Dmitry KAPLUN, Dmitry KLIONSKIY, Alexander VOZNESENSKIY, Vyacheslav GULVANSKIY

Saint Petersburg Electrotechnical University "LETI"

doi:10.15199/48.2015.02.12

Digital filter bank implementation in hydroacoustic monitoring tasks

Abstract. *The paper discusses digital filter bank implementation using the weighted overlap-add (WOLA) algorithm modification introduced in the context of multichannel signal processing. The suggested modification is applied to hydroacoustic monitoring tasks. High-performance software-hardware implementations on the basis of CPU (Central Processing Unit) and CUDA (Compute Unified Device Architecture) are compared and CUDA is shown to provide advantages for the filter banks studied.*

Streszczenie. *W artykule przedstawiono implementację banku filtrów cyfrowych opartych na algorytmie Weighted Overlap-Add (WOLA), zmodyfikowanego w kontekście przetwarzania sygnału wielokanałowego. Zaproponowane modyfikacje zostały zastosowane w hydroakustycznych zadaniach monitorowania. Dokonano porównania wysokowydajnych programowo-sprzętowych implementacji opartych na CPU (Central Processing Unit) oraz CUDA (Compute i Unified Device Architecture). Badania wykonane dla zdefiniowanego banku filtrów pokazały, że najbardziej korzystne jest zastosowanie CUDA. (**Implementacja banku filtrów w hydroakustycznych zadaniach monitorowania**).*

Keywords: hydroacoustic monitoring, filter bank, WOLA-algorithm, multichannel signal processing, software-hardware implementation.
Słowa kluczowe: monitorowanie hydroakustyczne, algorytm WOLA, przetwarzanie sygnału wielokanałowego.

Introduction

The term "monitoring" means a systematic or continuous data acquisition from some object or system [1]. Monitoring is often performed in a wide frequency band (wideband monitoring) and includes tracking the main system parameters and deviation search in these parameters. Any deviation is usually caused by abnormal functioning of an object and requires immediate action in order to prevent failures.

Wideband monitoring is frequently used in the following areas [1]:

- *hydroacoustics* – monitoring of water areas (coastal and marine waters), underwater and surface objects, study of navigation canals and near-shore waters, analysis of oceanic seismicity, etc.;
- *radio monitoring* – time-frequency processing of radiations, noise suppression, signal classification, signal parameter estimation, signal demodulation, direction-finding, etc.;
- *vibration analysis* – vibration measurements, vibration parameter estimation in the time and frequency domains, resonance frequency estimation; analysis of vibrations is important for spacecraft, aircraft, engines, turbines, machinery, etc.;
- *geophysics* – monitoring of seismic and geomagnetic activity in various regions of our planet.

The present paper is devoted to considering hydroacoustic monitoring tasks.

Hydroacoustic monitoring

The shelf regions of the world's oceans contain the vast majority of our planet's natural resources. In order to best develop these resources, it is necessary to create monitoring systems for the earth's water environment, which includes all the regions where human activity and impact are very high.

Fast estimation of water environment dynamics and the contemporary concept of hydroacoustic system construction are made possible by the multipurpose use of satellite and ground truth observation technologies. The creation of regional systems for such kinds of observation can be based on conventional ways of hydrologic and ecologic monitoring, but this might prove to be very expensive. At present above-ground remote methods of gathering information on water environment parameters have also become quite widespread. Among other promising observation techniques we can mention, for example, radar observations, which provide on-line information collection (wave characteristics and contamination degree of the water area under study).

One of the possibilities of reducing the cost of ground truth observations for ocean regions is multiple use of water environment observation techniques with the help of distant monitoring systems.

Furthermore, it is necessary to improve the existing mathematical models of hydroacoustic signals based on signal shaping in shelf regions. Besides, mathematical techniques employed for hydroacoustic information processing and interpretation have to be enhanced as well. Finally, key principles of hydroacoustic system construction have to be further developed. Mathematical tools for hydroacoustic information processing and analysis should be based on hydroacoustic signal generation techniques in shelf regions and time-spatial variability of acoustic fields in the water environment [2]. For this reason, development of mathematical tools should be followed by oceanological investigations.

Hydroacoustic monitoring systems

Among the main tasks of hydroacoustic monitoring are the ones listed below [3]:

- continuous observation of the underwater environment in various modes (active sonar, passive sonar, etc.);
- determination of an object's location in marine and riverine areas.

These tasks are normally solved by multipurpose hydroacoustic monitoring systems, which are equipped with a satellite-borne hydroacoustic antenna consisting of separate universal hydroacoustic antenna modules. Each module includes the following components:

- acoustic receivers;
- analog-to-digital module of signal preprocessing;
- digital signal generation block.

Antenna modules are intended for signal reception, analog signal preprocessing, and digital signal transmission to a computing system, where further processing is performed. Computer systems perform integrated multichannel signal processing, control, and diagnostics of the hydroacoustic monitoring system, and also visualization. All signal processing algorithms can be arranged the following way:

- stage 1 – spatiotemporal analysis, spectral analysis, adaptive signal processing, and signal detection;

• stage 2 – target coordinate detection, trajectory construction, and data classification;

• stage 3 – integrating underwater observation results from other systems.

Application of these algorithms also provides data output and data transmission to other systems for further processing.

WOLA-algorithm

Hydroacoustic monitoring tasks are often carried out using digital *FIR filters* [3-5] designed by the window or Chebyshev method [3-5]. Digital filters are also needed for *multichannel signal processing* [5], when it is necessary to control large water areas and reduce the amount of time allocated for signal analysis. One of the approaches to multichannel signal processing can be implemented by means of *digital filter banks* [3-8] with the help of high-performance software-hardware tools.

The present paper is devoted to the development of a WOLA modification for processing multichannel hydroacoustic signals.

Digital filter banks and their implementations have been studied for many years and DFT-modulated filter banks [3-5] have long been applied to various tasks related to multirate signal processing and wideband monitoring. Among the most significant achievements are the development of the polyphase form of a filter bank [6] and also the one-dimensional WOLA-algorithm [5].

DFT-modulated filter banks have several implementations. One of the simplest is a filter bank with complex modulation [5]. However, this implementation is not efficient since linear convolution in each channel is calculated for a large sampling rate (this leads to high computational costs). Therefore it is necessary to reduce the sampling rate, for instance, by choosing the polyphase structure [5].

One of the ways of implementing a polyphase filter bank is the WOLA algorithm, which performs *block-by-block* analysis [5]. Like a filter bank with complex modulation, the output signal is determined as

$$X_k(m) = \sum_{n=-\infty}^{\infty} h(mM-n)x(n)W_K^{-kn}, \tag{1}$$

where k is the number of a filter bank channel, m is the block number in WOLA (to be defined), $h(n)$ is the impulse response of a LF-prototype (low-frequency prototype) filter, M is the decimating factor of the input signal,

$$W_K^{-kn} = e^{-j\left[(2\pi k/K)n\right]}, \tag{2}$$

where $j=\sqrt{-1}$.

According to (1) it is possible to introduce a filter with the impulse response $h(mM-n)$ as a moving analysis window for extracting the sequence $y_m(n)$, which is then subjected to short-time Fourier transform (STFT). This is the key idea behind the algorithm. In the case of critical decimation, when $M=K$, WOLA is identical to a polyphase filter bank. The main difference is that WOLA is oriented to block-by-block analysis (rather than parallel processing) and the signal in question has to be split into sections of the length N_h and overlap equal to M samples.

The main factor of the filter bank implementation that affects computational complexity is the LF-prototype filter. The magnitude response of the filter coincides with the desired one for a single channel. LF-prototype order depends on a required channel bandwidth, rectangularity shape factor, and gain flatness in the passband of one single channel.

Thus, we have to create efficient FIR-filters using the minimal computational complexity criterion for reducing hardware and software costs.

The algorithm outlined above is intended for processing *one-dimensional signals* (the term "one-dimensional signal" is used as a synonym for "single-channel signal") presented in the form of a vector. In practice the original signal can consist of several one-dimensional signals (multichannel signal).

The input of the developed algorithm is as follows:

1) $\mathbf{x}(n)=\left[\mathbf{x}_0(n), \mathbf{x}_1(n), \dots, \mathbf{x}_{S-1}(n)\right]$ is a multichannel signal in the matrix form, where the sub-matrices $\mathbf{x}_0(n), \mathbf{x}_1(n), \dots, \mathbf{x}_{S-1}(n)$ can be presented as

$$\mathbf{x}_0=\begin{bmatrix} x_0(0) \\ x_0(1) \\ \vdots \\ x_0(N-1) \end{bmatrix}, \mathbf{x}_1=\begin{bmatrix} x_1(0) \\ x_1(1) \\ \vdots \\ x_1(N-1) \end{bmatrix}, \dots, \mathbf{x}_{S-1}=\begin{bmatrix} x_{S-1}(0) \\ x_{S-1}(1) \\ \vdots \\ x_{S-1}(N-1) \end{bmatrix}. \tag{3}$$

2) S is the size of a multichannel signal (number of one-dimensional signals).

3) M is the decimation factor, $M=1,\dots,K$, where K is the number of filter bank channels for each polyphase implementation corresponding to $x_i(n)$.

4) $h(n)$ is the impulse response of a LF-prototype filter (vector of size $1\times N_h$).

After obtaining a multichannel signal, weighting has to be applied:

$$\begin{aligned} &\tilde{x}_{mi}(n)=h(mM-n)x_i(n), \\ &i=0,\dots,S-1;\ n=0,\dots,N-1, \end{aligned} \tag{4}$$

where m is the block number (N_h is the block length). The total number of blocks P of the length N_h with the overlap $\left(N_h-M\right)$ is determined as

$$P=1+\left\lfloor \frac{N-N_h}{M} \right\rfloor, \tag{5}$$

where $\lfloor . \rfloor$ denotes rounding towards minus infinity. As a result of (4) we arrive at the matrix of weighted samples:

$$\tilde{\mathbf{x}}(n)=\left[\tilde{\mathbf{x}}_0(n), \tilde{\mathbf{x}}_1(n), \dots, \tilde{\mathbf{x}}_{S-1}(n)\right], \tag{6}$$

$$\tilde{\mathbf{x}}_0=\begin{bmatrix} \tilde{x}_0(0) \\ \tilde{x}_0(1) \\ \vdots \\ \tilde{x}_0(N-1) \end{bmatrix}, \tilde{\mathbf{x}}_1=\begin{bmatrix} \tilde{x}_1(0) \\ \tilde{x}_1(1) \\ \vdots \\ \tilde{x}_1(N-1) \end{bmatrix}, \dots, \tilde{\mathbf{x}}_{S-1}=\begin{bmatrix} \tilde{x}_{S-1}(0) \\ \tilde{x}_{S-1}(1) \\ \vdots \\ \tilde{x}_{S-1}(N-1) \end{bmatrix}. \tag{7}$$

After that the blocks of the length N_h are split into non-overlapping segments of the length K and we perform summation of these segments. The total number of segments Q of the length K each within the block of the length N_h is determined as $Q=\lceil N_h/K \rceil$, where $\lceil . \rceil$ denotes rounding towards infinity. As a result we obtain the following expressions:

$$\begin{aligned} &z_{0,m}(r)=\sum_{l=-\infty}^{\infty}\tilde{x}_{0,m}(r+lK),\ z_{1,m}(r)=\sum_{l=-\infty}^{\infty}\tilde{x}_{1,m}(r+lK), \\ &\dots,\ z_{S-1,m}(r)=\sum_{l=-\infty}^{\infty}\tilde{x}_{S-1,m}(r+lK), \end{aligned} \tag{8}$$

where $m = 0,...,P-1$, $r = 0,...,K-1$. The matrix $\mathbf{Z}$ can be written as

$$\mathbf{Z} = \begin{bmatrix} z_{00}(0) & z_{00}(1) & ... & z_{00}(K-1) \\ ... & ... & ... & ... \\ z_{0,P-1}(0) & z_{0,P-1}(1) & ... & z_{0,P-1}(K-1) \\ ... & ... & ... & ... \\ z_{S-1,0}(0) & z_{S-1,0}(1) & ... & z_{S-1,0}(K-1) \\ ... & ... & ... & ... \\ z_{S-1,P-1}(0) & z_{S-1,P-1}(1) & ... & z_{S-1,P-1}(K-1) \end{bmatrix}. \quad (9)$$

This matrix consists of sub-matrices and each one has P rows corresponding to blocks of the length N_h of each one-dimensional signal $x_i(n)$, $i = 0,...,S-1$. Thus, $\mathbf{Z}$ has the size $P \cdot S \times K$.

Next, DFT is applied to the matrix $\mathbf{Z}$. This operation can be implemented on the basis of vector DFT algorithms intended for calculating DFT of multichannel data:

$$\mathbf{Y} = VDFT\{\mathbf{Z}\}, \quad (10)$$

where $\mathbf{Y}$ is the resulting matrix after DFT computation, $VDFT$ is the operator of vector DFT.

There are two main approaches to vector DFT computation. One of them suggests application of one-dimensional DFT to each row of $\mathbf{Z}$. Overall, it is necessary to compute $P \cdot S$ one-dimensional DFTs of the length K each. The second approach is based on calculating one-dimensional DFT of the length $P \cdot S \cdot K$:

$$\mathbf{y} = DFT\begin{pmatrix} z_{00}(0) & z_{00}(1) & ... & z_{00}(K-1) \mid ... \\ ... \mid z_{S-1,P-1}(0) & z_{S-1,P-1}(1) & ... & z_{S-1,P-1}(K-1) \end{pmatrix}, \quad (11)$$

where DFT is the operator of one-dimensional DFT. In order to compute one-dimensional DFT, the matrix $\mathbf{Z}$ has to be converted into a vector (a sequence of matrix rows). There are several reasons for this transformation:

- an algorithm of vector DFT computation affects the total size of local memory of a DFT-processor. By reducing multichannel DFT to one-dimensional DFT it is possible to minimize computational memory and optimize the structure of a DFT-processor;
- computation of one-dimensional DFT of the length $P \cdot S \cdot K$ allows us to implement the computational procedure in a DFT-processor more efficiently in comparison with computing $P \cdot S$ different DFTs of the length K.

After computing DFT (11) of the vector $\mathbf{y}$ it is necessary to transform the result back into a matrix. The dimension of $\mathbf{Y}$ will coincide with that of $\mathbf{Z}$. Matrix $\mathbf{Y}$ will consist of several sub-matrices:

$$\mathbf{Y}(r) = \left[\mathbf{Y}_0(r), \mathbf{Y}_1(r), ..., \mathbf{Y}_{S-1}(r)\right]^T, \quad (12)$$

$$\mathbf{Y}_i(r) = \begin{bmatrix} y_{i,0}(0) & y_{i,0}(1) & ... & y_{i,0}(K-1) \\ & ... & & \\ y_{i,P-1}(0) & y_{i,P-1}(1) & ... & y_{i,P-1}(K-1) \end{bmatrix},$$

$$i = 0,...,S-1; \quad r = 0,...,K-1. \quad (13)$$

Finally, we have to form the matrix $\tilde{\mathbf{Y}}(r)$ by multiplying all the rows of $\mathbf{Y}(r)$ by the factor W_K^{-krM}:

$$\tilde{\mathbf{Y}}_i(r)\big|_{k-th\ line} = \mathbf{Y}_i(r)\big|_{k-th\ line} W_K^{-krM}, \quad r = 0,...,K-1;\ k = 0, ..., P-1. \quad (14)$$

To summarize, multichannel WOLA-algorithm can be presented as a sequence of steps:

1) weighting the multichannel signal $\mathbf{x}(n)$ in order to obtain $\tilde{\mathbf{x}}(n)$ according to (4)-(7).

2) splitting the multichannel signal (6) into non-overlapping segments of the length K and summation of segments of each one-dimensional signal according to (8);

3) application of vector DFT (10) to the matrix $\mathbf{Z}$ using one-dimensional DFT (11) of the length $P \cdot S \cdot K$ and computation of the matrix $\mathbf{Y}(r)$ according to (12)-(13);

4) computation of the matrix $\tilde{\mathbf{Y}}(r)$ according to (14).

The suggested modification of WOLA has a wider range of application than a multichannel polyphase filter bank. The polyphase filter bank is efficient in the case of critical decimation, while WOLA can be used no matter what the relation between M and K is.

Filter bank implementation

Filter banks are directed at parallel signal processing and therefore in order to improve their software-hardware implementation it is most reasonable to employ parallel structures. Among them are field-programmable gate arrays (FPGA) and processing devices based on CUDA [9]. CUDA implies using parallel graphical processors (GPU) for handling non-graphical tasks.

GPU is a coprocessor supplementing the functions of CPU; it also has its own memory and can execute a large number of separate threads in a parallel way.

The main advantages of CUDA are listed below:

1) compatibility with Windows, Linux, and Mac OS;

2) all software is developed using the extended C language (with additional tools for parallel programming and creating multiflow applications for CUDA);

3) no programmable interfaces (API) are used (such interfaces have a number of restrictions on creating multiflow applications).

Comparison of different software-hardware implementations was made for different forms of a DFT-modulated filter bank. The personal computer used for our experiments had the following configuration: CPU Intel Core i7-3630QM 2.4 GHz; RAM DDR3 16 Gb; OS Windows 7 64 bits; video card NVidia GeForce GT650M (384 core GPU, core frequency 850 MHz, video card memory 2 Gb).

In the tables below there are filter bank implementation results using CPU and CUDA. The *first* column contains data size (in samples and Mbytes), the *second* column ("CUDA without data transfer between RAM and video card memory") shows execution time for GPU without data transfer (from RAM to video card memory and backwards); the *third* column ("CUDA with data transfer between RAM and video card memory") gives the same information as the second column, but includes time for data transfer (from RAM to video card memory and backwards); the *fourth* column ("CPU") gives execution time for CPU. Regarding CUDA, Table 3 contains execution time only for the case when data transfer between RAM and video memory is taken into account.

The parameters of the filter bank and the prototype filter as follows: frequency range: [0…24] kHz; design

method for a LF-prototype filter: window method; bandwidth (one-sided): 94 Hz; LF-prototype filter order: 5141; shape factor: 1.24; signal suppression (on passband edge): 1 dB; signal suppression (on stopband edge): 120 dB; passband flatness: 0.5 dB.

The designed filter bank is intended for processing 5-channel signals ($S = 5$).

Table 1. Execution time (DFT-modulated filter bank with complex modulation)

Sample size (samples/ Mbytes)	Execution time for CUDA and CPU		
	CUDA without data transfer between RAM and video card memory	CUDA with data transfer between RAM and video card memory	CPU
3000000 / 11 Mb	1.44 sec.	1.66 sec.	260.542 sec.
8000000 / 30 Mb	3.76 sec.	4.33 sec.	680.000 sec.
16000000 / 61 Mb	6.75 sec.	7.90 sec.	1360.00 sec.
30000000 / 114 Mb	11.94 sec.	14.09 sec.	2550.00 sec.

Table 2. Execution time (polyphase structure and WOLA)

Sample size (samples/ Mbytes)	Execution time for CUDA and CPU		
	CUDA without data transfer between RAM and video card memory	CUDA with data transfer between RAM and video card memory	CPU
3000000 / 11 Mb	0.29 sec.	0.51 sec.	46.93 sec.
8000000 / 30 Mb	0.32 sec.	0.90 sec.	122.66 sec.
16000000 / 61 Mb	0.35 sec.	1.50 sec.	246.83 sec.
30000000 / 114 Mb	0.44 sec.	2.60 sec.	459.35 sec.

Table 3. Execution time (WOLA for the multichannel signal; $S = 5$)

Sample size (samples/ Mbytes)	Execution time for CUDA and CPU	
	CUDA with data transfer between RAM and video card memory	CPU
3000000 / 11 Mb	0.59 sec.	92.78 sec.
8000000 / 30 Mb	0.64 sec.	245.31 sec.
16000000 / 61 Mb	0.70 sec.	493.67 sec.

Table 1 shows execution time (for CUDA and CPU) for a DFT-modulated filter bank with complex modulation when a one-dimensional signal is processed. Table 2 is also related to a one-dimensional signal, but contains computation results for WOLA and the polyphase form in the case of critical decimation. Table 3 contains the results for a 5-channel signal ($S = 5$).

Detailed analysis of Tables I-III leads to the following conclusions:

- application of CUDA provides significant reduction in execution time (reduction in computational costs) in comparison with CPU;
- polyphase implementation form is the least time-consuming, whereas direct implementation form is the most time-consuming among the three different forms mentioned in the paper;
- increasing data size leads to a rise in the difference between execution times for CUDA and CPU;
- data transfer time is nearly half of total processing time for CUDA, which means that further improvements of CUDA have to be developed and tested;
- CUDA makes it possible to reduce hardware costs and to implement more complex computational signal processing algorithms.

Conclusion

In the paper we have considered hydroacoustic monitoring tasks and digital filter bank implementations for their solution. The modification of WOLA for multichannel signal processing has been developed and compared to a multichannel filter bank (polyphase form and the form with complex modulation). Software-hardware implementations have been analyzed and CUDA has been shown to significantly reduce processing time and increase performance (reduce computational costs) in comparison with CPU.

The paper is supported by the grant of the Russian Foundation for Basic Research ("My first grant") № 14-07-31250/14 and Contract № 02.G25.31.0058 dated 12.02.2013 (Board of Education of Russia).

REFERENCES

[1] D. Porcino, W. Hirt Ultra-Wideband Radio Technology: Potential and Challenges Ahead // IEEE Communications Magazine, 2003, Vol. 41(7), 66–74.

[2] R.P. Hodges Underwater Acoustics: analysis, design, and performance of sonar // UK, Wiley Interscience, 2010, 353 p.

[3] A. Antoniou, A. Digital Filters: Analysis, Design, and Applications // USA: McGraw-Hill, 1993, 689 p.

[4] S.K. Mitra Digital Signal Processing: A Computer-Based Approach // USA: McGraw-Hill, 1998, 940 p.

[5] R. E. Crochiere, L. R. Rabiner Multirate digital signal processing // USA: Prentice Hall, 1983, 411 p.

[6] M.G. Bellanger, G. Bonnerot, M. Coudreuse Digital filtering polyphase network: Application to sample rate alteration and filter banks // IEEE Trans. Acoust., Speech and Signal Processing. V. ASSP-24. Apr., 1976, 109-114.

[7] D.I. Kaplun, D.M. Klionskiy, A.S. Voznesenskiy, V.V. Gulvanskiy Order reduction of a low-frequency prototype filter by its magnitude response symmetrization in the task of filter bank design for wideband monitoring // International scientific and technical conference dedicated to the 50th anniversary of MREI-BSUIR, Belarus, Minsk, 2014, p. 270-271.

[8] D. I. Kaplun, D. M. Klionskiy, A. S. Voznesenskiy, V. V. Gulvanskiy Application of polyphase filter banks to wideband monitoring tasks // IEEE NW Russia Young Researchers in Electrical and Electronic Engineering Conference (ElConRusNW), Russia, Saint Petersburg, 2014, p. 97-100.

[9] D. B. Kirk, Wen-mei W. Hwu Programming Massively Parallel Processors: A Hands-on Approach / USA: Morgan Kaufmann, 2010, 280 p.

Authors: *associate prof. dr. Dmitry Kaplun, E-mail: mitya_kapl@front.ru; associate prof. dr. Dmitry Klionskiy, E-mail: klio2003@list.ru; senior researcher Alexander Voznesenskiy, E-mail: a-voznesensky@yandex.ru; senior researcher Vyacheslav Gulvanskiy, E-mail: slava-a-a@mail.ru; Saint Petersburg Electrotechnical University "LETI", ul. Professora Popova 5, 197376, Saint Petersburg, Russian Federation.*