Estimating frequency and phase in complex exponential embedded in complex Gaussian white noise using MLE

David Persson, Jon Erik Nordal

April 2025

Abstract

This project is a part of the course TTT4275 - Estimation, Detection and Classification. In the project we investigated the efficiency of estimating the frequency and phase of a complex exponential function embedded in complex Gaussian white noise by using a Maximum Likelihood Estimator (MLE) estimator based on the Fast Fourier Transform (FFT). Results show that the FFT estimator performs well at a high enough resolution, but is often too computationally expensive to be used in practice. In these cases, a computationally light FFT can be used together with a numerical search method to achieve similar estimation results with lower performance cost.

Contents

1 Introduction

1.1 Introduction to the problem

In this project we investigate the complex exponential function

$$x(t) = Ae^{i(\omega_0 t+\phi)} + w(t), \tag{1}$$

where we wish to estimate up to three unknown parameters, namely:

- The amplitude $A > 0$
- The frequency $\omega_0 > 0$
- The phase $-\pi < \phi < \pi$

We also assume w represents white, complex gaussian noise. This problem is of great practical interest as the need for estimating unknown parameters in sinusoidal functions often arises, for instance, in the field of communication technology.

Our solution of the problem is based on numerical approximations rather than deriving an explicit solution to the problem, as acquiring these solutions turns out to be practically impossible. More precisely, we will base the solution on the m-point (or m-bucket) FFT of the signal, which allows us to compute the MLE up to a precision determined by m. This is the first (naive) approach. However, acceptable precision requires such a large m that this is not feasible in practice. This motivates the second (refined) approach, which uses the FFT to obtain a rough estimate, and then refines it using a numerical search method.

2 Theory

This section is based on the theoretical results presented in [**?**]. This section will outline the theory that is required to understand the methods and results. We will start by defining the most central concepts of estimation/optimization theory, and then apply these concepts to the specific task of this project. Note that derivations will mostly be left out in order to not bloat the report.

2.1 Preliminaries of estimation theory

The estimation problem is given by the following:

- A number of samples, which are stochastic variables $x[0], x[1], ...x[N-1]$
- A parameter space Π
- A parameter vector $\theta \in \Pi$, which is considered unknown.

The goal is to construct an estimator, which is a function of the samples $\hat{\theta}(x[0], x[1], ..., x[N-1]) \in \Pi$, usually denoted simply $\hat{\theta}$. The goal is to make the output of this estimator statistically "close to" the true value θ.

Perhaps the most important criterion for an estimator is *unbiasedness*. An estimator is said to be unbiased if

$$\mathrm{E}[\hat{\theta}] = \theta. \tag{2}$$

Given an unbiased estimator, the next criterion of performance is the variance. Intuitively, if the variance is low, the chances of estimates being "close to" the true value are higher. Thus we wish to minimize the variance. It would be nice if we could achieve $\mathrm{var}(\hat{\theta}) = 0$, but an important theorem stops this from being possible. This is the famous Cramér-Rao Lower Bound (CRLB). However, to talk about the CRLB we first need the appropriate tools. Given an estimator $\hat{\theta}$ satisfying a certain[1] regularity condition, the Fisher information matrix is defined by

$$\left[\mathbf{I}(\hat{\theta})\right]_{ij} := -\mathrm{E}\left[\frac{\partial^2 \log(p(x;\theta))}{\partial\theta_i \partial\theta_j}\right], \tag{3}$$

where $p(x;\theta)$ is the multivariable pdf of the samples x given a parameter vector θ. This allows us to state the CRLB theorem, which says that, given some unbiased estimator $\hat{\theta}$,

$$\mathrm{var}(\hat{\theta}_i(x)) \geq \left[\mathbf{I}^{-1}(\hat{\theta})\right]_{ii}. \tag{4}$$

Here $\hat{\theta}_i(x)$ represents the i-th parameter to be estimated.

There are many different ways to create an estimator, but the one we will use in this project is the Maximum Likelihood Estimator (MLE), so we will now state the definition of this estimator. The fundamental idea of the MLE is to create an estimator which maximizes the probability of observing the observations which were made. The definition is simple,

$$\hat{\theta}_{MLE} := \arg\max_{\theta} p(x;\theta), \tag{5}$$

but there is an important subtlety here in that we consider p not as a function of the measurements x, but rather the parameter θ.

2.2 Preliminaries of optimization theory

Optimization is a mathematical tool based on finding minima, or maxima, of functions. It is a vast field filled with numerical techniques developed to solve these problems. In principle, the simplest optimization problem is stated as the following:

$$\max_{x} \quad f(x) \tag{6}$$

[1] `https://en.wikipedia.org/wiki/Fisher_information`

where f(x) is known as the objective function. A lot of numerical optimization techniques rely on the gradient of the objective function. In some applications, this may be a too strict requirement as the gradient may be too large or worse: it may not exist at all. In order to accommodate this, gradient-free optimization techniques have been developed. In our report we will be utilizing a few such methods, namely Nelder-Mead and Powell.

2.3 The estimator

As explained in section 1.1, we are attempting to estimate the frequency ω and phase ϕ in the signal

$$x(t) = Ae^{i(\omega_0 t + \phi)} + w(t), \tag{7}$$

where $w(t) \sim \mathcal{N}_{\mathcal{C}}(0, \sigma^2)\ \forall\ t \in \mathbb{R}$ is a complex gaussian distributed stochastic variable satisfying

$$\mathrm{E}[w(t)w(t+\tau)] = 0 \quad \forall\ \tau \in \mathbb{R}. \tag{8}$$

In other words, $w(t)$ represents complex Gaussian white noise.

Since we are going to use numerical methods to solve the problem, we will consider our signal as discrete in time, in other words a mapping $\mathbf{x} : \mathbb{Z} \to \mathbb{C}$. We thus define

$$x[n] = x(nT). \tag{9}$$

.

We will use the following constants in the calculations

- The sampling rate $F_s = 10^6$ Hz and period $T = 10^{-6}$ s.
- The true angular frequency $\omega_0 = 2\pi f_0\ \mathrm{rad\,s^{-1}}$, with $f_0 = 10^5$ Hz.
- The true phase $\phi = \frac{\pi}{8}$ rad.
- As mentioned in section 1, the amplitude $A = 1$ is considered known.
- The number of samples $N = 513$.

In order to simplify some of the calculations, we choose the samples to be symmetric about the origin, thus giving samples from $n = n_0$ through $n = n_0 + N - 1$. The starting point n_0 can be computed using $n_0 = -\frac{P}{N}$, where

$$P = \frac{N(N-1)}{2}. \tag{10}$$

Using our value $N = 513$, we get $n_0 = -256$.

Since we are considering a noisy signal, we will introduce the signal-to-noise-ratio (SNR), which is defined as

$$\mathrm{SNR} := \frac{A^2}{2\sigma^2}. \tag{11}$$

In the specific case of this task, many of these expressions can be further simplified. We will now present these simplifications. The SNR reduces to

$$\mathrm{SNR} = \frac{1}{2\sigma^2}, \tag{12}$$

the MLE is given by

$$\hat{\omega} = \arg\max_{\omega_0} \left| F(\omega_0) \right|, \tag{13}$$

where

$$F(w_0) = \frac{1}{N} \sum_{n=n_0}^{n_0+N-1} \mathbf{x}[n] e^{-i\omega_0 n t}. \tag{14}$$

Since the MLE is invariant to transformations we do not need to derive an MLE for the phase from scratch, as we can simply apply the transformation

$$\hat{\phi} = \arg e^{-i\hat{\omega} n_0 t} F(\hat{\omega}). \tag{15}$$

The CRLB of the estimator is given by the inequalities

$$\operatorname{var}(\hat{\omega}) \geq \frac{12\sigma^2}{A^2 T^2 N (N^2 - 1)} \tag{16a}$$

$$\operatorname{var}(\hat{\phi}) \geq \frac{12\sigma^2 (n_0^2 N + 2 n_0 P + Q)}{A^2 N^2 (N^2 - 1)}, \tag{16b}$$

Where P is defined as in equation 10, and

$$Q = \frac{N(N-1)(2N-1)}{6}. \tag{17}$$

Now, solving the optimization problem in equation 13 is not something we will attempt in this report. Instead we use Python to obtain an approximate solution. To do this we observe that equation 14 is the Discrete Fourier Transform (DFT) of the input signal x. Thus it can be approximated with an M-point Fast Fourier Transform (FFT), which divides the DFT into M frequency bins. The M-th frequency bin is given by

$$\omega_M = \frac{2\pi M}{MT}, \tag{18}$$

which means that we can approximate the MLE by taking

$$\hat{\omega}_{FFT} = \frac{2\pi m*}{MT}, \tag{19}$$

where

$$m* = \arg\max_{m} \mathrm{FFT}_M(\mathbf{x}). \tag{20}$$

Here FFT_M denotes the M-point FFT, and $\mathbf{x}$ is the measurement vector.

3 Implementation and results

The estimator was implemented using Python, relying mostly on the computing package `numpy`. The flow chart for converting samples into estimates is shown in figure 1. We pass the samples through numpy's `numpy.fft.fft`, then `argmax`, and finally use some relatively simple transformations of the data to extract the frequency and phase.

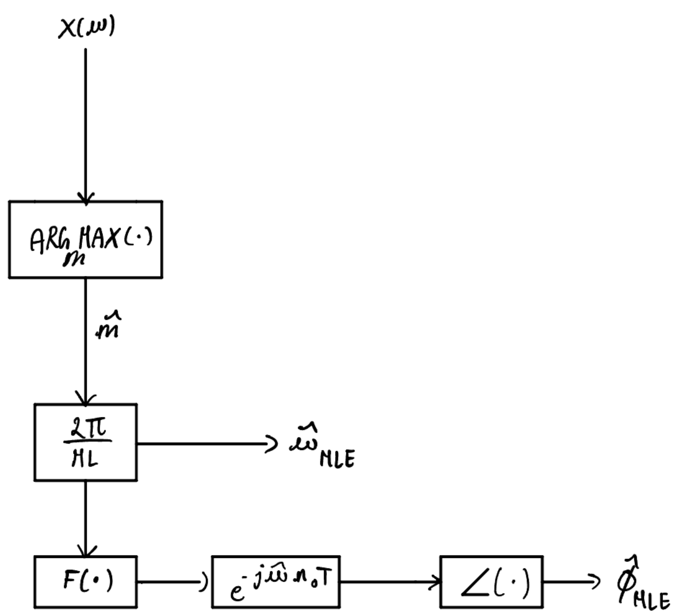

Figure 1: Flow chart describing the FFT-estimator for ω and ϕ

To find the variance of the estimator (which we cannot calculate by hand), we simply run the estimator a certain number of times, and compute both the variance (to compare with the CRLB) and the mean (to inspect whether the estimator is biased).

The M parameter in the M-point FFT is what we will be referring to as the "FFT size". This means that M represents the number of discrete frequencies the FFT can represent. This will be important to know as we now assess the performance of the FFT estimator.

3.1 Pure FFT estimator

In the first task we were instructed to run the FFT with all the FFT sizes in the set

$$\Phi := \{2^i \mid i = 10, 12, 14, 16, 18, 20\}, \tag{21}$$

and all the SNR levels (given in dB) in the set

$$\Sigma := \{-10, 0, 10, 20, 30, 40, 50, 60\}. \tag{22}$$

The result of computing the phase and frequency estimator variances for the different FFT sizes and SNR levels is shown in figures 2 and 3.

It is clear that the variance of the estimator goes below the CRLB as the SNR decreases. This is because the estimator is unbiased, which is especially clear when the SNR level is large compared to the FFT size, which is also visible in figures 4 and 5. In the case where the FFT size is large enough compared to the SNR level one can observe that the estimator attains the CRLB. This suggests that when $M \to \infty$, the FFT (or in that case, simply the fourier transform) could be an MVU estimator. Also note that some inaccuracies wrt. the CRLB are present due to the variance being computed empirically.

3.2 FFT estimator refined using numerical optimization

For FFT size 2^{20} and higher, the estimator takes impractically long to run, but this can be resolved using techniques from our theory of optimization.

More specifically, we use an FFT size of 2^{10} to obtain a rough result, and then run a numerical optimizer with the frequency bin from the FFT `argmax` computation as the starting point. Our initial goal is to maximize $|F(\omega_0)|$, so we then obtain the optimization problem 6 with $-|F(\omega_0)|$ as our objective function.

The choice of numerical optimizer is also worth discussing. In general, since the noise-free "true" signal consists only of a single frequency $\omega_0 = 2\pi 10^5$, we expect the fourier transform to have a large spike at this frequency, and then be flat all other places. Since the function to be optimized is the discrete fourier transform, we can conclude that a gradient-based method likely will not work well, since the gradient attains such large values.

The task description recommends using the Nelder-Mead method, which is a method of gradient-free optimization. We used the Nelder-Mead method as well as Powell's method (another gradient-free optimization method) and compared the results.

From figures 6, and 7, we can see that (both with Powell's method and Nelder-Mead) the optimization-based estimator attains the CRLB, even in the cases where the SNR is very high. We can also see that it suffers no breakdown

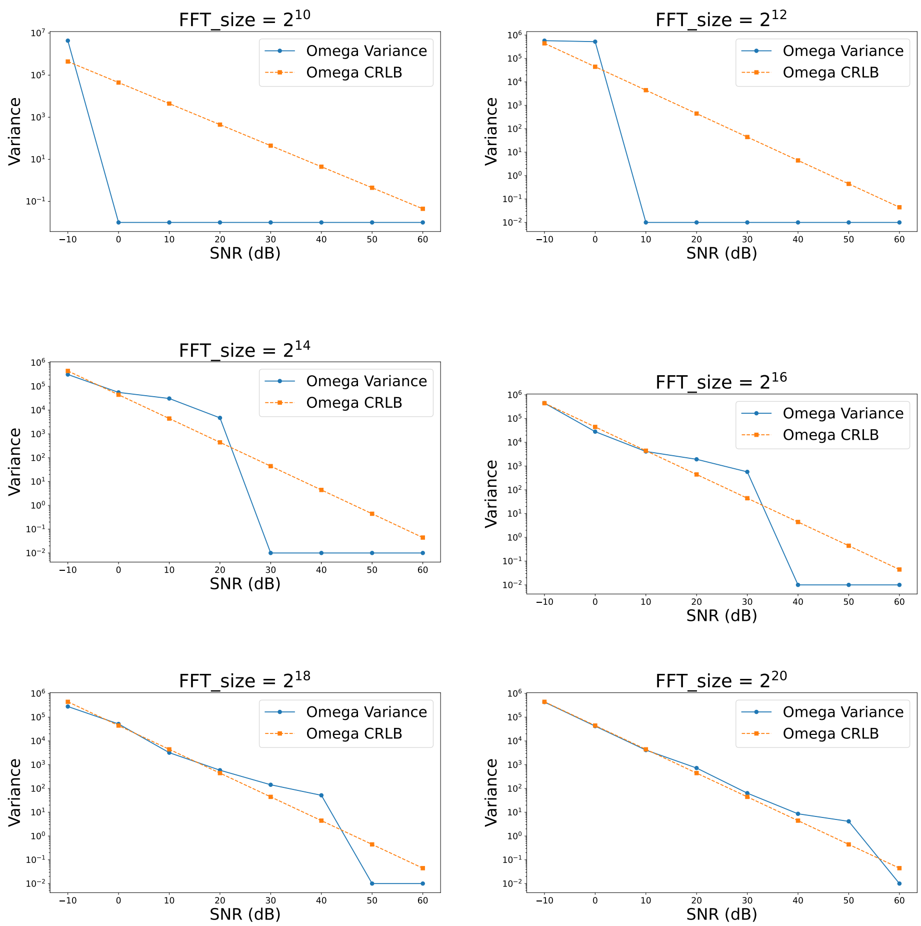

Figure 2: Omega estimates compared to the Cramer-Rao lower bound.

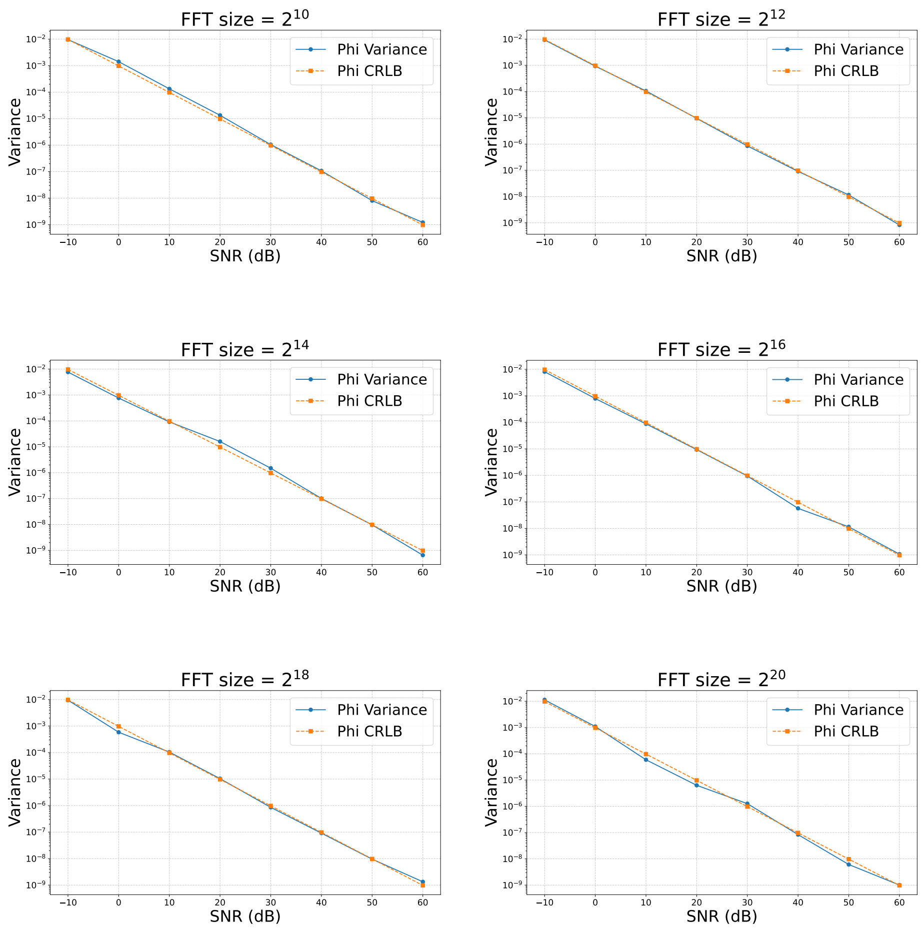

Figure 3: Phi estimates compared to the Cramer-Rao lower bound.

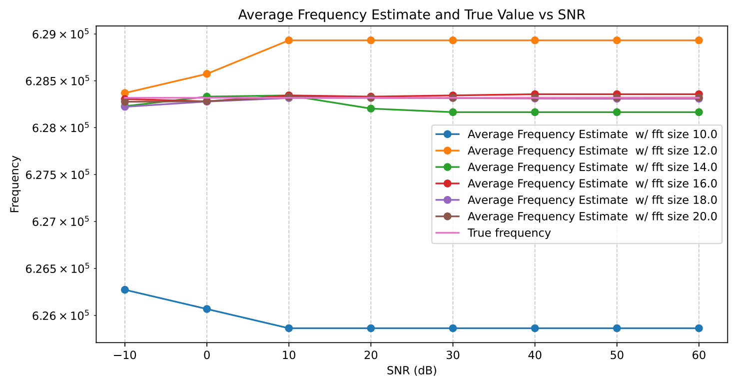

Figure 4: Average frequency estimate using pure FFT compared to true value

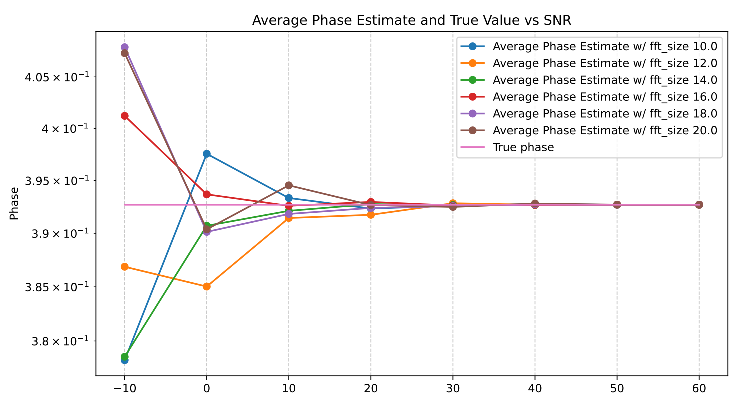

Figure 5: Average phase estimate using pure FFT compared to true value

like the pure FFT estimator does when the FFT size is low compared to SNR (as seen in figure 2, the variance drops to zero).

From figure 8 and 9 we can also see that the estimator approaches the true value of the parameters being estimated.

Finally, we can clearly see that both Nelder-Mead and and Powell exhibit good frequency, however the Powell method provided significantly faster runtimes of around 20 seconds compared to Nelder-Mead, which took about 50 seconds to run. The pure FFT estimator only took a little over 10 seconds to run even at the largest FFT size, but increasing the FFT size once more to 2^{22} (in order to get accuracy comparable to the numerical optimization method) increases the runtime to about 45 seconds. Thus the estimator using Powell's method has the best performance both in terms of accuracy and time, and is preferrable.

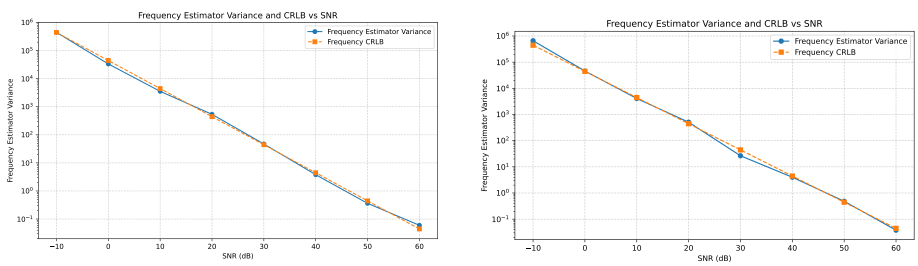

Figure 6: Comparison of frequency estimator performance between Nelder-Mead (left) and Powell (right).

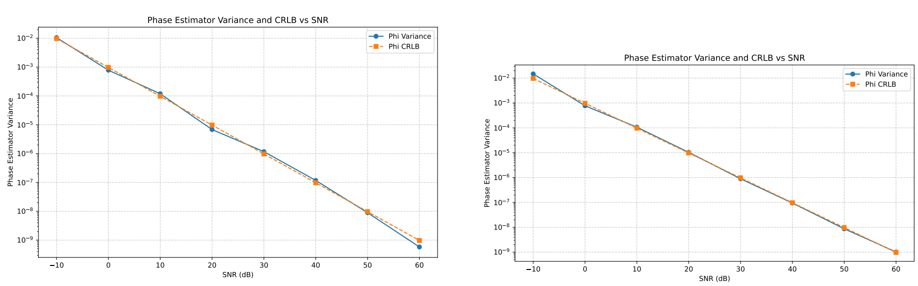

Figure 7: Comparison of phase estimator performance between Nelder-Mead (left) and Powell (right).

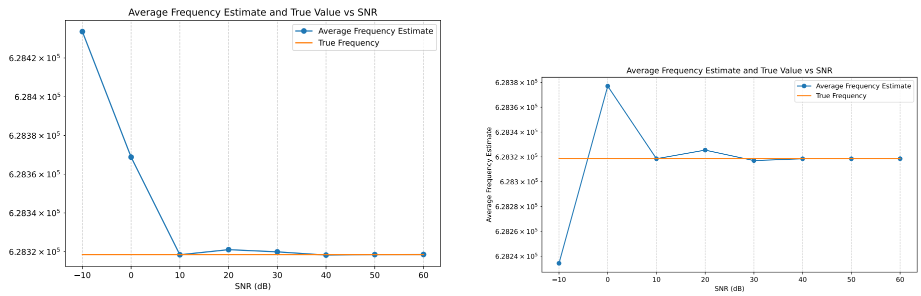

Figure 8: Comparison of average frequency estimate between Nelder-Mead (left) and Powell (right).

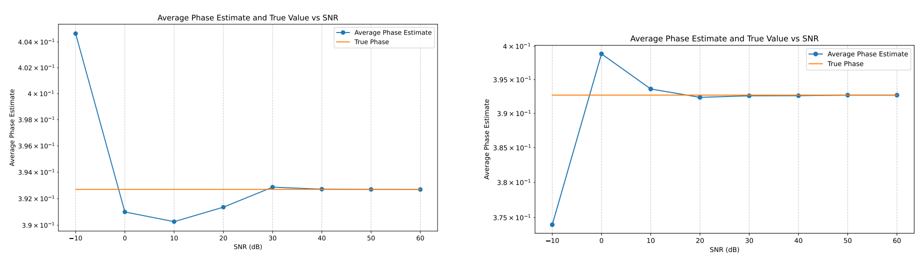

Figure 9: Comparison of average phase estimate between Nelder-Mead (left) and Powell (right).

4 Conclusion

The FFT-based estimator successfully estimates the frequency and phase, but is too computationally expensive. Therefore, the alternative method which uses the FFT to obtain a rough estimate and then refines it using numerical optimization is preferable as it achieves better performance with a lower computational cost.