Synchronization in OFDM Receivers for a TTP Wireless Interface

Master's Thesis

Thomas Raggl

submitted at
Institute of Broadband Communications
Graz University of Technology

in cooperation with
Department of Aviation
FH JOANNEUM Graz (University of Applied Sciences)

Supervisors: Prof. Dr. Erich Leitgeb (TU Graz)
Prof. Dr. Holger Flühr (FH JOANNEUM Graz)

Graz, January 2012

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

....

date (signature)

Foreword

The thesis presented here was developed in the course of the Master's thesis at Graz University of Technology at the Institute of Broadband Communications and in cooperation with the FH JOANNEUM (University of Applied Science) in Graz, from April to December 2011. Because of my personal interest in radio controlled model aircrafts and the strive for challenges in the field of system-on-chip design, I approached the Department of Aviation at FH JOANNEUM for a possible thesis topic. Luckily the head of the avionics group, Prof. Dr. Holger Flühr gave me the chance to conduct a very interesting problem related to remotely piloted aircrafts which was embedded in an ongoing research project. During the work on this thesis I faced several different challenging problems where I was able to acquire profound knowledge about different aspects of design process and delve into novel approaches in order to answer those problems.

I would like to thank Holger for giving me this opportunity, for his continuous ambitious support and for incorporating me to the team. Speaking of the team, I also would like to thank Klaus and Mario for their support and the enjoyable conversations we had.

My thanks go to Prof. Dr. Erich Leitgeb who agreed to supervise this thesis without hesitation and the many interesting and amusing Optikom meetings at TU Graz. Many thanks to the whole Optikom group, who always made me feel welcome and for accepting me as a part of the group.

Finally, a big thank you to my family, my parents, who always encouraged me on my journey, who supported my in every possible way to enable my studies in Graz. Many thanks go to my good friend Tim and especially my understanding girlfriend Verena who always stood by my side, even in difficult times.

Thomas Raggl, December 2011

Abstract

Remotely piloted aircrafts (RPAs) or vehicles (RPVs), also known as unmanned aerial vehicles (UAVs) or drones, have gained an increased interest for various application scenarios in the military and public sector. The Department of Aviation at the FH JOANNEUM in Graz is currently working on an experimental RPV serving as a future research platform. One specific project, called "CertLink" is concerned about the development of a certifiable datalink between the ground control station and the aerial vehicle which will employ an orthogonal frequency division multiplex (OFDM) scheme on the physical layer to realize a flexible, robust and redundant digital communication system. Based on a proprietary time-triggered protocol (TTP) a CertLink modem is designed to employ a "Wireless-TTP" datalink for safety-critical data exchange, providing a high degree of reliability and availability to a system intended for the aviation industry.

This thesis encompasses a general introduction to the CertLink concept and the proposed system architecture for the deployment. A comprehensive summary of the theoretical fundamentals of OFDM systems together with a unified mathematical model is provided along with the present challenges faced in the system design. One major challenge is concerned about the inherent sensitivity of OFDM systems to timing and frequency errors. Estimating and correcting for these errors is a prerequisite to the entire system design and can be summarized as the synchronization problem. Synchronization in OFDM receivers is the main focus of this thesis with an emphasis on a feasible module implementation in a field programmable gate array (FPGA) hardware.

As a pre-step to the actual hardware implementation a complete, fully configurable end-to-end OFDM system simulation model in MATLAB is developed in order to provide a reasonable testbed for dedicated synchronization algorithms. Selected algorithms are evaluated together with the overall system performance, establishing a foundation for the understanding of OFDM based modulation schemes and the receiver design. For the purpose of implementation a Model-Based Design approach is chosen by utilizing Simulink as a unified design environment, capable of generating hardware description language (HDL) code from an abstract model. The Simulink model implements the specific system requirements as a configurable, flexible, high-level end-to-end system model to realize an automated design flow for an appropriate hardware implementation and cosimulation. Within the OFDM receiver a mathematical description of a suitable synchronization algorithm is mapped to a hardware module demonstrating the capabilities of the design approach. Using a third-party hardware simulation tool the in-system verification is accomplished by cosimulation confirming the plausibility of the implementation results. A concluding discussion identifies some of the shortcomings of the simulations along with the practical difficulties, experienced during the realization of the design flow.

Kurzfassung

Unbemannte Luftfahrzeuge (RPAs, UAVs), auch Drohnen genannt, haben ein erhöhtes Interesse für verschiedene Anwendungsszenarien im militärischen und öffentlichen Sektor gewonnen. Die Abteilung für Luftfahrt an der FH JOANNEUM in Graz arbeitet derzeit an einem experimentellen RPA, welches als zukünftige Forschungsplattform dienen soll. Ein konkretes Projekt, genannt "CertLink" beschäftigt sich mit der Entwicklung einer zertifizierbaren Datenverbindung zwischen der Bodenstation und dem Luftfahrzeug, das ein orthogonales Frequenzmultiplexverfahren (OFDM) auf der physikalischen Schicht einsetzt, welches ein flexibles, robustes und redundantes digitales Kommunikationssystem realisieren wird. Basierend auf einem proprietären Time-Triggered Protocol (TTP) wird ein CertLink Modem entworfen, das einen "Wireless-TTP" Datenlink für sicherheitskritische Datenübertragungen einsetzt, um so ein hohes Maß an Zuverlässigkeit und Verfügbarkeit für ein in der Luftfahrt eingesetztes System zu bieten.

Diese Arbeit umfasst eine allgemeine Einführung in das CertLink Konzept und die geplante Systemarchitektur für den Einsatz dieses Systems. Eine umfassende Übersicht über die theoretischen Grundlagen der OFDM-Systeme, zusammen mit einem einheitlichen mathematischen Modell wird in Verbindung mit den gegenwärtigen Herausforderungen des System-Designs vorgestellt. Eine der größten Herausforderungen in OFDM-Systemen beschäftigt sich mit der vorhandenen Empfindlichkeit gegenüber Zeit- und Frequenzfehlern. Die Schätzung und Korrektur dieser Fehler ist eine Voraussetzung für das gesamte System-Design und kann als das Synchronisationsproblem zusammengefasst werden. Das Hauptaugenmerk dieser Arbeit liegt sich auf der Synchronisation in OFDM-Empfängern sowie auf einer praktikablen Umsetzung eines solchen Moduls in einem Field Programmable Gate Array (FPGA).

Als vorbereitender Schritt für die eigentliche Hardware-Implementierung wird ein vollständiges, voll konfigurierbares End-to-End Simulationsmodell eines OFDM Systems in MATLAB entwickelt, um eine vernünftige Testumgebung für dedizierte Synchronisationsalgorithmen zu bieten. Ausgewählte Algorithmen werden zusammen mit der gesamten Systemperformance evaluiert und ausgewertet, um eine Grundlage für das Verständnis der OFDM basierten Modulationsverfahren und des Empfänger-Designs zu schaffen. Für den Zweck der Umsetzung wird ein Model-Based Design Ansatz gewählt, welcher durch die Verwendung von Simulink eine einheitliche Design-Umgebung zur Verfügung stellt, die für eine automatische Generierung von Hardware-Beschreibungssprachen (HDL) Code aus einem abstrakten Modell geeignet ist. Das Simulink-Modell implementiert die spezifischen Systemanforderungen sowie ein konfigurierbares, flexibles, High-Level End-to-End System-Modell, um einen automatisierten Design-Flow für eine entsprechende Hardware-Implementierung und Co-Simulation zu realisieren. Innerhalb des OFDM-Empfängers wird eine mathematische Beschreibung eines geeigneten Synchronisations-Algorithmus in einem Hardware-Modul abgebildet, um die Fähigkeiten des Design-Ansatzes zu demonstrieren. Die Verwendung eines Hardware-Simulationstools eines Drittanbieters erlaubt die In-System-Verifizierung durch Co-Simulation und bestätigt die Plausibilität der gewonnenen Ergebnisse. Eine abschließende Diskussion zeigt einige Schwachstellen in den erstellten Simulationen auf und beschreibt die Schwierigkeiten bei der praktischen Umsetzung des Design-Flows.

Contents

List of Figures

List of Tables

"The wireless telegraph is not difficult to understand. The ordinary telegraph is like a very long cat. You pull the tail in New York, and it meows in Los Angeles. The wireless is the same, only without the cat."

Albert Einstein

1 Introduction

Unmanned Aerial Vehicles[1] (UAVs) or Remotely Piloted Aircrafts (RPAs) are a product of the steady growing technological advance and will have a considerable impact in the near future for civil and military applications. When the Wright brothers took the first controlled, human flight in mid December 1903 no one expected the aeronautic evolution we experience today. From simple fixed wing gliders over piston engine powered aircrafts to the most advanced military fighter planes of the 21st century, we have seen a remarkable development in the world of aviation. It is not surprising that this process of extraordinary engineering has not come to an end. Despite the major inventions of the Wright brothers and the following years of advance, considered as the "golden age" of aviation, it's history is much older. Model aircrafts are reported to be developed and built already around 400 B.C. (the famous model, known as "The Dove" invented by the Greek Archytas). Those early inventions helped to develop a deeper knowledge on aerodynamics and flight characteristics, moreover providing essential building blocks for a better understanding of aviation.

Despite the fact that unmanned aviation itself is viewed as a independent domain in the aviation industry, it underwent a similar substantial progress. The interest in remotely controlling model aircrafts using a radio control evolved before World War I and became even more popular afterwards. During that time the principle of remotely controlling a vehicle or simply a device was found in several military applications, e.g. guided missiles, bombs or aircraft dummies for air defense training in WWII, mainly motivated to minimize the casualties in high risk battlefield operations. Later on, further inventions in the field of radio electronics and electronics in general (transistor based radio control systems) as well as the advance of miniature internal combustion engines contributed to the ongoing popularity of model aircrafts, especially for hobbyists.

1.1 Motivation

For a certain time building and flying those models was seen as a hobbyist phenomenon that had no common practical use, until the US military readopted the concept. When the USAF discovered the rising number of pilots and planes lost, they started thinking about unmanned aircrafts and began to incorporate UAVs in their combat missions, the first time reported during the Vietnam War. In the early 1970s Israel started developing UAVs on their own, mainly for reconnaissance purposes. The vehicles delivered important information about enemy air defense systems and led to a severe advantage in the progress of the Lebanon War back in 1982. From that time on UAVs were indispensable in the battlefield and are heavily in use

1 According to the ICAO an aerial vehicle must have a pilot, whether it is manned (pilot in the cockpit) or unmanned (pilot at the control station), human or a machine. So the term RPA should be chosen instead of UAV.

today. Military UAVs are targeted at reconnaissance and attack missions simply differentiated by their payload. The technological progress in electronics and microelectronics enhanced the capabilities and functionality of the vehicles and allowed for a massive miniaturization of the overall system platforms. Such complex systems are now comprised of several functional units ranging from a central airborne computer, wireless communication links over essential navigation systems (GPS and Inertial Navigation Systems) to peripheral systems like surveillance cameras or radar equipment. The platforms provide sufficient flexibility to adapt to a variety of application scenarios. Today UAV systems are either controlled by a remote control center (human operator) or can operate autonomously, using artificial-intelligence systems.

Disregarding the military motivated approach, the concept of an unmanned aerial vehicle is portable to other considerably more useful general public applications. Such "Civil or Commercial UAVs" are intended to assist and improve the efficiency of national authorities like fire brigades, emergency medical services or police forces and are an integral part of researching activities in various fields. The type of an UAV is not necessarily restricted to fixed wing aircraft and therefore many different platforms are imaginable, e.g. a helicopter, called RUAV (Rotor-craft Unmanned Aerial Vehicle) or light-weight platforms like the popular quad-copters, called MAV (Micro Aerial Vehicle). So for instance think of an unmanned remotely controlled helicopter hovering over a forest fire and delivering specific data about the fire and the surroundings via infrared cameras or other instrumentation. The collected data can then be directly transmitted to the firefighters in the field or a remote control center, providing essential information about possible future actions or countermeasures to undertake. The risk of firefighters being trapped by the fire is minimized while the fire fighting measures can be optimized. Other possible scenarios are search and rescue operations, national border control, research activities or inspection and maintenance services on critical exposed or remote infrastructures like, e.g. electric power lines or pipelines. Such observation tasks handled by UAVs have another main economic advantage: maintenance and operation cost. Due to their light weight construction and small size, the deployment of those vehicles is much more cost-effective than compared to, e.g. a manned airplane. Their in-field operation time is significantly increased while still having lower overall cost. Hence, these circumstances give rise to even broader market acceptance in the near future.

A typical system for early remote control consisted of an analog ground radio transmitter (user input) and an analog radio receiver in the aerial vehicle itself. The medium air in between, in this context considered as a radio channel or simply a communication channel, propagates the emitted electromagnetic wave that carries all the control information. At that time the conveyed information was composed of simple on/off commands used for rudimentary control. Today's systems have a demand for more sophisticated, high performance digital communication links that allow for real-time, secure, high availability and long-range data exchange. Information data is not only used for command and control purposes but also to transmit large amount of data, e.g. on-line video-streams. The basic system setup is the same today as it was decades ago. However the requirements have considerably changed over time and drive the need for different design and implementation approaches.

An Unmanned Aerial System (UAS) describes the overall systems consisting of an UAV together with a Ground Control Station (GCS) and the air-to-ground/ground-to-air communication link. The Department of Aviation at FH JOANNEUM (University of Applied Science) in Graz, Austria is involved in a research project named *"Certifiable Data Link in Time Triggered-*

Architecture for Remote Control of Unmanned Aerial Vehicles", short *"CertLink",* which is concerned about developing a certifiable wireless communication data link as well as an UAS system architecture for a highly available, real-time system [1],[2]. CertLink provides the **groundwork** for this thesis and therefore will be described in more detail in the next section.

1.2 The "CertLink" Project

The CertLink project is funded by TAKE OFF, the Austrian aeronautics research and technology program founded in 2002 by the Federal Ministry for Transport, Innovation and Technology (BMVIT) and managed by the Austrian Research Promotion Agency (FFG). It aims to enhance the competence of the Austrian aeronautics industry by supporting research and development activities of (cited from [3]):

- *"industrial and service enterprises"*
- *"researchers from universities and non-university institutions"*
- and *"users from the aeronautics sector"*

1.2.1 Project Partners

Two additional research partners, namely *TTTech Computertechnik GmbH* and *Airborne Vision GmbH*, both located in Vienna, actively support the project:

TTTech has a *"focus on solutions for safe and reliable networks for the transportation and automation industries"* and is considered as the *"leading supplier of solutions for highly reliable electronic networks based on time-triggered technologies"* for the aerospace and automotive industry. They established platforms for time-triggered communication and develop certifiable safety modules ensuring reliability and robustness in digital communication applications (see [4]). The Time-Triggered Protocol (TTP) bus is one of the products that is established as the leading system for time-triggered solutions in the aerospace industry and has been in operational use for several years now. New designs, e.g. Boeing's 787 Dreamliner use TTP for their electrical power supply and distribution system.

Airborne Vision on the other hand concentrates on engineering a vertical take-off and land (VTOL) civil drone platform, called "CINEcopter HD35" that features a stabilized, multi-axis gimbal for mounting professional high-definition cameras. Based on the self-stabilizing gimbal and a remote camera control, such a platform will provide high-quality pictures for the film industry, broadcasting and aerial photography [5].

1.2.2 Project Definition

As mentioned before typical UAS combine three major systems: UAV, GCS and a bidirectional air-to-ground data link. Current data links come in different flavors which vary from primary flight control, air traffic control to flight mission purposes for payload operations, each having differing design and implementation requirements. A Command-and-Control (C2) link for instance is viewed as low-data rate, safety critical link demanding a high degree of availability,

reliability and security, compared to a task of streaming video data that just requires high data throughput. Thus, the core specifications of a data link have to be formulated in a heterogeneous manner, whereas the implementation side has to provide enough flexibility to incorporate such a design.

Currently FH JOANNEUM is working towards an UAS that will serve as a common experimental research platform for several research groups and laboratories, including aircraft design, aerodynamics and avionics, just to name a few. Accordingly the fixed-wing UAV which is part of the entire UAS, is named JOANNEUM Experimental Platform (JXP). Having a wingspan of seven meters and a takeoff weight below 20 kg the vehicle is able to carry up to three kilograms of payload and is powered electrically using state of the art battery technology.

Central aspects of a UAS are the flight control as well as the flight management, both handled based on a time-triggered bus system (TTP) that allows for reliable data exchange between defined bus members (nodes). TTP employs a deterministic communication flow throughout the network and uses mechanisms to ensure a secure data exchange by means of redundant implementation. TTP networks can be assumed to be distributed over ground and airborne systems. Each having different tasks to fulfill in order to establish an overall system functionality. Airborne architectures may integrate communication between redundant fly-by-wire flight control systems whereas ground systems could interchange data from different parts of the mission planning system.

1.2.2.1 “Wireless-TTP”

These two isolated architectures are both realized by a wire-based physical layer which defines their electrical properties. From a layer’s perspective, TTP sits on top of the RS-485 standard that is implemented in a redundant fashion. The question arises, why not to integrate ground and airborne systems into a **unified** network to ease the communication flow and inherently provide a reliable, highly available, secure data link based on the TTP specification. This can be done by defining the data link using a newly designed **wireless** PHY-layer. A “Wireless-TTP” bus would then incorporate the UAS avionics from the UAV and the GCS into a single TTP bus system architecture.

CertLink undertakes measures for designing such a time-triggered UAS architecture, that is certifiable and uses Commercially available Off-The-Shelf (COTS) components while still maintaining high availability. The project aims at several research areas:

Air-to-ground interface: Decides on the specification and implementation of the bidirectional wireless data link (CertLink-Modem)

System architecture: Based on the TTP core specifications, define system architecture to extend the ground system avionics to the UAV by means of an TTP wireless data link. Develop ways of a Wireless-TTP operational deployment.

Options for hardware integration: Trade-off between cost-effective implementations using commercially available hardware components and a flexible integration for potential optimizations in a field programmable gate array (FPGA).

Feasibility studies, implementation and ways of certification: How to implement the proposed system under consideration of several development and certification processes used in the aviation industry.

Discussions about the system availability can be found in [2] whereas the concept of the datalink is presented in [1].

1.2.2.2 Design Issues

The design of an air-to-ground interface has to account for several problems that arise in wireless communication systems and the underlying specifications of TTP, to ensure the same characteristics as for wire-bound interfaces. Some of the key aspects are concerned about the frequency usage, realizable data rate, wireless range, channel models and the attainable signal-to-noise ratio. To this day, there are no aeronautic frequency bands available that are specifically intended for UAS data links. The issues concerning the frequency allocation and regulatory actions will be addressed in the upcoming ITU-R World Radiocommunication Conference 2012 in Geneva, see [6]. Hence the wireless link has to be established in license free bands like the industrial, scientific and medical (ISM) radio band. The choice of using the ISM band for communication comes with additional general requirements such as limited radiated power and bandwidth which all have to be met. Since the ISM band is not designed to be used exclusively, the data link under consideration has to be adaptable by careful selection of modulation methods and error correction mechanisms.

A time-triggered bus architecture like TTP defines a deterministic communication scenario by introducing a common global system time and very strict designed cycle-based data exchange. To meet the requirement of a reliable, highly available communication the protocol employs a replication mechanism for data validation. For systems with safety-critical data links (e.g. C2), TTP demands a redundant physical layer in case of hardware failure. Therefore, Wireless-TTP has to incorporate methods for a similar redundant architecture. To maintain a global system time the CertLink modem needs to deal with rigorous timing constraints associated with timing and frequency synchronization issues linked to multi-path fading and Doppler effects (moving aircraft). Over-the-air (OTA) transmission delays as well as signal processing delays of the modem play another important role with respect to the TTP specification.

1.2.2.3 Design Decisions

Some of the design solutions will be briefly presented here. First of all the proposed the architecture as well as the implementation of the time triggered wireless data link requires flexibility. Since the development of such a CertLink modem is an ongoing research topic, there will certainly arise the need for change. Not only because it has a researching nature but because it has to support different kinds of data also. Some of these examples were already mentioned (C2, video streaming). ISM band communication inherently calls for flexibility to meet specific link requirements like SNR and communication range.

Conventional transceivers architectures in the ISM band are usually tailored for a certain application (e.g. WLAN, ZigBee). Typically they are realized in an ASIC and their architecture (modulation scheme, spectral characteristics, data rate) is fixed by design. Additional components include the analog RF front-end and a central processing unit that handles the digital communication interface. As one can see, there's virtually no room for improvement or flexibility. Therefore, CertLink will use a different architectural approach by utilizing a software-defined radio (SDR). Shortly explained, this design incorporates most of the signal chain in

the digital domain. Ideally the incoming signal (viewed from the receiver's side) from the antenna is directly sampled by a high-speed ADC, so all the signal manipulation is done digitally. This (optimal) configuration allows for a maximum of flexibility and hence is well suited for a CertLink modem architecture. Two implementation options come into consideration: DSPs and FPGAs. There is a great number of pros and contras for either side - CertLink decided on a FPGA-solution since it is optimal for rapid prototyping and again, provides the most flexibility.

The second issue is concerned about the robustness and redundancy of the data link. It is mainly motivated by the TTP specification and is essential for a reliable and certifiable communication. One of the most challenging things when dealing with terrestrial communication and reception in moving objects is multipath-fading and interference. Typically, a wireless communication system, as implemented by CertLink, faces a time-varying and highly dispersive (frequency selective) mobile radio channel due to multipath reception and the moving of the UAV in 3D-space. Employing a typical single carrier digital modulation scheme can handle such distortions, but only with great effort in the mobile receiver. This is especially true when the data link requires high data rates. In recent years a new modulation scheme evolved that uses a mulitcarrier, FDM approach. The technique is called OFDM (Orthogonal Frequency Division Multiplex) and uses several orthogonal subcarriers to modulate the data. Current standards like WLAN or DVB-T are already based on OFDM schemes. The core idea here is, that by spreading the data over several carriers the data rate can be reduced (the symbol duration is lengthened) while still maintaining spectral efficiency. The prolonged symbols and other OFDM specific methods help to mitigate the problems encountered by time-varying impulse responses in a multipath environment and prevent inter-symbol interferences (ISI). Although OFDM is very complex in its structure, it is considered as one of the key technology for future digital communication systems. That is only one aspect why CertLink will employ OFDM as the modulation scheme of choice. The issues on redundancy are encountered by duplicating the data on different subcarriers, denoted as spectral redundancy. Additional research, related to multi-antenna system will show whether the reliability of the data link can be further improved.

Despite the robustness against the mobile channel effects, OFDM needs special attention to the phase of synchronization. This type of modulation is extremely sensitive to frequency errors and timing errors. Such frequency errors may be introduced by different transmitter and receiver clocks or Doppler shifts that all have to be compensated for. Most of the work presented here deals with the OFDM receiver architecture and ways of synchronization. Section 2.1 will cover the OFDM fundamentals and characteristics in more detail.

The overall UAS will be demonstrated for C2 links of the JXP and payload transmission for a camera gimbal control.

1.3 Research Objectives

The scope of this thesis shall incorporate a variety of different aspects in the field of an OFDM PHY layer design. Theoretical basics of OFDM systems with a focus on the synchronization problem and a brief overview of the expected aeronautical channel models should provide some fundamental knowledge of the problems faced during the design process. The system architec-

ture shall be examined in more detail in order to explain the TTP concepts together with most of the requirements of a CertLink modem.

The goal of this thesis is to cover the problems of synchronization in OFDM receivers and find ways to implement a feasible algorithm for hardware deployment in an FPGA system. In order to define criteria for an appropriate selection of the algorithm a MATLAB based simulation script will be developed to provide a basic evaluation environment. Essentially, a *fully configurable* end-to-end system simulation should cope every component in the signal chain, ranging from test data generation to OFDM modulation over a suitable channel model to the OFDM demodulation in the receiver. Within this environment several synchronization methods shall be implemented, tested and evaluated using adequate performance measures. Additionally, the overall system performance should be analyzed in terms of bit error rate (BER) measurements in comparison to similar systems.

Based on the simulation results one algorithm should be chosen for a mapping to hardware. A VHDL description of the algorithm as well as a suitable testbed for system verification in conjunction with a cosimulation environment shall be established. A testbed incorporates the entire signal model of a CertLink modem in order to provide an appropriate test case scenario covering several use cases (different link configurations) of the system.

2 System Theory

This chapter will provide the theoretical concepts essential for following the system development. Starting off with the basics of orthogonal frequency division multiplex (OFDM) modulation schemes, also including a mathematical approach the necessary components for the CertLink OFDM-based physical layer, will be discussed. After establishing a common basis of a well behaved system the sensitivity of timing and frequency offsets can be elaborated, summarized as the "synchronization problem" in Section 2.2. Some general issues on the problem of synchronization as well as dedicated algorithms will be part of the discussion. The fundamentals of mobile radio channels together with a compact mathematical description will be provided at the end to complete the chapter.

2.1 OFDM Fundamentals

OFDM is a special form of a multicarrier transmission scheme which can be seen as a modulation or a multiplexing technique, according to [7]. In the last few years the demand for wireless high-data rate broadband communication systems for mobile telecommunications users was continuously increasing. In order to establish broadband communication of several megabits per second over a wireless mobile radio channel new technologies had to be developed. Mobile radio channels are characterized by multipath propagation where the moving receiver platform not only sees a single line of sight signal path but also many scattered, delayed radio waves due to reflections in the environment (mountains, buildings, etc.). This multipath reception (fading) in the receiver causes inter symbol interference (ISI) and severely degrades the system performance.

Using high-data rates in such a communication system leads to very short symbol times that are within the range of the delayed signal paths, making extensive adaptive equalization in the receiver mandatory. OFDM is introduced as a parallel transmission scheme to overcome the difficulties to implement low-cost, high-data rate receivers operating in multipath fading environments. Using OFDM provides robustness against frequency-selective fading and narrowband interference compared to single carrier systems by dividing the high-rate data stream into several low-rate streams transmitted over multiple parallel subcarriers, or subchannels (SC). This form of frequency division multiplexing (FDM) allows the multicarrier system to cope with signal fading where only a fraction of the SC is distorted by the interferer. The few affected carriers can be corrected by error-correcting measures. Single carrier systems on the other hand could experience a total link failure.

Based on early research in 1960 this concept of FDM and parallel transmission was adopted to form a classical parallel transmission scheme. Previously, the available frequency band was split into N non-overlapping subchannels, each one modulated by a different symbol and multiplexed together, forming a multicarrier modulation (MCM) scheme. Such a non-overlapping spectra

required dedicated guard bands for the receiver to demodulate the data (analog filter and analog demodulators), hence, leading to an inefficient spectrum utilization. In 1960 overlapping subchannels where proposed to efficiently use the available spectrum and save bandwidth. Overlapping means to reduce the intercarrier interference by introducing mathematically orthogonal subcarriers. Fig. 2.1 illustrates the optimization in bandwidth utilization by using orthogonal (bandlimited) overlapping subcarriers in MCM systems, as seen in Fig. 2.1b, compare to a classical MCM scheme in Fig. 2.1a.

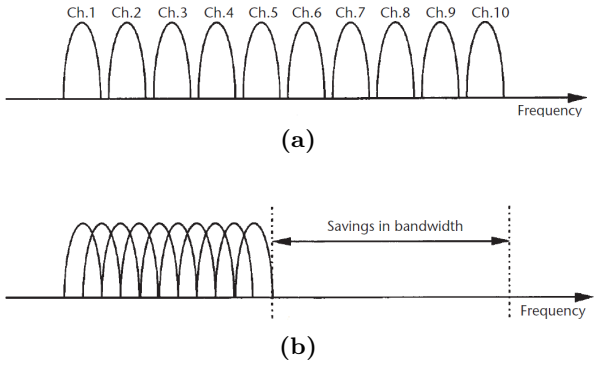


Figure 2.1: Optimization in bandwidth utilization by using orthogonal overlapping subcarriers, from [7]

Orthogonality allows to use a bank of demodulators in order to demodulate each subcarrier by shifting to DC and integrating over the symbol period. If these subcarriers are equally spaced by $1/T_u$, where T_u is the useful symbol period, the integration does not incorporate contributions from the adjacent carriers. Therefore the subcarriers are considered to be linearly independent or orthogonal. Weinstein and Ebert [8] proposed the use of the Discrete Fourier Transform (DFT), inverse Discrete Fourier Transform (iDFT) and digital implementations using the Fast Fourier Transform (FFT), inverse Fast Fourier Transform (iFFT) as a part of the modulation/demodulation process. In that way efficient high-speed implementations using special purpose computers were assumed to be feasible.

The modulated subcarriers occupy only a small bandwidth compared to the coherence bandwidth of the time-dispersive radio channel where the SCs only experience frequency flat fading, allowing for a simple equalization. Small bandwidths correlate to long symbol time durations; long compared to the time delay spread of the channel to minimize ISI.

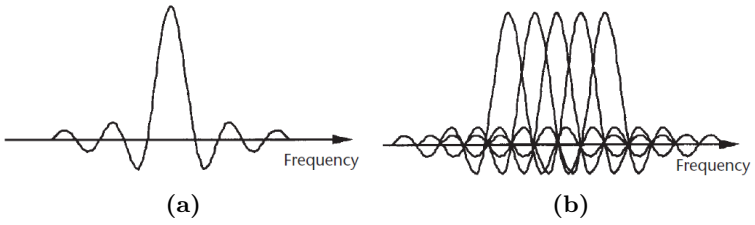


Figure 2.2: Frequency domain plot of a single subcarrier in **(a)** and the multiplexed overlapping spectra of an entire OFDM signal in **(b)**, from [7]

Recently several communication systems adopted OFDM for high-data rate transmission, e.g. ADSL, WLAN, DAB and DVB-T systems.

In summary some of the main OFDM advantages are listed below, from [7]:

- *"OFDM is an efficient way to deal with multipath; for a given delay spread, the implementation complexity is significantly lower than that of a single-carrier system with an equalizer."*
- *"In relatively slow time-varying channels, it is possible to enhance capacity significantly by adapting the data rate per SC according to the signal-to-noise ratio (SNR) of that particular SC."*
- *"OFDM is robust against narrowband interference because such interference affects only a small percentage of the SCs."*
- *"OFDM makes single-frequency networks possible, which is especially attractive for broadcasting applications."*

The drawbacks of the modulation scheme shall be the topic of the ongoing discussion as a basis for synchronization problems, also cited from [7]:

- *"OFDM is more sensitive to frequency offset and phase noise. "*
- *"OFDM has a relatively large peak-to-average-power ratio, which tends to reduce the power efficiency of the radio frequency (RF) amplifier."*

2.1.1 OFDM System Model

In the literature [8, 9, 10, 7, 11, 12] several different mathematical descriptions for OFDM systems are proposed to provide a solid background for understanding the concepts especially in conjunction with problems regarding synchronization. Although those mathematical models explain the same principles there is no consistency among those works - every author tries to invent a completely new mathematical model. Therefore, this subsection (and parts in Section 2.2) tries to define a unified and consistent OFDM system model with inspiration from the work in [7, 10, 11].

The typical signal chain of an OFDM system is composed of an OFDM transmitter and the corresponding OFDM receiver, both utilizing the iDFT/DFT as the central principle for modulating/demodulating the data constellations onto/from the orthogonal subcarriers.

2.1.2 OFDM Transmitter

The bitstream from the data source is typically encoded (FEC) and mapped to complex in-phase, quadrature-phase constellation points to form the symbols on the subcarrier level. The input samples of the iDFT consists of a block of N complex data symbols in the frequency domain (FD), taken from an arbitrary signaling set (PSK, QAM, etc.). The complex input samples $C \in \mathbb{C}$ in a I/Q notation are expressed as $C_n = C_{I_n} + C_{Q_n}$, $n = 0, 1, ..., N-1$, where N is the number of subcarriers or input length of the iDFT. The iDFT (block) output is in the time domain (TD) forming the baseband signal of the data symbols conveyed by N subcarriers, also denoted as the OFDM symbol. Be sure to distinguish between the complex symbols on the

subcarrier level (encoded and modulated data bits) and the OFDM symbol representing the OFDM modulator output (iDFT output). Usually N is a power of two to facilitate the efficient iFFT implementation having a computational complexity of $NlogN$ rather than N^2 as for the iDFT.

To mitigate the effects of multipath propagation over mobile radio channels, OFDM employs the concept of a guard interval insertion in the TD between the OFDM symbols. A simple guard interval (zero value) can perfectly eliminate inter-symbol interference (ISI)[1] due to the scattered signals at the receiver, but the present discontinuities in the time domain would introduce additional inter-carrier interference (ICI) in the frequency domain, hence degrading the performance. Therefore, the concept of guard intervals (GI) with a cyclic prefix (CP) technique is used to achieve periodicity by cyclically extending the OFDM symbol in the guard time. Essentially the last part of the symbol is copied to serve as a prefix. In that way the orthogonality among the subcarrier will be preserved despite the effects of a time-dispersive multipath channel. As a drawback, the guard interval concept forces a SNR loss since redundant data has to be transmitted and increases the separation of the subcarriers (enlarged symbol duration) - slightly increasing the bandwidth.

Incorporating the GI concept to the signal model, the i-th block of the iDFT output $s_i(n)$ can be expressed in a discrete-time notation as

$$s_i(n) = \begin{cases} \frac{1}{N} \sum\limits_{k=-\frac{N}{2}}^{\frac{N}{2}-1} C_i(k) e^{j2\pi nk/N} & , if\ -N_g \leq n \leq N-1 \\ 0 & , else \end{cases} \tag{2.1}$$

Collecting the output blocks in a vector $\mathbf{s}_i = [s_i(0), s_i(1), ..., s_i(N-1)]^{\mathrm{T}}$the cyclic prefix is appended to each block, such that $s_i(n) = s_i(n+N)$ for $-N_g \leq n \leq -1$, where N_g is the amount of samples for the guard interval. A symbol with the added cyclic prefix is then considered as one complete OFDM symbol. The concatenation of all iDFT output blocks (an OFDM frame) forms the sequence

$$s^{(T)} = \sum_i s_i(n - iN_T), \tag{2.2}$$

where $N_T = N+N_g$ is the total symbol or block length and the $^{(T)}$ emphasizes the representation of a sequence. Using the signal description from (2.1) the sequence evaluates to

$$s^{(T)} = \frac{1}{N} \sum_{i=-\infty}^{+\infty} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} C_i(k) e^{j2\pi(n-iN_T)k/N}, \tag{2.3}$$

referring to the i-th symbol (block) and the k-th subcarrier.

A continuous-time baseband expression can be found after performing discrete-time to continuous-time conversion (DC). The discrete block lengths then translate to $N \Rightarrow T_u$ (the useful symbol

1 ISI is also denoted as inter-block interference (IBI) in OFDM systems since portions of adjacent blocks of data symbols could distort the time domain signal due to destructive interference.

time), $N_g \Rightarrow T_g$ (the guard interval time duration) and $N_T \Rightarrow T$ (the total symbol duration). The signal model can now be stated in a continuous-time notation

$$s(t) = \frac{1}{T_u} \sum_{i=-\infty}^{+\infty} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} C_i(k) e^{j2\pi(t-iT)f_k} f(t-iT), \tag{2.4}$$

where $f(t)$ is an assumed rectangular pulse form defined as

$$f(t) = \begin{cases} 1 & , if\, 0 \leq t < T \\ 0 & , else \end{cases}$$

and $f_k = \frac{k}{T_u}$ is the frequency of the k-th subcarrier normalized to the useful symbol duration. The frequency separation of the subcarriers (subcarrier spacing or intercarrier spacing) is denoted as $\Delta f = f_k - f_{k-1} = \frac{1}{T_u}$. The complex baseband signal is then I/Q-modulated and up-converted to be sent over the radio channel.

2.1.3 OFDM Receiver

To follow the propagation over the physical channel a simple time-dispersive multipath channel model is introduced, refer to 3.3.2 for a more elaborate discussion. The channel is characterized by its time-varying (TV) impulse response (IR) denoted as $h(\tau;t)$ describing the channel behavior at time instant t due to an impulse stimulating the channel at time $t-\tau$ (τ is considered as the altering variable). The IR is expressed as

$$h(\tau;t) = \sum_{l=0}^{L-1} \alpha_l(t) e^{-\theta_l(t)} \cdot \delta(t-\tau_l(t)) = \sum_l h_l(t)\delta(t-\tau_l(t)),$$

where $\alpha_l(t)$ is the time-variant amplitude, $\theta_l(t)$ the time-variant phase and $\tau_l(t)$ the time-variant delay of the l-th propagation path, assuming a total of L paths.

The received signal at the ideal receiver is then written as the convolution of the TV-IR due to multipath fading and contaminated by thermal noise:

$$r(t) = h(\tau;t) * s(t) + w_n(t) = \int_0^{\tau_{max}} h(\tau;t) \cdot s(t-\tau) d\tau + w_n(t), \tag{2.5}$$

where $w(t)$ is a sample function of additive white Gaussian noise (AWGN) with variance $\sigma^2_{w_n}$ and τ_{max} is the maximum excess delay of the IR. If the channel is considered quasi-static over the duration of one OFDM symbol the IR can be simplified to $h(\tau) = \sum_l h_l \delta(t-\tau_l)$ which leads to

$$r(t) = \sum_l h_l s(t-\tau_l) + w_n(t). \tag{2.6}$$

If the signal is then AD-converted using the sampling frequency $f_s = \frac{1}{T_s}$ with the sampling period $T_s = \frac{T_u}{N}$, the discrete-time representation in the receiver (ADC output sequence) is

$$r^{(T)}(n) = \sum_{l=0}^{L-1} h(l) \cdot s^{(T)}(n-l) + w(n). \tag{2.7}$$

Here the sampled description for the IR is used - $h(l)$ is an element of the T_s-spaced samples of the overall IR denoted as the vector $\mathbf{h} = [h(0), h(1), ..., h(L-1)]^T$ assuming that the condition $L < N_g$ holds (the delay spread of the channel is shorter than the guard interval, hence no ISI/IBI). Rewriting (2.7) as a concatenation of adjacent blocks of length N_T gives

$$r^{(T)}(n) = \sum_{i} \sum_{l=0}^{L-1} h(l) \cdot s_i(n-l-iN_T) + w(n). \tag{2.8}$$

The ideal receiver then omits the guard interval where the N samples can be written as a vector $\mathbf{r}_i = [r_i(0), r_i(1), ..., r_i(N-1)]^T$. The i-th block is represented as

$$r_i(n) = \sum_{l=0}^{L-1} h(l) \cdot s_i(n-l) + w_i(n), \tag{2.9}$$

for $0 \leq n \leq N-1$. Each block of N samples is forwarded to the N-point DFT (preferably FFT) for OFDM demodulation. OFDM demodulation is accomplished by a bank of filters "matched" to the effective part T_u or N of the symbol. Applying the Fourier transformation to every block leads to

$$R_i(k) = \sum_{n=0}^{N-1} r_i(n) \cdot e^{-j2\pi nk/N}. \tag{2.10}$$

After a few additional mathematical derivations (see Chapter A in the Appendix) the final FD description at the receiver (DFT/FFT output) is given as

$$R_i(k) = H(k)C_i(k) + W_i(k), \tag{2.11}$$

where $C_i(k)$ are the recovered data symbols of the i-th block at the k-th subcarrier, $H(k)$ is the channel transfer function (CTF) over the k-th carrier and $W_i(k)$ represents noise contribution with power σ_w^2. Eqn. (2.11) shows a *perfectly synchronized* OFDM system which can be interpreted as *"a set of parallel transmissions over N Gaussian channels with different complex-valued attenuations H(k)"* [10]. Using coherent modulation techniques at subcarrier level demands channel estimation in order to retrieve the data from the signal constellation, which is summarized as "subcarrier recovery" speaking in OFDM terms (amplitude and phase reference is required for correct symbol detection). Non-coherent schemes on the other side can completely eliminate the need for subcarrier recovery by using differential detection based on adjacent symbol constellation points[13].

More information regarding OFDM systems and their characteristics can be found in the literature [12], [7], [14], [15], [11] and [16].

2.2 The Synchronization Problem

The previously introduced signal model refers to a perfect synchronized OFDM receiver where no timing and frequency error are present at the receiver. The basics of OFDM is simple in principle but the system design is far from being trivial. Moreover, *"synchronization represents one of the most challenging issues and plays a major role in the physical layer design"* as stated by Morelli et. al. [10]. In general, synchronization in a robust OFDM receiver is mandatory for detecting the unknown start of the OFDM frame (timing error) and to align the local oscillator frequencies from the modulator and demodulator (frequency error) [17, 18]. The task at hand is to estimate the errors and correct them in order to maintain a reasonable system performance. Should any of these errors remain uncorrected the orthogonality of the subcarriers is lost, hence, introducing IBI and ICI.

2.2.1 Synchronization Tasks

In practical systems Doppler shifts and instabilities of the local oscillators cause a carrier frequency offset (CFO) f_d between the modulating and demodulating sinusoids. Usually this offset is normalized to the subcarrier spacing leading to $\theta = N f_d T_s$ when using the definition for the SC spacing $\Delta f = \frac{1}{NT_s}$.

Timing errors in OFDM systems are introduced by the unknown arrival time of the OFDM blocks at the receiver, therefore the FFT demodulation window could be placed at the wrong position (which is equal to shifting the integration interval of a matched filter bank). A timing error, denoted as τ_d is commonly normalized to one sampling period T_s, where the normalized timing error is given by $\varepsilon = \frac{\tau_d}{T_s}$.

Incorporating both impairments to Eqn. (2.8) a more realistic (non-ideal, unsynchronized) system model can be developed

$$r^{(T)}(n) = e^{j2\pi\theta n/N} \sum_i \sum_{l=0}^{L-1} h(l) \cdot s_i(n - \varepsilon - l - iN_T) + w(n). \tag{2.12}$$

In order to successfully synchronize the system, the two unknown parameters θ and ε have to be estimated from the incoming sequence $r^{(T)}(n)$, denoting the estimates as $\hat{\theta}$ and $\hat{\varepsilon}$. Frequency synchronization means to use the frequency error estimate $\hat{\theta}$ to counteract the introduced frequency shift by counter-rotating the TD sequence with an angular phase factor $e^{-j2\pi\hat{\theta}n/N}$. Timing synchronization on the other hand uses the estimate $\hat{\varepsilon}$ to correctly realign the FFT window in the receiver. The frequency corrected samples should then lie within the range of $iN_T + \hat{\varepsilon} \leq n \leq iN_T + \hat{\varepsilon} + N - 1$ for OFDM demodulation.

2.2.2 Timing Errors

The DFT window, responsible for demodulating received discrete-time, should only contain samples from one OFDM symbol. Otherwise inter-block interference (IBI) occurs since energy from samples of adjacent blocks will be collected, essentially destroying the orthogonality among consecutive OFDM symbols. The guard interval between the symbols provides intrinsic

protection against IBI, hence, a certain range of this interval is not affected by the previous block. The unaffected length of the interval reduces[2] according to the present channel impulse response (CIR). If the timing estimate lies anywhere within that range no IBI will occur. That is, if the newly defined timing error $\Delta\varepsilon = \varepsilon - \hat{\varepsilon}$ is sufficiently small and belongs to the interval $-N_g + L - 1 \leq \Delta\varepsilon \leq 0$. Assuming a perfectly frequency synchronized system the DFT output will result only in a cyclic shift of the OFDM block. Because of the cyclic shift property[3] of the DFT a cyclic shift in time-domain will introduce only a linear phase factor in the frequency domain. A mathematical model based on Eqn. (2.11) looks like the following:

$$R_i(k) = e^{j2\pi k\Delta\varepsilon/N} H(k)C_i(k) + W_i(k) \tag{2.13}$$

The linear phase factor leads to a progressive phase rotation of the signal constellation in the frequency domain. Notice that a error of $\Delta\varepsilon = 1$sample introduces a phase shift of $\pm\pi$ to the outermost subcarriers $k = \pm\frac{N}{2}$.

If the timing error is outside of the interval stated above unwanted samples from adjacent blocks will contribute to the DFT output leading to IBI and followed by ICI:

$$R_i(k) = e^{j2\pi k\Delta\varepsilon/N} \alpha(\Delta\varepsilon)H(k)C_i(k) + I_i(k, \Delta\varepsilon) + W_i(k), \tag{2.14}$$

where $\alpha(\Delta\varepsilon)$ is an attenuation factor depending on the timing error and $I_i(k, \Delta\varepsilon)$ accounts for IBI/ICI modeled by a zero-mean random variable having a power of $\sigma_I^2(\Delta\varepsilon)$ [10].

2.2.3 Frequency Errors

As a result of an oscillator mismatch or due to Doppler shifts a frequency error the received signal experiences a shift in the frequency domain causing intercarrier interference. ICI arises from the adjacent subcarriers after demodulation, causing a loss of orthogonality. Additionally, the observed subcarrier amplitude is decreased due to a wrong sampling position of the frequency instant [7]. Assuming a synchronized system having no timing errors $\Delta\varepsilon = 0$ the DFT output of the frequency shifted signal can be modeled as

$$R_i(k) = e^{j\varphi_i} \sum_{q=0}^{N-1} H(q)C_i(q)f_N(\theta + q - k) + W_i(k), \tag{2.15}$$

where $\varphi_i = \frac{2\pi i\theta N_T}{N}$ and $f_N = \frac{sin(\pi x)}{Nsin\left(\frac{\pi x}{N}\right)} e^{j\pi x\frac{(N-1)}{N}}$ according to [10].

It is common in the literature to distinguish the frequency offset in an integer multiple of the subcarrier spacing and a fractional part. For an integer frequency offset the DFT output is shifted by an integer amount of subcarriers thus preserving orthogonality. The demodulated

2 In a well designed system the cyclic prefix interval is greater than the CIR to avoid IBI.

3 The time domain sequence $x[((n-m))_N]$, $0 \leq n \leq N-1$ translates to $e^{-j2\pi km/N}X[k]$, see [19]

symbols will then appear at the wrong positions. In the case of fractional offsets the orthogonality among the subcarriers is lost, leading to severe system performance degradation in terms of ICI.

The results by Morelli et. al. showed that in order to maintain a low SNR degradation after the synchronization, the timing errors should be within a few percent of the block length whereas remaining frequency errors of less than 4%-5% of the subcarrier spacing are tolerable.

2.2.4 Synchronization Algorithms

Today there exist a variety of synchronization algorithms to estimate the timing and frequency errors in the OFDM receiver by either using reference blocks (training symbols prepended to the data) or blind estimation schemes where the implicit redundancy of the CP is used. In systems using a frame-based transmission, where bursts of data are sent over the channel, the data aided methods are preferred since the synchronization process must be completed within a specific time window to recover the data. In continuous (broadcast) communication scenarios blind methods provide the benefit of avoiding the additional overhead despite the required (longer) observation time. The system design of CertLink modems will therefore focus on algorithms were specifically designed OFDM symbols are used to assist the synchronization process.

In general, the tasks during synchronization encompass the following:

- Detect the presence of a newly arrived OFDM frame (frame detection)
- Estimate the timing offset to align the DFT window while providing robustness against frequency offsets (unknown at this stage)
- Estimate the frequency offset to correct for the introduced frequency shift (fractional and integer offsets)

There are several approaches to accomplish those tasks either jointly or separately. The presented methods here provide a selection of algorithms which are considered to be feasible for a practical implementation in hardware. In recent years OFDM synchronization has been a focus of ongoing research and is still extensively studied. Morelli et. al. [10] tried to provide a comprehensive survey of the topic since available literature can only be found in a scattered form of journal and conference publications, making it very hard to get a unified picture of research results.

Schmidl and Cox [20, 21, 22] originally employed the concept of using reference blocks having a repetitive structure for estimating timing and frequency errors. The idea was to use robust correlation mechanisms over two specifically designed symbols to perform rapid acquisition of the symbol timing and frequency offsets. These symbols are put in front of the actual data symbols, thus adding some overhead to the OFDM frame. The first training symbol is designed to have to identical halves in the time domain for applying the symbol timing estimation algorithm and finding the fractional frequency offset. To construct such a identical symbol, a pseudo-noise (PN) sequence is modulated over the even frequencies wheres zeros are transmitted at odd frequency indices. The second symbol uses another PN-sequence at odd frequencies to perform channel estimation and another PN-sequence at the even frequencies to help estimating the integer frequency offset. The two halves of the first symbol remain unchanged after passing through the channel apart from the phase shift due to the CFO. Schmidl and Cox (S&C) propose an $N/2$-lag sliding window correlator to perform timing acquisition:

$$P(d) = \sum_{m=0}^{L-1} (r^*_{d+m} r_{d+m+L}), \tag{2.16}$$

assuming L complex samples in one half of the symbol. An iterative implementation is given as

$$P(d+1) = P(d) + (r^*_{d+L} r_{d+2L}) - (r^*_d r_{d+L}). \tag{2.17}$$

The parameter d is the time index for the sliding window length of $2L = N$. In principle, if the sliding window is perfectly aligned with the first training symbol the correlator exhibits a peak. The timing estimate is based on a maximum search over a normalized timing metric M defined as

$$M(d) = \frac{|P(d)|^2}{(R(d))^2}, \tag{2.18}$$

where $R(d)$ is the second half-symbol energy, calculated by

$$R(d) = \sum_{m=0}^{L-1} |r_{d+m+L}|, \tag{2.19}$$

which can also implemented iteratively by using

$$R(d+1) = R(d) + |r_{d+2L}|^2 - |r_{d+L}|^2. \tag{2.20}$$

The actual timing estimate is found by computing $\hat{\varepsilon} = \underset{\tilde{\varepsilon}}{argmax}\{M(\tilde{\varepsilon})\}$, either using a global maximization method or a proposed 90% averaging method where the two 90% values left and right to the global maximum are averaged to find the timing estimate $\hat{\varepsilon}$. Considering the inherent timing metric plateau of the S&C method the 90% averaging method shows a better performance [21]. In order to detect the arriving OFDM frame the metric is continuously monitored and compared to a threshold value which has to be appropriately designed with regard to false alarm and misdetection probabilities.

After passing through the channel the two first symbol halves experience a phase shift[4] of approximately $\phi = \pi T \theta$. This phase different can be directly estimated using

$$\hat{\phi} = angle(P(d)), \tag{2.21}$$

where d is usually substituted by the best timing point. If this phase estimate is withing $\pm\pi$ (corresponding to an interval of ± 1 subcarrier spacings) the fractional frequency estimate can be directly calculated by

4 This method was originally applied by Moose [23] which experienced only a very limited frequency acquisition range. S&C improved the estimator as presented here.

$$\hat{\theta}_f = \frac{\hat{\phi}}{\pi T}. \tag{2.22}$$

Otherwise the CFO estimate is distinguished between a fractional part denoted as θ_f and an integer part θ_i. Using the second symbol of the S&C method the additional integer part can be resolved and the overall estimate is evaluated by

$$\theta = \frac{\hat{\phi}}{\pi T} + \frac{2\hat{g}}{T}. \tag{2.23}$$

The parameter $\hat{g}$ resolves the ambiguities of the integer part which maximizes an additional metric $B(g)$. The metric uses the partly frequency corrected samples from the second symbol and translates them to the frequency domain to yield

$$B(g) = \frac{|\sum_{k\in X} x^*_{1,k+2g} v^*_k x_{2,k+2g}|^2}{2(\sum_{k\in X} |x_{2,k}|^2)^2}. \tag{2.24}$$

Here X defines the set of all even frequencies and x_1, x_2 represent the first and the second symbol in the FD domain, respectively. The additional term v_k is a differentially modulated sequence on the even frequencies of the second symbol. The range of assessable frequency offset is therefore extended to $\pm N$ subcarrier spacings. In order to reduce the overhead for estimating frequency errors Morelli et. al. [24] proposed a new training sequence of only one symbol containing *more* than two repetitive parts. The improved estimator has a large estimation range of $\pm L/2$ SC-spacings and shows almost equal performance as the S&C method.

For more information on the implementation details refer to Chapter 4, specifically Section 4.4.1.1 and Chapter 5. Details of the theoretical performance are available in [20],[21] and [22].

The method after S&C shows an inherent metric plateau which leads to some uncertainties regarding the start of the frame, related to the present estimation variance. Minn et. al. [25] tried to alleviate these shortcomings by proposing two improved approaches for the timing estimator. One of them tries to incorporate all samples of one symbol to calculate the symbol energy for normalizing the metric. Additionally the range of averaging is extended to a window $N_g + 1$ samples rather than using the 90% averaging method. The new timing metric is defined as

$$M(d) = \frac{1}{N_g + 1} \sum_{k=-N_g}^{0} M_f(d + k). \tag{2.25}$$

$M_f(d)$ is defined as

$$M_f(d) = \frac{|P(d)|^2}{R_f^2(d)}, \tag{2.26}$$

having

$$R_f(d) = \frac{1}{2}\sum_{m=0}^{N-1}|r(d+m)|^2. \tag{2.27}$$

The autocorrelation $P(d)$ is the same as in Eqn. (2.16).

The other approach incorporates a different training sequence structure to further optimize the sharpness of the timing metric reducing the estimation variance even more. Refer to [25] for details and results and the implementational aspects in Section 4.4.1.2.

In order to completely avoid the ambiguity in the S&C method and improve the results form [25] Park et. al. [26] presented a novel timing offset estimation for OFDM systems. A new conjugate symmetric training sequence is proposed resulting in an impulse-shaped metric by enlarging the difference between two adjacent samples of the timing metric. The metric is designed as

$$M(d) = \frac{|P(d)|^2}{(R(d))^2}, \tag{2.28}$$

where now the correlations are calculated as follows

$$P(d) = \sum_{k=0}^{N/2} r(d-k)\cdot r(d+k) \tag{2.29}$$

$$R(d) = \sum_{k=0}^{N/2} |r(d+k)|^2. \tag{2.30}$$

The value d which maximizes $M(d)$ yields the correct symbol timing improving the accuracy (smaller timing MSE) from the previous methods. Since the training symbol structure is similar to S&C the same frequency estimation method can be applied. The research results form Park et. al. can be found in [26] whereas simulation results are also presented in Section 4.4.1.3.

2.3 A Mobile Radio Channel

This section reviews some of the major characteristics and mathematical definitions of mobile radio channels. Channel modeling is a major research topic in mobile communication systems and is considered as the foundation and a prerequisite for the system design. Based on the works of Prasad, Hara and Harada [7], [12], [11] the key features of such channels will be discussed in the subsequent sections. The results presented here are not intended to provide a detailed presentation of the topic, rather a comprehensive overview to understand the link between mobile radio channels and orthogonal frequency division multiplex systems.

Considering a communication system consisting of a fixed base station and a mobile receiver, the propagation path of the radio waves is characterized by different impairments. Imagine the case where the base station transmitter transmits the signal in every direction using an omnidirectional antenna. The mobile receiver on the other side receives not only the direct, line of sight wave but also various different reflected, scattered and diffracted waves which are all superimposed at the receiver's antenna. As the path length of every wave changes, the time of arrival and the phase of each wave changes accordingly. The superposition of such delayed waves characterize a multipath propagation environment where the received signal fluctuates. A fluctuation in the signal level or power, where the signal is strengthened or weakened depending on the observation time is then called multipath fading [11].

A simple channel model shows how the multipath effects affect the received signals. Assuming a transmitted signal $x(t) = Re\{s(t)e^{j2\pi f_c t}\}$ with $s(t)$ as the equivalent baseband description and modulated with the carrier frequency f_c the received signal is modeled as

$$y(t) = \sum_{l=1}^{L} \alpha_l(t) x(t - \tau_l(t)), \tag{2.31}$$

where $\alpha_l(t)$ is the complex valued channel loss and $\tau_l(t)$ is the real-valued time delay for the l-th transmission path whereas L denotes the total number of paths seen by the receiver. The parameter $\alpha_l(t)$ and $\tau_l(t)$ are usually modeled as random stochastic processes. The equivalent baseband expression for the received signal is then given as

$$\begin{aligned} r(t) &= \sum_{l=1}^{L} \alpha_l(t) e^{-j\theta_l(t)} s(t - \tau_l(t)) \\ &= \int_{-\infty}^{+\infty} h(\tau; t) s(t - \tau) \mathrm{d}\tau. \end{aligned} \tag{2.32}$$

Here $\theta_l(t) = 2\pi f_c \tau_l(t)$ denotes the time varying phase of the l-th path. Signal fading is observed due the unpredictable changes of the time varying phases which are mainly dependent on the excess delay time $\tau_l(t)$ of the corresponding path. $r(t)$ can also modeled as the convolution of the time varying impulse response $h(\tau; t)$ with the transmitted signal $s(t)$. This IR characterizes the channel behavior at the time instant t due to a stimulating impulse at $t - \tau$ and evaluates to

$$h(\tau;t) = \sum_{l=1}^{L} \alpha_l(t) e^{-\theta_l(t)} \cdot \delta(t - \tau_l(t)). \tag{2.33}$$

Since $r(t)$ is expressed as the sum of many complex-valued stochastic processes (large number of random variables) the central limit theorem (CLT) applies where $r(t)$ can be modeled as a complex-valued Gaussian stochastic process. Such a process can be solely described by its mean and autocorrelation functions. The derivations show that the fluctuations of the received signal envelope follows a so called Rayleigh distribution wheres the phases show a uniform distribution. If $r(t)$ is described as a zero-mean Gaussian process (the receiver sees a line of sight path) then the probability density function reveals a Ricean distribution. A Ricean distribution is characterized by the Ricean K-factor which shows the relation between the power of the dominant ray path and the scattered paths.

2.3.1 Multipath Delay Profile

If the time varying impulse response (IR) is assumed to be wide sense stationary experiencing uncorrelated scattering (WSSUS), the multipath delay profile $\phi_h(\tau)$ can be derived from the autocorrelation of the IR. The multipath delay profile, multipath intensity profile or also called the delay power spectrum (DPS), describes *"the average output power of the channel as a function of time delay τ"[12]* and characterizes the frequency selectivity (or time diversity) of the channel.

2.3.2 Frequency-Selective Fading Channels

Translating the TV-IR to the frequency domain yields the time variant channel transfer function $H(f;t)$ due to the time variant delays $\tau_l(t)$. Performing the autocorrelation of the CTF (assuming WSSUS) leads to the "spaced frequency spaced time correlation function" $\phi_H(\Delta f;\Delta t)$, valuable to characterize the channel for all times and for all frequencies. Focusing on the frequency variations leads to the "spaced-frequency correlation function" $\phi_H(\Delta f)$ which describes the correlation between changes in frequency having a frequency separation of Δf . Based on the spaced frequency correlation function the "coherence bandwidth" $(\Delta f)_c$ is defined as the bandwidth over which the the channel variations $\{\alpha_l(t), \phi_l(t)\}$ are highly correlated. All frequencies within this bandwidth are attenuated in the same way therefore specifying a range over which the transfer function shows significant correlation[12]. The coherence bandwidth is typically used as a measure for frequency selectivity (or the degree of time dispersion). If the signal bandwidth is larger than the coherence bandwidth the channel is said to be "frequency selective" or "time dispersive". Otherwise the channel is considered to be "frequency non-selective" or "flat". In this context the root-mean-square (RMS) delay spread provides a similar measure.

2.3.3 Time-Selective Fading Channels

If only the time variations are of interest, then the "spaced-time correlation" $\phi_H(\Delta t)$ describes "*the correlation between time variations of the channel separated by Δt"[12].* The Fourier

transform of this correlation leads to the Doppler power spectrum $D_H(\varsigma)$ which in this case is used as a measure of frequency dispersion or time selectivity. Similar to the coherence bandwidth, a "coherence time" $(\Delta t)_c$can be defined. The coherence time characterizes the time duration where the channel variations $\{\alpha_l(t), \phi_l(t)\}$ are highly correlated. If the symbol time is larger than the coherence time then the channel is considered to be "time-selective" or "fast". Otherwise "time-non-selective" or "slow".

If the mobile receiver is traveling at a speed of v and the incoming waves arrive at different angles θ, a shift in frequency will be introduced due to the Doppler effect. This shift in frequency, the Doppler shift is defined as

$$\varsigma = \frac{v}{\lambda} cos\theta = f_D cos\theta \tag{2.34}$$

where $\lambda$is the wavelength of the carrier frequency f_c and f_D is denoted the maximum Doppler frequency at an arriving angle of $\theta = 0$ (towards the moving receiver). In a frequency domain description (Doppler power spectrum) the Doppler effect leads to a broadening of the signal spectrum[12].

For a more complete introduction to mobile radio channels refer to the literature in [7], [15], [11] or [12].

3 System Architecture

Now that the essential theoretical basics have been established, details of the system's architecture can be discussed more thoroughly. The purpose of this chapter is to provide an introduction to the TTP basics in order to understand how a wireless extension of the datalink can be designed. Based upon the fundamentals of the TTP concept the actual architecture of the CertLink modem will be explained. After a short review of two existing aeronautical datalinks the system requirements can be derived along with a description of the system components. The design choices mentioned in this context are related to a more detailed scientific research development process and therefore can not be reiterated to the fullest extent here.

3.1 TTP Basics for CertLink

The Time Triggered Protocol (TTP) is a fieldbus system developed for a reliable, safety-critical and fault-tolerant communication between distributed systems which fulfills hard real-time criteria and is well established in today's aerospace and automotive industry. TTTech provides a full range of different system design tools and equipment in order to rapidly integrate TTP as the solution of choice. Ranging from ASIC and IP solution of a TTP Controller other design tools for hardware and software development are available. This section only strives essential concepts of TTP with a focus on the PHY layer properties in conjunction with the development of CertLink modem, employing a certifiable datalink for a Wireless-TTP interface.

3.1.1 TTP Architecture

A TTP network typically consists of 4 to 64 bus members, denoted as *nodes*, which communicate over a common bus structure having a datarate of 5 Mbit/s when using a Modified Frequency Modulation (MFM) coding scheme. The data is transmitted over two distinct parallel bus systems, or channels[1] where the specification supports different physical layers, one of them being the RS-485 standard. It is a distinctive feature of a TTP system to identify the entire timing characteristics of the network communication at the start time of the system development. This timing scheme is stored in the so called Message Descriptor List (MEDL).

Establishing a time division multiple access (TDMA) communication scheme the entire set of nodes is called a *cluster*. Each bus member of that cluster has a defined time window (slot) assigned by the MEDL, in which he can transmit data packets, frames (messages) up to 240 Bytes in size. Notice that the two parallel sub-buses can possibly transmit different messages during the same time slot. The collection of all time slots is then called a TDMA *round* where

1 Usually denoted as Channel A and Channel B.

multiple rounds are summarized as a *cluster-cycle*. Such a hierarchical approach allows for different update rates of the messages, essentially determining the message periodicity. The assignment of the time windows can change from round to round.

3.1.1.1 TTP Timing

A node will experience different phases during the data transmission in the assigned time window. Previous to the actual transmission of the data, the Transmission Phase (TP) defines a Pre-Send Phase (PSP) where the TTP-Controller prepares the data to send and acquires the bus access. After the TP a Post-Reception Phase (PRP) specifies the time in which the receiving controller can process the incoming data. The bus is held in an idle state (IDLE Phase) after the PRP until the beginning of a new PSP. To complete the description of the TTP timing an Inter-Frame Gap (IFG) is introduced which lasts from the end of one frame to the beginning of the following frame.

3.1.1.2 Synchronization

As a time triggered protocol, TTP defines a global system clock which is used for synchronizing every node of the bus system. This global time represents an approximation to the average time of the members where the averaging requires at least four nodes in order to establish a fault-tolerant communication. The smallest time unit is defined as a Macrotick (MT) whereas the MT is derived from the actual processor clock (denoted as the Microtick). Using a sophisticated averaging algorithm, the macroticks of each node is held within a time span, called "*Precision*" π. To do so the TTP controller collects the variance of the received message time stamps and uses the median of these values to correct for the clock error.

3.1.1.3 TTP Frame Structure

A TTP frame is composed of a 4 bit header, 2-240 Bytes of payload data (messages) and a 3 Byte CRC. The header bits are placed prior to the message and is used for rudimentary control mechanisms. In a start-up scenario of a cluster, the protocol configures distinct nodes as "*cold starters*". To perform some sort of initialization and enabling the integration of nodes to the cluster they start sending out so called initialization frames (I-frames) where they contain only C-State data as payload. Otherwise, the normal frames (N-frames) contain user data and a special CRC sum. Additionally, an extended frame (X-frame) combines the features of I- and N-frames to provide a possibility for node re-integration.

The first header bit decides on the type of frame, whereas the other three bits cover the state change of a controller.

3.1.2 Wireless TTP

The setup for a Wireless TTP incorporates the distributed wire-bound bus systems and the CertLink modems which extend the TTP system over-the-air (OTA). Fig. 3.1 illustrates the design setup as a block diagram. As an example the moving UAV employs a TTP bus system

which could be required for the power-management or the rudder control. A fixed ground control station on the other side also uses a TTP bus structure, possibly used for tasks in flight management, flight control or in mission control scenarios.

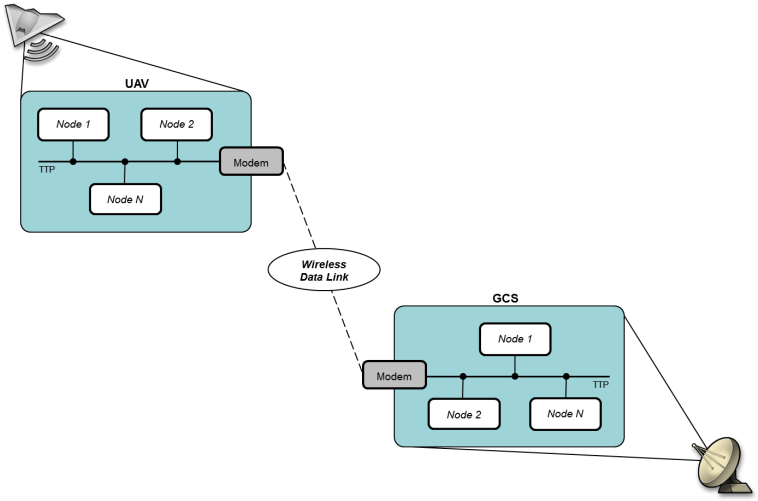


Figure 3.1: System architecture for a Wireless TTP extension in an example distributed wire-bound TTP system of an UAV and the ground control station

Using the MFM coding scheme for the TTP wire-bound PHY layer (RS-485) defined voltage level sequence lengths (depending on the timing of the encoded symbols) can be identified. Without using a TTP controller these voltage levels have to be correctly acquired in order to provide the data (MFM encoded symbols) to the modem's interface. Details on the data acquisition are out of scope of this thesis and are further investigated during the ongoing research by the project team.

Other research areas are concerned about the datalink specification of the various parameters to incorporate the new PHY layer to TTP. Issues concerning the different processing times in the receiver or the transmitter influence the deterministic behavior of the protocol and have to be carefully selected to ensure the proper operation of the datalink over the air. As mentioned in Chapter 1 the CertLink project team and TTTech cooperate on a certifiable link specification and work on several topics to propel the development process. Detailed design decisions on the timing will not be presented here since they are also part of ongoing research.

3.2 System Requirements

This section will present some of the numerous system requirements that are essential for the understanding of the overall modem concept. Since the final project outcome aims at a certifiable datalink the entire specification reclines on existing links to ease the certification process.

3.2.1 VDL Mode 2

The Very High Frequency (VHF) Digital Link (VDL) Mode 2 is an air interface for digital data communication between aircraft and ground-based systems. The European Telecommunication Standards Institute (ETSI) approved standard in the final release ETSI EN 301 841-1 (physical layer specification) operates in the VHF band between 117,975 MHz and 137,000 MHz with a 25 kHz channel spacing [27]. The VDL mode 2 datalink employs a differential modulation scheme (D8PSK) and is designed to be a subsystem (ground/air) of the Aeronautical Telecommunication Network (ATN). Based on the OSI layer model the VDL system defines the three lower layers consisting of the Physical Layer (Layer1), the Link Layer (Layer 2) and the Network Layer (Layer 3). ETSI [27] defines the lowest physical layer as follows:

"Layer 1 (Physical layer): provides transceiver frequency control, bit exchanges over the radio media, and notification functions. These functions are often known as radio and modulation functions. The physical layer handles information exchanges at the lowest level and manipulates bits. The physical layer handles modulation, data encoding and includes a forward error correction mechanism based on interleaving and Reed Solomon coding."

The other two layers will not be discussed here since they do not contribute to the actual topic of interest: the PHY layer of the datalink.

Basically the physical layer incorporates several tasks (cited from [27]):

- *"to modulate and demodulate radio carriers with a bit stream of a defined instantaneous rate to create an RF link;"*
- *"to acquire and maintain bit and burst synchronization between Transmitters and Receivers;"*
- *"to transmit or receive a defined number of bits at a requested time (packet mode) and on a particular carrier frequency;"*
- *"to add and remove a training sequence;"*
- *"to encode and decode the Forward Error Correction scheme;"*
- *"to measure received signal strength;"*
- *"to decide whether a channel is idle or busy, for the purposes of managing channel access attempts;"*
- *"to offer a notification service about the quality of link."*

After the receiver decodes the incoming data within the physical layer, it then propagates them for further processing in the upper layers. On the transmitter side the same principle applies,

where the data is received from the Layer 2 and is then properly processed in order to transmit the data bits over the radio frequency link.

As the system uses a D8PSK scheme (raised cosine filter, $\alpha = 0.6$) the data is differentially modulated where the three bits are transmitted as relative phase changes rather than absolute values of phase and amplitude. VDL Mode 2 requires the datastream to be divided into groups (blocks) of three consecutive data bits (LSB first). Zero padding shall be applied if necessary.

To provide means for synchronization and power ramp up procedures a training sequence is prepended to the data packet composed of several special function symbols (cited from [27]):

- *"Transmitter ramp up and power stabilization (5 symbols);"*
- *"synchronization and ambiguity resolution (16 symbols - the "unique word");"*
- *"reserved symbol (1 symbol);"*
- *"transmission length (a single 17 bit word);"*
- *"header FEC (5 bits)."*

In summary 88 bits of overhead for each data packet is required if the D8PSK scheme is used. Focusing more on the receiver architecture the following table will present a selection of the proposed specification. Note that all receiver requirements are based on an applied signal level of -87 dBm:

Uncorrected BER requirement (without FER)	$<10^{-3}$
Symbol rate capture range	±50ppm offset
Frequency capture range	±826Hz
Desired signal dynamic range	BER $<10^{-3}$if signal level is increased to -7 dBm
Frequency tolerance	±2 ppm of the carrier frequency
Modulation scheme	D8PSK (raised cosine filter, $\alpha = 0.6$)
Symbol rate (SR)	10 500 symbols/s, ±50ppm
Nominal bit rate	31 500 bit/s

Table 3.1: Selection of the VDL Mode 2 receiver requirements

Remark: The frequency capture range is composed of ±685Hz (maximum frequency error at 136,975 MHz) and ±141Hz as the maximum Doppler shift.

The most important requirement of the transmitter is the symbol rate which shall be 10 500 symbols/s ±50ppm for Mode 2. The symbol rate results in a nominal bit rate 31 500 bits/s.

The VDL Mode 4 standard extends the capabilities of the system to air/air communication, but uses a different (coherent) modulation scheme. For more information see [28].

3.2.2 L-DACS1

In order to provide another datalink example the future L-Band Digital Aeronautical Communications System Type 1 (L-DACS1) is shortly introduced in this section.

The Type 1 version or first option for implementing the system is a time division duplex (TDD) configuration, based on a OFDM modulation scheme and supporting gray-mapped QPSK, 16QAM and 64QAM. As the heavily used VHF band becomes unfeasible for Future Communications Systems (FCS) demanding higher data rates for an advanced Air Traffic Management (ATM), the Federal Aviation Administration (FAA) and EUROCONTROL initiated a study for a newly designed communication system in the lower part of the L-band (906-1164 MHz). Since other systems, e.g. Distance Measuring Equipment (DME), currently use the L-band, L-DACS1 is intended to work without any interference to existing, already operating systems [29].

The physical layer design utilizes an OFDM FDD system for an optimal spectral efficiency and to maximize the channel capacity. The system differentiates between a Forward Link (FL) and a Reverse Link (RL) for simultaneous transmission. The FL enables a continuous transmission while the reverse link is based on OFDM bursts. Without going into much detail the main parameters of the OFDM PHY layer are depicted in Table 3.2 on page 30 (valid for FL and RL).

FFT size: N_{FFT}	64
Sampling time: T_{sa}	1.6 µs
Sub-carrier spacing: Δf	9.765625 kHz
Useful symbol time: T_u	102.4 µs
Cyclic prefix ratio: G	11/64
Cyclic prefix time: T_{cp}	17.6 µs
OFDM symbol time: T_s	120 µs
Guard time: T_g	4.8 µs
Windowing time: T_w	12.8 µs
Number of used sub-carriers: N_u	50

Table 3.2: PHY layer parameters of the L-DACS1 OFDM system, cited from [29]

Using these parameters the total FFT bandwidth evaluates to 625.0 kHz - considering the guard bands the effective bandwidth reduces to 498.05 kHz. The maximum data rate in the FL is 1373.3 kbit/s (64QAM) and 1038.4 kbit/s (64QAM) in the case of the RL. A reference frequency accuracy of ±0.1ppm or better is defined for the ground station (GS) and ±1ppm for the aircraft station (AS). The Doppler shift between the GS and AS is 1.675 kHz at 850 knots and a carrier frequency of 1149.5 kHz.

3.2.2.1 Synchronization

One interesting aspect, regarding the development of a CertLink communication system, is the OFDM synchronization procedures employed in the L-DACS1 system. As for every OFDM system time and frequency estimation methods have to be employed.

L-DACS1 uses synchronization sequences embedded in two consecutive OFDM symbols defined in the frequency direction. The designed frequency mapping delivers four identical halves for the first symbol and two identical halves for the second symbol in the time domain. Using similar methods as described in [21] the time and frequency errors are exploited by utilizing the appropriate metrics as proposed in the paper. The receivers in the ground station as well as the aircraft station perform correlation methods (acquisition and tracking) to compensate for the errors. If coherent modulation schemes are used at the subcarrier level, channel estimation and equalization is mandatory in order to correctly demodulate the data. For this purpose additional pilot symbols are inserted in the frequency domain to provide measures to estimate the channel impulse response.

More information and a detailed description of the L-DACS1 specification is available in [30].

3.2.3 CertLink

As mentioned before the PHY layer of the CertLink modem will adequately adapt the specifications of the VDL Mode 2 physical layer to an OFDM based system.

A specified symbol rate (SR) of 10 500 OFDM symbols/s reveals a symbol duration of $T = \frac{1}{SR} = 95.24\mu s$ for one symbol. Defining this total OFDM symbol duration to be composed of a guard time T_g and the actual useful symbol time T_u ($T = T_{guard} + T_u$) and defining the guard interval (GI) to be 1/8, the guard time evaluates to 11.905 µs. Accordingly the effective symbol time is then $T_u = 83.333\mu s$. Based on this symbol time the frequency spacing can be calculated. Using $\Delta f = \frac{1}{T_u}$ the subcarrier frequency spacing (or subchannel separation) is then 12 kHz. Depending on the FFT length or the amount of subcarriers the required bandwidth can be calculated. Either by using a simply approximation which is calculated as:

$$B = \Delta f \cdot N, \tag{3.1}$$

where here N is the FFT length, or a more exact formula adopted from [12]:

$$B = \frac{2}{T} + \frac{N-1}{T\left(1 - \frac{T_g}{T}\right)}, \tag{3.2}$$

considering the effective bandwidth by incorporating additional terms accounting for the guard interval. A guard interval of 1/8 does not have a severe impact on the change in bandwidth requirements, so both formulas lead to a bandwidth of approximately 24.6 MHz under the assumption that *every* subcarrier is used and a total of 2048 carriers is available. If not all subcarrier are occupied the bandwidth can be presumed to have lower values.

Considering the FFT length again, switching to a digital representation is more intuitive. Therefore a 2048 sample FFT will be employed where additional 256 samples are reserved for the guard interval. Together, 2304 samples form one OFDM symbol.

Adopting the differential eight phase shift keying concept together with a Gray-encoding scheme from VDL Mode 2, three bits form one symbol at subcarrier level. In total 6144 bits or 2048 complex symbols could be theoretically conveyed in each OFDM symbol. Considering

the forward error correction mechanism (Reed-Solomon) and the redundancy in the frequency domain, the effective useful bits per symbol will be much lower.

Based on the TTP specification the total amount of required bits for one TTP frame (embedded in an OFDM frame) can be calculated. CertLink defines 2 TTP channels and 2 additional payload channels to cover a variety of transmission data. All four channels shall be structured the same way to ease the implementation effort. Each data frame can embed a maximum of 240 Bytes. The modem will not implement a dedicated TTP controller since the data should be transmitted transparently (decoding and encoding of the data is unnecessary). Because of the MFM encoding on the bus each bit will be sampled twice to correctly detect the data. To increase the availability of the wireless datalink a spectral redundancy is introduced to the OFDM scheme. Hence, each OFDM symbol carries redundant data (one part on the lower frequencies and the other part on higher frequencies - relative to the respective subcarrier center frequency). Adopting the FEC from the VDL Mode 2 link a Reed-Solomon block encoder provides a means to correct for burst transmission errors and is is implemented as RS(n,k), having n=255 and k=249. Accommodating for all of the previously mentioned specifications the number of uncoded transmission bits can be calculated as

$$n_{uncoded} = n_{channel} \cdot n_{TTP} \cdot n_{sampling} \cdot r \cdot 8, \tag{3.3}$$

where $n_{channel} = 4$ is the number of channels (2 TTP, 2 payload), $n_{TTP} = 240$ is the number of Bytes per frame, $n_{sampling} = 2$ is the sample rate for the bus and $r = 1$ is the redundancy. This leads to 15360 uncoded bits of data that has to be transmitted over the Wireless TTP interface. Incorporating the RS-encoding along with the required padding (block encoding) the input size for the encoder must fulfill the following condition $n_{RS} = y{\cdot}k\lceil log_2(n+1)\rceil$, where y is a integer multiple. This requirement increases the total (encoded) number of bits according to

$$n_{coded} = \lceil \frac{n_{uncoded}}{n_{RS}} \rceil n_{RS} = 16320\, bits, \tag{3.4}$$

where $\lceil . \rceil$ denotes the ceiling-function. Dividing this number by the specified FFT length of 2048, altogether 8 OFDM symbols (n_{OFDM}) are required. To determine the amount of symbols on the subcarrier level the following formula can be used

$$n_{dataSymbols} = \lceil n_{coded}/m/n_{\mathrm{OFDM}} \rceil, \tag{3.5}$$

where $m = 3$ is the modulation index (D8PSK). So each OFDM symbol conveys 680 complex data symbols or 2040 bits. These numbers describe the effective number of symbols (or bits) but with regard to redundancy this amount is doubled to 1360 and 4080, respectively. Setting this into perspective to the available 2048 subcarriers the utilization is around 66.4%. Computing the theoretical data rate R_{data} (number of effective bits per second) with respect to the given SR, evaluates to 20160000 bit/s whereas the transmission rate R (number total bits per second) is 21420000 bit/s. Splitting to each TTP subchannel (and payload) a data rate of 5040000 bit/s would be possible. With the current configuration the bandwidth reduces to 16.3 MHz since the subcarrier utilization is relatively low.

3.2.3.1 Frame Structure

The Header of the VDL Mode 2 specification is modified to contain only the stabilization sequence and the synchronization sequence for the power ramp up procedure. In order to perform OFDM synchronization 2 OFDM symbols shall be used to compensate for the timing error (FFT alignment) and frequency errors (fractional and integer offsets). Following the two synchronization symbols 8 data symbols serve as a container for the TTP channels. Hence, one OFDM frame consists of 10 OFDM symbols with a total time duration of $T_{frame} = 952.4\mu s$.

3.2.3.2 Miscellaneous

The frequency stability on the transmitter and receiver side shall be ±2ppm - leading to a relative frequency offset of 9600 Hz between TX and RX, assuming a carrier frequency of 2.4 GHz. The JXP platform speed is projected to be $25^{m}/_{s}$ with an operational LOS range of 5km. The moving UAV will therefore impose a Doppler shift of 200 Hz (speed of light $c = 3 \cdot 10^{8 m}/_{s}$). As discussed the system will operate in the 2.4 GHz ISM Band with an aspired TTP bus rate of at least 500 kbit/s.

3.2.3.3 Specification Summary

The specification of the CertLink physical layer is summarized in the following table:

Number of subcarriers: N	2048
FFT length: N_{FFT}	2048 samples
Guard interval ratio: GI	1/8
Subcarrier modulation scheme:	D8PSK
Guard interval length: N_g	256 samples
Symbol rate: SR	10 500 OFDM symbols/s
Symbol duration: T	95.24 µs
Useful symbol time: T_u	83.333 µs
Guard interval time: T_g	11.905 µs
Subcarrier spacing: Δf	12 kHz
Bandwidth: B	24.6 MHz
Code rate: CR	255/249
Number of TTP subchannels: $n_{channels}$	4
TTP frame size: n_{TTP}	240 Bytes
Bus sampling rate: $n_{sampling}$	2
Redundancy factor: r	1
Number of uncoded bits per OFDM Symbol: $n_{uncoded}$	15360
Number of coded bits per OFDM Symbol: n_{coded}	16320
Number of OFDM data symbols:	8
Number of synchronization symbols:	2
Subcarrier modulation index: m	3
Subcarrier utilization:	66.4%
Effective bandwidth: B_e	16.3 MHz
Effective data rate: R_{data}	20.2 Mbit
Transmission rate: R	21.4 MBit
Effective data rate per TTP channel: R_{TTP}	5 MBit
Total OFDM frame time duration: T_{frame}	952.4 µs

Table 3.4: Overview of the CertLink PHY specification

Remark: Parameters N_{used} and n already include redundancy, that is only half the carriers are relevant.

3.3 System Components

This section shortly discusses on which hardware platform the actual CertLink modem will be implemented and provides insight to the capabilities of the selected hardware. Furthermore the transmission channel, as being part of the system design, is elaborated in order to define a suitable channel model for system simulation.

3.3.1 Hardware Implementation

The physical layer of an CertLink modem is very similar to a typical OFDM layer which can be divided into an OFDM Transmitter and an OFDM Receiver. Those two modules are merged together to form an OFDM transceiver which sends and receives data over a radio channel. The details of the signal chain are explained in more detail in the following chapter Chapter 4.

One important aspect is concerned about the implementation of the modem since a rapid prototyping approach is envisioned. Instead of a pure Hardware Defined Radio (HDR) implementation a more flexible Software Defined Radio (SDR) architecture will be used which uses programmable devices to process digital baseband data at radio frequency. Since the processing functionality is provided by sophisticated silicon devices such as Digital Signal Processors (DSPs) or FPGAs the time to market is reduced while increasing the flexibility by entirely utilizing software upgrade. The concepts and basic architectures concerning SDRs are discussed in the literature [31]. As CertLink follows this approach, the signal processing power is provided by an Altera FPGA hardware platform. Specifically, an 780-pin Altera Cyclone III EP3C120F780C7N device is used which is mounted on a DSP development board appropriate for wireless applications and other signal processing tasks. The main features of the FPGA device can be summarized as (see [32]):

- *"119K logic elements (LEs)"*
- *"3,888 Kbits of memory"*
- *"288 18 x 18 multiplier blocks"*
- *"Four phase locked loops (PLLs)"*
- *"20 global clock networks"*
- *"531 user I/Os"*
- *"1.2 V core power"*

The development board features an on-board memory of 256 MB dual-channel DDR2 SDRAM, 8 MB SRAM and 64 MB of flash memory. The additional Data Conversion High-Speed Mezzanine Card (HSMC) is intended for high-speed, high-performance DSP applications using AD9254 14-bit 150 MS/s analog to digital converters and 14-bit resolution TI DAC5672 275 MS/s digital to analog converters.

3.3.2 Transmission Channel

To find an appropriate aeronautical channel model for system level simulations, some basic channel parameters have to be defined. This channel is assumed to be time-variant, dominated

by Ricean multipath fading (LOS component) and subject to Doppler effects. Based on the work in [33, 34] the most important parameters for a reasonable channel model can be found.

Assuming the aircraft to fly at a maximum height h_2 =5km, at a distance d =5km and having the antenna at a height h_1 =2m at the ground station the "*geometric path length difference Δl between the line-of-sight path and the dominant reflection along the vertical plane on the horizontal flat ground*" from [33] can be evaluated to:

$$\Delta l = \sqrt{h_1^2 + \left(\frac{h_1 d}{h_1 + h_2}\right)^2} + \sqrt{h_2^2 + \left(d - \frac{h_1 d}{h_1 + h_2}\right)^2} - \sqrt{d^2 + (h_2 - h_1)^2} = 2.83m \tag{3.6}$$

This path length corresponds to a path delay of $\Delta\tau = 9.43ns$. A worst case scenario multipath delay spread of $T_m = 10\Delta\tau$ leads to a coherence bandwidth $B_{cb} = \frac{1}{T_m}$ of 10.6MHz which in turn relates to a surely *frequency non-selective* channel for an OFDM system with a subcarrier spacing of 12 kHz.

Based on the research in [34] the Ricean K-factor is selected to be $K = 10$ (mean value of the considered scenarios evaluated by Haas). The Doppler power spectral density is approximated by a Gaussian power spectral density since the uniform distribution of the angle of arrival of the scattered components (Jakes model) does not hold for practical aeronautical channel models [34].

Hence, the channel model assumes one unscattered Ricean (LOS) path with $K = 10$ along with a scattered Rayleigh path relative to the line of sight component having a relative power of $R_p = -12$dB and a time delay of $T_m = 100ns$. See Fig. 3.2 for a schematic representation of the proposed multipath scenario used for system simulation.

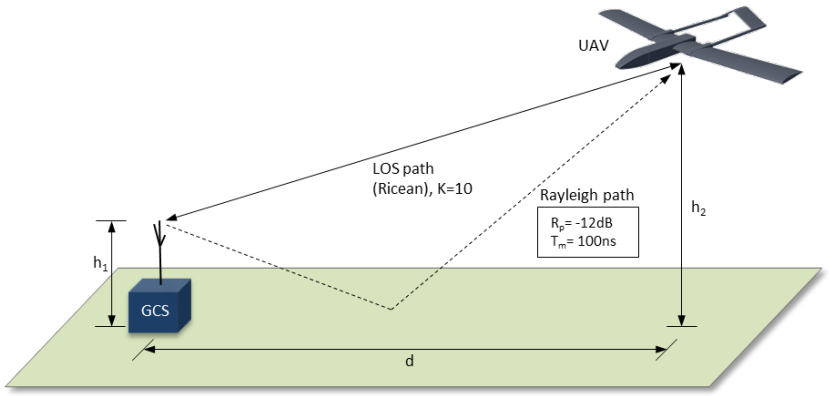


Figure 3.2: Illustration of the proposed multipath propagation channel model

4 System Simulation

To start off a design process for developing and implementing an OFDM based receiver in hardware, it is indispensable to perform some kind of system-level end-to-end simulation. There are two aspects that require the most attention. First and foremost, to fully understand the concepts of a digital communication system employing OFDM and second, to face and evaluate the difficulties of the synchronization process. Both, the fundamentals of Section 2.1 and the basics of synchronization presented in Section 2.2 provide the theory of operation. It is desirable to prove the theory by means of an executable specification in form of a simulation model. Having a fully functional system or simulation model, one is then able to see whether the requirements can be met by using adequate performance measures. There are several different ways to model a communication system. MATLAB has proven to be a valuable language to perform an abstract modeling approach for solving all kind of scientific problems. Reaching from natural sciences and financial problems to various engineering applications, MATLAB is the appropriate tool to use. It provides a unified platform for solving numerical problems, equipped with extensive functionality and flexibility. Especially speaking in terms of rapid prototyping, a designer wants to implement, test and evaluate algorithms quickly. Therefore, it is common practice to utilize an abstract way of modeling that is not concerned about complex specifics of the language in use. In that way MATLAB fits well as a first step of the entire design process.

All relevant aspects of the basic system structure can be represented in form of a computational model. To see things more clearly: the MATLAB script under consideration covers the transmitter model, an appropriate channel model and the receiver architecture for a CertLink modem. It simulates baseband data transmission in a sequential manner (frame-based communication), incorporates channel effects (AWGN, timing and frequency errors) and implements the receiver structure including all steps of synchronization except signal detection. This chapter gives insight in the modeling approach and presents details of implementation as well as system performance measures. This includes link-performance (BER-measurements) and characteristics of the synchronization algorithms.

4.1 Simulation Architecture

Since an OFDM receiver alone has no functional purpose, the simulation concept has to cover both, a transmitter and a receiver model, including the modulation/demodulation process as required for the CertLink modem. On one side there is the transceiver, being the data source, generating random binary data streams and performing OFDM modulation and on the other side the receiver, processing the incoming data, synchronizing to the frame and executing the following stages of demodulation. The simulation model has to cover all relevant parts of the signal processing chain. It aims at a certain degree of flexibility, in terms of parameter variation and calls for designated measures to evaluate the overall system. It is worth noticing here, that

the model will only consider baseband processing. So there is no attempt on describing the behavior in a realistic environment, because those aspects would focus more on RF engineering which is out of scope of this thesis. Such a realistic environment, as mentioned before, would include discussions about how to convert the data from discrete, digital domain to a continuous, analog domain, frequency up-converting, filtering, modulating the data (mixing) and vice versa on the receiver side. Additional impairments, such as DAC/ADC peculiarities, large PAPR and power amplifier characteristics (linearity) would have to be considered. Certainly, those problems will have to accounted for in the ongoing development process, but the proposed simulation architecture here focuses on a pure functional view and hence, describing the system in terms of baseband is absolutely sufficient.

The total simulation process is controlled via textual statements in the MATLAB script, which include parameters from the CertLink specification as well as simulation depended parameters, e.g. for controlling visual outputs. All relevant parameters of the OFDM simulation can be altered before each simulation run, therefore establishing a high degree of configurability.

In order to represent a possible communication scenario from one modem to another, the script is structured in a way to perform a frame-based data exchange. Frames can be understood as OFDM frames, containing several OFDM symbols which in turn carry the information. Speaking in terms of the MATLAB language, these frames are described by vectors or matrices propagating through the signal chain. This description is very powerful with regard to computational time, because vectors can be manipulated in parallel and hence decrease execution time. Normally, picturing a TTP communication, frames will be exchanged in a bursty manner - so there is virtually no continuous data stream. Since it is difficult to model a burst communication in a script based environment, the model will process frame after frame - depending on the desired maximum amount of frames to generate.

There are three main parameters that can change the simulation behavior:

- number of frames to send
- energy per bit to noise power spectral density ratio (E_b/N_0)
- frequency error

E_b/N_0 and the frequency error both have a certain sweeping range. In the case of E_b/N_0, the goal is to simulate different signal-to-noise ratios, modeling a changing link quality, whereas the frequency error sweep mainly aims at the evaluation of the synchronization algorithms, as discussed below.

The range for the signal-to-noise ratio per bit (or E_b/N_0), is chosen, such that enough data points are available for a reasonable BER comparison and that those points are distributed over different regions of interest. Because the BER tends to have a steep slope at higher SNR (at least for AWGN channels) it makes no sense to use a large amount of data points in that region. Areas of low SNR need a better resolution and therefore will reserve more points for computation. The BER of fading channels usually flattens out at higher SNR values, so the proposal of fewer data points also applies to this case. The total quantity of points has severe impact on the entire simulation time, since data transmission is done in a loop-fashion for all values of E_b/N_0. For now the range is defined from $0dB$ to $40dB$ either with an increasing step size or a logarithmic (equidistant) step size depending on the system configuration, although the range, being a system parameter, can be adapted for each simulation run.

It would be possible to incorporate even more parameters influencing the overall simulation behavior, but from a computational perspective this is not feasible[1]. During normal operation all values, except for E_b/N_0 are fixed. Of course it is possible to fix all parameters and have a look at one specific configuration.

The textual approach using MATLAB scripts features a quick way to establish an environment for a first concept of a modem design, but does not incorporate aspects of a *model-based* design. Nevertheless, most parts of the signal chain can be abstracted by sub-functions, therefore trying to encapsulate algorithmic parts to *"blocks of functionality"*. This gives an idea on how a model-based system could be structured and where which functionality is contained.

The following section describes the transmitter, channel and receiver in more detail.

4.2 CertLink Transmitter

In principle, the transmitter gathers data from the TTP bus and the payload interface, performs OFDM modulation and sends the data over a communication channel to the receiver. In this case, entirely done in baseband fashion - otherwise utilizing a RF link. A short overview shows which tasks are performed by the transmitter:

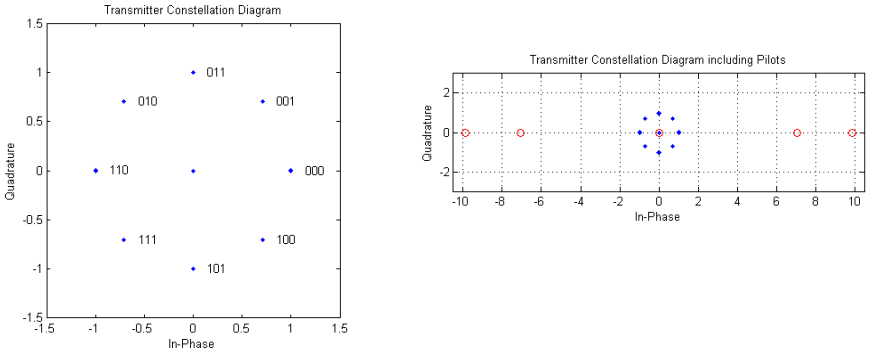
- Binary data generation
- RS-encoding (FEC)
- 8-DPSK modulation
- OFDM subcarrier mapping
- Pilot data generation
- Pilot insertion
- OFDM modulation (iFFT)
- Guard interval insertion

Since there is no interface to TTP in the model, the data has to be randomly generated by using uniformly distributed pseudo-random numbers restricted to the binary set $\{0, 1\}$. A vector of bits is formed, that includes the specified amount of information (depending on the number of TTP frames, payload frames and the intended sampling frequency of the TTP bus). At a first estimation, $15360\,bits$ are contained in one OFDM frame (2 TTP frames, 2 payload frames each $240\,bytes$ in size, sampling factor of 2). Notice, that no interleaving of TTP frames is done, because the frame was specified in a unified manner by the random number generator. The bit-stream is propagated to a RS-Encoder, employing measures for forward-error-correcting. Specified with a code-rate of n/k, where n being 255 and k is equal to 249, the coder adds additional information to the stream, hence enlarging the data vector. He does

1 Parameter sweeps can not be optimized by means of parallel computing since they have an inherent sequential behavior and consume a large part of computation time.

so by inserting redundant data bits at the end of each block of k-bits. The "RSEncoder" object from the Communication's Toolbox demands a bit input type consisting of an integer multiple of $k\lceil log_2(n+1)\rceil$-Blocks. Hence the encoder input has to be correctly padded, prior to encoding. Due to the principle of operation of the block-coder the data size increases from the padded vector length by the code-rate to 16320 $bits$.

Next, the data is differentially modulated by a 8-DPSK modulator (modulation order $M = 8$) conveying the information in *relative* phase transitions rather than absolute amplitude and phase. MATLAB provides a configurable modulator object which performs all necessary steps including the gray-encoding mapping scheme (constellation ordering). Three bits ($log_2 M$) are grouped together and mapped to a complex alphabet having eight distinct elements, hence reducing the vector size to a third of the original bit-stream. Fig. 4.1 illustrates the carrier modulation and pilot insertion in the frequency domain as constellation diagrams for the transmitted signal. The plot in Fig. 4.1a shows the 8-DPSK modulation of the data carriers after OFDM mapping, also visualizing the Gray-encoding scheme of the symbols consisting of groups of three bits. Two proposed pilot symbols are inserted in the frequency domain before applying the inverse Fourier transform that extend the constellation diagram. In Fig. 4.1b two BPSK modulated pilots are shown in *red* together with the data symbols in *blue*. Notice the enlarged symbol energy used for the pilot symbols as discussed further below.



(a) **(b)**

Figure 4.1: Constellation diagrams of the OFDM transmitter

At this stage the data is represented in the frequency domain, defined by a complex amplitude and phase and can be mapped to designated orthogonal subcarriers after parallelization. Parallelization means to distribute the information over a number of OFDM symbols - based on the requirements, eight such symbols are needed. Currently, at eight OFDM symbols, 680 parallel sub-channels are in use - considering redundancy, this amount doubles to 1360 channels. A frequency plan (not shown here) defines how the data mapping is done. The assignment itself is realized by an OFDM mapper method, deciding on which of the possible N_{iFFT}-frequencies each complex data sample will be transmitted and skips carriers where no data is assigned to. Frequencies containing no data are either due to linearity issues (DC), out of band radiation (Nyquist frequencies) or are reserved to future pilot symbols. OFDM mapping is implemented

in a sub-function performing several operations. Given the frequency plan specifications, the mapper computes the required vector indices for all frequencies and copies the data on the right locations. At this point the redundancy issue is covered by transmitting the same data on the positive and negative frequencies, consequently establishing a robust communication. Choosing $N_{\text{iFFT}} = 2048$ as the iFFT size, the mapping process results in a carrier utilization of about 67%, considering the number of parallel subchannels including redundancy.

The mapper output delivers a matrix consisting of 2048x8 elements. Each column corresponds to the complex information contained in one OFDM symbol, whereas each row specifies one distinct subcarrier frequency fitting the iFFT input. The synchronization algorithm implemented in the receiver requires OFDM symbols to determine the start of the frame, computing the timing offset and determining the frequency offset. For the purpose of pilot symbol data generation a function module is used which takes multiple input parameters. These parameters include the modulation scheme and the indices where to place the data symbols. The module generates two OFDM symbols (either BPSK or QPSK modulated) and appends them in front of the other eight symbols. See Fig. 4.1b for the resulting transmitter constellation diagram. Details about the module implementation will be described further below, see 4.3.2. All the data is fed to the iFFT, which synthesizes a time domain representation from the frequency domain description of the signal as shown in Fig. 4.2. Here the iFFT-synthesis of OFDM symbol #4 is presented for the in-phase component in the CertLink transmitter. The data is defined in the frequency domain using 2048 sub-carriers, whereupon the assignment of the parallel sub-channels is done via a OFDM mapping scheme. Redundancy is employed by duplicating the data on the other half of the spectrum - in between no data is assigned. Discrete time domain representation shows the superimposed orthogonal signals of each carrier, whereas no guard-interval is depicted here. The procedure, summarized as OFDM modulation, characterizes the transmitter output (a time signal) by a careful definition of orthogonal sub-carriers in the frequency domain.

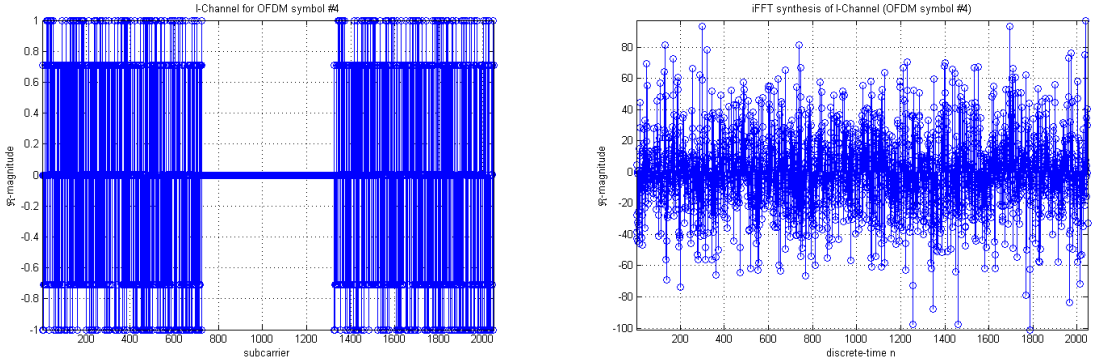


Figure 4.2: CertLink transmitter iFFT synthesis

To counteract inter-symbol interferences the time signal is extended by a guard-interval (GI) denoted as N_g, a cyclic repetition of original data, using matrix manipulation. The last 256 time domain samples ($1/8^{th}$ of FFT length) of each OFDM symbol (column vectors) are copied

and appended in parallel prior to the vectors resulting in a matrix size of 2304x10[2]. In this configuration the data is ready to be sent over the channel.

4.3 Channel Simulation

The following section focuses on how an AWGN can be theoretically described and simulated in MATLAB for an OFDM transmission system with respect to BER performance measurements. Additionally the characterization of timing and frequency errors together with their effects on the system is presented here.

4.3.1 AWGN Channel

The data is propagated through a non ideal baseband radio channel, moreover contaminating it with additive white Gaussian noise. In theory a sample function $w(t)$ from a random noise process, having a Gaussian amplitude distribution and a constant spectral density is added to the transmitter signal $s(t)$ causing distortions in the receiver, modeled by $r(t)$. Although such a model does not satisfy a realistic communication channel it establishes a first building block towards it. This first simulation concept will only cover such an AWGN channel for basic analysis of the system performance.

Performance measures like the BER are usually computed over a range of different normalized signal-to-noise ratios denoted as the "energy per bit to noise power spectral density ratio" (E_b/N_0). Therefore it is necessary to change the amount of additive noise power depending on the actual E_b/N_0 value under consideration. To realize a varying additive noise, an attenuation factor, μ is introduced leading to the basic relation (discrete notation)

$$r_{\mathrm{AWGN}}[n] = s[n] + \mu \cdot w[n]. \tag{4.1}$$

For computing μ based on the given E_b/N_0 additional relations are required. From the definition of E_b/N_0 a relation between signal and noise power can be developed [11]. The energy per bit is defined as

$$E_{\mathrm{b}} = \frac{P_{\mathrm{signal}}}{R_{\mathrm{bit}}}, \tag{4.2}$$

where P_{signal} is the carrier signal power and R_{bit} is the bit-rate. Similarly the noise power density can be written as

$$N_0 = \frac{P_{\mathrm{noise}}}{R_{\mathrm{symbol}}}. \tag{4.3}$$

2 The matrix dimension is determined by the number of FFT points N_{FFT}, the guard-interval length N_g and the number of OFDM symbols to transmit and can be expressed as $[N_{\mathrm{FFT}} + N_g]$x[#ofdmSymbols].

Since the FEC does not contribute to the SNR per bit an additional term has to be introduced which slightly lowers the desired E_b/N_0 for the channel. A "coded" E_b/N_0 is proposed accounting for the Reed-Solomon coding in order to meet the energy ratio at the encoder output, see [35]

$$\left(\frac{E_b}{N_0}\right)_c = \frac{E_b}{N_0} + 10\log_{10}\left(\frac{k}{n}\right). \tag{4.4}$$

Combining (4.2), (4.3) and (4.4) moreover explicitly stating the noise power, this resolves to

$$P_{\text{noise}} = \frac{R_{\text{symbol}}}{R_{\text{bit}}} \cdot P_{\text{signal}} \cdot 10^{\frac{-(E_b/N_0)_c}{10}}, \tag{4.5}$$

implicitly converting E_b/N_0 from decibels. Based on the noise power P_{noise} the attenuation factor a has to be expressed in terms of a voltage rather than power. Since the white noise is equally distributed in both the I and Q-channel, this leads to

$$a = \sqrt{\frac{1}{2}P_{\text{noise}}}. \tag{4.6}$$

The question arises how to compute the signal power of the transmitted OFDM signal because P_{signal} is required, in order to calculate the noise power. R_{symbol}, R_{bit}, and E_b/N_0 are already given as system parameters. Time domain OFDM signals consist of several signal parts which do not contribute to useful energy per bit. Among those are the pilot symbols, the guard-intervals of each OFDM symbol and the fact that not every subchannel is actually carrying data (FEC carriers). If none of these aspects are considered during the calculation of the signal power, the attenuation factor will simply not reflect the true, specified E_b/N_0. Therefore resulting in meaningless performance measures that actually can not be compared to single-carrier systems. Incorporating all non-contributing parts discarded by the receiver, concludes to

$$P_{\text{signal}} = \frac{\sum\limits_{\text{signal}} I^2 + Q^2}{\frac{N_{\text{symbols}}}{N_{\text{channels}}}} \cdot \frac{N_{\text{FFT}}}{N_{\text{FFT}} + N_{\text{GI}}}, \tag{4.7}$$

where I and Q are the in-phase and quadrature-phase signal components excluding the pilot symbols, N_{symbols} accounts for the number of OFDM data symbols, N_{channels} denotes all parallel sub-channels containing information, N_{FFT} is the number of FFT-points and N_{GI} relates to the number of samples used for the guard-interval. Eq. (4.7) normalizes the total signal power to **one** subcarrier modulated with useful data contributing to energy per bit.

Remark: It is possible to switch between two ways of computing the signal power. One by means of a predestined E_b/N_0 ratio and the other being based on SNRs. For the case of SNRs the total signal power is calculated by taking into account the pilot symbols and all the data symbols. As before, this power has to be normalized and scaled correctly with respect to the utilized subcarriers. Configuring the AWGN channel in terms of SNR is used wherever the performance of the estimators is evaluated (see Section 4.5), whereas E_b/N_0 ratios are required for bit error computations.

The AWGN channel model takes samples $w[n]$ from a normally distributed random process using the `randn`() function having zero mean and a variance $\sigma_w^2 = E\{|w|^2\}$, scales them by the factor μ and adds them to the signal, entering the channel. This relation is now interpreted as $r[n] = s[n] + w[n]$, where $w[n] = \mu \cdot w[n]$. In this way the channel accommodates for the varying E_b/N_0 by correctly determining the amount of noise power to add.

4.3.2 Frequency and Timing Errors

Given that OFDM systems are very sensitive to frequency and timing errors the channel has to model those impairments in order to represent a more realistic simulation environment. The goal here is to provide modifiable parameters that account for both types of errors used to test the implemented synchronization algorithms in the receiver.

4.3.2.1 Frequency Error

An additional frequency error can be modeled in the time domain as follows

$$r[n] = s[n]e^{j(\frac{2\pi\theta n}{N}+\phi)} + w[n], \tag{4.8}$$

where $r[n]$ and $s[n]$ are the received and transmitted samples respectively and $w[n]$ is the additive white noise. The introduced complex exponential, defined as the angular phase factor, is made up of the frequency error θ normalized to the subcarrier spacing Δf and an arbitrary initial phase offset ϕ. This time domain phase factor results in a frequency shift and destroys the orthogonality among the subcarriers, which would lead to unacceptable error rates if not corrected. Fig. 4.3 shows how different frequency errors change the phasor evolution over time. This representation depicts different frequency errors (angular phases) in the time domain over one symbol duration. The picture on the left shows a three-dimensional plot for two errors of 12.4 and 0.9 subcarrier spacings. Larger values of θ result in more frequent phasor rotations over time and a corresponding large shift in the frequency domain. Projection onto the imaginary axis results in the plot to the right, providing another view for better visual understanding.

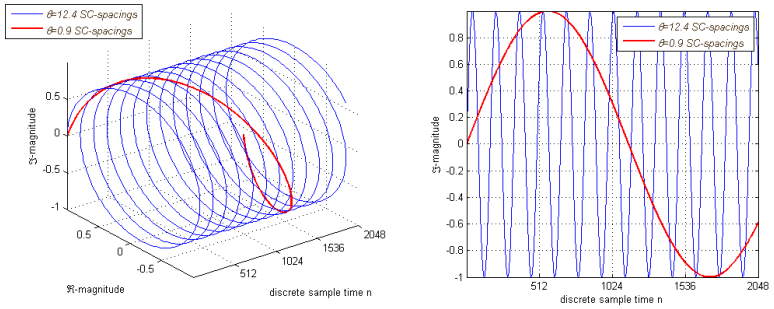
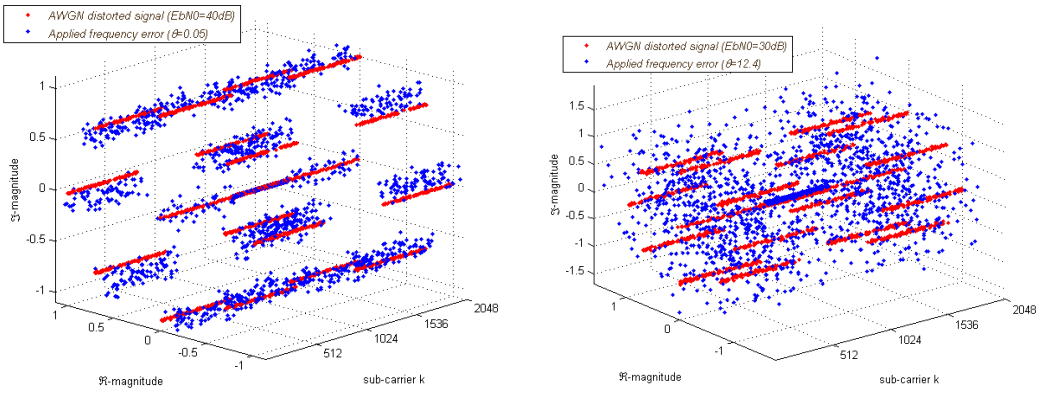


Figure 4.3: Time domain representation of the complex phase introduced by a frequency error



(a) **(b)**

Figure 4.4: Effects of different frequency errors in the frequency domain

It is a major challenge for the receiver to compensate for this shift in the frequency domain, making sure that the OFDM demodulation can be performed correctly. Frequency errors of about 4-5% of the subcarrier spacing can be tolerated without severely degrading the SNR, as stated in [10]. Two different cases of error impacts can be distinguished. Either the error is an integer multiple of the SC-spacing which leads to a frequency shift of integer positions or an error is appearing as a fraction of the frequency separation. The former case preserves orthogonality, although the carriers appear at the wrong FFT output in the receiver, the latter destroys orthogonality, therefore causing serious distortions. In Fig. 4.4 the effects of different frequency errors in the frequency domain are shown as a 3D evolution of the constellation diagrams. The plots show OFDM symbol #3 (8-DPSK modulated and carrier mapped) for each subchannel after passing the AWGN channel at different E_b/N_0, indicated in *red*. *Blue* points illustrate the impact of different values of θ on the data constellation. Fig. 4.4a shows a frequency error of 5% of the subcarrier spacing resulting in a minor shift which allows for correct demodulation without error compensation. In contrast, the plot in Fig. 4.4b depicts the severe distortions resulting from a much larger offset, making the demodulation of the data virtually impossible. The frequency error θ is either controlled via a single global system parameter or by a global vector, describing a certain range of errors.

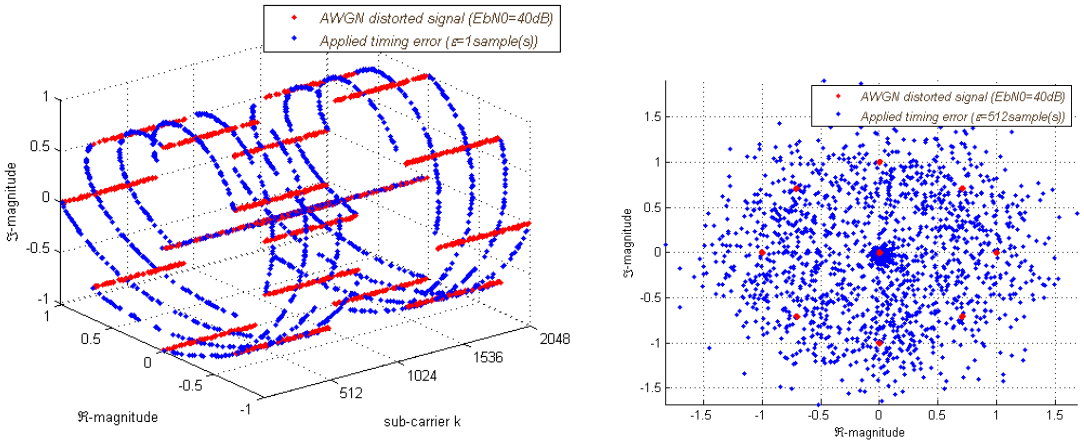
4.3.2.2 Timing Errors

Modeling a timing error is simply done by delaying the signal vector and inserting padding zeros. This approach tries to model variable "unknown" arrival times of the OFDM frame. Notice that there is a lack of realism when looking at a zero amplitude signal without any noise (as in this case with zero padding), since noise is naturally present at all times. Including the timing error model in (4.8) results in the following final equation implemented in the MATLAB script

$$r[n] = s[n - \varepsilon]e^{j(\frac{2\pi\theta n}{N} + \phi)} + w[n], \tag{4.9}$$

where ε is the delay expressed in terms of discrete time samples, normalized by the sample time T_s.

The receiver needs to estimate this timing offset in order to start the demodulation process using the FFT. If the FFT window is aligned in the range of the GI, only a linear phase rotation of the subcarriers can be observed, related to the time-shift property of the Fourier transform. Such a rotation can be resolved using either channel estimation by employing coherent modulation schemes or directly by differential modulation/demodulation. Computing the FFT outside the guard-interval results in inter-block interference and a loss of orthogonality among the subcarriers, hence leading to inter-carrier interference which in turn degrades the SNR compared to a perfectly synchronized system. Fig. 4.5 gives an impression how different timing errors impact the OFDM demodulation and shows the simulation results of timing error effects after FFT demodulation over one OFDM symbol, having 2048 subcarriers. Placing the FFT window inside the guard-interval (offset is one sample) already leads to a linear phase symbol rotation across the subcarriers, as shown in Fig. 4.5a. The plot Fig. 4.5b visualizes the data symbols using a wrong FFT window alignment outside the GI. The projection on the time axis shows how fatal the distortions are. Clearly the orthogonality is destroyed leading to a wrong FFT output resulting in a high BER - in fact the data on the subcarriers is lost and can not be recovered. *Red* data points relate to the output of the AWGN channel in the frequency domain whereas the *blue* points represent data after the FFT, when applying a timing offset.



(a) **(b)**

Figure 4.5: Simulation of timing error effects during FFT demodulation

4.4 CertLink Receiver

The CertLink receiver acts as a counterpart of the transmitter, implementing all steps of the modulation process in reverse order. Additionally, the receiver focuses on the important aspect of synchronization, especially on implementation and simulation. As described in Section 4.3, the transmitted signal encounters different distortions when passing the communication channel. An AWGN channel simulates different signal-to-noise ratios whereas frequency and timing errors emphasize more on OFDM related impairments. Without accurate synchronization no reasonable bit error rates can be achieved, since OFDM systems are very sensitive to frequency

errors, as demonstrated before. The process of synchronization consists of several sub-tasks that need to be executed before the actual OFDM demodulation takes place. These tasks are basically divided into parameter estimation (frequency error θ and timing offset ε) followed by error compensation.

This section concentrates on how different synchronization algorithms are implemented in MATLAB and compares them to find a suitable method for a CertLink receiver. The following bullet list roughly summarizes the computational steps of the receiver part:

- Parameter estimation (frequency error, timing offset)
- Error compensation (frequency error compensation, timing offset)
- Guard interval removal
- Pilot removal
- OFDM demodulation (FFT)
- OFDM subcarrier demapping
- 8-DPSK demodulation
- RS-decoding
- BER computation
- Estimator analysis

Section 2.2 already mentioned the theoretical aspects of synchronization in OFDM systems and gave an impression which algorithms are suitable for implementing in hardware. The developed MATLAB script evaluates three particular algorithms and compares them based on characteristic (statistical) measures from estimation theory. Arguably, a vast amount of algorithms are published in the field of OFDM synchronization, but since several papers already have drawn lots of comparisons and conclusions this thesis will not try to reiterate simulations for all of them. Instead a small selection is chosen to validate the ideas behind synchronization, always keeping in mind the implementation aspects regarding an executable hardware specification. This is justified by the fact that most algorithms available today are following nearly the same original concepts, most of the time only with minor modifications and improvements. The algorithms under discussion focus on pilot-assisted timing and frequency synchronization relying on repeating patterns in the transmitted signal. Basically, a sliding window correlation is applied, that tries to estimate for appearing errors using specifically designs metrics.

In order to provide a reasonable simulation framework for these algorithms in MATLAB, the incoming frames have to be slightly modified. A sliding window must me able to take continuous samples from a data-stream, but to this point all data is propagated via matrices or vectors through the signal chain. Imagine a timing offset of zero samples - then the first sample, part of the pilot of interest, is also the first element in the vector by design. No samples from previous frames can be accessed by the correlator, so to accommodate for this, each frame is (cyclically) shifted by half the frame size. In fact now the pilots are placed somewhere in the middle of the frame depending on the desired offset. Due to the cyclic shift, samples from the data symbols

are treated as “noise” for the duration of synchronization. After all errors are estimated, the shift is undone and the errors are compensated for.

It is not straightforward to compare different synchronization algorithms in a single simulation run. The reason being that each algorithm strongly depends on distinct pilot symbol schemes distinguished by their digital modulation scheme or structure. There can be only *one* certain pilot symbol type per cycle and this type has to be configured manually. Thus there is exactly one[3] algorithm providing the estimation parameters. However the comparison is absolutely possible if the synchronization methods use the same pilot structure or rather are compatible to each other. The metrics characterize the employed synchronization method and give a good impression about how large, e.g. the estimator variance is expected to be.

The implementation of the selected algorithms will now be discussed more thoroughly.

4.4.1 Timing and Frequency Synchronization

All implemented algorithms rely on finding the maximum correlation value utilizing repetitive parts of the pilot symbols embedded in the datastream. The design of this repetitive structure is crucial for the algorithm to work properly and heavily influences the outcome of the correlation metrics. In fact, the pilot symbol generation and incorporation in the datastream has to be accomplished by the transmitter. But since the topic is more related to the receiver functionality it is covered within this section.

4.4.1.1 Schmidl's Algorithm

Schmidl proposed a correlation metric based on a preamble $\rho = [A_L, A_L]$ having two identical halves A of length L (in the time domain), where $L = \frac{N}{2}$ and N is the number of FFT points[20, 21, 22]. In order to achieve this symmetry a QPSK-modulated pseudonoise sequence is transmitted on all the even frequencies whereas zeros are used for all odd frequencies. Based on the properties of the iDFT (iFFT), synthesizing this frequency description results in a time-symmetry. Additionally a second symbol is employed for estimating the integer frequency offset and providing a method for channel estimation. Because this system uses a non-coherent modulation for the payload - no channel estimation has to be performed and frequencies reserved for this estimation could be used otherwise, possibly for actual payload data. The structure of the second preamble uses two different modulated PN-sequence on even and odd frequencies. The type of pseudonoise sequence can be arbitrarily chosen without degrading the estimation performance, see[21]. So here, pseudorandom binary samples are generated by a random number generator (uniform distribution) using the `rand`() function.

Two versions of the Schmidl algorithm are implemented. First a straight forward version using the vector notation in MATLAB and second a recursive version focusing more on a feasible hardware implementation.

3 There is one fixed algorithm for frequency estimation but three different possibilities for timing estimation. Therefore, *two* different methods can possibly operate at the same time - they are chosen to be compatible.

The principle is the same for both: *Calculate the normalized autocorrelation between the received complex samples $r[n]$ and L-delayed samples $r[n-L]$ over a sliding window of length $2L$ to form a timing metric based on which the timing offset is estimated.*

The vectorized proposal directly implements the following non-causal, discrete autocorrelation

$$P[d] = \langle \mathbf{r}_d, \mathbf{r}_{d+L} \rangle, \tag{4.10}$$

where $P[d]$ is the dot[4] product of two windowed complex L-sample vectors $\mathbf{r} = [r[1], r[2], ..., r[L-1], r[L]]$, one being delayed by half the symbol length and d being the first sample in the sliding window. Essentially Eq. (2.16) is rewritten in a more compact vector form. Similarly, the received energy of the second symbol, used as a normalization factor is given as

$$R[d] = \langle |\mathbf{r}_{d+L}|, |\mathbf{r}_{d+L}| \rangle. \tag{4.11}$$

Scaling (4.10) by (4.11) results in the timing metric M, as proposed by Schmidl

$$\mathbf{M} = \frac{|\mathbf{P}|^2}{\mathbf{R}^2}. \tag{4.12}$$

Here the bold face letters represent vectors $\mathbf{P} = [P[1], P[2], ..., P[\Gamma - 1], P[\Gamma]]$ and $\mathbf{R} = [R[1], R[2], ..., R[\Gamma - 1], R[\Gamma]]$ whose length is depending on the computation interval size Γ. The running lag, denoted as $d \in \Gamma$ is defined over this range, that allows for a reasonable comparison of all three algorithms. The interval defines a window of $\Gamma = 4353$ samples centered around half the received signal length and was chosen to include two full OFDM symbols, one guard interval and an additional sample (Minn) which in turn leads to a nice timing metric diagram, as shall be seen later on in 4.5.

Although this first direct implementation looks intuitive, it requires L multiplications and $L-1$ additions for one element of $\mathbf{P}$ and even more for $\mathbf{R}$ because of the required absolute value. Utilizing a recursive implementation as suggested by Schmidl can significantly reduce the the computation time. Based on equations (2.17) and (2.20) the procedure looks like the following: Compute the first value $P[1]$ and $R[1]$ of the window using vector notation. Afterwards calculate all other values for $d > 1$ recursively using

$$P[d] = P[d-1] - r^*[d]r[d+L] + r^*[d+L]r[d+2L] \tag{4.13}$$

$$R[d] = R[d-1] + |r[d+2L|^2 - |r[d+L]|^2 \tag{4.14}$$

where $()^*$ denotes the complex conjugation.

For the first values of $\mathbf{P}$ and $\mathbf{R}$ the number of operations is equal to the previous case. But all other elements require only 2 multiplications and 3 additions for the case of $\mathbf{P}$ and slightly

4 About 15% of computation time can be saved if the correlation is explicitly implemented without using the `dot()` command from MATLAB

more for **R**, resulting in an improvement in terms of computation time by a factor *100*. It is worthwhile noticing here that it is possible, following [22], to compute a rough timing estimate using the regular implementation if not every sample is processed in (4.10), (4.11) and (4.12),respectively. Under these circumstances equal results regarding the computational complexity can be accomplished. This approach is intentionally used for frame detection and can not be directly applied in the context of a meaningful timing error correction. Simulation verified that if only one out of 100 samples is used for computing the timing metric, a coarse estimate is possible but additional effort has to be spent on calculating the best timing point. Since the recursive method is still superior, no other variant will be pursued further.

In order to compute the the actual timing error the 90% averaging method as described in Section 2.2.4 and suggested in [21] is facilitated, rather than a direct peak maximization. The former method calculates the maximum value and its index of the vector **M**. All samples above 90% of the maximum are extracted and those left and right of the global maximum are taken for averaging. The resulting index from the averaging process indicates the timing error inferred from the computational range - so to operate in the frame based simulation environment, the corresponding offset introduced by shifting of the samples has to be removed properly. Schmidl's method generally introduces a plateau inherent to the metric which leads to a large estimator variance and some uncertainty for the start of the frame. Averaging tries to find the center of this plateau, hence improving the performance.

Since Schmidl's method aims at a joint timing and frequency estimation the frequency offset can also be determined. The fractional part of the frequency error, denoted as $\hat{\theta}_f$ is directly calculated from Eqn. (2.22) by determining the phase of **P** at the best timing index $\hat{\varepsilon}$ and scaling by π. Fig. 4.6 depicts the evolution of the estimated fractional frequency error showing how the L-lag autocorrelation of the symmetric preamble ρ leads to approximately the same phase for the start of the frame. Here, the time dependent characteristics of the scaled angular phase over the interval Γ at $E_b/N_0 = 0dB$ and $\theta = 12.4$ SC-spacings is shown. Around the center of the discrete time window the normalized phase is identical to the fractional part of the appearing frequency offset. The square shows the estimated error directly depending on the previously estimated timing offset $\hat{\varepsilon}$.

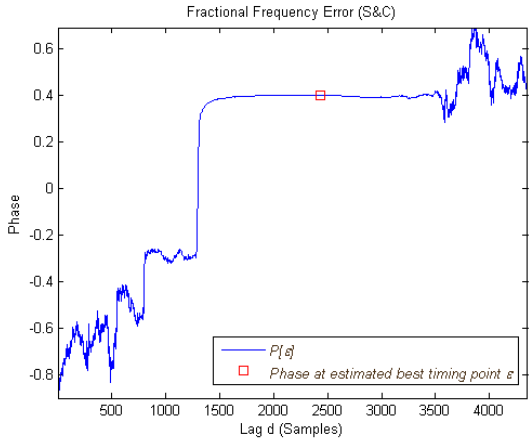


Figure 4.6: Evolution of the estimated fractional frequency error

This way of estimating the fractional frequency offset is used for all other algorithms since they

only provide measures to account for the single task of timing estimation. S&C also provide a method to identify integer frequency offsets, but because this involves computations in the frequency domain, it will be discussed separately.

4.4.1.2 Minn's Method

Minn's method tries to improve the estimator variance by a different way of averaging. However it facilitates the same preamble structure as S&C. The normalization factor for the timing metric is now given as

$$R_m[d] = \frac{1}{2}\langle|\mathbf{r}_{d+2L}|, |\mathbf{r}_{d+2L}|\rangle, \tag{4.15}$$

which calculates the average power over two OFDM symbols. So the new metric, denoted as $\mathbf{M}_f$ is then again averaged over a window Ω of length $N_g + 1$ by

$$\mathbf{M}_m = \frac{1}{\Omega}\sum \mathbf{M}_{f_\Omega}, \tag{4.16}$$

where the best timing point is simply calculated by finding $argmax(\mathbf{M}_m)$. The estimator variance is reduced at the expense of an increased computation time due to the additional expensive averaging. See Section 4.5 for results. Because the window size for averaging is different than for every other algorithm, the computation range Γ for the metrics has to accommodate for this to give meaningful and comparable results.

4.4.1.3 Park's Method

To improve the timing estimation of Minn et. al. even further, Park proposed a different preamble consisting of four parts $\rho_r = [C_{N/4}, D_{N/4}, C^*_{N/4}, D^*_{N/4}]$ that has a Hermitian symmetric ($C_{N/4}$ symmetric to $D_{N/4}$) structure leading to a sharp impulse shaped timing metric. To generate such a symbol structure a BPSK modulated pseudonoise sequence is mapped to all even frequencies before iFFT processing, whereas zeros are used for all odd frequencies. This simple frequency domain definition leads to the symmetric structure in the time domain. Because the basic symbol structure is similar to that of S&C, the S&C algorithm for finding the frequency offset can be applied to same the preamble ρ_r [26]. The algorithm implements two new correlations

$$P[d] = \langle\mathbf{r}_{d-L}, \mathbf{r}_{d+L}\rangle, \tag{4.17}$$

$$R[d] = \langle|\mathbf{r}_{d+L}|, |\mathbf{r}_{d+L}|\rangle, \tag{4.18}$$

where the vectors $\mathbf{r}$ are of length $L + 1$. Besides, the notation pictured here achieves is significantly faster implementation than a slow sequential mapping of equations (2.29), (2.30). The execution time of Park's algorithm is about 50% faster than that of Minn - so compared to every other implementation this method ranks second in terms of time complexity.

4.4.1.4 Timing Correction

Now that the basic implementation issues of the different estimators have been discussed - three different synchronization points are available for error compensation. As mentioned before, only one timing point can be used for further signal processing. Independent from the chosen algorithm, the following steps in the receiver remain the same:

First and foremost, the time-shift of the received sample vector is reversed and the timing error is compensated for. Since one frame is composed of a fixed amount of OFDM symbols and samples (depending on the system parameters) the timing error is corrected by shifting these samples to the left by the appropriate quantity. Therefore discarding the introduced channel delay. This procedure simulates the correct alignment for FFT demodulation in the receiver so that the signal vector contains only samples of interest.

As stated in Section 4.4.1.1 and explained in Section 2.2.4, the algorithm is also capable of estimating integer frequency errors, i.e. offsets in terms of multiple integer subcarrier spacings.

4.4.1.5 Integer Frequency Offset Estimation

To begin with, the algorithm corrects the first two OFDM symbols (containing the pilots) by the predetermined fractional frequency offset. A counter-rotation of the samples by multiplying with the complex exponential $e^{-j(2\pi n\hat{\theta}_f N^{-1})}$ will compensate for the error. The rotation leads to a frequency shift after applying the FFT - hence performing a realignment of the frequencies to integer multiples of the SC-spacing. Now the guard-interval can be removed from the frequency corrected pilots using matrix truncation. Both pilot symbols are transformed to the frequency domain and zero padded to two full OFDM symbols as required by the following normalized sliding correlation. The normalization factor can be computed *once* prior to correlation and is given as

$$G = 8\langle|\mathbf{X}_{2,e}|, |\mathbf{X}_{2,e}|\rangle^2, \tag{4.19}$$

where $\mathbf{X}_{2,e}$ denotes the vector of the second pilot for all even indices. So the vector length is effectively reduced to L-samples.

The metric B provides a measure to estimate the integer frequency offset using

$$B[g] = \frac{|\sum_{g=0}^{L-1} \mathbf{X}^*_{1,e+2g} v^*_k \mathbf{X}_{2,e+2g}|^2}{G}, \tag{4.20}$$

and searching for $\hat{g} = argmax(\mathbf{B})$ to maximize the metric vector $\mathbf{B} = [B[1], B[2], ..., B[L]]$. The parameter $g \in G$, $G = \{m \in \mathbb{Z} | 0 \leq m \leq L-1\}$ accounts for all possible even frequency shifts ranging from $[-L/2, +L/2]$ where $\hat{g}$ is the estimated offset factor. To translate this in terms of subcarrier spacing, $\hat{g}$ has to be properly scaled (offset removal and normalization) in order to reveal the final even integer frequency offset, which then has an effective range of $[-L, +L]$:

$$\hat{\theta}_i = 2(\hat{g} - \frac{L}{2} - 1). \tag{4.21}$$

The resulting frequency offset is then composed of the fractional and integer offset:

$$\hat{\theta} = \hat{\theta}_f + \hat{\theta}_i. \tag{4.22}$$

Fig. 4.7 shows the normalized metric $B(g)$ over the possible integer offsets, represented by g (as proposed by [20]). Notice that there is a large separation between correct and incorrect values due to the characteristics of the metric, as mentioned in [22]. So based on the statistics, it would be possible to use fewer samples to calculate $B(g)$ while still being able to correctly determine the integer offset. The simulation results are based on $B(g)$, calculated for a frequency offset $\theta = 6.9$ subcarrier spacings at $E_b/N_0 = 12dB$. As can be seen, the metric reveals a sharp impulse shaped characteristic with its maximum at index $g = 3$. This translates to an even integer frequency offset $\hat{\theta}_i$, corresponding to 6 SC-spacings, hence perfectly estimating the true offset θ_i.

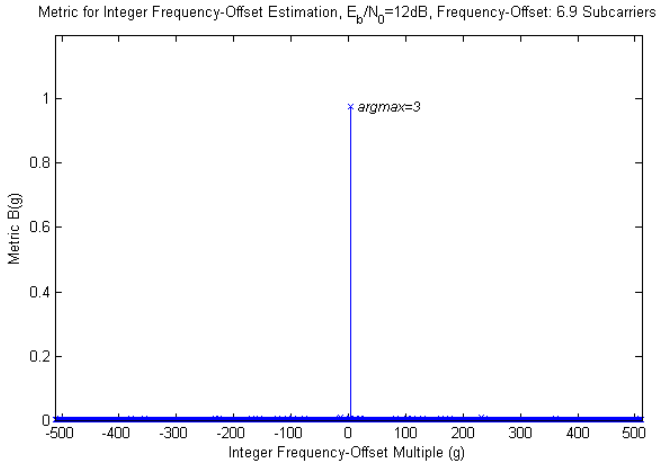


Figure 4.7: Metric $B(g)$ for integer frequency offset estimation

As soon as the entire frequency offset is estimated the timing-corrected samples are then frequency-corrected. The procedure for error compensation is exactly the same as previously described for the fractional frequency offset. This time all samples are multiplied by the complex exponential $e^{-j(2\pi n\hat{\theta}N^{-1})}$, so that each subcarrier is ideally realigned to its original position. Now that the synchronization process is complete, the guard interval is removed for every OFDM symbol. By simple matrix operations each symbol, represented as vector, is truncated by the right amount of reserved GI-samples in advance to the FFT.

4.4.2 OFDM Demodulation

The FFT size is matched to the iFFT size of the transceiver and should perfectly demodulate each parallel carrier of the synchronized data. Again, by matrix notation the FFT can easily be applied to obtain the FD-samples. By discarding the first two column vectors of the resulting matrix the pilot symbols are removed, since they already fulfilled their purpose. This leads to a matrix of size 2048x8 that can be processed by the OFDM demapper. Similar as in the transceiver this module selects all relevant indices (subcarrier) containing useful data and

extracts them. For now, the redundant data is simply *ignored*, hence only the data on the positive frequencies is used for further processing. Currently the demapper delivers a 680x8 matrix of complex data that is demodulated and Gray-decoded by a 8-DPSK demodulator object, opposite to that in the transceiver. The complex symbols are translated to a bit-stream and are handed over to the RS-decoder object. After forward-error-correction the data is ready for BER, FER calculation and analysis. All results and and data analysis will be discussed in the next section.

4.5 Simulation Analysis and Results

The main goal of this section is to give an overview of the achieved results. On one side the overall system performance and on the other side the performance and comparison of the implemented estimators for synchronization. Overall system performance covers the link performance of the physical layer by means of bit-errors and frame-errors whereas additional statistical measures highlight the characteristics of the synchronization algorithms in terms of timing and frequency errors.

To begin with, the results of the estimators are presented as follows.

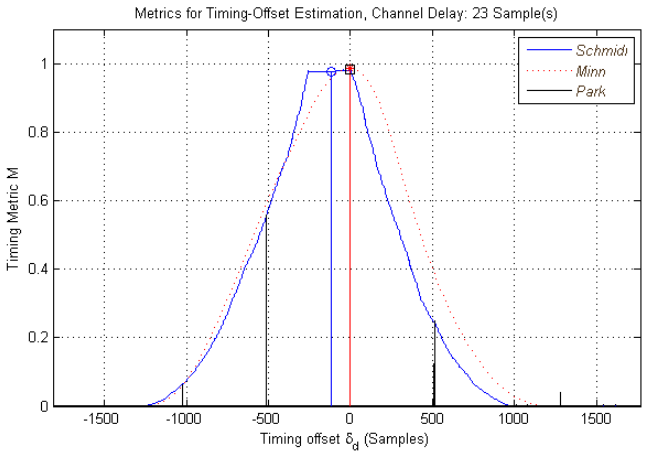


Figure 4.8: Normalized timing metrics after Schmidl, Minn and Park

Fig. 4.8 summarizes the results of timing-offset estimation for the three proposed algorithms and gives insight into how they perform in comparison to each other. All graphs are interpreted as the outcome of the correlation window sliding along the input samples, whereas the normalized timing metrics are shown over the defined interval Γ. The metrics and their corresponding estimates are shown for an introduced channel delay (timing offset) of 23 samples and are normalized to unity. Fig. 4.8 illustrates the abscissa in fragments of the relative timing offset $\delta_d = \hat{\varepsilon} - \varepsilon$ in order to be independent of the actual channel delay. The S&C algorithm clearly shows the metric plateau that typically (in case of an AWGN channel) extends over the length of the cyclic prefix - in this case equal to 256 samples. This unavoidable plateau normally leads to a large estimator variance if no additional treatment is applied. However, using the averaging method, as described earlier, limits the variance and places the estimate at about the center

of the plateau as can be seen from the simulation results. This time-index indicates the start of the useful part of the actual OFDM frame processed in the receiver therefore identifying the alignment of the FFT window. The true offset is indicated as $\delta_d = 0$ where the estimate coincides with the channel delay. Minn's method leads to smoothed slope and a better estimate, evident from the extra averaging. The third algorithm proposed by Park has the sharpest metric and exactly matches the true timing-offset at index 0. Fig. 4.8 can be viewed as an evolution of estimation algorithms for OFDM synchronization. Based on S&C each method shows a better performance in terms of the timing-estimate by utilizing new metrics or different preambles. To compare all three algorithms the BPSK modulated preamble from [26] was used because its structure is compatible to S&C and hence also to Minn's procedure. The figure shows a detailed view of resulting metrics after S&C for the recursive implementation. Fig. 4.9a shows a close up of the metric plateau and the estimated best timing point when using the 90% averaging method. In Fig. 4.9b the raw (unnormalized) autocorrelation $P[d]$ between the two pilot symbols is depicted. The last plot seen in Fig. 4.9c completes the picture by displaying the calculated power $R[d]$ of the second pilot symbol used as a normalization factor. Notice the rising power as the correlation window slides along the preamble samples. Every plot is computed over absolute lags d given in terms of the computation window Γ.

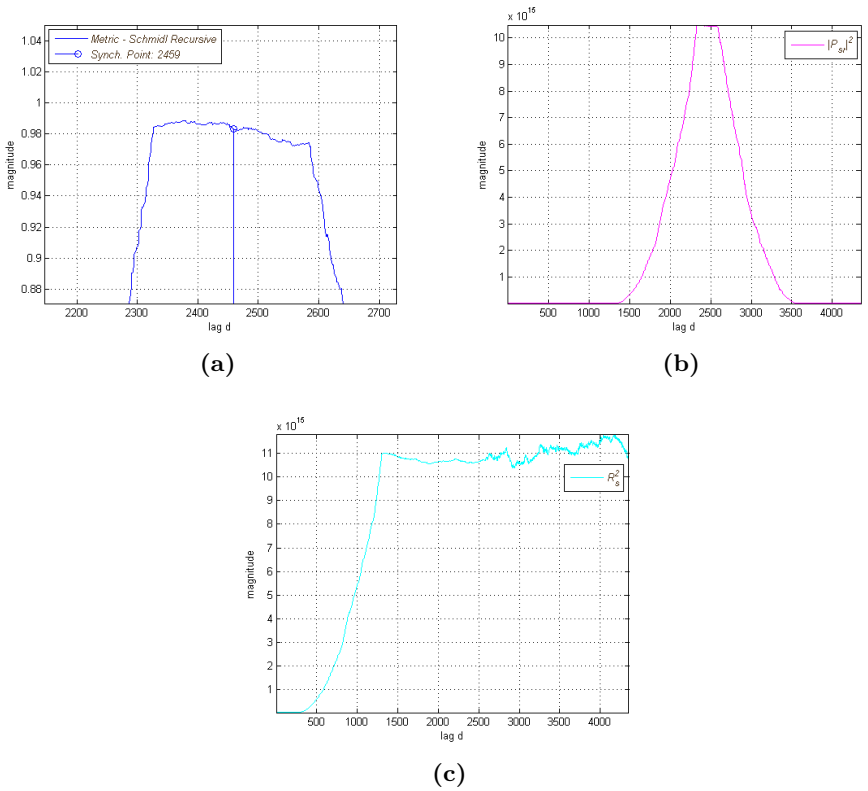
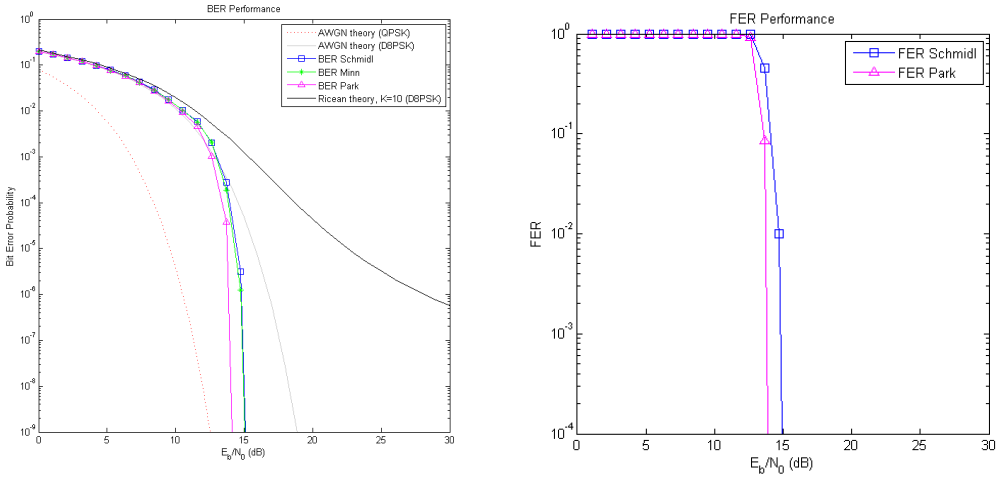


Figure 4.9: Detailed view of resulting metrics after S&C for the recursive implementation

Fig. 4.9 takes a closer look at the S&C results and shows some details concerned about the metrics itself. Depending on the SNR the metric of Schmidl's approach will change its characteristics as seen from Fig. 4.9a. It is clear that a simple global maximum search will likely

cause the estimate to fall of the plateau. Even a small deviation of a single sample can result in severe SNR degradation and ISI since the FFT would then take samples from an adjacent symbol as pictured in [21]. The flatness of the plateau could also be improved by using a different normalization factor as shown by [25], where they simply used all samples from one symbol to compute the half symbol energy. For completeness Fig. 4.9b and Fig. 4.9c visualize how the timing metric M is composed. Basically $P[d]$ is the unnormalized autocorrelation and already reflects the idea of a metric to detect frames and estimate timing-offset. The computation of $R[d]$ accounts for the symbol energy to normalize $P[d]$ to unity, if the the energy reaches a maximum (high SNR).



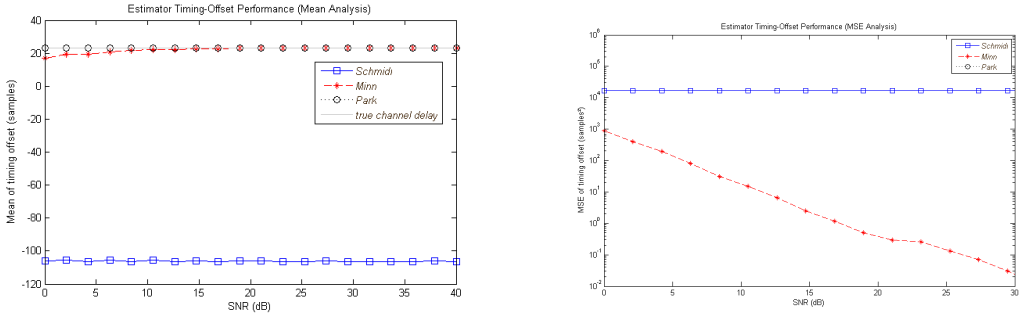
(a) **(b)**

Figure 4.10: BER and FER computations for the CertLink wireless data link

Fig. 4.10 compares the three (two) estimators based on their achieved bit error ratios and their frame error rate moreover visualizing the system performance of the synchronized CertLink wireless data link in an AWGN channel. In Fig. 4.10a a typical plot shows the bit-error probability over values of E_b/N_0 in comparison to selected theoretical BERs for single-carrier systems. Additionally, the three different synchronization algorithms are compared by means of their attained bit error probabilities. Under the current AWGN channel configuration the system shows a reduced BER over a theoretical single-carrier implementation due to FEC of the RS coding scheme. On the right Fig. 4.10b depicts the evolution of the frame error rate calculated over the same range of E_b/N_0 for two of the synchronization algorithms. All these theoretical BERs are either computed by using available MATLAB system functions for existing channel models or the `bertool` provided the data. Every one of the algorithms shows comparable results where the BER is improved over a theoretical uncoded AWGN channel for a D8PSK single-carrier system. This results can be explained by the fact that the system uses a FEC mechanism to attain lower bit error ratios over uncoded systems. While the algorithms after Minn and Schmidl show a similar performance in terms of their BER, Park's method leads to the best results since the timing offsets can be perfectly estimated, as seen below from Fig. 4.11. A theoretical BER performance under a Ricean fading channel is also included for convenience, which is configured to have a K factor of 10 and diversity equal to one. Note that

the data is generated by the `bertool` and provides only limited configuration options, therefore not representing the most accurate model. Fig. 4.10b distinguishes only two synchronization methods based on their frame error rates for completeness. Here, a frame error is present if one bit in an entire frame is erroneous and the plot again indicates the superior performance of Park's algorithm compared to Schmidl.

The following statistical analysis of the estimators is performed at a channel delay of 23 samples (1.1µs), a common frequency offset of 6.3 subcarrier spacings (75.6kHz) and a total of 400 simulations runs.

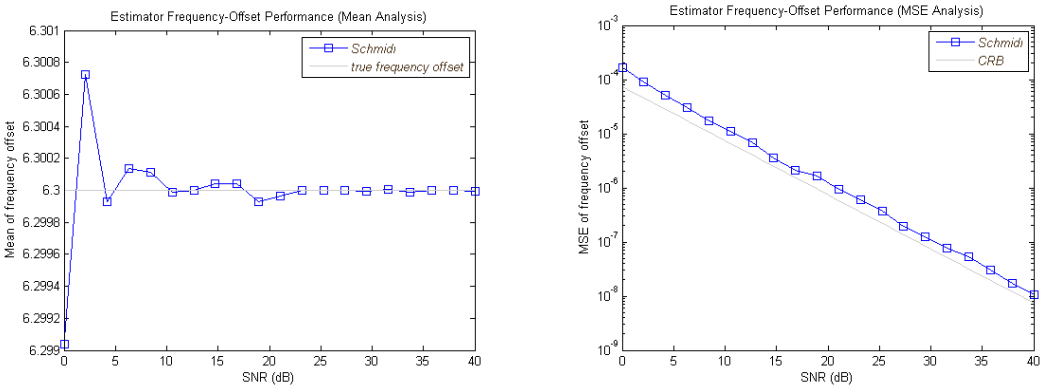


(a) **(b)**

Figure 4.11: Timing estimation performance of all three implemented algorithms

The performance of the timing-estimators in AWGN channels can be discussed with the help of Fig. 4.11 where the means and variances are compared. Fig. 4.11a visualizes the evolution of the expected value over different SNR values for 400 simulation runs. Together with all three simulation results the true channel delay of 23 samples is also pictured for reference. In Fig. 4.11b the second statistical performance measurement, the mean-squared error is shown for each implementation.

Again the previously discussed results about the timing-estimates are confirmed here. S&C shows that the expected value is shifted to approximately the middle of the metric plateau wheres for the other two cases the means are close to the true channel delay. Park's algorithms clearly outperforms the other two approaches. In the case of the mean-squared-error, shown in Fig. 4.11b, S&C reveals an estimation floor whereas Minn and Park do not. Since Park estimates the correct timing point under all conditions (AWGN) the variance is zero. From this follows that the simulation results of Park are not visible in the logarithmic diagram. Overall, Park's method performs best by means of the expected value and the estimator variance.



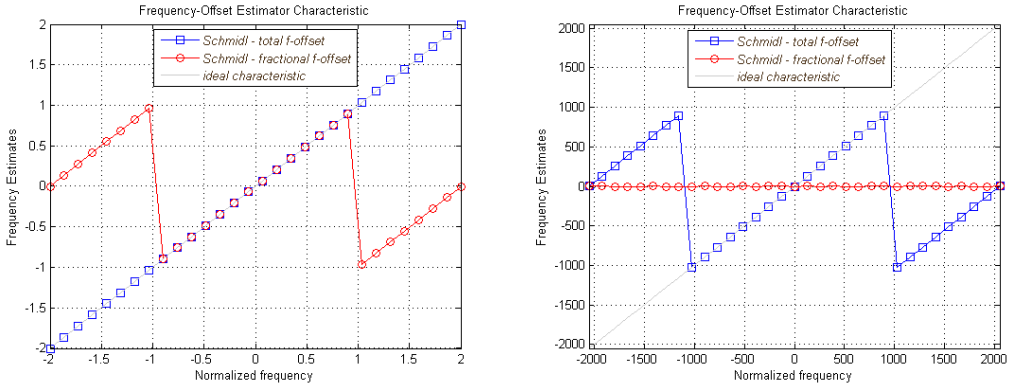
(a) **(b)**

Figure 4.12: Frequency estimation performance of S&C algorithm over different SNR

The same analysis can be applied to frequency offset estimation, as seen in Fig. 4.12, although only for the S&C algorithm since it is the only joint algorithm under consideration. The simulation results are plotted as a function of the SNR at a fixed frequency offset over 400 simulation runs. Fig. 4.12a shows the evolution of the expected value in comparison to the true offset. Here a combined frequency offset consisting of a fractional and an integer component was chosen. As the SNR increases the mean gradually approaches the true value. If the estimator variance or mean squared estimation error is observed in Fig. 4.12b, there is only a small offset of about $2dB$ from the bound. For this plot the Cramer-Rao bound of Morelli et. al, from [24] was chosen. The CRB is evaluated as

$$CRB = \frac{1}{2\pi} \frac{3(\mathrm{SNR})^{-1}}{N(1-1/N^2)}, \tag{4.23}$$

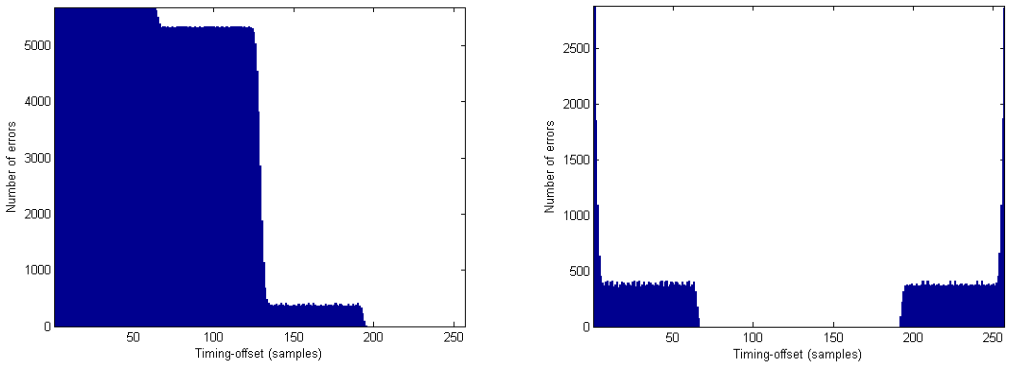
where the SNR is the total signal (entire OFDM frame) to noise power ratio and N denotes the FFT-length.



(a) **(b)**

Figure 4.13: Frequency estimator characteristics of the S&C algorithm

When comparing different frequency offset synchronization algorithm it is also common to have look at the estimator characteristics as a function of the normalized offset. This typical diagram illustrates an ideal (linear) characteristic as a reference where every frequency offset is perfectly matched. Generally speaking Fig. 4.13 compares the lock-in ranges of the fractional and the combined estimator algorithm over different normalized frequency-offset ranges. As emphasized in Fig. 4.13a, a fractional frequency offset can only range from $[-\pi, \pi]$ based on the use of the *atan* for calculating the phase difference of the two identical halves of the S&C preamble. Outside this range the ambiguities of the *atan* can not be resolved and therefore restrict the estimation range to only ± 1 subcarrier spacings. It is worthwhile noticing here that if the frequency error can be guaranteed to lie within one SC-spacing (e.g. precise oscillators with low tolerance) this approach is absolutely sufficient. Otherwise a combined version has to be employed that extends the range to $\pm L$ spacings, as depicted in Fig. 4.13b. Fig. 4.13b enlarges the viewable range of allowed frequency offsets from ± 2 SC spacings in Fig. 4.13a to the maximum of $\pm N/2$spacings for both estimation results.



(a) **(b)**

Figure 4.14: Adjustment effects of S&C timing-estimate in the frequency domain

An interesting phenomenon could be observed while developing and testing the S&C algorithm. As Speth et. al. state in [9] and [36], if the timing estimate lies within the range of the cyclic prefix, the orthogonality of the subcarriers is preserved although a linear phase rotation across the subcarriers will emerge after FFT demodulation. When using coherent modulation schemes this change in phase can not be distinguished from the changes in phase introduced by the channel transfer function and will be resolved by the mandatory channel estimation. The *same* applies for noncoherent differential modulation as employed in the system at hand. Fig. 4.14a shows the bit-errors versus different timing estimates between 0 and 256 samples, corresponding to the length of the guard interval. Despite the facts, the plot illustrates that only for small interval around the true[5] channel delay no bit-errors occur. Other timing points, especially at beginning, but also in the middle of the prefix show severe errors resulting in an unacceptable SNR degradation. Remember that S&C together with the 90%-method unfortunately forces the estimate to lie around the center of the metric. So, directly applying such a estimate for synchronization would lead to no reasonable system performance. The solution is found in

5 The useful portion of the first symbol starts after the cyclic prefix.

the frequency domain: the phase of the induced phasor is proportional to the timing error and can be compensated by a counter-rotation of $N_g/2$ samples using a complex exponential $e^{j(2\pi\mathbf{k}/N)N_g/2}$. Here the vector $\mathbf{k}$ represents every subcarrier from $[-\frac{N}{2}, \frac{N}{2}]$. After applying the compensation method the error distribution, as seen in Fig. 4.14b, reveals zero bit-errors occur around the center of the cyclic prefix which allows the S&C algorithm to operate as intended. The same applies for the other two algorithms, if their estimates are desired to be shifted in the TD to the middle of the cyclic prefix, as may be the case for ISI channels.

As a conclusion to this chapter Fig. 4.15 gives an overview of all "blocks of functionality" for the MATLAB simulation showing all major blocks and their relevant input/output signals. The reader is invited to use this block diagram together with the script, intended to speed up any future design or adaption process.

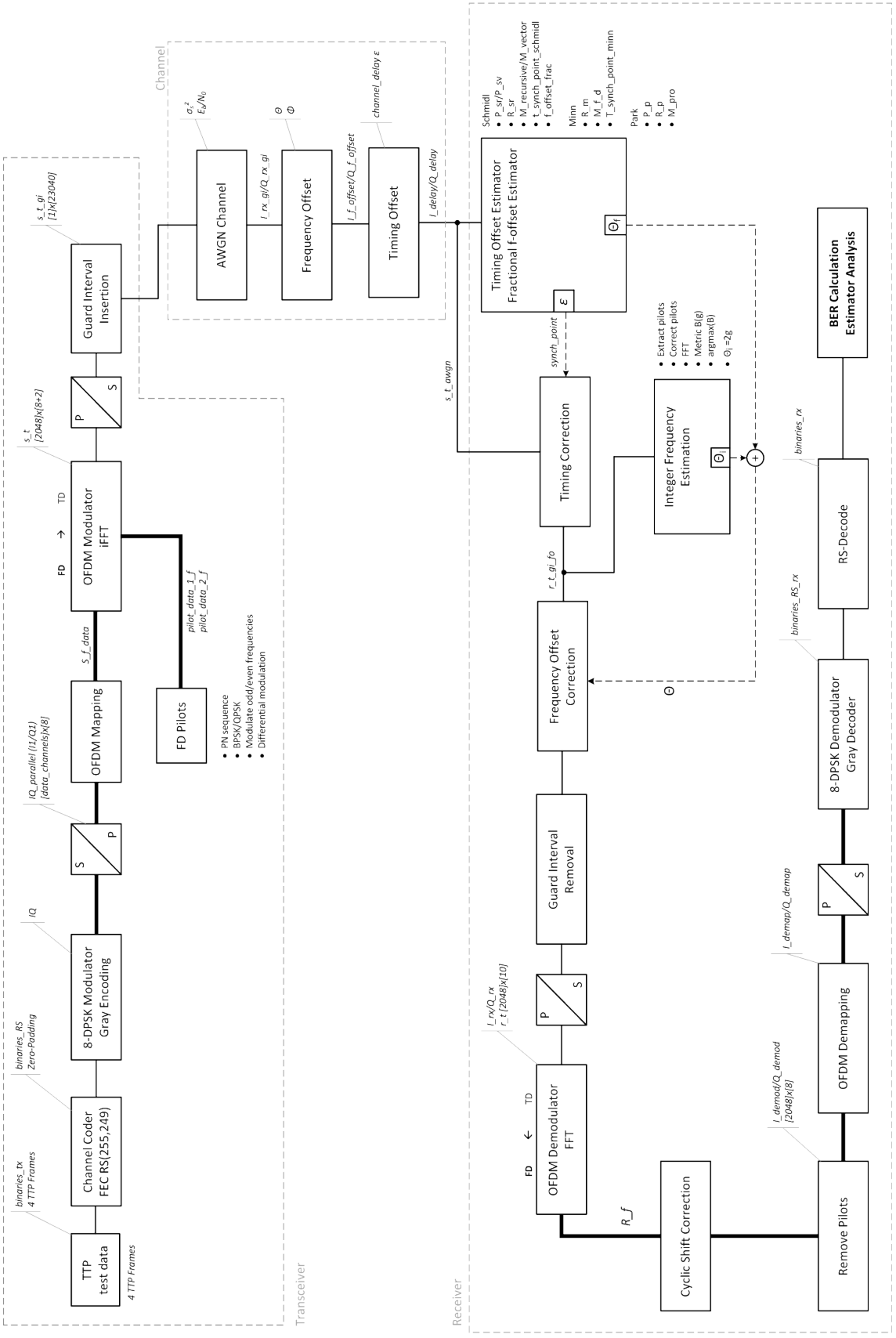


Figure 4.15: Block-diagram of MATLAB signal processing chain

5 Model-Based Implementation

Now that a functional MATLAB simulation framework is established, the task at hand is to select one sophisticated algorithm for a model-based hardware implementation in a FPGA.

A typical design flow would start off with a high-level model (e.g. JAVA) of the algorithm, that is already an executable specification and then start coding all the necessary HDL modules from there. If a JAVA model is assumed here (which could be a reasonable choice), it has to cover the design at it's lowest possible hierarchy by using basic elements such as multipliers, adders (maybe an "ALU" object with distinct functionality) and so forth. In that way the mapping to HDL is eased because the fundamental structure is already outlined and can be easily evaluated. So then every module that is part of the desired algorithm will be progressively translated to HDL until the system is complete and can be virtually assembled. After synthesizing the design[1] it needs to be tested in some hardware simulation environment. The JAVA model could for instance provide simple test-data, that is presented to the hardware description of the algorithm by using extensive test-bench scripts. Those scripts need to manage the control path manually, hence, much effort has to be put into a careful structured and clearly laid out design. Basically the test environment has to be implemented *twice* - one for the high-level model and one for hardware simulation and testing.

In principle, this approach makes sense and follows a clear top-down design method, where the task is first abstracted and then gradually improved until the lowest hierarchy (the physical layout) is reached. However, the transfer from a high-level model to an architectural HDL description is very error prone because of the translation "by hand". There is a certain breach within the flow that marks the weak point of the overall design. A mapping to HDL is not straightforward and is considered as an exhaustive task that has its own difficulties to cope with. Additionally, a large HDL design can quickly become quite complex that is hard to follow and to familiarize with in the first place.

Another main issue arises if the whole requirements - the specification of the implementation changes. This is assumed to be a normal process in today's design cycles and could emerge at any time. Following the flow described above, this would be considered a worst case scenario, especially if radical changes are expected. In this case, the new specification has to be represented by the high-level model and must be (again) mapped to a hardware description. Also a new testbench must be employed that covers the recent use cases from the requirements. All together, employing the design flow scheme described here, is very time-consuming to react to changes in the design - a rapid reiteration is simply not possible.

This thesis pursues a different approach that aims at a **unified** design environment where the executable specification - an abstract model - is also able to **directly** generate HDL code as well as testbenches for hardware cosimulation. The intention here is to use a design flow that

1 Here the flow is differentiated between ASIC and FPGA design

enables the engineer *"to work in the language of the problem rather than the language of the chip designer, e.g. VHDL or Verilog",* as cited from [37]. Normally the system designer works on an abstract level and develops the high-level model together with simulation and verification. The chip designer, on the other hand, "speaks" a different language at a much lower architectural level. However, he has to tightly cooperate with the system engineer and this holds also the other way around. Such a teamwork often bears inherent problems and misunderstandings mainly due to inefficient communication since they work on different domains each having different aspects to take care of. It would be of advantage if the gap between those two domains could be narrowed by utilizing the same environment, the same "language" for both sides. Or, in a larger extent, both sides are merged together, where the system architect and the chip designer even become the same person.

The vision is to focus more on an abstract view of a complex problem and on fine-tuning of algorithms rather than spending most of the time on mapping the system to a hardware description. In the end the engineer should be able to translate distinct elements from a system directly to a target hardware, either an ASIC or a FPGA, by a simple mouse-click.

5.1 Simulink Simulation Model

The *Simulink* software from The Mathworks is a visual Model-Based simulation and design tool that is already well established in the field of communication and signal processing. Due to its intuitive block oriented design it is predestined for such applications and eases the understanding of complex systems. As a part of the Simulink Toolbox the *Simulink HDL Coder* extends the capabilities even further by allowing for an **automated** generation of hardware description language (HDL) code from the Simulink model. In that way an intertwined tool-chain is established that realizes all required steps from the top-level design to a hardware description in a unified manner. Even the step of synthesis can be incorporated from third party vendors as well as automated test benches and simulation scripts for external hardware simulation tools. It is worthwhile noticing here that actually all of the previously envisioned proposals, in the context of the flawed design flow, can be satisfied and will be explained in more detail.

The subsequent sections will incrementally try to deepen the understanding of the chosen design approach and reveal all the benefits and drawbacks that have been discovered so far.

5.1.1 Model-Based Design

At first, a basic system simulation model in Simulink (covered in this section) has to be developed that reflects the CertLink OFDM PHY layer. Based on the MATLAB simulation, which is already structured in a modular way as seen in Fig. 4.15, the task is essentially a mapping to Simulink blocks. The difference now being that Simulink introduces a fundamental simulation time which allows for dynamic system modeling, thus representing a more realistic environment. This time dependency is expressed as a "sample time" parameter, which has nothing to do with the common engineering term of sampling a continuous analog signal. It rather specifies the rate of execution of a single block. Notice that the terminology holds for discrete-time, continuous-time or even hybrid systems. Some blocks can be parametrized by an explicit sample time where others define a implicit sample time that can not be changed and is

set automatically through inheritance. Typically, in a digital communication system the data generation block in the transmitter specifies a global, system-wide sample time which will then be inherited by all other components. The sample time for the CertLink modem is currently defined such that time duration of one OFDM frame exactly matches the requirements of the OFDM symbol-rate ($10500^{Symbols}/_{s}$).

In order to have a fully configurable system, a way of defining a range of system parameters must be established. There are different approaches to accomplish this. One of them would be to set parameters in different callback functions in the model properties dialog. Another way is to define a masked system parameter block which allows for a comfortable adaptation of the parameters via a graphical user-interface (GUI), see Fig. 5.1. Some parameters in the figure are gray, indicating that they are currently not cleared for user modification. This block interacts with a MATLAB script that is executed in the background where every parameter is specified. All parameters are collected and encapsulated in a *single* structure, called `sysParam`, which can be easily accessed by every system block via the point-operator.

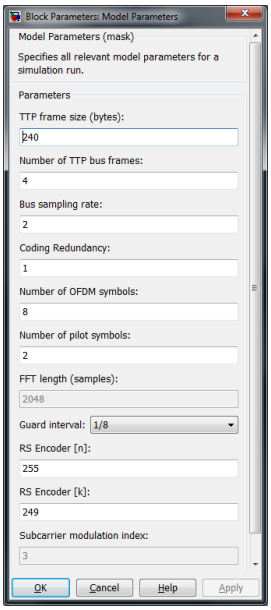


Figure 5.1: Graphical user interface containing all relevant system parameters for the Simulink model

Given that all parameters are available, the overall system can now be pictured in Fig. 5.2. This model is consistent with typical digital communication systems consisting of the transceiver, a channel model and the receiver, also matching the first simulation approach in Section 4.1. Here, a end-to-end physical layer Simulink model for the CertLink modem is presented, incorporating every part of the signal chain from transmitter, channel to the receiver. Notice the System Parameters block (on top) which allows for a comfortable graphical configuration of the entire system. The binary data streams are compared for error-rate calculation and are displayed during simulation time for the system operating under perfect channel conditions. This time, the channel model consists of a parametrized multipath fading channel together with a standard AWGN channel using the sophisticated Simulink blocks from the Communication Toolbox.

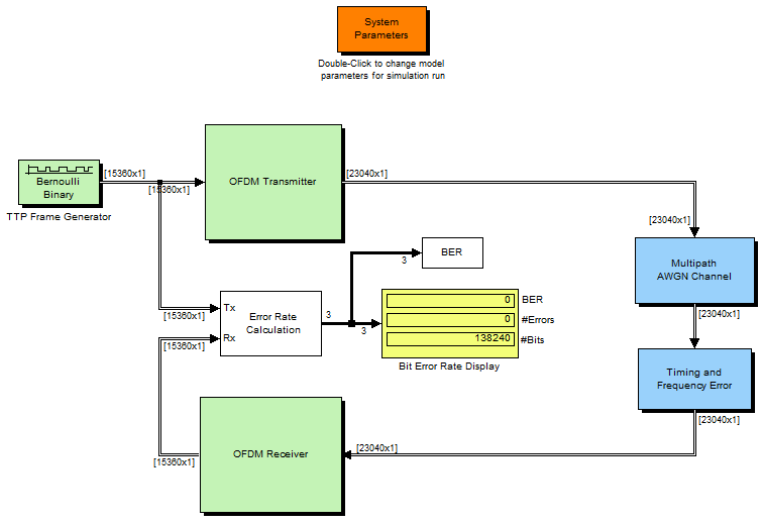


Figure 5.2: End-to-end PHY Simulink model for CertLink modem

The Simulink model apparently provides an impressive amount of abstraction. The diagram could also be misinterpreted as a purely visual representation of the system, e.g. for documentation purposes, but this model actually represents an executable fully configurable specification of the entire system. Actually it shows the first iteration of the modeling process towards a feasible hardware implementation. Besides that, Fig. 5.2 already illustrates how the system can be reduced to the most important blocks. Taking this diagram as a reference all sub-blocks can be progressively designed in more detail.

The Simulink software features the so called ”frame-based “processing in extension to a sample-based processing. Frame-based processing means, similar to signal processing in MATLAB, that the actual data (bits or integers) is propagated via vectors or matrices through the signal chain rather than sample by sample. This is a common way to represent data in real-time systems where high-rate samples of data are buffered to a block, or frame of say N samples which can be further processed at a lower rate equal to $1/N$. To put this into perspective: imagine an analog-to-digital converter which usually produces a datastream at a very fast rate. The digital signal processing blocks behind the ADC can not economically operate at such high rates and therefore need to use some kind of buffering beforehand. Simulink tries to reflect those characteristics in the modeling environment by providing such frame-based signals. Another benefit arises when frame-based signal types are used in real-time systems: as seen in Fig. 5.3 a sample-based operation requires an interrupt service routine (ISR) after each acquired sample to read the data from the hardware. By accumulating many samples during the data acquisition the throughput of the overall system can be increased, despite the fact that the frame-based operation introduces a certain amount of latency. The graphic essentially shows the comparison of the throughput rate between both signal types revealing the benefits of frame-based processing

Assigning aforementioned benefits to the Model-Based simulation, a *speed-up* in terms of simulation time can be observed because of the reduced block-to-block communication overhead. For more information referring to frame-based processing and the associated implementation issues, see [38].

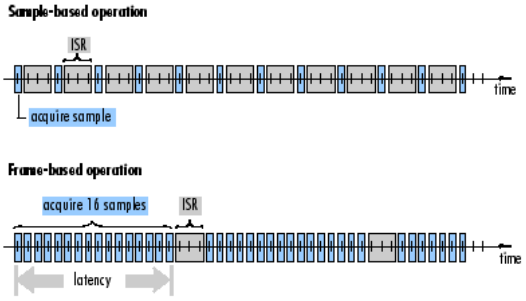


Figure 5.3: Sample-based versus frame-based operation in Simulink for real-time applications, from [38]

The TTP Frame Generator block in Fig. 5.2 is configured for frame-based processing and uses a Bernoulli binary generator, whereupon the simulation is entirely based on raw bits rather than integers. Similar to MATLAB, one frame is composed of 15730 *bits* (see Section 4.2) - as the amount is depended on the system parameters. Besides the frame size, also the related sample time must be specified. During the propagation through the different transmitter blocks additional samples will be appended to the frame (by FEC, OFDM mapping, etc.). So to match the required symbol rate SR on the channel, the sample time of the frame generator already needs to facilitate the correct update rate, i.e. the frame time has to fit the final frame size of 23040 *bits* although only 15730 *bits* will be generated. Notice that in the context of frame-based processing the sample time of the entire frame, here denoted as "frame time" and the actual time per sample ("sample time") have to be distinguished. Through buffering the sample time for the frame is reduced (slowed down) by the amount of samples per frame. So to put this into a mathematical perspective the requirement is to have a frame time of $\frac{1}{SR} = 952.38\mu s$ which is divided by the samples per frame (at the generator) to give a sample time of $\frac{952.38\mu s}{15730} = 62.004\mu s$. The frame duration is defined to be *constant* over the whole simulation time, independent of how many samples are appended.

Frames are generated one after the other, realizing a stream of data rather than explicit bursts. This does not truly reflect the real-world TTP scenario, where bursts of frames are expected to occur, but the techniques for receiving the data remains the same. However this way of modeling represents the worst case in which frames are received at the fastest rate possible imposing the most probable appearance of inter-block or inter-symbol interference. So if the receiver can handle the worst case performance-wise, all other cases will also work fine.

5.1.2 Transmitter

Fig. 5.4 shows the Simulink model for the OFDM Transmitter containing the essential blocks for signal processing and additional blocks for a graphical representation and measurements. The RS-Encoder and the D8PSK modulator are taken from the Communication Toolbox to speed up the design process. Reshaping prepares the data for the following OFDM mapping which is implemented as a "Embedded MATLAB" block which directly encapsulates the functionality as MATLAB code in a single Simulink block. A subcarrier mapping for a iFFT/FFT length of 2048 would otherwise be very hard to realize as an aggregation of atomic Simulink blocks. The embedded MATLAB code differs only slightly from a "ordinary" MATLAB function. Some issues regarding the access of vectors, data initialization and data types need more attention than usual.

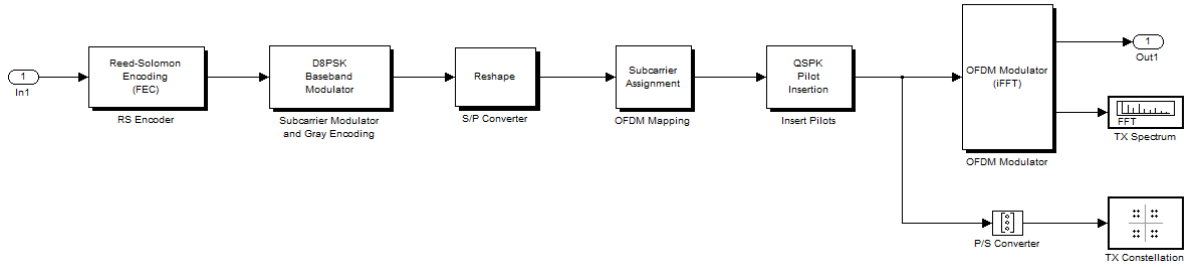


Figure 5.4: Simulink model for OFDM Transmitter

Some details of the pilot insertion and the OFDM modulator are presented in Fig. 5.5. First of all mind that *no* on-line generation of different pilot symbols is currently available. The reason being that pilot generation is directly coupled to the employed synchronization algorithm and needs special attention and additional effort with respect to its Model-Based implementation. Since the frequency domain pilot data (in this case QPSK modulated) is readily available from the MATLAB simulation it is appended to the datastream in a straightforward manner as depicted in Fig. 5.5a.

A look under the OFDM Modulator block reveals the subsystem as shown in Fig. 5.5b. The Simulink IFFT system block realizes the transfer from a frequency description to a time domain signal. After normalizing the output the cyclic prefix is inserted whereas a “Selector” block conveniently implements the insertion only by specifying a vector of indices where the guard needs to be placed for each symbol. The corresponding samples at the end of the symbol will be automatically copied on the correct position, therefore extending the signal size. In order to compute the spectrum of the transmitted signal the pilot data has to be extracted before computation. Now that the OFDM modulation process is complete, the signal data can be propagated through the transmission channel to the OFDM receiver.

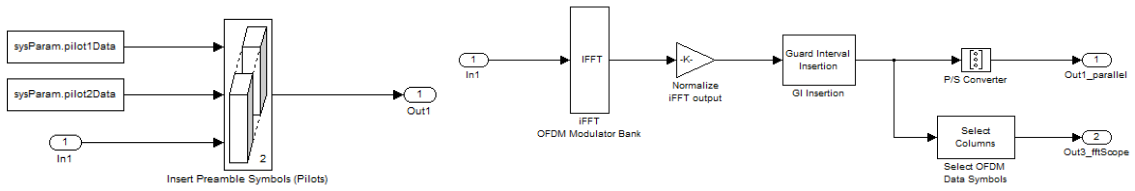


Figure 5.5: Sub-blocks of the OFDM Transmitter

5.1.3 Channel

Besides a standard AWGN channel, the Simulink channel model features also a Ricean fading model which was adapted to agree with the proposed aeronautical channel in Section 2.3. Because Simulink is intended for a dynamic system modeling the incorporation of time varying channels is more reasonable than for a MATLAB based simulation. Now actual movements of the receiver relative to the transmitter or vice versa can be simulated using applicable Doppler models. Additionally, the simulation of a line-of-sight multipath propagation model is available.

Simulink already implements fully configurable and technical mature system blocks for a digital communication channel that just need to be parametrized and incorporated into the design.

The high-level channel block is masked to provide full access to all of the underlying parameters of the subsystem pictured in Fig. 5.6. The three possible options for the behavior of the channel, selected by the "fadingMode" parameter, are the following:

No fading: Only the AWGN channel model is active. The signal is contaminated with additive white Gaussian noise, depending on the predestined E_b/N_0 parameter.

Flat fading: The channel defines a Gaussian Doppler spectrum having a maximum Doppler frequency f_D of 200 Hz.

Dispersive fading: Combines all channel models and adds an additional Rayleigh propagation path to the line-of-sight component to form a Ricean channel model.

Keep in mind that the AWGN is always active and that the channel configuration can be changed *on-the-fly*, i.e. during simulation time. All parameters are based on the discussed system architecture and the corresponding requirements in 4.1. The three options presented above will now be evaluated in more detail:

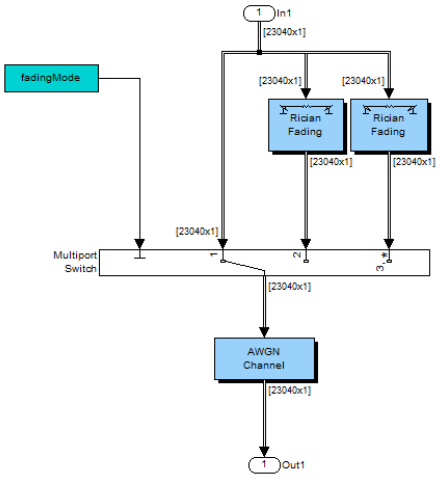


Figure 5.6: Baseband transmission channel for the end-to-end system

AWGN Channel: The AWGN channel essentially implements the same model as described in 4.3.1. As the Simulink model is only concerned about end-to-end link performance measures the additive noise must chosen appropriately in order to provide reasonable results. The model presented in Fig. 5.7 is built around the Simulink AWGN block that requires the noise variance as one of its input parameters. To compute this variance the signal power and the SNR per bit (E_b/N_0) must be available. Following the signal chain, the calculation of the signal power only addresses the proper data OFDM subcarriers (normalization) and uses the linear E_b/N_0 (40dB in this case with respect to FEC) to give the correct noise variance. The displays only provide informal visual aid to the user.

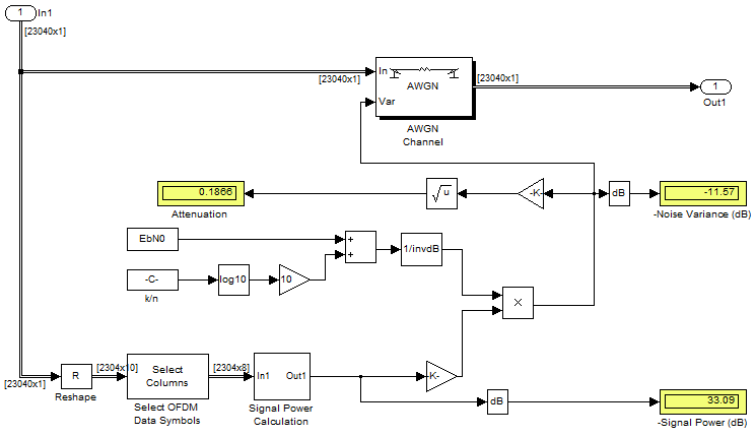


Figure 5.7: Detailed view of AWGN channel model

Fading Channel: Two different models can be chosen to simulate a more realistic aeronautical channel having additional properties concerning multipath propagation and Doppler effects. To account for a relative movement of the receiver to the transmitter the first Ricean model introduces only a single line-of-sight path having a Gaussian Doppler spectrum type. This Gaussian Doppler power spectrum is characterized by a standard deviation parameter that is normalized to the maximum Doppler frequency f_D and evaluates to variance $\sigma_{G,norm} = \frac{\sigma_G}{f_D} = \frac{1}{\sqrt{2}}$. In this way the Gaussian Doppler spread equals that of the Jakes Doppler spectrum. The single LOS path component is configured to have zero time delay and a 0dB gain. In this case the channel is said to be frequency flat, or frequency non-selective since the parameter variations are affected in the same way over the entire signal bandwidth. The direct path component is set to have a maximum Doppler frequency of $f_{D,max} = 200$Hz for an aircraft speed of $25^{\mathrm{m}}/\mathrm{s}$, and a Ricean factor of $K = 10$dB (which is also equal to 10 for a linear scale) as required for the Simulink model. Refer to 3.3.2 for the parameter decisions.

The second option considers a two-ray Ricean and therefore frequency selective fading model. In addition to the already mentioned Doppler effects in the first model, a second diffuse Rayleigh path with a time delay of 100ns and a relative power of -12dB is added to the dominant path. The overall path gain is normalized to 0dB.

Despite the current configuration of the aeronautic channel model, the parameters can be changed during simulation to adapt to alterable requirements.

Timing and Frequency Errors: To test the implemented synchronization algorithm a timing and a frequency error must be defined in the Model-Based simulation environment. Similar to the multipath channel model several options are available to chose from. Fig. 5.8 gives an impression how these different impairments can be modeled. Four different options for error addition are provided and selectable by the “modeSwitch” parameter. One option is to directly feed through the complex data, therefore adding no distortions. The other options represent every other permutations of a timing and a frequency error, i.e. either a timing error or a frequency error or both are active at the same time. Timing errors are simple delays to model different arrival times of the transmitter OFDM signal - compared to the MATLAB simulation, also fractional time delays are allowed Simulink with the help of a Fractional Delay block.

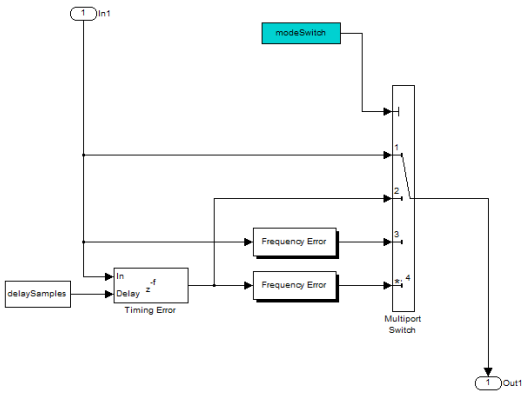


Figure 5.8: Model for additional channel impairments such as timing and frequency error for the OFDM system simulation

The frequency error models differences in the transmitter and receiver clocks and is specified in multiples of the subcarrier spacing. By multiplying the time domain data stream with the appropriate complex exponential the channel introduces a frequency shift. See also 4.3.2.1 and especially remember (4.8) for a better understanding.

Again, all parameters are defined in a GUI and can be altered on-the-fly, simulating different channel impairments for the OFDM receiver.

5.1.4 Receiver

In the OFDM receiver all operations from the transmitter are reversed, as already discussed in Section 4.2. In Fig. 5.9 a *first* iteration of the OFDM receiver block diagram is shown as implemented in Simulink including the reverse functional blocks to the transmitter and providing the binary datastream at its output for error comparison. Notice that the block diagram is read from *right* to *left* in order to follow the high-level signal chain concept and to maintain the signal flow direction. Both blocks, Timing and Frequency Offset Estimation and Frequency Offset Compensation do not perform any manipulation on the datastream since they are part of the continuing iterations of the receiver model. They will be described in the upcoming sections.

This iterative process progressively shows the design flow of this Model-Based approach where different modules are refined based on a coarse abstract model until a feasible model for a hardware implementation is developed. Both modules are part of the OFDM synchronization process which is mandatory before OFDM demodulation (FFT).

Assuming a perfectly synchronized system the first step is to remove the guard interval from the time domain samples prior to applying the FFT. This removal process is integrated into the OFDM Demodulator block and is therefore not shown in the figure. All subsequent steps are well-known from Section 4.4 and will not be reiterated here. Notice that the OFDM demapping, where all data carriers are extracted, is also implemented in an Embedded MATLAB block for convenience (see 5.1.2). As before, all redundant subcarriers, placed on the second half of the two-sided spectrum are discarded again during the demodulation process. After D8PSK demodulation and Reed-Solomon decoding the bit stream can be used for error computations.

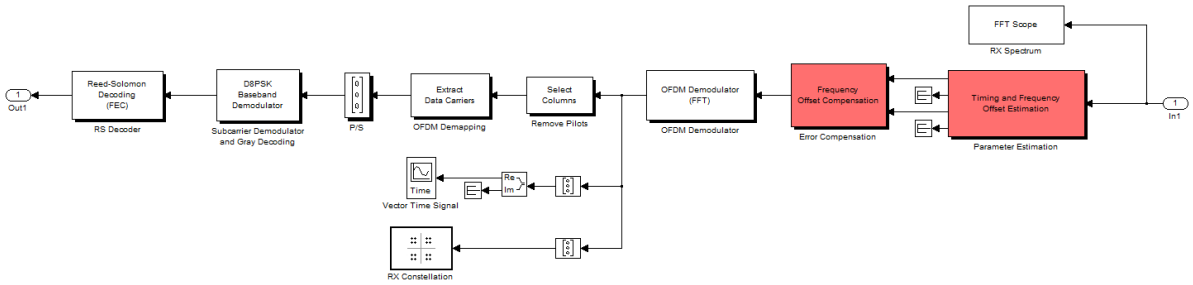


Figure 5.9: First iteration of the OFDM Receiver model in Simulink

5.1.5 BER Computation (bertool)

Briefly referring back to Fig. 5.2, the Error Rate Calculation block from Simulink allows the direct computation of bit errors and delivers all the important results at its output. A display visualizes the number of transmitted bits, the number of errors and the bit error ratio for direct feedback to the engineer. This block performs the computation on the basis of entire frames and provides options for delaying the calculation or to accommodate for delays in the receiver. Together with the `bertool` a powerful mechanism for link performance measures can be established. This tool enables the user to control the entire simulation process - it automatically can start and stop simulating the design, write and read parameter data and display the results in a separate figure. Measuring the link performance always considers the BER versus E_b/N_0, so it is evident that the tool must be able to influence the SNR per bit in some way. It does so, by using the global parameter `EbNo` that is applied to the "Multipath AWGN Channel" block. The system simulation is restarted for every value of `EbNo` and stops after a specified amount of errors or transmitted bits (stopping criteria). The computed BERs are logged and exported to the workspace.

Fig. 5.10 reveals the achieved performance of the OFDM system under using two different channel models with no additional interferences regarding time or frequency errors - therefore a perfect synchronization and demodulation in the receiver is supposed. One model considers a pure AWGN channel and the other defines the previously discussed dispersive two-ray Ricean model for K=10. Under AWGN the system's performance matches the theoretical BER when using RS encoding, with only a slight deviation at higher E_b/N_0 due to a reduced number of parameter values used. For the Ricean case an improved BER (steeper slope) can be observed when compared to the theoretical curve - the (coded) OFDM system proves its robustness against multipath fading.

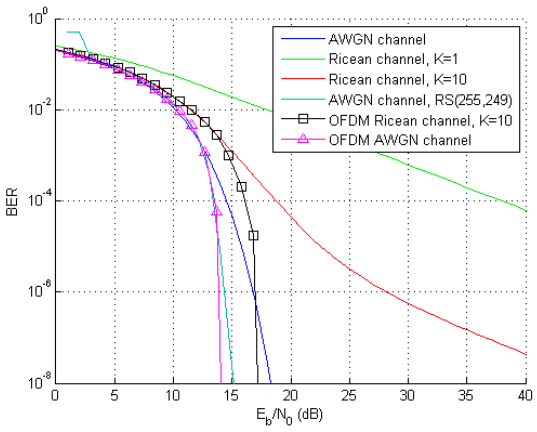


Figure 5.10: BER for the Simulink model assuming perfect synchronization in the receiver

5.1.6 Visualizing Channel Effects

To visualize the channel effects in the model, either the spectral analysis of the time signal or scatter plots (constellation diagrams) in the frequency domain can be used. These powerful measuring tools are continuously updated during the simulation time and thus provide insight to the influence on the channel impairments.

The Spectrum Scope is one of the tools used to visualize the frequency characteristics of the OFDM signal. In this case a scope having a FFT length of 2048 points is used (Hann windowing, two spectral averages) for displaying the two-sided spectrum. It is worthwhile noticing here that it must be ensured to used the correct sample time of the time series to get the scaling right. The signal is presented to the scope as frame-based, so the sample time is actually expressed as a frame time. In this case the scope utilizes the original sample time[2] to analyze the spectral characteristics. Fig. 5.11 shows the behavior of the OFDM system in an AWGN channel for the transmitter and the receiver respectively. The spectra in the figure are essentially an spectral overlay of all data symbols each containing 2304 samples (the guard interval is not removed here). On the receiver side in Fig. 5.11b the effects of changing E_b/N_0 configurations can be viewed, where the noise floor is adapted correspondingly (in this case $E_b/N_0 = 20$dB). At this configuration the system occupies about 16MHz of the spectrum since not all carriers are assigned to data symbols.

To extend the view to other than an AWGN channel, Fig. 5.12 illustrates the spectral conditions on the receiver side for frequency selective Ricean channels. The spectral shape is directly related to the channel transfer function resulting in more or less fading depending on the amount of signal paths. Fig. 5.12a presents the influence of the already mentioned two-ray Ricean channel whereas Fig. 5.12b extends the number of echo paths to four. Those four path exhibit decaying path gains (-6dB, -8dB,-10dB,-12dB) in addition to the LOS component, whereas the overall path gain is normalized to 0dB. The path delays increase linearly from 100µs to 400µs while still maintaining a K factor of 10. Both channels are examined at $E_b/N_0 = 40$dB.

2 Actually in this case the calculated OFDM *sampling* time T_s is used because the spectral analysis is related to the sampling process.

The simple two-ray Ricean channel shows only minor distortions whereas a five-ray multipath propagation already reveals severe distortions due to the large variation in attenuation of the channel's frequency response, typical for frequency selective fading.

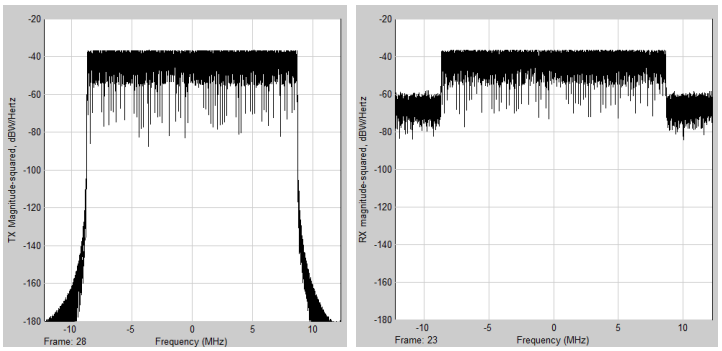


Figure 5.11: Transmitter and Receiver spectra of the OFDM system model operating in an AWGN channel

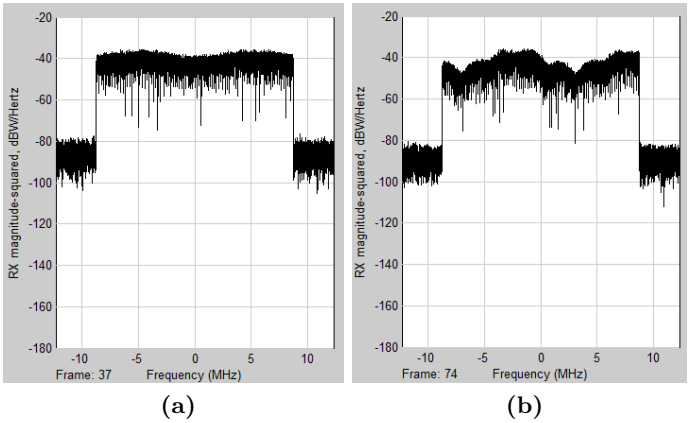
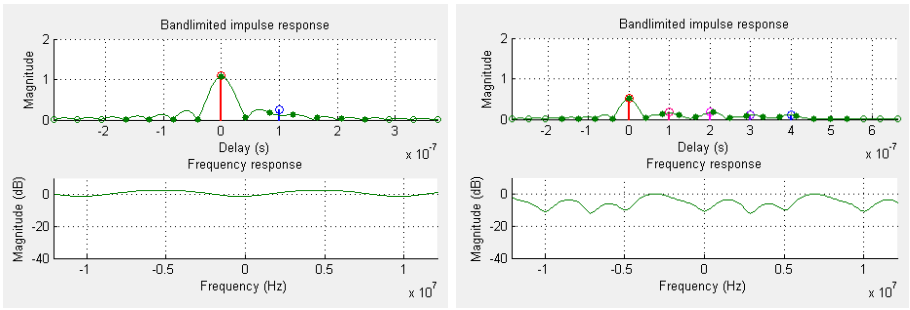


Figure 5.12: OFDM receiver spectra for the system model operating under different frequency selective fading channels

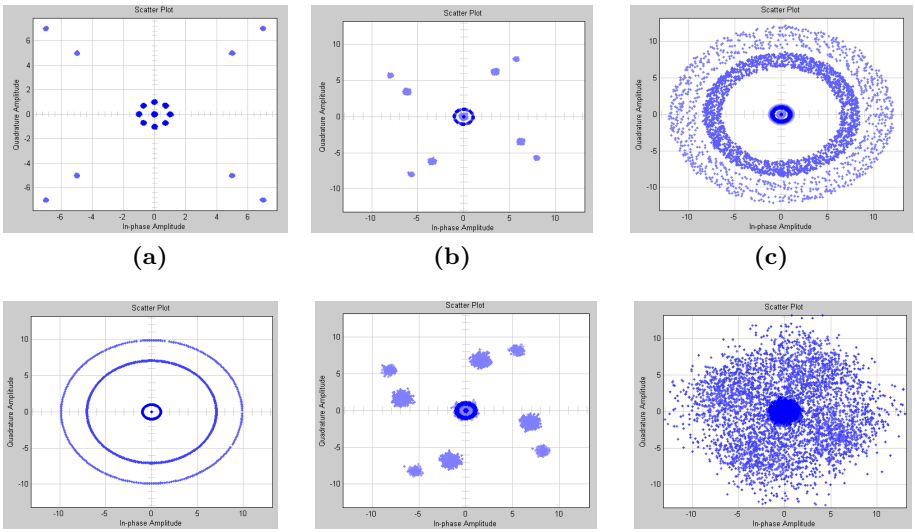
Fig. 5.13 provides some additional visual interpretation of the channels under discussion where the impulse and the frequency response is pictured for either channel in detail. The two ray channel in Fig. 5.13a shows only small changes in the frequency dependent transfer function wheres the deeper fades occur when the amount of arriving scattered paths is increased to four, as demonstrated in Fig. 5.13b. Both figures are conveniently generated by the Simulink Ricean Fading block and highlight only one of many other options for displaying the channel characteristics.



(a) **(b)**

Figure 5.13: Details of the impulse and frequency responses of the two Ricean channel configurations

Other channel effects are better understood if they are investigated in terms of scatter plots. Fig. 5.14 reveals a selection of different effects due to fading as well as to timing and frequency errors. Different effects on the constellation diagrams at the receiver are illustrated after applying the FFT for OFDM demodulation.



(a) **(b)** **(c)**

(d) **(e)** **(f)**

Figure 5.14: Different effects on the constellation diagrams at the receiver after applying the FFT for OFDM demodulation

If the signal at the receiver experiences no distortions except from additional white Gaussian noise, the constellation diagram of the FFT output behaves as depicted in Fig. 5.14a. The plot illustrates the results of perfect demodulation under perfect channel conditions apart from an AWGN channel at $E_b/N_0 = 40$dB. Essentially, this (and every other) diagram is a projection onto the time axis of all symbol constellations where each diagram is displayed for a different time instance. So in essence, the temporal evolution of the symbol constellation can be observed. The plot neatly presents the boosted QPSK modulated pilot symbols and the D8PSK modulated data symbols.

In Fig. 5.14b a flat fading, time selective channel imposes a Doppler shift to the received signal which produces variations in amplitude and phase (rotation of the constellation diagram) depending on the actual Doppler frequency. Here shown at a maximum Doppler frequency of $f_{D,max} = 200\text{Hz}$. In contrast, the frequency selective (two-ray) Ricean channel in Fig. 5.14c causes more distortions due to the secondary multipath components having a Rayleigh amplitude distribution. Changing the configuration of the Time and Frequency Error block also affects the FFT output as discussed in 4.3.2. The lower three plots demonstrate that the Simulink model is also capable of producing different errors impacting the demodulation process leading to a performance degradation and picture the linear phase rotations caused by timing offsets and the vulnerability of the OFDM system against frequency errors. Fig. 5.14d for instance, exhibits symbol rotation of the FD symbols due to a FFT window misalignment for a timing error of 20 samples. Note that a small frequency offset of 5% of the SC-spacing (600Hz), as seen in Fig. 5.14e still leads to a correct demodulation of the signal, while an offset ten times larger (0.5 SC-spacings) implicates fatal distortions (loss of orthogonality) and a degraded system performance.

The basic Simulink model is now considered complete. It is time to have closer look on how the intended design flow is supposed to work out. The following sections will present the ideas and how a specific synchronization is going to utilize this flow.

5.2 Design Flow - from Model to HDL

Based on the system model developed in Section 5.1 the next steps include the choice of one synchronization algorithm, the modeling of this algorithm and the task of preparing the model for a feasible hardware implementation.

Before deciding on the algorithm some basic statements about the intended design flow, especially the Simulink HDL Coder and its related precautionary measures, have to be made. The Simulink HDL Coder is able to automatically generate bit and cycle true HDL code (VHDL or Verilog), e.g. from Simulink models or MATLAB code. The generic HDL code generation process is independent of the target-hardware, thus enabling the system engineer to design and test an algorithm in hardware without worrying about specifics of the vendor. The following bullet list gives a short overview of the most important Simulink HDL Coder features (cited from [39]):

- *"Generation of target-independent, synthesizable HDL code from Simulink models, MATLAB code, and Stateflow charts*
- *Support for Mealy and Moore finite-state machines and control logic implementations*
- *Generation of test benches and EDA Simulator Link cosimulation models*
- *Resource sharing and subsystem-level retiming options for area-speed tradeoffs*
- *Simulink model optimization using timing constraint information and HDL synthesis tools*
- *Code-to-model and model-to-code traceability for DO-254*
- *Legacy code integration"*

The step of HDL code generation is the last one in a series of measures to actually be able to "click and generate". Starting from a theoretical description of the algorithm a Simulink model has to be established that reflects the basic functionality. This model or more precise this subsystem when speaking in terms of Simulink, can be developed by either using predefined blocks appropriate for HDL code generation or by using MATLAB statements encapsulated in an Embedded MATLAB block. The HDL coder only allows certain "HDL compatible" blocks and requires a slightly different way of writing translatable MATLAB code. All of compatible blocks are summarized in a separate library - so the choice is limited although a large collection is available. Along with ongoing new releases of the HDL coder this library is supposed to be continuously extended. For the Embedded MATLAB blocks, the coding style differs in its execution style since *persistent* variables are introduced. This enables the modeling of memories or system states that retain their content during different block calls. Usually, if all blocks are assembled to define the functionality of the model, a double precision version is designed in the first place, providing the maximum precision available for the system.

Since double precision or floating point operations in an FPGA are very costly in terms of resources, a fixed point implementation is more desirable. To achieve such a fixed-point implementation, Simulink provides tools like the Fixed-Point Advisor and the Fixed-Point Tool which both enhance the capabilities of designer by utilizing an assisted (automatic) selection of the word length, rounding modes, etc. If a reasonable implementation is assured (compatibility) the Simulink HDL Coder comes into play where it takes the fixed-point implementation and directly translates the model to a hardware description language code. Different options arise if the process of automatic code generation is complete. One being that also an additional model is generated based on the HDL - certain implementation options or results by the coder can be verified, e.g. delays, rounding, etc. Another option is to generate a testbed suitable for third-party hardware simulation tools. These testbeds include full HDL scripts and test cases to verify the design in a virtual hardware simulation environment. By using the EDA Simulator Link also cosimulation is possible. Here Simulink automatically generates an interface to an external hardware simulator like ModelSim and tests the hardware implementation against the fixed-point Simulink model.

Fig. 5.15 tries to visualize the basic steps of the described design flow. The diagram shows the major steps necessary to go from a theoretical description of an algorithm to a HDL description. Gray blocks indicate the helping tools between each procedure. With the ongoing development process details about certain aspects of the flow will be described more thoroughly.

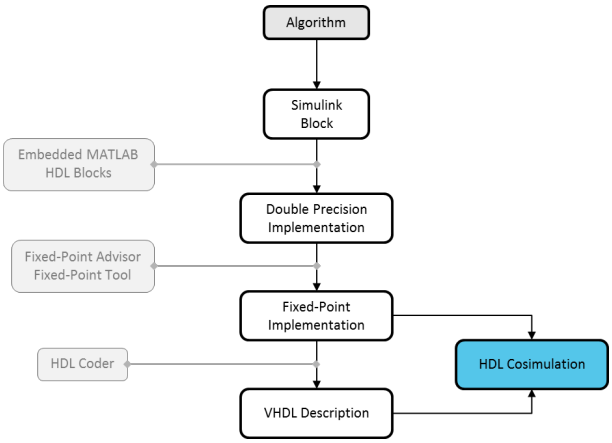


Figure 5.15: Simulink HDL Coder design flow

5.3 Implementation Details and Results

The subsequent section will elaborate how the actual implementation is done in hardware, firstly discussing the synchronization algorithm of choice and then following the design flow until a feasible hardware mapping is obtained. Moreover the adaptation of the overall system simulation, together with the issues of parameter estimation and the OFDM receiver design is described.

5.3.1 Synchronization Algorithm of Choice

Three estimation algorithms were investigated in Section 4.1 to evaluate their characteristics, performance and to gain more insight in the process of OFDM synchronization and the related issues when it comes to the implementation. Section 4.5 revealed the estimation performance of either method and came to the conclusion that the method after Park et. al. was superior over the other algorithms in terms of timing estimation analysis. However Park only proposes a way to estimate the timing offset while relying on a method similar to Schmidl's in order to estimate appearing frequency errors. This approach would include the need for implementing essentially two algorithms for two distinct tasks. Since it is currently not assessable what area requirements are needed for other receiver modules excluding the synchronization block the goal is to keep the implementation complexity low. If the receiver is finally completely implemented in hardware, trade-offs can be made concerning area and speed requirements. But since then, a strive for a low area implementation is more reasonable.

Because the method after Minn has a high computational complexity due to extensive averaging, the S&C algorithm is chosen for the OFDM synchronization in the CertLink receiver. Apart from the fact that S&C envision a joint estimation method of timing and frequency errors they also provide for supplemental information about strategies for frame detection and hardware implementation issues. In summary the S&C algorithm is preferred over the two methods due to:

- feasible low-area, low complexity implementation

- adequate for burst synchronization
- joint estimation of timing and frequency errors
- already in use by the industry (IEEE 802.11a standard)
- provides sophisticated method for frame detection
- additional information regarding implementation strategies available
- is considered the groundwork for all other subsequent synchronization method[3]

Now that the synchronization method is fixed, the algorithm must be described in terms of functional blocks for a Model-Based Design. Prior to that, some Simulink specific issues have to be addressed.

5.3.2 Refined Simulink Model

The HDL code generation for a Simulink subsystem is only possible if this subsystem is instanced in the top layer, i.e. it has to be defined on the upper most hierarchy level. Due to this constraint the Timing and Frequency Offset Estimation block, mentioned in 5.1.4 needs to be pulled out of the OFDM Receiver block representing sort of an "outer" receiver block. Furthermore the synchronization block will operate in a sample based fashion rather than frame based because of the underlying hardware complexity of parallel signal processing. Thus, the data from the channel is unbuffered and the sample time is increased correspondingly. Some additional modifications inside the OFDM receiver come along with this sample based operation which will be discussed in the course of the developing model.

Fig. 5.16 illustrates a refined Simulink model (second iteration) taking into account all the imposed constraints for automatic code generation. The DAC block on the transmitter side is a placeholder for future design iteration whereas the ADC channel covers the unbuffering of the frame based data from the baseband channel. The Parameter Estimation block contains the actual algorithm and provides three outputs (start of frame, timing offset, frequency offset) where two (start of frame, frequency offset) are actually used in the OFDM receiver. Since the computation in the estimation block introduces a certain delay the frame-based data stream has to be aligned (delayed) accordingly to ensure the correct timings whereas additional displays are used for presenting the estimation results. Other than that the top-level Simulink model did not change in its structure.

3 Scientific papers authored by Schmidl and Cox are the most cited in the domain of OFDM synchronization, see IEEE Xplore http://ieeexplore.ieee.org

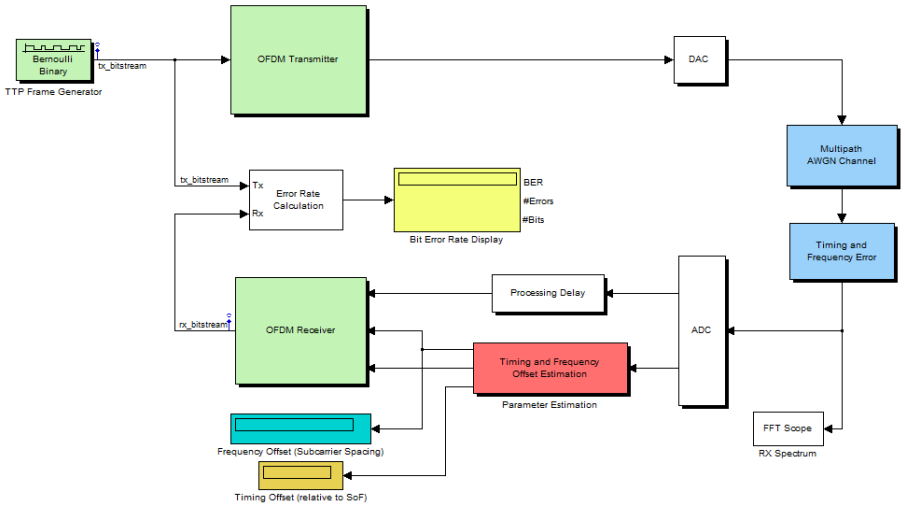


Figure 5.16: Second iteration of the Simulink model

5.3.3 Parameter Estimation

S&C's algorithm has been treated in detail in 4.4.1.1 for a script based simulation environment. The correlation metrics were calculated by using efficient vector notation and operation hence speeding up the simulation time. In a Simulink environment a more realistic, sample-by-sample based operation has to be established, thus requiring a different implementation approach adjusted for an architectural hardware description. In essence, as reiterated here, two complex sliding window autocorrelations have to be calculated based on a recursive computation. An analysis of the basic correlation equations show how such a recursive computation scheme evolves. The correlation P is defined as

$$P[d] = \sum_{m=0}^{L-1} r[m+d]^* r[m+d+L],$$

where $()^*$ denotes the complex conjugate. The equation can be expanded to

$$P[d] = \underbrace{r[d]^* r[d+L]}_{m=0} +$$
$$+ \boxed{\underbrace{r[d+1]^* r[d+1+L]}_{m=1} + \ldots + \underbrace{r[d+L-2]^* r[d+2L-2]}_{m=L-2} + \underbrace{r[d+L-1]^* r[d+2L-1]}_{m=L-1}},$$

representing a window of length L performing the complex autocorrelation of samples between one half symbol and the delayed version. Iterating the formula for one time step leads to

$$P[d+1] = \boxed{\underbrace{r[d+1]^* r[d+1+L]}_{m=0} + \underbrace{r[d+2]^* r[d+2+L]}_{m=1} + \ldots + \underbrace{r[d+L-2+1]^* r[d+2L-2+1]}_{m=L-2}} + \\ + \underbrace{r[d+L]^* r[d+2L]}_{m=L-1},$$

where the boxes indicate the common terms (intersecting set) between the two iteration steps. Taking into account these common terms leads to a simplified recursive definition of the correlation

$$\Rightarrow P[d+1] = P[d] - r[d]^* r[d+L] + r[d+L]^* r[d+2L]. \tag{5.1}$$

Rewriting (5.1) in a causal notation by shifting the sequence appropriately, gives the final equation for a suitable hardware mapping

$$P[d] = P[d-1] - r[d-2L]^* r[d-L] + r[d-L]^* r[d]. \tag{5.2}$$

The autocorrelation for the second half symbol (normalizing power) can be developed in the same way, where the equation is denoted as

$$R[d] = R[d-1] - |r[d-L]|^2 + |r[d]|^2. \tag{5.3}$$

Equations (5.2) and (5.3) form the basis of the entire algorithm. A closer look on the formulas reveals a similar structure of these sliding correlators that can conveniently implemented by a cascaded-integrator-comb filter structure as seen later on.

One major issue was not addressed during the development of the script-based simulation in MATLAB, namely **frame detection**. The reason for this being that in the MATLAB case no time dependency was present and the samples were always perfectly aligned in a vector or a matrix making the simulation of frame detection mechanism obsolete. In contrast, a real-world implementation *must* detect when a frame starts in order to trigger the further processing and determining the actual timing and frequency offset. As discussed in theory (Section 2.2.4) Schmidl et. al proposed a method for a robust frame detection which is entirely based on the two correlation values. Using a suitable threshold decision, the start of frame (SOF) can be reliably detected, moreover eliminating the need for explicitly computing the metric M, thus saving a costly division operation. The following list gives an impression on all the required steps for estimating and correcting timing and frequency errors of the S&C algorithm while focusing also on hardware implementation issues. It should present an overview or a guideline how the algorithm *could* be implemented, inspired by [22]. Based on the list the actual implementation steps can be discussed in more detail.

1. Compute the autocorrelation P[d] and the power estimate autocorrelation $R[d]$ iteratively. $R[d]$ could be used for an automatic gain control (AGC) for receiver power-up
2. Check for SOF, monitor $|P[d]|^2 > threshold \cdot (R[d])^2$ instead of $M[d] > threshold$ and detect SOF, which can be interpreted as a coarse timing estimate $\hat{\varepsilon}_c$ at the sample index $d_{opt,c}$, store threshold value in ROM or use an adaptive threshold

3. Compute $argmax(|P[d]|^2)$ whilst ignoring $(R[d])^2$ to give the fine timing estimate $\hat{\varepsilon}_{d_{opt}}$ at the sample time d_{opt} within the IBI-free interval (plateau)
4. Compute the phase $\phi = angle(P[d_{opt}])$ as the angle of the complex correlation value at the estimated timing index, use an efficient CORDIC based implementation to calculate the angle iteratively
5. Calculate the fractional frequency offset $\hat{\theta}_f = \frac{\phi}{\pi}$, store $\frac{1}{\pi}$ in a ROM with high precision
6. Correct the data samples by $\hat{\theta}_f$ using the complex exponential $e^{\frac{-j2\pi\hat{\theta}_f n}{N}}$, store $\frac{2\pi\hat{\theta}_f}{N} = \frac{\pi\hat{\theta}_f}{L}$ as a high precision constant and use a CORDIC iterative approximation for the trigonometric function
7. Align the FFT window according to the estimated timing offset $\hat{\varepsilon}_{d_{opt}}$
8. Compute the FFT of corrected pilot symbols $x_1[n] \multimap X_1[k]$, $x_2[n] \multimap X_2[k]$
9. Calculate $B[g]$ while reducing computational complexity by computing the normalization factor only once, use the a-priory known differentially modulated sequence v_k stored in ROM
10. Determine the integer frequency offset based on $g_{corr} = \underset{\tilde{g}}{argmax}(B[\tilde{g}])$
11. Shift partly corrected time domain samples by $\theta_i = 2g_{corr}$ subcarrier spacings before the FFT computation to correct for the integer frequency offset
12. Perform OFDM Demodulation on the corrected data samples using the FFT

The complete implementation of the algorithm would incorporate fractional and integer frequency estimation and correction where the integer part is calculated in the frequency domain. To save implementation space and reduce the complexity the procedure of estimating and correcting for the integer frequency offset is *ignored*, i.e. only offset in the range of ±1 SC-spacings (±12kHz) can be corrected. This range is sufficient if the oscillator accuracy is assumed to lie withing this frequency range. The assumption is reasonable because the quartz stability is defined to be $\pm 2ppm$ of the carrier frequency (see Chapter 3) which in turn resolves to a maximum of $\pm 4ppm$ (9600Hz) inaccuracy between transmitter and receiver[4]. Based on this decision, the steps 8. -11. can be skipped, meaning that no FD data is used for the algorithm, therefore reducing additional data communication overhead.

Fig. 5.17 illustrates the block diagram of the S&C algorithm where the mathematical description is mapped to a hardware description. Beginning from the right (intuitive view inherited from top-layer) the complex baseband data is divided into two branches. One for computing $P[d]$ and the other for $R[d]$ where each branch implements the recursive equations (5.2) and (5.3). The "R-branch" uses the absolute values of the complex input for the multiplication whereupon the "P-path" requires the delayed complex conjugated data along with the unchanged input. The CIC filter structure is the same for both signal paths consisting of a comb structure with an $L = N/2 = 1024$ samples delay and the integrator structure as shown in the Simulink model of Fig. 5.18. $|P[d]|^2$ and $(R[d])^2$ are computed for frame detection using the SOF threshold. The threshold value was determined empirically at different SNRs and channel

4 A headroom of 2400Hz is available to accommodate for Doppler frequency shifts.

conditions. A fixed threshold is only possible if the pilot symbol energy is boosted relative to the data symbols - this is the reason for using a higher energy symbol constellation for the preamble. Otherwise, the correlation metric would experience large amplitude variations for which an adaptive threshold would be necessary to adjust to different SNRs. The SOF Detection block decides if the threshold is reached and generates a single SOF pulse indicating that a data frame was detected in the received signal. This SOF pulse then riggers the *argmax* block which executes several tasks.

As the block's name already suggests, the argument (discrete time index) of a maximum correlation value, depending on the SOF, is computed. Beginning at the trigger the samples of $|P[d]|^2$ and $P[d]$ are sequentially stored in two separate buffers, each consisting of 512 entries. The buffer length is consciously chosen to make sure that all samples around the metric plateau are stored for computing the timing offset. Remember that the length of the plateau is equal to the guard interval length (256 samples) at maximum, so by choosing this buffer length the entire plateau samples around the optimal timing point should be included. Alongside, a maximum search determines the *global* maximum $|P[d_{opt}]|^2$ of the correlation. Normally, the best timing point is estimated based on the metric M but since the unnormalized autocorrelation does not vary much within the observed range, $(R[d])^2$ can be discarded. If the buffer is full the index of the global maximum is taken as the best timing point whereas the maximum value of P is used for the phase and fractional frequency offset computation. Notice that the buffer could also be used for the 90% averaging method but to keep the complexity to a minimum the straightforward implementation is employed here. While the estimated fractional frequency offset can be directly utilized by other receiver blocks, the timing estimate has to undergo additional processing. It is a prerequisite to translate this timing (integer) value into a simulation time related information, that somehow aligns the FFT computation window. To do so, the estimate is used as a preload value for a downward counter. Depending on the preload, the counter is triggered to start counting downwards at a constant slope. If an underflow occurs a unit pulse is generated for FFT control.

SOF Detection, *argmax* and the Counter blocks are comfortably implemented as Embedded MATLAB blocks also feasible for HDL code generation. It is worthwhile noticing here that in this refined Simulink model *not every* block is compatible to the HDL coder.

In order to compensate for the entire processing in the Parameter Estimation block a processing delay is introduced that delays the data samples accounting for the correlation over one full symbol (2048 samples) and the fixed buffer length (storing delay 512 samples). The frequency estimate, denoted as `theta_f` and the control signal for the FFT `FFT_start` are propagated to the actual OFDM receiver part.

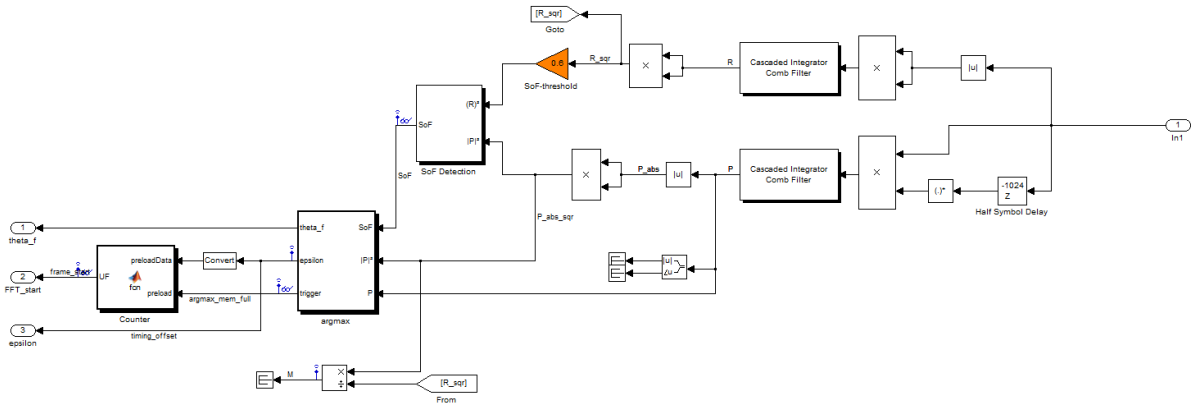


Figure 5.17: Block diagram for S&C estimation algorithm

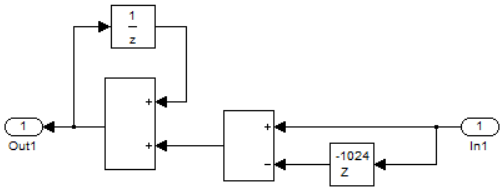


Figure 5.18: Simulink model for Cascaded Integrator Comb filter structure to perform the data correlation

5.3.4 OFDM Receiver

The OFDM Receiver structure is almost the same as in the first model concept except from the parameter estimation which moved to the top-level. The additional and adapted simulation concepts as well as newly designed system blocks are discussed in the following sections.

5.3.4.1 Frequency Error Correction

To correct the frequency offset the data is processed in the Error Compensation block shown in Fig. 5.19. The incoming data is corrected by fractional frequency error θ_f by using a complex exponential multiplication factor linearly distributed over all subcarriers. This block generates a linear ramp for the correcting complex phase factor depending on start of frame and limited by the number of samples per frame. The Linear Ramp block is essentially a up-counter with a synchronous enable input and a limit-detection that accounts for every sample n within the frame to construct the exponential $e^{\frac{-j2\pi\hat{\theta}_f n}{N}}$, where $\frac{2\pi}{N}$ is separately stored.

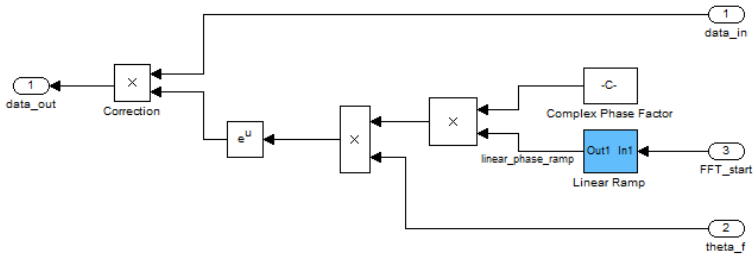


Figure 5.19: Model for frequency error compensation in Simulink

5.3.4.2 Streaming FFT

Data samples from the Error Correction block are then sent to the FFT for OFDM demodulation. The original Simulink model employed a frame-based system FFT block that could process frame after frame providing no control over the block. In a sample-based scenario this approach would not work since the timing estimate defines when to demodulate the data. Therefore, a sample-based HDL streaming FFT block is implemented that takes a start or control signal and processes blocks of samples either in a continuous or non-continuous manner. Remember that prior to applying the FFT the guard interval must be properly removed. Because the estimation block only provides *one* start pulse for the first OFDM symbol additional FFT start pulse have to be generated for each subsequent OFDM symbol. The FFT Frame Control block employs an up-counter with synchronous enable and a start pulse detection to "cut out" the cyclic prefix for every symbol by adjusting the FFT pulses correctly. If the FFT bank receives a valid pulse the incoming complex data samples will be transferred into the frequency domain inhibiting the output for a constant processing delay. Therefore the non-continuous operation mode is employed from the HDL FFT block. For each pulse exactly 2048 FD samples (corresponding to the FFT length) are made available at the FFT output.

5.3.4.3 FFT Buffer

Based on the fact that this FFT operates in sample-based fashion and other parts of the OFDM receiver still maintain a frame-based operation[5], a reasonable data alignment has to be established. Before aligning the samples to a frame the FFT output needs to be buffered. The reason for this being that the cut out samples corresponding to the guard interval lead to a zero data output at the FFT, hence inducing unwanted "data leaks". In order to concatenate the data again the `FFTBuffer` block is introduced. In essence, this block delays (displayed on block) and compresses the data by utilizing the data valid signal from the FFT and a single buffer. Delaying the data is accomplished by different write and read pointers where the offset inbetween is equal to the intended delay. This delay comprises all the "data gaps" from nine OFDM symbols plus additional samples to make sure that the read and write pointers do not overtake each other. The buffer length covers a full frame initialized with complex zeroes. As valid data arrives the module starts continuously reading from the buffer while the write

5 Notice that, e.g. the OFDM demapper is dependent on frame-based data since it is coded to operate on vectors and matrices.

pointer is initialized to the desired offset and starts filling the buffer from there. The block keeps track of the number of processed OFDM symbols and defines a `frame_out_valid` signal that decides whether the buffer output is valid or not. As soon as the predefined maximum number of symbols is reached, the block resets into its initialized state and waits for the next valid frame to arrive. Ideally, 20480 samples (ten OFDM symbols) are contained within a valid frame. The FFT buffer output is still sample-based, so to convert these samples to a frame-based vector, a Simulink Buffer block reduces the sample rate by 20480 and outputs a vector signal. The Buffer block is embedded in an enabled subsystem controlled by the `frame_out_valid` signal to build a frame, only consisting of *valid* data. Because all frame-based signals have a frame time corresponding to 23040 samples a sample rate adjustment needs to correct the aligned data. After parallelization and frequency shift compensation the data is ready to be processed by the following parts in the receiver.

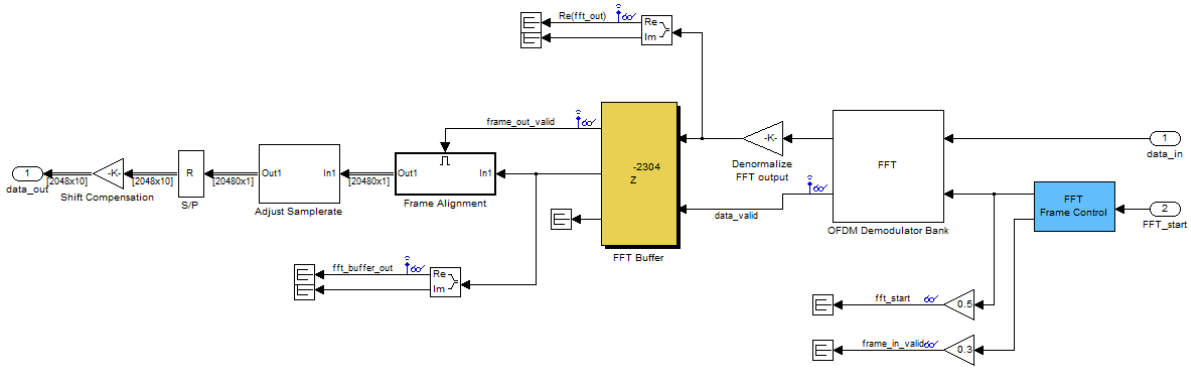


Figure 5.20: Detail of the OFDM Demodulator block

Fig. 5.20 shows the details of the OFDM Demodulator block including the streaming FFT implementation controlled by the FFT Frame Control block. The FFT output is buffered (FFT Buffer block) and aligned (Frame Alignment block) to the Simulink frame in order to process the data in a frame based manner. Notice that the FFT Buffer and the FFT Frame Control blocks are subsystems only containing Embedded MATLAB code.

5.3.5 Results

This section presents some of the intermediate results produced by the Simulink model in order to consolidate the explanations above. All simulations were performed under an AWGN channel ($E_b/N_0 = 40\text{dB}$) and a frequency error of $\theta = 0.9666$ subcarrier spacings.

5.3.5.1 Estimation Analysis

Fig. 5.21 depicts the time evolution of different signals appearing during the synchronization process utilizing the S&C algorithm. The complex baseband data (the figure only shows the in-phase part), as illustrated on the top of the figure, feeds the sliding window correlators within the Parameter Estimation block where each start of frame is detected based on the threshold decision as indicated in the same plot. The decision is based the correlator results shown as a

scaled version of $|P|^2$ and the second half symbol energy samples of R^2 whereas the actual metric M is only shown for convenience. The actual computation of the metric M would require an expensive division operation and is not intended to be used in the final implementation. Notice the boosted time domain samples on top which correspond to the pilot symbols marking the start of an OFDM frame. At the bottom of Fig. 5.21 the time evolution of the complex angle of P (compare to Fig. 4.6 from the MATLAB script) is shown. Within the range of the identical symbol halves the correlation samples of P show a constant phase compared to the random phase outside that region. Consequently this constant phase is used for the fractional frequency error computation.

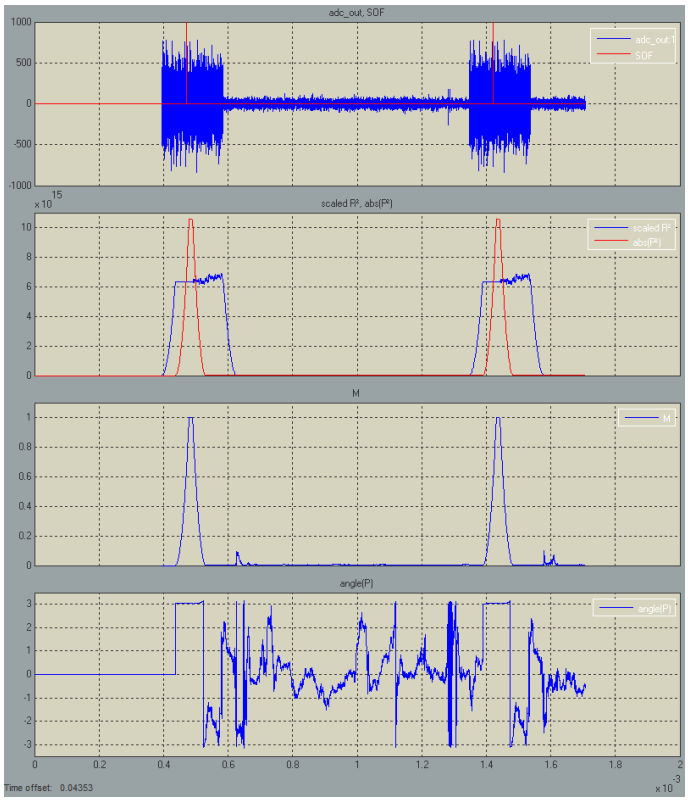


Figure 5.21: Simulink simulation analysis for the Parameter Estimation block

In Fig. 5.22 the results of the joint timing and frequency estimation method after S&C are illustrated. Each estimate is stored over the entire frame time and is updated if the new SOF is detected. Clearly the frequency estimate shows almost no variance[6] under these channel conditions, while the timing estimate considerably changes over the different frames being processed. The larger variance is explained by the fact that the Model-Based implementation uses global maximization method instead of 90% averaging. Despite the variance, the timing estimate still lies within the IBI-free plateau of the metric as depicted in Fig. 5.23. This plot is generated from the actual Embedded MATLAB code using extrinsic functions. It pictures an overlay of several simulation runs where the data content of the buffer (correlation data of

6 Notice that the scaling here is not optimal to point out the small variance, but even under proper scaling conditions the estimate would show rather small deviations from the true value.

$|P|^2$) and the timing estimates (based on a global maximum search) are visualized as markers over the entire buffer length of 512 samples relative to the detected SOF (sample 0). The figure is updated in a single plot where all the estimates appear to be inside the guard-interval. Remember that the timing estimate as in Fig. 5.22 is not directly used for FFT alignment and is illustrated here only for completeness.

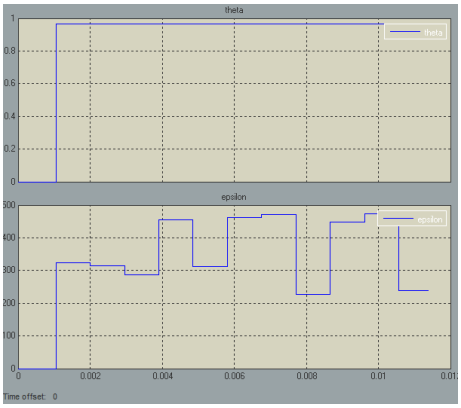


Figure 5.22: Time evolution of the frequency error and timing error estimates of the Model-Based design in Simulink over several OFDM frames

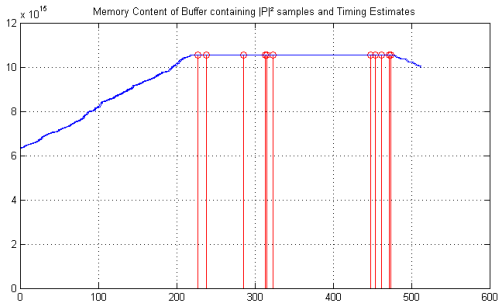


Figure 5.23: Visualization of buffer content in the *argmax* block including the timing estimates

5.3.5.2 Demodulation Analysis

Further analysis results of the simulation model can be seen in Fig. 5.24 where the different stages during demodulation of the frequency corrected signals is shown. These time-dependent signals picture the data flow of the signals entering the OFDM receiver, leaving the FFT processing unit and after FFT buffering along with different control signals.

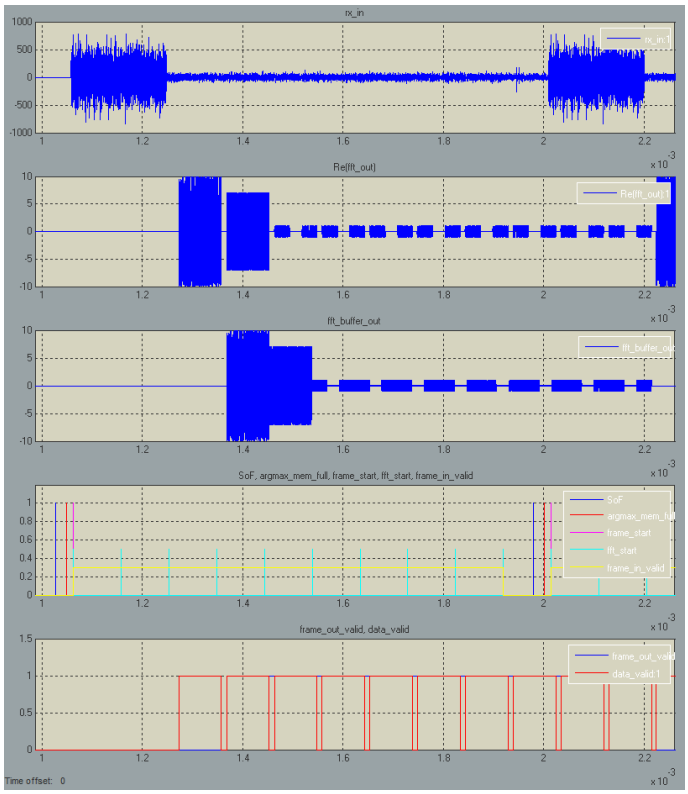


Figure 5.24: Simulink simulation analysis of the OFDM demodulation process

Starting from the delayed real-part samples on top, the FFT output and the buffered FFT samples gradually visualize the signal processing inside the OFDM receiver. Equally important are the corresponding control signals, especially the start pulses for the streaming FFT. The SOF marks the time instant where the incoming frame is detected and after a constant processing time in the *argmax* block has evaluated the actual timing estimate. Depending on the overflow in the buffer the first start pulse is generated followed by additional pulses emerging from the FFT Frame Control block. As mentioned before the spacing of these pulses incorporates the elimination of the guard interval shown in second signal plot. Notice the blanked FFT output of 256 samples after each block of data. The control signals of the bottom plot illustrate the valid time frames of the FFT output used by the FFT Buffer block to compress and realign the FD data (third plot). Observe the discontinuous data stream in the frequency domain for the eight data symbols, where the zeros correspond to the proposed subcarrier mapping scheme in the transmitter. After frame alignment the data can be represented in a scatter plot like in Fig. 5.25, where the constellation diagrams of all OFDM symbols are projected on the time axis. Obviously the synchronization works as intended, since the subcarrier symbols only experience a rotation due to the introduced phase factor resulting from the timing estimator variance. Remember that such linear phase factor can still be resolved when using a non-coherent demodulator. Disregarding the symbol rotation, no other distortions are visible in the constellation diagram despite the frequency error introduced in the channel. Based on this observation, the frequency error synchronization (estimation, compensation) works perfectly in the range of ± 1 subcarrier spacings.

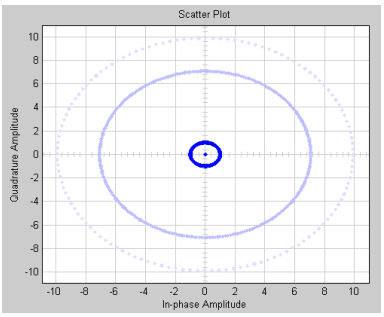
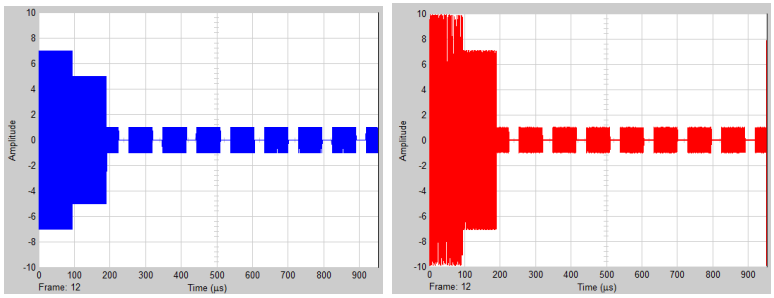


Figure 5.25: Scatter plot after FFT processing of the synchronized system in Simulink

Fig. 5.26 reveals how the sample based data in the receiver is aligned for frame based processing. As seen, the FD domain data, showing the modulated subcarriers of each OFDM symbol, before iFFT, in Fig. 5.26a and after FFT processing, in Fig. 5.26b is contained within one fixed frame (frame time of 952.38μs) as defined for the entire Simulink model. The conversion of sample to frame-based operation gives the user the opportunity to alleviate the ongoing tasks of demodulation until a frame-based bitstream is ready for BER computation.



(a) **(b)**

Figure 5.26: Comparison of TX and RX frame in the frequency domain

5.4 HDL Compatible Simulation Model

The next step is to go from the double precision model to a fixed point realization, feasible for a actual hardware implementation, i.e. a model, that is translatable to HDL code. There are several possible ways to get to a fixed point implementation of the algorithm. One being that the designer chooses all fixed point datatypes itself which is a rather complex task to get the data-sizing right the first time. Another way is to use existing tools that ease the step of implementation by providing additional automatism to the design flow. As mentioned in Section 5.2 MATLAB Simulink integrates the Fixed-Point Advisor and the Fixed-Point Tool for this purpose.

The Fixed-Point Advisor helps to prepare the model for automated HDL code generation by performing several setup tasks that configure the simulation model accordingly:

- Preparing the model for conversion (check supported blocks, setup of signal logging, generating reference data)

- Preparing data typing and scaling (block specific configuration, minimum-maximum values, hardware selection)
- Data typing and scaling (proposal, check for numerical errors, analyze and summarize data types)
- Prepare for code generation

This automated procedure ensures HDL coder compatibility in a step-by-step approach helping to speed up the design process. In order to generate an appropriate reference data the system model has to be adapted, especially the ADC block. Until now this block fulfilled the task of unbuffering the data for sample-based processing while still maintaining double precision. For a fixed-point implementation the actual conversion from double to fixed-point data types, or better the quantization process has to be covered, modeled in some way. As the final target hardware platform uses the Data Conversion High-Speed Mezzanine Card (HSMC) from Altera the built-in ADCs on the card will perform the conversion to the digital domain. In this case a Analog Devices AD9254 with 14-bit precision is used, having differential input range of $2V_{pp}$. Assuming[7] that the virtual channel simulation data only ranges between $\pm$1units (volts), the new ADC model quantizes the complex signal according to a full-scale-range (FSR) of $2V$. The least significant bit (LSB) is evaluated to be $\frac{FSR}{2^{14}}$, which in turn defines the scaling factor of the incoming signal. The model also incorporates data saturation to 14-bit and provides the signed 14-bit number, denoted as `sfix14` (one's complement) to the Parameter Estimation block. A double precision cast is necessary if the Fixed-Point Advisor collects the reference data.

Based on the fixed input range quantized to 14-bit the Advisor requires an initial guess for the internal precision (chosen to be 32bit) and proposes all the fixed-point data types in order to match the full precision implementation. To fine-tune the data types, the designer can choose to use the Fixed-Point Tool, which can improve word and fraction lengths (internal precision) during parameter estimation by using simulation data over several runs and providing visual feedback for the implementational differences. These tools automatically set the chosen data types for the Simulink blocks but can not directly change the Embedded MATLAB code. Therefore, a wrapper (cast) is required for every in- and output which can be removed if the data types are fixed. The Embedded MATLAB blocks then have to be changed in order to employ the proposed types by using so called "fi-Objects". Fi-Objects specify the numeric type of a variable and can handle several different data types.

As far as the rest of the model is concerned, only some minor modification are necessary for a flawless code generation. It is recommended to use the HDL Compatibility Checker before actually going for a fixed-point representation to check whether the model contains only translatable blocks or not. In the case of the S&C algorithm, some blocks have to be implemented differently or even excluded from the design.

The *abs_block* features HDL code generation only, if non-complex data is used[8]. The straightforward implementation by definition evaluates to $\sqrt{\Re(s)^2 + \Im(s)^2}$, if $s = a + jb$ is a complex number. A straightforward implementation of this equation would require 2 real multiplications,

7 The assumption is reasonable if the iFFT output of the transmitter remains unnormalized and if the receiver input is scaled properly (fixed coarse scaling factor to exploit almost the total value range).

8 When using the current Simulink HDL Coder 2.1

2 real additions and a square-root operation. Since the S&C method uses the absolute value computations only prior to a multiplication, this multiplication together with the square-root cancels out, thus saving some implementation space.

Regarding the *angle_block*, only a double precision implementation can be translated to HDL code. Since only fixed-point precision is utilized in the design the calculation of the fractional frequency offset has to be excluded from the Simulink subsystem. An additional external block (*f-offset*) takes the determined maximum value (real and imaginary part) of the correlation P, computes the angle and scales the result to reveal the actual offset. After casting to double the estimated value can be used for the following error correcting mechanisms.

Apart from these changes, the architectural model can now be translated to hardware description in VHDL. The resulting HDL compatible model differs only slightly from the double-precision model and is therefore not shown here. The generated VHDL code is bit-true to the MATLAB implementation (assuming no numerical errors) and features embedded hyperlinks in the code to reference the actual implemented MATLAB code lines. To emphasize the readability of the generated code compare listings 5.2 and 5.1 illustrating the implementation of the Counter module inside the Parameter Estimation block in VHDL and MATLAB. This counter uses a preload value (timing estimate, index) from the *argmax* block and generates the SOF pulse for the FFT window alignment.

Listing 5.1 Fixed-point Embedded MATLAB code of the Counter module.

```
function UF  = fcn(preloadData, preload)

% Down-counter with synchronous preload and underflow detection
nt = numerictype(0, 16, 0);
fm = hdlfimath;

% Setup persistent variables for counter functionality
persistent count_reg underflow reset;

% Initialization
if isempty(count_reg)
    count_reg = fi(0, nt, fm);
    underflow = false;
    reset = true;
end

% Generate underflow pulse (duration = 1 sample)
% hold reset
if underflow
 underflow = false;
 reset = true;
end

% Register transfer
UF = underflow;

% Load counter preload value on demand, clear reset flag
if preload
    count_reg = fi(preloadData, nt, fm);
    reset = false;
% Normal operation
% hold counter to zero if reset, otherwise
% detect underflow, reset counter
% or count downwards (normal operation)
else
    if reset == true
      count_reg = fi(0, nt, fm);
    else
      if count_reg == 0
        underflow = true;
        UF = underflow;
        count_reg = fi(0, nt, fm);
      else
        dec = count_reg - fi(1, nt, fm);
        count_reg = fi(dec, nt, fm);
      end
    end
end
```

Listing 5.2 Autogenerated VHDL code (snippet) of the Counter module.

```
ARCHITECTURE rtl OF Counter IS

  -- Signals
  SIGNAL preloadData_unsigned             : unsigned(15 DOWNTO 0);  -- uint16
  SIGNAL count_reg                        : unsigned(15 DOWNTO 0);  -- ufix16
  SIGNAL underflow                        : std_logic;
  SIGNAL reset_1                          : std_logic;
  SIGNAL count_reg_next                   : unsigned(15 DOWNTO 0);  -- ufix16
  SIGNAL underflow_next                   : std_logic;
  SIGNAL reset_next                       : std_logic;

BEGIN
  preloadData_unsigned <= unsigned(preloadData);

  Counter_1_process : PROCESS (clk, reset)
  BEGIN
    IF reset = '1' THEN
      count_reg <= to_unsigned(0, 16);
      underflow <= '0';
      reset_1 <= '1';
    ELSIF clk'EVENT AND clk = '1' THEN
      IF enb_1_8_0 = '1' THEN
        count_reg <= count_reg_next;
        underflow <= underflow_next;
        reset_1 <= reset_next;
      END IF;
    END IF;
  END PROCESS Counter_1_process;

  Counter_1_output : PROCESS (preloadData_unsigned, preload, count_reg, underflow, reset_1)
    VARIABLE dec : unsigned(16 DOWNTO 0);
    VARIABLE underflow_temp : std_logic;
    VARIABLE reset_temp : std_logic;
  BEGIN
    underflow_temp := underflow;
    reset_temp := reset_1;
    count_reg_next <= count_reg;
    --MATLAB Function 'Parameter Estimation/Counter': '<S85>:1'
    -- Down-counter with synchronous preload and underflow detection
    -- Setup persistent variables for counter functionality
    -- Initialization
    -- Generate underflow pulse (duration = 1 sample)
    -- hold reset
    IF underflow = '1' THEN
      --'<S85>:1:20'
      --'<S85>:1:21'
      underflow_temp := '0';
      --'<S85>:1:22'
      reset_temp := '1';
    END IF;
    -- Register transfer
    --'<S85>:1:26'
    UF <= underflow_temp;
    -- Load counter preload value on demand, clear reset flag
    IF preload = '1' THEN
      --'<S85>:1:29'
      --'<S85>:1:30'
      count_reg_next <= preloadData_unsigned;
      --'<S85>:1:31'
      reset_temp := '0';
      -- Normal operation
      -- hold counter to zero if reset, otherwise
      -- detect underflow, reset counter
      -- or count downwards (normal operation)
    ELSIF reset_temp = '1' THEN
      --'<S85>:1:37'
      --'<S85>:1:38'
      count_reg_next <= to_unsigned(0, 16);
    ELSIF count_reg = 0 THEN
      --'<S85>:1:41'
      underflow_temp := '1';
      --'<S85>:1:42'
      UF <= '1';
      --'<S85>:1:43'
      count_reg_next <= to_unsigned(0, 16);
    ELSE
      --'<S85>:1:45'
      dec := resize(count_reg, 17) - 1;
      --'<S85>:1:46'
      count_reg_next <= dec(15 DOWNTO 0);
    END IF;
    underflow_next <= underflow_temp;
    reset_next <= reset_temp;
  END PROCESS Counter_1_output;

END rtl;
```

The current model uses no optimization methods in terms of area and speed as provided by the HDL coder. As the resource utilization report in Table 5.1 on page 94 shows, the generated hardware implementation requires:

Multipliers	10
Adders/Subtractors	22
Registers	6160
RAMs	0
Multiplexers	25

Table 5.1: Resource utilization report of the generated Parameter Estimation block HDL implementation

The model employs a very specific, highly adapted fixed-point implementation to meet the double-precision model as best as possible. Hence, the reusability of the modules is limited making resource sharing or other optimization methods currently not feasible for the design. If the system designer chooses to lose the requirements on precision, a more optimized realization would certainly be possible.

To validate the generated model Simulink HDL coder offers a feature to automatically generate a testbench for HDL simulation or even a new model ready for cosimulation with third party tools, e.g. ModelSim from Mentor Graphics. Using cosimulation allows for a direct comparison of the fixed-point model (reference device under test) and the corresponding HDL model. Based on the HDL code, Simulink automatically generates and inserts an entire new model containing a cosimulation interface to ModelSim where the generated HDL code is represented as a black box having the exact same in- and outputs as the Simulink block. Each output is compared and visualized in different scope plots to show the difference of the implementation providing an in-system verification method. Refer to Fig. 5.27 for a block diagram of the cosimulation setup of the Parameter Estimation block.

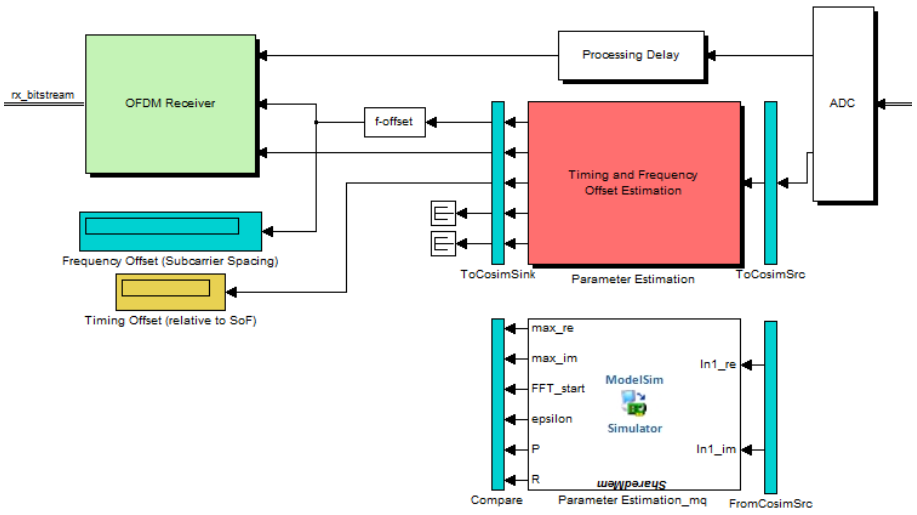
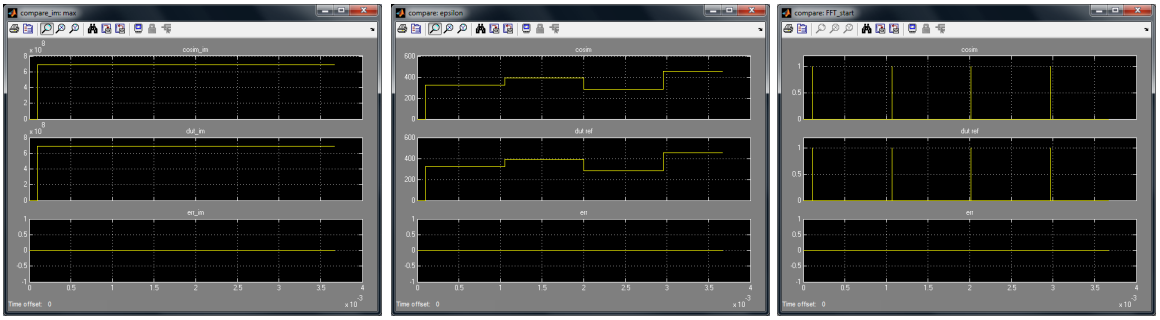


Figure 5.27: Generated HDL cosimulation testbench model in Simulink

Fig. 5.28 pictures the outcome of the cosimulation validation process between the reference

DUT and the *cosim* model. By using visual inspection the relevant block output for error compensation of timing and frequency errors proves to be identical in both models since the error is exactly zero during the entire simulation process. Fig. 5.28a shows time evolution of the maximum values of the correlation P (quadrature phase component) which is later on used for fraction frequency error computation whereas in Fig. **??**Cosimulation-epsilon: the results of the timing estimate, together with the corresponding FFT start pulses in **Fig. 5.28c** are illustrated. These results confirm that the generated model behaves as intended where the demodulation process following parameter estimation attains equal performance.



(a) **(b)** **(c)**

Figure 5.28: Comparison scope plots during cosimulation

In case of the HDL testbench, reference data is collected over a specified simulation time which is then used to construct test case scenarios for data verification. The testbench is loaded to ModelSim (assuming a compiled VHDL model) and executed automatically until all test cases are examined. Textual statements inform about the test results, i.e. passed or not passed.

For completeness, Fig. 5.29 reveals the HDL simulation results in ModelSim. The progress of the data signals is automatically updated during cosimulation where in this case the appropriate signals are represented in an analog format for convenience. The figure depicts all block output signals evaluated form the HDL model based on the Simulink input data (provided via the cosimulation interface). The clock frequency is directly related to the simulation time in Simulink and all control signals (`clk_enable`, `reset`, etc.) are automatically inserted in the testbench.

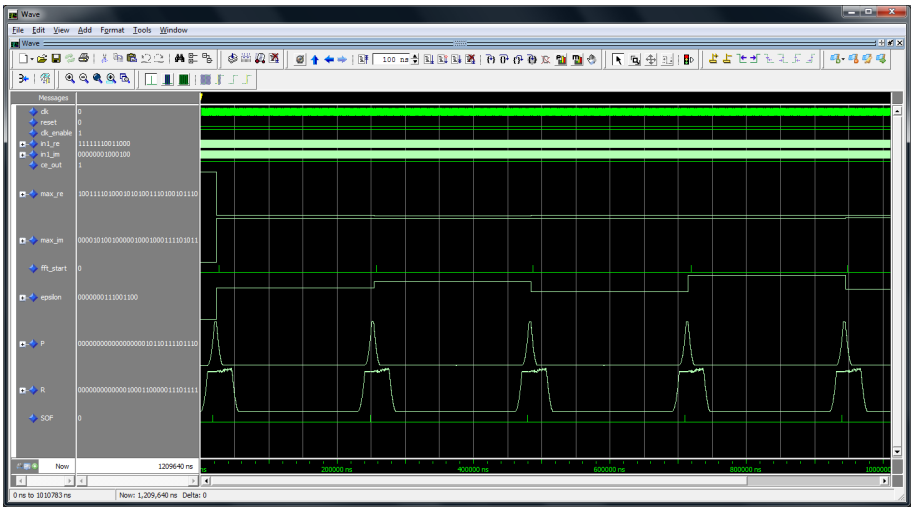


Figure 5.29: Wave diagrams in ModelSim during cosimulation

5.5 Altera DSP Builder

As Altera is considered the vendor of choice for the hardware platform of the CertLink modem Simulink HDL Coder reveals some drawbacks as the far as synthesizing is concerned. The reason for this being that the Coder in the (current) version 2.1 does not provide enough support in terms of an efficient mapping of the design to hardware and in-system verification. The mapping of the functional VHDL code to actual FPGA slices, i.e. generating the bitstream, is a separate issue to discuss. Mainly because it strongly depends on the hardware and the synthesis tool, how the HDL code is interpreted. In the case of a Model-Based design the FPGA vendors (Xilinx, Altera) provide specifically design blocks for Simulink that can be optimized for their hardware. By using only HDL compatible blocks from the Simulink library a rather inefficient design will result from the process of synthesizing. This is especially true if dedicated blocks for digital signal processing purposes are employed as generic blocks/code (CIC filters, FFT, modulators, ...). Despite the benefit of the Coder outcome being independent of the hardware platform it becomes dependent in the DSP case. Simple counters or memory would possibly lead to a feasible mapping to hardware but not so for special high performance applications like OFDM PHY layers.

Now, as mentioned, Simulink offers the possibility to integrate third party design flows for FPGA configuration (synthesizing, routing, etc.) but it lacks support for verification of Altera designs. By the term in-system verification, a FPGA-in-the-loop or hardware-in-the-loop (HIL) scenario is envisioned that can be realized by directly integrating a black box model (FPGA development board) in Simulink and performing FPGA cosimulation. This verification approach is possible for Xilinx devices but not for Alteras FPGAs.

To be able to test the entire design based on a Model-Based approach Altera's DSP Builder is the tool of choice. The DSP Builder extends the Simulink library with two additional toolboxes (Standard & Advanced Blockset) that offer several Altera specific, synthesizable intellectual properties (IP) modeled as Simulink blocks. Additionally the projected Altera Cyclone III development board is directly supported and can be incorporated to the design. Altera's DSP

Builder provides an entirely different hardware design flow as described for the Simulink HDL Coder as depicted in Fig. 5.30. Essentially, the HDL code generation capability of Simulink becomes obsolete.

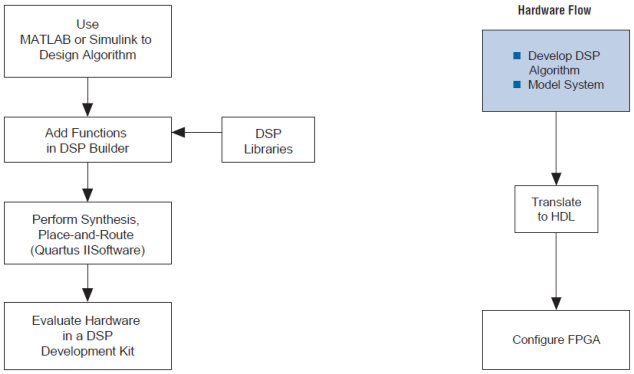


Figure 5.30: General and hardware specific design flow for Altera's DSP Builder

For a general introduction to DSP Builder and each blockset, the reader is referred to the exhaustive documentation in [40, 41, 42]. Here only a summary of the blockset features is mentioned in order to show an overview of the capabilities.

5.5.1 Standard Blockset

The standard blockset provides several libraries of design and interface Simulink blocks together with a library of DSP MegaCore functions from Altera intended for basic arithmetic and storage operations. See [40, 41] for more information. The following bullet list characterizes only the essential features for the standard blockset, cited from [40]:

- *"Cycle-accurate behavioral models"*
- *"Multiple clock domain management"*
- *"Control rich with backpressure support"*
- *"Access to specific hardware device features"*
- *"Hardware-in-the-loop (HIL) support enables FPGA hardware cosimulation"*
- *"Support for importing VHDL or Verilog HDL design entities"*
- *"Tabular and graphical state machine support"*
- *"Rapid prototyping using Altera DSP development boards"*
- *"SignalTap® II logic analyzer debugging support"*
- *"Direct instantiation of DSP IP cores"*

5.5.2 Advanced Blockset

The optional advanced blockset uses timing-driven IP blocks for high performance applications, such as filters, numerical oscillators and FFT computations. Some of the additional features of the advanced blockset are presented here (cited from [40]):

- *"Specification driven design with automatic pipelining and folding"*
- *"High level synthesis technology"*
- *"High performance timing-driven IP models"*
- *"Multichannel designs with automatically vectorized inputs"*
- *"Automatic generation of memory-mapped interfaces"*
- *"Simulink fixed-point types"*
- *"Single system clock for the main datapath logic"*
- *"Feed-forward datapath with minimum control"*
- *"Portability across different device families"*
- *"High-level resource trade-offs such as hard versus soft multipliers"*

5.5.3 HDL Coder and DSP Builder Coexistence

Although the essential Simulink HDL Coder design flow becomes essentially obsolete in the DSP Builder, there is still the possibility to design algorithms using the HDL Coder environment and integrate the results to the main flow of the DSP Builder. As Fig. 5.31 illustrates, two options are available to import different HDL designs by utilizing the HDL Import Block or the Subsystem Builder that provides additional features. So either use some third party legacy code or the HDL Coder outcome. In this way the benefits of a Simulink HDL Coder approach are maintained whereas the specific implementational issues of high performance DSP systems is covered by the Altera DSP Builder.

Regarding the synchronization design, the main building blocks could also be exchanged by Altera IPs for a more optimized synthesis design and performing system simulation and in-system verification in the DSP Builder.

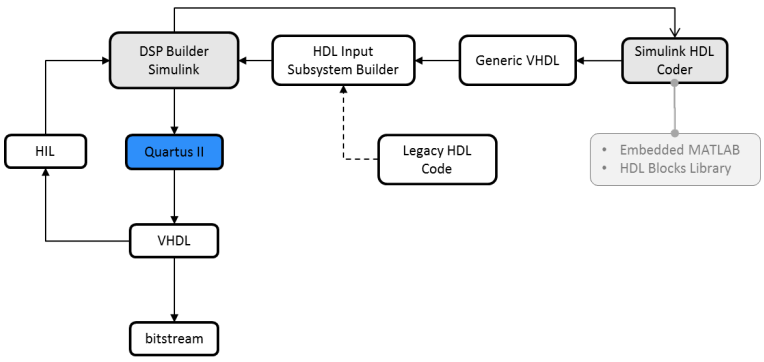


Figure 5.31: Design flow of DSP Builder incorporating Simulink HDL Coder generated VHDL code utilizing an extended design environment

6 Discussion and Final Results

This chapter reflects on the thesis while discussing the main results as well as the shortcomings at hand. Starting with the review of the initial problem description, the theoretical results together with the simulation and implementation results will be subsequently examined. A short concluding summary at the end of the chapter will give a comprehensive overview of the presented work.

In the scope of the CertLink project a modem is developed in order to extend a wire-based TTP bus system over a Wireless-TTP interface. To mitigate the effects of multipath fading channels, to provide robustness against narrowband interference and to employ means of spectral redundancy an OFDM modulation scheme is utilized for the certifiable wireless data link. As OFDM systems are very sensitive to timing and frequency offsets the physical layer design has to provide measures for receiver synchronization. One mandatory part of the system design is to implement sophisticated synchronization algorithms in hardware to ensure a reasonable system performance.

6.1 Simulation

A first simulation concept was proposed to accommodate for a complete executable system specification based on a set of requirements. The main purpose was to get familiar with the implementational aspects of a working OFDM scheme and to look at the overall system performance in terms of BER results. Concerning the receiver design, the task of synchronization was of most interest since it is a prerequisite in demodulating the incoming data packets. The simulation script tried to model the signal chain in a modular, structured way in order to be easily adaptable and interchangeable. Hence, future changes in the design can be conveniently tested and evaluated within the MATLAB simulation. As this script only provides a basis for the ongoing design process not every sub-module of the real physical layer is accurately modeled.

One of them being the TTP data and payload interface where currently only random data burst are generated in order to simulate the bus data. However, a reasonable detailed modeling at this abstract design level is not feasible since no actual hardware interface to MATLAB is available and the data acquisition/detection can be excluded from the PHY layer specification without affecting the functional behavior. The data interleaving between the TTP and payload information bits is not implemented because the specific design requirements were not fixed up to now.

Another general issue is concerned about the burst frame transmission modeling in MATLAB. As mentioned in Chapter 4 the propagation of the data is accomplished with the help of matrix notation and can therefore not exactly represent a realistic time-dependent dynamic system behavior. Despite the disadvantage this way of modeling leads to a rather short design

iteration where newly adaptations can be easily integrated into the signal processing chain. To cope with the lack of realism Simulink provides enough flexibility to facilitate a more natural modeling approach. This (along with other useful attributes) being the reason for mapping the MATLAB script to a Simulink environment.

As it is the case for most script based simulations the ability to configure the system is useful but can be very hard to follow without having any preknowledge about the intentions of the system designer. So basically the software switches and configurability options have to be appropriately documented and clearly presented to the user. Since the script is intended as pre-step towards a Model-Based Design the documentation is limited but sufficient.

As there are many different possible methods and algorithms for synchronizing an OFDM system a selection of three different approaches was made. This selection was based on extensive research in the field of OFDM synchronization and should illustrate the basic ideas and intentions moreover providing some insight in the evolution and improvements of such algorithms. Relying on one of the most recognized papers in the domain by Schmidl and Cox the concept of repetitive preamble structures was introduced and implemented along with the subsequent improved algorithms by Minn and Park. As researching activities progress it is quite natural that more advanced methods evolve which lead to more optimized design regarding several aspects, e.g. implementational complexity and performance measures. Since this is the case for most practical systems this thesis relies on the most fundamental method after S&C. One of the reasons being that almost every novel idea (in the same field) applied very similar concepts.

The evaluation of those algorithms in MATLAB showed to be very useful in terms of the process of synchronization (error estimation and correction) as well as to locate possible implementation difficulties and constraints. Every algorithm is provided with the same input data and estimates the corresponding parameters. Because only S&C has the ability to estimate (integer) frequency errors there is essentially no comparable alternative. The estimator performance plots used for comparison of the statistical mean and variance or the estimator characteristics were inspired by different scientific papers and should cover the most important measures for a reasonable conclusion on how "good" each algorithm operates. Other plots tried to reproduce and verify different results from the authors to check whether the implementation worked as intended. A three dimensional representation of the OFDM symbols in the frequency domain allowed to gain more insight on how the impairments of the channel actually impact the signal constellation at the receiver. Additionally it provided an alternative view on the OFDM multicarrier transmission concept where the subcarrier convey the information in parallel. Bit error rate computations revealed the overall system performance by manually analyzing biterrors in the resulting datastream of the receiver.

The script employs the ability to simulate AWGN and Ricean fading channels, although the multipath channel model was only implemented for testing purposes and did not actually contribute to any relevant or exhaustive performance measures. It would have been possible to use the `bertool` features to fully integrate a sophisticated standardized data analysis but only together with structural changes in the simulation design. However, some results for comparison of the BERs were generated by the `bertool` and have been conveniently imported to the simulation plots.

The different plots which have been shown throughout this thesis mostly differ in their SNR depiction. BER comparisons are usually based on the signal to noise ratio per bit where special attention has to be paid to OFDM systems to accommodate for each subcarrier of actual data

symbols by using explicit normalization. Normal SNR measurements for estimation analysis use a different normalization and account for each OFDM symbol embedded in the datastream. Without careful selection of these normalization factors the gained results would not reflect the true achieved performance in terms of BER or statistical (estimation related) measurements.

The flexible MATLAB simulation approach helped to take a first step towards an OFDM PHY layer hardware design including deepened knowledge about synchronization processes within the receiver architecture.

6.2 Implementation

Using the results from the script based simulation the whole design could be mapped to a more natural modeling design environment: Simulink. Dynamic system modeling, especially helpful for digital communication systems, is the key feature of Simulink. The Model-Based Design approach corresponds to a more natural system representation using graphical blocks of functionality. Additionally the easy-to-use measurement and analysis tools from the different toolboxes assist the rapid development for such a system.

As the MATLAB script was largely structured in a modular way the transfer to Simulink was very intuitive although the particular Simulink concepts had to be considered and incorporated to the design. One could argue that the entire design could have been developed directly in Simulink without the pre-step of using MATLAB scripts. This is true if enough Model-Based experience is available by the system designer, otherwise the two step procedure alleviates the process. Additionally a rough evaluation of certain aspects is easier done in MATLAB, especially due to the ability generating customized data plots. In retrospective, the development of a working, configurable end-to-end Simulink system did consume quite a lot of time due to limited previous knowledge.

The incorporation of the automated code generation tool Simulink HDL Coder required additional effort the get the system setup right. Several requirements had to be fulfilled in order to ensure a smooth and flawless code generation. Careful design choices of the Parameter Estimation block were necessary to directly translate the block oriented architecture to a functional hardware description.

6.3 Shortcomings and Future Improvements

The subsequent section will provide insight in some implementation specific issues and discuss their shortcomings together with possible future improvements.

6.3.1 Frame Misalignment

A mixture of frame and sample-based simulation degrades the overall system performance in terms of execution time. An entirely frame-based design decreases the execution time by propagating blocks of data rather than single bits through the signal chain. For a low-area design of the synchronization module for the receiver a sample-based approach is more reasonable and forced to mix both modeling concepts. As more and more components of the CertLink modem

will be prepared for automated code generation a complete sample-based design will be very likely to emerge if the HDL coder remains the first choice. Switching between frames and samples implications further issues during data conversion.

An illustrative example is the conversion of the FFT output in the receiver after synchronizing the system. Usually Buffer and Unbuffer blocks do the job of converting the data, however, in the case of the FFT Buffer block a control signal decides if a block of data is valid or not. Because the Simulink system Buffer block executes several tasks, embedding it in an Enabled Subsystem could cause inconsistent data if the timing is not perfect. Some problems emerged during the coding of the FFT Buffer code were the FFT output is rebuffered in order to eliminate the irrelevant data caused by the guard-interval removal. Problems, regarding the correct rebuffering were the input data should be exactly replicated only time-delayed. The reason for this could not be investigated in time leading to a frame misalignment for the further processing steps. Such a misalignment makes a BER comparison technically obsolete since wrong (time-shifted) bit sequences are compared to each other. Bearing in mind these shortcomings the system performance, *including* a functional hardware implementation of the selected synchronization algorithm, could not be accomplished within the Simulink framework.

6.3.2 Estimator Improvements

Nevertheless, MATLAB simulations showed the actual performance results of the S&C method, so, the potential Simulink BER measurements can be assumed to deliver similar results, disregarding the different method for finding the maximum of the timing metric (global maximization instead of 90% averaging). Improvement of the synchronization method itself is another issue to discuss. One easy step would be to incorporate an averaging method for the timing estimation rather than the currently implemented global maximum search. If the requirements of the receiver clock accuracy can not be fulfilled additional effort needs to be invested to enlarge the frequency estimation range by either using the second OFDM symbol or employing an entirely different (overhead optimized) version were only a single symbol is utilized for a joint estimation method [24]. Using the second symbol for the S&C requires to manipulate data in the frequency domain along with the computation of a new metric to estimate the integer frequency offset. Manipulations in the frequency domain also apply for the improved frequency estimation by Morelli et. al. were the OFDM symbol employs more than two repetitive parts. These operations in the FD could cause an increased receiver complexity and would not benefit a low-area implementation. If the entire synchronization architecture is redesigned (newly adopted algorithm) the steps from a theoretical model to an executable hardware specification will be rather similar to the approach presented in this thesis. The surrounding framework in Simulink is available allows for a rapid adaptation in case of a new design.

6.3.3 Hardware Mapping

If it comes to transferring the HDL code to an FPGA platform (Altera devices) the entire Simulink design flow reveals some drawbacks. Although the HDL Workflow Advisor allows to fully integrate the design into the Quartus design solution (synthesizing, place&route, etc.) the outcome can not be expected to be high-performance. Mainly because the Simulink environment itself does not provide highly optimized simulation models for Altera devices. Only in

conjunction with the add-on software DSP Builder from Altera new additional library blocksets become available. However, using these blocks urges the HDL Coder more to the background since the HDL code generation is implicitly accomplished by the DSP Builder. The entire simulation concept changes because of the imposed constraints by the Builder, where target specific configuration have to be considered more thoroughly differently than for the HDL Coder. A more systematic evaluation of the DSP Builder features will become more demanding as the CertLink modem design advances. Especially in combination with existing simulation architectures regarding the OFDM synchronization.

A co-existence is possible but the efficiency of such a heterogeneous design has to be estimated. During the early system development it was not so clear that the DSP Builder would become the actual environment of choice because other FPGA vendors, like Xilinx, showed to be better integrated into the Simulink design flow. Particularly, the integration of existing FPGA development boards along with a potential FPGA-in-the-loop, in-system verification approach. But despite the shortcomings of the current simulation design the underlying architecture of the synchronization module has proven to be valuable and it would require only little effort to map the existing subblocks of the design to more optimized blocks from the DSP Builder library. Changing the representation does not necessarily mean to change the functionality. Embedded MATLAB blocks can still be translated to a hardware description by the HDL Coder and imported to the DSP design.

7 Conclusions

The thesis was structured in a way to easily follow the system development beginning with an extensive introduction to the CertLink project in order to describe the concepts and project goals of a certifiable modem development. As the OFDM scheme is used in CertLink modems to cope with severe multipath fading and establish a reliable datalink it was convenient to provide a general introduction to OFDM systems along with their assets and drawbacks as seen in Chapter 2. In order to enable the reader to fully understand the concepts and mathematical approaches of the topic a unified mathematical model was introduced which tried to emphasize on a conclusive description from OFDM system theory to synchronization aspects (effects of timing and frequency errors) and aeronautical channel models. In that way the entire signal chain including all relevant components was explained in great detail. Based on this a more theoretical approach the important system architecture was covered to gain more insight in the requirements of the development process. The Time Triggered Protocol (TTP) was explained in a nutshell along with the theoretical background of existing aeronautical datalink specifications and the currently evolved system specifications.

Using this fundamental knowledge Chapter 4 began with a first simulation concept to cover most of the system peculiarities in a MATLAB environment. A fully configurable signal model was developed that allowed the user to evaluate the system's performance together with the analysis of selected OFDM synchronization algorithms. Essential BER performance measures for digital communication systems and statistical estimator analysis provided decisive information about the system parameter configuration and allowed to chose one algorithm for a feasible hardware implementation. The well known joint time- and frequency synchronization algorithm after S&C was selected to be implemented purely in hardware for a low-complexity receiver design using a Model-Based design method in the Simulink environment.

In the Simulink environment a dynamic end-to-end CertLink system model introduced a time-dependent characteristic to simulate the modem architecture in a more realistic scenario. Again a comfortable system configuration facilitated an on-the-fly alteration of the system behavior especially with respect to varying channel conditions. The unified design flow in use was based on a very abstract level description and gradually refined the model complexity in a hierarchical manner to show the benefits of a such contiguous design approach. To set the advantages in perspective a short description of an alternative but also typical approach was provided in order to reveal the superiority of the current design flow. The refinement procedure led to a single system block for parameter estimation to synchronize the receiver module to the appearing timing and frequency offsets on the transmission path.

By employing additional toolboxes from MATLAB Simulink the system block was prepared for the fully automated process of HDL code generation. Moreover the generated fixed-point model has been directly embedded in a cosimulation (hardware testbench) environment to systematically verify that the hardware description actually behaved analogous to the device under

test. The developed HDL model provided the groundwork for incorporating the synchronization module into a physical FPGA platform.

In conclusion, the thesis covered most of the aspects concerning the complex development process of an OFDM physical layer and revealed methods for an efficient design approach to implement system modules in hardware. Especially in terms of rapid prototyping it features possibilities to quickly react to changing requirements by performing thoroughly design reiterations following the Model-Based design flow.

Bibliography

[1] H. Flühr, M. Gruber, K. Kainrath, E. Knoll, T. Raggl, A. Gruber, M. Artner, and S. Morawitz, "Performance Evaluation of a Command-and-Control Data Link for Civilian Unmanned Aerial Vehicles," in *60. Deutschen Luft- und Raumfahrtkongress (DLRK 2011), 27.-29. September*, Congress Centrum Bremen, 09 2011.

[2] M. Gruber, H. Flühr, K. Kainrath, E. Knoll, A. Gruber, and M. Artner, "Systemarchitektur für unbemannte Flugsysteme mit COTS-Komponenten unter Zertifizierungsrandbedingungen," in *60. Deutschen Luft- und Raumfahrtkongress (DLRK 2011), 27.-29. September*, Congress Centrum Bremen, 09 2011.

[3] BMVIT. (2011) Innovation - Aeronautics. Bundesministerium für Verkehr, Innovation und Technologie. [Online]. Available: http://www.bmvit.gv.at/en/innovation/aeronautics/index.html

[4] TTTech Computertechnik AG. (2011) Web Presence. TTTech Computertechnik AG. Schoenbrunner Strasse 7, A-1040 Vienna, Austria. [Online]. Available: http://www.tttech.com/home/

[5] Airborne Vision GmbH. (2011) Zivile Film- und TV-KameraDrohne (VTOL UAV). Graumanngasse 7/B/5.Stock 1150 Wien. [Online]. Available: http://www.airbornevision.com/

[6] ITU-R, "2012 World Radiocommunication Conference," in *Agenda and References (Resolutions and Recommendations)*, 2012. [Online]. Available: http://www.itu.int/oth/R0C04000007/en

[7] R. Prasad, *OFDM for wireless communications systems*, ser. Artech House universal personal communications series. Artech House, 2004.

[8] S. Weinstein and P. Ebert, "Data Transmission by Frequency-Division Multiplexing Using the Discrete Fourier Transform," *Communication Technology, IEEE Transactions on*, vol. 19, no. 5, pp. 628 –634, october 1971.

[9] M. Speth, S. Fechtel, G. Fock, and H. Meyr, "Optimum receiver design for wireless broadband systems using OFDM. I," *Communications, IEEE Transactions on*, vol. 47, no. 11, pp. 1668 –1677, nov 1999.

[10] M. Morelli, C.-C. Kuo, and M.-O. Pun, "Synchronization Techniques for Orthogonal Frequency Division Multiple Access (OFDMA): A Tutorial Review," *Proceedings of the IEEE*, vol. 95, no. 7, pp. 1394 –1427, july 2007.

[11] H. Harada and R. Prasad, *Simulation and Software Radio for Mobile Communications*. Artech House, 2002, no. 1-58053-044-3.

[12] S. Hara and R. Prasad, *Multicarrier Techniques for 4G Mobile Communications*. Norwood, MA, USA: Artech House, Inc., 2003.

[13] O. Edfors, M. Sandell, J.-J. van de Beek, S. Wilson, and P. Ola Borjesson, "OFDM channel estimation by singular value decomposition," in *Vehicular Technology Conference, 1996. 'Mobile Technology for the Human Race'., IEEE 46th*, vol. 2, apr-1 may 1996, pp. 923 –927 vol.2.

[14] R. Stolle, "OFDM-Modulator," Labor für Hochfrequenztechnik, Tech. Rep., 2010.

[15] J. Proakis and M. Salehi, *Grundlagen der Kommunikationstechnik*, ser. et - Elektrotechnik. Pearson Studium, 2004.

[16] J. Stott, "The how and why of COFDM," Research and Development, EBU Technical Review, 1998.

[17] T. Pollet and M. Moeneclaey, "Synchronizability of OFDM signals," in *Global Telecommunications Conference, 1995. GLOBECOM '95., IEEE*, vol. 3, nov 1995, pp. 2054 –2058 vol.3.

[18] T. Pollet, M. Van Bladel, and M. Moeneclaey, "BER sensitivity of OFDM systems to carrier frequency offset and Wiener phase noise," *Communications, IEEE Transactions on*, vol. 43, no. 234, pp. 191 –193, feb/mar/apr 1995.

[19] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-time signal processing (2nd ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1999.

[20] T. Schmidl and D. Cox, "Low-overhead, low-complexity [burst] synchronization for OFDM," in *Communications, 1996. ICC 96, Conference Record, Converging Technologies for Tomorrow's Applications. 1996 IEEE International Conference on*, vol. 3, jun 1996, pp. 1301 –1306 vol.3.

[21] T. Schmidl and D. Cox, "Robust frequency and timing synchronization for OFDM," *Communications, IEEE Transactions on*, vol. 45, no. 12, pp. 1613 –1621, dec 1997.

[22] T. Schmidl and D. Cox, "Timing and Frequency Synchronization of OFDM Signals," U.S. Patent 5 732 113, March 24, 1998, appl. No.: 08/666,237. [Online]. Available: http://www.patents.com/us-5732113.html

[23] P. Moose, "A technique for orthogonal frequency division multiplexing frequency offset correction," *Communications, IEEE Transactions on*, vol. 42, no. 10, pp. 2908 –2914, oct 1994.

[24] M. Morelli and U. Mengali, "An improved frequency offset estimator for OFDM applications," *Communications Letters, IEEE*, vol. 3, no. 3, pp. 75 –77, mar 1999.

[25] H. Minn, M. Zeng, and V. Bhargava, "On timing offset estimation for OFDM systems," *Communications Letters, IEEE*, vol. 4, no. 7, pp. 242 –244, jul 2000.

[26] B. Park, H. Cheon, C. Kang, and D. Hong, "A novel timing estimation method for OFDM systems," *Communications Letters, IEEE*, vol. 7, no. 5, pp. 239 – 241, may 2003.

[27] ETSI, *Electromagnetic compatibility and Radio spectrum Matters (ERM); VHF air-ground Digital Link (VDL) Mode 2; Technical characteristics and methods of measurement for ground-based equipment; Part 1: Physical layer and MAC sub-layer*, ETSI European Standard (Telecommunications series) ETSI EN 301 841-1, August 2003, rEN/ERM-TG25-022-1.

[28] ETSI, *VHF air-ground and air-air Digital Link (VDL) Mode 4 radio equipment; Technical characteristics and methods of measurement for aeronautical mobile (airborne) equipment; Part 1: Physical layer*, ETSI European Standard ETSI EN 302 842-1, July 2011, rEN/AERO-00011-1.

[29] M. Sajatovic, B. Haindl, M. Ehammer, T. Gräupl, M. Schnell, U. Epple, and S. Brandes, "L-DACS1 System Definition Proposal: Deliverable D2," European Organisation for the Saftey of Air Navigation, Eurocontrol, Tech. Rep. CIEA15EN50002.10, 2009.

[30] M. Sajatovic, B. Haindl, C. Rihacek, M. Schnell, S. Gligorevic, and S. Brandes, "L-DACS1 System Definition Proposal: Deliverable D3 - Design Specifications for L-DACS1 Prototype," European Organisation for the Saftey of Air Navigation, Eurocontrol, Tech. Rep. CIEA15EN50003.10, 2009.

[31] P. Burns, *Software Defined Radio for 3G*. Artech, 2003, no. 1-58053-347-7.

[32] Altera, *Cyclone III Development Board Reference Manual*, Altera Corporation, 101 Innovation Drive San Jose, CA 95134, March 2008. [Online]. Available: www.altera.com

[33] K. Hofbauer, "Speech Watermarking and Air Traffic Control," Ph.D. dissertation, Graz University of Technology, Austria, 2009.

[34] E. Haas, "Aeronautical channel modeling," *Vehicular Technology, IEEE Transactions on*, vol. 51, no. 2, pp. 254 –264, mar 2002.

[35] The Mathworks, *Communications System Toolbox User's Guide*, The Mathworks, Inc., 3 Apple Hill Drive Natick, MA 01760-2098, 2011.

[36] M. Speth, F. Classen, and H. Meyr, "Frame synchronization of OFDM systems in frequency selective fading channels," in *Vehicular Technology Conference, 1997, IEEE 47th*, vol. 3, may 1997, pp. 1807 –1811 vol.3.

[37] C. Dick and F. Harris, "FPGA implementation of an OFDM PHY," in *Signals, Systems and Computers, 2003. Conference Record of the Thirty-Seventh Asilomar Conference on*, vol. 1, nov. 2003, pp. 905 – 909 Vol.1.

[38] The Mathworks, *DSP System Toolbox User's Guide*, 8th ed., The Mathworks, Inc., 3 Apple Hill Drive Natick, MA 01760-2098, September 2011.

[39] The Mathworks, *Simulink HDL Coder 2 User's Guide*, 2nd ed., The Mathworks, Inc., 3 Apple Hill Drive Natick, MA 01760-2098, April 2011.

[40] Altera, *DSP Builder Handbook Volume 1: Introduction to DSP Builder*, 2nd ed., Altera Corporation, 101 Innovation Drive San Jose, CA 95134, July 2011. [Online]. Available: www.altera.com

[41] Altera, *DSP Builder Handbook Volume 2: DSP Builder Standard Blockset*, 1st ed., Altera Corporation, 101 Innovation Drive San Jose, CA 95134, April 2011. [Online]. Available: www.altera.com

[42] Altera, *DSP Builder Handbook Volume 3: DSP Builder Advanced Blockset*, 2nd ed., Altera Corporation, 101 Innovation Drive San Jose, CA 95134, April 2011. [Online]. Available: www.altera.com

[43] H. Abdzadeh-Ziabari, M. Shayesteh, and M. Manaffar, "An improved timing estimation method for OFDM systems," *Consumer Electronics, IEEE Transactions on*, vol. 56, no. 4, pp. 2098 –2105, november 2010.

[44] B. Ai, Z. xing Yang, C. yong Pan, J. hua Ge, Y. Wang, and Z. Lu, "On the synchronization techniques for wireless OFDM systems," *Broadcasting, IEEE Transactions on*, vol. 52, no. 2, pp. 236 – 244, june 2006.

[45] A. Awoseyila, C. Kasparis, and B. Evans, "Robust time-domain timing and frequency synchronization for OFDM systems," *Consumer Electronics, IEEE Transactions on*, vol. 55, no. 2, pp. 391 –399, may 2009.

[46] A. Awoseyila, C. Kasparis, and B. Evans, "Improved preamble-aided timing estimation for OFDM systems," *Communications Letters, IEEE*, vol. 12, no. 11, pp. 825 –827, november 2008.

[47] J.-J. van de Beek, P. Borjesson, M.-L. Boucheret, D. Landstrom, J. Arenas, P. Odling, C. Ostberg, M. Wahlqvist, and S. Wilson, "A time and frequency synchronization scheme for multiuser OFDM," *Selected Areas in Communications, IEEE Journal on*, vol. 17, no. 11, pp. 1900 –1914, nov 1999.

[48] J.-J. van de Beek, M. Sandell, M. Isaksson, and P. Ola Borjesson, "Low-complex frame synchronization in OFDM systems," in *Universal Personal Communications. 1995. Record., 1995 Fourth IEEE International Conference on*, nov 1995, pp. 982 –986.

[49] S. D. Choi, J. M. Choi, and J. H. Lee, "An Initial Timing Offset Estimation Method for OFDM Systems in Rayleigh Fading Channel," in *Vehicular Technology Conference, 2006. VTC-2006 Fall. 2006 IEEE 64th*, sept. 2006, pp. 1 –5.

[50] H. Fang, X. Hu, and R. Xu, "An Implementation of Time and Frequency Synchronization for Carrier Interferometry of OFDM in an Underwater Acoustic Channel," in *Circuits, Communications and Systems, 2009. PACCS '09. Pacific-Asia Conference on*, may 2009, pp. 23 –26.

[51] H. Minn, V. Bhargava, and K. Letaief, "A robust timing and frequency synchronization for OFDM systems," *Wireless Communications, IEEE Transactions on*, vol. 2, no. 4, pp. 822 – 839, july 2003.

[52] G. Ren, Y. Chang, and H. Zhang, "A novel burst synchronization method for OFDM based WLAN systems," *Consumer Electronics, IEEE Transactions on*, vol. 50, no. 3, pp. 829 – 834, aug. 2004.

[53] G. Ren, Y. Chang, H. Zhang, and H. Zhang, "An Efficient Frequency Offset Estimation Method With a Large Range for Wireless OFDM Systems," *Vehicular Technology, IEEE Transactions on*, vol. 56, no. 4, pp. 1892 –1895, july 2007.

[54] G. Ren, Y. Chang, H. Zhang, and H. Zhang, "Synchronization method based on a new constant envelop preamble for OFDM systems," *Broadcasting, IEEE Transactions on*, vol. 51, no. 1, pp. 139 – 143, march 2005.

[55] D. Rudolph, "Vielträger-Modulation," TFH Berlin, Tech. Rep., 2004.

[56] K. Shi and E. Serpedin, "Coarse frame and carrier synchronization of OFDM systems: a new metric and comparison," *Wireless Communications, IEEE Transactions on*, vol. 3, no. 4, pp. 1271 – 1284, july 2004.

[57] B. Sklar, *Digital communications: fundamentals and applications*, ser. Prentice Hall Communications Engineering and Emerging Technologies Series. Prentice-Hall PTR, 2001.

[58] M. Speth, S. Fechtel, G. Fock, and H. Meyr, "Optimum receiver design for OFDM-based broadband transmission .II. A case study," *Communications, IEEE Transactions on*, vol. 49, no. 4, pp. 571 –578, apr 2001.

[59] M. Speth, S. Fechtel, G. Fock, and H. Meyr, "Broadband transmission using OFDM: system performance and receiver complexity," in *Broadband Communications, 1998. Accessing, Transmission, Networking. Proceedings. 1998 International Zurich Seminar on*, feb 1998, pp. 99 –104.

A Appendix: Derivation of the DFT Results in the OFDM Receiver

The i-th block in the time-domain is denoted as $r_i(n)$, for $0 \leq n \leq N-1$ and describes the discrete-time samples from the ADC output. In order to perform OFDM demodulation, essentially restoring the original frequency domain signal constellations from the transmitter, a N-point discrete Fourier transform (DFT) is applied to blocks of samples. The DFT output is denoted as $R_i(k)$ where the upper case symbols emphasize that the results are in the frequency domain and k represents the subcarrier index. If the DFT is applied to the i-th block $r_i(n)$ the notation looks like the following

$$R_i(k) = \sum_{n=0}^{N-1} r_i(n) e^{-j2\pi nk/N}.$$

Recalling the definition of $r_i(n)$ leads to

$$R_i(k) = \sum_{n=0}^{N-1} \left(\sum_{l=0}^{L-1} h(l) \cdot s_i(n-l) + w_i(n) \right) e^{-j2\pi nk/N},$$

which can be evaluated to

$$\begin{aligned} R_i(k) &= \sum_{n=0}^{N-1} \left(\sum_{l=0}^{L-1} h(l) \cdot s_i(n-l) e^{-j2\pi nk/N} + w_i(n) e^{-j2\pi nk/N} \right) \\ &= \sum_{n=0}^{N-1} \sum_{l=0}^{L-1} h(l) \cdot s_i(n-l) e^{-j2\pi nk/N} + \underbrace{\sum_{n=0}^{N-1} w_i(n) e^{-j2\pi nk/N}}_{W_i(k)}, \end{aligned}$$

where $W_i(k)$ already represents the Fourier transform of the noise contribution.

Further derivations incorporate the definition of the transmitted data block $s_i(n)$ to yield

$$\begin{aligned} R_i(k) &= \sum_{n=0}^{N-1} \sum_{l=0}^{L-1} \left(h(l) \cdot \frac{1}{N} \sum_{k'=-N/2}^{N/2-1} C_i(k') e^{-j2\pi(n-l)k'/N} e^{-j2\pi nk/N} \right) + W_i(k) \\ &= \frac{1}{N} \sum_{n=0}^{N-1} \sum_{l=0}^{L-1} \left(h(l) \sum_{k'=-N/2}^{N/2-1} C_i(k') e^{(j2\pi nk' - j2\pi lk' - j2\pi nk))/N} \right) + W_i(k) \end{aligned}$$

Some reordering and exchanging of sums leads to

$$\begin{aligned} R_i(k) &= \frac{1}{N}\sum_{k'=-N/2}^{N/2-1}\underbrace{\sum_{l=0}^{L-1}h(l)e^{-j2\pi lk'/N}}_{H(k')}\sum_{n=0}^{N-1}C_i(k')e^{j2\pi(k'-k)/N} + W_i(k) \\ &= \frac{1}{N}\sum_{k'=-N/2}^{N/2-1}H(k')C_i(k')\sum_{n=0}^{N-1}e^{j2\pi(k'-k)/N} + W_i(k), \end{aligned}$$

where $H(k')$ represents the Fourier transform of the channel impulse response, identified as sampled the channel transfer function (CTF) over the k'-th subcarrier. Evaluating the difference of the subcarrier indices $(k'-k)$ shows that if both are equal the sum $\sum_{n=0}^{N-1}e^{j2\pi(k'-k)/N}$ equals to N, which in turn yields to the final equation representing the DFT output for the i-th block as follows

$$\begin{aligned} R_i(k) &= \sum_{k'=-N/2}^{N/2-1}H(k')C_i(k') + W_i(k) \\ &= H(k)C_i(k) + W_i(k) \end{aligned}$$

Nomenclature

ADC	Analog to Digital Converter
AGC	Automatic Gain Control
ALU	Arithmetic Logical Unit
AS	Aircraft Station
ASIC	Application Specfic Integrated Circuit
ATM	Air Traffic Management
ATN	Aeronautical Telecommunication Network
AWGN	Additive White Gaussian Noise
BER	Bit Error Ratio
BMVIT	Federal Ministry for Transport, Innovation and Technology (german: Bundesministerium für Verkehr, Innovation und Technologie)
BPSK	Binary Phase Shift Keying
C2	Command-and-Control
CertLink	Certifiable Data Link in Time Triggered-Architecture for Remote Control of Unmanned Aerial Vehicles
CORDIC	COordinate Rotation DIgital Computer
COTS	Commercially available Off-The-Shelf
CP	Cyclic Prefix
CRB	Cramer Rao Bound
DAC	Digital to Analog Converter
DACS1	L-Band Digital Aeronautical Communications System Type 1
DC	Direct Current
DFT	Discrete Fourier Transform

DME	Distance Measuring Equipment
DPSK	Differential Phase Shift Keying
DSP	Digital Signal Processor
DUT	Device Under Test
DVB-T	Digital Video Broadcast - Terrestrial
ETSI	European Telecommunication Standards Institute
FAA	Federal Aviation Administration
FD	Frequency Domain
FDD	Frequency Division Duplex
FEC	Forward Error Correction
FFG	Austrian Research Promotion Agency (german: Österreichische Forschungsförderungsgesellschaft)
FFT	Fast Fourier Transformation
FL	Forward Link
FPGA	Field-Programmable Gate Array
FSR	Full Scale Range
GCS	Ground Control Station
GI	Guard Interval
GS	Ground Station
GUI	Graphical User Interface
HDL	Hardware Description Language
HDR	Hardware Defined Radio
HIL	Hardware-in-the-loop
IBI	Inter Block Interference
ICAO	International Civil Aviation Organization
iDFT	inverse Discrete Fourier Transform
iFFT	Inverse Fast Fourier Transformation

IFG	Inter-Frame Gap
IP	Intellectual Property
IR	Impulse Response
ISI	Inter-Symbol Interference
ISM	Industrial, Scientific and Medical Radio Band
ISR	Interrupt Service Routine
JXP	JOANNEUM Experimental Platform
LSB	least significant bit
MATLAB	MAtrix LABoratory
MAV	Micro Aerial Vehicle
MEDL	Message Descriptor List
MFM	Modified Frequency Modulation
MT	Macrotick
OFDM	Orthogonal Frequency Division Multiplex
OTA	Over-the-Air
PAPR	Peak-to-Average Power Ratio
PHY	Physical Layer
PN	Pseudo Noise
PRP	Post-Reception Phase
PSP	Pre-Send Phase
QPSK	Quadrature Phase Shift Keying
RF	Radio Frequency
RL	Reverse Link
ROM	Read Only Memory
RPA	Remotely Piloted Aircraft
RPV	Remotely Piloted Vehicle
RS	Reed Solomon

RUAV	Rotor-craft Unmanned Aerial Vehicle
SC	Sub-Carrier
SDR	Software-Defined Radio
SNR	Signal to Noise Ratio
SOF	Start of Frame
TD	Time Domain
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TP	Transmission Phase
TTP	Time-Triggered Protocol
TTTech	Time-Triggered Technology
TV	Time-Varying
TV-IR	Time Varying Impulse Response
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
USAF	United States Air Force
VTOL	Vertical Take Off and Landing
WLAN	Wireless Local Area Network