R/2R DAC Nonlinearity Compensation

Gabriel Kulig
Gustav Wallin

LiTH-ISY-EX--12/4616--SE
Linköping 2012
R/$2R$ DAC Nonlinearity Compensation

Thesis for Degree of Master of Science
Department of Electrical Engineering
Division of Electronics Systems
Linköping University

by

Gabriel Kulig
Gustav Wallin

ISRN: LiTH-ISY-EX--12/4616--SE
**Title**  
R/2R DAC Nonlinearity Compensation

**Authors**  
Gabriel Kulig  
Gustav Wallin

**Abstract**  
The resistor ladder (R/2R) digital-to-analogue converter (DAC) architecture is often used in high performance audio solutions due to its low-noise performance. Even high-end R/2R DACs suffer from static nonlinearity distortions. It was suspected that compensating for these nonlinearities would be possible. It was also suspected that this could improve audio quality in audio systems using R/2R DACs for digital-to-analogue (A/D) conversion.

Through the use of models of the resistor ladder architecture a way of characterizing and measuring the faults in the R/2R DAC was created. A compensation algorithm was developed in order to compensate for the nonlinearities. The performance of the algorithm was simulated and an implementation of it was evaluated using an audio evaluation instrument.

The results presented show that it is possible to increase linearity in R/2R DACs by compensating for static nonlinearity distortions. The increase in linearity can be quite significant and audible for the trained ear.

**Keywords**  
digital-to-analogue, D/A, digital-to-analogue converter, DAC, static, nonlinearity, distortion, compensation
Project Identity

Master’s Thesis in Electrical Engineering, Spring 2012
Linköping University, Linköpings Tekniska Högskola, ISY
Prevas Linköping

Students:
Gabriel Kulig, gabku460@student.liu.se
Gustav Wallin, guswa432@student.liu.se

Examiner:
Oscar Gustafsson, oscarg@isy.liu.se

Supervisor:
Niklas Andersson, niklasa@isy.liu.se

Supervisors at Prevas:
Hjalmar Nilsson, hjalmar.nilsson@prevas.se
Joacim Frisk, joacim.frisk@prevas.se
Abstract

The resistor ladder (R/2R) digital-to-analogue converter (DAC) architecture is often used in high performance audio solutions due to its low-noise performance. Even high-end R/2R DACs suffer from static nonlinearity distortions. It was suspected that compensating for these nonlinearities would be possible. It was also suspected that this could improve audio quality in audio systems using R/2R DACs for digital-to-analogue (A/D) conversion.

Through the use of models of the resistor ladder architecture a way of characterizing and measuring the faults in the R/2R DAC was created. A compensation algorithm was developed in order to compensate for the nonlinearities. The performance of the algorithm was simulated and an implementation of it was evaluated using an audio evaluation instrument.

The results presented show that it is possible to increase linearity in R/2R DACs by compensating for static nonlinearity distortions. The increase in linearity can be quite significant and audible for the trained ear.
Acknowledgements

First of all we would like to acknowledge our examiner Oscar Gustafsson. We would like to thank Niklas Andersson for supervising and supporting us throughout the thesis work.

We would like to thank Hjalmar Nilsson for his support and help with practical matters. We would also like to thank Joacim Frisk for insightful discussions.

Finally we would like to thank Naim Audio for making this thesis possible.
Contents

1 Introduction ..................................................................................................................................... 1
  1.1 Background .............................................................................................................................. 1
  1.2 Purpose ...................................................................................................................................... 1
  1.3 Overview .................................................................................................................................. 1
2 DAC Introduction ............................................................................................................................ 2
  2.1 Introduction .............................................................................................................................. 2
  2.2 The Functionality of the DAC ............................................................................................... 2
  2.3 Reconstruction .......................................................................................................................... 2
    2.3.1 Ideal Reconstruction Using the Sinc Function ............................................................... 2
    2.3.2 Reconstruction in Practice ......................................................................................... 3
  2.4 DAC Properties ......................................................................................................................... 4
    2.4.1 Static Properties ................................................................................................................. 4
      2.4.1.1 Offset Error ........................................................................................................... 4
      2.4.1.2 Gain Error ........................................................................................................... 4
      2.4.1.3 Differential Nonlinearity (DNL) Error ............................................................... 5
      2.4.1.4 Integral Nonlinearity (INL) Error ...................................................................... 5
      2.4.1.5 Monotonicity ...................................................................................................... 5
    2.4.2 Frequency-Domain Properties ......................................................................................... 6
      2.4.2.1 Signal-to-Noise Ratio (SNR) ............................................................................. 6
      2.4.2.2 Total Harmonic Distortion (THD) ..................................................................... 6
      2.4.2.3 Signal-to-Noise-and-Distortion Ratio (SNDR) .................................................. 6
      2.4.2.4 Total Harmonic Distortion Plus Noise (THD+N) .............................................. 6
      2.4.2.5 Spurious-Free Dynamic Range (SFDR) ............................................................. 7
      2.4.2.6 Effective Number of Bits (ENOB) ..................................................................... 7
  3 The R/2R DAC ............................................................................................................................... 8
  3.1 Introduction .............................................................................................................................. 8
  3.2 R/2R Structure Overview ........................................................................................................ 8
    3.2.1 A Simple Example .......................................................................................................... 9
  3.3 R/2R Ladder Structure With Unity Current Sources ................................................................ 13
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.1 Analysis of the R/2R Ladder Structure</td>
<td>13</td>
</tr>
<tr>
<td>3.3.2 Mathematical Proof of Binary Behaviour</td>
<td>17</td>
</tr>
<tr>
<td>3.3.2.1 Application of Derived Formulas</td>
<td>17</td>
</tr>
<tr>
<td>3.3.2.2 Proof of Binary Behaviour Using Mathematical Induction</td>
<td>18</td>
</tr>
<tr>
<td>3.3.3 Examination of Errors in the R/2R Ladder using Examples</td>
<td>21</td>
</tr>
<tr>
<td>3.3.3.1 Errors in Current Sources</td>
<td>21</td>
</tr>
<tr>
<td>3.3.3.2 Errors in Resistors</td>
<td>24</td>
</tr>
<tr>
<td>4 Measurements</td>
<td>28</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>28</td>
</tr>
<tr>
<td>4.2 The Discrete Fourier Transform</td>
<td>28</td>
</tr>
<tr>
<td>4.3 Sampling Theory</td>
<td>31</td>
</tr>
<tr>
<td>4.4 Measurement Platform</td>
<td>36</td>
</tr>
<tr>
<td>4.4.1 The DAC System</td>
<td>36</td>
</tr>
<tr>
<td>4.4.2 The Physical Setup</td>
<td>37</td>
</tr>
<tr>
<td>4.4.3 Data Sampling</td>
<td>38</td>
</tr>
<tr>
<td>4.5 Test Patterns and Data Extraction</td>
<td>39</td>
</tr>
<tr>
<td>5 The DAC Chip Model</td>
<td>44</td>
</tr>
<tr>
<td>5.1 Introduction</td>
<td>44</td>
</tr>
<tr>
<td>5.2 Internal DAC Chip Structure</td>
<td>44</td>
</tr>
<tr>
<td>5.2.1 R/2R Behaviour</td>
<td>44</td>
</tr>
<tr>
<td>5.2.2 Dual DAC Structure</td>
<td>45</td>
</tr>
<tr>
<td>5.2.2.1 The Complementary Architecture</td>
<td>45</td>
</tr>
<tr>
<td>5.2.2.2 The Offset Current Architecture</td>
<td>45</td>
</tr>
<tr>
<td>5.2.2.3 Usage of DAC Architectures in Simulations</td>
<td>46</td>
</tr>
<tr>
<td>6 Nonlinearity Compensation</td>
<td>47</td>
</tr>
<tr>
<td>6.1 Introduction</td>
<td>47</td>
</tr>
<tr>
<td>6.2 Algorithm Description</td>
<td>47</td>
</tr>
<tr>
<td>6.2.1 Description Using Text and Graphics</td>
<td>47</td>
</tr>
<tr>
<td>6.2.2 Pseudo Code Description</td>
<td>49</td>
</tr>
<tr>
<td>6.3 Algorithm Behaviour and Properties</td>
<td>49</td>
</tr>
<tr>
<td>6.3.1 Underestimating Property</td>
<td>49</td>
</tr>
<tr>
<td>6.3.2 Reduction of Static Distortions</td>
<td>50</td>
</tr>
<tr>
<td>6.3.3 Information Loss</td>
<td>51</td>
</tr>
</tbody>
</table>
6.3.4 Compression of Dynamic Range

6.4 Possible Improvements

6.4.1 Preservation of Detail

6.4.2 Always Check the Next Codeword

6.4.3 Normalizing Bit Levels Differently

7 Simulation, Evaluation and Test Results

7.1 Introduction

7.2 Simulations

7.2.1 Simulations of the K-grade PCM1704

7.2.1.1 Simulation With 0 dBFS Input Amplitude

7.2.1.2 Simulation With -20 dBFS Input Amplitude

7.2.2 Simulations of the Non-graded PCM1704

7.2.2.1 Simulation With 0 dBFS Input Amplitude

7.2.2.2 Simulation With -20 dBFS Input Amplitude

7.3 Evaluation

7.3.1 Evaluation Using Measurement Setup

7.3.2 Algorithm Implementation

7.3.3 Evaluation Using dScope Series III

7.3.3.1 FFT Plots

7.3.3.2 Sweeps

7.4 Discussion of Simulation and Evaluation Results

8 Conclusion and Future Work

8.1 Introduction

8.2 Conclusion

8.3 Future work

9 References
## Nomenclature

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC</td>
<td>Alternating current</td>
</tr>
<tr>
<td>DAC</td>
<td>Digital-to-analogue converter</td>
</tr>
<tr>
<td>dBFS</td>
<td>Decibels relative to full scale</td>
</tr>
<tr>
<td>DC</td>
<td>Direct current</td>
</tr>
<tr>
<td>DFT</td>
<td>Discrete Fourier transform</td>
</tr>
<tr>
<td>DNL</td>
<td>Differential nonlinearity error</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital signal processor</td>
</tr>
<tr>
<td>ENOB</td>
<td>Effective number of bits</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier transform</td>
</tr>
<tr>
<td>FPGA</td>
<td>Field programmable gate array</td>
</tr>
<tr>
<td>IDFT</td>
<td>Inverse Discrete Fourier transform</td>
</tr>
<tr>
<td>KCL</td>
<td>Kirchoff’s current law</td>
</tr>
<tr>
<td>KVL</td>
<td>Kirchoff’s voltage law</td>
</tr>
<tr>
<td>LSB</td>
<td>Least significant bit</td>
</tr>
<tr>
<td>MSB</td>
<td>Most significant bit</td>
</tr>
<tr>
<td>OP-amp</td>
<td>Operational amplifier</td>
</tr>
<tr>
<td>PAM</td>
<td>Pulse-amplitude modulation</td>
</tr>
<tr>
<td>SA</td>
<td>Successive approximation</td>
</tr>
<tr>
<td>SFDR</td>
<td>Spurious-free dynamic range</td>
</tr>
<tr>
<td>SNDR</td>
<td>Signal-to-noise-and-distortion ratio</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-noise ratio</td>
</tr>
<tr>
<td>TDFT</td>
<td>Time-discrete Fourier transform</td>
</tr>
<tr>
<td>THD</td>
<td>Total harmonic distortion</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Background
In most digital audio products the last stage, to convert the digital signal to the analogue domain, is a DAC. The resistor ladder (R/2R) DAC architecture is often used in high performance audio solutions due to its low-noise performance. This architecture is also considered to give a better sound experience than other DAC architectures. It is hard to define what constitutes a good sound experience and thus it is difficult to measure this property. However it is easy to verify that even a really good R/2R DAC suffers from nonlinearity distortions. It is also fairly easy to derive that non-ideal component values within the R/2R ladder contributes to these errors. Nonlinearity is a measurable property and it is reasonable to assume that something that affects the spectrum of the audio signal also will affect the sound experience. With this background it was suspected that an increase in linearity of the DAC would enhance the sound experience.

1.2 Purpose
The purpose of this master’s thesis was to examine the possibility of creating a method for increasing the linearity of digital-to-analogue conversion in high performance audio applications. Ultimately this method would be used for enhancing the audio experience in a high fidelity DAC system from Naim Audio. It was assumed that with knowledge about the nonlinearities and the structure of the used DAC chip it would be possible to create and implement an algorithm that can compensate for its distortions. This algorithm would then be implemented in hardware or software and a functioning prototype would be developed. The algorithm would then be tested using audio evaluation equipment.

1.3 Overview
Chapter 2 introduces the reader to the DAC concept and the different properties of the DAC. In chapter 3 the R/2R DAC and its properties are discussed. Chapter 4 describes background theory behind measuring signals digitally and the measurements that was performed on the DAC chip. Chapter 5 describes the models of the DAC chip that were used throughout the thesis. In chapter 6 the nonlinearity compensation algorithm that was developed is presented. Chapter 7 includes results from evaluation measurements. Finally chapter 8 includes a discussion and conclusion.
2 DAC Introduction

2.1 Introduction
In practically all of today’s modern audio equipment the audio source is digital. Before one can listen to the audio signal it must be converted from the digital to the analogue domain. The main component in this conversion is the digital-to-analogue converter (DAC). The DAC concept is introduced to the reader in this chapter.

2.2 The Functionality of the DAC
The basic function of a DAC is keeping the analogue output signal constant and equal to the digital input during one sample period $T = 1/f_s$. The output for an $N$-bit DAC held during a sample period $T$ is given by

$$A(nT) = \sum_{k=0}^{N-1} w_k \cdot b_k(nT) \quad (2.1)$$

where $b_k(nT)$ is the k-th bit of the input word at time $nT$ and $w_k$ is the weight of this bit. For a DAC with an offset binary-coded input the weights are given by

$$w_k = 2^k \cdot \delta, 0 \leq k \leq N - 1 \quad (2.2)$$

where $\delta$ is the analogue output value of the LSB. A DAC were each weight is twice as large as the preceding, like the one in (2.2), is referred to as binary weighted.

2.3 Reconstruction
Reconstruction is the process where an analogue signal is recreated from a digital signal. If no information is lost during the process the reconstruction is said to be ideal.

2.3.1 Ideal Reconstruction Using the Sinc Function
Ideal reconstruction of a digital signal as discussed in [2] can be done through pulse-amplitude modulation (PAM) of the digital signal with infinitely many sinc functions according to the formula

$$x_r(t) = \sum_{n=-\infty}^{\infty} x[n] \cdot \text{sinc} f_s(t - nT) \quad (2.3)$$

where $x_r(t)$ is the reconstructed signal, $x[n]$ is the digital signal, $f_s$ is the sample frequency and $T$ is the sample period. The sinc function is defined as

$$\text{sinc} \omega = \frac{\sin \pi \omega}{\pi \omega} \quad (2.4)$$

and can be seen in Figure 2.1. It is impossible to create an implementation of the PAM
described in (2.3) since it would have to be non-causal due to the infinite dilation of the sinc function in the time domain.

### 2.3.2 Reconstruction in Practice

As described in [2] the functionality of a DAC can be modelled as PAM of the digital input signal with square pulses (shown in Figure 2.3). The output signal from the DAC has a sinc-weighted frequency spectrum with infinite dilation. This can be realized intuitively seen by comparing the Fourier transforms of the sinc function and the square pulse, shown in Figure 2.2 and Figure 2.4. If the output signal of the DAC is filtered using a reconstruction filter an ideal recreated signal could be acquired. In practical reconstruction applications DACs are used together with an approximation of the ideal reconstruction filter, the low-pass filter.

Figure 2.1: The sinc function.

Figure 2.2: The Fourier transform of the sinc function.

Figure 2.3: The square pulse.

Figure 2.4: The Fourier transform of the square pulse.
2.4 DAC Properties
In this section different properties of DACs are introduced. These are used for measuring DAC performance.

2.4.1 Static Properties
The ideal input-output characteristic of a DAC is a staircase with uniform step size over the complete dynamic range [1]. This means that each analogue step should be equal to the output of one LSB. The input-output characteristics of an ideal and non-ideal 3-bit DAC are shown in Figure 2.5.

![Figure 2.5: Ideal and non-ideal input-output characteristics of a 3-bit DAC.](image)

2.4.1.1 Offset Error
An offset error is equal to a non-zero output for a zero-input (denoted $d_0$ in Figure 2.5). It corresponds to a DC offset and could be removed by DC blocking the signal.

2.4.1.2 Gain Error
The static gain error affects the gain of the DAC throughout the dynamic range. It can be either linear or nonlinear (illustrated in Figure 2.6). A linear gain error does not change the linearity of the DAC since it corresponds to an increase or decrease in amplification.
2.4.1.3 Differential Nonlinearity (DNL) Error

The differential nonlinearity is used to describe how the transitions in the dynamic range deviate from the ideal step size. The output difference between the non-ideal and ideal DAC for a input code $k$ is denoted $d_k$ and is defined as

$$d_k = A_{\text{real}}(k) - A_{\text{ideal}}(k)$$  \hspace{1cm} (2.5)

where $A_{\text{ideal}}(k)$ and $A_{\text{real}}(k)$ are the output signals, for input code $k$, of the ideal and non-ideal DACs respectively (as illustrated in Figure 2.5). The output difference for input code 0, called $d_0$, is the offset error. DNL is defined as

$$DNL_k = d_k - d_{k-1}, \quad k \geq 1$$  \hspace{1cm} (2.6)

2.4.1.4 Integral Nonlinearity (INL) Error

The integral nonlinearity is defined as the deviation of the non-ideal output, with the offset and linear gain error removed, from a straight line. The INL is defined this way since offset and linear gain errors do not degrade the linearity of the DAC. There are several ways of removing the offset and gain error when calculating the INL [4]. The method used in this thesis is to create a best-fitted line using the least square method as described in [3]. The best-fitted line is denoted $A_{\text{best-fit}}(k)$ and the INL is calculated as

$$INL_k = A_{\text{real}}(k) - A_{\text{best-fit}}(k)$$  \hspace{1cm} (2.7)

2.4.1.5 Monotonicity

A DAC is said to be monotonic if its output is guaranteed to increase with an increasing input. According to [4] and [5] a DAC is monotonic if

$$\max |DNL_k| \leq 1 \text{ LSB}$$  \hspace{1cm} (2.8)
and

$$\max |INL_k| \leq 0.5 \text{ LSB} \quad (2.9)$$

holds true. However, a DAC might still be monotonic even if (2.8) and (2.9) are not met [5].

2.4.2 Frequency-Domain Properties

2.4.2.1 Signal-to-Noise Ratio (SNR)
The signal-to-noise ratio describes the relationship between the signal and noise power. SNR expressed in dB is defined as

$$SNR = 10 \cdot \log_{10} \left( \frac{P_{signal}}{P_{noise}} \right) \quad (2.10)$$

where $P_{signal}$ and $P_{noise}$ is the signal and noise power respectively.

2.4.2.2 Total Harmonic Distortion (THD)
Total harmonic distortion is the ratio between the total power of harmonic distortions and the signal power. THD expressed in dB is defined as

$$THD = 10 \cdot \log_{10} \left( \frac{\sum_{k=2}^{\infty} P_k}{P_{signal}} \right) \quad (2.11)$$

where $P_{signal}$ is the signal power and $P_k$ is the power of the k-th harmonic.

2.4.2.3 Signal-to-Noise-and-Distortion Ratio (SNDR)
The signal-to-noise-and-distortion ratio is a combination of SNR and THD. It accounts for both the noise and harmonic distortions. The definition of SNDR expressed in dB is

$$SNDR = 10 \cdot \log_{10} \left( \frac{P_{signal}}{P_{noise} + \sum_{k=2}^{\infty} P_k} \right) \quad (2.12)$$

2.4.2.4 Total Harmonic Distortion Plus Noise (THD+N)
Total harmonic distortion plus noise describes the ratio between the total power of harmonic distortions plus noise and the signal power. THD expressed in dB is defined as

$$THD + N = 10 \cdot \log_{10} \left( \frac{P_{noise} + \sum_{k=2}^{\infty} P_k}{P_{signal}} \right) \quad (2.13)$$

which is equal to the negation of the SNDR. THD+N is usually given in dB or as a percentage.
2.4.2.5 **Spurious-Free Dynamic Range (SFDR)**

The spurious-free dynamic range describes the power difference between the signal and the largest spurious (unwanted) spectral component, which usually is a harmonic distortion. SFDR expressed in dB is defined as

\[
SFDR = 10 \cdot \log_{10}\left(\frac{P_{\text{signal}}}{P_x}\right)
\]  

(2.14)

where \(P_x\) is the power of the largest unwanted spectral component. An example of a frequency spectrum with SFDR marked is shown in Figure 2.7.

![Example frequency spectrum](image)

**Figure 2.7: Example frequency spectrum.**

2.4.2.6 **Effective Number of Bits (ENOB)**

Effective number of bits is a way of measuring the SNDR in bits. The relation between ENOB and SNDR is described in [4] as

\[
ENOB = \frac{\text{SNDR} - 1.76}{6.02}
\]

(2.15)
3 The R/2R DAC

3.1 Introduction
The R/2R DAC offers a way of converting a digital signal to an analogue signal while using only approximately \( 3n \) resistors, where \( n \) is the number of input bits. The output from an R/2R DAC can be either a current or a voltage. The following chapter will focus on the properties and characteristics of R/2R DACs.

3.2 R/2R Structure Overview
The principle behind an R/2R DAC is that a reference current or voltage is fed to a resistor ladder structure. It is divided and controlled such that only a desired portion of the reference current/voltage propagates to the output. Another important feature of the R/2R DAC is that the ladder is binary weighted, which is a direct consequence of current/voltage sharing over the resistors. This means that the input bits of the ladder will represent a binary fractal value. The fractal value tells how much of the reference current/voltage that carries on to the output. For example, an input word of \( 111_{\text{BIN}} \) will correspond to that the fraction \( 0.111_{\text{BIN}} = 7/8 \) of the reference voltage or current will propagate to the output. The schematic in Figure 3.1 shows a model of the R/2R DAC found in [1]. This model is used as reference in this chapter when the functionality and the characteristics of the R/2R DAC are explained and/or highlighted.

![Figure 3.1: Schematic of the reference ladder.](image)

The R/2R ladder shown in Figure 3.1 is of current-steering type. It is fed a reference current denoted \( I_{\text{ref}} \). The rightmost 2R resistor yields the MSB current and the second leftmost yields the LSB current. The resistor closest to the output serves as a termination resistance to the ladder. In Figure 3.1 are a number of logic switches. Through these, parts of \( I_{\text{ref}} \) can be added or removed from the output. Each of these switches are controlled by an input bit, if the bit is active the switch will connect the corresponding part of \( I_{\text{ref}} \) to the output. The most significant bit (MSB) controls the rightmost switch and the least significant bit (LSB) controls...
the leftmost switch. The parts of $I_{ref}$ not added to the output will be fed into a grounded connection. Shown in Figure 3.2 below is a generic node of the ladder. As can be seen in the figure, the current flowing into the node is shared between the shunt resistor and the sum of the serial and input resistance of the remaining part of the ladder.

![Figure 3.2: A generic node of the R/2R-ladder.](image)

### 3.2.1 A Simple Example

Let us examine the functionality by examining a 3-bit DAC receiving a certain input. Consider the 3-bit R/2R ladder shown in Figure 3.3. The input is $111_{BIN}$ corresponding to all switches connecting the R/2R ladder to the output.

![Figure 3.3: A 3-bit R/2R ladder with all switches in on-position.](image)
Only the termination resistor is connected to ground. Calculations of currents are done node-wise and the results are then added to the output through the use of Kirchoff’s current law (KCL). Node and current names in the R/2R ladder have been introduced in Figure 3.4.

**Figure 3.4: The R/2R ladder with node and current names.**

The connection to $I_{out}$ in each is replaced with a connection to ground since the output of the ladder is connected to the virtual ground of the om-amp in practical uses [1].

**Calculations for node 3**

A schematic of node 3 is shown in Figure 3.5.

**Figure 3.5: Schematic for node 3 calculations.**

Current sharing gives

$$I_3 = \frac{2R}{2R + 2R} \cdot I_5 = \frac{I_5}{2}$$

(3.1)

The equivalent resistance of node 3, denoted $R_{eq,3}$, is calculated as
Calculations for node 2
A schematic of node 2 is shown in Figure 3.6.

![Figure 3.6: Schematic for node 2 calculations.](image)

The value of $R_{eq,3}$ is taken from (3.2). Current sharing over the shunt resistor gives

$$I_2 = \frac{R + R_{eq,3}}{2R + R + R_{eq,3}} \cdot I_4 = \frac{2R}{4R} \cdot I_4 = \frac{I_4}{2}$$  \hspace{1cm} (3.3)

The equivalent resistance of node 2 is calculated as

$$R_{eq,2} = \left( \frac{1}{R + R_{eq,3}} + \frac{1}{2R} \right)^{-1} = \left( \frac{1}{2R} + \frac{1}{2R} \right)^{-1} = \left( \frac{1}{R} \right)^{-1} = R$$  \hspace{1cm} (3.4)

At this point it will be of interest to calculate how much of the incoming current that propagates to node number 3. In this case we get

$$I_4 - I_2 - I_5 = 0 \Leftrightarrow I_4 = I_2 + I_5 \Leftrightarrow$$

$$I_5 = I_4 - I_2 = \{ (3.3) \} = I_4 - \frac{I_4}{2} = \frac{I_4}{2}$$  \hspace{1cm} (3.5)

Calculations for node 1
A schematic of node 1 is shown in Figure 3.7.
Once again current sharing gives \[ I_1 = \frac{R + R_{eq2}}{2R + R + R_{eq2}} \cdot I_{Ref} = \{ (3.4) \} = \frac{2R}{4R} \cdot I_{Ref} = \frac{I_{Ref}}{2} \] (3.6)

When KCL is applied we get.

\[
\begin{align*}
I_{ref} - I_4 - I_1 &= 0 \\
I_4 &= I_{ref} - I_1 - \frac{I_{ref}}{2} = \frac{I_{ref}}{2} 
\end{align*}
\] (3.7)

Using the expressions for the currents \( I_1 \) and \( I_4 \) expressed in terms of \( I_{ref} \) the other node currents can be calculated using KCL. Using (3.3) we get

\[ I_2 = \frac{I_4}{2} = \{ (3.7) \} = \frac{1}{2} \cdot \frac{I_{ref}}{2} = \frac{I_{ref}}{4} \] (3.8)

(3.5) gives us

\[ I_5 = \frac{I_4}{2} = \{ (3.7) \} = \frac{I_{ref}}{4} \] (3.9)

and finally (3.1) gives

\[ I_3 = \frac{I_5}{2} = \{ (3.9) \} = \frac{I_{ref}}{8} \] (3.10)

As can be seen in the schematic of the entire ladder, i.e. the one shown in Figure 3.4, \( I_{out} \) is the sum of the currents \( I_1, I_2 \) and \( I_3 \). Using (3.6), (3.8) and (3.10) we get

\[ I_{out} = I_1 + I_2 + I_3 = \left( \frac{1}{2} + \frac{1}{4} + \frac{1}{8} \right) \cdot I_{ref} = \frac{7}{8} \cdot I_{ref} \] (3.11)

This is the expected output value from the DAC as explained in the beginning of the chapter.
The calculations used above are basically the same as those that are applicable for more advanced R/2R ladders.

3.3 R/2R Ladder Structure with Unity Current Sources.
The model presented in section 3.2 had one single large current or voltage input. The drawback with such a model is that relatively large currents will pass through the MSB node and those close to it. In [4] this problem is described and the use of differently sized switches is mentioned as a normal way of coping with this problem. A way of removing the need for different sized switches is to replace the reference current source with a number of equally sized current sources placed as shown in Figure 3.8 below. In [4] it is pointed out that this will often reduce the speed of the circuit, however for some applications the speed of the DAC circuit will not be an issue. An advantage with such a structure is that the current sources can be closely matched. If the first bit, that is the MSB, is placed outside of the R/2R ladder the magnitude of the unit current sources can be halved while the functionality is still maintained.

![Figure 3.8: The R/2R ladder structure with equal current sources.](image)

3.3.1 Analysis of the R/2R Ladder Structure
The structure in Figure 3.8 has been redrawn in Figure 3.9 with different notations. The number of input bits is equal to $N$. If a bit in the incoming data word is enabled the corresponding current source is connected to the R/2R ladder, if not the current source is connected to ground. The most significant bit (MSB) controls the rightmost switch and the least significant bit (LSB) controls the leftmost switch.
Since $R_{op}$ is small when compared to the input impedance of the operational amplifier practically no current will be drawn through the op-amp. $I_{out}$ will therefore be equal to the current drawn through $R_{op}$. When only the MSB is enabled $I_{out}$ will be equal to the current $I$.

Consider the structure when only one current source, $b_M$ ($M < N$), is connected to the R/2R ladder (only one bit enabled). A simplified circuit illustrating this is shown in Figure 3.10. The op-amp connection has been replaced with a connection to ground since its negative input acts as virtual ground [9].

Simplification of the circuit makes it easier to calculate the output current, $I_{out}$. $R_{T,1}$ and $R_{S,1}$ are coupled in parallel with $R_{T,2}$ between node 2 and ground. These resistors can be replaced by an equivalent resistance $R_{R,left,2}$ where

$$R_{R,left,2} = \left( R_{T,1} + R_{S,1} \right) || R_{T,2} \quad (3.12)$$

In the same way $R_{R,left,2}$ and $R_{S,2}$ are coupled in parallel with $R_{T,3}$ and can be replaced by an equivalent resistance between node 3 and ground. Since the structure is repetitive we can generalize (3.12) by first assigning
\[ R_{R,\text{left},1} = R_{T,1} \]  

(3.13)

and then get

\[ R_{R,\text{left},n} = (R_{R,\text{left},n-1} + R_{S,n-1})||R_{T,n}, \quad n \in [2, M] \]

(3.14)

Using (3.14) all resistors to the left of node \( M \) can be replaced by \( R_{R,\text{left},M} \). A schematic where this has been done is shown in Figure 3.11.

![Figure 3.11: Simplification of the R/2R ladder with a single current source.](image)

In order to calculate \( I_{\text{out}} \) we first calculate \( I_M, \ldots, I_{N-2} \). Further simplification of the circuit allows us to do this in a fashionable manner. Let

\[ R_{R,\text{right},N} = R_{S,N-1} \]

(3.15)

\( R_{R,\text{right},N} \) is then coupled in parallel with \( R_{T,N-1} \) and the combination of these is coupled in series with \( R_{S,N-2}, R_{R,\text{right},N}, R_{T,N-1} \) and \( R_{S,N-2} \) can be replaced with \( R_{R,\text{right},N-1} \), where

\[ R_{R,\text{right},N-1} = (R_{R,\text{right},N}||R_{T,N-1}) + R_{S,N-2} \]

(3.16)

We can generalize (3.16) by letting

\[ R_{R,\text{right},n} = (R_{R,\text{right},n+1}||R_{T,n}) + R_{S,n-1}, \quad n \in [M + 1, N - 1] \]

(3.17)

\( R_{R,\text{right},N} \) is given by (3.15). Using (3.17) we can replace all the resistors to the right of node \( M \) with \( R_{R,\text{right},M+1} \). This is illustrated in Figure 3.12.
Now $I_M$ can be calculated using current division. We get

$$I_M = \frac{R_{R,\text{left},M}}{R_{R,\text{left},M} + R_{R,\text{right},M+1}} \cdot I \quad (3.18)$$

Now that $I_M$ is known $I_{M+1}$ can be calculated as

$$I_{M+1} = \frac{R_{R,M+1}}{R_{R,M+1} + R_{R,\text{right},M+2}} \cdot I_M \quad (3.19)$$

The generalisation of (3.19) becomes

$$I_n = \frac{R_{T,n}}{R_{T,n} + R_{R,\text{right},n+1}} \cdot I_{n-1}, \quad n \in [M + 1, N - 1] \quad (3.20)$$

Using (3.20) the general formula for calculating a current $I_k$ can be written as

$$I_k = \left( \prod_{n=M+1}^{k} \frac{R_{T,n}}{R_{T,n} + R_{R,\text{right},n+1}} \right) \cdot \frac{R_{R,\text{left},M}}{R_{R,\text{left},M} + R_{R,\text{right},M+1}} \cdot I, \quad \begin{cases} M \in [1, N-2] \\ k \in [M+1, N-1] \end{cases} \quad (3.21)$$

For $M = N-1$, $I_{N-1}$ can be calculated using (3.18). $I_{out}$ is equal to $I_{N-1}$ as can be seen in Figure 3.11. We can now write a general formula for $I_{out}$ by combining (3.18) and (3.21) into

$$I_{out} = \begin{cases} \left( \prod_{n=M+1}^{N-1} \frac{R_{T,n}}{R_{T,n} + R_{R,\text{right},n+1}} \right) \cdot \frac{R_{R,\text{left},M}}{R_{R,\text{left},M} + R_{R,\text{right},M+1}} \cdot I, \quad M \in [1, N-2] \\ \frac{R_{R,\text{left},M}}{R_{R,\text{left},M} + R_{R,\text{right},M+1}} \cdot I, \quad M = N-1 \end{cases} \quad (3.22)$$

Since all of the components in the circuit are linear the superposition theorem applies [9]. This implies that when several current sources are connected to the R/2R ladder the output current can be calculated as the sum of contributions from the different current sources.
3.3.2 Mathematical Proof of Binary Behaviour

In this section it is shown mathematically that the R/2R ladder structure in Figure 3.8 is binary weighted. As mentioned earlier, the MSB of the input controls the rightmost current source (current source $N$). This current source draws current $I$ from the output when the MSB is enabled. In order for the structure to be binary weighted, the current drawn from the output by current source $N - x$ should be equal to $I/2^x$ when its control bit is enabled.

3.3.2.1 Application of Derived Formulas

The formulas derived in section 3.3.1 are valid for a more general case. These formulas can be simplified when applied on the structure in Figure 3.8. Consider the case when a single current source, $b_M (M < N)$, is connected to the R/2R ladder as is shown in Figure 3.13.

![Figure 3.13: The R/2R ladder with a single current source connected.](image)

$M$ could of course be equal to 1 in this case, corresponding to the leftmost current source being active. Using the notations from Figure 3.10 we get

\[ R_{T,1} = R \]  
(3.23)

\[ R_{T,n} = 2R, \quad n \in [2, N - 1] \]  
(3.24)

\[ R_{S,n} = R, \quad n \in [1, N - 1] \]  
(3.25)

(3.23)-(3.25) inserted into (3.13) and (3.14) yields

\[ R_{R,left,1} = R \]  
(3.26)

and

\[ R_{R,left,n} = (R_{R,left,n-1} + R)/2R, \quad n \in [2, M] \]  
(3.27)

Since $R_{R,left,1} = R$ we will get
\[ R_{R,\text{left},2} = (R + R) \| 2R = R \]  
(3.28)

and in the same way

\[ R_{R,\text{left},n} = R, \quad n \in [1, M] \]  
(3.29)

Formula (3.29) implies that the circuit in Figure 3.13 is equivalent to the circuit shown in Figure 3.14 below.

![Figure 3.14: Simplification of the R/2R ladder with a single current source connected.](image)

Inserting (3.24)-(3.25) into (3.15) and (3.17) yields

\[ R_{R,\text{right},N} = R \]  
(3.30)

and

\[ R_{R,\text{right},n} = (R_{R,\text{right},n+1} \| 2R) + R, \quad n \in [M + 1, N - 1] \]  
(3.31)

With (3.24) and (3.29) inserted in (3.22) we get

\[
I_{\text{out}} = \begin{cases} 
\left( \prod_{n=M+2}^{N} \frac{2R}{2R + R_{R,\text{right},n}} \right) \cdot \frac{R}{R + R_{R,\text{right},M+1}} \cdot I, & M \in [1, N - 2] \\
\frac{R}{R + R_{R,\text{right},M+1}} \cdot I, & M = N - 1
\end{cases}
\]  
(3.32)

3.3.2.2 Proof of Binary Behaviour Using Mathematical Induction

Due to the recursive properties of the derived formulas, mathematical induction is used to prove that the structure is binary weighted. In order to prove this it must be proven that current source \( N - x \) adds a \( 1/2^x \) current contribution to \( I_{\text{out}} \) when connected to the R/2R ladder.
The Basis
The case $x = 0$ is trivial since current source $N$ is connected directly to the input of the OP-amp. It will draw current $I$ from the output when connected to the R/2R ladder. For current source $N - 1$, inserting $M = N - 1$ into formula (3.32) and combining this with (3.30) yields

$$I_{\text{out}} = \frac{R}{R + R} \cdot I = \frac{I}{2}$$

(3.33)

So far we have proven that the structure is binary weighted for bits (current sources) $N$ and $N - 1$ corresponding to $x = 0$ and $x = 1$. These cases are however not sufficient as basis in the proof. This is because they are not included in the part of formula (3.32) covering cases where $M \in [1, N - 2]$. In order to have a valid basis we need to prove that when current source $N - 2$ is connected to the R/2R ladder, it draws current $I/4$ from the output. With $M = N - 2$ in (3.32) we get

$$I_{\text{out}} = \left( \prod_{n=N-2}^{N} \frac{2R}{2R + R_{R,\text{right},n}} \right) \cdot \frac{R}{R + R_{R,\text{right},N-1}} \cdot I = \frac{2R}{R + R_{R,\text{right},N-1}} \cdot I = \frac{R}{2} \cdot I = \frac{I}{4}, \quad Q.E.D.$$

Inductive Step
Assume that current source $N - k, k \geq 2$, draws current $I/2^k$ from the output when connected to the R/2R ladder. Call this current $I_{\text{out},N-k}$ i.e. $I_{\text{out},N-k} = I/2^k$. Inserting $M = N - k$ into (3.32) gives us

$$I_{\text{out},N-k} = \left( \prod_{n=N-k+2}^{N} \frac{2R}{2R + R_{R,\text{right},n}} \right) \cdot \frac{R}{R + R_{R,\text{right},N-k+1}} \cdot I = \frac{2R}{R + R_{R,\text{right},N-k+1}} \cdot I = \frac{I}{2^k}$$

(3.34)

Current $I_{\text{out},N-(k+1)}$ is the current drawn from the output by current source $N - (k + 1)$ when it is connected to the R/2R ladder. Using (3.32) it can be written as

$$I_{\text{out},N-(k+1)} = \left( \prod_{n=N-k+1}^{N} \frac{2R}{2R + R_{R,\text{right},n}} \right) \cdot \frac{R}{R + R_{R,\text{right},N-k}} \cdot I = \frac{2R}{R + R_{R,\text{right},N-k+1}} \cdot \left( \prod_{n=N-k+2}^{N} \frac{2R}{2R + R_{R,\text{right},n}} \right) \cdot \frac{R}{R + R_{R,\text{right},N-k}} \cdot I$$

(3.35)

Denote
\[ \Phi = \left( \prod_{n=N-k+2}^{N} \frac{2R}{2R + R_{R,right,n}} \right) \]  

(3.36)

Inserting this into (3.35) gives us

\[ I_{out,N-(k+1)} = \frac{2R}{2R + R_{R,right,N-k+1}} \cdot \frac{R}{R + R_{R,right,N-k}} \cdot \Phi \cdot I = \{(3.31)\} = \]

\[ = \frac{2R}{2R + R_{R,right,N-k+1}} \cdot \frac{R}{R + (R_{R,right,N-k+1})2R + R} \cdot \Phi \cdot I = \]

\[ = \frac{2R}{2R + R_{R,right,N-k+1}} \cdot \frac{R}{2R + \frac{2R \cdot R_{R,right,N-k+1}}{2R + R_{R,right,N-k+1}}} \cdot \Phi \cdot I = \]

\[ = \frac{2R}{2R + R_{R,right,N-k+1}} \cdot \frac{R \cdot (2R + R_{R,right,N-k+1})}{2R \cdot (2R + R_{R,right,N-k+1}) + 2R \cdot R_{R,right,N-k+1}} \cdot \Phi \cdot I = \]

\[ = \frac{R}{2R + 2R_{R,right,N-k+1}} \cdot \Phi \cdot I \]  

(3.37)

Inserting (3.36) into (3.34) gives us

\[ \Phi \cdot \frac{R}{R + R_{R,right,N-k+1}} \cdot I = \frac{I}{2^k} \Leftrightarrow \]

\[ \Leftrightarrow \Phi \cdot R \cdot 2^k = R + R_{R,right,N-k+1} \Leftrightarrow \]

\[ \Leftrightarrow R_{R,right,N-k+1} = 2^k \cdot \Phi \cdot R - R \]  

(3.38)

By combining this with (3.37) we get

\[ I_{out,N-(k+1)} = \frac{R}{2R + 2(2^k \cdot \Phi \cdot R - R)} \cdot \Phi \cdot I = \]

\[ = \frac{R}{2 \cdot 2^k \cdot \Phi \cdot R} \cdot \Phi \cdot I = \frac{1}{2} \cdot \frac{2^k \cdot I}{2^k \cdot I} = \frac{I}{2^{k+1}} \]  

(3.39)

And thus we have proven that

\[ I_{out,N-k} = \frac{I}{2^k} \Rightarrow I_{out,N-(k+1)} = \frac{I}{2^{k+1}}, \quad k \geq 2 \]  

(3.40)

We know from the basis that

\[ I_{out,N-k} = \frac{I}{2^k}, \quad k = 2 \]  

(3.41)
and hence the relationship

\[ I_{out,N-x} = \frac{I}{2^x}, \quad x \geq 2 \]  \hspace{1cm} (3.42)

has been proven valid through mathematical induction. The cases \( x = 0 \) and \( x = 1 \) are covered in the basis and the proof that the structure is binary weighted is therefore complete. \( Q.E.D. \)

### 3.3.3 Examination of Errors in the R/2R Ladder using Examples

A non-ideal DAC suffers from deviations when it comes to component values. For example the resistance values in the ladder will differ from the ideal values \( R \) and \( 2R \). Also the current drawn by the current sources will be different for different current sources; ideally all these current should be equal. The different kinds of errors affect the output in different ways. To be able to compensate for these faults it is of interest to know how these errors manifests. Therefore errors will be introduced in the previously examined model and the impact of these will be studied.

The contribution to the output current from current source \( N - x \) in the ideal case could be written as

\[ I_{cont,N-x} = \frac{I_{ref}}{2^x} \]  \hspace{1cm} (3.43)

as proved in section 3.3.2. However when it comes to a non-ideal ladder one could assume that component errors in both resistors and current sources affects this current and that it could be written as

\[ I_{cont,N-x} = (I_{ref} + \Delta I_{ref}) \cdot \frac{A + C \cdot \delta}{B + D \cdot \delta} \]  \hspace{1cm} (3.44)

where \( A/B = 1/2^x \) and \( \Delta, \delta, A, B, C, D \) are real numbers. Note that if the errors denoted \( \Delta \) and \( \delta \) are zero (3.43) and (3.44) are equal.

#### 3.3.3.1 Errors in Current Sources

We start our study by adding an error to a current source. This is modeled in [6] as a unit current source and a current source in parallel generating the error. An illustration of this is found in Figure 3.15. The unit current source is denoted \( I \) and the error current source is denoted \( \Delta I \).
This model is applicable to all of the current sources connected to the ladder.

In the following example an error is introduced in one of the current sources of a 4-bit DAC. The 4-bit DAC with error is shown in Figure 3.16. The input data word in this case would be 0010\textsubscript{BIN}.

The schematic in Figure 3.16 is simplified by removing the disconnected current sources and replacing some of the resistors with equivalent resistances. The op-amp connection is replaced by a connection to ground since its input acts as virtual ground. The simplified schematic is shown in Figure 3.17
Further simplifications are made by calculating the equivalent resistances of the part of the circuit to the left and right of the current source connection. The left equivalent resistance is given by

\[
\left( \frac{1}{2R} + \frac{1}{2R} \right)^{-1} = R
\]  \hspace{1cm} (3.45)

and the right one by

\[
R + \left( \frac{1}{R} + \frac{1}{2R} \right)^{-1} = R + \left( \frac{3R}{2R^2} \right)^{-1} = \frac{5R}{3}
\]  \hspace{1cm} (3.46)

The further simplified circuit is shown in Figure 3.18.
The current \( I_1 \) which is marked in Figure 3.18 is needed in order to calculate the output current. KCL and current sharing is applied and this results in

\[
I_1 = \frac{R}{R + \frac{5R}{3}} \cdot (I + \Delta I) = \frac{3R}{3R + 5R} \cdot (I + \Delta I) = \frac{3}{8} \cdot (I + \Delta I) \tag{3.47}
\]

The resistor to the right is removed and replaced with the original resistor structure as shown in Figure 3.19. This is done in order to calculate \( I_{out} \).

![Figure 3.19: Schematic with unfolded resistor structure to the right.](image)

Using current sharing we can calculate \( I_{out} \) as

\[
I_{out} = \frac{2R}{R + 2R} \cdot I_1 = \frac{2}{3} \cdot I_1 = \frac{2}{3} \cdot \frac{3}{8} \cdot (I + \Delta I) = \frac{1}{4} \cdot (I + \Delta I) \tag{3.48}
\]

If the DAC was ideal the output would be equal to \( I/4 \). This can be realized by inserting \( x = 2 \) into (3.43). In the expression for the non-ideal output we have the term \( I + \Delta I \) instead of \( I \). By studying (3.22) one can realize that such a substitution is valid for a general case. As has been shown, adding an error to a current source which is connected to the R/2R ladder adds a proportional error to the output current. In other words an error in the current source will result in an offset error on the output.

### 3.3.3.2 Errors in Resistors

Another very important type of error to examine is those caused by to non-ideal resistance values. Consider the same circuit setup as in the previous example but with an error in a resistor instead. This is shown in Figure 3.20.
Figure 3.20: R/2R DAC with resistor error in the ladder.

The schematic is simplified by calculating equivalent resistances to the left and right of the current source connected to the ladder. As before the op-amp is replaced by a virtual ground and the inactive current sources are removed. The left equivalent resistance, denoted $R_{eq,L}$, is given by

$$R_{eq,L} = \left( \frac{1}{2R + \delta R} + 1 \right)^{-1} = \frac{4 + 2\delta}{4 + \delta} \cdot R \quad (3.49)$$

and the right one by

$$R + \left( \frac{1}{R} + \frac{1}{2R} \right)^{-1} = R + \left( \frac{3R}{2R^2} \right)^{-1} = \frac{5R}{3} \quad (3.50)$$

The simplified circuit is shown in Figure 3.21.
Current sharing gives

\[
I_1 = \frac{R_{eqL}}{R_{eqL} + \frac{5R}{3}} \cdot I = \frac{3R_{eqL}}{3R_{eqL} + 5R} \cdot I = \frac{3 \cdot (4 + 2\delta)}{4 + \delta} \cdot \frac{R}{3R \cdot (4 + 2\delta) + 5R \cdot (4 + \delta)} \cdot I = \frac{12R + 6\delta R + 20R + 5\delta R}{12R + 6\delta R + 20R + 5\delta R} \cdot I = \frac{32R + 11\delta R}{32R + 11\delta R} \cdot I
\]

(3.51)

By unfolding the right equivalent resistance into its original resistor structure \(I_{out}\) can now be calculated. The unfolded schematic is shown in Figure 3.22.

![Figure 3.22: Schematic with the right equivalent resistance unfolded.](image)

We get

\[
I_{out} = \frac{2R}{2R + R} \cdot I_1 = \frac{2}{3} \cdot I_1
\]

(3.52)

Combining (3.51) and (3.52) gives

\[
I_{out} = \frac{2}{3} \cdot \frac{12R + 6\delta R}{32R + 11\delta R} \cdot I = \frac{8 + 4\delta}{32 + 11\delta} \cdot I
\]

(3.53)

Note that if the error \(\delta\) is equal to zero, then \(I_{out}\) is equal to \(I/4\) as it should be. An important thing to notice is that when there is an error in a resistor in the ladder, it will not lead to an offset at the output.
Consider the example shown in Figure 3.23. The input data word is set to 0100\textsubscript{BIN} in this case, which should yield $I_{\text{out}} = I/2$ and the same error is present once again.

The same calculations as in the previous example are applied which gives

$$I_{\text{out}} = I \cdot \frac{16 + 6\delta}{32 + 11\delta}$$  \hspace{1cm} (3.54)

As shown in (6.43) this is almost the expected value of $I_{\text{out}}$ except for the $\delta$ terms. The two similar examples show that the output current suffers from the same kind of errors in both cases. However though they suffer from the exact same error in the ladder the error in the output current differs. When (3.53) is compared to (3.54) it is easy to see that different nodes are affected differently from the same error. Since every resistor is non-ideal the output will almost certainly contain some distortions introduced from the R/2R-ladder.
4 Measurements

4.1 Introduction
As explained in the chapter about R/2R DACs errors in the ladder affects the output in an unwanted way. In order to break down the output and identify the actual non-wanted components in the output values, measurements had to be done. For this a measurement setup had to be constructed - a platform on which different measurements could be executed. The platform had to be able to subject the device under test (DUT) to a known test signal or pattern, and then capture the response and process the data and display the result in a human friendly manner. Included in this chapter are descriptions of the different parts of the test platform and a brief look at some sampling theory.

4.2 The Discrete Fourier Transform
When using computers for calculating a Fourier transform the discrete Fourier transform (DFT) has to be used. When using Matlab for calculating Fourier transforms the used implementation is called fast Fourier transform (FFT) which is an implementation of the discrete Fourier transform (DFT). This section will give a brief background of some properties of the DFT.

When the calculations are done using a computer a finite number of samples must be used. The number of samples used when computing the transform are often denoted N and the samples are called points. In this chapter these will be referred to as points while sample will be used to refer to a part of a longer signal in the time domain that is subjected to transformation. The same notation will be used here as in [2]; that is capital letter denotes the transformed signal and lower case letters denotes signals in the time domain. The definition of the Fourier transform of a time discrete signal is found in [2]. It is written as

\[
X[\Omega] = \sum_{n=-\infty}^{\infty} x[n] e^{-j\Omega n} \tag{4.1}
\]

where \( \Omega \) represents the normalized angular frequency, and \( x[n] \) denotes a continuous signal which has been sampled. The discrete Fourier transform can be viewed as a sampled version of the Fourier transform of a time discrete signal. That is if the Fourier transform of a sampled signal would to be sampled one would obtain the DFT. A DFT could be created from the transform in (4.1) by taking N points from \( X[\Omega] \) with steps of \( 2\pi/N \) between each value of \( \Omega \) when \( \Omega \) lies in the interval \( [0,2\pi] \). In other words pick N points evenly distributed from one period of the transformed discrete signal and adjust the sum so that it goes from zero to \( N - 1 \). This will give The DFT. As stated in [2] this can be written as

\[
X_N[k] = X[\Omega]|_{\Omega = k\frac{2\pi}{N}}, \quad k = 0, 1, \ldots, N - 1 \tag{4.2}
\]

This gives the following expression for the DFT in relation to the sampled signal
$$X_N[k] = \sum_{n=0}^{N-1} x[n] e^{-j(k/N)n} \quad (4.3)$$

Also stated in [2] is the fact that if values of the DFT outside the interval \([0, N - 1]\) are calculated a periodic repetition of the values within the interval will appear. This can be shown using (4.3) as

$$X_N[k + N] = \sum_{n=0}^{N-1} x[n] e^{-j(k+\mathcal{N})\frac{2\pi n}{N}}$$

$$= \sum_{n=0}^{N-1} x[n] e^{-j(k)\frac{2\pi n}{N}} e^{-j\mathcal{N}\frac{2\pi n}{N}}$$

$$= \sum_{n=0}^{N-1} x[n] e^{-j(k)\frac{2\pi n}{N}} e^{-j2\pi n}$$

$$= \sum_{n=0}^{N-1} x[n] e^{-j(k)\frac{2\pi n}{N}} = X_N[k] \quad (4.4)$$

This shows that the spectrum of the DFT will repeat itself over a period of \(N\).

As mentioned previously when calculations are performed on a computer only a finite number of points can be used. Thus it may be necessary to limit the number of points that will be included. This can be done in different ways however it is important to keep in mind that if the period of the incoming signal is long compared to the number of points used information will be lost. This effect is called truncating. An effect of truncating is so called spectral leakage. If the DFT is not chosen wisely, a situation may arise where the leakage is sampled and therefore frequencies not present in the sampled signal are introduced.

Let us study a small example to illustrate these effects. Truncation is done with a window function. The window is in the form of a rectangular function with the description

$$w[n] = \begin{cases} 1, & \text{on the interval} \\ 0, & \text{outside the interval} \end{cases} \quad (4.5)$$

The test signal will consist of a time discrete sine waveform onto which the window will be applied, the sinusoid is denoted \(x[n]\) and written as

$$x[n] = \sin \left( \frac{\pi}{4} n \right) \quad (4.6)$$

The multiplication between the window function and the sin will result in a truncated signal in the time domain. However as known from basic transform theory a multiplication in the time domain corresponds to a convolution in the frequency domain. As shown in section 2.3.2 a rectangular shape in the time domain will give a sinc function in the frequency domain. And since a pure sinusoid consists of only one frequency it will correspond to a dirac in the
frequency domain. The spectrum of a sinusoid is shown in Figure 4.1. The effect of the
convolution between a dirac and a sinc is essentially that, the sinc function is copied, scaled
and moved to where the dirac impulses originally appears. As can be seen in the Figure 4.2
the spectrum is filled with superpositioned sinc functions spreading energy into the spectrum,
as stated before this effect is called spectral leakage.

Figure 4.1 Frequency spectrum of a sine wave, the spectrum is periodic with $2\pi$.

Figure 4.2: The frequency spectrum of the windowed signal.
The form of the superpositioned spectrum still has the characteristics of the sinc function, which includes ripple outside of the main lobes. Between the peaks of the ripple are zero value points evenly distributed in the spectrum. As shown before the DFT can be viewed as a sampled time discrete Fourier transform, therefore the length of the window and the number of points in the DFT will make all the difference. As can be seen in Figure 4.2, sampling with the right number of points will adjust the space between samples so that it fits perfectly to the original transform. That is, it will yield one sample from the peak of the main frequency and it will hit the zero values in between the ripple waves, thus completely bypassing the effects of spectral leakage and the result will be much like the analytical expression.

If a mismatch between the waveform and the length of the DFT occurs, unwanted effects will occur. That is, the distance in between sampled points will be such that the main lobe is not sampled at the peak and the zero value points are not sampled. Thus this gives a DFT indicating the presence of several non-existing frequencies, with respect to the analytical transform. In the windowed TDFT these frequencies do of course exist due to leakage, but they are unwanted. This problem is discussed in [2]. To combat this problem the number of points in the DFT must coincide with a number of whole periods of the examined signal. The mathematical relation between the number of points and the incoming signal, when the true spectrum is wanted is given by [2] as

$$N = m \cdot \frac{2\pi}{\Omega}$$

(4.7)

where $N$ is the number of points, $m$ is the number of periods and $\Omega$ is the normalized angular frequency of the periodic signal. It is also stated in [2] that when dealing with non-periodic signals there is no way of eliminating the spectral leakage frequencies, but different forms of window functions can suppress these more or less.

### 4.3 Sampling Theory

The DAC takes a digital input and the output is analogue, in order to perform the analysis it is helpful to bring the output-data in to a computer. This in turns means that an analogue-to-digital converter (ADC) is required to convert the analogue output to a series of digital values. To perform frequency analysis Fourier transforms were required. Two issues connected to the transformation where considered in the design of the test platform, these where sampling coherency and the length of the DFT. Both of these issues will be discussed in this section.

As known from Fourier analysis [11] any signal that is periodic in an interval could be given a Fourier transform. The part which is to be transformed is extended from $-\infty$ to $+\infty$ this can lead to some complications when the selected part is to be transformed. To illustrate the problem a sine wave is generated and then transformed. The sine wave is periodic so selecting exactly one period would be enough for the Fourier transform.

To illustrate the effects of discontinuities in the sample that is transformed two different samples are generated and transformed. The two samples are shown in Figure 4.3.
The first graph shows a whole sinus period, during the transformation it will be treated as if replicated and repeated an infinite number of times. The second graph shows an extension of this period. In the third graph however a slightly longer part of the sinusoid has been sampled. It will also be treated as if replicated an infinite number of times during transformation. Because of the nonsymmetrical appearance of the wave, once it is repeated discontinuities will appear between the repeated periods. Graph number four shows a set of repeated periods and the discontinuities that are formed in between. This is a direct result of faulty selection of period time. When using a computer to perform the transform a discrete transform will be used. In this thesis Matlabs implementation of FFT has been used. What is received is from the FFT a set of frequency components that describes the signal which is transformed. The discontinuities caused by a faulty chosen sample will introduce large collection of frequency components which are not present in the FFT of the original signal. Since the incoming signal is a sinusoid it should only consist of one frequency component. These faulty frequency components will result in faulty signal power being spread across the spectrum, making it harder to gather information about how the sinusoid is affected by the circuit. For example nonlinearities should introduce undesired energy peaks in the frequency spectrum. If the peaks are small there is a risk that the noise caused by bad sampling will drown the noise caused by nonlinearities.

The discrete Fourier transforms of the two sinusoid samples from Figure 4.3 are shown in Figure 4.4. The red line displays the spectrum of the faulty period and the blue displays the spectrum of the correct one.
In Figure 4.4 it is easy to see that the difference between a perfectly sampled signal and a slightly faulty sampled signal is large. It is also easy to understand that small nonlinearity spectral components would be drowned in the spectral components introduced by a faulty sample length.

Even if the sampling period is chosen so that perfect synchronization between the sampling and the sampled signal is achieved there is still a hazard to consider. This hazard arises when the start and end points of the sample are equal. Since the signal is treated as if it was periodically extended the same value would be repeated twice in a row. This forms a discontinuity which will spread energy into the surrounding spectrum, causing faulty frequency components. This problem is highlighted in the following example. A perfect period and one where the start and end points are equal are shown in Figure 4.5.
As in the previous example two signals and a representation of how they will be treated by the DFT are displayed in Figure 4.5. The first one has the wanted size, thus making the start point of the next period fit precisely to the end of the first one. This gives a continuous signal of the expected form. In the second case, the last point is equal to the first one, resulting in a discontinuity when repeated. Note that a horizontal line now connects two adjacent wave forms. Figure 4.6 shows the connection between two adjacent periods of the sample with faulty length.
When this signals are transformed into the frequency domain it is obvious that the discontinuities formed by this kind of sampling will introduce unwanted spectral components, much like the previous kind of error introduced by sampling. The DFT of this sample and the sample of perfect length is shown in Figure 4.7 below.
In many cases the signal to be measured and transformed is not known a priori, therefore making it hard to sample without receiving unwanted discontinuities. The countermeasure for this is called windowing, it is described in [2] and in [1]. Using a window is always a tradeoff, typical effects of windowing are as mentioned in [2] widening of the spectrum and ripple. Maloberti mentions in [1] the fact that if the sample points of the DFT cover a whole number of the periods there is no need for windowing. In the case studied in this thesis the signal generating, the sampling and the length of the DFT is controllable therefore no specific windowing function has been used. This is true for the fundamental tone, that is the generated signal. Nonlinearities within the circuit could result in frequencies in the examined output which does not follow the same relations described in (4.7). Because of this a higher resolution might be of use to ease the effect of leakage and thus making it easier to detect the nonlinearities. The number of points used could be increased without violating the (4.7) by including more periods of the signal with in the sample. If two periods are used, then the sampled signal has a periodicity of 2N and more points can be included in the DFT, thus increasing the resolution.

4.4 Measurement Platform

The main components of the measurement platform and how they are interconnected is shown in Figure 4.8. During measurements the digital signal is generated by a DSP inside the DAC system. It is then converted to an analogue signal by the DAC chip, which also lies inside the DAC system. The analogue output of the DAC system is converted to a digital signal in the ADC. The signal is then fed back to the DSP and stored in a memory in the DAC system. Once the measurement is complete the data is transferred to a PC using a JTAG-connection.

![Diagram of Measurement Platform](image)

Figure 4.8: Measurement platform overview.

4.4.1 The DAC System

The DAC system consists of the DAC from Naim Audio which is a commercial system for hi-fi audio digital-to-analogue conversion. To avoid misconceptions in this section the Naim DAC system will be referred to as the DAC system and the PCM1704 DAC chip from Texas
Instruments will be referred to as the DAC chip. The main components of the DAC system are the following: The DAC chip, power supplies, clock generators and the ADSP-21369 digital signal processor (DSP) from Analog Devices. An overview of the DAC system is shown in Figure 4.9.

![Figure 4.9: An overview of the DAC system.](image)

The central part in the signal flow during measurements is the DSP since it generates the test signal, receives measured data and manages the connection to the PC. The onboard power supply units are arranged so that it generates stable voltages for both analogue and digital usage, in a set of different amplitudes (−12 V,+12V etc.).

### 4.4.2 The Physical Setup

The ADS1274 ADC from Texas Instruments, from now on referred to as the ADC, is mounted upon the test board ADS1274EVM from Texas Instruments, from now on referred to as the EVM. The EVM consist of components peripheral to the ADC which are there to ease the use of the ADC. It consists of among other things DIP switches for configuring the ADC, input buffers and connection pins that are connected to the ADC. The ADC is capable of high resolution 24-bit analogue-to-digital conversion. For further information about the ADC please see the manufacturer’s data sheet.

By making some minor modifications to the DAC system the EVM board could be mounted on the DAC system motherboard. The required power inputs to the EVM were acquired from the DAC system power supplies. This gave a stable source of power as well as common
reference levels. In order to ensure synchronized measurements the ADC was clocked by the same clock as the DSP. The ADC was connected to the DSP via the use of reconfigured test pins. The DAC system output was connected to the input of the ADC.

As mentioned before a PC was used to collect the data from the DSP through a JTAG-connection. The tool used to connect the PC to the DSP was the *HP USB ICE* emulator from Analog Devices. During measurements this emulator was used to run a program on the DSP. The program generated a certain digital output signal, for example a sine wave with specific amplitude. The digital signal was converted to an analogue signal by the DAC chip and then back to an analogue signal by the ADC. The ADC fed the signal back to the DSP. Once the data was back in the DSP it could be stored in a data buffer. When a preset amount of data had been collected the program halted and the data could be collected by the PC.

### 4.4.3 Data Sampling

In section 4.3 it is explained why it is important that the sample points are placed so that the measured data adds up to a collection of whole periods of the measured signal. Both the DSP and the ADC were clocked by the DAC system clock generation circuit. Because of this it was possible to ensure synchronization between the A/D conversion and the test signal generation. This made it possible to guarantee that the desired number of periods was captured within the DSP data buffer. Even though the start of the sample series might not be at the beginning of a period, meaning zero for a sine, an even number of periods was obtained. This allowed for a DFT to be performed on the data without introducing spectral leakage.

Once the data had been transferred to the PC, Matlab was used for signal processing. In Figure 4.10 an example of processed data is shown. The distortions, which are plotted in blue in this figure, were extracted from a periodic mean value of the output from the DAC system when it was fed with a sinusoidal input of amplitude $-5.2 \text{ dBFS}$ (decibels relative to full scale). The signal was removed from a DFT of this periodic mean value and the distortions could then be acquired using an inverse discrete Fourier transform (IDFT). A scaled down version of the periodic mean value is also plotted in red in this figure to illustrate where certain distortions occur. The resulting signal after the signal processing described above should ideally consist of a constant line since the white noise is removed by the use of a periodic mean value. Notice the symmetry in the distortions around the positive and negative parts of the sinusoid.
4.5 Test Patterns and Data Extraction

As previously explained in chapter 3 deviations from the nominal component values in the ladder will lead to errors in the output signal. The purpose of the measurements was to analyse and identify how these errors affect the output signal. Even though the number of possible input words is finite, a 24-bit DAC still has a very large amount of possible input combinations. For smaller DACs it could be possible and perhaps easier to conduct exhaustive testing and extract the error from each word. In this case however a time efficient and scalable approach was desired. Given the nature of the DAC and how superpositioning could be used to construct the output signal as shown in chapter 3 it was suspected that determining the output values of the individual bits would be enough.

A brief look at loop formed between the DSP output, the DAC chip, the ADC and the DSP input shows a few obstacles making it hard to determine the bit values individually. An overview of this loop is shown in Figure 4.11.
A quick look at this sketch of the signal loop shows two possible points of measuring. These points were easily accessed on the DAC system motherboard. The first one lies as close to the DAC chip as possible i.e. directly after the I/V-conversion stage. External equipment was not meant to be connected at this point and therefore the signal was severely distorted when the input signal to the ADC was collected at this point. The next suitable point for measurement was the output from the DAC system. A DC blocker at the output prevented the use of static test signals so the only way to get past it was to use an alternating signal. Toggling between bipolar zero to the desired bit combination should have given data suitable for bit value extraction. The data collected in this way suffered from drift and other disturbance such as charge/discharge effects which made it unsuitable for bit value extraction.

Due to the nature of the binary number system every number containing a single non-zero bit is equal to the word containing all preceding bits plus one LSB. In other words $40_{HEX}$ is equal to $3F_{HEX} + 01_{HEX}$ in the same way $80_{HEX}$ is equal to $7F_{HEX} + 01_{HEX}$. Alternating between a word containing a single non-zero bit and the word containing all preceding bits generates a small alternating signal with a DC offset. This offset could be removed by the blocker and leave only the difference between the two levels. Ideally the difference should be equal to one LSB as previously mentioned however due to distortions this is not true and the difference might be somewhat larger (positive or negative).

Let $b_n$ denote the output value of bit $n$. The output of an input word could then be written

$$A = \sum_{k=1}^{N} d(k) \cdot b_k$$

(4.8)

where $d(k)$ is digit $k$ in the word. The difference between e.g. $80_{HEX}$ and $7F_{HEX}$ could then be written as

$$\Delta = b_8 - b_7 - b_6 - b_5 - b_4 - b_3 - b_2 - b_1$$

(4.9)
Measuring the output difference between each word containing a single non-zero bit and the word which is one LSB smaller in a DAC would lead to a set of equations. These equations form an equation system and for an 8-bit DAC it would look like

\[
\begin{bmatrix}
\Delta_8 \\
\Delta_7 \\
\Delta_6 \\
\Delta_5 \\
\Delta_4 \\
\Delta_3 \\
\Delta_2 \\
\Delta_1 \\
\end{bmatrix} = \begin{bmatrix}
1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
0 & 1 & -1 & -1 & -1 & -1 & -1 & -1 \\
0 & 0 & 1 & -1 & -1 & -1 & -1 & -1 \\
0 & 0 & 0 & 1 & -1 & -1 & -1 & -1 \\
0 & 0 & 0 & 0 & 1 & -1 & -1 & -1 \\
0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix} \begin{bmatrix}
b_8 \\
b_7 \\
b_6 \\
b_5 \\
b_4 \\
b_3 \\
b_2 \\
b_1 \\
\end{bmatrix},
\]

(4.10)

where \( \Delta_x \) denotes the output difference between two adjacent levels. The equation system is solvable since the determinant of the equation matrix is non-zero [10].

A problem that arose when toggling between two adjacent code words was that the difference between the two output levels was too small to be detected by the ADC. The solution was to introduce amplification in the EVM buffer stage giving the setup shown in Figure 4.12.

![Figure 4.12: Amplification in the buffer stages.](image)

It was discovered that by increasing the gain of the EVM buffer also altered the filtering properties of the buffer even though the RC time constant was unaltered. The EVM functioned as a HP filter and the performed changes introduced a higher cut-off frequency. Since the ADC sampling frequency was 48 kHz a low signal frequency was desired in order to preserve detail in the measurements. Tests showed that 2 kHz was a high enough frequency to pass the EVM high pass filter. It is possible that another frequency would yield even better test results. The derivation of an optimal frequency however falls outside the scope of this thesis.
Figure 4.13: A measured output difference between two adjacent input words.

Figure 4.12 shows a periodic mean value of a measured output difference between two adjacent input words. By creating a mean value of the higher and lower level the output difference between the different code words could be calculated. Due to the fact each bit is faulty there is a chance that the output of the smaller input value is larger than the output of the larger input value. The calculation above only gives the absolute size of the fault. It is important to determine direction of the fault, in other words the sign of $\Delta_\zeta$ in (4.10) has to be decided. In order to do this the same alternating signal was measured with one difference, this time the higher input level was intentionally applied to the DAC a longer part of the period than the lower level. A periodic mean value of such a measurement is shown in Figure 4.14.
Figure 4.14: A measured output difference sign extraction.

As can be seen in this figure the higher level is much longer than the lower. Since it is known that the longer level corresponds to the higher input value the conclusion can be drawn that the output of the higher input value is larger than that of the lower input value.

Once the size and direction of each fault was known the equation system in (4.10) could be solved using simple linear algebra. The bit levels obtained through these measurements were affected by an unknown gain introduced by each step from the DAC chip output to the ADC input. To counter for this the obtained values were normalized with respect to the nominal value of the MSB.
5 The DAC Chip Model

5.1 Introduction
In order to be able to compensate the static nonlinearity distortions of the DAC chip correctly knowledge about its conversion structure was needed. For this purpose a model of the DAC chip was developed. This model has been used in simulations for development and evaluation of the nonlinearity compensation. Included in this chapter is the deduction of the DAC model and an analysis of the DAC structure.

5.2 Internal DAC Chip Structure
By studying the PCM1704 datasheet [8] and a field applications engineer training document from Burr Brown [7] knowledge about the internal DAC structure was obtained.

5.2.1 R/2R Behaviour
In the FAE training document [7] one can see that the DAC conversion is of R/2R type. This was verified through measurements by noting that the distortions of the DAC output signal had typical R/2R characteristics. An example of this is shown in Figure 5.1. The distortions, which are plotted in blue in this figure, were extracted from a periodic mean value of the output from the DAC system when it was fed with a sinusoidal input of amplitude \(-5.2\,\text{dBFS}\). The signal was removed from a DFT of this periodic mean value and the distortions could then be acquired using an IDFT. A scaled down version of the periodic mean value is also plotted in red in this figure.

![Figure 5.1: Plot of R/2R distortions.](image-url)
The R/2R Ladder Structure
The R/2R ladder structure that has been used for simulating the R/2R DACs is shown in Figure 5.2. This architecture has the benefit that the switch sizes can be uniform since an equal amount of current is flowing through each switch. It is therefore suitable for usage in an integrated circuit design. This structure is elaborately analysed in section 3.3.

![Figure 5.2: The Used R/2R Ladder Structure.](image)

5.2.2 Dual DAC Structure
According to the datasheet of the PCM1704 [8] it consists of two separate 23-bit DACs, both connected to the output. The output of the PCM1704 can swing between $-1.2\ mA$ and $+1.2\ mA$. Measurements have showed that distortions of the negative and positive parts of a signal are completely different as can be in Figure 5.1. Two different architectures that could have this behaviour were considered as plausible.

5.2.2.1 The Complementary Architecture
The first architecture that was considered can be seen in Figure 5.3. In this architecture only one of the DACs is active at a time and the other is set to output zero. DAC A gives a positive contribution to the output when activated, and DAC B a negative contribution. In this way the output can swing between $-I_{\text{max}}$ and $I_{\text{max}}$ if the maximal current output of each DAC is $I_{\text{max}}$.

![Figure 5.3: The complementary DAC architecture.](image)

5.2.2.2 The Offset Current Architecture
The second architecture that was under consideration is shown in Figure 5.4. The bias current is equal to $I_{\text{max}}$, the maximal current output of each DAC. The lowest output voltage is given
when both DACs are set to output zero. The output current, $I_{out}$, will then be equal to $-I_{max}$ due to the bias current. For $-I_{max} \leq I_{out} \leq 0$ only DAC B is active and its output varies between 0 and $I_{max}$. $I_{out} = 0$ is reached when DAC B is set to output $I_{max}$. For $0 \leq I_{out} \leq I_{max}$ DAC B outputs $I_{max}$ and DAC A is active and outputs a value between 0 and $I_{max}$. The output current of this architecture can swing between $-I_{max}$ and $I_{max}$.

5.2.2.3 **Usage of DAC Architectures in Simulations**

The complementary architecture has been used in simulations for developing and simulating ways of nonlinearity compensation throughout the thesis. The effects on the offset current architecture have always been considered although they were not simulated.
6 Nonlinearity Compensation

6.1 Introduction
There are many ways of compensating for nonlinearities. The focus of this thesis has been to investigate a possible software method for reducing static nonlinearities of R/2R DACs. The method presented has been created with a DSP or FPGA implementation in mind. An algorithm was developed, implemented and tested. This chapter describes the algorithm, its properties and possible ways of improving it. Any references to “the algorithm” in this chapter are references to this algorithm.

6.2 Algorithm Description
The algorithm that has been developed is a kind of successive approximation (SA) algorithm. To be used it requires knowledge about how the output of each input bit of the DAC deviates from its ideal value. How this information is gathered is described in section 4.5.

6.2.1 Description Using Text and Graphics
In short, the algorithm takes an input word that was supposed to be sent to the DAC. With knowledge about the faults in each bit level of the DAC it constructs another word that is the closer to the ideal output value of the original word. One could say that the algorithm adjusts the input to the DAC so that the resulting output of the DAC is closer to the wanted output. The algorithm does this by successively adding bits to the output word while making sure that the corresponding output never exceeds the ideal output level. When doing this it goes from largest to smallest bit.

Let us consider a 4-bit DAC with non-ideal output values for each bit. Its ideal and non-ideal output values are shown in Figure 6.1. Ideal output values are coloured in white and actual output values are coloured in grey. The size of each value is represented by the horizontal length of each bar.

![Figure 6.1: Ideal and non-ideal output values of a 4-bit DAC.](image)

Assume that we want to output the analogue value corresponding to the digital input word $1110_{BIN}$. The ideal and actual output value of this input word are compared in Figure 6.2.
Figure 6.2: Ideal and actual output of the 4-bit DAC for input word 1110BIN.

The ideal output value is referred to as the wanted output value as well. Now let us compensate this input word step for step in order to demonstrate how the compensation algorithm works.

**Step 1**
The wanted output value is compared to the actual output of the bit 3 (MSB). Since it is larger than the output of bit 3, bit 3 is added to the output word. A rest is formed by subtracting the output value of bit 3 from the wanted output value. An illustration of this is shown in Figure 6.3.

**Step 2**
In the second step the rest value is compared to the output of bit 2. Since the rest value is largest of the two, bit 2 is added to the output word. A new rest is formed by subtracting the output value of bit 2 from the rest value. All of this is illustrated in Figure 6.4.

**Step 3**
In the third step of the compensation, the rest value is compared to bit 1. Since the rest value is smaller than bit 1, the output word and the rest value are unchanged. This is shown in Figure 6.5.

**Step 4**
In the fourth and final step in the compensation of this 4-bit word the rest value is compared to the output of bit 0. As can be seen in Figure 6.6 the rest value is larger than the output of bit 0 and bit 0 is therefore added to the output word. The rest value formed in this step is equal to
the difference between the ideal output value and the compensated output value assuming that the output of each bit is known exactly.

$\text{Incoming rest value}$  
$\text{Bit 0 output}$  
$\text{Outgoing rest value}$

Figure 6.6: Fourth and final step of the compensation.

The compensated output word is $1101_{\text{BIN}}$.

6.2.2 Pseudo Code Description

Shown below is C-style pseudo code demonstrating the algorithms behaviour. $n$ is the number of bits in the incoming data word.

```
rest value = wanted output value;
for (i = 0; i < n; i++){
    if (rest value >= output value of bit i){
        rest value = rest value-output value of bit i;
        add bit i to output value;
    }
}
```

6.3 Algorithm Behaviour and Properties

6.3.1 Underestimating Property

During the execution of the algorithm a certain bit only is added to the output word if it the rest value is larger than this bit. Because of this the output value is always smaller than the ideal output value. Let us continue on with the 4-bit DAC compensation example from section 6.2.1. A comparison of the compensated and non-compensated outputs as well as their relation to the ideal output is shown is in Figure 6.7.

```
Ideal output
Non-compensated output
Compensated output
```

Figure 6.7: Comparison between a compensated and non-compensated output.

As can be seen in the figure the compensated output is closer to the ideal output value. This is not always the case due to the fact that the algorithm is always underestimating the ideal output value; adding an extra LSB might yield a result closer to the ideal output. However the compensated output is usually closer to the ideal output than the non-compensated output and the algorithm is able to reduce static distortions significantly. This will be shown in succeeding sections that are including simulations and evaluations.
6.3.2 Reduction of Static Distortions
Consider the input output curve showed in Figure 6.8.

This curve exhibits the static distortions that are typical for R/2R DACs, although they are somewhat exaggerated. Notice how patterns are repeated throughout the curve due to the binary coding. For example the errors occurring in the first half of the curve are identical to those in the second half. The difference between these parts is that the MSB is zero in the first half, and one in the second half. Let us compare two segments of the non-compensated and compensated output of this DAC. These are shown in Figure 6.9 and Figure 6.10.

The first error that occurs is a rising error at around code word 30. This is combatted by the algorithm by just raising the output amplitude. The rising error at around code word 60 is not
as easily removed. At this point the algorithm has to choose between two levels with no values in between. Since the algorithm always underestimates the ideal output value the compensated output gets stuck at the lower level for some samples. The compensated output is increased again once the input has increased more than the size of the increasing error.

### 6.3.3 Information Loss

Since some code areas are mapped to the same output code information loss can occur when using the algorithm. Simulations based on measured values have shown that the size of these code areas are around 3-7 codes wide in a 24-bit DAC. These made up 7% of the dynamic range in this simulation. As mentioned before these faults occur when switching a lot of bits, most noticeably when performing transitions of the type ...0111... to ...1000... (adjacent codewords). The same fault that occurs at the transition between e.g. 03F<sub>HEX</sub> and 040<sub>HEX</sub> will occur when switching from 53F<sub>HEX</sub> to 540<sub>HEX</sub>. When the input signal is a sinusoid and one of its peaks is inside one of these areas clipping can occur as can be seen in Figure 6.11.

![Figure 6.11: Clipping of a sinusoid.](image)

Since the peak is symmetric, the effect is doubled; the same input codes are used twice during the rise and fall of the sinusoid.

### 6.3.4 Compression of Dynamic Range

Some compression of the dynamic range is usually present which is expected when one think about how the algorithm functions. The compression influences the larger input codes in the
way that these are saturated i.e. all of them are mapped to a certain smaller value. A simulation based on measured values of a 24-bit DAC showed that for the DAC system used for this thesis the total amount of saturated codes made up 0.001% of the total dynamic range.

6.4 Possible Improvements

6.4.1 Preservation of Detail
The information loss described in section 6.3.3 could be combatted by not modifying bits in a lower part of the input word. For example the lowest 4 bits could go untouched from the input word to the compensated output word. This could lead to detail preservation in the compensated output signal at the cost of an introducing the nonlinearity distortions of the 4 lowest bits. However the distortions in the lower bits could probably be disregarded when compared to the size of the faults shown in Figure 6.10 and Figure 6.11. These bits could still be included in the calculations of the compensation depending on what would yield the best results.

6.4.2 Always Check the Next Codeword
By always comparing the compensated output word with the word that is one LSB larger and choosing the word creating the best output, distortions could be reduced even more. This is so because of the underestimating property of the algorithm. Simulations have shown that introducing such a check could improve the THD+N.

6.4.3 Normalizing Bit Levels Differently
Once the bit levels had been extracted, as described in section 4.5, they were normalized with respect to the nominal value of the MSB. Ideally the output of the MSB should be equal to the sum of the output of all other bits plus one LSB, i.e.

$$b_N = b_1 + \sum_{k=1}^{N-1} b_k$$  \hspace{1cm} (6.1)

where $b_k$ is the output of bit $k$, $b_N$ is the output of the MSB. However due to distortions this is not true in non-ideal DACs. If the right side of (6.1) is much larger than the left side or vice versa a different normalisation might yield a better result.
7 Simulation, Evaluation and Test Results

7.1 Introduction
In order to be able to verify the functionality of the algorithm and to evaluate its performance it was implemented in a hardware prototype. It was then evaluated using audio evaluation equipment. The algorithm was also simulated using Monte Carlo simulations. This chapter describes the hardware implementation, the simulations, the evaluations and the results of these. There is no way to present in a written form how the sound quality was affected by the compensation. Therefore the results presented in this chapter are measurable properties.

7.2 Simulations
A Matlab model of the DAC chip was created. The complementary DAC architecture shown in section 5.2.2.1 and the R/2R ladder architecture presented in section 3.3 were used for simulating the D/A conversion. The simulations are best-case simulations i.e. the fault in each DAC is known exactly during the compensation. The DAC performance was measured in THD+N or SNDR during the simulations. In the DAC chip datasheet [8] the following numbers specified by the manufacturer regarding the THD+N performance of the PCM1704 DAC can be found:

<table>
<thead>
<tr>
<th>Chip</th>
<th>Typical</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>In %</td>
<td>In dB</td>
</tr>
<tr>
<td>VO = 0 dB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCM1704U</td>
<td>0.0025</td>
<td>-92.04</td>
</tr>
<tr>
<td>PCM1704U-J</td>
<td>0.0015</td>
<td>-96.48</td>
</tr>
<tr>
<td>PCM1704U-K</td>
<td>0.0008</td>
<td>-101.94</td>
</tr>
<tr>
<td>VO = -20 dB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCM1704U</td>
<td>0.008</td>
<td>-81.94</td>
</tr>
<tr>
<td>PCM1704U-J</td>
<td>0.007</td>
<td>-83.10</td>
</tr>
<tr>
<td>PCM1704U-K</td>
<td>0.006</td>
<td>-84.44</td>
</tr>
</tbody>
</table>

Table 7.1: THD+N performance of the PCM1704.

The performance is given at different input amplitudes (VO) and for different quality grades of the DAC. The DACs compensated in this thesis were of K-grade (denoted PCM1704U-K in Table 7.1)

Once simulations were started it was noticed that introducing even very small faults in the component values of the circuit yielded low performance in THD+N. In the FAE document [7] one can see that laser trimming of current sources have been performed. Because of this it was assumed that the errors in the current sources were sufficiently small to be disregarded. Even with ideal current sources small faults in the resistors led to poor DAC performance. It was therefore concluded the design of the PCM1704 had been well done.

7.2.1 Simulations of the K-grade PCM1704
In order to get performance similar to the PCM1704 K-grade in the simulated DACs the standard deviation of the resistor faults was set to 0.008% of the resistor value. Monte Carlo simulations with 10 000 simulated DACs were run.
7.2.1.1 Simulation With 0 dBFS Input Amplitude
In the first simulation the simulated input signal had an amplitude of 0 dBFS. A histogram of the simulated DACs SNDR-performance is shown in Figure 7.1. Note that this is the performance without compensation enabled.

Figure 7.1: Histogram showing SNDR performance of simulated DACs.

A histogram showing the increase in SNDR due to the compensation is shown in Figure 7.2.

Figure 7.2: Histogram showing SNDR improvement of simulated DACs when using compensation.

As can be seen the compensation improves the SNDR of all the simulated DACs.
7.2.1.2 Simulation With -20 dBFS Input Amplitude

In the second simulation the simulated input signal had an amplitude of $-20 \, dBFS$. The SNDR performance of the simulated DACs is shown in Figure 7.3.

Figure 7.3: Histogram showing SNDR performance of simulated DACs.

A histogram showing the increase in SNDR due to the compensation is shown in Figure 7.4.

Figure 7.4: Histogram showing SNDR improvement of simulated DACs when using compensation.

As can be seen the compensation improves the SNDR of all the simulated DACs.
7.2.2 Simulations of the Non-graded PCM1704
The performance of the non-graded PCM1704 (denoted PCM1704U in Table 7.1) was also mimicked in simulations. The standard deviation in the resistor faults was set to 0.025% of the resistor value in order to get the correct performance. As before a Monte Carlo simulation with 10 000 simulated DACs was run.

7.2.2.1 Simulation With 0 dBFS Input Amplitude
In Figure 7.5 a histogram of the simulated DACs SNDR-performance is shown. As before this is the performance with compensation disabled.

![Histogram showing SNDR performance of simulated DACs.](image)

Figure 7.5: Histogram showing SNDR performance of simulated DACs.

A histogram showing the increase in SNDR due to the compensation is shown in Figure 7.6.
As can be seen the compensation improves the SNDR of all the simulated DACs.

7.2.2.2 Simulation With -20 dBFS Input Amplitude

In Figure 7.7 a histogram of the simulated DACs SNDR-performance is shown. As before this is the performance with compensation disabled.
A histogram showing the increase in SNDR due to the compensation is shown in Figure 7.8.

![Figure 7.8: Histogram showing SNDR improvement of simulated DACs when using compensation.](image)

As can be seen the compensation improves the SNDR of all the simulated DACs.

### 7.3 Evaluation

#### 7.3.1 Evaluation Using Measurement Setup

The measured results shown in this section were captured using the measurement setup presented in section 4.4. Matlab was used for generating the plots. Sinusoids of different amplitude were fed to the system during the measurements with compensation turned on and off. A mean period of the measured data was created in order to remove any white noise. A DFT of the mean period was then calculated. In order to extract the distortions from this data the input signal spectral components were removed from the DFT. By calculating an IDFT of the resulting spectral information the distortion data was acquired. The distortions were plotted in order to show how they are affected by the compensation. Alongside with the distortions is a scaled down version of the input signal in order to show where certain distortions occur. The DFT where the input signal spectral components have not been removed are also plotted.

In Figure 7.9, extracted distortions from measurements with and without compensation are shown. In this case the input was a sinusoid with frequency of 100 Hz and amplitude of 0.55 relative to full swing. The distortions are plotted in blue, and a scaled down version of the input signal in red.
Figure 7.9: Distortions plotted with and without compensation.

The corresponding frequency spectrums of these signals are shown in Figure 7.10. In the frequency spectrum plots however the input signal has not been removed.
Figure 7.10: Corresponding frequency spectrums to the measured signals in Figure 7.9.

The distortions from Figure 7.9 are plotted overlaid on each other in Figure 7.11 in order to clarify the effects of compensation. The distortions from the non-compensated signal are plotted in blue, and the distortions from the compensated signal are plotted in green. A scaled down version of the input signal is plotted in red.
Figure 7.11: Comparison between extracted noise curves.

As can be seen in both the time and frequency domain the compensation clearly reduces the effects of the distortions.

In Figure 7.12 extracted distortions similar to those in Figure 7.9 are plotted. In this case the input was a sinusoid with frequency $100 \text{ Hz}$ and relative amplitude of $2^{-10} (-60 \text{ dBFS})$. This means that only the least significant 14 bits are used in the D/A conversion. The distortions are plotted in blue, and a scaled down version of the input signal in red.
Figure 7.12 Distortion plotted with and without compensation.

The corresponding frequency spectrums of these signals are shown in Figure 7.13. In the frequency spectrum plots however the input signal has not been removed.
Figure 7.13: Corresponding frequency spectrums to the measured signals in Figure 7.12.

The distortions from Figure 7.12 are plotted overlaid on each other in Figure 7.14 in order to clarify the effects of compensation. The distortions from the non-compensated signal are plotted in blue, and the distortions from the compensated signal are plotted in green. A scaled down version of the input signal is plotted in red.
Figure 7.14: Comparison between extracted noise curves.

As has been shown, even at as low input amplitudes as $-60 \text{ dBFS}$ the compensation reduces the effects of the distortions, although it is not as clearly visible as when using a larger input signal.

### 7.3.2 Algorithm Implementation

The hardware prototype developed was a modified version of the DAC from Naim Audio. The algorithm was implemented in the ADSP-21369 DSP from Analog Devices.

### 7.3.3 Evaluation Using dScope Series III

The measured results shown in this section were captured using an audio evaluation instrument called dScope Series III from PrismSound. Both the input and the output of the DAC system was connected to this instrument as it generated the input signal and analyzed the output signal. The sample rate of the digital input signal was $192 \text{ kHz}$. This was the highest sample rate that the instrument could generate.

#### 7.3.3.1 FFT Plots

The dScope Series III was used for generating time-averaged FFTs of the output signal of the DAC with compensation turned on and off for different input signals. Plots of the results from this are shown in this section. The axes in the plots have been scaled so that focus lies on the undesired frequencies and therefore the whole fundamental is not visible in the plots. The examples shows four different combinations of input signal properties. The frequency of the
incoming signal is either 100 Hz or 1 kHz while the amplitude changes from 0 dBFS to −20 dBFS.

In Figure 7.15 the input signal was a sinusoid with frequency 1 kHz and amplitude 0 dBFS.

![Compensated FFT compared to non-compensated FFT](image)

**Figure 7.15:** Comparison between an FFT of the compensated and non-compensated signal.
In Figure 7.16 the input signal was a sinusoid with frequency 100 Hz and amplitude 0 dBFS.

![Compensated and Non-Compensated FFT Comparison](image_url)

Figure 7.16: Comparison between an FFT of the compensated and non-compensated signal.
In Figure 7.17 the input signal was a sinusoid with frequency 1 kHz and amplitude -20 dBFS.

Figure 7.17: Comparison between an FFT of the compensated and non-compensated signal.
In Figure 7.18 the input signal was a sinusoid with frequency 100 Hz and amplitude 0 dBFS.

![Compensated and Non-Compensated FFT Comparison](image)

Figure 7.18: Comparison between an FFT of the compensated and non-compensated signal.

As can be seen in these figures, a lot of spurious spectral components are removed by the compensation algorithm. The noise floor is unchanged as expected.

7.3.3.2 Sweeps
The dScope Series III was used for generating frequency and amplitude sweeps of the THD+N of the output signal of the DAC with compensation turned on and off. Plots of the results from this are shown in this section.

**Frequency Sweeps**
The frequency of the input signal was varied between 20 Hz and 20 kHz while the amplitude was held constant. This was done while the compensation was turned on and off. In Figure 7.19 the amplitude of the input signal was 0 dBFS, in Figure 7.20 it was −20 dBFS and in Figure 7.21 it was −60 dBFS. In all of the plots the red curve corresponds to the compensation being turned off, while the blue curve corresponds to the compensation being turned on.
Figure 7.19: Frequency sweep at amplitude 0 dBFS.

Figure 7.20: Frequency sweep at amplitude -20 dBFS.
As can be seen in these plots the THD+N is practically lowered at all frequencies for the tested amplitudes. At low input amplitudes and high frequencies (as shown in Figure 7.21) noise has a larger impact since distortions are already low. Therefore it might seem as though the conversion with compensation yields a worse result than without compensation. However consecutive runs have shown that the results vary a lot in this region of operation and therefore no conclusion could be drawn.

Amplitude Sweeps
The amplitude of the input signal was varied between 0 dBFS and -80 dBFS while the frequency was held constant. This was done while the compensation was turned on and off. In Figure 7.22 the frequency of the input signal was 100 Hz, in Figure 7.23 it was 1 kHz and in Figure 7.24 it was 10 kHz. In all of the plots the red curve corresponds to the compensation being turned off, while the blue curve corresponds to the compensation being turned on.
Figure 7.22: Sweep of amplitude over a constant frequency of 100 Hz.
Figure 7.23: Sweep of amplitude over a constant frequency of 1000 Hz.
As can be seen in these plots the THD+N is practically lowered at all amplitudes for the tested frequencies.

### 7.4 Discussion of Simulation and Evaluation Results

As has been shown in both simulations and evaluations of the algorithm it has the ability of reducing the R/2R distortions. The results presented in the simulations section are somewhat optimistic since what is simulated is the best-case since the exact bit level outputs are known when compensating. However the results presented in the evaluation section are very realistic since they are measurements of a real world implementation.
8 Conclusion and Future Work

8.1 Introduction
The ultimate goal of this thesis was to obtain knowledge that could be used in order to increase the sound quality in a hi-fi audio digital-to-analogue conversion. The main objective however was to investigate if it would be possible to compensate for nonlinearities in the output signal caused by faulty component values. These two aims should be separated for two reasons. The first reason is that even though it was desired by everyone involved that increased linearity would lead to an enhanced audio experience there were never any guarantees for this. This is so because of the fact that linearity is measurable property whereas the audio quality experienced by a human is not. However this does not imply that the linearity is irrelevant; assuming that increasing linearity is correlated with increasing audio quality is not an unfounded assumption since it generally is true. The second reason would be that the results produced in this thesis would not be limited to audio applications. Instead it would be applicable to R/2R digital-to-analogue conversion in general. With this in mind we can now proceed to the conclusion.

8.2 Conclusion
In this thesis a way of compensating for nonlinearities associated with R/2R DACs has been developed. A compensation algorithm was developed, simulated, implemented and evaluated. Results so far have proven that the algorithm is successful in reducing nonlinearities in R/2R digital-to-analogue conversion. The implementation has also been subjected to listening tests performed at Naim Audio and the feedback from these tests has been positive.

This thesis resulted in a functioning prototype of a system using the developed compensation algorithm. Though the audio quality was improved it is suspected that it could be improved even more. However as a master’s thesis is limited in time this was left for future work.

8.3 Future work
Some possible improvements to the algorithm have been suggested in section 6.4, i.e. keeping the lowest part of the input word untouched, always evaluating the word that is one LSB larger than the compensated word and using a different bit level normalization. All of these improvements could be investigated in future work. Another large area included in future work would be to improve the quality of the measurements.
9 References

Appendix A: Division of work

Before the thesis work was started the pre-study was conducted in parallel in order to cover as much material as possible. Although there were a lot of dependencies between the different sections of this project, the aim was to parallelize the work as much as possible. This resulted in that a lot of the modeling work that was done in Matlab was performed simultaneously. Also simulations preformed on captured data could be run by both users on different computers. However the tasks that required measuring or modification on the actual hardware could not be parallelize to any greater extent. It was discovered that this was beneficial for the thesis work since both of the students were involved in the whole project. Both students could verify the work that had been done and could correct for errors. During the writing of the thesis report, the writing of chapters was divided between the two authors. The section below describes how the chapters were divided.

Division of thesis report

Chapter 1: Both authors contributed equally to this chapter.

Chapter 2: Written by Gabriel.

Chapter 3: Written by Gustav except for section 3.3.1-3.3.2 which were written by Gabriel.

Chapter 4: Written by Gustav.

Chapter 5: Written by Gabriel.

Chapter 6: Written by Gabriel.

Chapter 7: Both authors contributed equally to this chapter except for section 7.2 which was written by Gabriel.

Chapter 8: Both authors contributed equally to this chapter.