Short-critical-path algorithm for allpass transform

Maxim I. Porhun ✉ and Maxim I. Vashkevich
Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus
✉ E-mail: porhun@bsuir.by

Allpass transform is a key element of the discrete systems which produces signal processing in warped frequency domain. The paper presents a new short-critical path algorithm for allpass transform, which essentially consists of calculating the outputs of allpass filters chain. The algorithm is based on the proposed dual allpass filter structure, which computes the outputs of a cascade of two allpass filters in one computational pass. The dual allpass filter structure is obtained using the state variable representation of the allpass filter. The efficiency of the proposed algorithm is analyzed in the context of allpass transform implementation using a general purpose CPU. The results of the experiment show that the use of the proposed algorithm accelerates the calculation of the allpass transform by 1.81 times.

Introduction: Systems based on allpass transform (APT) [1–3] are widely used for various tasks in digital signal processing. Application of APT is a simple and effective way to achieve non-uniform frequency resolution in discrete-time systems [3], which is especially important for hearing frequency selectivity modeling. Formally, the APT consists of the implementation of a chain (cascade) of allpass IIR filters, and the output of the transform is the outputs of each filter in the cascade.

Here we bring several examples of APT usage. In [4], the APT is used to implement a warped discrete Fourier transform (WDFT) with a non-uniform frequency resolution. In particular, WDFT is used to detect signals with multiple narrow-band components. In [1], IIR filters are implemented using an allpass chain to build audio equalizers that take into account hearing resolution. In [5], APT is embedded into the Burg spectrum analysis procedure to improve resolution at low frequencies.

The main issue in development of systems based on APT is the implementation of an allpass chain (APC) [3, 6]. In practical applications, the length of the chain reaches 1024 elements [3]. The operations in the APC must be carried out in the original sampling rate which increases the computational cost and time. An important aspect of APC implementation is that the computational algorithm contains data dependency, which makes it difficult to parallelize the algorithm in its original formulation.

We propose an accelerated algorithm for the computation of the outputs of APC which is based on dual allpass filter structure. The dual allpass filter is designed to obtain the outputs of a cascade of two allpass filters in a one computational pass. The proposed dual allpass filter structure has the same critical path as original allpass filter structure, which compute single filter output. The rest of the paper follows the next plan: (i) description of the allpass filter and its computational graph; (ii) the analysis of the computation time of the APC; (iii) description of the proposed structure of dual allpass filter; (iv) evaluation of the proposed short-critical-path algorithm; (v) concluding remarks.

Allpass chain: In this section, we describe the conventional approach that is used to implement allpass chain, which essentially implements APT. Allpass filter has the following transfer function:

$$A(z) = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}, \; |\alpha| < 1, \tag{1}$$

where α is a coefficient, which controls the frequency warping.

Using the inverse z-transform (1) can be converted to difference equation:

$$y(n) = x(n-1) + \alpha(y(n-1) - x(n)). \tag{2}$$

where $x(n)$ – filter input, $y(n)$ – filter output.

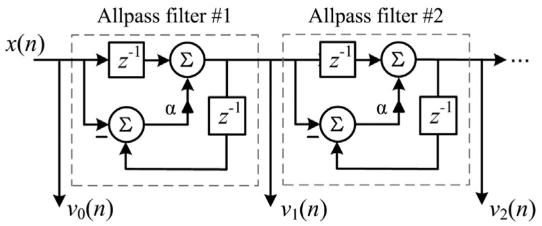

Fig. 1 *Allpass chain*

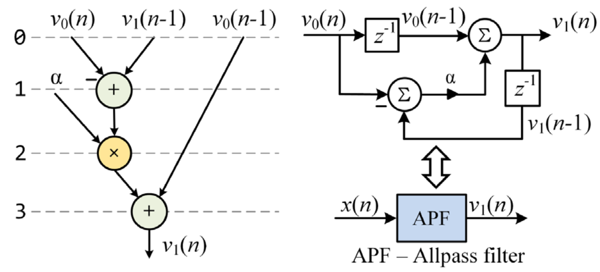

Fig. 2 *Directed computational graph of the allpass filter*

From computational point of view implementation of (2) requires 2 additions and 1 multiplication. Using filter structure that based on the (2) the APC can be implemented as shown in Figure 1.

It is easy to see that the scheme on Figure 1 is serial in nature and possesses a data dependency. For example, $v_2(n)$ cannot be computed until the computation of $v_1(n)$ is completed. Figure 2 shows the directed computational graph of allpass filter (2). The graph is divided into layers. Each layer contains operations that can be computed in parallel.

It can be seen that the graph of allpass filter has three layers (the 0th layer is the input data). Therefore, if the APC (see Figure 1) consists of N elements, then the total number of layers in its computational graph is $3N$. The number of layers is important, it shows the possibility to parallel the algorithm, since calculations in each layer depend on the results of the previous layer. We can assume that the calculation time of one layer is equal to time of one processor cycle. This means that the implementation of the APC as shown in Figure 1 will take $3N$ processor cycles. The aim of this paper is to design algorithm for calculating outputs of APC with reduced number of layers in computational graph by increasing the degree of parallelism. The difficulty in implementing the discrete-time systems based on APT is that when a new input sample $x(n)$ comes, all N outputs of the APC should be recalculated. Moreover, the computational process in system based on APT cannot be carried out until the outputs of all elements of the APC are calculated. Thus, the time of calculation of all outputs of the APC becomes crucial. This is the main motivation for developing short-critical-path algorithm for APT.

Dual allpass filter structure: In this section we describe new allpass filter structure that will be used as a building block to construct short critical path APT. For our purpose we will use state variable description of allpass filter (2) [7]:

$$\begin{aligned} y(n) &= (1-\alpha^2)w(n-1) - \alpha x(n) \\ w(n) &= \alpha w(n-1) + x(n), \end{aligned} \tag{3}$$

where $w(n)$ is an internal state variable.

System (3) can be rewritten in a matrix form:

$$\begin{bmatrix} w(n) \\ y(n) \end{bmatrix} = \begin{bmatrix} \alpha & 1 \\ (1-\alpha^2) & -\alpha \end{bmatrix} \begin{bmatrix} w(n-1) \\ x(n) \end{bmatrix}. \tag{4}$$

We will use the matrix form (4) to construct the dual allpass filter structure. Suppose we have a cascade of two allpass filters that implemented according to Equation (4). Each filter in the cascade has its own state variable ($w_1(n)$ – for first filter and $w_2(n)$ – for second) and output signal ($v_1(n)$ and $v_2(n)$, respectively). We want to describe the cascade of two filters as a matrices multiplication operation. For this reason let us define input vector as $[w_2(n-1) \quad w_1(n-1) \quad v_0(n)]^T$ (here $v_0(n)$

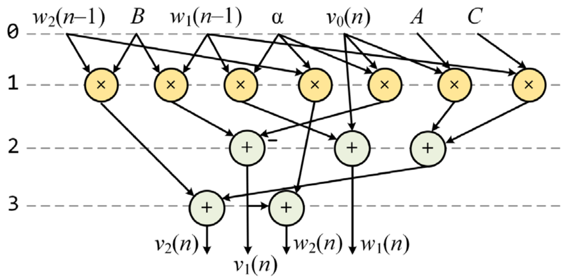

Fig. 3 *Directed computational graph of the allpass filter*

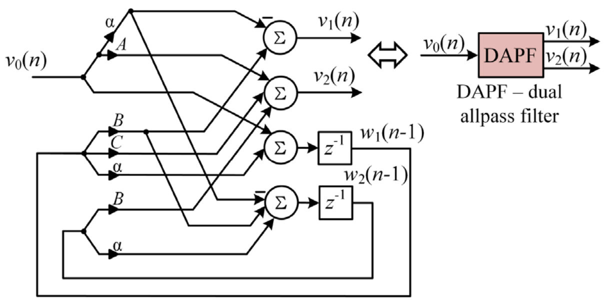

Fig. 4 *Block-scheme of the dual allpass filter*

is equivalent to input signal of the APC as shown in Figure 1), and the output vector as $[w_2(n) \quad w_1(n) \quad v_2(n) \quad v_1(n)]^T$. After that the cascade of two allpass filter can be represented as

$$\begin{bmatrix} w_2(n) \\ w_1(n) \\ v_2(n) \\ v_1(n) \end{bmatrix} = \begin{bmatrix} \alpha & 0 & 1 \\ 0 & 1 & 0 \\ (1-\alpha^2) & 0 & -\alpha \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \alpha & 1 \\ 0 & (1-\alpha^2) & -\alpha \end{bmatrix} \begin{bmatrix} w_2(n-1) \\ w_1(n-1) \\ v_0(n) \end{bmatrix}. \tag{5}$$

After multiplication of the matrices the expression (5) can be simplified:

$$\begin{bmatrix} w_2(n) \\ w_1(n) \\ v_2(n) \\ v_1(n) \end{bmatrix} = \begin{bmatrix} \alpha & B & -\alpha \\ 0 & \alpha & 1 \\ B & C & A \\ 0 & B & -\alpha \end{bmatrix} \begin{bmatrix} w_2(n-1) \\ w_1(n-1) \\ v_0(n) \end{bmatrix}, \tag{6}$$

where $A = \alpha^2$, $B = (1-\alpha^2)$ and $C = -\alpha(1-\alpha^2)$. This equation defines the dual allpass filter. In Figure 3 the matrix equation (6) is represented in the form of computational graph.

From Figure 3, it can be seen that computational graph of the dual allpass filter (DAPF) has 3 layers. It is the same number of layers as in graph of single allpass filter in Figure 2. Now, if we construct the allpass chain with N elements using the DAPF, the depth of the resulting computational graph will be twice as shorter as when using the standard allpass filter structure (i.e. $\frac{3}{2}N$ instead of $3N$).

Block scheme of the DAPF is presented in Figure 4.

The computational costs of the DAPF is 7 multiplications and 5 additions. This is more than 2 multiplications and 4 additions that are required for a cascade of two allpass filters implemented according to the expression (2). Therefore, the proposed DAPF structure has an increased computational complexity, but leads to reduce the computation time due to parallelization.

Evaluation: The proposed DAPF (Figure 4) and direct form of allpass filter (Figure 2) were implemented using C language to evaluate their efficiency in context of implementing the allpass transform using a general purpose CPU. Therefore our evaluation address the following question: the CPU time, required for calculation of output of APC with N elements using direct form of allpass filter and using DAPF.

We tested two implementations of APC that are based on the different allpass structures. The CPU time of each variant of implementation were evaluated 10 times, then the obtained values were averaged. The scheme of the experiment is given in Figure 5. The input signal $x(n)$ is a discrete

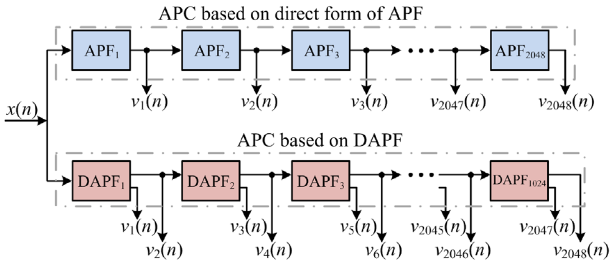

Fig. 5 *Experiment scheme*

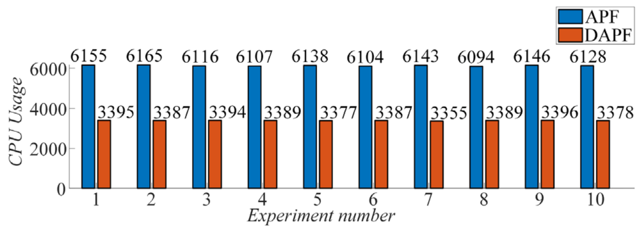

Fig. 6 *Run-time measurements of software implementations of APC based on direct form of allpass filter and DAPF*

sinusoid with a length of 10^6 samples (sampling rate 44.1 kHz). The signal passed through APC with $N = 2048$ elements. To analyze the CPU time, we used the built-in profiler of the Visual Studio 2019 IDE and a PC with Intel Core i7-37700 processor (clock frequency 3.4 GHz). The source C code used for experiments is available in public repository [8].

Experimental results: The run-time measurements obtained in experiment are given in Figure 6. The results show that run-time of the APC based on direct form of allpass filter equals to 6129.6 ± 22 CPU time units, while the implementation based on DAPF requires 3384.7 ± 11 CPU time units (Figure 6). Thus, the implementation of the APT is accelerated by 1.81 times due to the use of DAPF.

Conclusion: The paper proposes short-critical-path algorithm that accelerates the computation of APT. The algorithm is based on the new dual allpass filter structure, which is obtained using state variable description approach. We compare the proposed algorithm of computation of APT with algorithm based on direct form of allpass filters. The results of the experiments showed that the use of the proposed algorithm accelerates the calculation of the APT outputs by 1.81 times.

Author contributions: Maxim Porhun: Data curation, software, validation, visualization, writing - review and editing. Maxim Vashkevich: Conceptualization, data curation, formal analysis, methodology, software, validation, visualization, writing - review and editing.

Conflict of interest statement: The authors declare no conflicts of interest.

Data availability statement: The data that support the findings of this study are openly available in Short-critical-path_algorithm_for_allpass_transform at https://github.com/Maxim-Porhun/Short-critical-path_algorithm_for_allpass_transform, reference number 8.

Received: *4 July 2023* Accepted: *9 October 2023*
doi: 10.1049/ell2.12986

References

1 Bank, B.: Warped, kautz, and fixed-pole parallel filters: A review. *J. Audio Eng. Soc.* **70**(6), 414–434 (2022)

2 Ma, T., Wei, Y., Lou, X.: Reconfigurable nonuniform filter bank for hearing aid systems. *IEEE/ACM Trans. Audio, Speech, Language Process.* **30**, 758–771 (2021)

3 Gülzow, T., Engelsberg, A., Heute, U.: Comparison of a discrete wavelet transformation and a nonuniform polyphase filterbank applied to spectral-subtraction speech enhancement. *Signal Process.* **64**(1), 5–19 (1998)
4 Kwon, O.: Efficient detection of multi-narrowband using the warped discrete Fourier transform. *J. Comput. Sci.* **6**(1), 26–33 (2018)
5 Roth, K., et al.: Frequency warped Burg's method for AR-modeling. In: 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pp. 5–8. IEEE, Piscataway (2003)
6 Kleimola, J., et al.: Sound synthesis using an allpass filter chain with audio-rate coefficient modulation. In: Proceedings of the 12th International Conference on Digital Audio Effects (DAFx-09), pp. 305–312. Como, Italy (2009)
7 Roberts, R.A., Mullis, C.T.: *Digital signal processing*. Addison-Wesley Longman Publishing Co., Inc., Reading, MA (1987)
8 Porhun, M., Vashkevich, M.: Short-critical-path algorithm for allpass transform. https://github.com/Maxim-Porhun/Short-critical-path_algorithm_for_allpass_transform. Accessed 29 June 2023