NTNU – Trondheim
Norwegian University of
Science and Technology

Synchronization in Digital Receivers

Faraz Barzideh

Master of Science in Communication Technology
Submission date: July 2013
Supervisor: Lars Magne Lundheim, IET
Co-supervisor: Sverre Wichlund, Nordic Semiconductor

Norwegian University of Science and Technology
Department of Electronics and Telecommunications

NTNU – Trondheim
Norwegian University of
Science and Technology

Norwegian University of Science and Technology
(NTNU)

Master Thesis

Synchronization in Digital Receivers

Author:
Faraz Barzideh

Supervisors:
Professor Lars Lundheim
NTNU
Dr. Sverre Wichlund
Nordic Semiconductor

A thesis submitted in fulfillment of the requirements
for the degree of Masrer of Science

in

Signal Processing Group
Department of Electronics and Telecommunication

July 2013

Norwegian University of Science and Technology

Abstract

Information Technology, Mathematics and Electrical Engineering
Department of Electronics and Telecommunication

Masrer of Science

Synchronization in Digital Receivers

by Faraz Barzideh

Estimating the frequency of received signals is an attractive topic in communication theory. In this work, some of these estimators are introduced. Also, a generalization of two of these estimators are derived. Performance of the estimators are compared together for several preamble lengths, signal-to-noise values and steps. Generalized estimators attain the Cramer-Rao lower bound for lower signal-to-noise ratio values if the right step value is used. The computation complexity of the estimators is calculated and discussed. Also it is shown that computation complexity depends on how an algorithm is implemented by calculating computation complexity for two separate implementations of a single estimator.

Acknowledgements

First and Foremost, I would like to express my sincere gratitude to my supervisor Professor Lars Lundheim for the continuous support, for his patience, motivation, enthusiasm, immense knowledge and for carefully reading and commenting on countless revisions of this manuscript. I could not have imagined having a better advisor and mentor for my master thesis. I also would like to express my gratitude to my co-supervisor Dr. Sverre Wichlund for his insightful comments and suggestions.

My sincere thanks also goes to Solomon Tesfamicael for his friendly support and advices. Special thanks to Susanne Sandell for proofreading the manuscript.

Last but not the least, I would like to thank my family for their love, concern and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The estimation of the frequency of a single complex sinusoid in white Gaussian noise is a problem in signal processing which has received much attention, mainly due to its use in digital receivers.

Every digital transmission system contains several oscillators. In order to be able to extract the data out of the received signals, the oscillators in the receiver and the transmitter should be matched in frequency . In reality however, for many reasons this rarely happens for example because of unmatched oscillators in transmitter and receiver or because of Doppler Effect. As a result, there is a need for methods which can estimate the carrier frequency.

This report focuses on such methods. The aim for this report is to find algorithms that use low energy and at the same time have enough accuracy. Chapter 2 describes noise models and their subsequent estimators. Some popular estimators along with their generalized versions will be introduced. Also, this report introduces generalized version of two estimators proposed by Hua Fu and Pooi Yuen Kam. Chapter 3 discusses performance of the estimators in different conditions while Chapter 4 explains about the computation complexity of the estimators. Chapter 5 concludes this work by comparing the results from the previous chapters and rating the estimators for use in Bluetooth low energy.

Chapter 2

Estimators

2.1 Assumptions

In digital communication, some protocols contain a sequence of known symbols called preamble which is used to synchronize the transmission. This way the receiver can estimate the carrier frequency easier.

Consider that the received signal has the following form

$$x(t) = Ae^{j(\omega_0 t+\theta)} + z(t) \qquad t = 0, 1, \ldots, N-1 \tag{2.1}$$

where ω_0 and θ are deterministic with unknown values and N is the length of the preamble. The noise, $z(t)$, is a zero mean complex white Gaussian process. It is possible to write $z(t)$ as $z_I(t) + jz_Q(t)$ in which $z_I(t)$ is the parallel component and $z_Q(t)$ is the perpendicular component to the transmitted signal phasor [1]. The real and imaginary parts both have variance of $\frac{\sigma^2}{2}$ and are uncorrelated with each other.

Large enough signal-to-noise(SNR), $\frac{A^2}{\sigma^2}$, makes it possible to approximate the signal in the following way[2]

$$x(t) \approx Ae^{j(\omega_0 t+\theta+u(t))} \qquad t = 0, 1, \ldots, N-1. \tag{2.2}$$

The phase of $x(t)$ is

$$\angle x(t) = \omega_0 + \theta + u(t) \qquad t = 0, 1, \ldots, N-1. \tag{2.3}$$

Here, $u(t)$ is a zero mean white Gaussian process and can be expressed as [3]

$$u(t) = \tan^{-1}[\frac{z_Q(t)}{A + z_I(t)}]. \tag{2.4}$$

High enough SNR ($\frac{A^2}{\sigma^2} \gg 1$), $|z_I(t)| \ll A$, makes it possible to approximate the measurement noise,$u(t)$, as

$$u(t) \approx \frac{z_Q(t)}{A} \tag{2.5}$$

in which the noise is a zero mean Gaussian with variance of $\frac{\sigma^2/2}{A^2}$. This model was used by Tretter in [4].

The noise can also be expressed as [3]

$$u(t) = \sin^{-1}[\frac{z_Q(t)}{|x(t)|}]. \tag{2.6}$$

Assuming high SNR and using the approximation that $\sin^{-1}(x) \approx x$, we can write equation 2.6 as

$$u(t) \approx \frac{z_Q(t)}{|x(t)|}. \tag{2.7}$$

In this case, the noise is a zero mean Gaussian with variance of $\frac{\sigma^2/2}{|x(t)|^2}$.

The exact statistical model for the noise is given by [5]

$$p(u(t)||x(t)|) = \frac{\exp\left[\frac{|x(t)|A}{\sigma^2/2}\cos\left(n(t)\right)\right]}{2\pi I_0(\frac{|x(t)|A}{\sigma^2/2})} \qquad -\pi \leq u(t) < \pi \tag{2.8}$$

where $I_0(\cdot)$ is the modified Bessel function of the first kind and of order zero.

If SNR is high, equation 2.8 can be approximated by a Gaussian probability distribution function (PDF) leading to

$$p(u(t)||x(t)|) = \frac{1}{\sqrt{2\pi\frac{\sigma^2/2}{|x(t)|A}}}\exp\left(-\frac{u^2(t)}{2\frac{\sigma^2/2}{|x(t)|A}}\right) \qquad -\pi \leq u(t) < \pi \tag{2.9}$$

which shows that the noise variance in this model is $\frac{\sigma^2/2}{|x(t)|A}$.

Equation 2.10 is the *a priori* for the noise when SNR is high and is the same model, equation 2.5, used by Tretter[5]

$$p(u(t)) = \frac{1}{\sqrt{2\pi\frac{\sigma^2/2}{A^2}}}\exp[-\frac{u^2(t)}{2\frac{\sigma^2/2}{A^2}}] \qquad -\infty \leq u(t) < \infty. \tag{2.10}$$

2.2 Derivation of Estimators

It is possible to accurately estimate the frequency by finding the maximum likelihood estimator for the signal model in equation 2.1. The problem is that sometimes the amount of computation is prohibitive even with an FFT implementation[2], so it is preferable to use simpler estimators. This report discusses the methods proposed in [2, 5–7]. Also a generalized versions of the estimators proposed by Hua Fu and Pooi Yuen Kam in [5] is derived.

Since the aim is to only estimate the frequency, ω_0 , the following approach will be taken by defining

$$\Delta(t) = \angle x(t+1) - \angle x(t) \qquad t = 0, 1, \ldots, N-1. \tag{2.11}$$

It is also possible to write equation 2.11 as

$$\Delta(t) = \angle[x(t+1)x(t)^*] \qquad t = 0, 1, \ldots, N-1. \tag{2.12}$$

Inserting equation 2.3 into equation 2.12 leads to

$$\Delta(t) = \omega_0 + v(t) \qquad t = 0, 1, \ldots, N-1 \tag{2.13}$$

where $v(t) = u(t+1) - u(t)$ is a zero mean colored Gaussian noise. This approach results in a DC in colored Gaussian noise problem. The maximum likelihood estimator for such a problem has the following form [2]:

$$\hat{\omega}_0 = \frac{\mathbf{1}^T\mathbf{C_v^{-1}}\mathbf{\Delta}}{\mathbf{1}^T\mathbf{C_v^{-1}}\mathbf{1}} \tag{2.14}$$

where $\mathbf{C_v}$ is the covariance matrix of $\mathbf{v}$ which is defined as $\mathbf{v} = [v(1)\ \ v(2)\ldots\ \ v(N-1)]$, $\mathbf{1} = [1\ \ 1\ldots 1\ \ 1]^T$ and $\mathbf{\Delta} = [\Delta(1)\ \ \Delta(2)\ \ \ldots\Delta(N)]^T$. The entries of C, denoted by $c_{i,j}$, depend on the chosen noise model.

Using Tretter's model, equation 2.5, the entries of C will be [2]

$$c_{i,j} = \begin{cases} \frac{\sigma^2}{A^2}, & \text{if } i = j \\ -\frac{\sigma^2}{2A^2}, & \text{if } |i-j| = 1 \\ 0, & \text{if } |i-j| > 1. \end{cases} \tag{2.15}$$

If the model from equation 2.7 is used, the entries of C will be [5]

$$c_{i,j} = \begin{cases} \frac{\sigma^2/2}{|x(t+1)|^2} + \frac{\sigma^2/2}{|x(t)|^2}, & \text{if } i = j \\ -\frac{\sigma^2/2}{|x(i)|^2}, & \text{if } i-j = 1 \\ -\frac{\sigma^2/2}{|x(j)|^2}, & \text{if } j-i = 1 \\ 0, & \text{if } |i-j| > 1. \end{cases} \tag{2.16}$$

Finally, if the noise model from equation 2.10, is used the entries of C will be [5]

$$c_{i,j} = \begin{cases} \frac{\sigma^2/2}{|x(t+1)|A} + \frac{\sigma^2/2}{|x(t)|A}, & \text{if } i = j \\ -\frac{\sigma^2/2}{|x(i)|A}, & \text{if } i-j = 1 \\ -\frac{\sigma^2/2}{|x(j)|A}, & \text{if } j-i = 1 \\ 0, & \text{if } |i-j| > 1. \end{cases} \tag{2.17}$$

If the covariance matrix from equation 2.15 is used, equation 2.14 can be rewritten as a closed form used by Kay in [2]. The resulting estimator is

$$\hat{\omega}_0 = \sum_{t=0}^{N-2} w(t) \angle [x(t)^* x(t+1)] \tag{2.18}$$

where $w(t) = \frac{\frac{3}{2}N}{N^2-1}\{1 - [\frac{t-(\frac{N}{2}-1)}{\frac{N}{2}}]^2\}$. This estimator is known as weighted phase averager [2]. $w(t)$ is called the windowing function. It should be noted that $w(t)$ is independent of signal. As seen in figure 2.1 for three different values of N, $w(t)$ is a symmetric function. This property will be helpful in Chapter 4 where the computation complexity of estimators is calculated.

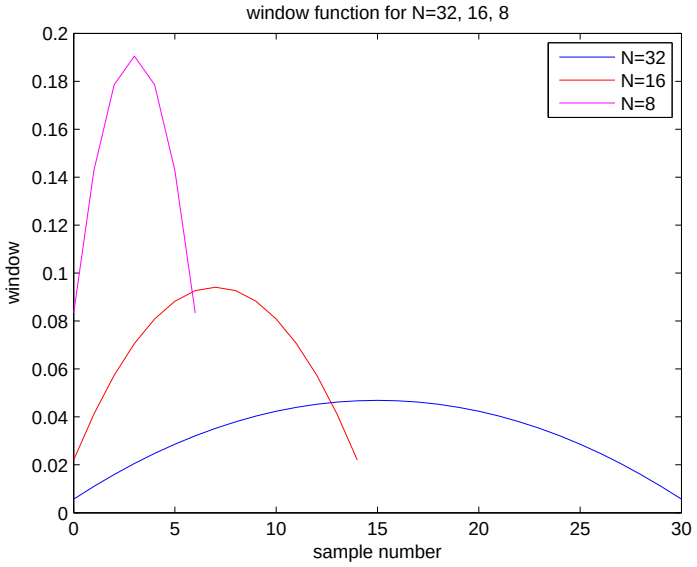

FIGURE 2.1: Windowing function for weighted phase averager for $N = 32$

If instead of using the windowing function in equation 2.18, a flat windowing function is applied, the following estimator will result:

$$\hat{\omega}_0 = \frac{1}{N-1} \sum_{t=0}^{N-2} \angle[x(t)^* x(t+1)] \tag{2.19}$$

which is called as unweighted phase averager [2]. A flat windowing function means neglecting the fact that the noise is colored. This leads to loss of performance. Using a flat windowing function, the estimator in equation 2.19 becomes

$$\frac{1}{N-1} \sum_{t=0}^{N-2} [\angle x(t+1) - \angle x(t)]$$

$$= \frac{1}{N-1} [\angle x(N+1) - \angle x(1)]$$

which is unbiased and has variance of $var(\hat{\omega}_0) = \frac{1}{(N-1)^2 \frac{A^2}{\sigma^2}}$ [2].

There is lower bound for the variances achievable for unbiased estimators which is called Cramer-Rao lower bound (CRLB)[8]. So comparing the variance of the estimators against CRLB tells us how far we are from the least possible variance for an unbiased estimator. For this problem and assuming that SNR is high enough that the signal can be approximated to have the form from equation 2.2, CRLB= $\frac{6}{\frac{A^2}{\sigma^2} N(N^2-1)}$ [2]. The ratio of variance to CRLB is

$$\frac{var(\omega_0)}{CRLB} = \frac{N(N+1)}{6(N-1)} \approx \frac{N}{6} \tag{2.20}$$

which shows that for large N, the performance loss is substantial. Note that the ratio only depends on N. This leads to a fixed distance relative to CRLB when N is fixed.

For high SNR values, the following estimator is equivalent to the estimator in equation 2.19[2] meaning that they produce the same results. The estimator is called a linear prediction estimator [2] and has the following form

$$\hat{\omega}_0 = \angle \frac{1}{N-1} \sum_{t=0}^{N-2} x(t)^* x(t+1). \tag{2.21}$$

2.3 Generalized Estimators

Generalized versions of some of the estimators introduced in the previous section have been used in [6, 7, 9, 10] where $D\omega_0$ is estimated first and then through dividing by D, $\hat{\omega}_0$ is resulted. In this report, D will be called "step" meaning the steps between samples. Using D, equations from the previous section are rewritten as follows

$$\Delta(t) = \angle x(t+D) - \angle x(t) = \angle[x(t+D)x^*(t)] \qquad t = 0, 1, \ldots, N-1 \tag{2.22}$$

$$= D\omega_0 + v(t) \qquad t = 0, 1, \ldots, N-1 \tag{2.23}$$

where $v(t) = u(t+D) - u(t)$.

Using the procedure mentioned above, MLE estimator(equation 2.14), becomes

$$\hat{\omega}_0 = \frac{1}{D} \frac{\mathbf{1}^T \mathbf{C_v^{-1}} \boldsymbol{\Delta}}{\mathbf{1}^T \mathbf{C_v^{-1}} \mathbf{1}}. \tag{2.24}$$

By changing the step from 1 to an arbitrary value D, the covariance matrices change too.

For Tretter's model(equation 2.5), the covariance matrix will be

$$c_{i,j} = \begin{cases} \dfrac{\sigma^2}{A^2}, & \text{if } i = j \\ -\dfrac{\sigma^2}{2A^2}, & \text{if } |i-j| = D \\ 0, & \text{Otherwise} \end{cases} \tag{2.25}$$

which is called "Generalized weighted phase averager (GWPA) ". By looking more carefully at equation 2.25, it can be seen that the covariance matrix is symmetric. It is possible to write the covariance matrix as $\mathbf{C_v} = \frac{\sigma^2/2}{A^2} \cdot \mathbf{C}$ so $\mathbf{C_v^{-1}} = \frac{A^2}{\sigma^2/2}\mathbf{C^{-1}}$. Applying this to equation 2.24, the dependency on SNR vanishes from both numerator and denominator. This gives the possibility to find a closed form for GWPA similar to equation 2.18.

C is symmetric meaning that its inverse is also symmetric. So $\mathbf{1}^T\mathbf{C^{-1}}$ is symmetric too. Also $\mathbf{1}^T\mathbf{C^{-1}}\mathbf{1}$ is just the summation of the elements of $\mathbf{1}^T\mathbf{C^{-1}}$ and is constant. This pattern makes it possible to write GWPA in a closed form. The only issue that remains is to find the symmetric vector $\frac{1}{D}\frac{\mathbf{1}^T\mathbf{C^{-1}}}{\mathbf{1}^T\mathbf{C^{-1}}\mathbf{1}}$.

In [7], Rosnes and Vahlin divided $1 \leq D \leq (N-1)$ into three sub intervals $D \leq N/3$, $N/3 \leq D \leq N/2$ and $D \geq N/2$.

They derived a closed form for the case when $N/3 \leq D \leq N/2$. Under this condition the estimator is

$$\hat{\omega}_0 = \frac{1}{D}\frac{1}{N-1}\sum_{t=1}^{N-D} W_g(t)\angle[x(t)^* x(t+D)] \tag{2.26}$$

where $W_g(t)$ is the windowing function is

$$W_g(t) = \begin{cases} \dfrac{2}{(3N-5D)}, & \text{if } t = D, D+1, \ldots, N-D-1 \\ \dfrac{1}{(3N-5D)}, & \text{if } t = N-D, N-D+1, \ldots, 2D-1 \\ \dfrac{2}{(3N-5D)}, & \text{if } t = 2D, 2D+1, \ldots, N-1 \end{cases} \tag{2.27}$$

As expected, the windowing function is symmetric.

It is mentioned in [7] that if $D > N/2$ the covariance matrix becomes diagonal. Using equation 2.25, the windowing function becomes $\frac{1}{D(N-D)}$ and GWPA becomes

$$\hat{\omega}_0 = \frac{1}{D}\frac{1}{N-D}\sum_{t=1}^{N-D} \angle[x(t)^* x(t+D)]. \tag{2.28}$$

Finding a windowing function for $D \leq N/3$ is not straight forward. Yet based on previous discussions, it is already known that it will be symmetric.

Having this knowledge can help a designer to store half of the values of $\frac{\mathbf{1}^T\mathbf{C}^{-1}}{\mathbf{1}^T\mathbf{C}^{-1}\mathbf{1}}$ and use them instead of calculating the inverse of covariance matrix.

For the noise model in equation 2.7, we derive a generalized covariance matrix by introducing an arbitrary step D. The noise, v, is zero mean so

$$C = \begin{bmatrix} E(v(0)v(0)) & \dots & E(v(N-D)v(0)) \\ E(v(0)v(1)) & \dots & E(v(N-D)v(1)) \\ \dots & \dots & \dots \\ E(v(0)v(N-D)) & \dots & E(v(N-D)v(N-D)) \end{bmatrix} \tag{2.29}$$

Now we calculate the entries. The entries of $\mathbf{C_v^{-1}}$, $c_{i,j}$, can be divided into four different categories depending on their indexes.

$$c_{i,j} \in \begin{cases} \text{Group one} & \text{if} \quad i = j \\ \text{Group two} & \text{if} \quad i - j = D \\ \text{Group three} & \text{if} \quad j - i = D \\ \text{Group four} & \text{The rest} \end{cases} \tag{2.30}$$

For group one the entries are

$$c_{i,j} = E((v(n)v(n)) = E((u(n+D) - u(n)(u(n+D) - u(n)))$$

$$= E(u^2(n+D) - 2u(n+D)u(n) + u^2(n))$$

Since the noise samples are independent of each other, $E(u(n+D)u(n)) = 0$, so

$$c_{i,j} = E(u^2(n+D)) + E(u^2(n)) = \frac{\sigma^2/2}{|x(t+1)|^2} + \frac{\sigma^2/2}{|x(t)|^2}. \tag{2.31}$$

For the second group the entries are

$$c_{i,j} = E(v(i)v(j))$$

Since $i - j = D$ so $i = j + D$, then

$$c_{i,j} = E((u(i+D)-u(i))(u(j+D)-u(j))) = E((u(j+2D)-u(j+D)(u(j+D)-u(j)))$$

$$= E(u(j+2D)u(j+D) - u(j+2D)u(j) - u^2(j+D) + u(j+D)u(j))$$

and again because the noise samples are independent of each other, everything except $E(-u^2(j+D))$ is zero so

$$c_{i,j} = E(-u^2(j+D)) = -\frac{\sigma^2/2}{|x(i)|^2}. \tag{2.32}$$

The situation for group three is similar to group two.

For group four the entries are

$$c_{i,j} = E(v(n)v(m)) = E((u(n+D) - u(n))(u(m+D) - u(m)))$$

$$= E(u(n+D)u(m+D) - u(n+D)u(m) - u(n)u(m+D) + u(m)u(n)) = 0. \tag{2.33}$$

So the covariance matrix is

$$c_{i,j} = \begin{cases} \dfrac{\sigma^2/2}{|x(t+1)|^2} + \dfrac{\sigma^2/2}{|x(t)|^2}, & \text{if } i = j \\ -\dfrac{\sigma^2/2}{|x(i)|^2}, & \text{if } i - j = D \\ -\dfrac{\sigma^2/2}{|x(j)|^2}, & \text{if } j - i = D \\ 0, & \text{Otherwise} \end{cases} \tag{2.34}$$

which is a Generalized version of the estimator proposed by H. Fu and P.Y. Kam in [3] and for the sake of simplicity it will be called GHP.

Equation 2.17 is generalized to have the following covariance matrix. The entries of this matrix can be calculated similarly to the way the entries of GHP were calculated.

$$c_{i,j} = \begin{cases} \frac{\sigma^2/2}{|x(t+1)|A} + \frac{\sigma^2/2}{|x(t)|A}, & \text{if } i = j \\ -\frac{\sigma^2/2}{|x(i)|A}, & \text{if } i - j = D \\ -\frac{\sigma^2/2}{|x(j)|A}, & \text{if } j - i = D \\ 0, & \text{Otherwise} \end{cases} \tag{2.35}$$

and is called GLHP.

Because of dependence of covariance matrices of GHP and GLHP on the signal, it is not possible to pre-calculate and store the windowing function weights beforehand. Also, a closed form is not easily obtained. The only thing that can be said is the form of covariance matrix. Similar to GWPA, when $D \geq N/2$ the covariance matrix is diagonal. If D is chosen so that $D \geq N/2$, then no matrix inversion is needed because inverse of a diagonal matrix is a diagonal matrix with entries that are inverse of the entries of the original matrix. Yet this process should be done every time we receive signals.

Generalized version of estimators in equations 2.19 and 2.21 are

$$\hat{\omega}_0 = \frac{1}{D}\frac{1}{N-1}\sum_{t=1}^{N-D} \angle[x(t)^* x(t+D)] \tag{2.36}$$

$$\hat{\omega}_0 = \frac{1}{D}\angle[\frac{1}{N-1}\sum_{t=1}^{N-D} x(t)^* x(t+D)] \tag{2.37}$$

and these estimators are called generalized unweighted phase averager (GUWPA) and generalized unweighted linear predictor(GULP) respectively. Note that GUWPA is the same as GWPA when $D \geq N/2$.

As can be seen, equations 2.18 to 2.21 are special cases of the generalized forms when $D = 1$.

It should be noted that there is an upper limit for D. It should have a value so that $D\omega_0 < \pi$, otherwise aliasing problems occur. Figure 2.2 shows the relation between D and ω_0.

Also, it is not true that the higher the value of D is, the better the estimators become. In the following, this is proven for estimator in equation 2.36 :

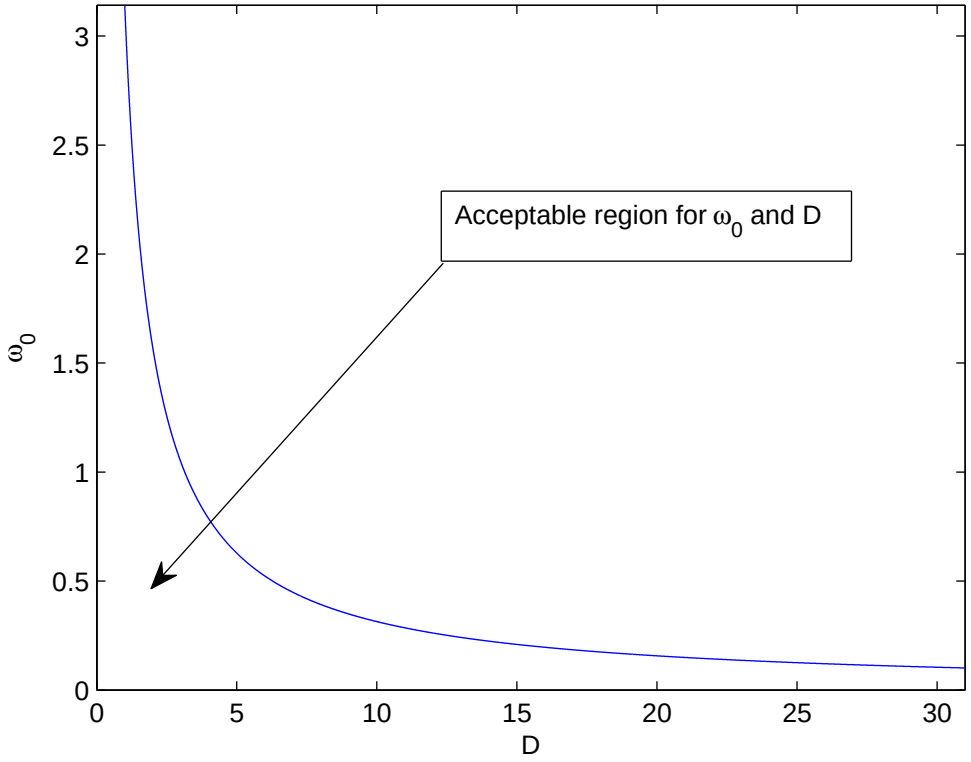

FIGURE 2.2: Relation between ω_0 and D

The unweighted phase averager is

$$\hat{\omega}_0 = \frac{1}{D}\frac{1}{N-D}\sum_{t=1}^{N-D} \angle[x(t)^* x(t+D)] = \omega_0 + \frac{1}{D(N-D)}\sum_{t=1}^{N-D} u(t+D) - u(t) = \omega_0 + U \tag{2.38}$$

For $D \leq N/2$

$$U = \sum_{t=1}^{D} -u(t) + \sum_{t=D+1}^{N-D} u(t) - u(t) + \sum_{N-D+1}^{N} u(t) = \sum_{t=1}^{D} -u(t) + \sum_{t=N-D+1}^{N} u(t) \tag{2.39}$$

For $D > N/2$

$$U = -\sum_{t=1}^{N-D} u(t) + \sum_{t=D+1}^{N} u(t) \tag{2.40}$$

Since $u(t) \sim \mathcal{N}(0, \sigma_u^2)$ and is white, the variance of U becomes

$$\sigma_U^2 = \begin{cases} 2D\sigma_u^2, & \text{if } D \leq N/2 \\ 2(N-D)\sigma_u^2, & \text{if } D > N/2. \end{cases} \tag{2.41}$$

Variance of $\hat{\omega}_0$ is $\sigma_{\hat{\omega}_0}^2 = \frac{1}{D^2(N-D)^2}\sigma_U^2$ and therefore

$$\sigma_{\hat{\omega}_0}^2 = \begin{cases} \dfrac{2}{D(N-D)^2}\sigma_u^2, & \text{if } D \leq N/2 \\ \dfrac{2}{D^2(N-D)}\sigma_u^2, & \text{if } D > N/2. \end{cases} \tag{2.42}$$

Figure 2.3 shows the variance of GUWPA for different values of D when $N = 32$. It shows that the variance is the highest when $D = 1$ and $D = 31$. The local maximum at $D = 16$ happens because the variance function changes rules at $D = \frac{N}{2}$. Also, the variance has two minimums. In order to find the places where minimum variance occurs analytically, we take the derivation and find for what values of D it equals to zero.

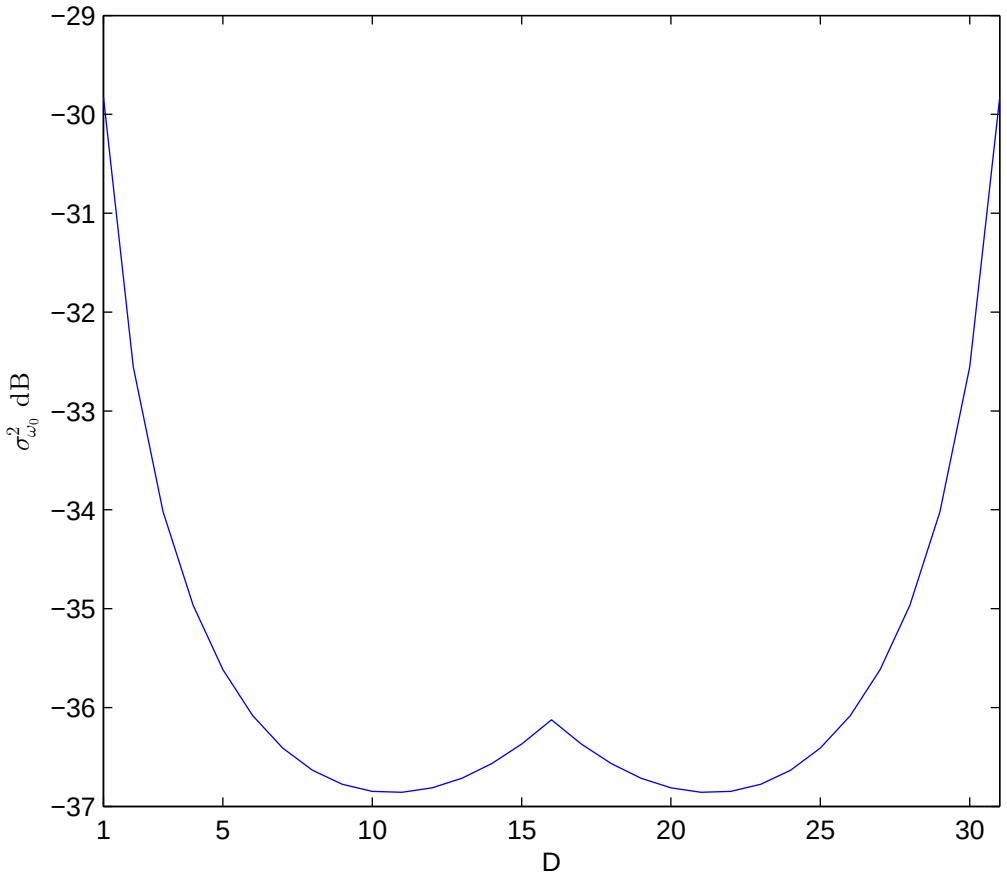

FIGURE 2.3: Variance of GUWPA when $N = 32$ for different values of D

$$\frac{d}{dD}\frac{1}{D(N-D)^2} = \frac{-3D^2 - N^2 + 4ND}{(D(N-D)^2)^2} = 0 \Rightarrow D = \frac{N}{3} \text{ for } D \leq \frac{N}{2} \tag{2.43}$$

$$\frac{d}{dD}\frac{1}{D^2(N-D)} = \frac{3D^2 - 2ND}{(D^2(N-D))^2} = 0 \Rightarrow D = \frac{2N}{3} \text{ for } D > \frac{N}{2} \tag{2.44}$$

So the two minimums occur at $\frac{N}{3}$ and $\frac{2N}{3}$. After solving the derivation, four point are found but two of them are invalid because they are out of range.

An intuitive way to understand why higher step values do not necessarily improve performance is to note that by increasing the value of steps we are in fact decreasing the number of our samples.

It is important to notice that the variance of GUWPA is independent of ω_0. This can be seen from equation 2.42. Also this is true for other estimators as well.

The variance for GWPA, GHP and GLHP is $\mathbf{1}^T\mathbf{C}^{-\mathbf{1}}\mathbf{1}$. Since none of the covariance matrices depend on ω_0, the variance is not dependent on ω_0 as well. For GULP the situation is similar.

Figure 2.4 shows that the variance of the estimators is nearly constant and confirms that the estimators are not dependent on ω_0.

Next chapter will discuss the performance of the estimators for different conditions.

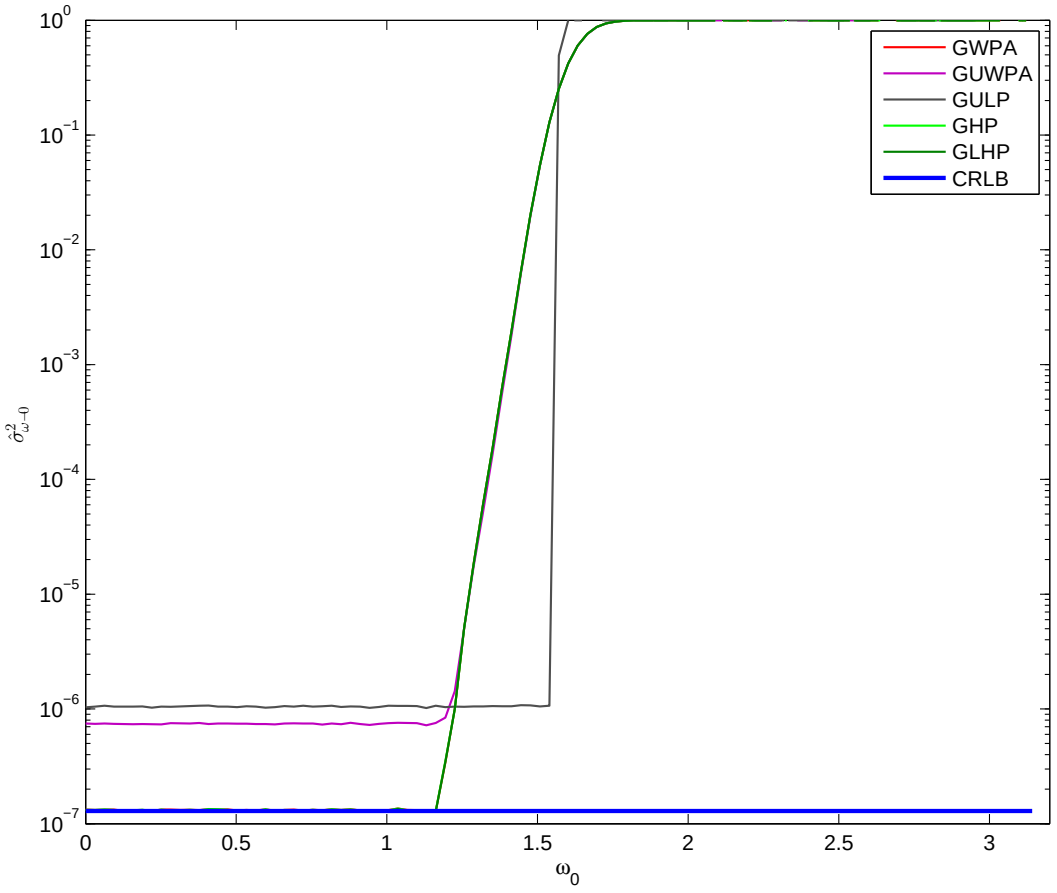

FIGURE 2.4: $\hat{\sigma}^2_{\omega_0}$ as a function of ω_0

Chapter 3

Performance assessment

3.1 Simulation Results

In this section, the performance of the estimator introduced in previous chapter is shown and discussed for several conditions. Three preamble lengths used for the simulations are $N = 4$ [11] for Bluetooth v2 [12], $N = 8$ for Bluetooth v4 and $N = 32$ for zigbee[13].

For comparing the performance of the estimators, we estimate mean square error(MSE) for different values of SNR, N or D ,depending on the type of graph. This performance metric is given by

$$\hat{\sigma}^2_{\omega_0} = \frac{1}{M} \sum_{i=0}^{M-1} (\hat{\omega}_{0_i} - \omega)^2 \tag{3.1}$$

where M is the number of simulations and $\hat{\omega}_{0_i}$ is the estimated value of ω_0 in each simulation. The results are then plotted against each other and CRLB. For simplicity we use the short name of the estimators. Table 3.1 summarizes the estimators, their short names, corresponding equation numbers and their colors in graphs.

Full Name	Short Name	Equation number	Color
Generalized uweighted phase avereger	GUWPA	2.36	Purple
Generalized uweighted Linear predictor	GULP	2.37	Gray
Generalized weighted phase avereger	GWPA	2.25	Red
Generalized H. Fu and P.Y. Kam	GHP	2.34	Light Green
Generalized Linear H. Fu and P.Y. Kam	GLHP	2.35	Dark Green

TABLE 3.1: A guide for reading plots

The estimators can be categorized into two distinct groups. Weighted estimators and unweighted estimators. The first group consists of GWPA, GHP and GLHP. The second group consists of GUWPA and GLUP.

3.2 $\hat{\sigma}^2_{\omega_0}$ as a function of SNR for different preamble lengths

Figure 3.1, shows the performance of estimators in form of $\hat{\sigma}^2_{\omega_0}$(in log scale) versus SNR(in dB scale) for different preamble lengths.

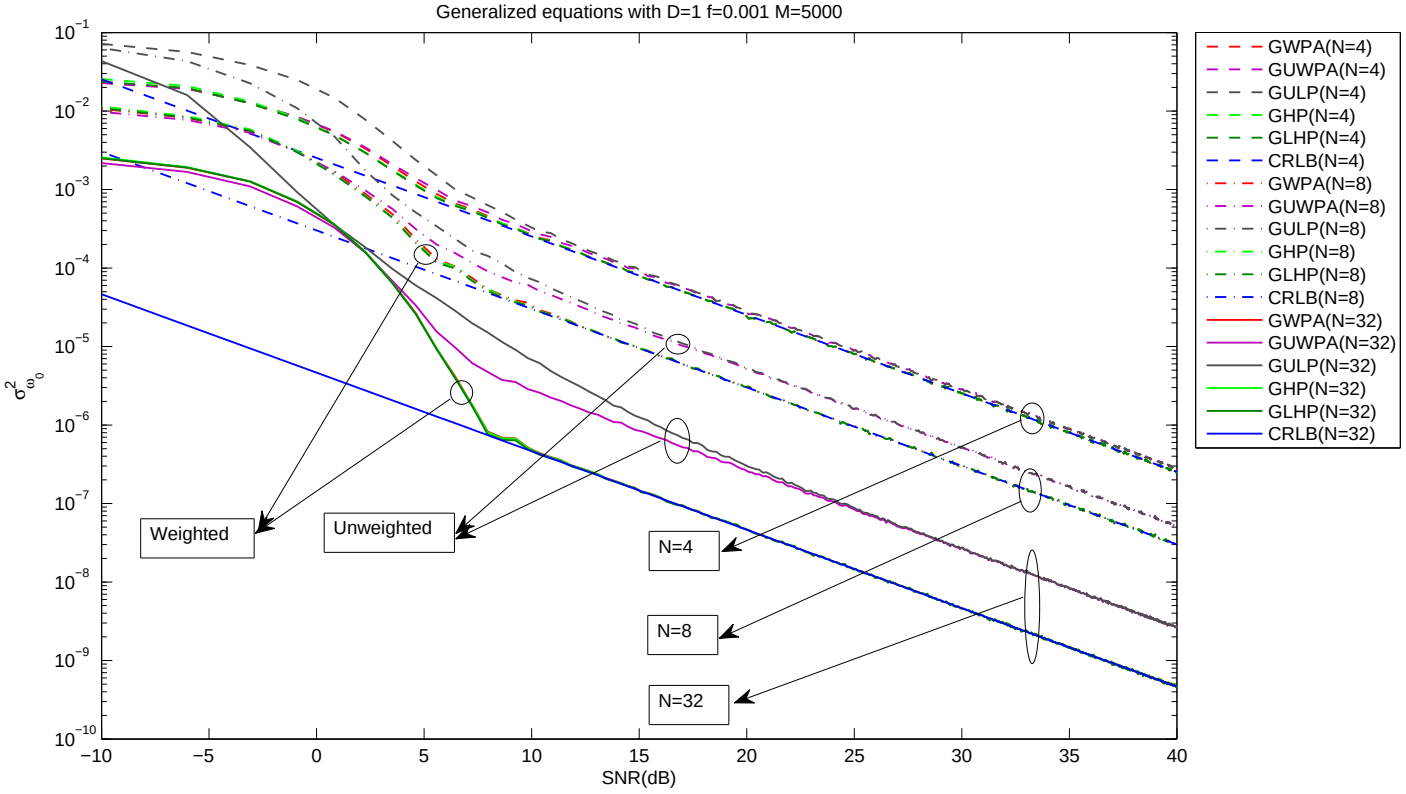

FIGURE 3.1: Performance of estimators for a $N = 4, 8, 32$, $D = 1$

These estimators are based on equation 2.2 which means that they are suitable for high SNR values. This is visible from the graphs in figure 3.1. For SNR values less than 5 dB, deviation from CRLB is high. The increase in SNR will decrease the deviation from CRLB. For higher values of N and even for high SNRs, the unweighted estimators do not reach CRLB. Instead they maintain a fixed distance from it. The higher the preamble length, N, the larger is the fixed distance relative to CRLB. The reason for this behavior is, as mentioned in the previous chapter, due to lack of proper windowing[2]. The results from this figure confirm the analytical results obtained in the previous chapter.

As expected the higher the preamble length(N) is, The better the estimators get. As shown later in the chapter, this improvement in performance is not linearly dependent on N.

Although GHP and GLHP are more accurate but this matter is not visible from figure 3.1. Figure 3.2 shows that as N increases, the difference between the performance of GHP, GLHP and GWPA decreases. This means that it is possible to use GWPA when the preamble length is long instead of GHP and GLHP without apparent loss of performance.

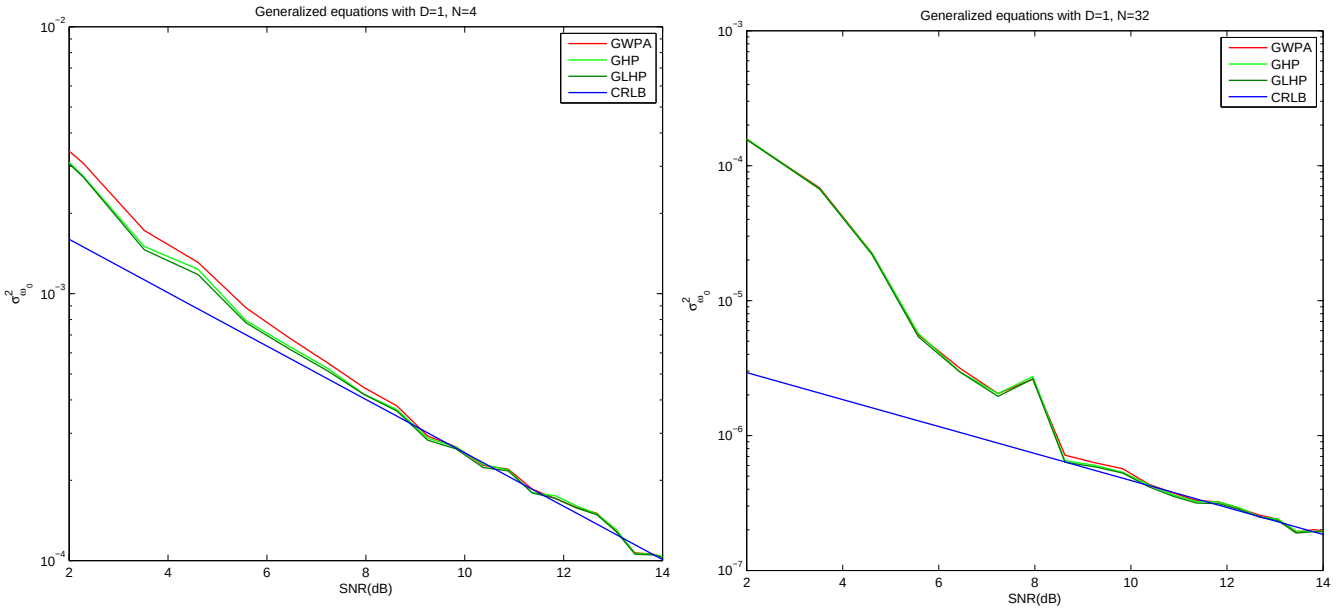

FIGURE 3.2: Performance of estimators for a $N = 4, 8, 32$, $D = 1$

3.3 $\hat{\sigma}^2_{\omega_0}$ as a function of D of different preamble lengths and SNR values

In this section, $\hat{\sigma}^2_{\omega_0}$ is plotted against D for SNR$= 0, 10, 16$ dB. It was mentioned in the last chapter on page 12, that not all values of D are suitable. This can be seen from the following figures as well.

Figure 3.3 shows that as SNR increases the performance of the estimators improves. For SNR$= 0$ dB, the best value of D is 3. For higher SNR values, $D = 1$ is best for weighted estimators and $D = 3$ for unweighted ones. Note that for high SNR values and $D \geq N/2$, the performance of all estimators is similar. As mentioned in the previous chapter, the reason for this similarity is that the covariance matrix of the weighted estimators becomes

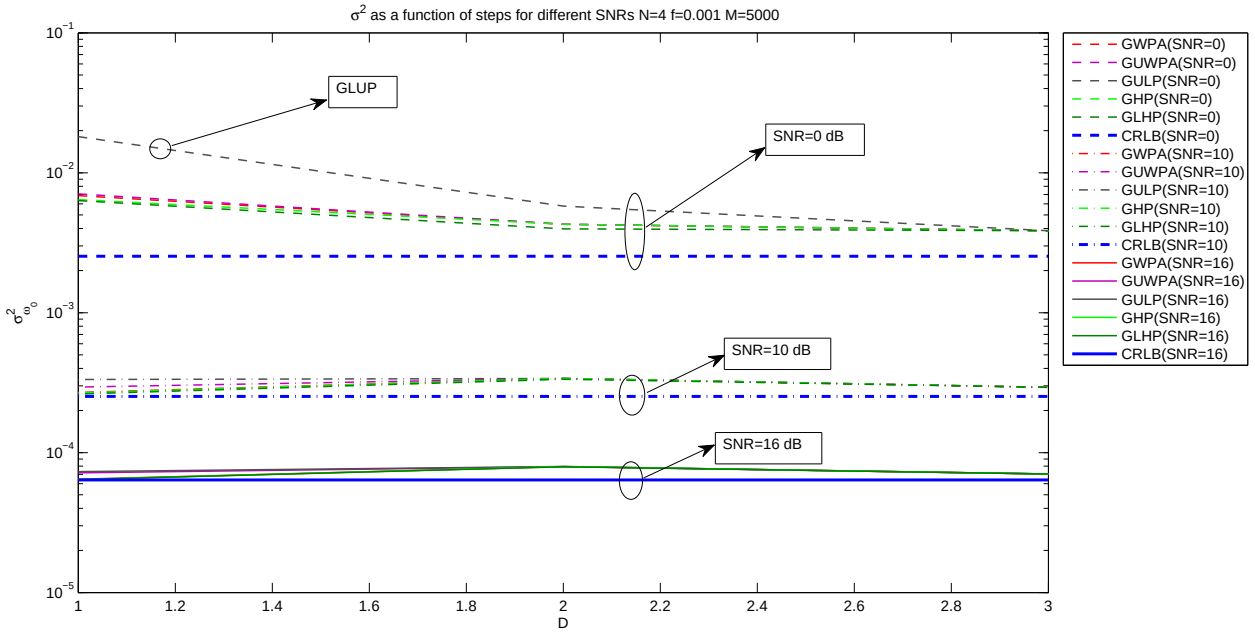

FIGURE 3.3: Performance of estimators for a $N = 4$ and different SNRs versus D

diagonal so they become similar to the unweighted estimators. This is proven for GWPA. For GHP and GLHP, it should be noted that their covariance matrices are similar. In fact the entries if covariance matrix of GWPA are approximations of covariance matrices of GHP and GLHP which leads to the similarity of their covariance matrix entries. This similarity in performance gives the option of using simpler estimators and get the same performance as the complex ones.

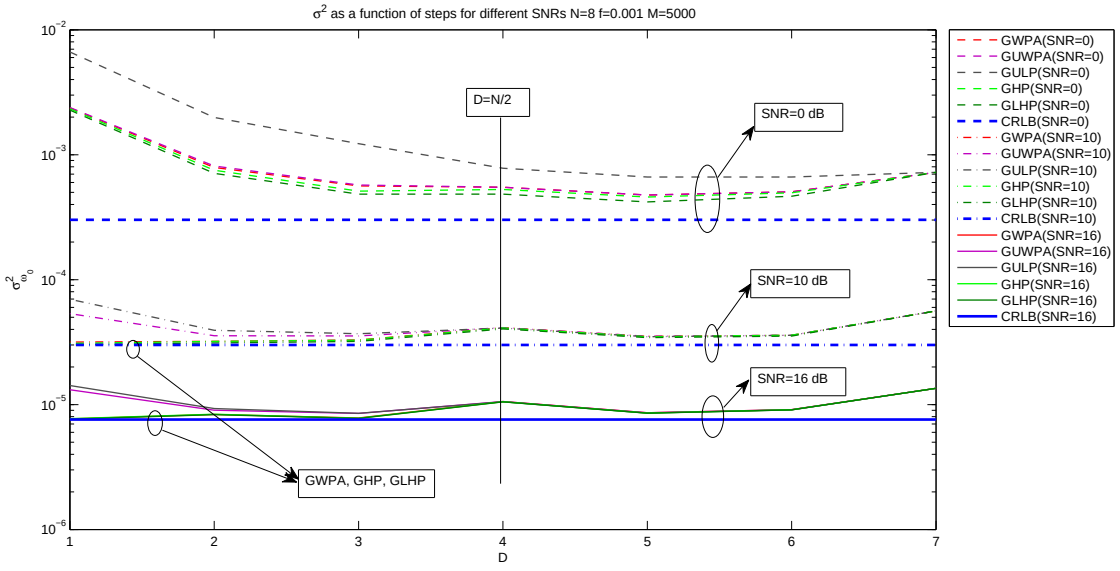

FIGURE 3.4: Performance of estimators for a $N = 8$ and different SNRs versus D

Figure 3.4 shows the performance of estimators when $N = 8$. Since the estimators are for high SNR conditions, not even the weighted ones reach CRLB when SNR= 0 dB. For the other two values of SNR, the weighted estimators reach CRLB for $D \leq 3$ and then when $D \geq N/2$ their performance matches the unweighted ones.

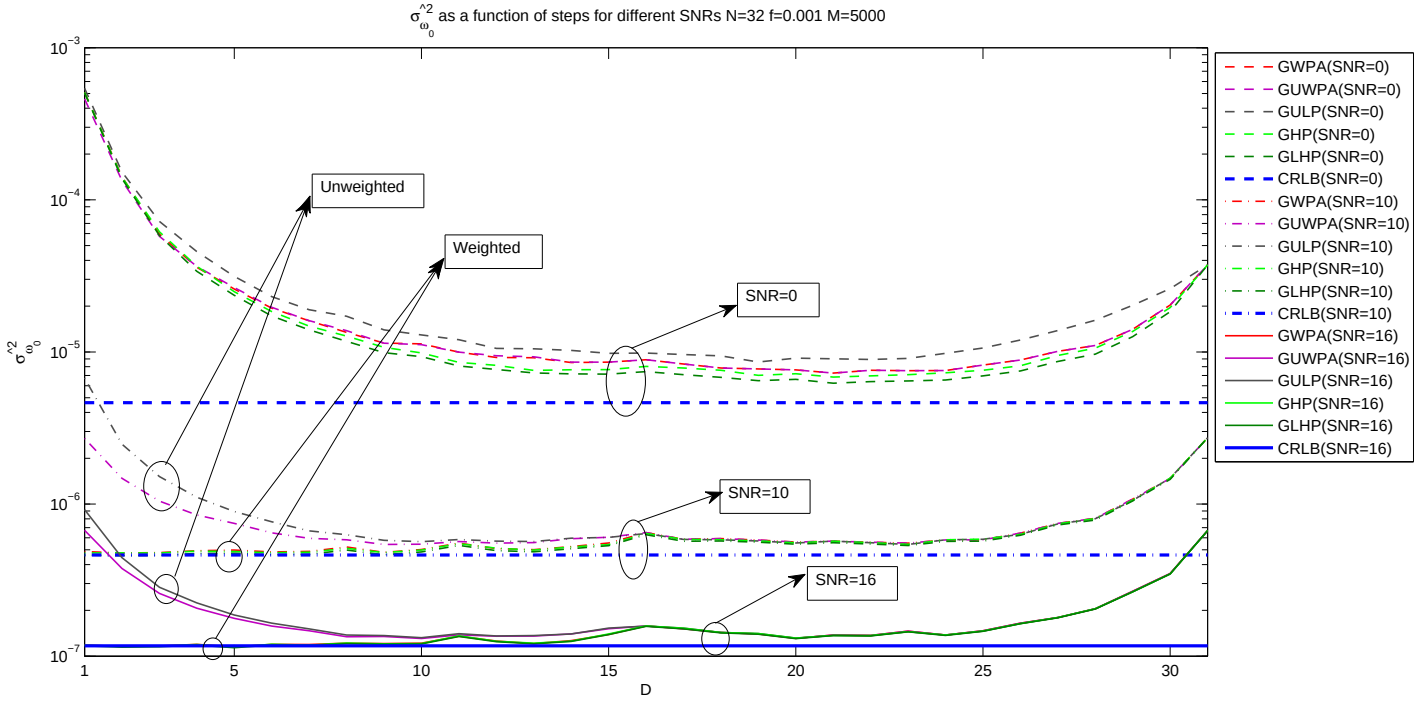

FIGURE 3.5: Performance of estimators for a $N = 32$ and different SNRs versus D

Figure 3.5 shows the performance of the estimators for $N = 32$. For low SNRs, all the estimators have low performance. As SNR increases, the performance of the weighted estimators for $D \leq \frac{N}{2}$ improves dramatically and their variances reach CRLB. For $D \geq \frac{N}{2}$, all the estimators show the same performance which is expected based on the results in the previous chapter. There are two minimums for GUWPA too. They occur around $D = N/3$ and $D = 2N/3$ as predicted analytically on page 14.

3.4 $\hat{\sigma}^2_{\omega_0}$ as a function of N for different SNR values

Figure 3.6 shows that as N increases, the estimators become more and more accurate. It should be noted that the estimators improve with different rates. It is obvious that the higher the SNR gets, the more accurate the estimators become. For high enough SNR values, the weighted estimators reach CRLB, whereas the unweighted ones keep a distance from it. Meaning that they are not asymptotically efficient which is already known from page 7. For SNR= 0 dB, GULP shows the worst performance compared

to other estimators, while others show similar performance. This effect is also visible in figures 3.3 to 3.5 for SNR= 0.

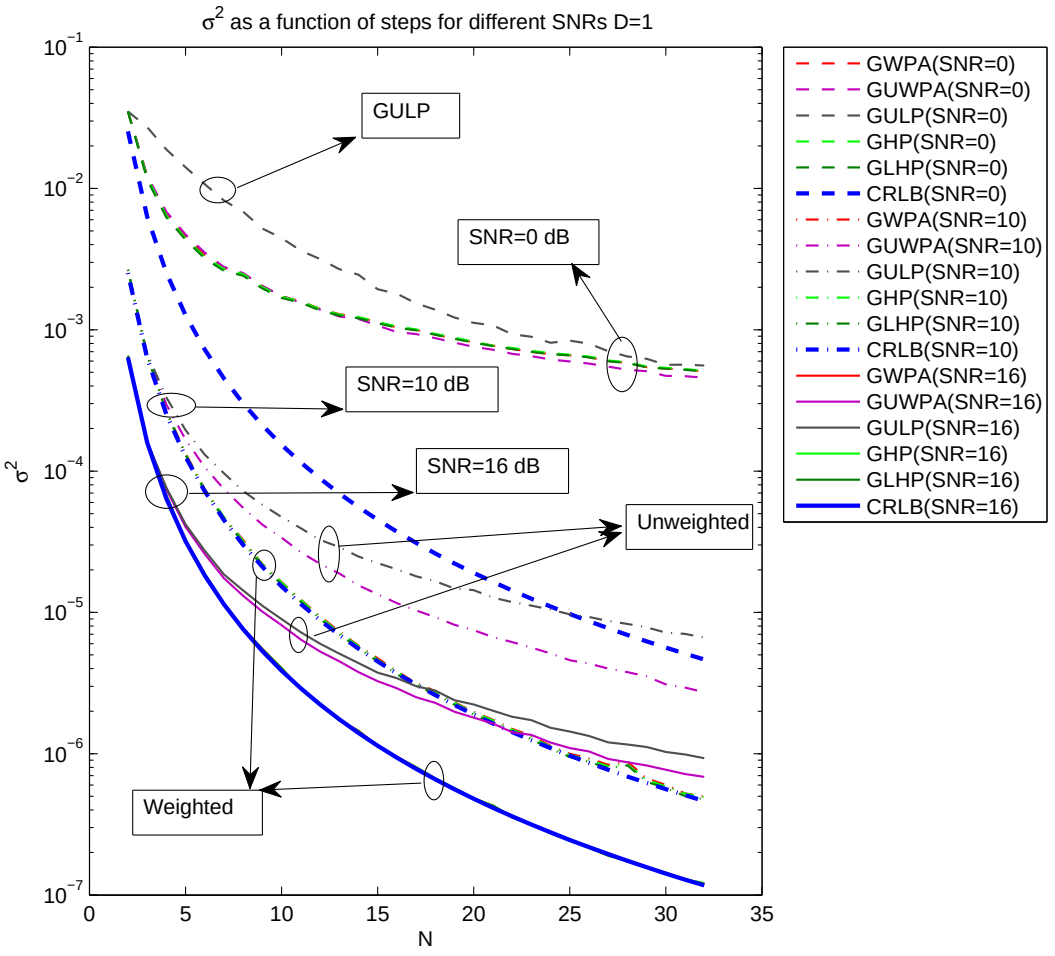

FIGURE 3.6: Performance of estimators versus N and for different SNRs and $D = 1$

Figure 3.7 shows the effect of steps on the estimators when $D = 3$. This value was chosen because for high SNR values and $N = 8, 32$, the weighted estimators reach CRLB. Having such step has increased the rate of performance improvement for all the estimators, which was obvious based on the figures of the last section.

3.5 $\hat{\sigma}^2_{\omega_0}$ as a function of SNR for different preamble lengths and $D = 3$

Figure 3.8 shows the effect of steps on $\hat{\sigma}^2_{\omega_0}$ versus SNR. It is seen from previous section that for $D = 3$, the estimators have better performance improvement rate than when $D = 1$.

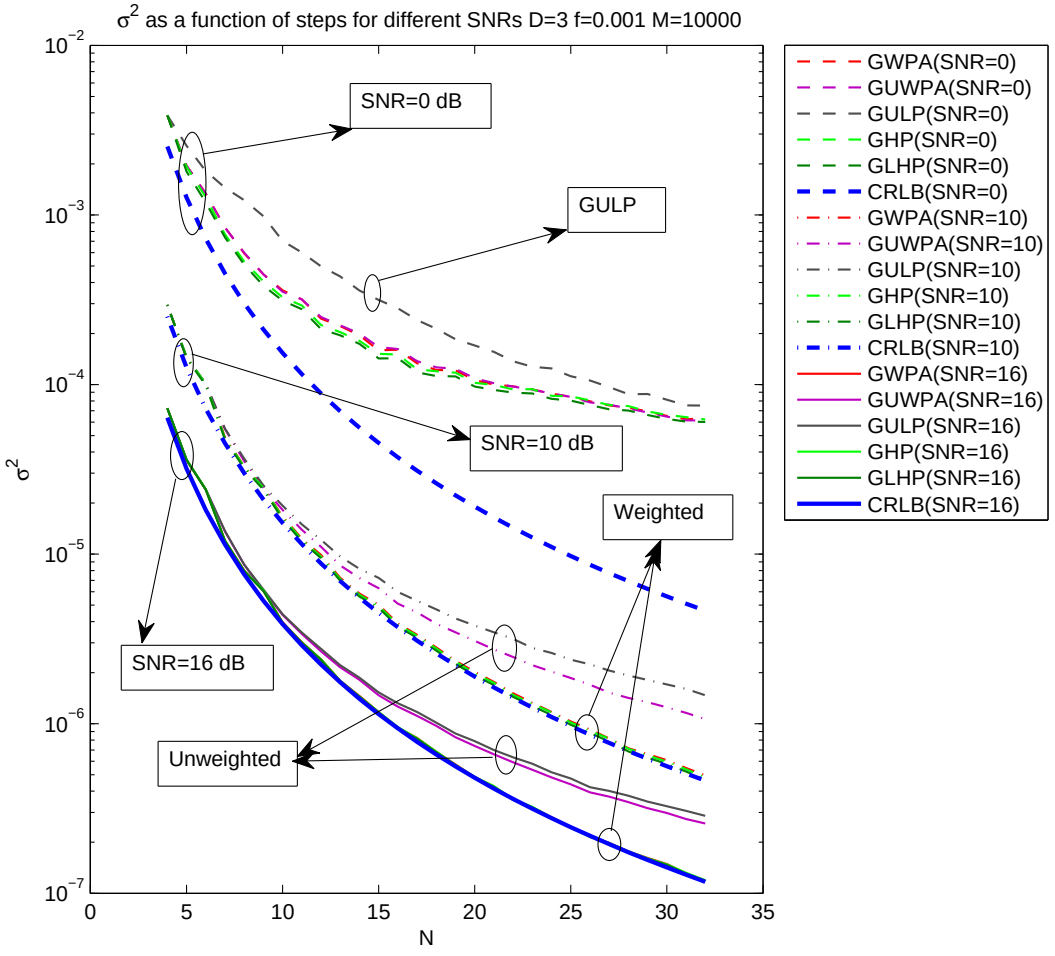

FIGURE 3.7: Performance of estimators versus N and for different SNRs and $D = 3$

Comparing figures 3.1 and 3.8 for $N = 4$ shows that when $D = 3$ all estimators show the exact same behavior and reach CRLB for lower SNR values compared to when $D = 1$. For $N = 8$ in figure 3.8 all the estimators, reach CRLB. This is not true when $D = 1$. For the case of $N = 32$, for both steps we have two distinct sets of estimators. The weighted estimators reach CRLB and the unweighted ones maintain a fixed distance from it. Yet it should be noted that the variance of the unweighted estimators is lower when $D = 3$. This is also visible in figures 3.3 to 3.5. The reason for this is that the ratio of variance to CRLB decrease when moving from $D = 1$ to $D = 3$. This effect is visible from figure 3.9 for GUWPA.

There is a local maximum on $N = 6$ for the graph for $D = 3$. The reason for this is because the variance of GUWPA is different for $D \leq N/2$ and $N \geq N/2$.

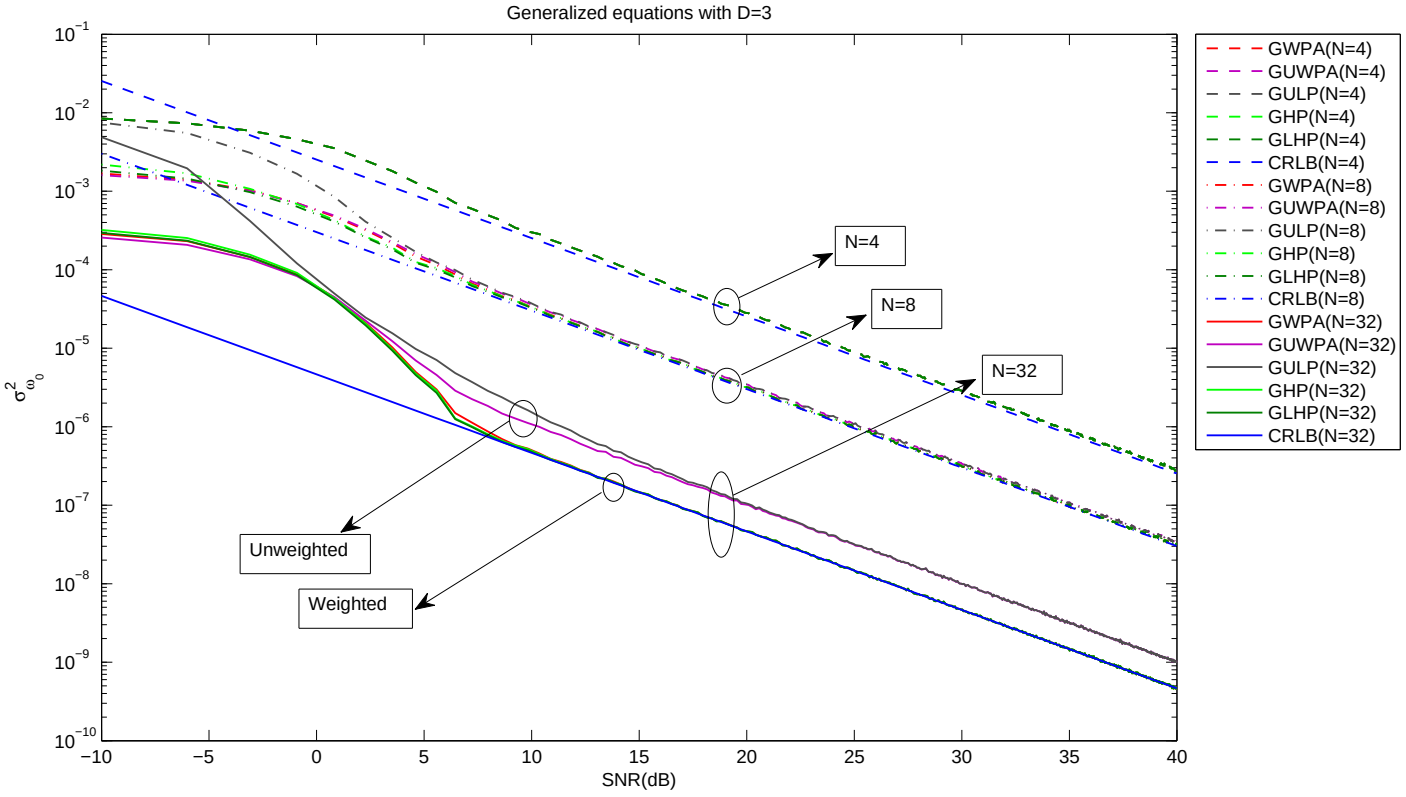

FIGURE 3.8: Performance of estimators versus SNR and for $D = 3$

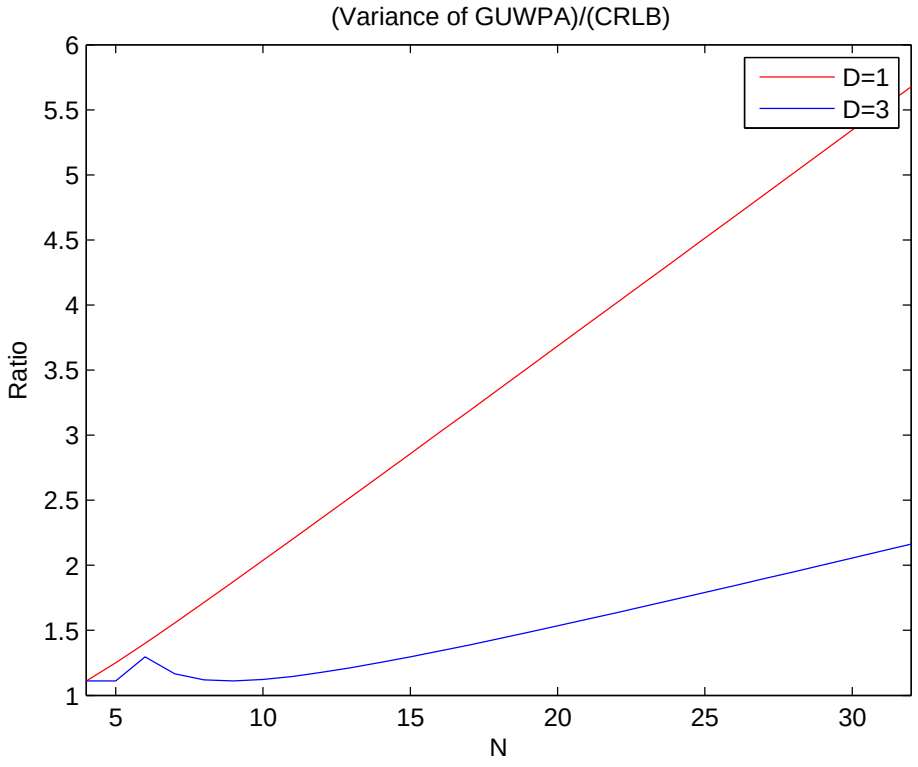

FIGURE 3.9: Performance of estimators versus SNR and for $D = 3$

Chapter 4

Computation Complexity

4.1 What is computation complexity and why is it important?

In academia, people mostly care about the performance of an algorithm or a model. However, performance is one of the parameters in industry . The other parameter is the cost. Every application has some limitations. These limitations define what algorithm or method can be used. There are many things that can be considered costly for a certain application in industry such as computation complexity and chip area. Here we calculate the computation complexity of the estimators. The reason for this is that the more operations there are, the more the consumed energy is. Algorithms used in devices with limited energy source such as mobile phones that use battery are in this category. On the other hand a computer does not have such restriction on energy, so it can use more complex algorithms.

Computation complexity means how many operations(summations, multiplications and divisions) are needed for a process and also tells how the number of operations grow if the samples grow large. For example, for adding two vectors of size N, we will need N additions. For finding the inner product of two vectors, we will need N multiplications and $N-1$ additions. Meaning that for calculating inner product we will need $2N-1$ operation. If the number of samples get large the part with N grows while the constant remains the same. For the sake of simplicity in this report, only the part that grows faster than others is considered for the computation complexity and other parts are neglected. So the complexity of inner product is N

4.1.1 Multiplication of two complex numbers

Multiplication of two complex number needs three real multiplication[14]. Consider two complex numbers $a + jb$ and $c + jd$. The answer is to multiplication is

$$(a + jb)(c + jd) = (ac - bd) + j(ad - bc).$$

It is possible to find each term by

$$a(c + d) - d(a + b) = ac - bd$$

and

$$a(c + d) - c(b - a) = ad - bc$$

meaning that only $a(c+d)$, $d(a+b)$ and $c(b-a)$ should be calculated to find the result.

4.1.2 Calculating the phase of a signal

There are many ways to calculate the phase of a signal. Here we use "Coordinate Rotation Digital Computer" or CORDIC[15]. This method only uses addition, subtractions and bit shifts and is suitable for the time when multipliers are not available.

In order to find the phase of a signal using CORDIC, the signal is rotated for L times (number of iterations) to zero out the imaginary part. After this step, the amount of rotations are added up together and the negative of the accumulated angles is the phase of the signal. The higher value for L means higher precision. CORDIC is an iterative algorithm and in each iteration uses three additions one for changing the real and imaginary parts and one for adding up the angle. Here we only count how many angle operations are done.

4.1.3 Calculating cost for matrix multiplication

Consider two matrices of size $N \times M$ and $M \times P$. To find the complexity, first we find the number of operations for calculating one entry which is $2M - 1 (M - 1$ additions and M multiplications). Now since the resulting matrix has $N \times P$ entries, it can be conclude that the number of operations needed is $2M - 1 \times N \times P((M - 1) \times N \times P$

summations and $M \times N \times P$ multiplications). So the computation complexity of matrix multiplication is $M \times N \times P$.

4.1.4 Calculating cost of matrix inversion

We calculate the cost for Gaussian elimination method. So for start we have to make all the entries bellow the first entry in the first row, zero. To make one zero, we will need N additions and multiplications. Since we want to do this with $N-1$ entries, we will have $N(N-1)$ operations. For the second entry of the second row we will need order of $(N-1)(N-2)$ operations to zero out the entries below. The addition of these operations will be

$$\sum_{i=1}^{N} i(i-1) = \sum_{i=1}^{N} i^2 - \sum_{i=1}^{i} i = \frac{N(N-1)(2N-1)}{6} - \frac{N(N+1)}{2}. \tag{4.1}$$

The cubed part grows faster than the other parts so the computation complexity of Gaussian elimination method is N^3.

There are other methods for calculating the inverse matrix too. In general the complexity of an algorithm mostly depends on how it is implemented. This will be made more clear later.

4.2 Calculating complexity for the estimators

Let's start by finding the computation complexity for the simplest estimators GUWPA and GULP.

GUWPA is

$$\hat{\omega}_0 = \frac{1}{D}\frac{1}{N-D}\sum_{t=1}^{N-D} \angle[x(t)^* x(t+D)]. \tag{4.2}$$

The value $\frac{1}{D(N-D)}$ can be pre-calculated and multiplied to the result of the sum. Also, $\angle[x(t)^* x(t+D)]$ can be written as $\angle x(t+D) - \angle x(t)$ to avoid complex multiplications. To make things more easier, we expand the summation and simplify the answer to get

$$\hat{\omega}_0 = \frac{1}{D}\frac{1}{N-D}\sum_{t=1}^{N-D} [\angle x(t+D) - \angle x(t)].$$

$$= \frac{1}{D}\frac{1}{N-D}[x(N) + x(N-1) + \ldots + x(N-D-1) - x(1) - \ldots - x(D)]. \quad (4.3)$$

So overall, regardless of N, one multiplication and $2D-1$ additions and subtractions are needed. Table 4.1 shows the number of operations that are needed in order to calculate $\hat{\omega}_0$. The total amount is written as $sS + mM + aA + dD$ meaning that s summations, m multiplications, a angles operations and d devisions are needed.

Summation	Multiplication	Angle	Total
$2D-1$	1	$2D$	$(2D-1)S + (1)M + (2D)A$

TABLE 4.1: Amount of operations needed to calculate $\hat{\omega}_0$ using GUWPA

So the computation complexity of GUWPA is not dependent on number of samples.

For GULP we have

$$\hat{\omega}_0 = \frac{1}{D}\angle[\frac{1}{N-1}\sum_{t=1}^{N-D} x(t)^* x(t+D)] \quad (4.4)$$

Unlike GUWPA, it is not possible to simplify GULP much. Only that $\frac{1}{D}$ and $\frac{1}{N-1}$ can be pre-calculated. So one angle operation, $3(N-D)+2$ real multiplications and $(N-D)-1$ additions are needed. Table 4.2 shows the amount of calculations needed for GULP.

Summation	Multiplication	Angle	Total
$N-D-1$	$3(N-D)+2$	1	$(N-D-1)S + (3(N-D)+2)M + (1)A$

TABLE 4.2: Amount of operations needed to calculate $\hat{\omega}_0$ using GULP

Now we look at weighted phase averager(WPA)

$$\hat{\omega}_0 = \sum_{t=0}^{N-2} w(t)\angle[x(t)^* x(t+1)]. \quad (4.5)$$

$$= \sum_{t=0}^{N-2} w(t)[\angle x(t+1) - \angle x(t)]. \quad (4.6)$$

The windowing functions, $w(t)$, is not related to signal and can be calculated once, then stored and used. This means that it has no role in computation complexity of WPA yet it takes memory space. We can reduce the needed memory space if we take into account

the symmetric nature of the windowing function. The closed form driven by Rosnes and Vahlin in equation 2.26, has the same computation complexity.

Table 4.3 shows the amount of operations needed to estimate ω_0.

Summation	Multiplication	Angle	Total
$2(N-1)$	$N-1$	$2(N-1)$	$(2N-2)S+(N-1)M+(2N-2)A$

TABLE 4.3: Amount of operations needed to calculate $\hat{\omega}_0$ using WPA

Now we look at GWPA which uses

$$\hat{\omega}_0 = \frac{1}{D}\frac{\mathbf{1}^T\mathbf{C_v^{-1}\Delta}}{\mathbf{1}^T\mathbf{C_v^{-1}1}} \tag{4.7}$$

with covariance matrix entires

$$c_{i,j} = \begin{cases} \dfrac{\sigma^2}{A^2}, & \text{if } i=j \\ -\dfrac{\sigma^2}{2A^2}, & \text{if } |i-j|=D \\ 0, & \text{Otherwise.} \end{cases} \tag{4.8}$$

In the following, two different methods for implementing this estimator are discussed.

The first method is to calculate equation 4.7 every time. The inverse of covariance matrix is pre-calculated. Table 4.4 shows the amount of operations needed to calculate $\hat{\omega}_0$.

Summation	Multiplication	Devision	Angle	Total
$2N^2+N-1$	$2N^2+1$	1	$2N$	$(2N^2+N-1)S+(2N^2+1)M+(2N)D+(1)A$

TABLE 4.4: Amount of operations needed to calculate $\hat{\omega}_0$ using first implementation of GWPA

The second method is based on the discussions of page 8. We use the fact that the covariance matrix can be written as $\frac{\sigma^2/2}{A^2}.\mathbf{C}$ where entries of $\mathbf{C}$ are shown by equation 4.9

$$\gamma_{i,j} = \begin{cases} 1, & \text{if } i=j \\ -\dfrac{1}{2}, & \text{if } |i-j|=D \\ 0, & \text{Otherwise.} \end{cases} \tag{4.9}$$

If $\mathbf{C_v}$ is inversed, we get $\frac{A^2}{\sigma^2/2}\mathbf{C^{-1}}$. Putting this in equation 4.7, we see that $\frac{A^2}{\sigma^2/2}$ vanishes which makes it possible to pre-calculate and store $\frac{\mathbf{1}^T\mathbf{C^{-1}}}{D\mathbf{1}^T\mathbf{C^{-1}1}}$ and use it for all the received signals. Table 4.5 shows the amount of operations needed to implement GWPA.

Summation	Multiplication	Angle	Total
$2N-1$	N	$2N$	$(2N-1)S+(N)M+(2N)A$

TABLE 4.5: Amount of operations needed to calculate $\hat{\omega}_0$ using second implementation of GWPA

As it was mentioned before, the computation complexity of an algorithm depends on the method it is implemented in.

The computation complexity for GHP and GLHP are as follows.

When calculating equation 4.7, the covariance matrix and its inverse should be calculated each time the signal is received. First we look at the amount of operations that are needed to calculate the covariance matrix. $|x(t)|^2$ is calculated by squaring the signal amplitude thus avoiding complex multiplication. There are algorithms that can estimate the amplitude of $x(t)$ using only additions and bit shifts. One of these algorithms is CORDIC. It is noticeable that $\frac{\sigma^2}{2}$ simplifies from numerator and denominator. Table 4.6 shows this amount for the estimators.

Method	Sum	Multiplication	devisions	Total
GHP and GLHP	$N-D$	$3N-3D$	$4N-6D$	$(N-D)S+(3N-3D)M+(4N-6D)D$

TABLE 4.6: Amount of operations needed to calculate the covariance matrix of GHP and GLHP

Complexity of calculating the inverse of covariance matrix is N^3 and complexity of calculating equation 4.7 is $N^2(2N^2$ multiplications and additions). Computation complexity of calculating GHP and GLHP is shown by table 4.7.

Method	summation	multiplication	Devision	Angle
Equation 4.7	$2(N^2-2ND+D^2+N-D-1)$	$2N^2$	N^2	$N-D$

TABLE 4.7: Computation complexity of calculating of equation 4.7 for GHP and GLHP

Among all these terms, the inversion needs the most amount of operation and so is the dominating term. This means that in order to calculate GHP and GLHP at least N^3 operations are needed.

Table 4.8, shows the dominating term in computation complexity of the estimators.

In the next chapter we consider the results of both chapter 3 and chapter 4 and make a conclusion.

Method	Equation Number	Cost
GUWPA	2.36	$Not dependent on N$
GULP	2.37	N
WPA	2.18	N
GWPA(first implementation)	2.25	N^2
GWPA(second implementation)	2.25	N
GHP	2.34	N^3
GLHP	2.35	N^3

TABLE 4.8: Computation complexity of the estimators

Chapter 5

Discussion and Conclusion

5.1 Discussion

It can be seen from Chapter 3 that the weighted estimators show better performance compared to the unweighted ones, but as shown in Chapter 4, some are much more complex too. Based on the results of the last two chapters, GUWPA seems to be the best choice for low power applications because its complexity is independent of the number of samples and its error can be tolerated for low N's. The second best is the second implementation method of GWPA because its computation complexity is linear and for high enough SNR, it reaches CRLB. Although GHP and GLHP have the best performances(slightly better than GWPA as shown in Chapter 3), yet because of their computational complexity which stems from inverting the signal dependent covariance matrix, they are not suited for low energy applications.

From tables 4.1 and 4.2 it can be seen that for $D < \frac{2N}{3}$, GULP has more multiplications than GUWPA, WPA and GWPA. Since it shows the same performance as GUWPA for high SNR values GUWPA is more preferable.

Tables 5.1 and 5.1 sort the estimators based on their performance and computation complexity.

Cost Rank	Method	Eq. NO	Performance Rank	Method	Eq. NO
1	GUWPA	2.36	1	GLHP	2.35
1	WPA	2.18	2	GHP	2.34
1	GWPA(2nd)	2.25	3	GWPA(2nd)	2.25
1	GULP	2.37	3	GWPA(1st)	2.25
2	GWPA(1st)	2.25	4	WPA	2.18
3	GHP	2.34	5	GUWPA	2.36
3	GLHP	2.35	6	GULP	2.37

5.2 Conclusion

Single frequency estimators were introduced to recover the frequency of a received signal from a preamble. It was shown that using an arbitrary step size can produce better results given that the step is chosen wisely. This improvement comes with a cost and that is loss of the range of frequencies which can be used. Generalized versions of the estimators proposed by Hua Fu and Pooi Yuen Kam was derived. It was proven for one estimator that not all step are suitable and higher step values will not necessarily improve performance. The performance of all of the estimators were evaluated for several conditions. It was shown that as N grows, the performance of estimators improves too yet not in a linear way. Also the computation complexity of the estimators was calculated. It was shown that it is possible to reduce the complexity of generalized weighted phase averager by pre-calculating and storing its weights which make GWPA a suitable candidate for low energy applications. The most suitable candidate was GUWPA because its complexity does not depend on N and it needs less amount of operations.

Bibliography

[1] H. Fu and P.Y. Kam. Ml estimation of the frequency and phase in noise. In *Global Telecommunications Conference. IEEE*, pages pp. 1–5, Nov. 27 2006-Dec. 1 2006.

[2] S. Kay. A fast and accurate single frequency estimator. *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37:pp. 1987–1990, Dec. 1989.

[3] H. Fu and P.Y. Kam. Improved weighted phase averager for frequency estimation of single sinusoid in noise. *Electronics Letters*, vol. 44:pp. 247–248, Jan. 2008.

[4] S. Tretter. Estimating the frequency of a noisy sinusoid by linear regression. *IEEE Transactions on Information Theory*, vol. 31:pp. 832–835, Nov. 1985.

[5] H. Fu and P.Y. Kam. Weighted phase averager for frequency estimation of a noisy single sinusoid: Application of the observation phase noise model. In *IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*, pages pp. 1923–1927, Sept. 2009.

[6] Kai-Hsin Chen and Hsi-Pin Ma. A low power zigbee baseband processor. *Internationa SoC Design Conference(ISOCC)*, vol. 1:pp. 40–43, Nov. 2008.

[7] E. E. Rosnes and A. Vahlin. Generalized kay estimator for the frequency of a single complex sinusoid. In *IEEE International Conference on Communications(ICC)*, volume vol. 10, pages pp. 2958–2962, Jun. 2001.

[8] Steven M. Kay. *Fundamentals of Statistical Signal Processing, Estimation Theory*. Prentice Hall, 1993.

[9] J.K. Wolf and J.W. Schwartz. Comparison of estimators for frequency offset. *IEEE Transactions on Communications*, vol. 38:pp. 124–127, Jan. 1990.

[10] F. Classen, H. Meyr, and P. Sehier. An all feedforward synchronization unit for digital radio. In *43rd IEEE Vehicular Technology Conference*, pages pp. 738–741, May. 1993.

[11] Roger Garvert. Bluetooth low energy. URL www.fte.com/docs/Ble_101_frontline.pps.

[12] Agilent Technologies. BluetoothÂő measurement fundamentals, application note. URL cp.literature.agilent.com/litweb/pdf/5988-3760EN.pdf.

[13] Jie Liu Simon Gerber, Chieh-Jan Mike Liang. Surviving wi-fi interference in low power zigbee networks. URL www.disco.ethz.ch/lectures/fs11/seminar/slides/gerber.pdf.

[14] I. Munro. Some results concerning efficient and optimal algorithms. *3rd Annual ACM Symposium on Theory of Computing*, pages pp. 40–44, 1971.

[15] Jack E. Volder. The cordic trigonometric computing technique. *IRE Transactions on Electronic Computers*, vol. EC-8:pp. 330–334, Sept. 1959.