


# Comparative Analysis of Interpolation Techniques for FFT-Based Frequency Estimation


**Gamze Cabadag**

(Ankara Yıldırım Beyazıt University, Ankara, Turkey)

 <https://orcid.org/0000-0002-9338-7595>, [gamzecedadag@gmail.com](mailto:gamzecedadag@gmail.com)


**Ali Degirmenci**

(Ankara Yıldırım Beyazıt University, Ankara, Turkey)

 <https://orcid.org/0000-0001-9727-8559>, [alidegirmenci@aybu.edu.tr](mailto:alidegirmenci@aybu.edu.tr)

**Omer Karal**

(Ankara Yıldırım Beyazıt University, Ankara, Turkey)

 <https://orcid.org/0000-0001-8742-8189>, [omerkaral@aybu.edu.tr](mailto:omerkaral@aybu.edu.tr)

**Abstract:** Fast Fourier transform (FFT) is a widely used method for frequency estimation in electronic support systems. However, when the intermediate frequency (IF) of the radar signal is not an exact multiple of the FFT resolution, the correct frequency value cannot be obtained in the FFT computation. Therefore, interpolation methods are used to improve the frequency obtained from the FFT result. In this study, 12 different interpolation techniques (Jain, corrected Jain, Quinn, improved Quinn, Jacobsen, Macleod, Ding, Voglewede, mobile industrial (MI), Candan, rectangular-window-based interpolation, and Hanning window based interpolation) used in the literature have been extensively analyzed on radar signals contaminated with Laplace and Gaussian noise at different SNR values. In addition, in order to observe the performance of the techniques in different frequency bands, the bandwidth was changed to between 100 and 1000 MHz, and 100 Monte Carlo simulations were applied for each frequency. From the experimental analysis results, the improved Quinn technique showed the best performance for both noises. In addition to accuracy evaluations, the computational complexity of each interpolation technique was analyzed in terms of floating-point operations (FLOPs). The FLOPs cost of the FFT was uniformly included in all methods to ensure fair comparison. Results showed that while all techniques operate within a similar computational range, methods like Jain and Candan exhibit lower FLOPs costs, whereas the improved Quinn method, despite its higher complexity, achieves the best estimation accuracy.

**Keywords:** Fast Fourier Transform, Frequency Estimation, Gauss Noise, Interpolation, Laplace Noise

**Categories:** H.3.1, H.3.2, H.3.3, H.3.7, H.5.1

**DOI:** 10.3897/jucs.156911

## 1 Introduction

In electronic support systems, which is a sub-branch of electronic warfare, the extraction and examination of radar signal parameters are necessary for the detection and classification of threat radar. These parameters, called “pulse descriptor word” (PDW), include five types of information about the radar signal: time of arrival (TOA); pulse width (PW); pulse amplitude (PA); carrier frequency; and angle of arrival (AOA). The extraction of

the frequency parameter is of great importance in many fields such as power electronics and mechanical applications [Santamaria et al. 2000, Martinez et al. 2019, Zhang et al. 2021, Huerta-Lopez et al. 2000], but it is also important in electronic warfare applications for accurate PDW extraction and detection of threat radar [Ortatatlı et al. 2016, Tsui 1995].

Frequency estimation methods are basically divided into two different computational domains, i.e., time domain and frequency domain. While methods in the time domain are based on the analysis of sequential signals, methods in the frequency domain are based on the analysis of various signal spectra. In addition, estimation methods in the time domain are divided into two groups, i.e., parametric and nonparametric. Nonparametric methods include Pisarenko's, multiple signal classification (MUSIC), estimation of signal parameters via rotational invariance techniques (ESPRIT), minimum norm (MN); parametric methods include autoregressive (AR), moving average, and Prony's—all of which can be given as examples. Although estimation methods in the time domain provide high accuracy, these methods are difficult to apply to real-time systems due to computation complexities. For this reason, estimation methods in the frequency domain are preferred in real-time systems.

FFT, a faster version of the discrete Fourier transform (DFT) [Gasior and Gonzalez 2000, Polap 2018], is the most widely used frequency-based estimation method because it requires less computation. In FFT, the frequency resolution is the ratio of the sampling frequency to the number of FFT points. Frequency resolution can be improved by increasing the number of FFT points; however, this requires more computations and time delay for real time applications. Therefore, interpolation techniques can be used to increase the frequency accuracy without increasing the number of FFT points. The literature has investigated various methods to improve frequency accuracy based on interpolation. However, the performance of the methods has not been extensively analyzed for different frequency bands under different noise distributions. In this study, the performances of 12 different interpolation techniques (Jain, corrected Jain, Quinn, improved Quinn, Jacobsen, Macleod, Ding, Voglewede, mobile industrial (MI), Candan, rectangular-window-based interpolation, and Hanning window-based interpolation) are analyzed. Unlike in the literature, the signal contaminated by Gaussian and Laplace noise and signal-to-noise ratio (SNR) values changed from 2 to 30 dB. In addition, to observe the performance of the methods in different frequency bands, the bandwidth was changed to between 100 and 1000 MHz, and 100 Monte Carlo simulations were applied for each frequency.

The rest of this work is shown as follows: In Section 2, a detailed summary of the current literature review is presented. In Section 3, the methods are explained theoretically, and their formulas are given. In Section 4, comprehensive simulations are made, and analysis results and interpretations are shared. Finally, in Section 5, concluding remarks and issues related to future studies are presented.

## 2 Related Works

FFT is a DFT-based method used to compute the frequency-domain equivalent of signals in the time domain [Ai and Lomakin 2024, Cavado 2024, Jiao et al. 2023, Rajaby and Sayedi2022]. In the frequency domain, the frequency resolution is computed as the ratio of the sampling frequency to the number of FFT samples. After the signal of the threat radar is converted into an IF signal by the radar receiver, the extraction of frequency and other PDW parameters begins. When the IF frequency of the threat radar is not

an exact multiple of the FFT resolution, the frequency cannot be accurately estimated because it will appear in the frequency bins. Therefore, interpolation methods are used to improve the FFT result. In the literature, researchers have recently used different interpolation methods. For example, Gasior and Gonzalez [2000] preferred parabolic and Gaussian interpolation methods. Also, in the time domain Rectangular, Triangular, Hann, Hamming, Blackman, Blackman–Harris, Nuttall, Blackman–Harris–Nuttall, and Gaussian windows were applied, respectively, before Gaussian and parabolic interpolation methods. A performance comparison was made with a gain factor, which was computed as the ratio of the largest error without interpolation to the largest error after interpolation. From the simulation results, both methods increase the frequency resolution; however, the Gaussian method gives better results than the parabolic. These methods have the lowest error rate and the highest gain factor in the Gaussian window. When other windows are applied, the gain factor can be sorted from lowest to the highest as follows: rectangular; triangular; Hann; Hamming; Blackman; Blackman–Harris; Nuttall, Blackman–Harris–Nuttall; and Gaussian.

Candan [2011] presented a new formula derived from the Jacobsen interpolation method. Parabolic, Quinn, Macleod, Jacobsen, and a recommended method named as “Jacobsen with bias correction” were compared in terms of bias and root mean square error (RMSE). In the absence of noise, and the number of FFT points ( $N$ ) is 8, the least-biased estimator is Jacobsen, and the most-biased method is parabolic interpolation; other methods have approximately similar results. According to the author, for low/medium  $N$  values, the proposed method will give accurate results in radar signal processing applications such as Doppler frequency estimation at high SNR values.

Candan, in his another study Candan [2013], performed a detailed bias analysis in order to eliminate the bias effect in the Jacobsen with the bias-correction method [Candan 2011]. Using the result of bias analysis, a bias-reducing stage, following the correction stage, has been proposed. This stage successfully increases the performance at high SNR values.

Since Gasior and Gonzalez [2000] used the signal processed in a rectangular window in their study, Candan [2015] eliminated the need to use rectangular windows by adapting the deviation correction according to the selected window type. Although most of the frequency estimators in the literature are derived for specific window types, in the Duda [2011]’s study, bias correction can be adapted to different windows. The advantage of Candan’s study over Duda’s work, which requires large-degree polynomial calculations, is that it improves performance without increasing computational complexity. The main idea of Candan’s suggested method is to use the proposed estimator twice at different operating points. The same idea is seen in the estimator proposed by Aboutanios and Mulgrew [2005] for the rectangular-window data. Both studies aim to remove estimator bias by moving the operating point. Numerical results in three sets were presented to examine the performance of the proposed method. The first set presents the performance of the proposed method over a wide range of SNR, the second examines an RMSE comparison of the estimator for different windows, and the third analyses the effect of zero-padding [Candan 2013].

Iglesias et al. [2014] applied the zero-padding method before the FFT operation, and frequency was estimated using parabolic interpolation. The authors concluded that parabolic interpolation decreases the RMSE in regard to taking directly the most powerful FFT bin. Arva et al. [2017] examined the MI method, developed by a Romanian company. Since the Jacobsen method gives accurate results compared with other methods in the literature, the MI and the Jacobsen methods were compared. An FFT analyzer was used for verification of methods based on the experimental results; however, the MI method

is more accurate than the Jacobsen method.

Niranjan et al. [2021] examined the rectangular-window-based, Hamming-window-based, and curve-fitting-based interpolation methods. Further, to observe the effect of windowing before FFT, the authors applied the Hamming window in the time domain before curve-fitting interpolation, as the authors mentioned that frequency resolution can be raised by increasing the number of FFT points. In order to show that the frequency improves as FFT points increase, the authors applied the four different methods by choosing the number of FFT points as 256, 512, 1024, 2048, and 4096, respectively. Frequency range was changed from 1100 to 1120 MHz with 0.5 MHz resolution, and a performance comparison was tested at each FFT point for each method. From the presented results, it is obviously seen that RMSE and peak error decreases as the number of FFT points increases for FFT measured frequency and each interpolation method.

RMSE and peak error were the lowest in the case of the curve-fitting interpolation technique with the Hamming window. The error rates of the remaining methods can be listed from least to most: curve-fitting-based, Hamming-window-based, and rectangular-window-based. As seen in the simulation results, as the FFT points increase, the frequency can be improved without the need for interpolation methods; however, this increases the mathematical operations and requires hardware support.

In the literature, although some authors directly used well-known interpolation methods, others made changes to obtain a more accurate frequency value. The improved Quinn method, [Quinn 1994] which is also called Quinn's second estimator, can be given as an example of these methods, which are obtained by improving the existing one. Also, Minda et al. [2020] made changes in the Jain interpolation method procedure since the frequency result deviates from the actual value if the two points used for the computation of the Jain interpolation method are not in the same lobe. Therefore, the authors proposed a new method that selects two relevant points from the same lobe. The zero-padding method was also used in the procedure of this method, which is called the "corrected Jain's algorithm". According to the results of observations made at different signal time lengths by using the Jain and corrected Jain's algorithm, it can be seen that the error in the frequency estimation decreases as the signal time length increases for both methods; however, the corrected Jain's algorithm has a lower error rate than the Jain's algorithm.

Minda et al. [2020] compared five different interpolation methods regarding their error rate. The methods used in this study were divided into two categories, depending on the number of points required for interpolation and methods that use complex or real parts of the FFT result for the interpolation computation. For real part methods, two-point Quinn and three-point Jacobsen methods were applied. For the complex part category, two-point Jain and three-point Ding and Voglewede methods were performed. In the first step, Quinn, Jacobsen, Jain, Ding, and Voglewede methods were compared according to their error rates. Based on the simulation results, the Jacobsen method has the lowest error rate, Quinn has the second-lowest error rate, and Jainn, Ding, and Voglewede have approximately similar error results. In the second stage, the authors changed the Jain interpolation method with the corrected Jain's algorithm proposed in [Minda et al. 2020]. Although it cannot be said clearly which interpolation method gives more accurate results, it is seen that each method improves the frequency at different time intervals. However, the algorithm of Jacobsen and that of Quinn provide the most accurate results regardless of signal length.

### 3 Material and Methods

FFT-based digital receivers are widely used in electronic support systems. An intermediate frequency (IF) signal coming from the radar receiver's RF chain block is digitized with a high-speed analog-to-digital converter (ADC) [Niranjan et al. 2021]. FFT can be used to find the frequency of digitized signals; since an exact frequency value is often not found due to frequency resolution; however, the frequency value found in the FFT result can be improved using interpolation methods. Fig. 1 shows the block diagram of the interpolation process.

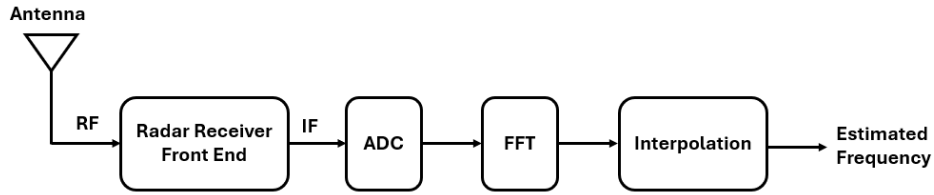


Figure 1: Block diagram of interpolation technique applied frequency estimation.

If a sampled sequence of a sinusoidal signal is  $g[n]$ ,

$$g[n] = G \exp \left[ j2\pi \frac{f_0}{f_s} n \right], \quad n = 0, \dots, N - 1 \quad (1)$$

where  $G$  is the amplitude,  $f_0$  is the signal frequency;  $f_s$  is sampling frequency,  $N$  number of FFT points, and  $n$  is the index of the samples. Frequency resolution, that is, the distance between two frequency bins, is as shown in Formula (2) [Minda et al. 2020].

$$\Delta f = \frac{f_s}{N} \quad (2)$$

The discrete Fourier transform of the  $G[n]$  at  $k$  bin shows the amplitude of this signal; the frequency spectrum of FFT is shown in Fig. 2 [Minda et al. 2020].

$$G_k = \sum_{n=0}^{N-1} g[n] e^{-j(2\pi/N)nk}, \quad G_k = \text{Re}G_k + j\text{Im}G_k \quad (3)$$

$$\text{Re}G_k = \sum_{n=0}^{N-1} g[n] \cos\left(\frac{2\pi}{N}nk\right), \quad (4)$$

$$\text{Im}G_k = - \sum_{n=0}^{N-1} g[n] \sin\left(\frac{2\pi}{N}nk\right)$$

$$G_k = \text{Re}G_k + \text{Im}G_k \quad (5)$$

$$|G_k| = \sqrt{(\text{Re}G_k)^2 + (\text{Im}G_k)^2} \quad (6)$$

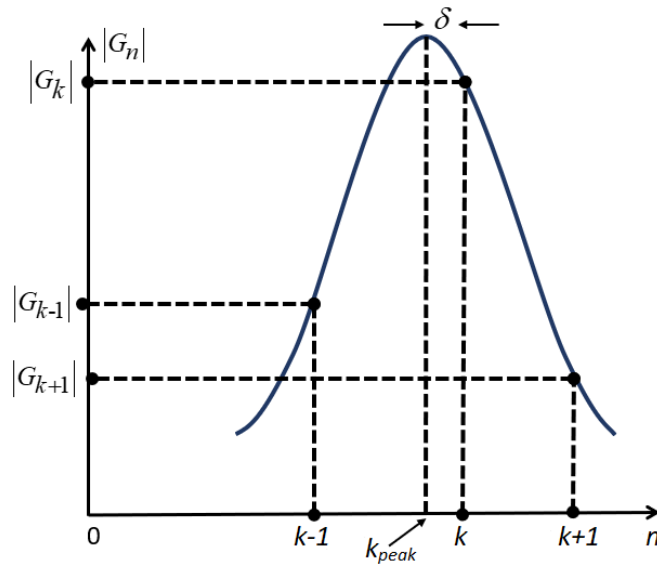


Figure 2: Frequency spectrum of FFT.

After formulas (3-6) [Minda et al. 2020], calculations are completed and  $G_k$  values are found; in order to improve the frequency, a two-step process is used. In the first stage, as a result of the standard FFT process, it is also named “coarse estimation”, the value of  $k$  that maximises the value of  $G_k$  and its neighbourhoods are found. In the second stage, with the help of the formulas given in Table 1 and Table 2, the correction coefficient value is calculated for each method. By using the calculated correction coefficient value  $\delta$ , the improved frequency value can be recalculated using Formula (7) [Minda et al. 2020].

$$f_c = (k + \delta)\Delta f \quad (7)$$

The common purpose of all interpolation techniques is to improve the frequency result; the difference from each other is determined via the correction coefficient calculation methods. In general, methods differ in that they use the magnitude of complex  $G_k$  or only magnitude of the real part of  $G_k$  and number of neighbours used for interpolation. For example, Quinn and Jacobsen use only the magnitude of the real part of  $G_k$ ; however, Jain, Ding, and Voglewede use the magnitude of complex part of  $G_k$ . Also, Quinn and Jain methods use  $G_k$  and its one neighbour value in the correction coefficient calculation, while the Jacobsen, Ding, and Voglewede methods use  $G_k$  and its two neighbour values. Twelve methods used in this work are divided into two groups according to their use of complex magnitude or real magnitudes of  $G_k$  while calculating the correction coefficient. The first group, i.e., the complex part magnitude group, includes Jain, improved Jain, Ding, Voglewede, MI, Hanning window-based, and rectangular-window-based methods; the method formulas are given in Table 1. The formulas belonging to the second group consisting of Quinn, improved Quinn, Jacobsen, Macleod, and Candan methods, which calculate the correction coefficient with real part magnitude, are given in Table 2.

Interpolation Method Name	Formula
<b>Jain Method</b> [Minda et al. 2020]	$\text{if } G_{[k-1]} > G_{[k+1]}$ $a = \frac{G_{[k]}}{G_{[k-1]}}, \delta = \frac{a}{a+1}, \quad f_{\text{jain}} = (k-1 + \delta) \frac{f_s}{N}$ $\text{else}$ $a = \frac{G_{[k+1]}}{G_{[k]}}, \delta = \frac{a}{a+1}, \quad f_{\text{jain}} = (k + \delta) \frac{f_s}{N}$
<b>Corrected Jain Method</b> [Minda et al. 2020]	<ul style="list-style-type: none"> <li>– <b>FFT calculated for original signal and coordinates of the maximiser and its neighbors are found.</b></li> <li>– <b>For zero-padding, 2 points zero with zero amplitude added to the original signal.</b></li> <li>– <b>FFT calculated for zero-padded signal and coordinates are determined for maximiser and its neighbors.</b></li> <li>– <b>To select proper neighbor of maximiser,</b>  <math display="block">\text{If } G_{[k*]} &lt; G_{[k]}, \text{ (where } G_{[k*]} \text{ the amplitude of maximizer for zero - padded signal)}</math> </li> </ul> $a = \frac{G_{[k+1]}}{G_{[k]}}, \delta = \frac{a}{a+1}, \quad f_{\text{imp\_jain}} = (k + \delta) \frac{f_s}{N}$ $\text{else}$ $a = \frac{G_{[k]}}{G_{[k-1]}}, \delta = \frac{a}{a+1}, \quad f_{\text{imp\_jain}} = (k-1 + \delta) \frac{f_s}{N}$
<b>Ding Method</b> [Minda et al. 2020]	$\delta = \frac{G_{[k+1]} - G_{[k-1]}}{G_{[k-1]} + G_{[k]} + G_{[k+1]}}, \quad f_{\text{ding}} = \frac{f_s}{N} (k + \delta)$
<b>Voglewede Method</b> [Minda et al. 2020]	$\delta = \frac{G_{[k+1]} - G_{[k-1]}}{2(G_{[k]} - G_{[k-1]} - G_{[k+1]})}, \quad f_{\text{voglewede}} = \frac{f_s}{N} (k + \delta)$
<b>MI Method</b> [Arva et al. 2017]	$U = \frac{G_{[k]}}{G_{[k+1]}}, L = \frac{G_{[k]}}{G_{[k-1]}}$ $\text{if } U = L$ $dF = 0$ $\text{else}$ $\text{if } U < L$ $dF = (2 - U) / (1 + U)$ $\text{else - if } U > L$ $dF = (L - 2) / (1 + L)$ $\text{end}$ $\delta = dF,$ $f_{\text{arva}} = \frac{f_s}{N} (k + \delta)$
<b>Hanning Window-Based Method</b> [Salih 2012]	$\delta = \frac{2G_{[k+1]} - G_{[k]}}{G_{[k]} + G_{[k+1]}}, \quad f_{\text{hanning}} = \frac{f_s}{N} (k + \delta)$
<b>Rectangular-Window-Based Method</b> [Salih 2012]	$\delta = \frac{G_{[k+1]} + G_{[k-1]}}{2G_{[k]} - G_{[k-1]} + G_{[k+1]}}, \quad f_{\text{rectangular}} = \frac{f_s}{N} (k + \delta)$

Table 1: Interpolation methods based on complex part

## 4 Simulation Results and Discussion

In this section, 12 estimation methods are compared with respect to accuracy, and the simulation results are validated with the help of MATLAB 2019a. To verify accuracy of the methods, the frequency of the input signal changed between 100 to 1000 MHz. Also, in this study, the sampling frequency is 5 GHz, and the number of FFT points is 256 [Kulkarni et al. 2013].

### 4.1 Evaluation Criteria

The RMSE, which is widely used in the literature, uses the Euclidean distance to show how far the predicted values deviate from the actual values. The RMSE can be expressed

Interpolation Method Name	Formula
<b>Quinn Method</b> [Duda 2011]	$- a_1 = \text{real} \left\{ \frac{G_{[k-1]}}{G_{[k]}} \right\} \quad a_2 = \text{real} \left\{ \frac{G_{[k+1]}}{Y_{[k]}} \right\}$ $- \delta_1 = \frac{a_1}{1-a_1} \quad \text{and} \quad \delta_2 = \frac{-a_2}{1-a_2}$ $- \text{if } \delta_1 > 0 \text{ and } \delta_2 > 0, \delta = \delta_2$ $- \text{else, } \delta = \delta_1$ $- f_{\text{quinn}} = \frac{f_s}{N}(k + \delta)$
<b>Improved Quinn Method</b> [Quinn 1994]	$- a_1 = \text{real} \left\{ \frac{G_{[k-1]}}{G_{[k]}} \right\} \quad a_2 = \text{real} \left\{ \frac{G_{[k+1]}}{G_{[k]}} \right\}$ $- \delta_1 = \frac{a_1}{1-a_1} \quad \text{and} \quad \delta_2 = \frac{-a_2}{1-a_2}$ $- \delta = \frac{\delta_1 + \delta_2}{2} - K(\delta_1^2) + K(\delta_2^2)$ $- K(x) = \frac{1}{4} \log(3x^2 + 6x + 1) - \frac{\sqrt{6}}{24} \log\left(\frac{x+1-\frac{\sqrt{2}}{3}}{x+1+\frac{\sqrt{2}}{3}}\right)$ $- f_{\text{imp\_quinn}} = \frac{f_s}{N}(k + \delta)$
<b>Jacobsen Method</b> [Candan 2013]	$\delta = \text{real} \left\{ \frac{G_{[k-1]} - G_{[k+1]}}{2G_{[k]} - G_{[k-1]} - G_{[k+1]}} \right\}, \quad f_{\text{jacobsen}} = \frac{f_s}{N}(k + \delta)$
<b>Macleod Method</b> [Candan 2013]	$\tau = G_k$ $R_m = \text{real} \{ Y_m \tau^* \}, \quad m = k-1, k, k+1$ $\gamma = \frac{R_{k-1} - R_{k+1}}{2R_k + R_{k-1} + R_{k+1}} \quad \text{and} \quad \delta = \frac{\sqrt{1+8\gamma^2}-1}{4\gamma}$ $f_{\text{macleod}} = \frac{f_s}{N}(k + \delta)$
<b>Candan Method</b> [Candan 2011]	$\delta = \frac{\tan(\pi/N)}{\pi/N} \cdot \text{real} \left\{ \frac{G_{[k-1]} - G_{[k+1]}}{2G_{[k]} - G_{[k-1]} - G_{[k+1]}} \right\}$ $f_{\text{candan}} = \frac{f_s}{N}(k + \delta)$

Table 2: Interpolation methods based on real part

as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^N \| a(i) - \hat{a} \|^2}{N}} \quad (8)$$

where  $a$  is original value,  $\hat{a}$  is predicted value, and  $N$  is the number of samples.

In performance-critical signal processing systems such as radar and electronic support applications, computational efficiency is a key metric alongside estimation accuracy. For this purpose, the complexity of each interpolation method in terms of floating point operations (FLOPs) is evaluated, a standard unit of computational cost that represents basic arithmetic operations such as addition, multiplication, division, and square root. While accuracy metrics such as RMSE reflect estimation performance, FLOP counts provide insight into algorithmic feasibility for real-time or embedded implementations. In this study, we estimate the FLOP requirements of each interpolation method based on the dominant arithmetic expressions involved in peak-based frequency estimation. This analysis helps balance the trade-off between accuracy and implementation cost, particularly in resource-constrained radar processing systems.

#### 4.2 Contamination of Signal by Different Noise

When a signal with any frequency comes to the radar receiver, it includes not only its own frequency but also the noise. Estimated frequency may deviate from its actual value due to noise. Since the type of the noises found in the environment and their source are not known in advance, different kinds of noises are applied to the signal before the frequency estimation process [Koziarski et al. 2020, Blinowski and Szczypiorski 2017]. Since Gaussian noise is often used in other studies, in this study, in addition to Gaussian noise, Laplace noise was also applied at different SNR levels. Although similar to the normal distribution, the Laplace distribution has steeper structures [Karal 2017, Zhang et al. 2021, Cabadag and Karal 2022]. To show that these two distributions are different from each other, formulas for the Gaussian and Laplace distributions are given in Formulas (9-10); the visuals of these two noise distributions at the same variance and mean are plotted in Fig. 3.

$$f_{Gauss}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (9)$$

$$f_{Laplace}(x) = \frac{1}{2\lambda} e^{-\frac{|x-\mu|}{\lambda}} \quad (10)$$

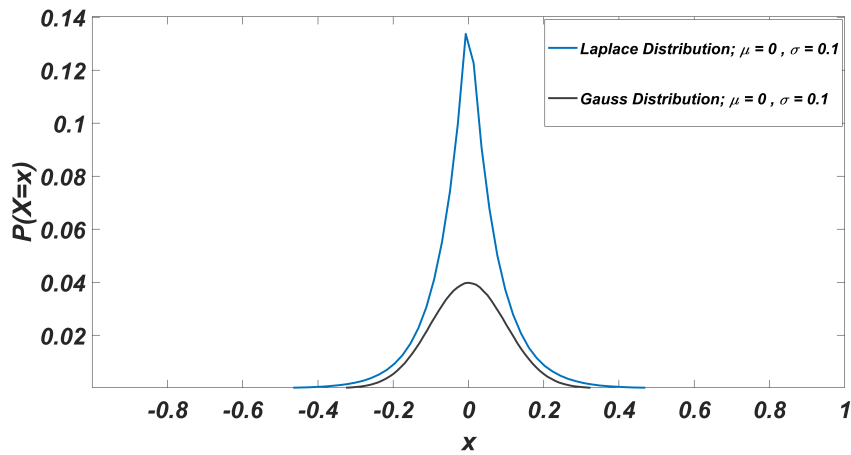


Figure 3: Probability density functions of Gaussian and Laplace distribution

#### 4.3 Simulation Results

In order to analyze the effect of the noise and different frequencies of each method more clearly, analyses were made at 2, 10, and 30 dB SNR values of the noises that have zero mean and variable variance. For each frequency and SNR value, 100 Monte Carlo simulations were applied. The errors of estimated frequencies are listed in Table 3, Table 4, Table 5, Table 6, and Table 7. In order to see the effect of the applied interpolation

method on the FFT result, first, the FFT result without interpolation is given in a related table, and these results are compared with the interpolation method result. In the tables, the unit for SNR is dB, and all of the frequency errors are MHz.

SNR	Noise model	FFT	Jain	C. Jain	Quinn	L. Quinn	Jacobsen	Macleod	Ding	Voglewede	MI	Candan	Rectangular	Hanning
30	Gaussian	2.343	<b>0.015</b>	0.989	<b>0.015</b>	0.047	0.024	0.052	1.572	2.071	14.38	2.343	0.023	14.38
	Laplace	2.343	<b>0.013</b>	0.991	<b>0.015</b>	0.042	0.024	0.049	1.572	2.070	14.38	2.343	0.024	14.38
10	Gaussian	2.343	0.660	0.944	0.300	0.233	<b>0.024</b>	0.545	1.584	2.074	14.05	2.343	0.027	14.52
	Laplace	2.343	0.809	1.000	0.306	0.247	<b>0.025</b>	0.553	1.625	2.086	13.72	2.343	0.251	14.22
2	Gaussian	2.343	2.620	1.324	1.184	0.875	0.921	2.281	2.005	2.146	10.73	2.343	<b>0.847</b>	13.46
	Laplace	2.343	2.384	1.332	1.163	<b>0.822</b>	0.891	2.413	1.859	2.090	11.04	2.343	1.079	13.23

Table 3: RMSE results for 100 MHz

SNR	Noise model	FFT	Jain	C. Jain	Quinn	L. Quinn	Jacobsen	Macleod	Ding	Voglewede	MI	Candan	Rectangular	Hanning
30	Gaussian	7.031	0.045	2.312	0.051	0.046	<b>0.025</b>	0.067	3.942	4.635	5.618	7.031	<b>0.026</b>	5.618
	Laplace	7.031	<b>0.045</b>	2.318	0.049	0.041	<b>0.025</b>	0.064	3.939	4.634	5.604	7.031	<b>0.025</b>	5.604
10	Gaussian	7.031	<b>0.189</b>	2.906	<b>0.189</b>	0.195	0.320	0.211	3.922	4.604	5.555	7.031	0.318	5.555
	Laplace	7.031	0.193	2.467	0.191	<b>0.189</b>	0.283	0.204	3.929	4.620	5.580	7.031	0.287	5.580
2	Gaussian	7.031	1.107	5.480	<b>0.699</b>	0.745	1.138	0.807	4.264	4.803	5.689	7.031	1.102	5.933
	Laplace	7.031	0.992	5.230	0.779	<b>0.787</b>	1.130	0.865	4.207	4.450	5.566	7.031	1.127	5.636

Table 4: RMSE results for 300 MHz

SNR	Noise model	FFT	Jain	C. Jain	Quinn	L. Quinn	Jacobsen	Macleod	Ding	Voglewede	MI	Candan	Rectangular	Hanning
30	Gaussian	7.812	<b>0.011</b>	3.692	<b>0.011</b>	0.012	0.026	0.102	3.818	4.042	3.604	7.812	19.55	1.112
	Laplace	7.812	<b>0.010</b>	3.693	0.011	0.018	0.024	0.102	3.816	4.042	3.605	7.812	19.55	1.093
10	Gaussian	7.812	0.211	3.618	0.207	<b>0.199</b>	0.282	0.204	3.756	3.945	3.642	7.812	19.53	1.318
	Laplace	7.812	0.201	3.679	0.199	<b>0.196</b>	0.294	0.217	3.798	4.025	3.628	7.812	19.54	1.082
2	Gaussian	7.812	0.856	3.623	<b>0.734</b>	0.743	1.145	0.778	4.058	3.989	3.870	7.812	20.03	3.450
	Laplace	7.812	<b>0.728</b>	3.610	0.768	0.827	1.292	0.802	4.037	3.917	3.738	7.812	19.91	3.281

Table 5: RMSE results for 500 MHz

#### 4.4 Discussion

In this study, performance of the interpolation algorithms was compared by RMSE evaluation criteria. Although the interpolation methods used in the study generally improved the FFT result, some interpolation methods increased the error in the FFT result even more at some frequency values. To better compare the techniques, Gaussian

SNR	Noise model	FFT	Jain	C. Jain	Quinn	L. Quinn	Jacobsen	Macleod	Ding	Voglewede	MI	Candan	Rectangular	Hanning
30	Gaussian	3.125	0.022	1.359	0.022	<b>0.019</b>	0.020	0.062	2.329	2.809	13.24	3.125	6.421	9.338
	Laplace	3.125	0.026	1.366	0.025	0.211	<b>0.021</b>	0.063	2.328	2.808	13.22	3.125	6.430	9.320
10	Gaussian	3.125	0.351	1.623	0.226	<b>0.210</b>	0.252	0.407	2.322	2.806	13.11	3.125	6.422	9.367
	Laplace	3.125	0.510	1.491	0.222	<b>0.205</b>	0.236	0.422	2.360	2.817	12.73	3.125	6.523	9.082
2	Gaussian	3.125	2.457	2.826	<b>0.889</b>	0.904	1.052	1.969	2.296	2.746	9.776	3.125	6.954	8.667
	Laplace	3.125	2.903	2.904	0.959	<b>0.788</b>	0.848	1.960	2.402	2.798	9.085	3.125	7.082	8.074

Table 6: RMSE results for 700 MHz

SNR	Noise model	FFT	Jain	C. Jain	Quinn	L. Quinn	Jacobsen	Macleod	Ding	Voglewede	MI	Candan	Rectangular	Hanning
30	Gaussian	3.906	0.040	19.91	0.040	0.026	<b>0.022</b>	0.049	2.685	3.359	11.59	3.906	0.022	11.59
	Laplace	3.906	0.044	19.90	0.045	0.030	<b>0.020</b>	0.053	2.683	3.358	11.58	3.906	0.020	11.58
10	Gaussian	3.906	0.215	19.98	0.212	<b>0.194</b>	0.210	0.384	2.698	3.362	11.57	3.906	0.214	11.57
	Laplace	3.906	0.239	19.94	0.239	0.228	0.257	0.385	2.723	3.371	11.55	3.906	<b>0.216</b>	11.55
2	Gaussian	3.906	2.222	19.99	1.029	<b>0.821</b>	0.930	1.274	2.794	3.380	91.93	3.906	0.832	11.49
	Laplace	3.906	2.919	19.46	0.997	<b>0.823</b>	1.011	1.409	2.936	3.417	8.103	3.906	1.076	10.93

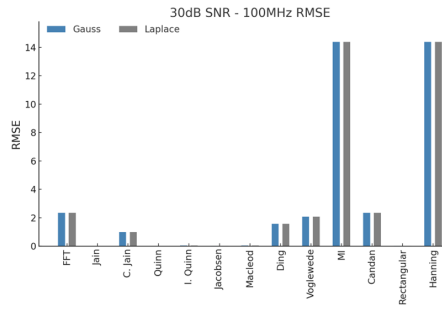
Table 7: RMSE results for 1000 MHz

and Laplace noises were added at different SNR levels. Further, in order to obtain more accurate results, 100 Monte Carlo simulations were performed for each frequency at each added SNR level. In order to see at which frequency the methods give more accurate results, 100, 300, 500, 700, and 1000 MHz frequency values were selected from the 100 to 1000 MHz bandwidth. It is impossible to directly interpret which method gives the best results because each can provide better results than the other at different SNR and frequency values. Therefore, the errors obtained for each frequency value are analyzed in separate tables.

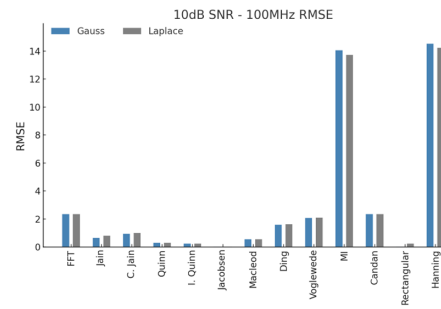
As shown in Table 3, for the 100 MHz frequency, the performance of interpolation methods varies considerably depending on the SNR level and noise model. At 30 dB SNR, the Jain and Quinn methods provide the lowest RMSE under both Gaussian and Laplacian noise, significantly outperforming the classical FFT estimate. Notably, Jacobsen and Rectangular-window-based interpolation also show competitive performance with RMSE values around, while Hanning and MI perform poorly with RMSEs exceeding 14.3MHz. At 10 dB SNR, Jacobsen achieve the lowest RMSE, followed closely by Rectangular and Improved Quinn. However, Jain's accuracy drops significantly, particularly under Laplacian noise, indicating its sensitivity to non-Gaussian conditions. Under 2 dB SNR, the estimation errors increase for all methods, but Improved Quinn, Jacobsen, and Macleod still provide notably better results compared to the raw FFT. Window-based estimators such as Hanning and MI exhibit large errors, indicating poor robustness in low-SNR, high-noise scenarios. In conclusion, for 100 MHz, Jacobsen, Improved Quinn, and Rectangular-window-based interpolation methods offer a favorable balance between accuracy and robustness, making them suitable candidates across varying SNR levels.

According to Table 4, for 300 MHz frequency value, except for the Candan method, all other methods improved the frequency value compared with the FFT result without interpolation. Different methods performed well at different SNR levels. As SNR levels decrease, similar to 100 MHz, the performance of the interpolation methods began to deteriorate, except for the Hanning window-based interpolation, Candan, MI, Voglewede, and Ding methods.

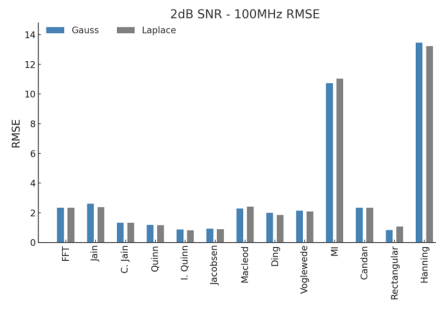
As shown in Table 5, for the 500 MHz frequency, the performance of interpolation



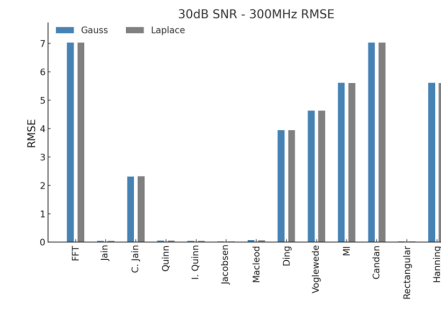
(a)



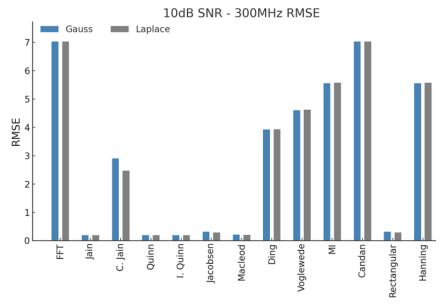
(b)



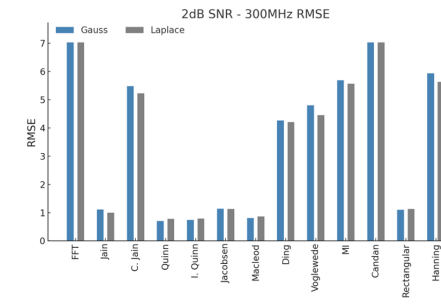
(c)



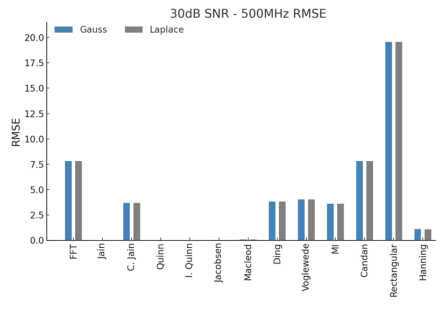
(d)



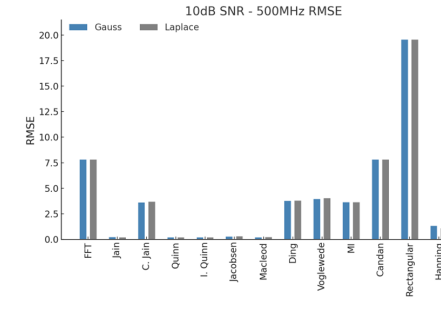
(e)



(f)

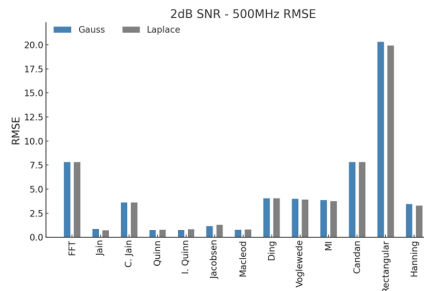


(g)

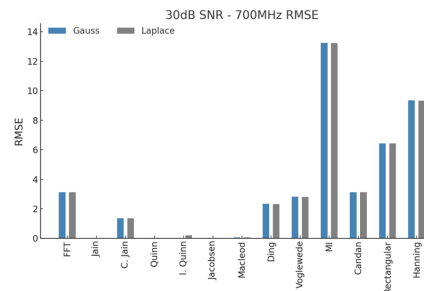


(h)

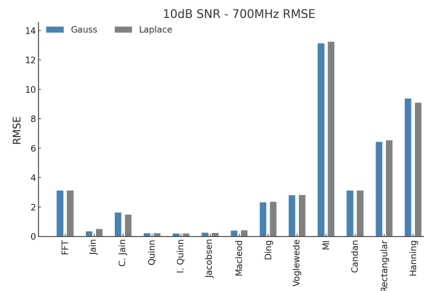
Figure 4: RMSE comparison of frequency estimation methods under Gaussian (Gauss) and Laplacian (Laplace) noise across varying SNR levels and center frequencies



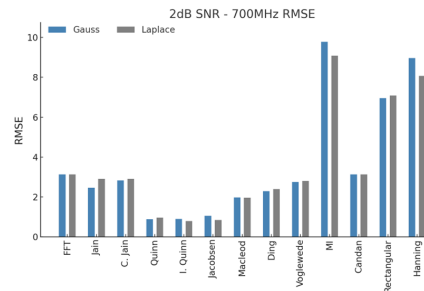
(i)



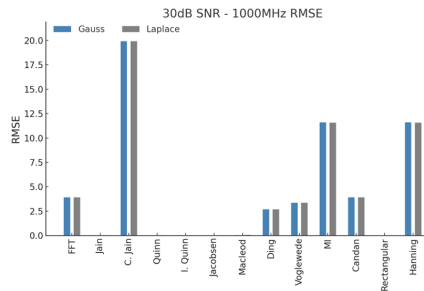
(j)



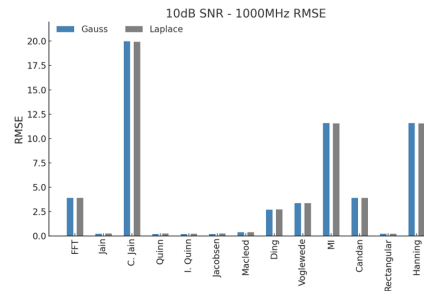
(k)



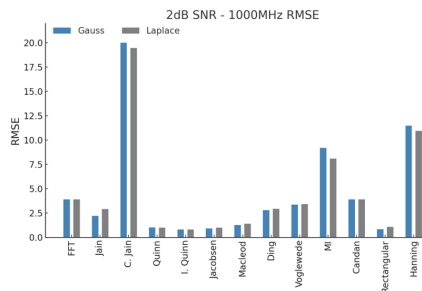
(l)



(m)



(n)



(o)

Figure 4: (continued) RMSE comparison of frequency estimation methods under Gaussian (Gauss) and Laplacian (Laplace) noise across varying SNR levels and center frequencies

methods demonstrates clear sensitivity to the SNR level and noise distribution. At 30 dB SNR, the Jain and Quinn methods exhibit the lowest RMSE values under both Gaussian and Laplacian noise, significantly outperforming the baseline FFT estimate Improved Quinn follows closely, while Jacobsen and Macleod remain competitive with RMSE values below 0.1 MHz. In contrast, window-based estimators such as Rectangular and Hanning yield substantially higher error rates, with MI also performing poorly. At 10 dB SNR, interpolation-based estimators continue to outperform the FFT. The Improved Quinn method achieves the best accuracy, followed by Quinn and Jacobsen, while C. Jain suffers a moderate degradation in performance. Again, MI, Rectangular, and Candan produce the highest error values, indicating poor robustness under mid-level SNR conditions. At 2 dB SNR, as expected, the estimation errors increase significantly across all methods. However, Improved Quinn, Quinn, and Jacobsen still offer lower RMSE compared to the FFT, with values. Meanwhile, window-based approaches and Candan perform poorly, indicating limited suitability in low-SNR scenarios. In summary, for 500 MHz, Jain, Quinn, Improved Quinn, and Jacobsen are consistently the most effective interpolation methods across varying SNR levels, with Improved Quinn providing a favorable trade-off between accuracy and robustness. Conversely, window-based estimators are not recommended due to their consistently high estimation errors.

According to Table 6, for 700 MHz, the MI, rectangular, and Hanning methods increased the error rate, and the improved Quinn showed the best performance. Although the Jacobsen and Quinn methods are not as accurate as the improved Quinn, performance of these three methods is almost the same.

When Table 7 is examined, rectangular, Jacobsen, and improved Quinn show similar performance for 1000 MHz. The MI and Hannig were the most successful methods.

The bar charts in Figures 4a-4o illustrate the RMSE performance of 13 different frequency estimation techniques under Gaussian and Laplacian noise at multiple SNR levels and center frequencies. As expected, increasing noise power (i.e., decreasing SNR) leads to a noticeable degradation in estimation accuracy across all methods. In addition to the numerical RMSE values summarized in Tables 3-7, bar chart visualizations were included to provide a clearer comparative illustration of frequency estimation errors across different methods, noise models, and SNR levels. As observed from the bar charts, the interpolation method that yields the lowest frequency estimation error varies depending on the center frequency and the SNR level. While the optimal interpolation method in terms of RMSE varies depending on the frequency and SNR, the methods proposed by Jain, Quinn, Improved Quinn, Jacobsen, and Macleod tend to outperform others across a wide range of conditions.

All the tables, show that the result of the Candan method does not change according to the SNR, and it is directly the same as the FFT result. Since it was stated that the Candan method should be used with a rectangular window [Candan 2011], this method was also performed after multiplying the signal with a rectangular window in the time domain, but there was no change in the result. Minda et al. [2020] analyzed five different methods, including Jacobsen and Quinn methods, and, based on the simulation results, the Jacobsen and Quinn methods perform better than others. This study, compared the 12 methods, including Jacobsen, Quinn, and improved Quinn. When the simulation results were analyzed, the improved Quinn performed better than the Jacobsen and Quinn. It was also observed that the performance of the interpolation methods did not differ much compared with the Gaussian or Laplace noise. In Fig. 5, the RMSE versus SNR graph shows how the RMSE value changes as the SNR level increases. Although there was no noticeable change in the RMSE value as the SNR increased in the Hanning, MI, corrected Jain, rectangular, Candan, and Ding methods, it was observed that the error

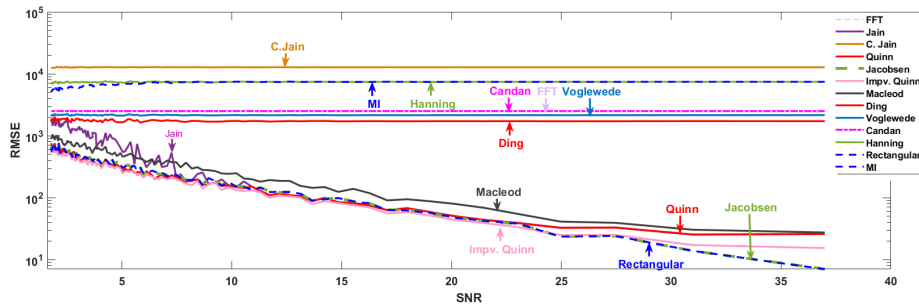


Figure 5: RMSE change according to SNR for 12 methods

rate decreased as the SNR increased in other methods.

#### 4.5 FLOP-Based Computational Complexity Assessment

In this study, floating-point operation (FLOP) counts are computed using the algorithmic complexity model, wherein each elementary arithmetic operation is assigned a unit computational cost. This abstraction enables performance evaluation and comparison of interpolation methods independently of hardware-specific execution latencies or implementation efficiencies. The FLOP cost assignments used in this work are consistent with widely accepted conventions in the numerical linear algebra literature, such as those presented by Golub and Van Loan [1996] and Dongarra et al. [2003]. Table 8 summarizes the FLOP cost per operation.

Operation	Symbol	FLOP Count	Description
Addition	$a + b$	1	Basic floating-point addition
Subtraction	$a - b$	1	Counted equivalent to addition
Multiplication	$a \times b$	1	Single floating-point multiplication
Division	$a \div b$	1	Treated as a standard binary operation
Square root	$\sqrt{a}$	1	Considered as a basic arithmetic operation
Exponentiation (integer)	$a^n$	$n-1$	Interpreted as $n-1$ multiplications
Dot product ( $length = n$ )	$\sum (a_i \times b_i)$	$2n$	$n$ multiplications and $n$ additions

Table 8: Flops cost per operation

In this study, all frequency estimation methods rely on an initial Fast Fourier Transform (FFT) to convert the time-domain signal into the frequency domain. Therefore, the FLOP (floating-point operation) cost of one FFT per run is included in the total FLOP count for every method, ensuring a fair comparison. Total FLOP counts were calculated based on 10,000 executions per method (100 Monte Carlo runs for each of 100 noise levels).

Among the methods:

- The lowest computational costs were observed in the Jain, Candan, and Ding methods, each requiring approximately 102.51 million FLOPs. These methods involve only a few simple arithmetic operations such as addition and division.

- The Improved Quinn method had the highest FLOP count, around 102.64 million FLOPs, due to its use of more complex functions like logarithms and square roots.
- The FFT-only (no interpolation) approach resulted in 102.42 million FLOPs, representing the minimal cost of spectral transformation and peak detection alone.

Table 9 summarizes the total estimated FLOP cost of each interpolation method over the full simulation setup (100 Monte Carlo runs  $\times$  100 noise levels = 10,000 executions), with the FFT cost uniformly included. Overall, the results show that all methods lie within a narrow range of total FLOP cost, and that the majority of the computational burden arises from the FFT itself. The additional cost of interpolation is relatively small in comparison.

Method	Key Operations	FLOPs per Run (incl. FFT)	Total FLOPs (100 $\times$ 100 runs)
FFT (no interpolation)	FFT + peak detection only	10,242	102,420,000
Jain	FFT + 3 reads, 2 divisions, 2 adds, 1 mult	10,251	102,510,000
Corrected Jain	Zero-padded FFT + interpolation logic	10,249	102,490,000
Quinn	FFT + 3 reads, 2 divs, 2 adds, 2 subs, 2 real ops, 1 mult	10,254	102,540,000
Improved Quinn	Quinn 1st + 2 logs, 2 powers, 4 adds, 2 sqrt	10,264	102,640,000
Jacobsen	FFT + 3 subs, 3 divs, 1 tanh, 2 mults	10,254	102,540,000
Macleod	FFT + 3 mults, 3 real, 3 conj, 3 adds, 2 divs, 2 sqrt	10,261	102,610,000
Ding	FFT + 2 subs, 3 adds, 1 div	10,248	102,480,000
Voglewede	FFT + 2 subs, 3 adds, 1 div	10,248	102,480,000
Candan	FFT + 3 subs, 3 adds, 1 tan, 2 mults	10,251	102,510,000
Rectangular window	FFT + 3 adds/subs + 1 div	10,246	102,460,000
Hanning window	FFT + 3 adds/subs + 1 div	10,246	102,460,000
Arva	FFT + 4 divs, 3 adds/subs, 1 logical comparison	10,250	102,500,000

Table 9: Total Estimated FLOPs per each interpolation

Although this study focuses on classical, non-learning-based interpolation methods due to their interpretability and low computational complexity, future work may explore comparisons with adaptive and learning-based estimators, such as reinforcement learning approaches used in dynamic decision-making under noise and resource constraints [Li et al. 2020, Mohajer et al. 2024].

## 5 Conclusion

One of the most critical parameters of the pulse descriptor word is the frequency of the threat radar, which is extracted by electronic support systems by listening to the combat environment. FFT is a widely used method for frequency estimation. Because it is critical to accurately estimate the frequency, many different interpolation methods have been developed to improve the accuracy of FFT output frequency. In this study, threat radar frequency is estimated by FFT and 12 different interpolation techniques (Jain, corrected Jain, Quinn, improved Quinn, Jacobsen, Macleod, Ding, Voglewede, Mobil Industrial (MI) method, Candan, rectangular-window-based interpolation, and Hanning window-based interpolation) are applied to the FFT result. The performance of the evaluated interpolation methods was assessed based not only on estimation accuracy in terms of RMSE, but also on computational efficiency through FLOPs-based complexity

analysis, providing a comprehensive comparison across both accuracy and resource cost dimensions.

According to simulation results for RMSE, although the interpolation method yielding the lowest RMSE may vary with frequency and SNR conditions, the approaches developed by Jain, Quinn, Improved Quinn, Jacobsen, and Macleod consistently demonstrate superior performance across different scenarios. The most unsuccessful method with the highest RMSE is the Hanning window-based interpolation method.

From the computational complexity perspective, interpolation methods differ significantly in terms of their arithmetic demands. Simpler techniques such as Jain, Candan, and Ding exhibit lower computational overhead, as they rely on basic arithmetic operations like addition, subtraction, and division. In contrast, more advanced methods—particularly the Improved Quinn approach—introduce higher processing complexity due to their use of transcendental functions, including logarithmic and square root operations. Although these methods impose a greater computational burden, they often offer improved estimation accuracy, highlighting the trade-off between performance and efficiency. Therefore, the selection of an appropriate method should consider both the estimation quality and the available processing resources. Among the twelve interpolation techniques presented in this study, the most suitable method should be selected based on the specific requirements and constraints of the intended application.

In this study, only interpolation techniques were examined; windowing techniques haven't been studied. In the future, windowing and interpolation methods can be used together to further improve FFT accuracy. In addition, similar improvements can be made for other interpolation methods; further, as with the improvements made to the Quinn algorithm, an improved Quinn algorithm is obtained, and the error rate is reduced. Moreover, since mostly improved Quinn has good performance in the relevant frequency and noise levels, this method can be compared with other frequency estimation methods and a new method to be developed in future studies.

## References

- [Aboutanios and Mulgrew 2005] Aboutanios, E., Mulgrew, B.: "Iterative frequency estimation by interpolation on Fourier coefficients"; *IEEE Trans. Signal Process.* 53, 4 (2005), 1237-1242. <https://doi.org/10.1109/TSP.2005.843719>.
- [Ai and Lomakin 2024] Ai, F., Lomakin, V.: "Fast Fourier Transform periodic interpolation method for superposition sums in a periodic unit cell"; *Computer Physics* 304, (2024), 109291. <https://doi.org/10.1016/j.cpc.2024.109291>.
- [Arva et al. 2017] Arva, M.C., Bizon, N., Micu A.: "Proposal of an accurate and low complexity method for frequency estimation using DFT interpolation"; *40<sup>th</sup> International Conference on Telecommunications and Signal Processing (TSP)*, (2017), 474-479. <https://doi.org/10.1109/TSP.2017.8076031>.
- [Blinowski and Szczypiorski 2017] Blinowski, G.J., Szczypiorski, K.: "Steganography in VLC Systems"; *Journal of Universal Computer Science* 23, 5 (2017), 454-478. <https://doi.org/10.3217/jucs-023-05-0454>.
- [Cabadağ and Karal 2022] Cabadağ, G., Karal, Ö.: "Analysis of Prony's and Pisarenko Frequency Estimation Methods at Different Bandwidths, Different Noise And Variances"; *30<sup>th</sup> Signal Processing and Communications Applications Conference (SIU)*, (2022), 1-4. <https://doi.org/10.1109/SIU55565.2022.9864904>.

- [Candan 2011] Candan Ç.: “A method for fine resolution frequency estimation from three DFT samples”; *IEEE Signal Process Lett.* 18, 6 (2011), 351-354. <https://doi.org/10.1109/LSP.2011.2136378>.
- [Candan 2013] Candan Ç.: “Analysis and further improvement of fine resolution frequency estimation method from three DFT samples”; *IEEE Signal Process Lett.* 20, 9 (2013), 913-916. <https://doi.org/10.1109/LSP.2013.2273616>.
- [Candan 2015] Candan Ç.: “Fine resolution frequency estimation from three DFT samples: Case of windowed data”; *Signal Processing* 114, (2015), 245-250. <https://doi.org/10.1016/j.sigpro.2015.03.009>.
- [Cavedo 2024] Cavedo, F., Esmaili, P., Norgia, M.: “Study of the Errors in Interpolated Fast Fourier Transform for Interferometric Applications”; *Metrology* 4, 1 (2024), 117-130. <https://doi.org/10.3390/metrology4010008>.
- [Dongarra et al. 2003] Dongarra, J., Luszczek, P., Petitet, A.: “The LINPACK Benchmark: Past, Present and Future”; *Concurrency Comput. Pract. Exper.* 15, 9 (2003), 803–820. <https://doi.org/10.1002/cpe.728>.
- [Duda 2011] Duda K.: “DFT interpolation algorithm for Kaiser–Bessel and Dolph–Chebyshev windows”; *IEEE Trans. Instrum. Meas.* 60, 3 (2011), 784-790. <https://doi.org/10.1109/TIM.2010.2046594>.
- [Fang et al. 2012] Fang, L., Duan, D., Yang, L.: “A new DFT-based frequency estimator for single-tone complex sinusoidal signal”; *IEEE MILCOM IEEE Military Communications Conf.*, (2012), 1-6. <https://doi.org/10.1109/MILCOM.2012.6415812>.
- [Gasior and Gonzalez 2000] Gasior, M., Gonzalez, J.L.: “Improving FFT frequency measurement resolution by parabolic and Gaussian spectrum interpolation”; *AIP Conference Proceedings, American Institute of Physics* 732, 1 (2000), 276-285. <https://doi.org/10.1063/1.1831158>.
- [Golub and Van Loan 1996] Golub, G.H., Van Loan, C.F.: “Matrix Computations (3rd ed.)”; JHU press (1996). <https://doi.org/10.56021/9781421407944>.
- [Huerta-Lopez et al. 2000] Huerta-Lopez C.I., Shin, Y., Powers, E.J., Roesset, J.M.: “Time-frequency analysis of earthquake records”; *12<sup>th</sup> World Conference on Earthquake Engineering Auckland* (2000). <https://doi.org/10.1006/mssp.2000.1321>.
- [Iglesias et al. 2014] Iglesias, V., Grajal, J., Yeste-Ojeda, O., Garrido, M., Sánchez, M.A., López-Vallejo, M.: “Real-time radar pulse parameter extractor”; *IEEE Radar Conf.*, (2014), 0371-0375. <https://doi.org/10.1109/RADAR.2014.6875617>.
- [Jiao et al. 2023] Jiao, Y., Zhang, F., Huang, Q., Liu, X., Li, L.: “Analysis of interpolation methods in the validation of backscattering coefficient products”; *Sensors* 23, 1 (2023), 469. <https://doi.org/10.3390/s23010469>.
- [Karal 2017] Karal Ö.: “Maximum likelihood optimal and robust Support Vector Regression with Incosh loss function”; *Neural Networks* 94, (2017), 3-21. <https://doi.org/10.1016/j.neunet.2017.06.008>.
- [Koziarski et al. 2020] Koziarski, L.M., Law Cyganek, B., Koc, O.N., Kara, A.: “A study on pattern recognition with the histograms of oriented gradients in distorted and noisy images”; *J. Univers. Comput.* 26, 4 (2020), 454-478. <https://doi.org/10.3897/jucs.2020.024>.
- [Kulkarni et al. 2013] Kulkarni, A.S., Vijesh, P., Paranjape, H.V., Reddy, K.M.: “Approaches towards implementation of multi-bit digital receiver using fast Fourier Transform”; *Def. Sci. J.* 63, 2 (2013), 198. <https://doi.org/10.14429/dsj.63.4264>.
- [Lai et al. 2019] Lai, Z., Xiao, Z., Zhang, G., Wang, G., Liu, Y., Shen, H., Gao, X.: “Application of FFT interpolation correction algorithm based on window function in power harmonic analysis”; *IOP Conf. Ser.: Earth Environ. Sci.* 253, 3 (2019), 032184. <https://doi.org/10.1088/1755-1315/252/3/032184>.

- [Li et al. 2020] Li, R., Wang, C., Zhao, Z., Guo, R., Zhang, H.: "The LSTM-based advantage actor-critic learning for resource management in network slicing with user mobility"; *IEEE Commun. Lett.* 24, 9 (2020), 2005-2009. <https://doi.org/10.1109/LCOMM.2024.3501956>.
- [Martinez et al. 2019] Lai, Z., Xiao, Z., Zhang, G., Wang, G., Liu, Y., Shen, H., Gao, X.: "Application of FFT interpolation correction algorithm based on window function in power harmonic analysis"; *IOP Conf. Ser.: Earth Environ. Sci.* 253, 3 (2019), 032184. <https://doi.org/10.1088/1755-1315/252/3/032184>.
- [Minda et al. 2020] Minda, A.A., Barbinita, C.I., Gillich, G.R.: "A Review of Interpolation Methods Used for Frequency Estimation"; *ROM. J. ACOUST. VIB.* 17, 1 (2020), 21-26.
- [Minda et al. 2020] Minda, A.A., Lupu, D., Gillich, G.R.: "Improvement of Jain's algorithm for frequency estimation"; *Stud. Univ. Babeş-Bolyai Eng.* 65, 1 (2020), 115-122. <https://doi.org/10.24193/subbeng.2020.1.12>.
- [Mohajer et al. 2024] Mohajer, A., Hajipour, J., Leung, V.C.: "Dynamic offloading in mobile edge computing with traffic-aware network slicing and adaptive TD3 strategy"; *IEEE Commun. Lett.* 29, 1 (2024), 95-99. <https://doi.org/10.1109/LCOMM.2024.3501956>.
- [Niranjan et al. 2021] Niranjan, R.K., Rao, C.B.R., Singh, A.K.: "Performance Comparison of FFT based Frequency Estimation using different Interpolation Techniques for ELINT Systems"; 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), (2021), 819-834. <https://doi.org/10.1109/ICAECT49130.2021.9392605>.
- [Ortatatlı et al. 2016] Ortatatlı, İ.E., Orduyılmaz, A., Serin, M., Özdil, Ö., Yıldırım, A., Gürbüz, A.C.: "Real-time frequency parameter extraction for electronic support systems"; 24<sup>th</sup> Signal Processing and Communication Application Conference (SIU), (2016), 105-108. <https://doi.org/10.1109/SIU.2016.7495687>.
- [Quinn 1994] Quinn, B.G.: "Estimating frequency by interpolation using Fourier coefficients"; *IEEE Trans. Signal Process.* 42, 5 (1994), 1264-1268. <https://doi.org/10.1109/78.295186>.
- [Pan et al. 2022] Pan, J., Zhang, S., Xia, L., Tan, L., Guo, L.: "Embedding Soft Thresholding Function into Deep Learning Models for Noisy Radar Emitter Signal Recognition"; *Electronics* 11, 14 (2022), 2142. <https://doi.org/10.3390/electronics11142142>.
- [Polap 2018] Polap, D.: "Model of identity verification support system based on voice and image samples"; *Journal of Universal Computer Science* 24, 4 (2018), 460-474. <https://doi.org/10.3217/jucs-024-04-0460>.
- [Rajaby and Sayedi2022] Rajaby, E., Sayedi, S.M.: "A structured review of sparse fast Fourier transform algorithms"; *Digital Signal Processing* 123, (2022), 103403. <https://doi.org/10.1016/j.dsp.2022.103403>.
- [Salih 2012] Salih, S.: "Fourier Transform: Signal Processing"; Rijeka, Croatia, IntechOpen, (2012). <https://doi.org/10.5772/813>.
- [Santamaria et al. 2000] Santamaria, I., Pantaleon, C., Ibanez, J.: "A comparative study of high-accuracy frequency estimation methods"; *Mech. Syst. Signal Process.* 14, 5 (2000), 819-834. <https://doi.org/10.1006/mssp.2000.1321>.
- [Tsui 1995] Tsui, J.B.: "Digital Techniques for wideband receivers"; 2<sup>nd</sup> Ed. SciTech Publishing, London, Boston, Artech House, (1995), 819-834. <https://doi.org/10.1049/SBRA511E>.
- [Zhang et al. 2001] Zhang, F., Geng, Z., Yuan, W.: "The algorithm of interpolating windowed FFT for harmonic analysis of electric power system"; *IEEE Trans. Power Delivery* 16, 2 (2001), 160-164. <https://doi.org/10.1109/61.915476>.
- [Zhang et al. 2021] Zhang, S., Pan, J., Han, Z., Guo, L.: "Recognition of noisy radar emitter signals using a one-dimensional deep residual shrinkage network"; *Sensors* 21, 23 (2021), 7973. <https://doi.org/10.3390/s21237973>.