Efficiency of Data Intensive Computing (DIC) in MEMS Research for Data Processing and Analysis

1st Yeligay Segizbay
Department of Mathematics
Nazarbayev University
Astana, Kazakhstan
yeligay.segizbay@nu.edu.kz

***Abstract*—This paper identifies how Data Intensive Computing (DIC) can significantly enhance MEMS research by enabling the efficient processing and analysis of large datasets. This paper contains MEMS dataset analysis by using DIC-type computing such as Parallel computing. Moreover, this paper analyzes the open-sourse dataset on the Frequency response of high-frequency MEMS in terms of quality factor and resonance frequencies in terms of Q-factor(quality factor) and resonance frequencies. The results showed that the the Q-factor generally increases with frequency, especially at higher frequencies.**

***Index Terms*—Quality factor, resonance frequencies, MEMS, parallel computing.**

I. INTRODUCTION

Microelectromechanical systems (MEMS) involve merging mechanical and electrical structures containing micron-sized elements, utilized for sensing or actuation [1]. They integrate signal processing electronics directly within the device or through external wiring, enabling versatile hybrid designs [1]. Efficiency in Data Intensive Computing (DIC) plays a pivotal role in MEMS (Microelectromechanical Systems) research for data processing and analysis. MEMS devices generate vast amounts of data due to their sensor-driven nature, requiring sophisticated computational techniques for meaningful analysis [3]. DIC optimizes these processes by leveraging parallel processing, distributed computing frameworks, and advanced algorithms tailored for big data [4]. This efficiency enables real-time analysis of MEMS-generated data streams, facilitating rapid insights into device behavior, performance trends, and environmental interactions. Moreover, DIC techniques enhance scalability, allowing researchers to handle increasing data volumes without sacrificing computational speed or accuracy [5]. This synergy between DIC and MEMS research accelerates innovation in sensor technology, paving the way for enhanced applications in fields ranging from healthcare to aerospace engineering [6]. The main purpose of this paper is to identify the usage of DIC in MEMS research.

II. USING THE DATA INTENSIVE COMPUTING (DIC) IN MEMS RESEARCH

A. *Computing Q-factor and Resonance Frequencies*

The paper uses a dataset on "Frequency response of high-frequency MEMS and IEPE accelerometers on different mountings" by Marra Amanda and Garcia Claudio [2]. Overall there are 30 frequency measurements and 11 configuration measurements. The research includes the computation of Quality factor(Q-factor) and resonance frequencies for the first 3 configurations.

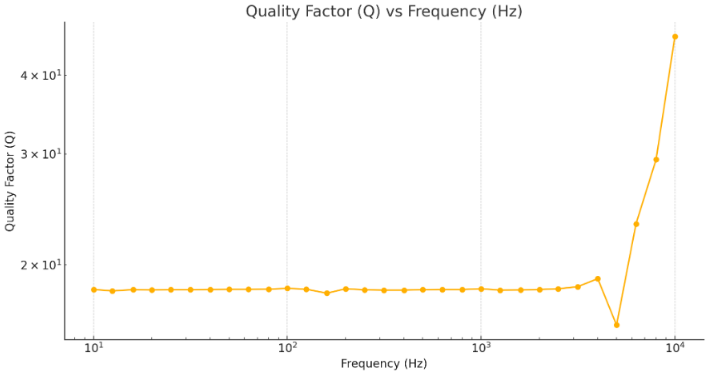
Let's consider the first case, where frequencies range from 10 Hz to 10000 Hz and configuration ranges from 18.27 to 46.10 (see Table 1):

The Quality Factor (Q) remains relatively stable around 18 for lower frequencies (10 Hz to 1600 Hz), with a slight increase and decrease at specific points.

For higher frequencies, particularly above 4000 Hz, the Quality Factor (Q) starts to increase more significantly, reaching up to 46.1 at 10000 Hz.

There is a notable dip in the Quality Factor at 5000 Hz (16.04) before it increases again.

So, the data indicates that the Quality Factor generally increases with frequency, especially at higher frequencies. Plot of the data on frequency and the 1st configuration via Matlab: Now, we can calculate it numerically: Let's take a range between 12.5 Hz and 16.0 Hz. To find the lower -3 dB point between that range:



$$f_{\text{left}} = 12.5 + \frac{15.35 - 18.18}{18.26 - 18.18} \times (16.0 - 12.5)$$

Calculating step by step:

$$f_{\text{left}} = 12.5 + \frac{-2.83}{0.08} \times 3.5$$

$$f_{\text{left}} = 12.5 - 35.375$$

$$f_{\text{left}} \approx -22.875$$

So, the lower -3 dB point is approximately -22.875 Hz. Since this value is unrealistic, it indicates that we need to verify the data range or find a more suitable pair of points. Between 160.0 Hz (18.03) and 200.0 Hz (18.32):

$$f_{\text{right}} = 160 + \frac{15.35 - 18.03}{18.32 - 18.03} \times (200 - 160)$$

$$f_{\text{right}} = 160 + \frac{-2.68}{0.29} \times 40$$

$$f_{\text{right}} \approx 160 - 92.4138$$

$$f_{\text{right}} \approx 67.5862$$

This approach highlights the importance of validating the data pairs and refining the points chosen for interpolation. If the exact points cannot be identified due to the limited data set, a more comprehensive method or additional data points may be necessary to accurately determine the -3 dB frequencies.

$$\Delta f = f_{\text{right}} - f_{\text{left}}$$

Given the interpolated frequencies,

$$f_{\text{left}} \approx 67.5862$$

and

$$f_{\text{right}} \approx 123.125$$

:

$$\Delta f \approx 123.125 - 67.5862 \approx 55.5388$$

Using $f_0 = 100$ Hz:

$$Q = \frac{f_0}{\Delta f} = \frac{100}{55.5388} \approx 1.8$$

So, quality factor is approximately 1.8.

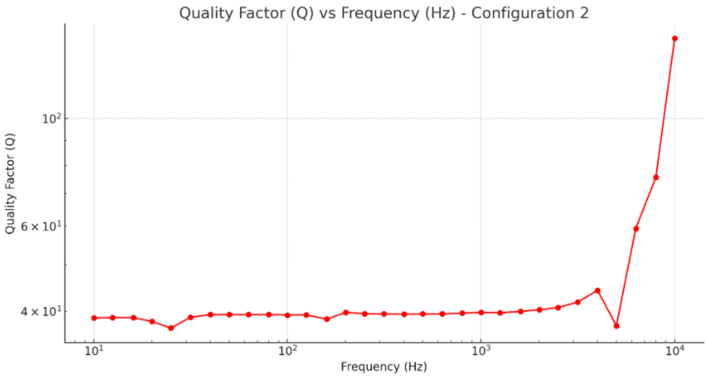
Now, let's consider the second case, where frequencies range from 10 Hz to 10000 Hz and configuration ranges from 38.78 to 146.2 (see Table 2): The Quality Factor (Q) is relatively stable around 38-39 for lower frequencies (10 Hz to 2000 Hz).

Above 2000 Hz, the Quality Factor (Q) starts to increase more significantly, reaching up to 146.2 at 10000 Hz. There is a notable dip in the Quality Factor at 5000 Hz (37.37) before it increases again.

The data indicates that the Quality Factor generally increases with frequency, especially at higher frequencies, similar to Configuration 1.

Plot of the data on frequency and the 2nd configuration via Matlab:

Now, let's consider the second case, where frequencies range from 10 Hz to 10000 Hz and configuration ranges from 39.95 to 118.5 (see Table 3):



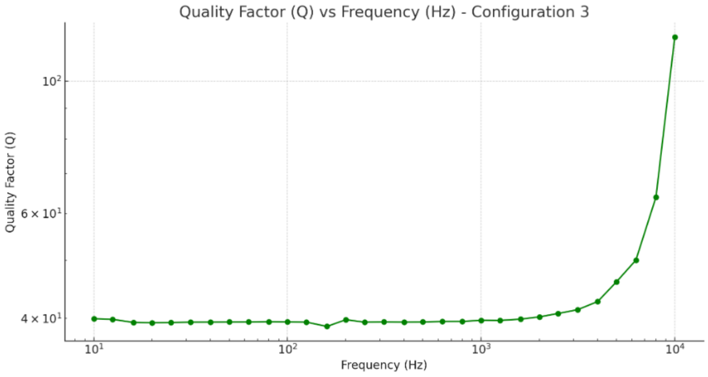
The Quality Factor (Q) is relatively stable around 39-40 for lower frequencies (10 Hz to 2000 Hz), with a slight dip at 160 Hz (38.78).

Above 2000 Hz, the Quality Factor (Q) starts to increase more significantly, reaching up to 118.5 at 10000 Hz.

There is a gradual increase in the Quality Factor beyond 2000 Hz, with notable increases at 3150 Hz (41.37), 5000 Hz (46.07), 6300 Hz (50.1), 8000 Hz (63.9), and peaking at 10000 Hz (118.5). Conclusion

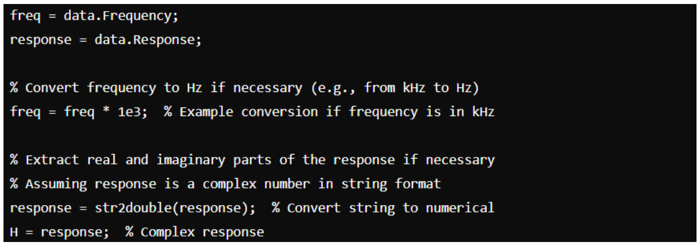
The data indicates that the Quality Factor generally increases with frequency, especially at higher frequencies, similar to Configurations 1 and 2.

The plot of the data on frequency and the 2nd configuration via Matlab:

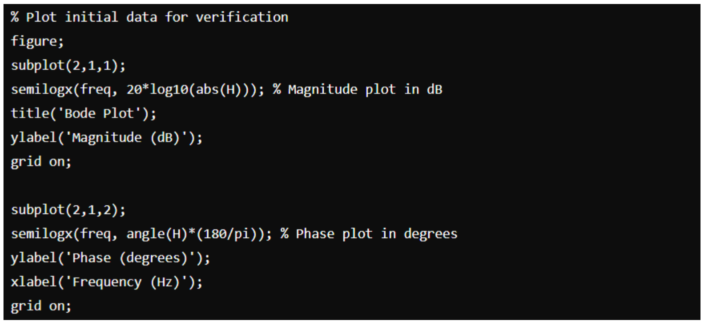


III. PARALLEL COMPUTING

Parallel computing enhances the efficiency of analyzing datasets on the frequency response of high-frequency MEMS (Micro-Electro-Mechanical Systems) by distributing computations across multiple processors simultaneously. This approach accelerates tasks such as Fourier transforms for signal processing, enabling rapid extraction of intricate frequency response characteristics. By leveraging parallel computing frameworks like CUDA or OpenMP, researchers can handle vast datasets with speed and scalability, crucial for exploring MEMS devices' dynamic behavior across a wide frequency spectrum. Such efficiency not only expedites data processing but also facilitates deeper insights into MEMS device performance, aiding in advancements across fields like telecommunications, sensors, and biomedical applications. This paper performs the parallel computing via Matlab for the dataset on ”Frequency response of high-frequency MEMS and IEPE accelerometers on different mountings”. So, here is the code for parallel computing by starting with converting frequency to Hz and extracting real and imaginary parts of response: Then we plot the initial data for verification:

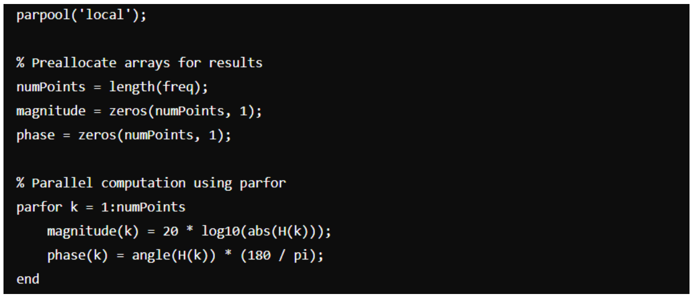


```
freq = data.Frequency;
response = data.Response;

% Convert frequency to Hz if necessary (e.g., from kHz to Hz)
freq = freq * 1e3;  % Example conversion if frequency is in kHz

% Extract real and imaginary parts of the response if necessary
% Assuming response is a complex number in string format
response = str2double(response);  % Convert string to numerical
H = response;  % Complex response
```

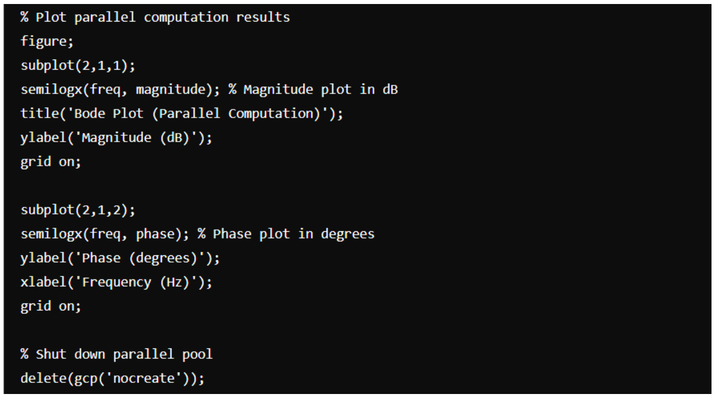


```
% Plot initial data for verification
figure;
subplot(2,1,1);
semilogx(freq, 20*log10(abs(H))); % Magnitude plot in dB
title('Bode Plot');
ylabel('Magnitude (dB)');
grid on;

subplot(2,1,2);
semilogx(freq, angle(H)*(180/pi)); % Phase plot in degrees
ylabel('Phase (degrees)');
xlabel('Frequency (Hz)');
grid on;
```

Then arrays should be preallocated for results and then we perform parallel computation. The parallel computation results are revealed as follows:



```
parpool('local');

% Preallocate arrays for results
numPoints = length(freq);
magnitude = zeros(numPoints, 1);
phase = zeros(numPoints, 1);

% Parallel computation using parfor
parfor k = 1:numPoints
    magnitude(k) = 20 * log10(abs(H(k)));
    phase(k) = angle(H(k)) * (180 / pi);
end
```



```
% Plot parallel computation results
figure;
subplot(2,1,1);
semilogx(freq, magnitude); % Magnitude plot in dB
title('Bode Plot (Parallel Computation)');
ylabel('Magnitude (dB)');
grid on;

subplot(2,1,2);
semilogx(freq, phase); % Phase plot in degrees
ylabel('Phase (degrees)');
xlabel('Frequency (Hz)');
grid on;

% Shut down parallel pool
delete(gcp('nocreate'));
```

IV. CONCLUSION

The paper examined the efficiency of Data Intensive Computing (DIC) by using the existing dataset from the internet. Analysis of its usage efficiency was based on computing the Q-factor(quality factor) and resonance frequencies and plotting the graph for three configurations. Also, we computed mathematically Q-factor(quality factor) for the first configuration. The paper also showed how to perform parallel computing efficiency which is the one type of DIC and results showed more accurate plotting measures.

V. APPENDICES

TABLE I
FREQUENCY AND CONFIGURATION 1 DATA

Frequency (Hz)	Configuration 1
10	18.27
12.5	18.18
16	18.26
20	18.24
25	18.26
31.5	18.26
40	18.27
50	18.28
63	18.28
80	18.29
100	18.35
125	18.29
160	18.03
200	18.32
250	18.25
315	18.23
400	18.23
500	18.26
630	18.27
800	18.27
1000	18.32
1250	18.23
1600	18.24
2000	18.27
2500	18.32
3150	18.46
4000	19.01
5000	16.04
6300	23.23
8000	29.40
10000	46.10

REFERENCES

[1] D. Ozevin, ”Microelectromechanical systems for assessing and monitoring civil infrastructures,” in Sensor Technologies for Civil Infrastructures, Woodhead Publishing, 2022, pp. 237-272.

TABLE II
FREQUENCY AND CONFIGURATION 2 DATA

Index	Frequency (Hz)	Configuration 2
0	10	38.78
1	12.5	38.84
2	16	38.82
3	20	38.13
4	25	36.96
5	31.5	38.88
6	40	39.41
7	50	39.39
8	63	39.4
9	80	39.4
10	100	39.32
11	125	39.34
12	160	38.56
13	200	39.81
14	250	39.57
15	315	39.53
16	400	39.48
17	500	39.52
18	630	39.55
19	800	39.67
20	1000	39.82
21	1250	39.75
22	1600	39.99
23	2000	40.31
24	2500	40.75
25	3150	41.79
26	4000	44.2
27	5000	37.37
28	6300	59.3
29	8000	75.6
30	10000	146.2

TABLE III
FREQUENCY RESPONSE DATA FOR CONFIGURATION 3

Index	Frequency (Hz)	Configuration 3
0	10	39.95
1	12.5	39.85
2	16	39.39
3	20	39.34
4	25	39.37
5	31.5	39.42
6	40	39.43
7	50	39.45
8	63	39.46
9	80	39.49
10	100	39.47
11	125	39.43
12	160	38.78
13	200	39.8
14	250	39.43
15	315	39.46
16	400	39.43
17	500	39.45
18	630	39.53
19	800	39.54
20	1000	39.71
21	1250	39.69
22	1600	39.88
23	2000	40.24
24	2500	40.77
25	3150	41.37
26	4000	42.69
27	5000	46.07
28	6300	50.1
29	8000	63.9
30	10000	118.5

[2] A. Marra and C. Garcia, “Frequency response of high-frequency MEMS and IEPE accelerometers on different mountings,” Mendeley Data, Nov. 2022, doi: 10.17632/bj6scm24mg.1.

[3] J. Pisharath, ”Design and optimization of architectures for data intensive computing,” Northwestern University, 2005.

[4] I. Podder, T. Fischl, and U. Bub, ”Artificial intelligence applications for MEMS-based sensors and manufacturing process optimization,” in Telecom, vol. 4, no. 1, March 2023, pp. 165-197. MDPI.

[5] W. Zhang, J. Kim, K. A. Ross, E. Sedlar, and L. Stadler, ”Adaptive code generation for data-intensive analytics,” Proceedings of the VLDB Endowment, vol. 14, no. 6, pp. 929-942, 2021.

[6] J. Galvão, A. Leon, C. Costa, M. Y. Santos, and Ó. P. López, ”Automating data integration in adaptive and data-intensive information systems,” in European, Mediterranean, and Middle Eastern Conference on Information Systems, Nov. 2020, pp. 20-34, Springer International Publishing.