Electronic Supplementary Information

A Derivative-Based Signal Processing Framework for Real-Time Event Detection in Impedance Flow Cytometry

Brendan Wurts[1], Charlie Jindrich[1], Yu Gong[2], Joshua Ojih[3], Ming Hu[3], Kun Yin[4], Leilei Shi[1*]

[1]*Department of Engineering, School of Engineering, Computing, and Mathematics, College of Charleston, Charleston, SC 29424, USA, shil1@cofc.edu*

[2]*Department of Physics and Astronomy, School of Natural and Environmental Sciences, College of Charleston, Charleston, SC 29424, USA*

[3]*Department of Mechanical Engineering, University of South Carolina, Columbia, SC 29208, USA*

[4] *School of Global Health, Chinese Center for Tropical Diseases Research, Shanghai Jiao Tong University School of Medicine, Shanghai 200025, China*

**Corresponding author: Leilei Shi shil1@cofc.edu*

Algorithm Description

To ensure smooth and accurate computation of the derivative across the entire signal, the algorithm employs different difference approximations for boundary and interior points. Specifically, for the first data point, the derivative is approximated using the forward difference method:

$$f'(x_1) \approx f(x_2) - f(x_1) \tag{1}$$

For each interior point x_i, the derivative is calculated using the central difference approximation:

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_{i-1})}{2} \tag{2}$$

For the last data point, the derivative is approximated using the backward difference method:

$$f'(x_n) \approx f(x_n) - f(x_{n-1}) \tag{3}$$

After computing the derivative, a threshold is applied to filter out noise while preserving the essential features of the signal. This thresholding process helps in detecting peaks that correspond to significant events, such as the passage of cells or particles.

Synthetic Data Generation

In order to test the performance of the derivative method, synthetic data streams with various noise levels have been generated by a custom MATLAB script. The script simulates realistic impedance flow cytometry signals by combining multiple noise sources, including pink noise, periodic noise, and optional white noise, with a clean signal composed of synthetic events. Each event represents the passage of a cell or particle through the detection region and is modeled as a bipolar Gaussian functions.[11, 27]

The synthetic signals were generated over a 30-second duration with a sampling frequency of 7196 Hz, with some datasets generated at up to 5× this rate to simulate high-resolution conditions, resulting in at least 215,880 data points per stream. To mimic different experimental conditions, noise levels were varied systematically. The pink noise level was kept constant, using values measured directly from our system to reflect the actual background noise observed in the experimental setup. In contrast, the periodic noise levels were systematically varied to evaluate their impact on signal quality and the robustness of the derivative method. Periodic noise components were introduced at different amplitudes and frequencies to represent interference from electrical equipment, harmonics from switching electronics, mechanical oscillations, or other environmental factors.

The clean synthetic events were randomly distributed within the data streams based on a Poisson process, with an average event rate of 5 events per second. [11, 28] This process models the random arrival of cells or particles over time. These data streams provide a controlled environment for testing, allowing for a thorough evaluation of the derivative method's capability to accurately detect and characterize cell passage events under different noise conditions.

To test the noise robustness of the derivative signal processing method, we systematically introduced both white noise and periodic noise to simulate realistic signal conditions observed in our impedance flow cytometry data. The periodic noise components were designed to match the frequencies encountered in our real-world system, which typically arise from electrical and environmental interference. Specifically, we introduced periodic noise with frequency components ranging from 3.5 Hz to 180 Hz, which were selected based on the noise characteristics observed in our experimental setup. The corresponding amplitudes for each frequency were normalized and scaled using a factor to control the strength of the noise in relation to the signal. The periodic noise was added to the clean signal through a combination of sinusoidal waveforms.

In addition to periodic noise, we introduced band-limited white noise to simulate random background interference. White noise was generated across a frequency range of 0 Hz to 100 Hz and was scaled using a factor to control its amplitude. This allowed us to simulate different levels of random noise interference commonly seen in experimental setups. The pink noise component, however, was kept constant throughout the experiments with a fixed of 5×10^{-6} V, a value observed from real-world experiments. Pink noise was generated by filtering white noise in the frequency domain using a $1/\sqrt{f}$ filter, which accurately models the frequency-dependent noise typically observed in biological systems where lower frequencies exhibit more noise power.

Various noise levels used in our experiments, with selected levels for both periodic and white noise ranging from 5×10^{-7} to 1.5×10^{-3} V for white noise and from 1×10^{-5} to 3×10^{-4} V for periodic noise.

These variations enabled us to evaluate the robustness of our signal processing techniques in different noise environments. To quantify the impact of the noise on signal quality, we converted the noise magnitude to dB Signal-to-Noise Ratio (SNR) using the following MATLAB function. This function calculates the SNR based on the root mean square (RMS) power of the signal (p_s) and noise (p_n) within specific event windows, applying the equation:

$$SNR(dB) = 10 \times log_{10}\left(\frac{p_s}{p_n}\right) \tag{5}$$

Detailed timing benchmarks and implementation specifications

Notch Filter and Detrending Implementation Details

To mitigate periodic noise components commonly observed in experimental IFC signals—particularly those originating from power line interference and instrumentation drift—we implemented a multistage filtering routine as part of the traditional signal processing pipeline. The routine applies a cascade of second-order infinite impulse response (IIR) notch filters targeting

60 Hz, 120 Hz, and 180 Hz, each designed with a 3 Hz bandwidth around the center frequency. The normalized notch frequencies are calculated relative to the Nyquist frequency, and zero-phase filtering is applied using MATLAB's filtfilt function to prevent phase distortion.

In addition to the targeted notches, a custom IIR comb filter is employed to suppress broadband periodic interference centered around 10 Hz and its harmonics. The comb filter is configured using a modified version of MATLAB's iircomb() function with a quality factor Q=20, which ensures narrowband suppression while maintaining spectral fidelity outside the target frequencies.

Table S1. Computation time comparison between the traditional, derivative-based, and notch-filter-augmented signal processing methods across three simulated streams.

Stream	Traditional (s)	Derivative (s)	Traditional with notch (s)
1	0.250253	0.1912549	22.3129093
2	0.2483794	0.1963162	21.888916
3	0.2454775	0.2004076	22.2193944
4	0.2452065	0.1948337	22.4994633
5	0.249409	0.1974944	22.9284328
6	0.2466393	0.1876977	23.1507793
7	0.2446639	0.194513	22.2710444
8	0.2477022	0.2007753	21.9141999
9	0.2479345	0.188675	22.3039037
10	0.2477414	0.1902445	22.1215999
11	0.2621318	0.1966699	22.1086149
12	0.3084714	0.1845371	22.3308218
13	0.2847584	0.1987662	22.2838563
14	0.2595529	0.2632998	22.2759221
15	0.2573706	0.2087798	22.0179143
16	0.2659007	0.2156529	22.8107856
17	0.2542133	0.1915435	22.5786573
18	0.256759	0.2063403	21.8976197
19	0.2633408	0.1995616	22.2599239
20	0.2668829	0.189159	22.027936
21	0.2515984	0.200792	21.8115279
22	0.2783156	0.2014806	22.0520356
23	0.262323	0.2039773	21.882631
24	0.2740025	0.2057383	21.824548
25	0.2670045	0.2115863	21.7730944
26	0.2761044	0.2088654	21.4100123
27	0.2623766	0.2019398	21.7878581
28	0.273261	0.2088413	23.3894202

29	0.2635957	0.2025438	22.1428759
30	0.2642339	0.2087329	22.7720115
Average	0.26085347	0.20170067	22.23495699

[1] F. Caselli and P. Bisegna, "A simple and robust event-detection algorithm for single-cell impedance cytometry," *IEEE Transactions on Biomedical Engineering,* vol. 63, no. 2, pp. 415-422, 2015.

[2] F. Caselli and P. Bisegna, "Simulation and performance analysis of a novel high-accuracy sheathless microfluidic impedance cytometer with coplanar electrode layout," *Medical Engineering & Physics,* vol. 48, pp. 81-89, 2017.

[3] U. Hassan and R. Bashir, "Coincidence detection of heterogeneous cell populations from whole blood with coplanar electrodes in a microfluidic impedance cytometer," *Lab on a Chip,* vol. 14, no. 22, pp. 4370-4381, 2014.