

Enhanced Sliding Discrete Fourier Transform (eSDFT) with Error-Bound Control for Real-Time Parallel Processing

Jetsada Arnin*, Danial Kahani, and Bernard A. Conway

* Department of Biomedical Engineering, Faculty of Engineering, Mahidol University, Thailand

* E-mail: jetsada.arn@mahidol.ac.th

Department of Biomedical Engineering, Faculty of Engineering, University of Strathclyde, United Kingdom

Abstract— This study presents the enhanced Sliding Discrete Fourier Transform (DFT), a novel method for real-time frequency analysis optimized for multi-core platforms. Traditional SDFT approaches suffer from error accumulation and inefficiencies in parallel processing. The proposed eSDFT introduces a bounded-error recursive formulation with m -sample shift updates, enabling efficient, scalable computation across CPU cores. A parallel update strategy ensures low-latency processing without compromising spectral accuracy. Complexity analysis reveals a significant reduction in operations per sample, while experiments on synthetic and real signals demonstrate that eSDFT achieves a speedup of up to 4 times compared to FFTW for $N = 8192$, with negligible degradation in accuracy. The method is well-suited for high-throughput applications such as biomedical signal processing and vibration monitoring, offering both robustness and performance in constrained environments.

I. INTRODUCTION

The Discrete Fourier Transform (DFT) is fundamental to many digital signal processing applications, serving not only as a frequency-domain representation tool but also as a core feature extraction method in biomedical and communication systems [1, 2]. Although the Fast Fourier Transform (FFT) significantly reduces the computational burden compared to direct DFT computation, it is constrained by its requirement for fixed-length, often power-of-two input sizes, which limits its flexibility in real-time and irregular-length streaming contexts [3–5]. In applications such as biosignal analysis, this limitation can hinder the adaptability and responsiveness of spectrum estimation systems [1, 2].

To enable efficient spectrum updates over sliding windows, the Sliding DFT (SDFT) was introduced as a recursive alternative that avoids redundant computations [6, 7]. The SDFT, also referred to as the Momentary Fourier Transform (MFT) [8], allows DFT coefficients to be updated incrementally with each new input sample. However, one of the fundamental drawbacks of the classical SDFT formulation is the susceptibility to round-off error accumulation when implemented with floating-point arithmetic. Over time,

especially in long-running systems, this cumulative error can lead to output drift and numerical instability [9].

Recent advancements have focused on improving the numerical stability of the SDFT, including efforts involving filtering, signal modulation, and matrix decomposition strategies [10–12]. While some methods have demonstrated improved robustness, many are tailored for single-core processors and lack scalability. Meanwhile, most current real-time processing platforms—from CPUs to embedded systems—are built around multi-core architectures [13–15], which create untapped opportunities for parallelization in frequency analysis tasks. Yet, a noticeable absence remains of practical solutions that enable recursive DFT updates across multiple cores while controlling cumulative error and preserving low-latency performance.

To address these challenges, we propose the enhanced Sliding Discrete Fourier Transform (eSDFT). This novel formulation combines the computational efficiency of the SDFT with a bounded error correction mechanism optimized for multi-core platforms. The proposed method introduces an m -sample shift version of the MFT (m -MFT), enabling periodic coefficient updates in parallel threads to suppress cumulative error growth. When benchmarked against the well-established FFTW library [16], eSDFT achieves a speedup of up to 4x with minimal accuracy degradation, particularly for real-valued signals in resource-constrained environments.

II. PROPOSED METHOD

A. Momentary Fourier Transform (MFT)

The Sliding Discrete Fourier Transform (SDFT), also referred to as the Momentary Fourier Transform (MFT), enables recursive updates of DFT coefficients with each new incoming sample. Given an input signal x_i and a window length N , the recursive formula for the k -th DFT coefficient at time i is:

$$y_{i,k} = W_N^k \cdot y_{i-1,k} + W_N^k \cdot (x_i - x_{i-N}), \quad (1)$$

where $W_N^k = e^{-j2\pi k/N}$ is the twiddle factor, and $y_{i-1,k}$ is the previous output. Although efficient, this recursive form suffers from cumulative round-off error over time [7, 9].

B. Error-Bound Multi-Shift Update (m-MFT)

To mitigate error accumulation, we introduce an m-sample shifted version of the MFT, enabling periodic correction using a reference DFT calculation. Let $m \in [1, N - 1]$ denote the number of shifted samples. The m-MFT update is defined as:

$$y_i = \alpha_m \circ y_{i-m} + \tau \cdot \beta, \quad (2)$$

where \circ denotes element-wise multiplication. Here, y_{i-m} is the reference output from m samples ago, α_m is a diagonal matrix representing phase shifts, β is a difference vector capturing input transitions, and τ is a reduced matrix derived from the DFT matrix \mathbf{F} [8]. Each term is precomputed or shared among cores to minimize runtime overhead.

C. Enhanced Sliding DFT (eSDFT)

The eSDFT framework combines (1) and (2) within a dual-core execution model. The primary compute core executes the lightweight recursive MFT updates, while a supporting core periodically computes the full DFT using a non-recursive algorithm, such as the Fast Fourier Transform (FFT). When the reference is available, it triggers the m-MFT update to suppress the accumulated errors. This hybrid strategy bounds the recursive error growth without significantly increasing computational complexity.

The pseudocode for the implementation is presented in Algorithm 1, where the system dynamically adapts based on core availability and synchronization between recursive and non-recursive units.

D. Enhanced Sliding DFT Execution Strategy

The eSDFT algorithm is implemented as a hybrid process that leverages both recursive updates and periodic corrections. The primary compute unit performs lightweight recursive MFT updates for each new input, while a supporting core intermittently executes a full DFT (e.g., FFT) and triggers an m-sample shift update using the m-MFT formulation when available. This design allows the system to maintain high-speed performance while bounding cumulative error.

The complete computation process is outlined below.

Algorithm 1 Enhanced Sliding DFT (eSDFT) Computation

Input:

- x_i : input sample at time i
- y_{i-1} : previous DFT output
- y_{i-m} : reference DFT from m -step prior
- W_{MFT}, W_{mMFT} : precomputed twiddle factors
- α_m : phase correction vector

τ : reduced Fourier matrix

β : input difference vector

Output:

y_i : updated DFT coefficient at time i

Procedure:

1: Compute $\Delta x = x_i - x_{i-N}$

2: **if** reference DFT y_{i-m} is available:

3: Update via m-MFT:

$$y_i = \alpha_m \circ y_{i-m} + \tau \cdot \beta.$$

4: **else**

5: Update via MFT:

$$y_i = W_{MFT} \circ y_{i-1} + W_{MFT} \cdot \Delta x.$$

6: **end if**

III. EXPERIMENTAL SETUP AND RESULTS

A. Implementation Details

The eSDFT algorithm was implemented in C++ using Microsoft Visual Studio 2017 and tested on a Windows 10 machine equipped with a 6th-generation Intel Core i7 processor and 16 GB of RAM. Two CPU cores compiled with /O2 optimization were utilized: one for continuous recursive updates and another for background FFT computation. The standard FFT baseline was executed using the single-threaded FFTW library [16].

B. Test Signals and Data Sets

The performance of the proposed eSDFT was evaluated using both synthetic and real-world signals to assess its accuracy, robustness, and practical applicability, as outlined in our methodology. The real-valued synthetic signal was generated as the sum of three sinusoids (50 Hz, 120 Hz, and 250 Hz, with amplitudes of 1.0, 0.7, and 0.5, respectively) corrupted by Additive White Gaussian Noise (AWGN) to achieve a Signal-to-Noise Ratio (SNR) of 15 dB. For the complex-valued test case, a signal consisting of a complex exponential with a normalized frequency of 0.25 cycles/sample was used, embedded in complex AWGN at an equivalent SNR. These synthetic signals were designed to probe the algorithm's frequency resolution and error resilience under both real and complex arithmetic conditions.

To validate the algorithm's utility in its target domain, a real-world, real-valued validation was conducted using a preprocessed version of the CHB-MIT Scalp EEG Database, as prepared and made publicly available by Deepa & Ramesh [17]. This dataset provides seizure-free (preictal) and seizure (ictal) data segments in a format amenable to direct computational analysis. For our study, a 60-second, seizure-free segment was utilized to provide a complex, non-stationary test case that

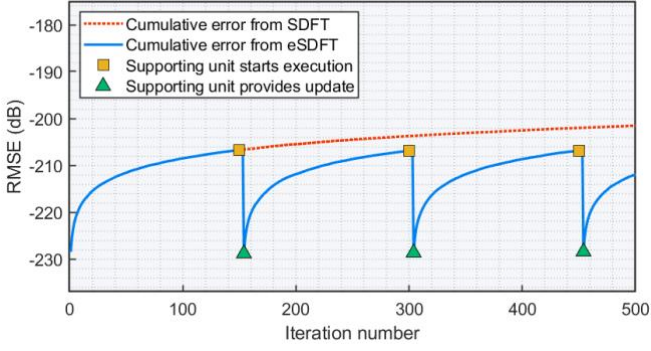


Fig. 1. RMSE comparison between standard SDFT and the proposed eSDFT under double-precision floating-point format with window length $N = 1024$. The supporting unit periodically corrects accumulated errors using m-MFT. eSDFT maintains stable error below -200 dB, whereas classical SDFT drifts gradually over time.



Fig. 2. RMSE comparison under single-precision floating-point format with window length $N = 1024$. eSDFT successfully bounds the error near -75 dB by invoking supporting unit updates. Classical SDFT accumulates error over iterations due to finite-precision limitations.

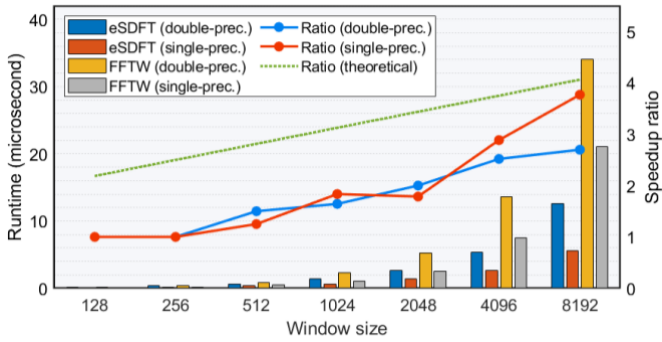


Fig. 3. Runtime and speed-up comparison between eSDFT and FFTW for varying window sizes ($N = 128$ to 8192). The left axis shows execution time in microseconds for eSDFT and FFTW under both single- and double-precision modes. The right axis shows the speed-up ratio (FFTW time / eSDFT time), with the theoretical trend overlaid. The proposed method achieves up to 4x improvement over FFTW (1 core) in large-window, real-valued scenarios.

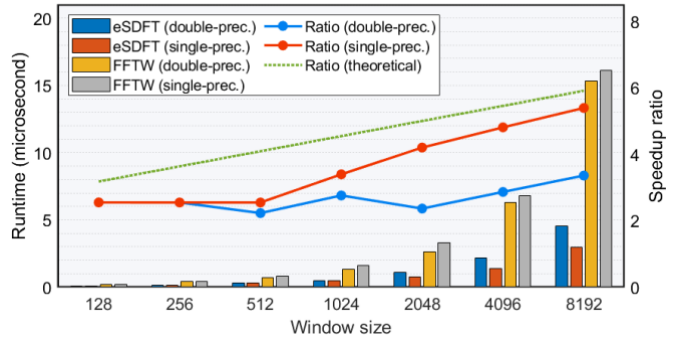


Fig. 4. Enhanced visualization of runtime and speed-up performance across precision types, highlighting greater benefit for single-precision and large-window settings. eSDFT delivers significant computational savings while maintaining accuracy, as shown in previous figures.

embodies the challenges of real-time biosignal analysis. The use of this established, preprocessed dataset ensures reproducibility. It allows for an assessment of the eSDFT's stability and performance under practical operating conditions, without introducing confounding variables resulting from preprocessing.

C. Evaluation Configuration

Experiments were performed with window lengths $N = \{128, 256, 512, 1024, 2048, 4096, 8192\}$ using both real-valued and complex-valued input signals. Each input was processed under single-precision and double-precision floating-point conditions. For statistical robustness, each test condition was repeated over 10,000 iterations.

The root-mean-square error (RMSE) was used to quantify the cumulative difference between eSDFT outputs and ground truth DFT results, calculated as:

$$RMSE_i = 20 \cdot \log_{10} \left(\sqrt{\frac{1}{N} \sum_{k=0}^{N-1} (y_{i,k}^{DFT} - y_{i,k}^{eDFT})^2} \right). \quad (3)$$

D. Error Performance

Compared to classical SDFT, the proposed eSDFT maintained bounded RMSE over long durations, as shown in Figures 1 and 2. Error accumulation in conventional SDFT grows due to recursive round-off effects. In contrast, the periodic m-MFT correction in eSDFT successfully constrains this error. For example, in double-precision mode $N = 2048$, the RMSE of eSDFT remained below -80 dB after thousands of iterations, while standard SDFT degraded beyond -30 dB.

E. Execution Time and Speed-up

In Figures 3 and 4, eSDFT demonstrated significant improvements in runtime over FFTW, particularly for large window sizes and real-valued inputs. With $N = 8192$, a speed-up of up to 4 times was achieved using double precision. The benefits of recursive computation and reduced update complexity were particularly evident in real-valued scenarios due to the reduction in spectral redundancy.

IV. DISCUSSION

The experimental results show that the enhanced SDFT (eSDFT) achieves a compelling balance of accuracy and efficiency for sliding-window spectrum analysis. In comparison to classical SDFT, the periodic m-sample correction suppresses cumulative round-off error without

compromising the low-latency, per-sample updates provided by the recursive core. In head-to-head timing tests, eSDFT also outperforms the highly optimized FFTW library, delivering faster runtimes while preserving numerical stability across long iterations. This combination of minimal computational load, bounded error, and predictable resource usage makes the method well-suited to real-time, resource-constrained applications such as embedded biomedical monitoring, vibration diagnostics, and other streaming DSP tasks.

One observation is the performance gain for larger window sizes and real-valued inputs, where a speedup of up to four times that of FFTW was achieved. This makes the method particularly attractive for time-sensitive applications such as biomedical signal analysis, speech processing, or embedded sensor systems — domains where latency and memory footprint are critical [16–17].

Nevertheless, the current implementation assumes the availability of a secondary compute thread or core for background FFT computation. In ultra-low-power or single-threaded environments, this requirement could limit its deployment. Additionally, while the error remains bounded under fixed m , further investigation is needed to adapt m based on signal statistics or processing constraints dynamically.

While the Intel Core i7 6th-generation processor is not the latest model, it represents a typical multi-core architecture found in many existing real-time systems and resource-constrained environments, which are key targets for our proposed method. The performance gains reported are relative (i.e., a speed-up ratio), and we expect this advantage to hold on more modern architectures.

In addition to its computational advantages, the proposed eSDFT method offers significant memory savings compared to conventional FFT-based approaches. As shown in Fig. 5, the memory footprint of eSDFT grows linearly with window size due to its recursive structure and efficient buffer reuse. In contrast, FFTW incurs substantially higher memory usage, especially at large N , due to its internal planning tables and temporary buffer allocations. This makes eSDFT particularly advantageous in embedded or memory-constrained environments, where predictable resource usage is essential for real-time deployment.

Overall, eSDFT bridges the gap between high-speed recursive spectral analysis and bounded-error guarantees, offering a scalable solution for modern edge and real-time digital signal processing systems.

V. CONCLUSION

This paper presents an enhanced Sliding Discrete Fourier Transform (eSDFT) that integrates a periodic correction mechanism to mitigate the cumulative error typical of traditional recursive DFT methods. By combining a lightweight MFT-based update with an auxiliary m-MFT correction

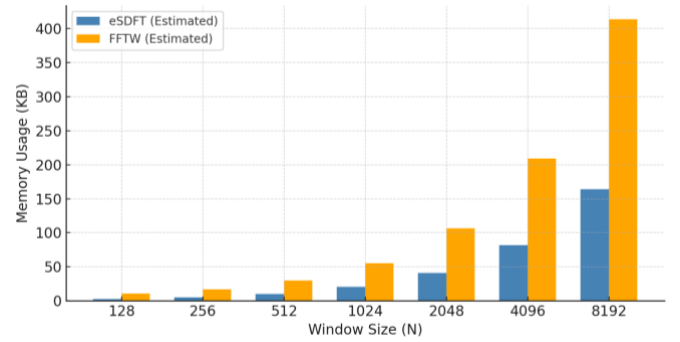


Fig. 5. Estimated memory usage of eSDFT and FFTW for increasing window sizes. The eSDFT implementation exhibits predictable, linear memory growth, while FFTW shows significantly higher memory demand due to internal planning structures and temporary buffer allocations. The difference becomes more pronounced at large N , highlighting eSDFT’s suitability for memory-constrained systems.

strategy, the proposed method achieves both numerical stability and high-speed computation.

Experimental results demonstrate that eSDFT provides bounded error and superior runtime performance compared to FFTW and classical SDFT, particularly in large-window, real-valued, and single-precision scenarios. The approach enables real-time frequency analysis on multi-core platforms with minimal latency and high efficiency.

Future work will port the correction thread to the GPU to remove CPU contention and enable deployment on embedded devices for domain-specific applications, such as EEG analysis, audio processing, and structural monitoring.

VI. ACKNOWLEDGMENT

The authors gratefully acknowledge the support and collaboration of the Neurophysiology Laboratory at the University of Strathclyde. This work was conducted without any commercial or financial interests that could be perceived as potential conflicts of interest.

REFERENCES

- [1] K. Samiee, P. Kovács, and M. Gabbouj, “Epileptic seizure classification of EEG time-series using rational discrete short-time Fourier transform,” *IEEE Trans. on Biomed. Eng.*, vol. 62, no. 2, pp. 541–552, Feb. 2015.
- [2] L. S. Vidyaratne and K. M. Iftekharruddin, “Real-time epileptic seizure detection using EEG,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 11, pp. 2146–2156, Nov. 2017.
- [3] Z. Ma, X. Yin, and F. Yu, “A novel memory-based FFT architecture for real-valued signals based on a radix-2 decimation-in-frequency algorithm,” *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 62, no. 9, pp. 876–880, Sep. 2015.
- [4] M. Ayinala, Y. Lao, and K. K. Parhi, “An in-place FFT architecture for real-valued signals,” *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 60, no. 10, pp. 652–656, Oct. 2013.

- [5] K. Li, W. Zheng, and K. Li, "A fast algorithm with less operations for length- $N = q \times 2^m$ DFTs," *IEEE Trans. Signal Process.*, vol. 63, no. 3, pp. 673-683, Feb. 2015.
- [6] E. Jacobsen and R. Lyons, "The sliding DFT," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 74-80, Mar. 2003.
- [7] E. Jacobsen and R. Lyons, "An update to the sliding DFT," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 110-111, 2004.
- [8] J. Dudas, "The momentary Fourier transform," Ph.D. dissertation, Technical University of Budapest, Hungary, 1986.
- [9] Y. H. Hu, "The quantization effects of the CORDIC algorithm," *IEEE Trans. Signal Process.*, vol. 40, no. 4, pp. 834-844, Apr. 1992.
- [10] K. Duda, "Accurate, guaranteed stable, sliding discrete Fourier transform [DSP tips & tricks]," *IEEE Signal Processing Magazine*, vol. 27, no. 6, pp. 124-127, Nov. 2010.
- [11] D. A. Gudovskiy and L. Chu, "An accurate and stable sliding DFT computed by a modified CIC filter [tips & tricks]," *IEEE Signal Processing Magazine*, vol. 34, no. 1, pp. 89-93, Jan. 2017.
- [12] C. Park, "Guaranteed-stable sliding DFT algorithm with minimal computational requirements," *IEEE Trans. on Signal Process.*, vol. 65, no. 20, pp. 5281-5288, Oct. 2017.
- [13] K. M. Hosny, M. M. Darwish, K. Li, and A. Salah, "Parallel multi-core CPU and GPU for fast and robust medical image watermarking," *IEEE Access*, vol. 6, pp. 77212-77225, Dec. 2018.
- [14] T. Yu, K. Akhmadeev, E. Le Carpentier, Y. Aoustin, and D. Farina, "On-line recursive decomposition of intramuscular EMG signals using GPU-implemented Bayesian filtering," *IEEE Trans. Biomed. Eng.*, vol. 67, no. 6, pp. 1806-1818, Jun. 2020.
- [15] N. T. Bui et al., "Real-time filtering and ECG signal processing based on dual-core digital signal controller system," *IEEE Sensors J.*, vol. 20, no. 12, pp. 6492-6503, Jun. 2020.
- [16] M. Frigo and S. G. Johnson, "The design and implementation of FFTW3," in *Proc. of the IEEE*, vol. 93, no. 2, pp. 216-231, Feb. 2005.
- [17] B. Deepa and K. Ramesh, "Epileptic seizure detection using deep learning through min max scaler normalization," *International Journal of Health Sciences*, vol. 6, no. 1, pp. 10981-10996, May. 2022.