Master Thesis Project (EE: 191211219, EmSys: 192199978)

Multi-Rate Discrete Fourier Transform Characteristics: Results of Models, Simulations, and Measurements

Wendo Beuker MSc. Thesis Electrical Engineering & Embedded Systems

> Faculty of Electrical Engineering, Mathematics & Computer Science

December 2019

Supervisors prof. dr. ir. B. Nauta dr. ir. A-J. Annema M. Huiskamp MSc Chair of Integrated Circuit Design

Dr. Ir. A.B.J. Kokkeler Chair of Computer Architecture for Embedded Systems

Report number: 067.3826

Confidential until November 2023

Faculty of Electrical Engineering, Mathematics and Computer Science University of Twente P.O. Box 217 7500 AE Enschede The Netherlands

UNIVERSITY OF TWENTE.

Abstract

The Discrete Fourier Transform (DFT) is a technique to assess data in a sampled signal from a frequency point of view. This technique is widely used in industry; from wireless communication to mobile devices. A limitation of the DFT is that the highest recoverable frequency component (the so called Nyquist frequency) is limited by the sample frequency of the measurement system. At the Integrated Circuit Design group of the University of Twente, an extension to the DFT is developed. This extension eliminates this limitation on the expense of multiple measurements at different sample frequencies. This extension is called the Multi-Rate Discrete Fourier Transform (MR-DFT). The intended use case of the MR-DFT is in the field of characterising class-E Power Amplifiers to compensate for PVT spread and load impedance mismatch.

The MR-DFT is the main focus of this work. In this work a measurement setup to measure the MR-DFT is presented and measurement results of QAM constellations are presented. With the current setup, frequencies of $1.75 \cdot f_{nyquist}$ are measurable. Also the mathematical background of the MR-DFT is presented. Furthermore, the effect of noise on the MR-DFT is described and a model for the impact of noise is presented. This model is compared to simulations and measurements.

As case study, the effect of quantization noise on the noise floor of the output of the MR-DFT is investigated. Although the noise (floor) profiles of the measurements, model, and simulations do not match in an absolute manner, the noise floors have a similar profile.

The model is a step into the right direction but it is suspected some of the assumptions are not correct, deviating the model results from simulations.

It is suspected that the measurements deviate from the simulation due to a limitation in the measurement setup and since other noise sources are present (like jitter) in the measurement setup. They seem to have a larger impact on the noise floor than the quantization noise, making it impossible to assess the effect of the quantization noise.

Although the results on the expected noise floor of the MR-DFT of measurements, simulations, and a model are not (exactly) in line, this work is a great step toward unravelling the mysteries of the MR-DFT.

Contents

Ι	Int	trodu	ction	6										
1	Introduction													
2	2 Introduction to the MR-DFT													
	2.1	Discre	te Fourier Transform and Aliasing	11										
	2.2	Multi-	Rate Discrete Fourier Transform	12										
	2.3	Increa	se of Spectral Resolution	14										
	2.4	MR-D	FT Measurement Setup Implications	14										
11	N	IR-DI	FT Mathematics	18										
3	Mat of N	thema /IR-DH	tical Background T	19										
	3.1	List of	f Symbols	19										
	3.2	Termi	nology and Basic Principles	20										
		3.2.1	Discrete Fourier Transform, Bins and Harmonics	20										
		3.2.2	Sample Set and its Discrete Fourier Transform	21										
		3.2.3	Collection of Sample Sets	21										
		3.2.4	Beat Period	21										
		3.2.5	Requirements for a Collection of Sample Sets	21										
		3.2.6	Harmonic Matrices	21										
		3.2.7	Subdivision of Harmonic Matrices	24										
		3.2.8	Collection of Discrete Spectra	25										
	3.3	MR-D	FT in a Mathematical Context	25										
		3.3.1	Number of Reconstructed Frequency Bins	26										
		3.3.2	Frequency Resolution of MR-DFT	27										
		3.3.3	MR-DFT-Matrices and \vec{Y}	27										
	3.4	Nyqui	st-Shannon Criterion and the MR-DFT	29										

III Digital Design

 $\mathbf{32}$

	4.1	Measurement Goal and Plan	33												
	4.2	Requirements													
	4.3	Hardware Selection and Consequences	34												
		4.3.1 FPGA and ADCs	34												
		4.3.2 Arbitrary Waveform Generator	34												
	4.4	Conceptual Design	35												
5	Syst Des	tem Components and sign Choices	37												
	5.1	Clock Generations and Clock Domains	37												
		5.1.1 Clock Generation	37												
	5.2	ADC to FIFO	37												
		5.2.1 Connectivity to A2F	38												
		5.2.2 Clock Domains and the A2F	38												
		5.2.3 Clock and Data Synchronisation	39												
		5.2.4 Counter and Timestamping	41												
		5.2.5 Data to FIFO	41												
	5.3	FIFOs	42												
		5.3.1 FIFO: COTS vs Custom	42												
		5.3.2 Connectivity of FIFO	42												
		5.3.3 Memory Array	42												
		5.3.4 Control Logic	42												
	5.4	Operating System	43												
	5.5	Timing Issues	43												
IV	/ 1	Measurement Setup, Method, and Results	46												
6	Mea	asurement Setup and Method	47												
	6.1	Measurement Setup	47												
		6.1.1 Clock Synchronisation	47												
		6.1.2 Data Synchronisation	47												
		6.1.3 AWG Settings	49												
	6.2	Measurement Method	50												
		6.2.1 Non-OFDM QAM16	50												
		6.2.2 OFDM QAM16 without Symbol Rotation	50												

		6.2.3 OFDM QAM16 with Symbol Rotation	51
		6.2.4 OFDM QAM64 with Symbol Rotation	51
		6.2.5 Random Signals	51
7	Mea	asurement Results	53
	7.1	Non-OFDM QAM16	53
	7.2	OFDM QAM16 without Symbol Rotation	55
	7.3	OFDM QAM16 with Symbol Rotation	56
	7.4	OFDM QAM64 with Symbol Rotation	58
	7.5	Comparison and Conclusions	61
V	N	IR-DFT and Noise	62
8	Dist	turbances and Noise in MR-DFT	63
	8.1	Disturbances: A Spectral Approach	63
		8.1.1 ADC Sampling Clock Jitter and (MR-)DFT	63
	8.2	Disturbances: A Statistical Approach	64
		8.2.1 The Statistics of Quantization Error and the DFT	64
	8.3	MR-DFT Noise Shaping	66
9	Noi Sim	se Comparison: Theory, nulations, and Measurements	71
	9.1	Model	71
	9.2	Simulations	72
		9.2.1 Sampled Random Signal and MR-DFT (Noise Floor) Behaviour	72
		9.2.2 Quantized Cosine and MR-DFT Noise Floor Behaviour	74
	9.3	Measurements	77
		9.3.1 Wideband Noise	77
		9.3.2 Quantization Noise in Sine Measurements	78
	9.4	Discussion of Results	79
	9.5	Discussion of Measurement Method	80
	9.6	Conclusion	80
V	ΙI	Discussion, Conclusion, and Recommendations	82
10	Dis and	cussion, Conclusion, l Recommendations	83

10.1 Summary	83
10.2 Discussion	84
10.3 Conclusion	84
10.4 Recommendations	85
Bibliography	88
Appendix A Algorithm to Count Number of Solvable Linear Equations	89
A.1 Algorithm 1	89
A.2 Algorithm 2	90
Appendix B Frequency Locking Error and (MR-)DFT	92
Appendix C Lyapunov CLT Simulations	95
Appendix D Complex Standard Normal Distribution	99
Appendix E Squared Normal Distribution and its Relation to the Chi-Squared Distribution	101

Part I

Introduction

Chapter 1: Introduction

In the current society, humankind would like to have the world in their hand. Humankind would like to be able to communicate with everybody, always, and everywhere. Mobile communication enables humankind to do so. At the moment, it is possible to message friends, read newspapers, or order your groceries while travelling to work, during meetings, or even while walking your dog. This need of humankind goes hand in hand with the digitisation of society; data production experiences an exponential growth rate [1]. The increasing amounts of mobile generated data by both consumers as well as devices (i.e. Internet of Things devices/services) both enables as well as drives the development for all kind of RF applications like 4/5G, Bluetooth, and LoRaWAN to connect all these devices to the ever growing internet. Novel engineering solutions are needed to transmit, receive, and handle these exponentially growing amounts of (mobile) data in an effective and (power) efficient manner.

To enable the transmission and handling of these ever increasing data streams, RF-transceivers are more and more integrated in CMOS technology. Where 60 years ago you needed a microwave sized device to only receive AM radio, nowadays smartphones support numerous wireless protocols via fully integrated CMOS RF-transceivers. The transmitter part of these fully integrated transceivers are more and more build around switched mode Power Amplifiers (PA). The advantage of using switched mode PAs over class A/AB PAs is in their increased power efficiency at the cost of lower linearity and more complex driving circuitry.

The performance of (switched mode) PAs is negatively influenced by Process, Voltage and Temperature (PVT) spread as well as load impedance variations. PVT spread results in a decrease in both linearity and efficiency [2]. Handling mobile communication devices significantly alters the impedance of the antenna/load [3] which results in an altered radiation efficiency and reflection of power back toward the PA. The latter could damage the PA.

The power reflection problem can be tackled by designing robust PAs. A robust PA includes enough margin in its design to not be damaged by reflected power. This results in an increase of area and power usage of such a PA. Also this does not counter the efficiency degradation due to PVT spread. Therefore, the common solution at the moment is to compensate load impedance variations and PVT spread via Adaptive Digital Pre-Distortion (ADPD) in combination with adaptive matching networks and tunable bias networks[2]. Adaptive matching networks can already be implemented efficiently [4]. The current compensation techniques include the use of temperature sensors, DC-sensors, power detectors [5], and peak detectors [6] to control the hardware to compensate for PVT spread and load impedance variations [2].

For proper control of the Switch Mode PA via ADPD matching networks, information about the linearity and load impedance is necessary. This information can be retrieved from the harmonics of the RFwaveform [2][7]. Especially for Class-E AP, the first three or four harmonics are of utmost interest. However, in the case of PVT spread and the load impedance variations compensation techniques discussed before, the necessary information about the RF-waveform is lost [2]. [2] proposes a circuit topology which directly measures the output of the PA in a high ohmic manner such that the necessary RF-waveform information can be measured while the transmitter is in service.

Besides this, another problem arising is the increase of carrier frequency for radio applications. There is a lot of free spectrum in the mm-Wave spectrum between 10 GHz and 100 GHz usable for e.g. 5G applications [8]. At the moment of writing, most countries have devoted a frequency band to 5G between 24 GHz and 28 GHz and the Federal Communications Commission reserved spectrum from 64 - 71 GHz for use of 5G in the United States [9]. The 3rd harmonic of the 28 GHz band is around 85 GHz. An ADC with at least 170 GS/s is needed to measure this signal. Luckily, these are available. The author could find two oscilloscopes at the market with sample rates over 200 GS/s. These cost over a million per piece and weigh more than 25 kg. So they cannot be included within a PA for mobile applications. At the moment of writing, it is not possible to measure the third or fourth harmonic of PAs operating in this 5G specific, and most of the the mm-Wave spectrum in general, on chip. A solution is needed to measure the output harmonics of a Switch Mode PA for mm-Wave applications.

One of the properties of PVT spread and load impedance variations is that they are relatively slow

processes. Load impedance variations are in the milliseconds scale [6] since one cannot pick-up a mobile phone within a microsecond. Also PVT spread variations are within this millisecond range. So the problem boils down to how to measure low frequency modulation of the high frequency RF carrier and harmonics without losing signal waveform information.

To tackle this problem, Maikel Huiskamp of the ICD group at the University of Twente, proposed an expansion to the Discrete Fourier Transform (DFT). This expansion, that is called the Multi-Rate DFT (MR-DFT), enables to reconstruct frequency information from higher Nyquist zones¹ at the expense of time, without increasing the sample frequency.

Research Questions

This thesis is devoted to the MR-DFT: How does it work? Can the algorithm be expressed in a mathematical fashion? Is the algorithm usable for measurement? What is the effect of noise on the algorithm? To unravel the properties of the MR-DFT, three research questions are formulated:

- 1) Can a mathematical framework be constructed that describes the MR-DFT?
- 2) Is it possible to create a measurement setup that measures higher Nyquist zones via the MR-DFT?
- 3) What is the impact of noise on the MR-DFT and can a model be constructed and verified?

This research is performed for the Integrated Circuit Design group and the Computer Architecture for Embedded Systems group at the University of Twente to finish a double degree in both Electrical Engineering and Embedded Systems.

Document Structure

The document is divided in five parts. Part I introduces the research topic in general. The introduction is found in Chapter 1 and Chapter 2 provides an introduction in the MR-DFT.

Part II is devoted the the mathematical framework which describes the MR-DFT in a mathematical fashion. Chapter 3 describes how the MR-DFT can be describes in formulas and equations, it states what requirements should be met, and what data is relevant for performing a MR-DFT.

In Part III, the design of the measurement instrument, used to perform MR-DFT measurements, is presented. Chapter 4 describes the goal of the measurement instrument and the conceptual design of the measurement instrument. In Chapter 5, the actual digital design of the measurement instrument is presented.

The next part, Part IV, presents the measurement on the MR-DFT by the measurement instrument. Chapter 6 presents the measurement setup and measurement method. Chapter 7 presents and discusses the measurement results and anomalies found.

Part V covers the influence of noise on the MR-DFT. In Chapter 8, the effect of noise sources and other disturbances on the output of the MR-DFT is investigated. In this chapter a model for the effect of noise on the MR-DFT is presented. Chapter 9 presents the results of the noise model, as well as simulations of noise on the MR-DFT and measurements of noise on the MR-DFT. The results of the model, simulations, and measurements are compared and discussed and conclusions are drawn from it.

The last part is Part VI. This part contains the discussion, conclusion, and recommendations of this thesis. All these matters are discussed in Chapter 10.

The rest of the document consists of an Bibliography and a set of appendices.

¹Higher Nyquist zones refers to all Nyquist zones larger than zone 1.

Acknowledgement

I would like to thank Dr. Ir. Anne-Johan Annema, Dr. Ir. Andr Kokkeler, and Maikel Huiskamp MSc. for supervising my master thesis. Furthermore, I would like to thank Oguz Meeter MSc. for this help setting-up the FGPA/SoC system. Also, I would like to thank Ir. Bert Molenkamp for this VHDL, Quartus, and ModelSim support. At last, I would like to thank Dr. Chris Zeinstra for his help with statistics. Without the support of these people, I could have never reached the results I did.

Chapter 2: Introduction to the MR-DFT

This chapter provides an introduction into the Multi-Rate Discrete Fourier Transform (MR-DFT). It qualitatively describes the MR-DFT by means of an example. Also, some properties of the MR-DFT are introduced as well as the intended application area and implications for the measurement setup. The next chapter, Chapter 3, provides the mathematics involved in the MR-DFT.

Note that the MR-DFT algorithm is not an invention of the author but is invented at the Integrated Circuit Group at the University of Twente. The information in this chapter is mostly based on [10].

2.1 Discrete Fourier Transform and Aliasing

The Fourier Transform is a signal analysis technique in which a periodic signal is decomposed into the frequency components it consists out of. The Fourier Transform is performed on time continues periodic signal. A time discrete version of the Fourier Transform exists, which is called the Discrete Fourier Transform (DFT). Mathematically speaking, the DFT looks like:

$$X_{k} = \frac{1}{N} \cdot \sum_{n=0}^{N-1} x[n] \cdot e^{\frac{-j \cdot 2\pi \cdot n \cdot k}{N}}$$
(2.1)

in which x[n] is the time discrete input signal, N is the number of samples of x[n], X_k is the output vector of the DFT, and j is the imaginary unit. The DFT can also be applied to complex signal but in this work it is assumed that the converted signals are real.

 X_k is the complex output vector of the DFT and represents the frequency spectrum. An element of X_k is called a "bin" and contains information about a small part of the spectrum. The absolute value of the bin represents the amplitude of the frequency component in that part of the spectrum and the phase of X_k represents the phase of this frequency component. In this manner, the original discrete signal can be reconstructed by

$$x[n] = \sum_{k=0}^{N-1} X_k \cdot e^{\frac{j \cdot 2\pi \cdot n \cdot n}{N}}$$
(2.2)

which for real signals evaluate to:

$$x[n] = \sum_{k=0}^{N-1} |X_k| \cdot \cos\left(\frac{2\pi \cdot n \cdot k}{N} + angle(X_k)\right)$$
(2.3)

in which angle(X) is the operation resulting in the phase of the complex number X.

A bin of X_k spans a bandwidth equal to:

$$\Delta f = \frac{f_s}{N} \tag{2.4}$$

in which f_s is the sample frequency of the data. Δf is normally referred to as the bin size or the frequency resolution of the DFT. The centre frequency in a DFT bin can be calculated by:

$$f_{k,centre} = k \cdot \Delta f \tag{2.5}$$

This means that bin k of \vec{X} (X_k) spans the frequency spectrum from $f_{k,centre} - \Delta f/2$ Hz to $f_{k,centre} + \Delta f/2$ Hz except for bin 0, the "DC bin", which spans from 0 Hz to $\Delta f/2$ Hz¹.

Due to the discrete nature of the sample moments and the limited number of samples, some frequencies cannot be distinguished. The highest distinguishable frequency is half of the sample frequency and is called the Nyquist frequency. Higher frequencies appear as lower frequencies after a DFT. This is shown in Figure 2.1. This images shows a plot of 2 sines which are sampled at the same moment (orange circles). Although the bottom sine has a frequency 5 times higher than the top one, the sampled data is identical. The DFT on both data sets results in the same frequency component. This effect is called aliasing and it is said that the high frequency sine is "folded" to the low frequency bin.

¹For real input signals.



Figure 2.1: The origin of aliasing. Due to the limited number of discrete sample moments, not all frequencies can be distinguished.

2.2 Multi-Rate Discrete Fourier Transform

Aliasing in the DFT occurs in predictable patterns and can be "misused" to recover extra information. Take a real signal x(t) composed out of a DC component R_0 and 6 complex harmonics at frequencies 1 Hz to 6 Hz. The real parts referred to as R_1 to R_6 and imaginary parts as S_1 to S_6^{-2} . The signal is sampled with 8 evenly in time spaced samples (e.g. at 8 Samples/second) and the transformed to the frequency domain. Folding patterns are observed for the 5th and higher harmonic (aliasing). The folding of the real parts is represented in Table 2.1a and the folding of the imaginary parts is represented in Table 2.2a. The rows of the tables represent bins and the columns represent harmonics. When a row contains " R_2 " and " R_6 ", it means that both the real part of the second and the sixth harmonic end up in this bin. When a column contains " S_3 " and " $-S_3$ " in the rows of bin 3 and 5, it means that the imaginary part of the third harmonic ends up in bin 3 and in bin 5³.

How can the effective bandwidth be increased (find higher harmonics⁴) without increasing the sampling frequency? Assume the same part of signal x(t) is also sampled with 7 evenly in time spaced samples (at 7 S/s), the higher harmonics fold to different bins than in the previous example. The folding of the real parts is represented in Table 2.1b and the folding of the imaginary parts is represented in Table 2.2b. Table 2.1 and Table 2.2 show that the folding patterns of the two sample sets differ. This results in a system of linear equations that can be solved to reconstruct harmonic information from higher Nyquist zones (actually it results in two systems: one for the real part of the complex harmonics and one for the imaginary part of the complex harmonics). In this system of linear equation (shown in tabular form in Table 2.1 and Table 2.2 and in mathematical from in Equation 2.6 and Equation 2.7), DC and 6 harmonics can be reconstructed. The linearly independent columns are shown in green in Table 2.1 and Table 2.2⁵. This results in two systems of linear equations:

²Within the mathematics of the MR-DFT, matrix operations are preformed. To prevent mixing of the imaginary part of harmonics and the identity matrix, the imaginary part of harmonics is denoted by S instead of I.

³Note that the minus sign of the imaginary parts originates from the complex conjugate of the complex harmonic $(R_x + j \cdot S_x)$ due to negative frequencies. After a DFT, the negative frequencies end up in the harmonic bins of positive frequencies in a mirrored fashion.

⁴Higher harmonics refers to harmonics from higher Nyquist zones (zones 2 and up).

 $^{^{5}}$ Actually, there are 8 independent columns for the real part and 7 for the imaginary part. Since it makes no sense to only reconstruct the real part of the 7th complex harmonic, bin 4 of Table 2.1a kept out of the system of linear equations. Also bin 0 of Table 2.2a is not shown in green since the imaginary part of DC is 0 by definition.

$Re(X_{8,0}) = R_0$		$Im(X_{8,0}) = 0$	
$Re(X_{8,1}) = R_1$		$Im(X_{8,1}) = S_1$	
$Re(X_{8,2}) = R_2 + R_6$		$Im(X_{8,2}) = S_2 - S_6$	
$Re(X_{8,3}) = R_3 + R_5$	(2.6)	$Im(X_{8,3}) = S_3 - S_5$	(2.7)
$Re(X_{7,1}) = R_1 + R_6$		$Im(X_{7,1}) = S_1 - S_6$	
$Re(X_{7,2}) = R_2 + R_5$		$Im(X_{7,2}) = S_2 - S_5$	
$Re(X_{7,3}) = R_3 + R_4$		$Im(X_{7,3}) = S_3 - S_4$	

In this context (and the context of the rest of the report) $X_{c,k}$ is the k-th bin of the DFT of a sample set containing c samples of x(t). In this example k runs from 0 to c-1 and c is either 8 or 7.

If these systems of linear equations are independent, the systems are solvable. For the given example, R_0 and R_1 are extracted from $\operatorname{Re}(X_{8,0})$ and $\operatorname{Re}(X_{8,1})$. R_6 is found by subtracting R_1 from $\operatorname{Re}(X_{7,1})$. Subtracting R_6 from $\operatorname{Re}(X_{8,2})$ results in R_2 . Continue this process to retrieve the real part of all harmonics. A similar procedure is used to reconstruct the imaginary part of the selected harmonics. The solution of this system of linear equations is shown in Equation 2.8 and Equation 2.9.

$$\begin{array}{ll} R_{0} = Re(X_{8,0}) & S_{0} = 0 \\ R_{1} = Re(X_{8,1}) & S_{1} = Im(X_{8,1}) \\ R_{6} = Re(X_{7,1}) - R_{1} & S_{6} = S_{1} - Im(X_{7,1}) \\ R_{2} = Re(X_{8,2}) - R_{6} & (2.8) & S_{2} = Im(X_{8,2}) + S_{6} & (2.9) \\ R_{5} = Re(X_{7,2}) - R_{2} & S_{5} = S_{2} - Im(X_{7,2}) \\ R_{3} = Re(X_{8,3}) - R_{5} & S_{3} = Im(X_{8,3}) + S_{5} \\ R_{4} = Re(X_{7,3}) - R_{3} & S_{4} = S_{3} - Im(X_{7,3}) \end{array}$$

		Har	mon	ics -	real pa	\mathbf{rt}									
Ding	0	1	2	9	1	F	Harmonics - real part								
DIIIS	U	1	4	э	4	0	0	Bins	0	1	2	3	4	5	6
0	2*R0								0*D0	-	-	- -	-	-	•
1		R1						U	2*R0						
1		101	DO					1		R1					R6
2			R2				R6	2			DO			DE	
3				R3		R5		4			n2			nə	
4				100	0*D4			3				R3	R4		
4					2*R4			4				R3	$\mathbf{R}4$		
5				R3		R5		-1			DO	100	104	D F	
6			DO				DE	5			R2			R5	
0			n2				no	6		R1					R6
7		R1								101				<u> </u>	100
		1	·	I	1	I				(b)					
			(a)								. ,				

Table 2.1: Tabular representation of aliasing of the real parts of an eight-point DFT (a) and of a seven-point DFT (b). The rows represent the harmonic bins and columns represent the (real part) of the harmonic information. The tables shows in which bin which harmonic ends up; the tables visualise the aliasing patterns. The rows in green show the linearly independent rows.

This concept can be extended to more sample sets thereby providing the possibility to recover more harmonics. For example: for 4 evenly spaced sets containing 8, 7, 6 and 5 samples, which are sampled over the same piece of signal, can recover up to 10 harmonics.

Harmonics - imaginary part										-					
Ding	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$								Harmonics - imaginary p						oart
DIIIS	U	T	4	э	4	0	U	Bins	0	1	2	3	4	5	6
0								0	-	-		-	-	-	-
1		T1						U							
-			10				TC	1		I1					-I6
2			12				-10	2			12			-15	
3				I3		-I5					14	TO	TA	-10	
1					14 14 = 0			3				13	-14		
4					14 - 14 = 0			4				-I3	I4		
5				-I3		I5		- F			10			TE	
6			-I2				16	Э			-12			10	
7		T1					10	6		-I1					I6
1		-11						L			(1	\	1		L
				(a)						(b)				

Table 2.2: Tabular representation of aliasing of the imaginary parts of an eightpoint DFT (a) and of a seven-point DFT (b). The rows represent the harmonic bins and columns represent the (imaginary part) of the harmonics information. The tables shows in which bin which harmonic ends up; the tables visualise the aliasing patterns. The rows in green show the linearly independent rows.

2.3 Increase of Spectral Resolution

Like normal DFTs, in MR-DFTs it is possible to increase the frequency resolution by extending the sampling set length while keeping the sample frequencies constant. So, for example, instead of using 8, 7, 6 and 5 point sets, use $g \cdot 8$, $g \cdot 7$, $g \cdot 6$ and $g \cdot 5$ point sets (in which g is an integer larger than one). This increases the spectral resolution by a factor of g^{-6} . If spectral leakage is a serious problem for an application, increasing the spectral resolution might be a solution⁷.

2.4 MR-DFT Measurement Setup Implications

The need for measuring multiple sample sets over the same signal waveform has some implications for the measurement method and therefore also for the measurement setup. The sample sets can be measured in two manners: parallel in time or sequential in time. This section introduces the two manners and discusses its implications.

In case of a parallel measurement, the different sample sets are measured at the same time. Multiple equivalent measurement systems should be in place which operate in parallel and independently. The phase difference between the various sample clocks is preferably zero or should be known.

In case of a sequential measurement, the different sample sets are measured one at the time. Measuring in a sequential manner sounds counter intuitive since signals differ over time. There are two reason why it is possible in the intended application:

- The processes responsible for PVT-spread and load impedance variations are relatively slow (ms range, see Chapter 1).
- Due to the used modulation techniques, the same symbol is (on average) transmitted again relatively fast (ms range).

The second reason needs some more explanation. Wireless communication system normally send their data via a modulation scheme like Quadrature Phase Shift Keying (QPSK) or Quadrature Amplitude

 $^{^{6}}$ Strictly spoken, the increase of resolution by a factor of g is not correct. For the exact number of recoverable harmonics, see subsection 3.3.1.

⁷I.e. in an application in which a signal of interest and an interferer are in neighbouring bins. Due to a small frequency offset of the sampling clock, spectral energy of the interferer might leak into the bin of the signal of interest and the other way around, making the two signal indistinguishable. Increasing the spectral resolution increases the bin spacing between the signal of interest and the interferer. Since the amount of leaked energy due to a sampling clock offset decreases in a sinc-like manner, increasing the bin spacing significantly decreases the leakage of energy between the bins of the interferer and the signal of interest



Figure 2.2: Example of the sequential sampling process of symbol X for 4 sample sets. The symbol duration is not long enough to sample all 4 sample sets during the first occurrence of symbol X. The last two sample sets are sampled during the next occurrence of symbol X.

Modulation (QAM). The symbol rate per (sub-)carrier of these kind of systems is generally orders of magnitude lower than the carrier frequency. The same symbol is therefore transmitted for a significant amount of time. For example, LTE-Advanced (4G) uses 15 ksps (kilo symbol per second) per sub-carrier [11] while the carrier frequency is in the 800+ MHz bands; the time one symbol is transmitted is 67 µs. Therefore, it might be possible to capture multiple sample sets of the symbol within this symbol time in a sequential manner. Otherwise, in case of 64QAM modulation⁸ every symbol is transmitted on average every 4 ms. So, if it is impossible to capture all sample sets during the 67 µs period a symbol is transmitted, on average within 4 ms the symbol occurrences again and the remaining remaining sample sets can be captured.

Figure 2.2 visualises the sequential concept. Of the 4 sample sets, the first two are sampled during the first occurrence of symbol X. The last two are sampled during the next occurrence of symbol X. Since PVT variations and load impedance variations are slow processes (within the millisecond range, as discussed in Chapter 1), waiting for the next occasion of the same symbol to capture the remaining sample sets is expected to be feasible and valid.

When measurements are used as input for the MR-DFT algorithm, the measurement setup should be able to perform parallel or sequential measurements. A possible implementation of the sequential measurement method is shown in Figure 2.3. On top, the transmitter chain with a Local Oscillator (LO), mixer, Power Amplifier (PA), and antenna and matching network is shown. The N-path filter (denoted by "Passive Mixer") is used to sample and mix down the RF-waveform. In case e.g. 8 samples are captured per beat period, the N-path filter will consist of 8 paths. In case e.g. 7 samples are captured one of the branches of the N-path filter is not used. A Delay Locked Loop (DLL) with variable length generates the evenly spaced clock phases (derived from the RF carrier frequency) to drive the switches of the N-path filter. In this foregoing examples, the DLL contains 8 or 7 delay elements depending on which sample set is being captured at the moment. A baseband ADC can now be used to convert the captured RF waveform to the digital domain. This digital data can be used as input for the MR-DFT.

⁸The 3GPP LTE-Advanced specifications also include higher order modulation schemes (up top 256QAM). Due to noise requirements of the higher order modulation schemes it seems unlikely that these will be used in practical scenarios [12].



Figure 2.3: Implementation of sequential measurement method. The Rotation Delay Locked Loop (which could also be replaced by an Phase Locked Loop structure) is capable of the generation the necessary evenly spaced clock cycles to drive an N-path filter. The N-path filter mixes the RF waveform to baseband. The waveform is captured by a "low frequency" baseband ADC. The MR-DFT is performed in the digital domain. Image retrieved from [2].

Part II

MR-DFT Mathematics

Chapter 3: Mathematical Background of MR-DFT

This chapter provides the mathematical background of the Multi-Rate DFT (MR-DFT). The chapter starts with a List of Symbols. Afterwards, the terminology and the basic principles used in the theory are introduced. The next section introduces the mathematics of the MR-DFT itself and provides an example. The last sections covers the Nyquist-Shannon criteria and the effect of it on the MR-DFT.

Note that the author is not the inventor of the MR-DFT algorithm itself. The MR-DFT is invented at the Integrated Circuit Design group at the University of Twente. The mathematical framework for the MR-DFT (presented in this chapter) is created by the author.

3.1 List of Symbols

This section contains a list of symbols. Not all symbols in the list are used in this chapter. Some are introduced in Chapter 8. Note that the definition used for \mathbb{N} include all non-negative integers (like the definition in ISO 80000-2). If only positive integers are meant, $\mathbb{N}_{>0}$ is used.

x(t)	Time continues signal.
\vec{x}_c	The Sample Set of $x(t)$ containing exactly c samples per beat period T_b . So loosely
	spoken a set of samples of a time discreet version of $x(t)$, which is sampled at c/T_b
	Samples/s.
$x_c[n]$	The <i>n</i> -th sample of <i>c</i> -th Sample Set op $x(t)$, in which $n \in [0, N_c - 1]$.
\vec{X}_c	Column vector containing Discrete Fourier transformed version of c-th Sample Set,
	thus a discrete spectrum of \vec{x}_c .
$X_{c,k}$	Frequency bin k of Discrete Fourier transformed version of c -th Sample Set (i.e.
	element k of \vec{X}_c : $X_{c,k} = \vec{X}_c(k)$). $k \in [0, N_{c-1}]$
$ec{Y}$	Column vector of length M containing linear independent frequency bins of \vec{X}_c for all c .
\vec{Y}_{r}	Column vector of length M containing the real part of \vec{Y} . Strictly spoken this is not
- 1	correct and a better definitions is given in subsection 3.3.3.
\vec{Y}_i	Column vector of length M containing the imaginary part of \vec{Y} . Strictly spoken this
- 1	is not correct and a better definitions is given in subsection 3.3.3.
c, ci	c and ci are used to identify Sample Sets: c in general and ci for a specific Sample Set.
,	The i in <i>ci</i> should be interpreted as a subscript. Since <i>ci</i> is used in subscripts most
	of the time, this would result in a subscript in a subscript. This is not readable.
	Therefore, the i in ci is not written as subscript. The value of c is equal to the
	amount of samples per beat period of \vec{x}_c . I.e. the amount of samples of \vec{x}_c that
	spans T_b . $i \in \{0, 1,, d-1\}$ such that $c \in \{c0, c1,, c(d-1)\}, ci \in \mathbb{N}_{>0}, ci \neq cj$
k	Counter for frequency bins of a certain DFT transform. $k \in [0, N_c - 1]$
d	The number of Sample Sets within a Collection of Sample Sets.
$t_{c,n}$	The moment in time at which the n -th sample of Sample Set c is sampled, i.e.
_	$x_c[n] = x(t_{c,n}).$
\vec{t}	A vector containing all ideal sample moment within a beat period of a Collection \vec{of}
→	Sample Sets for all Sample Sets. Some sample moments occur multiple times in t .
t_{uniq}	A vector containing all unique elements of t . The sample moments that occur
	multiple times in t , only occur once in t_{uniq} .
$t_{uniq,a}$	Element a of t_{uniq} : $t_{uniq,a} = t_{uniq}(a)$
$\#t_{uniq,a}$	The number of times $t_{uniq,a}$ occurs in t .
T_c	Time between two samples of Sample Set c. This implies that $T_c \cdot c = T_b$.
T_b	Beat period of a Collection of Sample Sets.
$f_s(t), f_{s,c}(t)$), An ideal sample clock, the sample clock of the <i>c</i> -th Sample Set, the frequency of $(1, 1)$
$f_{s,c}$	$f_{s,c}(t)$. I.e $\frac{1}{f_{s,c}} = T_c$

$F_s(\omega), \vec{F}_{s,c}(\omega)$	The Fourier transform of a sample clock $f_s(t)$, the Fourier transform of the sample
	clock of the <i>c</i> -th Sample Set $f_{s,c}(t)$.
M	Number of reconstructed frequency bins.
m	Counter for reconstructed frequency bins. $\{m \in \mathbb{N} 0 \le m < M - 1\}$
N_c	Number of samples in c-th Sample Set (and therefore also the number of frequency
	bins in the DFT of the c -th Sample Set). This is always a multiple of c , which is
	defined as $N_c = c \cdot g$.
n	Mostly used as counter for different bins. In that case $n \in [0, N_c - 1]$.
g	The number of beat periods within a Sample Set, thus $g = \frac{N_c}{c}$. Note that this is valid for all c . $\{g \in \mathbb{N}_{>0}\}$
Q	The number of unsolvable linear equation due to shared sample moments between (multiple) Sample Sets of a Collection.
Ĥ	Column vector of length M containing the reconstructed frequency bins. The output of the MR-DFT.
H_m	Reconstructed harmonic/frequency bin m. $H_m = \vec{H}(m)$
R	M by M matrix. \mathbf{R}^{-1} performs the reconstruction of the real parts of the output
	harmonics of the MR-DFT. Thus this matrix contains the aliasing patterns of real
	parts of the different DFT sets. Part of the "MR-DFT-matrices".
\mathbf{R}_{c}	Z by M matrix, used to construct R . Contains the part of R which is contributed
	by the c-th Sample Set. Z is an integer that differs from case to case.
\mathbf{S}	M by M matrix. S^{-1} performs the reconstruction of the imaginary parts of the
	output bins of the MR-DFT. Thus this matrix contains the aliasing patterns of the
	imaginary parts of the different DFT sets. Part of the "MR-DFT-matrices".
\mathbf{S}_{c}	Z by M matrix, used to construct S . Contains the part of S which is contributed
	by the <i>c</i> -th Sample Set. Z is an integer that differs from case to case.
$\mathbf{A}_{c,r}$	N_c by M matrix containing the aliasing patterns of the real part of the c-th Sample
•	Set. Part of the "Harmonic Matrices".
$\mathbf{A}_{c,i}$	N_c by <i>M</i> matrix containing the anasing patterns of the imaginary part of the <i>c</i> -th Sample Set. Part of the "Harmonic Matrices"
\vec{A} (m, r)	Sample Set. 1 at of the Harmonic Matrices . $P_{over m} of \mathbf{A}$
$A_{c,r}(n, :)$	Now n of $\mathbf{A}_{c,r}$.
$A_{c,i}(n,:)$	NOW n OI $\mathbf{A}_{c,i}$.
$A_{c,r}(n,m)$	Element (n,m) of $\mathbf{A}_{c,r}$.
$A_{c,i}(n,m)$	Element (n,n) of $\mathbf{A}_{c,i}$.

3.2 Terminology and Basic Principles

3.2.1 Discrete Fourier Transform, Bins and Harmonics

The Discrete Fourier Transform (DFT) transforms a finite sequence of evenly spaced values¹ (assumed to be a time discrete signal with samples spaced evenly in time) into a series of complex values representing an equal length set of evenly spaced frequencies of which the time discrete signal is composed of. The DFT of a signal x[n], containing N samples (n = 0,1,.., N-1), is calculated via

$$X_{k} = \frac{1}{N} \sum_{n=0}^{N-1} x[n] \cdot e^{-\frac{2\pi j}{N}kn}$$

$$= \frac{1}{N} \sum_{n=0}^{N-1} x[n] \cdot [\cos(2\pi kn/N) - i \cdot \sin(2\pi kn/N)]$$
(3.1)

x[n] is referred to as the input of the DFT and \vec{X} is referred to as the output of the DFT. k Runs from 0 to N-1 in integer steps. X_k is referred to as the k-th bin. The k-th harmonic ends up in the k-th bin (but also other harmonics will end-up in the k-th bin). See section 2.1 for more information on the DFT.

¹In general the values may be complex but in this work the values are assumed to be real.

3.2.2 Sample Set and its Discrete Fourier Transform

A Sample Set \vec{x}_c is a set of samples obtained by sampling the time continuous signal x(t) using equally spaced sample moments in time. The Sample Set contains the samples $x_{c,0}, x_{c,1}, \dots, x_{c,N_c-1}$. The set identifier c indicates the number of samples in \vec{x}_c per beat period T_b . $c \in \{c0, c1, \dots, c(d-1)\}$. For example, if a Collection contains 2 Sample Sets: \vec{x}_{c0} has 8 samples per beat period and \vec{x}_{c1} has 7 samples per beat period, than c0 = 8 and c1 = 7. Note that a specific case of e.g. Sample Sets, DFTs of Sample Sets, etc can be addressed by ci^2 in which $i \in \{0, 1, \dots, d-1\}$.

The DFT of \vec{x}_c is addressed by \vec{X}_c . The vector \vec{X}_c contains all frequency bins of the DFT of \vec{x}_c , thus \vec{X}_c contains N_c elements. $X_{c,k}$ is the k-th bin of \vec{X}_c .

3.2.3 Collection of Sample Sets

A Collection of Sample Sets contains d different Sample Sets. The different Sample Sets of a Collection of Sample Sets, sample the same (part) of signal x(t) but at a different sample frequency. A Collection of Sample Sets is abbreviated as Collection in the proceeding.

3.2.4 Beat Period

In this work, the beat period T_b is defined as the first moment in time all sets in a Collection have a sample that is sampled at exactly the same moment in time, assuming the first sample of the sets of the Collection are also sampled at the exact same moment in time. This is very similar to the commonly used definition of beat period within RF applications.

3.2.5 Requirements for a Collection of Sample Sets

A Collection should meet the following requirements.

- Every Sample Set of a Collection contains N_c evenly (in time) spaced samples of the same signal. The spacing between two samples of \vec{x}_c is called T_c .
- Within a Collection, there are not two Sample Sets which use the exact same number of samples to span the beat period T_b . This implies that $N_{ci} \neq N_{cj}$ for $c \in \{c0, c1, ..., c(d-1)\}, ci \neq cj$
- All Sample Sets within a Collection span g beat periods. g is at least one, but might be larger. g is an integer: $\{g \in \mathbb{N}_{>0}\}$
- The first ci samples of Sample Set \vec{x}_{ci} , for all ci, span the (same) beat period T_b . This also holds for every second, third, etc, ci samples if g > 1.
- For all Sample Sets within a Collection, the first sample of a beat period is sampled at exactly the same moment in time, i.e. $t_{c,0+a\cdot N_c}$ is equal for all c. Note that $a \in \mathbb{Z}$ when assuming a infinitely long signal x(t). The sample clocks thus have a rising edge at the same time: a common rising edge.

3.2.6 Harmonic Matrices

The Harmonic Matrices $\mathbf{A}_{c,r}$ and $\mathbf{A}_{c,i}$ provide information about the aliasing patterns of the DFT of \vec{x}_c . The matrices $\mathbf{A}_{c,r}$ and $\mathbf{A}_{c,i}$ show which harmonic ends up in which bin via aliasing in \vec{X}_c . $\mathbf{A}_{c,r}$ contains the aliasing patterns of the real parts of the DFT while $\mathbf{A}_{c,i}$ contains this information for the imaginary part of the DFT³. An example of $\mathbf{A}_{c,r}$ is provided in Table 2.1a. Table 2.2a shows an example of $\mathbf{A}_{c,i}$.

²Preferably the index i would have been in subscript. However, ci is used as subscript most of the time and using a subscript within a subscript turned out to be unreadable. Therefore, the i in ci is not written as subscript.

³Note that the Harmonic Matrices $\mathbf{A}_{c,r}$ and $\mathbf{A}_{c,i}$ are not specific for MR-DFTs. The represent the aliasing patterns of a (classical) DFT. MR-DFT differs from the classical DFT since it uses the aliasing patterns of multiple Sample Sets to

The Harmonic Matrix is an N_c -by-M matrix, means that N_c harmonics bins are present (an N_c -point DFT is performed on the Sample Set) and it is tried to reconstruct M harmonics⁴. Every row in the Harmonic Matrix represents a frequency bin and every column represents a harmonic frequency. If a certain bin of the Harmonic Matrix (so a certain row of the Harmonic Matrix) has a non-zero value in a certain column, it means that the harmonic related to that column ends up in that frequency bin (for the DFT of certain Sample Set). The non-zero value indicates how many times the harmonic ends up in that bin.

 $\vec{A}_{c,r}(n,:)$ and $\vec{A}_{c,i}(n,:)$ represent the *n*-th row of $\mathbf{A}_{c,r}$ and $\mathbf{A}_{c,i}$. The first column has index 0 such that $0 \le n < N_c - 1$.

Construction of the Harmonic Matrices

The structure for the real and imaginary parts of a Harmonic Matrix are almost identical except that the aliased harmonics of the imaginary part end-up sign-flipped. Therefore, only the construction of $\mathbf{A}_{c,r}$ is considered.

To improve readability, N_c is abbreviated to N in the continuation of subsection 3.2.6.

The first real part of the zero-th harmonic (DC, R_0) of the positive frequencies and negative frequencies end up in the zeroth bin:

$$\begin{bmatrix} 2 \cdot R_0 & \dots & \dots \\ 0 & \dots & \dots \\ 0 & \dots & \dots \\ \vdots & \ddots & \dots \\ 0 & \dots & \dots \end{bmatrix}$$
(3.2)

The first harmonic R_1 ends up in the first bin and the (N-1)-th bin⁵

$$\begin{bmatrix} 2 \cdot R_0 & 0 & \dots & \dots \\ 0 & R_1 & \dots & \dots \\ 0 & 0 & \dots & \dots \\ \vdots & \vdots & \ddots & \dots \\ 0 & R_1 & \dots & \dots \end{bmatrix}$$
(3.3)

The second harmonic R_2 ends up in the second bin and the (N-2)-th bin

$$\begin{bmatrix} 2 \cdot R_0 & 0 & 0 & \dots & \dots \\ 0 & R_1 & 0 & \dots & \dots \\ 0 & 0 & R_2 & \dots & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & R_2 & \dots & \dots \\ 0 & R_1 & 0 & \dots & \dots \end{bmatrix}$$
(3.4)

This pattern continues and if $m > \frac{N}{2}$ it starts to look like a cross. If m > N - 1, the N-th harmonic (R_N) , which should end up in bin N, is folded back to the bin zero.

improve the spectral bandwidth.

⁴At this moment it is not of interest whether reconstructing M harmonics is possible with the given sets of samples. The maximum value of M is explained in subsection 3.3.1.

 $^{^{5}}$ This has to do with the fact that for a negative frequencies, after a DFT, can be found in bins N/2 to N-1. Since the input signals are real, the real parts of positive and negative frequency are equal.

$2 \cdot R_0$	0	0		0	0	$2 \cdot R_N$	
0	R_1	0		0	R_{N-1}	0	
0	0	R_2		R_{N-2}	0	0	
	÷	÷	÷	:	÷	÷	÷
0	0	R_2		R_{N-2}	0	0	
0	R_1	0		0	R_{N-1}	0	

The next harmonic, R_{N+1} , should end up in bin -1 and N+1. These are aliased to bin N-1 and 1 respectively. So the diagonal lines "switch" direction.

$$\begin{bmatrix} 2 \cdot R_0 & 0 & 0 & \dots & 0 & 0 & 2 \cdot R_N & 0 & \dots \\ 0 & R_1 & 0 & \dots & 0 & R_{N-1} & 0 & R_{N+1} & \dots \\ 0 & 0 & R_2 & \dots & R_{N-2} & 0 & 0 & 0 & \dots \\ \vdots & \vdots \\ 0 & 0 & R_2 & \dots & R_{N-2} & 0 & 0 & 0 & \dots \\ 0 & R_1 & 0 & \dots & 0 & R_{N-1} & 0 & R_{N+1} & \dots \end{bmatrix}$$
(3.6)

Also this pattern continues. When $m > \frac{3 \cdot N}{2}$, then a second cross-like pattern appears and when $m > 2 \cdot N - 1$, a second switch of directions of the diagonal lines appears. This pattern repeats until column M-1 is reached.

In a mathematical expression this results in that:

$$A_r(n,m) = \delta ((n-m)\%N) + \delta ((n+m)\%N)$$
(3.7a)

$$A_{i}(n,m) = \delta \left((n-m)\%N \right) - \delta \left((n+m)\%N \right)$$
(3.7b)

In which $A_r(n,m)$ is the element of \mathbf{A}_r at row n and column m, $A_i(m,n)$ is the element of \mathbf{A}_i at row n and column m, $\delta()$ is the Kronecker delta function and % is the modulo operator. Indexing of rows and columns start at 0, such that $n \in [0, 1, 2, ..., N - 1]$ and $m \in [0, 1, 2, ..., M - 1]$.

Introducing subscript c again, Equation 3.7 becomes

$$A_{c,r}(n,m) = \delta \left((n-m)\%N_c \right) + \delta \left((n+m)\%N_c \right)$$
(3.8a)

$$A_{c,i}(n,m) = \delta \left((n-m)\%N_c \right) - \delta \left((n+m)\%N_c \right)$$
(3.8b)

$$A_{c,i}(0,0) = 1 \tag{3.8c}$$

In which $A_{c,r}(n,m)$ is the element of $\mathbf{A}_{c,r}$ at row n and column m and $A_{c,i}(n,m)$ is the element of $\mathbf{A}_{c,i}$ at row n and column m. In subsection 3.3.3, it turns out that it is useful to introduce S_0 to ensure that the system of linear equations is solvable (S_m is the imaginary part of the m-th harmonic)⁶. Since S_0 is by definition equal to 0, adding it does not change the problem. Therefore $A_{c,i}(0,0)$ is set to 1.

The outline of $\mathbf{A}_{c,i}$ (the imaginary matrix of the Harmonic Matrices) for N_c is even is shown in Equation 3.9. Note that the harmonic and its (folded) complex conjugate cancel out in column $n \cdot g \cdot N_c/2$ (in which n is a positive integer and $1 \le n \cdot g \cdot N_c/2 \le M-1$). For N is odd the outline of $\mathbf{A}_{,i}$ is shown in Equation 3.10.

⁶Another solution is to remove the first row and column of $\mathbf{A}_{c,i}$, resulting in a N_{c} -1 by M-1 matrix for $\mathbf{A}_{c,i}$. From a consistency point of view, it has been chosen to keep $\mathbf{A}_{c,r}$ and $\mathbf{A}_{c,i}$ the same size and add S_0 . From a performance point of view, it makes sense to remove the first row and first column of $\mathbf{A}_{c,i}$.

$\begin{bmatrix} S_0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ $	$egin{array}{c} 0 & S_1 & \ 0 & \ \vdots & \ 0 & \ 0 & \ \vdots & \ 0 & \ -S_1 & \ \end{array}$	$ \begin{array}{c} 0 \\ 0 \\ S_2 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ -S_2 \\ 0 \end{array} $	· · · · · · · · · · · · · · · · · · ·	$\begin{array}{c} 0 \\ 0 \\ 0 \\ \vdots \\ S_{\frac{N_c}{2} - 1} \\ 0 \\ -S_{\frac{N_c}{2} - 1} \\ \vdots \\ 0 \\ 0 \end{array}$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$egin{array}{c} 0 & -S_{N_c-0} & & \\ & \vdots & & \\ 0 & & & \\ 0 & & & \\ 0 & & & \\ S_{N_c-1} & & \\ \end{array}$	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	$egin{array}{c} 0 \\ S_{N_c+1} \\ 0 \\ dots \\ 0 \\ 0 \\ dots \\ 0 \\ -S_{N_c+1} \end{array}$	··· ··· ·· ··· ·· ··	(3.9)
$\begin{bmatrix} S_0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$	$egin{array}{ccc} 0 & S_1 & & \ 0 & & \ \vdots & 0 & & \ 0 & & \ \vdots & 0 & & \ -S_1 & & \ \end{array}$	$egin{array}{c} 0 \\ 0 \\ S_2 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ -S_2 \\ 0 \end{array}$	···· ··· ··· ··· ···	$\begin{array}{c} 0\\ 0\\ 0\\ \vdots\\ -S_{\left\lfloor \frac{N_c}{2} \right\rfloor}\\ -S_{\left\lfloor \frac{N_c}{2} \right\rfloor}\\ \vdots\\ 0\\ 0 \end{array}$	$\begin{matrix} 0\\ 0\\ \vdots\\ -S_{\lceil \frac{N_1}{2}}\\ S_{\lceil \frac{N_2}{2}}\\ \vdots\\ 0\\ 0\end{matrix}$	···· ··· ··· ··· ··· ··· ···	$egin{array}{c} 0 \ -S_{N_c-1} \ 0 \ dots \ 0 \ 0 \ \ddots \ 0 \ S_{N_c-1} \end{array}$	0 0 0 : 0 0 : 0 0 0 -	$egin{array}{c} 0 \\ S_{N_c+1} \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ -S_{N_c+1} \end{array}$	···· ··· ··· ··· ··· ···	(3.10)

For every Sample Set within the Collection there is one set of Harmonic Matrices. So for a Collection of Sample Sets, a total of d sets of Harmonic Matrices should be constructed.

3.2.7 Subdivision of Harmonic Matrices

For easier reference to certain parts of the Harmonic Matrices, these parts have been given their own identifier: top half, centre row and bottom half. Although intuitively it sounds that the top half and bottom half are of equal size and the centre row only exists for Harmonic Matrices with odd N_c , this is both not the case. Be aware of this. For all three definitions, it is assumed that $N_c > 2$. The next three paragraphs define the top half, centre row and bottom part.

Top half

For an N_c -by-*M* Harmonic Matrix, the top half is defined as rows 0 to $\left\lceil \frac{N_c}{2} \right\rceil - 1$.

Centre row

For an N_c -by-M Harmonic Matrix with even N_c , the centre row is defined as the $\frac{N_c}{2}$ -th row. Centre rows are only defined for Harmonic Matrices when N_c is even.

Bottom half

For an N_c -by-M Harmonic Matrix with even N_c , the bottom half is defined as rows $\frac{N_c}{2} + 1$ to N_c -1. For an N_c -by-M matrix when N_c is odd, the bottom half is defined as rows $\left\lceil \frac{N_c}{2} \right\rceil$ to N_c -1.

3.2.8 Collection of Discrete Spectra

If a Collection of Sample Sets is transformed to the frequency domain via a DFT, a Collection of discrete spectra $\vec{X}_{c0}, \vec{X}_{c1}, ..., \vec{X}_{c(d-1)}$ is retrieved.

3.3 MR-DFT in a Mathematical Context

In the MR-DFT, two systems of linear equations are constructed and solved to reconstruct the output harmonics. The system of linear equations for reconstruction of the real part of the output harmonics is shown in Equation 3.11 and the system for reconstruction of the imaginary part is shown in Equation 3.12.

$$\mathbf{R} \cdot Re(\vec{H}) = Re(\vec{Y}) = \vec{Y}_r$$

$$Re(\vec{H}) = \mathbf{R}^{-1} \cdot \vec{Y}_r$$
(3.11)

$$\mathbf{S} \cdot Im(\vec{H}) = Im(\vec{Y}) = \vec{Y}_i$$

$$Im(\vec{H}) = \mathbf{S}^{-1} \cdot \vec{Y}_i$$
(3.12)

which can be combined in one equation to

$$\vec{H} = \mathbf{R}^{-1} \cdot \vec{Y}_r \cdot + j \cdot \mathbf{S}^{-1} \cdot \vec{Y}_i \tag{3.13}$$

In Equation 3.11 and Equation 3.12, **R** and **S** are the so called MR-DFT-matrices. The rows of **R** and **S** consist of a selection of the rows of $\mathbf{A}_{ci,r}$ and $\mathbf{A}_{ci,i}$ and \vec{Y} contains the corresponding DFT bins (with \vec{Y}_r and \vec{Y}_i being the real and imaginary part of \vec{Y}^7). The structure of **R**, **S**, and \vec{Y} is explained in subsection 3.3.3. \vec{H} contains the reconstructed harmonics. The structure of \vec{H} is:

$$\vec{H} = \begin{bmatrix} H_0 \\ H_1 \\ \vdots \\ H_{M-1} \end{bmatrix}$$
(3.14)

In which H_m are the reconstructed harmonics/frequency bins.

The observant reader probably asks itself what happened to half of the signal power. The energy of a frequency component is divided over its positive frequency bin and negative frequency bin. In case of a real input signal, the energy is divided 50-50 over the positive and negative frequency bin. In the case presented, the negative frequency bins are the once found in the bottom-half of the Harmonic Matrices. As explained in footnote 8 on 27, the bottom half of the Harmonic Matrices is equal to the top half (except for the complex conjugate) and may be omitted. The DFT bins of the Collection of Discrete Spectra accordingly are also omitted, so the energy in the negative frequency bins is lost (ignored). Therefore, an extra factor of 2 should be added somewhere in the mathematical framework. This may e.g. via dividing the Harmonic Matrices by 2 or adding a factor of 2 in Equation 3.13. The last option has been chosen, resulting

$$\vec{H} = 2 \cdot \mathbf{R}^{-1} \cdot \vec{Y}_r \cdot + j \cdot 2 \cdot \mathbf{S}^{-1} \cdot \vec{Y}_i \tag{3.15}$$

⁷It turns out that this is not strictly true since \vec{Y}_r and \vec{Y}_i might use different elements of \vec{Y} . This is covered in more detail in subsection 3.3.3. For now assume that $\vec{Y}_r = Re(\vec{Y})$ and $\vec{Y}_i = Im(\vec{Y})$.

3.3.1 Number of Reconstructed Frequency Bins

An algorithm to determine the maximum number of reconstructed harmonics is empirically determined. This is done by having a close look at the Harmonic Matrices as well keeping in mind that a system of linear equations is constructed and solved. Using these two properties, the following observations can be done.

- In essence the MR-DFT is solving a system of linear equations. Every sample within a Collection makes that one equation of the system is solvable. Since the reconstructed harmonic consist out of a real and imaginary part and complex conjugate operation is not a linear operation, for every reconstructed harmonic two linear equations are necessary. This makes that a DFT consisting out of x samples can be used to reconstruct at most the ceil(x/2) harmonics. The ceil-operation (rounding up) originates from the fact the DC component (H_0) only has a real part and therefore only one equation is used to reconstruct it.
- All Collections have at least one common sample moment per beat period. This is a sample moment that is shared between all Sample Sets withing the Collection. In maths, this looks like $a \cdot T_{ck} = b \cdot T_{cl}$ in which $a \in 0, 1, ..., ck-1, b \in 0, 1, ..., cl-1$ and $ck \neq cl$. For all Collections this is the first sample of all Sample Sets, but there are Collections which have more common sample moments. For example, the Collection consisting out of the two Sample Sets with c0 = 9 and c1 = 6, has 3 common sample moments per beat period: $t_{9,0} = t_{6,0}, t_{9,3} = t_{6,2}$, and $t_{9,6} = t_{6,4}$. Since these common samples should result in the same sampled value, they don't add any extra information. Therefore, every common sample moment lowers the amount of solvable linear equations.

The total decrease depends on the multiplicity of the common sample moment. For example: given the Collection consisting of the Sample Sets with c0 = 9, c1 = 6, and c2 = 1, there are 3 common sample moments. These are: $t_{9,0}$ with multiplicity 3 (common sample moment shared between 3 Sample Sets), $t_{9,3}$ with multiplicity 2, and $t_{9,6}$ with multiplicity 2. A common sample moment decreases the amount of solvable equations with its multiplicity - 1. In the example, the amount of solvable linear equations is decreased by 4 (or loosely spoken the amount of reconstructable harmonics is lowered by 2). In general, if a sample moment is shared by b Sample Sets, the amount of solvable linear equations is lowered by b - 1.

• The MR-DFT of a Collection containing Sample Sets c0, c1, ..., c(d-1) and g > 1, can also be approached as a the MR-DFT of a Collection containing the Sample Sets $N_{c0}, N_{c1}, ..., N_{c(d-1)}$ $(= c0 \cdot g, c1 \cdot g, ..., c(d-1) \cdot g)$. Note that this introduces at least g - 1 extra common sample moment.

Given these observations, the following equation is derived:

$$M = \left\lceil \frac{\sum_{c} (c \cdot g) - Q}{2} \right\rceil \text{ in which } c \in \{c0, c1, \dots, c(d-1)\}$$
(3.16)

In which Q is the sum over the multiplicity - 1 of all sample moments in the Collection. This can be written as:

$$Q = \sum_{a} \left(\# t_{uniq,a} - 1 \right) \cdot g \quad \text{in which } a \text{ runs over all elements of } \vec{t_{uniq}}$$
(3.17)

In which \vec{t} are all (ideal) sample moments of the Sample Sets within a Collection, $\vec{t_{uniq}}$ are the unique elements of \vec{t} , and $\#t_{uniq,a}$ is the number of times the a-th element of $\vec{t_{uniq}}$ occurs in \vec{t} (the multiplicity of $t_{uniq,a}$). Appendix A provides two MATLAB algorithms to calculate Q.

Equation 3.16 is validated for all possible combinations of 2 up to 4 Sample Sets with the largest Sample Set containing up to 20 samples and for 1 up to 10 beat periods (a total of 7350 cases). In all these cases, the number of reconstructable harmonics has never been predicted wrong. Therefore, it is assumed that this equation is correct.

3.3.2 Frequency Resolution of MR-DFT

The frequency resolution of the output of the MR-DFT (the bandwidth per output bin) is equal to the frequency resolution of the different Sample Sets. This number can be calculated via

$$\Delta f = \frac{f_{s,c}}{N_c} \tag{3.18}$$

and holds for every value of c (since the frequency resolution is equal for all Sample Sets within a Collection).

3.3.3 MR-DFT-Matrices and \vec{Y}

The matrices \mathbf{R}^{-1} and \mathbf{S}^{-1} can be calculated by applying the Moore-Penrose pseudo-inversion to a concatenated version of all Harmonic Matrices (Equation 3.19a and Equation 3.19b). This solution does work but is not a very elegant. This section provides an algorithm to end-up with invertible MR-DFT-matrices.

The Harmonic Matrices should be concatenated in a column-like fashion 8 as shown in Equation 3.20a and Equation 3.20b.

$$\mathbf{R} = \begin{bmatrix} \mathbf{A}_{c0,r} \\ \mathbf{A}_{c1,r} \\ \vdots \\ \mathbf{A}_{cd-1,r} \end{bmatrix}$$
(3.20a)
$$\mathbf{S} = \begin{bmatrix} \mathbf{A}_{c0,i} \\ \mathbf{A}_{c1,i} \\ \vdots \\ \mathbf{A}_{cd-1,i} \end{bmatrix}$$
(3.20b)

In the same manner \vec{Y} can be constructed out of the DFTs of the different Sample Sets:

$$\vec{Y} = \begin{bmatrix} \vec{X}_{c0} \\ \vec{X}_{c1} \\ \vdots \\ \vec{X}_{cd-1} \end{bmatrix}$$
(3.21)

The resulting MR-DFT-matrices are not yet square thus some rows are linearly dependent. To find which rows should be left out, bring the transpose of \mathbf{R} to row reduced echelon form. Find the indexes of the

$$\mathbf{R}_{c} = \begin{bmatrix} A_{c,r}(0,:) \\ \vec{A}_{c,r}(1,:) \\ \vdots \\ \vec{A}_{c,r}(\lceil \frac{c}{2} \rceil - 1,:) \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} \mathbf{R}_{c0} \\ \mathbf{R}_{c1} \\ \vdots \\ \mathbf{R}_{c(d-1)} \end{bmatrix}$$
(3.19a)
$$\mathbf{S}_{c} = \begin{bmatrix} \vec{A}_{c,i}(0,:) \\ \vec{A}_{c,i}(1,:) \\ \vdots \end{bmatrix} \quad \mathbf{S} = \begin{bmatrix} \mathbf{S}_{c0} \\ \mathbf{S}_{c1} \\ \vdots \end{bmatrix}$$
(3.19b)

$$\begin{bmatrix} \vdots \\ \vec{A}_{c,i}(\lceil \frac{c}{2} \rceil - 1, :) \end{bmatrix} \begin{bmatrix} \vdots \\ \mathbf{S}_{c(d-1)} \end{bmatrix}$$

⁸It can be observed that the bottom half of the Harmonic Matrices is equal to the top half. From a performance point of view, removing these parts of the matrices before concatenating, is desirable. If present, the centre row should not be removed. This results in the relations for the MR-DFT-matrices as in Equation 3.19. Don't forget to select the right elements of the different \vec{X}_c to construct \vec{Y}_c and \vec{Y} in the same manner.

columns of all pivot elements⁹ and select the rows of **R** with same index as the pivot columns. Remove all other rows. Also, select the same elements of $Re(\vec{Y})$ and leave out all other elements to construct \vec{Y}_r . Do the same for **S**, $Im(\vec{Y})$, and \vec{Y}_i . Note that other rows can be selected for **R** than for **S**.

This algorithm is validated by performing every possible MR-DFT with at least 2 and up to 4 Sample Sets, a maximum Sample Set length of 20 samples per beat period, and up to 10 beat periods of data. This results in a total of 7350 simulations that has been performed of which no one has failed. Therefore, it is assumed that this method is correct.

This procedure is clarified by an example. The matrix **R** and \vec{Y}_r will be constructed for a Collection containing 2 Sample Sets. The two sets have 4 (c0) and 3 (c1) samples per beat period and a total of 3 beat periods is sampled (g = 3). First calculate M via Equation 3.16. This results in M = 9, so the output contains 9 bins: DC and 8 harmonics. The next step is to construct the Harmonic Matrices $\mathbf{A}_{4,r}$ and $\mathbf{A}_{3,r}$. This results in:

$$\mathbf{A}_{4,r} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{A}_{3,r} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$(3.22)$$

To save space, the procedure for constructing ${\bf R}$ in footnote 8 on page 27 and Equation 3.19 is used. This results in:

But this matrix has a rank of 9 while it has 10 rows, so there is one rows which is linearly dependent on the other ones. By transposing and bringing it to row reduced echelon form, it turns out that that row 9 $(\vec{A}_{3,r}(4,:))$, indexing of rows start at 0) has no pivot element and thus that row 9 should be removed from **R**. This results in the real part of the MR-DFT-matrices:

$$\mathbf{R} = \begin{vmatrix} A_{4,r}(0,:) \\ \vec{A}_{4,r}(1,:) \\ \vec{A}_{4,r}(2,:) \\ \vec{A}_{4,r}(3,:) \\ \vec{A}_{4,r}(4,:) \\ \vec{A}_{4,r}(5,:) \\ \vec{A}_{3,r}(1,:) \\ \vec{A}_{3,r}(2,:) \\ \vec{A}_{3,r}(3,:) \end{vmatrix} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$
(3.24)

⁹See for example https://en.wikipedia.org/wiki/Pivot_element for more information on pivot elements.

And the \vec{Y}_r is:

$$\vec{Y}_{r} = Re \begin{pmatrix} \begin{bmatrix} X_{4,0} \\ X_{4,1} \\ X_{4,2} \\ X_{4,3} \\ X_{4,4} \\ X_{4,5} \\ X_{3,1} \\ X_{3,2} \\ X_{3,3} \end{bmatrix} \end{pmatrix}$$
(3.25)

The same procedure also hold for constructing \mathbf{S} .

Now all matrices and vectors for the MR-DFT are known, the output of the MR-DFT (\vec{H}) can be calculated by performing Equation 3.15.

3.4 Nyquist-Shannon Criterion and the MR-DFT

This section describes the Nyquist criterion, the consequences of it for the DFT, and if it holds for the MR-DFT. The Nyquist criterion, although stated by Shannon in [13], states that: "If a function f(t) contains no frequencies higher than BW cycles per second, it is completely determined by giving its ordinates at a series of points spaced 1/(2BW) seconds apart." Shannon states that the sample frequency (f_s) should be $f_s \ge 2BW$ to be able to recover the frequency information in BW via a DFT. More modern version of the Nyquist criterion mostly state that the series of points should be spaced strictly less than 1/(2BW) seconds apart [14]. This implies that $f_s > 2BW$. This more modern version of the Nyquist-Shannon criterion is going to be used in this work. The implications for a DFT are that if a DFT is performed on a signal that contains frequency information equal or higher the Nyquist frequency f_{nyq} $(f_{nyq} = f_s/2)$, this ends-up in the bandwidth up to BW via folding/aliasing. Via the Poisson Summation Formula, aliasing can be expressed as:

$$G(f) = \sum_{k=-\infty}^{\infty} F(f - k \cdot f_s) = \sum_{n=-\infty}^{\infty} T_s \cdot f(n \cdot T) \cdot e^{-j2\pi \cdot n \cdot T_s \cdot f}$$
(3.26)

in which G(f) is the DFT of a sampled version of f(t), F(f) is the (continuous) Fourier transform of f(t), f_s the sample frequency, and $T_s = 1/f_s$.

This implies that any information in the band from BW to 2BW (assuming the sample frequency is 2BW), is folded to the band of BW to 0, band from 2BW to 3BW is aliased back to the band of 0 to BW, and so on¹⁰. This reveals an interesting property of the DFT. As long as only information is present in the band of $n \cdot BW$ to $(n + 1) \cdot BW$, this information can be retrieved without any problem due to folding to the band of 0 to BW. Note that it should be known upfront in which band information is present. Also the information should be confined to a band between $n \cdot BW$ and $(n+1) \cdot BW$ in which n is a positive integer. If the information is e.g. in the band between 2BW and $(n+1) \cdot BW$, both are aliased to the same band between 0 and BW/2 and corrupt each other. It is therefore of utmost importance that n is an integer. So the Nyquist-Shannon criterion in combination with Poissons Summation Formula result in that information with a bandwidth up to 1BW can be retrieved as long as this information is available within a band between $n \cdot BW$ and $(n+1) \cdot BW$ and the remainder of the spectrum is completely empty¹¹.

The question is if this also holds of the MR-DFT. The short answer is no. The long answer is probably; the presented Harmonic Matrices are tailored to reconstruct the frequencies from 0 Hz to $(M-1) \cdot \Delta f$ but it might be able to create Harmonic Matrices with which higher frequency bands can be reconstructed.

¹⁰Strictly spoken, this is not correct since folding also does some complex conjugation and when complex input signals are used things get even more complex. For the essence of the MR-DFT in the proposed applications (having a real input signal), the foregoing explanation is sufficient. ¹¹For exact phase recovery it should be known in which bandwidth between $n \cdot BW$ and $(n + 1) \cdot BW$ the information

¹¹For exact phase recovery it should be known in which bandwidth between $n \cdot BW$ and $(n+1) \cdot BW$ the information is present

The Harmonic Matrices and MR-DFT-matrices provided in this chapter and thesis only works for reconstruct the frequency information within the bandwidth from 0 Hz to $M \cdot \Delta f$ Hz, in which Δf is the bin size is given by Equation 3.18. Therefore, they cannot be used to reconstruct any other frequency information except for the band from 0 Hz to $M \cdot \Delta f$ Hz¹². This is confirmed by simulations.

However, it is expected that it is possible to reconstruct the frequency information out of any band from $a \cdot M \cdot \Delta f$ Hz to $(a+1) \cdot M \cdot \Delta f$ Hz, in which a is a non-negative integer, as long as the remainder of the spectrum is empty. This is expected to be possible since it is possible to construct Harmonic Matrices that represent the folding patterns for any band from $a \cdot M \cdot \Delta f$ Hz to $(a+1) \cdot M \cdot \Delta f$ Hz.

It might actually be possible to reconstruct any band $a \cdot \Delta f$ Hz to $b \cdot \Delta f$ Hz as long as b - a < Mor b - a < M - 1 and the all bands outside of the range from $a \cdot \Delta f$ Hz up to $b \cdot \Delta f$ Hz are empty. This heavily depends on how the harmonic information is folded back per Sample Set and if enough independent linear equations are present to solve the system of linear equations. The verification of these ideas are kept out of the scope of this research and are added to the recommendations.

¹²Except for the hypothetical special case in which the folding pattern of the harmonics within the band from $a \cdot M \cdot \Delta f$ Hz up to $(a + 1) \cdot M \cdot \Delta f$ Hz, for $a \in \mathbb{N}$, are exactly equal for all Sample Sets. It is investigated if this special case exists.

Part III

Digital Design

Chapter 4: Measuring the MR-DFT: Goal and Conceptual Design

Part III describe the development of an MR-DFT measurement instrument (abbreviated to Instrument). Chapter 4 describes the goal of the Instrument, requirements, the selected hardware, and the conceptual design on block level. Building on this chapter, Chapter 5 presents the in-depth design of the various system blocks as well as problems faced during the design process. Subsequent chapters in Part IV, describe the measurement setup as well as the measurement results.

This chapter is divided in 4 sections. Section 4.1 introduces the goal of measurement instrument. The next section describes the requirements the Instrument should meet as well as requirements on the measurement setup. Section 4.3 presents the selected hardware and why this hardware is used. The last section, section 4.4, elaborates on the digital design of the Instrument and.

4.1 Measurement Goal and Plan

The designed Instrument implements a proof-of-concept measurement instrument for measuring Sample Sets in combination with the MR-DFT. A diagram of the system is shown in Figure 4.1. For simplicity, it is chosen that the Instrument performs an MR-DFT on two Sample Sets. To prevent timing and phase issues, the necessary measurements are chosen to be performed parallel in time (instead of sequential in time). This implies that the Instrument must be able to sample a signal at least at two different sample rates. To handle high data rates from the Analog-to-Digital Converters (ADCs), the Instrument is based on SoC platform containing both an FPGA and ARM processor (so called HPS); the FPGA handles the instantaneous high data volumes and the ARM processor is used for data movement and long term storage. The sampled signal is generated by an Arbitrary Waveform Generator (AWG). This provides the ability to perform e.g. measurements on QAM constellations¹. In the proof-of-concept demonstration, the processing of the sampled data is performed offline, on a MATLAB script on a regular computer.



Figure 4.1: High-level diagram of the measurement setup.

4.2 Requirements

The AWG must have least 10 bits resolution over full scale range. The output signal must approach a continuous signal: sample rate must be at least 100 times higher than the highest recoverable frequency.

The ADCs need sufficient resolution. If OFDM like signals over 200-250 channels are measured, give or take, around 7 to 8 bits of precision are lost per channel. If QAM16 constellations are send (which use 3 different signal amplitudes), give or take at least 10 bits of precision is necessary. All ADCs must have an individual sample clock. Sample rates are at least 10 MHz.

The FPGA must contain PLLs (to generate multiple clock signals) and three clock output pins (for two sample clocks and a synchronisation clock). Furthermore, the FPGA must have at least two clock

¹Quadrature Amplitude Modulation (QAM) is a industry wide used method for modulating data into an RF carrier.

networks for the two sample clocks and must be able to run at sample clock frequencies. The FPGA and HPS must be connected by an interconnect or bus that allows to exchange data.

The HPS must be able to run a high level Operating System (like Linux) and must support functionality to send data to another PC for offline processing (preferable Ethernet support).

Given the time limitations of a master thesis, there was no time available to develop custom hardware.

4.3 Hardware Selection and Consequences

This section describes the selected hardware, why this hardware is selected, and what the consequences of the selected hardware are.

4.3.1 FPGA and ADCs

A Terasic SoCKit development board was selected to build the Instrument around. This board provides an Altera Cyclone V Device (5CSXFC6D6F31C6N) of Intel^{2 3}. This development board is chosen since this board was available within the ICD group and it fitted the requirements of section 4.2.

The Altera Cyclone V is what Altera (now Intel) calls a "SoC". In FPGA jargon this is a chip with FPGA fabric and a general purpose processor. The FPGA fabric and processor are connected by a bus that allows to exchange data between the two systems. The digital processor (called Hard Processor System, or in short HPS) is based on the ARM architecture and is capable of running a high-level Operating System like Linux.

For this development board, multiple expansion boards (called "daughter board" by Terasic) are available. Terasic offers the High Speed AD/DA Card (ADA), containing the Analog Devices AD9248 ADC IC. This is a two channel, 14 bits, 65 MS/s ADC, having individual sample clocks for the two channels. The input range of the ADCs is -1 V to +1 V. The ADCs have an SINAD between 71 and 72 dB (ENOB between 11.5 and 11.7). Given it fits the requirements and it is the only available of-the-shelf ADC board for this development board, this board is selected.

Consequences of Terasic High Speed AD/DA Card

The consequence of using these ADCs is that the maximum sample frequency is 65 MS/s and two data channels are available. Given these limitations, it is chosen to perform an MR-DFT with 8 and 7 samples per beat period. \vec{x}_8 is sampled at 64 Ms/s and \vec{x}_7 at 56 Ms/s. Given those sample rates, the ADCs generate a combined data rate of at least 210 MB/s.

Arbitrarily, it is chosen to sample a total of 32 consecutive beat periods. This makes that \vec{x}_8 contains 256 samples ($N_{c8} = 256$) and \vec{x}_7 contains 224 samples ($N_{c7} = 224$). The consequence of g = 32 is that the DFT bins are 250 kHz wide (see Equation 3.18) and the highest recoverable output bin/harmonic is at 55.75 MHz (see Equation 3.16).

4.3.2 Arbitrary Waveform Generator

As AWG, the Keysight M8190A is chosen. This high-end AWG meets the requirements (also the requirements in combinations with the ADCs) and is available at the lab of the Integrated Circuit Design group of the University of Twente. The Keysight M8190A has a resolution of 14 bits over 700 mVpp full scale output range (for 50 Ω loads) at up to 8 GS/s.

 $^{^{2}}$ For documentation regarding this device, see the Cyclone V Handbook Volume 1 [15] and 2 [16].

 $^{^{3}}$ The used product is en Altera product. Since then, Intel has acquired Altera. Altera and Intel are used intertwined in this report.
4.4 Conceptual Design

Figure 4.1 shows a high level overview of the demonstration setup. In this section, a more detailed overview of the ADC-FPGA-HPS data chain is given. The challenge in chain is that the ADCs produce more than 200 MB/s while the bus between the FPGA and HPS cannot handle those data rates. Also, the storage capacity of the HPS is limit (flash SD-card, capacity will be in the order of 8 GB or 16 GB). The challenge is how to reduce the high volume data streams of the ADCs such that it can be handled and stored by the HPS.

Figure 4.2 shows a block diagram of the conceptual design of this data chain. The challenge have been tackled in two ways. The data streams of the ADCs first encounter a "data valve" and than some buffers are present. The "data valve" ensures that all data related to one measurement (256 and 224 data points) are stored in the buffers such that the HPS has sufficient time to transfer all data of a measurement to its file system.

The "data valve", called ADC to FIFO (A2F), forwards only data under certain conditions. The A2F also provides in some synchronisation between measurements and the AWG. The A2F is implemented within the FPGA fabric.

The buffers, implemented by First In First Out (FIFO) memories, make sure that the HPS has enough time to copy all data related to a measurement to its file system. A FIFO memory is used to ensure the data is buffered in chronological order. Since the two ADCs generate separate data streams, two FIFOs are used (one for every data stream). The FIFOs are implemented in the FPGA fabric.

Now the data of one measurement is in the file system of the HPS, it is accessible and can be moved (e.g. via Ethernet) for off-line processing.

The different blocks (ADCs, A2F, FIFOs, and HPS) should be connected. Both ADCs have a parallel output interface which runs at the sample clock rate. This parallel bus is connected to the A2F.

The A2F moves data to the FIFOs at high data rates and needs some status information from the FIFOs (e.g. if the FIFOs are ready to receive data). So it should be possible to move data in two directions between the A2F and the FIFOs. Altera provides standard busses that can be used within the FPGA Fabric. Due to the low complexity, ease of use, and streaming data is present, the Avalon Streaming Interface⁴ is used to stream the data between the A2F and the FIFOs. Note that the Avalon Streaming Interface is a uni-directional bus, so two busses (one for each direction) are present between the A2F and a FIFO (hence the two arrows in Figure 4.2).

The HPS and the FIFOs also exchange data, but in one direction. The HPS needs some status information about the FIFOs (e.g. if the FIFOs are ready to be read out) and the HPS should move data from the FIFO to its file system. Altera provides some busses to connect the HPS to the FPGA fabric. Due to the relatively low complexity and ease of use, the Avalon Memory Mapped Interface has been selected for this task. Note that the FIFO data output port is mapped to another memory address than status output port (hence the two arrows in Figure 4.2).



Figure 4.2: A schematic overview of the system design of the digital system.

Since multiple different clock frequencies are used, clock domain crossing is imminent. Within the FPGA fabric, this is not a problem since the hardware can be tightly controlled. However, the clock domain crossing between the FIFO and HPS might cause problems. To prevent any problems due to clock domain crossing at this clock domain edge, the FIFOs is first completely filled with data before the HPS starts reading. Since the FIFOs are 224 and 256 entries deep, there is more than enough time for the output of the flipflops to stabilise before reading occurs.

⁴For the Avalon Interface specifications, see [17].

Chapter 5: System Components and Design Choices

This chapter provides a detailed description of the different components of the digital part of the Instrument. It also covers why certain design choices are made. The implementation of the ADC to FIFO block are considered in section 5.2. Section 5.3 explains the implementation of the FIFO buffers. The next short section covers the Operating System used. The last section addresses some timing issues. But first the clock generation and the different clock domains within the Instrument are described. For a high level design overview of the Instruments design, see Chapter 4.

5.1 Clock Generations and Clock Domains

The Instrument, as described in Chapter 4 needs 3 different clock domains:

- CLK64: 64 MHz clock for sampling data and handling this data stream.
- CLK56: 56 MHz clock for sampling data and handling this data stream.
- CLKHPS: A clock source to clock the busses adjacent to the HPS. These busses have a clock speed of 64 MHz.

In principle the same clock that provides CLK64 or CLK56 could also be used to provide the clock for CLKHPS. In practise this is also the case. Since it is unknown what the internals of the clock network in the HPS are and how they affect the clock signal (mainly phase), the CLKHPS is seen as a separate clock domain.

5.1.1 Clock Generation

The Intel Cyclone V Device family provides, among other reference clocks, a couple of 50 MHz reference clocks and a PLL IP¹ (which provides both integer and fractional functionality). The 50 MHz clock source and PLL IP are used to generate the clock signals for the clock domains CLK64, CLK56, and CLKHPS. The PLL IP is not able to synthesise the two required clock signals out of the 50 MHz reference clock. A PLL IP is used to downscale the 50 MHz clock to 16 MHz. A second PLL-IP is used to upscale this clock to both 56 MHz and 64 MHz². Since both clock signals originate from the same PLL-IP, the signals are "in phase" in the sense that they have a common rising edge. This has been validated by inspecting the clock signals with an oscilloscope (Agilent DSAZ204A).

The same PLL-IP that generates the 64 MHz and 56 MHz clock signals, also generates an 8 MHz clock signal/beat period. This 8 MHz clock shares a common rising edge with 64 MHz and 56 MHz sample clocks; it defines the beat period of the two sample clocks. This clock is referred to as Sync Clock and is used for synchronisation purposes (see subsection 6.1.1).

5.2 ADC to FIFO

The first digital block encountered in the FPGA is the ADC to FIFO (A2F) block. Figure 5.1 shows a schematic overview of this block. This block takes the parallel output of the ADCs and forwards it to the FIFO (via the "Data to FIFO" sub-blocks). To ensure proper phase measurement, some synchronisation is necessary. This is performed by the "Clock and Data Sync" sub-block. The "Clock

 $^{^1\}mathrm{See}$ [18] for the documentation on the Altera PLL IP.

²Within the Quartus Platform Designer, a PLL IP implementation can have up to 4 outputs having different output frequencies. Normally a single PLL only outputs one clock signal. Since the Platform Designer is a high-level drag-and-drop-like tool, it is unknown how the multiple outputs are realised. See [18] for more information.



Figure 5.1: A schematic view of the implementation of the ADC to FIFO digital block.

and Data Sync" sub-block controls the "Data to FIFO" sub-blocks such that data is only forwarded when certain synchronisation conditions are met. The following sections dive into the details of the function of the A2F block. But first, the connectivity and clock domains are considered.

5.2.1 Connectivity to A2F

The A2F is connected to the rest of the system by six busses:

- Two from the ADCs to the A2F (one per ADC). These are parallel busses with a width of 15 bits: 14 for data and 1 out-of-range bit.
- Two data busses from the A2F to the FIFOs (one per FIFO). These streaming busses stream data from the A2F to the FIFOs. These 32 bits wide busses are implemented by Avalon Streaming Interface.
- Two data busses from the FIFOs to the A2F (one per FIFO). These streaming busses provide access to the Control and Status Register (CSR) of the FIFOs. See subsection 5.3.4 for more information. These 32 bits wide busses are implemented by Avalon Streaming Interface.

5.2.2 Clock Domains and the A2F

The A2F operates within the CLK56 and CLK64 clock domains since the two AD channels operates at two different clock speeds (namely 64 Ms/s and 56 Ms/s). In Figure 5.1, the CLK64 is depicted in red and the CLK56 is depicted in blue. The A2F is also connected to the FPGA wide asynchronous reset. If the reset is active, all registers and states machines are reset to their initial value.

5.2.3 Clock and Data Synchronisation

The MR-DFT algorithm requires that the first sample of all sample sets is sampled at the exact same time (the clocks responsible for the sampling should have a common rising edge). The "Clock and Data Sync" (CDS) sub-block of Figure 5.1 controls the "Data to FIFO" sub-blocks and makes sure only data is sent to the FIFOs when certain synchronisation conditions are met. This is necessary to ensure proper measurement of the phase. What the synchronisation conditions are, is explained in subsection 5.2.3 and Figure 5.2.3.

Clock Synchronisation

The first sample of all sample sets should be sampled at the exact same time. There are two possible solutions to comply with this requirement: Either start sampling at a common rising edge of both clocks or include extra data in the output to the HPS which provides information about when common rising edges of both clocks are in the data set. It has been chosen to implement the first solution³: Starting the data sets at a common rising edge of both sample clocks.

The Clock and Data Sync sub-block performs the synchronisation of the two data streams (see Figure 5.1). This sub-block uses the 56 MHz clock to sample the 64 MHz clock. The working principle is shown in Figure 5.2. At the common rising edge, it is unknown what value is sampled. The next 3 rising edges of the 56 MHz clock sample the high value (H) of the 64 MHz clock. Subsequently, the 56 MHz clock samples three low values. The next rising edge of the 56 MHz clock is a common rising edge.

The implementation of this digital block is simple. If it detects three sequential low sampled values, it sends a Clock Sync Pulse to the two Data to FIFO sub-blocks marking the next upcoming clock edge as a common rising edge⁴.

As seen in Figure 5.2, close to the 4th (+/- 53 ns) and 5th (+/- 70 ns) rising edge of the 56 MHz clock, the 64 MHz clock has a clock edge (in this case a falling edge). If the edges of the 64 MHz clock are not sharp, this could lead to wrong sampling results (and thus in non-deterministic behaviour). The difference between these clock edges is approximately 1.12 ns. Due to the nature (28 nm) and max clock speeds of this FPGA (up to 550 MHz [19]), it is expected that this is sufficient time between the clock edges for accurate sampling. Although not formally validated, it is investigated if any non-deterministic behaviour could be observed. In two measurements of one hour (one at room temperature and one at $60+^{\circ}C^{5}$) no non-deterministic effects is observed. Therefore, it is assumed that this clock synchronisation technique is usable for our use case.

Besides the clock signals, the Clock Sync sub-block also uses the Control Status Register (CSR) information of both FIFOs as input values. The sub-block checks if the FIFOs are ready to receive data. If not, the Clock Sync Pulse is not send.

Data Synchronisation

Another problem arising is how to ensure the phase of the measured signal. When e.g. a QAM constellation is measured, measuring the phase causes difficulties. To ensure proper phase measurement of all symbols, the measurements of all symbols in a constellation should start at a fixed phase shift (preferable 0) relative to the start of the symbol waveform. If not, the amplitude of the symbols within the QAM constellation can be determined, but the phase cannot. Within an integrated measurement setup this is not a problem⁶ but in the Instrument this is a problem. So an extra data synchronisation step is necessary.

The addition of a reference tone to the input is not sufficient for absolute phase measurements. The

³Note that this was not a real choice during the implementation phase. At that moment the second possibility was not yet though of. Because it is expected that both solutions perform equally well, there is no need to re-evaluate this choice. ⁴Note that for the 64 MHz clock, not the first rising edge after the Clock Sync Pulse but the second one is the common

rising edge. Hence, the clock sync pulse to this sub-block is delayed by 1 clock cycle. ⁵This test has been performed by aiming a hot air gun at 125°C at the package of the FPGA, while loading both cores of the HPS by calculating some floating point multiplication.

⁶E.g. on chip when signal generation and measurement are performed by the same circuits as suggested in Figure 2.3.



64 MHz(red) and 56 MHz(blue) clock

Figure 5.2: Clock synchronisation process by sampling the 64 MHz clock by the 56 MHz clock. Top: The two sample clocks. Bottom: The sampled values when the 64 MHz clock is sampled by the 56 MHz clock.



Time

Figure 5.3: Absolute and relative phase difference between symbol (blue) and reference tone (red). While at the start at the symbol the absolute phase difference is 0, at the start of period 2 and 3, the relative difference between the symbol and reference tone is not zero.

3	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Timestamp									V	0					S	an	npl	.e c	lat	a											

Figure 5.4: The structure of the output of the D2F sub-block (and thereby the data output structure of the A2F block). V = valid bit, O = Out of Range bit.

phase difference between the reference tone and the symbol is measurable as long as the symbol is not an integer multiple of the reference tone the local phase difference between the reference tone and the symbol differs from period to period (of the reference tone). This is depicted in Figure 5.3. The absolute phase difference between the reference tone (red) and symbol (blue) is zero (see "Start Symbol"). When a measurement starts in period 2 or 3 of the reference tone, it seems like a valid (non zero!) phase of the symbol is measured. Since the symbol does not start at the beginning of the that period of the reference tone, the phase measurement is invalid. Since the measurement does not provide when the symbol starts, it is impossible to tell if the measured phase difference is the absolute phase difference between reference tone and symbol.

To tackle this problem, an extra sync signal is added to the Clock and Data Sync sub-block to mark the start of a symbol. This signal is denoted as "Synchronisation" in Figure 5.1. At the beginning of every symbol, the AWG sends a synchronisation pulse to the FPGA board. Using this pulse, the Instrument ensures that it first sample of the Sample Sets is always within the first period of the reference signal such that absolute phase measurements of symbols are possible. The implementation of this synchronisation in the measurement setup is discusses in more detail in subsection 6.1.2.

5.2.4 Counter and Timestamping

The next sub-block in the A2F is called "Counter". This sub-block is responsible for providing "timestamps". It generates a timestamp by counting the 64 MHz clock. This timestamp is sent to both "Data to FIFO" sub-blocks, which combine it with the sampled data. These timestamps are added because it was expected that the designed Instrument might experience timing issues. Having this timestamps makes sure that in the post processing data integrity can be checked. For more info about these timing issues, see section 5.5.

The counter is a 16 bit gray code counter⁷ which counts clock pulses of the 64 MHz clock and overflows to zero. In this way every sample gets a unique timestamp (per data stream).

5.2.5 Data to FIFO

The last sub-block of the A2F block is "Data to FIFO"-block (D2F). The D2F block adds a timestamp to the data of the ADC and forwards it to the FIFOs when the CDS signals it to do so. The block combines the data of 2 inputs: ADC data (14 bits data and 1 out of range bit) and a timestamp (16 bits) from the Counter sub-block. If the CDS signals a common rising edge is upcoming and the FIFO is ready to receive data (detected via the Control and Status Register of the FIFO), the block combines the two sets and adds an extra valid bit. An overview of the output word structure of the D2F is shown in Figure 5.4. Note that the "V"-bit stand for valid and the "O"-bit is the Out of Range bit. The valid bit is added to detect any burst interruptions. The A2F sub-block must send a sequential set of data to the FIFOs. If the 'ready'-signal of the FIFO goes low while the FIFO is not yet filled (according to the CSR-data), the valid bit is lowered. This means that there has been an interrupt in the burst of data to the FIFO⁸. Although the A2F block keeps sending data, it could not be guaranteed that this data is saved in the FIFO. Due to the implementation of the FIFO it is necessary to fill it completely even though the data might be invalid. The Valid-flag is used to check for any corrupt/invalid data.

⁷Thanks to Ir E. Molenkamp for providing VHDL container with ready-to-use gray code utilities.

 $^{^{8}}$ Note that the system is designed that this should not able to happen anytime. However, it is of uttermost importance that the datasets in the FIFOs are sequential. Therefore this bit has been introduced to be 100% sure of this.

5.3 FIFOs

The FIFOs fulfil the role of data buffer between HPS and ADCs. The FIFO contains a large array of 32 bit memory elements and the number of memory elements is programmable. An overview of the FIFO block is shown in Figure 5.5. This section explains why custom FIFOs are designed, the connectivity to the FIFO, the memory array implementation, and the CSR implementation.

5.3.1 FIFO: COTS vs Custom

FIFOs are widely used blocks in digital design. It would be senseless to design one, if also a Commercial of the Shelf (COTS) implementation is available. Altera provides a standard FIFO in the IP library included with the Quartus Platform Designer. Therefore, it was tried to include this IP in the system. It turned out that this FIFO is quite slow. It needs several clock cycles to store a data word. To overcome this problem a custom FIFO implementation is designed.

5.3.2 Connectivity of FIFO

A FIFO is connected with 4 busses to the rest of the system:

- One streaming data bus from the A2F to the FIFO for receiving data. This 32 bits wide bus is implemented by Avalon Streaming Interface.
- One streaming data bus from the FIFO to the A2F for sending CSR information. This 32 bits wide bus is implemented by Avalon Streaming Interface.
- Two memory mapped busses to a system bus of the HPS. One for retrieving data out of the FIFO and one to retrieve CSR information for the HPS. These 32 bits wide busses are connected to the HPS via the Avalon Memory Mapped Interface.

5.3.3 Memory Array

The memory array is an array of registers. The memory array is 32 bits wide. The FIFOs depth is programmable at HDL (Hardware Description Language) level. In the Instrument, the FIFOs are 256 and 224 entries deep since larger values (512+) do not fit in the FPGA.

5.3.4 Control Logic

The control logic controls the memory and provides Control and Status information (CSR) on the input and output ports of the FIFO. First of all it keeps track of the fill-level of the FIFO memory via a write pointer and it keeps track of which elements are red via a read pointer.

The FIFO is implemented by four processes with different tasks: *Reading, writing, CSR in, and CSR out.* All these processes use the shared read and write pointer. The *writing* process is responsible forwards the incoming data to the memory array and increasing the write pointer. The *reading* process is responsible for forwarding the data from the memory array (at the location of the current read pointer) to the output port of the FIFO and increasing the read pointer. The *CSR in* process provides CSR info to the A2F block at the input of the FIFO and the *CSR out* process provides CSR info to the HPS at the output of the FIFO.

The CSR port provide status information about the state FIFO ⁹. The word structure of the CSR input and output port is visualised in Figure 5.6. It outputs both the current read pointer and current write pointer as well as two status bits: a "ready" flag and a "full" flag. The ready flag is high when the FIFO

 $^{^{9}}$ CSR is an abbreviation for Control and Status Register. In the presented case, this port provides only status information. The term "CSR" is widely used in the field and therefore the port has not been renamed to CR or C-port.



Figure 5.5: A schematic representation of the FIFO implementation showing the memory array, control logic, and output registers.

$31 \ 30$	29 2	8 27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			wr	ite	po	oin	ter	•										re	ad	po	oint	er						F	R

Figure 5.6: The structure of the output of the CSR ports. R = Ready bit, F = Full bit.

is ready to receive new data (when the read and write pointer are zero). The full flag signals the HPS that it could start reading data and it signals the A2F that it should stop sending data. This flag is high when the FIFO is filled completely (write pointer is equal to FIFO depth).

The memory array is reset to all zeros after it has been read completely. This is not strictly necessary, but makes it easier to spot any invalid data (the probability that all zeros are written to a FIFO entry is virtually zero).

5.4 Operating System

The Linux distribution Xillinux is used as Operating System. This is an Ubuntu based Linux distribution for SoCs of Xilinx but there is also one version that supports the Intel Cyclone V SoCs. This Linux distribution is used over e.g. Buildroot since it works out-of-the-box and includes Ethernet capabilities out-of-the-box.

5.5 Timing Issues

The TimeQuest Timing Analyzer of the Quartus toolset is used to analyse timing issues in the designed system. It turned out that the system experienced quite some timing issues (5000+). All found problems appear at the crossing of clock domains. This could result in meta-stability and wrong register states. Although normally these problems are tackled by using synchronisation flip-flops, these flip-flops are not included in this design. The reason for this is two sided. First of all, due to the high data rates it would be quite a difficult task to get this up-and-running in a right manner. Secondly, there seems no need for

this; the timing problems do exist but do not result in meta-stability and wrong register states. This is the result of that most timing problems are reported at the clock domain crossing in the FIFO. Because the FIFO is first filled completely before it is read completely, no data crosses a clock domain without being buffered for a considerable long time. The only unbuffered signal that crosses the a clock domain boarder in the FIFOs is the data in the CSR. In this case mainly the full flag of this register is of interest. This flag signals the HPS to start reading data out of the FIFO. This flag is only raised when the FIFO is filled completely. If this bit experiences any meta-stability when changing from low to high, it may result in a delayed raise of the bit by a clock cycle.

Besides the foregoing, there can be thought of one condition within the FIFOs that results in the wrong read-out of the write pointer due to meta-stability. This happens when the last entry of the FIFO is filled. In this case, it could happen that the write pointer is being updated when a clock pulse within the CLKHPS domain is present. It could happen that the write pointer is already stable while the data being written is not. In that case the filled flag is raised while the data written in the last entry of the FIFO is not yet stable. However, this poses no problems because the HPS first reads all other entries of the FIFO. When the HPS eventually reads the last entry of the FIFO, the data in this register had plenty of time to become stable. So this situation poses no problem at all.

Although no problems are expected, there are still two measures introduced to detect timing issues and reduce the impact of meta-stability. Firstly, the read and write pointers in the FIFO are stored in a gray code format such that meta-stability could only affect one bit of these pointers (see also subsection 5.2.4). The second measure is the introduction of timestamps to data. If any data-points are lost due to timing issues, this is observable.

Although it is not formally validated if timing errors occur in the system, it is investigated if they are detectable on outputs of the system. Therefore, some deterministic data has been fed into the system (instead of data from the ADC). Two measurements of an hour have been performed: One at room temperature and one at $60+^{\circ}C^{10}$. All the output data was as expected and also the timestamps did not indicate any possible missed data-points. The results of these test is that the effect of possible timing issues cannot be observed at the output of the system. Therefore, it is assumed that the timing issues do not manifest themselfs at the output of the system within the working conditions of the system.

 $^{^{10}}$ This test is performed by aiming a hot air gun at 125°C at the package of the FPGA, while loading both cores of the HPS by calculating some floating point multiplication to produce some extra heat.

Part IV

Measurement Setup, Method, and Results

Chapter 6: Measurement Setup and Method

Part IV is devoted to the measurement setup, method, and measurement results. Chapter 6 describes the measurement setup, problems faced during the design of it, and how the measurements are performed. Chapter 7 presents the results of the measurements.

Chapter 6 is divided into 2 section. Section 6.1 introduces the measurement setup and synchronisation problems faced in the design phase. Section 6.2 describes the measurement method and different measurements. The measurement results are presented in Chapter 7 (for QAM measurements) and section 9.3 (random signals/noise).

6.1 Measurement Setup

A schematic overview of the measurement setup is shown in Figure 6.1. The setup consist of an Arbitrary Waveform Generator (AWG), two power splitters, the Instrument, and some wires. The AWG (Keysight M8190A) is used to generate a signal of interest (SoI) on channel 1. This SoI is split by a power splitter (HP 11667B) and sent to both ADCs. Channel 2 of the AWG is used to generate a Data Synchronisation Pulse (Pulse) This Pulse is used for synchronisation purposes (see subsection 6.1.2). Figure 6.2 displays a picture of the setup. The labels in the Figure 6.2 refer to: Instrument (1), AWG (2), HP 11667B power splitter (3), Mini-Circuits ZFRSC-123-S+ power splitter (4).

6.1.1 Clock Synchronisation

Since the SoI are QAM constellations, the phase of the measured frequency components should be well known. To make this possible, all sample clocks within the measurement setup are locked. The same PLL IP generates both sample clocks, ensuring them to be in-phase (see also subsection 5.1.1). The same PLL-IP that generates the sample clocks, also generates an 8 MHz Sync Clock (see subsection 5.1.1 for more about the Sync Clock). The AWG uses the Sync Clock as reference for its sample clock. Within the AWG it is upconverted to the 7.168 GHz sample clock. The AWG sample clock is thus locked to the beat period of the sample clocks of the Instrument via the Sync Clock.

6.1.2 Data Synchronisation

To perform proper phase measurements, the time between the start of a symbol and the start of a measurement should be measurable. This problem is already discussed in Figure 5.2.3. This section shortly introduces the problem again and describes the implemented solution.

The validity of the measured phase can be ensured in two manners: Either a measurement should always start at a known time difference relative to the start of a symbol or the phase difference between the start of a symbol and the start of a measurement should be measurable (e.g. via a reference signal). Since absolute timing cannot be guaranteed within the measurement setup, the latter is implemented (see section 6.2) via a reference tone. This implies that the measurements should start within the first period of this reference tone (since the phase difference between the reference tone and the SoI may vary from period to period of the reference tone). An extra synchronisation step of the data is necessary to ensure that the measurements always start within the first period of the reference signal. This subsection dives into the implementation of the data synchronisation. Figure 6.3 shows a diagram of the data synchronisation system.

Figure 6.3 shows the Sync Clock (as described in subsection 6.1.1) in red. The Sync Clock is (besides as reference clock) also used as external trigger for the AWG. When the AWG receives an external trigger (depicted by arrow 1 in Figure 6.3), the AWG starts generating the SoI (the blue waveform in Figure 6.3) which is already stored in its memory. The AWG repeats the SoI until the stop signal is send. Figure 6.3



Figure 6.1: A schematic overview of the measurement setup used to measure the MR-DFT.



Figure 6.2: A picture of the measurement setup. Figure 6.1 shows a diagram of the measurement setup. 1: The Instrument. 2: AWG. 3: Power splitter (HP 11667B). 4: Power splitter (Mini-Circuits ZFRSC-123-S+).



Figure 6.3: The data synchronisation process. In blue the data output. In purple the Pulse. In green of a measurement is capturing data or not and in red the 8 MHz Sync Clock. Arrow 1 represents the triggering of the AWG by the Sync Clock to start generating the data signal. Arrow 2 represents the triggering of the Instrument by the Pulse to start a measurement

shows this by the repetition of the blue waveform. Since the reference tone is equal to the beat period of the SoI, the SoI is an integer number of beat periods long, and the sample clock of the AWG is phase locked to the Sync Clock, the SoI always has a constant phase difference between the start of a symbol and the start of the beat period of the Instrument (Sync Clock). This is shown in Figure 6.3 by that a new blue waveform always starts at a rising edge of the red Sync Clock. For simplicity, in Figure 6.3 the constant phase difference is zero. In reality, there is a bit of constant phase difference and phase difference is smaller than the beat period.

It should be ensured that a measurement starts in the first beat period of the SoI. Therefore, the AWG generates also a data synchronisation pulse (Pulse) to mark the start of the next repetition of the SoI. Figure 6.3 shows the pulse in purple. When the Instrument receives the Pulse (and is ready to capture data), it starts capturing data (depicted by arrow 2 in Figure 6.3). Figure 6.3 shows in green if the Instrument is capturing data. The Instrument is going to capture data until the FIFOs are filled. In Figure 6.3, the FIFOs are filled at the last rising edge of the red Sync Pulse. In this manner it is ensured that measurements always start within the first beat period of the SoI

6.1.3 AWG Settings

For signal generation, an Arbitrary Waveform Generator is used. In this case, the used AWG is the Keysight M8190A, more specifically the 14 bit, 8 GS/s version. This AWG has an output voltage of 0.7 Vpp for a 50 Ω load (or 0.7 V amplitude for high impedant load)¹. Due to the use of a power splitter (HP 11667B) in the signal path, this voltage is reduced to 0.35 Vpp at the terminals of the ADC. The AWG sample clock of 7.168 GHz is based on the 8 MHz Sync Clock provided by the instrument and internally upconverted to 7.168 GHz. The intended applications for the MR-DFT measures on continues signals. Therefore, it has been chosen to use a high sample frequency such that the generated signal approaches a continues signal. This sample clock frequency of 7.168 GHz is chosen since it is a multiple of 56 · 64. Both the 56 MHz sample clock period as well as the 64 MHz sample clock period span an integer number of AWG sample clock periods (or mathematically saying 7.168 · 10⁹%56 · 10⁶ == 0 and 7.168 · 10⁹%64 · 10⁶ == 0, in which % is the modulo operation). In the design of the measurement,

 $^{^{1}}$ The Keysight M8190A also offers amplified output ports, with an output voltage of a couple of Volts peak-peak. However, the signal quality of these ports is horrible (i.e. a lot of overshoot). Therefore, these outputs have not been used for signal generation.

when the Pulse was not yet added to the design, this was though to be important since the waveforms generated by the AWG have a length of exactly 32 beat periods. After adding the Pulse, this is not important anymore but was not changed.

6.2 Measurement Method

To verify the MR-DFT algorithm and analyse the impact of noise on the MR-DFT accuracy, measurements are performed. This section elaborates on the used measurement method.

For all these measurements, two sample sets are measured; ADC-A samples at 64 MS/s and ADC-B samples at 56 MS/s. In the common beat period (125 nS, 8 MHz) ADC-A takes 8 samples while ADC-B takes 7 samples. Data of a total of 32 beat periods is stored; 256 sequential samples of ADC-A and 224 sequential samples of ADC-B.

The data measured are QAM constellation. This kind of data is selected since QAM modulation is widely used in protocols to send data wirelessly. Therefore, in the intended application (characterising a class-E PA), the probability is high that the class-E PA is amplifying similar waveforms. Due to limitations in the measurement equipment, during a measurement the waveform per channel contains only one symbol. This one symbol is repeated indefinitely until the next symbol is loaded into the AWG. This has as a consequence that effects Inter-Symbol Interference don't effect the measurement results.

After a measurement, the data is transferred to a computer for post processing. The post processing consists of performing a MR-DFT with c0 = 8, c1 = 7, and g = 32. This results in a bin bandwidth of 250 kHz and a total number of reconstructed bins of 224 (see also Equation 3.16). The highest reconstructed bin is at 55.75 MHz. In all measurements a reference signal of 8 MHz (the beat frequency) is added as a reference for the phase measurements (see subsection 6.1.2 for more information about the reference tone). 191 channels from 8.25 MHz to 55.75 MHz are used for measurements.

A MATLAB script controls the AWG via the VISA protocol and controls the Instrument via SSH. The measured data is transferred via FTP from the Instrument's file system to the system that processes the data. A different MATLAB script is used to process the data.

The different measurements performed are:

- non-OFDM QAM16
- OFDM QAM16 without symbol rotation
- OFDM QAM16 with symbol rotation
- OFDM QAM64 with symbol rotation
- Random signals

The next subsection elaborates on how these measurements are performed and in what fashion.

6.2.1 Non-OFDM QAM16

In this measurement QAM16 constellations are sent in all channels sequentially in time and every symbol of a constellation is measured independently. This means that for every symbol per channel a separate waveform is generated. The reference signal uses half of the output amplitude while the other half is used for the SoI. For this measurement run a total of 3056 (191 channels times 16 symbols per channel) measurements are performed to measure all constellations in all frequency channels.

6.2.2 OFDM QAM16 without Symbol Rotation

In this measurement a QAM16 constellation is sent in all channels at once. This represents an OFDMlike signal. In every channel the same symbol is send. This implies that the reference signal and all send symbols have a maximum amplitude of 3.9 mVpp at the output of the AWG (or 1.5 mVpp at the inputs of the Instrument). This number originates from the full range of the AWG (700 mVpp) divided by the number of channels (191 + 1 reference tone) multiplied by a factor of 1.07 due to a bit of destructive interference between the symbols in different channels. This run needs to perform 16 measurement to measure all constellations in all channels.

6.2.3 OFDM QAM16 with Symbol Rotation

In this measurement run a QAM16 constellation is sent in all channels at once. This represents an OFDM-like signal. In this case a kind of symbol rotation is used: In channel 1 symbol 1 is send, in channel 2 symbol 2 is send and so on. In channel 17 again symbol 1 is send. In this manner the signals in different channels do add less in a constructively manner such that more voltage headroom is left for all signals. This results in a maximum amplitude per channel of 9.89 mVpp at the output of the AWG (or 4.94 mVpp at the inputs of the Instrument). This run needs to perform 16 measurements to measure all constellations in all frequency channels.

6.2.4 OFDM QAM64 with Symbol Rotation

In this measurement run a QAM64 constellation is sent in all channels at once. This represents an OFDM-like signal. In this case, also symbol rotation is used like in subsection 6.2.3. All channels have a maximum amplitude of 13.74 mVpp (or 6.77 mVpp at the inputs of the Instrument). This run needs to perform 64 measurements to measure all constellations in all frequency channels.

6.2.5 Random Signals

In this case, every sample of the output of the AWG is sampled from a uniform distribution between -350 mV and 350 mV at the output of the AWG (or -175 mV to 175 mV at the inputs of the Instrument) such that a wide band random signal can be measured.

Chapter 7: Measurement Results

This chapter presents the measurement results, measured with the measurement setup described in section 6.1 and measured via the methods treated in section 6.2^1 . The results of every measurement run are discussed in a separate section. The last section is devoted to the comparison of the results and some conclusions.

In this chapter, the Error Vector Magnitude (EVM) is used as a measure for the quality of measured constellations. The EVM is defined as in [20]

$$EVM = 100 \cdot \sqrt{\frac{\sum_{k=0}^{N-1} |e_k|^2}{N \cdot P_{avg}}}$$
(7.1)

wherein N is the number of symbols in the constellation (so 16 for QAM16), e_k is the error vector (and is defined as the difference between the ideal point of the constellation and the measured point), and P_{avg} is the average power of the ideal points of the constellation.

It turned out that the first of the 10 constellations measured has a significant lower EVM then the other 9. It is unknown why this is the case. Since this seems like outliers, these measure are neglected within the EVM numbers.

EVMs are calculated per constellation. Afterwards, the EVM of all 9 constellations is averaged. The minimum, average, and maximum EVM of the 9 constellations is graphically shown for all measurements. For calculating an EVM, the constellations should be rotated and normalised such that the orientation and scaling becomes equal to that of an ideal constellation. The scaling is done by calculating the ratio of the absolute values between the measured and ideal constellation for a point. This ratio is averaged over all points in the constellation. The same is done for the phase; the phase difference is calculated per point and averaged over the constellation. Afterwards, the constellation is rotated by the found phase difference and scaled by the found ratio. Note that this does not necessarily result in the lowest EVM.

Precise and accurate are used as defined in [21]. Accuracy is the measure to be close the other measurements while precision is the measure to be close to the intended value.

7.1 Non-OFDM QAM16

Figure 7.1 and Figure 7.2 show the results of 10 runs of the measured QAM 16 constellation in a per channel basis (as described in treated in subsection 6.2.1). Figure 7.1 shows the constellation measured in channel 1 (250 kHz @ 8.25 MHz). Figure 7.2 shows the constellation measured in channel 191 (250 kHz @ 55.75 MHz). The blue circles are the measurement points and the orange crosses are the ideal positions of the constellation. Figure 7.2 shows more spread than Figure 7.1 and the spread seems to be in a circular direction around the origin.

The EVM of channel 1 is between 0.076% and 0.098% (average 0.088%) whereas the EVM of channel 191 is between 1.14% and 2.14% (average 1.63%). Figure 7.3 shows the average EVM of all channels of this measurement.

Figure 7.3 shows spikes for certain channels. The centre frequency of these channels are a multiple of 4 MHz (i.e. channel 16 @ 12 MHz, channel 32 @ 16 MHz, and so on). The constellations belonging to these channels show a loss in precision but are still accurate. That is the cause of the spikes. The loss in precision for channels representing a multiple of 8 MHz might be caused by peaks in the noise floor due to a bad assumption regarding the quantization noise distribution (see Figure 9.3 and Figure 9.4). Why the other channels show these spikes is unknown.

It generally seems to be the case that the highest frequency channels show the most spread in the measured points. In the proceeding of this chapter, only the QAM constellation of the channel 191 (250 kHz @ 55.75 MHz) is presented unless there is a clear reason to show other channels.

¹For the measurements on noise, see Chapter 9.



Figure 7.1: The results of 10 runs of non-OFDM QAM16 measurements for channel 1 (250 kHz channel around 8.25 MHz). The measurement points are shown in blue and the ideal point in orange. The EVM of the shown constellation is between 0.30% and 0.41%.



Figure 7.2: The results of 10 runs of the non-OFDM QAM16 measurements for channel 191 (250 kHz channel around 55.75 MHz). The measurement points are shown in blue and the ideal point in orange. The EVM of the shown constellation is between 1.18% and 2.41%.



Figure 7.3: The average EVM of all channels for the constellations of the QAM16 non-OFDM measurements. In blue the average EVM, in orange the maximum EVM, and in yellow the minimum EVM. Note that the values of the average, minimum, and maximum EVM (on average) are very close and might be hard to distinguish.

7.2 OFDM QAM16 without Symbol Rotation

Figure 7.4 shows the results of a measured QAM16 constellation in OFDM fashion without symbol rotation (as described in subsection 6.2.2). The figure shows the constellations measured in channel 191 (250 kHz @ 55.75 MHz). Figure 7.4 shows that the measured points are relatively precise but not very accurate. The deformations occurring in Figure 7.4 is seen in all higher frequency channels. Therefore, it is expected to be the result of a systematic error. Multiple simulations has been performed to find if this specific deformation could be mimicked. These include sweeps over: Phase mismatch between ADCs, gain mismatch between ADCs, jitter (both random and deterministic per clock period), spurious tones, aliasing problems, and combinations of the foregoing. Non of the simulations resulted in the observed deformation. It is also checked if the AWG generates a clean QAM constellation and this seems to be the case as well. So the cause of the deformations is unknown. Furthermore, the voltage level of the SoI utilises the full scale of the AWG output (350 mVpp at the inputs of the Instrument), which is not in the order of the LSB-quantization step (122 μ V) or ENOB-levels(644 μ V) of the ADCs. Although the voltage levels per channel (1.9 mVpp) are close to the quantization levels, due to the OFDM fashion of the signal it is not expected that (deterministic) quantization errors cause the observed behaviour.

The EVM of channel 1 (250 kHz @ 8.25 MHz) ranges from 8.34% to 10.64% (average 9.40%) whereas the EVM channel 191 (250 kHz @ 55.75 MHz) ranges from 16.11% to 19.02% (average 17.38%). Figure 7.5 shows the EVM per channel.

Figure 7.5 shows one spike in channel 96. The script used to rotate and normalise the constellations such that they can be compared to an ideal constellation is unable to rotate this constellation to the right orientation. The peaks at 64 and 160 seem to be constellations which are heavily deformed but also wrongly normalised.



Figure 7.4: The results of 10 runs of OFDM QAM16 measurements without symbol rotation for channel 191 (250 kHz channel around 55.75 MHz). The measurement points are shown in blue and the ideal point in orange. The EVM of the shown constellation is between 16.12% and 19.02%.



Figure 7.5: The average EVM of all channels for the constellations of the QAM16 OFDM measurements without symbol rotation. In blue the average EVM, in orange the maximum EVM, and in yellow the minimum EVM.

7.3 OFDM QAM16 with Symbol Rotation

Figure 7.6 shows the results of a measured QAM16 constellation in OFDM fashion using symbol rotation (as described in subsection 6.2.3). Less constructive interference is present when adding the waveforms of all symbols sent in all channels (compared to section 7.1), providing headroom to increase the signal

amplitude of all symbols by a factor of 2.75 (compared to section 7.2). The constellations measured in channel 191 (250 kHz @ 55.75 MHz) are shown. Most points in the constellations are at the expected position. Some symbols are still precise but not accurate. The reason for this is unknown but since it looks like a systematic error, it is expected to have the same cause as the deformations observed in section 7.2.

The EVM of channel 1 (250 kHz @ 8.25 MHz) ranges from 6.78% to 8.05% (average 7.31%). The EVM of channel 191 (250 kHz @ 55.75 MHz) ranges from 4.10% to 5.46% (average 4.87%). It is unknown why the EVM for lower frequency channels is higher than for higher frequency channels. Figure 7.7 shows the EVM per channel. In Figure 7.7 a peak is observed in channel 96. This constellation is deformed and the normalisation seems to be off. Furthermore, some "levels" are observed on the left of channel 96 (channel 1-32, channel 33-64, channel 65-96). The exact cause is unknown but it might be related to the levels shown in the noise floor (see Chapter 9 and footnote 2 at page 58). The low frequency constellations are less precise than the high frequency constellations, resulting in a higher EVM. Why the lower frequencies are less precise is unknown.



Figure 7.6: The results of 10 runs of OFDM QAM16 measurements with symbol rotation for channel 191 (250 kHz channel around 55.75 MHz). The measured points are shown in blue, the ideal points are shown in orange. The EVM of the shown constellation is between 4.10% and 5.46%.



Figure 7.7: The average EVM of all channels for the constellations of the QAM16 OFDM measurements with symbol rotation. In blue the average EVM, in orange the maximum EVM, and in yellow the minimum EVM.

7.4 OFDM QAM64 with Symbol Rotation

Figure 7.8 and Figure 7.9 show the results of a measured QAM64 constellation in OFDM fashion using symbol rotation (as described in subsection 6.2.4). Figure 7.8 shows channel 16 (250 kHz @ 16 MHz) and Figure 7.9 shows channel 191 (250 kHz @ 55.75 MHz). The constellations in channel 16 are also shown since they are more precise but less accurate than those in channel 19. It is expected that the constellations in channel 16 are more precise but it is not expected that they are less accurate since with the same jitter specs, the deviation due to jitter is lower at lower signal frequencies (the relative effect of jitter is larger at higher signal frequencies). Looking through the constellations in the different channels for this measurement run, it seems characteristic that constellation in some channels are more accurate than in other channels. No patterns have been discovered so it is unclear why this is the case.

The EVM in channel 1 (250 kHz @ 8.25 MHz) ranges from 6.49% to 6.91% (average 6.69%), the EVM of channel 16 ranges from 3.82% to 4.16% (average 4.07%), and the EVM of channel 191 ranges from 3.37% to 4.47% (average 3.88%). Figure 7.10 shows the EVM per channel.

Also the EVM plot of Figure 7.10 shows some spikes. Similar to the previous measurements, these constellations have been inspected and it turns out that they are not properly orientated resulting in faulty EVM values (see Figure 7.11 for an example). Furthermore, the graph looks similar to that of other EVM measurement and also some levels can be observed from set of 32 channels (distinguishable are channels 1-32, 33-64, and 161-191). It is highly probable that this is related to the used MR-DFT configuration².

²In the used MR-DFT configuration, the reconstruction of the harmonics is done in blocks of 32 bins. The performed MR-DFT is similar to that Table 2.1 and Table 2.2 except that every bin in these tables are 32 in this configuration. So the first 64 MR-DFT output bins can directly be extracted from \vec{X}_8 . Output bins 191-224 are extracted from the last 32 bins of \vec{X}_7 by subtracting the bins 32-63 of \vec{X}_8 . Bins 64-95 are extracted from \vec{X}_8 by subtracting the MR-DFT output bins 191-224 from bins 64-95 (of \vec{X}_8) and so on. The author does not believe that it is not a coincident that the 32 bins wide "levels" emerging in the EVM graphs are not related to this calculation.

.

	QAM64	4 conste	llation C)FDM, s	ymbol re	tation	
		Channel	16 (250	kHz @	12 MHz)	Om ×id	leasuement leal
Q	O k	1	۵	1	8	۰. ۲	ð
ש	•	•	ه م	Q	Ŏ	9	o×
•	8	8	9	8	Q	Č	8
Ĩ	٢	8	2	9	8	ø	۶
ו• -	6 ·	-4 🦻 -:	² 😵	8	2 8 4		ô 👩
(đ	•	8	8	8	8	Ŏ
×	Q	8	Q	e	0	×O	٢
•	ŏ	Ŷ	ر ې	()	8	đ	O

Figure 7.8: The results of 10 runs of OFDM QAM64 measurements with symbol rotation for channel 16 (250 kHz channel around 16 MHz). The measured points are shown in blue, the ideal points are shown in orange.



Figure 7.9: The results of 10 runs of OFDM QAM64 measurements with symbol rotation for channel 191 (250 kHz channel around 55.75 MHz). The measured points are shown in blue, the ideal points are shown in orange.



Figure 7.10: The average EVM of all channels for the constellations of the QAM64 OFDM measurements with symbol rotation. In blue the average EVM, in orange the maximum EVM, and in yellow the minimum EVM.



Figure 7.11: A QAM constellation which is not orientated correctly by the scripts used for visualising the constellations and calculating the EVM. In this case, the results of 10 runs of OFDM QAM64 measurements with symbol rotation for channel 23 (250 kHz channel around 13.75 MHz) are shown. The measured points are shown in blue, the ideal points are shown in orange.

7.5 Comparison and Conclusions

This chapter shows the resulting QAM constellations measured in different manners. The results at high frequencies and low voltage levels show a high precision but low accuracy, resulting in deformed constellation (OFDM QAM16 without symbol rotation). When amplitude per channel is increased, the precision of the constellation rapidly increases (OFDM QAM16 with symbol rotation). Since the deformation in the OFDM QAM16 without symbol ration measurements are consistent over different measurement runs, it is expected that this is caused by a systematic error/disturbance (e.g. quantization or thermal noise, jitter, distortion, lack of ADC resolution, and alike) or systematic error in the measurement setup. It is not expected that the shown deformation is a characteristic of the MR-DFT (since it has not been observed in other MR-DFT measurements performed with different measurement setups). Simulations sweeping over phase mismatch between ADCs, gain mismatch between ADCs, jitter (both random and deterministic per clock period), spurious tones, aliasing problems, and combinations of the foregoing are performed, but the results do not match the measured deformations. The exact cause of the deformations is unknown

Table 7.1 shows the EVM for channel 1, 16, and 191 for the different measurement presented in this chapter. Generally, it can be concluded that the EVM decreases when symbol amplitude increase (non-OFDM vs OFDM and non-symbol rotation vs symbol rotation). When larger symbol amplitudes are use, the relative effect of noise and distortion decreases. Furthermore, looking at the different plots showing the EVM over the channels (Figure 7.3, Figure 7.5, Figure 7.7, and Figure 7.10), the difference between the minimum, average, and maximum EVM increases with increasing frequency. It is expected that this is jitter related since the relative impact of jitter increases for higher frequency signals.

The conclusion is that the MR-DFT is capable of reconstructing the constellations, also for channels in higher Nyquist zones.

Channel	QAM16 non-OFDM	OFDM16	OFDM16 Symbol Rotation	OFDM64 Symbol Rotation
1	0.088%	9.40%	7.31%	6.69%
16	0.38~%	9.55%	6.40%	4.07%
191	1.63%	17.38%	4.87%	3.88%

Table 7.1: A overview of the EVMs presented in this chapter. The EVM of channel 1, 16, and 191. The presented measurements are: QAM16 non-OFDM (section 7.1), OFDM16 (subsection 6.2.2, no symbol rotation applied), OFDM16 Symbol Rotation (subsection 6.2.3), and OFDM64 Symbol Rotation (subsection 6.2.4).

$\mathbf{Part}~\mathbf{V}$

MR-DFT and Noise

Chapter 8: Disturbances and Noise in MR-DFT

Part V covers the impact of disturbances of the input signal or ADC sample clock on the output bins of the MR-DFT. Chapter 8 introduces the theory and mathematics involved in the impact of disturbances on noise floor of the MR-DFT. Chapter 9 compares the theory, simulations, and measurements on the impact of (mainly quantization) noise on the output of the MR-DFT.

A number of disturbances¹ degrade the performance of the MR-DFT. This chapter shows two approaches to determine the effect of disturbances on the output of the MR-DFT. A spectral approach is introduced and as example the effect of ADC sample clock jitter is investigated. Section 8.2 presents a statistical approach and as example the effect of quantization error is investigated. The last section of this chapter presents an analysis of the noise shaping properties of the MR-DFT and as an example a model for the effect of the quantization error on the noise floor of the MR-DFT is presented.

Initially, for an on-chip MR-DFT implementation the different sample clocks were planned to be generated by a Delay Locked Loop (DLL). These devices suffer from a possible Locking Errors. The effect of a DLL Locking Error on the MR-DFT was investigated. Plans have changed and this analysis is not relevant anymore. Therefore, it has been moved to Appendix B.

8.1 Disturbances: A Spectral Approach

The MR-DFT is composed of all linear operations. To analyse the impact of the disturbances on the reconstructed output bins \vec{H} , the vector \vec{Y} is decomposed in an ideal vector \vec{Y}_{sig} and a vector caused by the disturbance under investigation \vec{Y}_{noise} . Doing so, the effect of the disturbance on the output of the MR-DFT can be calculated as:

$$\vec{H} = 2 \cdot \mathbf{R}^{-1} \cdot Re(\vec{Y}) + j \cdot 2 \cdot \mathbf{S}^{-1} \cdot Im(\vec{Y}) = 2 \left(\mathbf{R}^{-1} \cdot Re(\vec{Y}_{sig}) + \mathbf{R}^{-1} \cdot Re(\vec{Y}_{noise}) + j \left(\mathbf{S}^{-1} \cdot Im(\vec{Y}_{sig}) + \mathbf{S}^{-1} \cdot Im(\vec{Y}_{noise}) \right) \right)$$
(8.1)

The effect of the disturbance on the output of the MR-DFT is thus equal to

$$H_{m,noise} = 2 \cdot \vec{R_m^{-1}} \cdot \vec{Y_{r,noise}} + j \cdot 2 \cdot \vec{S_m^{-1}} \cdot \vec{Y_{i,noise}}$$

$$(8.2)$$

in which $\vec{R_m^{-1}}$ is the *m*-th row of \mathbf{R}^{-1} and $\vec{S_m^{-1}}$ is the *m*-th row of \mathbf{S}^{-1} . The columns of \mathbf{R}^{-1} and \mathbf{S}^{-1} determine how \vec{Y}_{noise} ends up in \vec{H} .

Since the MR-DFT-matrices (\mathbf{R}^{-1} and \mathbf{S}^{-1}) depends on the used configuration/setup, it it not straight forward to give qualitative results about the effect of some disturbances on the MR-DFT results.

As an example, the sample clock jitter is discussed in subsection 8.1.1.

8.1.1 ADC Sampling Clock Jitter and (MR-)DFT

The proposed jitter analysis technique is similar to those of ADC jitter analysis (as for example described in [14] or [22]).

Sampling with an ideal sample clock in the time domain is equivalent to the multiplication of the signal with a(n infinitely continuing) train of evenly spaced delta pulses:

$$s[n] = s(t) \cdot \sum_{n = -\infty}^{\infty} \delta(t - n \cdot \Delta T_s) = s(t) \cdot f_s(t)$$
(8.3)

 $^{^1\}mathrm{Disturbances}$ refers to all non-idealities like noise, jitter, and so on.

In the frequency domain representation of a train of evenly spaced infinitely continuing delta pulses is as well a train of evenly spaced infinitely continuing delta pulses [23]:

$$F_s(\omega) = \mathscr{F}\left(\sum_{n=-\infty}^{\infty} \delta(t - n \cdot \Delta T)\right) = \frac{2\pi}{\Delta T} \sum_{n=-\infty}^{\infty} \delta\left(\omega - \frac{2n\pi}{\Delta T}\right)$$
(8.4)

In which \mathscr{F} is the Fourier transform operation. Multiplying in the time domain is convolving in the frequency domain. The effect of sampling in the frequency domain is to convolving the Fourier transform of the signal with a train of evenly spaced infinitely continuing delta pulses. The DFT of signal s[n] is equal to

$$S(\Omega) = S(\omega) * F_s(\omega) \tag{8.5}$$

In the foregoing, it is assumed that the sample clock is ideal; every clock pulse is exactly timed and infinitely steep. In reality, clock signals do not behave in this manner; clock edges are not exactly evenly spaced (jitter) and clock edges are not infinitely steep. These time domain effects, result in phase skirts around the delta peaks in the frequency domain. For a discussion of clock jitter and the effects of it, see, for example, [22] or [24].

The consequences of jitter for a DFT is that the DFTs of the (ideally sampled) Collection of Sample Sets should be convolved with the frequency spectrum of the sample clocks:

$$\vec{X}_c = \mathscr{F}(\vec{x}_c) * \vec{F}_{s,c} \tag{8.6}$$

For the MR-DFT this implies that the convolution of the spectra of the sample clock with jitter and the Sample Sets are used in the matrix multiplication with the MR-DFT-matrix (Equation 3.15).

8.2 Disturbances: A Statistical Approach

In this approach, the impact of a disturbance or noise source on the output of the MR-DFT is analysing by propagating the properties of the statistical distribution of the disturbance to the output of the MR-DFT. The disturbance causes sample errors with a certain distribution. Via the Collection of Discrete Spectra and Equation 8.1 this error propagates to the output of the MR-DFT. The essence of this approach is to keeps track of the statistical properties of the effect of the error in the various steps in the MR-DFT. This enables to determine the statistical properties of the effect of the error on the output of the MR-DFT, providing a measure for the expected deviation due to the disturbance.

As an example, subsection 8.2.1 discusses this approach applied to quantization noise. This approach can be generalised to other noise sources.

8.2.1 The Statistics of Quantization Error and the DFT

This section assumes that the quantization error is uniformly distributed among $\left[-\frac{1}{2}LSB, \frac{1}{2}LSB\right]$ (see for example [14] for the requirements for this assumption).

Assume the signal x(t) is sampled at integer multiples of T,

$$x[n] = x(n \cdot T) \text{ for } n \in \mathbb{Z}$$

$$(8.7)$$

y[n] is the quantized version of x[n],

$$y[n] = x[n] + \epsilon[n] \tag{8.8}$$

in which $\epsilon[n]$ is the quantization error for sample n. ϵ is uniformly distributed within $\left[-\frac{1}{2}LSB, \frac{1}{2}LSB\right]$,

$$\epsilon = U\left(-\frac{LSB}{2}, \frac{LSB}{2}\right) \tag{8.9}$$

Since the DFT is a linear operation, the DFT of y[n] is equal to

$$\vec{Y} = DFT(y[n]) = DFT(x[n]) + DFT(\epsilon[n])$$
(8.10)

The DFT(x[n]) is the part of interest of Y_k and the $DFT(\epsilon[n])$ is the part of Y_k caused by the quantization error. The DFT of the quantization error is defined as

$$E_{k} = \frac{1}{N} \sum_{n=0}^{N-1} \epsilon[n] \cdot e^{\frac{-j2\pi kn}{N}}$$
(8.11)

In case of N=8, Equation 8.11 evaluates in:

$$E_{k} = \frac{1}{N} \left(\epsilon[0] + \epsilon[1] \cdot e^{\frac{-j\pi k}{4}} + \epsilon[2] \cdot e^{\frac{-j\pi k}{2}} + \epsilon[3] \cdot e^{\frac{-j3\pi k}{4}} + \epsilon[4] \cdot e^{-j\pi k} + \epsilon[5] \cdot e^{\frac{-j5\pi k}{4}} + \epsilon[6] \cdot e^{\frac{-j3\pi k}{2}} + \epsilon[7] \cdot e^{\frac{-j7\pi k}{4}} \right)$$
(8.12)

Equation 8.12 results in a sum of samples (sampled out of the same uniform distribution) multiplied by a complex vector with unity length. So loosely speaking, the uniform distributions within this sum are rotated around the origin but do have the same standard deviation.

The Central Limit Theorem is used to determine the distribution of E_k . The CLT states that a sum of independent random variables, all with equal distribution, approaches a normal distribution when summed (due to convolution of the Probability Density Functions) with the mean and variance:

$$\mu_t = \sum_n \mu_n$$

$$\sigma_t^2 = \sum_n \sigma_n^2$$
(8.13)

In Equation 8.12, the summed uniform distribution do not have equal distributions due to the multiplication by a complex vector. The standard CLT is not applicable. The Lyapunov CLT, a variant of the CLT, states that the sum of not necessary identical distributions tends toward a normal distribution if certain criteria are met². Moreover, a statement of [25] is interpreted by the author as that the Lyapunov CLT can be easily generalised for complex numbers. Due to a lack of statically knowledge, the author is not able to check the Lyapunov criteria. Simulations have been performed to do so. The results of these simulations are in accordance with Equation 8.13.

Assuming the (Lyapunov) CLT is applicable to this situation, the expected value and the variance of the error (due to the quantization) are found via Equation 8.11 and Equation 8.13. The uniform distribution are first multiplied by the factor 1/N of Equation 8.12:

$$\mu_n = E\left[U\left(-\frac{LSB}{2N}, \frac{LSB}{2N}\right)\right] = 0$$

$$\mu_t = E\left[E_k\right] = \sum_{n=0}^{N-1} \mu_n = 0$$

$$\sigma_n^2 = var\left(U\left(-\frac{LSB}{2N}, \frac{LSB}{2N}\right)\right) = \frac{1}{12}\left(\frac{LSB}{2N} + \frac{LSB}{2N}\right)^2 = \frac{1}{12}\left(\frac{LSB}{N}\right)^2$$

$$\sigma_t^2 = var(E_k) = \sum_{n=0}^{N-1} \sigma_n^2 = N \cdot \sigma_n^2 = \frac{1}{12 \cdot N}(LSB)^2$$
(8.14)

 $^{^{2}}$ The author has been told by a mathematician that it is hard to prove the Lyapunov criteria but that for "standard" statistical problems (like this one) the criteria are met. So assuming that these criteria are met is a save bet. To verify this statement, simulations have been performed. The results of the simulations are in accordance with Equation 8.13.

These results are verified by simulations. An in-depth discussion of the simulations and the results can be found in Appendix C. The conclusion is that the statistics of E_k follow the Lyapunov CLT for sufficient large values of N (give or take >4). The distribution of the quantization error within a DFT bin converges toward a real (1D) or complex (2D) normal distribution. The mean and variance of the amplitude in DFT bin k, due the quantization error, are equal to:

$$E[E_k] = 0$$

$$var(E_k) = \sigma_{qe}^2 = \frac{1}{12 \cdot N} LSB^2$$
(8.15)

and the distribution of E_k is equal to:

$$E_k = CN\left(0, \frac{1}{12 \cdot N} (LSB)^2\right) \text{ for } k\% \frac{N_c}{2} \neq 0$$

$$E_k = N\left(0, \frac{1}{12 \cdot N} (LSB)^2\right) \text{ for } k\% \frac{N_c}{2} = 0$$
(8.16)

in which $CN(\mu, \sigma^2)$ is the circular symmetric complex normal distribution (see Appendix C and Appendix D) and $N(\mu, \sigma^2)$ is the normal distribution. The power in a bin due to the quantization error is equal to

$$E[E_k^2] = var(E_k) + E[E_k]^2 = \sigma_{qe}^2 = \frac{1}{12 \cdot N} LSB^2$$
(8.17)

which is equivalent to the standard formula for the quantization error for an >6 bits ADC (see for example [14]).

The next section describes how these results effect the output of the MR-DFT.

8.3 MR-DFT Noise Shaping

This section mathematically describes the cause of the noise floor shaping property of the MR-DFT. In the proceeding of this section, quantization noise is taken into account but the approach is generally applicable to noise in general. Keep in mind that the effect of correlation of noise sources and (in)dependence of noise samples might differ from noise source to noise source.

This section redefines \vec{Y} . This section assumes that \vec{Y}_r and \vec{Y}_i contain the real and imaginary parts of the same DFT bins. This implies $Re(\vec{Y}) = \vec{Y}_r$ and $Im(\vec{Y}) = \vec{Y}_i$. As explained in subsection 3.3.3, this is not necessarily the case and therefore technically not correct but it greatly reduces the complexity of notations making the principles easier understandable.

Given Equation 8.16, the statistics of the different bins of \vec{Y}_n are:

$$\vec{Y}_{n} = \begin{bmatrix} N(0, \sigma_{c0}^{2}) \\ CN(0, \sigma_{c0}^{2}) \\ \vdots \\ CN(0, \sigma_{c0}^{2}) \\ N(0, \sigma_{c1}^{2}) \\ CN(0, \sigma_{c1}^{2}) \\ \vdots \\ CN(0, \sigma_{c1}^{2}) \\ \vdots \\ N(0, \sigma_{c(d-1)}^{2}) \\ \vdots \\ CN(0, \sigma_{c(d-1)}^{2}) \\ \vdots \\ CN(0, \sigma_{c(d-1)}^{2}) \\ \vdots \\ CN(0, \sigma_{c(d-1)}^{2}) \end{bmatrix}$$

$$(8.18)$$

in which σ_{ci}^2 is equal to $\frac{1}{12 \cdot N_{ci}} \cdot (LSB)^2$ (Equation 8.15) and $CN(\mu, \sigma^2)$ is a complex normal distribution. Note that this is also strictly not correct. Some of the complex normal distribution evaluate to a normal distribution (see also the explanation around Equation C.2). For the sake of complexity, this is omitted in the notations in this chapter but is taken into account in simulations presented in section 9.1.

Due to the circular symmetry, the complex normal distribution can be written as (see Appendix D for a discussion of the complex normal distribution and some of its properties):

$$CN\left(0,\sigma_{c}^{2}\right) = N\left(0,\frac{\sigma_{c}^{2}}{2}\right) + j \cdot N\left(0,\frac{\sigma_{c}^{2}}{2}\right)$$

$$(8.19)$$

The various σ_{ci} in Equation 8.18 can be expressed into one another. Since

$$\sigma_{ci}^2 = \frac{1}{12 \cdot N_{ci}} (LSB)^2 \tag{8.20}$$

and LSB is equal for all different Sample Sets³, it holds that

$$\sigma_{ci}^2 = \frac{N_{cj}}{N_{ci}} \sigma_{cj}^2 \text{ for all } ci, cj$$
(8.21)

Combining Equation 8.19 and Equation 8.21 into Equation 8.18 and using σ_{c0} as reference, \vec{Y}_n becomes:

$$\vec{Y}_{n,eq} = \begin{bmatrix} N(0, \sigma_{c0}^2) \\ N(0, \sigma_{c0}^2/2) + j \cdot N(0, \sigma_{c0}^2/2) \\ \vdots \\ N(0, \sigma_{c0}^2/2) + j \cdot N(0, \sigma_{c0}^2/2) \\ N(0, \frac{N_{c0}}{N_{c1}} \sigma_{c0}^2/2) + j \cdot N(0, \frac{N_{c0}}{N_{c1}} \sigma_{c0}^2/2) \\ \vdots \\ N(0, \frac{N_{c0}}{N_{c1}} \sigma_{c0}^2/2) + j \cdot N(0, \frac{N_{c0}}{N_{c1}} \sigma_{c0}^2/2) \\ \vdots \\ N(0, \frac{N_{c0}}{N_{c(d-1)}} \sigma_{c0}^2/2) + j \cdot N(0, \frac{N_{c0}}{N_{c(d-1)}} \sigma_{c0}^2/2) \\ \vdots \\ N(0, \frac{N_{c0}}{N_{c(d-1)}} \sigma_{c0}^2/2) + j \cdot N(0, \frac{N_{c0}}{N_{c(d-1)}} \sigma_{c0}^2/2) \\ \vdots \\ N(0, \frac{N_{c0}}{N_{c(d-1)}} \sigma_{c0}^2/2) + j \cdot N(0, \frac{N_{c0}}{N_{c(d-1)}} \sigma_{c0}^2/2) \end{bmatrix}$$

$$(8.22)$$

 $\vec{Y}_{n,eq}$ is split in a real and imaginary part. Note again that this is not strictly correct since the elements selected for \vec{Y}_r and \vec{Y}_i may differ. For the easy of simple notations, this is neglected.

$$\vec{Y}_{r,n,eq} = Re(\vec{Y}_{n,eq}) \tag{8.23a}$$

$$\vec{Y}_{i,n,eq} = Im(\vec{Y}_{n,eq}) \tag{8.23b}$$

Now the MR-DFT comes into play. Recall from Equation 3.15 that the MR-DFT is defined as

$$\vec{H} = 2 \cdot \mathbf{R}^{-1} \cdot \vec{Y}_r + j \cdot 2 \cdot \mathbf{S}^{-1} \cdot \vec{Y}_i \tag{8.24}$$

The part of H_m due to noise is equal to:

$$H_m = 2 \cdot \vec{R_m^{-1}} \cdot \vec{Y_{r,n,eq}} + j \cdot 2 \cdot \vec{S_m^{-1}} \cdot \vec{Y_{i,n,eq}}$$
(8.25)

The expected noise floor is simply the expected noise power per bin. Statistically speaking this is $E[H_m^2]$ and is equal to

 $^{^{3}\}mbox{The same quantization error/noise performance is assumed for all Sample Sets and/or ADCs used to sample the Sample Sets.$

$$E\left[|H_{m}|^{2}\right] = E\left[|2 \cdot \vec{R_{m}^{-1}} \cdot \vec{Y_{r,n,eq}} + j \cdot 2 \cdot \vec{S_{m}^{-1}} \cdot \vec{Y_{i,n,eq}}|^{2}\right]$$

$$= E\left[\sqrt{(2 \cdot \vec{R_{m}^{-1}} \cdot \vec{Y_{r,n,eq}})^{2} + (2 \cdot \vec{S_{m}^{-1}} \cdot \vec{Y_{i,n,eq}})^{2}}\right]$$

$$= E\left[(2 \cdot \vec{R_{m}^{-1}} \cdot \vec{Y_{r,n,eq}})^{2} + (2 \cdot \vec{S_{m}^{-1}} \cdot \vec{Y_{i,n,eq}})^{2}\right]$$

$$= 4 \cdot E\left[(\vec{R_{m}^{-1}} \cdot \vec{Y_{r,n,eq}})^{2}\right] + 4 \cdot E\left[(\vec{S_{m}^{-1}} \cdot \vec{Y_{i,n,eq}})^{2}\right]$$
(8.26)

In which

$$E\left[(\vec{R_m^{-1}} \cdot \vec{Y}_{r,n,eq})^2\right] = E\left[\sum_l (R_{m,l}^{-1})^2 (Y_{r,n,eq,l})^2\right] + E\left[\sum_{k \neq l} R_{m,k}^{-1} R_{m,l}^{-1} Y_{r,n,eq,k} Y_{r,n,eq,l}\right]$$
(8.27a)

$$E\left[(\vec{S_m^{-1}} \cdot \vec{Y}_{i,n,eq})^2\right] = E\left[\sum_l (S_{m,l}^{-1})^2 (Y_{i,n,eq,l})^2\right] + E\left[\sum_{k \neq l} S_{m,k}^{-1} S_{m,l}^{-1} Y_{i,n,eq,k} Y_{i,n,eq,l}\right]$$
(8.27b)

For k, l = 0, 1, 2, ..., M - 1 and $R_{m,k}^{-1}$ and $S_{m,k}^{-1}$ are respectively the k-th element of $\vec{R_m^{-1}}$ and $\vec{S_m^{-1}}$. In the same manner $Y_{r,n,eq,k}$ and $Y_{i,n,eq,k}$ are respectively the k-th element of $\vec{Y_{r,n,eq}}$ and $\vec{Y_{i,n,eq}}$.

Squared terms

Recall from equation Equation 8.27 that the expected noise power per bin is equal to the sum of the expected values of two squared terms and two cross terms. The square terms always correlate since they are a multiplication with themselves. Furthermore, due to the square nature of this term the value of a square term is always positive.

Using the properties of Equation E.5 and Equation E.6a of the Chi-squared distribution (see Appendix E for a short introduction of the squared normal distribution and its relation to the Chi-squared distribution) results in:

$$E\left[\sum_{l} (R_{m,l}^{-1})^2 (Y_{r,n,eq,l})^2\right] = \sum_{l} \left((R_{m,l}^{-1})^2 \cdot E\left[Y_{r,n,eq,l}^2\right] \right) = \sum_{l} \left((R_{m,l}^{-1})^2 \cdot var(Y_{r,n,eq,l}) \right)$$
(8.28a)

$$E\left[\sum_{l} (S_{m,l}^{-1})^2 (Y_{i,n,eq,l})^2\right] = \sum_{l} \left((S_{m,l}^{-1})^2 \cdot E\left[Y_{i,n,eq,l}^2\right] \right) = \sum_{l} \left((S_{m,l}^{-1})^2 \cdot var(Y_{r,n,eq,l}) \right)$$
(8.28b)

In which the operator var(X) returns the variance of X.

Cross terms

Recall from equation Equation 8.27 that the expected noise power per bin is equal to the sum of the expected values of two squared terms and two cross terms. In the cross terms, the expected value of a sum is taken. Since the expected value of a sum is equal to the sum of the expected values and multiplication by a constant can be taken outside of the expected value operator, the cross terms are equal to:

$$E\left[\sum_{k\neq l} R_{m,k}^{-1} R_{m,l}^{-1} Y_{r,n,eq,k} Y_{r,n,eq,l}\right] = \sum_{k\neq l} R_{m,k}^{-1} R_{m,l}^{-1} \cdot E\left[Y_{r,n,eq,k} Y_{r,n,eq,l}\right]$$
(8.29a)

$$E\left[\sum_{k\neq l} S_{m,k}^{-1} S_{m,l}^{-1} Y_{i,n,eq,k} Y_{i,n,eq,l}\right] = \sum_{k\neq l} S_{m,k}^{-1} S_{m,l}^{-1} \cdot E\left[Y_{i,n,eq,k} Y_{i,n,eq,l}\right]$$
(8.29b)

In which $E[Y_{r,n,eq,k}Y_{r,n,eq,l}]$ can be written as E[Z] with $Z = Y_{r,n,eq,k} \cdot Y_{r,n,eq,l}$. When two random variables are independent, the expectation of their products is the product of their expectations [26]. $Y_{r,n,eq,k}$ is constructed out of the DFT of the different Sample Sets within a Collection. The sampled noise (in this case the quantization error) is expected to be independent between different sample sets, since sample moments are mostly not at the same moment in time. Therefore, the output of the DFTs between different sample sets are expected to be independent. Simulations are performed to verify if the different DFT bins within the DFT of a Sample Set are independent. In the simulation, a random signal is sampled and it is checked if the $E[Y \cdot Y]$ is equal to $E[Y] \cdot E[Y]$. Although the mathematics and simulations are not completely understood, the result seem to show that $Y_{r,n,eq,k}$ and $Y_{r,n,eq,l}$ are independent. Due to time limitation, this has not been further investigated (and is added as a recommendation). Given this results, it is assumed that $E[Y_{r,n,eq,k} \cdot Y_{r,n,eq,l}] = E[Y_{r,n,eq,k}] \cdot E[Y_{r,n,eq,l}]$ such that

$$\sum_{k \neq l} R_{m,k}^{-1} R_{m,l}^{-1} \cdot E\left[Y_{r,n,eq,k} Y_{r,n,eq,l}\right] = \sum_{k \neq l} R_{m,k}^{-1} R_{m,l}^{-1} \cdot E\left[Y_{r,n,eq,k}\right] \cdot E\left[Y_{r,n,eq,l}\right] = 0$$
(8.30a)

$$\sum_{k \neq l} S_{m,k}^{-1} S_{m,l}^{-1} \cdot E\left[Y_{i,n,eq,k} Y_{i,n,eq,l}\right] = \sum_{k \neq l} S_{m,k}^{-1} S_{m,l}^{-1} \cdot E\left[Y_{i,n,eq,k}\right] \cdot E\left[Y_{i,n,eq,l}\right] = 0$$
(8.30b)

since $E[Y_{r,n,eq,k}] = E[Y_{r,n,eq,l}] = 0.$

Conclusion

Assuming that $Y_{r,n,eq,k}$ and $Y_{r,n,eq,l}$ are independent, it follows that

$$E\left[|H_m|^2\right] = 4 \cdot \sum_l \left((R_{m,l}^{-1})^2 \cdot var(Y_{r,n,eq,l}) \right) + 4 \cdot \sum_l \left((S_{m,l}^{-1})^2 \cdot var(Y_{r,n,eq,l}) \right)$$
(8.31)
Chapter 9: Noise Comparison: Theory, Simulations, and Measurements

This chapter presents the results of the model proposed in section 8.3, MATLAB simulations, and measurements performed with the Instrument. After the presentation of the results, the results are compared and discussed in section 9.4, the measurement method is discussed in section 9.5, and section 9.6 draws a conclusion.

This chapter uses the MR-DFT in a comparable manner as the Instrument, such that results can be compared. This includes that per beat period, 2 sample sets are samples: One samples 8 samples per beat period and the other 7. Both sample sets span 32 beat periods. All MR-DFT applications performed or referred to in this chapter are of this kind.

Four different cases are used to compare the results of the simulations, model, and measurements. These are:

- Unity: In this case noise or noisy signals are sampled out of a uniform distribution with a variance of 1/12. This a distribution that spans 1 (e.g. ranging [-0.5 0.5] or [0 1]) hence "unity".
- FS: FS refers to Full Scale. This case uses random signals which range over the maximum signal amplitude of the Arbitrary Waveform Generator (AWG). This is from -0.175 V to 0.175 V.
- 14 bits: This refers to the quantization error due to 14 bits quantization over the range [-1 1]. This is the resolution of the Analog Devices AD9248, the ADCs used in the Instrument. The quantization error in this case is assumed to be uniformly distributed ranging $[-2^{-14} 2^{-14}]$.
- 11 bits: This refers to the quantization error due to 11 bits quantization over the range [-1 1]. This is comparable to the ENOB value of the Analog Devices AD9248, the ADCs used in the Instrument¹. The quantization error in this case is assumed to be uniformly distributed ranging $[-2^{-11} \ 2^{-11}]$.

In case of simulations and measurements (section 9.2 and section 9.3), the simulations and measurements are performed 1000 times (unless otherwise stated) and 1000 MR-DFT spectra are calculated. All shown spectra are the result of a bin-wise average of the absolute value of all 1000 values retrieved for that output bin:

$$H_m = \frac{1}{N} \sum_{n=1}^{N} |H_{m,n}| \tag{9.1}$$

in which $H_{m,n}$ is the *m*-th bin of the n-th simulation run and N is the number of averaged runs (1000 unless otherwise stated).

To show the effect of quantization noise on the noise floor, the output of the MR-DFT of a quantized signal is subtracted from the output of the MR-DFT of the same non-quantized signal. The absolute value of difference per bin between the ideal and quantized version for every run is averaged over all runs in a bin-wise fashion:

$$\Delta H_m = \frac{1}{N} \sum_{n=1}^{N} |H_{m,n} - H_{quan,m,n}|$$
(9.2)

in which $H_{quan,m,n}$ is the *m*-th bin of the MR-DFT of the quantized signal of run n.

9.1 Model

Figure 9.1 present the effect of different signal and noise patterns on the output bins of the MR-DFT according to the noise model (presented in subsection 8.2.1).. The different figures in Figure 9.1 represent:

¹Due to the nature of the rounding operation used in the simulations, it is not possible to round to decimal numbers. So it is not possible to round to an ENOB of ~ 11.6 . Therefore, rounding to 11 bits has been used.

- A The expected output of a MR-DFT performed on a random input signal with a variance of 1/12. E.g. a random signal sampled out of a uniform distribution between 0 and 1 (unity-case).
- B The expected output of a MR-DFT performed on a random input signal with a variance of $0.35^2/2$. E.g. a random signal sampled out of a uniform distribution between -0.175 and 0.175 (FS case).
- C The expected effect on the noise floor of the output of the MR-DFT caused by quantizing a signal with 14 bits of resolution in the range of -1 to 1 (14 bits case).
- D The expected effect on the noise floor of the output of the MR-DFT caused by quantizing a signal with 11 bits of resolution in the range of -1 to 1 (11 bits case).

The found MR-DFT noise profiles are used to compare the model, simulations, and measurement in section 9.4.



Figure 9.1: The results of the model of noise floor characteristic of the MR-DFT for different inputs: A) The output of a MR-DFT of an random input with a variance equal to 1/12. For example the MR-DFT of a uniformly distributed signal between 0 and 1. B) The output of a MR-DFT of a random signal with a variance of $0.35^2/12$. For example the MR-DFT of a uniformly distributed signal between -0.175 and 0.175. C) The expected noise floor in the output of the MR-DFT due to quantization noise for 14 bits quantization. D) The expected noise floor in the output of the MR-DFT due to quantization.

9.2 Simulations

For simulating the noise performance of the MR-DFT, two different simulations are performed. Subsection 9.2.1 shows the impact of a sampled random signal on the noise floor of the output of the MR-DFT. Subsection 9.2.2 shows the impact of a quantized cosine on the noise floor of the output of the MR-DFT.

9.2.1 Sampled Random Signal and MR-DFT (Noise Floor) Behaviour

Figure 9.2 shows the results of simulations on the MR-DFT. The MR-DFT spectrum is simulated for random signals, quantized versions of the same random signals, and the difference between the two



Figure 9.2: The results of simulations of the MR-DFT and the effect on the noise floor. A) Absolute output of the MR-DFT performed on a random signal with every sample out of a uniform distribution between 0 and 1, averaged over 1000 runs. B) The 1000 random signals of A quantized by 11 and 14 bit resolution. C) The MR-DFT performed on the (expected) quantization error for 11-bits and 14-bits quantization. D) The data of B minus the data of A. Thus the output of the MR-DFT of the quantized versions of the random signal minus the MR-DFT of the (unquantized version of) random signal. E) The output of the MR-DFT performed on a uniformly distributed random signal between -0.175 and 0.175 (the maximum output range of the AWG).

foregoing spectra. For these simulation, random signals are sampled out of a uniform distribution within MATLAB. It is ensured that both Sample Sets contain the same value at common sample moments. The results are plotted in Figure 9.2. The different figures within Figure 9.2 represent:

- A The result of the MR-DFT on a signal sampled out of a uniform distribution ranging [-0.5 0.5].
- B In blue the MR-DFT of the signal of A quantized with 11 bits resolution. In orange the signal of A quantized with 14 bits resolution. The orange and blue waveform are on top of each other and are identical to the waveform of figure A. The quantization errors are very small compared to the signal and make no visible difference on this logarithmic scale.
- C This figure shows the results of MR-DFT simulations on the expected quantization error. The effect of the expected quantization error has been simulated by performing the MR-DFT on a random signal sampled out of a uniform distribution ranging [-LSB/2 LSB/2]. For the 11-bits case

(in blue), the samples are sampled from $U(-2^{-11}, 2^{-11})^2$. For the 14-bits case (in orange), the samples are sampled from $U(-2^{-14}, 2^{-14})$.

- D This figure shows the effect of quantization, but from a simulation point of view. D shows the plot of A subtracted from the plots of B (according to Equation 9.2).
- E This figure shows a plot that is similar to figure A, expect that the samples are from a uniform distribution ranging [-0.175 0.175], the FS-case. This figure is added for comparison of measurements to the simulations.

Plot C and D of Figure 9.2 are identical. This leads to the conclusion that the quantization error indeed has a uniform distribution between -LSB/2 and LSB/2 for a random input signal.

9.2.2 Quantized Cosine and MR-DFT Noise Floor Behaviour

This section presents the results of simulation of the MR-DFT performed on a sampled and quantizing a cosine. To only show the effect of the quantization, the result of the MR-DFT on the non-quantized cosine is subtracted from the MR-DFT of the quantized cosine (according to Equation 9.2).

The cosine used in this chapter has an amplitude of 1^3 , is sampled with 8 and 7 samples per beat period and has a frequency of 1/8-th of frequency of the 8-samples-per-beat-period sample clock. The data of 32 beat periods is used to perform the MR-DFT upon (g = 32). This is equivalent to the measurement setup with the Instrument with two sample clocks at 64 MHz and 56 MHz. In that case the cosine has an equivalent frequency of 8 MHz.

Quantized Cosine Results

All results are shown in Figure 9.3:

- A The output of the MR-DFT performed on a 8 MHz equivalent (ideal) cosine.
- B The output of the MR-DFT performed on a 11 bits quantized version (in blue) of the ideal cosine and a 14 bits quantized version (in orange) of the ideal cosine. The bins that have no value, have an absolute value equal or close to 0. On a logarithmic scale this is equivalent to $-\infty$. MATLAB does not plot those points.
- C The effect of 11 bits quantization on the output spectrum of the MR-DFT.
- D The effect of 14 bits quantization on the output spectrum of the MR-DFT.

The sample rates are an integer multiple of the signal frequency. This makes that the assumption that quantization error follows a uniform distribution and is independent is not correct. The quantization errors become dependent and correlations show-up. The spikes shown in Figure 9.3 are a result of this. This effect can be suppressed by introducing (random) sample clock jitter for all samples. The next section shows the effect of sample clock jitter on the correlation of the quantization error.

Introducing Jitter

To suppress the spikes and to comply with the assumption of uniformly distributed quantization errors, jitter is introduced to all sample moments. The introduced jitter is uniform distributed between $-0.0005 \cdot \pi$ radians and $0.0005 \cdot \pi$ radians⁴. In the equivalent setup in which the cosine has a frequency of 8 MHz, this

²The 11 bits quantization range from -1 to 1 such that the $LSB = 2/2^{11}$ or $LSB/2 = 2^{-11}$.

 $^{^{3}}$ Note that the amplitude of the cosine doesn't matter for investigating the effect of quantization as long as the full range of the signal (cosine) is quantized by at least 7 bits (one of the requirements for assuming uniformly distributed quantization noise).

⁴This is the jitter for the 64 MHz clock. The 56 MHz clock has a uniformly distributed jitter between $-0.0005 \cdot 7/8 \cdot \pi$ and $0.0005 \cdot 7/8 \cdot \pi$ such the time displacement of the clock edges of both clocks is in the same range.



Figure 9.3: The results of the MR-DFT on an 8 MHz equivalent cosine with an amplitude of 1. A) Result of the MR-DFT performed on the ideal cosine. B) Results of the MR-DFT performed on the cosine quantized by 14 bits and 11 bits precision. For bins with no data point, the bin content was equal to zero (or $-\infty$ on a logarithmic scale). Note that the orange plot is on top of the blue one. C) The effect of 11 bits quantization on the output spectrum of the MR-DFT. D) The effect of 14 bits quantization on the output spectrum of the MR-DFT.

is equivalent to uniform distributed jitter between -3.9 and 3.9 ps for both sample clocks. The results are shown in Figure 9.4:

- A The MR-DFT of a cosine sampled by sample clocks with jitter. The noise floor shaping can be clearly seen and is caused by random sample errors due to the jitter.
- B The MR-DFT of a quantized cosine sampled by sample clocks with jitter. In blue 11 bits quantization is used. In orange 14 bits quantization is used.
- C Zoomed in version of B.
- D The effect of only quantization noise on the noise floor. This is calculated by subtracting the plot of A from the plots of B (in accordance with Equation 9.2). The difference between the two plots is approximate 18 dB, which is equal to the ratio of the variances of the quantization error caused by 11 bits quantization and the variance of the quantization error caused by 14 bits quantization $(10 \cdot log\left(\frac{LSB_{11}^2/12}{LSB_{14}^2/12}\right) = 18.06)$. A bit of peaking is already observed in the blue plot.

The figures show that the spikes have disappeared. So the addition of some extra randomness makes that the assumption of uniformly distributed quantization error is valid. However, the noise floor is now not only caused by the quantization error but also by the effect of jitter on the samples. Comparing Figure 9.4 C and D shows that the largest part of the noise floor is not caused by the quantization error but by jitter.

The current model does not include jitter. The method of section 8.2 and section 8.3 can be used to construct a model keeping both jitter and the quantization error into account. Due to time limitations this analysis has not been performed and is added as a recommendation.

There are two remaining questions. The first question is if the used jitter value is realistic and the right distribution is used for the jitter. The short answer is: probably not. Some short measurement have





Figure 9.4: The effect of sampling jitter on the MR-DFT of a cosine. A) Output of MR-DFT on jittered cosine. B) Output of the MR-DFT on the cosine of A quantized by 11 bits resolution (in blue) and 14 bits resolution (in orange). C) Zoomed in version of B. D) The effect of only the quantization noise for 11 bits quantization resolution (in blue) and 14 bits quantization resolution (in orange). These plots are calculated by subtracting the plot of A from the plots of B.

been performed on the jitter of the sample clocks. Although the results of the measurements are not completely understood, the random part of the jitter is around 17 ps. Also it is expected that the jitter follows a normal-like distribution. However, it is hard to validate this since there is not a lot known about the jitter specs of the clock sources available within the Intel Cyclone V FPGA. Due to time limitations no (extra) measurements on the sample clock jitter have been performed.

To investigate the effect of different jitter values on the output of the MR-DFT, a sweep over 4 jitter values have been performed. In all 4 cases, the jitter is sampled out of a uniform distribution but the range differs. The ranges are: +/-39 ps, +/-3.9 ps, +/-0.39 ps, and +/-0.2 ps. The results are shown in Figure 9.5. The 4 different plots show the MR-DFT output for the 4 jitter values. The blue plots show the noise floor for 11 bits case and the orange show the noise floor for the 14 bits case. The plots show that the energy moves from the noise floor to the spikes for smaller values of the jitter. This is especially the case for the 11 bits quantization and in lesser amount for the 14 bits quantization.

This supports that for small or no jitter, the assumption that the quantization noise is uniformly distributed is not true. Also, that the 14 bits case is less effected by jitter is in line with the uniformly distributed quantization error assumption. Due to a higher resolution, smaller jitter values still result in large enough deviations in the sampled value such that the quantization errors are (kind of) random.

The second remaining question is if the noise shaping within measurements is caused by a combination of jitter and quantization error and thus that the assumption that the quantization error is uniformly distributed is valid. It cannot be concluded that when no spikes are observed, this is solely caused by jitter. Also other processes (like thermal noise) might be able to add enough "randomness" to make sure that the quantization error becomes (kind of) uniformly distributed even though the sample frequency is an integer multiple of the signal frequency.

Which process(es) cause the scrambling of the sampled values is hard to say. In the case of the measurement setup, it might be caused by noise generated by the Arbitrary Waveform Generator, ADCs, thermal noise here and there, irradiation from the ether, and so on. Taking all these noise sources into account when determining the MR-DFT output noise floor characteristics is hard since the statistical properties of most noise sources are unknown.



Figure 9.5: The effect of quantization noise and sample clock jitter on the MR-DFT noise floor for different values of jitter. A) Sample clock jitter uniformly distributed between +/-39 ps. B) Sample clock jitter uniformly distributed between +/-3.9 ps. C) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed between +/-0.39 ps. D) Sample clock jitter uniformly distributed

9.3 Measurements

The measurements presented in this section, are performed by the Instrument (introduced in Chapter 4 and Chapter 5) and measurement setup (introduced in section 6.1). Two different signals are measured: Uniformly distributed random signal and a sine wave. The first measurement is to investigate if the effect of the random signals on the noise floor can be measured, the second measurement is to investigate if the effect of the quantization error can be observed in the noise floor. Note that the ADC inputs of the ADA daughter board have an input low-pass filter with a corner frequency at ~ 100 MHz. A wide band input signal (like uniformly distributed noise), is filtered before being sampled. The effect of this filter on the statistical independency of samples is unknown. It is therefore expected that it is not possible to correctly measure wide band noise signal. Since no better measurement methods are available it still is given a try.

9.3.1 Wideband Noise

Wideband noise is generated and measured by the Instrument. The noise is generated by generating a random signal out of a uniform distribution between -0.175V and 0.175V. Every sample of the signal (containing 7.168 GS/s) is another value out of the uniform distribution. The resulting signal can therefore be approached as wideband noise. In this case, instead of averaged over 1000 measurements, 10000 measurements are used for averaging. The result is shown in Figure 9.6.

The expected pattern in the noise floor is visible but it looks like there is some filtering visible in some of the "levels" (e.g. from bin 32 to 63 and from bin 64 to 95). This might be caused by the input filters of the Instrument. To exclude the effect of the input filter from the measurements, the noise/signal should be injected after the input filters. This is tried in the subsection 9.3.2.



Figure 9.6: The results of the MR-DFT on wideband noise measurements.

9.3.2 Quantization Noise in Sine Measurements

To produce a better measurement on the noise shaping properties of the MR-DFT, the noise should be injected after the input filters of the Instrument. One way to do so is to asses the quantization noise of the Instrument. In this section it is tried to measure this noise. This is done by measuring a sine with an frequency of 8 MHz and an amplitude of 0.175 V. The absolute value of the MR-DFT of 1000 measurement runs is averaged. The phase of the sine of every run is randomly selected out of a uniform distribution between 0 and 2π . The results are plot in Figure 9.7.



Figure 9.7: The MR-DFT of measurements on a sine at a frequency of 8 MHz.

First of all the 8 MHz peak has a power of -13.57 dB. This is equivalent to an amplitude of 210 mV (175 mV expected). Measurements with a scope on the output of the AWG has shown that the AWG outputs a bit higher output voltage than expected. This can account for this amplitude of 210 mV. Since the amplitude spans more than 7 bits of the Instrument's resolution, this extra bit of output voltage does not influence the expected quantization error (and actually makes the assumption that the quantization error is uniformly distributed a bit better).

Figure 9.7 shows some peaking at multiples of 8 MHz. It might be harmonic distortion of either the AWG or the Instrument. However due to the appearance and nature of the peaks, it is expected that this is caused by the quantization error (as in Figure 9.3). The amount of available jitter within the Instrument seems not to be enough to scramble the signal samples enough such that the quantization error becomes independent.

The noise floor does show shaping of the noise floor in the expected shape. The main source of this noise

is hard to guess. As discussed in subsection 9.2.2 and shown in Figure 9.4, the effect of jitter might be larger than that the quantization error. Also other noise sources (e.g. AWG output noise, thermal noise, irradiation) play a role in this measurement. Since the characteristics and/or spectra of the other noise sources are unknown, it is impossible to pinpoint the sources and their contributions.

9.4**Discussion of Results**

The simulated, measured, and theoretical value of the different "noise floor levels" (the flat parts of the noise floor of about 32 bins wide) are shown in Table 9.1. The meaning of the different phrases used in the table, can be found in the caption of the table.

	Simulations			Theory			Measurements		
flat part $\#$	Unity	\mathbf{FS}	11 bits	Unity	\mathbf{FS}	11 bits	Unity	\mathbf{FS}	Qnoise
1	-30,23	-39,1	-90,05	-34,87	-43,99	-95,08	not measured	-54,2	-94,74
2	$-25,\!69$	-34,74	-85,85	-29,9	-39,02	-90,11	not measured	$-51,\!66$	-90,58
3	-24,34	$-33,\!48$	-84,41	$-27,\!64$	-36,76	-87,85	not measured	-51,05	-88,44
4	-23,9	-32,99	-84,12	-26,17	$-35,\!28$	-86,37	not measured	-51,02	-87,05
5	-24,05	-33,26	-84,21	-26,79	-35,91	-87	not measured	-51,21	-87,45
6	-25,02	-34,39	-84,9	-28,55	$-37,\!67$	-88,76	not measured	-51,74	-89,25
7	-27,17	-36,33	$-87,\!66$	-31,56	$-40,\!68$	-91,77	not measured	$-53,\!24$	-92,12

Table 9.1: The power values (in dB) of the different "noise levels" (the flat parts in the noise floor). In this table "flat part #" represents the flat part, numbered from left to right. "unity" refers to the case in which uniform distributed noise with a variance of 1/12 is used. "FS" refers to the full-scale case, in which uniform distributed noise between -0.175 and 0.175 is used. "11 bits" refers to the case that should be equivalent to the effect of 11 bits quantization on the noise floor (uniform distributed noise between -2^{-11} and 2^{-11}). "Qnoise" refers to the measurements of Figure 9.7, in which the effect of quantization noise of 11.6 ENOB is investigated. All values are in dB.

Table 9.1 shows two interesting points. First of all, the measurements do not match the simulations and theory. Secondly, there is a difference between the simulations and the model but this is (more or less) a constant factor for every flat part over all three cases. The differences are calculated and presented in Table 9.2.

Difference simulations and mode								
flat part #	Unity	\mathbf{FS}	11					
1	4,64	4,89	5,03					
2	4,21	$4,\!28$	4,26					
3	3,3	3,28	$3,\!44$					
4	2,27	$2,\!29$	$2,\!25$					
5	2,74	$2,\!65$	2,79					
6	3,53	$3,\!28$	$3,\!86$					
7	4,39	$4,\!35$	4,11					

lel

Table 9.2: This table shows the difference between the noise floor levels of the simulations and the model in dB. For the meaning of the different phrases used in this table, refer to the caption of Table 9.1.

Table 9.2 shows that the three simulations differ from the model by a certain factor for ever flat part. The model does not fit the simulations and it is expected that this is the result of a wrong assumption or wrong assumptions. The most obvious culprit is the assumption made in subsection 8.2.1 at page 69 about the expected value of the cross terms in the expected noise power in the output bins of the MR-DFT. This assumption is only loosely supported by simulations. Therefore, proving mathematically the expected value of the cross terms is added as recommendation.

The results of the measurements are hard to interpret. Although the expected shape in the noise is

observed (see Figure 9.6), it is questionable how trustworthy the measurements are since a large part of the broadband noise is filtered by the input filter of the ADC channels. A better technique would be to inject the noise after the input filter or remove the input filter all together.

It is tried to fix this by injecting a cosine at 8 MHz and use the quantization error as random noise to inspect the noise shaping properties of the MR-DFT. Although some noise shaping is observed (see Figure 9.7, also a lot of spikes are observed. It is expected that these spikes are caused by correlation between the quantization errors (the quantization errors are not independent) since the assumption that the quantization errors are uniformly distributed does not hold for this case (because the sample frequencies are an integer multiple of the signal frequency). It seems that the available jitter does not scramble the sample values enough to cause enough randomness in the quantization errors such that the errors become independent.

9.5 Discussion of Measurement Method

For the measurements, an 8 MHz signal is generated and measured. After finishing the measurements, it was realised that the 8 MHz signal might not be the best choose. Firstly, as already discussed, because the sample frequencies are an integer multiple of the signal frequency (although this did reveal the effect of jitter on the correlation peaks and noise floor). Secondly, because within the used MR-DFT algorithm (with 8 and 7 samples both spanning 32 periods), only the 32th bin of \vec{X}_8 is necessary to reconstruct the 8 MHz input signal. Effect like gain mismatch between the two ADC channels, differences in jitter between the two sample clocks, and alike do not show up in the results of both the measurements and simulations. It is expected that the effect of the former phenomena does not significantly influences the noise floor shape but it has been observed that gain mismatch between the two channels can result in spikes in the output of the MR-DFT.

A better frequency for the signal of the performed measurements in this chapter could have been, for example, 40 MHz or 48 MHz. These frequencies need elements of both \vec{X}_8 and \vec{X}_7 to be reconstructed since it is larger than the Nyquist frequency of both sample clocks. Also, the sample clocks are no integer multiple of 40 MHz or 48 MHz. And 40 MHz and 48 MHz are multiples of 8 MHz. The period of the 8 MHz signal fits an integer multiple times within the total length of the two Sample Sets. This results in no discontinuity in the sampled data when assuming the sampled data is repeated indefinitely (an assumption done when performing a DFT transform on a time limited dataset). In turn, no artefacts are observed in \vec{X}_8 and \vec{X}_7 due to a discontinuity. Since 40 MHz and 48 MHz are multiples of 8 MHz, this also holds for 40 MHz and 48 MHz.

After performing all the measurements, it was noticed in the datasheet of the used ADCs (AD9248) that the ADCs use the half of the sample clock period to sample the input signal. However it has not been found in the datasheet in which half of the clock period the input stage is in track-mode and in which half it is in hold-mode. If the track-mode is from the rising edge to the falling edge this might pose a problem. The input voltage level is then "measured" at the falling edge of the sample clocks (assuming no input filtering, infinitely fast sample capacitor charging, and so on). In the current design it is assumed that the data is sampled at the rising edge of the sample clocks instead of the falling edge. This has implications for the sample moments and the output of the MR-DFT. It should be clear when the samples are taken and if this is at the falling edge, the output of the DFTs should be compensated for this difference in phase (before performing the MR-DFT) or the sample clocks should be inverted. This is added as recommendation.

9.6 Conclusion

Although the same kind of pattern is visible in the noise floor of the MR-DFT of simulations, model, and measurements, they do not match.

The wrong assumption about the distribution of the quantization error also plays a role in the matching between the model and simulations of the quantization error. This can be suppressed by adding jitter but this introduces a new noise source which is not taken into account within the model. In case of a random input signal, the model and simulation still do not match. The difference between the different levels in the noise floor, seems to be a constant factor per level. The results suggest that a factor has been missed within the model. It is expected that a wrong assumption has been made somewhere in the process. The most obvious culprit is the assumption made in subsection 8.2.1 at page 69 about the expected value of the cross terms in the expected noise power in the output bins of the MR-DFT. This should be investigated further.

A contributor to the mismatch between the measurements and the model and simulations might be the presence of other noise sources. They are not taken into account but might have a far larger influence than the effect of quantization (like jitter). Furthermore, the input filter of the ADCs is certain to influence the measurements of wide band noise but it is unknown in which manner.

Regarding the measurement method, a better signal frequency can be used (e.g. 40 MHz or 48 MHz). Furthermore, it should be thought of how to directly measure the quantization error instead of other effect (like jitter, input filters, and other noise sources) or incorporate the effect of other noise sources within the model. A possible solution for the input filter problem could be removing the ADC input filters.

Part VI

Discussion, Conclusion, and Recommendations

Chapter 10: Discussion, Conclusion, and Recommendations

Recently, a so-called Multi-Rate Discrete Fourier Transform algorithm is developed at the Integrated Circuit Design group of the University of Twente. The algorithm enables the reconstruction of information from higher Nyquist zones (Nyquist zones 2+) by utilising multiple sample rates to sample the same waveform. This work presents the theoretical background of the Multi-Rate Discrete Fourier Transform as well as the impact of non-idealities (such as random noise, quantization noise, and jitter) on the accuracy of the output of the Multi-Rate Discrete Fourier Transform. This chapter summarises the main results, conclusions, and sums up the recommendations.

10.1 Summary

Humankind generates more and more data every year and strives to increase wireless communication throughput significantly with the instruction of techniques as 4G, 5G, and LoRaWAN. In a transceiver, a Power Amplifier (PA) amplifies the modulated data signal before it is transmitted through the ether. For efficiency reasons, switched mode PA, like class-E devices, are used. Class-E PAs are susceptible for PVT spread and load impedance variations, which reduces linearity and efficiency of the PA [2]. The state of operation of class-E PAs can be deduced from the information in the first 3 or 4 harmonics of the output of the PA [2] [7]. The information in the harmonics can be used as input to control the class-E PA such that it operates in a safe and efficient operation region. The next generations wireless communication protocols (like 5G) utilise the mm-Wave spectrum (e.g. 5G uses a frequency band around 28 GHz [8] [9]). The first couple of harmonics end up the around hundred GHz or higher. Measuring these frequency components is not possible in toady's IC technology. New technology must be developed to tackle this measurement problem. A possible candidate is the Multi-Rate Discrete Fourier Transform (MR-DFT) algorithm (Chapter 1).

The MR-DFT utilises data sets of the same waveform sampled at various sample rates. Due to the use of multiple sample rates, the aliasing patterns of the frequency information in the higher Nyquist zones differs for the distinct sets of sampled data. The different aliasing patterns can be converted to a system of linear equations which is used to reconstruct (a part of) the harmonic information in the higher Nyquist zones (Chapter 2 and 3).

In this work, a measurement setup is presented that is able to capture different set of sampled data at multiple sample frequencies. The measurement instrument, based around an SoC platform (an FPGA with an ARM core), utilises two ADCs to sample the same signal (generated by an Arbitrary Waveform Generator) at 64 MS/s and 56 MS/s. In the time (beat period) that the first ADC samples 8 samples, the second ADC samples 7 samples. In total, the sample sets span 32 beat periods, resulting in sample sets of 256 samples and 224 samples respectively. With these sample sets, frequency information up 55.75 MHz is recovered with a bin width of 250 kHz (Chapter 4 and 5).

The measurement setup is used to measure (mainly) QAM16 constellations in the different MR-DFT output bins. The constellations are send both in parallel in 191 MR-DFT output bins as well as one per output bin at the time. The QAM constellations can be reconstructed. For small signal amplitudes (maximum symbol amplitude of a few millivolt to 1 millivolt for the whole constellation) the constellation quality degrades fast and there seems to be some systematic error which results in deformation of the constellations (Chapter 6 and 7).

Also the impact of statistical processes (like noise, quantization error, jitter, and similar) on the MR-DFT is analysed. A model for the impact of the quantization error on the noise floor is presented¹ and compared to simulations and measurements. The results show that a similar shape in the noise floor of the model, simulations, and measurements is observed but the absolute level of the noise floor differs

 $^{^{1}}$ The model and approach to come to the model can be easily adapted for other statistical processes like noise, jitter, and so on.

between the model, simulations, and measurements (Chapter 8 and 9).

10.2 Discussion

The results of the measurements on QAM constellations are promising. They support that the MR-DFT works as an algorithm on real-life measured data and show that it is possible to reconstruct frequency components in higher Nyquist zones. In measurements at low symbol amplitude (couple of millivolts) deformation in the constellations are observed which seem to have a systematic cause. Although the cause of the deformations is unknown, it is not suspected that the deformations are a consequence of the MR-DFT algorithm. Therefore, this does not decrease the credibility of the MR-DFT algorithm to measure the first couple of harmonics of class-E Power Amplifiers. Furthermore, in the intended applications (the output of a class-E PA) decent signal levels are expected, such that symbol amplitudes in the millivolt range are not expected to occur.

The only possible problem at the moment might be the ability to acquire all Sample Sets in an acceptable time limits. The Sample Sets for the MR-DFT can be acquired in two manners: parallel or sequential. In the parallel manner, the various Sample Sets are measured by their own set of hardware (mixer/sampler and ADC) at the exact same moment in time.

The use of modulations in symbols (e.g. QAM) offers also the possibility to measure the various Sample Sets in a sequential in time manner, sharing the measurement hardware. When symbol x occurs, it is measured to capture the first or first couple of Sample Sets. The remaining Sample Sets can be captures at next occurrence of symbol x. As long as the repetition rate of symbol x is high enough (longer than the time it takes for PVT spread and load impedance variations to change the class-E PA operation state, it is expected that this is in the order of a millisecond [6]) the sequential measurement method works. If QAM constellations with a high number of symbols (e.g. QAM256) are transmitted, the symbol repetition rate might be so low that the PVT spread and load impedance variation are faster than the time expected time between two occurrences of symbol x.

If sampling of all Sample Sets in a sequential manner takes too much time, also a hybrid form between sequential and parallel sampling can be thought of. In such a system (based on the circuit of Figure 2.3), for every Sample Set a passive mixer/sampler (and the necessary clock generation circuitry) is present while the baseband ADC is shared. In this manner, all Sample Sets can be sampled over the same part of the waveform while only one ADC is necessary². This enables to perform all sampling on the same symbol/signal on the expense of chip area (compared to the sequential method) or on the expense of time (compared to the parallel method).

10.3 Conclusion

The main findings in the work presented in the report, concerning the implementation and verification of the MR-DFT, are summarised following the research questions stated in the introduction. The three research questions are:

- 1) Can a mathematical framework be constructed that describes the MR-DFT?
- 2) Is it possible to create a measurement setup that measures higher Nyquist zones via the MR-DFT?
- 3) What is the impact of noise on the MR-DFT and can a model be constructed and verified?

The first research question is positively answered. A mathematical framework to represent the MR-DFT in a mathematical context, is presented. It shows how the MR-DFT works, what requirements are present for signals and Sample Sets. It also presents a method to come up with a minimal selection of rows of the Harmonic Matrices to still perform the MR-DFT using a matrix inversion (instead of

 $^{^{2}}$ The area of the clock circuitry and passive mixer is of the same order of magnitude as the chip area of the baseband ADC[27]. So area wise, it makes sense to omit the repetition of the baseband ADCs.

using the Moore-Penrose pseudo-inversion on the complete set of Harmonic Matrices). This leads to the conclusions that an all including mathematical frameworks is presented.

The second research question is also positively answered. A measurement setup is constructed that can measure the Sample Sets needed for an MR-DFT. The setup, based upon a FPGA/ARM platform using an ADC expansion board, is able to measure QAM16 and QAM64 constellations in all MR-DFT bins simultaneously, thus constructing frequency information from higher Nyquist zones. This leads to the conclusions that a measurement setup is presented that can be used in combination with the MR-DFT.

The answer to the last research question is not so decisive as the answer to the other two question. The theory related to the impact of noise and disturbances on the MR-DFT is presented. Using a statistical approach, a model of the noise floor shaping properties of the MR-DFT due to quantization errors is presented. This approach can be generalised and used for all kind of noise sources and disturbances.

This model has been compared to simulations and measurements. Unfortunately the model, simulations, and measurements do not match although the noise floor profiles look similar. The difference between the model and simulations seems to be a constant factor. It is suspected that one of the assumption, used to construct the model, is not correct.

This leads to the conclusion that the presented model is not fully correct and used measurement setup does not work for characterising the effect of noise on the MR-DFT. Extra work is needed to find out where the deviation originates from and how proper measurements can be performed.

Summarising, a model for the impact of the noise on the MR-DFT is presented and compared to simulations and measurements. The model, simulations, and measurements do not match but the observed noise floors have similar profiles. This answers the third research question.

10.4 Recommendations

The recommendations, following from this study, are ordered in four different sections: mathematics, QAM measurements, Instrument, and recommendations related to noise and disturbances (including theory, simulations, and verification via measurements).

Mathematics

- The mathematics are now discussed in an "engineering-manner". A lot of the mathematics is based on assumptions and validated by simulations. Although a lot of results seem to fit the simulations, this is not a water-proof approach. The mathematical, and especially statistical, knowledge of the author is limited so it is expected that the theory and model are not free of errors (which is supported by simulations). Therefore, it is recommended to have a thorough look at the mathematics by a mathematician such that assumptions can be proven right or wrong (like the applicability of the Lyapunov Central Limit Theorem or the expected value of the cross term in the expected value of the output power in the MR-DFT due to statistical processes).
- In a normal DFT, it is possible to measure the frequencies from i.e. BW to 2BW by using aliasing/folding if the rest of the spectrum is empty. It is expected that similar measurements can also be performed with the MR-DFT. It can be investigated if it is possible to reconstruct frequencies higher $M \cdot \Delta f$ by using Harmonic Matrices that describe the folding patterns of this frequency information for the different Sample Sets.

QAM Measurements

• In the QAM measurements, deterministic deformations of the constellations are observed. The cause is unknown (and could range from problems with the measurement setup to problems with the MR-DFT algorithm itself). Before proper application of the MR-DFT is possible, the cause should be known or it should be proven that this deterministic deformation does not show up in the application.

Instrument

• It is unclear if the track-mode (sample-mode) of the input stage of the Analog Devices AD9248 is from the rising sample clock edge to the falling edge or from the falling edge to the rising edge. This has implications for the sample moment and therefore for at least the phase of the different DFT bins. This also affects the MR-DFT. The exact sample moment of the Analog Devices AD9248 should be clarified and, if necessary, data processing should be adapted to the result.

Noise and Disturbances

- Come up with a clear method that enables to measure noise/random signals to verify the models and simulations. A possible way could be by physically removing the ADC input filters at the ADA daughter board but also a completely different measurement setup might be of use.
- Investigate the effect of jitter on the output of the MR-DFT and incorporate this noise source within the noise analysis.
- The output of the MR-DFT is influenced by the gain and phase mismatch of the current Instrument but it is unknown how. Within the intended application, in which the same hardware is used to sample all Sample Sets, this is expected to be less of a problem. It would still be informative to come up with a method or theory to assess the effect of phase and gain mismatch between the different Sample Sets.
- The effect of the expected value of the cross terms within the expected noise power per bin in the output of the MR-DFT is based on assumptions which are only loosely validated by simulations. It is expected that this might be the cause of discrepancy between the model and the simulations. Therefore, it is recommended to prove what the effect of the expected value of the cross terms in the expected noise power per output bin has on the output noise floor of the MR-DFT.

Bibliography

- Peter Ffoulkes. insideBIGDATA: Guide to use of big data on an industrial scale. https: //insidebigdata.com/white-paper/guide-big-data-industrial-scale/, 2017. Retrieved on 26th February 2019.
- [2] M. Huiskamp, A. Annema, and B. Nauta. A delay spread cancelling waveform characterizer for rf power amplifiers. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 65(12):1834–1838, Dec 2018.
- [3] Kevin R Boyle, Yun Yuan, and Leo P Lighart. Analysis of mobile phone antenna impedance variations with user proximity. *IEEE Transactions on antennas and propagation*, 55(2):364–372, 2007.
- [4] Ettore Lorenzo Firrao, Anne-Johan Annema, Frank E van Vliet, and Bram Nauta. On the minimum number of states for switchable matching networks. *IEEE transactions on circuits and systems I:* regular papers, 62(2):433–440, 2015.
- [5] Steven M Bowers, Kaushik Sengupta, Kaushik Dasgupta, Benjamin D Parker, and Ali Hajimiri. Integrated self-healing for mm-wave power amplifiers. *IEEE Transactions on Microwave Theory* and Techniques, 61(3):1301–1315, 2013.
- [6] Youngchang Yoon, Hyoungsoo Kim, Hyungwook Kim, Kun-Seok Lee, Chang-Ho Lee, and James S Kenney. A 2.4-ghz cmos power amplifier with an integrated antenna impedance mismatch correction system. *IEEE Journal of Solid-State Circuits*, 49(3):608–621, 2014.
- [7] Paolo Colantonio, Franco Giannini, Ernesto Limiti, and Valeria Teppati. An approach to harmonic load-and source-pull measurements for high-efficiency pa design. *IEEE Transactions on Microwave* theory and Techniques, 52(1):191–198, 2004.
- [8] Lili Wei, Rose Qingyang Hu, Yi Qian, and Geng Wu. Key elements to enable millimeter wave communications for 5g wireless systems. *IEEE Wireless Communications*, 21(6):136–143, 2014.
- [9] Editorial Team everything RF. 5g frequency bands. https://www.everythingrf.com/community/ 5g-frequency-bands, February 2018. Retrieved at 26th April 2019.
- [10] Maikel Huiskamp. Sherpas user committee meeting 4. Unpublished presentation.
- [11] Marcin Dryjanski and Moe Rahnema. From LTE to LTE-Advanced Pro and 5G. Artech House, September 2017.
- [12] Harri Holma, Antti Toskala, and Jussi Reunanen. LTE small cell optimization: 3GPP Evolution to Release 13. John Wiley & Sons, 2016.
- [13] C. E. Shannon. Communication in the presence of noise. Proceedings of the IEEE, 72(9):1192–1201, Sep. 1984.
- [14] Marcel JM Pelgrom. Analog-to-digital conversion. Springer, 2nd ed edition, 2013.
- [15] Altera Corperation. Cyclone v device handbook volume 1: Device interfaces and integration. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/ cyclone-v/cv_5v2.pdf, October 2019. Retrieved at 14th October 2019.
- [16] Altera Corperation. Cyclone v device handbook volume 2: Transceivers. https://www.intel.com/ content/dam/www/programmable/us/en/pdfs/literature/hb/cyclone-v/cv_5v3.pdf, October 2018. Retrieved at 14th October 2019.
- [17] Intel Corporation. Avalon[®] interface specifications. https://www.intel.com/content/dam/www/ programmable/us/en/pdfs/literature/manual/mnl_avalon_spec.pdf, October 2019. Retrieved at 14th October 2019.

- [18] Altera. Altera phase-locked loop (altera pll) ip core user guide. https://www.intel.com/content/ dam/www/programmable/us/en/pdfs/literature/ug/altera_pll.pdf, June 2017. Retrieved at 14th October 2019.
- [19] Altera Corperation. Cyclone v device overview. https://eu.mouser.com/pdfdocs/ Altera_cv_51001.pdf, December 2012. Retrieved at 16th November 2019.
- [20] comm.evm. https://www.mathworks.com/help/comm/ref/comm.evm-system-object.html. Retrieved at 23th of April 2019.
- [21] Accuracy and precision. https://en.wikipedia.org/wiki/Accuracy_and_precision, October 2019. Retrieved at 11th November 2019.
- [22] Brad Brannon. Sampled systems and the effects of clock phase noise and jitter. Analog Devices Application Note AN-756, 2004.
- [23] Samir S Soliman and 1935 Srinath, Mandyam D. Continuous and discrete signals and systems. Upper Saddle River, N.J.; London : Prentice-Hall International, 2nd ed edition, 1998. Previous ed.: 1990.
- [24] Gil Engel. The power spectral density of phase noise and jitter: Theory, data analysis, and experimental results. Analog Devices Application Note AN-1067, 2010.
- [25] N Gallagher. Quantizing schemes for the discrete fourier transform of a random time-series. IEEE Transactions on Information Theory, 24(2):156–163, 1978.
- [26] Product distribution. https://en.wikipedia.org/wiki/Product_distribution, October 2019. Retrieved at 27th October 2019.
- [27] Maikel Huiskamp. Private conversation with maikel huiskamp on 11th november 2019. Private conversation.
- [28] Dennis C Gilliland. Integral of the bivariate normal distribution over an offset circle. Journal of the American Statistical Association, 57(300):758–768, 1962.
- [29] Werner R. W. ScheinHardt. Kansrekening en statistiek inf/tel. Reader, 4 2010. Reader for bachelor course "Probability Theory and Statistics" for Electrical Engineering at University of Twente.

Appendix A: Algorithm to Count Number of Solvable Linear Equations

This appendix provides two MATLAB implementations of algorithms that return the number of reconstructable harmonics by counting the number of unsolvable linear equations of the system of linear equation due to shared sample moments between (multiple) Sample Sets withing a Collection and subtracting this from the total number of samples in the Collection. The first algorithm was the first try to tackle the problem. The implementation is not very elegant. The second one is a more elegant solution and probably more efficient.

A.1 Algorithm 1

Algorithm 1 is shown in Listing A.1. The first algorithm assumes two input variables, a vector with the length of the Sample Sets (c-values) and the number of periods the Sample Sets are sampled (g-value). A maximum of 4 Sample Sets can be provided to this function.

The algorithm works as follows. It first determined the sample times of the different Sample Sets (lines 15-16, 25, 38) normalised to the sample times of the first given Sample Set. Note that the sample times are rounded. This is necessary to convert a certain float format within MATLAB to another float format. Although not completely understand why and how, it turns out that is necessary for the *intersec* function to return the right output. Via the *intersect* function, it is determined what the common sample moments are between the second and the first Sample Sets (line 17) and this number is counted (line 20). The output of this count is the number of sampled values that do not add extra information since they are present multiple times.

Afterwards, the common sample moments between the third Sample Set and set one and two is determined if a third Sample Set is provided (lines 25-27), any values that occur more than once are filtered out (line 30), and the resulting common sample moments are counted and added to the previously found number of sampled values that add no extra information (line 32). In lines 39 to 48, the same is done for the fourth Sample Set (if provided, line 38).

In line 52 the maximum number of solvable equations is calculated and decreased by the number of found double sampled values and used to calculate the number of recoverable harmonics (including DC).

Listing A.1: Algorithm 1, used to determine the number of recoverable harmonics.

```
function [numOfHarms] = findNumOfHarms(sampleSets, periods)
2
   %UNTITLED5 Calculates number of output harmonics to be expected
3
   %
       SampleSets is a row vector containing the number of samples per
      sample
   %
       set. Periods contains the number of periods present
4
5
6
7
   ss_l = size(sampleSets, 2);
8
9
   if ss_1 > 4
       disp("ERROR: Collection contains more than 4 Sample Sets.");
11
       return
12
   end
13
14
   \% determine sample moments for sets 1 and 2 and find shared sample
      moments
   sm_1 = round(1:1:sampleSets(1,1)*periods, 3);
15
   sm_2 = round(1:sampleSets(1,1)/sampleSets(1,2):sampleSets(1,1)*periods,
16
       3);
   int_12 = intersect(sm_1, sm_2);
17
```

```
18
   \% Count the number of shares sample moment thus dependent equitions
      within
   % the system of linear equitions
19
20
   dependent_eq_cnt = size(int_12, 2);
22
   \% If a third sample set is present, determine sample moments and find
23
   % shared sample moments with set 1 and 2
24
   if ss_1 > 2
       sm_3 = round(1:sampleSets(1,1)/sampleSets(1,3):sampleSets(1,1)*
25
          periods, 3);
26
       int_23 = intersect(sm_2, sm_3);
       int_13 = intersect(sm_1, sm_3);
27
28
       \% Join int_23 and int_13 while filtering out values that occur
29
       % multipletimes.
       union_3 = union(int_13, int_23);
       % increase counter
       dependent_eq_cnt = dependent_eq_cnt + size(union_3, 2);
   end
34
36
   \% If a fourth sample set is present, determine sample moments and find
37
   \% shared sample moments with set 1 and 2 and 3
   if ss_1 > 3
38
       sm_4 = round(1:sampleSets(1,1)/sampleSets(1,4):sampleSets(1,1)*
          periods, 3);
40
       int_14 = intersect(sm_1, sm_4);
41
       int_{24} = intersect(sm_2, sm_4);
42
       int_{34} = intersect(sm_3, sm_4);
43
       \% Join int 14, int_24 and int_34 while filtering out values that
          occur
       % multipletimes.
       union_4_1 = union(int_14, int_24);
45
       union_4_2 = union(union_4_1, int_34);
47
       % increase counter
       dependent_eq_cnt = dependent_eq_cnt + size(union_4_2, 2);
48
49
   end
50
   % Calculate
   numOfHarms = ceil((sum(sampleSets*periods)-dependent_eq_cnt)/2);
```

A.2 Algorithm 2

Algorithm 2 is shown in Listing A.2. The first algorithm assumes two input variables, a row vector with the length of the Sample Sets (*c*-values) and the number of periods the Sample Sets are sampled (*g*-value).

The algorithms makes a vector containing all the sample moments of all the Sample Sets (lines 8-10). Note that the sample times are rounded. This is necessary to convert a certain float format within MATLAB to another float format. Although not completely understand why and how, it turns out without this rounding operations the statements at line 17 and 18 will not return the right output. At line 17, all unique Sample Sets are determined. The *bsxfun* function-call in combination with the inner sum, determines how often every unique sample moment appears in the *sample_moments*-vector. All sample moments occurring more than once do not add additional value since the sampled value are equal over all appearances. The one occurrence of the sample adding value is subtracted in the resulting vector. The resulting vector contains the number of times the sample moments occur more than once. The sum of this vector is the amount of redundant information within the Sample Set.

In line 20 the maximum number of solvable equations is calculated and decreased by the number of found double sampled values and used to calculate the number of recoverable harmonics (including DC).

Listing A.2: Algorithm 2, used to determine the number of recoverable harmonics.

```
function [numOfHarms] = findNumOfHarms4(sampleSets, periods)
2
   %UNTITLED5 Calculates number of output harmonics to be expected
3
   %
       SampleSets is a row vector containing the number of samples per
      sample
   %
       set. Periods contains the number of periods present
4
5
6
   sample_moments = [];
7
   for i=1:1:size(sampleSets,2)
8
       sample_moments = [sample_moments round(1:sampleSets(1,i)/sampleSets
9
          (1,1):sampleSets(1,i)*periods, 3)];
10
   end
11
   % next 2 lines are retrieved from
12
   % https://nl.mathworks.com/matlabcentral/answers/120062-how-to-count-
13
      the-occurrences-of-a-value-for-an-matrix.
   \% The first line determines the unique sample moments, the second line
14
15
   \% determiens how often they occure, subtracts 1 for and sums the
      results.
   \% In this way it determines only the the sample moments that occur more
16
       than once
17
   unique_sm=unique(sample_moments);
   dependent_eq_cnt = sum(sum(bsxfun(@eq,sample_moments(:),unique_sm))-1);
18
19
   numOfHarms = ceil((sum(sampleSets*periods)-dependent_eq_cnt)/2);
20
```

Appendix B: Frequency Locking Error and (MR-)DFT

In the proposed application of the MR-DFT, it was planned to generate multiple (evenly spaced) clock phases for sampling the output of a PA via a Delay Locked Loop (DLL). In a DLL, frequency locking problems are very probable. This means that the total delay over the delay chain of the DLL is a bit more or less than exactly, which results in a small deviation in the absolute sampling moment. This can be caused by e.g. mono-stability in phase detectors. The output clock phases are not exactly in lock with the input clock and therefore this is called a frequency lock error (FLE).

During the design stage, the clock phase generator has been changed from a DLL to a PLL, making frequency locking problems far less likely. Since the analysis in this subsection was already performed, it included in this report but moved from Chapter 9 to here.

The approach to investigate the influence of FLE on the MR-DFT will be to first investigate how the results of the DFT are influenced by an FLE and then to check how this influence will affect the result of the MR-DFT. Let start at the definition of the FLE. Lets say a cosine is sampled with 8 samples per period (a cosine signal has been chosen since in the given application a QAM/OFDM like signal will be generated, which is just a cosine with amplitude and phase modulation). Normally this means that:

$$x_8[n] = \cos(2\pi \frac{n}{8})$$
 for $n = 0, 1, ..., 7$ (B.1)

In case of a locking error, this becomes:

$$x_8[n](L_e) = \cos\left(2\pi(1+L_e)\frac{n}{8}\right) \text{ for } n = 0, 1, .., 7$$
(B.2)

The FLE L_e results in a leakage of frequency energy to different bins¹.

The FLE example can be generalised to a sampling of a certain frequency that is sampled by a clock signal with a frequency lock error and phase shift ϕ .

$$x_{c}[n](L_{e},\phi) = \cos(2\pi(1+L_{e})\frac{n}{c}+\phi) \text{ for } n = 0, 1, ..., N_{c}-1$$

= $\cos(2\pi(1+L_{e})\frac{n}{c}) \cdot \cos(\phi) - \sin(2\pi(1+L_{e})\frac{n}{c}) \cdot \sin(\phi)$ (B.3)

Taking the DFT of $x_c[n]$ results in:

$$X_{c,k}(L_e,\phi) = \frac{1}{N_c} \sum_{n=0}^{N_c-1} \left(\cos(2\pi(1+L_e)\frac{n}{c}) \cdot \cos(\phi) - \sin(2\pi(1+L_e)\frac{n}{c}) \cdot \sin(\phi) \right) \cdot e^{\frac{-2j\pi nk}{N_c}}$$

$$= \frac{\cos(\phi)}{N_c} \sum_{n=0}^{N_c-1} \cos(2\pi(1+L_e)\frac{n}{c}) \cdot e^{\frac{-2j\pi nk}{N_c}} - \frac{\sin(\phi)}{N_c} \sum_{n=0}^{N_c-1} \sin(2\pi(1+L_e)\frac{n}{c}) \cdot e^{\frac{-2j\pi nk}{N_c}}$$
(B.4)

Instead of using one complex summation to get the (complex) results of the DFT, the result (and thereby the summation) could also be split in real part $a_{c,k}(L_e, \phi)$ and an imaginary part $b_{c,k}(L_e, \phi)$ such that

¹The problem could also approached from a different view. Because of the FLE, the original signal does not exactly fit within one frequency bin and therefore will leak to surrounding bins in a sinc-like manner due to the use of a square windows in the DFT. In this section it is investigated where the energy will end-up. It turns out that a linearly increasing L_e results in leakage patterns within bins that are sinc-like shaped but do differ from a sinc.

$$a_{c,k}(L_e,\phi) = \frac{\cos(\phi)}{N_c} \sum_{n=0}^{N_c-1} \cos(2\pi(1+L_e)\frac{n}{c}) \cdot \cos(\frac{2\pi kn}{N_c}) - \frac{\sin(\phi)}{N_c} \sum_{n=0}^{N_c-1} \sin(2\pi(1+L_e)\frac{n}{c}) \cdot \cos(\frac{2\pi kn}{N_c}) = p(L_e,\phi) + q(L_e,\phi) b_{c,k}(L_e,\phi) = \frac{-j \cdot \cos(\phi)}{N_c} \sum_{n=0}^{N_c-1} \cos(2\pi(1+L_e)\frac{n}{c}) \cdot \sin(\frac{2\pi kn}{N_c}) + \frac{j \cdot \sin(\phi)}{N_c} \sum_{n=0}^{N_c-1} \sin(2\pi(1+L_e)\frac{n}{c}) \cdot \sin(\frac{2\pi kn}{N_c}) = r(L_e,\phi) + s(L_e,\phi)$$
(B.6)

For the sake of a clear derivation, $a_{c,k}(L_e, \phi)$ and $b_{c,k}(L_e, \phi)$ have been in split in $p(L_e, \phi) + q(L_e, \phi)$ and $r(L_e, \phi) + s(L_e, \phi)$ in Equation B.5 and Equation B.6. First $p(L_e, \phi)$ term of $a_{c,k}(L_e, \phi)$ is investigated. Afterwards $q(L_e, \phi)$, $r(L_e, \phi)$ and $s(L_e, \phi)$ are discussed as well.

Using the trigonometric identity that $2\cos(\alpha) \cdot \cos(\beta) = \cos(\alpha - \beta) + \cos(\alpha + \beta)$, and the identity $N_c = c \cdot g$ it follows that

$$p(L_e, \phi) = \frac{\cos(\phi)}{N_c} \sum_{n=0}^{N_c - 1} \cos(2\pi (1 + L_e) \frac{n}{c}) \cdot \cos(\frac{2\pi kn}{N_c})$$

$$= \frac{\cos(\phi)}{2N_c} \sum_{n=0}^{N_c - 1} \left[\cos(2\pi \frac{n}{c} (1 + L_e - \frac{k}{g})) + \cos(2\pi \frac{n}{c} (1 + L_e + \frac{k}{g}))\right]$$

$$= \frac{\cos(\phi)}{2N_c} \left(\sum_{n=0}^{N_c - 1} \cos(2\pi \frac{n}{c} (1 + L_e - \frac{k}{g})) + \sum_{n=0}^{N_c - 1} \cos(2\pi \frac{n}{c} (1 + L_e + \frac{k}{g}))\right)$$
(B.7)

Via the Dirichlet kernel (or Lagrange's trigonometric identities), it can be shown that

$$\sum_{k=0}^{n} \cos(kx) = \frac{\sin((n+\frac{1}{2}) \cdot x)}{2 \cdot \sin(\frac{x}{2})} + \frac{1}{2}$$
(B.8)

Substituting x for either $\frac{2\pi}{c}(1+L_e-\frac{k}{g})$ or $\frac{2\pi}{c}(1+L_e+\frac{k}{g})$, this results in

$$p(L_e, \phi) = \frac{\cos(\phi)}{2N_c} \left(\frac{\sin((N_c - 1 + \frac{1}{2}) \cdot \frac{2\pi}{c} (1 + L_e - \frac{k}{g}))}{2 \cdot \sin(\frac{2\pi}{2 \cdot c} (1 + L_e - \frac{k}{g}))} + \frac{1}{2} + \frac{\sin((N_c - 1 + \frac{1}{2}) \cdot \frac{2\pi}{c} (1 + L_e + \frac{k}{g}))}{2 \cdot \sin(\frac{2\pi}{c} (1 + L_e + \frac{k}{g}))} + \frac{1}{2} \right)$$
$$= \frac{\cos(\phi)}{4N_c} \left(\frac{\sin((N_c - \frac{1}{2}) \cdot \frac{2\pi}{c} (1 + L_e - \frac{k}{g}))}{\sin(\frac{\pi}{c} (1 + L_e - \frac{k}{g}))} + \frac{\sin((N_c - \frac{1}{2}) \cdot \frac{2\pi}{c} (1 + L_e + \frac{k}{g}))}{\sin(\frac{\pi}{c} (1 + L_e - \frac{k}{g}))} + \frac{\sin((N_c - \frac{1}{2}) \cdot \frac{2\pi}{c} (1 + L_e + \frac{k}{g}))}{\sin(\frac{\pi}{c} (1 + L_e - \frac{k}{g}))} + 2 \right)$$
(B.9)

The derivation of $p(L_e, \phi)$ ends at this point for now. The derivation of $q(L_e, \phi)$ goes in a similar fashion. First the trigonometric identity $2\sin(\alpha) \cdot \cos(\beta) = \sin(\alpha + \beta) + \sin(\alpha - \beta)$ as well as the definition of $N_c = c \cdot g$ are applied to $q(L_e, \phi)$:

$$q(L_e, \phi) = \frac{-\sin(\phi)}{N_c} \sum_{n=0}^{N_c-1} \sin(\frac{2\pi n}{c} (1+L_e)) \cdot \cos(\frac{2\pi nk}{N_c})$$

$$= \frac{-\sin(\phi)}{2N_c} \left(\sum_{n=0}^{N_c-1} \sin\left(\frac{2\pi n}{c} (1+L_e+\frac{k}{g})\right) + \sum_{n=0}^{N_c-1} \sin\left(\frac{2\pi n}{c} (1+L_e-\frac{k}{g})\right) \right)$$
(B.10)

(B.14)

Via Lagrange trigonometric identities it can be shown that:

$$\sum_{n=0}^{N} \sin(nx) = \frac{\sin((N+1)\frac{x}{2})}{\sin(\frac{x}{2})} \sin\left(\frac{N \cdot x}{2}\right)$$
(B.11)

Using this identity, it follows that:

$$q(L_e, \phi) = \frac{-\sin(\phi)}{2N_c} \left(\frac{\sin(\frac{N_c \pi}{c}(1 + L_e + \frac{k}{g}))}{\sin(\frac{\pi}{c}(1 + L_e + \frac{k}{g}))} \sin\left((N_c - 1)\frac{\pi}{c}(1 + L_e + \frac{k}{c})\right) + \frac{\sin(\frac{N_c \pi}{c}(1 + L_e - \frac{k}{g}))}{\sin(\frac{\pi}{c}(1 + L_e - \frac{k}{g}))} \sin\left((N_c - 1)\frac{\pi}{c}(1 + L_e - \frac{k}{c})\right) \right)$$
(B.12)

In a similar fashion, $r(L_e, \phi)$ and $s(L_e, \phi)$ of $b_{c,k}(L_e, \phi)$ can be evaluated to:

$$r(L_{e},\phi) = \frac{-j \cdot \cos(\phi)}{2N_{c}} \cdot \left(\frac{\sin(\frac{\pi N_{c}}{c}(1+L_{e}+\frac{k}{g}))}{\sin(\frac{\pi}{c}(1+L_{e}+\frac{k}{g}))}\sin\left((N_{c}-1)\frac{\pi}{c}(1+L_{e}+\frac{k}{g})\right) - \frac{\sin(\frac{\pi N_{e}}{c}(1+L_{e}-\frac{k}{g}))}{\sin(\frac{\pi}{c}(1+L_{e}-\frac{k}{g}))}\sin\left((N_{c}-1)\frac{\pi}{c}(1+L_{e}-\frac{k}{g})\right)\right)$$

$$s(L_{e},\phi) = \frac{-j \cdot \sin(\phi)}{4N_{c}}\left(\frac{\sin((N_{c}-\frac{1}{2}) \cdot \frac{2\pi}{c}(1+L_{e}-\frac{k}{g}))}{\sin(\frac{\pi}{c}(1+L_{e}-\frac{k}{g}))} - \frac{\sin((N_{c}-\frac{1}{2}) \cdot \frac{2\pi}{c}(1+L_{e}+\frac{k}{g})}{\sin(\frac{\pi}{c}(1+L_{e}+\frac{k}{g}))} + 2\right)$$
(B.13)

such that $a_{c,k}(L_e, \phi)$ and $b_{c,k}(L_e, \phi)$ end up as

$$a_{c,k}(L_e,\phi) = \frac{\cos(\phi)}{4N_c} \left(\frac{\sin((N_c - \frac{1}{2}) \cdot \frac{2\pi}{c} (1 + L_e - \frac{k}{g}))}{\sin(\frac{\pi}{c} (1 + L_e - \frac{k}{g}))} + \frac{\sin((N_c - \frac{1}{2}) \cdot \frac{2\pi}{c} (1 + L_e + \frac{k}{g})}{\sin(\frac{\pi}{c} (1 + L_e + \frac{k}{g}))} + 2 \right) - \frac{\sin(\phi)}{2N_c} \left(\frac{\sin(\frac{N_c \pi}{c} (1 + L_e + \frac{k}{g}))}{\sin(\frac{\pi}{c} (1 + L_e + \frac{k}{g}))} \sin\left((N_c - 1)\frac{\pi}{c} (1 + L_e + \frac{k}{c})\right) + \frac{\sin(\frac{N_c \pi}{c} (1 + L_e - \frac{k}{g}))}{\sin(\frac{\pi}{c} (1 + L_e - \frac{k}{g}))} \sin\left((N_c - 1)\frac{\pi}{c} (1 + L_e - \frac{k}{c})\right) \right)$$
(B.15)

$$b_{c,k}(L_e,\phi) = \frac{-j \cdot \cos(\phi)}{2N_c} \cdot \left(\frac{\sin(\frac{\pi N_c}{c}(1+L_e+\frac{k}{g}))}{\sin(\frac{\pi}{c}(1+L_e+\frac{k}{g}))} \sin\left((N_c-1)\frac{\pi}{c}(1+L_e+\frac{k}{g})\right) - \frac{\sin(\frac{\pi N_c}{c}(1+L_e-\frac{k}{g}))}{\sin(\frac{\pi}{c}(1+L_e-\frac{k}{g}))} \sin\left((N_c-1)\frac{\pi}{c}(1+L_e-\frac{k}{g})\right) \right) - \frac{j \cdot \sin(\phi)}{4N_c} \left(\frac{\sin((N_c-\frac{1}{2}) \cdot \frac{2\pi}{c}(1+L_e-\frac{k}{g}))}{\sin(\frac{\pi}{c}(1+L_e-\frac{k}{g}))} - \frac{\sin((N_c-\frac{1}{2}) \cdot \frac{2\pi}{c}(1+L_e+\frac{k}{g}))}{\sin(\frac{\pi}{c}(1+L_e+\frac{k}{g}))} + 2 \right)$$
(B.16)

It is now known how the DFT of a cosine signal is disturbed by an FLE. In a similar fashion it can be shown how the DFT of a sine is disturbed by an FLE (or add a phase shift of $-\pi/2$ for ϕ). The same technique can also be used for other signals but these might not result in analytic solutions.

The outcome of the MR-DFT could now be calculated by performing Equation 3.15. It seems not to be possible to provide an analytic expression for this since the form of MR-DFT-matrix differs significantly from for different sample sets.

Appendix C: Lyapunov CLT Simulations

This Appendix presents the results of simulations performed on the DFT of the quantization error to verify if statistical properties of the output of this DFT comply with the Lyapunov CLT.

In this simulation N random number, which are uniformly distributed within the full quantization range (-4 to 4), are sampled¹. This set of numbers is called the signal. The DFT of the signal is calculated and the DFT of a quantized version of the signal is calculated. The range of quantization is from -4 to 4 with 11 bit precision. The DFT of the original signal and quantized signal are subtracted in a bin-wise fashion and the result is plotted in a 2D histogram (real part on the x-axis and imaginary part on the y-axis).

The plots show that for a sufficient large number of runs (1000000 is used) and for a sufficiently large N, the distribution of the quantization error in most frequency bins tend to a circularly-symmetric complex normal distribution ("2D-Gaussian", see Appendix D for more information about the complex normal distribution). Figure C.1 shows an example of such a histogram. In the proceeding of this chapter, the circularly-symmetric complex normal distribution is referred to as complex normal distribution since in all discussed cases the complex normal distributions are circularly-symmetric (unless otherwise stated). The complex normal distribution is denoted by $CN(\mu, \sigma^2)$ and this refers to the definitions of Equation D.1. The effect of the quantization error on the DFT bins is equal to:

$$E_k = CN\left(0, \frac{1}{12 \cdot N} (LSB)^2\right) \tag{C.1}$$

for all k except the special cases discussed in the next subsection.

Given Equation C.1, implicitly the statistics of \vec{Y}_{noise} are known and Equation 8.1 can be used to calculate the impact of quantization noise on the output bins of the MR-DFT. section 8.3 presents a model for the noise floor of the MR-DFT using the results of Equation C.1 and Equation 8.1.

Special cases

Not for every bin and every N, the distribution converges in a complex normal distribution. In case N is not large enough, the Probability Density Function (PDF) of the quantization error in a bin does not yet converges to a Gaussian-like. For example N=3 bin 1, is a hexagonal cone. For N is 4, the resulting PDF already starts to resemble a Gaussian-like shape. However, for this set, the different rotated uniform distributions have an absolute angle of 90 degrees (or 180 degrees, the next paragraph discusses that case). This makes that the resulting PDF has a square base and is not circular/radial symmetric.

Another special case is those where the added uniform probabilities are rotated by $n \cdot \pi$. For an 8-point DFT, this is the case for bin 0 (since $e^{-j2\pi k \cdot 0/8} = e^0$) and 4 (since $e^{-j2\pi k \cdot 4/8} = e^{-j\pi \cdot k}$). In these cases, the absolute value² of the complex exponentials in Equation 8.12 always evaluate to 1. Therefore, the output of the DFT of the quantization error is reel; the complex normal distribution evaluates to a normal distribution for $k\% \frac{N_c}{2} = 0$ (in which % is the modulo operation):

$$CN(0,\sigma^2) \to N(0,\sigma^2)$$
 (C.2)

Figure C.2 shows an example of a simulation showing this effect is. Figure C.2 shows a histogram of the 4-th bin of an 8 point DFT of the quantization error of a random signal ranging from -4 to 4, quantized with 11 bits of precision, for 1000000 different runs.

 $^{^{1}}$ A uniform distribution was favoured over a normal distribution due to the known range. Using a normal distribution for the samples of the signal, results in possible signal values out of the quantization range. In that case, not only the quantization error distribution will be measured but the range limitations of the system. Also, the assumption that the quantization error is uniformly distributed requires that the signal utilises the full quantization range. In case of i.e. a normal distribution, most of the weight of the distribution is around zero while less is at the borders of the quantization range.

range. ²Since, in the end, the quantity of interest is the average noise power and not the momentarily noise voltage, the sign of the evaluated complex exponential is not of interest in comparing e^0 and $e^{-j \cdot k}$.



Figure C.1: (a): A 2D histogram of the result of the quantization error on bin 0 of an 8 point DFT. For this histogram 1000000 runs have been performed. (b): A cross section of the histogram of Figure C.1a. This histogram is created by making a histogram of the values of Figure C.1a with abs(y) < 0.0005.



Figure C.2: (a): A 2D histogram of the result of the quantization error on bin 1 of an 8 point DFT. For this histogram 1000000 runs have been performed. (b): A cross section of the histogram of Figure C.2a. This histogram is created by making a histogram of the values of Figure C.2a with abs(y) < 0.0005.

The second special case is the zeroth bin/DC bin. Since all complex exponentials within the DFT sum of the DC bin evaluate to 1, the DC bin is the sum of all quantization errors. Therefore, the normal CLT is applicable to this bin and the distribution of the quantization error within the DC bin converges to a normal distribution for sufficiently large N.

Conclusion

Giving these results, it seems likely that the distribution of the quantization error within a DFT bin converges toward a real (1D) or complex (2D) normal distribution. The mean and variance of the amplitude in DFT bin k, due the quantization error, are equal to:

$$E[E_k] = 0$$

$$var(E_k) = \sigma_{qe}^2 = \frac{1}{12 \cdot N} LSB^2$$
(C.3)

such that

$$E_k = CN\left(0, \frac{1}{12 \cdot N} (LSB)^2\right) \text{ for } k\% \frac{N_c}{2} \neq 0$$

$$E_k = N\left(0, \frac{1}{12 \cdot N} (LSB)^2\right) \text{ for } k\% \frac{N_c}{2} = 0$$
(C.4)

Note that the variance is equal for all bins, regarding if the bin is a special case or not. The power in a bin due to the quantization error is equal to

$$E[E_k^2] = var(E_k) + E[E_k]^2 = \sigma_{qe}^2 = \frac{1}{12 \cdot N} LSB^2$$
(C.5)

which is equivalent to the standard formula for the quantization error for an >6 bits ADC (see for example [14]).

The next section describes how these results effect the output of the MR-DFT.

Appendix D: Complex Standard Normal Distribution

The information in primarily based on information from:

- https://en.wikipedia.org/wiki/Complex_normal_distribution
- https://en.wikipedia.org/wiki/Multivariate_normal_distribution
- dr.ir. Mark Oude Alink

The Complex Normal distribution is a distribution that describes complex random numbers of which the real and imaginary part are jointly normal. Since only the complex standard normal distribution is used in this work (or a scaled version of it), only the (scaled) complex standard normal distribution is discussed. The complex standard normal distribution is a complex normal distribution of which the real and imaginary part are orthogonal normal distributions (independent/no correlation), have a zero mean and have a standard deviation of 0.5^{-1} :

$$CN(0,1) = N(0,0.5) + j \cdot N(0,0.5)$$
(D.1)

The complex normal distribution is a bivariant normal distribution. Therefore the Probability Density Function (PDF) can be formulated by:

$$f(r,i) = \frac{1}{2\pi\sigma_r\sigma_i\sqrt{1-\rho^2}} e^{-\frac{1}{2(1-\rho^2)}\left(\frac{(r-\mu_r)^2}{\sigma_r^2} + \frac{(i-\mu_i)^2}{\sigma_i^2} - \frac{2\rho(r-\mu_r)(i-\mu_i)}{\sigma_r\sigma_i}\right)}$$
(D.2)

in which r is the real part, i the imaginary part and ρ is the correlation between r and i. Note that in this research σ_r and σ_i are equal.

Relation to bivariant normal distribution and correlation

In the case of this work, the complex normal distribution is circularly symmetrical, which means that real and imaginary part are independent. The symmetry axis is around the origin. This implies that ρ is zero and that μ_r and μ_i are both zero. It turns out that in that case, the bivariant normal distribution is just a multiplication of the PDFs of the two normal distributions it is based on:

$$f(r,i) = \frac{1}{2\pi\sigma_r\sigma_i} e^{-\left(\frac{r^2}{2\sigma_r^2} + \frac{i^2}{2\sigma_i^2}\right)} = \frac{1}{\sqrt{2\pi\sigma_r^2}} e^{-\frac{r^2}{2\sigma_r^2}} \cdot \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{i^2}{2\sigma_i^2}} = f(r) \cdot f(i)$$
(D.3)

Scaling the complex standard normal distribution

In this research, a scaled version of the complex standard normal distribution is used. This section will investigate how the variance reacts to scaling of the complex standard normal distribution by a factor of a. Lets scale f(r, i) by a:

$$f(\frac{r}{a},\frac{i}{a}) = \frac{C}{2\pi\sigma_r\sigma_i} e^{-\left(\frac{r^2}{2a^2\sigma_r^2} + \frac{i^2}{2a^2\sigma_i^2}\right)}$$
(D.4)

¹The notation of a complex normal distribution is similar to the normal distribution when the "covariance matrix"matrix Λ is equal to 1. In that case, the notation for the complex normal distribution is: $CN(\mu, \sigma^2)$. In case of the complex standard normal distribution this becomes CN(0, 1).

An unknown factor C is added to make sure that the integral of over the PDF is equal to one. Furthermore if a particular (1D) distribution is scaled by a factor a, both the mean and standard deviation should be scaled by a. For a normal distribution this implies that:

$$a \cdot N(\mu, \sigma) = N(a\mu, a\sigma)$$

$$f(\frac{r}{a}) = \frac{1}{\sqrt{2\pi a^2 \sigma_r^2}} e^{-\frac{r^2}{2a^2 \sigma_r^2}}$$
(D.5)

In Equation D.3 is is shown that $f(r,i) = f(r) \cdot f(i)$. This implies that

$$f(\frac{r}{a}, \frac{i}{a}) = f(\frac{r}{a}) \cdot f(\frac{i}{a})$$

= $\frac{1}{\sqrt{2\pi a^2 \sigma_r^2}} e^{-\frac{r^2}{2a^2 \sigma_r^2}} \cdot \frac{1}{\sqrt{2\pi a^2 \sigma_i^2}} e^{-\frac{i^2}{2a^2 \sigma_i^2}}$
= $\frac{1}{2\pi a^2 \sigma_r^2 a^2 \sigma_i^2} e^{-\left(\frac{r^2}{2a^2 \sigma_r^2} + \frac{i^2}{2a^2 \sigma_i^2}\right)}$ (D.6)

Such that C is $1/a^4$. This shows that f(r/a, i/a) has the form of complex normal distribution. This implies that $a \cdot CN\left(0, (\sigma_r^2 + \sigma_i^2)\right) = CN\left(0, a^2(\sigma_r^2 + \sigma_i^2)\right)$. In case σ_r and σ_i are equal (which is the case in this work), $a^2\sigma = a^2\sigma_r/2 = a^2\sigma_i/2$. Note that the value of C could also be determined by noting that $\int \int f(r/a)f(i/a)drdi = \int \int f(r/a, i/a)drdi = 1$. [28] shows a technique to solve this integral, but it is out of the scope of this work to do so.

Appendix E: Squared Normal Distribution and its Relation to the Chi-Squared Distribution

This appendix elaborates on the statistical properties of the squared normal distribution and how that distribution is related to a Chi-squared distribution and the Gamma distribution. Used sources: [29], https://en.wikipedia.org/wiki/Chi-squared_distribution,

https://en.wikipedia.org/wiki/Normal_distribution#General_normal_distribution,

https://stats.stackexchange.com/questions/93383/square-of-normal-distribution-with-specific-variance/122864

The distribution X^2 is the distribution under investigation, in which X is a normal distribution:

$$X \sim N(0, \sigma^2) \tag{E.1}$$

The Chi-squared distribution with k degrees of freedom is a summation of k squared independent standard normal distributions and denoted as χ_k^2 :

$$\chi_k^2 \sim \sum_k (N(0,1))^2$$
 (E.2)

Since the distribution under investigation is not a standard normal distribution but a normal distribution, it should be transformed to a standard normal distribution.

$$\frac{X}{\sigma} \sim N(0,1) \tag{E.3}$$

Such that

$$\frac{X^2}{\sigma^2} \sim \chi_1^2 \tag{E.4}$$

or that

$$X^2 \sim \sigma^2 \chi_1^2 \tag{E.5}$$

Since the expected value of the Chi-squared distribution is equal to k and the variance to 2k, the expected value and variance of the squared normal distribution are:

$$E\left[X^2\right] = E\left[\sigma^2 \cdot \chi_1^2\right] = \sigma^2 \cdot k = \sigma^2$$
(E.6a)

$$var(X^{2}) = var(\sigma^{2} \cdot \chi_{1}^{2}) = \sigma^{2} \cdot 2 \cdot k = 2 \cdot \sigma^{2}$$
(E.6b)