Master of Science Thesis in Communication Systems
Department of Electrical Engineering, Linköping University, 2017

Cell Acquisition and Synchronization for Unlicensed NB-IoT

Eskil Jörgensen

LINKÖPING UNIVERSITY

Master of Science Thesis in Communication Systems

Cell Acquisition and Synchronization for Unlicensed NB-IoT

Eskil Jörgensen

LiTH-ISY-EX--17/5082--SE

Supervisor: **Daniel Verenzuela**
ISY, Linköpings universitet
Y.-P. Eric Wang
Ericsson, Inc

Examiner: **Erik G. Larsson**
ISY, Linköpings universitet

Division of Communication Systems
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Abstract

Narrowband Internet-of-Things (NB-IoT) is a new wireless technology designed to support cellular networks with wide coverage for a massive number of very cheap low power user devices. Studies have been initiated for deployment of NB-IoT in unlicensed frequency bands, some of which demand the use of a frequency-hopping scheme with a short channel dwell time. In order for a device to connect to a cell, it must synchronize well within the dwell time in order to decode the frequency-hopping pattern. Due to the significant path loss, the narrow bandwidth and the device characteristics, decreasing the synchronization time is a challenge. This thesis studies different methods to decrease the synchronization time for NB-IoT without increasing the demands on the user device. The study shows how artificial fast fading can be combined with denser reference signalling in order to achieve improvements to the cell acquisition and synchronization procedure sufficient for enabling unlicensed operation of NB-IoT.

Sammanfattning

Narrowband Internet-of-Things (NB-IoT) är en ny trådlös teknik som är designad för att hantera mobilnät med vidsträckt täckning för ett massivt antal mycket billiga och strömsnåla användarenheter. Studier har inletts för att operera NB-IoT i olicensierade frekvensband, varav några kräver att frekvenshoppande spridningsspektrum, med kort uppehållstid per kanal, används. För att en användarenhet ska kunna ansluta till en basstation måste den slutföra synkronisingsfasen inom uppehållstiden, så att basstationens hoppmönster kan avkodas. På grund utav den stora signalförsvagningen, den smala bandbredden och användarenhetens egenskaper är det en stor utmaning att förkorta synkroniseringstiden. Detta examensarbete studerar olika metoder för att förkorta synkroniseringstiden i NB-IoT utan att öka kraven på användarenheten. Arbetet visar att artificiell snabb-fädning kan kombineras med tätare referenssignalering för att uppnå förbättringar i synkroniseringsprocessen som är tillräckliga för att möjliggöra operation av NB-IoT i olicensierade frekvensband.

Acknowledgments

I would like to express my thanks to the whole team at *Ericsson Research Silicon Valley* for a great time in their friendly and exciting working environment. In particular, I would like to thank the radio team for many interesting discussions and the large amount of constructive feedback they offered. My greatest thanks goes to my supervisor Eric Wang for his continuous support and genuine interest in the study. He unhesitatingly and accurately answered any questions I came up with, and those were many.

I also would like to thank my academic supervisor Daniel Verenzuela and my examiner Erik G. Larsson, for providing valuable feedback on the thesis while being flexible and patient with my own way of working. Finally, I want to thank Gunnar Bark, Ali Khayrallah and Thomas Cheng for actively supporting the initiative to make this project possible in the first place.

Santa Clara, June 2017
Eskil Jörgensen

Contents

Notation

Sets, distributions and operators

Notation	Definition
$\mathbb{R}$	The set of real numbers
$\mathbb{C}$	The set of complex numbers
$\mathbb{T}$	The circle group, $\mathbb{T} = \{z \in \mathbb{C} : \lvert z\rvert = 1\}$
S^n	The unit n-sphere, $\{\mathbf{x} \in \mathbb{R}^{n+1} : \lVert\mathbf{x}\rVert = 1\}$
$\mathbf{Gr}(r, V)$	The set of r-dimensional subspaces of V
$\mathcal{CN}(0, \Gamma)$	Circularly-symmetric normal distribution
$\mathbb{E}\{X\}$	Expectation of X
$\mathbf{Var}\{X\}$	Variance of X
$\mathbf{Cov}\{X, Y\}$	Covariance of X and Y
$\mathbf{X}^H$	Hermitian transpose of matrix $\mathbf{X}$
$\lVert\mathbf{x}\rVert_p$	p-norm of vector $\mathbf{x}$
$\lvert x\rvert$	Absolute value of scalar x
x'	Time derivative of x

Design parameters

Notation	Definition
B	Base sequence
s	Code cover
f_c	Carrier frequency
f_s	Sampling rate
α	Forgetting factor
$\rho(\tau)$	Synchronization metric
w_k	Metric summation weight
λ_G	Genie detection threshold
λ_P	Peak detection threshold
$\mathbf{W}$	Codebook matrix

Abbrevations

Abbrevation	Definition
3GPP	3rd Generation Partnership Project
5G	5th Generation Mobile Networks
BS	Base Station
CAZAC	Constant Amplitude Zero Autocorrelation Waveform
CFO	Carrier Frequency Offset
CP	Cyclic Prefix
CRO	Carrier Raster Offset
DL	Downlink
ECC	Electronic Communications Committee
ECDF	Empirical Cumulative Distribution Function
EGB	Equal Gain Beamforming
EIRP	Equivalent Isotropically Radiated Power
ERP	Equivalent Radiated Power
FAR	False Alarm Rate
FCC	Federal Communications Commission
FHSS	Frequency-hopping Spread Spectrum
GSM	Global System for Mobile Communications
ICI	Intercarrier Interference
ICS	Initial Cell Search
ISI	Intersymbol Interference
ISM	Industrial, Scientific and Medical
ITU	International Telecommunication Union
LTE	Long-Term Evolution
MAI	Multiple-Access Interference
MCL	Maximum Coupling Loss
MF	Merit Factor
MIB	Master Information Block
mMTC	Massive Machine Type Communication
MTC	Machine Type Communication
NB-IoT	Narrowband Internet of Things
NB-IoT-U	NB-IoT in Unlicensed Spectrum
NPSS	Narrowband Primary Synchronization Signal
NSSS	Narrowband Secondary Synchronization Signal
OFDM	Orthogonal Frequency-Division Multiplexing
PRB	Physical Resource Block
PSL	Peak Sidelobe Level
PST	Primary Synchronization Time
QAM	Quadrature Amplitude Modulation
RRC	Root-raised-cosine
SNR	Signal-to-Noise Ratio
SRD	Short Range Devices
UE	User Equipment
UL	Uplink

1 Introduction

This chapter will set the stage for the thesis by introducing the studied problem. The problem is first motivated by describing how the solution would help in the ongoing developments within the field of wireless networks. The purpose of the study is then declared, followed by a particularized problem formulation. The chapter ends with a precaution on the limitations of the study and an outline of the following chapters.

1.1 Motivation

During the last decade, there has been a rapid growth in the number of mobile connected devices. This growth has been projected to last for years ahead, which implies great challenges for development in wireless communication technology. In the "IMT Vision for 2020" [1], one of three major upcoming use case categories is envisioned to be the so called *massive machine type communication* (mMTC). mMTC refers to connectivity for a large number of devices with high demands on affordability, battery life and coverage — but with lower demands on throughput and latency.

Narrowband Internet of Things (NB-IoT) is a new technology that was introduced by the *3rd Generation Partnership Project* (3GPP) standards organization in its Release 13 as a candidate for enabling mMTC. A critical design feature for NB-IoT is a synchronization scheme (consisting of signaling and algorithms) that is simple enough to meet the demands on device cost and power consumption, but robust enough to perform under extreme coverage conditions.

In a typical cellular communications system such as NB-IoT, cell acquisition and synchronization is the first task a user needs to perform in order to connect to a

cell. This task consists of a few subtasks: detection of a suitable cell to connect to; coarse and fine estimation of timing, frequency and phase for this cell; and acquirement of its specific cell ID. In order to facilitate these tasks, many cellular systems employ carefully designed synchronization reference signals broadcasted by the base station of each cell. A series of studies were carried out within the 3GPP community to find such reference signals for NB-IoT, that enable robust synchronization by devices with low-end receivers. One of the main studies that led to actual standardization of synchronization reference signals is [2].

Deployment of NB-IoT in unlicensed bands (called NB-IoT-U in this thesis) might be desirable as a way to provide more spectrum at a low cost and is now being investigated in several organizations. Some of the bands that are under consideration have regulations that enforce wireless devices to use frequency hopping, limiting the dwell time on a single carrier frequency down to fractions of a second. For a NB-IoT user to connect to the network, it must be able to synchronize and detect the cell ID before it hops to a new carrier. The current synchronization scheme, although being robust and computationally efficient, have been shown to exceed this time frame for some demanded coverage conditions [3]. The synchronization time thus need to be improved for NB-IoT-U to be successful.

In order to increase the coverage of NB-IoT and to facilitate adaption for unlicensed bands, studies of enhancing the current NB-IoT synchronization design are of great interest.

1.2 Purpose

The purpose of this thesis is to aid the design of NB-IoT-U by studying the performance of various methods for initial cell acquisition and synchronization in NB-IoT. The methods under consideration mainly refers to different synchronization signals on the transmitter side and different signal processing techniques on the receiver side.

1.3 Problem Formulation

The main problem seen in today's standard is considered to be synchronization time. Decreasing this time would require the receiver to have access to more informative signals, to process current signals more effectively or to decrease demands on accuracy. Decreasing the accuracy will affect subsequent perceived *signal-to-noise ratio* (SNR) — increasing communication error rates — and is not considered an alternative in this study. More informative signals could be achieved by improving the channel, increasing signal power or by changing the reference signals at the transmitter. The latter would require an undesirable change to the current NB-IoT standard but is still a viable option. More effective processing of current signals on the other hand might require more expensive receiver devices or a greater receiver power consumption, both of which are undesirable for mMTC usage. To address these problems, the following *research questions* are stated:

- How can NB-IoT synchronization time be decreased while keeping accuracy high and receiver complexity low?
- What methods could be good candidates for NB-IoT-U?
- Would those candidates require a change of the current standard?

In order to answer these questions, we need well defined metrics. We define *primary synchronization time* (PST) as the amount of time it takes for a user device, after the start of a cell search, to successfully acquire timing and frequency information. What counts as successful is up to the receiver algorithm to decide, but this decision will affect the accuracy. To help us answer the first research question, the 90th percentile of PST among devices is set as the main metric, but *empirical cumulative distribution function* (ECDF) plots or tables of the PST will be provided. The 90th percentile is a common capability metric used for design of wireless systems to balance high robustness against low overhead. It is also a more reliable statistic under limited sample sizes, as compared to higher percentiles.

The accuracy mentioned in the same question is measured by *residual timing and frequency errors after successful synchronization*. Our metric for accuracy will be *false alarm rate*[1] (FAR), which we define as the fraction of devices having residual errors above a certain acceptable limit. The exact limit is set to make sure that communication can proceed under decent error rates and will be determined in Chapter 7. A target FAR of 5% will serve as a guideline, this also being a common trade-off between robustness and overhead.

The demand on receiver complexity will be met by choosing a baseline receiver algorithm that is known to be cheap and then modify the algorithm in ways that are guaranteed not to increase complexity significantly.

While the first question aims to study a variety of methods and compare them in a qualitative way, the second question aims to select from these methods two or three specific configurations. These configurations will be evaluated quantitatively using the specific requirements and conditions defined later in Section 4.3 and Section 8.1, with the aim of having 90% of all users meeting these requirements. The last question is easy to answer for a given method: new standardization is required only if the synchronization signals are changed.

1.4 Limitations

The thesis studies how the capability of current NB-IoT synchronization can be extended. Since the capability of a system is determined by its limits, we will restrict our study to the most challenging cases:

- Only low speed users, to increase the effect of prolonged deep fades.
- Only low SNR, corresponding to a very large path loss.

[1] False alarms are often referred to as *type I errors*.

However, due to time constraints, interference has not been modeled. Interference may have significant negative effects on performance and should be studied in future work.

While all major steps of the cell acquisition procedure will be mentioned and described to some extent, the experiments will be limited to synchronization of timing and frequency. This step is considered the most demanding, both in terms of latency and in computational resources — timing in particular.

1.5 Thesis Outline

Chapter 2 explains what synchronization is, first in a more general sense and then for the case of OFDM[2] systems. The different parts and stages of the OFDM synchronization procedure are outlined together with common ways to solve them. A few previous studies will be mentioned, some of which treat ideas that are later used in our own study.

Chapter 3 has two purposes. It first describes NB-IoT: why it was designed and how. With an overview of some important NB-IoT design features, the synchronization algorithm used in this study is then explained in more detail.

Chapter 4 introduces the concept of unlicensed spectrum, the most important bands suitable for NB-IoT-U deployment and specifies how the requirements of these bands affect our study.

Chapter 5 introduces two studied methods that are based upon modification or extension of the synchronization reference signals.

Chapter 6 introduces the third studied method, which is based upon transmit antenna diversity.

Chapter 7 specifies our experiment on several levels. It explains the simulations that are ran: their common setup, their different parameters, what methods they represent, in what order they are ran, how they depend on each other and how it all will help us answer our questions.

Chapter 8 presents the results of all the experiments described in Chapter 7. Note that the study is carried out in several phases, such that the result of one phase affect the experiment setup of the next. Each top-level section in Chapter 8 corresponds to one such phase and the dependency is explained in Chapter 7.

Chapter 9 concludes our study by discussing the results and their implications. The chapter aims to discuss to what extent the results let us answer the questions from Section 1.3 but also what other implications they might have beyond the scope of this thesis. The final section on future work lists a number of ways in which the study could be extended in order to answer our research questions with more confidence.

[2]This term will be explained in detail in Section 2.2.

2

Synchronization in OFDM Systems

As stated in Chapter 1, synchronization is a critical part of the NB-IoT design. It is however not always given a thorough treatment in undergraduate curricula. To make sure the reader is familiar with the basics, the following chapter will make a brief introduction to the problem of synchronization. It will thereafter give a quick overview of various methods that have previously been used for synchronization in OFDM systems.

2.1 Synchronization in General Terms

Before considering synchronization in the case of NB-IoT, it may be useful to think about the problem in more general terms to get a conceptual sense of its fundamentals. If we turn to the Oxford English Dictionary[1], we find *synchronization* to be described as

> *The operation or activity of two or more things at the same time or rate.*

This definition accurately captures the essence of synchronization in a diverse set of situations, ranging from traffic to music performance to digital communications. Translating this into a more formal description, we will refer to these *things* as units, s_i, numbered by an index set $I \ni i$, that together form a system $S = \{s_i : i \in I\}$. We will quantify the *operation or activity* of the respective units by an internal time state process $v_i(t)$ indexed by external[2] time $t \in \mathbb{R}$. By using this notation, we can define synchronization by means of *same time or rate* as

[1] `https://en.oxforddictionaries.com/definition/synchronization`, 2017-04-16

[2] Consider a global time reference in a non-relativistic system.

satisfying $v_i(t) \approx v_j(t)$ or $v_i'(t) \approx v_j'(t)$ for $i, j \in I$. We will now introduce a few useful ways to categorize systems based on their particular situations.

One useful way to categorize systems is whether their respective units have operations that are *linear* or *cyclic*. If they operate linearly, each time instance is unique and it makes sense to use real time variables $v \in \mathbb{R}$. If on the other hand the units operate in a cyclical manner, where their inner state repeat the same trajectory over and over, the state could then be considered equivalent over from one period to next. It may then be convenient to use a time variable that is also periodic, e.g. $v \in \mathbb{T}$, and refer to the rate as *frequency*.

Next, we can categorize systems based on the topology of the information flow between units during synchronization. In a *centralized* case, one master unit will set the pace to be followed by the complete system, much like the conductor of an orchestra. In a *decentralized* case, the system is divided into subsystems, each having its own master unit with the task of synchronizing with other subsystems or higher level master units. A third type of topology is the *distributed* case, where every unit have the same authority and adapt to surrounding units.

Systems may also be categorized by how synchronization information is passed around. In *passive* systems, units directly observe the operation of other units and use this observation for synchronization. In *active* systems, signals are communicated specifically to aid the synchronization process. This distinction is important in communication systems, since synchronization signals and payload information often compete for the exact same physical resources. In this thesis we will study a *periodic*, *centralized*, *active* system.

2.1.1 Synchronized Communication

Digital communication systems are heavily dependent on synchronization for effective operation. To illustrate why, we will use a common model with one transmitter unit sending a digital message over a waveform channel to a receiver unit. The transmitter includes a modulator that maps bits $X[n]$ onto a continuous waveform $S(t, X; \theta_T)$, where the stochastic process $\theta_T(t)$ represents the timing and frequency state of the modulator. The channel filters S according to the channel function $h(S; \theta_C)$ and stochastic channel state $\theta_C(t)$. Finally, there is noise $N(t)$ added to the received waveform. This is depicted in Figure 2.1.

The task of the receiver is to map the received signal $Y(t, S; \theta_C, N) = h(S; \theta_C) + N(t)$ back to data bit estimates $\hat{X}[n]$ as accurately as possible. If the transmitter design and channel statistics are completely known, all uncertainty can be expressed as a joint probability $p(X, \theta_T, \theta_C, N)$ and the channel capacity[3] C is given as

$$C = \sup_{p_X(x)} I(X; Y). \tag{2.1}$$

[3] In this thesis, *channel capacity* refers to the theoretical maximal achievable information rate over a channel and *cell capacity* to the maximal number of simultaneous users a cell can serve.

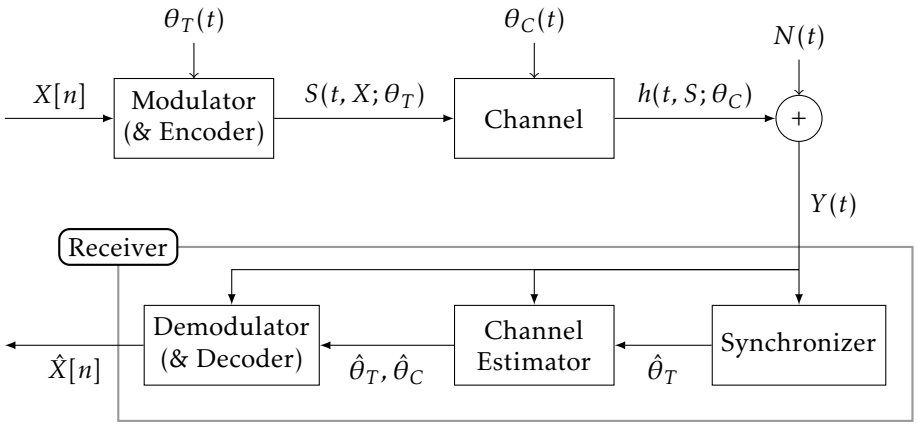

Figure 2.1: *Block model of a communication link*

C can be reached by using a sophisticated coding scheme with redundancy added accordingly. This is depicted in part a) of Figure 2.2.

As it turns out, tackling this joint probability directly is typically intractable, with any coding scheme approaching channel capacity being far too complex. The way to mitigate this is to use a modular receiver design, with each module handling one source of randomness. A synchronizer for θ_T, a channel estimator for θ_C and a decoder for N. This subdivision of the receiver is shown in the lower part of Figure 2.1 and has a decoupling effect on the distribution, simplifying it significantly. Each module has a refining role in the signal processing pipeline, ideally removing additional entropy introduced by its corresponding randomness source.

So how would one estimate the parameters $\theta = \{\theta_T, \theta_C\}$? There are some techniques that exploit redundancies inherent to the transmission scheme — and we will discuss this briefly in Section 2.4 — but for links with high spectral efficiency, an overwhelming fraction of the received entropy will be caused by X, impeding inferences made on θ. A somewhat more robust solution would be to simply introduce redundancy directly on the actual symbols. This is usually done by dividing the transmission frame into pure data symbols and reference symbols (also known as pilot- or training sequences), as shown in Figure 2.2 b). The reference symbols are used to estimate θ and compensate its effects on the signal. A system that uses reference signals would be an *active* system in the sense described earlier in this section.

By having a confident estimation of θ, we are in the situation where only the additive noise provides randomness on top of X, and this can conveniently be handled by channel coding with redundancy introduced by the encoder. The partitioning of the receiver into simple modules is of course a design compromise causing some residual noise and errors, but the advantages of increased simplicity far outweigh this problem.

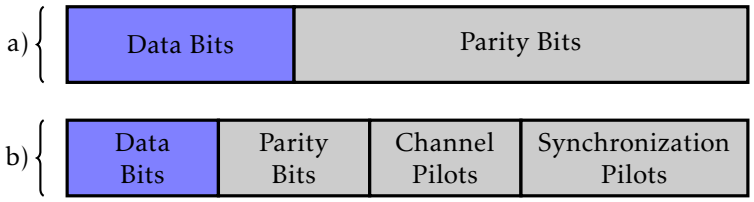

Figure 2.2: *Frame structure using: a) optimal codec b) modular design*

Replacing data symbols by pilots will reduce the immediate data rate, but also reduce the error rate, which means less redundancy requirements in the codec. Instead of being set by theoretical calculations, the pilot rate is usually set by considering the specific application at hand and its specific requirements on throughput, latency and reliability. In the next section, we will describe the modulation- and multiplexing scheme used in NB-IoT in order to enable more detailed discussions regarding synchronization schemes.

2.2 Orthogonal Frequency-Division Multiplexing

Many modern high-data-rate wireless communication systems — including NB-IoT, *Long-Term Evolution* (LTE) and WiFi — are based on *orthogonal frequency-division multiplexing* (OFDM). OFDM is a scheme for modulation and multiplexing of digital data, hence, the usage of OFDM has important implications on synchronization design.

The idea of OFDM is to use a relatively low symbol rate and then do *quadrature amplitude modulation* (QAM) on the harmonics of the symbol rate coherently. By doing this, one effectively get *frequency division multiplexing* with orthogonal *subcarriers*. Since the harmonics are integer multiples of the symbol rate there is a constant *subcarrier spacing*, which creates a rectangular two-dimensional grid of resources, delimited by symbol time on one axis and its inverse (subcarrier spacing) on the other. Every point on the grid can hold one QAM symbol, which creates an opportunity to schedule data transmissions in both time and frequency. Figure 2.4 illustrates time-frequency grids.

By keeping the symbol rate low, the channel delay spread will not destroy the symbol completely, and we can even afford a guard interval on each symbol to prevent *intersymbol interference* (ISI). The most common way to allocate the guard interval is to use a so called *cyclic prefix* (CP) where the last samples are extended cyclically to prefix the symbol. Another way of interpreting the low symbol rate is as a low subcarrier spacing, effectively creating a set of channels with a bandwidth narrow enough to experience flat fading. This is a main reason for OFDM being widely used in fading wireless environments. The maximum bandwidth under which the channel can practically be considered flat is called *coherence bandwidth* and is approximately proportional to the delay spread. All the different parameters defining the OFDM waveform — including subcarrier

spacing, CP length and larger scale structures for allocation of resources — can be set in a variety of ways. A specific setting of these parameters is called a *numerology*[4].

As it turns out, the scheme described above is equivalent to a Fourier series. The fact that a sampled Fourier series is equal to an *inverse discrete Fourier transform* of the same coefficients enables a convenient digital implementation of the baseband modulation. The demodulation procedure can similarly be based on a discrete Fourier transform, complemented by the usual downconversion and symbol detection used in a QAM system. A simplified view of an OFDM receiver can be seen in Figure 2.3.

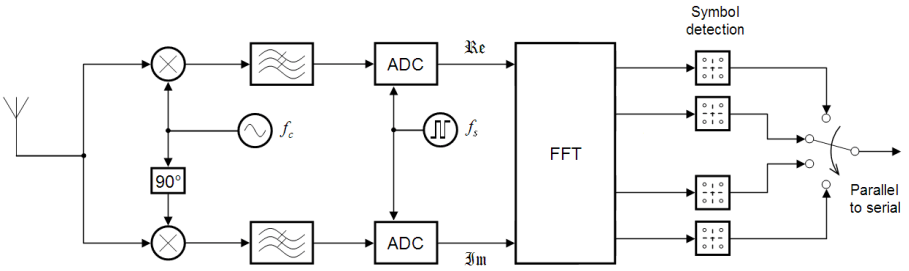

Figure 2.3: *OFDM demodulator with local oscillator (f_c) and sample clock (f_s) separated. [Source: Wikimedia Commons (remix)]*

2.2.1 OFDM Synchronization Tasks

Even though OFDM is a compelling scheme for fading channels, it is largely dependent on accurate synchronization. The effects of deficient synchronization are discussed in Section 2.2.2. But first, the main tasks required for synchronization in OFDM are described.

Symbol and frame timing

To demodulate a received signal into symbols, the receiver needs to know the *symbol timing*, i.e. the time instance marking the start of a new symbol. Otherwise, the information of one symbol will distort the demodulating of another, causing ISI. Due to the lack of an exact common time reference (and the unknown *time delay*), the symbol timing information has to be extracted from the signal itself.

In addition, the receiver also needs information on which symbol constitutes the start of a new frame. Since transmission schemes typically employ a hierarchical framing structure with multiple types of data, the index of the symbol marking a complete frame period (i.e. the *frame timing*) must be known.

[4]Not to be confused with mystical beliefs.

Carrier Frequency Offset

Similarly to timing, also frequency information has to be learned. Assuming prior knowledge in the receiver of the carrier frequency the transmitter intend to use, f_c, there might still be a frequency discrepancy due to Doppler shift and to local oscillator instability on both sides. This is called *carrier frequency offset* (CFO). With c being the speed of light, d the distance between the units and LO_{Tx} and LO_{Rx} the local oscillator frequency errors (usually expressed in ppm) of the transmitter and receiver respectively, the aggregate CFO effect will be given as the scaling factor

$$\phi = \frac{1 + LO_{Tx}}{1 + LO_{Rx}} \cdot \frac{c}{c + d'}. \tag{2.2}$$

This scaling can be interpreted as an area-preserving scaling of the axes in the time-frequency grid, illustrated by Figure 2.4. This factor is usually very small and have a negligible effect on a single symbol. However, CFO will affect the down-conversion, leading to a baseband frequency shift proportional to f_c, and f_c is often much larger than the bandwidth. This frequency shift will cause *inter-carrier interference* (ICI) that reduces carrier orthogonality. Also, time-drift will add up over many symbols, again causing ISI.

Note that the effect we describe here is usually more accurately ascribed to the *sampling clock error*. We assume here that the sampling frequency, f_s, and local oscillator frequency, f_c, originate from the same source and abuse the term CFO (as commonly done in literature). Pure frequency offset will instead be referred to as *carrier raster offset* (CRO).

Carrier Raster Offset

Prior knowledge of f_c will not always be complete. For synchronization to be feasible, the prior knowledge should include a reasonably small set of frequencies to search through: a *search raster*. By searching through this raster, one should be guaranteed to hit the correct frequency only by a small error, the CRO. The CRO has the effect on an actual *offset*, i.e. a shift in frequency.

While CFO causes time drift, CRO does not. But synchronization schemes are typically designed to correct the total perceived frequency error f_e = CFO + CRO. This leads to a problem when compensating for time drift. If for example CFO >> CRO, the system can assume the time drift being proportional to f_e and the CRO will then lead to a residual time-drift due to overcompensation.

Channel estimation and equalization

As described in Section 2.1.1, channel estimation and equalization may be done after synchronization. This procedure is in some sense analogous to synchronization, with channel reference symbols providing redundancy for parameter estimation. Channel *coherence time* is defined as the maximum time period under which the channel can practically be considered time-invariant. By doing a channel estimation at an interval considerably smaller than the channel coher-

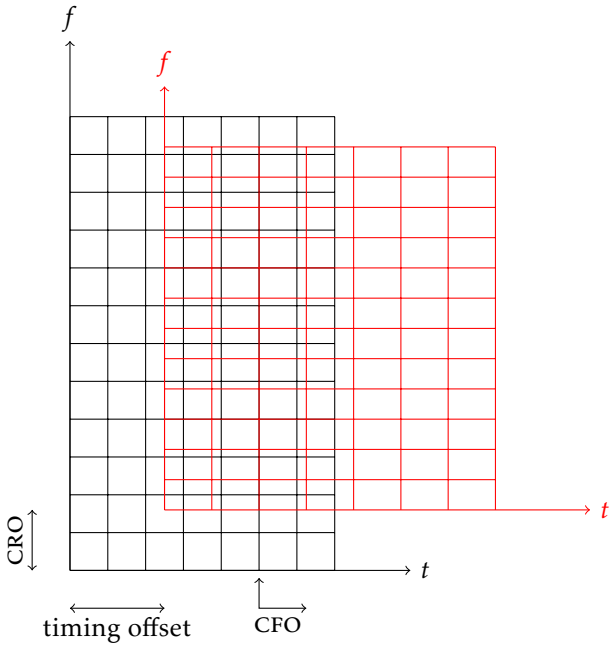

Figure 2.4: *Timing offset, CFO and CRO. The scaling of the axes preserves area and will for high carrier frequencies be perceived as another constant offset.*

ence time, the coding scheme will not need to cover up for any entropy induced on the received signal by the channel dynamics.

One important issue that we have not yet mentioned is that of carrier phase, which can vary due to instabilities in the transmitter or receiver oscillators. We will consider this effect as part of the channel and leave the equalizer to compensate for the effect.

2.2.2 Effects of Deficient Synchronization

We have mentioned synchronization tasks, with the most important effects being ISI and ICI. These are obviously to be avoided and we will here give an example of how their effects can be quantified. The effects are measured by how much the perceived SNR gets affected.

Note that in many multi-user systems, OFDM is used for multiple access, thus the effects of poor synchronization of one user are not only degrading its performance, but also affecting other users, which is called *multiple-access interference* (MAI). In this case, synchronization is not only a matter of increasing channel capacity for ones own link, which makes the discussions in Section 2.1.1 about redundancy rather more complicated.

OFDM Timing errors

SNR loss, γ, can be calculated as a function of the timing error, $\Delta\tau$, from four quantities: signal power $\mathbb{E}\{|S(t)|^2\}$; noise power σ_N^2; interference from ISI and ICI, modeled as a zero-mean variable with variance $\sigma_I^2(\Delta\tau)$; and an attenuation factor, $\alpha^2(\Delta\tau)$, representing the fact that part of the signal will be outside the sampling window. By assuming that the channel and frequency is known, and normalizing the channel impulse, we get [4] that

$$\gamma(\Delta\tau) := \frac{\mathrm{SNR}^{(ideal)}}{\mathrm{SNR}^{(real)}} = \frac{\mathbb{E}\left\{|S(t)|^2\right\}/\sigma_N^2}{\mathbb{E}\left\{|S(t)|^2\right\}\alpha^2(\Delta\tau)/[\sigma_N^2+\sigma_I^2(\Delta\tau)]} = \frac{1}{\alpha^2(\Delta\tau)}\left[1+\frac{\sigma_I^2(\Delta\tau)}{\sigma_N^2}\right]. \tag{2.3}$$

OFDM Frequency errors

Similarly, when the channel and timing is known, the SNR loss can be calculated from the frequency error, ϵ, as follows. By normalizing the channel impulse, we have [4]

$$\gamma(\epsilon) := \frac{\mathrm{SNR}^{(ideal)}}{\mathrm{SNR}^{(real)}} = \frac{1}{|f_n(\epsilon)|^2}\left\{1+\frac{\mathbb{E}\left\{|S(t)|^2\right\}}{\sigma_N^2}[1-|f_n(\epsilon)|^2]\right\} \approx 1+\frac{\mathbb{E}\left\{|S(t)|^2\right\}}{3\sigma_N^2}(\pi\epsilon)^2 \tag{2.4}$$

where n is the number of available subcarriers and

$$f_n(\epsilon) = \frac{\sin(\pi\epsilon)}{n\sin(\pi\epsilon/n)} e^{j\pi\epsilon(n-1)/n}. \tag{2.5}$$

2.3 Cellular OFDM and Initial Cell Search

In cellular networks, connectivity is provided to all *user equipment*s (UEs) in a cell by a *base station* (BS). Transmission from BS to UE is referred to as *downlink* (DL) and the opposite direction is called *uplink* (UL). The BS often coordinates and schedules traffic for a large number of UEs within the cell and makes sure that they communicate efficiently with users or networks outside the cell. It is therefore appropriate to have each UE adapt to the rest of the system, with the BS acting as a *centralized* source of timing by providing DL reference symbols.

The DL synchronization will provide everything the UE needs for reception, but due to Doppler shift and time delay, UL will still suffer from MAI, unless some UL synchronization takes place. After DL synchronization, the MAI can be resolved either by sophisticated signal processing at the BS, or by providing feedback of Doppler shift and time delay on the DL, to have the UE compensate.

When a new UE wants to connect to a cell, it first needs to detect the cell and any specific system information before attempting to connect. It is also important that coarse synchronization happens on DL before UL, to minimize the MAI caused by the UE. This first procedure is called *cell acquisition* or *initial cell search* (ICS). After ICS is done, the UE can move on to the *random access* procedure. When successful communication has started, the UE needs to do fine synchronization continuously, referred to as *tracking*. Tracking can be based upon adaptive signal processing techniques but may also rely on the same algorithm used for ICS. We will only treat ICS. Following is an exemplary outline of the ICS procedure:

1. The UE powers on.
2. Check SIM-card for band or raster information.
3. Scan entire raster for cell power profile.
4. Try to synchronize on loud candidate frequencies:
   a) Frame or symbol timing (our main focus)
   b) Frequency estimation
   c) Frame numbering
   d) Channel and phase estimation
   e) Cell ID and system information
5. Do random access procedure (with UL synchronization).
6. Request scheduling.

7. Start communication and do tracking continuously.

2.4 OFDM Downlink Methods

The core part of timing and frequency synchronization consists of two steps. The first could simply be stated as the estimation of *shift* in time and frequency of the DL signal. As we know from Fourier analysis, a shift in one domain corresponds to a *rotation* in the other. Consequently, this step can be done by estimating shift, rotation or a combination of both. The general problem, given received signal Y, can be expressed as

$$(\hat{t}, \hat{f}) = \underset{(t,f)}{\operatorname{argmax}} L_1(Y, t, f) \tag{2.6}$$

where the cost function L_1 ideally should be based on high level metrics such as minimization of PST or maximization of perceived SNR, but will in practical cases be based on common parameter estimators like *maximum likelihood* or *linear minimum mean squared error*, or on other heuristics. Figure 2.5 illustrates an example of an estimation metric as a function of timing error.

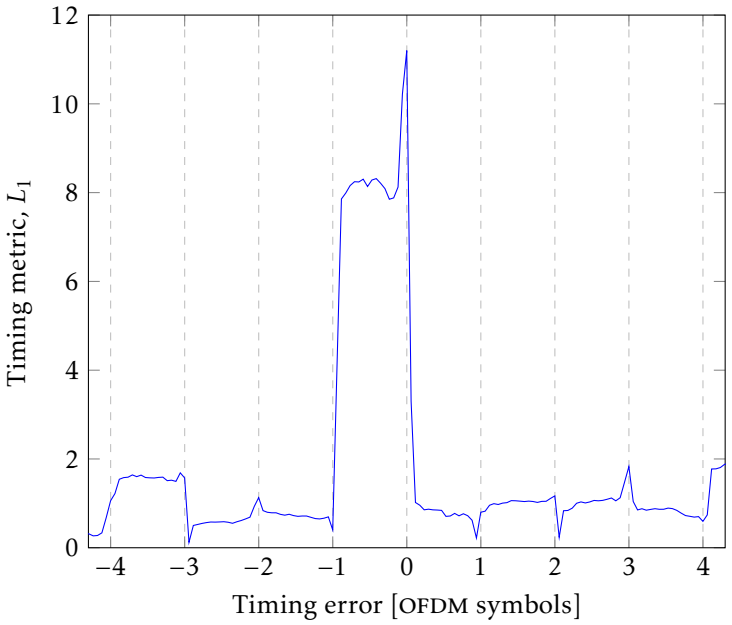

Figure 2.5: *Example of a timing metric. The data is taken from the prestudy and the cost function is $|\rho|$ (defined in the next chapter).*

The second part is a matter of detection. Due to factors such as power consumption and MAI, it may not be worthwhile to attempt communication unless the previous estimation is accurate enough. Thus there should be some detection

threshold, λ, determining whether the DL signal estimation is confident enough. The confidence metric, L_2, will typically be a function of a subset of L_1 and its derivatives:

$$L_2\left(\left\{\frac{\partial^{m+n}L_1}{\partial t^m\,\partial f^n}\right\}\right) \lessgtr \lambda \tag{2.7}$$

Detection is often treated quite easily by adjusting λ to some given target FAR. The more intricate problem is that of estimation. The estimation in equation (2.6) can be simplified if the time can be decoupled from frequency. This is analogous to the modular receiver design but take it one step further, and similarly suffer from some loss in maximal achievable performance. This cost of this loss could be made insignificant in comparison with the reduced cost by the much less complex design.

Pilot-free approaches

We mentioned earlier that synchronization methods may or may not be based on pilot signals. Pilot-free methods need to exploit some characteristic of the waveform design. Examples of this include searching for power profiles or utilizing silent periods such as guard intervals. Utilization of recurring silence imply a reliance on a constant stream of data packets or other statically allocated symbols to be transmitted, which may not always be provided. Other pilot free methods exist, for example utilizing redundancy in the CP of OFDM, which is often significantly longer than delay spread and therefore carries some redundancy. The pitfalls of this method are several: it also relies on continuous transmissions; the CP is only a fraction of a symbol, thus providing only a small SNR gain; occasional long channel delay spreads may impede potential gains from this method. Pilot-free methods have been shown to perform at an unsatisfactorily low accuracy [4].

The acquisition procedure, as outlined in Section 2.3, consists of several steps of increasing granularity. Pilot-free methods may have a place early in this process, where a crude measurement such as *received signal strength indication* can provide guidance. So the more compelling alternative is to use pilot-based methods. There are different varieties of these and some are introduced below for specific phases of the synchronization procedure.

Coarse timing

Since, in OFDM systems, timing estimation is typically the first thing that happens (Section 2.3), any timing method should be robust enough to handle a relatively large CFO. It should also be robust to the unknown channel. While very precise timing can be hard to achieve at this stage, a coarse estimate is easier. The coarse timing estimate can be used to do frequency synchronization before a fine timing is finally done. The distortion caused by the CFO and the channel can make it unfruitful to do cross-correlation of the received signal with a memorized copy of the pilot. One solution to this is to use as pilot a sequence, B, duplicated in time, and instead detect it by a sliding auto-correlation window [5]. If each copy of B is

much longer than the delay spread (as should be guaranteed by the CP), the channel distortion on each copy, B, should be similar, yielding only a negligible effect on the auto-correlation. The effect of the CFO will be a complex time rotation on the signal, which by using auto-correlation will be seen as nothing but a complex rotation between the symbols. The magnitude of the sliding auto-correlation will peak when the window matches the pilot, if the pilot is large enough to prohibit random data from creating false alarms. A refined version of this uses more than two copies, $[B, B, -B, B]$, to sharpen the peak [6]. The binary sequence of copies is called code cover in this thesis. What constitutes a good code cover will be discussed later.

Fine timing and tracking

At the end of the ICS procedure, more accurate timing can be achieved for example by using a higher sampling rate. Sub-sample level accuracy can be incorporated into the channel estimate. Due to oscillator instabilities and varying Doppler shift — but more importantly, residual time drift — the fine timing then has to be tracked. The time drift can be captured explicitly by tracking the varying time delay of the *channel impulse response* maximal taps [4].

Frequency acquisition

Many methods for frequency acquisition in principle rely on measuring the CFO induced phase shift between symbols. Since phase is periodic, only the phase shift modulo 2π can be measured. This is equivalent to determine the frequency offset from the nearest sub-carrier and is called *fractional frequency* estimation. To get a complete frequency estimate the correct subcarrier must be found, which is called *integer frequency*. Finding the fractional frequency is by nature a continuous problem, while integer frequency can be found by hypothesis testing on a number of plausible subcarriers. [4]

An example of how fractional frequency can be found is given by the previously described timing method that uses two copies of B [5]. Here, the phase between the copies is utilized. In the same paper, a method is described for finding integer frequency by using an extended reference sequence, $[B, B, P_1, P_2]$, where P_1 and P_2 are pseudo-noise reference sequences. After fractional frequency is found and compensated for, a discrete Fourier transform of the signal is done followed by a cross-correlation in frequency domain with memorized copies of P_1 and P_2. In this way, the integer frequency can be found. A similar approach using a smaller overhead is introduced in [7]. This technique is part of the synchronization algorithm used in this study.

Frequency tracking

The need for frequency tracking stems from same reason as we do time tracking. Usually, it suffices to repeat the acquisition procedure. Other classical approaches that are used for this are based upon *phase-locked loops*, which can be deployed using error signals in time or frequency domain.

2.4.1 Pilot Characteristics

How should one choose the B from the earlier part? This question is central to both synchronization design generally and also to this thesis specifically. As mentioned earlier, synchronization corresponds to estimating shift or rotation in time and frequency. Shifts are found by correlation, which requires sequences with good *auto-correlation properties*, i.e. a sharp peak at zero and low side-lobes for non-zero shifts of the auto-correlation function. Rotations conversely are easily found on sequences with nearly constant amplitude. Not surprisingly, constant amplitude translates to the described good auto-correlation properties by Fourier transform. Thus, for pilot sequences to facilitate synchronization of time and frequency, they need to have zero auto-correlation and constant amplitude in both domains. Such sequences are called *constant amplitude zero auto-correlation waveform* (CAZAC).

3

Narrowband Internet of Things

With the previous chapter introducing some aspects of synchronization for OFDM systems, we will now have a look at what this means for Narrowband Internet of Things. This chapter will introduce the purpose and capabilities of NB-IoT as well as its design features and limitations relevant to our study. The first section expands on Chapter 1 and gives a background of the motivation for creating NB-IoT. It is followed by a short description of parts of the air interface, with emphasis of features relevant to synchronization. We will then describe in detail how the synchronization system of NB-IoT is designed and it how it relates to the concepts in the previous chapter, encompassing both the transmitter and receiver side of the DL.

3.1 LTE, 5G and mMTC

Cellular communication is an example of a technology that has been very successful in the last decades, with recent standards such as LTE enabling the outstanding use today of hundreds of millions of smartphones with services such as high-definition video calls between continents. In the coming years however, the number of connected devices as well as the number of use cases is expected to increase massively, much driven by so called *machine type communication* (MTC). This requires a continuous fast paced research and development of wireless networks, as today's (although capable) networks will be insufficient in a few years.

There is a widespread consensus within the telecom industry that the *5th Generation Mobile Networks* (5G) devices can suitably be divided into three major kinds of applications. These were defined [1] by *International Telecommuni-*

cation Union (ITU) as being: *enhanced mobile broadband*, which is the natural improvement of today's mobile broadband with higher data rates supporting video streaming and virtual reality; massive machine type communication (mMTC), which refers to a large quantity of low-end devices such as sensor- and actuator networks, meters (water, gas, electric, or parking), home automation, etc; and *ultra-reliable low latency communications* which refers to highly critical links for applications such as remote surgery, vehicular communication systems and cloud-based control of drones and robots. Their requirements are summarized in Figure 3.1.

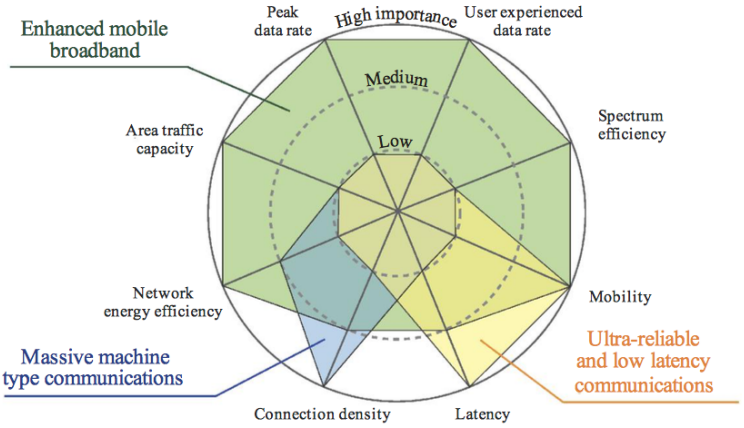

Figure 3.1: *The importance of key capabilities in different usage scenarios. [Source: IMT Vision, ITU]*

Part of 3GPPs 5G transition strategy includes a new radio interface called *5G New Radio*, which presumably will meet all the requirements of 5G. The details of this development are not yet clear, but what is certain is that the transition to 5G will depend on deployments of *5G New Radio* in tandem with extensions of LTE (as of now called LTE Advanced Pro or 4.5G).

This thesis concerns only the mMTC part of 5G. LTE has been developed primarily with mobile broadband in mind, and only to some extent for MTC. Recent work by 3GPP for MTC include *EC-GSM-IoT* and the LTE-MTC[1] cost reductions introduced in *release 13*. To more convincingly meet the needs of 5G mMTC, a new technology was introduced in release 13, namely NB-IoT, with commercial products already on the market as of early 2017. NB-IoT is somewhat based on LTE, sharing the numerologies for the OFDM DL, as well as tail-biting convolutional codes with interleaving and rate matching from LTE [8]. The design, however, is modified and streamlined to meet the following requirements [9]:

- High cell capacity: >52 500 devices
- Low cost: <$5 per device

[1] Also known as LTE-M or eMTC, but more formally referred to as LTE CatM1/M2

- Low power consumption: >10 years with a 5 Wh battery
- Improved coverage: 20 dB better than GSM[2]
- Relaxed device data rate requirements
- Relaxed latency requirements

3.2 Downlink Physical Layer

The complete air-interface of NB-IoT is specified in [10]. The first version only supports half-duplex frequency-division duplex. By using OFDM, a time-frequency grid as illustrated in Figure 2.4 naturally arises. The parameters and dimensions are here set equal to the LTE numerology, but only one 180 kHz *physical resource block* (PRB) will be used, as shown in Figure 3.4. Only the *normal cyclic prefix* of nine samples is used, resulting in 128 + 9 = 137 samples per symbol.

By using only one PRB, the carrier will fit well within a 200 kHz band[3]. This carrier is meant to be deployed in one of three different deployment modes as shown in Figure 3.2: LTE *inband*, LTE *guardband* or *standalone*. Because of the design similarities, LTE inband deployment is possible with the BS incorporating the NB-IoT PRB into its entire carrier and the UE being agnostic to the surrounding signals. In the standalone case, substantial pulse shaping might be needed to adhere to GSM emission regulations. To cope with the different circumstances, these different deployment modes might use different levels of DL power boosting.

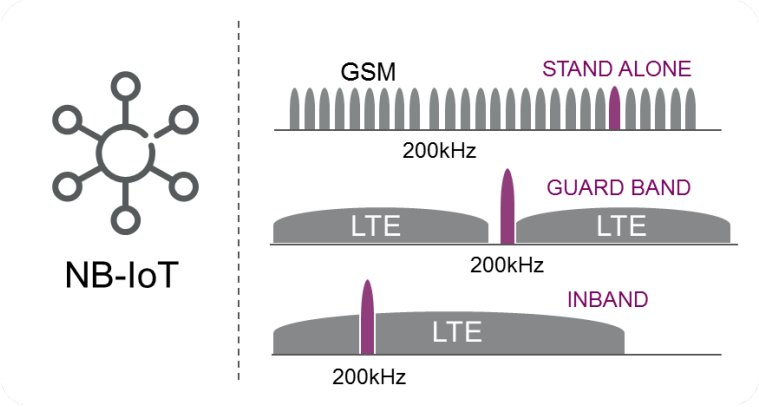

Figure 3.2: *NB-IoT deployment modes. [Source: Ericsson Research Blog]*

The CRO discussed in Section 2.2.1 is present in NB-IoT, and the effects are central for the synchronization. In the ICS, the UE is required only to search for a carrier on a 100 kHz raster spacing. The deployment of NB-IoT PRBs is done with this in

[2] *Global System for Mobile Communications*

[3] The exact carrier footprint can be adapted to different deployments by varying the pulse shaping filter.

Table 3.1: *NB-IoT physical channels and signals.*

Channel	Main purposes
Broadcast channel	System information, frame number
Control channel	Scheduling, acknowledgements, paging, etc
Shared channel	Higher level payload data, etc
Reference signals	Demodulation phase reference
NPSS	Frame boundary, frequency
NSSS	Frame number, cell ID

mind. For inband and guardband deployments however, the PRB must be aligned within the LTE grid. By deploying NB-IoT on a wisely chosen subset of LTE PRBs, the NB-IoT carrier can be placed ±2.5 kHz or ±7.5 kHz from nearest frequency raster point (i.e. a multiple of 100 kHz). For standalone deployments, the PRB can be placed exactly on the raster. The prior offset uncertainty in the UE of up to 7.5 kHz constitutes the CRO.

3.2.1 Physical Channels and Signals

In 3GPP standards, a *physical channel* is a set of physical resources in the time-frequency grid dedicated for transmission of some specific information, with specifications on the transmission format. In NB-IoT, the physical channels are based on an essential subset of the LTE physical channels. An initial capital *N* have been added to their abbreviations to emphasize that they are specific to NB-IoT. Even though the physical channel specifications are changed from LTE, their respective purposes remain mostly unchanged, as summarized in Table 3.1. As we will see, this thesis concerns only the synchronization signals of NB-IoT: NPSS (*Narrowband Primary Synchronization Signal*) and NSSS (*Narrowband Secondary Synchronization Signal*) — primarily[4] NPSS.

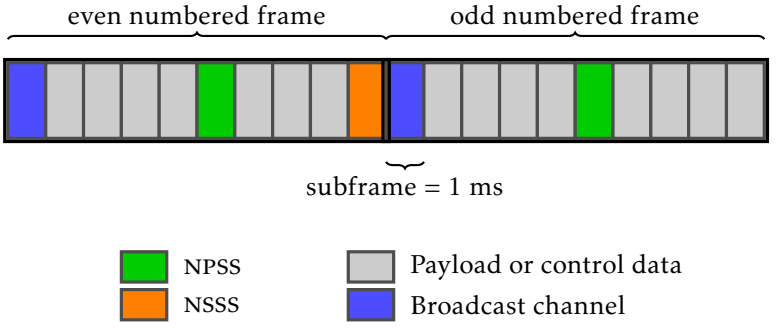

Figure 3.3: *NB-IoT frame structure*

Figure 3.3 illustrates the basic time-domain structure. Time is divided up into

[4]No pun intended.

10 ms blocks called *frames*, which in turn are subdivided into *subframes*. Each subframe is allocated to one channel or signal at a time. One exception to this arises in inband deployments, where the control channel of LTE is allocated to the first three symbols of each subframe. Since the UE cannot know the deployment mode before ICS, these symbols are left empty in the subframes of NPSS, NSSS and the broadcast channel for all NB-IoT deployment modes. In addition to this, LTE reference signals are spread out in all inband subframes. The small rectangles in Figure 3.4 are called *resource elements* and each contains one coefficient of an OFDM symbol. The dimensions of a resource element are given by the subcarrier spacing, 15 kHz. Figure 3.4 illustrates how the LTE reference signals punctures the NPSS. The puncturing does only happen in inband deployments and can in this case be treated as noise on the synchronization signals.

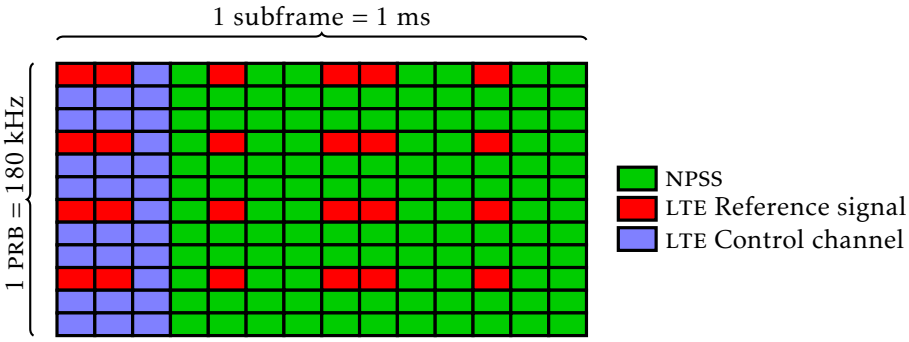

Figure 3.4: *NPSS resource mapping, inband deployment. Each row represents one subcarrier.*

3.3 Synchronization in NB-IoT

Due to the challenges associated with wide coverage areas and narrow frequency bands, the initial procedures such as synchronization and random access are crucial to the system design in NB-IoT. They are the most power hungry, the hardest and also the limiting factor that will determine the performance of the system.

The challenge stems from a number of reasons: First of all mMTC UEs may be deployed in places with very poor coverage. Secondly, due to the low cost of UEs, low quality components should be assumed, with inaccurate crystal oscillators causing CFO of as much as 20 ppm. At the frequency bands around 900 MHz, which we consider for NB-IoT-U (see Chapter 4), this corresponds to a CFO of 18 kHz. Thirdly, the UE does not know the CRO before ICS, which has to be accounted for by the synchronization. Adding the CFO and the CRO, we see that the synchronization has to deal with an initial frequency disparity equal to as much as two subcarriers. For this reason, the synchronization is designed to be very robust. Robust design can easily lead to high computational complexity, so active efforts has to be taken in order to keep the algorithm simple.

As mentioned earlier, both NPSS and NSSS take up one subframe. The actual signals are composed of sophisticated sequences. The NPSS has a hierarchical structure, consisting of a *base sequence* and a *code cover*. This structure is crucial for our study and will be described below.

3.3.1 Base Sequence

Wireless communication standards are full of reference signals of different kinds and many of them are CAZAC. One such type of sequences that have seen successful use in the third and fourth generation wireless technology is *Zadoff-Chu sequences* [11]. They are defined as

$$d_u(n) = \exp\left(-j\frac{\pi u n(n+1+2q)}{N_{ZC}}\right) \tag{3.1}$$

where the parameter u is called root index, q is the shift of the sequence and N_{ZC} is the length of the sequence. In NB-IoT, the parameters are set to $q = 0$, $u = 5$ and $N_{ZC} = 11$. The length-11 sequence is mapped to each OFDM symbol of the NPSS. So if there are twelwe subcarriers, why not use $N_{ZC} = 12$? The answer comes from the fact that if N_{ZC} is prime, then the time-domain sequence will also be Zadoff-Chu, and as it turns out also periodic and symmetric. Periodicity allows for smoother time-domain concatenation of the OFDM symbols and the symmetry can be exploited for cheaper receiver signal processing.

As can be seen from n appearing twice in the equation (3.1), the sequence is chirp-like (i.e. having an increasing rate of change), which is related to it being CAZAC. One good reason Zadoff-Chu sequences has been used in many standards is that the cross-correlation is zero for sequences of different shifts, q, and constant for different roots, u. For NB-IoT, the important aspect is not the auto-correlation itself, but the fact that it has constant amplitude in time (improving the cubic metric) and frequency domain (limiting spurious emissions).

3.3.2 Code Cover

The second layer of the NPSS structure is the code cover. It is the same kind of code cover that is described in Section 2.4, with B here being the base sequence. Each NPSS OFDM symbol consists of B or $-B$, as shown in Figure 3.5. The NPSS code cover is ([2])

$$s = [+ + + + - - + + + - +], \tag{3.2}$$

so the NPSS block, H, in Figure 3.5 can be expressed as

$$H_{n,k} = d_5(n) \cdot s(k), \tag{3.3}$$

where n and k index rows and columns respectively.

3.3.3 Reciever NPSS Processing

Regardless of its properties, the performance of a particular NPSS will only be as good as the UEs ability to process it. Therefore, the NPSS we described here, was

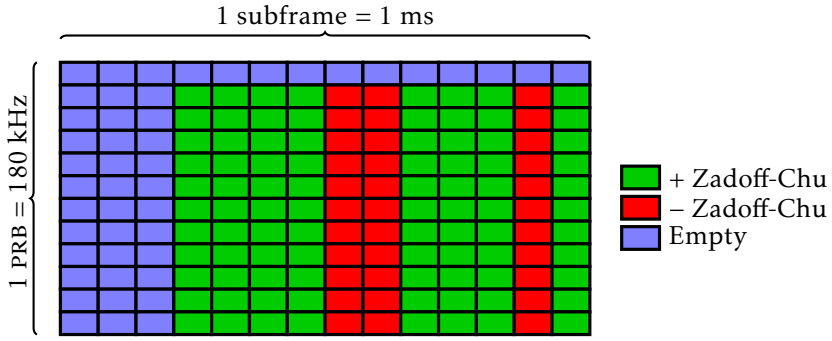

Figure 3.5: *NB-IoT NPSS structure, standalone deployment.*

proposed to 3GPP together with a corresponding receiver algorithm [12]. The combination of the signal and the algorithm was shown to be robust and computationally cheap [12]. The algorithm used in this report is based on this 3GPP contribution and its revisited version [2], with only minor changes.

The principles behind the algorithm are based upon methods we described in Chapter 2. Timing estimation is done akin to [6] and the frequency estimation done akin to [7]. These techniques are in this algorithm combined using a single metric function for timing and frequency. The signal processing steps of the algorithm are outlined below, similarly to the description in [2].

1. Downsampling

The standard sampling frequency in NB-IoT is 1.92 MHz, but the NPSS processing uses a reduced sampling rate of 240 kHz, for computational simplicity. The algorithm will process one 10 ms frame at a time and try to find among the samples $\tau \in [1, 2, \ldots, 2400]$, the sample, τ_0, where the NPSS starts.

2. Autocorrelation

For each new sample, τ, the vector $\mathbf{r}(\tau) = [\mathbf{r}_1(\tau), \mathbf{r}_2(\tau), \ldots, \mathbf{r}_K(\tau)]$ is formed from the most recent samples[5]. K is the length of the code cover. $K = 11$ for NB-IoT. Each $\mathbf{r}_i(\tau)$ has the length of one symbol and $\mathbf{r}(\tau)$ the length of the NPSS. For $\tau = \tau_0$, $\mathbf{r}(\tau)$ will match the NPSS, a fact that is the basis for the following processing. The first step is to reapply the code cover, s, and do cross-correlation between the symbols:

$$A_k(\tau) = \frac{1}{K-k} \sum_{m=1}^{K-k} (s(m+k)\,\mathbf{r}_{m+k}(\tau))\,(s(m)\,\mathbf{r}_m(\tau))^H, \qquad k = 0, 1, 2, 3, 4 \tag{3.4}$$

k is limited to 4 to keep processing complexity low. Thanks to auto-correlation properties of s, the magnitude of $A_k(\tau)$ will peak for $\tau = \tau_0$. To see this, note that for $\tau = \tau_0$, the reapplication of s will produce a sequence of K identical symbols

[5] Note that samples from the previous frame might be needed.

(not counting channel effects and noise), while for other values of τ, the code cover will not match the signal and instead create a new pseudo-random pattern. When $\tau = \tau_0$, the CFO induced phase rotation, θ, between adjacent symbols will be given by

$$\mathbb{E}\{A_k(\tau_0)\} \propto e^{jk\theta}. \tag{3.5}$$

Thus, timing and frequency can be extracted from the magnitude and phase of $A_k(\tau)$ respectively. This is the basis for the entire algorithm.

3. Exponential smoothing

Recall that the SNR might be very poor, so $A_k(\tau)$ is too noisy to provide reliable estimates. For this reason, the synchronization algorithm utilizes the retransmissions of the NPSS to do accumulation over several frames

$$\overline{A_k(\tau)}_n := \alpha \cdot \overline{A_k(\tau)}_{n-1} + (1-\alpha) \cdot A_k(\tau) \tag{3.6}$$

where n denotes the current frame number and the bar notation indicates the accumulation operation defined by the equation. The forgetting factor, α, is adjusted to the expected time-drift of the system, as explained in Section 5.1.

4. Coherent combining

The next step is to extract the actual metric,

$$\rho(\tau) := \sum_{k=0}^{3} w_k \overline{A_{k+1}(\tau)}\, \overline{A_k(\tau)}^* \tag{3.7}$$

where w_k are fixed weights that were set for minimum mean squared error in this study. Notice the similarity with equation (3.4). This step can be seen as coherent combining at a higher level. Thus, timing and frequency is still contained in the complex value $\rho(\tau)$, but with better accuracy than in $A_k(\tau)$. The accuracy of $\rho(\tau)$ is expected to increase for every new frame due to the accumulation in the previous step. The next step in the processing is to decide when the estimate is accurate enough. Either 5a or 5b can be used for this decision.

5a. Peak detection

The metric, $\rho(\tau)$, is expected to have a large magnitude for $\tau = \tau_0$. If the magnitude has a large peak for some τ, the estimate is likely to be accurate. With

$$(\rho_{\max}, \hat{\tau}) = \max_{\tau} \left\{ \frac{|\rho(\tau)|}{\sum_i |\rho(i)|} \right\}, \tag{3.8}$$

$\hat{\tau}$ can be used as timing estimate if $\rho_{\max} > \lambda_P$. Otherwise the accumulation procedure should continue. λ_P is called the *Peak detection threshold* and should be set to balance PST and FAR.

5b. Genie detection

When evaluating the estimation performance of different synchronization schemes, the false alarms can make the comparison more complicated. One way to deal

with this is to get rid of the detection problem by using a detection rule here referred to as *Genie detection*. It works as follows: $\hat{\tau}$ is used as timing estimate if $|\hat{\tau} - \tau_0| < \lambda_G$, otherwise the accumulation procedure should continue. λ_G is called the *Genie detection threshold*, and the name stems from the fact that the receiver needs to magically know τ_0 to do the detection. This rule is impossible to implement in a UE and is only used for the simulation.

6. Fractional frequency

When the timing is estimated, a fractional frequency estimate, $\hat{f}_F$, can be acquired according the principle in equation (3.5). The estimate is given by

$$\hat{f}_F \coloneqq \frac{128}{137} \cdot \frac{\arg\{\rho(\hat{\tau})\}}{2\pi}, \tag{3.9}$$

where the first factor compensates for the length-9 CP.

7. Integer frequency

The integer frequency estimate, $\hat{f}_I$, can be acquired by hypothesis testing over the subcarriers that are reasonable, given the maximum CFO and CRO in the system model. With a 20 ppm CFO at 900 MHz and a CRO of up to 7.5 kHz for example, the maximum integer frequency offset is limited to two subcarriers. The estimate is given as

$$\hat{f}_I \coloneqq \frac{128}{137} \underset{f_I \in \{\pm 2, \pm 1, 0\}}{\operatorname{argmax}} C_{\mathbf{r},\mathrm{NPSS}}(\hat{\tau}, \hat{f}_F + \frac{128}{137} f_I), \tag{3.10}$$

where $C_{\mathbf{r},\mathrm{NPSS}}$ denotes cross-correlation of $\mathbf{r}(\hat{\tau})$ counter-phase rotated according to $\hat{f}_F + \frac{128}{137} f_I$, with a copy of the NPSS.

3.3.4 Receiver NSSS Processing

The NSSS is also based upon Zadoff-Chu sequences, but with different root indices, and is scrambled according to one of several predefined binary sequences. The root index and the index of the scrambling pattern can encode digital information. The cell ID is encoded, together with the last three significant bits of the current frame number [8]. Thus, timing can be achieved within a window of eight frames, or 80 ms. The NSSS can also be used to refine the fractional frequency estimation by correlation in the frequency domain at the higher sampling rate of 1.92 MHz. NSSS processing is much easier than NPSS processing, and was not investigated in detail in this study. Instead, an off-the-shelf algorithm already implemented in the simulator was used.

3.3.5 Master Information Block

The remaining four bits of the frame number are bundled together with other system information and transmitted in the broadcast channel as a packet called *master information block* (MIB). The MIB is divided up into eight subblocks, each of which is repeated in every first subframe of eight consecutive frames (see Fig-

ure 3.3). Thus it takes 640 ms to transmit one MIB before any new system information can be transmitted.

4

Operation in Unlicensed Spectrum

Initial standardization of NB-IoT has been completed and user chip-sets are already available as of 2017. Like for many other wireless standards, NB-IoT will likely be updated to enable more features or improve on previous ones. One major theme in 5G research is the migration of cellular systems into new frequency bands. This includes exploiting the very wide millimeter-wave bands as well as refarming bands that was previously used for services that are now obsolete[1]. A third kind is the migration into unlicensed bands, which entail a new set of design challenges brought by the specific regulations in these bands. The challenges associated with NB-IoT-U — adaption of NB-IoT into unlicensed spectrum — is the main motivation for this thesis. This chapter will first provide a brief introduction to these bands and why they are useful, adding to the motivation provided in Section 1.1. This chapter then introduces the most suitable frequency bands for NB-IoT-U and what limitations their regulations impose on synchronization.

4.1 Frequency Bands

Regulation and administration of the radio spectrum is done to various degrees and in various ways globally (by ITU), regionally (e.g. by *Electronic Communications Committee* (ECC) in the EU) and nationally (e.g. by *Federal Communications Commission* (FCC) in the U.S.). This process is quite complicated, with many participating organizations. The role of global and regional organizations is to set standards and provide guidelines to be followed by individual nations, but in the end the radio spectrum is actually regulated by national government agencies. They divide the spectrum into bands and decide who can use each band,

[1] The NB-IoT standalone deployment mode is an example of refarming GSM bands.

for what purpose and under which conditions. [13, Chapter 2]

Bands used for civil communication systems can either be licensed or unlicensed. Licensed bands are bands that require a license for operation. A license usually entails the sole right of operation in a specific band and is made available by a government agency through auctioning. With a licensed band an operator can thus have full control over the interference within the band, enabling very efficient multiple access and coordination schemes. The auction price per useful MHz in the U.S. have been on the order of hundreds of millions of dollars in the last decades.

Unlicensed bands on the other hand are open for anyone to use, as long as one follows whatever rules are imposed on these bands (e.g. by using certified equipment). Among other things, these rules specify how coexistence is achieved by the use of multiple access methods. These methods — be it power limitations, duty cycles, spectrum spreading or back off in contention schemes — will prevent each user from operating in the most efficient manner, even when they are alone. And when they are not alone, the interference might be unpredictable and too strong for reliable operation.

There are important advantages of unlicensed bands. One obvious advantage is affordable operation. Thanks to unlicensed bands, WiFi access points can be bought and installed by single households. Also, smaller commercial organizations may use these bands to set up internal wide area networks without having to pay millions of dollars for a license. Another advantage is that the spectrum may be used more efficiently by the market as a whole, as license holders typically do not use their bands at all times or at all geographical places.

The advantages of both kinds of spectrum may be combined by transmitting in two bands simultaneously. *Licensed Assisted Access* is an example of an LTE technology that transmits all the crucial control information in a licensed band while using (often underutilized) unlicensed bands to boost the user data rates. This might serve as an appealing compromise to large operators, which might be afraid to see their spectrum assets lose value if purely unlicensed cellular technologies catch on.

Unlicensed bands might be very suitable for mMTC, whose demands are affordability and cell capacity rather than throughput and reliability. NB-IoT operation in unlicensed bands (here called NB-IoT-U) has been discussed in the 3GPP and MulteFire communities. The two primary markets considered for NB-IoT-U are the U.S. and EU and we limit our study to these regions. In the next section, we will explore the most suitable frequency bands and how their regulations might force a change of the current technology.

4.2 Regulations

Different emission regulations will apply for different types of devices. The FCC for example classify radiators into three categories: intentional radiators, unin-

tentional radiators and incidental radiators. Wireless communication devices are intentional radiators [14]. Rules for communication type of devices typically include:

- What the specific operating frequencies are.
- How much power a device can emit.
- How much spectral leakage or what modulation methods that are allowed.
- How much spurious emissions from harmonics that are allowed.
- Specific interference mitigation schemes, such as duty cycles or spectrum spreading.

The most important regulation is the power emission limit. Since the amount of interference caused to another device is dependent only on the field strength at its location, the regulations are usually set as to limit the maximum field strength produced at a given distance. The maximum field strength will be in the direction of the main lobe. Hence a device with a low antenna gain will be allowed a larger total output power. Power regulations are typically stated as ERP (*equivalent radiated power*) or EIRP (*equivalent isotropically radiated power*), quantities standardized by IEEE. EIRP is defined as the amount of power an *isotropic antenna* would need to create the maximum allowed field strength for a given distance. Similarly, ERP is defined as the power required by an half-wave dipole antenna. Since ideal half-wave dipoles have an antenna gain of 2.15 dBi[2], we have

$$\text{EIRP} = \text{ERP} + 2.15 \text{ dB}. \tag{4.1}$$

This conversion is useful since the FCC and the ECC use different units.

4.2.1 US ISM Bands

The FCC use the term ISM bands (*Industrial, Scientific and Medical*) for *ultra high frequency* bands that allow unlicensed operation by low power devices. They are regulated in a document commonly called *Title 47, Part 15* [15]. In *subpart C* of this document, the operational ISM regulations can be found.

While there are many different ISM bands to choose from, we will narrow down our choices as follows. First of all, high carrier frequencies have two important disadvantages. They tend to have poor propagation properties and they require power hungry and expensive active components. Since NB-IoT is already struggling to increase its coverage and reduce the device cost and power consumption, we will limit our considerations. We draw the line at 3 GHz, only allowing frequencies under this value.

There are two ISM bands below 3 GHz intended for communication: The 915 MHz band (902–928 MHz) and the 2.4 GHz band (2400–2483.5 MHz). The advantage of the 2.4 GHz band is its worldwide adoption and its high bandwidth.

[2] dBi (dB isotropic) refers to the main lobe gain as compared with an isotropic antenna.

These features have made the band very popular for many use cases — some of the most well known being WiFi, Bluetooth and microwave ovens. All this usage cause a high interference floor that would aggravate NB-IoT coverage. Due to this interference, and to the fact that the band still use a rather high carrier frequency, we rule out the band.

We will now study the 915 MHz band in more detail. By limiting the emissions to EIRP = -1.23 dBm[3], the band can be used freely with no further major restrictions. This power is not enough to create a functional cellular system. By following a *frequency-hopping spread spectrum* (FHSS) scheme specified by FCC, much higher output powers are allowed. There are different power limits depending on how many criteria that are fulfilled by the system. The maximum possible power of EIRP = 36 dBm is allowed if:

- The 20 dB bandwidth is less than 250kHz.
- FHSS is used and
  - At least 50 channels are used for hopping.
  - These channels have non overlapping 20 dB bandwidths.
  - At most 0.4 s dwell time per channel per 20 s period is used.
  - The hopping follows a pseudo-random pattern.
- The antenna gain is at least 6 dBi.

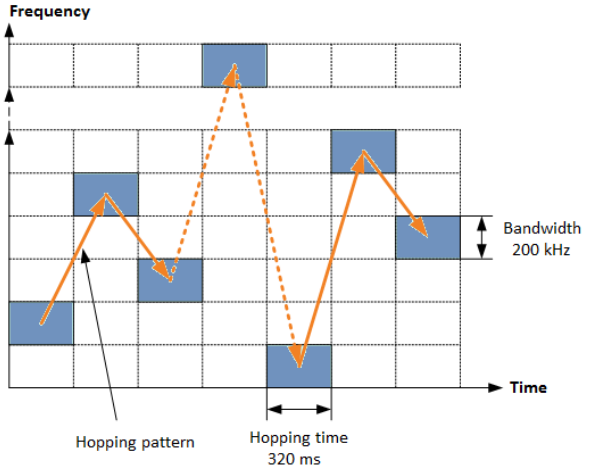

Figure 4.1: *Pseudo-random hopping pattern.*

An illustration of FHSS is given in Figure 4.1. Since the hopping pattern must be pseudo-random, it has to be different between nearby BSs for them to be considered separate systems. Therefore, the NB-IoT UE will not know the exact hopping

[3]dBm measures power in dB with 1 milliwatt as reference.

pattern until is has decoded the cell ID, which happens in the end of the synchronization process when the NSSS is detected. At this point, the timing is known up to a multiple of 80 ms. Therefore, a feasible way to synchronize the UE to the BSs channel hopping is to use a channel dwell time that is a multiple of 80 ms. The UE can then listen to one channel until the NPSS and NSSS is detected and then do hypothesis testing of the number of 80 ms periods until the next hop by sampling both the current and the next channel.

Furthermore, by having the MIB transmission interval of 640 ms being a multiple of the channel dwell time, the MIB boundary is given by knowing the hopping boundary, the hopping pattern and the index of the current channel. This simplifies MIB detection. The largest channel dwell time satisfying both this criterion and the 0.4 s limit is 320 ms. The conclusion here is that NB-IoT-U should be able to finish both NPSS and NSSS processing within 320 ms.

4.2.2 EU Short Range Devices

In the European Union, low power radiating devices are called *Short Range Devices* (SRD). These correspond to ISM in the U.S. and include NB-IoT devices. The Electronic Communications Committee (ECC) in EU provides recommendations for how frequency bands should be allocated and used. The recommendations for SRD are found in a document called *Rec 70-03* [16] and are supposed so be implemented by the member states. [14]

There is a 2.4 GHz band also for SRD, which we exclude based upon previous arguments. There are also three bands around 434 MHz, which have a low power limit and does not allow voice. The remaining bands are listed in Table 4.1 together with their relevant regulations. The 869.4 (to 869.65) MHz band is by far the best candidate in terms of power, which is a crucial factor for NB-IoT coverage: The ERP is 27 dBm corresponding to an EIRP of 29.15 dBm. By concatenating many channels, this band can also be operated in a mode that utilizes the full bandwidth of 250 kHz, which matches the 915 MHz ISM band nicely and allows for a similar implementation. The 10% duty cycle means that: A device can transmit for a maximum total time of 360 s per hour; each transmission can have a maximum length of 36 s; a minimum off time of 3.6 s is required between transmissions. This should, however, be easily manageable once adaption to the ISM dwell time is handled.

4.3 Conclusions

The 915 MHz ISM band and the 869.4 MHz SRD band are the most promising for commercial NB-IoT-U deployment. What we learned from studying their regulations is that a number of adaptions has to be made. The changes or settings that affect our study are the following:

1. The synchronization should not take more than 320 ms.
2. A pulse shaping filter might be needed to make sure the 20 dB bandwidth

Table 4.1: *SRD regulations at interesting frequencies.*

Frequency Band (MHz)	Channel BW	ERP (dBm)	Duty Cycle
868.0 – 868.6	No limit	14	<1%
868.7 – 869.2	No limit	14	<0.1%
869.3 – 869.4	<25 kHz	10	No limit
869.4 – 869.65	<25 kHz[1]	27	<10%
869.7 – 870.0	No limit	7	No limit

[1] Channels may be concatenated.

lies within 250 kHz.

3. The EIRP limits are 29.15 dBm and 36 dBm for EU and the U.S. respectively.
4. A carrier frequency of 900 MHz should be a good assumption in our experiment.

These points are incorporated into our study as described in Chapter 7.

5

Temporal Diversity

This is the first chapter of two that will describe and specify the methods that are studied in this thesis. The chapter starts with an introduction of how coherent combining can be used for *time diversity*. Then follows two sections, each describing a method that is based upon coherent combining.

5.1 Coherent Combining

The synchronization accuracy is essentially limited by the SNR of the received signal. One way of increasing the SNR is to transmit at a higher signal power. This is often not a feasible solution, for several reasons. Higher transmit power can increase the detrimental effects of hardware non-linearities, it obviously leads to a higher power consumption and it might be prohibited by regulations. Moreover, it may lead to higher interference within a system, which in turn actually degrades performance.

Another way of achieving a higher total energy of the received signal is to extend it in time. By allocating more time resources to the reference signals, more redundant information could be sent and enable a less uncertain estimation. But instead of looking at this as increased information, it could it could be treated as an increased SNR, if the signals are *combined coherently*. By transmitting the same reference signal several times and then adding the copies together in the receiver, the fact that the noise realizations are independent would lead to a higher SNR. The power of coherently added signals scales quadratically, while the power of added independent noise scales linearly. With N transmissions of signal s with

additive white Gaussian noise, n, the resulting SNR would be

$$\mathrm{SNR}_{\mathrm{new}} = \frac{\mathbb{E}\left\{|Ns|^2\right\}}{\mathbb{E}\left\{N\,|n|^2\right\}} = N\frac{\mathbb{E}\left\{|s|^2\right\}}{\mathbb{E}\left\{|n|^2\right\}} = N \cdot \mathrm{SNR}_{\mathrm{old}} \tag{5.1}$$

and the SNR gain would thus be $10\log_{10} N$ dB. This in an example of time diversity.

In the real world, estimated target values might change with time. In our case, timing offset will change due to time-drift and frequency offset will change due to oscillator instabilities and varying Doppler shifts. Time drift is expected to be the dominant factor. The solution to the time-variance is to apply a tracking method, such as a linear filter. If s is the signal, and y is the estimate, an infinite impulse response filter would be given by

$$y[n] = \sum_{i=0}^{P} b_i s[n-i] - \sum_{i=1}^{Q} a_i y[n-i]. \tag{5.2}$$

with P and Q being design parameters. The filter coefficients would ideally be calculated using *linear minimum mean squared error*.

The NPSS processing used in this thesis applies such a filter in equation (3.6), with $P = 0$, $Q = 1$, $a_1 = \alpha$, $b_0 = 1 - \alpha$ and $s[n]$ being the auto-correlation value $A_k(\tau)$ instead of directly using the signal $\mathbf{r}$. $A_k(\tau)$ could equally well be thought of as a signal in noise. Due to the complexity of the system model, we will set α experimentally. The rest of this chapter presents two studied methods, which both rely on time diversity and coherent combining.

5.2 Method 1: NPSS Densification

Recall from Chapter 3 that one out of ten subframes in every frame contains the NPSS. Every NPSS on its own provides an opportunity for synchronization but since the SNR can be very low, the UE will combine the NPSSs coherently as described above. In NB-IoT-U however, there is a limited time window of opportunity for synchronization, before the BS hops to a new carrier frequency. In order to increase the probability of success within this time window, we will further increase the NPSS energy by allocating even more subframes. Specifically we will introduce a new NPSS per frame, and refer to this scheme as *densification*, or the *dense* case. As Figure 5.1 shows, the original NPSS will be moved to a later subframe in order to allow for uniform spacing among NPSS subframes without affecting the broadcast channel. Uniform spacing allows for the same efficient receiver algorithm described in Section 3.3.3, but with half the accumulation period. This would have the advantage that only half the buffer memory would be needed in the UE receiver.

One might believe that an extra NPSS subframe would lead to significant overhead, occupying a potential payload subframe. This is not necessarily the case,

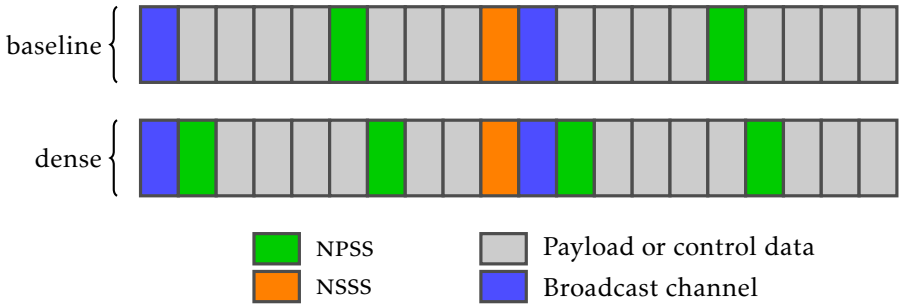

Figure 5.1: *New dense frame structure with an extra NPSS subframe.*

however. Recall from Chapter 4 that NB-IoT-U would use a frequency hopping pattern of at least 50 hopping channels. The idea is that one or a few of these can be used as *discovery channels*. The discovery channels would be known by the UE and contain the extra (or even all of the) synchronization signals as well as other important system information. By synchronizing to one of these channels, the UE can then communicate over all the other channels with significantly lower overhead. The natural question then becomes, why not use more than two NPSS subframes? The answer is that 10 — the number of subframes in a frame — has an inconvenient prime factorization, which would complicate the synchronization design. And two subframes, with an increased SNR of $10\log_{10} 2 \approx 3$ db, is also expected to be enough. Another reasonable prediction that could be made about this method is that with half the accumulation period, the time-drift is smaller and thus the optimal α should be higher.

5.3 Method 2: NPSS Enhancement

If adding a full subframe may seem drastic, an alternative is to add only a few symbols by exploiting a property of NB-IoT-U: it is standalone and therefore no LTE control information is transmitted in the same band. This means that the first 3 symbols in every subframe (see Figure 3.4) are available. They could be used to extend the NPSS, either by prepending the existing code cover, or by replacing it altogether. An important problem now arises: how should the new code cover be designed?

5.3.1 Design Guidelines

The ultimate goal of the code cover design is to improve the estimation and detection by having the metric, $\rho(\tau)$, in equation (3.7) exhibiting a narrow mainlobe and low sidelobes. In the case of high SNR, a narrow mainlobe should be a good goal, in order to improve estimation accuracy. In low SNR however, more focus should be put on the sidelobes, in order to increase detection accuracy and decrease the FAR. By setting a high detection threshold and relying on the coherent accumulation to provide a more accurate metric $\rho(\tau)$, more focus could once

again be put on the mainlobe.

We mentioned earlier that good auto-correlation properties of the code cover is the design target. This is not entirely true however. By studying equation (3.4), we see that we rather want good auto-correlation properties *after* the signal has already been mixed with shifted versions of itself. Actually, in the NPSS processing, correlation-like multiplications are carried out thrice before reaching the metric: $s(m) \cdot \mathbf{r}_m(\tau)$ and $(s(m+k)\mathbf{r}_{m+k}(\tau)) \cdot (s(m)\mathbf{r}_m(\tau))^H$ and $\overline{A_{k+1}(\tau)} \cdot \overline{A_k(\tau)}^*$.

Ideally, the code cover, s, should be jointly optimized with the combining weights w_k, under channel model assumptions, in order to get desirable statistics of the metric, $\rho(\tau)$, that lead to good PST and FAR, but that is a daunting task. Instead, we will try to argue that good auto-correlation properties of s could be a good approximation to get low sidelobes of $\rho(\tau)$, and thus a decent design guideline: Consider the frequency domain. The correlation operations — mixing after time-shifts — correspond to convolutions of rotated spectra. Having low sidelobes in the time-domain requires a spectrum that is the opposite: broad and constant amplitude. This can be achieved if the spectrum of the original sequence also has these properties — you will not get a broad spectrum by convolving narrow functions. Thus, the original sequence should have good auto-correlation properties.

5.3.2 Minimum-sidelobe Binary Codes

With the code cover applied to the base sequence, and with NPSS processing happening every sample, the code cover effectively gets auto-correlated at shifts smaller than one symbol, i.e. smaller than one of its binary elements. Due to this, it is relevant to talk about the mainlobe width. While searching for a suitable code cover however, this sample-level auto-correlation is demanding. Considering only correlation of the binary sequence on its own turned out to be more practical. By doing that, only the magnitude of the sidelobes can be measured.

This was done in [17]. In that study, computer searches were done — up to lengths far greater than our available 14 symbols — in order to find minimum *peak sidelobe level* (PSL) binary sequences. For a binary sequence $s \in \{-1, 1\}^N$, PSL is defined as

$$\mathrm{PSL}(s) = \max_{1 \le k \le N-1} \left| \sum_{n=1}^{N-k} s_n s_{n+k} \right| \tag{5.3}$$

and indicates the *maximum* self interference of s. An additional metric, *merit factor* (MF), was used in that study to indicate the *average* self interference. It is formed by averaging the side-lobes and is defined as

$$\mathrm{MF}(s) = \frac{N^2}{2\sum_{k=1}^{N-1} \left(\sum_{n=1}^{N-k} s_n s_{n+k} \right)^2}. \tag{5.4}$$

These two metrics were used as design guidelines in this study.

Table 5.1: *NPSS code covers. Low PSL and high MF are sought.*

Length	Type	Sequence	PSL	MF
11	Baseline (3GPP)	[+ + + + − − + + + − +]	2	2.9
14	Extension	[− + + + + + + + − − + + + − +]	4	1.9
11	Special Barker	[+ + + − − − + − − + −]	1	12.1
13	Barker	[+ + + + + − − + + − + − +]	1	14.1
14	Minimum PSL	[+ + + + + + + − − + + − + − +]	2	5.2

5.3.3 Barker Codes

Barker codes are codes with PSL = 1. There are also some Barker codes with the sidelobes limited to $\{0, 1\}$, i.e. no negative sidelobes. These are called *special Barker codes*. Barker codes are known only up to length 13, and special Barker only up to 11. It has been shown that no larger Barker codes exist, at least up to length 10^{22}. Barker codes where the first go-to solution that were considered, but two other code covers were also tried.

5.3.4 Proposal

The following decisions were made.

A length-14 sequence with optimal PSL and MF metrics should be tried: the minimum PSL sequence from [17] was chosen.

A suboptimal length-14 sequence that is a pure extension of the current standard should be tried, if such an alternative was found: such a code cover has been proposed in [18], and could therefore be used. The reason for this choice is that a BS using this code could be backward compatible with older version UEs.

An optimal length-11 sequence should be tried in order to investigate whether improvements can be made without allocating additional symbols: a special Barker code was chosen.

A code cover of length 12 or 13 should be tried, if such a code was found with very good PSL and MF metrics: a length-13 Barker code was chosen.

Table 5.1 summarizes all studied code covers.

6

Spatial Diversity

In the previous chapter, we discussed methods to improve reception by making the reference signals more dense in time. In this chapter, we will show methods that might improve the overall performance of the scheme by altering the channel conditions as to mitigate the detrimental effects of fading. These methods rely on spatial diversity, using multiple transmit antennas. As we will see, a good way of doing this is to create artificial fast fading by blindly looping through a carefully chosen codebook.

6.1 Fading Channels in NB-IoT

Many different channel models incorporate a propagation phenomenon called fading, which is the change in reception quality brought on by time-varying channels. One important type, *multipath fading*, is the effect not mainly of objects shadowing the path but of the inverse: objects in the environment cause reflection and diffraction of waves, which leads to multiple incoming rays at the receiver location interfering constructively or destructively. An illustration of this effect is given in Figure 6.1.

Due to movement of both these objects and the UE, the channel will change and the perceived signal quality may fluctuate as the UE moves through weak and strong spots. This is called *fast fading* and may or may not be a desired channel property in a cell. In the case of NB-IoT, a great deal of UEs will be stationarily deployed[1] in a static environment. If in addition to this, the fast fading is not fast enough, a particularly unfavorable situation can occur. With a Doppler spread of less than 1 Hz for example, a substantial fraction of users would experience a

[1] For example the electric meter far down in your basement.

deep fade (i.e. a very destructive interference) throughout the 320 ms synchronization window.

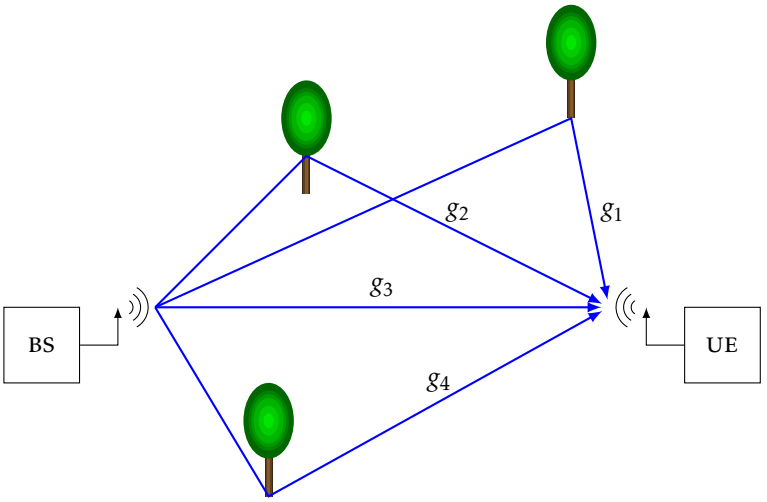

Figure 6.1: *Single-antenna multipath propagation*

This is where diversity comes in. By having several transmit antennas, we can utilize the fact that not all of them will be at a deep fade at any give time. Each antenna will create a unique link to the receiver, with its own channel realization, as shown in Figure 6.2. By transmitting the signal on each antenna simultaneously and applying a unique filter to each of them, the total perceived channel can be changed. Using N transmit antennas, the situation could also be interpreted as a single-antenna channel with N adjustable rays. The goal is to set all the filters so as to get a total channel response with a good quality, or at least to mitigate the effects of deep fades.

The problem during synchronization is that neither the BS nor the UE has got any channel state information, and also that no feedback can be sent in the UL. Hence, the diversity scheme can not be adaptive to any particular UE's situation. *maximal ratio precoding* and *zero-forcing precoding*, for example, require channel state information at the transmitter and can not be used. *Space-time block codes* (e.g. the Alamouti code for two antennas) require channel state information at the receiver and can not be used. Actually, there is no way to increase the expected SNR at an arbitrary UE under these circumstances. What can be done, however, is to average out the SNR over time and space by blindly altering the channel. By regularly changing the filtering at each antenna, a faster fading will be experienced at each UE. This could increase the chance that each UE has at least some period of good reception during the 320 ms synchronization period. We call this technique *artificial fast fading*.

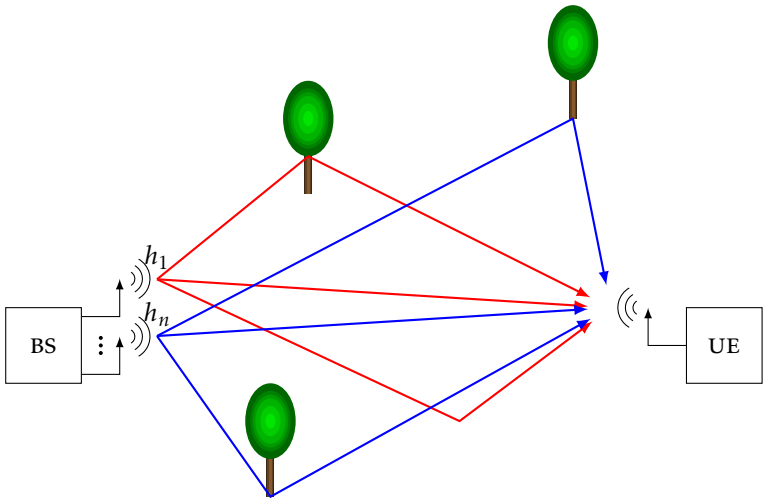

Figure 6.2: *Multi-antenna multipath propagation*

6.2 Motivation

In this section, the usefulness of artificial fast fading will be motivated mathematically for the case of two transmit antennas. The argument can easily be extended to larger number of antennas.

A common wireless channel model is the so called *tapped delay line*, where each tap represents one propagation path — one ray — and has some attenuation and some delay. On frequency bands smaller than the coherence bandwidth (given by the delay spread), the effect of the delay is a phase shift of the incoming waveform. Each ray, i, can then be represented by a complex number, g_i, containing information about the attenuation and phase shift. The phase shift can often be modelled as uniformly distributed. When there are many rays, the central limit theorem comes into effect, and the total channel, $h = \sum_i g_i$, can be approximated by a circularly-symmetric complex normal distribution. This model is called Rayleigh fading.

If multiple transmit antennas are used and, additionally, the spacing between adjacent equally polarized antennas is larger than half the wavelength, the link can be treated as a vector of independent elements $\mathbf{h} = (h_1, h_2) \sim \mathcal{CN}(0, \sigma_h I)$. Figure 6.3 illustrates a channel realization for a two-antenna link. The sum $h = h_1 + h_2$ in the figure, will determine the gain of the channel, but h_1 and h_2 are independent and unknown to the BS.

Consider the scheme in Figure 6.4. By multiplying the NPSS by -1 in every odd frame, we practically get total channel $h_a = h_1 + h_2$ on even frames and $h_b = h_1 - h_2$ on odd frames. When h_a is small, h_b is likely to be large, and the accumulation done in equation (3.6) (described in Section 5.1) will average out the quality. We

will now show that this scheme will lower the variance of the received signal power while keeping expectancy unchanged.

The accumulation in equation (3.6) is done on the auto-correlation values A_k from equation (3.4). The useful signal in A_k is proportional to $|h|^2$. Thus, it makes sense to study the sum absolute square of the channel coefficients over two frames when evaluating the quality of the accumulation. The expected accumulated signal power does not change when h_b is used, as opposed to only using h_a:

$$\mathbb{E}\left\{|h_a|^2 + |h_b|^2\right\} = \mathbb{E}\left\{|h_a|^2\right\} + \mathbb{E}\left\{|h_b|^2\right\} = \mathbb{E}\left\{|h_a|^2\right\} + \mathbb{E}\left\{|h_a|^2\right\} = \mathbb{E}\left\{|h_a|^2 + |h_a|^2\right\} \quad (6.1)$$

The variance however, will decrease when the scheme is used:

$$\begin{aligned}\mathbf{Var}\left\{|h_a|^2 + |h_b|^2\right\} = \mathbf{Var}\left\{|h_a|^2\right\} + \mathbf{Var}\left\{|h_b|^2\right\} + 2\,\mathbf{Cov}\left\{|h_a|^2, |h_b|^2\right\} = \\ = /\text{i.i.d.}/ = 2\,\mathbf{Var}\left\{|h_a|^2\right\} = 2\sigma_h^4 \quad (6.2)\end{aligned}$$

The last equality can be calculated using the fact that the square of a Rayleigh random variable is exponentially distributed. With the assumption that the channel coherence time is significantly larger than a few frames, the variance when the scheme is not used is:

$$\mathbf{Var}\left\{|h_a|^2 + |h_a|^2\right\} = 4\,\mathbf{Var}\left\{|h_a|^2\right\} = 4\sigma_h^4 \quad (6.3)$$

The takeaway from this section is that artificial fast fading will preserve the expected received signal power for every user but make sure that the signal power, averaged over a few frames, stays closer to the expected value. This could lead to a situation where a larger proportion of the UEs can synchronize within 320 ms.

6.3 Codebook Design

How should the extra antennas be used to take advantage of the diversity? Since the channel can be represented using complex coefficients, any channel realization with this representation could be reached by scaling each antenna gain with some complex number. This corresponds to adjusting the power amplifier as well as introducing a phase shift on each antenna. Since the receiver will perceive the total sum of all channels, the received signal could be written using an inner product, as

$$y = \mathbf{h}^H \mathbf{w} x + n, \quad \mathbf{h} \sim \mathcal{CN}(0, \rho_h I), \quad n \sim \mathcal{CN}(0, \rho_n) \quad (6.4)$$

where the vector $\mathbf{w} \in \mathbb{C}^N$ is called the precoder vector and contains the complex scaling factors described earlier. N is the number of transmitter antennas. $\mathbf{w}$ can, within some constraints, be chosen freely at the transmitter side. A very common constraint to have is that of a limited transmitter power, P, translating to $\|\mathbf{w}\|_2^2 \leq P$. The BS design we expect for deployment of NB-IoT uses a single power amplifier chain, setting all gains equal and bounding our choices by $\|\mathbf{w}\|_\infty \leq P$. In this situation, called *equal gain beamforming* (EGB), only the phase matters:

$\mathbf{w} \in \mathbb{T}^N$ and we get the situation illustrated in Figure 6.3.

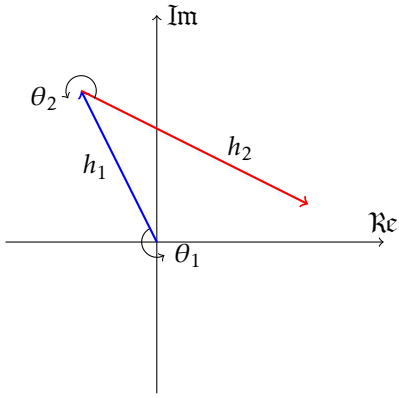

Figure 6.3: *By phase rotation on each antenna, the resulting channel can be varied.*

The example in Section 6.2 used two alternating precoders: $\mathbf{w}_a = [1, 1]^T$ and $\mathbf{w}_b = [1, -1]^T$. This configuration can be written in matrix form as

$$\mathbf{W} = [\mathbf{w}_a \quad \mathbf{w}_b] = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{6.5}$$

where each column is a precoder. We call this type of matrix a codebook and refer to the number of columns, M, as the *codebook length*. For each new NPSS transmission, a new precoder is used. This process repeats, once all precoders has been used. The transmission periodicity is thus equal to the codebook length times the time interval between two NPSSs. Our design challenge is now to choose $\mathbf{W} \in \mathbb{T}^{N \times M}$ in a way that maximizes the chance for a UE to synchronize within 320 ms.

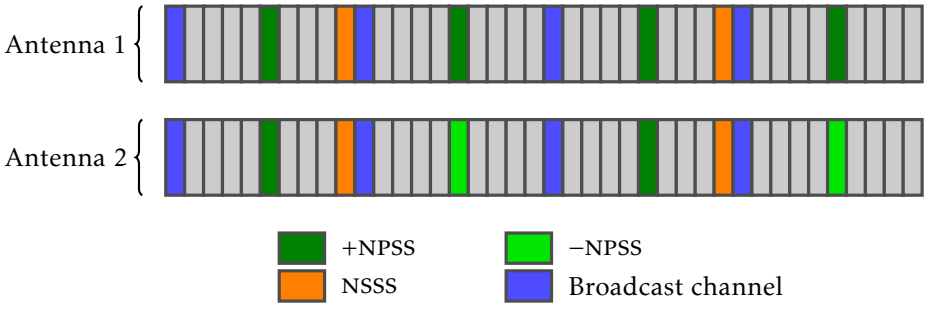

Figure 6.4: *Example of subframe structure using antenna diversity.*

6.3.1 Grassmannian Line Packing

Poor channel quality is avoided by making the channel realizations as diverse as possible. This amounts to picking precoders $\mathbf{w}_k$ that are, in some sense, far apart. The received signal power for a single transmission is proportional to $\mathbf{h}^H\mathbf{w}\mathbf{w}^H\mathbf{h}$, so in order to achieve a diverse codebook, its rows should be spread out in a directional sense. The outer product of $\mathbf{w}$ is independent of phase shifts on $\mathbf{w}$. Moreover, since the algorithm accumulates the timing metric instead of the actual signal, our performance is invariant to uniform precoder phaseshifts.

So we want to spread out $\mathbf{w}$ in all directions of $\mathbb{C}^N$ as measured by inner products, i.e. to do sphere packing on the complex projective space $\mathbb{CP}^{N-1}$. This space is equal to $\mathbf{Gr}(1, \mathbb{C}^N)$, where the r-th order Grassmannian manifold, $\mathbf{Gr}(r, V)$, of the vector space V, is defined as the set of r-dimensional subspaces of V. For this reason, the problem is equivalent to *Grassmannian line packing* and is hence sometimes referred to as *Grassmannian precoding*.

Since $\mathbb{T}^N$ is not a vector space, our problem is not directly Grassmannian line packing, but is usually regarded as line packing on $\mathbf{Gr}(1, \mathbb{C}^N)$ with EGB constraints. Grassmannian line packing on high dimensional complex space is an open problem in mathematics. The case with one antenna is of course trivial. The two-antenna case has been solved by proving the isometric isomorphism $\mathbf{Gr}(1, \mathbb{C}^2) \cong S^2$ [19]. The EGB case for two antennas is trivial. For higher dimensions however, numerical approximations are commonly used, but can be hard to find because of the non-convexity of the optimization problem [20].

6.3.2 Hadamard Patterns

If $M = N$, that is if the codebook is a square matrix, the optimal Grassmannian line packing will be achieved by any unitary matrix. For example, any orthonormal basis in $\mathbb{R}^N$ is optimal, which under the EGB constraint turns to be the Hadamard matrices, if the dimension N is a power of two. Hadamard matrices are often defined as an example of generalized Fourier matrices, but can conveniently be defined recursively as

$$\mathbf{H}_{i+1} = \begin{bmatrix} \mathbf{H}_i & \mathbf{H}_i \\ \mathbf{H}_i & -\mathbf{H}_i \end{bmatrix}, \quad \mathbf{H}_0 = [1] \tag{6.6}$$

6.4 Method 3: Artificial Fast Fading

This section specifies the set of codebooks that constitute Method 3. Hadamard matrices up to $\mathbf{H}_4$ will be used. Larger sizes are not expected to help much, since the synchronization already must be done within a handful of frames. Moreover, while the quality averaging effects for a few antennas might be useful, diminishing returns are expected for additional antennas. As mentioned earlier, the EGB case for $N = 2$ is easy. Since only the direction of $\mathbf{w}$ matters, the first row can be kept unchanged without loss of generality. For an M-sized code, the second row should contain the M-th roots of unity. We will choose the optimal Grassmannian

Table 6.1: *Proposed codebooks.*

Antennas	**Precoders**	Matrix	Type
1	1	$\mathbf{H}_0$	Baseline
2	2	$\mathbf{H}_1$	Hadamard
2	4	$\mathbf{M}_2$	Optimal
4	4	$\mathbf{H}_2$	Hadamard
4	8	$\mathbf{M}_4$	Numerical
8	8	$\mathbf{H}_3$	Hadamard
16	16	$\mathbf{H}_4$	Hadamard

packing for $M = 4$:

$$\mathbf{M}_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & j & -j \end{bmatrix} \tag{6.7}$$

A precoder for $N = 4, M = 8$, was found using a numerical optimization software package provided by [20] and it is listed as $\mathbf{M}_4$ among the other codebooks in Table 6.1. The package contains sequential smooth optimization algorithms for the Grassmannian packing problem, with and without the EGB constraint.

7

Experimental Setup

This chapter describes how the theory, background and methods introduced so far can be used to provide answers to the questions asked in the introduction. A sequence of simulations are described whose results can be tied directly to our metrics. The first section outlines a step by step procedure and the second section specifies the parameter settings for each simulation.

7.1 Experiment Overview

As stated in Chapter 1, we want to *study* the performance of *different methods* for initial cell acquisition and synchronization. Studying here refers to implementation and simulation of the suggested methods under coverage conditions that could be considered realistic but challenging for NB-IoT-U. The tool that will be used for this is a proprietary Matlab link simulator developed at Ericsson AB.

7.1.1 Objectives and Methods

As a reminder of what this study is about, we will here recapitulate our studied methods and research objectives. We have described three methods, two of which have several configurations:

- NPSS densification — illustrated in Figure 5.1.
- NPSS enhancement — with the five settings shown in Table 5.1.
- Artificial fast fading — with the seven settings shown in Table 6.1.

These methods will all be investigated by simulation as an attempt of answering the following questions from Section 1.3:

1. How can NB-IoT synchronization time be decreased while keeping accuracy high and receiver complexity low?
2. What methods could be good candidates for NB-IoT-U?
3. Would proposed candidates require a change of the current standard?

Question 1 will be approached by making a comparison of the three methods. The comparison will be relative to each other, with the main benchmark being a low PST for almost all users. Accuracy will be controlled by using the Genie detection rule. The complexity issue is handled by limiting the setup to use the receiver algorithm from Section 3.3.3.

Question 2 will be answered by picking good candidate methods, set the simulation parameters as realistically as possible and then compare the metrics with the requirements. From Chapter 4, we learned that the time limit for full synchronization is 320 ms. We need 90% of users to achieve this with a FAR not greater than 5%. The FAR can be measured by using the Peak detection rule.

Question 3 will be answered by observing whether the answers to Question 2 requires changes to the existing NPSS or its placement within the frame. A pure extension of the NPSS does not and neither does the use of method 3.

7.1.2 Experiment Procedure

The study will be carried out in a 4-step procedure. Each step will provide settings or other information for subsequent steps.

Step 1: Prestudy

The purpose of the prestudy is to determine how the rest of the study should be set up, in order to yield more reliable results. The first thing to do is to make sure that the bandwidth requirement stated in Section 4.3 is met by finding a suitable transmit filter. The reason we do this is that a transmit filter may have significant effects on performance.

The other thing that is done in the prestudy is finding a relationship between PST (corresponding to NPSS processing) and the full synchronization time (corresponding to both NPSS and NSSS processing). The conclusion from Section 4.3 is that full synchronization time should take a maximum of 320 ms. The main purpose of this thesis regards only PST, which is the interesting and most challenging part of the synchronization, but NSSS processing may take some non-negligible time and must be taken into account while answering Question 2 above. Thus a new smaller time limit should be set for PST.

Step 2: Method comparison

All synchronization methods outlined in Section 7.1.1 will be simulated using the same baseline parameters (with the filter chosen in the Step 1) and with the only changes being whatever is implied by the respective method.

These results will serve two purposes. They will be used directly to answer Question 1 by comparing the methods to the baseline and to each other. They will also be used to select two or three candidate methods that seem promising for meeting the demands of NB-IoT-U declared in Section 7.1.1 and the Step 1. The candidates are chosen from the methods tested in this step but may also be combinations of these. The candidates will be studied in more detail in subsequent steps in order to answer Question 2 and 3.

Step 3: Algorithm fine-tuning

The purpose of this step is to tune the receiver algorithm for each of the candidate methods chosen in Step 2 so as to maximize performance. To simplify the study and make the comparison more straightforward, Step 1 & 2 eliminate the effects of false alarms by applying the Genie detection rule. This is not realistic and does not give a full picture of the performance. To answer Question 2 and 3, the candidates are evaluated using the Peak detection rule.

There is a trade-off between PST and FAR. A relaxed detection threshold, λ_P, leads to faster synchronization at the cost of more false alarms. The forgetting factor, α, also affects this but its relationship is not monotonic. When α is set for maximum-ratio combining, the perceived SNR is optimal, leading to a minimum FAR. The relationship between α and PST is more complicated and depends also on λ_P and the SNR of the individual user. In principle, a suboptimal α will lead to more false alarms which lowers the PST, but also to more *false negatives*[1] that increase the PST. In Step 3, first α and then λ_P will be set.

Step 4: Evaluation

In the last step, the candidates are evaluated using the parameters set in Step 3. The simulations will have an increased sample size for more reliable statistics. The results will include both tables and plots showing the distributions of both PST and residual errors. These results ought to be reliable enough to answer Question 2 & 3.

7.2 Simulation Setup

The system model is based upon previous studies done on NB-IoT synchronization for standalone deployment ([3], [2], [18]) but is modified according to Section 4.3. This section summarizes the fixed system parameters as well as the baseline simulation parameters and how they change for all simulation instances.

7.2.1 Basic Assumptions

The system consists of a UE receiver in a single cell with the BS transmitting over a time period of approximately 10 seconds — long enough to assure that the UE will synchronize. There are no additional cells to cause ICI nor any interfering

[1] False negatives are often referred to as *type II errors* and occur when the estimate although being accurate is deemed too uncertain for detection.

Table 7.1: *Fixed parameters*

Parameter	Value
Number of cells	1
Deployment mode	Standalone
Channel model	Typical Urban 1
Fading	Modified Jakes
Doppler spread	1 Hz
Carrier frequency	900 MHz
Receiver antennas	1
SNR	-8.5 dB
Sampling rate	1.92 MHz

users. No frequency-hopping is done as the purpose is to determine the synchronization time actually required, however long it may be. The fixed parameters — parameter that stay unchanged throughout the study — are summarized in Table 7.1 and explained in the following paragraphs.

BS Transmission

The BS transmits only one NB-IoT PRB with all associated physical channels at a 900 MHz carrier frequency. The transmission is done in standalone deployment mode, which means that there is no LTE control information allocated nor any puncturing of NPSS or NSSS by LTE reference signals. The transmit filter to use is to be determined in Step 1 and is used for all simulations in subsequent steps. One transmit antenna is used as a baseline, but more are added when Method 3 is used.

Initial offsets

Initial timing offset is uniformly distributed over the DL periodicity of 80 ms. For the NPSS processing, this translates into an initial offset uniformly distributed on the interval [-5 ms, 5 ms] at a granularity of the sampling frequency f_s = 1.92 MHz.

The total CFO, is given by equation (2.2). The UE velocity is negligible when the Doppler spread is 1 Hz and the carrier frequency is 900 MHz. NB-IoT should support cheap mMTC devices with 20 ppm oscillator errors, which translates into an error of ±18 kHz. To prove robustness also for small BS instabilities and larger UE errors, the initial CFO is set to ±50 kHz, with equal probability.

The maximum CRO in NB-IoT is ±7.5 kHz which arises due to the fact the the UE may not know the deployment mode until after synchronization. The CRO is in this study fixed to 7.5 kHz, leading to a total offset of 7.5 ± 50 kHz. The time-drift is modelled as proportional only to the CFO.

Channel model

The channel model is a tapped delay line model called *Typical Urban 1* based on measurements carried out within the *COST 207* project. Its power delay profile, consisting of 12 taps, is defined in [21]. Each tap is Rayleigh fading, implemented by the modified Jakes' model [22] with 16 rays per tap and the Doppler spread set to 1 Hz. This Doppler spread reflects the challenge in mMTC of having many UEs being slow moving or stationary.

Link budget

When assessing the coverage performance of a wireless technology, 3GPP frequently uses a quantity called *maximum coupling loss* (MCL). This is defined as the maximum total loss in signal gain from BS antenna port to UE antenna port under which said technology can operate. The MCL includes the effects of antenna gains, path loss and loss due to cables, walls and slow fading. The normal coverage goal for NB-IoT is MCL = 144 dB, as set by 3GPP in [9]. A coverage of MCL = 154 dB is used in this thesis, to aim towards the extended coverage capability for mMTC desired by ITU. This number will help us set the target SNR using

$$\text{SNR} = \underbrace{\text{EIRP} - \text{BS antenna gain} - \text{MCL}}_{\text{Signal}} - \underbrace{10\log_{10}(1000 \cdot k_B \cdot T_0 \cdot F \cdot \Delta f)}_{\text{Noise}}. \quad (7.1)$$

Using a BS antenna gain of 6 dB, standard noise temperature $T_0 = 15$ °C, noise figure $F = \sqrt{10} = 5$ dB and the NB-IoT bandwidth of $\Delta f = 180$ kHz, the resulting target SNR is −8.5 dB. Equation (7.1) can then be applied in the opposite direction to find out that our results will apply equally well to the EU band under MCL = 154 - 36 + 29.2 = 147.2 dB, where the radiation limits from Section 4.3 were used.

Receiver synchronization algorithm

The algorithm from Section 3.3.3 is used, where the signal processing is done at the reduced sample rate of 240 kHz for reduced computational complexity. α is set to 0.9 for Step 1 & 2. Step 1 & 2 use Genie detection with $\lambda_G = 2.5$ μs and Step 3 & 4 use Peak detection. NSSS processing is carried out only in Step 1 and is done as described in Section 3.3.4.

Iterations and random seeds

A simulation will be run for many iterations, each of which provides a completely new realization of one BS and one UE created by a new random seed. The seed is used to randomize three things: initial offsets, channel ray phases and receiver noise. 1000 iterations are used per parameter setup in Step 1–3, which should produce a reasonably low variance of the 95th percentile statistic. In Step 4, 10000 iterations are used.

7.2.2 Varying Parameters

This section explains all simulations and for each one how the parameters are set. Baseline settings are listen in Table 7.2 and are applied to every simulation

Table 7.2: *The varying parameters, exemplified by the baseline case.*

Parameter	Value
NPSS density	1 / frame
Code cover	3GPP
Transmit antennas	1
α	0.9
λ_G	2.5 μs
λ_P	15
Iterations	1000
Transmit filter	see Section 8.1

unless stated otherwise.

Prestudy

Transmit filters may be needed to limit the 20 dB bandwidth to 250 kHz. A *power spectral density* estimate will be formed by generating transmit signals in the simulator using different filters. The alternative of not using a filter at all will be compared to the alternative of using a filter previously used in the simulator for studies on GSM refarming and to the alternative of designing a suitable *root-raised-cosine* (RRC) filter. The RRC filter will be limited to a maximum of 61 taps and designed for a flat passband response while meeting the leakage limit. Worth mentioning is that an RRC filter is used at the receiver.

Using the results from the filter selection, an affine relationship will be found between PST and full synchronization time by linear regression. The model assumed here is that the NSSS processing time consists of a constant mean independent of the NPSS processing time plus a dependent part proportional to NPSS. The PST corresponding to a full synchronization time of 320 ms will be used as our target.

Comparison

Method 1 will be tested by running the baseline (with NPSS in subframe 6) and by running the dense case (with NPSS in subframes 2 & 7). The forgetting factor will be kept unchanged although the dense case is expected to favor a higher value of the parameter. The chance for adjustment comes in Step 3.

Method 2 will be tested similarly, with the baseline compared to all code covers.

Method 3 will be tested with different numbers of antennas that all transmit the same signal, changed only by the precoder. The power is spread evenly between antennas and normalized to sum up to the same transmit power used in the baseline. One precoder corresponds to a column in the codebook matrix and is multiplied element wise to the antennas. Each new NPSS transmission will change precoder to the next one in the matrix, e.g. such that a 2×4 codebook has a period time of 4 frames.

Algorithm fine-tuning

The candidates selected in Step 2 correspond to a specific setting of one method, such as a specific code cover or codebook. Based on the parameters from Step 2, each candidate are run in Peak detection mode with a number of settings for α. Next, the best α is used for simulations with a number of settings for λ_P.

The choice of these parameters amount to a the trade-off between PST and FAR and thus FAR needs to be defined, i.e. in terms of maximum acceptable residual errors. One part of the prestudy not yet mentioned in this report is a comparison of residual frequency and timing errors. As it turned out from that study, the timing estimation is by large the most challenging part. Given an accurate timing, the subsequent frequency error was well within the 50 Hz requirement defined in [9]. We thus define FAR based on a timing error of 5 μs, to cover up for the low sample rate of 240 kHz used for NPSS processing. The 3GPP requirements are 2.5 μs but are on the other hand set very tightly and may indeed be relaxed. [9]

Evaluation

10000 iterations are used in Step 4. Moreover, the best α and λ_P from Step 3 are used. The rest is unchanged from Step 2.

8 Results

This chapter presents the results of all steps described in the previous chapter. Most experiments measure how PST is distributed for a particular parameter setup. The ECDF plots generated using this data look similar to exponential distributions, and are rather smooth. For this reason, these plots are in most places replaced by ECDF tables, which facilitate assessment of performance. The study is carried out in several steps (as outlined in Section 7.1.2) and each top-level section in this chapter corresponds to one such step. Decisions made based upon the results of one step are motivated in the corresponding section.

8.1 Prestudy

In addition to the following, some less interesting prestudies were done. These include replication of previous studies ([3], [2], [18]) and coarse tuning of the receiver algorithm.

8.1.1 Pulse Shaping

An RRC filter was designed with two criteria in mind: meeting the 20 dB bandwidth of 250 kHz and keeping the frequency response flat within the 180 kHz PRB. Three parameters were used for this: roll-off factor (β), oversampling factor (OS) and filter length (truncation). Another filter that was found in the preexisting simulator was also considered. That filter had been used for studies on licensed standalone deployments for NB-IoT and was designed to fit the GSM-mask.

The transmit signal was passed through these filters and power spectral densities of the outputs were estimated. The one-sided Welch estimates are shown in Figure 8.1. The GSM filter meets the 20 dB bandwidth but is showing decline at the

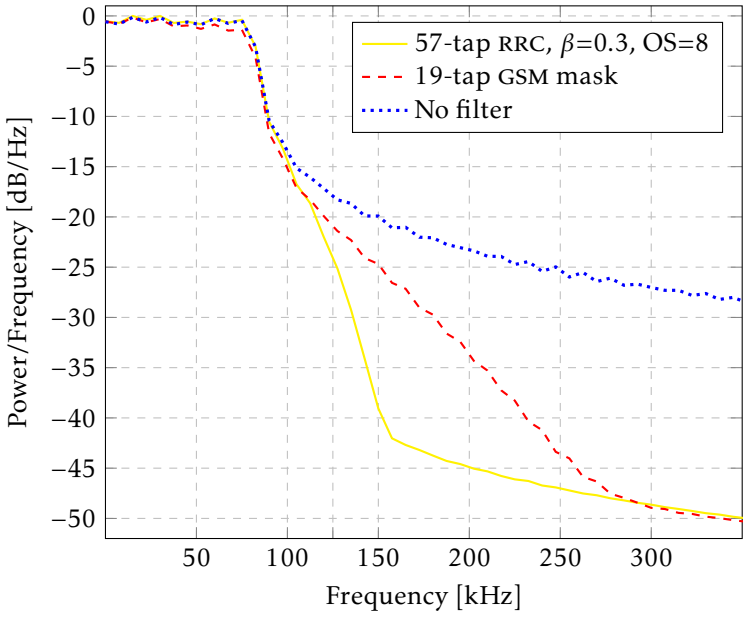

Figure 8.1: *Welch PSD estimate of the transmit signal, showing the 20 dB bandwidths for different transmit filters.*

Table 8.1: *PST [ms] for different transmit filters*

ECDF	No filter	GSM mask	RRC
50%	50	50	40
90%	270	280	260
95%	430	420	340
99%	820	810	770

edge of the PRB. The RRC filter on the other hand meets the 20 dB bandwidth with an even larger marginal while keeping the in-band response similar to the unfiltered signal. Table 8.1 shows that the performance of the GSM filter is noticeably but not overwhelmingly worse than that of the RRC filter. The RRC filter, having a length of 57 taps, unfortunately suffered from prohibitively long simulation times — an effect that might arise also in BS implementations. The GSM filter was chosen for our study and more rigorous filter designs could be considered for future studies.

8.1.2 Target Latency

Performing a large number of NSSS processing simulations was out of the scope of this study. One simulation using the baseline setup was done however, with some results plotted in Figure 8.2 together with a linear fit. The regression line can be

interpreted as showing the (sample) average time required for NSSS processing, given that the NPSS processing took a certain time. The linear model is motivated by the assumption that the NSSS processing relies on low residual NPSS processing errors, which given a robust detection rule should be largely uncorrelated to the SNR, leading to the constant term. The scaling factor is due to the fact that NSSS processing still relies on good reception of the NSSS, which depend on the SNR.

$y = 320$ ms gives us $x = 259.4$ ms (see linear fit in Figure 8.2), which means that when NPSS takes (i.e. PST is) 259.4 ms, the full synchronization is expected to take 320 ms. Our objective is to maximize the worst case performance as opposed to the average case. On the other hand, the NSSS processing used in the simulations are off-the-shelf and perhaps far from optimal. 260 ms is chosen as the 90th percentile PST target, with the repercussions of this compromise addressed in Chapter 9.

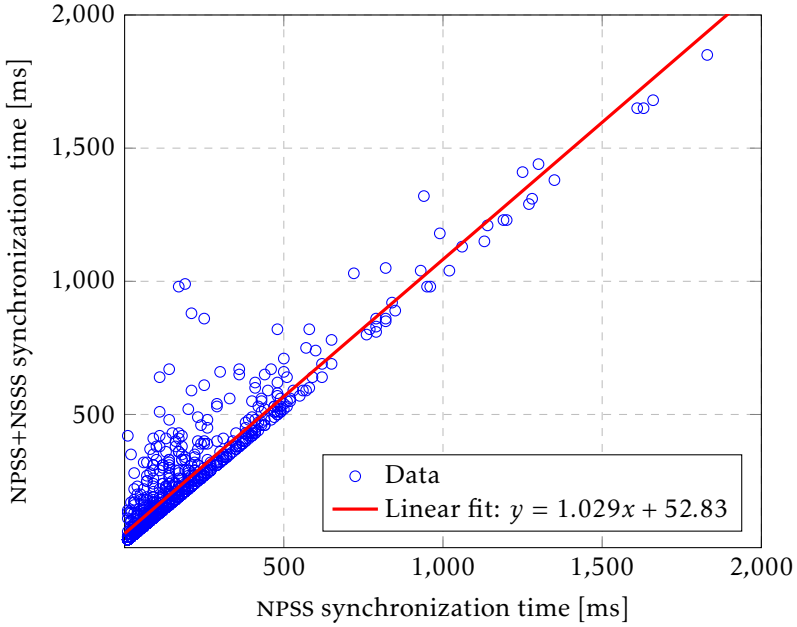

Figure 8.2: *Relationship of synchronization time between primary and secondary synchronization.*

8.2 Method Comparison and Selection

This section corresponds to Step 2. Each of the three methods are tested using 1000 iterations and the GSM filter.

Table 8.2: PST [ms] for different NPSS densities. Left case is baseline.

ECDF	1 NPSS / Frame	2 NPSS / Frame
50%	50	25
90%	280	170
95%	420	240
99%	810	500

Table 8.3: PST [ms] for different code covers.

ECDF	11 Baseline	11 Barker	13 Barker	14 Leukhin	14 Extension
50%	50	50	50	40	40
90%	280	300	260	250	240
95%	420	430	370	360	320
99%	810	1080	730	780	740

8.2.1 NPSS Densification

The results of Method 1 are shown in Table 8.2. Note that all values are multiples of the NPSS periodicity: 10 ms and 5 ms respectively. Note also that both variants use the baseline forgetting factor $\alpha = 0.9$ applied to each new NPSS instance. The higher density setting will experience less time drift during this period and should ideally use a higher forgetting factor.

8.2.2 NPSS Enhancement

The results of Method 2 are shown in Table 8.3. Note that 14 symbols are required to synchronize 90% of the users in less than 260 ms. Note also how the performance gains compare to Method 1, where the number of symbols was doubled.

8.2.3 Artificial Fast Fading

The results of Method 3 are shown in Table 8.4.

Table 8.4: PST [ms] for different code books.
(Tx elements × codebook length)

ECDF	1x1	2x2	2x4	4x4	4x8	8x8	16x16
50%	50	40	30	30	40	30	30
90%	280	170	170	110	110	80	80
95%	420	250	240	150	150	110	100
99%	810	510	550	320	250	160	150

Table 8.5: *PST [ms] and FAR for different forgetting factors. Candidate 1: 4x4 codebook, 1 NPSS per frame*

ECDF	α = **0.90**	**0.93**	**0.95**	**0.96**	**0.97**
50%	60	70	70	80	90
90%	220	220	220	230	250
95%	310	300	300	310	320
99%	520	480	480	480	490
FAR	8.2%	7.7%	6.9%	6.2%	5.7%

8.2.4 Candidate Selection

By comparing all previous results to our target of 90%, it seems that denser NPSS alone, or alternatively 2 transmit antennas alone, would be more than enough to meet our goals. Further investigation showed that this is not the case. The reason is that the previous simulations studied estimation performance, using Genie detection. When applying Peak detection, false alarms are introduced, which in turn forces the use of very strict detection thresholds that lead to longer synchronization times.

After a quick investigation, the following 2 candidate methods were chosen:

- 1 NPSS per frame with 4 transmit antennas.
- 2 NPSS per frame with 2 transmit antennas.

The choice of codebook seemed rather arbitrary based on the results, so the Hadamard codebooks were chosen. To limit the scope of the study, Method 2 (with its relatively low performance gains) was not studied further.

8.3 Algorithm Fine-tuning

For the 2 chosen candidate methods, the next step is to first choose α and then λ_P. α is chosen for minimal 90th percentile PST. λ_P is chosen for the trade-off between a 260 ms 90th percentile PST and a 5% FAR.

8.3.1 Candidate 1

Candidate 1 uses 1 NPSS per frame and 4 transmit antennas with the $\mathbf{H}_2$ codebook.

Forgetting factor

The results in Table 8.5 show that the best worst-case PST performance was achieved for $\alpha = 0.93$ and $\alpha = 0.95$. The average, $\alpha = 0.94$, was chosen as a compromise.

Table 8.6: PST [ms] and FAR for different detection thresholds. Candidate 1: 4x4 codebook, 1 NPSS per frame, $\alpha = 0.94$

ECDF	λ_P = **15**	**16**	**17**	**18**	**19**	**20**
50%	70	70	80	80	90	90
90%	220	240	250	270	280	300
95%	300	320	340	360	400	430
99%	480	520	590	600	710	760
FAR	7.1%	6.4%	5.7%	4.9%	4.5%	4.3%

Table 8.7: PST [ms] and FAR for different forgetting factors. Candidate 2: 2x2 codebook, 2 NPSS per frame

ECDF	α = **0.90**	**0.93**	**0.95**	**0.96**	**0.97**
50%	40	40	45	45	50
90%	255	240	225	220	230
95%	380	355	325	310	330
99%	875	665	655	595	575
FAR	7.8%	7.7%	7.2%	5.9%	5.1%

Threshold

The results in Table 8.6 show that $\lambda_P = 17$ yields a 90% PST below 260 ms, but a FAR greater than 5%. $\lambda_P = 18$ was chosen, since the FAR is just below 5%. Note that the PST ECDF values are given as multiples of 10 ms frames and that the percentile values are rounded upwards. The exact ECDF value for 260 ms might be close to 90% and will be illustrated in the evaluation Step.

8.3.2 Candidate 2

Candidate 2 uses 2 NPSSs per frame and 2 transmit antennas with the $\mathbf{H}_1$ codebook.

Forgetting factor

The results in Table 8.7 show that the best worst-case PST performance is achieved for $\alpha = 0.96$, which was chosen.

Threshold

The results in Table 8.8 show that only $\lambda_P = 17$ satisfies both the PST and FAR requirements. This value was chosen for evaluation.

Table 8.8: *PST [ms] and FAR for different detection thresholds. Candidate 2: 2x2 codebook, 2 NPSS per frame, α = 0.96*

ECDF	λ_P = **15**	**16**	**17**	**18**	**19**	**20**
50%	45	50	55	55	60	60
90%	220	245	255	265	280	295
95%	310	335	380	390	405	420
99%	595	655	665	725	725	905
FAR	5.9%	5.5%	4.3%	4.1%	3.1%	2.6%

Table 8.9: *Parameter settings for the 2 candidates*

Metod	**NPSS density**	**Tx Antennas**	**Precoders**	α	λ_P
Baseline	1	1	1	0.90	15
Candidate 1	1	4	4	0.94	18
Candidate 2	2	2	2	0.96	17

8.4 Evaluation

In this section, the two proposed candidates are tested using ten times the previous sample size: 10000 iterations. Baseline performance is added as a reference and note that its parameters have not been fine-tuned. The methods and their parameter settings are summarized in Table 8.9. The data is shown in plots this time. Figure 8.3 shows PST zoomed to interesting intervals and Figure 8.4 shows timing errors.

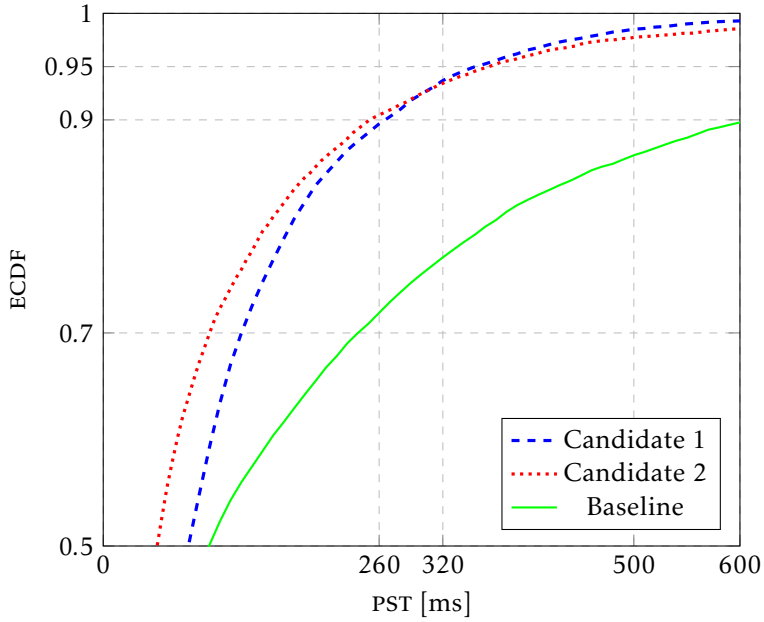

Figure 8.3: *Evaluation: NPSS detection time*

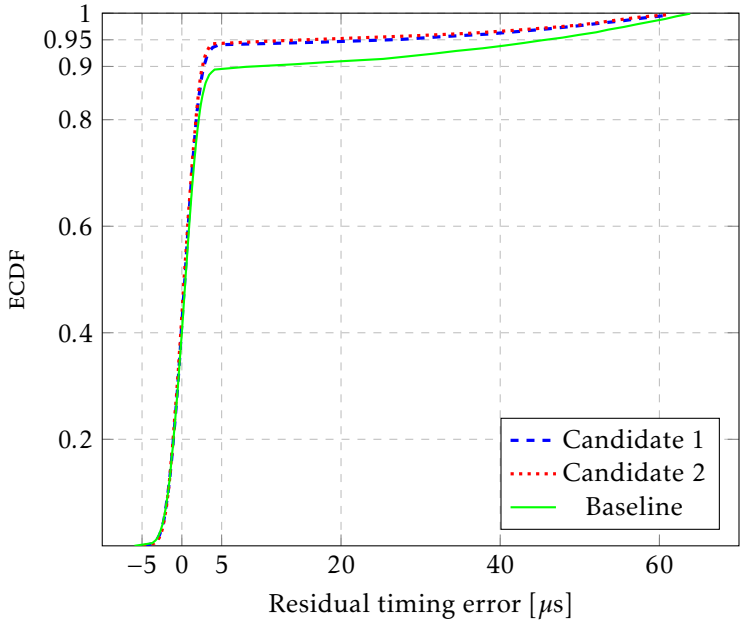

Figure 8.4: *Evaluation: Residual timing error*

9 Discussion and Conclusion

This chapter concludes our study by discussing the results, their applicability and their implications. The first section summarizes the results from Chapter 8 and points out interesting data that do or do not agree with the theoretical parts of the thesis. After that, the general research method and its possible shortcomings is discussed. A conclusion is then made, reconnecting the previous discussion to the stated purpose and research questions from Chapter 1. The final section on future work lists a number of ways in which the study could be extended in order to answer our research questions with more confidence.

9.1 Results

In this section, the results are commented.

9.1.1 Prestudy

The RRC filter performed better than the GSM filter, which is not surprising, given that it is more flat within the 180 kHz PRB. It also performed better than not using a filter, which may be explained by it being better matched to the RRC receive filter. The GSM filter performed similarly to the unfiltered case, which was not expected and not fully understood.

The processing latency of NPSS and NSSS seemed surprisingly uncorrelated. NSSS latency was expected to scale faster with NPSS, since a particular SNR realization should affect both steps. The fact that NPSS took up the majority of the time was expected.

9.1.2 Method Selection

Extension of the code cover showed slight improvement, but of course not as good improvement as the NPSS densification, which in principle corresponds to adding 11 symbols to the code cover.

The fact that the Barker codes did not come out superior was a bit interesting, given the argument that good auto-correlation properties should suit the synchronization scheme. The fact that the extended baseline sequence performed well was welcomed, for it is the only alternative that can provide backward compatibility.

Transmit antenna diversity was showed to yield a good improvement but only up to eight antennas. This is thought to have two main reasons. For the first 80 ms, only eight precoding patterns can be used, and by using eight antennas, these patterns will be orthogonal and thus optimal. Additional antennas can only be helpful if the synchronization takes more than 80 ms. Secondly, by using eight antennas, the channel quality will already be averaged out significantly, with diminishing returns on each additional antenna. Note also that increasing only the number of antennas is much more effective than increasing only the codebook length. The effect of having a number of precoders larger than the number of antennas was not explored theoretically in the thesis and had to be tested experimentally. The idea behind using this was that successful synchronization could perhaps be achieved by making sure at least one channel realization is *very good*, as opposed to relying on accumulation and the average channel quality.

9.1.3 Algorithm Fine-Tuning

The two chosen values for α did not differ a lot, considering the limited gains that can be made by very high values for α during the relatively short time frame needed for NPSS processing. No prediction was made regarding which candidate should end up with the highest α. Candidate 1 with many antennas should favor a high value, to effectively utilize the different realizations. But Candidate 2 should also favor a high value, since the time drift between two consecutive NPSS transmissions is smaller. Both candidates has values higher than the interval $[0.5, 0.9]$ suggested by [2].

No particular observations were done regarding the detection threshold.

9.1.4 Evaluation

Both methods performed significantly better than the baseline in both PST and FAR at the same time. This should be expected, given the results from the method selection. Candidate 1 barely miss the target of a 90th percentile PST of 260 ms while Candidate 2 barely achieve it. Note also from Figure 8.3 that in terms of PST, Candidate 1 favors worst-case users more compared to Candidate 2, which favors average users more. The residual errors that lie within [-2 μs, 2 μs] seem to be uniformly distributed. These could be explained as quantization errors caused by the lower NPSS processing sampling rate of 240 kHz.

Both candidates perform similarly in terms of FAR and barely miss the target of 5 μs for 5% of the users. Improvement of the FAR turned out to be very challenging, which stems from a peculiar effect visible in Figure 8.4: while most of the timing errors were confined to 5 μs, some errors were as large as 65 μs, corresponding to one OFDM symbol. And these larger errors were only visible for positive values.

The explanation for this comes down to how the receiver algorithms fundamentally works. For all code covers, the timing metric has a sharp peak centered at zero, but with rather high and asymmetric side lobes within the symbol time, even when no channel or noise is added. Figure 2.5 illustrates this effect. The data is taken from the prestudy and shows the value of $L_1 = \rho$ after 30 accumulations. When lower values of λ_P was used, essential false alarms occurred: those errors appeared to be uniformly distributed over the whole frame of [-5 ms, 5 ms], which one should expect from false alarms.

9.2 The Work in a Wider Perspective

The purpose of this thesis was twofold: to increase performance for NB-IoT and to enable NB-IoT-U. Better performance of NB-IoT might lead to a faster or more widespread adoption of mMTC devices, which in turn could be a step in the direction towards *The Networked Society*. The Networked Society is a vision of a time when all things that could benefit from connectivity will be connected. The societal impacts of such a situation are hard to predict but will undoubtedly be vast — far beyond the scope of this discussion.

The adaption of NB-IoT for unlicensed spectrum could have many effects, some of which are discussed in Section 4.1. For example, it could lead to an overall more efficient use of the limited amount of available radio spectrum. It could open up for small operators or businesses to set up wireless networks, small or large. It could also increase the prevailing interference in these bands, potentially obstructing some of the current usages.

9.3 Conclusions

The purpose of this thesis was to aid the design of NB-IoT-U by studying the performance of various methods for initial cell acquisition and synchronization in NB-IoT. This goal could be considered fulfilled, since the results could be used to make decisions on either further research or on system design. In the following section, the specific research questions stated in the problem formulation are addressed.

9.3.1 Answers to the Research Questions

How can NB-IoT synchronization time be decreased while keeping accuracy high and receiver complexity low?

It can be decreased significantly either by introducing an extra NPSS per frame,

or by using artificial fast fading with two, four or eight antennas. An extension of the current NPSS to cover a whole subframe would decrease synchronization time slightly.

What methods could be good candidates for NB-IoT-U?

It has been shown with two candidate methods that for users with a coupling loss as weak as 154 dB, 90% will detect NPSS within 260 ms with only 5% having a residual timing error greater than 5 μs. One of the methods uses four antennas, while the other uses only two antennas but an additional NPSS per frame. Given that NSSS processing could benefit from similar methods, and that MCLs lower than 154 dB are acceptable, these methods should suffice to handle the synchronization for successful NB-IoT-U operation.

Would those candidates require a change of the current standard?

The use of BS transmit diversity would not require any change to the current standard since, by using the studied receiver algorithm, the UE is agnostic to the additional antennas. NPSS densification would definitely require a change to the standard and not be backward compatible since the broadcast channel hinders a uniform spacing of two NPSSs within a frame without moving the prevalent NPSS. Extending the NPSS by appending three new symbols would require the standard to be changed, but the sequence would be backward compatible with old UEs.

9.3.2 Implications

Operation of NB-IoT in unlicensed spectrum was approved as a work item in the MulteFire organization in June 2017. This means that the organization will initiate efforts to research and standardize the corresponding technology. Cell acquisition and synchronization is one of the more challenging parts of this process. This thesis provides insight for that development.

9.4 Future Work

This section describes some ways in which this study could be extended to provide more reliable information in the synchronization design.

The full synchronization procedure, consisting of NPSS and NSSS processing, should be measured. While doing this, improvements to the NSSS processing should be explored and taken into account.

Interference could be added into the model to more accurately reflect the environment in unlicensed spectrum. Interference could easily and perhaps accurately be modeled as an additional source of white noise. Separate studies should be carried out to motivate this and to determine appropriate interference levels.

Proper design of transmit filters could be done to potentially show improved results.

The optimal code cover should be used. It could be found by joint optimization of code cover with the combining weights, w_k. Knowing exactly how this should be carried out require some more thorough theoretical analysis of the estimation bounds and how they relate to high percentile PSTs. Of course, it could still be worthwhile to explore totally different synchronization algorithms.

Antenna code books could be studied further. First of all it should be clarified whether increased code book sizes provide any useful gain. Then code book sorting could be studied to make sure the precoders are deployed in the correct order (applicable only to the case of large code books). Finally, studies on the effect of relaxing the EGB requirements should be done.

Totally different methods for transmit diversity could also be studied. *Differential space-time codes* is an example of a group of transmit diversity schemes that does not require channel knowledge in the transmitter nor in the receiver. The applicability of such schemes in the synchronization phase could be interesting to explore.

List of Figures

List of Tables

Bibliography

[1] ITU-R, *IMT Vision Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond*, September 2015. Recommendation M.2083-0. Cited on pages 1 and 19.

[2] Q. Incorporated, "NB-PSS and NB-SSS Design (Revised)," TS R1-161981, 3rd Generation Partnership Project (3GPP), Mar. 2016. Cited on pages 2, 24, 25, 51, 57, and 66.

[3] A. Adhikary, X. Lin, and Y.-P. E. Wang, "Performance evaluation of NB-IoT coverage." Submitted to IEEE Veh. Technol. Conf. (VTC), September 2016. Cited on pages 2, 51, and 57.

[4] M. Morelli, C. C. J. Kuo, and M. O. Pun, "Synchronization techniques for orthogonal frequency division multiple access (OFDMA): A tutorial review," *Proceedings of the IEEE*, vol. 95, pp. 1394–1427, July 2007. Cited on pages 12, 15, and 16.

[5] T. M. Schmidl and D. C. Cox, "Robust frequency and timing synchronization for OFDM," *IEEE Transactions on Communications*, vol. 45, pp. 1613–1621, December 1997. Cited on pages 15 and 16.

[6] K. Shi and E. Serpedin, "Coarse frame and carrier synchronization of OFDM systems: A new metric and comparison," *IEEE Transactions on Wireless Communications*, vol. 3, pp. 1271–1284, July 2004. Cited on pages 16 and 25.

[7] M. Morelli and U. Mengali, "An improved frequency offset estimator for OFDM applications," *IEEE Communications Letters*, vol. 3, pp. 75–77, March 1999. Cited on pages 16 and 25.

[8] Y.-P. E. Wang, X. Lin, A. Adhikary, A. Grövlen, Y. Sui, Y. W. Blankenship, J. Bergman, and H. S. Razaghi, "A primer on 3GPP narrowband internet of things (NB-IoT)," *CoRR*, vol. abs/1606.04171, 2016. Cited on pages 20 and 27.

[9] 3GPP, "Cellular system support for ultra-low complexity and low through-

put Internet of Things (CIoT)," TS 45.820-d10, 3rd Generation Partnership Project (3GPP), Nov. 2015. Cited on pages 20, 53, and 55.

[10] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation," TS 36.211, 3rd Generation Partnership Project (3GPP), Jan. 2017. Cited on page 21.

[11] B. Popovic, "Generalized chirp-like polyphase sequences with optimum correlation properties," *IEEE Transactions on Information Theory*, vol. 38, pp. 1406–1409, July 1992. Cited on page 24.

[12] Q. Incorporated, "NB-PSS and NB-SSS Design," TS R1-161936, 3rd Generation Partnership Project (3GPP), Mar. 2016. Cited on page 25.

[13] E. Dahlman, S. Parkvall, and J. Sköld, *4G, LTE-Advanced Pro and The Road to 5G*. London: Elsevier Science, 2016. Cited on page 30.

[14] M. Loy, R. Karingattil, and L. Williams, "ISM-Band and Short Range Device Regulatory Compliance Overview," TS SWRA048, Texas Instruments, May 2005. Cited on pages 31 and 33.

[15] FCC, "Code of Federal Regulations, Title 47, Part 15," TS 47 CFR 15, Federal Communications Commission (FCC), Feb. 2017. Cited on page 31.

[16] ECC, "ERC Recommendation 70–03," TS 70–03, European Conference of Postal and Telecommunications Administrations (CEPT), Feb. 2017. Cited on page 33.

[17] A. Leukhin and E. Potehin, "Binary sequences with minimum peak sidelobe level up to length 68," *ArXiv e-prints*, Dec. 2012. Cited on pages 38 and 39.

[18] Huawei, HiSilicon, Neul, "On cell search and system acquisition time improvements," TS R1-1707027, 3rd Generation Partnership Project (3GPP), May 2017. Cited on pages 39, 51, and 57.

[19] R.-A. Pitavaland, H.-L. Määttänen, K. Schober, O. Tirkkonen, and R. Wichman, "Beamforming codebooks for two transmit antenna systems based on optimum grassmannian packings," *IEEE Transactions on Information Theory*, vol. 57, pp. 6591 – 6602, Oct 2011. Cited on page 46.

[20] A. Medra and T.-N. Davidson, "Flexible codebook design for limited feedback systems via sequential smooth optimization on the grassmannian manifold," *IEEE Transactions on Signal Processing*, vol. 62, pp. 1305 – 1318, Jan 2014. Cited on pages 46 and 47.

[21] 3GPP, "Radio transmission and reception," TS GSM 05.05, 3rd Generation Partnership Project (3GPP), Mar. 1996. Cited on page 53.

[22] P. Dent, G. Bottomley, and T. Croft, "Jakes' fading model revisited," *IEEE Electronic Letters*, vol. 29, pp. 1162–1163, June 1993. Cited on page 53.