NTNU
Det skapende universitet

Digital signalrekonstruksjon for tidsmultipleksa analog-til-digital konvertere

Gard Wold Sætre

Master i elektronikk
Innlevert: Juli 2012
Hovedveileder: Nils Holte, IET
Medveileder: Sigve Tjora, Hittite

Norges teknisk-naturvitenskapelige universitet
Institutt for elektronikk og telekommunikasjon

Problem Description

In very high bandwidth applications, such as wireless communication systems and instrumentation, analog-to-digital converters (ADC's) need to operate at extremely high sampling rates. To achieve such speeds with today's integrated circuits (IC) processing technologies are rarely feasible. A solution to this is to time-interleave several converters, allowing each one to run at a lower rate. A time-interleaved ADC (TIADC) is as such a polyphase realization of the analog to digital (AD) conversion. If the interleaved ADC cores are identical, the polyphase decomposition is lossless, but in practice there will be offset-, gain- and timing-mismatches between them, causing unwanted distortion. In particular timing mismatches are difficult to detect and compensate. Promising research has been made into fully digital compensation of timing mismatches, using digital estimators and adaptive filter banks to reconstruct the ideally matched scenario. The assignment is divided into two main parts which are:

1. To study existing literature and published solutions for signal reconstruction in the precence of timing mismatches in high performance TIADC's.
2. To develop a Matlab framework for modeling TIADC mismatches, and use this for analysis of different calibration schemes.

Preface

This thesis presents the work carried out in my final semester at NTNU, in the field of *digital communications*, at the Institute of Electronics and Tele-communication (IET). The assignment was to implement a method for signal reconstruction from non-uniform samplers.

First off I would like to thank my family for all the help and support they have given me through all my years as a student. Thank you very much!

I would like to thank my supervisor Nils Holte from IET for his patience, help and direction on the thesis.

I would also like to thank Morten Flå, Irja W. Sætre and Bjarte L. Nystøyl for proof-reading and help on my layout, spelling and inability to finish a sentence before starting a new one.

Lastly I would like to thank my fraternity St. Omega, especially all the users of 'Loftet' who have made the last four years really fun!

Abstract

As digital communication requires faster and faster systems to handle the increased bandwidth and required sampling rate, using a single ADC might not be sufficient. One solution to this problem is to use TIADCs which eliminates the constant need for faster and more accurate ADCs. When using TIADC the overall system speed is equal to the combined speed of all the individual ADCs.

In this report, the reconstruction of a band limited signal from a 2-TIADC and a 4-TIADC system, assuming periodic nonuniform sampling, when the timing-mismatch is known, has been shown. The chosen method is based on multirate filter banks, where the filter coefficients are found by interpolation with a raised cosine function. The chosen input-signal was a sine signal with a frequency of 350 MHz, and the sampling frequency was set as 1 GHz. The filter-lengths are defined by k and the intervall is defined as $[-2k-1, 2k]$.

The simulation results show that the reconstruction of the input signal from a 2-TIADC system, can with a k varying from 10 to 45, depending on the roll off factor, be achieved within an error limit of -100dB (MSE). The results for the 4-TIADC system showed an increased error and thus, an increased k is needed to achieve a reconstruction within the limit. The results indicated that k had to be increased to have a value between 15 and 50.

The 2-TIADC setup was also tested to see how it handled an increasing timing-mismatch ranging from 10-99% of T_s with k chosen as 25, 50 and 100. The results showed that reconstruction within the error limit could be achieved for a timing-mismatch of 0.7, 0.85 and 0.95% T_s respectively.

Sammendrag

Siden digital kommunikasjon krever raskere og raskere systemer for å kunne håndtere økende behov for båndbredde og samplingsrate, kan systemer med èn ADC være utilstrekkelig. En løsning på dette problemet er å bruke TIADC systemer, som kan eliminere nødvendigheten til å stadig utvikle raskere og mer nøyaktige ADCer. Dette siden den totale hastigheten til et TIADC system er lik summen av de individuelle ADCene.

I denne oppgaven har signalrekonstruksjon av et båndbegrenset signal fra et 2-TIADC og et 4-TIADC system, med antakelsene ikke-uniform periodisk sampling og kjent tidsforskyvning blitt vist. Den valgte metoden er basert på multirate filterbanker, hvor filterkoeffisientene blir funnet via interpolasjoen med en *raised cosine funksjon*. Det valgte inngangssignalet er et sinus signal med frekvens på 350 MHz, og samplingsfrekvensen ble valgt til 1 GHz. Lengden på de digitale filtrene blir bestemt av k og intervallet definert som $[-2k-1, 2k]$.

Simuleringsresultatene viste at inngangssignalet fra 2-TIADC systemet kan, med k valgt mellom 10 til 45 avhengig av roll-off faktoren, rekonstrueres med feil innen den satte feilgrensen på -100dB (MSE). Resultatene for 4-TIADC systemet viste seg å ha større feil, og for å kunne oppnå en rekonstruksjon innen feilgrensen, må k økes til mellom 15 og 50 avhengig av roll-off faktoren.

Systemet for 2-TIADC ble også testet for å se hvordan metoden håndterte en økende tidsforskyvning. Tidsforskyvningen varierte fra 0.1-0.99% av T_s, og verdiene av k som ble testet var 25, 50 og 100. Resultatene viste at inngangssignalet kan rekonstrueres med en tidsforskyvning på henholdsvis 0.7, 0.85 og 0.95 T_s med en feil innen den valgte grensen.

Contents

List of Figures

List of Tables

Abbreviations

AD	Analog to Digital
ADC	Analog to Digital Converter
CT	Continous Time
DSP	Digital Signal Processing
DT	Discrete Time
Fs	Sampling Frequency
IC	Integrated Circuits
IR	Impulse Response
PR	Perfect Reconstruction
SNDR	Signal To Noise and Distortion Ratio
SNR	Signal to Noise Ratio
TIADC	Time Interleaved ADC
Ts	Sampling Period

Chapter 1

Introduction

As communication is getting faster and faster, the demand for high-speed wireless digital communication increases. The rate of communication systems are limited by the analog-to-digital converters (ADCs) and their sampling frequency and accuracy. As the analog limit is pushed forward, problems arise, as the use of a single ADC might not meet either the required sampling speed, or the required efficiency with respect to cost and power usage. A solution to this is the usage of time interleaved analog to digital converters (TIADCs), first introduced by Black and Hodges in 1980 [2]. However, TIADCs introduce new errors which are *gain-*, *offset-* and *timing mismatch*. Because of the high sampling rate in such systems, the timing mismatch has the biggest impact and offers the biggest challenge to correct.

In this thesis, TIADCs will be explored and a method for reconstructing a signal from its non-uniform samples will be presented. The correction of errors introduced when using TIADCs can be done both analog or digital. This thesis will focus on correction with the use of digital signal processing (DSP). In order to perform the correction, the timing-mismatch has to either be known beforehand, or estimated. Different methods for estimating this, can be found in [4][6][9][13].

There exist many and various methods for signal reconstruction when the timing mismatch is known, or estimated beforehand. Some of these methods can be found in [4][7][10][11]. In relation to this thesis, the method presented by Prendergast, Levy and Hurst is the most essential.

The work presented in [7] proposes a method for signal reconstruction from non-uniform periodic samplers using multi rate filter banks, where the reconstruction is performed using poly-phase matrices. One of the essential similarities between this method and the one presented in this thesis, is how the coefficients for the reconstruction filters are found. Both methods uses an interpolation technique to recreate the sampled signal. In the method proposed by Prendergast et. al. a *sinc-function* together with a window function was used, which differs from the *raised cosine interpolation* used in this thesis.

In this thesis the timing-skew is assumed to be known and the focus will be 'black-box signal reconstruction' from 2-TIADC and 4-TIADC systems using a method based on interpolation functions as filters. The analysis of the implemented method will focus on the accuracy of the reconstructed signal compared to a required minimum error.

1.1 Thesis outline

The thesis starts with a short introduction to the theme in chapter 1, which also gives a short background of previous work in this field. Chapter 2 presents relevant theory, starting with the basics of sampling theory and ADCs and continues with the more advanced theory for TIADCs. This chapter ends with signal reconstruction theory for 2-TIADC and M-TIADC assuming non-uniform periodic sampling and known timing-mismatch. Chapter 3 shows block diagrams to explain the different parts of the reconstruction method. It also explains some differences between the theory and the implemented code. In chapter 4 the simulation results are presented and commented. The last chapter presents a conclusion of the chosen method and topics for future work.

Chapter 2

Theory

This chapter will first explain the basics of signal sampling and ADCs, and continue to the advanced topic of TIADCs. After that, timing-mismatch will be presented and explained. The two last sections will focus on theory for signal reconstruction from non-uniform periodic samplers. The reconstruction theory will start with signal reconstruction for a 2-TIADC system and then generalize into M-TIADC.

2.1 Signal Sampling and digitalizing

Analog signals are a part of the everyday life. These are electromagnetic waves with varying frequency and amplitude. We use them in cell phones, wifi, radio and many more applications. The reason for using analog signals is that we cannot transfer digital signals directly between devices i.e, from a wireless router to a computer.

The information we want to send is attached to a carrier wave through modulation and then sent from the transmitter to the receiver. In the receiver, the signal is demodulated and the wanted information is acquired from the signal with the use of an ADC. The general principle of how an ADC works can be seen in Figure 2.1[8].

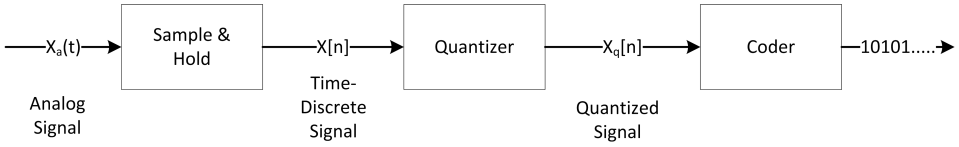


Figure 2.1: The basic steps of an ADC.

Signal sampling is the first step in turning an analog signal into a digital signal. In this stage a continuous-time signal x(t) is turned into a discrete-time (DT)

signal x[n], by taking a sample every T_s seconds. When assuming periodic uniform sampling, the sampled DT signal can be written as

$$x[n] = x(nT_s), \text{ for } -\infty < n < \infty \tag{2.1}$$

where T_s is defined as $1/F_s$, and F_s is the sampling frequency [8].

To be able to reconstruct the original signal from the discrete-time signal, the sampling period T_s has to fulfill the Shannon-Nyquist theorem to avoid aliasing. Aliasing refers to an effect that causes different signals to become indistinguishable. This theorem gives us the lowest possible sampling frequency, depending on the input-signal frequency, when assuming uniform and periodic sampling [8].

$$F_s > 2B = 2 * f_{max} \tag{2.2}$$

where B is the bandwidth of the signal, found by f_{max} which is the highest frequency in the input-signal.

The second step in the creation of a digital signal is *quantization* of the time-discrete signal. The quantization of a sampled signal converts the values from a infinite to a finite number of digits [8].

$$x_q[n] = Q[x[n]] \tag{2.3}$$

The error introduced when representing the continuous-valued signal as a finite set of discrete values is called quantization error. The size of the error, $e_q[n]$, is dependent on the chosen step-size, Δ, as shown in equation 2.4 [8].

$$-\frac{\Delta}{2} \geq e_q[n] \leq \frac{\Delta}{2} \text{ where } e_q[n] = x_q[n] - x[n] \tag{2.4}$$

The final stage in the digitalization of an analog signal is assigning a unique binary number to each quantization level. When having L levels we need at least L different binary numbers. With a sequence length of b bits, we can create 2^b different binary numbers. This gives us the minimum number of bits required, which is the smallest integer greater or equal to $log_2(L)$ [8].

2.1.1 Analog-to-Digital Converters errors

When using ADCs the input signal is subject to different types of errors. For a single ADC setup these errors can be summarized into three categories, which are *sampling jitter*, *quantization noise* and *non-linearities* [8].

Using a single ADC has its advantages, but the overall system speed is limited by the ADCs sampling rate. The advantages of using a single ADC is the fact that

ADCs have been used for a relatively long time, and thus the former mentioned errors have been thoroughly investigated and solutions exists to correct them. Because of this the errors will not be discussed further.

2.2 Signal Reconstruction

As previously mentioned, an input signal $x(t)$ can be recreated from its sampled state $x[n]$ if the sampling frequency is above the Shannon-Nyquist limit. This property is important as it enables for the ability to check that the sampled signal is correct.

Perfect reconstruction can be achieved if the reconstructed signal $\hat{x}$ can be written as shown in equation 2.5 [7].

$$\hat{x}[n] = ax[n - d] \tag{2.5}$$

In the equation above a is a non-zero integer and d is an integer. Signal reconstruction can be performed through the use of interpolation, which will be explained in the following section.

2.2.1 Interpolation

Interpolation in the traditional sense can be explained as upsampling, where zeros are inserted in between the original values in the signal and thus increasing the sampling rate of the signal.

$$x_{inter}[n] = x[nT_s/I] \tag{2.6}$$

Where I is the interpolation factor. Interpolation is often used in combination with an interpolation function to approximates values for the zeros by using the known values and the properties of the interpolation function. This property of an interpolation function makes it ideal for recreating signals. When assuming uniform sampling a signal can be reconstructed by interpolating the signal $x[n]$ with an interpolation function $g[n]$ sampled at the zero crossings. Within DSP this is equivalent, in the time domain, to convolution and the reconstructed signal can be written as shown in equation 2.7 [8].

$$x_{recon}[n] = x[n] * g[n] = \sum_{k=-\infty}^{\infty} x[k]g[n - k] \tag{2.7}$$

There exists many different interpolation functions with different properties, depending on both the input-signal and the properties of the output-signal. Two of

the most important factors in a real time communication system are delay through the circuit and the required bandwidth. Choosing a fitting interpolation function can minimize the delay and bandwidth. Two of these functions are *sinc* and *raised cosine function* which will be explored in the following sections.

Sinc function

The *sinc* function has the properties of an ideal low-pass filter, and is defined in the time domain as described in equation 2.8[1].

$$g_{sinc}(t) = \frac{sin(\pi t/T)}{\pi t/T} \tag{2.8}$$

The ideal low-pass properties can better be seen when transforming $g_{sinc}(t)$ to the frequency domain function $G_{sinc}(f)$

$$G_{sinc}(f) = \begin{cases} 1 & |f| \leq f_c \\ 0 & otherwise \end{cases} \tag{2.9}$$

where f_c is the cut-off frequency. A common problem with sinc is that it is defined from $-\infty$ to ∞ and thus the impulse-respons, $g_{sinc}(t)$, has an infinite length. In implementations, the length of the function is limited to an interval $[-M, M]$ which introduces edge effects caused by the side lobes. The side lobes of sinc decays at a rate $\propto \frac{1}{t}$. The error introduced by these edge effects can be limited by choosing a sufficiently large M. Due to this fact, the sinc function is not suited in this implementation because of the extra delay the filter length introduces.

Raised cosine function

The raised cosine function is also defined from $-\infty$ to ∞, but the decay of the side lobes is faster. This increased decay minimizes the errors due to edge effect, and thus a shorter filter length is required. In the time domain it is defined as shown in equation 2.10 [1].

$$g_{rcos}(t) = \left(\frac{sin(\pi t/T_s)}{\pi t/T_s}\right)\left(\frac{cos(\pi \alpha t/T_s)}{1-(2\alpha t/T_s)^2}\right) \tag{2.10}$$

α is the excess bandwidth and has a value between 0 and 1. For α=0 the raised cosine function equals the sinc function. When transforming $g_{rcos}(t)$ to the frequency domain it can be written as shown in equation 2.11 [1].

$$G_{rcos}(f) = \begin{cases} T & , |f| \leq \frac{1-\alpha}{2T} \\ Tcos^2\left[\frac{\pi T}{2\alpha}\left(|f| - \frac{1-\alpha}{2T}\right)\right] & , \frac{1-\alpha}{2T} < |f| \leq \frac{1+\alpha}{2T} \\ 0 & , \frac{1+\alpha}{2T} < |f| \end{cases} \tag{2.11}$$

The side lobes of $g_{rcos}(t)$ decays with a rate $\propto \frac{1}{t^3}$. The difference between the *sinc* and *raised cosine* filters can be seen in Figure 2.2.

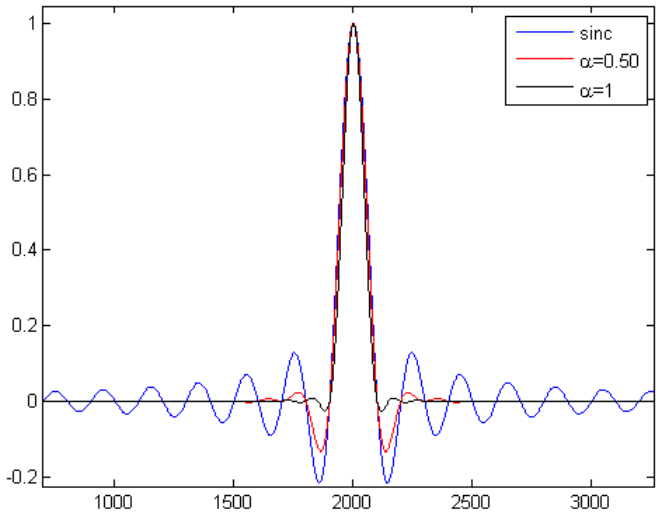


Figure 2.2: The difference between the impulse responses for sinc and raised cosine functions.

From Figure 2.2 one can see the difference in decay of the side lobes between the sinc and the raised cosine filter with varying roll-off factor, α.

2.3 Principle of Time Interleaved Analog-to-Digital Converters

The main principle of TIADCs, is the use of two or more ADCs on the same input signal. The reason for this is to create a fast ADC system with the use of slower individual ADCs. A generalized setup can be seen in Figure 2.3 [3].

The goal is to achieve a required sampling rate, F_s, by using multiple ADCs. The sampling rate of each of the sub-ADCs is F_s/M. When multiplexing the samples from each ADC the signal will be in the correct order. Thus the overall system achieves a required sampling rate with the use of slower individual ADCs.

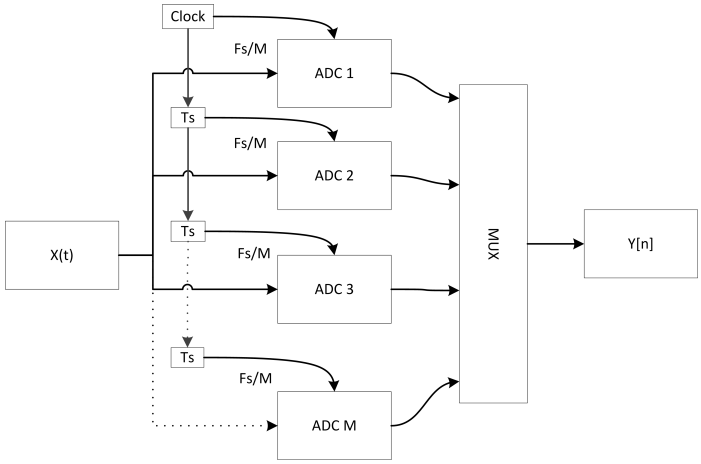


Figure 2.3: The general setup for use of TIADCs.

The problem with using TIADCs is the introduction of certain types of errors compared to only using a single ADC. These errors are in addition to the ones common when using a single ADC which were mentioned in chapter 2.1.1. In a TIADC system the signal is also affected by *gain-*, *offset-* and *timing-mismatch*. Where the latter one, because of the high sampling rate, has the biggest effect on the sampled signal and is the most difficult error to compensate for.

2.3.1 Timing skew

The basic effect of timing skew is that the input signal will not be sampled at the correct time. Instead of getting the correct $x[n] = x(nT_s)$, the actual sampled signal will be

$$x[n] = x(nT_s + \tau) \tag{2.12}$$

where τ is the timing-mismatch [3]. In a TIADC system the individual samplers can be expressed as

$$x_i[n] = x(MnT_s + \tau_i) \tag{2.13}$$

where M is the total number of samplers and i represents the individual samplers.

The effect of timing-mismatch in the frequency domain is the appearance of spurious components which decrease the *signal to noise and distortion ratio* (SNDR) [12]. These frequency components will appear periodically at the intervals shown in equation 2.14, where i represent the individual ADCs [6].

$$f_{noise} = \pm f_{in} + \frac{i}{M}, i = 0, 2, .., M - 1 \tag{2.14}$$

For a sinusoid signal in time domain, the effects of timing-mismatch can be seen in the amplitude of the sampled signal. The error will be largest where the slew-rate for the input signal is greatest. Slew-rate is defined as the rate of change in a signal [6]. For example, the slew-rate for a sine-signal is greatest at the zero-crossing. This is illustrated in Figure 2.4.

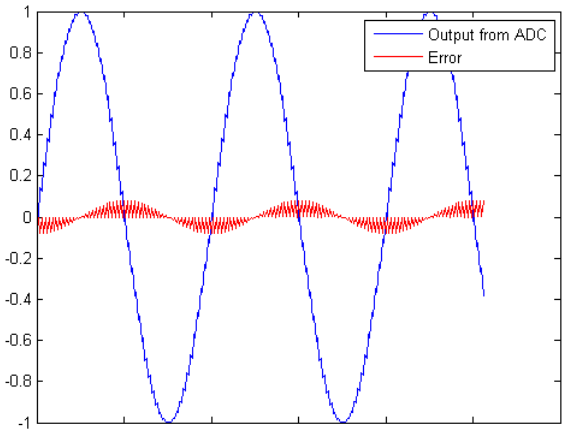


Figure 2.4: The error caused by timing-mismatch in time domain.

2.4 Multirate aspect

Another aspect of TIADC systems is the multirate aspect. This is a term used for systems with sample-rate-conversion, or in other words, systems where the sampling rate is changed at least once [8]. An illustration of a multirate system can be seen in Figure 2.5

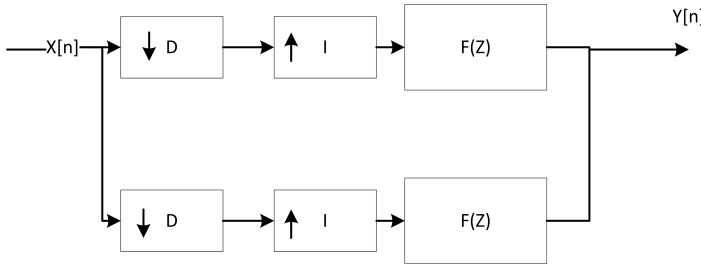


Figure 2.5: Illustration of a multirate system.

In Figure 2.5 the D stands for decimation (downsampling) and the I for interpolation (upsampling). The values for D and I can only be integers, but by utilizing both interpolation and decimation the sampling rate can be changed to a fractional number. For example by using D=2 and I=3 the final sampling rate will be, $F_{end} = \frac{3F_{start}}{2}$. When changing the sampling rate, the order of decimation and interpolation is important, as it is possible to end up with a signal which is sampled with a frequency below the Shannon-Nyquist limit.

In the TIADC case the sampled signals have the same frequency but different phases and thus, a TIADC system can be seen as a multirate system. For illustration see Figure 2.6.

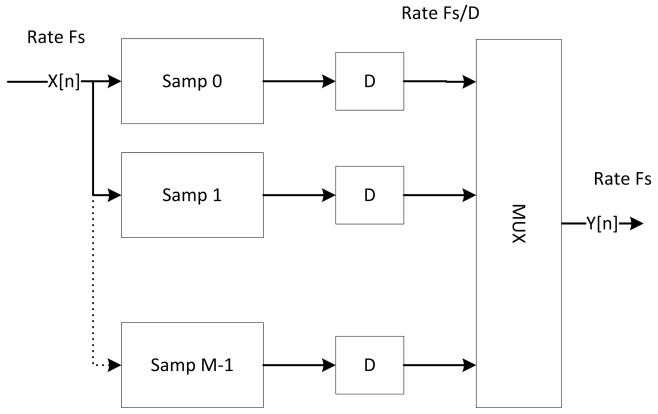


Figure 2.6: Illustration of the multirate aspect of TIADC.

2.5 Signal reconstruction for a 2-TIADC setup

The presented theory on signal reconstruction from a 2-TIADC system is based on the method by Nils Holte [5]. This section will explain the theory of a signal reconstruction method for a 2-TIADC system, assuming there is time-skew in one of the ADCs. The last part of this section will propose a generalized theory for signal reconstruction from M-ADC systems.

This method uses raised cosine interpolation (see section 2.2). The reason for this is to get a faster decay of the side lobes, compared to using sinc interpolation. This will shorten the required filter length and minimize the added signal delay through the reconstruction circuit.

The notation will be x for ideal sampled signal, y for non-ideal sampled signal and f for the interpolation function. Vectors will have an underline $\underline{F}$, and matrices will have a tilde beneath them, $\underset{\sim}{F}$.

In an extremely simplified way the method can be illustrated by the following logic. Assuming that $A = B * C$, then B can be found by $B = A * C^{-1}$.

Assuming that the interpolation function is limited in time to the intervall $t \in [-2k-1, 2k]$ and sampled at the zero-crossings, then the non-ideal samples can be recreated by using equations 2.15 and 2.16.

$$\begin{aligned} y[2n] &= \sum_{k=-2K-1}^{2K} x[2n-k]f(k) \\ &= \sum_{k=-2K}^{2K+1} x[2n+k]f(-k) = \underline{f}_{eR}^T \underline{x}_{2n} \end{aligned} \tag{2.15}$$

$$\begin{aligned} y[2n+1] &= \sum_{k=-2K-1}^{2K} x[2n-k]f(k+1+\tau) \\ &= \sum_{k=-2K}^{2K+1} x[2n+k]f(-k+1+\tau) = \underline{f}_{oR}^T \underline{x}_{2n} \end{aligned} \tag{2.16}$$

where

$$\underline{f}_{eR}^T = [f(2k), \cdots, f(0), \cdots, f(-2k-1)] \tag{2.17}$$

$$\underline{f}_{oR}^T = [f(2k+1+\tau), \cdots, f(1+\tau), \cdots, f(-2k+\tau)] \tag{2.18}$$

$$\underline{x}_{2n}^T = [x[2n-2k], \cdots, x[2n], \cdots, x[2n+2k+1])] \tag{2.19}$$

The notation e and o stands for even and odd respectively to differentiate between the odd and even part of the sampled signal. As shown above, the first step is to recreate the sampled signal, $y[n]$, by using an interpolation function as a filter.

This is done by convolving the ideal sampled signal, $x[n]$, with the interpolation function $f[k]$.

As equation 2.15 and 2.16 only recreates a part of the $y[n]$, we define extended vectors spanning $k \in [-2L, 2L+1]$ to account for the entire length of $y[n]$. When applying the extended vectors to the ideal x and the sampled y we get two extended vectors. The new vectors can be found by equation 2.20 and 2.21.

$$\underline{x}_{E,L}^T = [x[-2L], \cdots, x[2L+1]] \tag{2.20}$$

$$\underline{y}_{E,L}^T = [y[-2L], \cdots, y[2L+1]] \tag{2.21}$$

Another way of viewing the extended vector $\underline{y}_{E,L}$ is by using matrix formulation, thus transforming the reconstruction for the sampled signal as shown in equation 2.22.

$$\underline{y}_{E,L} = \underbrace{\begin{bmatrix} \underline{f}_{eR}^T & \underline{0} & . & \underline{0} \\ \underline{f}_{oR}^T & \underline{0} & . & \underline{0} \\ . & . & . & . \\ \underline{0} & \underline{0} & \underline{f}_{eR}^T & \underline{0} \\ \underline{0} & \underline{0} & \underline{f}_{oR}^T & \underline{0} \\ . & . & . & . \\ \underline{0} & . & \underline{0} & \underline{f}_{eR}^T \\ \underline{0} & . & \underline{0} & \underline{f}_{oR}^T \end{bmatrix}}_{F_{L,L+k}} \underline{x}_{E,L+k} \tag{2.22}$$

The matrix $F_{L,L+k}$ shown above contains all the information needed to recreate $y[n]$. Assuming this information is correct (yielding a $\hat{y}$ with error below a set limit), then $F_{L,L+k}$ also contains the necessary information to create the inverse interpolation functions needed to recreate the original signal, x. Another formulation of the $\underset{\sim}{F}$ matrix can be seen in equations 2.23 and 2.24.

$$\underset{\sim}{F}_{L,L+k} = \begin{bmatrix} \underline{F}_k & \cdots & \underline{F}_0 & \dots & \underline{F}_{-k} & \underline{0} & \underline{0} \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ \underline{0} & \underline{F}_k & \cdots & \underline{F}_0 & \cdots & \underline{F}_{-k} & \underline{0} \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ \underline{0} & \underline{0} & \underline{F}_k & \cdots & \underline{F}_0 & \cdots & \underline{F}_{-k} \end{bmatrix} \tag{2.23}$$

Where the individual sub-matrices of $\underset{\sim}{F}$ can be written as:

$$\underline{F}_k = \begin{bmatrix} f(k) & f(k-1) \\ f(k+1+\tau) & f(k+\tau) \end{bmatrix} \tag{2.24}$$

2.5.1 Least Mean Square formulation 2-TIADC system

Assuming that reconstruction may be done by inverse interpolation functions $g_e[k]$ and $g_o[k]$ found from $f_e[k]$ and $f_o[k]$ with time discrete impulse responses, then, $\underline{\hat{x}}$, can be written as shown in equation 2.25.

$$\underline{\hat{x}} = [x_{2n}(-2k), x_{2n+1}(-2k), ..., x_{2n}(2k+1), x_{2n+1}(2k+1)] \tag{2.25}$$

where

$$\hat{x}[2n] = \sum_{k=-2K}^{2K+1} y[2n+k]g_e[-k] = \underline{g}_{eR}^T \underline{y}_{2n} \tag{2.26}$$

$$\hat{x}[2n+1] = \sum_{k=-2K}^{2K+1} y[2n+k]g_e[-k] = \underline{g}_{oR}^T \underline{y}_{2n} \tag{2.27}$$

where

$$\underline{g}_{eR}^T = [g_e[-2k], \cdots, g_e[2k+1]] \tag{2.28}$$

$$\underline{g}_{oR}^T = [g_o[-2k], \cdots, g_o[2k+1]] \tag{2.29}$$

$$\underline{y}_{2n}^T = [y[2n-2k], \cdots, y[2n+2k+1]] \tag{2.30}$$

In this case the interpolation functions are assumed to be symmetric around zero, as can be seen from $f[k]$ which is defined from $[-2k-1, 2k]$. Because of this we can, without loss of generality, set $n = 0$, and find the even and odd samples of $\hat{x}$

$$\hat{x}[0] = \underline{g}_{eR}^T \underline{y}_0 = \underline{y}_0^T \underline{g}_{eR} \tag{2.31}$$

$$\hat{x}[1] = \underline{g}_{oR}^T \underline{y}_0 = \underline{y}_0^T \underline{g}_{0R} \tag{2.32}$$

where

$$\underline{y}_0 = \underline{y}_{E,k} = \underset{\sim}{F}_{k,2k} \underline{x}_{E,2k} \tag{2.33}$$

By combining equation 2.33 and 2.31 we can find an expression for the reconstructed discrete-time signal $x[n]$ for n=0.

$$\hat{x}[0] = \underline{g}_{eR}^T \underset{\sim}{F}_{k,2k} \underline{x}_{E,2k} \hat{x}[1] = \underline{g}_{oR}^T \underset{\sim}{F}_{k,2k} \underline{x}_{E,2k} \tag{2.34}$$

The expression for $\hat{x}[0]$ shown above assumes perfect reconstruction. This is however not achievable, as the timing-mismatch will be estimated and thus, not 100% correct.

When taking into account the fact that the reconstructed signal $\hat{x}[n]$ will not be equal to $x[n]$, the best case scenario can be achieved by finding the least-mean square (LMS) estimate for $\hat{x}[n]$. When formulating the LMS estimate for this problem, it can be expressed as shown in equation 2.35.

$$\begin{aligned} D_e &= ||\underset{\sim}{F}^T_{k,2k}\underline{g}_{eR} - d_{1,e}||^2 \\ &= \underline{g}^T_{eR}\underset{\sim}{F}_{k,2k}\underset{\sim}{F}^T\underline{g}_{eR} - 2\underline{g}^T_{eR}\underset{\sim}{F}_{k,2k}d_{1,e} + 1 \end{aligned} \tag{2.35}$$

Minimization of the error gives us

$$\underset{\sim}{F}_{k,2k}\underset{\sim}{F}^T_{k,2k}\underline{g}_{eR} = \underset{\sim}{F}_{k,2k}d_{1,e} \tag{2.36}$$

When rewriting equation 2.36 to find the reconstruction vector g_{eR} and likewise for g_{oR} we get the expressions shown in equation 2.37 and equation 2.38 [1].

$$\underline{g}_{eR} = \left[\underset{\sim}{F}_{k,2k}\underset{\sim}{F}^T_{k,2k}\right]^{-1}\underset{\sim}{F}_{k,2k}d_{1,e} \tag{2.37}$$

$$\underline{g}_{oR} = \left[\underset{\sim}{F}_{k,2k}\underset{\sim}{F}^T_{k,2k}\right]^{-1}\underset{\sim}{F}_{k,2k}d_{1,o} \tag{2.38}$$

where

$$d^T_{1e} = [\overbrace{0,0,...,0}^{4k},1,0,...,0]}_{8k+2} \tag{2.39}$$

And for odd samples

$$d^T_{1o} = [\overbrace{0,0,...,0}^{4k+1},1,0,...,0]}_{8k+2} \tag{2.40}$$

The two vectors $\underline{g}_{eR}$ and $\underline{g}_{oR}$ contain all the reconstruction information required for the best possible reconstruction of $x[n]$.

Another way to evaluate the performance of the reconstruction is to rearrange equation 2.37 and 2.38 to find d_{1e} and d_{1o}. These calculations will for an ideal reconstruction yield the results shown in equation 2.41 and 2.42.

[1] Because of singular matrices, the Moore-Penrose pseudoinverse have been used.

$$\underline{d}_{1,e}^T = \underline{g}_{eR}^T \underset{\sim}{F}_{k,2k} \tag{2.41}$$

$$\underline{d}_{1,o}^T = \underline{g}_{oR}^T \underset{\sim}{F}_{k,2k} \tag{2.42}$$

The re-calculation of d can also be used as a condition to see if the reconstruction should work, independent of the actual reconstruction result.

2.6 Generalized reconstruction for M-TIADC setup

As mentioned earlier, the method for 2-TIADC signal reconstruction can be generalized to apply for an arbitrary even number of ADCs. The notation will be the same used for the 2-TIADC theory, with an added variable i which represents the individual samplers $[0, 1, ..., M-1]$.

Same as the 2-ADC approach we assume that the interpolation function is limited to the intervall $[-2k-1, 2k]$. Then the sampled signals can be expressed

$$y[Mn] = \sum_{k=-2K}^{2K+1} x(Mn+k)f(-k+\tau_0) = \underline{f}_{0R}^T \underline{x}_{Mn} \tag{2.43}$$

$$y[Mn+1] = \sum_{k=-2K}^{2K+1} x(Mn+k)f(-k+1+\tau_1) = \underline{f}_{1R}^T \underline{x}_{Mn} \tag{2.44}$$

..

$$\begin{aligned} y[M(n+1)-1] &= \sum_{k=-2K}^{2K+1} x(Mn+k)f(-k+(M-1)+\tau_{M-1}) \\ &= \underline{f}_{(M-1)R}^T \underline{x}_{Mn} \end{aligned} \tag{2.45}$$

where

$$\underline{f}_{iR} = [f(2k+i+\tau_i), \cdots f(\tau_i), \cdots f(-2k-1+i+\tau_i)] \tag{2.46}$$

$$\underline{x}_{Mn}^T = [x[Mn-2k], \cdots x[Mn], \cdots, x[Mn+2k+1]] \tag{2.47}$$

Defining extended vectors spanning $k \in [-2L, 2L+1]$. Applying this to the vectors for x and y and finding the new expressions for the extended x and y as shown in equations 2.48 and 2.49.

$$\underline{x}_{E,L}^T = [x[-2L], \cdots, x[2L+1]] \tag{2.48}$$

$$\underline{y}_{E,L}^T = [y[-2L], \cdots, y[2L+1]] \tag{2.49}$$

$$y_{E,L} = \underbrace{\begin{bmatrix} \underline{f}_{0R}^T & \underline{0} & \underline{0} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \underline{f}_{(M-1)R}^T & \underline{0} & \underline{0} \\ \cdot & \cdot & \cdot \\ \underline{0} & \underline{f}_{0R}^T & \underline{0} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \underline{0} & \underline{f}_{(M-1)R}^T & \underline{0} \\ \cdot & \cdot & \cdot \\ \underline{0} & \underline{0} & \underline{f}_{0R}^T \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \underline{0} & \underline{0} & \underline{f}_{(M-1)R}^T \end{bmatrix}}_{F_{L,L+k}} \underline{x}_{E,L+k} \tag{2.50}$$

where the matrix $\underset{\sim}{F}_{L,L+k}$ contains the information needed to recreate $y[n]$ and thus, the information needed to find the inverse interpolations function g_i.

2.6.1 Least Mean Square formulation for M-TIADC

Assuming that the reconstruction may be done by a corresponding functions with time discrete impulse respons $g_i[k]$, where i is the number of individual samplers. Then, the reconstructed signal for the individual samplers can be expressed

$$\hat{x}[Mn] = \sum_{k=-2K}^{2K+1} y[Mn+k]g_0[-k] = \underline{g}_{0R}^T \underline{y}_{Mn} \tag{2.51}$$

..

$$\hat{x}[M(n+1)-1] = \sum_{k=-2K}^{2K+1} y[Mn+k]g_{M-1}[-k] = \underline{g}_{(M-1)R}^T \underline{y}_{Mn} \tag{2.52}$$

where

$$\underline{g}_{iR}^T = [g_i[-2k], \ldots, g_i[2k+1]] \tag{2.53}$$

$$\underline{y}_{Mn}^T = [y[Mn-2k], \cdots, y[Mn+2k+1]] \tag{2.54}$$

Without losing generality, we can set n=0 and find expressions for the reconstructed signal from the individual samplers

$$\hat{x}[0] = \underline{g}_{0R}^T \underline{y}_0 = \underline{y}_0^T \underline{g}_{0R} \tag{2.55}$$

$$\hat{x}[1] = \underline{g}_{1R}^T \underline{y}_1 = \underline{y}_1^T \underline{g}_{1R} \tag{2.56}$$

$$\vdots$$

$$\hat{x}[M-1] = \underline{g}_{(M-1)R}^T \underline{y}_{M-1} = \underline{y}_{M-1}^T \underline{g}_{(M-1)R} \tag{2.57}$$

Assuming the reconstruction will not be perfect, the best case scenario can be achieved with LMS estimation of the reconstructed signal $\hat{x}[n]$. The LMS formulation can be seen in equation 2.58.

$$\begin{aligned} D_0 &= ||\underset{\sim}{F}_{S,F}^T \underline{g}_{0R} - d_{1,0}||^2 \\ &= \underline{g}_{0R}^T \underset{\sim}{F}_{S,F} \underset{\sim}{F}_{S,F}^T \underline{g}_{0R} - 2\underline{g}_{0R}^T \underset{\sim}{F}_{S,F} d_{1,0} + 1 \end{aligned} \tag{2.58}$$

For the generalized method, the dimensions of $\underset{\sim}{F}$ will be $[S \times F]$ where $S = M(2k+1)$ and $F = ((2M+4)k+2)^2$.

When minimizing equation 2.58 we get the following expression for the reconstruction vectors g_{iR}

$$\underline{g}_{iR} = \left[\underset{\sim}{F}_{S,F} \underset{\sim}{F}_{S,F}^T\right] \underset{\sim}{F}_{S,F} d_{1i} \tag{2.59}$$

where

$$d_{1i}^T = [\overbrace{0, \cdots, 0}^{(M+2)k+i} \underbrace{1, \cdots, 0}_{(2M+4)k+2}] \tag{2.60}$$

The vectors $g_i[k]$ contains all the information needed to reconstruct the input-signal $x[n]$.

Same as for the 2-TIADC setup, the performance of the reconstruction can be measured by rearranging equation 2.59 for calculating d_{1i}, which gives us equation 2.61.

$$\underline{d}_{1i}^T = \underline{g}_{iR}^T \underset{\sim}{F}_{M(2k+1),(2M+4)k+2} \tag{2.61}$$

[2]The dimensions for $\underset{\sim}{F}$ were found by simulation in MatLab for M=2,4,8.

Chapter 3

Method

This chapter will present block diagrams of the MatLab implemented system, used for signal reconstruction from a 2-TIADC system assuming non-uniform periodic samplers and known timing-mismatch.

3.1 Theory versus MatLab

When implementing the signal reconstruction method in MatLab som changes had to be made. These changes occurred because of the MatLab indexing which starts at 1. This is different from the theory which assumes that the indexing starts at 0. In other words, the theory assumes that the timing-mismatch affects all *odd* samples, but because of the MatLab indexing the timing-mismatch affects the *even* samples.

In the implemented code this was accounted for by changes in f_{eR}, f_{oR}, d_{1e} and d_{1o}. The changes can be seen in the equations shown below.

$$f_{eR}^T = [f(2k+1), \cdots, f(0), \cdots, f(-2k)] \tag{3.1}$$

$$f_{oR}^T = [f(2k+\tau), \cdots, f(\tau), \cdots, f(-2k-1-\tau)] \tag{3.2}$$

$$d_{1e}^T = [\overbrace{0, \cdots, 0}^{4k+1}, 1, 0, \cdots, 0] \tag{3.3}$$

$$d_{1o}^T = [\overbrace{0, \cdots, 0}^{4k}, 1, 0, \cdots, 0] \tag{3.4}$$

Because of the changes to the previously mentioned vectors, changes also had to be made in the matrix $\underset{\sim}{F}$. The changes made to $\underset{\sim}{F}$ was in the order of the vectors f_e and f_o. In the implementation the matrix $\underset{\sim}{F}$ started with the values from f_o and

then the values from f_e, which is the opposite order of what is shown in equation 2.22.

3.2 Sine input-signal

The first step in the implementation was to create and sample a sine signal using the wanted signal-frequency, sampling-frequency and timing-mismatch. After finding $x[n]$ and $y[n]$ the interpolation function $f(t)$ was created.

3.2.1 Error calculations

The MSE calculations were found by equation 3.5 ignoring the values which contained edge-effects since this error can be eliminated.

$$MSE_{dB} = 10log_{10}(max((x[n] - \hat{x}[n])^2)) \tag{3.5}$$

3.2.2 Reconstruction from 2-TIADC system

The first step in the reconstruction of the non-ideal sampled signal $\hat{y}[n]$, was assigning values from $f(t)$ to the interpolation functions f_e and f_o. This was done as shown in equations 3.1 and 3.2, where the values correspond to the zero-crossings in $f(t)$[1] [2].

The next step was reconstruction of $y[n]$. This was done by convolving the input-signal $x[n]$ with f_e and f_o. After removing the zero-padding the resulting $\hat{y}[n]$ was compared to $y[n]$. The setup used can be seen in Figure 3.1.

[1] See Appendix A for the created MatLab functions used in the 2- and 4-TIADC systems.

[2] See appendix B for the MatLab code used in the signal reconstruction from the 2-TIADC system.

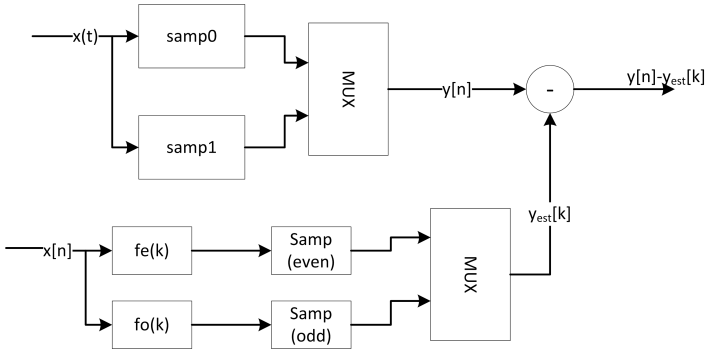


Figure 3.1: The setup used for reconstruction of $y[n]$ from the 2-TIADC system.

After finding values for α and k which gave an error below the chosen limit, the reconstruction of the input-signal $x[n]$ could begin. The first step was formulation the reconstruction vectors g_e and g_o using equations 2.37 and 2.38. After finding these vectors, $x[n]$ could be reconstructed by convolving g_e and g_o with $y[n]$. The MatLab system can be seen in Figure 3.2.

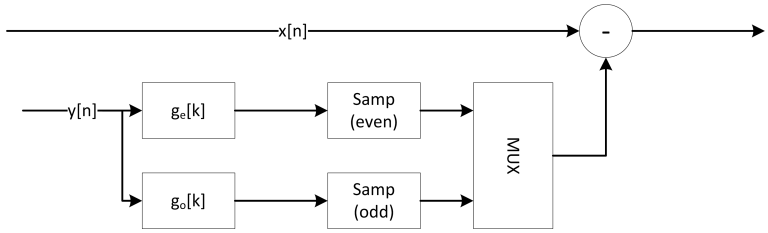


Figure 3.2: The setup used for reconstruction of $x[n]$ from the 2-TIADC system.

3.2.3 Reconstruction from 4-TIADC system

When expanding from 2-TIADC to 4-TIADC, the equations from the generalized theory have been applied with M=4[3]. In the implementation, the changes applied in the 2-TIADC setup regarding the vectors f_{iR},and d_i were extended to the 4-TIADC system.

The setup used to recreate the non-ideal sampled signal $y[n]$ for the 4-TIADC system can be seen in Figure 3.3.

[3]See appendix C for the MatLab code used in the signal reconstruction from a 4-TIADC system.

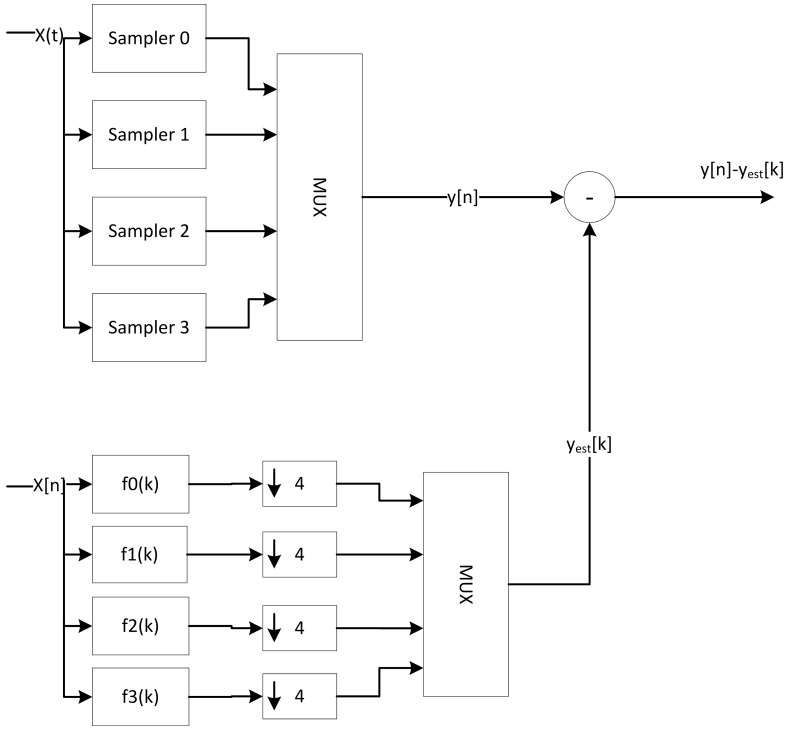


Figure 3.3: The setup used to recreate $y[n]$ for a 4-TIADC system.

The last step was to reconstruct the input signal $x[n]$. This was done using a setup similar to the setup for 2-TIADC, but extended to apply for 4 samplers. The setup can be seen in Figure 3.4.

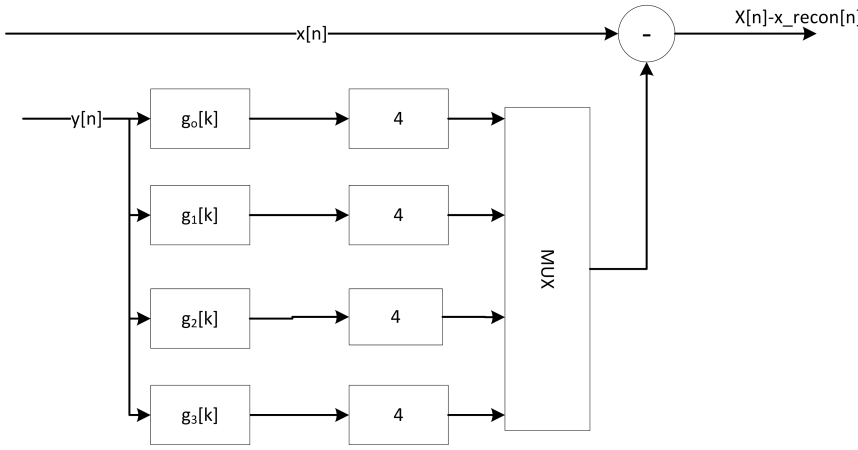


Figure 3.4: The setup used to recreate $x[n]$ for a 4-TIADC system.

Chapter 4

Simulation Results

This chapter will present the chosen signals, their values and the simulated results. The various simulation results will also be commented. The first part will present the results of signal reconstruction from a 2-TIADC system, and the second part will show the results for a 4-TIADC system. The presented MSE results were found by using the equation 3.5 and the condition mentioned in section 3.2.1.

The effect of errors in τ will not be presented as there is no adaptive filtering, and thus the error introduced by using erroneous values for the timing-mismatch will be constant and independent of k.

4.1 Signal reconstruction for a 2-TIADC system

As a test of the interpolation method, a sine signal was sent through the reconstruction circuit with various roll-off factors. The signal-frequency and the other chosen values that ensures a thorough test, can be seen in Table 4.1.

4.1.1 Reconstructing the sampled signal in a 2-TIADC system

After the initial test using parameters α=0.2 and k=50, which gave a an estimation error within the boundaries of the tolerated error mentioned in Table 4.1, further simulations were performed. When using convolution, the length of the filter and the value of the roll-off factor α plays a big part in the added delay and the accuracy in the output signal. A longer filter length will decrease the error, but it will also increase the added delay. In the sense of trade-offs, an increased roll-off factor will give less error, but it will also increase the needed bandwidth. Finding suitable

Table 4.1: The chosen values for the input-signal used in the method for signal reconstruction from the 2-TIADC system.

Variabel	Value	Unit
f	350	[MHz]
α_1	70% (0.3)	$[F_s/2]$
α_2	80% (0.2)	$[F_s/2]$
α_3	90% (0.1)	$[F_s/2]$
α_4	95% (0.05)	$[FS/2]$
F_s	1	[GHz]
Time-skew	$0.05T_s$	[s]
Error Limit	-100	[dB]
k	10-200, step-size 10	

values for both α and the filter length is important for the end result, and one way to determine these are through simulations.

The results of the simulations using k and α shown in Table 4.1 can be seen in Figure 4.1. See Appendix E.1 for the full size figure.

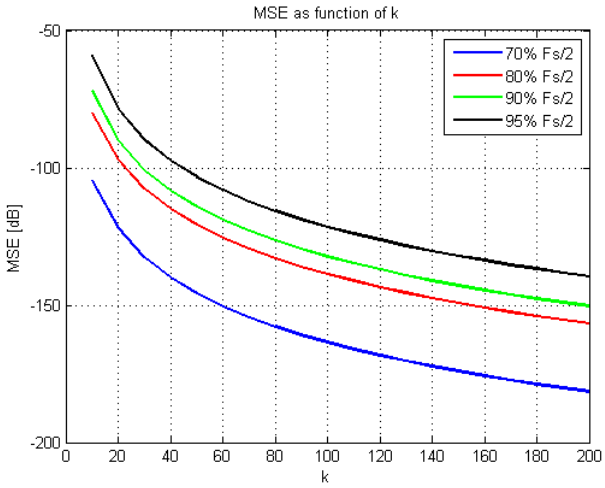


Figure 4.1: MSE as a function of k with varying α for $y[n]$ for the 2-TIADC system.

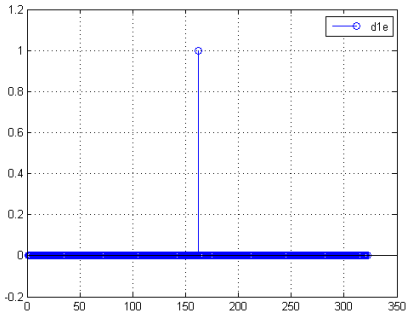
From Figure 4.1 it is easy to see that the error decreases when k and α increases. When comparing the graphs with the error limit which is -100dB, the required filter length can by found by using a k between 20 and 45 depending on α.

4.1.2 Reconstructing the input signal in a 2-TIADC system

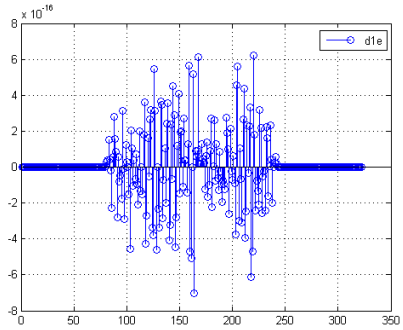
The next step after recreating $y[n]$, was using the information found in the interpolation vectors f_e and f_o to create the reconstruction vectors g_e and g_o.

As mentioned in the theory, one way to check the method is to recalculate the vectors d_{1e} and d_{1o}. These calculations were carried out, and the results can be seen in Figure 4.2 and 4.3. For full size figures see Appendix D.1-4.

1. $g_{eR}^T \underset{\sim}{F}_{k,2k} = d_{1e}^T$
2. $g_{oR}^T \underset{\sim}{F}_{k,2k} = d_{1o}^T$

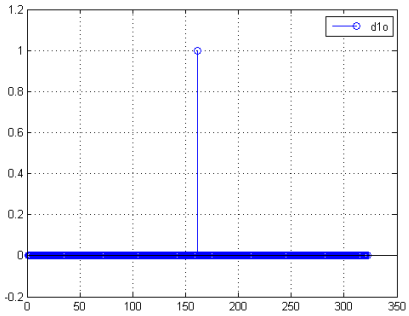


(a) The calculated d_{1e}

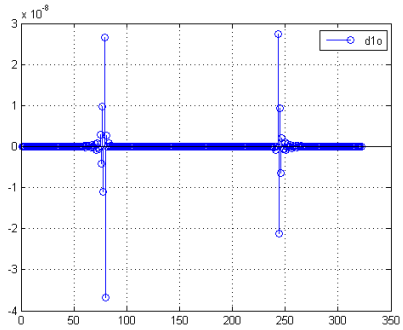


(b) Illustration of d_{1e} without the maximum

Figure 4.2: Illustration of the calculated d_{1e} for the 2-TIADC system.



(a) The calculated d_{1o}



(b) Illustration of d_{1o} without the maximum

Figure 4.3: Illustration of the calculated d_{1o} for the 2-TIADC system.

From Figure 4.2 and Figure 4.3 it is easy to see that the recalculated d_{1o} is not identical to the original one. This result was expected as the reconstruction is not perfect. The error is however sufficiently small to consider the result as correct.

The next and final step, was to use the reconstruction vectors g_e and g_o recreate the input signal $x[n]$. The setup can be seen in Figure 3.2, and the MSE results as a function of k with varying α can be seen in Figure 4.4. See Appendix E.2 for the full size figure.

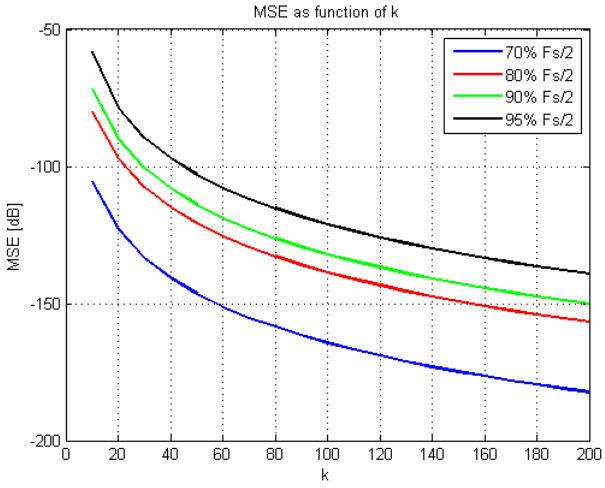


Figure 4.4: The MSE between $x[n]$ and $\hat{x}[n]$ as function of k with varying α for the 2-TIADC system.

From Figure 4.4 it is easy to see that reconstruction of the input-signal can be achieved with $k = 10$ and $\alpha = 0.3$ within the the set error limit. By choosing a larger k the excess bandwidth can be reduced. These results are expected as there is a trade-off between filter-length and excess bandwidth.

4.1.3 Effect of varying timing-mismatch

As a test of the reconstruction methods ability to handle increasing timing-mismatch the setup for 2-TIADC was tested with different time-skews. The chosen values can be seen in Table 4.2.

Table 4.2: The different variable and their values used to test the reconstruction method for increasing timing-mismatch.

Variable	Value	Unit
k_1	25	
k_2	50	
k_3	100	
τ_1	0.1-0.99, step-size 0.01	$[Ts]$
α	0.3	

The results from the simulations can be seen Figure 4.5 and Figure 4.6. See Appendix E.5-6 for the full size figures.

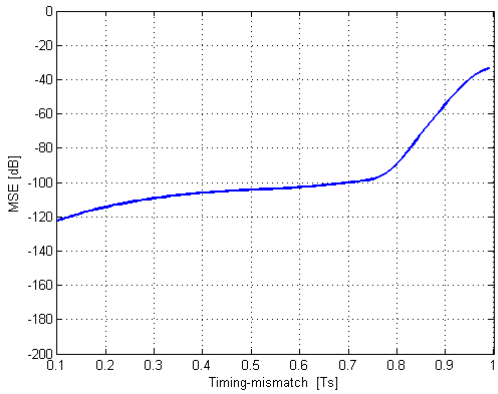


Figure 4.5: MSE as function of τ for k=25 for the 2-TIADC system.

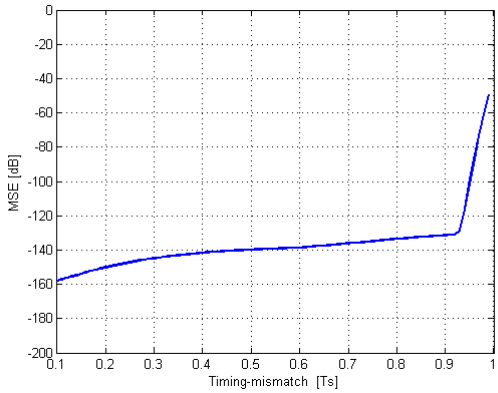


Figure 4.6: MSE as function of τ for k=100 for the 2-TIADC system.

The results shown in the figures above show that the error increases as τ increases. The error increases moderately until a threshold, dependent on k, where it starts to increases exponentially. The figures also show that with a large k, the method can handle reconstruction of a signal with an added timing-mismatch of $\approx 0.95Ts$, when comparing to the tolerated error of $-100dB$.

4.2 Signal reconstruction for a 4-TIADC system

The signal reconstruction from 4 samplers were simulated with the values shown in Table 4.3.

Table 4.3: The variable and values used for simulating signal reconstruction from a 4-TIADC system.

Variable	**Value**	**Unit**
k	10-200, step-size 10	
τ_0	0	[Ts]
τ_1	-0.05	[Ts]
τ_2	0.06	[Ts]
τ_3	-0.04	[Ts]
α_1	70% (0.3)	[Fs/2]
α_2	80% (0.2)	[Fs/2]
α_3	90% (0.1)	[Fs/2]
α_4	95% (0.05)	[Fs/2]

4.2.1 Reconstructing the sampled signal from a 4-TIADC system

The first simulation was reconstructing the non-ideal sampled signal $\hat{y}[n]$. The simulation results can be seen in Figure 4.7. See Appendix E.6 for the full size figure.

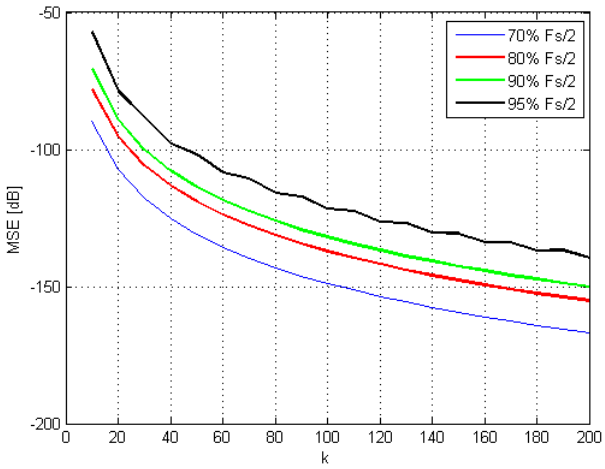


Figure 4.7: Simulated MSE for the 4-TIADC setup when reconstructing $y[n]$.

From Figure 4.7 it is easy to see an increased error, especially for a low k and $\alpha = 0.3$ when compared to the results for the 2-TIADC system. The simulation results also show some irregularities for $\alpha = 0.05$ $(95\%Fs/2)$, these might be caused by the *sine* input-signal.

4.2.2 Reconstructing the input-signal from a 4-TIADC system

The reconstruction of the input signal $x[n]$ was done using the setup seen in Figure 3.4. The results can be seen in Figure 4.8. See Appendix E.7 for the full size figure.

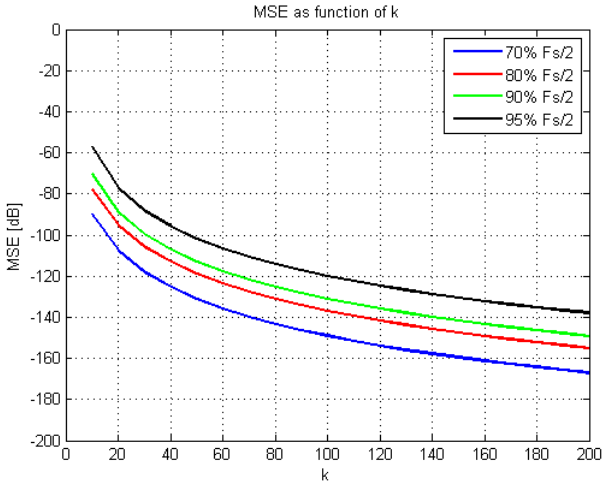
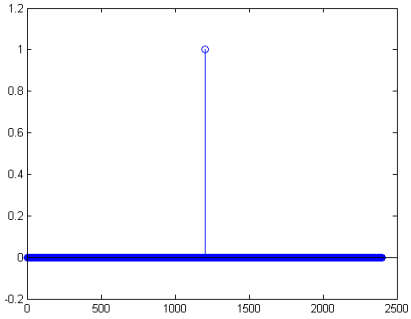


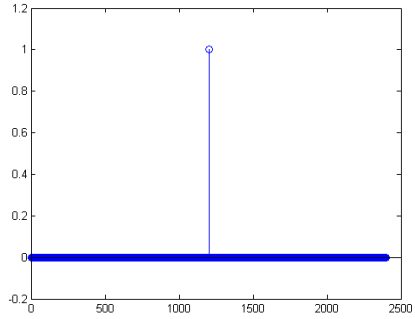
Figure 4.8: Simulated MSE for 4-TIADC setup when reconstructing $x[n]$.

The results from Figure 4.8 are expected as the MSE decreases as k increases, and the results are similar to the results achieved in the reconstruction of the sampled signal $y[n]$. The main difference is the irregularities for $\alpha = 0.05$ seen in Figure 4.7, which have disappeared when reconstructing $x[n]$.

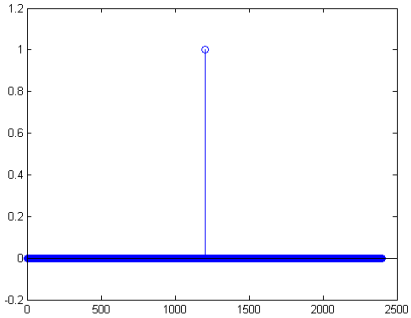
As an additional check of the results, the vectors d_{10}, d_{11}, d_{12} and d_{13} were calculated to see if the result equaled delta-pulses. The new d_is had a maximum value of 1 and an error within the limit. The results for d_{10} and d_{11} without the maximum can be seen in Figure 4.9a-d. See Appendix D.5-12 for the full size figures.



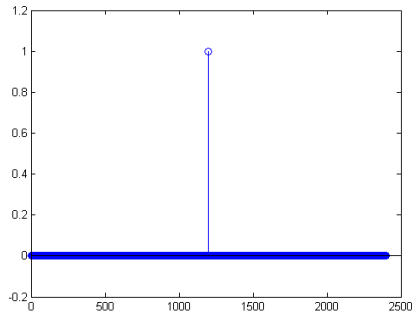
(a) The calculated d_{10}



(b) Illustration of d_{11}



(c) The calculated d_{12}



(d) Illustration of d_{13}

Figure 4.9: Illustration of the error of the calculated d_i for the 4-TIADC system.

From the figures shown above, the recalculations of the d_is gave the correct results, as all the vectors have the form of a delta-pulse. Although the vectors are not perfect delta-pulses, the errors are sufficiently small to be satisfied with the results.

Chapter 5

Conclusion

In this thesis, a method for signal reconstruction from a 2-TIADC and M-TIADC system has been explained theoretically. A possible solution for signal reconstruction from 2- and 4-TIADC has been implemented in MatLab, when assuming known timing-mismatch and non-uniform periodic sampling. The results were presented as MSE compared to a chosen error limit of -100 dB.

The simulation results from the 2-TIADC showed that the input-signal could be reconstructed with an error below the limit by choosing k as low as 10 and the roll-off factor, α as 0.3. The same result could also be achieved with $k = 45$ and α set to 0.05. The 2-TIADC system was also tested to see how the method could handle an increasing timing-mismatch. The results indicated that an increasing timing-mismatch could be corrected within the error limit for mismatch up to 95% of T_s. This maximum was achieved with $k = 100$. When lowering k to 25, the maximum mismatch which could be corrected was lowered to 70% of T_s.

The simulations of the 4-TIADC system also achieved signal reconstruction within the error limit. The biggest difference from the 2-TIADC was the error when $\alpha = 0.3$. Following the results, k would have to be increased to 15 to achieve a reconstructed signal withing the error limit.

The overall impression of the chosen signal reconstruction method after testing, is that the method achieves the objective for the chosen input-signal. The reconstructed input-signal is well within the error limit of -100dB. One downside is that the method was only tested with a sine signal, and not signals with a bigger bandwidth.

5.1 Future work

In order to better the signal reconstruction method and make it more realistic, since the timing-mismatch will be estimated in real life applications, is the use of adaptive filtering. This would enable the possibility for simulations of the effects of increasing error in the estimated timing-mismatch.

Another topic for future work is the implementation of a digital signal. A sine signal is sufficient for testing, but in a more realistic test, the input-signal would be digital i.e. a PAM signal.

Bibliography

[1] John R. Barry, Edward A. Lee, and David G. Messerschmitt. *Digital Communication*. Springer, 3th edition, 2004.

[2] William C. Jr Black and David A. Hodges. Time interleaved converter arrays. *IEEE Journal of Solid State Circuits*, SC-15, 1980.

[3] Vijay Divi. *Estimation and Calibration Algorithms for Distributed Sampling Systems*. PhD thesis, Massachusetts Institute of Technology, 2008.

[4] Vijay Divi and Gregory W. Wornell. Blind caloibration of timing skew in time-interleaved analog-to-digital converters. *IEEE Journal of Selected Topics in Signal Processing*, 3, 2009.

[5] Nils Holte. Reconstruction of a non ideally sampled signal for two multiplexed samplers, march 2012. revision 28.06.2012.

[6] Naoki Kurosawa, Haruo Kobayashi, Kaoru Maruyama, Hidetake Sugawara, and Kensuke Kobayashi. Explicit analysis of channel mismatch effects in time-interleaved adc systems. *IEEE Transactions on Circuits and Systems: Fundamental Theory and Applications*, 48, 2001.

[7] Ryan S. Prendergast, Bernard C. Levy, and Paul J. Hurst. Reconstruction of band-limited periodic nonuniformly sampled signals through multirate filter banks. *IEEE Transactions on Circuits and Systems: Regular Papers*, 51, 2004.

[8] John G. Proakis and Dimitris G. Manolakis. *Digital Signal Processing, Principles, Algorithms and Applications*. Pearson Prentice Hall, 4th edition, 2007.

[9] Munkyo Seo, Mark J. W. Rodwell, and Upamanyu Madhow. Blind correction of gain and timing mismatches for a two-channel time-interleaved analog-to-digital converter: Experimental verification.

[10] Stefan Tertinek and Christian Vogel. Reconstruction of nonuniformly sampled bandlimited signals using a differentiator-multiplier cascade. *IEEE Transactions on Circuits and Systems: Regular Papers*, 55, 2008.

[11] K. M. Tsui and S. C. Chan. A versatile iterative framework for the reconstruction of bandlimited signals from their nonuniform samples. *Journal of Signal Processing Systems*, 62, 2011.

[12] Christian Vogel and Christian Doppler. Comprehensive error analysis of combined channel mismatch effects in time-interleaved adcs. 2003. Conference Publication.

[13] Yuexian Zou, Bo Li, and Xiao Chen. An efficient blind timing skews estimation for time-interleaved analog-to-digital converters. 2011. Conference Publication.

Appendices

Appendix A

MatLab illustration code and functions

Ideal 2-ADC sampling

```
%% Matlab code for an ideal 2ADC system
%Variables
f          = 1e5;
Fs         = 1e7;
Ts         =  1/Fs;
LENGTH     = 2^12;
RunTime    = (LENGTH-1)*Ts;
m          = 2;

t   = 0:Ts:RunTime;
t_sampler0 = 0:m*Ts:RunTime;
t_sampler1 = Ts:m*Ts:RunTime;

%Sampling

x1 = sin(2*pi*f*t_sampler0);
x2 = sin(2*pi*f*t_sampler1);

%Multiplexing
x=mux2(x1,x2);
```

Illustration of 2-TIADC with mismatches

```
%% Matlab code for non-ideal TI-ADC system

f = 1e5;
Fs = 1e7;
Ts = 1/Fs;
NFFT = 2^12;
m = 2;

RunTime = (NFFT-1)*Ts;
t = 0:Ts:RunTime;
t_sampler0 = 0:m*Ts:RunTime;
t_sampler1 = Ts:m*Ts:RunTime;

%Gain mismatch, dg
dg = [1  0.98];

%Offset mismatch
do = [0  0.3];


%Time-skew
dt = [0  0.005];

x1 = do(1) + dg(1)*sin(2*pi*f*t_sampler0 + dt(1));
x2 = do(2) + dg(2)*sin(2*pi*f*t_sampler1 + dt(2));

%mulitplexing
x = mux2(x1,x2);
```

Raised cosine function

```
function cosrolloff = cosRollOffCont(t,alpha,sampFreq)

if nargin<2
    alpha=0.5;
end
if nargin<3
    sampFreq=1e9;
end

Ts = 1/sampFreq;
tmax = max(t);
tmin = min(t);
tlength = length(t)-1;

stepsize = (1/(tlength/(abs(tmax)+abs(tmin))));
num=zeros(1,length(t));
den=zeros(1,length(t));
cosrolloff=zeros(1,length(t));

k=0;
for i=tmin:stepsize:tmax;
    k=k+1;
    num(k) = sin(pi*i/Ts)*cos(alpha*pi*i/Ts);
    den(k) = (pi*i/Ts)*(1-(2*alpha*i/Ts)^2);

    %Using if to bypass the matlab 'inf' for 0/0 and X/0
    %Because of matlab using ~stepsize and not the actual
    %value (1-(2*alpha*i/Ts)^2) vil never yield zero, but a
    %value ~Xe-16. Because of this i'm using e-15 as a
    %threshold seeing as the %neighbouring values are ~e-3

    if abs((1-(2*alpha*i/Ts)^2))<1e-15
        cosrolloff(k)=(pi/4)*sinc(i/Ts);

    elseif (pi*i/Ts)==0
        cosrolloff(k)=cos(pi*alpha*i/Ts)/(1-(2*alpha*i/Ts)^2);

    else
        cosrolloff(k) = num(k)/den(k);

    end
end
end
```

Multiplexing function for 2 input signals

```
function out = mux2(signal_1, signal_2)

%Function that multiplexes two signals.
%The output function starts with the first sample
%in signal_1 and then the first sample in signal_2
%and so on.


out=zeros(2*length(signal_1),1);
out(1:2:end) = signal_1;
out(2:2:end) = signal_2;

end
```

Multiplexing function for 4 input signals

```
function out = mux4(signal_1, signal_2, signal_3, signal_4)

%Function that multiplexes four signals.
%The output function starts with the first sample
%in signal_1 and then the first sample in signal_2
%and so on.

out=zeros(4*length(signal_1),1);
out(1:4:end) = signal_1;
out(2:4:end) = signal_2;
out(3:4:end) = signal_3;
out(4:4:end) = signal_4;

end
```

Appendix B

MatLab code for 2-TIADC

```
%This is the MatLab code used for signal reconstruction from
%a 2-TIADC system with non-uniform periodic samplers.

clear all;
close all;

%Variables
amp = 1;
LENGTH = 2^12;
F = 350e6;
Fs = 1e9;
Ts = 1/Fs;
nr_samp = 2;
TIMESKEW = 0.05;
k=10;

%Sample vectors
t = (-LENGTH/2:1:LENGTH/2-1)*Ts;
t_sampler0=(-LENGTH/2:nr_samp:LENGTH/2-1)*Ts;
t_sampler1=(-LENGTH/2+1+TIMESKEW:nr_samp:LENGTH/2+TIMESKEW)*Ts;

%Signal sampling
IDEAL_SAMPLED_SIGNAL = amp*sin(2*pi*F*t)';
SIGNAL_SAMPLER0      = amp*sin(2*pi*F*t_sampler0);
SIGNAL_SAMPLER1      = amp*sin(2*pi*F*t_sampler1);% + TIMESKEW);

%Multiplexing the sampled signal
NON_IDEAL_SAMPLED_SIGNAL = mux2(SIGNAL_SAMPLER0,SIGNAL_SAMPLER1);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%% Raised cosine interpolation function %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Raised Cosine interpolation function
alpha=0.3;
stepsize = 0.001;
P = 1/stepsize;
skew = TIMESKEW/stepsize;
TIME_VECTOR = (-1000:stepsize:1000).*Ts;
f_x = cosRollOffCont(TIME_VECTOR, alpha, Fs);
middle_fx = (length(f_x)-1)/2+1;

%Assigning values to fe and fo
fe = zeros(4*k+2,1);
mid_fe = length(fe)/2;
fe(mid_fe)          = f_x(middle_fx);
fe(mid_fe-1:-1:1) = f_x(mid_fx-P:-P:mid_fx-(P*((mid_fe-2))));
fe(mid_fe+1:1:end)= f_x(mid_fx+P:P:mid_fx+(P*((mid_fe-1))));

fo = zeros(4*k+2,1);
mid_fo = length(fo)/2+1;
fo(mid_fo)          =f_x(mid_fx-skewSamp);
fo(mid_fo-1:-1:1) =f_x(mid_fx-(P+skew):-P:mid_fx-((P)*((mid_fo))));
fo(mid_fo+1:1:end)=f_x(mid_fx+(P-skew):P:mid_fx+((P)*((mid_fo-1))));

%Reversing fe and fo
foR = fo(end:-1:1);
feR = fe(end:-1:1);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%% Recreation of y[n] %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Convolution to find y[n]
yoddhat = conv(IDEAL_SAMPLED_SIGNAL, feR, 'full');
yevenhat = conv(IDEAL_SAMPLED_SIGNAL, foR, 'full');

%Removing zero-padding and decimation
yoddhat = yoddhat(2*k+2:2:end-(2*k));
yevenhat = yevenhat(2*k+2:2:end-(2*k));

%multiplexing to get the recreated y[n] without zero-padding
```

```
yhat = mux2(yodd,yeven);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%% Reconstruction of x[n] %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Creating the reconstruction matrix F
Feo = zeros(length(fe)/2, length(fe));
for i = 1:length(fe)/2
    for j = 1:length(fe)
        Feo(2*i-1,(2*i-1)+j-1) = foR(j);
        Feo(2*i,(2*i-1)+j-1)   = feR(j);
    end
end

%Creating the vectors de and do
deven = zeros(size(Feo,2),1);
deven(4*k+2) = 1;
dodd  = zeros(size(Feo,2),1);
dodd(4*k+1) = 1;

%Finding the reconstrucion vectors ge and go
ge = pinv(Feo*Feo')*Feo*deven;
go = pinv(Feo*Feo')*Feo*dodd;

%Convolution to find the reconstructed x[n]
xhatteven = conv(NON_IDEAL_SAMPLED_SIGNAL,ge,'full');
xhattodd  = conv(NON_IDEAL_SAMPLED_SIGNAL,go,'full');

%Removing the zero-padding and Decimation by 2
xhatteven1 = xhatteven(2*k+2:2:end-(2*k));
xhattodd1  = xhattodd(2*k+2:2:end-(2*k));

%The reconstructed x[n]
xhatt = mux2(xhatteven1,xhattodd1);
```

Appendix C

MatLab code for 4-TIADC

```
clear all;
close all;

%Variables
LENGTH = 2^12;
F = 350e6;
Fs = 1e9;
Ts = 1/Fs;
k=20;
alpha=0.2;

M = 4;
delta0 = 0;
delta1 = -0.05;
delta2 = 0.06;
delta3 = -0.04;

%time vectors
t = 0:Ts:(LENGTH-1)*Ts;
t_sampler0 = 0:M*Ts:(LENGTH-1)*Ts;
t_sampler1 = ((1+delta1):M:(LENGTH-1+delta1))*Ts;
t_sampler2 = ((2+delta2):M:(LENGTH-1+delta2))*Ts;
t_sampler3 = ((3+delta3):M:(LENGTH-1+delta3))*Ts;


%Sampling
y0 = sin(2*pi*F*t_sampler0);
y1 = sin(2*pi*F*t_sampler1);
y2 = sin(2*pi*F*t_sampler2);
```

```
y3 = sin(2*pi*F*t_sampler3);

IDEAL_SIGNAL = sin(2*pi*F*t)';
NON_IDEAL = mux4(y0,y1,y2,y3);

stepsize = 1/1000;
periode = 1/stepsize;
P = periode;
t_RCI = (-1000:stepsize:1000)*Ts;
f = cosRollOffCont(t_RCI,alpha,Fs);
mid_f = (length(f_tot)-1)/2+1;


%Assigning values for the individual interpolation functions
f0 = zeros(4*k+2,1);
D0 = delta0/stepsize;
mid_f0 = length(f0)/2;
f0(mid_f0) = f_tot(middle_f);
f0(mid_f0-1:-1:1) = f(mid_f-P+D0:-P:mid_f-((4*k+2)/2*P-1));
f0(mid_f0+1:1:end) = f(mid_f+P+D0:P:mid_f+((4*k+2)/2*P+1));

f1 = zeros(4*k+2,1);
D1 = delta1/stepsize;
mid_f1 = (length(f1))/2+1;
f1(mid_f1) = f(mid_f-D1);
f1(mid_f1-1:-1:1)=f(mid_f-(P+D1):-P:mid_f-((4*k+2)/2*P+1));
f1(mid_f1+1:1:end)=f(mid_f+(P-D1):P:mid_f+((4*k+2)/2*P-1));

f2 = zeros(4*k+2,1);
D2 = delta2/stepsize;
mid_f2 = (length(f2))/2+2;
f2(mid_f2) = f(mid_f-D2);
f2(mid_f2-1:-1:1) = f(mid_f-(P+D2):-P:mid_f-((4*k+2+4)/2*P));
f2(mid_f2+1:1:end) = f(mid_f+(P-D2):P:mid_f+((4*k+2-4)/2*P));

f3 = zeros(4*k+2,1);
D3 = delta3/stepsize;
mid_f3 = (length(f3))/2+3;
f3(mid_f3) = f_tot(middle_f-D3);
f3(mid_f3-1:-1:1) = f(mid_f-(P+D3):-P:mid_f-((4*k+2+4)/2*P));
f3(mid_f3+1:1:end) = f(mid_f+(P-D3):P:mid_f+((4*k+2-4)/2*P));

%Reversing the time-vectors
f0R = f0(end:-1:1);
f1R = f1(end:-1:1);
```

```
f2R = f2(end:-1:1);
f3R = f3(end:-1:1);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%% Reconstruction of y[n] %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Convolution
y0_hatt_temp = conv(IDEAL_SIGNAL, f0R);
y1_hatt_temp = conv(IDEAL_SIGNAL, f1R);
y2_hatt_temp = conv(IDEAL_SIGNAL, f2R);
y3_hatt_temp = conv(IDEAL_SIGNAL, f3R);

%Removing zero-padding
y0_hatt = y0_hatt_temp(2*k+2:4:end-(2*k));
y1_hatt = y1_hatt_temp(2*k+2:4:end-(2*k));
y2_hatt = y2_hatt_temp(2*k+2:4:end-(2*k));
y3_hatt = y3_hatt_temp(2*k+2:4:end-(2*k));

%Multiplexing
y_hatt = mux4(y0_hatt,y1_hatt,y2_hatt,y3_hatt);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%% Reconstruction of x[n] %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Reconstruction matrix F
Fi = zeros((2*k+1)*nr_samp,(2*nr_samp+4)*k+2);

for i=1:(4*k+2)/2
 for j=1:4*k+2
  Fi(4*i,(nr_samp*i-(nr_samp-1))+j-1) = f0R(j);
  Fi(4*i-1,(nr_samp*i-(nr_samp-1))+j-1) = f1R(j);
  Fi(4*i-2,(nr_samp*i-(nr_samp-1))+j-1) = f2R(j);
  Fi(4*i-3,  (nr_samp*i-(nr_samp-1))+j-1) = f3R(j);

 end
end

%The delta-pulses d_i
d10 = zeros((2*nr_samp+4)*k+2,1);
d11 = zeros((2*nr_samp+4)*k+2,1);
d12 = zeros((2*nr_samp+4)*k+2,1);
d13 = zeros((2*nr_samp+4)*k+2,1);
```

```
d10((nr_samp+2)*k+4) = 1;
d11((nr_samp+2)*k+3) = 1;
d12((nr_samp+2)*k+2) = 1;
d13((nr_samp+2)*k+1) = 1;

%Creating the inverse interpolation functions
g0 = pinv(Fi*Fi')*Fi*d10;
g1 = pinv(Fi*Fi')*Fi*d11;
g2 = pinv(Fi*Fi')*Fi*d12;
g3 = pinv(Fi*Fi')*Fi*d13;

%Convolving
x0hat = conv(NON_IDEAL,g0);
x1hat = conv(NON_IDEAL,g1);
x2hat = conv(NON_IDEAL,g2);
x3hat = conv(NON_IDEAL,g3);

%Decimation to get every 4th sample
x0 = x0hat(4*k+4:4:end-(4*k));
x1 = x1hat(4*k+4:4:end-(4*k));
x2 = x2hat(4*k+4:4:end-(4*k));
x3 = x3hat(4*k+4:4:end-(4*k));

%The reconstructed x[n]
xhat = mux4(x0,x1,x2,x3);
```

Appendix D

Illustration figures - Delta pulses

The recreated delta-pulses for the 2-TIADC system

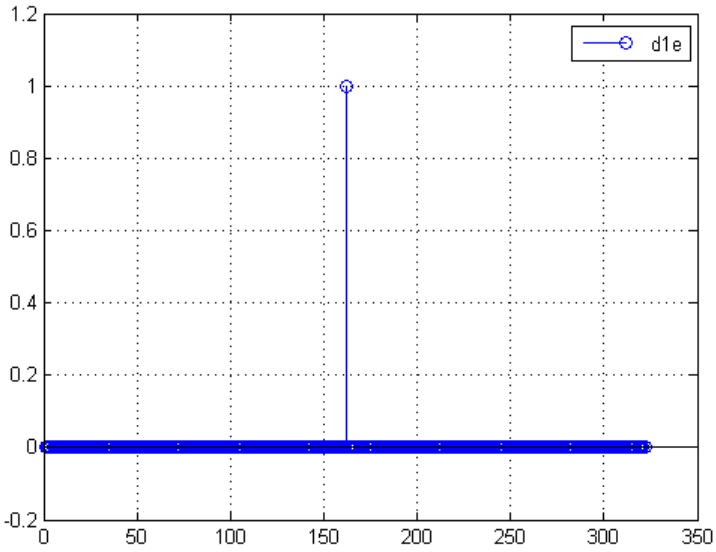


Figure D.1: The calculated d_{1e} for 2-TIADC system.

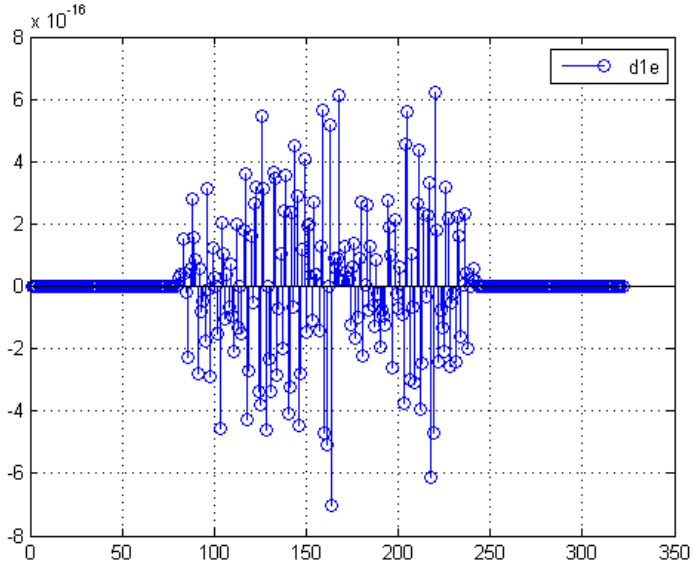


Figure D.2: The calculated errors of d_{1e} for 2-TIADC system.

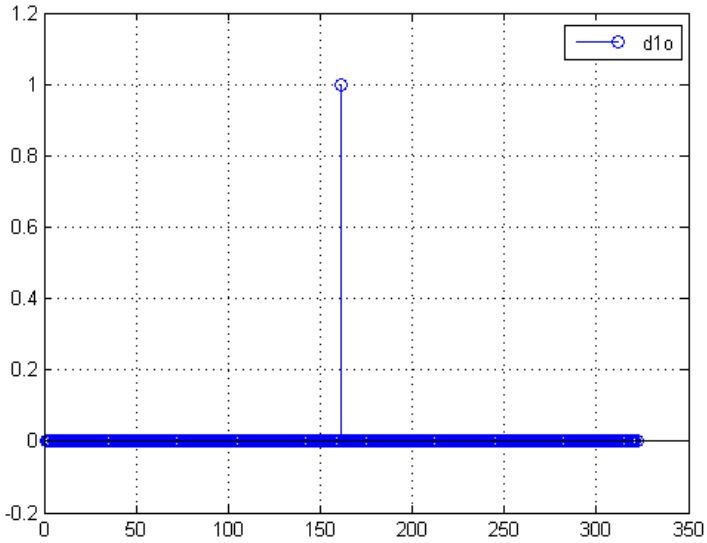


Figure D.3: The calculated d_{1o} for 2-TIADC system.

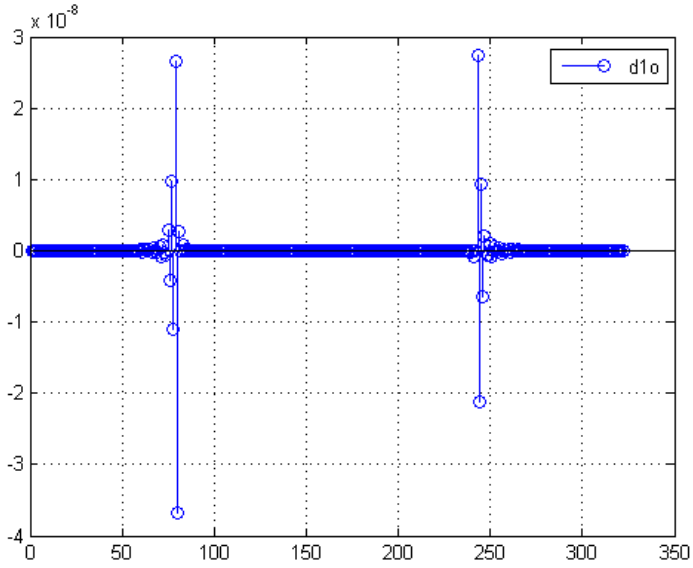


Figure D.4: The calculated errors of d_{1o} for 2-TIADC system.

The recreated delta-pulses for the 4-TIADC system

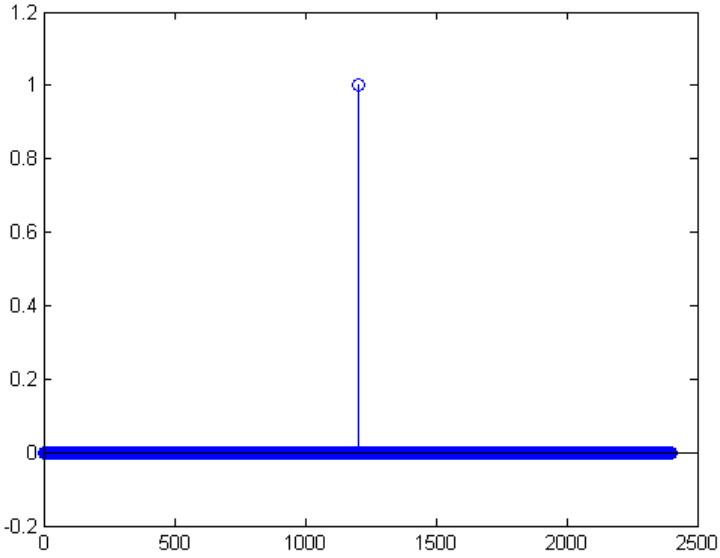


Figure D.5: The calculated d_{10} for the 4-TIADC system.

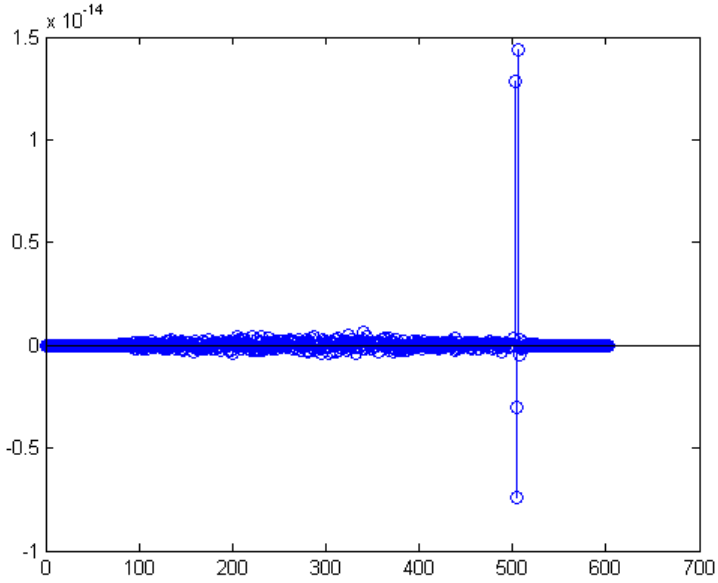


Figure D.6: The calculated errors of d_{10} for the 4-TIADC system.

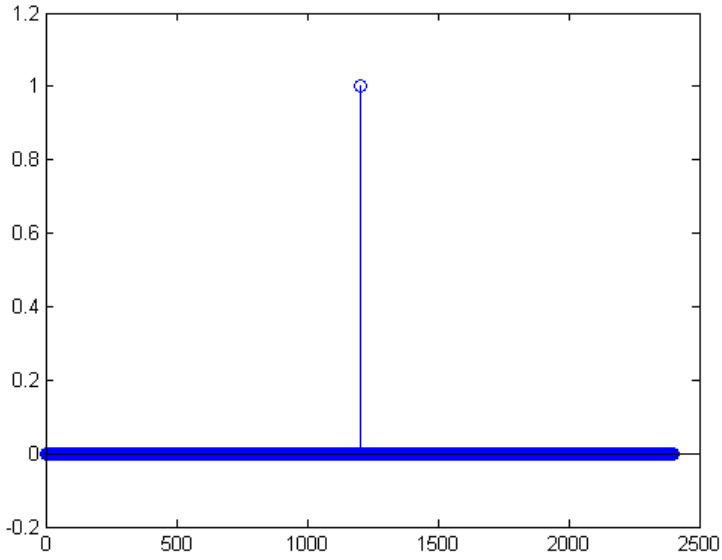


Figure D.7: The calculated d_{11} for the 4-TIADC system.

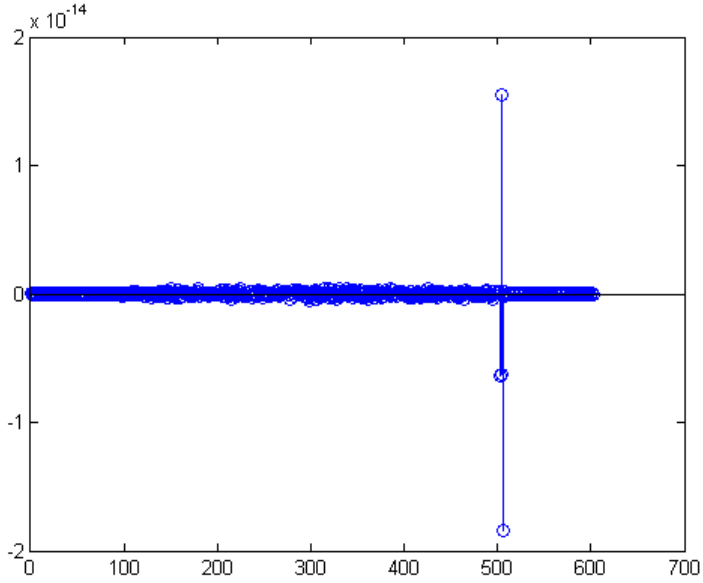


Figure D.8: The calculated errors of d_{11} for the 4-TIADC system.

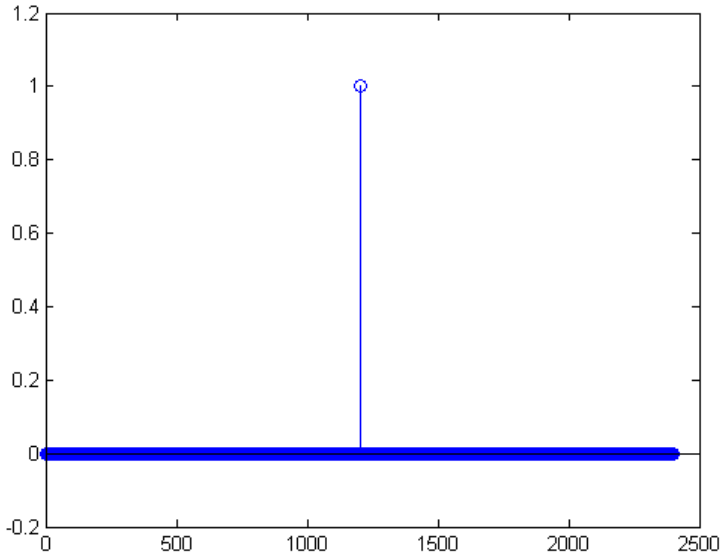


Figure D.9: The calculated d_{12} for the 4-TIADC system.

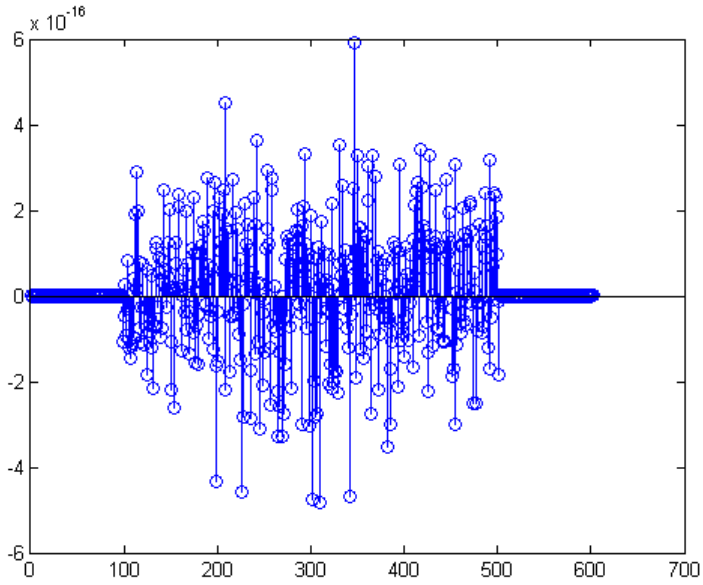


Figure D.10: The calculated errors of d_{12} for the 4-TIADC system.

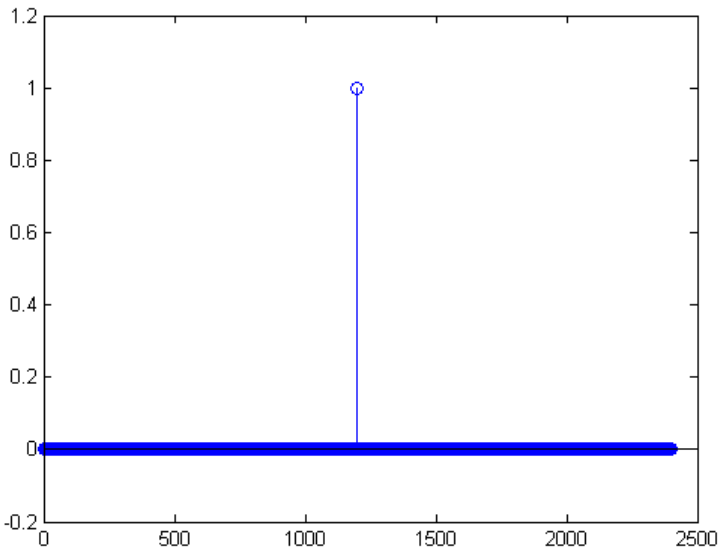


Figure D.11: The calculated d_{13} for the 4-TIADC system.

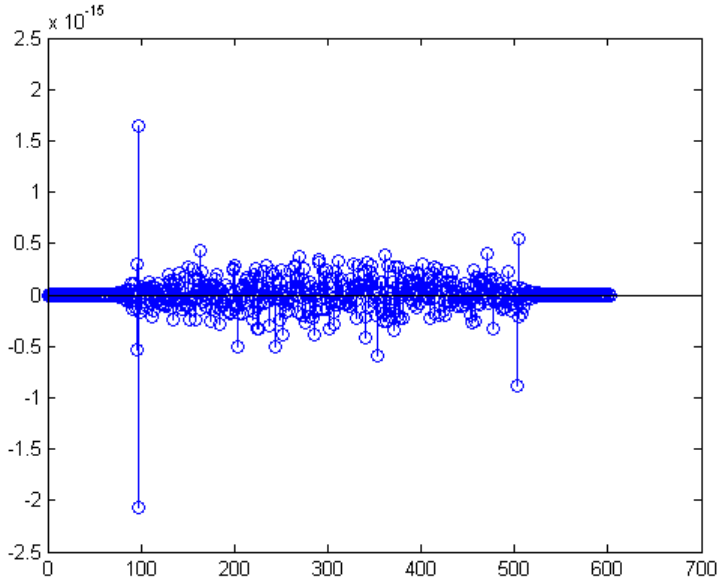


Figure D.12: The calculated errors of d_{13} for the 4-TIADC system.

Appendix E

Illustration figures - MSE

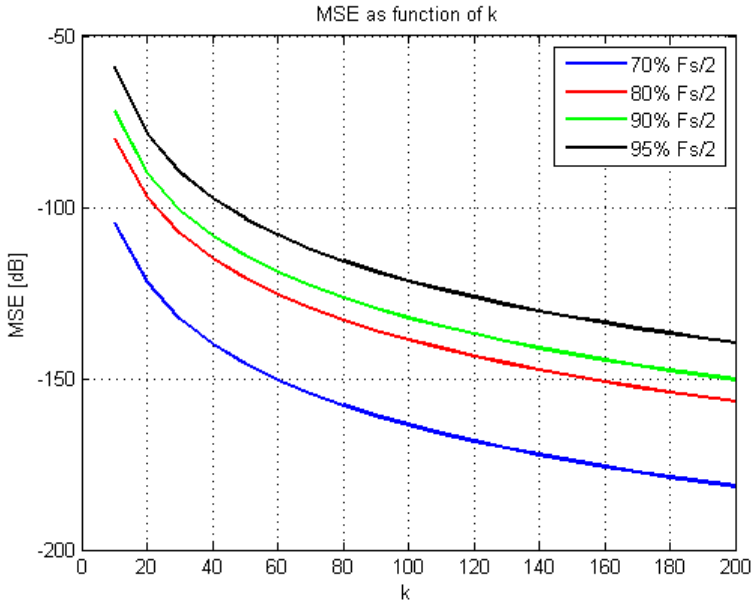


Figure E.1: Illustration of the MSE when reconstructing $y[n]$ for the 2-TIADC system

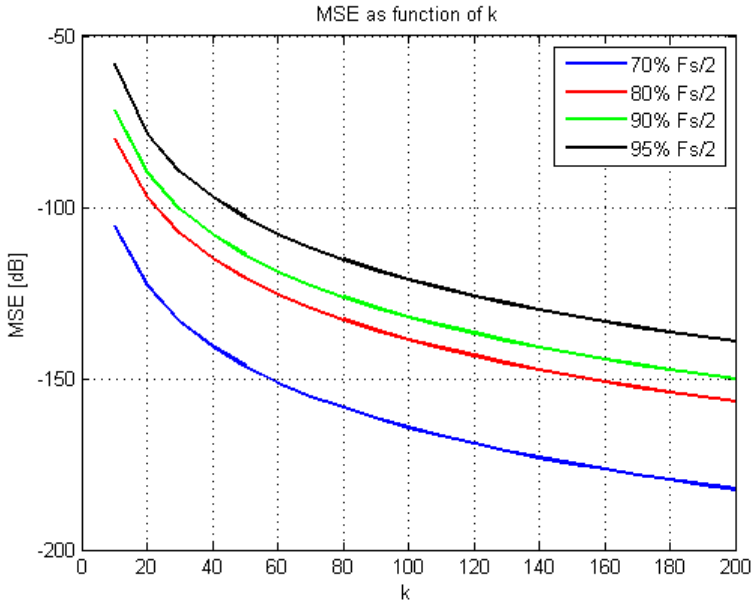


Figure E.2: Illustration of the MSE when reconstructing $x[n]$ for the 2-TIADC system

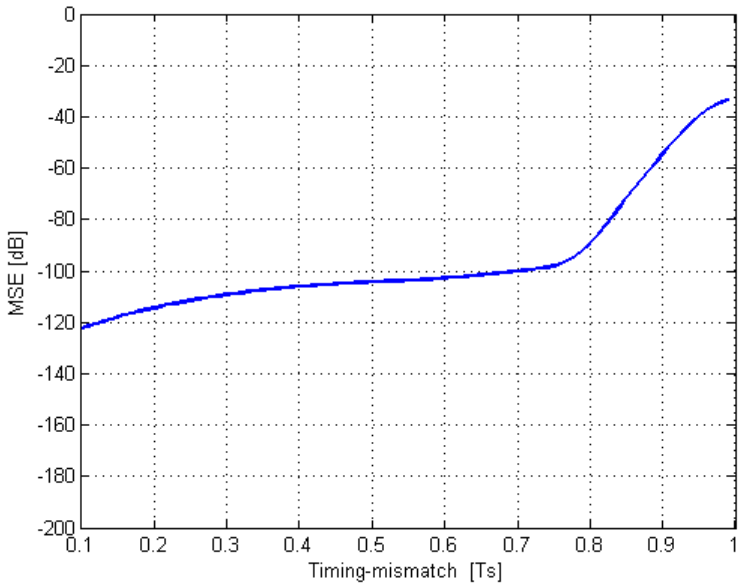


Figure E.3: Illustration of MSE as function of τ for k=25 for the 2-TIADC system

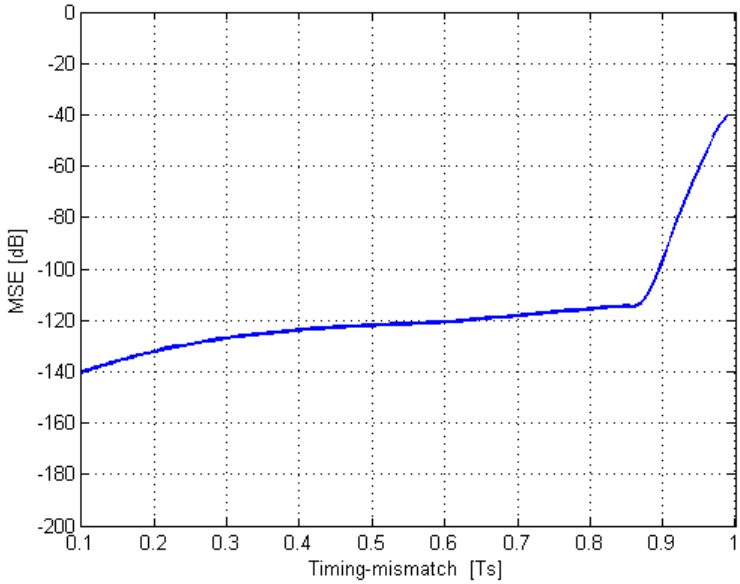


Figure E.4: Illustration of MSE as function of τ for k=50 for the 2-TIADC system

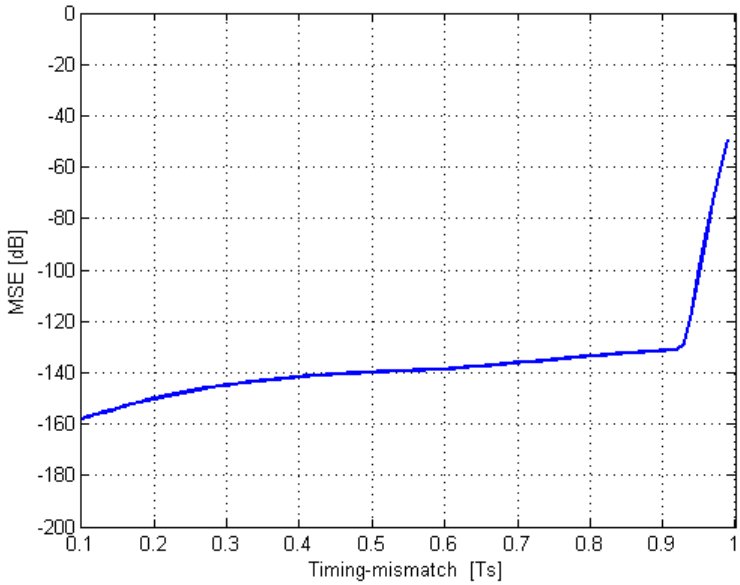


Figure E.5: Illustration of MSE as function of τ for k=100 for the 2-TIADC system

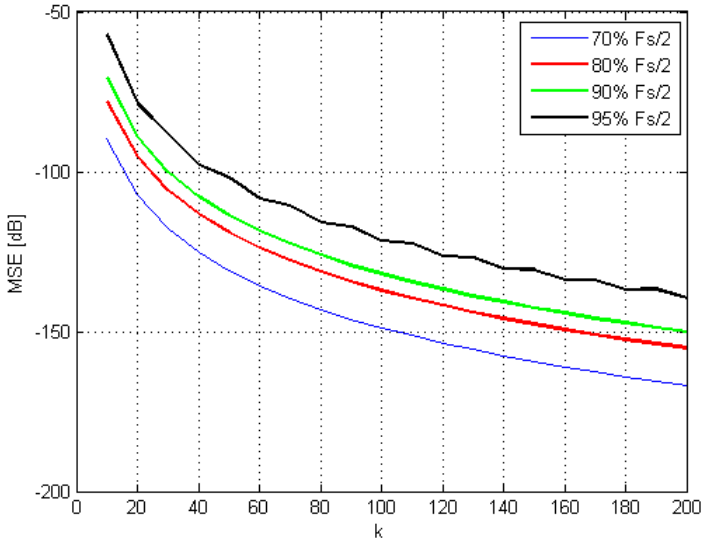


Figure E.6: Illustration of the MSE when reconstructing $y[n]$ for the 4-TIADC system

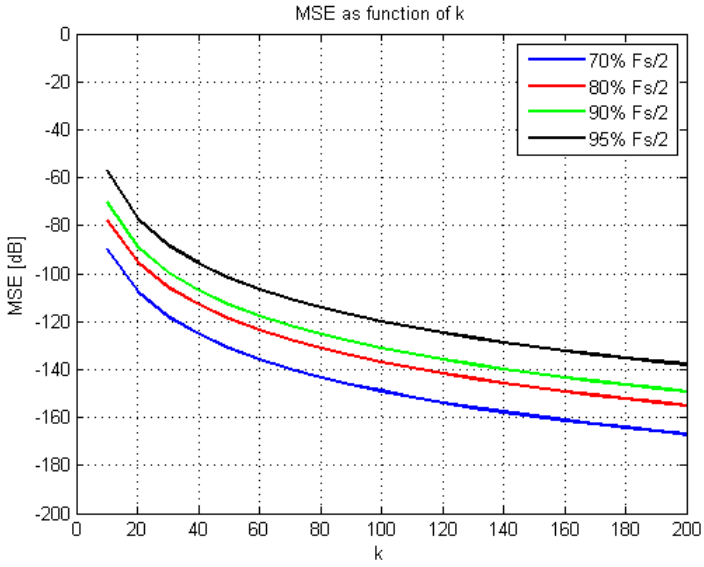


Figure E.7: Illustration of the MSE when reconstructing $x[n]$ for the 4-TIADC system

Appendix F

zip file attachment

Content of attached zip-file:

```
Matlab code\2ADC.m
Matlab code\4ADC.m
Matlab code\cosRollOffCont.m
Matlab code\mux2.m
Matlab code\mux4.m
```