January 2008

Radar Waveform Design for Classification and Linearization of Digital-to-Analog Converters

Cagatay Capar
University of Massachusetts Amherst

RADAR WAVEFORM DESIGN FOR CLASSIFICATION AND LINEARIZATION OF DIGITAL-TO-ANALOG CONVERTERS

A Thesis Presented

by

CAGATAY CAPAR

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL AND COMPUTER ENGINEERING

September 2008

Electrical and Computer Engineering

RADAR WAVEFORM DESIGN FOR CLASSIFICATION AND LINEARIZATION OF DIGITAL-TO-ANALOG CONVERTERS

A Thesis Presented

by

CAGATAY CAPAR

Approved as to style and content by:

Dennis L. Goeckel, Chair

Patrick A. Kelly, Member

Hossein Pishro-Nik, Member

C. V. Hollot, Department Head
Electrical and Computer Engineering

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

Every practical system shows nonideal behavior. Due to limited precision, poor design, imperfect components or ignored external conditions, systems perform different than what they are intended to do. It is the engineer's task to find ways of making performance in the face of such flaws as close to ideal as possible. In this thesis we are trying to enhance systems that are components of important applications.

The thesis work we are presenting is mainly divided into two research projects. The first one is a target identification project for radars, and the second one is a predistorter mechanism design to compensate for nonlinearities of digital-to-analog converters (DACs). In this chapter, we explain our motivation for trying to solve these problems, give a short background needed to understand our effort and finally summarize our contributions. In the radar target identification project, we try to find the optimal signal to be used in automobile radars to maximize their target detection performance. The details of this project are given in Chapter 2. Our second project is about an enhancement to digital-to-analog converters to compensate for nonlinearities. We try to mitigate this undesired effect by constructing a predistorter. The details of this project are explained in Chapter 3. We present our conclusions derived from our work on these projects in Chapter 4.

1.1 Motivation

Radar systems are built with the aim of detection of a target. As with all real systems, radars suffer from nonideal conditions which lead to incorrect decisions.

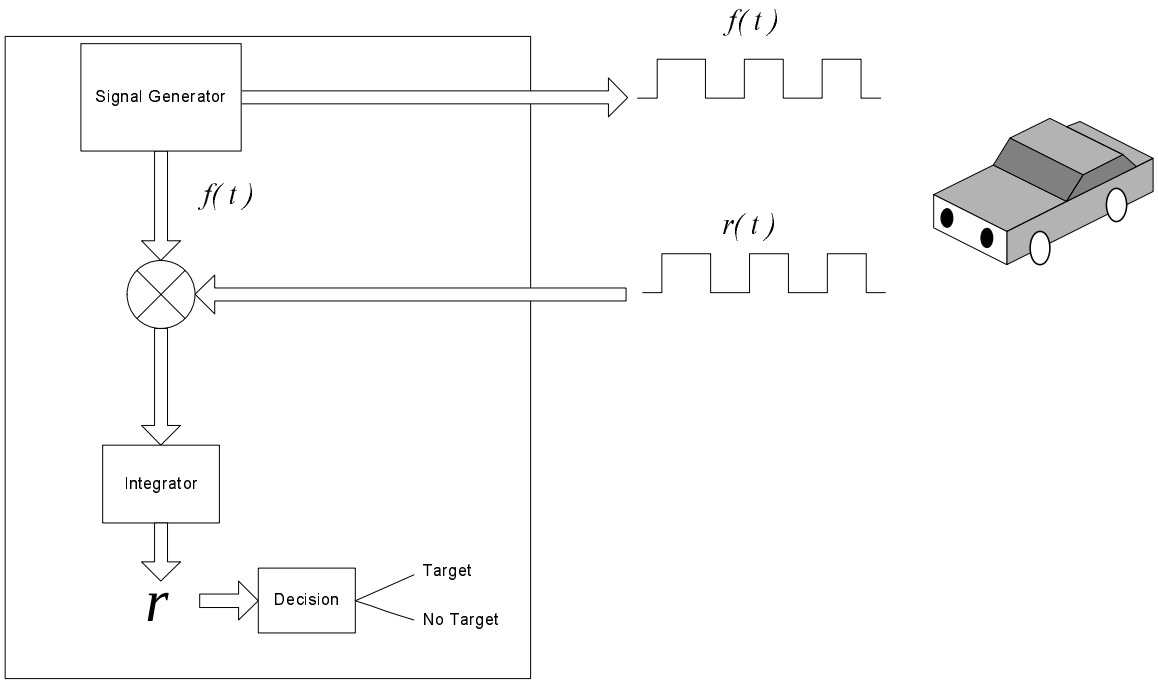

Figure 1.1. Radar Model.

These incorrect decisions may be in the form of failing to detect an actual target or wrongly declaring the presence of a target (false alarm).

Our research problem is about the radars placed in the bumpers of automobiles. Their task is to continuously check for other vehicles and to warn the car to slow down in the case of a detection. Figure 1.1 explains how these radars work. The radar system sends a signal, and, if there is another vehicle in the range, this vehicle reflects the radar signal back. As soon as the radar detects the vehicle, it warns the car to slow down to avoid collision. These radars are already in use; however, they require an important enhancement. Although they are often successful in detecting other vehicles, they sometimes give false alarms and slow the car unnecessarily. These false alarms are mostly because of small objects around the road that reflect the radar signal back.

If we can find a way to improve their identification performance to minimize false alarms, we can make this system a more reliable one. In other words, we want to make these radars more intelligent ones which can distinguish vehicles from other objects. The signals used in the radar systems we are working on are often selected as the simplest one to generate, ignoring their potential effect on detection performance. In our project, we try to answer the question of whether it is possible or not to increase detection performance by altering the radar signal. If the answer to this question is yes, the radar system in hand can be made a better one with just a simple change on the signal generated.

Having explained all of this, we can now specify our problem as a *radar target identification problem*. Here it should be noted that there has already been a lot of work done for similar problems. See [1] for an example. A unique aspect of our problem that makes it different than those works is the constraints determined by the radar we are working on. In most of the radar target classification literature, the reflected signal from a possible target is fully available to the algorithm. This enables an easier study with the help of increased information. The radars used in the cars, however, base their decision on the correlation of the sent and the received signal. In other words, we have access to the information that can be derived only from the result of this correlation. This makes the linear properties of the reflection process inapplicable in our project.

One can view the target identification problem as hypothesis testing where the two hypotheses are the targets of concern and the targets to ignore. Hypothesis testing is a well studied subject and we can make use of its principles. Our problem, however, is not a simple hypothesis testing problem. Every vehicle can have a completely different reflection and, likewise, not all the small objects can be represented by a single reflected signal. Thus, our hypotheses are the collection of all the possible reflections from a large object against all the possible reflections from a small object.

This makes our problem fall into the area of robust hypothesis testing, where the hypotheses are represented by uncertainty classes. Our aim is to maximize the worst case performance.

Last of all, we should also mention a possible outcome of our study. As we stated above, our ultimate goal is to choose the best signal to send. Along with comparing the available signals in terms of their identification performances, we can also guess what the optimal signal would be were we not constrained with the capabilities of our radar. If this optimal signal significantly improves identification, we can even propose a change to the radar circuit. Or, on the contrary, we might be able to state that the signals in the available set are already sufficient and eliminate the question whether we are losing optimality by being constrained to that set.

As in the first project, our aim in the second project is to lessen the adverse effects of nonideal conditions of a particular type of system, namely a digital-to-analog converter. DACs are used in every digital system where the output has to be a physical analog signal. For example, the sound cards in computers take the digital media file and process it, but, independent of what they do digitally, they need to convert the final signal to analog so that we can hear it. Since DACs are very important components of many electronic systems we use, their improvement has a broad potential application area.

Ideally, DACs are built to generate the analog signal corresponding to its digital counterpart. However, in real DACs, the analog output differs from the theoretical one in the sense that the relation with the digital input is not linear. This can be seen from Figure 1.2, where the dashed line shows the ideal input-output relation. The solid curve is a typical input-output plot of a DAC and it deviates from the dashed line and shows a nonlinear characteristic.

What we attempt to do is to build a predistorter system that will change the digital input in such a way that the analog output is close to the ideal one. In other

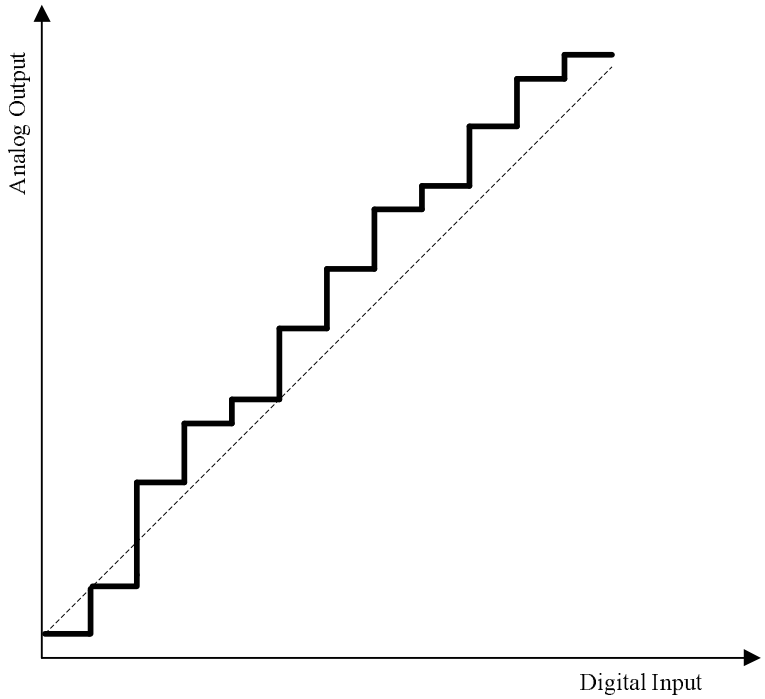

Figure 1.2. DAC output vs. input plot in ideal (dashed) and practical (solid) cases.

words, we will precompensate for the nonlinearity of the DAC. That will enable us to generate the required output without changing the DAC itself. Precompensation for nonlinearity is not a new idea and is often used for power amplifier circuits. We are going to try to apply this idea to DACs. With the help of the knowledge about amplifier precompensation, we think we can understand what can be done for DACs. Unlike for the amplifiers, however, the compensation system for a DAC has to be of low-complexity because of the high sampling rate.

1.2 Background

Let us start with the radar target identification project. We have to know the role of the radar signal in the detection to understand how changing it can help us. So, we first explain the basic details of the radar we are working on. Second, we will

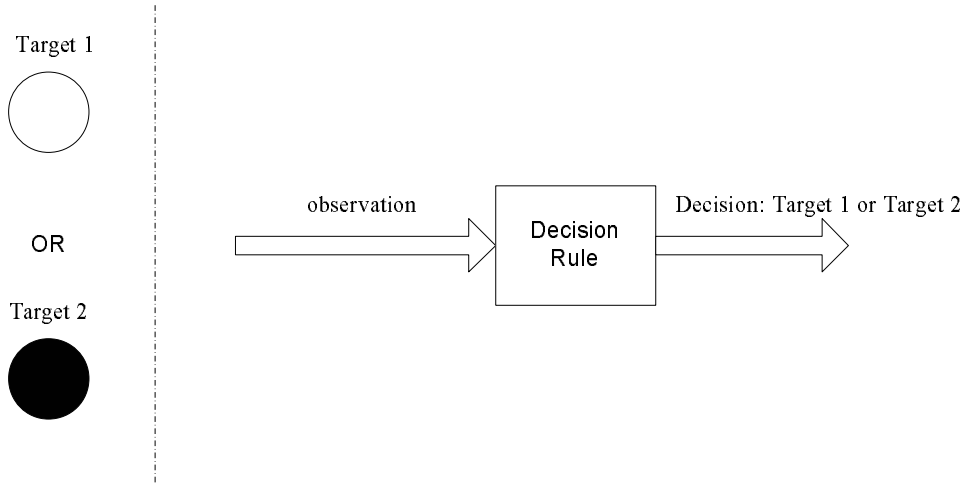

Figure 1.3. Simple Hypothesis Testing

give the basics of hypothesis testing along with a specific area of such, namely robust hypothesis testing.

Figure 1.1 shows the basic components of the radar system. As can be seen, the radar is sending $f(t)$. After sending this signal, the radar receives a signal, $r(t)$. This might be a reflection from a target or just a noise signal in the situation of no target. What our radar does is simply correlate (multiply and integrate) $f(t)$ and $r(t)$ and end up with a scalar r. Based on this scalar, the radar decides whether a target is present or not. The basic idea is that, if a target is not present, the returned signal will mostly consist of noise which does not give a large correlation. However, if there is a target, the observed signal, $r(t)$, will be a closer signal to $f(t)$. This will give a larger correlation than the first case.

Based on this model, let us rephrase our above mentioned goal. We are trying to choose $f(t)$ that results in the best detection. How the detection performance is measured will be explained in Chapter 2.

Hypothesis testing is to decide between some choices (hypotheses) based on an observation. Figure 1.3 explains how hypothesis testing works. Either target 1 or

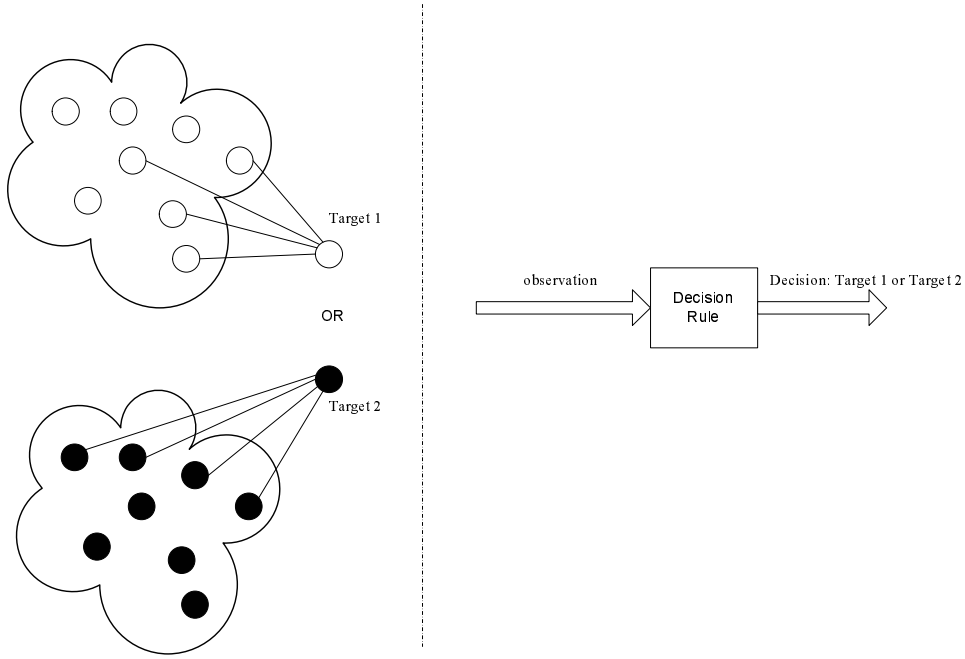

Figure 1.4. Hypothesis testing between uncertainty classes

target 2 is present, but our observation is not perfect. This means that, for example, target 1 might lead to different observations each time we check. Based on this noisy observation, we are trying to come up with the optimal decision rule. The optimal decision rule is the one that minimizes the average cost of decision. When we come to our problem, target 1 represents large objects and target 2 represents small objects. The observation is the correlation we talked about above, and the decision is whether to ignore the target or take action.

As we explained in the preceding section, this model is too simple for our particular problem. Target 1 and target 2 cannot be described by a single representative. They, instead, are instances from a large class of targets. In detection language, these are called uncertainty classes. The situation is explained in Figure 1.4. Large objects and small objects constitute two classes we are trying to distinguish between. Each time, a member from one of these two classes is present and we try to decide based on our observation from this representative. Thus, our aim here is to find a decision rule

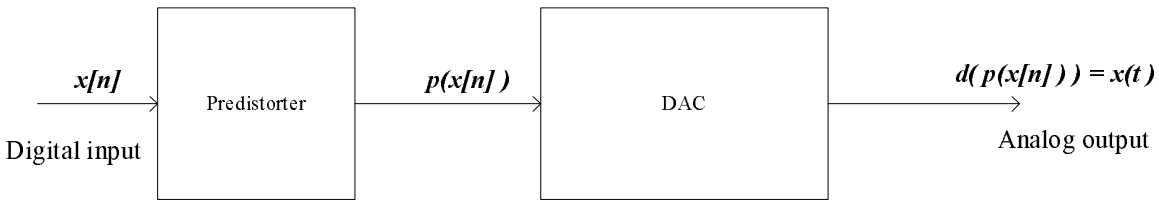

Figure 1.5. DAC with predistorter.

that will minimize the cost over all possible pairs from these classes. Our problem becomes *robust hypothesis testing*.

The success of the decision rule we find, of course, depends on our measure. In our case, we desire our system to maximize its performance for the worst case. This type of optimization is called *minimax optimization*. So, a more precise definition of our problem is minimax robust hypothesis testing. There are works done in this area of detection as well [2], [3], [4].

In our work, we first try to solve the simple hypothesis testing problem by assuming fixed targets. Then we try to generalize our findings to family of targets. As we will see, the properties of the target classes significantly affects the complexity of the problem.

When it comes to the DAC enhancement project, the best way to explain the problem is with the help of the input-output relation of a typical DAC.

Figure 1.2 explains this relation. In a digital-to-analog conversion, every digital input level (i.e. code) corresponds to a different analog value. Most often every single bit increase in the digital code results in a fixed amount of increase in the analog output. This difference between successive codes are called LSBs for short (stands for Least Significant Bit). This fixed difference leads to a linear relation between the digital input and the analog output. This is shown by the dashed line in the plot. In the implementation of DACs, however, it is very hard to realize equal amplitude

differences between successive codes. This results in a nonlinear input-output relation. This nonlinearity causes distorted outputs which we are trying to fix.

The above mentioned relation is a good instructive example of a source of nonlinearity in DACs. However, it is far from being the only reason behind the nonlinearity. There are other effects explained in Chapter 3 and they become more dominant under common operating conditions. However, it is important to know that, in our project we are not trying to change the nonideal performance of the DAC. We are accepting it as a fact and modeling it as well as possible. After modeling, we try to find a way to change the input so that it produces what is desired, i.e. we predistort the input for nonlinearity compensation. Figure 1.5 shows how the predistorter mechanism is implemented. Suppose we have the digital input $x[n]$ which we are trying to convert to $x(t) = a(x[n])$. However, the DAC produces $d(x[n]) \neq a(x[n])$. We precompensate for this difference by first producing $p(x[n])$, where $d(p(\cdot)) = a(\cdot)$. So, when $p(x[n])$ is passed through the DAC, the output is $d(p(x[n])) = a(x[n]) = x(t)$, the desired output.

It should be obvious that predistorter design heavily relies on modeling. Unfortunately, DAC modeling is difficult in general [6]. The nonlinear behavior is caused by many different imperfections present in the DAC circuit, and this prevents the existence of a comprehensive model. A natural way to overcome this modeling problem is to use models of some of the most dominant nonlinear errors. In this project, we use a model developed for *dynamic DAC errors* which are described in Chapter 3 [9]. We try to understand how this DAC model causes nonlinearity.

1.3 Contribution

The radar systems we are trying to improve have the capability to send different kinds of signals. In terms of detection performance, however, it is *not known* whether a particular type of signal is better than another. Our main contribution in this

project is to find out whether sending a specific signal results in better detection. This requires us to do a comparison study between these signals. The search for the performance differences between the available signals also gives us signs about the optimal signal to send, independent of its availability. We provide waveform optimization results under several scenarios with varying complexity levels.

For the second project, our contribution is in terms of theoretical and simulation analyses of the nonlinearity caused by certain types of DAC errors. These type of errors are input-dependent and we use a model previously developed to describe them [9]. However, how the dependence on the input affects nonlinearity is not immediately obvious from a given input-output function provided by this DAC model. In this thesis, we present nonlinearity analyses for different functions relating the input to the error. We show that the nonlinearity observed at the output is significantly dependent on how the DAC error is modeled. This provides insight to what kind of output distortion one should expect for different definitions of error functions. This insight should in turn help the design of a predistorter aimed to eliminate those analyzed nonlinearities.

CHAPTER 2

RADAR WAVEFORM DESIGN FOR TARGET CLASSIFICATION

In this project, our objective is to find the optimal discrimination pulse for car radars to distinguish between two families of targets under the constraints such as the modulation type, energy, etc.

As with every engineering problem, we first need a model of the physical problem. There are different concerns to be accounted for in the model design. These can be listed as:

1. Constraints on the pulse generated by the car radar.
2. Imperfect observation of the reflected signal due to effects such as noise, fading, etc.
3. Description of the target classes.
4. Definition of the detection performance measure.

We follow a path that starts with the simplest assumptions and gradually gets more complicated. Starting with a simple performance measure, we compare the detection performance of different signals generated with methods such as amplitude modulation, phase modulation, etc. Our findings are given in Section 2.1.

After that, we add the noise and fading affects to our model. The reflected signals now have unknown parameters. This also changes our performance measure. We present our work in this new model in Section 2.2.

Finally, we take into account the fact that the targets we are trying to classify cannot be represented by single reflected signals. Our classification problem is indeed between two families of targets rather than two distinct targets. The work for classification between families of targets is given in Section 2.3. We end the chapter with a short summary presented in Section 2.4.

2.1 A Simple Performance Measure

First we try to find out what the optimal pulse would be if we had no constraints in design. In order to find this theoretical optimal pulse, we work assuming two *distinct* targets having given impulse responses. Let $h_1(t)$ and $h_2(t)$ be the impulse responses of the two targets we try to identify. Let $f(t)$ be the pulse we try to optimize for the best identification. We start with the idea that we achieve the best identification when the difference between the responses of these two targets is maximized. So, the quantity we are trying to maximize (call K) can be given as:

$$K = \int_{-\infty}^{\infty} (h_1(t) * f(t) - h_2(t) * f(t))^2 dt. \tag{2.1}$$

Using Parseval's Theroem, K can be rewritten as:

$$\begin{aligned} K &= \int_{-\infty}^{\infty} |H_1(f)F(f) - H_2(f)F(f)|^2 df \\ &= \int_{-\infty}^{\infty} |H_1(f) - H_2(f)|^2 |F(f)|^2 df. \end{aligned} \tag{2.2}$$

Because we have restrictions on our pulse, we cannot (in general) produce the optimal pulse that will maximize K. What we should do is to try to maximize K under our constraints, i.e. approximate the optimal pulse as much as possible. The methods of signal generation can be various such as amplitude modulation, phase modulation, etc.

2.1.1 Unquantized Amplitudes

We first assume our pulse to be a sequence of signals ($s(t)$) weighted with some real coefficients (c_k). In general, we assume $f(t)$ to be expressed as:

$$f(t) = \sum_{k=0}^{N-1} c_k s(t - kT_s), \tag{2.3}$$

where T_s is the signal duration. The Fourier transform of this signal and its magnitude can be written as:

$$\begin{aligned} F(f) &= \sum_{k=0}^{N-1} S(f) c_k e^{-j2\pi f k T_s}, \\ |F(f)| &= |S(f)||\sum_{k=0}^{N-1} c_k e^{-j2\pi f k T_s}|. \end{aligned} \tag{2.4}$$

Using the Euler identity, $|F(f)|^2$ can be written as:

$$\begin{aligned} |F(f)|^2 &= |S(f)|^2 \left((\sum_{k=0}^{N-1} c_k \cos 2\pi f k T_s)^2 + (\sum_{k=0}^{N-1} c_k \sin 2\pi f k T_s)^2 \right) \\ &= |S(f)|^2 \left(\sum_{k=0}^{N-1} c_k^2 + \sum_{m=0}^{N-1} \sum_{n \neq m, n=0}^{N-1} 2 c_m c_n \cos 2\pi (m-n) f T_s \right). \end{aligned} \tag{2.5}$$

Then

$$\begin{aligned} K = \int_{-\infty}^{\infty} |H_1(f) - H_2(f)|^2 |S(f)|^2 (\sum_{k=0}^{N-1} c_k^2 + \\ \sum_{m=0}^{N-1} \sum_{n \neq m, n=0}^{N-1} 2 c_m c_n \cos 2\pi (m-n) f T_s) df, \end{aligned} \tag{2.6}$$

where $|H_1(f) - H_2(f)|$ and $|S(f)|$ are assumed given (In our calculations, we use $s(t)$ to be a rectangular function of duration T_s, and we try several $|H_1(f) - H_2(f)|$'s). We

try to maximize K with respect to c_k's. When we equate the derivative with respect to any coefficient c_k to zero, we obtain the following expression:

$$\begin{aligned}\frac{\partial}{\partial c_k}K &= \int_{-\infty}^{\infty}|H_1(f)-H_2(f)|^2|S(f)|^2\left(2c_k+\sum_{m=0}^{N-1}2c_kc_m\cos 2\pi(k-m)fT_s\right)df \\ &= 0. \end{aligned} \tag{2.7}$$

So, we have N equations with N unknowns. When we write these equations in matrix form, we get

$$\mathbf{Bc}=0, B_{ij}=\int_{-\infty}^{\infty}|H_1(f)-H_2(f)|^2|S(f)|^2\cos 2\pi(i-j)fT_s, \tag{2.8}$$

where $\mathbf{B}$ is an $N\times N$ matrix and $\mathbf{c}$ is the $N\times 1$ vector of c_k's. In order to have nonzero solutions, $\mathbf{B}$ should be singular. However, we realized that this was not the case in general. Hence, the only solution is $\mathbf{c}=\mathbf{0}$, which minimizes K rather than maximizing it. So, we need to put the energy constraint into the calculation. The constraint we choose is $\sum_{k=0}^{N-1}c_k^2=1$. We use the Lagrange Multipliers method to optimize $\mathbf{c}$ under this constraint. So, a new variable λ is included and the new equations are obtained by taking the derivative

$$\frac{\partial}{\partial c_k}(K+\lambda(1-\sum_{k=0}^{N-1}c_k^2))=0 \tag{2.9}$$

which produces N nonlinear equations. The equation $\sum_{k=0}^{N-1}c_k^2=1$ is the $(N+1)^{st}$ equation. So, we now have $N+1$ nonlinear equations with $N+1$ unknowns. Using MATLAB we are now able to solve these equations and find a nonzero solution for the optimum $\mathbf{c}$ vector.

2.1.2 Two Phases

In the previous section, the optimum coefficients found can be any real number, which is not possible in practice. In this section, we allow our signals to have fixed

amplitude and one of two possible phases. In addition, we allow "no signal" for a particular signal duration. As a result, we are constrained to use $c_k \in \{-1, 0, 1\}$. So, what we have to do is to calculate K for all the 3^N permutations of $\mathbf{c}$ and pick the one that corresponds to the largest K.

We can make a frequency interpretation at this point which will give us an idea about what the optimal pulse would look like. Note that the theoretical optimal waveform is the one that has nonzero frequency content constrained in the band where $|H_1(f) - H_2(f)|$ is also nonzero. This can be seen from Eq. 2.2. So, the optimal waveform we find should be the one which is concentrated most in this band compared to the other $3^N - 1$ alternatives.

We present our simulation results later in the section, however, two points should be mentioned here.

1. Since K is a function of the magnitude of the Fourier transform of the pulse, we always have two optimal $\mathbf{c}$ vectors, which differ by a minus sign.

2. For all the cases where $|H_1(f) - H_2(f)|$ is lowpass, i.e. whenever it can be defined as

$$\begin{aligned} |H_1(f) - H_2(f)| &\neq 0, \qquad |f| \leq f_0 \\ |H_1(f) - H_2(f)| &= 0, \qquad \text{else,} \end{aligned} \tag{2.10}$$

   the optimum coefficients are $\mathbf{c}_{opt} = [1, 1, 1, 1, 1, 1, 1]^T$ (and also $-\mathbf{c}_{opt}$ as mentioned). This is due to the fact that if $|H_1(f) - H_2(f)|$ has this property, the coefficients of all the $c_m c_n$ terms ($2\int_{-\infty}^{\infty} |H_1(f) - H_2(f)|^2 |S(f)|^2 cos(2\pi(m-n)fT_s)$) in Eq. 2.6 become positive. This requires all $c_m c_n$ products to be maximum. More intuitively, the reason for this second observation is that $|F(f)|$ acquires the largest frequency content around DC only when all c_k's are 1 (or -1).

2.1.3 Continuous Phase

In this section, we analyze the case where each signal in the pulse has a fixed amplitude and can have any phase in $[0, 2\pi)$. Under these assumptions, the lowpass equivalent of our pulse becomes

$$f(t) = \sum_{k=0}^{N-1} e^{-j\theta_k} s(t - kT_s),\ \theta_k \in [0, 2\pi). \tag{2.11}$$

Let us start analyzing the continuous phase with the $N = 2$ case. When $N = 2$, our pulse becomes

$$f(t) = e^{-j\theta_1} s(t) + e^{-j\theta_2} s(t - T_s)$$

with the Fourier Transform

$$\begin{aligned} F(f) &= S(f)(e^{-j\theta_1} + e^{-j\theta_2} e^{-j2\pi f T_s}) \\ |F(f)| &= |S(f)||\cos\theta_1 + \cos(\theta_2 + 2\pi f T_s) - j(\sin\theta_1 + \sin(\theta_2 + 2\pi f T_s))|. \end{aligned}$$

We need the magnitude squared which is

$$\begin{aligned} |F(f)|^2 &= |S(f)|^2((\cos\theta_1 + \cos(\theta_2 + 2\pi f T_s))^2 + (\sin\theta_1 + \sin(\theta_2 + 2\pi f T_s))^2) \\ &= |S(f)|^2(2 + 2\cos(\theta_1 - \theta_2 - 2\pi f T_s)). \end{aligned} \tag{2.12}$$

Now K becomes

$$K = \int_{-\infty}^{\infty} |H_1(f) - H_2(f)|^2 |S(f)|^2 (2 + 2\cos(\theta_1 - \theta_2 - 2\pi f T_s)) df.$$

Calling $|H_1(f) - H_2(f)|^2 |S(f)|^2 = A(f)$, where $A(f)$ is a positive valued even function,

$$K = 2\int_{-\infty}^{\infty} A(f) df + 2\int_{-\infty}^{\infty} A(f)\cos(\theta_1 - \theta_2 - 2\pi f T_s) df. \tag{2.13}$$

The first term in the summation above is a constant. What we are trying to maximize is the second term. The first observation is that K is a function of $\theta_1 - \theta_2$. This means that we will try to find the optimum phase difference, and any two phases with a difference of this optimum value will maximize K. Now let $(\theta_1 - \theta_2)$ be ϕ. Dropping the 2, the second term of this summation is then

$$\begin{aligned} C &= \int_{-\infty}^{\infty} A(f)\cos(\phi - 2\pi f T_s)df \\ &= \int_{-\infty}^{\infty} A(f)\cos(2\pi f T_s)df\cos(\phi) + \int_{-\infty}^{\infty} A(f)\sin(2\pi f T_s)df\sin(\phi) \\ &= C_1\cos(\phi) + C_2\sin(\phi). \end{aligned} \tag{2.14}$$

Since $A(f)$ is an even function of f, $C_2 = 0$ and we are left with

$$C = C_1\cos(\phi)$$

to maximize. Obviously if C_1 is positive, $\phi = 0$ maximizes C and, if C_1 is negative, $\phi = \pi$ is the answer. So, any two phases such that $\theta_1 - \theta_2 = 0$ or π will maximize K, depending on what $A(f)$ is.

When we generalize this case to N signals, Eq. 2.12 becomes

$$|F(f)|^2 = |S(f)|^2(N + 2\sum_{m=0}^{N-1}\sum_{n\neq m,n=0}^{N-1}\cos(\theta_m - \theta_n - 2(n-m)\pi f T_s)). \tag{2.15}$$

So, for the current case, K is given as:

$$\begin{aligned} K &= N\int_{-\infty}^{\infty} A(f)df \\ &+ \int_{-\infty}^{\infty} A(f)\sum_{m=0}^{N-1}\sum_{n\neq m,n=0}^{N-1}\cos(\theta_m - \theta_n - 2(n-m)\pi f T_s)df. \end{aligned} \tag{2.16}$$

Again the second part of K is the one to be maximized. Let us rewrite this part using a trigonometric identity.

$$\begin{aligned} C &= \int_{-\infty}^{\infty} A(f) \sum_{m=0}^{N-1} \sum_{n\neq m, n=0}^{N-1} \cos(\theta_m - \theta_n) \cos(2(n-m)\pi f T_s) df \\ &+ \int_{-\infty}^{\infty} A(f) \sum_{m=0}^{N-1} \sum_{n\neq m, n=0}^{N-1} \sin(\theta_m - \theta_n) \sin(2(n-m)\pi f T_s) df. \end{aligned}$$

As in the previous case, the second term in this summation is zero since $A(f)$ is even. So, C is in fact

$$C = \int_{-\infty}^{\infty} A(f) \sum_{m=0}^{N-1} \sum_{n\neq m, n=0}^{N-1} \cos(\theta_m - \theta_n) \cos(2(n-m)\pi f T_s) df. \tag{2.17}$$

Now, it is a good idea to collect the terms with the same $(n-m)$'s. By doing this, C can be rewritten as:

$$\begin{aligned} C &= \sum_{i=1}^{N-1} \int_{-\infty}^{\infty} A(f) \cos(2i\pi f T_s) df \sum_{n=1}^{N-i} cos(\theta_n - \theta_{n+i}) \\ &= \sum_{i=1}^{N-1} C_i \sum_{n=1}^{N-i} cos(\theta_n - \theta_{n+i}). \end{aligned} \tag{2.18}$$

It can be confirmed that this expression fits the one we found for $N = 2$. Again we call phase differences ϕ_i's, so we have $(N-1)$ ϕ values to solve for. As N gets larger, C becomes more complicated. Unlike the $N = 2$ case, the dependence of the optimum ϕ_i values on the C_i values is highly increased. The optimum values for given C_i's are easily found by simulation; however, it is difficult to come up with an analytical solution.

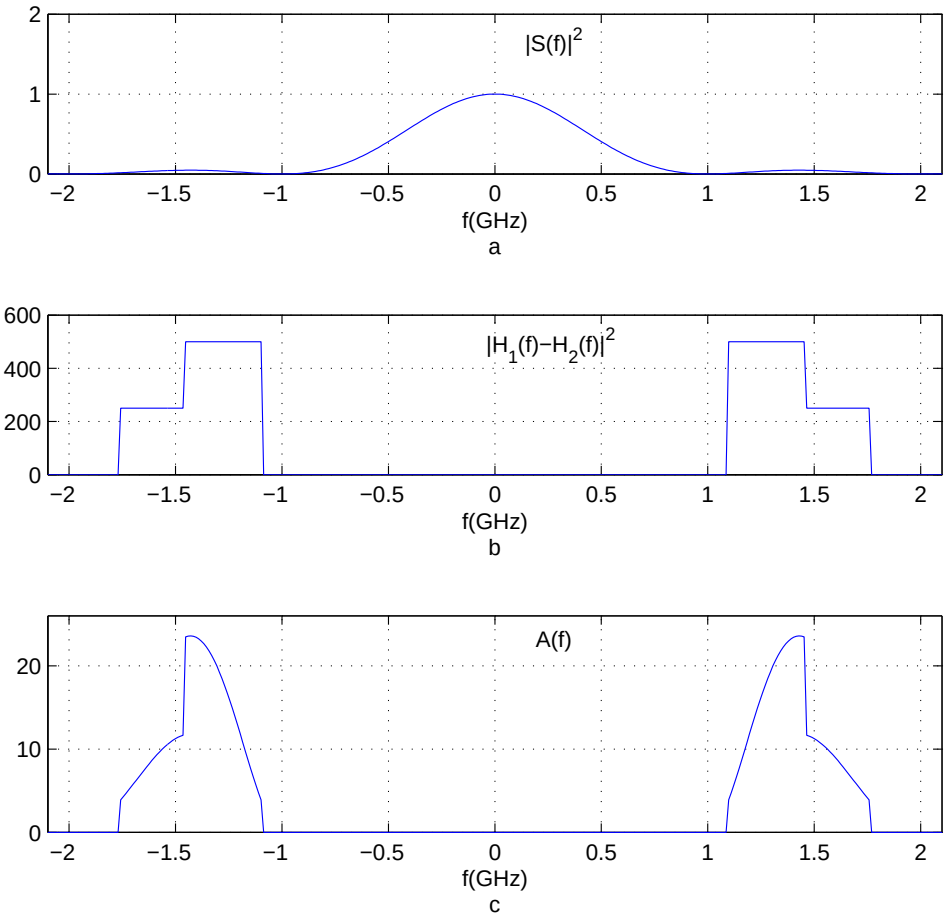

Figure 2.1. The functions used for the first numerical example.

2.1.3.1 Numerical Example 1

As an illustrative example, we find the optimum ϕ_i values for the $N = 7$ case. For this purpose, first we need to assume some functions for $|H_1(f) - H_2(f)|^2$ and $|S(f)|^2$. We assume the signals in Figure 2.1 to be the case for our simulation.

For $N = 7$, C in Eq. 2.18 can be expressed as:

$$C = \sum_{i=1}^{6} C_i \sum_{n=1}^{7-i} cos(\theta_n - \theta_{n+i}).$$

In order to gain more insight, let us write C in the open form.

$$\begin{aligned} C &= C_1\left(\cos(\theta_1-\theta_2)+\cos(\theta_2-\theta_3)+\ldots+\cos(\theta_6-\theta_7)\right) \\ &+ C_2\left(\cos(\theta_1-\theta_3)+\cos(\theta_2-\theta_4)+\ldots\cos(\theta_5-\theta_7)\right)+... \\ &+ C_6\cos(\theta_1-\theta_7). \end{aligned} \tag{2.19}$$

As for $N = 2$ case, if we assign

$$\phi_i = \theta_i - \theta_{i+1}, \tag{2.20}$$

C becomes

$$\begin{aligned} C &= C_1(\cos(\phi_1)+\cos(\phi_2)+\ldots+\cos(\phi_6)) \\ &+ C_2(\cos(\phi_1+\phi_2)+\cos(\phi_2+\phi_3)+\ldots+\cos(\phi_5+\phi_6))\ldots \\ \ldots &+ C_6\cos(\phi_1+\phi_2+\phi_3+\phi_4+\phi_5+\phi_6). \end{aligned} \tag{2.21}$$

As we see, we have six variables to optimize in this case. Any seven phase values having differences according to the optimum ϕ values will maximize C and hence K. Also note that C_i's are given by

$$C_i = \int_{-\infty}^{\infty} A(f)\cos(2i\pi f T_s)df. \tag{2.22}$$

With the functions given in Fig. 2.1, the C_i values are found to be

$$C_1 = -8.0405, C_2 = -1.5488, C_3 = 1.198,$$
$$C_4 = -0.83449, C_5 = 0.60015, C_6 = -0.54565.$$

After finding the C_i values for the given $A(f)$ function, our task is to calculate C for all possible ϕ combinations. Of course, when simulating, we cannot realize continuous

phase. We have to assign ϕ_i values from a finite set of phases. If we let each ϕ_i to take values from M discrete phases, the number of possible ϕ combinations is M^6.

In our calculations, we try M values ranging from 2 to 16. For each M value, ϕ_i's take values from the set $A_M = \{0, \frac{2\pi}{M}, \frac{4\pi}{M}, ..., \frac{2(M-1)\pi}{M}\}$. We evaluate C for each possible 6-permutation of these values. The ϕ vector corresponding to the largest C gives us the optimum phase differences. After finding ϕ_{opt}, we calculate K for these optimal values. This value, K_{max}, is the maximum possible K that can be achieved using a finite phase set of size M. The values are given in Table 2.1. In addition to the table, the graph depicting the dB loss in K with decreasing M values is given in Fig. 2.2. The values on the graph are found by dividing the K value for the given M to the K value for $M = 16$, namely

$$\Delta_K(M) = 10 \log \frac{K(M)}{K(16)}. \tag{2.23}$$

2.1.3.2 Comments on the Results

First of all, it should be noted that the ϕ_{opt} values in Table 1 are not unique. In some cases, there are more than one set that give the same K_{max}. We just include one of them in the table. Secondly, it should be kept in mind that, these are the optimum values for ϕ_i's, not θ_i's. Any θ_i combination satisfying the optimum ϕ_i values will maximize K.

When we look at our numerical results, the first observation is that no significant increase in K_{max} is achieved by increasing the number of phases. A second point is, for some cases, increasing M even decreases K_{max}. The change from $M = 2$ to $M = 3$ is an example. It should be noted that the set of possible ϕ values for $M = 2$ is not a subset of the set for $M = 3$. So, ϕ_{opt} for $M = 2$ is not available for the case with three distinct phases (and for any case where M is odd). Hence, increasing M does not necessarily improve identification.

Table 2.1. Optimal phase differences(ϕ_{opt}) and maximum K values that are achieved using M discrete ϕ values.

M	ϕ_{opt}	K_{max}	M	ϕ_{opt}	K_{max}
2	$[\pi, \pi, \pi, 0, \pi, \pi]$	194.07	10	$[\pi, \frac{4\pi}{5}, \frac{3\pi}{5}, \frac{4\pi}{5}, \frac{4\pi}{5}, \frac{4\pi}{5}]$	203.4
3	$[\frac{2\pi}{3}, \frac{2\pi}{3}, \frac{2\pi}{3}, \frac{2\pi}{3}, \frac{2\pi}{3}, \frac{2\pi}{3}]$	185.34	11	$[\frac{10\pi}{11}, \frac{8\pi}{11}, \frac{8\pi}{11}, \frac{8\pi}{11}, \frac{8\pi}{11}, \frac{10\pi}{11}]$	204.85
4	$[\pi, \pi, \pi, 0, \pi, \pi]$	194.07	12	$[\frac{7\pi}{6}, \frac{7\pi}{6}, \frac{4\pi}{3}, \frac{4\pi}{3}, \frac{7\pi}{6}, \frac{7\pi}{6}]$	204.13
5	$[\frac{4\pi}{5}, \frac{4\pi}{5}, \frac{4\pi}{5}, \frac{4\pi}{5}, \frac{4\pi}{5}, \frac{4\pi}{5}]$	203.12	13	$[\frac{14\pi}{13}, \frac{16\pi}{13}, \frac{18\pi}{13}, \frac{16\pi}{13}, \frac{16\pi}{13}, \frac{14\pi}{13}]$	205.17
6	$[\pi, \pi, \frac{2\pi}{3}, \frac{2\pi}{3}, \frac{2\pi}{3}, \pi]$	200.08	14	$[\frac{8\pi}{7}, \frac{8\pi}{7}, \frac{9\pi}{7}, \frac{9\pi}{7}, \frac{9\pi}{7}, \frac{8\pi}{7}]$	204.96
7	$[\frac{6\pi}{7}, \frac{6\pi}{7}, \frac{6\pi}{7}, \frac{4\pi}{7}, \frac{6\pi}{7}, \frac{6\pi}{7}]$	203.01	15	$[\frac{14\pi}{15}, \frac{4\pi}{5}, \frac{2\pi}{3}, \frac{2\pi}{3}, \frac{4\pi}{5}, \frac{14\pi}{15}]$	205.13
8	$[\pi, \frac{3\pi}{4}, \frac{3\pi}{4}, \frac{3\pi}{4}, \frac{3\pi}{4}, \pi]$	203.73	16	$[\frac{7\pi}{8}, \frac{3\pi}{4}, \frac{3\pi}{4}, \frac{3\pi}{4}, \frac{3\pi}{4}, \frac{7\pi}{8}]$	205.35
9	$[\frac{8\pi}{9}, \frac{8\pi}{9}, \frac{2\pi}{3}, \frac{2\pi}{3}, \frac{8\pi}{9}, \frac{8\pi}{9}]$	204.44			

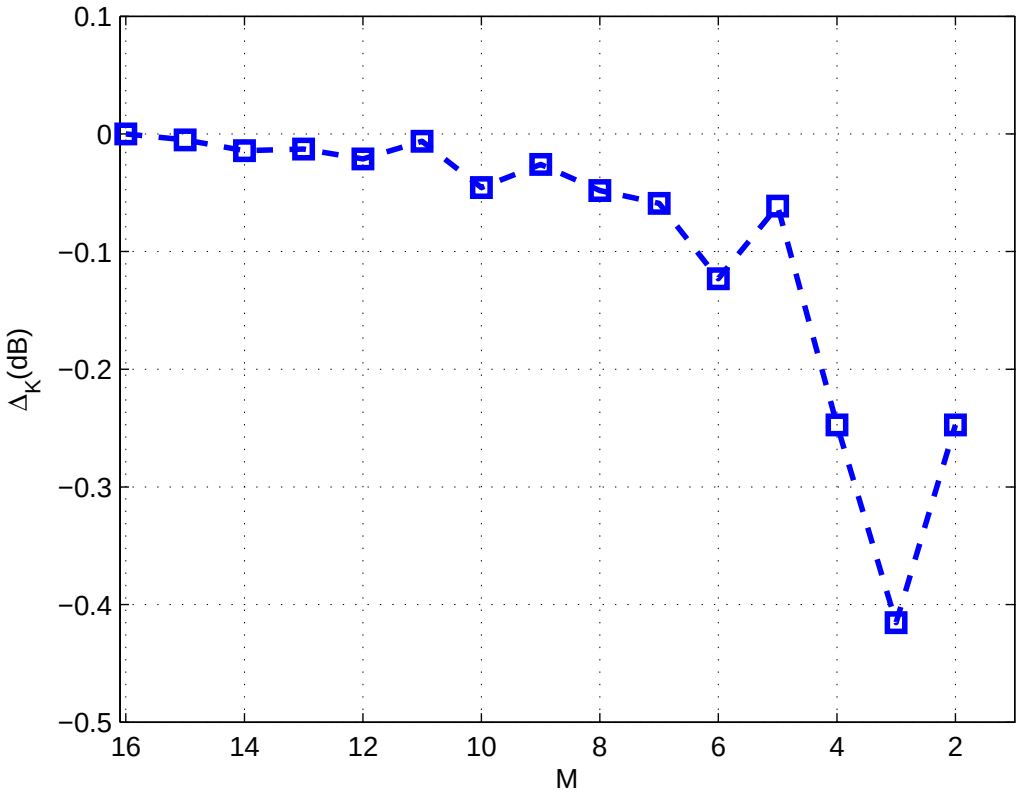

Figure 2.2. Decrease in K (Δ_K) with decreasing number of distinct phases.

When we compare the cases $M = 2, 4, 8, 16$, we see that K_{max} values are the same for the first two. In other words, adding $\frac{\pi}{2}$ and $\frac{3\pi}{2}$ to the set does not change the solution. However, for $M = 8$, we are able to obtain a higher K value. So, adding the new values, $\frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}$ and $\frac{7\pi}{4}$, to our set helps us to find a better solution. This is confirmed by the fact that ϕ_{opt} includes $\frac{3\pi}{4}$ for $M = 8$. A similar observation can also be made for $M = 16$.

As a final remark, it is important to see that, these results are completely determined by the C_i values. So, if the assumed functions are changed, only the C_i values have to be recalculated and the rest of the procedure is the same. As an interesting side note, when all C_i's are positive, the solution is trivial and all ϕ_i's equal to zero maximizes C (see Eq. 2.21).

Based on these results, let us try to confirm the frequency interpretation. In Section 2.1, we had given our performance measure to be

$$K = \int_{-\infty}^{\infty} |H_1(f) - H_2(f)|^2 |F(f)|^2 df. \tag{2.24}$$

This result suggests that the optimal pulse is the one that has its frequency content bounded to the nonzero region of $|H_1(f) - H_2(f)|$. This also means that the optimal pulses we find must have the highest frequency content in this region compared to the non-optimal alternatives. In order to check our results, let us compare two pulses in terms of their frequency contents.

For $M = 16$, the optimal ϕ combination is found to be (see Table 2.1)

$$\phi_{opt} = [\frac{7\pi}{8}, \frac{3\pi}{4}, \frac{3\pi}{4}, \frac{3\pi}{4}, \frac{3\pi}{4}, \frac{7\pi}{8}].$$

With these ϕ values, an optimal θ combination can be selected as:

$$\theta_{opt} = [\frac{19\pi}{4}, \frac{31\pi}{8}, \frac{25\pi}{8}, \frac{19\pi}{8}, \frac{13\pi}{8}, \frac{7\pi}{8}, 0].$$

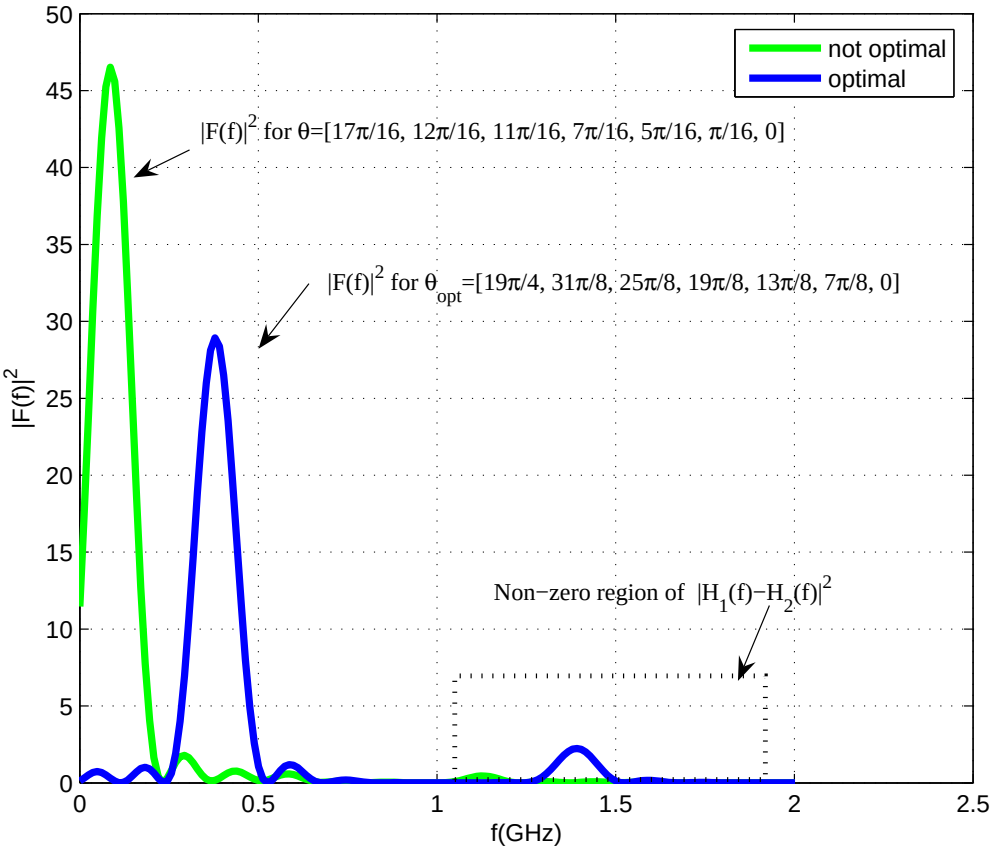

Figure 2.3. $|F(f)|^2$ plotted for two different phase values.

These θ values are sufficient to write the optimal pulse using Eq. 2.11. For comparison, let us choose another θ combination *arbitrarily* to be

$$\theta = [\frac{17\pi}{16}, \frac{12\pi}{16}, \frac{11\pi}{16}, \frac{7\pi}{16}, \frac{5\pi}{16}, \frac{\pi}{16}, 0].$$

The Fourier transforms of these two pulses are given in Figure 2.3. The optimal pulse has obviously higher frequency content in the region where $|H_1(f) - H_2(f)|^2$ is located. It is easy to see that when multiplied with $|H_1(f) - H_2(f)|^2$ and integrated, which results in K by definition, the optimal pulse will give a higher result.

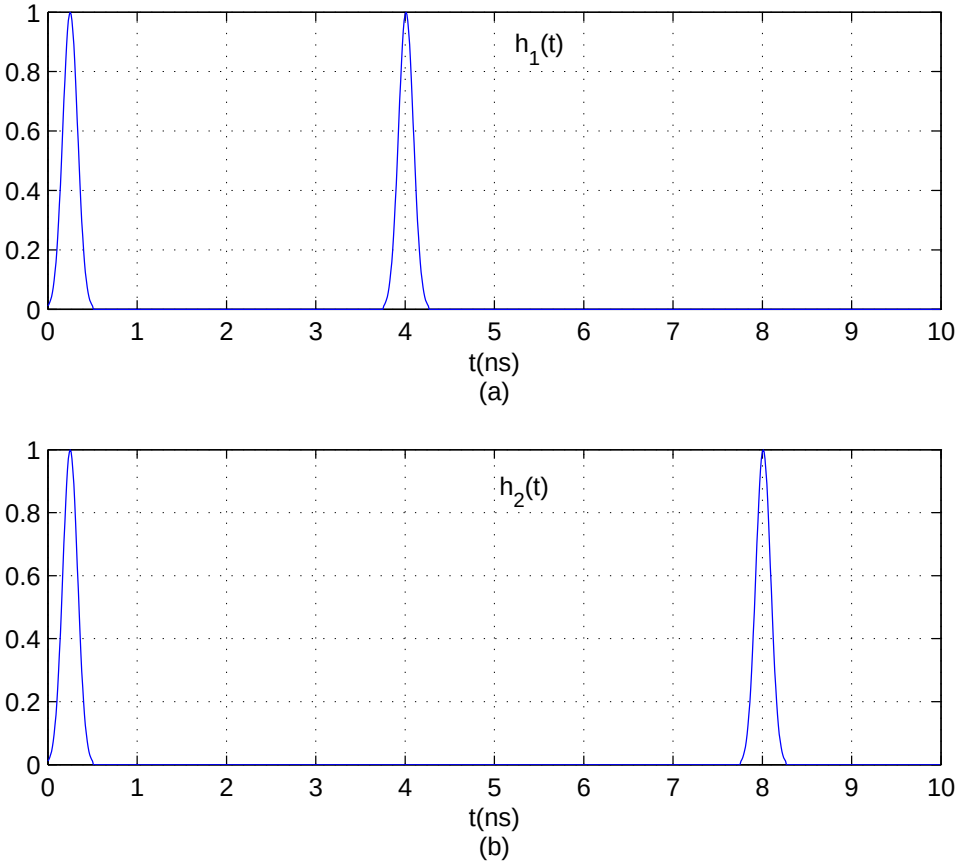

Figure 2.4. Impulse responses for the second numerical example.

2.1.3.3 Numerical Example 2

In this section, we repeat our simulations with different impulse responses. The impulse responses are given in Figure 2.4. The first impulse response has two Gaussian-like pulses located at $T_s/4$ and $4T_s$. These two pulses are located at $T_s/4$ and $8T_s$ for the second impulse response.

For these impulse responses, $|H_1(f) - H_2(f)|^2$ becomes as given in Figure 2.5. See that $A(f) = |H_1(f) - H_2(f)|^2|S(f)|^2$ changes accordingly. What we do is simply recalculate the C_i values given by

$$C_i = \int_{-\infty}^{\infty} A(f)\cos(2i\pi f T_s)df \tag{2.25}$$

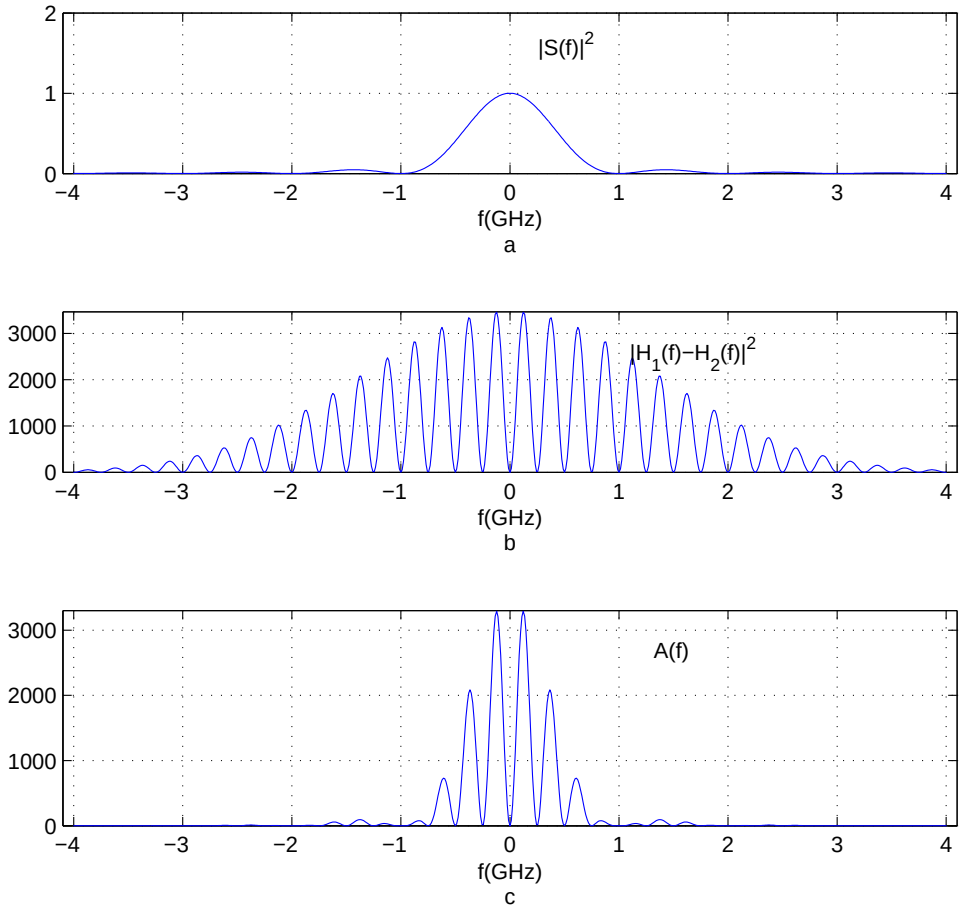

Figure 2.5. Functions used for the second numerical example

and calculate C using Eq. 2.21 for each possible ϕ combination. The C_i values for this case are found to be

$$\begin{aligned} C_1 &= 81.007, C_2 = 0.0011, C_3 = -40.523, \\ C_4 &= -787.82, C_5 = -40.523, C_6 = -0.0011. \end{aligned}$$

Again the optimal pulse is the one that achieves the maximum C value. ϕ_{opt} and K_{max} values for changing M are given in Table 2.2.

Table 2.2. Optimal phase differences(ϕ_{opt}) and maximum K values for the second experiment where the functions in Fig. 2.5 are used.

M	ϕ_{opt}	K_{max}	M	ϕ_{opt}	K_{max}
2	$[0, 0, \pi, 0, 0, 0]$	16728	10	$[\frac{\pi}{5}, \frac{\pi}{5}, \frac{2\pi}{5}, \frac{\pi}{5}, \frac{\pi}{5}, \frac{\pi}{5}]$	16815
3	$[0, 0, \frac{2\pi}{3}, 0, 0, 0]$	14244	11	$[\frac{2\pi}{11}, \frac{2\pi}{11}, \frac{4\pi}{11}, \frac{2\pi}{11}, \frac{2\pi}{11}, \frac{2\pi}{11}]$	16640
4	$[0, 0, \pi, 0, 0, 0]$	16728	12	$[\frac{\pi}{3}, \frac{\pi}{3}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{3}, \frac{\pi}{3}]$	16841
5	$[0, 0, \frac{8\pi}{5}, \frac{8\pi}{5}, 0, 0]$	15814	13	$[\frac{2\pi}{13}, \frac{2\pi}{13}, \frac{4\pi}{13}, \frac{4\pi}{13}, \frac{2\pi}{13}, \frac{2\pi}{13}]$	16713
6	$[0, 0, \pi, 0, 0, 0]$	16728	14	$[\frac{\pi}{7}, \frac{\pi}{7}, \frac{3\pi}{7}, \frac{2\pi}{7}, \frac{\pi}{7}, \frac{\pi}{7}]$	16838
7	$[0, \frac{2\pi}{7}, \frac{2\pi}{7}, \frac{2\pi}{7}, \frac{2\pi}{7}, 0]$	16344	15	$[\frac{2\pi}{15}, \frac{2\pi}{15}, \frac{2\pi}{5}, \frac{4\pi}{15}, \frac{2\pi}{15}, \frac{2\pi}{15}]$	16742
8	$[\frac{\pi}{4}, \frac{\pi}{4}, \frac{\pi}{4}, \frac{\pi}{4}, \frac{\pi}{4}, \frac{\pi}{4}]$	16787	16	$[\frac{\pi}{8}, \frac{\pi}{8}, \frac{3\pi}{8}, \frac{3\pi}{8}, \frac{\pi}{8}, \frac{\pi}{8}]$	16840
9	$[\frac{16\pi}{9}, \frac{16\pi}{9}, \frac{16\pi}{9}, \frac{16\pi}{9}, \frac{16\pi}{9}, \frac{16\pi}{9}]$	16530			

In this second example, the results are even closer to each other compared to the first numerical example. Again, increasing M did not result in a better performance.

An important comparison can be made with the case with *no modulation*, which means having all θ values equal to zero, i.e. using only a rectangular pulse of duration $7T_s$. The K value obtained in this case is 6789, which is significantly smaller than the values in Table 2.2. In fact, this value is less than half of the smallest value in Table 2.2, a 3.2 dB difference. This proves that by applying phase to our signals, we can achieve better target identification.

Having seen the K_{max} values so close to each other for different M's, we want to know what the global maximum of the K function is, i.e. the K_{max} value achieved when any phase is allowed for each signal in the pulse. To find the answer, we need to find the maximum value C in Eq. 2.21 can take. The numerically calculated maximum of C is found to be approximately 2907. Using Eq. 2.16 to calculate K, we find K_{max} for continuous phase to be 16844. This is an interesting observation as this value is very close to the values in Table 2.2. This means the values we achieve with finite number of phases are already close to the global maximum. In fact, we obtain a value very close to the global maximum with even 2 distinct phases. This also explains why we cannot achieve improvement by increasing the number of phases. Since we are already close to the limit, we cannot get higher values by increasing M.

Finally, in addition to the M phase values, we also allow the pulse to be OFF for some signal durations. In other words, we allow a zero coefficient along with the already existing M different coefficients. We investigate whether this makes any changes in terms of reaching a larger K value. Previously, we were comparing M^7 possible signals. With allowing signals to be OFF, we are now comparing $(M+1)^7$ possible signals. We observe that the K_{max} values achieved remain the same. This is somewhat expected since we are losing energy if some of the coefficients are zero. So, for a fair comparison, we rescale the amplitudes for all the combinations with

at least one zero coefficient. In this case, we have the same energy for all possible $(M+1)^7$ signals. This time the K_{max} values change. These values are achieved by a combination of coefficients including zero.

The values achieved with the OFF case are given in Table 2.3 . The simulations for this case take much longer because it is not possible to simplify the calculation of K. So, we present here the values for $M = 2, 3, 4$ and 5 only.

Table 2.3. Maximum K values achieved for the second experiment with allowing zero coefficient (OFF).

M	K_{max}
2 + OFF	17679
3 + OFF	14732
4 + OFF	17679
5 + OFF	16553

2.2 Signals with Unknown Parameters

Up to this point, we have always assumed that the responses of targets to our signal are completely known. In this part, we will consider the situation where this is not true. We will be regarding the responses as having a random phase, random amplitude and additive noise. For example, the response of the first target to our radar signal will be modeled as:

$$r_1(t) = \alpha e^{j\theta}(f(t) * h_1(t)) + n(t), \tag{2.26}$$

where α and θ are random variables and $n(t)$ is the additive noise.

It is important to state our understanding of the problem in the first place. We consider the target identification as a binary detection problem. In this problem, we try to choose between two targets each corresponding to a different signal waveform with random parameters.

In order to develop our understanding of this detection problem, we first present the general simple binary hypothesis testing, then we pass to the composite hypothesis case where unknown parameters come into play. After that, we explain the detection problem where the hypotheses correspond to waveforms. We finish with presenting our case where we decide between waveforms with unknown parameters.

2.2.1 Binary Hypothesis Testing

In the general problem of detection, we try to decide between choices which are referred as hypotheses. In the binary detection case, we have two hypotheses to choose from, namely H_0 and H_1. What we have to do is to develop a decision rule based on our observations.

Our observations correspond to an observation vector, $\mathbf{r}$. We develop a decision rule based on the conditional probability densities of these observations.

Here, we will not go into the detail of developing this decision rule. Assuming that the a priori probabilities of the two hypotheses, P_0 and P_1, are known, a decision based on the Bayes criterion can be made. Using the conditional probabilities of our observation given the two hypotheses, we can decide on which hypothesis is true based on the following decision rule:

$$\frac{f_{\mathbf{R}|H_1}(\mathbf{r}|H_1)}{f_{\mathbf{R}|H_0}(\mathbf{r}|H_0)} \gtrless_{H_0}^{H_1} \frac{P_0(C_{10}-C_{00})}{P_1(C_{01}-C_{11})}, \tag{2.27}$$

where C_{ij} is the cost of claiming H_i under H_j. This decision rule can be simplified as:

$$\Lambda(\mathbf{r}) \gtrless_{H_0}^{H_1} \eta. \tag{2.28}$$

$\Lambda(\mathbf{r})$ is the likelihood ratio and η is the threshold. So, the binary detection problem turns into a *likelihood ratio test*. For our present discussions, we will try to construct the likelihood ratio and the threshold will be considered as given. In fact, as can be seen, only $\Lambda(\mathbf{r})$ depends on the observations and the calculation of the threshold can be ignored in developing the decision rule from the observations. Here, it is important to see that $\Lambda(\mathbf{r})$ is one-dimensional regardless of what the dimension of $\mathbf{r}$ is. The likelihood ratio test (LRT) is the basics of detection problems in general.

2.2.1.1 Composite Hypothesis

A composite hypothesis is one, for which the observation depends on a random variable that we denote θ. As stated above, in any case, we try to write the likelihood ratio $\Lambda(\mathbf{r})$. For the case in which we know the probability density function of θ, we can write the conditional probabilities using this density. For composite hypotheses, the likelihood ratio is obtained as:

$$\Lambda(\mathbf{r}) = \frac{f_{\mathbf{R}|H_1}(\mathbf{r}|H_1)}{f_{\mathbf{R}|H_0}(\mathbf{r}|H_0)} = \frac{\int_\theta f_{\mathbf{R}|\theta,H_1}(\mathbf{r}|\theta,H_1) f_\Theta(\theta) d\theta}{\int_\theta f_{\mathbf{R}|\theta,H_0}(\mathbf{r}|\theta,H_0) f_\Theta(\theta) d\theta}. \tag{2.29}$$

Once this likelihood ratio is found, the LRT is completely defined and we are back to the case in the previous section.

2.2.1.2 Detection of Signals

In the previous sections, we considered the observations as finite random vectors, $\mathbf{r}$. In the detection of signals, the observations are continuous random waveforms, $r(t)$.

In the simplest binary signal detection, the received waveforms under the hypotheses can be written as:

$$\begin{aligned} H_0 &: r(t) = s_0(t) + n(t), \\ H_1 &: r(t) = s_1(t) + n(t), \end{aligned} \tag{2.30}$$

where $n(t)$ is the additive noise. Here, our aim is again to write the likelihood ratio. However, since the observations are waveforms, the extra step here is to reduce the observations into a set of random variables. We will achieve this using basis functions to represent the continuous waveforms. Once we do this, the problem will turn into the same one in composite hypothesis case.

2.2.1.3 Detection of Signals with Unknown Parameters

Here, the observations are continuous waveforms depending on parameters which are random variables. For this case, the received waveforms under the hypotheses can be written as:

$$H_0 : r(t) = s_0(t, \theta) + n(t),$$
$$H_1 : r(t) = s_1(t, \theta) + n(t), \tag{2.31}$$

where θ is a random variable with known probability density. It is obvious that this case is the continuous counterpart of the composite hypothesis case. Similar to that case, what we have to do is to find the conditional probability densities assuming θ is given. This task is the same for the previous section; it requires reduction of the continuous waveform to a set of random variables. After that, using the probability density of θ, we will try to write the likelihood ratio using Eq. 2.29.

2.2.2 Target Identification Problem

2.2.2.1 Problem Definition and Procedure

Having seen these binary detection problems in general, we can now identify our problem. Our problem can be regarded as the *detection of signals with unknown parameters*. This is the one covered in the previous section. For our particular problem, the received waveforms under the two hypotheses can be given as (to avoid confusion we will use the notation H_1 and H_2):

$$H_1 : r(t) = s_1(t, \alpha, \theta) = \alpha e^{j\theta}(f(t) * h_1(t)) + n(t),$$
$$H_2 : r(t) = s_2(t, \alpha, \theta) = \alpha e^{j\theta}(f(t) * h_2(t)) + n(t), \tag{2.32}$$

where α and θ are random variables with given probability densities, $h_1(t)$ and $h_2(t)$ are the given target impulse responses and $f(t)$ is the radar waveform we attempt to design.

Having defined our problem, we can determine the procedure to follow.

i. Assuming α and θ are given, reduce the observed waveform $r(t)$ to a random vector $\mathbf{r}$.

ii. Find the conditional probabilities $f_{\mathbf{R}|\Theta,\alpha,H_1}(\mathbf{r}|\theta,\alpha,H_1)$ and $f_{\mathbf{R}|\Theta,\alpha,H_2}(\mathbf{r}|\theta,\alpha,H_2)$.

iii. Using the given probability densities $f_\alpha(\alpha)$ and $f_\Theta(\theta)$, find the conditional probabilities $f_{\mathbf{R}|H_1}(\mathbf{r}|H_1)$ and $f_{\mathbf{R}|H_2}(\mathbf{r}|H_2)$.

iv. Construct the likelihood ratio $\Lambda(\mathbf{r})$ and find the decision rule.

v. Apply performance analysis on the decision rule found and extract the relation of the performance to the radar waveform $f(t)$.

vi. Using this relation, find the optimal waveform that will yield the best detection performance.

2.2.2.2 Analysis

We start with a simplification that will make our calculations easier. Since α and $e^{j\theta}$ are random variables, their product is another random variable. We call this random variable β. We assume that β is a complex-valued Gaussian with zero mean and variance σ_β^2.

We define

$$\begin{aligned} \beta &= \alpha e^{j\theta}, && (2.33)\\ u_1(t) &= f(t) * h_1(t), && (2.34)\\ u_2(t) &= f(t) * h_2(t). && (2.35) \end{aligned}$$

The received waveforms under the two hypotheses are

$$\begin{aligned} H_1 &: r(t) = \beta u_1(t) + n(t),\\ H_2 &: r(t) = \beta u_2(t) + n(t). && (2.36) \end{aligned}$$

When the signal $r(t)$ is received, the radar correlates it with $f(t)$. So, in fact what we will use is the result of this correlation operation rather than the waveform itself. Defining

$$r = \int_0^T r(t)f(t)dt, \tag{2.37}$$

the random variables we will have under the two hypotheses are

$$H_1 : r = \beta u_1 + n, \tag{2.38}$$

$$H_2 : r = \beta u_2 + n, \tag{2.39}$$

where

$$u_1 = \int_0^T u_1(t)f(t)dt, \tag{2.40}$$

$$u_2 = \int_0^T u_2(t)f(t)dt, \tag{2.41}$$

$$n = \int_0^T n(t)f(t)dt. \tag{2.42}$$

Here $r(t)$ and $n(t)$ are random waveforms, and, r and n are random variables. Assuming $n(t)$ to be white Gaussian noise, n is a Gaussian random variable. On the other hand, $u_1(t)$ and $u_2(t)$ are deterministic signals, so u_1 and u_2 are not random.

Now we can try to find the probability density function of r under the two hypotheses. Under both hypotheses, r is the sum of two independent complex-valued Gaussian random variables. So, r is also a complex-valued Gaussian random variable, which we can write as:

$$r = r_{real} + jr_{imag}. \tag{2.43}$$

Here r_{real} and r_{imag} are real-valued Gaussian random variables.

We assume that β and n are complex-valued random variables for which real and imaginary parts are independent and have the same Gaussian densities, with means zero and variances $\frac{\sigma_\beta^2}{2}$, $\frac{\sigma_n^2}{2}$, respectively.

Let us have a look at the means and the variances of r_{real} and r_{imag} under H_1.

$$E[r_{real}|H_1] = E[\beta_{real}u_1 + n_{real}] = 0, \tag{2.44}$$

$$E[r_{imag}|H_1] = E[\beta_{imag}u_1 + n_{imag}] = 0, \tag{2.45}$$

and the variances are

$$\begin{aligned} E[r_{real}^2] &= E[(\beta_{real}u_1 + n_{real})^2] \\ &= E[\beta_{real}^2]u_1^2 + 2u_1E[\beta_{real}]E[n_{real}] + E[n_{real}^2] \\ &= \frac{\sigma_\beta^2}{2}u_1^2 + \frac{\sigma_n^2}{2} \\ &\triangleq \frac{\sigma_{r_1}^2}{2}. \end{aligned} \tag{2.46}$$

Similarly

$$\begin{aligned} E[r_{imag}^2] &= E[(\beta_{imag}u_1 + n_{imag})^2] \\ &= \frac{\sigma_\beta^2}{2}u_1^2 + \frac{\sigma_n^2}{2} \\ &\triangleq \frac{\sigma_{r_1}^2}{2}. \end{aligned} \tag{2.47}$$

So, now we can write the pdf's of r_{real} and r_{imag}.

$$f_{R_{real}}(r_{real}) = f_{R_{imag}}(r_{imag}) = \frac{1}{\sqrt{\pi\sigma_{r_1}^2}}e^{\left(-\frac{r_{real}^2}{\sigma_{r_1}^2}\right)}. \tag{2.48}$$

So, *r is also a complex-valued random variable whose real and imaginary parts are independent and have the same zero-mean Gaussian distribution.*

Now, let us write the probability density function of r. It is easy since the real and imaginary parts are independent.

$$\begin{aligned} f_R(r) = f_R(r_{real} + jr_{imag}) &= f_{R_{real}}(r_{real}) f_{R_{imag}}(r_{imag}) \\ &= \frac{1}{\pi\sigma_{r_1}^2} e^{\left(-\frac{r_{real}^2}{\sigma_{r_1}^2} - \frac{r_{imag}^2}{\sigma_{r_1}^2}\right)} \\ &= \frac{1}{\pi\sigma_{r_1}^2} e^{\left(-\frac{r_{real}^2 + r_{imag}^2}{\sigma_{r_1}^2}\right)} \\ &= \frac{1}{\pi\sigma_{r_1}^2} e^{\left(-\frac{\|r\|^2}{\sigma_{r_1}^2}\right)}. \end{aligned} \tag{2.49}$$

Having the pdf's, we can write the likelihood ratio and the decision rule. If we assume that the hypotheses are equiprobable, the cost of correct decision is zero, and the cost of failures in both ways are the same, the threshold η in Eq. 28 becomes 1.

Our decision rule is

$$\Lambda(r) = \frac{f_{R|H_1}(r|H_1)}{f_{R|H_2}(r|H_2)} \gtrless_{H_2}^{H_1} 1. \tag{2.50}$$

Replacing the pdf's,

$$\frac{\frac{1}{\pi\sigma_{r_1}^2} e^{-\frac{\|r\|^2}{\sigma_{r_1}^2}}}{\frac{1}{\pi\sigma_{r_2}^2} e^{-\frac{\|r\|^2}{\sigma_{r_2}^2}}} \gtrless_{H_2}^{H_1} 1, \tag{2.51}$$

$$\frac{\sigma_{r_2}^2}{\sigma_{r_1}^2} e^{\left(\frac{\|r\|^2}{\sigma_{r_2}^2} - \frac{\|r\|^2}{\sigma_{r_1}^2}\right)} \gtrless_{H_2}^{H_1} 1, \tag{2.52}$$

$$ln(\frac{\sigma_{r_2}^2}{\sigma_{r_1}^2}) + \|r\|^2 (\frac{1}{\sigma_{r_2}^2} - \frac{1}{\sigma_{r_1}^2}) \gtrless_{H_2}^{H_1} 0. \tag{2.53}$$

Assuming $\sigma_{r_1}^2 > \sigma_{r_2}^2$,

$$\|r\|^2 \gtrless_{H_2}^{H_1} \frac{\sigma_{r_1}^2 \sigma_{r_2}^2}{\sigma_{r_1}^2 - \sigma_{r_2}^2} ln(\frac{\sigma_{r_1}^2}{\sigma_{r_2}^2}) = \gamma. \tag{2.54}$$

So, the only statistic that we use for detection, i.e. *the sufficient statistic*, is $\|r\|^2$. The next step is to find the probability of error to evaluate the performance. If we can manage to write the probability of error in terms of $f(t)$, we will have a value to minimize by selecting the optimal waveform.

Now let us try to write the probability of error, $P(e)$. Under H_1, we make an error if $\|r\|^2$ is less than the threshold γ. So, the conditional probability of error can be written as:

$$P(e|H_1) = P(\|r\|^2 < \gamma|H_1). \tag{2.55}$$

We know that r is a complex-valued Gaussian random variable. What we need to know is the distribution of $\|r\|^2$. If $\|r\|^2 = r_{real}^2 + r_{imag}^2$ is divided by $\frac{\sigma_{r_1}^2}{2}$, then it has a *chi-square distribution* with the degree of freedom equal to 2. So, we say that

$$\frac{\|r\|^2}{\sigma_{r_1}^2/2} \sim \mathcal{X}_2^2. \tag{2.56}$$

We can now write the conditional probability of error.

$$\begin{aligned} P(e|H_1) &= P(\|r\|^2 < \gamma|H_1) \\ &= P(\frac{r_{real}^2 + r_{imag}^2}{\sigma_{r_1}^2/2} < \frac{\gamma}{\sigma_{r_1}^2/2}) \\ &= F(\frac{2\gamma}{\sigma_{r_1}^2}), \end{aligned} \tag{2.57}$$

where $F(\cdot)$ is the cdf of the $\mathcal{X}_2^2$ distribution. Likewise, under H_2, the probability of error is

$$P(e|H_2) = P(\|r\|^2 > \gamma|H_2). \tag{2.58}$$

Following the same procedure,

$$\begin{aligned} P(e|H_2) &= P(\|r\|^2 > \gamma|H_2) \\ &= P(\frac{r_{real}^2 + r_{imag}^2}{\sigma_{r_2}^2/2}) > \frac{\gamma}{\sigma_{r_2}^2/2}) \\ &= 1 - F(\frac{2\gamma}{\sigma_{r_2}^2}). \end{aligned} \tag{2.59}$$

We found the conditional probability of errors. Now we can calculate the total probability of error.

$$\begin{aligned} P(e) &= \frac{1}{2}F(\frac{2\gamma}{\sigma_{r_1}^2}) + \frac{1}{2}\left(1 - F(\frac{2\gamma}{\sigma_{r_2}^2})\right) \\ &= \frac{1}{2}\left(1 + F(\frac{2\gamma}{\sigma_{r_1}^2}) - F(\frac{2\gamma}{\sigma_{r_2}^2})\right). \end{aligned} \tag{2.60}$$

Using Eq. 2.54 to replace the value of γ,

$$P(e) = \frac{1}{2}\left(1 + F\left(\frac{2\sigma_{r_2}^2}{\sigma_{r_1}^2 - \sigma_{r_2}^2} ln(\frac{\sigma_{r_1}^2}{\sigma_{r_2}^2})\right) - F\left(\frac{2\sigma_{r_1}^2}{\sigma_{r_1}^2 - \sigma_{r_2}^2} ln(\frac{\sigma_{r_1}^2}{\sigma_{r_2}^2})\right)\right). \tag{2.61}$$

So, in order to minimize $P(e)$ we should have $\sigma_{r_1}^2$ as large as possible and $\sigma_{r_2}^2$ as small as possible. In other words, we should maximize (using Eq. 2.47)

$$K = \frac{\sigma_{r_1}^2}{\sigma_{r_2}^2} = \frac{\sigma_\beta^2 u_1^2 + \sigma_n^2}{\sigma_\beta^2 u_2^2 + \sigma_n^2}. \tag{2.62}$$

This is expected as under both hypotheses r is a Gaussian random variable with the same mean. The only parameter that distinguishes the two hypotheses are the variances. So, it would be better to have these as far apart as possible. This is more easily seen if we write the probability of error in terms of K (plotted in Fig. 2.7).

$$P(e) = \frac{1}{2}\left(1 + F\left(\frac{2}{K-1} ln(K)\right) - F\left(\frac{2K}{K-1} ln(K)\right)\right). \tag{2.63}$$

Finally, replacing the values of u_1 and u_2,

$$K = \frac{\sigma_\beta^2 \left(\int_0^T (f(t) * h_1(t)) f(t) dt\right)^2 + \sigma_n^2}{\sigma_\beta^2 \left(\int_0^T (f(t) * h_2(t)) f(t) dt\right)^2 + \sigma_n^2}. \tag{2.64}$$

So, we managed to figure out how to find the optimal $f(t)$ to minimize probability of error. The optimal waveform is defined as:

$$f_{opt}(t) = \underset{f(t)}{\operatorname{argmax}}\, K. \tag{2.65}$$

The next step is to find the way to come up with $f_{opt}(t)$ that maximizes K, given $h_1(t)$, $h_2(t)$, σ_β^2 and σ_n^2.

2.2.2.3 Discretization

It turns out that K with the continuous form in Eq. 2.64 is hard to maximize. It is a functional of $f(t)$, and is not easily worked on. To come up with an easier model,

we decide to maximize the discretized version of it. The new definition we will work on is

$$K = \frac{\sigma_\beta^2 \left(\sum_{n=0}^{N-1} f[n] \sum_{i=0}^{N-1} f[i] h_1[n-i] \right)^2 + \sigma_n^2}{\sigma_\beta^2 \left(\sum_{n=0}^{N-1} f[n] \sum_{i=0}^{N-1} f[i] h_2[n-i] \right)^2 + \sigma_n^2}. \tag{2.66}$$

Here we think of K as a function of the sample values of $f[n]$ and we try to find the optimum $f[0], f[1], ..., f[N-1]$ values that maximize K.

The maximization of K with respect to these values are done with a computer simulation. We have to assume discrete $h_1[n]$ and $h_2[n]$ to do the simulation. The impulse responses assumed are given in Fig. 2.6. We also need to assume some values for σ_β^2 and σ_n^2 in order to be completely able to write K. We assume them to be both 1.

We find the optimum $f[n]$ sequences that maximize K, with changing energy values. In Fig. 2.8 and Table 2.4 we give the calculated sequences with energies equal to $1, 10, 20$ and 50.

As can be seen, the maximum K values achieved are not so big and they somehow converge to a limit. Even worse, the corresponding probability of errors are too large for these K values. However, this also shows how large probability of errors would be obtained when a non-optimal sequence is used. To see this, we also generate random sequences and find their probability of errors with theoretical analysis and simulation (with energy equal to 20). The results are given in Fig. 2.9. It can be seen how bad probability of errors we get with randomly generated sequences. This shows the difficulty of the problem of detection between same-mean, different-variance distributions. In Fig. 2.10 the corresponding K values for the same sequences are given.

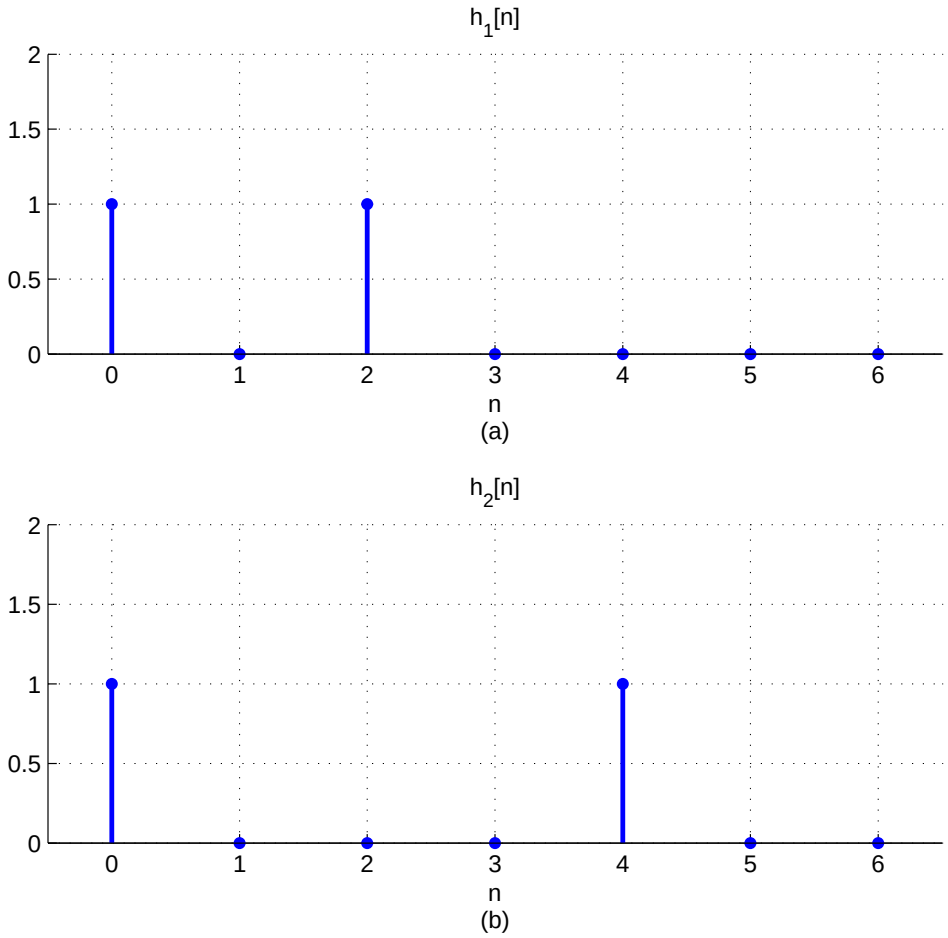

Figure 2.6. Discrete impulse responses used for the simulation

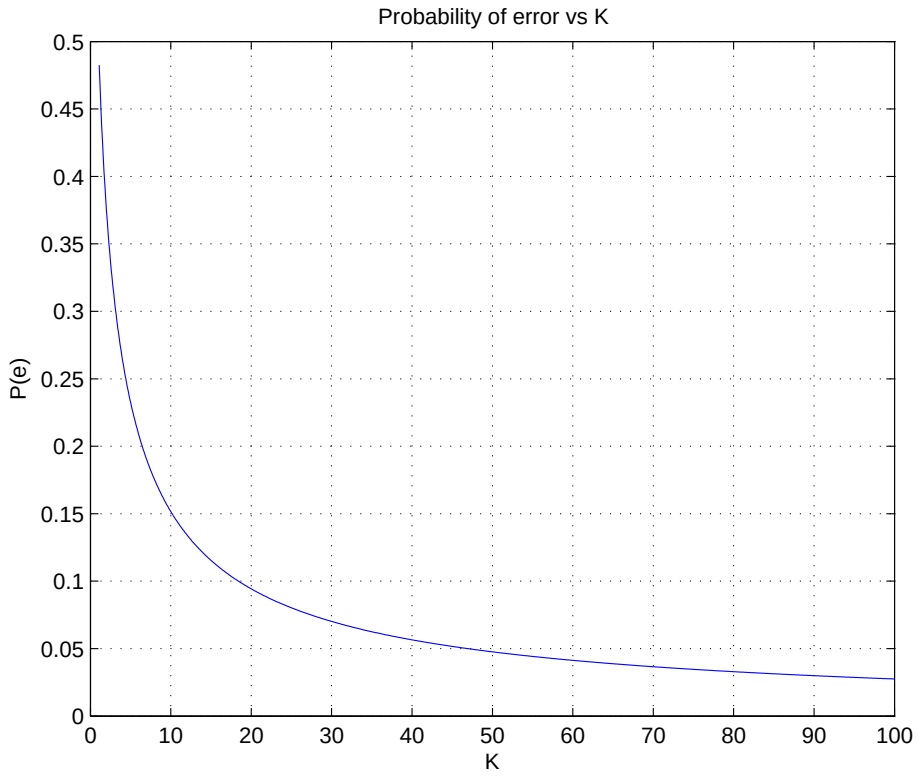

Figure 2.7. P(e) vs. K.

Table 2.4. Maximum K values achieved with changing energy values.

energy	$f_{opt}[n]$	K
1	$[0.5618\ \ 0.0\ \ 0.4300\ \ 0.0\ \ -0.4293\ \ 0.0\ \ -0.5613]^T$	2.1178
10	$[1.6906\ \ 0.0\ \ 1.4635\ \ 0.0\ \ -1.4635\ \ 0.0\ \ -1.6906]^T$	6.2224
20	$[2.3883\ \ 0.0\ \ 2.0726\ \ 0.0\ \ -2.0726\ \ 0.0\ \ -2.3884]^T$	6.3745
50	$[3.7753\ \ 0.0\ \ 3.2782\ \ 0.0\ \ -3.2786\ \ 0.0\ \ -3.7751]^T$	6.4187

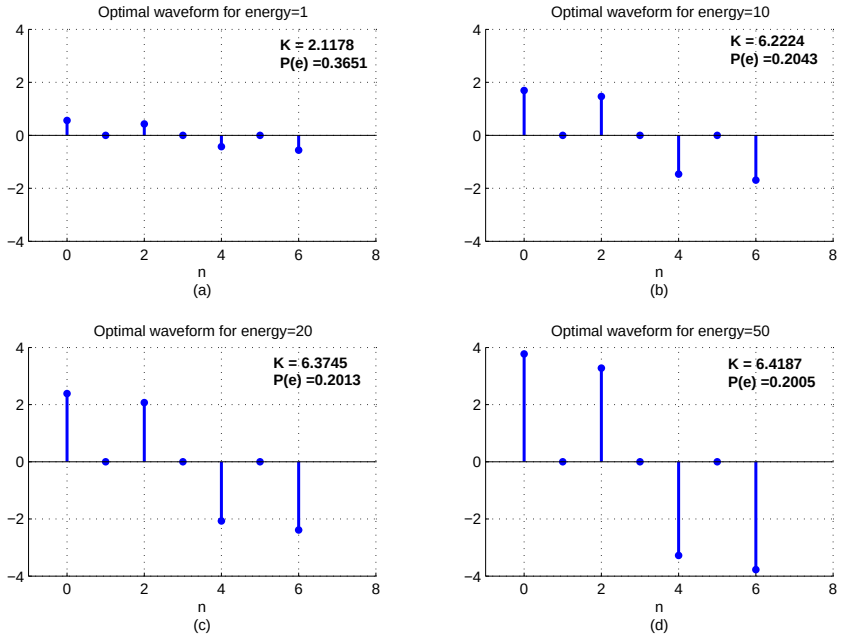

Figure 2.8. Optimum waveforms and the corresponding K and $P(e)$ values for different energy values.

2.2.2.4 More Practical Waveforms

In this section, we present the results calculated for more practical waveforms where the samples, $f[n]$, have the same fixed amplitudes. The resulting waveforms are given in Table 2.5 and Fig. 2.11. It can be observed that a very slight decrease in K happens with this new constraint.

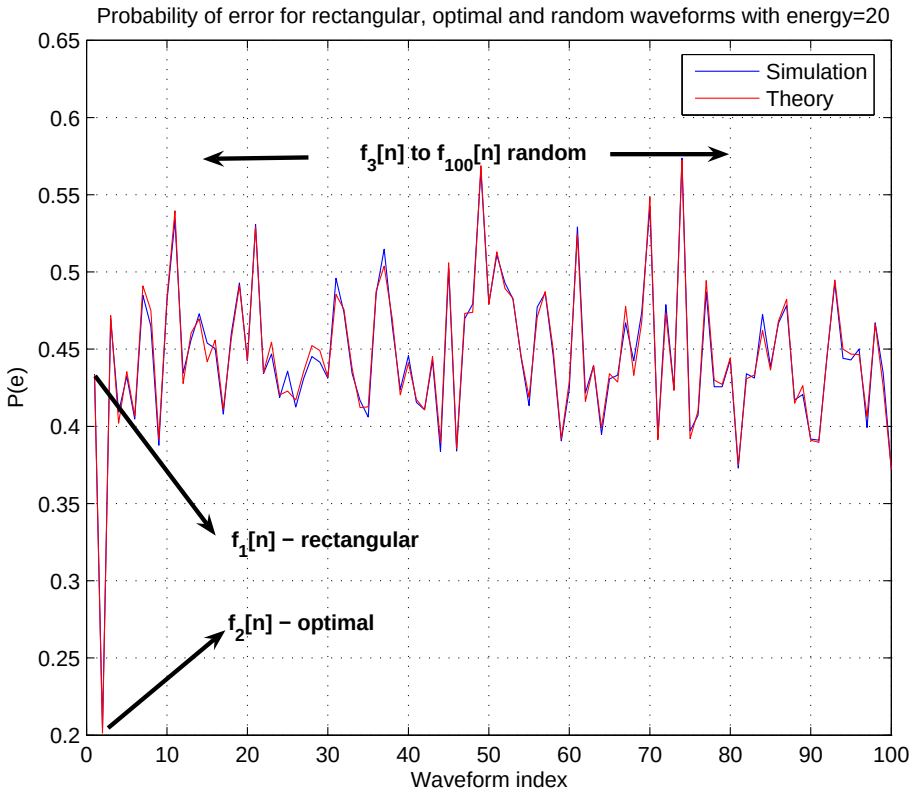

Figure 2.9. P(e) for rectangular, optimal and random waveforms with energy=20.

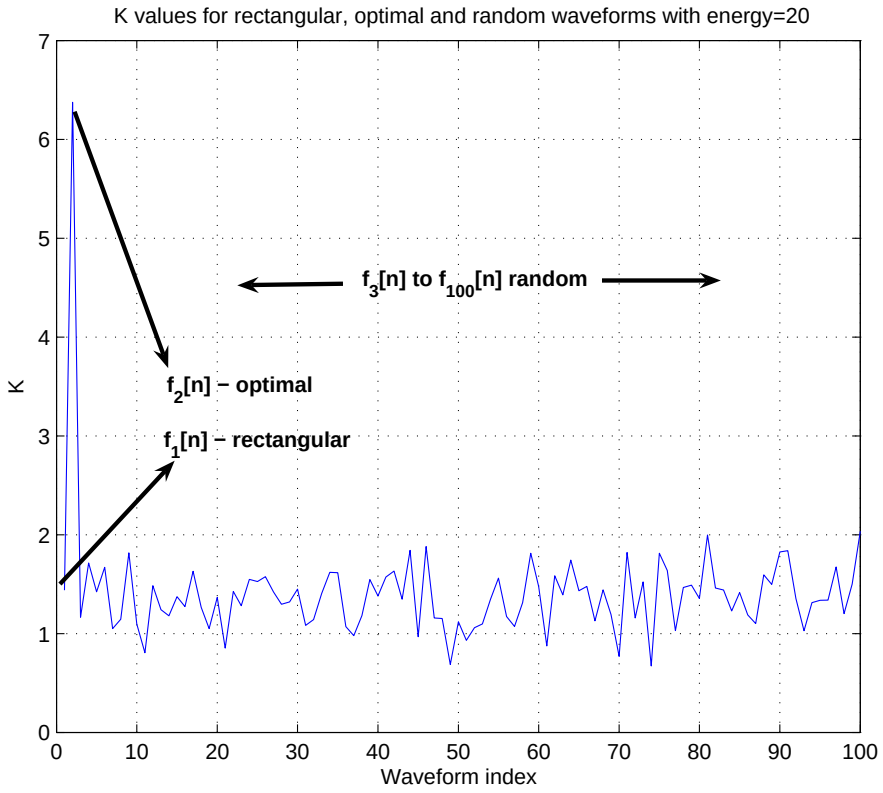

Figure 2.10. K values for rectangular, optimal and random waveforms with energy=20.

Table 2.5. K values achieved with more practical waveforms than in Table 2.4 with various energy values.

energy	$f[n]$	K
1	$[0.5\ \ 0.0\ \ 0.5\ \ 0.0\ \ -0.5\ \ 0.0\ \ -0.5]^T$	2.05
10	$[\sqrt{10}/2\ \ 0.0\ \ \sqrt{10}/2\ \ 0.0\ \ -\sqrt{10}/2\ \ 0.0\ \ -\sqrt{10}/2]^T$	6.0481
20	$[\sqrt{20}/2\ \ 0.0\ \ \sqrt{20}/2\ \ 0.0\ \ -\sqrt{20}/2\ \ 0.0\ \ -\sqrt{20}/2]^T$	6.1980
50	$[\sqrt{50}/2\ \ 0.0\ \ \sqrt{50}/2\ \ 0.0\ \ -\sqrt{50}/2\ \ 0.0\ \ -\sqrt{50}/2]^T$	6.2416

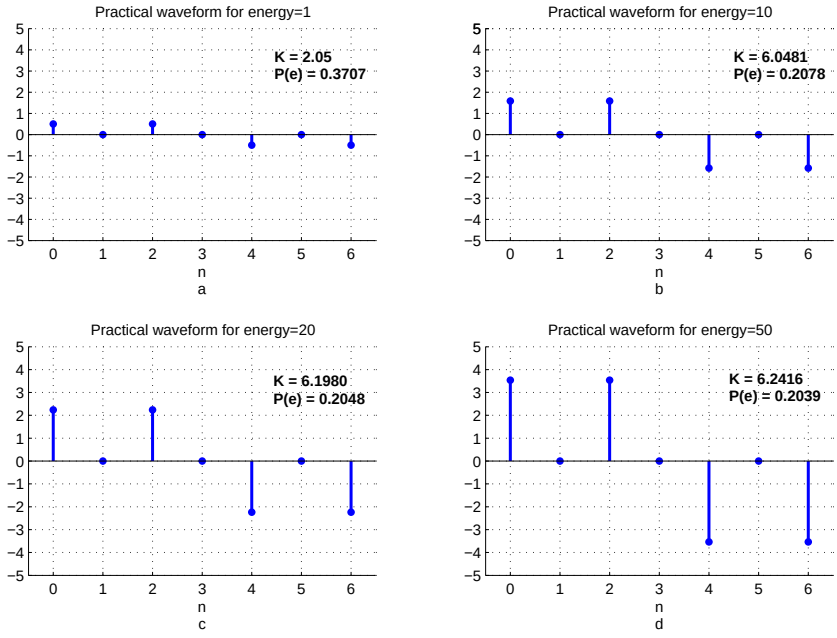

Figure 2.11. More practical waveforms which are alternatives to the ones in Fig. 2.8.

.

2.2.2.5 Changing the Correlator Function

Up to this point, we assumed our radar transmitter sending $f(t)$ and receiver correlating the received signal with the same waveform. Now, we try to see whether changing the correlating signal in the receiver to another waveform, call $g(t)$, can improve identification performance, or more directly, increase K.

First, let us write the new (discretized) definition for K, which is nothing but Eq. 2.66 with a slight change.

$$K = \frac{\sigma_\beta^2 \left(\sum_{n=0}^{N-1} g[n] \sum_{i=0}^{N-1} f[i] h_1[n-i] \right)^2 + \sigma_n^2}{\sigma_\beta^2 \left(\sum_{n=0}^{N-1} g[n] \sum_{i=0}^{N-1} f[i] h_2[n-i] \right)^2 + \sigma_n^2}. \tag{2.67}$$

Now we try to maximize K with respect to the variables $g[0]$,...,$g[N-1]$,$f[0]$,...,$f[N-1]$. We use numerical calculations again to optimize K. We follow the same procedure in Section 2.2.2.3 and find the optimum $f[n]$ and $g[n]$ sequences.

We give the results in Table 2.6, Fig. 2.12 and Fig. 2.13. Here the important observation is that we are able to obtain dramatic increases in K, and accordingly much lower probability of errors. It can be concluded that *using different waveforms for the transmitter and receiver can significantly increase our performance.*

A detail about the values in Table 2.6 is that the waveforms for energy values 20 and 50 are not optimal in reality. Since the computer simulation gives unreliable results for such a high value of energy, we just applied a scaled version of $f_{opt}[n]$ and $g_{opt}[n]$ for energy = 10. Even for this case, the K values are much higher than the case summarized in Table 2.4.

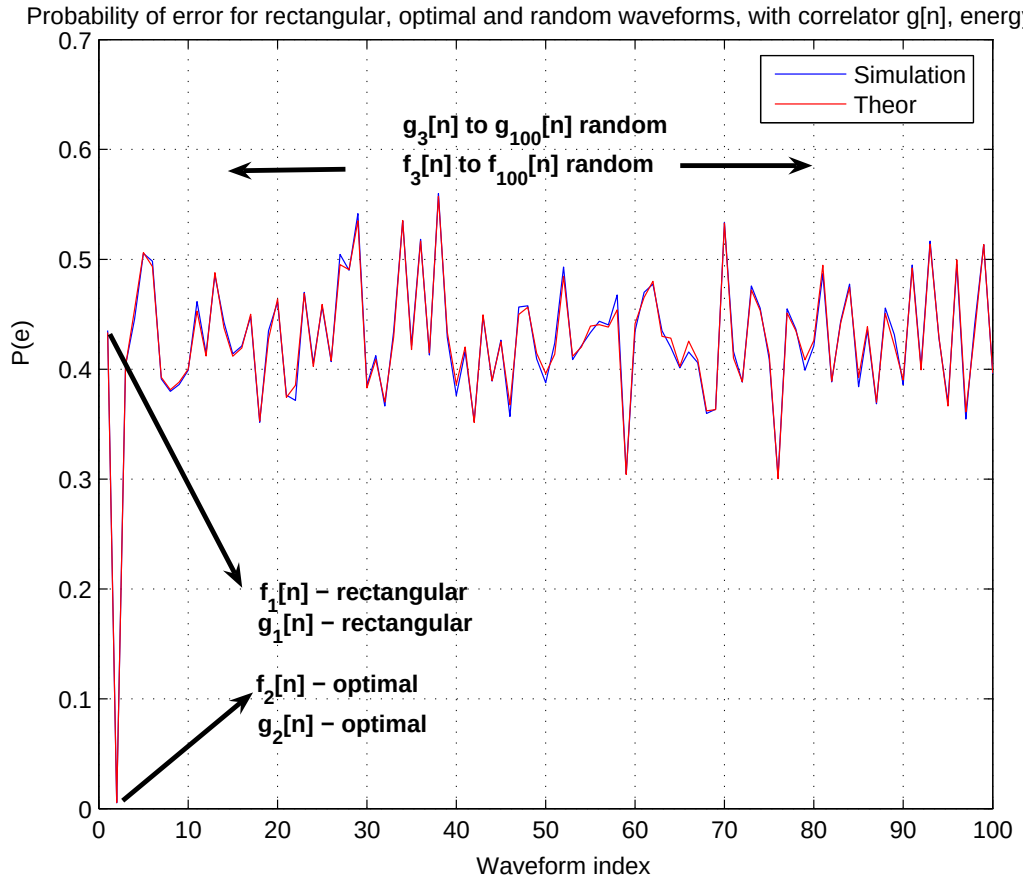

Figure 2.12. P(e) for rectangular, optimal and random waveforms with energy=20, with correlator g[n].

2.2.2.6 Using Phase Modulation

In this part, we repeat our simulations for the case where the samples of radar signal $f[n]$ and the correlator signal $g[n]$ have the same amplitudes but changing phase values. In order to optimize our signals for this new design constraint, we have to review our model. Let us quickly remember our variables. Under the two hypotheses the received waveforms were

$$H_1 : r(t) = \beta u_1(t) + n(t),$$
$$H_2 : r(t) = \beta u_2(t) + n(t). \tag{2.68}$$

Table 2.6. Maximum K values achieved with changing energy values and with different correlator ($g[n]$).

energy	optimal waveforms	K
1	$f_{opt}[n] = [0.6865 \;\; 0.0 \;\; 0.2874 \;\; 0.0 \;\; -0.6265 \;\; 0.0 \;\; -0.2317]^T$	2.8869
	$g_{opt}[n] = [0.2317 \;\; 0.0 \;\; 0.6266 \;\; 0.0 \;\; -0.2877 \;\; 0.0 \;\; -0.6863]^T$	
10	$f_{opt}[n] = [1.4367 \;\; -1.4607 \;\; 1.2432 \;\; 0.8479 \;\; -1.682 \;\; 0.4842 \;\; -0.6886]^T$	168.7093
	$g_{opt}[n] = [0.47627 \;\; -0.48424 \;\; 1.5226 \;\; -0.8479 \;\; -0.23089 \;\; 1.4607 \;\; -2.0772]^T$	
20	$f_{opt}[n] = \sqrt{2}[1.4367 \;\; -1.4607 \;\; 1.2432 \;\; 0.8479 \;\; -1.682 \;\; 0.4842 \;\; -0.6886]^T$	668
	$g_{opt}[n] = \sqrt{2}[0.47627 \;\; -0.48424 \;\; 1.5226 \;\; -0.8479 \;\; -0.23089 \;\; 1.4607 \;\; -2.0772]^T$	
50	$f_{opt}[n] = \sqrt{5}[1.4367 \;\; -1.4607 \;\; 1.2432 \;\; 0.8479 \;\; -1.682 \;\; 0.4842 \;\; -0.6886]^T$	4009
	$g_{opt}[n] = \sqrt{5}[0.47627 \;\; -0.48424 \;\; 1.5226 \;\; -0.8479 \;\; -0.23089 \;\; 1.4607 \;\; -2.0772]^T$	

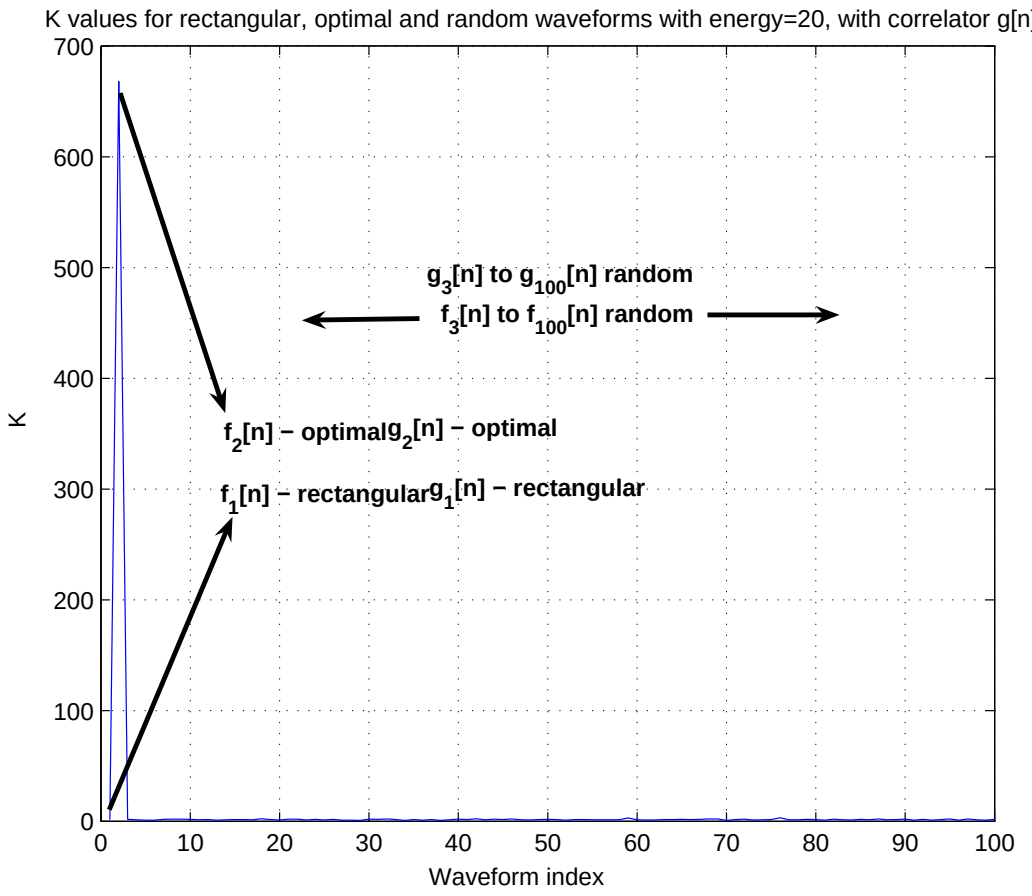

Figure 2.13. K values for rectangular, optimal and random waveforms with energy=20,with correlator g[n].

Recalling that the correlator function $g(t)$ is in general different than $f(t)$, again we define

$$r = \int_0^T r(t)g(t)dt, \tag{2.69}$$

$$u_1 = \int_0^T u_1(t)g(t)dt, \tag{2.70}$$

$$u_2 = \int_0^T u_2(t)g(t)dt, \tag{2.71}$$

$$n = \int_0^T n(t)g(t)dt. \tag{2.72}$$

With these definitions, the random variables we have under the two hypotheses are

$$H_1 : r = \beta u_1 + n, \tag{2.73}$$

$$H_2 : r = \beta u_2 + n. \tag{2.74}$$

The most important difference compared to the amplitude modulation case is that $u_1(t)$ and $u_2(t)$ are now complex baseband signals. So, the calculation regarding the moments of the random variable r_{real} and r_{imag} changes slightly. Without going into too much detail, we say that the only change is the expression of the variances of these random variables. Their variances are now given by

$$\sigma_{r1}^2 = \sigma_\beta^2 \|u_1\|^2 + \sigma_n^2, \tag{2.75}$$

$$\sigma_{r2}^2 = \sigma_\beta^2 \|u_2\|^2 + \sigma_n^2. \tag{2.76}$$

Other than this change, all probability of error calculations are the same.

We make our simulations for the case where $f[n]$ and $g[n]$ are allowed to be different. Our aim is to find the phase values that minimize the probability of error. In our simulations, we assume $f[n]$ and $g[n]$ to be consisting of 7 pulses each. Hence,

our purpose is to find the optimum 14 phase values. In the following simulation, we allow the pulses to have either 0 or π as the phase value. The pulses are also allowed to be OFF, meaning that there is no signal at all for that particular duration. So, any pulse can have either of the values of $\{-A, A, 0\}$, A depending on the energy constraint.

The results are given in Table 2.7. It can be easily seen that even with two phases we can have K values close to the amplitude modulation case. $f[n]$ and $g[n]$ values are the scaled versions of each other for the last 3 energy values. An interesting observation is that these signals make the output for the second target zero. In other words, when this radar and the correlator signal is used, the u_2 value in Eq. 2.62 turns out to be zero which helps maximize K.

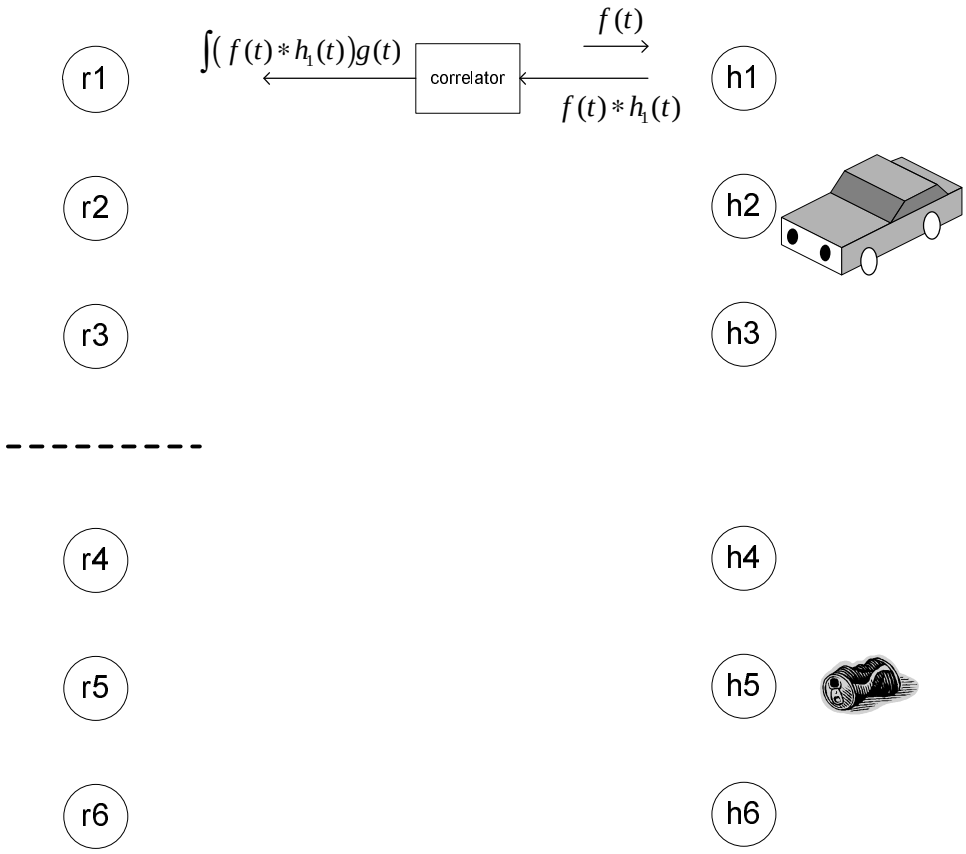

Figure 2.14. Family of targets.

2.3 Family of Targets

In this project, we are trying to distinguish between two kinds of targets which differ by their sizes, a coke can and a car for example. In reality, each of these two target classes consists of a large number of impulse responses. But, formerly, we were only able to find the optimal signals when each target class is assumed to be consisting of a single impulse response. Of course, this is a too unrealistic assumption. For instance, each coke can might have a different impulse response (or even a single one can have different impulse responses depending on the angle of incidence). So, the next goal is to distinguish between classes of targets that consist of a number of impulse responses. Again, we are trying to find the optimal $f(t)$ and $g(t)$ based on what $\|r\|$ they correspond to.

Table 2.7. Maximum K values achieved with changing energy values using phase modulation.

energy	optimal waveforms	K	$P(e)$
1	$f_{opt}[n] = [-1/\sqrt{2}\ \ 0\ \ 0\ \ 0\ \ 1/\sqrt{2}\ \ 0\ \ -0]^T$	2.6667	0.3265
	$g_{opt}[n] = [-0.5\ \ 0\ \ -0.5\ \ 0\ \ 0.5\ \ 0\ \ 0.5]^T$		
10	$f_{opt}[n] = [-\sqrt{10}/\sqrt{3}\ \ 0\ \ -\sqrt{10}/\sqrt{3}\ \ 0\ \ \sqrt{10}/\sqrt{3}\ \ 0\ \ 0]^T$	151	0.0196
	$g_{opt}[n] = [0\ \ 0\ \ -\sqrt{10}/\sqrt{2}\ \ 0\ \ 0\ \ 0\ \ -\sqrt{10}/\sqrt{2}]^T$		
20	$f_{opt}[n] = [-\sqrt{20}/\sqrt{3}\ \ 0\ \ -\sqrt{20}/\sqrt{3}\ \ 0\ \ \sqrt{20}/\sqrt{3}\ \ 0\ \ 0]^T$	601	0.0061
	$g_{opt}[n] = [0\ \ 0\ \ -\sqrt{20}/\sqrt{2}\ \ 0\ \ 0\ \ 0\ \ -\sqrt{20}/\sqrt{2}]^T$		
50	$f_{opt}[n] = [-\sqrt{50}/\sqrt{3}\ \ 0\ \ -\sqrt{50}/\sqrt{3}\ \ 0\ \ \sqrt{50}/\sqrt{3}\ \ 0\ \ 0]$	3571	0.0013
	$g_{opt}[n] = [0\ \ 0\ \ -\sqrt{50}/\sqrt{2}\ \ 0\ \ 0\ \ 0\ \ -\sqrt{50}/\sqrt{2}]$		

Figure 2.14 illustrates a simple model where both target classes constitute of three impulse responses. The car and coke can class has three impulse responses each ($h_1(t)$, $h_2(t)$, $h_3(t)$ are the cars, $h_4(t)$, $h_5(t)$, $h_6(t)$ are the coke cans). We know from the last section that the statistic we use for the decision is the norm of the result of the correlation. Let us assume that $\|r_1\|$, $\|r_2\|$, and $\|r_3\|$ correspond to the car responses. Similarly, $\|r_4\|$, $\|r_5\|$, and $\|r_6\|$ are the coke can responses. What we would like is to have the car and coke can responses as far apart as possible. This will decrease the probability of error. Let us write our goal.

$$\textit{Maximize } \Big(\min(\|r_1\|, \|r_2\|, \|r_3\|) - \max(\|r_4\|, \|r_5\|, \|r_6\|) \Big).$$

We make this problem easier by assuming the coke can responses will be zero. Of course, this means that we have to choose our signals from the ones that have zero response from the coke can. Keeping this in mind, the new goal is

$$\textit{Maximize } \min(\|r_1\|, \|r_2\|, \|r_3\|).$$

Let us explain the situation better. We first select an $(f(t), g(t))$ pair. When we use it, it gives us some $(\|r_1\|, \|r_2\|, \|r_3\|)$ in return. The minimum of these values tells us how good we did, because it corresponds to the worst case scenario.

This problem can be viewed as a game. The game we are playing here has the following rules:

i. There are two players.

ii. We are the one who plays first.

iii. We select our playing card from the set of all $(f(t), g(t))$ pairs that have zero coke can response.

iv. After we make our move, our opponent plays by choosing one of his car impulse responses.

v. $\|r\|$ is how much we gain from this game (payoff).

Since we do not know which move our opponent will make, we should be ready to encounter an unfriendly opponent who tries to minimize what we gain. This means that we should *maximize our minimum gain.* This problem is called a minimax problem.

Now consider this situation: We know that for a given target impulse response we can find the optimal signals. For our simple example, consider Table 2.8. Suppose we want to maximize the minimum gain based on the values in this hypothetical table. Let us try to choose the best signal pair. If we choose $(f_1(t), g_1(t))$ or $(f_3(t), g_3(t))$, our minimum gain will definitely be smaller than 5 since the best we can gain with the second target is 5. So, in order to solve our problem, it seems we should choose $(f_2(t), g_2(t))$ since it is the best choice in the worst situation. In order to see what happens when we make that decision, let us look at Table 2.9 where we choose $(f_2(t), g_2(t))$ as the signal pair to be used.

As can be seen, the minimum gain we get is 4. If $(f_2(t), g_2(t))$ performed better than 5 for $h_1(t)$ and $h_3(t)$, it would definitely be the minimax optimal signal pair. Unfortunately, in this case, the optimal signal pair for the worst case does not work in the general minimax sense. So, the optimal signal pair may be some other $(f_4(t), g_4(t))$ which remains to be found. This example shows how minimax optimization for family of targets can be harder compared to the optimization for a given target. However, the good news is that there are certain conditions a problem may carry where the optimal signal for the worst case is guaranteed to be the minimax optimal. If the particular problem in hand carries these conditions, the solution is easily found.

In [3], the authors discuss the conditions under which the optimal signal set for the worst case also works in general. When the sufficient conditions explained there are met, we can just find the optimal signal pair for the worst-case target and be sure that they are the best signals we can use. This type of solution is called a

Table 2.8. Optimal signals and their corresponding $\|r\|$ values.

Target impulse response	Optimal Signals	$\|r\|$
$h_1(t)$	$(f_1(t), g_1(t))$	6
$h_2(t)$	$(f_2(t), g_2(t))$	5
$h_3(t)$	$(f_3(t), g_3(t))$	9

Table 2.9. The corresponding $\|r\|$ values when the optimal signal pair for target 2 is used.

Target impulse response	Used Signals	$\|r\|$
$h_1(t)$	$(f_2(t), g_2(t))$	4
$h_2(t)$	$(f_2(t), g_2(t))$	5
$h_3(t)$	$(f_2(t), g_2(t))$	7

saddle point solution. In order to be able to check whether our problem accepts a saddle point solution, we should define the target classes explicitly. However, this requires the knowledge of physical characteristics of the responses by small and large objects. Thus, modeling of the classes is the largest step to be taken towards a solution. When appropriate modeling is done, radar waveforms giving more robust classification performances can be found.

2.4 Summary

In this chapter, we present our work on waveform design for car radars. We attempt to find the best waveform that distinguishes large objects from small ones. Several optimal waveforms are calculated under different constraints regarding how the waveform is generated and how the reflecting signal is processed. We also investigate the case in which the responses from the targets are assumed to be random signals. For this case, we calculate a new performance measure and find the optimal waveforms. In all these scenarios, the results are found under the assumption of two distinct responses representing large and small objects. However, for more realistic results, these targets should be assumed to be forming two "classes of responses". With this more realistic assumption, the problem turns into the problem of classification between two families of targets. This is a more difficult problem in general, and needs to carry certain conditions in order to be solved easily. At this point, a physical model of the problem becomes crucial to the solution. Once this model is obtained, a realistic performance measure can be found and the optimal waveform can be calculated.

CHAPTER 3

DIGITAL-TO-ANALOG CONVERTER LINEARIZATION

Most of the current systems employ digital processing of signals. Digital signal processing has many advantages over its analog counterpart some of which can be listed as flexibility of design, accuracy, ease of storage and lower cost. These advantages make digital signal processing the natural choice for countless practical applications. Because the signals to be processed and/or the signals to be produced at the output are often inherently analog, a digital signal processor needs to include an analog-to-digital converter (ADC) at the input end and a digital-to-analog converter (DAC) at the output end. In order to fully exploit the above-mentioned advantages of working in the digital domain, the distortion caused by these converters has to be minimized.

Like all practical systems, DACs show some nonideal behavior. This often shows itself as a nonlinear input-output characteristic. Linearization is the process of reversing this effect. In our problem, we will use the method of precompensation. Precompensation (or predistortion) means changing the input to the system to get the desired output. This way we compensate for the nonlinearity.

The important question is how to change the input. Answering this question requires the knowledge of the nonlinear characteristic of the DAC. The nonlinearity of the DAC comes from many different phenomena present in the DAC architecture. Some of the sources of nonlinearity may be more dominant than the others depending on the operating conditions. In order to design a predistorter for the DAC, we need a DAC model in the first place. The DAC model needs to account for at least the most

significant sources of nonlinear behavior. Once we have the model, i.e. the input-output relation of the DAC, we can analyze its nonlinear effect on the input. This analysis is crucial as it gives insight to what can be done at the input to compensate for the nonlinearity.

In this chapter, we first give a general description of DAC errors, some of which cause nonlinearity. In Section 3.2, we explain what predistortion is and what it means for the special case of DAC linearization. We present the DAC model we use in Section 3.3. The important nonlinearity analysis of this DAC model for different scenarios is given in Section 3.4. We end the chapter with the summary in Section 3.5.

3.1 DAC Errors

Digital-to-analog converters map the digital input to the corresponding analog output. The input to a DAC is a digital signal from a finite set of levels (or codes). Every input code corresponds to an analog value. The input-output function of a theoretical 3-bit DAC is plotted in Figure 3.1. Based on this input-output relation, an example of an ideal time domain response of this DAC can be plotted as in Figure 3.2.

Real DAC input-output relations and time domain responses, however, deviate from these ideal curves in several ways. These deviations may be referred to as simply errors. A certain type of error may or may not be of nonlinear nature. Often nonlinear errors are the ones causing the undesired response. An important reason why nonlinear errors are unwanted is the fact that they cause spectral regrowth. When a signal is passed through a nonlinear system, the bandwidth it occupies is broadened. This can be mostly harmful in communication applications where the output signals need to be strictly confined to the predefined frequency interval.

DAC errors are usually categorized as *static errors* and *dynamic errors*. Very shortly, static errors can be described as the deviations from the ideal function in

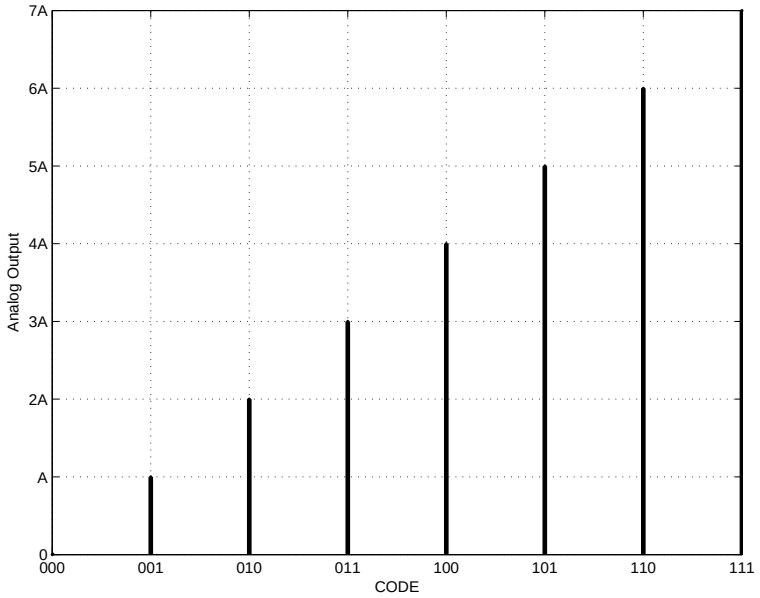

Figure 3.1. Ideal DAC input-output function.

Figure 3.1. Dynamic errors are the ones that show themselves during the transition from one digital code to another which make the real response look different than the ideal one given in Figure 3.2.

Static Errors

Static errors of DACs are the differences between the *settled* real response and the ideal response. They can be listed as offset error, gain error, differential nonlinearity and integral nonlinearity. In order to explain these errors in the simplest way, let us consider the input-output function of a real 3-bit DAC in Figure 3.3.

Offset error is the value where the real DAC input-output function crosses the y-axis. It is the response of the DAC to the digital input word corresponding to zero. For the example in Figure 3.3 the offset error is $0.2A$. This error *does not* contribute to the nonlinearity of a DAC.

Gain error is the error in the slope of the line connecting the zero-scale and full-scale output of a DAC. Ideally, for uniform quantization, this line should make 45°

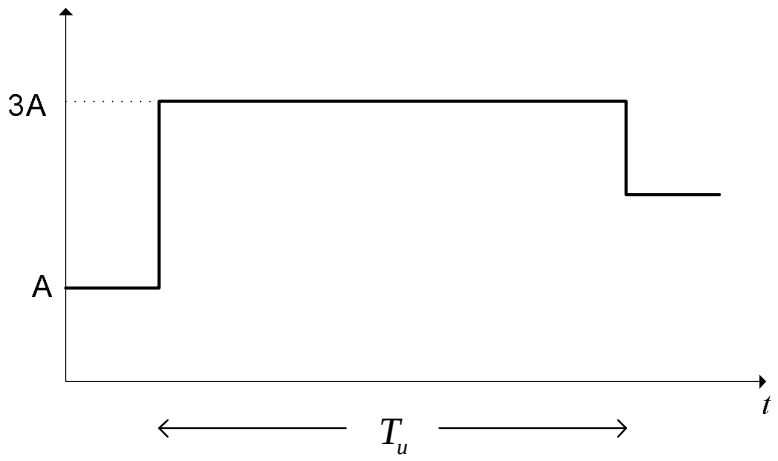

Figure 3.2. Ideal response of the DAC in Figure 3.1 to the digital sequence [001,011,010]. T_u is the update period of the DAC.

with the x-axis. Like the offset error, gain error *does not* contribute to the nonlinearity of a DAC. As the name implies, the error is just in the overall gain.

The only static errors that cause nonlinearity are the differential and integral nonlinearity. To understand these errors, first note that for an ideal DAC, every one bit increase in the code corresponds to the same amount of analog output increase. This amount (A in Figure 3.1) is called a one LSB (least significant bit) difference, or just one LSB for short. If the DAC is not able to fix this value, the input-output relation will be different.

For the simple 3-bit DAC example, the real input-output relation is given in Figure 3.3. As can be seen, the difference between successive analog outputs are not equal to 1 LSB. They differ for different successive code pairs. For example, between code 000 and 001 the difference is $1.1A$ although it is supposed to be A. This $0.1A$ difference is defined as the Differential Nonlinearity (DNL). As these DNLs accumulate over codes, every analog output level deviates from the ideal. This deviation is called Integral Nonlinearity (INL). For our example, INL for 001 is $0.3A$. An INL figure for a commercially available 16-bit DAC is given in Figure 3.4 [10].

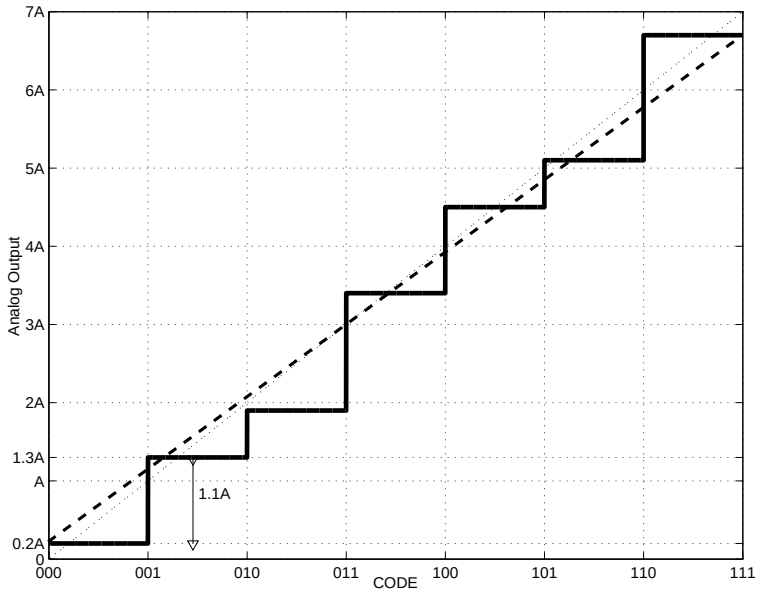

Figure 3.3. A nonideal 3-bit DAC input-output function example.

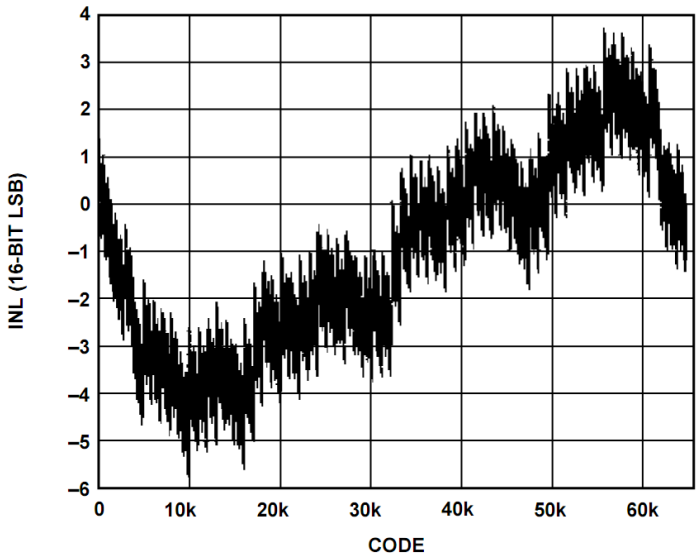

Figure 3.4. Plot of INL versus the digital input measured for a commercially available DAC [10].

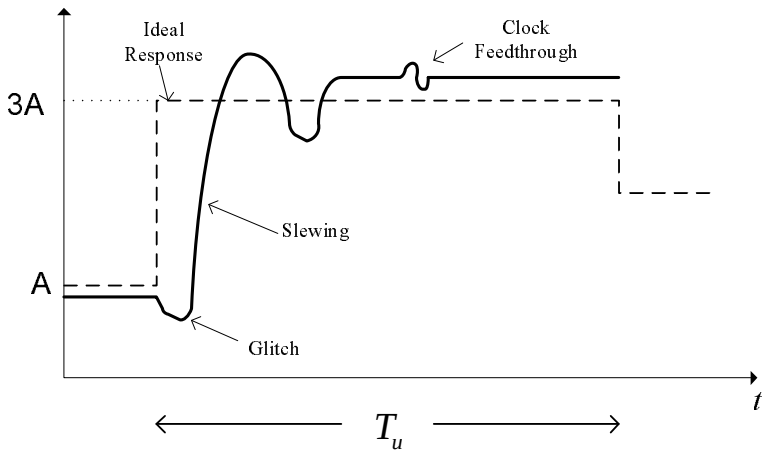

Figure 3.5. Example of a real DAC time-domain response.

Dynamic Errors

Unlike static errors, dynamic errors involve the nonideal behavior of the analog DAC output during the transition from one digital code to the next one. We will explain these errors again by referring to a typical DAC time-domain response given in Figure 3.5 [6]. The dynamic errors of a DAC can be listed as glitches, slewing and clock feedthrough. There are also other effects like switch memory and timing jitter that we will assume to have negligible impact.

Glitches are the unwanted impulsive changes in the output amplitude at the start of the transition period. Glitches occur due to lack of exact synchronization of switches in DACs. Consider a transition from the code 01111 to 10000 for a 5-bit DAC. Let us assume each bit is controlled by a switch. For this code transition, one switch (MSB) needs to be closed and the remaining four of them need to open all at the same time. If, however, the MSB switches faster, we will temporarily see the full-scale output corresponding to 11111. As can be deduced from this explanation, glitches depend on the input signal and one can expect higher glitch errors with increasing switch activity [5]. For example, high output signal frequency leads to larger step sizes and more switching activity. Although glitches are signal-dependent er-

rors, we will assume they will only increase the noise floor in the frequency spectrum instead of exciting some harmonics [5]. Note that, for eliminating nonlinear effects, reducing the harmonics power is more of a concern.

Slewing effect comes from the fact that the DAC cannot change its output value instantaneously. The output value ramps up from the previous level to the current one. In other words, there is a nonzero settling time for the output to reach the desired value. Slewing effect is one of the reasons of frequency distortion and is likely to create undesired harmonics [5]. In fact, what really causes nonlinearity is the nature of slewing being *code-dependent* [6]. We will see later why this is the case. One can expect the slewing effect to be higher for larger step sizes. Hence, larger output signal frequency creates larger slewing error.

Clock Feedthrough (CFT) is a dynamic error caused by the capacitive coupling between the clock signal controlling a switch and the output signal [7]. CFT occurs when the switches turn on and off. This means CFT happens twice for a given update period, T_u [5]. Because of that, CFT is assumed to produce a harmonic only at half the update frequency $f_u/2$ [6]. CFT may also cause an error in the settled value [7], but for current-steering DACs the effect is expected to be transient like the glitch effect [5].

Having seen the static and dynamic errors, let us shortly comment on their relative importance. Generally for small DAC speeds and low signal frequencies static errors tend to dominate. The situation reverses for higher update rates and signal frequencies. Dynamic errors are the dominant cause of distortion in this case [6].

3.2 Predistortion

In general we can count two ways of combating nonidealities of systems. The first and maybe the most obvious way would be to make the system a better one. This may be in terms of redesigning the system, replacing the components, etc. However, if

we are constrained to work with the given system, we have to admit the situation and do our best. The second method then would be to put additional systems that make the overall input-output relation closer to the ideal one. Although the cancelation of the nonidealities makes the perfect solution, this may not be easy, if not impossible.

Predistortion is a method of the second type mentioned above. What we do in predistortion is to *give the "wrong" input to get the "right" output*. Let us explain what we mean. In general, because we do not have an ideal system in hand, the output will be different than what we intended. However, a different input this time can give an output closer to ideal one. So, we predistort the input, i.e. change the input *before* it enters the system. As explained in the Introduction, predistortion involves an additional system between the input and the nonideal system in hand. In our case, the nonideal system is the DAC and the predistorter system is the one we are trying to design. So, the question to be answered is: What is the predistorter mechanism which makes the overall DAC system closest to the ideal? By "ideal" here, we mean "linear". We will see later that a system's linearity is usually inspected by means of frequency spectrum analysis. We will be hoping to see a clean frequency spectrum at the output end when designing our predistorter.

A predistorter can be implemented as an online or an offline one. An online predistorter adapts itself as the system continues running. Offline predistorter, on the other hand, will require a calibration period before the overall system starts operating. Once the predistorter is set, it will be used as a fixed (not adaptive) system. In this project, we are trying to linearize DACs with offline predistorters.

Before ending the description of predistortion, it should be noted that the success of the predistorter depends on the quality of the model assumed, i.e. how well it matches reality. On the other hand, a good model is likely to be a complex one, which in turn complicates the predistorter design. This is an important trade-off in choosing a model. DACs in general are hard to model, and most probably there is

no model that covers the complete input-output relation of a DAC, let alone having a simple one [6]. As a matter of fact even a single type of error, e.g. glitches, can be extremely hard to model [5]. In conclusion, we are constrained to work with an incomplete DAC model and we will be seeking a model that covers most dominant nonlinearities under our assumed operating conditions.

3.3 DAC Model

As mentioned in the previous section, the success of linearization is closely tied to what is selected as the DAC model. In fact, modeling of the system can be thought of as the starting point to the linearizer design. This is because the crucial nonlinear input-output relation is given by the model.

A DAC model has to be architecture-dependent. In other words, how the actual DAC is implemented as a circuit determines the model in the first place. DACs can be implemented in several different architectures. In this project, we assume the DAC to be of current-steering architecture, as most high-speed DACs are this type [7]. Upon the selection of the architecture, a convenient DAC model can now be looked for. Unfortunately, the literature is not rich in DAC modeling, at least when compared to modeling (and compensation) of ADCs. However, there are some relatively recent works in modeling of the dynamic errors of DACs [8, 9]. As mentioned in Section 3.1, for high update rates and/or high output signal frequencies, dynamic errors tend to dominate the DAC behavior. This allows us to use a model which is focued on the dynamic errors. In this project, we assume the DAC will have the nonlinear input-output relation given by the model developed in [9].

The model we assume is developed by some component-wise simplifications to an actual current-steering DAC circuit. Every nonideal component, like current sources, switches, output wires, etc., is represented by an approximate equivalent circuit. After that, the combination of all these components gives us an overall DAC model.

This approximate DAC circuit is given in [9] as in Figure 3.6. This circuit is further simplified and represented as the input-dependent circuit in Figure 3.7. Of course, for our purposes, what matters is the nonlinear equation deduced from the finalized DAC model. The circuit in Figure 3.7 has the following input-output relation:

$$y(t) = y(0) + (I(x)R - y(0)) \left(1 - e^{-\frac{t}{RC(x)}} \right), \quad 0 < t \leq T_u, \tag{3.1}$$

where we assume the output to be defined as the voltage value taken at the designated node. $I(x)$ is the ideal current produced for the code x on the interval $(0, T_u]$, where T_u is the update period of the DAC. The most interesting point in this model is the code-dependent capacitor $C(x)$. We had mentioned in Section 3.1 that the code-dependence of the slewing effect is the main reason behind the nonlinearity rather than its existence alone. This can be easily seen from the equation. If the capacitor $C(x)$ was a constant for all possible codes, then this would still be a linear input-output relation although it does not produce the desired sample-and-hold output. In order to have a general understanding of what kind of an output we would see according to this equation, consider the plots in Figure 3.8. The two curves are the different outputs to the discrete-time input $x[\cdot] = [0, 2, 5, 1, 4]$. The black curve is the ideal sample-and-hold signal while the blue one is the slewed output calculated by Equation 3.1 (for this plot $C(x)$ is assumed to be proportional to $1/|x|$). The first obvious difference between the curves is the finite rate of state change in the slewed curve. However, also notice that for every period there is a different time constant. This is most evident on $(3T, 4T]$ where the change is so slow that the value at $4T$ cannot even come close to the ideal value $y(4T) = 1$.

Assuming that $C(x)$ is given explicitly, the time domain relation given in Equation 3.1 is enough to describe the nonlinear input-output function of this DAC model. However, frequency spectrum analysis is a more convenient method to explore in what

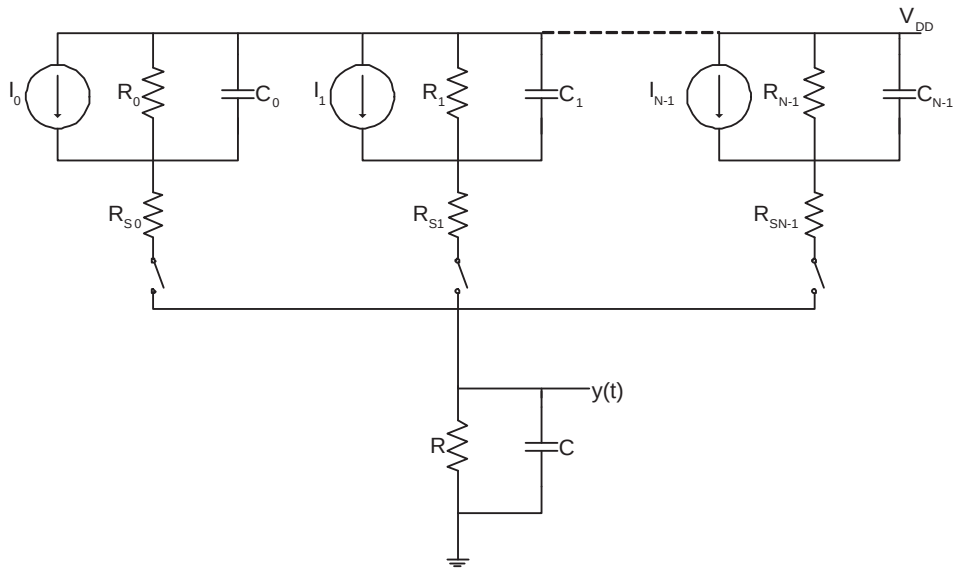

Figure 3.6. Simplified current-steering DAC circuit [9].

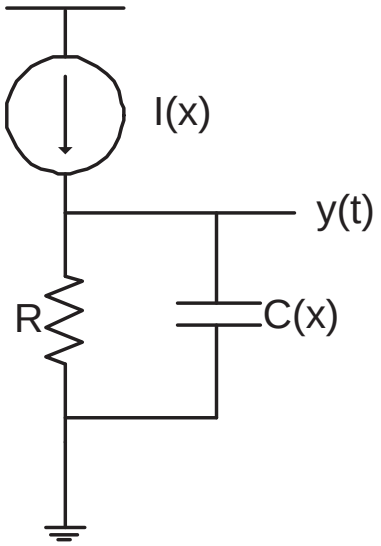

Figure 3.7. DAC model used in this project [9].

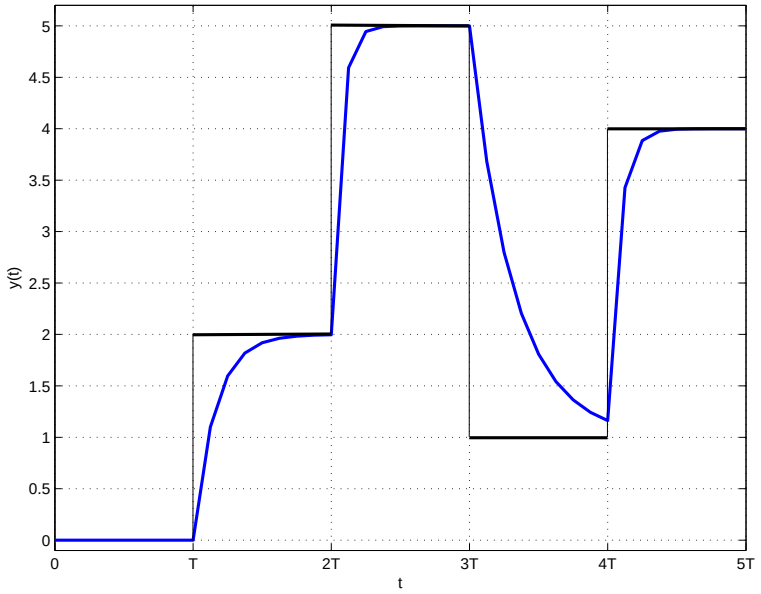

Figure 3.8. Ideal sample-and-hold output of the DAC to the input $x = [0, 2, 5, 1, 4]$ (black curve) and the output according to Equation 3.1 (blue curve).

ways this function effects a given input signal. For nonlinearity analysis, it is common to use a sinusoid as the input and investigate the harmonics present in the output spectrum. The locations, strength or number of harmonics can give valuable information about the system under concern. As a matter of fact, some specifications used to assess DAC performance are obtained using this kind of spectrum analysis. For example, a common DAC specification called Spurious Free Dynamic Range (SFDR) is defined as the difference in the powers of the fundamental and the largest harmonic in the output spectrum when a sinusoid is applied to the DAC. In the analysis of our model given by Equation 3.1, we study the DAC nonlinearity by means of frequency spectrum analysis.

3.4 Nonlinearity Analysis of the DAC Model

In order to come up with a linearizer, we should first understand what kind of a nonlinear effect Equation 3.1 implies. But this equation is incomplete as long as $C(x)$

is not clearly defined. In this section, we explain how different definitions of $C(x)$ affect the nonlinearity. However, we first do some simplifications on Equation 3.1 to make it easier to work on. Let us first rewrite the output for an arbitrary instant of time:

$$y(nT_u + t) = y(nT_u) + (x[n] - y(nT_u)) \left(1 - e^{-\frac{t}{RC(x[n])}} \right), \quad 0 < t \leq T_u. \tag{3.2}$$

Notice that we replaced $I(x)R$ by $x[n]$ because they are both equal to the ideal output value for the interval $(nT_u, (n+1)T_u]$. The above equation shows that $y(nT_u + t)$ depends on $x[n]$ and $y(nT_u)$. $y(nT_u)$ itself is determined by an equation similar to Equation 3.2:

$$\begin{aligned} y((n-1)T_u + T_u) &= y((n-1)T_u) + \\ &\quad (x[n-1] - y((n-1)T_u)) \left(1 - e^{-\frac{T_u}{RC(x[n-1])}} \right). \end{aligned}$$

As can be seen, $y(nT_u)$ depends on $y((n-1)T_u)$ and $x[n-1]$. As we take these dependencies back to the output value corresponding to the first input code, we see that the value of the output at any instant is dependent on *all* the previous input values. This complicates the analysis because in order to write the output as a function of the input only, we should have a very long equation which involves the repeated application of the same relation given in Equation 3.2. We can make an approximation here to simplify Equation 3.2. If we assume a big time constant, we can say $y(nT_u^+) \approx x[n-1]$. The equation becomes

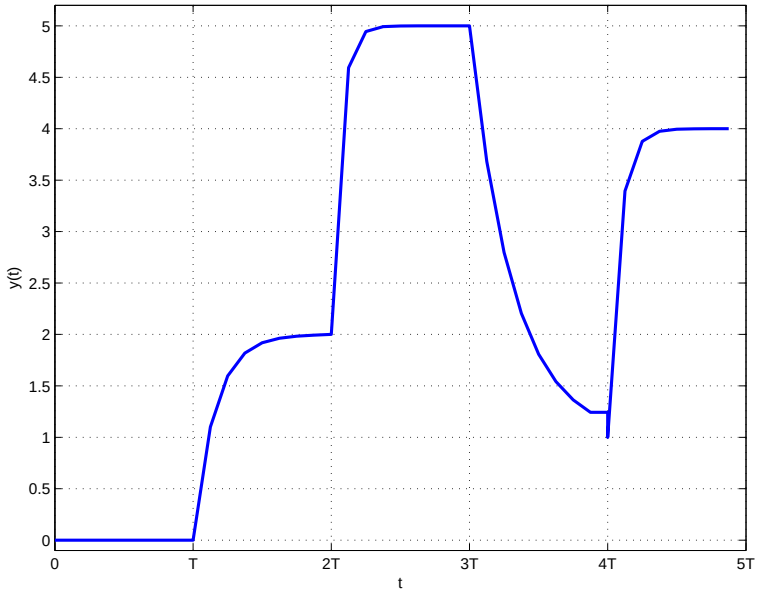

Figure 3.9. DAC output to the input $x = [0, 2, 5, 1, 4]$ according to Equation 3.3.

$$y(nT_u + t) = x[n-1] + (x[n] - x[n-1]) \left(1 - e^{-\frac{t}{RC(x[n])}} \right), \quad 0 < t \leq T_u. \tag{3.3}$$

Equation 3.3 is simpler because this time the value of the output depends only on the current and the previous input value. Notice that, here, we do not ignore the error made in the value of $y(nT_u)$, but rather we ignore the fact that in the next period, the output will rise from the value it just reached. More clearly, this modification makes the output discontinuous. Figure 3.9 shows the output for the same input as in Figure 3.8 but now calculated using Equation 3.3.

It is clear that, as long as $RC(x) \neq 0$, the value of $y(\cdot)$ will fall short to reach the intended value at the update instants. In other words, if we sample $y(\cdot)$ at those instants we will see nonzero error values. Let us explain this by rewriting Equation 3.3 by considering the output as the discrete-time signal $y[\cdot]$ generated by samples of $y(\cdot)$ at multiples of T_u.

$$y[n+1] = x[n-1] + (x[n] - x[n-1]) \left(1 - e^{-\frac{T_u}{RC(x[n])}} \right), \tag{3.4}$$

where $y[n] \triangleq y(nT_u)$.

The error made at the $n+1$st sample is the difference between $y[n+1]$ and its ideal value, $x[n]$. Using Equation 3.4, we can define the discrete-time error signal $e[\cdot]$ as:

$$\begin{aligned} e[n] = y[n] - x[n-1] &= x[n-2] + (x[n-1] - x[n-2]) \left(1 - e^{-\frac{T_u}{RC(x[n-1])}} \right) \\ &\quad -x[n-1] \\ &= (x[n-2] - x[n-1]) e^{-\frac{T_u}{RC(x[n-1])}}. \end{aligned}$$

With this definition, now the *real* output can be thought of as an error signal added to the *ideal* output. The error signal $e[\cdot]$ is the part causing nonlinearity. The source of the nonlinearity is not the existence of $e[\cdot]$, but its nonlinear dependence on $x[\cdot]$. We need to explore how the definition of $C(x)$ effects the nonlinearity. In the following sections we analyze what happens in different scenarios.

3.4.1 $C(x)$ independent and identically distributed in x

In Equation 3.4, we know that $C(x)$ implies there is a different capacitance for every update period. As a starting analysis, we first consider the extreme case where the capacitance is a random variable independent of the input signal. This analysis will provide a good basis for the more realistic situations where $C(x)$ is assumed to be a random variable whose statistics depend on x.

We start by some definitions to make the analysis easier. We can define a new discrete-time signal $z[n] \triangleq x[n-2] - x[n-1]$. With this definition,

$$e[n] = z[n]e^{-\frac{T_u}{RC(x[n-1])}} \quad .$$

Now, we define a discrete-time random process r_n, such that $r_n = T_u/RC(x[n-1])$. This makes $z[n]$ a random process and it can be more appropriately denoted as z_n. Finally, with all these definitions, $e[\cdot]$ also becomes a random process and can be defined as:

$$e_n = z_n e^{-r_n}. \tag{3.5}$$

As mentioned above, we will do the nonlinearity analysis by means of frequency spectrum calculation. e_n is a random process and we need to calculate its power spectral density (psd). If we assume the input process to be wide-sense stationary, z_n will also be a wide-sense stationary process. If we consider r_n to be a white noise process independent of z_n, then the autocorrelation function of e_n can be calculated as:

$$\begin{aligned}
R_e[n_1, n_2] = E\{e_{n_1} e_{n_2}\} &= E\{z_{n_1} z_{n_2} e^{-r_{n_1}} e^{-r_{n_2}}\} \\
&= E\{z_{n_1} z_{n_2}\} E\{e^{-r_{n_1}} e^{-r_{n_2}}\} \\
&= \begin{cases} R_z[0] E\{e^{-2r_{n_1}}\} & \text{if } n_1 = n_2 \\ R_z[n_2 - n_1] E\{e^{-r_{n_1}}\} E\{e^{-r_{n_2}}\} & \text{if } n_1 \neq n_2 \end{cases} \\
&= \begin{cases} R_z[0] c_1 & \text{if } n_1 = n_2 \\ R_z[n_2 - n_1] c_2 & \text{if } n_1 \neq n_2 \end{cases} \\
R_e[m] &= \begin{cases} R_z[m] c_1 & \text{if } m = 0 \\ R_z[m] c_2 & \text{if } m \neq 0 \end{cases} \\
&= R_z[m](c_2 + \delta[m](c_1 - c_2)).
\end{aligned}$$

In the above equations, c_1, c_2 are constants defined by $c_1 \triangleq E\{e^{-2r_{n_1}}\}$ and $c_2 \triangleq E\{e^{-r_{n_1}}\}E\{e^{-r_{n_2}}\}$. Observe that e_n is also a wide-sense stationary process (in fact it is also necessary to show e_n has a constant mean which is an easy task). The power spectral density of e_n can be calculated by taking the Fourier transform of its autocorrelation function $R_e[m]$. Then,

$$\begin{aligned} S_e(\omega) = \mathcal{F}\{R_e[m]\} &= \mathcal{F}\{R_z[m](c_2 + \delta[m](c_1 - c_2))\} \\ &= c_2 S_z(\omega) + (c_1 - c_2)R_z[0]. \end{aligned}$$

From the above description, we see that multiplying z_n by e^{-r_n} has the effect of scaling $S_z(\omega)$ and adding a constant function to it. If the input is a sinusoid, $S_z(w)$ will only have the fundamental harmonic. Then, this process will produce no additional harmonics, but instead, it will increase the noise floor in the spectrum. In order to check this analysis, we simulate a DAC using Equation 3.2, with the replacement of $r_n = 1/RC(x[n])$ with a white Gaussian random process. We assume the DAC update rate to be $f_u = 1/T_u = 100$ MHz. For the input, we use a sampled cosine at 13 MHz, $x[n] = cos(2\pi\frac{13}{100}n)$. The output spectrum is seen in Figure 3.10. The spectrum is clean in terms of harmonics, but the overall noise floor is definitely increased which is consistent with our theoretical analysis.

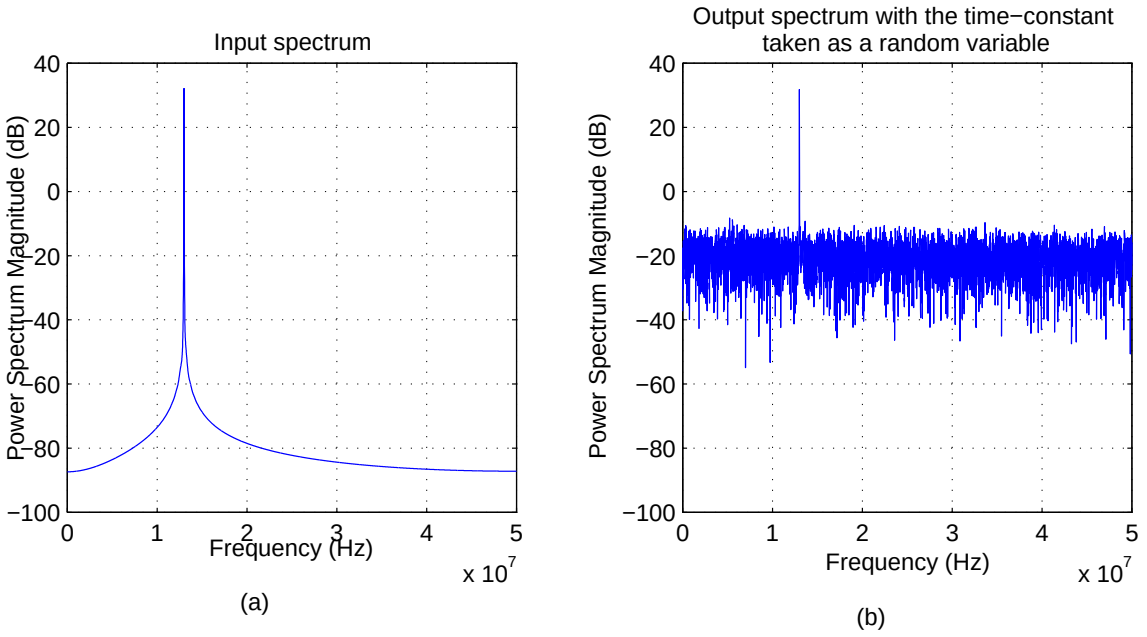

Figure 3.10. Input (a) and output (b) signal power spectra for a DAC simulated according to Equation 3.2. $1/RC(x[n])$ is replaced by a white Gaussian random process r_n. The input is a cosine signal at 13 MHz and the DAC update rate is 100 MHz.

3.4.2 $C(x)$ as a deterministic function

In the above discussion, we observed that if the output capacitance is a random variable independent of the input, the only effect on the input spectrum is in terms of adding some constant noise. This is consistent with the claim that nonlinearity comes from the dependence of the slewing on the input. Here, we explore a special case of this dependence where C is a deterministic function of x.

In order to make our analysis look better we first define the function

$$f(x) = \frac{T_u}{RC(x)}. \tag{3.6}$$

This implies the following relation for the error signal:

$$\begin{aligned} e[n] &= z[n]e^{-f(x[n-1])} \\ &= (x[n-2] - x[n-1])e^{-f(x[n-1])}. \end{aligned} \tag{3.7}$$

Here, it should be noted that our focus is on how this relation affects nonlinearity. The main question regarding the relation in Equation 3.7 is: How does a signal change when it is multiplied by an exponential function which has the signal in its exponent? To answer this question, it is much more convenient to work on the following continuous signal. The discussion that follows can be easily extended to the discrete-time case. Let $e(t)$ be defined as:

$$e(t) \triangleq x(t)e^{-f(x(t))}. \tag{3.8}$$

We need to find the relation between $E(f)$ and $X(f)$. If we define $g(t) \triangleq f(x(t))$, and use Taylor series expansion,

$$\begin{aligned} e(t) &= x(t)e^{-f(x(t))} \\ &= x(t)e^{-g(t)} \\ &= x(t)(1 - g(t) + \frac{g^2(t)}{2} - \frac{g^3(t)}{3!} + ...). \end{aligned}$$

In order to understand what happens in the spectrum we need to know what $g(t)$ is. For example if $g(t) = x(t)$,

$$\begin{aligned} e(t) &= x(t)(1 - g(t) + \frac{g^2(t)}{2} - \frac{g^3(t)}{3!} + ...) \\ &= x(t)(1 - x(t) + \frac{x^2(t)}{2} - \frac{x^3(t)}{3!} + ...) \\ &= x(t) - x^2(t) + \frac{x^3(t)}{2} - \frac{x^4(t)}{3!} + ... \quad . \end{aligned}$$

Looking at this relation above, if $x(t)$ is a sinusoid, $E(f)$ will contain all the harmonics with decreasing amplitudes. Now, let us look at another $g(t)$ definition. This time suppose $g(t) = cos(x(t))$.

$$\begin{aligned} e(t) &= x(t)(1 - g(t) + \frac{g^2(t))}{2} - \frac{g^3(t)}{3!} + ...) \\ &= x(t)(1 - (1 - \frac{x^2(t)}{2} + \frac{x^4(t)}{4!} + ...) + \frac{1}{2}(1 - \frac{x^2(t)}{2} + \frac{x^4(t)}{4!} + ...)^2 + ...). \end{aligned}$$

If $x(t)$ is a sinusoid, it is obvious that this time we will not see the even harmonics in $E(f)$. In order to check the validity of these statements, we repeat the simulation in Section 3.4.1 with the only change that we replace the random variable r_n with $g[n] = f(x[n])$. The results are given in Figures 3.11 and 3.12. The spectrum plots reveal that when we have $f(x) = x$, we see all the harmonics, while $f(x) = cos(x)$ does not excite the even numbered harmonics. These are all consistent with our theoretical findings.

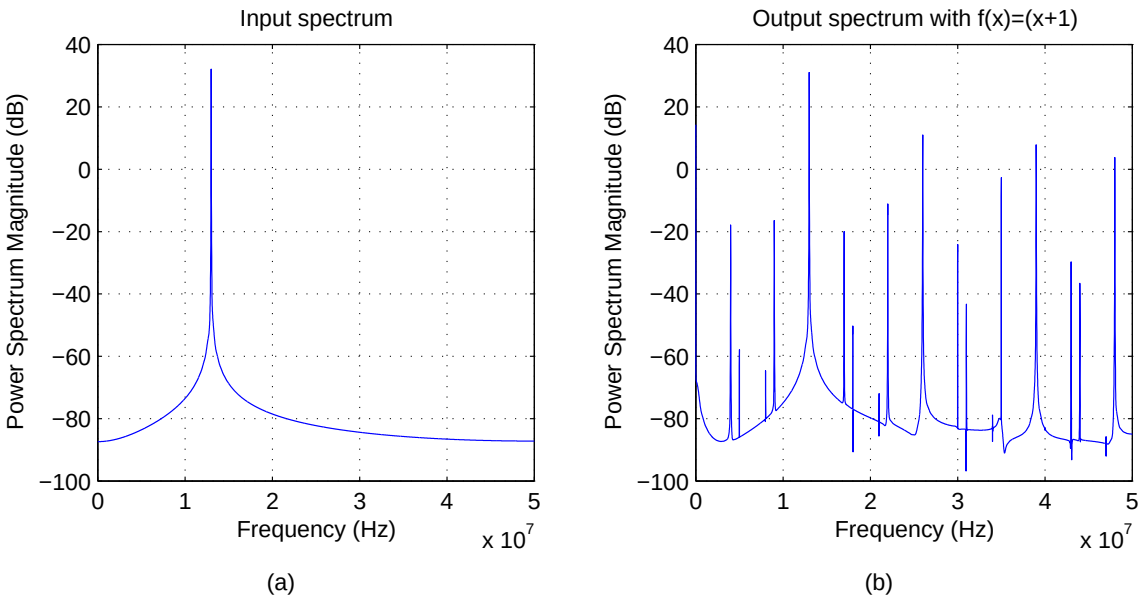

Figure 3.11. Input (a) and output (b) signal power spectra for a DAC simulated according to Equation 3.2. $1/RC(x[n])$ is replaced by $x[n] + 1$ (not $x[n]$, because it would have produced negative time constants). The input is a cosine signal at 13 MHz and the DAC update rate is 100 MHz. Observe that there are many harmonics present with changing amplitudes.

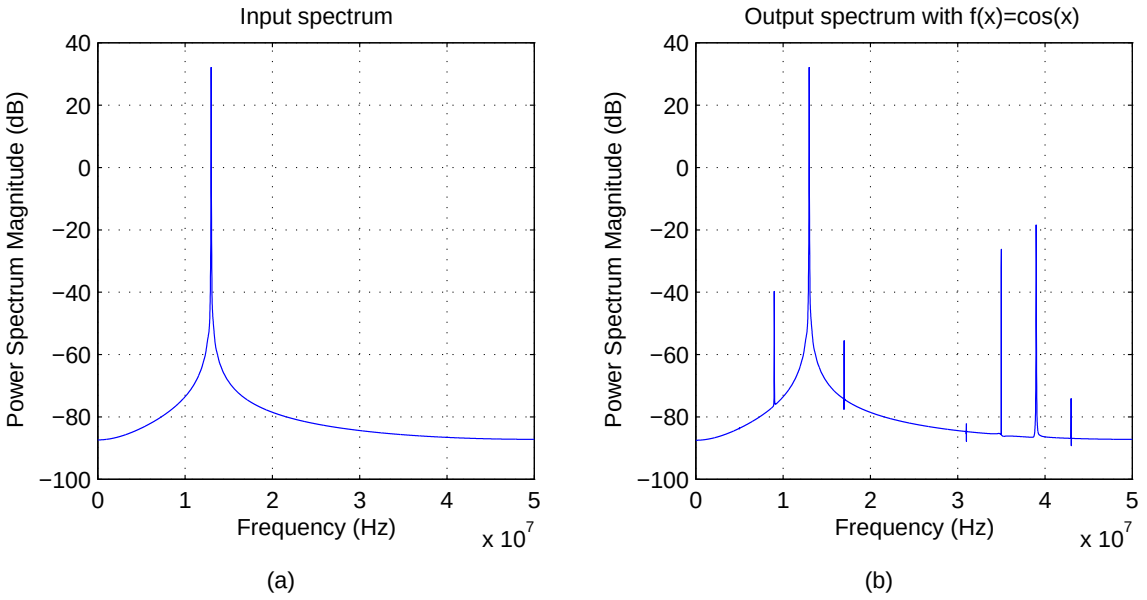

Figure 3.12. Input (a) and output (b) signal power spectra for a DAC simulated according to Equation 3.2. $1/RC(x[n])$ is replaced by $cos(x[n])$. The input is a cosine signal at 13 MHz and the DAC update rate is 100 MHz. Observe that even numbered harmonics are not present.

3.4.3 $C(x)$ increasing with $|x|$

In the previous sections, we studied what could happen for different definitions of the time constant in Equation 3.2. These cases, covered in Section 3.4.1 and 3.4.2, involve general definitions and the results presented give an idea about of what type of distortion each scenario would lead. If we want to see what a realistic error function could cause, we should go back to our circuit model and try to relate $C(x)$ to x in a reasonable way.

Consider the circuit models given in Figures 3.6 and 3.7. The capacitance $C(x)$ in Figure 3.7 is a combination of the capacitances present in the DAC circuit. If we assume that the only capacitances in the DAC are the output capacitances of the current sources (denoted by C_i's in Figure 3.6) and the load capacitance (denoted by C), we can write $C(x)$ in terms of these capacitances. Before doing that, we first neglect the resistances of the current sources (R_i's) and switches (R_{s_i}'s). This makes the equivalent circuit a very simple one, where all the capacitances come in parallel. Of course, for every digital input, there is a different equivalent circuit due to varying combinations of open and close switches. Let us define the set of indices of the closed switches by $S = \{i|$ Switch i is closed $\}$. Then we can write $C(x)$ as:

$$C(x) = C + \sum_{i \in S} C_i. \tag{3.9}$$

Here we defined $C(x)$, but we are not done yet to start our nonlinearity analysis. The last thing remaining is to relate the digital input to the set S. A first observation may be that we should have more switches closed for higher amplitude inputs. This would be a true assumption if we assume all the current sources in Figure 3.6 to be unit current sources and each current source is controlled by a switch. This assumption would be true for thermometer DACs, where the number of switches and unit current sources are both equal to the number of input levels [7].We further assume the output capacitance of the current sources to be equal, i.e. $C_i = C_0$, $\forall i$. If we assume the

DAC to be unity-gain, then $C(x)$ can be written as $C(x) = C + C_0\frac{|x|}{LSB}$. Here, LSB is the amount of change in the output value corresponding to one bit change in the input (note that x is a quantized value).

Now we refer to Equation 3.8. In the previous discussion, we had found the relation between the output distortion and the time-constant by using this error signal definition. In this specific case,

$$f(x(t)) = \frac{1}{R(C + C_0\frac{|x(t)|}{LSB})} \quad . \tag{3.10}$$

Remember that the Taylor expansion of $f(x(t))$ had helped us to understand what type of distortion we would have in the output spectrum. When the input to the DAC is a sinusoid, based on what powers are present in the Taylor approximation of $e(t)$, we can predict the harmonics that would be seen in the spectrum. We now repeat the same analysis here. Note that $f(x(t))$ in Equation 3.10 is an even function. This means the Taylor expansion of $f(x(t))$ will not contain odd powers.

$$f(x(t)) = a_0 + a_2x^2(t) + a_4x^4(t) + ... \quad . \tag{3.11}$$

Then

$$\begin{aligned} e(t) = x(t)\Big(1 - (a_0 + a_2x^2(t) \quad &+ \quad a_4x^4(t) + ...) \\ &+ \quad \frac{1}{2}(a_0 + a_2x^2(t) + a_4x^4(t) + ...)^2 - ...\Big) \quad . \end{aligned}$$

As can be seen, $e(t)$ does not contain any even powers of $x(t)$. Like in the case analyzed in Section 3.4.2, where $f(x) = cos(x)$, the even numbered harmonics will not be present in the output spectrum. As a conclusion, we were able to do the nonlinearity analysis for a more realistic situation. This analysis can be repeated for

less simplified cases, e.g. with the switch and current sources resistances also taken into account.

We check this analysis with a DAC simulation. As in previous sections, the input signal is a cosine at 13 MHz. The 10-bit DAC is assumed to be have an update rate of $f_u = 100$ MHz. We assume the circuit elements to have values $R = 70\,\Omega$, $C = 20$ pF and $C_0 = 5$ fF. The input and output signal spectra are given in Figure 3.13. The simulation results agree with our calculations as the output spectrum does not have even numbered harmonics.

Considering the simulation results in this chapter, it is important to note that, we worked on *approximate* functions in our theoretical analyses but used the *exact* model in simulations. For example, in both the random and the deterministic case, we used Equation 3.3 instead of Equation 3.2 for simplification. Further, we neglected the fact that the error signals were written in terms of the *modified* input sequence, $x[n-2]-x[n-1]$, not the exact input sequence, $x[n]$. These simplifications -although they do change the output signal- were not expected to disturb the nonlinearity analysis. For example, if an operation on $x[n]$ does not produce any harmonics, then the same should be true for the operation on $x[n-2]-x[n-1]$ as well. On the other hand, we used the exact model given by Equation 3.2 in the simulations. The simulation results justified our assumption that the nonlinearity analysis still remains to be correct despite the above-mentioned simplifications.

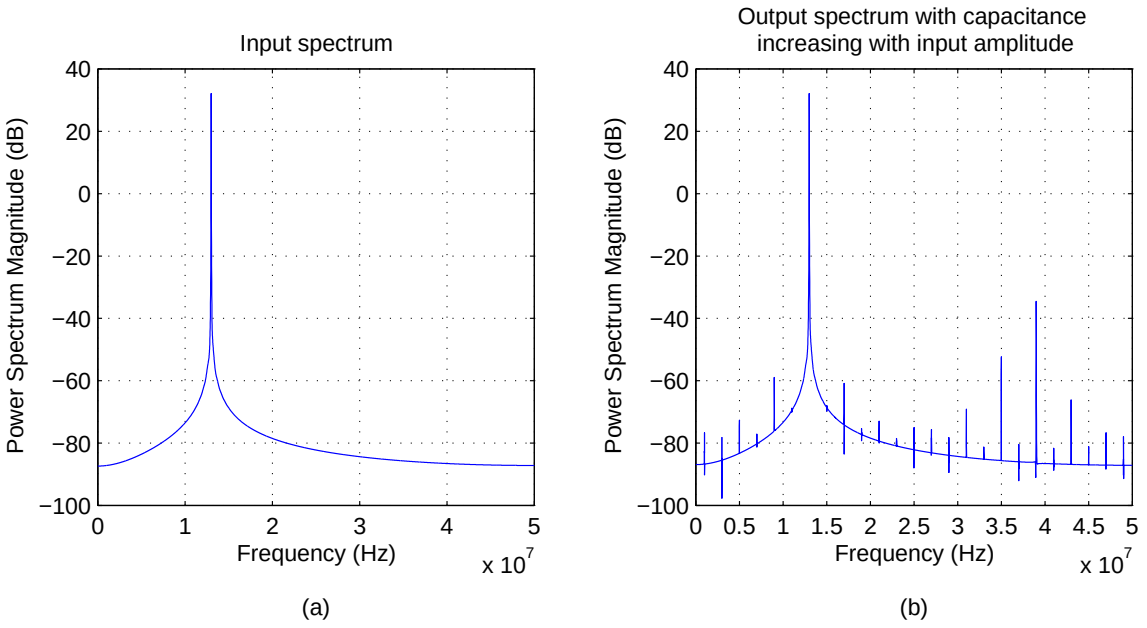

Figure 3.13. Input (a) and output (b) signal power spectra for a 10-bit DAC simulated according to Equation 3.2. $C(x[n])$ is replaced by $C + C_0 \frac{|x[n]|}{LSB}$ where $R = 70\Omega$, $C = 20$ pF and $C_0 = 5$ fF. The input is a cosine signal at 13 MHz and the DAC update rate is 100 MHz. Even harmonics are not present.

3.5 Summary

In this chapter, we present our work on digital-to-analog converter (DAC) linearization. We attempt to achieve a more linear input-output relation by modifying the input to a DAC. This method, called predistortion, requires a good understanding of the DAC errors contributing to the nonlinearity. We present the most prominent DAC errors by explaining their effects on the DAC output. In order to be able to modify the input accordingly, a DAC model needs to be adopted. Based on the most dominant of the DAC errors, we choose a simple DAC model and we explain how it is extracted from the actual DAC circuit. This model gives us a mathematical expression of the input-output relation which can be used to understand how the nonlinearity occurs. The error function, i.e. how the error depends on the input signal, determines the type of the nonlinearity observed at the output. Through theoretical analyses and computer simulations, we investigate how different error function definitions lead to different types of nonlinearities. We present our results in terms of frequency spectrum calculations.

CHAPTER 4

CONCLUSION

We present our work on two research projects. In the radar target classification project, we tried to enhance the detection performance of car radars. Our aim was to find the best signal to send to maximize classification between small and large objects. We obtained the results given in Chapter 2. Having these results in hand, we have the following conclusions.

Once given a performance measure, it is a rather trivial task to compare the signals from the available set. However, the essential part of the problem is the modeling to define this measure. As long as our model assumes two distinct impulse responses representing targets of concern and targets to ignore, we are able to find the best signal to send. As the constraints on the available signals get more relaxed we are able to achieve better classification performance. In addition, using a different signal for correlation helps us improve this performance. These findings suggest changes on the car radar system for better detection.

Once we leave the assumption of two distinct impulse responses, the problem gets more complicated. Now we have to do classification between classes of targets. Our objective becomes maximizing the worst case performance. This makes the problem a minimax robust optimization problem. This type of optimization is more difficult and needs special conditions to be easily solved. However, because of the lack of a model of the physical targets, we are not able to check whether these conditions are satisfied. Measurements on real targets are needed to be done to model the target

behavior for small and large objects. This way, we can make a more reliable choice of signal.

For the second project, we worked on a predistorter design for digital-to-analog converters (DACs). Although the main goal is to eliminate nonlinearities of DACs, the big challenge showed itself in understanding why these nonlinearities happen in the first place. As the predistorter design significantly depends on the DAC model assumed, a crucial first step has to be finding a behavioral model of a DAC. Unfortunately, a DAC is not easy to model. DAC errors causing nonlinearity originate from a wide range of factors and even a single selected type of error is very hard to model. So, we resorted to models of DAC errors we found to be most dominant. We investigated how this model would affect a given input signal by several theoretical and simulated analyses. These analyses constitute our main contribution in this project. We observed that the type of dependence the errors have on the input significantly affects how the nonlinearity shows itself.

BIBLIOGRAPHY

[1] S. M. Sowelam and A. H. Tewfik, "Waveform selection in radar target classification," *IEEE Transactions on Information Theory,* vol.46, no.3, pp. 1014-1029, May 2000.

[2] P. J. Huber, "A Robust Version of the Probability Ratio Test," *The Annals of Mathematical Statistics,* vol. 36, no. 6, pp. 1753-1758, Dec 1965.

[3] S. Verdu and H. V. Poor, "On minimax robustness: A general approach and applications," *IEEE Transactions on Information Theory*, vol.30, no.2, pp. 328-340, Mar 1984.

[4] V. V. Veeravalli, T. Basar, and H. V. Poor, "Minimax robust decentralized detection," *IEEE Transactions on Information Theory,* vol.40, no.1, pp. 35-40, Jan 1994.

[5] J. J. Wikner, "Studies on CMOS Digital-to-Analog Converters" *Linköping Studies in Science and Technology,* Dissertation No. 667, Linköping, 2001.

[6] P. Hendriks, "Specifying communications DACs" *IEEE Spectrum,* vol.34, no.7, pp. 58-69, July 1997.

[7] B. Razavi, *Principles of Data Conversion System Design.* New York: IEEE Press, 1995.

[8] K. O. Andersson and J. J. Wikner, "Characterization of a CMOS current-steering DAC using state-space models," in *Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems,* vol. 2, no., pp. 668-671 vol.2, 2000.

[9] K. O. Andersson,"Studies on Performance Limitations in CMOS DACs," *Linköping Studies in Science and Technology,* Thesis No. 976, Linköping, Nov. 2002.

[10] AD9776A/AD9778A/AD9779A, Datasheet, Analog Devices, Norwood, MA, 2007.

[11] F. H. Irons, D. M. Hummels, and S. P. Kennedy, "Improved compensation for analog-to-digital converters," *IEEE Transactions on Circuits and Systems,* vol.38, no.8, pp. 958-961, Aug. 1991.