$$

The following equation shows the relationship among the sampling length N, number of overlapping points D, number of segments K and segmental length L.

$$\mathrm{N} = \mathrm{L} + (\mathrm{L} - \mathrm{D}) \times (\mathrm{K} - 1)$$

where $\mathrm{K} = \dfrac{N-L}{L-D} + 1$

(2) Apply a smooth window (usually Hamming window) w(n) to each segment of data, and, obtain $X_i(N)$ and $Y_i(N)$ through Fourier transform.

(3) Compute the cross power spectral values of two signals in the current segment: $P_{xy} = X_i^*(N).\times Y_i(N)$, where $X_i^*(N)$ is the complex conjugate of $X_i(N)$.

(4) Add and average the cross powers of all segments and obtain the CPSD values:

$$S_x(e^{j\omega}) = \frac{1}{K}\sum_{i=1}^{K}\frac{1}{LU}P_{xy}$$

here $U = \dfrac{1}{L}\sum_{n=0}^{L-1} w^2(n)$, it indicates the mean power of the window function. In addition, $LU = \sum_{n=0}^{L-1} w^2(n)$ expresses the energy of the window function w(n) with length L[6].

2.2 Serial Implementation of the CPSD Algorithm

The serial implementation of the SPSD algorithm based on Welch algorithm can be described as follows:

Input: sample data x(n), y(n), data length x_len, segment length seg_len, and sampling rage Fs.

Computing:

```
(1) overlap of neighboring segments (usually half of seg_len) and number of
    segments n_ffts
(2) for i = 0 to seg_len -1
begin loop
        Generate Hamming window win[i];
        Calculate the accumulated value of u: u += win[i]*win[i];
        i++;
    end loop.
(3) for start_seg=1 to x_len-seg_len+1
begin loop
        end_seg=start_seg+seg_len-1;
        apply Hamming window to the data of x(n), y(n) between
    [start_seg,end_seg];
        do Fourier transform onto windowed data and obtain fft_x, fft_y;
        pgram =  fft_y .*conj(fft_x);
        //calculate the cross power spectrum of these two signals in current
    segment .conj(fft_x) means the complex conjugate of fft_x.
        $P_{xy}$ +=  pgram;  // summate power spectra
        start_seg += seg_len-overlap;
    end for loop
  (4)Output: power spectrum estimates: Pxy/( n_ffts×Fs×u)
```

In the above algorithm, start_seg and end_seg are starting point and end point, respectively.

3. Parallel Algorithm of the CPSD

3.1 Program Structure of PAcpsd

From the steps of Welch spectrum estimate algorithm in 2.1, it can be seen that this algorithm is easy to be implemented in parallel. In this paper, the parallel algorithm uses the Master-Slave structure. After the master processor reads original sample data, number of segments n_ffts is determined from the overall length of data x_len, segmental length seg_len and overlap the overlapping length of neighboring segments. Then the master processor distributes equally each segment of data into every slave processing node and step (2, 3) is conducted by each slave processing node. Later on, the master processor collects all results given by the slave processors and finishes step (4). So far, the resultant cross power spectrum can be obtained. The program structure is demonstrated as Figure 2. Since the parallel implementation of fast Fourier transform which is involved in step (2) has been maturely developed, we need not give relevant details. Some references are available [7].

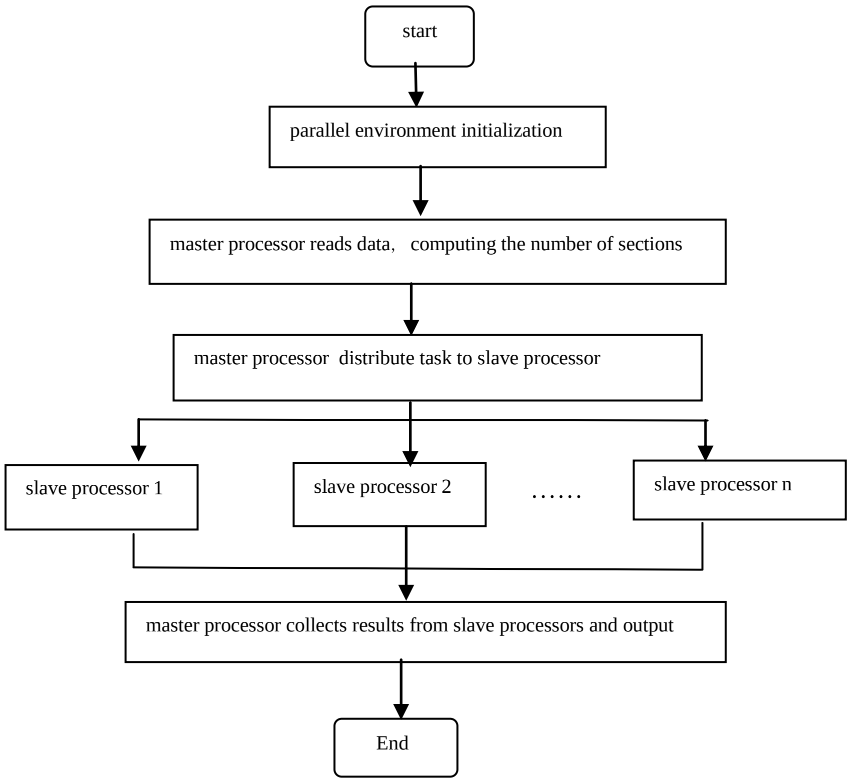

start
parallel environment initialization
master processor reads data, computing the number of sections
master processor distribute task to slave processor
slave processor 1
slave processor 2
……
slave processor n
master processor collects results from slave processors and output
End

Figure 2. Program Structure of PAcpsd

3.2 Distribution of Tasks

The method that calculates the work load per processor is: firstly calculate the least number of tasks per processor:

AvgNum = n_ffts / size

If the number of processors size cannot divide n_ffts without remainder, RNum=nffts mod size processors will be assigned AvgNum+1 processing tasks. Assume extra tasks are assigned to processors with small numbers, then the number of tasks that the processor numbered rank is assigned would be:

$$\text{MyNum}=\begin{cases}\text{AvgNum}+1 & \text{rank}<\text{RNum}\\ \text{AvgNum} & \text{other nodes}\end{cases}$$

Data range in which each processor is assigned is

startPos = i× (seg_len - overlap); stopPos = startPos + seg_len - 1;

where startPos is the start position of the sample data, stopPos is the stop position of the sample data, i is the index of segment, which lies in the range 0～n_ffts-1.seg_len is the length of each segment and overlap is the length of overlap.

4. Experimental Results and Analysis

4.1 Problem Description

For practical sample data, especially those in the area of EEG research, EEG signals show different power spectral characteristics for different status, e.g. sleeping and awake. Traditional data sampling for EEG employs only 20 leads. As the technology evolves, current EEG devices can include as many as 256 leads. From the point of view of data analysis, the more the leads are, the more the sample data become, i.e., more time is needed for the analysis. Specifically, spectral analysis in time domain has approached the upper limit of single-CPU computation. As a result, it is necessary to obtain results of data analysis more quickly through parallel computing of multiple CPUs.

This paper adopts milti-channel EEG data (sampling rate 1000Hz and band-pass filtering 0.5~90Hz) to test the algorithm.

4.2 Experimental Environment and Evaluation Standard

The experiment was conducted on the high-performance cluster servers of National Linux Technology Training and Development Center in Hunan University of Arts and Science. This cluster uses Lenovo DeepComp1800 with 8 computing nodes, 1 control node and 1 storage node. Each computing node has 2 Intel Xeon 2.8GHz processors, a 2GB memory and a SCSI 74G hard disk. The software environment is RedHat Linux 9.0, MPICH. Matlab (R2011B, Mathworks Inc) on Windows XP is used as the comparison.

In the experiment, the speeding ratio Speedup is employed to evaluate the time performance of PAcpsd. The formula to compute Speedup is: Speedup= Ts/ Tp, where Ts is the time of serial operations and Tp is the time of parallel operations.

4.3 Analysis of Experimental Results

Cross power spectrum computation of the test data was conducted on the control platform by adopting the cpsd function, *i.e.* cpsd(a,b,hamming(256),128,256,1000) and the results are shown in Figure 3, where the cpsd parameters mean that sample data a, b are split up into segments, each segment has 256 data points, the overlap of neighboring segments is 128, Hamming window is applied, nfft is 256 and the sampling rate is 1000Hz. Results of PAcpsd running on the cluster are shown in Figure 4.

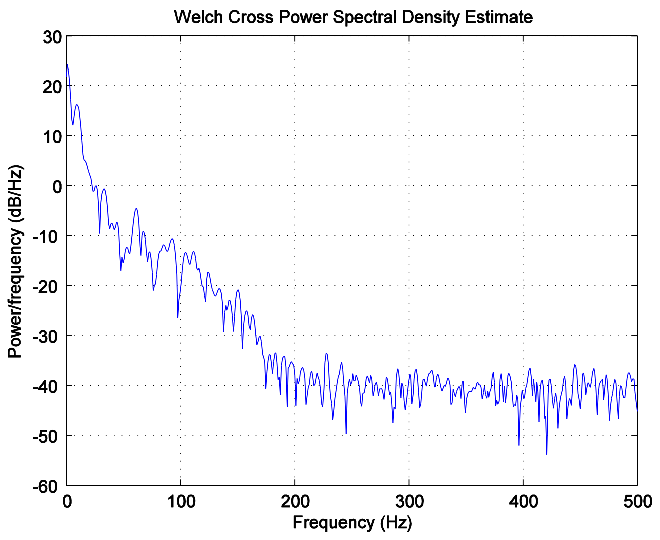


Figure 3. Result of cpsd in Mtlab

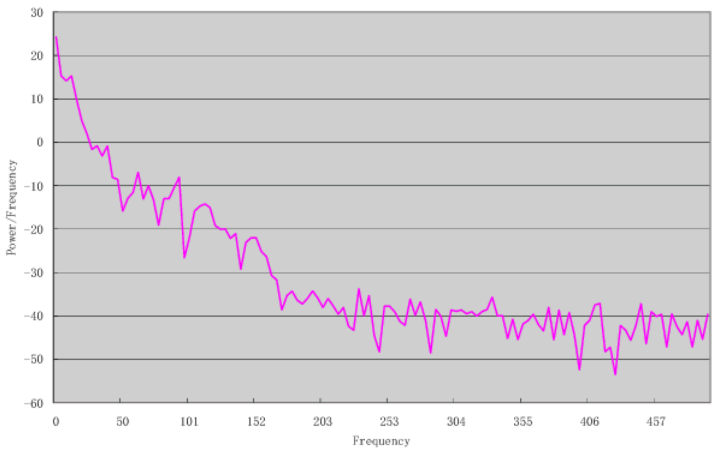


Figure 4. Result of PAcpsd

Comparison of Figure 3 and Figure 4 indicates that the results given by PAcpsd and those provided by Matlab are consistent, which proves that correctness of PAcpsd. To verify the time performance of PAcpsd, computations were conducted on different amounts of processors and the elapsed times for all cases are listed in Table 1.

Table 1. Different Amounts of Processors and the Elapsed Times

Size	serial	2	3	4	5	6	7	8
elapsed times (s)	0.098	0.057	0.04	0.029	0.023	0.020	0.017	0.014

The Speedups are shown in Figure 5, which indicates that as the number of processors increases, the Speedup of PAcpsd grows basically in a linear manner. So that PAcpsd is proven to be effective in reducing running time.

5. Conclusion

This paper proposed PAcpsd - a parallel algorithm of cross power spectral density based on Welch's algorithm. With the support of MPI, the Master-Slave parallel

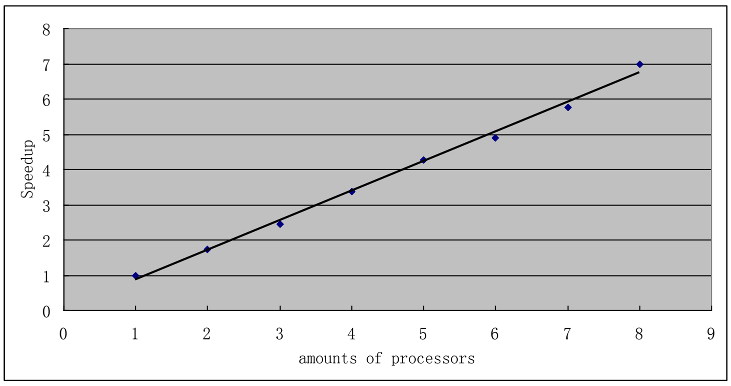


Figure 5. Speedup of PAcpsd

mechanism is employed to implement the algorithm and guarantee its time performance. Computing results in Matlab are basically identical with those run by PAcpsd for the same sample data, which shows the correctness of PAcpsd. Experimental results of PAcpsd running on different amounts of processors indicate that PAcpsd effectively reduced running time while yielding correct results. The prototype of PAcpsd proposed by this paper is completely based on open-source software and advantageous with low cost and high efficiency.

Acknowledgements

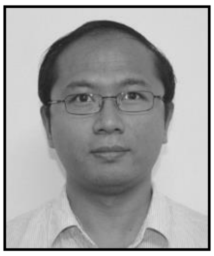
This work was financially partly supported by the Natural Science Foundation of China (61178051), supported by National Key Technology Research and Development Program of the Ministry of Science and Technology of China (NO.2011BAK04B04), supported by Natural Science Foundation of Hunan Province of China (12JJ9022), and supported by the Construct Program of the key Discipline in Hunan University of Arts and Science.

References

[1] F. Du, L. Tang and J.-q. Ding, "Improvement and application of PSD and PWELCH [J]", China Measurement & Test, vol. 36, no. 1, (**2010**), pp. 93-96.
[2] A. Sarkar and C. S. Manohar, "Critical Cross Power Spectral Density Functions and the Highest Response of Multi-Supported Structures Subjected to Multi-component Earthquake Excitations [J]", Earthquake Engineering and Structural Dynamics, vol. 25, (**1996**), pp. 303-315.
[3] M. Rahmani, A. Akbari, B. Ayad and B. Lithgow, "Noise cross PSD estimation using phase information in diffuse noise field [J]", Signal Processing, vol. 89m, no. 5, (**2009**), pp. 703-709.
[4] P. D. Welch, "The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms [J]", IEEE Transactions On Audio And Electroacoustics, vol. 15, no. 2, (**1967**), pp. 70-73.
[5] P. Stoica and R. Moses, "Spectral Analysis of Signals", Prentice Hall, New Jersey, (**2005).**
[6] Z. Shen, L. Wang and F. Chen, "Research on Spectral Estimation Using Welch Method Based on Analysis of Finite-length Sequence [J]", Computer Simulation, vol. 27, no. 12, (**2010**), vol. 27, no. 12, (**2010**), pp. 391-395
[7] G. Chen, "Parallel Algorithm Practice", Higher Education Press, Beijing, (**2004).**

Authors

Qi Xiong, he is senior engineer in the Hunan University of Arts and Science, PRC. He received his M.S. degree from Huazhong University of Science and Technology, China, in 2005. He has published more than 10 papers in journals and conferences. His research interests include parallel computing and computer networks.



Nanhui Chen, he is an associate professor in Kunming Institute of Zoology, Chinese Academy of Sciences. He got his Bachelor Degree of Medicine in Hunan Medical University, China in 1993. In 1998 he got his Ph.D. degree in Shanghai Brain Research Institute, China. From 1998 to 2000 he did the postdoc work in LSN/NIMH, USA. Since 2000 he did research work in KunMing Institute of Zoology in China. His research interests include neural information processing and parallel computing

Meisen Pan, he is a professor in the Hunan University of Arts and Science, PRC. He received his B.Sc. degree from Hunan Normal University, China, in 1995, the M.S. degree from Huazhong University of Science and Technology, China, in 2005, and the Ph.D. degree from Central South University, China, in 2011. He has published more than 40 papers in journals and conferences. His research interests include biomedical image processing, information fusion, and software engineering.