



UNIVERSITÄT  
HEIDELBERG  
ZUKUNFT  
SEIT 1386

FAKULTÄT FÜR INFORMATIK

STUDIENGANG MEDIZINISCHE INFORMATIK MASTER



*ANALYSE UND EVALUATION DER WIRKUNG DES DOPPLER-EFFEKTS  
AUF VERSCHIEDENE MODULATIONSVERFAHREN DER GÄNGIGEN  
FUNKTECHNIK*

*Stefan Völker*

Heilbronn, 30. April 2015

**MASTER THESIS**

# **Master Thesis**

zur Erlangung des akademischen Grades

## **Master of Science**

an der

**Hochschule Heilbronn und  
Universität Heidelberg**

Fakultät für Informatik

Thema: Analyse und Evaluation der Wirkung des Doppler-Effekts auf verschiedene Modulationsverfahren der gängigen Funktechnik

Vorgelegt von: Stefan Völker

1. Prüfer: Prof. Dr. Oliver Kalthoff

2. Prüfer: Prof. Dr. Rolf Bendl

Abgabedatum: Heilbronn, 30. April 2015

## Abstract

Die Popularität der drahtlosen Datenübertragung wächst, seit dem der Standard 802.11 im Jahr 1997 vom Institute of Electrical and Electronics Engineers (IEEE) für die private Nutzung ausgegeben wurde, stetig an. Hierbei beschränken sich die meisten Anwendungen auf das Wesentliche: die Übertragung von Nutzdaten zwischen einem Access Point und verschiedenen Endgeräten. Der konventionelle Weg der Datenverarbeitung in drahtlosen Netzwerk-Modulen sieht vor, dass zunächst digitale Daten in analogen Verfahren moduliert werden. Nach der Übertragung als elektromagnetische Welle, werden die analogen Daten wieder von eingebetteten Systemen in einen digitalen Datenstrom übersetzt und zum Weiterverarbeiten weitergereicht.

Durch die steigende Rechenleistung von Computern gibt es auch andere Wege der Übertragung. Mit der Idee, den Datenstrom nicht analog weiterzuverarbeiten, sondern ihn in digitaler Form zu modulieren oder zu demodulieren entstanden die ersten ‚Software-Radios‘. Zu Beginn der drahtlosen Ära wurden Software-Radios zum Beispiel dazu genutzt, um Sicherheitslücken in Wi-Fi Übertragungen zu finden. Heutzutage finden Software-Radios immer mehr Anwendungen in wissenschaftlichen Bereichen abseits der üblichen Datenübertragung.

Als alternative Vorreiter gelten die Arbeiten der Universität Washington „Whole-Home Gesture Recognition Using Wireless Signals“ (Quifan et al., 2013a) und die des Massachusetts Institute of Technology „See Through Walls with Wi-Fi“ (Adib F. and Katabi D., 2013). Die erst genannte Anwendung beruht auf der Theorie des Doppler-Effekts für elektromagnetische Wellen und zeigt auf, dass es möglich ist, die Übertragungsverfahren des IEEE 802.11 zu überwachen und Bewegungen nahe den Antennen zu erfassen. Die Zweite zeigt auf, dass ähnliche Ergebnisse auch ohne den Doppler-Effekt sondern nur durch die gemessenen Reflexionen von Signalen erreicht werden können. Die Möglichkeiten dieser Techniken reichen von einer Gestensteuerung, die ohne zusätzliche Sensoren am Menschen auskommt, bis zum Visualisieren von Bewegungen hinter Wänden.

Der Gegenstand der vorliegenden Arbeit ist grundlegende Problematiken der Detektion des Doppler-Effekts in digitalen Modulationen aufzuzeigen. Hierfür werden Grundlagen der Empfangstechnik erläutert. Weiterhin wird der Einsatz aktueller Grafikkarten in Software-Radios zur Detektion des Doppler-Effekts untersucht.

## Inhalt

|  |     |
|--|-----|
| Abkürzungsverzeichnis .....  | III |
| 1. Vorwort .....   | 1   |
| 1.1. Beschreibung des Problems .....   | 1   |
| 1.2. Motivation und Hintergrund .....  | 2   |
| 1.3. Präzisierung beschriebener Modulationsverfahren .....                                     | 2   |
| 1.4. Limitierungen durch das verwendete Empfangsgerät .....                                    | 3   |
| 2. Theoretischer Hintergrund.....  | 4   |
| 2.1. Radartechnik .....  | 4   |
| 2.2. Der relativistische Doppler-Effekt .....  | 5   |
| 2.3. Software Defined Radios.....  | 6   |
| 2.3.1. Bandbreite .....  | 7   |
| 2.3.2. Bandbreitenauflösung.....   | 8   |
| 2.3.3. Empfangstechnik .....   | 9   |
| 2.3.4. Probleme bei der Durchführung der FFT .....   | 10  |
| 2.4. Betrachtete Modulationsverfahren .....  | 11  |
| 2.4.1. Nutzen von Modulationsverfahren .....   | 11  |
| 2.4.2. Analoge Modulationsverfahren .....  | 12  |
| 2.4.3. Konstellationsdiagramm .....  | 13  |
| 2.4.4. Binäre Phasenmodulation und Quadraturphasenumtastung .....                              | 14  |
| 2.4.5. Quadraturamplitudenmodulation.....  | 15  |
| 2.4.6. Orthogonales Frequenzmultiplexverfahren .....   | 16  |
| 2.4.7. Modulation und Demodulation am Beispiel des IEEE 802.11 .....                           | 17  |
| 2.5. Bewegungsdetektion in hochfrequenten Signalen .....                                       | 22  |
| 2.5.1. Vorteile analoger Schaltkreise zur Detektion .....                                      | 22  |
| 2.5.2. Bestehende wissenschaftliche Projekte.....  | 23  |
| 2.5.3. Problematik der Erkennung kleiner Frequenzänderungen in<br>breitbandigen Signalen ..... | 26  |
| 2.5.4. Der Einsatz von Grafikkarten in SDR Anwendungen.....                                    | 26  |
| 2.6. Mögliche Vorgehensweisen .....  | 29  |
| 3. Ausgeführte Methoden .....  | 31  |
| 3.1. Mathematische Betrachtung des Doppler-Effekts .....                                       | 31  |
| 3.2. Numerische Betrachtung.....   | 34  |

|        |  |    |
|--------|--|----|
| 3.3.   | Erstellung und Visualisierung von Signalen .....                   | 35 |
| 3.4.   | Verwendete Entwicklungsumgebung und Bibliotheken .....             | 38 |
| 3.5.   | Validierung der Methoden .....                                     | 39 |
| 3.6.   | Umsetzung einer IEEE 802.11 Simulation .....                       | 41 |
| 3.6.1. | Mögliche Nutzen der Simulation .....                               | 41 |
| 3.6.2. | Fragestellungen bei der Umsetzung .....                            | 41 |
| 3.6.3. | Umsetzung der Modulation und Demodulation .....                    | 43 |
| 3.7.   | Umsetzung einer Spektralanalyse auf einer GPU .....                | 45 |
| 3.7.1. | Idee der Umsetzung .....   | 45 |
| 3.7.2. | Durchführung der Umsetzung .....                                   | 45 |
| 3.7.3. | Erste Ergebnisse .....   | 46 |
| 4.     | Ergebnisse .....   | 49 |
| 4.1.   | Betrachtung des Doppler-Effekts in der IEEE 802.11 .....           | 49 |
| 4.2.   | Erstellung der Umgebung zur Signalverarbeitung auf einer GPU ..... | 49 |
| 4.3.   | Versuch der Simulation des Doppler-Effekts .....                   | 49 |
| 4.4.   | Umsetzung eines SDR mit GPU Unterstützung .....                    | 50 |
| 5.     | Diskussion .....   | 51 |
| 6.     | Ausblick .....   | 52 |
| 7.     | Abbildungsverzeichnis .....  | 53 |
| 8.     | Literaturverzeichnis .....   | 54 |
| 9.     | Tabellenverzeichnis .....  | 55 |
| 10.    | Appendix .....   | 56 |
| 10.1.  | Vergleich der Grafikkarten: .....                                  | 56 |
| 10.2.  | Matlab Umsetzung zur Betrachtung des Doppler-Effekts .....         | 57 |
| 11.    | Eidesstattliche Erklärung .....                                    | 58 |

## Abkürzungsverzeichnis

|      |   |   |
|------|---|---|
| ADC  | - | Analog-Digital-Converter                          |
| BPSK | - | Binary Phase Shift Keying                         |
| CPU  | - | Central Processing Unit                           |
| DSSS | - | Direct Sequence Spread Spectrum                   |
| DVB  | - | Digital Video Broadcasting                        |
| FFT  | - | Fast Fourier Transformation                       |
| FPGA | - | Field Programmable Gate Array                     |
| GPU  | - | Graphics Processing Unit                          |
| IEEE | - | Institute of Electrical and Electronics Engineers |
| ISM  | - | Industrial, Scientific and Medical                |
| LTE  | - | Long Term Evolution                               |
| MIMO | - | Multiple Input Multiple Output                    |
| OFDM | - | Orthogonal Frequency-Division Multiplexing        |
| QAM  | - | Quadrature Amplitude Modulation                   |
| SDR  | - | Software Defined Radio                            |

## **1. Vorwort**

In dieser Thesis wird der relativistische Doppler-Effekt für elektromagnetische Wellen in bereits etablierten Anwendungen, wie drahtlosen Netzwerken oder Long Term Evolution (LTE), untersucht und analysiert. Hieraus ergeben sich umfangreiche Themengebiete, welche physikalische Phänomene, mathematische Zusammenhänge und elektronische Gegebenheiten beinhalten. Hierfür wird der von der IEEE festgelegte Standard 802.11, der festlegt, wie ein drahtloses Netzwerk aufgebaut ist, erläutert und auf den Einfluss des Doppler-Effekts sowie dessen Auswertungsmöglichkeiten durch Software-Radios untersucht.

Zu Beginn dieser Arbeit wird ein grundlegender Einblick in die Thematiken und Funktionsweisen der einzelnen Soft- und Hardware Bausteine gegeben. Hierzu wird aufgezeigt, wie die induzierten Frequenzverschiebungen des Doppler-Effekts, in elektronischen Anwendungen detektiert werden können.

Im weiteren Verlauf wird untersucht, ob durch die Parallelisierung von Algorithmen auf Grafikkarten ein Beitrag zur Auswertung des Doppler-Effekts geleistet werden kann. Hiervon sollen zukünftige Entwicklungen zu diesem Thema, die auf der Auswertung des Doppler Effekts in drahtlosen Netzwerken basieren, profitieren können.

### **1.1. Beschreibung des Problems**

Der Doppler-Effekt wirkt auf die elektromagnetischen Wellen der drahtlosen Datenübertragung als Frequenzverschiebung. Er entsteht wenn sich der Sender oder Empfänger eines drahtlosen Signals bewegt. Bewegende Objekte innerhalb der Signalwege reflektieren das Signal und funktionieren somit als Zwischen-sender, die durch ihre Bewegungen den Doppler-Effekt induzieren. Hierbei definiert der Begriff Mehrwegeempfang, dass ein Signal über unendlich viele Wege vom Sender zum Empfänger gelangt. Also ist ein Empfänger in der Lage, Bewegungen innerhalb des Mehrwegeempfangs zu detektieren. Beschränkt wird der Mehrwegeempfang durch Abschwächungen im Signalverlauf. Diese treten sowohl bei längeren Distanzen auf, als auch durch Reflektion, Streuung, Absorption, Brechung und / oder Beugung. Abbildung 1 (Stretch J., v2.2) verdeutlicht die unterschiedlichen Interferenzen die auftreten können.

Der Doppler-Effekt kann somit als minimale Frequenzverschiebung mit geringer Einwirkung auf das Signal verstanden werden. Dass es möglich ist, diesen trotzdem zu messen zeigt die Umsetzung von WiSee (Quifan et al, 2012) der Universität Washington.

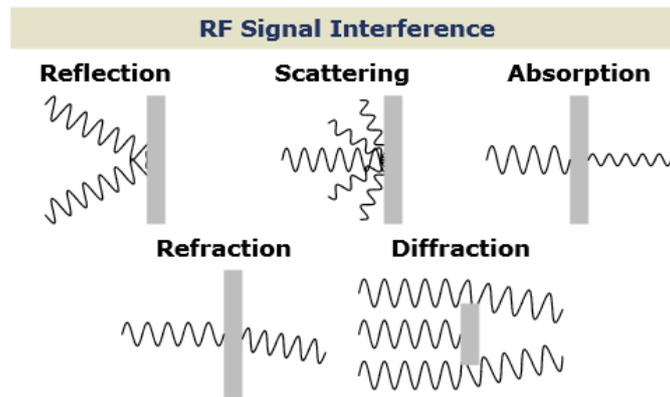


Abbildung 1: Interferenzen, bei elektromagnetischen Wellen

## 1.2. Motivation und Hintergrund

Bestehende Arbeiten zeigen, dass es möglich ist durch Bewegungen induzierte Änderungen in elektromagnetischen Wellen, zu messen und auszuwerten. Hierdurch kann beispielweise eine Gestensteuerung realisiert werden, die ohne Zuhilfenahme weiterer Sensoren an Körpern funktioniert. Weiterhin zeigt die Arbeit (Chen, V., 2006), dass es möglich ist anhand der Auswertung elektromagnetischer Wellen zwischen unterschiedlichen sich bewegendenden Personen zu unterscheiden. Die Realisierung eines derartigen Systems basiert in der Regel auf der Auswertung des Doppler-Effekts. Die vorliegende Arbeit interessiert sich für die Wirkung des Doppler-Effekts auf Methoden der drahtlosen Datenübertragung.

## 1.3. Präzisierung beschriebener Modulationsverfahren

Viele der heutigen Standards in der drahtlosen Signalübertragung setzen auf das orthogonale Frequenzmultiplex Verfahren (OFDM). Hierzu gehören Teile der physikalischen Schicht des IEEE 802.11, der terrestrische digitale Fernsehempfang, die Übertragung von Daten auf Mobiltelefone durch Long-Term-Evolution u.v.m.

Der Einsatz von OFDM bedingt zwangsläufig die Verwendung weiterer spezifischer Modulationsverfahren (siehe Abbildung). Diese unterscheiden sich durch die möglichen Übertragungsraten und die Anfälligkeit des Signals auf Störungen. Sie werden abhängig von der verfügbaren Signalqualität zwischen den Endgeräten eingesetzt, um Fehler in der Datenübertragung zu vermeiden.

| Modulations |            |             |
|-------------|------------|-------------|
| Scheme      | Modulation | Throughput  |
| DSSS        | DBPSK      | 1 Mbps      |
|             | DQPSK      | 2 Mbps      |
|             | CCK        | 5.5/11 Mbps |
| OFDM        | BPSK       | 6/9 Mbps    |
|             | QPSK       | 12/18 Mbps  |
|             | 16-QAM     | 24/36 Mbps  |
|             | 64-QAM     | 48/54 Mbps  |

Abbildung 2: Kombinationen von Modulationsverfahren

Abbildung 2 (Stretch J., v2.2) zeigt die verschiedenen Kombinationen von Modulationsverfahren und der daraus resultierenden Übertragungsgeschwindigkeit.

Das Direct Sequence Spread Spectrum (DSSS) Verfahren wird in dieser Arbeit nicht berücksichtigt, da die Untersuchung der Wirkung des Doppler-Effekts auf OFDM die aktuellen Anwendungen digitaler Datenübertragung abdeckt. Diese unterscheiden sich auf physikalischer Ebene durch die Verwendung in verschiedenen ISM (Industrie, Wissenschaft und Medizin) Frequenzbändern. Deshalb wird in diesem Projekt vorrangig die Auswertung der OFDM auf die Entstehung und Detektion von Doppler-Frequenzen untersucht.

#### 1.4. Limitierungen durch das verwendete Empfangsgerät

Die in der Arbeit präsentierten, aufgenommenen Signale werden mit einem Empfänger, der normalerweise zum Empfang von digitalem terrestrischem Fernsehen verwendet wird, aufgenommen. Über einen Software Hack können die empfangenen Signale direkt ausgelesen werden. Hierdurch wird die Funktionsweise eines voll funktionstüchtigen Empfängers realisiert. Limitierungen ergeben sich aus verschiedenen Kenndaten des Empfängers. Beispielsweise kann die maximal zur Verfügung stehende Anzahl an Abtastpunkten, die bei  $3,2 \cdot 10^6$  Abtastpunkten pro Sekunde liegt, genannt werden. Diese liegt bei professionellen Empfangsgeräten in der Größenordnung zwischen  $40 \cdot 10^6$  und  $100 \cdot 10^6$ .

## 2. Theoretischer Hintergrund

Um die Möglichkeit der Gestenerkennung durch den Doppler-Effekt in modulierten elektromagnetischen Wellen zu untersuchen bedarf es einiger Grundlagen bezüglich

- Radaranwendungen,
- der Physik des Doppler-Effekts,
- Software-Radios,
- Modulationstechniken und Signalverarbeitung.

### 2.1. Radartechnik

Die Wirkung und Auswertung des Doppler-Effekts wird als Verfahren in der Radartechnik verwendet. Hierzu dient der Dauerstrich-Radar, der eine Welle fester Frequenz in der Regel durch eine Richtantenne aussendet und die Reflektionen von Objekten wieder empfängt. Die feste Frequenz spielt hierbei eine zentrale Rolle, da der Doppler-Effekt besagt, dass sich die Frequenz der ausgesendeten Welle verändert, wenn sich das reflektierende Objekt bewegt. Durch die Verwendung von Richtantennen bzw. Parabolantennen wird die Richtung der Messung bestimmt.

In bestehenden Anwendungen wird das Problem der Richtungserkennung durch die Verwendung mehrerer Antennen gelöst. Somit können herkömmliche Antennen eingesetzt werden, um die Position der wieder empfangenen Reflektionen festzustellen. Das Verfahren läuft unter dem Begriff Multiple Input Multiple Output (MIMO).

Um das Prinzip des Dauerstrich-Radars auf gängige Methoden der drahtlosen Datenübertragung anwenden zu können, bedarf es einer Auswahl an Modulationsverfahren, die ähnlich dem Dauerstrich-Radar feste Frequenzen verwenden. Diese Bedingung ist für die in der IEEE 802.11 spezifizierten Verfahren der DSSS und OFDM erfüllt. Weitere Details hierzu sind im Kapitel „Betrachtete Modulationsverfahren“ zu finden.

## 2.2. Der relativistische Doppler-Effekt

Der Doppler Effekt beschreibt eine Frequenzänderung die zustande kommt, wenn sich der Sender oder der Empfänger einer akustischen oder elektromagnetischen Welle bewegt. Befindet sich ein Objekt im Signalweg zwischen Sender und Empfänger werden die Wellen daran reflektiert. Das Objekt gilt somit als passiver Sender, dessen Sendeenergie aus der Reflexion des Signals hervorgeht. Bewegt sich dieses Objekt tritt der Doppler-Effekt auf.

Viele Anwendungen der Radartechnik machen sich den Doppler-Effekt zu Nutzen. So finden sich, neben der hier beschriebenen Gestenerkennung in drahtlosen Netzwerken, viele radarspezifische Anwendungen des Doppler-Effekts hauptsächlich in der Flugnavigation oder für militärische Zwecke wieder.

Die allgemeine Formel zur Berechnung des relativistischen Doppler Effekts lautet:

$$f = f_0 \sqrt{\frac{1-\beta}{1+\beta}} \quad (2.2.1)$$

mit  $\beta = v/c$ . Für kleine Geschwindigkeiten (also  $\beta \ll 1$ ) kann in eine Potenzreihe in  $\beta$  entwickelt werden. Näherungsweise gilt dann:

$$f = f_0 (1 - \beta + \frac{1}{2}\beta^2). \quad (2.2.2)$$

Die Formel für den transversalen Doppler Effekt beschreibt eine Bewegung des Objekts quer zum Empfänger:

$$f = f_0 \sqrt{1 - \beta^2}. \quad (2.2.3)$$

Bei niedrigen Geschwindigkeiten kann die Gleichung wieder in einer Potenzreihe in  $\beta$  entwickelt werden:

$$f = f_0 (1 - \frac{1}{2}\beta^2). \quad (2.2.4)$$

Aus den Formeln (Halliday, 2. Auflage) geht hervor, dass der Doppler Effekt eine Frequenzänderung abhängig von der ursprünglichen Frequenz induziert. Hierbei gilt, dass der Einfluss des Effekts umso größer ist, je größer die Ursprungsfrequenz ist. Hierzu kann die Abhängigkeit zwischen Wellenlänge und Frequenz betrachtet werden:

Unter Betrachtung der oben aufgeführten Formeln kann angenommen werden, dass die Frequenz einer elektromagnetischen Welle ausschlaggebend für den Einfluss einer Bewegung auf diese Welle ist. Der Zusammenhang zwischen Frequenz und Wellenlänge kann der Gleichung (2.2.5) entnommen werden:

$$\lambda = \frac{c}{f} \quad (2.2.5)$$

Die Formel zeigt, dass die Wellenlänge  $\lambda$  abhängig von der Lichtgeschwindigkeit bzw. deren Ausbreitungsgeschwindigkeit und der Frequenz ist. Die Ausbreitungsgeschwindigkeit einer elektromagnetischer Wellen ist durch die Lichtgeschwindigkeit fest definiert. Deshalb gilt, dass die Wellenlänge mit steigender Frequenz kleiner wird.

Abschließend kann gesagt werden, dass der Doppler-Effekt bei größeren Frequenzen und somit kleineren Wellenlängen eine größere Ausprägung besitzt.

### 2.3. Software Defined Radios

Zum Detektieren des Doppler-Effekts in digitalen Modulationen, wird eine Empfangsanlage für hochfrequente Signale benötigt. Als solche werden Software Defined Radios (kurz: SDR) verwendet. Der Begriff SDRs beinhaltet die elektronischen Komponenten und die dazugehörige Software zum Signalempfang.

Die elektronischen Komponenten eines SDR bieten die Möglichkeit Signale bis zu einer von der Hardware begrenzten Frequenz zu empfangen.

Beispiele hierfür sind:

|                              | Verwendeter SDR  | HackRF         | bladerF           |      | Ettus USRP     |       |          |
|------------------------------|------------------|----------------|-------------------|------|----------------|-------|----------|
|                              |                  |                | x40               | x115 | B200           | B210  | N210     |
| <b>Empfangbares Spektrum</b> | 24 MHz – 1,7 GHz | 30 MHz – 6 GHz | 300 MHz – 3,8 GHz |      | 50 MHz – 6 GHz |       | 100 MHz  |
| <b>Bandbreite</b>            |                  | 20 MHz         | 28 MHz            |      | 61.44 MHz      |       | 100 MHz  |
| <b>Sample Größe (ADC)</b>    | 8 bit            | 8 Bit          | 12 Bit            |      | 12 Bit         |       | 14 Bit   |
| <b>Sample Rate</b>           | 3,2 MS/s         | 20 MS/s        | 40 MS/s           |      | 61,44 MS/s     |       | 100 MS/s |
| <b>FPGA Logik Elemente</b>   | keine            | keine          | 40k               | 115k | 75k            | 150k  | 50k      |
| <b>Kosten</b>                | ~20€             | 310€           | 390€              | 600€ | 615€           | 1000€ | 1760€    |

Tabelle 1: Liste verschiedener Software Radios

Die Tabelle<sup>1</sup> zeigt gängige Software-Radios. Aus der Tabelle ersichtlich ist, dass die meisten Software-Radios einen FPGA besitzen. Dies ist ein Baustein, der die empfangenen Signale verarbeiten kann. Die Programmierung eines FPGA kann vom Benutzer durchgeführt werden.

Der Unterschied zwischen einem SDR und herkömmlichen Komponenten zur drahtlosen Datenübertragung liegt in der Verarbeitung der Signale. Während beispielweise dedizierte WLAN Empfänger die Signalverarbeitung fest in der Elektronik implementiert haben, wird bei SDRs die Signalverarbeitung, wie Modulation und Demodulation, durch Software gesteuert und umgesetzt. Hieraus

<sup>1</sup> zum Teil entnommen aus: [www.taylorkillian.com/2013/08/sdr-showdown-hackrf-vs-bladerf-vs-usrp.html](http://www.taylorkillian.com/2013/08/sdr-showdown-hackrf-vs-bladerf-vs-usrp.html)

ergibt sich bei SDRs eine gesteigerte Relevanz der Datenverarbeitung und Datenspeicherung innerhalb der Signalverarbeitung.

Während WLAN Endgeräte nur Parameter der jeweiligen Übertragungsstandards und die Nutzdaten zur Verfügung stellen, wird mithilfe von SDRs das eigentliche Signal ausgelesen. Ein SDR ist somit einem Oszilloskop sehr ähnlich.

Da durch SDRs eine Visualisierung von hochfrequenten Signalen ermöglicht wird, ist zu untersuchen, von welchen Parametern eines SDRs die Detektion des Doppler-Effekts abhängig ist.

### **2.3.1. Bandbreite**

Der Begriff Bandbreite entstammt der Signalverarbeitung und Nachrichtentechnik. Zur Erklärung des Begriffs wird hier das menschliche Hörvermögen als Beispiel herangezogen.

Das menschliche Ohr ist in der Lage Frequenzen zwischen 16 Hz und 20 kHz wahrzunehmen. Dies entspricht der wahrnehmbaren Bandbreite. Bezieht man dies auf die Umsetzung einer Modulation, kann der Zusammenhang zur Technik einfach erklärt werden.

Bei der Übertragung von Musik auf analoge Radio-Empfänger wird von der Sendestation das eigentliche Nutzsignal – die hörbare Musik – auf eine höhere Frequenz moduliert. Hierdurch ändern sich die Frequenzen des Signals. Die Abbildung 3, zeigt einen Auszug aus SDRSharp, einem gängigen Programm für SDRs. Der Graph zeigt das empfangene Signalspektrum der Größe 3.2 MHz um eine Mittenfrequenz (roter Strich) von rund 91,1 MHz. Zur tatsächlichen Auswertung des empfangenen Signals ist eine Bandbreite von 250 kHz notwendig. Diese ist im Bild weiß hervorgehoben. Hieraus lässt sich durch Demodulation wieder das eigentliche Signal – die Musik – rekonstruieren.

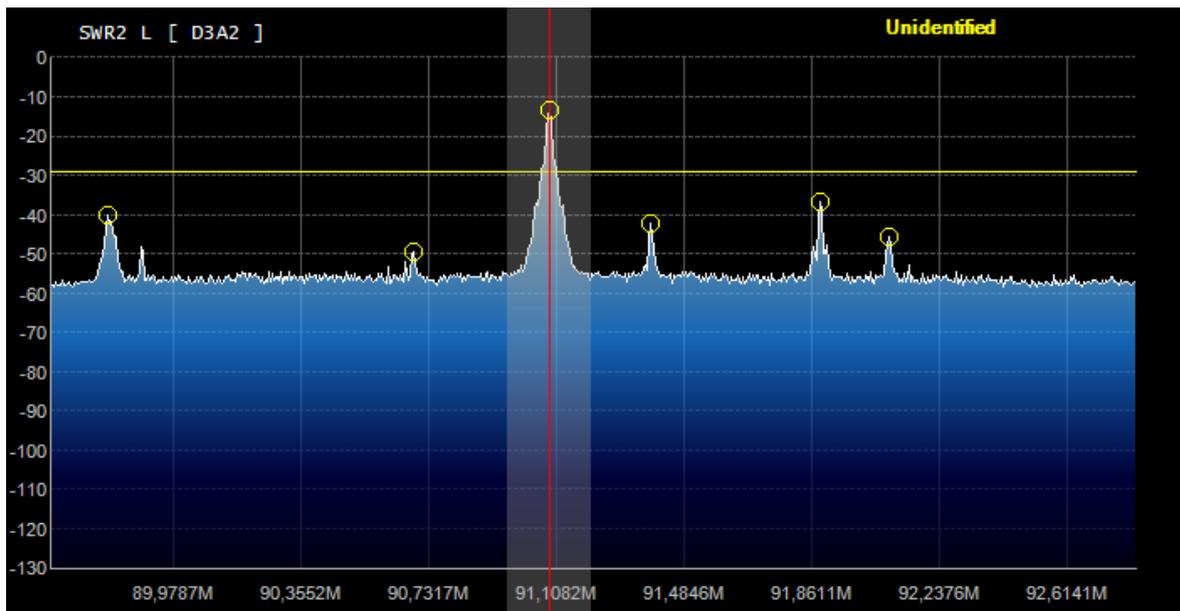


Abbildung 3: Auszug aus SDRSharp

Mithilfe von Modulationsverfahren können beliebige Signale auf beliebige Frequenzen moduliert werden. Um das Signal wieder rekonstruieren zu können, muss der Empfänger in der Lage sein, die Bandbreite des Signals mit genügender Auflösung zu empfangen.

### 2.3.2. Bandbreitenauflösung

Ein Qualitätskriterium des Empfangs ist die Bandbreitenauflösung (engl.: Resolution Bandwidth, RBW). Diese ist vorrangig festgelegt durch die verwendete Hardware sowie der Anzahl an Samples der durchgeführten Fourier Transformation (FFT). Mithilfe der FFT, werden Signale zwischen dem Zeitbereich und dem Frequenzbereich (Spektrum) transformiert.

Um die Auswirkung des Doppler-Effekts im Frequenzbereich effektiv darstellen zu können, muss eine möglichst hohe Bandbreitenauflösung erzielt werden. Hierdurch können die minimalen Frequenzänderungen im Spektralbereich sichtbar werden.

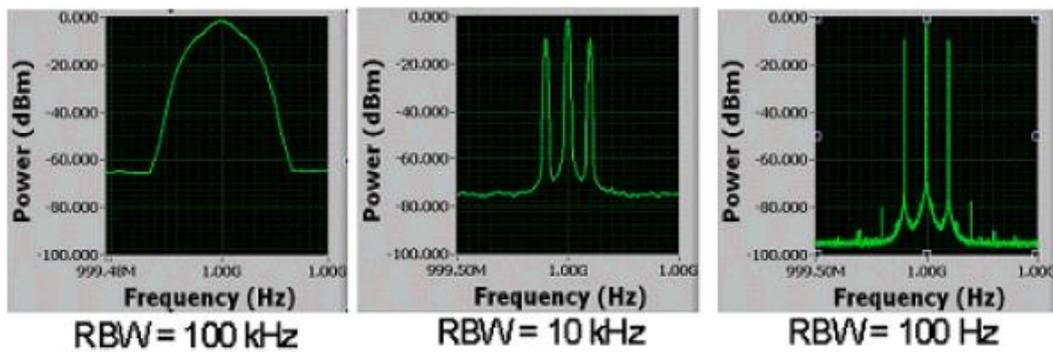


Abbildung 4: verschiedene Bandbreitenaufösungen desselben Signals

Abbildung 4 (NI-Tutorial-3983-en, 2014) zeigt die Bandbreitenaufösung eines Signals, das mit unterschiedlicher Anzahl an Samples in den Frequenzbereich transformiert wird. Ein Sample ist ein diskreter abgetasteter Wert des Signals. Hardwareseitig wird die maximal verfügbare Anzahl an Samples pro Sekunde durch den im Empfänger verbauten Analog- Digital-Wandler bestimmt.

Um eine möglichst hohe Bandbreitenaufösung zu erhalten, ist es notwendig, das Signal möglichst häufig abzutasten und mit einer möglichst großen Anzahl an Samples die FFT durchzuführen. Die höchste hierbei zu erreichende Bandbreitenaufösung beträgt 1 Hz.

Hieraus ergibt sich die theoretische Betrachtung, dass zum Beispiel die ursprüngliche Frequenz eines Signals von 2,4 GHz (nach IEEE 802.11) die Durchführung einer FFT mit mindestens 4,8 GS benötigt, um eine Auflösung von 1 Hz zu erreichen, sowie dem einhalten des Abtasttheorems gerecht zu werden. Dies würde eine extrem hohe Rechenleistung erfordern.

### 2.3.3. Empfangstechnik

In der Empfangstechnik gibt es zwei Arten von Empfängern. Dies sind der Geradeausempfänger und der Überlagerungsempfänger.

Um den Doppler-Effekt auf einer Übertragungsfrequenz von 2,4 GHz zu detektieren eignet sich der Geradeausempfänger nicht. Die Theorie des Geradeausempfängers besagt, dass das an der Antenne ankommende Signal verstärkt und direkt ausgelesen wird. Hierdurch entsteht der entscheidende Nachteil, dass die maximale abtastbare Frequenz dem Nyquist Kriterium unterliegt, welches besagt, dass die maximal rekonstruierbare Signalfrequenz der halben Abtastrate entspricht. Die Abtastrate ist direkt abhängig von der Konvertierungsrate des verwendeten Analog-Digital-Wandler der im Empfänger verbaut ist.

ADCs, die derzeit auf dem Markt sind, reichen bei ausreichender Präzision in ihrer Abtastrate lediglich in den unteren einstelligen Gigahertz Bereich. Somit ist eine Umsetzung, im Bezug zur Detektion des Doppler-Effekts im 2,4 GHz Band mit Geradeausempfängern nicht möglich, da die Übertragungsfrequenzen zu hoch sind.

Beim Überlagerungsempfänger, wird das empfangene Signal vor der Digitalisierung auf eine geringere Mittenfrequenz gemischt. Hierdurch bezieht sich das Abtasttheorem von Nyquist nicht auf die Übertragungsfrequenz, sondern auf die verwendete Mittenfrequenz. Die zentrale Rolle zur Detektion des Doppler-Effekts spielt hierbei die Art der Mischung des hochfrequenten Signals auf die Mittenfrequenz. Hieraus ergibt sich eine Übertragungsfunktion, die darauf zu untersuchen ist, ob kleine Frequenzänderungen erhalten bleiben.

Zur Detektion des Doppler-Effekt in der drahtlosen Datenübertragung im 2,4 GHz Band durch ein SDR ist ein Überlagerungsempfänger notwendig.

#### **2.3.4. Probleme bei der Durchführung der FFT**

Die Anzahl der Samples mit der die FFT durchgeführt werden muss um eine Bandbreitenauflösung von 1 Hz zu erreichen bezieht sich auf das empfangene Signal nachdem es auf die Mittenfrequenz des Überlagerungsempfängers gemischt wurde.

Betrachtet man hierzu die Abbildung 5, die mit einem SDR mit einer Sample Rate von 3.2 MS/s, einer Mittenfrequenz von 3,57 MHz und einer FFT mit 16382 Samples erstellt wird ergibt sich eine theoretische Bandbreitenauflösung von über 400Hz.

Stellt man in SDRSharp die Größe der FFT auf das Maximum von rund 4 Mega Samples wird eine theoretische Bandbreitenauflösung von ca. 2 Hz ist erreicht. Die Aktualisierungsrate des Programmablaufs verringert sich jedoch spürbar.

Die Probleme, die sich bei der Durchführung der FFT mit 4 MS ergeben, sind die hierfür notwendige Berechnungsdauer, als auch die verfügbare Anzahl an Samples, die durch den SDR zur Verfügung gestellt werden.

In der Dokumentation von SDRSharp wird eine maximale Auflösung von 16384 Abtastpunkten, unabhängig von verwendeten Empfängern empfohlen, um einen flüssigen Programmablauf zu gewährleisten.

Hieraus ergibt sich die Notwendigkeit, die FFT performanter zu berechnen und zu untersuchen, ob aktuelle Hardware die Durchführung größerer FFTs in geringerer Zeit ermöglicht.

## 2.4. Betrachtete Modulationsverfahren

In diesem Kapitel werden grundlegende Modulationsarten und deren praktischer Einsatz erläutert.

Zunächst werden die Grundlagen der Modulation erläutert, bevor ein genauerer Einblick in das Verfahren gegeben wird. Im IEEE 802.11g Standard werden folgende Modulationsverfahren zusammen mit OFDM genannt:

- Binäre Phasenmodulation (BPSK)
- Quadraturphasenumtastung (QPSK)
- Quadraturamplitudenmodulation (QAM)

### 2.4.1. Nutzen von Modulationsverfahren

Bevor der Einsatz der einzelnen Modulationen erläutert wird, soll dieses Kapitel die Frage klären, weshalb Modulationsverfahren notwendig sind und welchen Nutzen sie bringen.

Modulationsverfahren dienen in erster Linie dazu, verschiedenste Signale so zu verändern, dass daraus ein analoges Signal mit definierten Eigenschaften (Frequenz, Störanfälligkeit) entsteht, mit dem durch Demodulation das ursprüngliche Signal rekonstruiert werden kann. Häufig wird hierbei ein Signal geringerer Frequenzen auf höhere Frequenzen moduliert. Dies hat physikalische Gründe, wie das Ausbreitungsverhalten der Wellen bei unterschiedlichen Frequenzen oder dem Unterdrücken von Störeinflüssen.

Ein wichtiger Faktor der Modulation ist der Zuwachs an Trägerfrequenzen mit steigender Frequenz. Hierzu ein Beispiel:

Betrachtet man wieder das menschliche Hörvermögen, sowie die Modulation, wie sie beim Radio Empfang verwendet wird, kann man sich die Frage stellen, wie es möglich ist mehrere Radiosender durch ein und dasselbe Medium zu übertragen. Aus der Bandbreite der hörbaren Akustik, ergeben sich rund 20.000 verschiedene Frequenzen die jeder Radiosender übertragen muss. Wird dieses Signal auf eine Frequenz von 100MHz moduliert, gilt unter der Annahme, dass der Modulationsgrad einem Ganzen entspricht, dass rund um die Mittenfrequenz von 100MHz wieder 20.000 Abtastpunkte entstehen müssen. Das Signal, das ursprünglich von 16Hz bis 20kHz reichte, reicht nun von 99,99MHz bis 100,01MHz und besitzt ebenso viele Frequenzen wie das originale Signal. Rund um die Mittenfrequenz von 100MHz befinden sich weitere freie Bänder, auf denen andere Stationen senden können.

Der Nutzen von Modulationsverfahren kann wie folgt zusammengefasst werden:

- Anpassen der Übertragungsfrequenzen an räumliche Gegebenheiten um die Signalabschwächung möglichst gering zu halten.
- Konvertierung der Nutzdaten zwischen verschiedenen Übertragungsmedien (akustisch – digital – elektromagnetisch).
- Effizientere Nutzung des Übertragungsmediums, durch den Zuwachs an Trägerfrequenzen.

### 2.4.2. Analoge Modulationsverfahren

In Wi-Fi Anwendungen zeigt sich der Trend zur digitalen Modulation. Als Beispiel zeigt sich hier die Verwendung von Digital Video Broadcasting (DVB) das derzeit in verschiedenen Medien die rein analoge Modulation von Signalen verdrängt. Die steigende Verwendung von digitalen Modulationsverfahren schließt analoge Modulationsverfahren nicht aus. Im Gegenteil, betrachtet man verschiedene Formen der gängigen digitalen Modulationen genauer, basiert jedes digitale Modulationsverfahren einem analogen.

Folgende Abbildung erklärt die Anwendung von Amplitudenmodulation (AM), Frequenzmodulation (FM) und Phasenmodulation (PM):

$$s_T(t) = \overset{AM}{\hat{s}_T} \sin\left(\overset{FM}{\hat{\omega}_T} t + \overset{PM}{\hat{\varphi}_T}\right)$$

Amplitudenmodulation:  $\hat{s}_T = f(s_M(t))$

Frequenzmodulation:  $\omega_T = f(s_M(t))$

Phasenmodulation:  $\varphi_T = f(s_M(t))$

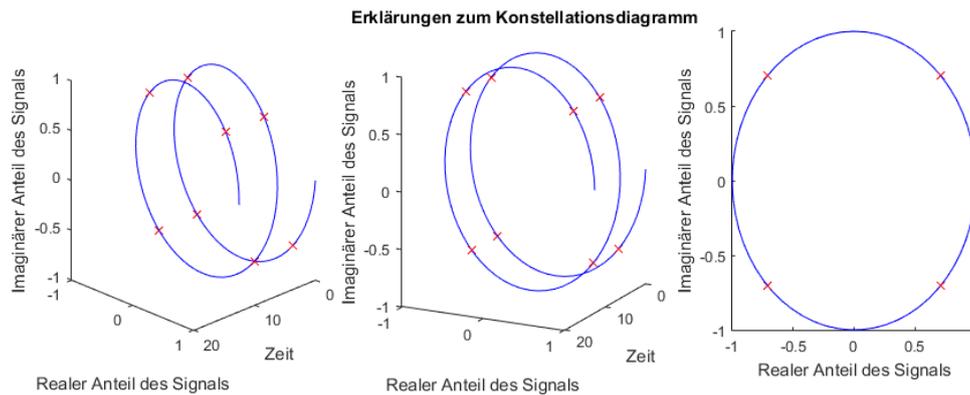
**Abbildung 5: Mathematische Grundlagen analoger Modulationsverfahren**

Die Abbildung 5 ist nachgebildet aus (Böge W. und Pläßmann W, 2007). Formelzeichen mit niedergestelltem ‚T‘ beziehen ihre Eigenschaften von der Trägerfrequenz, ein niedergestelltes ‚M‘ kennzeichnet das zu modulierende Signal.

### 2.4.3. Konstellationsdiagramm

Die Ergebnisse der hier gezeigten digitalen Modulationsverfahren werden in der Regel als Konstellationsdiagramm angezeigt. Hier wird kurz das Zustandekommen des Konstellationsdiagramms erklärt.

Die empfangenen Signale werden durch komplexe Zahlen repräsentiert. Der reale und imaginäre Anteil wird von einer Phasenverschiebung um 90 Grad voneinander getrennt. Um dies zu verdeutlichen wurde unabhängig von jedweder Modulation die folgende Grafik erstellt.



**Abbildung 6: Erklärungen zum Konstellationsdiagramm**

Die Abbildung zeigt dreimal denselben Graphen. Zunächst im dreidimensionalen Raum, dann wird wie Rotation um die y- und z-Achse verringert. Der dritte Graph zeigt den Signalverlauf als Kreis, die x-Achse ist nicht mehr zusehen. In der Praxis werden die beiden Signalanteile mit  $I(t)$  und  $Q(t)$  bezeichnet. Die Codierung des Signals geschieht durch das Verändern der Phase und/oder Veränderungen an der Amplitude. Die roten Kreuze stellen die Abtastpunkte für die Rekonstruktion der Daten dar. Dies sind die Konstellationen, die den notwendigen Abtastpunkten zur Rekonstruktion des digitalen Nutzsignals entsprechen. Eine genauere Beschreibung kann den einzelnen dargestellten Modulationsverfahren entnommen werden.

#### 2.4.4. Binäre Phasenmodulation und Quadraturphasenumtastung

Bei der BPSK werden binäre Werte einzeln übertragen. Hierfür wird die Phasenlage des Signals verwendet, die zwischen den Werten  $45^\circ$  und  $225^\circ$  wechselt, um eine 0 oder 1 zu übertragen. Durch die geringe Variation des Signals kann selbst bei schlechtem Empfang eine Übertragung sichergestellt werden.

Die QPSK moduliert ähnlich der BPSK, mit dem Unterschied, dass die Phasenlage nur um  $90^\circ$  verändert wird. Hierdurch kann der doppelte Datendurchsatz der BPSK erreicht werden. Pro Trägerfrequenz können somit 2 Bits übertragen werden.

Die Abbildung 7 (IEEE 802.11, 2012) zeigt die beiden Modulationsverfahren, die Pfeile repräsentieren die möglichen Konstellationen.

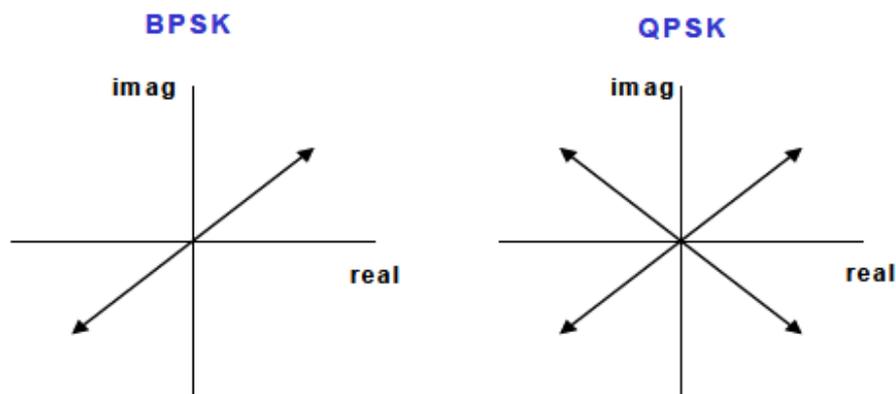


Figure 19-21—BPSK and QPSK signaling with the I/Q channels maximally energized

Abbildung 7: Exemplarische Konstellationsdiagramme der BPSK und QPSK

Die beiden Modulationsverfahren BPSK und QPSK werden im weiteren Verlauf nicht auf die Wirkung des Doppler-Effekts untersucht. Sie sind jedoch hier beschrieben, um das Verständnis für die Modulationen zu komplettieren.

### 2.4.5. Quadraturamplitudenmodulation

Die QAM wird bei ausreichender Übertragungsqualität angewandt. Sie nutzt sowohl die Amplitudenmodulation als auch die Phasenmodulation. Das zu übertragende Signal wird in zwei Komponenten,  $I(t)$  und  $Q(t)$  aufgeteilt, die um  $90^\circ$  verschobene Phasen aufweisen. Die Amplituden der Komponenten werden anschließend mit den Nutzdaten multipliziert. Entsprechend der Übertragungsqualität wird nochmals zwischen der 16-QAM und 64-QAM unterschieden. Der Unterschied zwischen den beiden Verfahren ist, dass bei einer 16-QAM die Amplituden vier verschiedene Werte, bei der 64-QAM 16 verschiedenen Werten annehmen können.

$$A \cos(2\pi f_c t + \varphi) = I \cos(2\pi f_c t) - Q \sin(2\pi f_c t) \quad (2.4.5)$$

Die Formel 2.4.5 (NI-Tutorial-2898-en) zeigt die mathematische Darstellung der QAM.

Im Fall einer 16-QAM wird ein Eingangssignal mit 4 Bit in zwei Signale  $I(t)$  und  $Q(t)$  mit jeweils 2 Bit aufgeteilt. Diese werden mit der Trägerfrequenz multipliziert. Das Empfangskriterium der QAM ist, dass Sender und Empfänger dieselbe Phasenlage besitzen.

Die Ergebnisse der QAM werden durch ein Konstellationsdiagramm angegeben. Die Entstehung eines Konstellationsdiagramms ist in Kapitel 2.4.3 erläutert. Dies zeigt auf der Ordinate den Verlauf der Amplitude von  $I(t)$  und  $Q(t)$  auf der Abszisse.

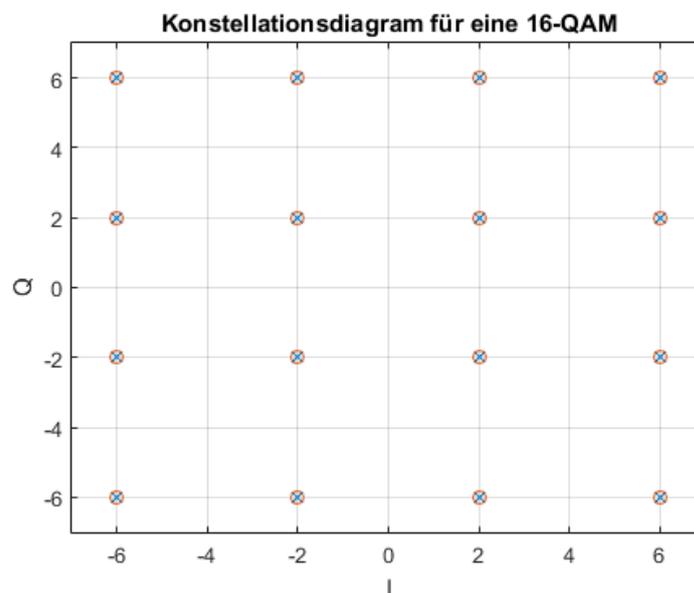


Abbildung 8: Beispielhaftes Konstellationsdiagramm einer 16-QAM

#### 2.4.6. Orthogonales Frequenzmultiplexverfahren

Das orthogonale Frequenzmultiplexverfahren (engl.: Orthogonal Frequency-Division Multiplexing, OFDM) beschreibt die Verwendung mehrerer im Frequenzbereich zueinander orthogonal stehender Trägerfrequenzen. Das OFDM hat das Ziel, Störungen die durch benachbarte Träger innerhalb eines Kanals entstehen, zu vermeiden. Eine Umsetzung der OFDM geschieht im Normalfall durch die Anwendung der inversen FFT auf die Ergebnisse vorangehender Modulationen.

Spezifischer wird die Umsetzung in der IEEE 802.11 durch folgende Formel beschrieben:

$$r_{SUB}(t) = w_{TSUB}(t) \sum_{k=-N_{ST}/2}^{N_{ST}} C_k \exp(j2\pi\Delta_f)(t - T_{GUARD}) \quad (2.4.6.1)$$

Hieraus ergibt sich nach Anwendung der Eulerschen Formel:

$$\exp(j\varphi) = \cos(\varphi) + j \sin(\varphi) \quad (2.4.6.2)$$

$$r_{SUB}(t) = w_{TSUB}(t) \sum_{k=-N_{ST}/2}^{N_{ST}} C_k [\cos(2\pi\Delta_f) + j \sin(2\pi\Delta_f)](t - T_{GUARD}) \quad (2.4.6.3)$$

Die Gleichung (2.4.6.1) zeigt, dass zur Berechnung der OFDM die inverse FFT verwendet werden kann. Die Gleichung (2.4.6.3) zeigt, dass die Umsetzung ebenso durch Addition der beiden um 90 Grad verschobenen Phasen möglich ist. Dies ermöglicht eine Simulation des Signals im hochfrequenten Bereich unter Einbezug des Doppler Effekts.

In Kapitel 2.1 wird bereits erwähnt, dass das OFDM die für die Erkennung von Frequenzverschiebungen in der drahtlosen Datenübertragung geeignet ist. Das Verfahren teilt den Nutzdatenstrom auf verschiedene konstante Trägerfrequenzen auf. Die Frequenzen sind über die gesamte Übertragung hinweg konstant und variieren nur in ihrer Amplitude. Dies gleicht dem Prinzip des Dauerstrichradars.

### 2.4.7. Modulation und Demodulation am Beispiel des IEEE 802.11

In diesem Kapitel wird die Funktionsweise von Teilen des IEEE Standards zur Datenübertragung erläutert. Hierzu werden Kenndaten der Übertragung sowie die Anwendung der Modulationsverfahren dargestellt. Spezifischer wird hier der IEEE 802.11g Standard untersucht. Dieser verwendet die bereits erklärten Modulationsverfahren.

Ein Rahmen (engl. Frame) wird in der Übertragungstechnik auch als Paket bezeichnet. Grundsätzlich erfolgt hierbei eine Unterscheidung zwischen:

- Control Frames
- Data Frames
- Management Frames

Die verschiedenen Paketformen dienen der Kommunikation zwischen den Endgeräten und dem Zugangspunkt. Durch die Control- und Management-Pakete wird beispielsweise die verwendete Bandbreite, Modulationsart etc. festgelegt. Im Folgenden wird die Übertragung eines Datenpakets genauer betrachtet.

#### **Allgemeine Bedingungen**

Der Standard legt zunächst das 2,4 GHz ISM Band zur Datenübertragung fest. Dieses wird regional bedingt in unterschiedliche Kanäle aufgeteilt. Für die Region Deutschland, ist das Band in 13 Kanäle unterteilt. Diese haben einen Abstand von 5MHz zueinander.

Zur Übertragung stehen drei verschiedene Bandbreiten zur Verfügung 5 MHz, 10 MHz und 20 MHz. Zwischen Ihnen wird unterschieden, wenn aufgrund vieler Drahtlosnetzwerke wenig Bandbreite verfügbar ist.

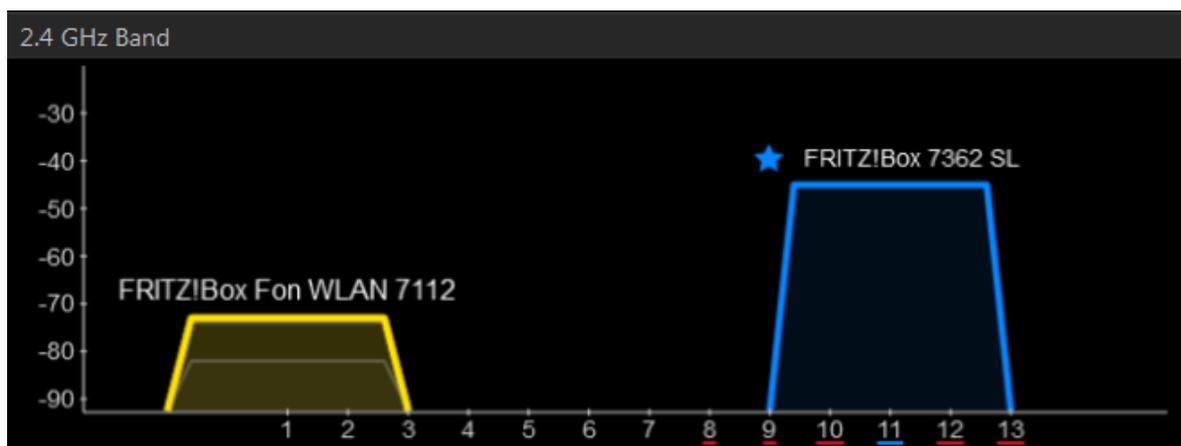


Abbildung 9: Darstellung der verwendeten Bandbreite

Die Abbildung zeigt die Spektren zweier Zugangspunkte mit je 20 MHz Bandbreite. Der blau eingezeichnete Sender sendet auf Kanal 11 mit einer Bandbreite von 20 MHz. Somit sind die Kanäle 9 bis 13 ebenfalls belegt. Die Abbildung ist ein Auszug aus inSSIDer, einem Programm zur Analyse von Drahtlosen Netzwerken.

Im Beispiel einer Bandbreite von 20 MHz ergibt sich ein Abstand von 312,5 MHz zwischen den einzelnen Sub-Trägern. In folgender Abbildung 10 (NI-Tutorial-3740-en, 2014) wird das Prinzip verdeutlicht.

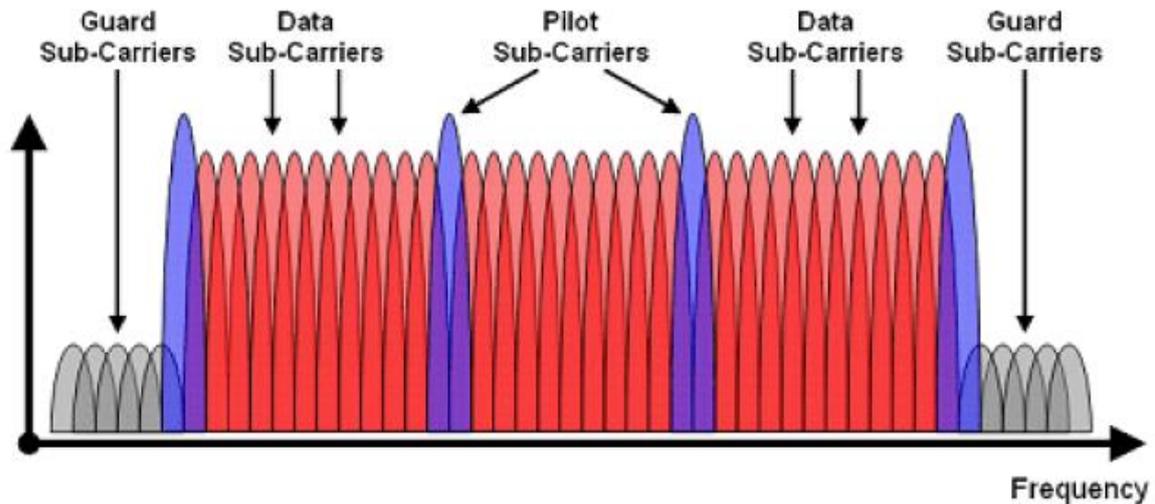


Abbildung 10: schematische Aufstellung der einzelnen Trägerfrequenzen

Ein einzelner Rahmen der Übertragung besteht aus 64 Subträgern und der eigentlichen Trägerfrequenz, die selbst keine Nutzdaten überträgt. Um Interferenzen zwischen den einzelnen Kanälen zu vermeiden werden die ersten 5 und die letzten 6 Subträger ausgelassen. Diese werden als Guards bezeichnet. Weitere vier Subträger werden als sogenannte Piloten verwendet. Diese dienen der Synchronisation der Übertragung fest. Die Nutzdaten werden von 46 Subträgern übertragen.

## Modulation

Die einzelnen Daten Sub-Träger werden mit den bereits genannten Modulationsverfahren BPSK, QPSK oder QAM Moduliert.

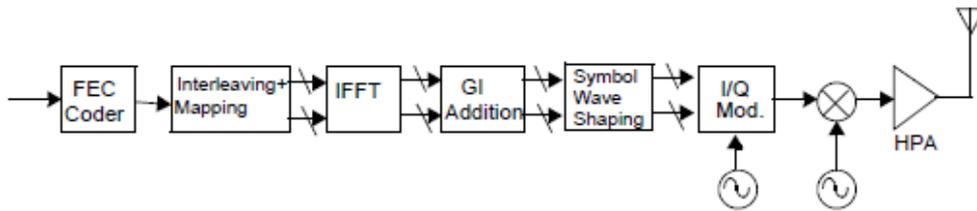


Abbildung 11: Blockdiagramm zum Senden eines Datenstroms

Das Blockdiagramm (IEEE 802.11, 2012) zeigt den Weg der Signalgenerierung.

- Vorwärtsfehlerkorrektur (FEC):  
Die Daten werden durch einen Faltungscodiercode ersetzt, der die Daten redundant beinhaltet, um bei der Decodierung mögliche Fehler zu korrigieren.
- Verschachtelte Zuordnung (IM):  
Die Daten werden mittels BPSK, QPSK oder QAM moduliert.
- Inverse Fouriertransformation:  
Anwendung der OFDM (Siehe Kapitel 2.4.6)
- GI Ergänzung:  
Die Guards werden hinzugefügt
- Symbol Wellen Modellierung:  
Anwendung von Fensterfunktionen um die Spektrale Leistungsdichte zu regulieren
- I/Q Modulation  
Die zugeordneten Nutzdaten werden auf die zwei um  $90^\circ$  verschobenen Phasen moduliert.
- Mischen der Zwischenfrequenz auf die Übertragungsfrequenz.
- Verstärkung mit einem High Power Amplifier (HPA):  
Das Signal wird zur Übertragung verstärkt.

Der Standard gibt weiterhin ein Beispiel für den Rahmen einer Übertragung mit 20MHz Bandbreite an. Zur Messung eines solchen Rahmens wird eine Bandbreitenauflösung von 100kHz, sowie eine dargestellte Bandbreite von 30MHz zur Visualisierung eines Rahmens empfohlen.

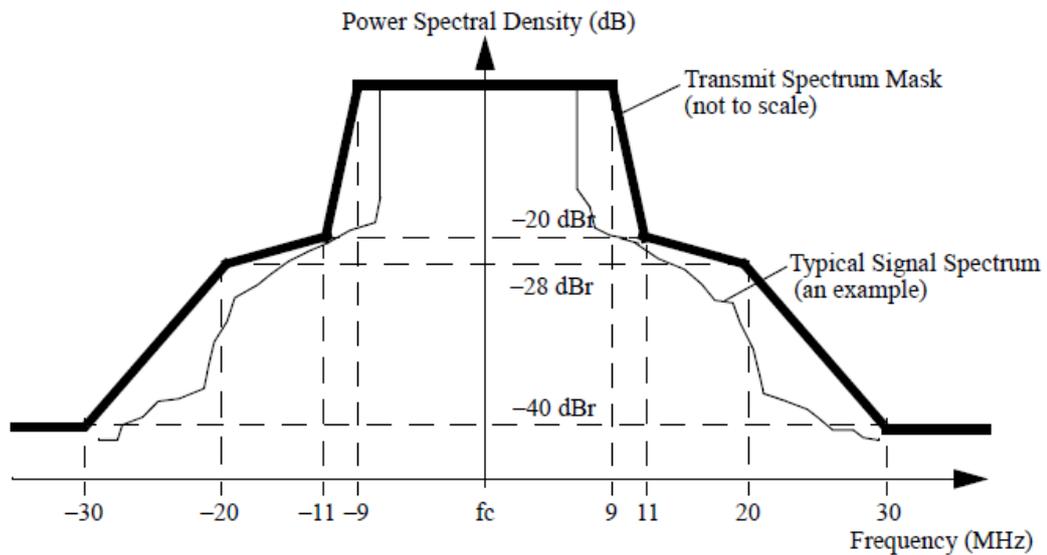


Abbildung 12: Beispiel eines Signals im Spektrum

Die Abbildung 12 (IEEE 802.11, 2012) zeigt die Maske und ein Beispiel eines Signals das zum Versenden bereitgestellt wird.  $f_c$  bezeichnet die Trägerfrequenz. Die spektrale Leistungsdichte auf der Ordinate zeigt, dass rund um das eigentliche Signal von  $\pm 9$  MHz abzüglich der freigelassenen Guards, ein Rauschen entsteht. Dies ist von der Fensterfunktion, die bei der Symbol-Wellen-Modellierung angewandt wird abhängig. Die IEEE gibt keine feste Fensterfunktion vor. Das tatsächlich benötigte Fenster ist abhängig von der verwendeten Hardware.

Neben den Empfehlungen der IEEE im 802.11 Standard gibt das Werk, „Waveshaping of Multicarrier Signal for Data Transmission Over Wireless Channels“ von H. Nikookar Aufschluss über die Wirkung verschiedener Fenster.

## Demodulation

Die Demodulation ist die Rückgewinnung der Nutzdaten im Empfänger bzw. Sender. Im Folgenden wird die Demodulation des Signals erklärt.

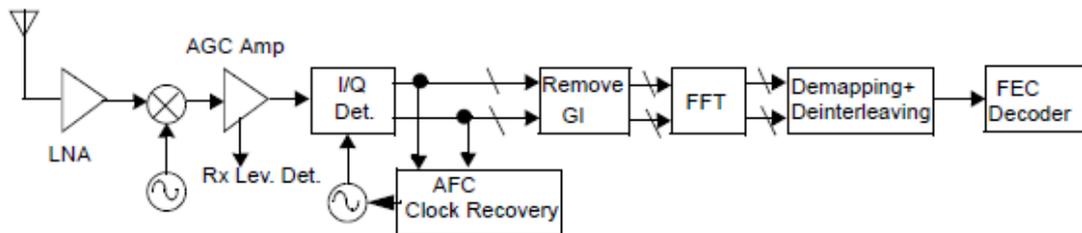


Abbildung 13: Blockdiagramm zum Empfang eines Datenstroms.

Zur Demodulation zeigt Standard ebenso ein exemplarisches Blockschaltbild (IEEE 802.11, 2012). Dies wird in folgenden Schritten erörtert.

- Verstärkung des Signals mit einem Low Noise Amplifier (LNA):  
Ein Verstärker mit möglichst geringem Rauschen verstärkt das empfangene Signal.
- Mischen der Übertragungsfrequenz auf die Zwischenfrequenz mit anschließender automatischer Verstärkung.
- I/Q Demodulation:  
Das Signal wird wieder in zwei Phasen aufgeteilt, wobei die Frequenz des Signals durch eine automatische Regelung rekonstruiert wird
- GI Löschung:  
Die Guards werden abgeschnitten
- Fourier Transformation :  
Auflösen der OFDM (Siehe Kapitel 2.4.4).
- Rekonstruktion der verschachtelten Zuordnung:  
Demodulation der BPSK, QPSK oder QAM.
- Durchführung der Vorwärtsfehlerkorrektur:  
Auflösen der redundanten Daten und Wiederherstellung des eigentlichen Datenstroms.

Die tatsächlichen Parameter der Modulation und Demodulation sowie der Signalübertragung, sind abhängig von den verwendeten Chips, Schaltkreisen und Verstärkern. Die IEEE 802.11 Spezifiziert die Standards und Randbedingung zur Übertragung.

## 2.5. Bewegungsdetektion in hochfrequenten Signalen

Zunächst werden hier bereits realisierte Anwendungen des Doppler-Effekts aufgezeigt und die verschiedenen Funktionsweisen kurz erläutert. Ebenso wird eine Anwendung der Bewegungsdetektion diskutiert, die ohne die Auswertung des Doppler-Effekts realisiert ist.

### 2.5.1. Vorteile analoger Schaltkreise zur Detektion

Der Doppler Effekt findet bereits in vielen Bereichen der Technik Anwendung. Bezogen auf die Bewegungen des Menschen, werden Doppler Sensoren verwendet um durch Bewegungen beispielsweise automatische Türöffner oder Lichtschalter zu steuern. Hierfür gibt es rein analoge Schaltkreise, die die Detektion des Doppler-Effekts ermöglichen. Im Folgenden wird die Funktionsweise des InnoSent IPS-265 betrachtet. Das Modul wird in Deutschland ebenso unter der Bezeichnung RSM2650 von B+B Sensors geführt und ist nicht größer als ein zwei Euro Stück.

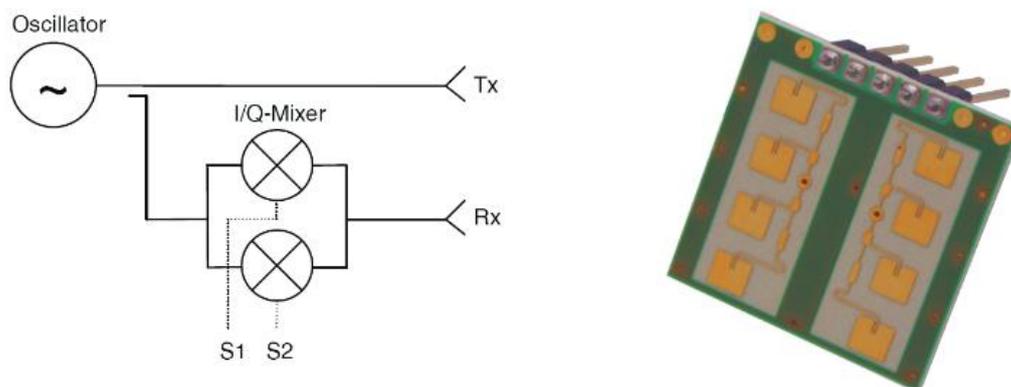


Abbildung 14: Blockdiagramm und Antennen Array des IPS-265

Der Sensor (B+B Sensors) sendet und empfängt Frequenzen im 24GHz ISM-Band mit je vier Antennen. Durch die hohe Frequenz wird nach den Folgerungen aus Kapitel 2.2 eine bessere Detektion der Bewegung erreicht. Der Empfang läuft, wie auch die Auswertung der genannten Modulationen, über einen I/Q Mixer, der in zwei um 90 Grad verschobenen Phasen, das Ausgangssignal bereitstellt. Laut Datenblatt muss das Ausgangssignal mit einer Bandbreite zwischen 20 bis 900 Hz zum Weiterverarbeiten aufbereitet werden. Durch den Vergleich der Signale in analogen Schaltkreisen, reicht eine geringe Bandbreite aus, um Bewegungen in den Signalwegen zu erkennen.

Die Kenndaten des Moduls zeigen, dass durch rein analoge Technik der Doppler-Effekt ohne großen Berechnungsaufwand detektiert werden kann. Dies wird realisiert indem die empfangenen Signale mit der ursprünglichen Frequenz verglichen werden. Die notwendige Bandbreite der Signalauswertung ist

wesentlich geringer als bei einer Umsetzung mit SDRs. Dies ist ein entscheidender Vorteil bei der Detektion des Doppler-Effekts in einer analogen Schaltung.

### 2.5.2. Bestehende wissenschaftliche Projekte

Folgend werden Anwendungen, die auf dem Doppler-Effekt in digitalen Modulationen beruhen, diskutiert. Diese sind WiSee (Quifan et al, 2012) und Wi-Vi (Adib F. and Katabi D., 2013).

#### WiSee

WiSee beschäftigt sich mit der Möglichkeit einer Gestensteuerung die darauf basiert, dass eine drahtlose Datenübertragung, nach dem IEEE 802.11 Standard, mitgeschnitten wird. Wenn die Hochfrequenten Signale im Raum auf eine Bewegung treffen, wird in den Signalen der Doppler-Effekt induziert. Die einzelnen Signale werden ausgewertet und mit definierten Signalen verglichen, die einer Geste entsprechen. Durch dieses Verfahren wird eine Gestensteuerung realisiert die ohne Sensoren am menschlichen Körper auskommt.

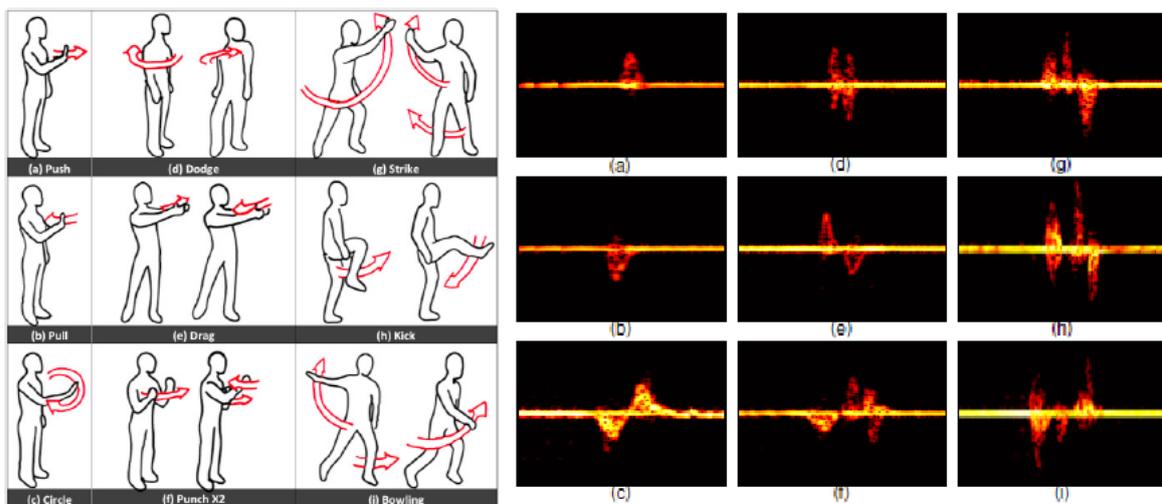


Abbildung 15: Mögliche Gesten bei WiSee

Die Abbildung 15 (Quifan P. et al., 2013a) zeigt links die ausführbaren Gesten, die WiSee detektieren kann. Rechts sind die dazugehörigen Signalverläufe dargestellt. Die Anwendung läuft im Hintergrund und detektiert alle Bewegungen in der Reichweite der Datenübertragung. Hierdurch entsteht das Problem, dass ungewollt Bewegungen als Gesten erkannt werden. Dies wird verhindert indem vor der Eingabe einer Geste zunächst eine Start-Geste ausgeführt werden, um das System zu aktivieren. Danach können die Gesten einzeln ausgeführt oder kombiniert werden um verschiedene Interaktionen durchzuführen.

Das Prinzip von WiSee basiert darauf, die empfangenen Rahmen der Übertragung zunächst so zu modifizieren, dass die Nutzdaten aller Rahmen angeglichen werden. Danach werden die Rahmen aneinandergehängt um damit eine Spektralanalyse durchzuführen, die eine Bandbreitenauflösung von einem Herz erreicht. Auf diese Weise können die durch den Doppler-Effekt induzierten Frequenzänderungen detektiert werden.

Zu achten ist bei der Umsetzung auf die verwendete Hardware, die es ermöglicht die FFT mit einer hohen Anzahl an Abtastpunkten durchzuführen. Näheres hierzu folgt in den nächsten Kapiteln.

### **Wi-Vi**

Eine weitere Anwendung, die sich mit der Detektion von Bewegungen in den Datensätzen drahtloser Kommunikation beschäftigt ist Wi-Vi (Adib F. and Katabi D., 2013). Der Ansatz von Wi-Vi ist unabhängig von der Detektion des Doppler-Effekts. Trotzdem soll hier die Funktionsweise erläutert werden, da die Anwendung das Ziel verfolgt, Bewegungen in drahtloser Kommunikation zu detektieren.

Die Detektion der Bewegung wird realisiert, indem zunächst ein Signal versendet und wieder empfangen wird. Dabei werden die Reflektionen der Umgebung aufgezeichnet, dies ist vergleichbar mit einem Radarquerschnitt. Im zweiten Schritt wird dasselbe Signal nochmals ausgesendet und die beiden Ergebnisse voneinander subtrahiert. Ein Objekt, das sich in dieser Zeit ausreichend bewegt hat, reflektiert das Signal zu einem anderen Zeitpunkt wieder. Hierdurch kommt es dazu, dass das errechnete Signal nur die Reflektionen des sich bewegenden Objekts beinhaltet.

Die Bewegungserkennung erfolgt also nur durch den Vergleich entstehender Reflexionen.

Aus der Vorgehensweise von Wi-Vi ergeben sich, gegenüber der Detektion von Bewegung anhand des Doppler-Effekts, sowohl Vor- als auch Nachteile. Da die Subtraktion der Signale im Zeitbereich erfolgt, kann auf eine Spektralanalyse verzichtet werden. Genauer gesagt werden die Signale bereits in analogen Schaltkreisen miteinander verglichen, bevor diese in digitale Signale zur Auswertung umgesetzt werden. Es ist also keine Durchführung einer FFT erforderlich, die das hochfrequente Signal mit einer hohen Bandbreitenauflösung transformiert. Die Problematik rund um die Bandbreite der Auswertung entfällt.

Für Wi-Vi ergeben sich folgende Nachteile:

- Die Signale werden von Wi-Vi zwar ähnlich den Verfahren einer Datenübertragung moduliert, jedoch sind sie zur eigentlichen Datenübertragung nicht brauchbar. Es wird also kein Signal mitgeschnitten sondern ausschließlich zur Detektion von Bewegungen versendet.
- Die Bewegungen müssen ausreichend große Distanz aufweisen um erkannt werden zu können. In der Wi-Vi Publikation wird eine minimale Bewegung von 50cm vorausgesetzt. Die minimale Größe der Bewegung ergibt sich aus dem vierfachen der Wellenlänge der Sendefrequenz.

### ***Verwendete Hardware***

Beide Anwendungen verwenden einen Ettus USRP N210 als SDR. Dieser gilt als hochauflösender Software-Radio mit integriertem, programmierbarem Logikbaustein zur Weiterverarbeitung der Signale. Hierbei handelt es sich um einen Spartan 3A-DSP 3400 FPGA, der die Echtzeit-Verarbeitung der Signale ermöglicht. Die maximale Datenmenge des verbauten Analog-Digital-Konverters ist mit 100 MS/s und einer Genauigkeit von 14-Bit angegeben.

### ***Weiterführende Möglichkeiten der Bewegungsdetektion***

Das Werk „Millimeter-Wave Doppler Spectrum and Polarimetric Response of Walking Bodies“ (Vahidpour M. and Sarabandi K.) zeigt wie aufschlussreich das Doppler-Spektrum zusammen mit einem Radarquerschnitt ist. In der Arbeit werden mittels der beiden Verfahren Rückschlüsse auf das sich bewegende Objekt oder die sich bewegende Person gezogen. Die Arbeit deutet darauf hin, dass es möglich ist, durch die Kombination der Anwendungen zwischen unterschiedlichen Personen, Geschlechtern, Größe und Gewicht zu unterscheiden.

### ***Schlussfolgerungen***

Zusammenfassend kann gesagt werden, dass

- durch Messung des Doppler-Effekts eine präzisere Beschreibung von Bewegungen möglich ist als durch den Vergleich von Reflexionen.
- die Auswertung des Doppler-Effekts in digitalen Umsetzungen, durch die benötigte Bandbreitenauflösung, eine hohe Rechenleistung erforderlich macht.
- in den bisher beschriebenen Situationen mit hochauflösenden SDRs gearbeitet wird, die ebenso die Fähigkeit besitzen die Signale durch dedizierte, programmierbare Bausteine zu verarbeiten.

### 2.5.3. Problematik der Erkennung kleiner Frequenzänderungen in breitbandigen Signalen

Ein nennenswertes Problem bei der Auswertung des Doppler-Effekts in digital modulierten Signalen liegt bei der Umsetzung von analogen Signalen in digitale Signale. Zur Rekonstruktion der Nutzdaten, also zur Demodulation, reicht eine geringe Bandbreitenauflösung aus. Diese ergibt sich aus dem Abstand der einzelnen Trägerfrequenzen. SDRs sind deshalb dafür konzipiert die Signale mit einer ähnlichen Bandbreitenauflösung aufzuzeichnen. Die notwendige Bandbreitenauflösung von 1 Hz muss deshalb auf anderen Wegen erreicht werden.

In der Umsetzung von WiSee wird hierfür ein Lösungsansatz genannt. Dieser besteht aus dem Zusammensetzen empfangener Signale. Hierzu betrachtet WiSee die einzelnen Rahmen der Datenübertragung. Zunächst werden die Nutzdaten aus den empfangenen Datensätzen angeglichen, dann werden die Rahmen aneinander gehängt, um mittels FFT eine größere Bandbreitenauflösung zu erhalten.

$$X_n = \sum_{k=1}^N x_k e^{-\frac{i2\pi kn}{2N}} + \sum_{k=N+1}^{2N} x_k e^{-\frac{i2\pi kn}{2N}} \quad (2.5.3)$$

Die Formel (2.5.3) zeigt den theoretischen Ansatz der WiSee Implementierung zur Detektion des Doppler Effekts. Die empfangene Bandbreite wird entsprechend oft hintereinander gesetzt, um eine höhere Auflösung im Frequenzbereich zu erhalten. Spezifischer wird beschrieben, dass das Signal mit dem demonstrierten Verfahren auf eine Länge von einer Sekunde zusammengesetzt wird und somit eine Bandbreitenauflösung von 1Hz erreicht wird.

### 2.5.4. Der Einsatz von Grafikkarten in SDR Anwendungen

Die Idee die aufgenommenen Daten eines SDR mit einer Grafikkarten (GPU) weiterzuverarbeiten besteht bereits. GPUs bieten die Möglichkeit Auswertungsalgorithmen zu parallelisieren. Diese werden dadurch schneller ausgeführt. Die Publikation von (Nyländen T. et al., 2010) untersucht die Verwendung einer GPU für eine konventionelle Long Term Evolution (LTE) Implementierung. Die Ergebnisse der Arbeit zeigen, dass die Rechenleistung der GPU ausreichend ist, während das eigentliche Problem darin liegt, die per SDR aufgenommenen Daten in den Speicher der Grafikkarte zu übertragen. Weiterhin stellt das verzweigte parallele Rechnen eine Herausforderung für die angewandten Algorithmen dar.

Die Rechenarbeit, die durch GPUs in SDR Anwendungen übernommen werden kann, wird in den bisher beschriebenen Projekten durch einen FPGA ausgeführt. Der Vergleich der Rechenleistung zwischen GPU und FPGA ist in verschiedenen

Bereichen ein wiederkehrendes und bereits mehrfach diskutiertes Thema. Eine Recherche zeigt auf, dass die Erfolgsfaktoren abhängig von den umzusetzenden, meist mathematischen, Problemen sowie von den darauf basierenden Algorithmen sind.

Die beiden Hardwarekomponenten zeichnen sich durch eine hohe Rechenleistung aus. Dennoch liegen ihnen zwei unterschiedliche Architekturen zu Grunde. Diese werden im Folgenden beschrieben.

### **FPGAs**

Field Programmable Gate Arrays sind Chips der digitalen Datenverarbeitung, die in komplizierten Hardwarebeschreibungssprachen wie VHDL (Very High Speed Hardware Description Language) und Verilog programmiert werden. Ein FPGA ist ein einzelner Chip, der ohne weitere Peripherie nicht auskommt. Deshalb werden diese in der Regel als fertige Entwickler-Boards vertrieben.



**Abbildung 16: Beispiel eines FPGA Entwickler-Boards**

Die Abbildung 16 (Terrasic, 2012) zeigt einen FPGA mit Programmierschnittstelle und diverser Peripherie wie Bewegungssensoren, Analog-Digital-Konverter, etc. Die hohe Rechenleistung wird bei FPGAs erreicht, indem pro Taktzyklus mehrere Operationen ausgeführt werden (NI-Tutorial-6984-de, 2012). Die tatsächliche Leistung ist abhängig von den verfügbaren Gates / Transistoren die in dem FPGA verbaut sind und der Programmierung der Chips, die maßgeblich dafür verantwortlich ist, wie viele der verfügbaren Gates verwendet werden.

Die Peripherie eines FPGA muss durch einzelne Vorgänge gesteuert und ausgelesen werden. Selbst bei fertigen Entwickler-Boards müssen, die einzelnen Hardwarekomponenten studiert werden und in der einer Hardwarebeschreibungssprache angesprochen und gesteuert werden. Implementierungen in Hardwarebeschreibungssprachen benötigen viel Zeit, sowie detaillierte Kenntnisse zu den einzelnen Bausteinen.



**Abbildung 17: Der Ettus USRP N210 Software-Radio**

In den genannten Anwendungen wird ein Ettus USRP N210 Software-Radio (Abbildung 17<sup>2</sup>) verwendet. Dieser besitzt einen integrierten FPGA, der es ermöglicht die Rechenoperationen mit einer großen Anzahl an Samples und somit hohen Bandbreitenauflösung durchzuführen. Dies ist, wie in den bisherigen Kapiteln erörtert, die Voraussetzung zur Detektion des Doppler-Effekts in digitalen Modulationsverfahren.

### **GPUs**

Wie bereits oben erläutert, kann die Rechenleistung eines FPGAs und eines GPUs je nach Anwendungsfall miteinander verglichen werden. Die große Leistungsfähigkeit eines FPGAs entsteht dadurch, dass innerhalb eines einzelnen Taktzyklus mehrere Operationen durchgeführt werden können. Eine GPU hingegen arbeitet wie ein herkömmlicher Prozessor, benötigt also pro Operation mehrerer Taktzyklen. Die große Rechenleistung eines GPUs entsteht durch parallele Rechenvorgänge. Derzeitige High-End GPUs besitzen bis zu 3000 Prozessoren, die Daten parallel verarbeiten können.

Weiterhin ist die Umsetzung von Programmen auf einer GPU nicht von einzelnen Hardwarekomponenten abhängig. Die Programmierung geschieht in der Regel in C/C++ Derivaten und fest definierten Programmierstandards wie CUDA, OpenCL,

---

<sup>2</sup> [www.ettus.com/product/details/UN210-KIT](http://www.ettus.com/product/details/UN210-KIT)

OpenGL u.v.m. Hierdurch entsteht im Vergleich zu Implementierungen auf FPGAs eine deutlich kürzere Entwicklungsdauer.

In der Publikation (Nyländen T. et al.) werden sowohl das verzweigte parallele Rechnen wie auch die langsame Zuweisung und der Transfer von Speicher als limitierende Faktoren für die Implementierung von SDRs auf GPUs angeführt. Die Zuweisung und der Transfer von Daten auf die GPU, kann durch die heutigen Standards Graphics Double Data Rate 5 (GDDR5) und Peripheral Component Interconnect Express 3.0 (PCIe 3.0) neu betrachtet werden. Im Appendix findet sich ein Vergleich, zwischen den beiden Grafikkarten, die in der Publikation und in der hier vorliegenden Arbeit verwendet werden.

Der Anstieg der Rechenleistung derzeitiger System- und Grafikhardware legt einen Versuch der Umsetzung eines SDRs mittels GPU nahe. Da die Taktrate, die Anzahl der Kerne und die Speicheranbindung aktueller GPUs ist gestiegen ist.

Unter Verwendung aktueller GPUs gibt es bereits eine Implementierung zur Visualisierung der Signale. ‚gr-fosphor‘ der Open Source Mobile Communications (OSMOCOM) ist vorgesehen um in GNU-Radio große Bandbreiten zu visualisieren. Laut der angegebenen Benchmarks können Datendurchsätze von bis zu 220 MS/s erreicht werden.

## **2.6. Mögliche Vorgehensweisen**

In der bisherigen Ausarbeitung werden die Grundlagen der Doppler-Detektion in Verfahren der drahtlosen Nachrichtentechnik erläutert. Folgend wird gezeigt, dass das weitere Vorgehen der Arbeit in zwei Szenarios unterteilt werden kann:

- Es wird untersucht, inwiefern sich die Empfangstechnik verschiedener SDRs auf die Auswertung des Doppler-Effekts auswirkt.
- Es wird untersucht, inwieweit die Auswertung des Doppler-Effekts in digitalen Verfahren durch den Einsatz von GPUs verbessert werden kann.

Die beiden Vorgehensweisen unterscheiden sich in den zu verwendenden Materialien.

## **Weiterführende Untersuchungen zur Empfangstechnik**

Aus den vorangegangenen Beschreibungen geht hervor, dass die Wirkung des Doppler-Effekts in digitalen Modulationen auf einer hohen Übertragungsfrequenz stattfindet. Um diese zu empfangen und zu digitalisieren wird ein Überlagerungsempfänger benötigt. Dieser mischt die hohe Übertragungsfrequenz auf eine bauteilspezifische Mittenfrequenz herunter. Dem Wissensstand dieser Arbeit zufolge ist unklar, ob hierbei die kleinen Frequenzänderungen - die durch Bewegungen induziert werden – durch das Mischen auf eine niedrigere Frequenz verloren gehen.

Um präzisere Aussagen zu diesem Thema machen zu können bedarf es dem Untersuchen der Übertragungsfunktionen, die dem Frequenzmischer zugrunde liegen.

### **Der Einsatz von GPUs in Software-Radios:**

Aus der Erklärung der Bandbreitenauflösung geht hervor, dass zum Erreichen einer möglichst hohen Bandbreitenauflösung eine hohe Rechenleistung erforderlich ist. Durch eine hohe Bandbreitenauflösung können kleinste Veränderungen der Frequenz des Signals im Spektrum detektiert werden. Die Rechenleistung steigt mit zunehmender Anzahl an Abtastpunkten. Am Beispiel von SDRSharp wurde gezeigt, dass von der Gesamtzahl der Abtastpunkte eines Signals häufig nur ein Bruchteil auf Endgeräten verarbeitet wird.

Weiterhin erläutert das Kapitel 2.5.4 die Möglichkeit der Umsetzung eines Software-Radios auf einer GPU. GPUs erlauben die Verarbeitung großer Datenmengen parallel auf mehreren Prozessoren.

Hieraus ergibt sich die Vorgehensweise, zu untersuchen, ob die Leistung aktueller Grafikkarten ausreichend ist, um hochfrequente Signale zu simulieren. Es wird untersucht, ob es mit den Standards GDDR5 und PCIe 3.0 möglich ist, die aufgenommenen Daten mit hinreichender Geschwindigkeit auf die GPU zu laden. Mit den Ergebnissen kann die maximale mögliche Anzahl an Abtastpunkten bei der Durchführung der FFT bestimmt werden.

Die Untersuchung der beiden Vorgehensweisen kann klären, wie präzise zum aktuellen Stand der Technik die Auswertung des Doppler-Effekts durchgeführt werden kann. Um den zeitlichen Rahmen dieser Arbeit einzuhalten, wird nur die Ausarbeitung zum Einsatz von GPUs in SDRs betrachtet.

### 3. Ausgeführte Methoden

Es erfolgt zunächst eine analytische Betrachtung des Doppler-Effekts. Diese wird ausgedehnt auf die entsprechenden Modulationsverfahren.

Weiterhin wird die Umsetzung eines SDR auf einer GPU beschrieben. Dies beinhaltet die

- numerische Betrachtung,
- Validierung der Methoden,
- Diskussion einer Simulation,
- und Erstellung einer GPU SDR Umsetzung.

#### 3.1. Mathematische Betrachtung des Doppler-Effekts

Im Folgenden soll zunächst der Doppler Effekt aufgrund seines mathematischen Ursprungs betrachtet werden. Hierzu wird angenommen, dass im ISM 2,4GHz Band eine Frequenzverschiebung erzeugt wird. Betrachtet wird eine Bewegung, die inmitten des Signalwegs verläuft, sowie eine Querbewegung, die durch den transversalen Doppler-Effekt beschrieben wird.

Vergleicht man zunächst die allgemeinen Formeln mit deren Potenzreihe, fällt auf, dass die Potenzreihen aufgrund des Terms  $\pm\beta^2$  mit  $\beta = v/c$  nicht geeignet für eine digitale Umsetzung sind. Mit  $v = 1 \text{ m/s}$  ergibt sich  $\beta = 3.34 \cdot 10^{-9}$ . Wird dies quadriert, benötigt man mehr als 16 Nachkommastellen. Daher werden die Potenzreihen nicht betrachtet.

Möchten man herausfinden, wie groß der Einfluss des Doppler-Effekts auf die Signale ist, so kann man die Formeln nach der Geschwindigkeit und der Frequenzänderung auflösen.

Demnach gilt:

$$f = f_0 \cdot d \quad (3.1.1)$$

mit  $f = 2,4\text{Ghz}$  ergibt sich:

$$d = \frac{(2,4 \cdot 10^9 + \Delta f)}{2,4 \cdot 10^9}. \quad (3.1.2)$$

Eingesetzt in Formel 2.2.1 und aufgelöst nach v ergibt sich:

$$v = \frac{c - c \cdot d^2}{d^2 + 1}. \quad (3.1.3)$$

Sowie für den transversalen Doppler-Effekt aus Formel 2.2.3:

$$v = -c \cdot d^2 + c. \tag{3.1.4}$$

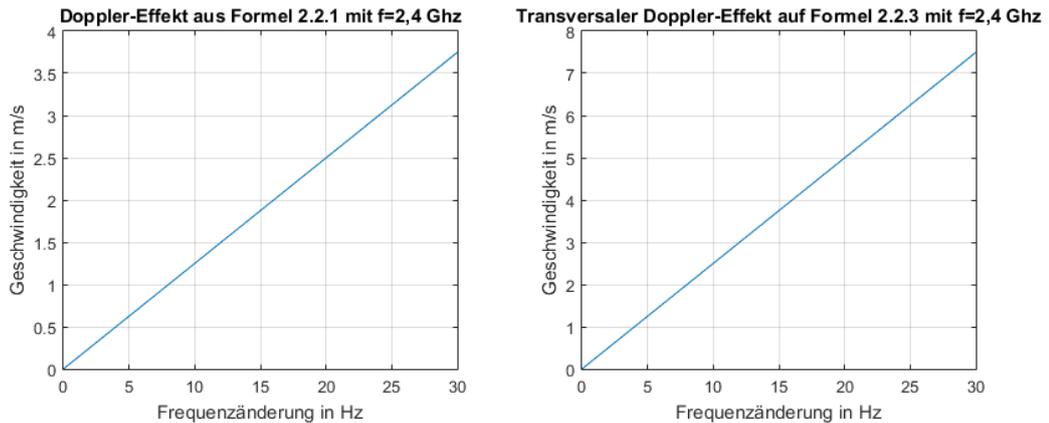


Abbildung 18: Frequenzverschiebung des Doppler-Effekts

In der IEEE 802.11 wird das 20MHz Frequenzband in 64 Subträger aufgeteilt. Die ersten fünf und die letzten sechs Subträger sind die Guards und bleiben ohne Daten, um Störeinflüsse zwischen benachbarten Kanälen zu vermeiden. Die Piloten werden berücksichtigt, da sie wie die Träger zur Nutzdatenübertragung der Wirkung des Doppler-Effekts unterliegen. Die einzelnen Träger haben einen Abstand von 312.5kHz. Berücksichtigt man dies bei der Analyse, kann man davon ausgehen, dass die Frequenzverschiebung in den Subträgern unterschiedlich groß ausfällt.

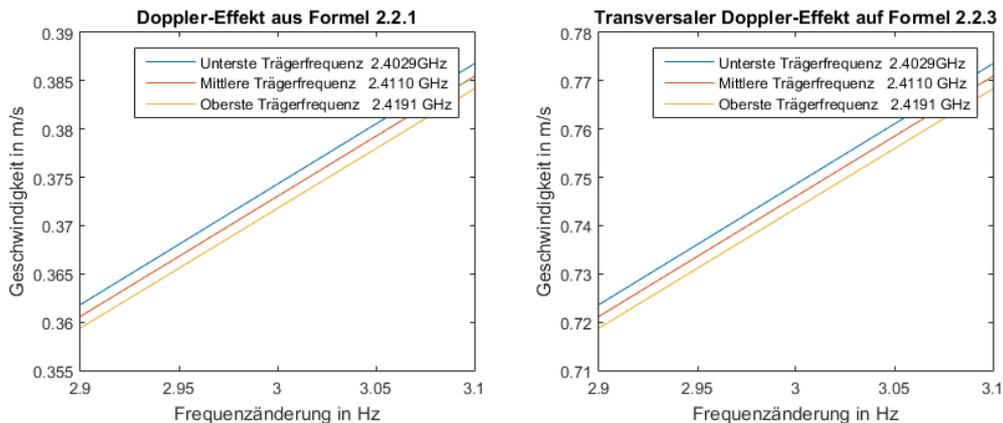


Abbildung 19: Frequenzverschiebung bei unterschiedlichen Trägerfrequenzen

Die Abbildung zeigt, dass die Annahme korrekt, jedoch der Unterschied sehr gering ist. Mit einer normalen Bandbreitenauflösung von 1 Hz ist die Änderung nicht relevant. Dennoch ist die Steigung der drei Geraden nicht gleich. Abbildung

21 zeigt den geringen Unterschied zwischen Geschwindigkeit und resultierender Frequenzverschiebung äußeren Trägerfrequenzen.

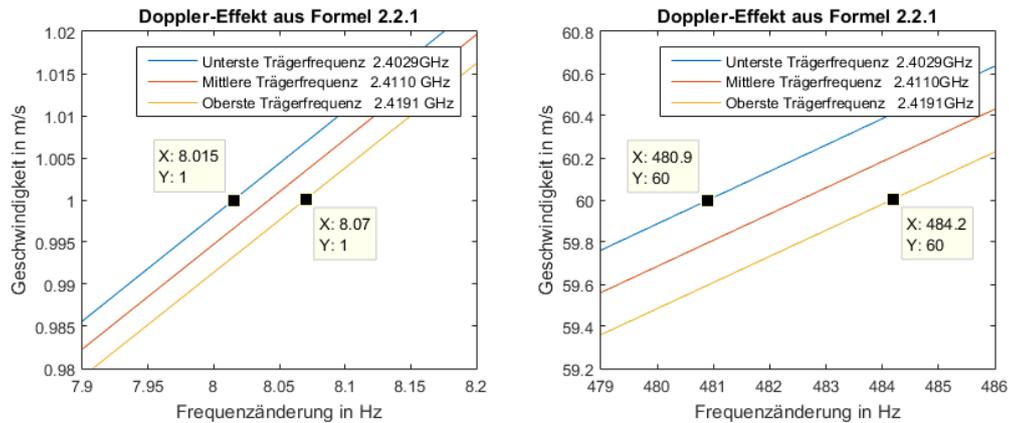


Abbildung 20: Verdeutlicht wird die unterschiedliche Steigung der Geraden

In Simulationen zu OFDM Modulationen wird der Doppler-Effekt häufig als einfache Verschiebung im Frequenzbereich realisiert. Aus den hier gewonnenen Erkenntnissen kann geschlossen werden, dass dies nicht den physikalischen Tatsachen entspricht. Während bei kleinen Bewegungen der induzierte Doppler-Effekt auf den verschiedenen Trägern nur einen geringen Unterschied von 0.055 Hz aufweist, ist bei einer Bewegungsgeschwindigkeit von 60 m/s bereits ein Unterschied von 3.3 Hz vorhanden. Eine Differenz von 1 Hz ist im errechneten Beispiel bereits bei einer Geschwindigkeit von rund 20 m/s ersichtlich.

Die zur Auswertung angefertigten Skripte sind im Appendix unter 7.2 zu finden.

### 3.2. Numerische Betrachtung

Dieses Kapitel soll die Frage klären, ob bei der Detektion des Doppler-Effekts die Verwendung von Datentypen, die eine hohe Präzision besitzen notwendig ist. Hierfür müssen die Anwendungsfälle der Simulation und der eigentlichen Auswertung getrennt voneinander betrachtet werden.

Die Verwendung von möglichst präzisen Datentypen bringt eine längere Berechnungsdauer sowie eine höhere Speicherlast mit sich. Hierdurch entstehen Nachteile in Bezug auf die Leistung und den Datendurchsatz einer Anwendung. Da jeder einzelne Datenpunkt im Speicher abgelegt werden muss, entsteht bei doppelter Präzision auch die doppelte Speicherlast. Ein Abtastpunkt vom Typ ‚double‘ benötigt 64 Bits, ein Abtastpunkt vom Typ ‚float‘ 32 Bits. Überträgt man dies auf eine Bandbreite von 20 MHz, die aus 40MS/s errechnet werden soll, so benötigt man im Idealfall - ohne die Berücksichtigung komplexer Zahlen und eventuell redundant allokierten Speichers – eine Speicherlast von 640 MB bei doppelter Präzision. Bei einfacher Präzision wird nur die Hälfte benötigt.

Um mögliche Einbußen bei der Berechnungsdauer auf einer GPU zu ermitteln, wurde in Benchmark programmiert. Dieser beinhaltet:

- Füllen des Speichers mit einer einfachen Sinusfunktion
- Berechnen der FFT

Der Benchmark misst die Dauer, die für 50 Berechnungen gebraucht wird. Die Ergebnisse zeigen, dass die Berechnung mit einfacher Präzision 100ms benötigen. Für die doppelte Präzision dauert die Auswertung 300ms. In diesem einfachen Anwendungsfall brauchen die Berechnungen auf doppelter Präzision das Dreifache an Zeit.

Die Präzision von Software-Radios ergibt sich aus den verwendeten Analog zu Digital Konvertern. Derzeit verfügbare Konverter besitzen eine Genauigkeit von 8-14 Bits. Demnach ist es ausreichend die Verarbeitung für praktische Anwendungen, wie z.B. der Spektralanalyse, mit kleinen Datentypen durchzuführen.

Weiterhin ist zu prüfen, wie sich die unterschiedlichen Datentypen auf die Erstellung/Simulation eines Signals mit minimaler Frequenzänderung auswirken. Hierzu wird in C/C++ eine Tabelle erstellt.

```
Frequency <double>      | Float Sine          | Double Sine
2409.999991961105000    | -0.00002780        | -0.00005050986657
24099.99991961105100   | -0.00027803        | -0.00050509865898
240999.99919611053000  | +0.09084430        | -0.00505096509455
2409999.99196110530000 | +0.39833161        | -0.05048839080493
24099999.91961105200000 | +0.86473411        | -0.48389378591258
240999999.19611052000000 | +0.85285503        | +0.94322136223173
2409999999.96110530000000 | +0.58939207        | -0.24195743179277
```

Abbildung 21: Numerische Betrachtung kleiner Frequenzverschiebungen

Begonnen wird in der abgebildeten Tabelle mit einer Frequenz von 2411 Hz, die mit einer Bewegung von 1 m/s verrechnet wird. In jeder Zeile wird die Frequenz um den Faktor 10 erhöht. Die letzte Zeile zeigt die Werte für eine ursprüngliche Frequenz von 2,411 GHz wie im IEEE 802.11 Standard definiert. Aus Abbildung 12 ist ersichtlich, dass bei einer Geschwindigkeit von 1m/s eine Frequenzänderung von ungefähr 8 Hz im 2,4GHz Band erzeugt wird. Dies bestätigt der letzte Wert der Tabelle.

Die Tabelle zeigt, dass sich die in einfacher Präzision errechneten Werte für hohe Frequenzen sehr stark von denen in doppelter Präzision unterscheiden. Der letzte Frequenz-Wert mit 2,4 GHz gilt auf dem verwendeten System als Grenzwert. Wird versucht die Frequenz mit 24GHz zu erzeugen, treten numerische Fehler auf.

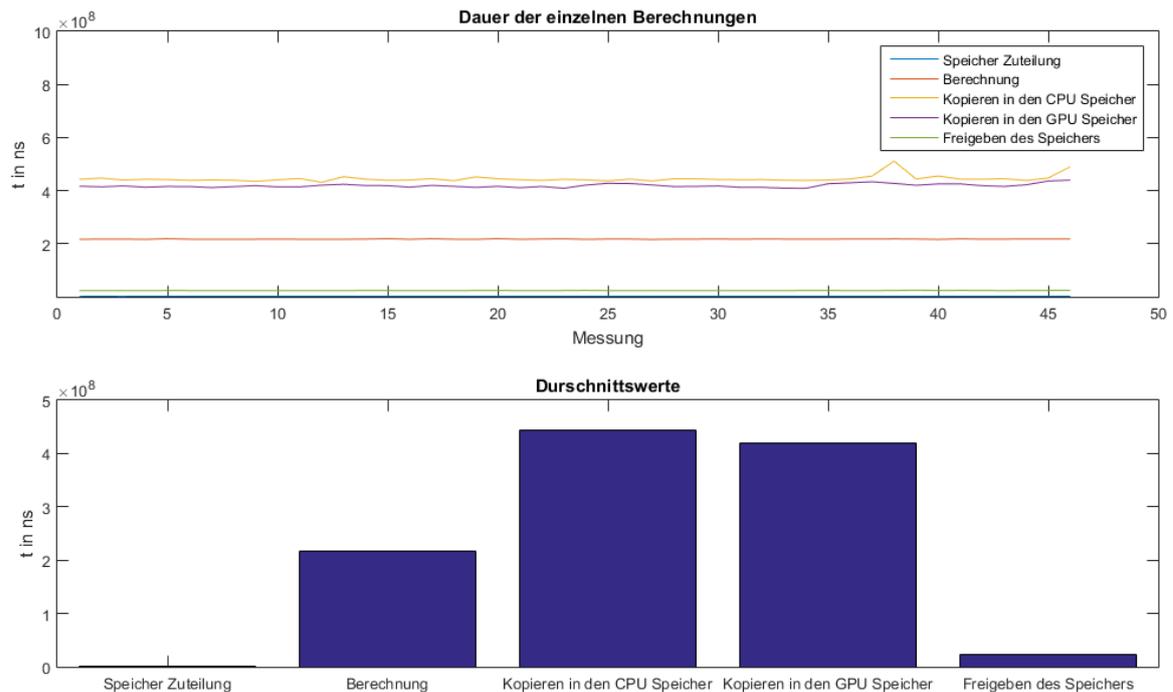
Darüber hinaus wird deutlich, dass jeder Wert nur mit 16 Stellen dargestellt werden kann. Wie in der Tabelle ersichtlich, besitzt die errechnete Frequenz unter Einbezug des Doppler-Effekts bei 2,4 GHz nur noch sieben Nachkommastellen. Um eine möglichst präzise Aussage zu über den Einfluss des Doppler-Effekts in einer Simulation zu gewährleisten, wird in der folgenden Anwendung nur mit einer Frequenz von 2,4 MHz simuliert. Um die Wirkung des Doppler-Effekts trotzdem relativ zum 2,4 GHz Band zu beschreiben, wird diese mit dem Faktor 1000 verrechnet. Mathematisch betrachtet findet eine einfache Kommaverschiebung statt, um den numerischen Fehler der Berechnungen möglichst gering zu halten.

Schlussfolgernd kann gesagt werden, dass zur Erstellung bzw. Simulation eines hochfrequenten Signals die Verwendung von doppelter Präzision notwendig ist. Für die Weiterverarbeitung können anschließend die Nachkommastellen abgeschnitten und es kann mit einfacher Präzision weitergerechnet werden. Auf diese Art und Weise, ist die Ausgabe der Samples einer Simulation - wie in einem SDR - mit einer geringen Präzision realisiert. Trotzdem sind die physikalischen Eigenschaften korrekt simuliert.

### **3.3. Erstellung und Visualisierung von Signalen**

Zur Verarbeitung von simulierten und empfangenen Signalen wird in dieser Arbeit CUDA verwendet. CUDA ist eine Plattform vom Grafikkartenhersteller NVIDIA, die es erlaubt parallele Berechnungen auf GPUs durchzuführen.

In einer frühen Projektphase wurde zum Erstellen der Signale ein Versuch durchgeführt, der Aufschluss über die Möglichkeiten der Signalverarbeitung mit CUDA geben soll. Hierzu wird eine Sinusfunktion mit rund 300 MS auf der GPU erzeugt und die einzelnen Berechnungszeiten gemessen. Um verwertbare Ergebnisse zu bekommen wird der Vorgang 50-mal wiederholt.

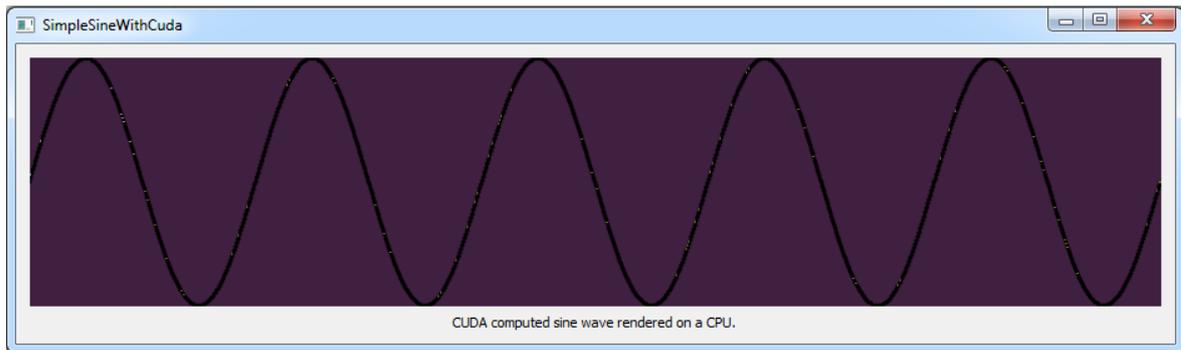


**Abbildung 22: Zeiten der einzelnen Vorgänge**

Der Versuch zeigt, dass mithilfe der GPU die hohe Anzahl an Samples verarbeitet werden kann. Ebenso geht hervor, dass das Kopieren von Daten zwischen den CPU- und GPU-Speichern tatsächlich ein Problem darstellt. Die übertragene Datenmenge liegt jedoch weit über der notwendigen Sampleanzahl. Im Versuch wurden die Daten mit doppelter Präzision also 64 Bits erzeugt. Die übertragene Speicherlast liegt demnach bei ungefähr 2,4 GB. Dies entspricht mehr als dem halben zur Verfügung stehenden Speicher auf der GPU. Die gesamte Dauer eines Durchlaufs liegt bei 1,1 Sekunden.

Um das erstellte Signal weiter zu untersuchen, wurde versucht dieses mit einfachen Mitteln zu darzustellen. Hierfür werden die Daten vom Speicher der GPU in den Speicher der CPU kopiert und anschließend als Punkte gezeichnet. Hierbei entstand ein Flaschenhals in der Implementierung, da die CPU zum Zeichnen der Daten um ein vielfaches länger braucht, als die Generierung auf der Grafikkarte. Ebenso dauert eine Iteration – Beispielsweise zum Finden von Nullstellen im Signal - durch den Datensatz mit der CPU bis zu einer halben Sekunde.

Zunächst wird versucht nicht alle Werte, sondern nur jeden x-ten Wert zu zeichnen. Ein fließender Programmablauf wird hierbei ansatzweise möglich, wenn nur jeder  $10^4$ -te Wert gezeichnet wird.



**Abbildung 23: Darstellung der mit CUDA berechneten Signale.**

Ebenso wird versucht, die Darstellung der Samples durch die externe Bibliothek QCustomPlot zu realisieren. Diese bietet den Vorteil, dass bereits alle Funktionen, beispielsweise zum Zoomen, Auslesen von Datenpunkten etc. implementiert sind. Hierbei ergibt sich jedoch ebenfalls das Problem, dass die Datenmenge zu groß ist und hierdurch der Aufbau der Zeichnung sehr lange dauert.

Um die Probleme des langsamen Zeichnens, des zeitaufwendigen Iterierens durch die Daten auf der CPU und des Verschiebens der Daten zwischen dem GPU und CPU Speicher zu lösen, wird in den folgenden Anwendungen das Zeichnen der Datensätze durch OpenGL realisiert.

Die Open Graphics Library (OpenGL) ist eine Programmierschnittstelle die es in diesem Fall ermöglicht die Grafikkarte ebenso zum Zeichnen zu verwenden. Die Signale können dadurch auf der GPU generiert bzw. ausgewertet und anschließend ebenso durch parallele Rechenoperationen gezeichnet werden. In den weiteren Kapiteln zeigt sich, dass durch die Verwendung einer GPU die Verarbeitung und Visualisierung großer Datenmengen beschleunigt wird.

### 3.4. Verwendete Entwicklungsumgebung und Bibliotheken

Die Umsetzung der Theorie geschieht hauptsächlich in der Programmiersprache C/C++ mit Visual Studio 2013 als Entwicklungsumgebung. Weiterhin wird das Qt Framework (64 Bit Version) mit OpenGL Unterstützung verwendet. Für die Berechnungen auf dem GPU wird CUDA zusammen mit Nvidia Nsight 4.1 eingesetzt. Um die mit CUDA errechneten Datensätze mit OpenGL zu zeichnen, wird der OpenGL Extension Wrangler (GLEW) mit eingebunden.

Liste der verwendeten Bibliotheken:

|                                  | Zweck  | Version | Verwendete Bibliotheken   |
|----------------------------------|--|---------|---|
| <b>Qt Framework</b>              | Steuerung des Programmablaufs                  | 5.3.1   | qtmain.lib<br>Qt5Core.lib<br>Qt5Gui.lib<br>Qt5OpenGL.lib<br>opengl32.lib<br>glu32.lib<br>Qt5Widgets.lib |
| <b>CUDA</b>                      | Berechnungen auf GPU                           | 6.5     | cuda.lib<br>cudart.lib<br>cufft.lib<br>cufftw.lib   |
| <b>OpenGL Extension Wrangler</b> | Unterstützung von CUDA und doppelter Präzision | 1.11.0  | glew.lib  |

Tabelle 2: Liste verwendeter Bibliotheken

Anhand dieser Bibliotheken wird eine Art Interface zur Berechnung und Visualisierung von mathematischen Problemen auf einer GPU erarbeitet. Mit Kenntnissen in C/C++/Cuda/OpenGL können Modifikationen an den mathematischen Berechnungen und der Visualisierung des Programms vorgenommen werden.

### 3.5. Validierung der Methoden

Die aufgezeigte, programmierte Umgebung dient hauptsächlich der Spektralanalyse simulierter und aufgenommener Signale. Zur Umsetzung der Analyse wird die in CUDA verfügbare FFT verwendet. Um sicherzustellen, dass die Anwendungen korrekte Ergebnisse liefern, wird eine Validierung der Umgebung durchgeführt, indem die Anzeige der Spektralanalyse überprüft wird. Wie aus der numerischen Betrachtung hervorgeht, kann eine Simulation bei 2,4 MHz durchgeführt werden. Deshalb wird die Anzahl der Samples für die Validierung auf 8,36 MS festgelegt.

#### Darstellung von minimaler und maximaler Frequenz

Im ersten Schritt wird untersucht, ob die Frequenz von 1 Hz sowie die Frequenz in Höhe der Hälfte der Abtastrate korrekt angezeigt werden. Diese ergibt sich als maximale Frequenz nach dem Abtasttheorem.

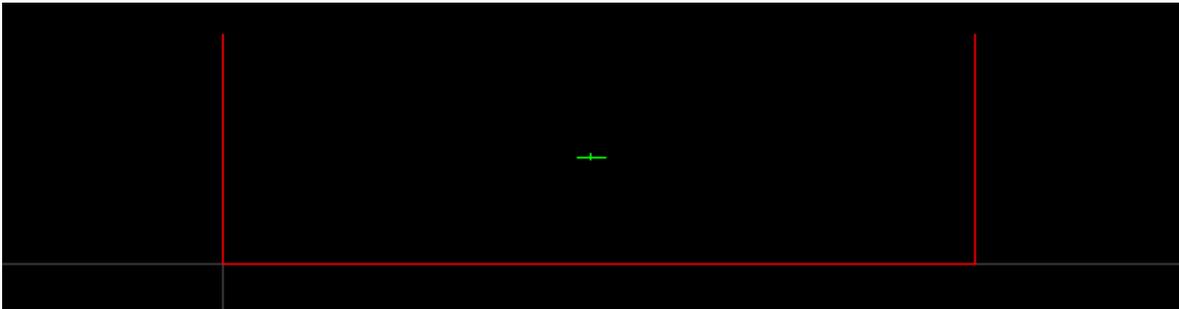


Abbildung 24: Darstellung der niedrigsten und höchsten Frequenzen.

Die Abbildung zeigt, dass die erstellten Frequenzen von 1 Hz und 4,18 MHz bei einer Samplerate von 8,36 MS korrekt dargestellt werden.

#### Validierung von Frequenzen eines Kanals nach IEEE 802.11

Unter Berücksichtigung der numerischen Betrachtung werden die 64 Trägerfrequenzen eines eigentlich 20 MHz breiten Übertragungskanal zur Validierung herangezogen.

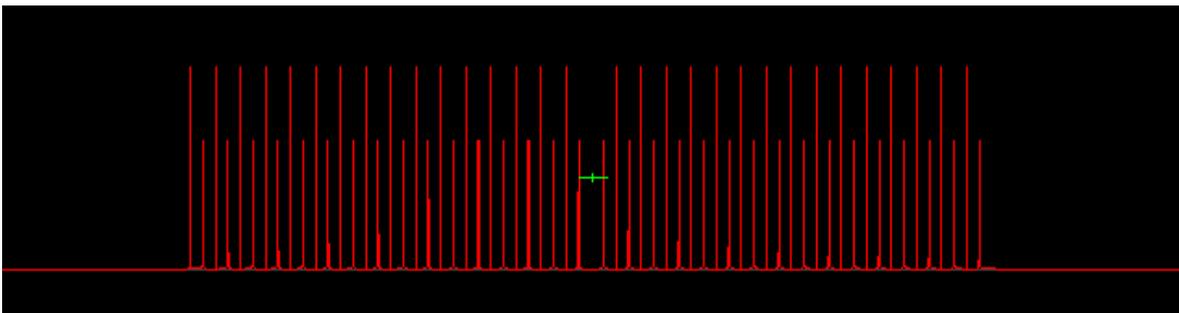
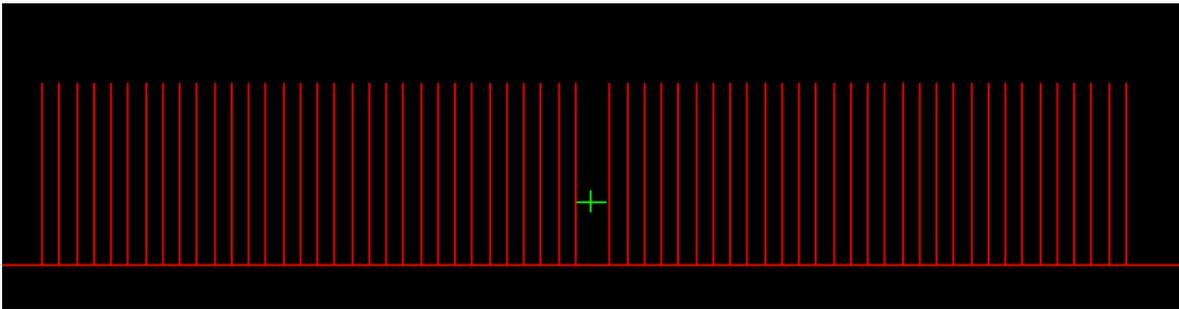


Abbildung 25: Darstellung der Trägerfrequenzen eines OFDM Symbols

Die dargestellten Frequenzen zeigen nicht alle dieselbe Amplitude. Der Grund hierfür ist, dass der Abstand der Trägerfrequenzen bei einer Bandbreite von 20 MHz 312.5 kHz beträgt. Die angewandte Kommaverschiebung der Werte ergibt demnach einen Abstand der Trägerfrequenzen von 312,5 Hz. Bei der FFT werden die Frequenzen, die ein halbes Hertz beinhalten, auf die nächst niedrigere und höhere Frequenz aufgeteilt.

Die Validierung zeigt, dass die Darstellung funktioniert, jedoch die aufgrund der Numerik durchgeführte Kommaverschiebung für die Simulation des IEEE 802.11 Standards in der Spektralanalyse nicht präzise genug ausgewertet wird.



**Abbildung 26: Darstellung der gerundeten Trägerfrequenzen eines OFDM Symbols**

Wird der Abstand der Trägerfrequenzen in einer Simulation von 312,5 Hz auf 312 Hz abgerundet, erübrigt sich das Problem. Weiterhin muss die Anzahl der Samples nicht gesteigert werden um das Abtasttheorem einzuhalten, wie es der Fall wäre, wenn die Kommaverschiebung nur um zwei Stellen durchgeführt wird.

### 3.6. Umsetzung einer IEEE 802.11 Simulation

Um eine Vorstellung davon zu bekommen, wie sich der Doppler-Effekt auf die modulierten Signale auswirkt, wird hier versucht eine Simulation zu erstellen. Wie in den bereits bestehenden Anwendungen aufgeführt, wird nun versucht eine OFDM umzusetzen. Nutzdaten sollen durch eine 16-QAM in das Signal moduliert werden. Hierfür wird versucht das Signal nach den Formeln 2.4.5 und 2.4.6.3 zu erzeugen. Die Simulation basiert auf rein mathematischen Methoden und lässt elektronische bausteinspezifische Gegebenheiten aus.

#### 3.6.1. Mögliche Nutzen der Simulation

Die Durchführung der Simulation soll dazu dienen, bisherige Annahmen zu bestätigen. Dies betrifft den aufgezeigten Hintergrund der Bandbreite, die Modulationsverfahren und die mathematische Betrachtung des Doppler-Effekts.

#### 3.6.2. Fragestellungen bei der Umsetzung

Eine grundlegende Frage gilt dem Zeitpunkt, an dem die Daten auf die Trägerfrequenzen moduliert werden. In der IEEE 802.11 wird angegeben, dass zur Übertragung eines Rahmens mit einer Bandbreite von 20 MHz eine Zeitspanne von 4  $\mu$ s notwendig ist. Aus der Übertragungsdauer ergibt sich bei einer Frequenz von 2,4 GHz, dass 9600-mal die Wellenlänge übertragen wird. Unklar ist hierbei, an welchen Stellen der Welle die Konstellationen der QAM in das Signal übertragen werden.

Es wird angenommen, dass die Konstellationen bei einer bestimmten Phasenlage des Signals aufmoduliert werden, um zu verhindern, dass diese durch den Nulldurchgang des Signals verloren geht. In der Umsetzung werden die Konstellationen jeweils bei einer Phasenlage von  $\pi/4$  mit einem Abstand von  $\pi/2$  auf die Trägersignale moduliert.

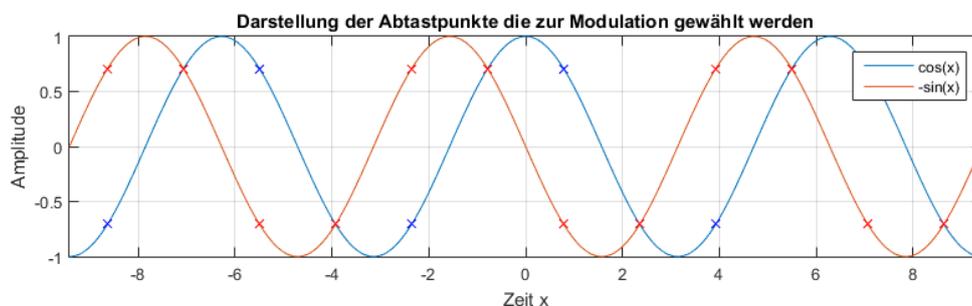


Abbildung 27: Darstellung der Abtastpunkte die zur Modulation gewählt werden

Werden die Konstellationen in kürzeren Abständen auf den Träger moduliert, so kommt es dazu, dass diese bei einem Nulldurchgang mit dem Signal multipliziert werden. Die Rekonstruktion der Konstellationen ist dann nicht mehr möglich.

Mit den gemachten Annahmen ist die Demodulation bzw. das Wiederberechnen der Konstellationen möglich. Für diese Annahme wurde in der IEEE 802.11 keine Angabe gefunden.

Aus der Formel 2.4.5.3 ergibt sich die Anwendung einer inversen FFT zur Umsetzung einer OFDM. Unter Berücksichtigung des Doppler-Effekts, muss die Transformation mit der Frequenzverschiebung implementiert werden. Hierzu wird das Signal aus Sinus und Kosinus Funktionen zusammengesetzt.

Hierzu ergibt sich die folgende Fragestellung:

Ein Signal, das mehrere Frequenzen beinhaltet wird normalerweise durch die Addition der Sinus und Kosinus Werte im Zeitbereich erstellt. Die Anwendung einer BPSK, QPSK oder QAM erfordert, dass die Werte, die für die einzelnen Trägerfrequenzen entstehen, mit unterschiedlichen Phasen oder Amplituden multipliziert werden.

Wird dies für alle Werte des Signals über die 53 Informationstragenden Frequenzen angewandt, beinhaltet jeder einzelne Abtastpunkt alle Informationen des zu übertragenden Nutzsignals. Wie kann hieraus die Phasen bzw. Amplituden der vorangehenden Modulation errechnet werden?

Der Versuch, die Phasen und Amplituden durch die Anwendung einer FFT – wie es in der IEEE 802.11 spezifiziert ist - zu rekonstruieren, bringt keine verwertbaren Ergebnisse.

### 3.6.3. Umsetzung der Modulation und Demodulation

Unter Berücksichtigung der Fragestellungen werden Funktionen zur Simulation der Modulation und Demodulation unter CUDA erstellt. Die Eingegebenen Konstellationen werden anhand dieser Funktionen korrekt wieder ausgegeben.

Die entwickelte Funktion zur Modulation lautet:

```
__global__ void modulation (float3 *out, unsigned int width, double doppler){  
  
    unsigned int x = blockIdx.x*blockDim.x + threadIdx.x;  
  
    double u = (x / (double)(width));  
    double phase = 0;  
    double freq = 0;  
    double I = 0;  
    double Q = 0;  
    int p = 0;  
  
    int i = x % 64;  
    p = (-32 + i);  
    freq = ((frequency)+(p*sidebands)) * doppler;  
    phase = ((CUDA_PI02)*x) + (CUDA_PI04 / freq);  
  
    I = cos(phase * freq) * (double)getData(false, &i);  
    Q = -sin(phase * freq) * (double)getData(true, &i);  
  
    out[x] = make_float3(u, I, Q);  
  
}
```

Die Funktion benötigt zur Eingabe die Länge des Signals und den nach Formel 2.2.1 berechneten Doppler-Effekt. Sie wird für jeden erstellten Abtastpunkt parallel aufgerufen. Es wird davon ausgegangen, dass jeder errechnete Abtastpunkt nur eine Trägerfrequenz und eine Konstellation besitzt. Die Konstellationen werden von der Funktion `getData(bool IQ, int n)` zur Verfügung gestellt.

Die entwickelte Funktion zur Demodulation lautet:

```
__global__ void demodulation (float3 *in, float3 *data, unsigned int width){  
  
    unsigned int x = blockIdx.x*blockDim.x + threadIdx.x;  
  
    double u = x / (double)(width);  
    double phase = 0;  
    double freq = 1;  
    double I = 0;  
    double Q = 0;  
    signed int p = 0;  
  
    int i = x % 64;  
    p = (-32 + i);  
    freq = ((frequency)+(p*sidebands));  
    phase = ((CUDART_PI02)*x) + (CUDART_PI04 / freq);  
  
    if (data[i].x == 0 && data[i].y == 0){  
        I = cos(phase * freq) * in[x].y * 2;  
        Q = -sin(phase * freq) * in[x].z * 2;  
    }  
    else {  
        I = (data[i].y + cos(phase * freq) * in[x].y * 2) / (double)2;  
        Q = (data[i].z - sin(phase * freq) * in[x].z * 2) / (double)2;  
    }  
  
    data[i] = make_float3(I, Q, i / (double)64); // (i % 16) / double(32));  
  
}
```

Die Demodulation findet statt, indem die vorher - in der Modulation erstellten - Abtastpunkte wieder mit dem Trägersignal multipliziert werden. Um die Amplitude anzugleichen, wird mit dem Faktor 2 multipliziert. Unterschieden wird, ob für die Konstellation bereits ein Wert vorhanden ist oder nicht. Ein Wert ist bereits vorhanden, wenn die Signallänge die 64 Abtastpunkte überschreitet. In diesem Fall wird zwischen dem vorhandenen Wert und dem bestehenden gemittelt.

### 3.7. Umsetzung einer Spektralanalyse auf einer GPU

Um den Einsatz einer GPU in Software-Radio Anwendungen zu hinterfragen wird eine Spektralanalyse auf einer GPU realisiert. Hierzu werden Datensätze aus einem Empfänger ausgelesen und im Frequenzbereich dargestellt.

#### 3.7.1. Idee der Umsetzung

Die Idee der Umsetzung basiert auf der in WiSee verwendeten Technik. Hierzu werden die Samples eines SDRs ausgelesen und solange aneinander gereiht, bis eine definierte Anzahl an Samples erreicht wird. Dies entspricht der Idee von WiSee nach der Formel 2.5.3 mit dem Unterschied, dass die Nutzdaten nicht angeglichen werden.

Mit dem eingesetzten Empfänger, kann maximal eine Frequenz von 1766 MHz aufgenommen werden. Anhand der maximalen Frequenz, die mit dem Empfänger überwacht werden kann, kann kein Signal nach dem IEEE 802.11 Standard überwacht werden. Der Zweck des Programms liegt darin, zu überprüfen, welche Bandbreite maximal mit der angegebenen GPU verarbeitet und visualisiert werden kann. Wichtig hierbei sind die Berechnungszeiten, sowie die Zeit die notwendig ist, um die Daten in den Speicher der GPU zu kopieren.

#### 3.7.2. Durchführung der Umsetzung

Zur Umsetzung werden die bisher genannten Bibliotheken verwendet. Hinzu kommt die Bibliothek rtl-sdr der OSMOCOM, die zum Ansprechen des Empfängers notwendig ist.

In Programmcode ausgedrückt wird folgendes realisiert:

```
generiere Array für Abtastpunkte auf GPU;  
wiederhole:  
    lese x Abtastpunkte aus Empfänger;  
    verschiebe alle vorhandenen Abtastpunkte auf GPU um x nach hinten;  
    kopiere neue Abtastpunkte auf GPU;  
    führe FFT über maximale Menge an Abtastpunkten durch;  
    zeichne Ergebnisse;  
solange bis: Abbruch durch Benutzer
```

Aus dem Empfänger werden pro Iteration rund 0.5 MS ausgelesen. Maximal liefert der Empfänger 3,2 MS/s. Das Array, indem die Samples auf dem GPU abgelegt werden, wird beim Initialisieren mit Nullen beschrieben. Hieraus ergibt sich je nach

maximaler Array-Größe eine Vorlaufzeit bis tatsächlich alle Werte des Arrays mit den Abtastpunkten des Empfängers belegt sind. Diese beträgt bei den durchgeführten Messungen zwischen wenigen Sekunden bis zu einer halben Minute.

Der Zweck des Programms ist wie oben erwähnt, die maximale Anzahl an Samples Herauszufinden, die mit der GPU im Spektralbereich dargestellt werden kann. Um hierüber einen genaueren Aufschluss zu erhalten, werden die Zeiten der einzelnen Berechnungen gemessen.

### 3.7.3. Erste Ergebnisse

Folgend werden die Ergebnisse zur Umsetzung der Spektralanalyse auf einer GPU präsentiert. Gemessen werden die Zeiten der folgenden Berechnungen:

- Entstehender Overhead durch CUDA (Mapping und Unmapping)
- Kopieren des Speichers
- Durchführen der FFT
- Zeichnen der Ergebnisse der Spektralanalyse

Hierzu werden drei Messungen durchgeführt:

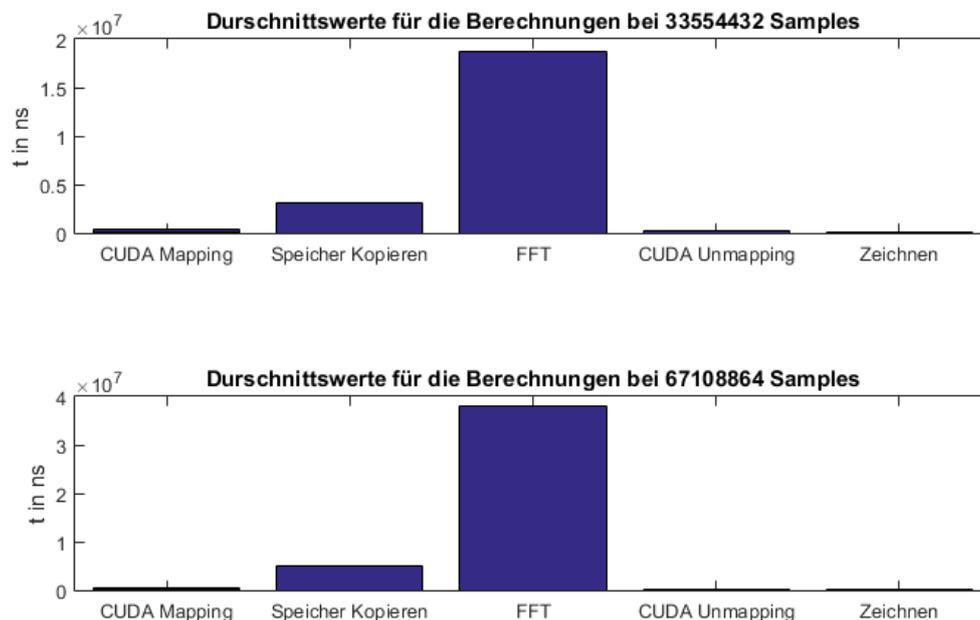
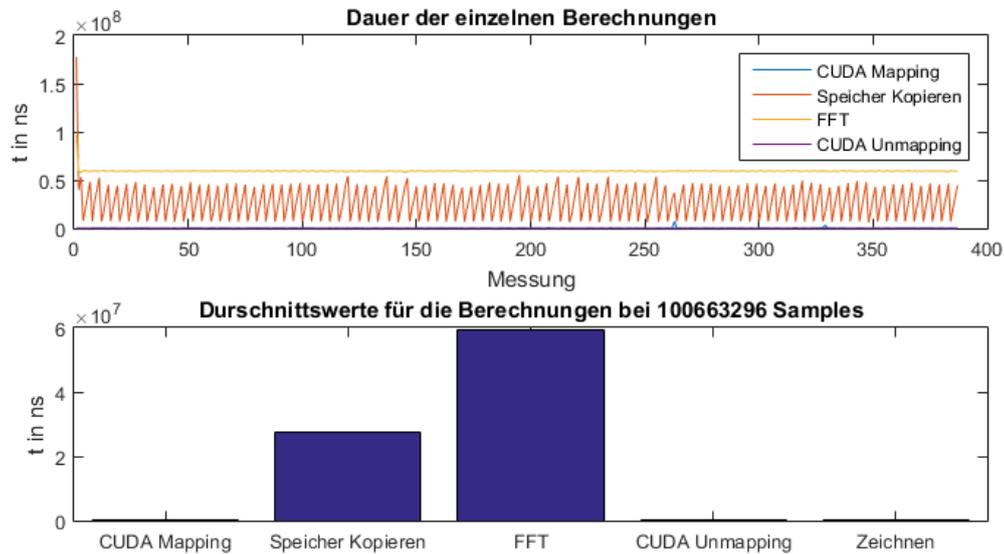


Abbildung 28: Ergebnisse der Messung mit rund 33 MS und 67 MS.

Die ersten beiden Messungen werden mit ungefähr 33 MS und 67 MS durchgeführt. Die Abbildung 28 zeigt, dass die Dauer der einzelnen Vorgänge im Millisekunden-Bereich liegt.

Die größte mögliche Anzahl an Samples die hier pro Berechnung erreicht werden liegt bei rund 100 MS. Hierzu werden die Ergebnisse genauer dargestellt.



**Abbildung 29: Ergebnisse der Messung mit rund 100 MS**

Abbildung 29 zeigt, dass die Berechnungen sogar bei einem Datendurchsatz von 100 MS im Millisekunden-Bereich liegen. Bei einer weiteren Erhöhung der Samplezahl ergibt sich das Problem, dass der Arbeitsspeicher der GPU nicht mehr ausreicht.

Um die tatsächlich anfallende Speicherlast zu messen, wird mit NVIDIA Inspector die aktuelle Auslastung der GPU Prozessoren und des GPU Speichers während der Laufzeit des Programms überwacht (siehe Abbildung 30). Hieraus geht hervor, dass die Berechnungen mit 100 MS eine Speicherlast von ungefähr 2,7 GB verursachen. Diese resultiert aus dem benötigten Speicher für das Ablegen der Samples und zusätzlich notwendigem Speicher, der für die FFT benötigt wird. Dies zeigt, dass der Speicher der GPU, der 4 GB groß ist am oberen Limit ist.

Die resultierende Visualisierung der der Spektralanalyse wird in folgender Abbildung ersichtlich.

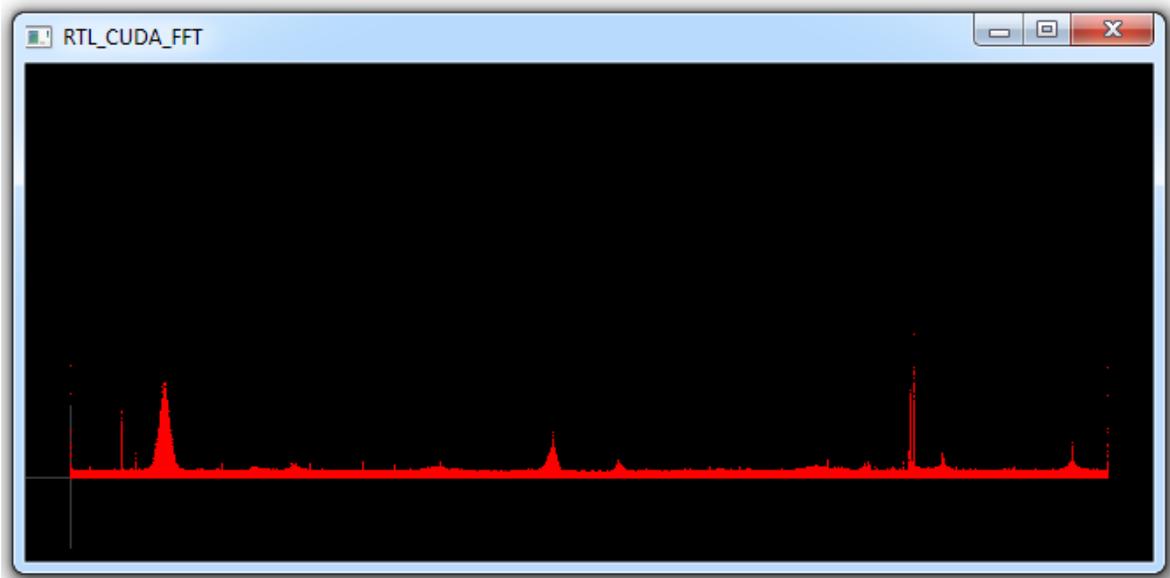


Abbildung 30: Dargestellt wird das Spektrum um 100 MHz, das mit 100 MS/s berechnet wird.

Zusammenfassend kann folgende Tabelle erstellt werden:

| Anzahl der Verwendeten Samples | Durchschnittliche Dauer der Berechnung und des Zeichnens |
|--------------------------------|--|
| 33554432                       | 22 ms  |
| 67108864                       | 43 ms  |
| 100663296                      | 87 ms  |

Tabelle 3: Ergebnisse der SDR Umsetzung

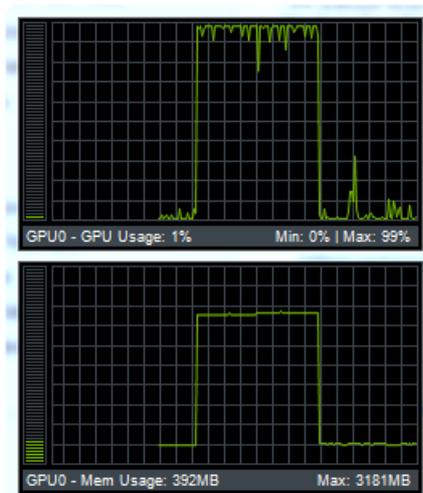


Abbildung 31: Last auf der GPU während des Programmablaufs

## **4. Ergebnisse**

Die hier aufgeführten Ergebnisse beziehen sich auf den aktuellen Projektstand und beinhalten die Auswertung mathematischer Betrachtungen, die Umsetzung der Implementierung sowie die Performanz der Simulation.

### **4.1. Betrachtung des Doppler-Effekts in der IEEE 802.11**

Die mathematische Betrachtung des Doppler-Effekts nach dem IEEE 802.11 Standard hat ergeben, dass die durch den Doppler-Effekt induzierten Frequenzverschiebungen auf den unterschiedlichen Trägerfrequenzen verschiedene Größenordnungen besitzen. Die Untersuchung geht davon aus, dass die Bewegung unmittelbar zwischen Sender und Empfänger verläuft. Zwischen der obersten und untersten Trägerfrequenz der Datenübertragung resultiert bei einer Geschwindigkeit von 20 m/s ein Unterschied der Frequenzverschiebung um 1 Hz. Eine Bewegung von 1 m/s ergibt mathematisch betrachtet eine Frequenzverschiebung um 8 Hz. Durch diese Angaben ist es möglich, festzulegen welche Bandbreitenauflösung zur Detektion des Doppler-Effekts notwendig ist.

### **4.2. Erstellung der Umgebung zur Signalverarbeitung auf einer GPU**

Im Rahmen der durchgeführten Untersuchungen und Umsetzungen war es notwendig immer wieder große Datenmengen zu verarbeiten. Bei der Auswertung von empfangenen Daten in SDRs ist es üblich, die Berechnungen so auszuführen, dass sie in annähernder Echtzeit umgesetzt werden können. In vielen SDRs sind hierfür FPGAs ein fester Bestandteil. Dies sind zusätzliche Hardwarekomponenten, die aufwendig programmiert werden müssen. Um dieses Problem zu umgehen, wurde eine Umgebung erstellt, die es ermöglicht die Daten in ähnlicher Geschwindigkeit auf modernen Grafikkarten (GPUs) zu verarbeiten. Es ergeben sich die Vorteile, dass die Umsetzung von Algorithmen zur Weiterverarbeitung in hohen Programmiersprachen erstellt werden können und die GPU auch zur Darstellung der Datensätze/Signale eingesetzt werden kann.

Um dies zu realisieren wurde in C/C++ mit dem Qt-Framework und den CUDA und OpenGL Bibliotheken eine Umgebung geschaffen, die es ermöglicht in wenigen Schritten Änderungen an Berechnungen durchzuführen. Die Umgebung wurde zur Spektralanalyse von Signalen validiert. Sie ermöglicht sowohl die Umsetzung von aufgenommenen Signalen als auch die Simulation von Signalen.

### **4.3. Versuch der Simulation des Doppler-Effekts**

In der hier beschriebenen Theorie wird die Funktionsweise der drahtlosen Datenübertragung nach dem IEEE 802.11 Standard detailliert betrachtet. Hierzu wird mit der Formel 2.4.6.3 eine Möglichkeit aufgezeigt, die Frequenzverschiebungen des Doppler-Effekts in die Erstellung eines Simulierten Signals einfließen zu lassen.

Die Praktische Umsetzung der Simulation brachte keine auswertbaren Ergebnisse, jedoch neue weiterführende Fragestellungen bezüglich der Simulation von Modulationen. Die im Standard gegebenen Informationen sind zur technischen Umsetzung und nicht zur Simulation der Signale geeignet. Die neuen Fragestellungen (siehe Kapitel 3.6.2) können aufgrund der Komplexität des Themas als Teilergebnisse zur Simulation betrachtet werden.

#### **4.4. Umsetzung eines SDR mit GPU Unterstützung**

In WiSee, einer in der Arbeit erläuterten Technik, welche den Doppler-Effekt in drahtlosen Anwendungen detektiert, wird dargestellt, dass durch das hintereinander Reihen von aufgenommenen Signalen einer Datenübertragung die Detektion des Doppler-Effekts möglich ist.

Die hierfür notwendige große Rechenleistung begründet in dieser Arbeit den Einsatz einer GPU. Um zu testen ob die Leistung einer GPU an dieser Stelle ausreichend ist, werden die empfangenen Daten eines low budget SDRs mit der GPU nach dem oben beschriebenen Prinzip weiterverarbeitet.

Die Ergebnisse zeigen, dass das erstellte Programm in der Lage ist 100 MS innerhalb von rund 90 Millisekunden zu verarbeiten. Diese Zeit beinhaltet das Kopieren der Daten in den GPU Speicher, die Berechnung der FFT und die Visualisierung im Frequenzbereich.

Bedenken, dass beispielsweise die Verbindung zwischen dem SDR und der GPU zu langsam ist, können anhand der Ergebnisse der Arbeit ausgeräumt werden. Sowohl die Übertragung der Daten auf die GPU, als auch deren Verarbeitung ist auf der verwendeten Grafikkarte NVIDIA GTX 760 möglich. Limitiert wird die Anzahl der maximal zu verarbeitenden Daten durch den verfügbaren Arbeitsspeicher der GPU. In diesem müssen die Abtastpunkte abgelegt werden können und Ressourcen zur Weiterverarbeitung vorhanden sein.

Die 100MS reichen je nach Mittenfrequenz des Empfängers aus, um das Abtasttheorem einzuhalten. Empfangene Signale können hierdurch mit einer genügenden Auflösung dargestellt und auf die Wirkung des Doppler-Effekts untersucht werden.

## 5. Diskussion

Die Arbeit untersucht durchgängig den Bezug zwischen angewandten Modulationsverfahren und dem Doppler-Effekt in elektromagnetischen Wellen. Hierbei wird sich größtenteils auf die Auswertung von mathematischen Funktionen verlassen. Die Überprüfung der Funktionen in praktischen Anwendungen konnte in der Arbeit nur bedingt realisiert werden. Für eine praktischere Vorgehensweise fehlte die Möglichkeit Signale in geeigneter Qualität aufzunehmen. Die gegebenen Möglichkeiten führten dazu, dass versucht wurde den Einfluss des Doppler-Effekts zu simulieren. Der Versuch der Simulation blieb weitgehend erfolglos, jedoch wurde hierdurch ein tiefes Verständnis für die Vorgänge bei der Modulation und Demodulation von Signalen erlangt.

Weiterhin basieren Teile der hier angefertigten Ausarbeitungen auf der Idee von WiSee. Zu dieser Idee ist keine detaillierte Dokumentation verfügbar, lediglich das Prinzip wird im zugehörigen Paper beschrieben. Dieses beinhaltet keine detaillierten Beschreibungen zur eingesetzten Empfangstechnik. Diese ist zur Detektion des Doppler-Effekts ebenso entscheidend, wie die Methoden zur Weiterverarbeitung der Signale. Bei einer genaueren Betrachtung des Inhalts des WiSee Papers kann man annehmen, dass die tatsächliche technische Umsetzung nur bedingt den beschriebenen Formeln und Vorgehensweisen entspricht.

Die erstellten Umsetzungen zeigen, dass der Einsatz von GPUs in Software-Radios durchaus sinnvoll und zweckmäßig ist. Diskussionswürdig sind die eingesetzten Frameworks zur Umsetzung. Die Verwendung von CUDA erleichtert die Realisierung von Ideen auf einem GPU, da bereits viele Funktionen wie beispielsweise die FFT vorhanden sind. Theoretisch wäre auch eine Umsetzung nur mit OpenGL als Schnittstelle zur GPU möglich. Dies hätte den Vorteil, dass versucht werden könnte den Speicherverbrauch, der in dieser Arbeit als limitierender Faktor gilt, einzuschränken. Hierdurch könnte mit der gleichen Hardware ein noch größerer Datendurchsatz erreicht werden.

Eine weitere Leistungssteigerung kann nicht nur durch den Einsatz von besseren Grafikkarten erreicht werden, sondern auch durch das Zusammenschalten mehrerer Grafikkarten. Hieraus ergibt sich jedoch eine neue Problematik, die gesondert untersucht werden muss.

## **6. Ausblick**

Komplexe praktische Anwendungen des Doppler-Effekts in elektromagnetischen Wellen sind bisher nur im Flugverkehr, der Marine und beim Militär zu finden. Die hier aufgezeigten, bisher nur von der Wissenschaft verwendeten, Umsetzungen zeigen, dass sich dies in der Zukunft ändern kann. Hieraus können sich für Privathaushalte und Industrie neue Anwendungsgebiete ergeben, die bisher noch nicht absehbar sind. Dazu ist es notwendig, den Doppler-Effekt mit möglichst einfachen Mitteln einsetzen zu können. Diese Arbeit ist ein Schritt in die Richtung, die Auswertung des Doppler-Effekts durch den Einsatz von Grafikkarten zu vereinfachen.

## 7. Abbildungsverzeichnis

|  |    |
|--|----|
| Abbildung 1: Interferenzen, bei elektromagnetischen Wellen.....                              | 2  |
| Abbildung 2: Kombinationen von Modulationsverfahren .....                                    | 3  |
| Abbildung 3: Auszug aus SDRSharp.....  | 8  |
| Abbildung 4: verschiedene Bandbreitenaufösungen desselben Signals .....                      | 9  |
| Abbildung 5: Mathematische Grundlagen analoger Modulationsverfahren .....                    | 12 |
| Abbildung 6: Erklärungen zum Konstellationsdiagramm .....                                    | 13 |
| Abbildung 7: Exemplarische Konstellationsdiagramme der BPSK und QPSK .....                   | 14 |
| Abbildung 8: Beispielhaftes Konstellationsdiagramm einer 16-QAM.....                         | 15 |
| Abbildung 9: Darstellung der verwendeten Bandbreite .....                                    | 17 |
| Abbildung 10: schematische Aufstellung der einzelnen Trägerfrequenzen .....                  | 18 |
| Abbildung 11: Blockdiagramm zum Senden eines Datenstroms .....                               | 19 |
| Abbildung 12: Beispiel eines Signals im Spektrum.....  | 20 |
| Abbildung 13: Blockdiagramm zum Empfang eines Datenstroms.....                               | 21 |
| Abbildung 14: Blockdiagramm und Antennen Array des IPS-265 .....                             | 22 |
| Abbildung 15: Mögliche Gesten bei WiSee .....  | 23 |
| Abbildung 16: Beispiel eines FPGA Entwickler-Boards.....                                     | 27 |
| Abbildung 17: Der Ettus USRP N210 Software-Radio .....                                       | 28 |
| Abbildung 18: Frequenzverschiebung des Doppler-Effekts.....                                  | 32 |
| Abbildung 19: Frequenzverschiebung bei unterschiedlichen Trägerfrequenzen ..                 | 32 |
| Abbildung 20: Verdeutlicht wird die unterschiedliche Steigung der Geraden .....              | 33 |
| Abbildung 21: Numerische Betrachtung kleiner Frequenzverschiebungen .....                    | 34 |
| Abbildung 22: Zeiten der einzelnen Vorgänge.....   | 36 |
| Abbildung 23: Darstellung der mit CUDA berechneten Signale.....                              | 37 |
| Abbildung 24: Darstellung der niedrigsten und höchsten Frequenzen. ....                      | 39 |
| Abbildung 25: Darstellung der Trägerfrequenzen eines OFDM Symbols .....                      | 39 |
| Abbildung 26: Darstellung der gerundeten Trägerfrequenzen eines OFDM Symbols.....            | 40 |
| Abbildung 27: Darstellung der Abtastpunkte die zur Modulation gewählt werden                 | 41 |
| Abbildung 28: Ergebnisse der Messung mit rund 33 MS und 67 MS. ....                          | 46 |
| Abbildung 29: Ergebnisse der Messung mit rund 100 MS.....                                    | 47 |
| Abbildung 30: Dargestellt wird das Spektrum um 100 MHz, das mit 100 MS/s berechnet wird..... | 48 |
| Abbildung 31: Last auf der GPU während des Programmablaufs .....                             | 48 |

## 8. Literaturverzeichnis

Adib F. und Katabi D., 2013: See Through Walls with Wi-Fi!, Massachusetts: Massachusetts Institute of Technology

B+B Sensors: Datenblatt Radar Bewegungsmelder-Modul Stereo RSM2650

Böge W. und Plaßmann W, 2007: Formeln und Tabellen Elektrotechnik, Springer Verlag

Chen, V., 2006: Micro-Doppler Effect in Radar: Phenomenon, Model and Simulation Study, IEEE Transactions on aerospace and electronic systems, Vol. 42, No. 1

IEEE 802.11, 2012: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications

Quifan P. et al., 2013a: Whole-Home Gesture Recognition Using Wireless Signals, Washington: University of Washington

Nyländen T. et al., 2010: A GPU Implementation for two MIMO-OFDM Detectors, Finland: University of Oulu

NI-Tutorial-3740-en, 2014: OFDM and Multichannel Communication Systems

NI-Tutorial-3983-en, 2014: Resolution Bandwidth (RBW)

NI-Tutorial-2898-en, 2014: Quadrature Amplitude Modulation (QAM)

NI-Tutorial-6984-de, 2012: Einführung in die FPGA-Technologie: Die 5 größten Vorteile

Stretch J., v2.2: IEEE 802.11 WLAN - packetlife.net

Terrasic, 2012: DE-0-Nano User Manual

## 9. Tabellenverzeichnis

|  |    |
|--|----|
| Tabelle 1: Liste verschiedener Software Radios ..... | 6  |
| Tabelle 2: Liste verwendeter Bibliotheken .....      | 38 |
| Tabelle 3: Ergebnisse der SDR Umsetzung .....        | 48 |
| Tabelle 4: Vergleich der Grafikkarten .....          | 56 |

## 10. Appendix

### 10.1. Vergleich der Grafikkarten:

| Modellbezeichnung   | Quadro FX 1700  | GV-N760OC-4GD rev2.1  |
|---------------------|---|-----------------------|
| <b>Allgemein</b>    |   |                       |
| Cuda Kerne:         | 32  | 1152                  |
| CUDA Version:       | 1.1   | 3.0                   |
| Chiptakt:           | 460 MHz   | 1084 MHz bis 1228 MHz |
| Bus Interface:      | PCI-E 1.0 x16   | PCI-E 3.0 x16         |
| Datenrate für PCIe: | 4 GB/s  | ~15 GB/s              |
| <b>Speicher</b>     |   |                       |
| Busbreite:          | 128 Bit   | 256 Bit               |
| Speichertakt:       | 400 MHz   | 3004 MHz              |
| Größe:              | 512 MB  | 4 GB                  |
| Speichertyp:        | DDR2  | DDR5                  |
| ECC:                | Ja  | Nein                  |
| <b>Sonstiges</b>    |   |                       |
| OpenGL Version:     | 2.1   | 4.5                   |
| <b>Quelle:</b>      | <a href="http://de.wikipedia.org/wiki/Nvidia_Quadro">http://de.wikipedia.org/wiki/Nvidia_Quadro</a> | Testsystem            |

Tabelle 4: Vergleich der Grafikkarten

## 10.2. Matlab Umsetzung zur Betrachtung des Doppler-Effekts

```
fcenter = 2.411e9;
c = 299792458;

x = 1:0.005:500;
dx = zeros(3,length(x));

for i = -1:1:1
    f(i+2) = fcenter + (312.5e3 * 26 * i)
    for n = 1:length(x)
        dx(i+2,:) = (f(i+2) + x)/f(i+2);
    end
end

figure;
subplot(1,2,1);
plot(x,abs(v1))
title('Doppler-Effekt aus Formel 2.2.1')
xlabel('Frequenzänderung in Hz')
ylabel('Geschwindigkeit in m/s')
legend('Unterste Trägerfrequenz 2.4029GHz','Mittlere Trägerfrequenz 2.4110 GHz','Oberste Trägerfrequenz 2.4191 GHz')
set(gca,'xLim',[2.9 3.1]);

subplot(1,2,2);
plot(x,abs(v3))
title('Transversaler Doppler-Effekt auf Formel 2.2.3')
xlabel('Frequenzänderung in Hz')
ylabel('Geschwindigkeit in m/s')
legend('Unterste Trägerfrequenz','Mittlere Trägerfrequenz','Oberste Trägerfrequenz')
legend('Unterste Trägerfrequenz 2.4029GHz','Mittlere Trägerfrequenz 2.4110 GHz','Oberste Trägerfrequenz 2.4191 GHz')
set(gca,'xLim',[2.9 3.1]);

figure;
subplot(1,2,1);
plot(x,abs(v1))
title('Doppler-Effekt aus Formel 2.2.1')
xlabel('Frequenzänderung in Hz')
ylabel('Geschwindigkeit in m/s')
legend('Unterste Trägerfrequenz 2.4029GHz','Mittlere Trägerfrequenz 2.4110 GHz','Oberste Trägerfrequenz 2.4191 GHz')
set(gca,'xLim',[7.9 8.2]);
set(gca,'yLim',[0.98 1.02]);

subplot(1,2,2);
plot(x,abs(v1))
title('Doppler-Effekt aus Formel 2.2.1')
xlabel('Frequenzänderung in Hz')
ylabel('Geschwindigkeit in m/s')
legend('Unterste Trägerfrequenz','Mittlere Trägerfrequenz','Oberste Trägerfrequenz')
legend('Unterste Trägerfrequenz 2.4029GHz','Mittlere Trägerfrequenz 2.4110GHz','Oberste Trägerfrequenz 2.4191GHz')
set(gca,'xLim',[479 486]);
```

## **11. Eidesstattliche Erklärung**

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, alle Ausführungen, die andere Schriften wörtlich oder sinngemäß entnommen wurden, kenntlich gemacht sind und die Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Heilbronn, April 2015

Stefan Völker