UNIVERSITY OF TWENTE.

Department of Electrical Engineering
Telecommunication Engineering Group

Synchronization performance of noise-based frequency offset modulation

by

Thomas Schellekens

Master thesis
June 2010

Supervisor: dr.ir. M.J. Bentum
Advisors: dr.ir. A. Meijerink
dr.ir. G.J. Heijenk

Summary

The aim of this thesis is to investigate whether noise-based frequency offset modulation (FODMA) is a viable candidate for a wideband radio link in an ultra-low-power wireless sensor network (WSN). Ultra-low-power wireless sensor networks are characterized by low duty-cycles and short data packets. Wideband radio transmission has many advantages such as robustness against narrowband fading and interference, and no requirement for a radio frequency license. It is therefore attractive to use a wideband transmission scheme in a WSN. There are different ways to achieve wideband transmission, and we will be looking at Direct Sequence Spread Spectrum (DSSS) and FODMA.

Wideband radio receivers are divided into functional parts, namely, the front-end, the despreading and sampling block, and the baseband processing block. For many applications, the front-end can be assumed to consume most power. We seek to minimize the power consumption of the front-end through reducing the time required for the despreading and sampling block to synchronize, such that we can reduce the time the front-end is operating. The despreading and sampling block needs to synchronize several parameters to be able to despread succesfully and sample succesfully. Long synchronization periods are a problem when data packets are very short, since the synchronization period then requires relatively much energy. To save energy in a ultra-low-power WSN with short data-packets, FODMA was proposed since it has a suspected short synchronization time. This thesis aims to quantify the synchronization period for a FODMA receiver, and compare it to a benchmark wideband system, DSSS.

The required synchronization time for DSSS was found to depend on the spreading factor, that is, the ratio between the bandwidth of the signal on the channel and the bandwidth of the information signal at baseband. This is because larger spreading factors require longer pseudo-noise (PN) codes when a single symbol is spread using the full code, and the DSSS system has to perform a search over the code to find the right starting position of the code. The DSSS receiver also requires some time for 'close code tracking'. Once the starting position of the code is found and tracked, symbol timing can be derived when a single symbol is spread using the entire PN code.

The synchronization for FODMA was found to depend on the acquisition behaviour of a phase-locked loop (PLL), since the frequency offset created by the oscillator in the

transmitter has to be phase-locked in the receiver. The time for a PLL to enter phase lock was found to decrease inversely proportional with the loop gain. Higher loop gains are thus desirable for quick acquisition, but they also make the PLL sensitive to noise, that is, high gains can cause the PLL to lose lock when there is a small disturbance to the signal. The relationship between the loop gain and mean time to lose lock were found in the literature. Some modifications were made to the PLL to deal with a biphase-modulated carrier signal. For FODMA, the symbol timing estimation was found to become dominant above SNRs of 16 dB. Above these SNRs, the time for the PLL to enter phase lock can become much shorter than one bit period.

For FODMA, the required synchronization time was found to be independent of the spreading factor. FODMA has a synchronization time in the order of one hundred to just several bit periods for a range of SNRs (1–100). This is much shorter than the synchronization time for DSSS, whose synchronization time in bit periods is about twice the spreading factor for large SNRs (> 10), below which it is larger.

This leads us to conclude that FODMA would be a good radio transmission scheme for ultra-low-power wireless sensor networks with low duty-cycles and short packets, at least as far as the synchronization performance is concerned. However, FODMA does require a better signal quality at the front-end to achieve the same bit-error-rate as DSSS. This is in fact at least 11 dB, depending on the received SNR per bit and the spreading factor, and it remains to be seen in which scenarios the shorter synchronization time of FODMA outweighs this drawback.

Contents

List of acronyms

ADC	analog-digital converter
AWGN	additive white Gaussian noise
BER	bit error rate
BPSK	binary phase shift keying
BB	baseband
CDMA	code division multiple access
CMOS	complementary metal oxide semiconductor
DHTR	delay-hopped transmitted-reference
DLL	delay-locked loop
DPSK	differential phase shift keying
DSP	digital signal processing
DSSS	direct sequence spread spectrum
EMI	electro-magnetic interference
EEMCS	Electrical Engineering, Mathematics and Computer Science
FCC	Federal Communications Commission
FHSS	frequency hopping spread spectrum
FODMA	frequency offset division multiple access
FSR	frequency-shifted reference
IC	integrated circuit
IEEE	Institute of Electrical and Electronic Engineers

IR	impulse radio
IF	intermediate frequency
LFO	low-frequency oscillator
LNA	low-noise amplifier
MAC	medium access control
OFDM	orthogonal frequency division multiplexing
PAM	pulse amplitude modulation
PAN	personal area network
PDF	probability distribution function
PLL	phase-locked loop
PN	pseudo-noise
PSK	phase shift keying
RF	radio-frequency
RFID	radio frequency identification
SNR	signal-to-noise ratio
SRR	short range radio
TDL	tau-dither loop
TE	Telecommunication Engineering
TR	transmit-reference
UT	University of Twente
UWB	ultra-wideband
VCC	voltage-controlled clock
VCO	voltage-controlled oscillator
WSN	wireless sensor network

List of symbols

A	Amplitude of the incoming signal
B_f	Bandwidth of the second bandpass filter in DSSS acquisition
B_l	Loop noise bandwidth of the phase-locked loop
B_i	Pre-squarer filter bandwidth of the phase-locked loop
E_b	Energy per bit
J	Number of symbols used in symbol timing estimation
K	False alarm penalty
K_1	Amplitude of the locally generated signal in the PLL
K_m	Multiplier gain
K_o	VCO gain
L	Signal attenuation due to chip update misalignment
N_0	Single sided noise spectral density
P_fa	False alarm probability
P_d	Detection probability
q	Number of search positions in serial search DSSS acquisition
R	Data rate
S	Spreading factor
S_opt	Optimal spreading factor for FODMA
S_L	Squaring loss for phase-locked loop
S_m	Power spectral density of the modulation format
$\overline{T}_\mathrm{acq}$	Mean acquisition time
T_c	Chip time
T_d	Dwell time of the integration block
T_pll	Required time for the phase-locked loop to enter phase-lock
T_s	Mean time between cycle slips
U	Signal to noise ratio for symbol timing estimation
α	Excess bandwidth factor of the Nyquist-shaped data pulse
β	Ratio of the loop bandwidth to the pre-squarer filter bandwidth
γ	SNR after despreading
γ_DSSS	SNR after despreading for DSSS

γ_{FODMA} SNR after despreading for FODMA
δ Correlator spacing of the DSSS tracking system
ζ Damping factor of the phase-locked loop
η DSSS acquisition decision treshold
θ Phase of the incoming signal in the PLL
$\hat{\theta}$ Phase of the locally generated signal in the PLL
μ True symbol timing
$\hat{\mu}$ Estimated symbol timing
ρ Inverse of phase error variance in the phase-locked loop
τ Gain-normalized time for the PLL
ϕ Phase error of the PLL
ϕ_ϵ Allowable phase error of the PLL

Chapter 1

Introduction

1.1 Context

This master thesis deals with radio transmission aspects of wireless sensor networks (WSNs). WSNs can be used for many applications, and they may have widely different requirements. Existing WSN applications range from wildlife monitoring [1] to ingestable sensors used in medicine [2]. In the former case, transmission ranges of more than one kilometer are required. In the latter, a transmission range of 50 centimeters is sufficient. The data rates involved can also vary greatly, from megabits to just bits per second. The transmission range, together with the the data rate, largely determines the energy requirements of nodes in the network. In this thesis, the WSNs considered are ultra-low power WSNs.

Ultra-low power WSNs are envisioned for applications where small amounts of data need to be sent over short ranges. An example would be the monitoring of humidity in agricultural fields, where humidity readings (only a few bytes) are transmitted twice per day via an ad-hoc wireless network infrastructure. The limited functionality of the network allows for relatively cheap production of the individual nodes and small-scale integration. This is a step towards so called 'smart dust' networks [3], where individual nodes are envisioned to be the size of dust particles. Within the Electrical Engineering, Mathematics and Computer Science (EEMCS) faculty of the University of Twente (UT) [4], of which the Telecommunication Engineering (TE) group is part, WSNs are one of the focus research areas. The TE group, within its short range radio (SRR) programme, focuses on the radio transmission aspects of these ultra-low-power WSNs.

Radio transmission of the data within an ultra-low-power WSN would obviously be responsible for a large part of the energy consumption of a node. Keeping the power consumption for radio transmission minimal is very desirable, since a node usually utilizes a battery which holds limited energy.

Replacing the batteries in WSNs is often impractical or not economical, for example

when the WSN is integrated into a structure, or scattered over a large area. Low power consumption possibly also allows energy-scavenging schemes to power the nodes [5]. Energy-scavenging schemes draw ambient energy from the environment, such as solar, vibrational or radio-frequency (RF) energy. In a small research project preceding this master thesis, an inventarisation of ultra-low-power WSNs was made [6].

Wideband radio transmission of sensor data has several advantages in WSNs, and it is the subject of Section 1.2. Subsequently, we will motivate the use of a particular type of wideband transmission for ultra-low-power WSNs, transmit-reference (TR) modulation, in Section 1.3. In Section 1.4, the main research objective will be detailed. In Section 1.5, the outline of this thesis will be presented.

1.2 Wideband radio

As the name implies, wideband radio uses a large bandwidth for transmission. In wideband transmission, the power spectral density is very low, although the total power remains the same. This is illustrated in Figure 1.1.

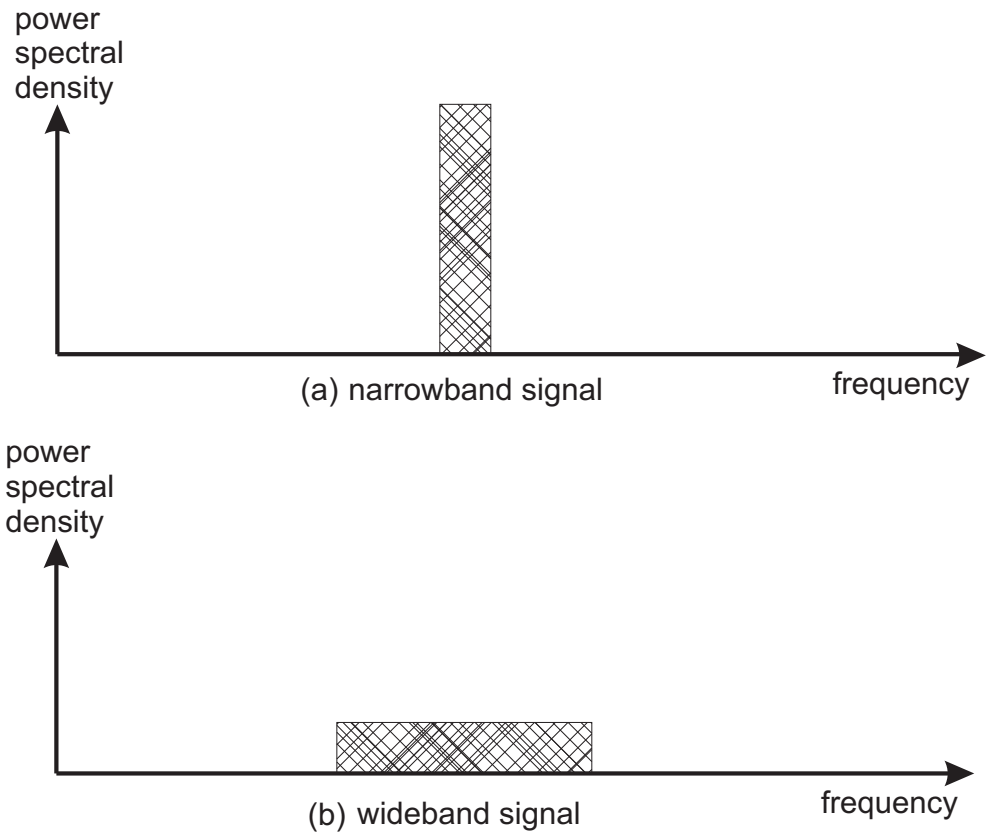

Figure 1.1: Power spectral density of a narrowband (a) and wideband (b) signal

There are four important advantages of using wideband radio, some of which specifically apply to WSNs. The first is that the emitted power per hertz of bandwidth is

so low that the interference caused to licensed users (using narrowband carriers) is very small. Vice versa, a narrowband carrier only affects a very small portion of the frequency spectrum used by wideband systems, such that the interference to the wideband systems is also very small. This allows both systems to coexist. A wideband system therefore usually does not have to obtain a license for use of certain frequencies, given limits on transmitted power defined by the Federal Communications Commission (FCC). Without the requirement of obtaining a license for use of part of the frequency spectrum, wideband radio allows new applications to be deployed much easier.

The second advantage is the immunity of wideband systems to multipath fading. When a narrowband signal is transmitted, it might happen that the receiver receives the line-of-sight signal, but also other versions of the signal which are reflected off various surfaces. If these reflected signals arriving at the receiver are out of phase (having a delay of half the wavelength), they interfere destructively so that the signal power decreases. The environment in which multiple signal reflections are received is called a multipath environment. Using wideband transmission, the frequencies that destructively interfere with each other at the receiver given a certain delay (a specific multipath environment) are only a small portion of the total frequency range, and plenty of signal power remains. WSNs are often employed indoors, where many surfaces may cause reflections. WSNs therefore are likely to benefit from wideband transmission. In a multipath environment, a rake receiver can be used to 'collect' several of the reflected signals and combine the signal energy contained in them by seperately synchronizing and demodulating all the reflections [7], at the cost of additional receiver complexity. Rake receivers will not be considered in this thesis. Wideband systems are robust to the possible fading caused by the reflected signals at the receiver. However, if no rake receiver is used, there is still a loss of signal energy, namely that energy that is contained in the reflections, to which the receiver is not synchronized.

A third advantage of using wideband transmission is that it allows for localization of the nodes. Wideband signals have very short autocorrelation periods, which allows time-of-arrival estimates of high accuracy. This allows for precise spatial localization when the source of the signal has a known position. The precision of the localization procedure depends on the bandwidth [8].

The fourth advantage of wideband transmission is that the requirement for medium access control (MAC) is smaller. That is, in a multiple-access scenario, there is less need for coordinating medium access, since real collisions cannot take place, that is, simultaneous wideband transmissions from other sources only raise the noise floor slightly, and do not necessarily cause packet loss.

The attractive features of wideband transmission are such that presently, several systems already employ wideband transmission for SRR, such as Zigbee [9], Blue-

tooth [10], and several wireless LAN systems. These systems use different methods to achieve wideband transmission. The three basic approaches used by wideband systems are called direct sequence spread spectrum (DSSS), frequency hopping spread spectrum (FHSS), and impulse radio (IR), respectively.

In the following three subsections, these three wideband transmission methods will be detailed. Subsection 1.2.4 will motivate the use of wideband transmission for WSNs, and highlight the synchronization issues in such networks, and ultra-low power WSNs in particular.

1.2.1 Direct sequence spread spectrum

A DSSS transmitter achieves large bandwidths by mixing a wideband signal with the information signal. This wideband signal is called the 'chip sequence' and has a larger bandwidth than the information signal. The receiver can then perform demodulation by mixing the received signal with the same chip sequence. This mixing operation is illustrated in Figure 1.2, and it results in a wideband signal as illustrated in Figure 1.1.

In Figure 1.2, the original, narrowband information signal illustrated in the time domain is at the top and denoted by (a). It corresponds to the narrowband signal (a) illustrated in Figure 1.1. The second signal from the top in Figure 1.2 is the wideband chipping sequence with which the information signal is mixed. This results in the bottom signal in Figure 1.2, signal (c), which is a wideband signal that contains the original, narrowband, information signal.

To demodulate the signal, the receiver has to mix the received signal with the same chip sequence. This has to happen in phase, that is, the receiver has to find the right alignment between the incoming (transmitted) chip sequence position and its locally generated chip sequence position, or demodulation will fail. This alignment procedure requires a period of synchronization in the DSSS receiver, and we will study it in detail in Chapter 3. A DSSS receiver can be combined with a rake receiver architecture [7], although this will make synchronization more complex, since all the rake 'fingers' (who collect different multipath components) have to be synchronized to the individual multipath components.

1.2.2 Frequency hopping spread spectrum

Another technique that achieves wideband transmission is frequency hopping spread spectrum (FHSS). In FHSS, the transmitter uses a narrowband carrier, but only transmits for a short period, before 'hopping' to another frequency. Over time, the transmitter transmits many short 'bursts' at many frequencies, such that, time-averaged, a wideband signal with low spectral density results. Because FHSS uses a narrowband carrier, it is not inherently immune against multipath fading or immune against inter-

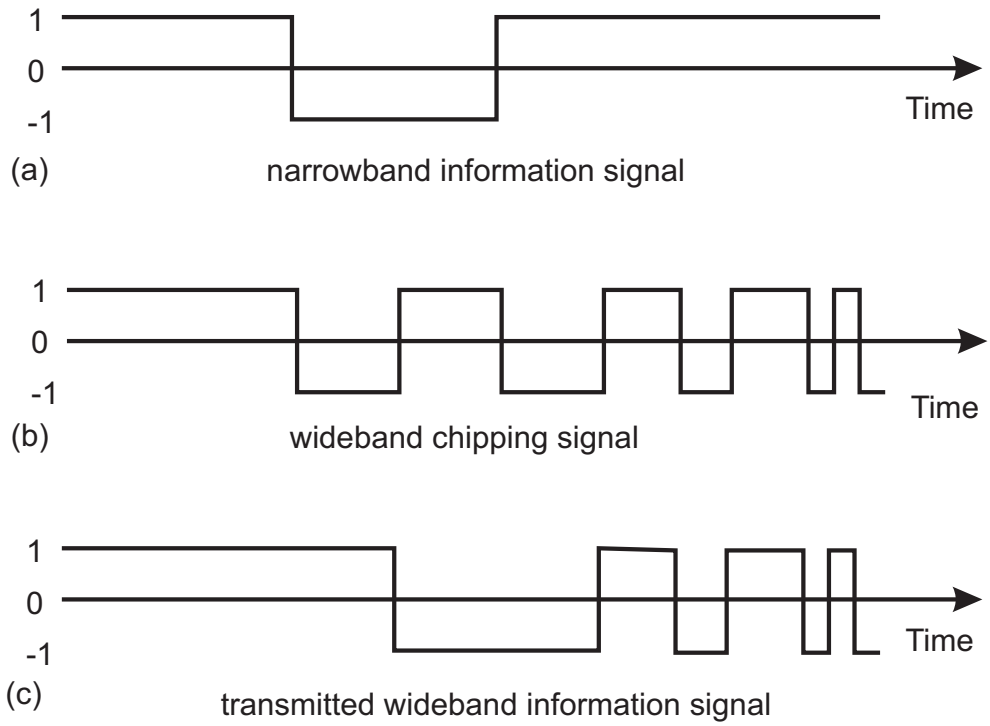

Figure 1.2: Spreading operation of DSSS, with (a) the information signal, (b) the chip sequence, and (c) the transmitted signal

ference from other narrowband systems. To mitigate some of these effects, FHSS can utilize adaptive hopping. Adaptive hopping involves the avoidance of frequencies that experience fading or interference. Possibly, it can also adjust the time it transmits on one frequency, and the frequency spacing of hops to favorably affect the transmission. The synchronization in FHSS requires that the receiver waits for a certain number of hops before it can lock onto the hopping sequence, which should be known to the receiver in advance. FHSS is not suitable for combination with a rake receiver architecture [11], since the resolvability of the multipath components is not good for this (instantaneously narrowband) system.

1.2.3 Impulse radio

Wideband transmission can also be done by transmitting very short (nanosecond) wideband pulses to which pulse position modulation is applied [12]. IR suffers from long synchronization times since the starting instant of the received pulse has to be estimated with high precision. Considerable effort is required to generate the pulses, and make them fit the spectral mask (the legal power spectral density as defined by the FCC) [13]. IR is appropriate in combination with a rake receiver, since the short duration of the pulses makes them highly resolvable in time, which also makes IR suitable for applications that require localization.

1.2.4 Wideband radio for ultra-low-power sensor networks

Existing systems such as Zigbee, currently popular for WSNs, already employ wideband radio for transmission. IEEE 802.15.4 [9], the standard for low-data-rate wideband personal area network (PAN) applications, facilitates the development of these types of applications. However, wideband transmission schemes usually suffer from long synchronization times [14, 15]. For the DSSS wideband system introduced in Section 1.2.1, the receiver has to synchronize its local generation of the spreading code (chip sequence) with the received signal. FHSS requires time to lock on to the hop sequence. Pulse-based wideband transmission schemes require time to estimate the starting instant of the pulse.

In ultra-low-power WSNs, small amounts of data are sent periodically. In the example of an agricultural WSN, humidity readings could be sent twice per day. This is called a low-duty-cycle scenario. In a low-duty-cycle scenario, the individual nodes are in sleep mode most of the time. When they wake up, they take a measurement (usually requiring little energy) and set up a radio connection to transmit their data (requiring a lot of energy). This means that every time data needs to be sent, the radio connection needs to be set up (synchronized). In a classical narrowband coherent receiver, setting up a radio connection involves phase-locking the local oscillator of the receiver to the received signal (enabling amplification of the signal out of the noise), and estimating the symbol timing, that is, estimating the optimal instant to sample the signal. In a wideband receiver, synchronization may require more steps, such as locking to the chip sequence (in DSSS), locking to the hopping sequence (in FHSS), or estimating the starting instant of the pulse (for IR).

For ultra-low-power WSNs, the synchronization time needs to be reduced as much as possible, since for small amounts of sensor data (short transmission periods), the synchronization time may comprise a relatively large portion of the transmission time, and transmission and reception of data comprises a large part of the energy consumption of the node (see Section 2.2). In the receiver, the largest energy consumer is the front-end, where the received signal is amplified. This is a key assumption and it will be detailed in Chapter 2.

One wideband transmission scheme with a (suspected) short synchronization time is transmit-reference modulation. It is also known as transmitted-reference modulation [15]. In Section 1.3, we will explain the basics of transmit-reference modulation. A shorter synchronization time would likely yield significant energy savings for both the transmitter and receiver in ultra-low-power sensor networks. Although rake receivers are not considered in this thesis, we note that a shorter synchronization period could apply to synchronization of the individual rake fingers as well.

1.3 Transmit-reference modulation

The principle of transmit-reference modulation has been known since 1922 [16]. Various architectures employing TR principles have been developed since, as described in Chapter 2 of [15]. An example of a transmit-reference transmitter and receiver is given in Figure 1.3. In Figure 1.3, TR modulation is illustrated for the case where a time-offset (delay) is used in combination with a pulse source. In Figure 1.4, the signals travelling through the different branches of the transmitter are illustrated. The pulses from the source (signal (a) in Figure 1.4) are split into two paths. The pulse travelling through the upper path in Figure 1.3 is the 'reference' pulse, which is unmodulated (signal (b) in Figure 1.4). The pulses travelling through the lower path are modulated with polar NRZ data symbols, +1, -1 (in this example). Subsequently, the modulated pulse travelling through the lower branch is delayed with a certain delay (signal (c) in Figure 1.4). This ensures that after summation of the signals from both branches, they can be distinguished. The sum of the signals is put on the channel (signal (d) in Figure 1.4). In effect, there are two signals (pulse trains) on the radio channel, namely the reference signal (signal (b)) and the modulated, delayed signal (signal (c)).

In the receiver (illustrated on the right side of Figure 1.3), the received signal (signal (a) of Figure 1.5), is split into two paths. The signal travelling through the upper branch is the undelayed received signal (signal (b) of Figure 1.5). The signal travelling through the lower branch is delayed with the same delay used in the transmitter, illustrated as signal (c) of Figure 1.5. The undelayed and delayed signal are subsequently mixed with each other, producing product (d) of Figure 1.5, which represents the data that was used in the transmitter.

The illustration of the system of Figure 1.3 utilizes pulses as carrier. However, it is also possible to use wideband noise as carrier, because the TR receiver needs no knowledge of the source signal [17], as it only correlates the two signals. The TR system then operates exactly the same. The transmitter has a modulated and delayed source signal in one branch, a reference signal in the other branch. The receiver also has exactly the same architecture. The advantage of using a noise source is that noise

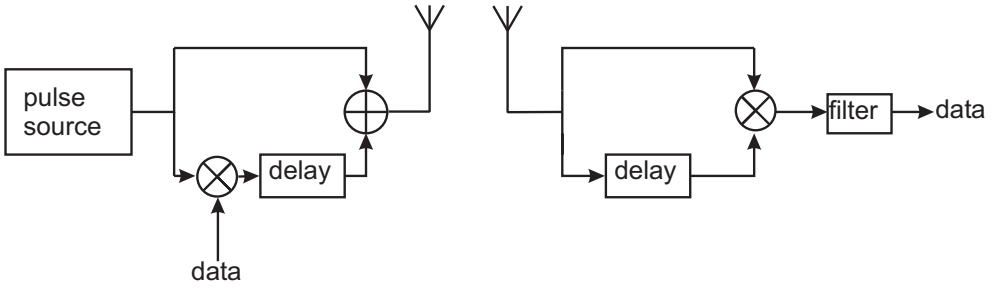

Figure 1.3: Transmit-reference transmitter and receiver

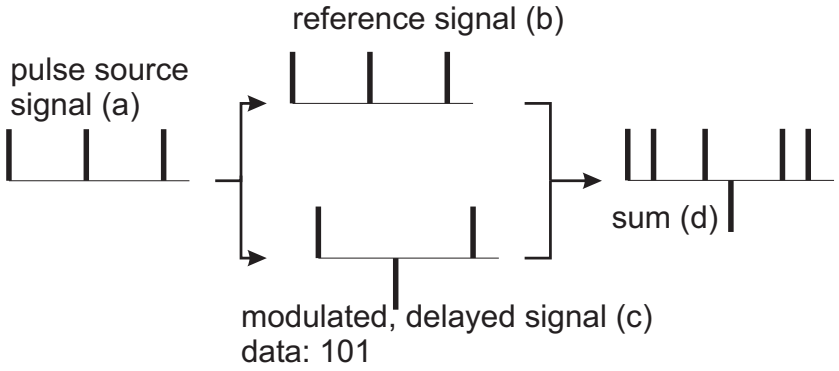

Figure 1.4: Pulses through the TR transmitter

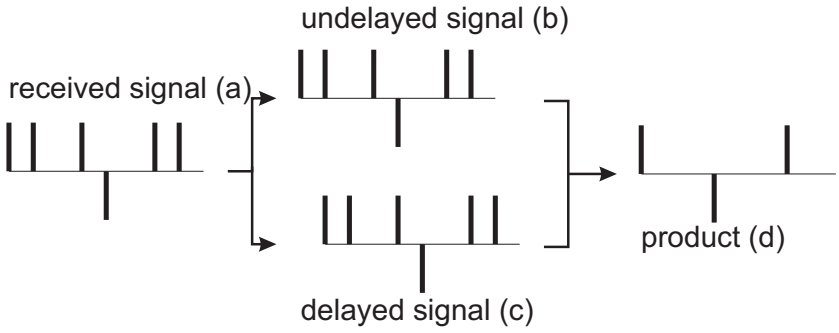

Figure 1.5: Pulses through the TR receiver

is easier to shape than pulses, which needs pulse dithering to fill the spectral mask, the legally allowed power spectral density. Since the exact shape of the signal is not important when a TR-style receiver is used, wideband noise can be used as well.

Although the time offset TR modulation scheme illustrated in Figure 1.3 is useful for understanding TR concepts, implementing a time-delay on chip is difficult [18]. This is because the delay needs to be longer than the coherence time of the noise source if the delayed and undelayed signals are to be orthogonal, which in the gigahertz bandwidth range translates to a delay line of centimeters length. Therefore, instead of using a time-delay, a frequency-offset is used. A TR system with noise as carrier and a frequency offset instead of a time offset is illustrated in Figure 1.6, as proposed in [18].

In Figure 1.6, the noise signal travelling through the lower branch is mixed with the data, after which it is multiplied by the signal coming from a low-frequency oscillator (LFO), at a frequency which much higher than the bit rate [19]. The subsequent transmission happens just as in the pulse-based time offset TR system illustrated in Figures 1.3, 1.4 and 1.5. In the receiver, the frequency offset used in the transmitter should be matched, that is, the same frequency should be used. Moreover, the receiver LFO should oscillate in phase with the oscillation in the received signal. The phase-locking of the receiver LFO is part of the synchronization procedure of the frequency-offset TR receiver, and it is subject of Chapter 4.

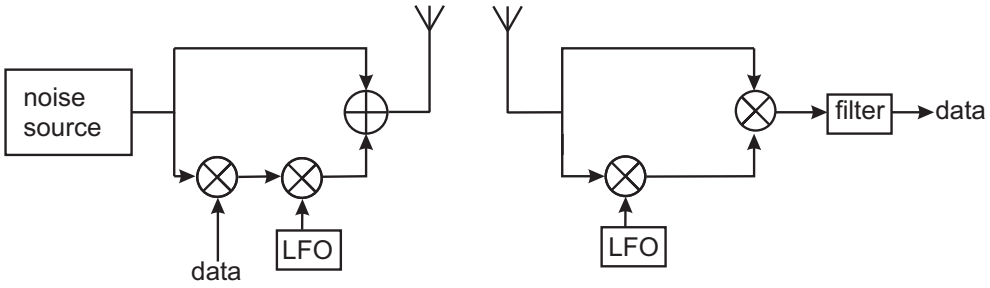

Figure 1.6: Noise-based frequency offset TR modulation

Tunable LFO components are readily available as integrated components, usually in the form of a voltage-controlled oscillator (VCO), and are much more easily integrated into a small-scale circuit than a time delay. Frequency-offset TR modulation can also be done with pulses as carrier, and this was investigated in [20]. Frequency-offset TR modulation with noise as carrier was examined by Shang [19] and Balkema [21] at the TE group. Shang found the theoretical bit error probability behaviour versus the signal-to-noise ratio (SNR) per bit, and Balkema found the same behaviour in his experimental set-up. Small-scale integration of the system in CMOS was investigated by Mahrof [22], who performed simulations of different system architectures. The performance of noise-based frequency offset TR modulation in frequency-selective fading channels was subject of a paper by Meijerink [23], where noise-based frequency offset TR modulation was found to require 10 dB higher SNR per bit to achieve the same bit error rate (BER) as a coherent binary phase shift keying (BPSK) system. Noise-based frequency offset TR modulation is not expected to have as good a BER performance as a coherent phase shift keying (PSK) system. Its supposed strength is its applicability to ultra-low-power WSNs. The research by Shang, Balkema, Mahrof and Meijerink characterize noise-based frequency offset TR modulation in terms of its BER performance. Actual application of noise-based frequency offset TR modulation requires knowledge of its BER performance as well as its synchronization behaviour. Therefore, this master thesis will aim to quantify the synchronization time of a noise-based frequency offset TR modulation scheme, to see whether noise-based frequency offset TR modulation is indeed a viable candidate for radio transmission in ultra-low-power WSNs. Shang and Balkema [19, 21] have both investigated the multi-user performance of the TR system using noise as a carrier and a frequency offset. They name the system frequency offset division multiple access (FODMA), for its capacity to allow multiple access on a channel enabled by using different LFO frequencies. We will also use the acronym FODMA for noise-based frequency offset TR modulation.

1.4 Research objective

FODMA is a candidate transmission scheme for ultra-low-power wireless sensor networks. Its attractive features are wideband transmission, the requirement of only a few components (implementation simplicity), and its suspected short synchronization time. A short synchronization time is expected to generate significant power savings for WSN nodes which only receive small amounts of data in a low-duty-cycle scenario, which is further illustrated in Section 2.2. In this master thesis, we will quantify the synchronization time of FODMA, and compare it to the synchronization time of the most basic and well-known wideband scheme, DSSS. DSSS was chosen to be the benchmark wideband system because it is a simple system compared to FHSS, and simplicity of implementation is an important criterion for simple sensor nodes [18]. Furthermore, it is in popular use in WSNs, through the ZigBee standard, a DSSS-based SRR platform. Only the synchronization time of the receiver will be characterized, from which we will draw conclusions about the viability of the use of this technique in ultra-low-power wireless sensor networks.

1.5 Outline of the thesis

In Chapter 2, a model of a radio receiver is given, facilitating a comparison between a FODMA receiver and a DSSS receiver. Chapter 2 also discusses synchronization for radio receivers in general, and details the assumptions under which the comparison is made. Chapter 3 analyzes the synchronization time of a simple type of DSSS system. Chapter 4 will analyze the synchronization time for the FODMA technique. Chapter 5 will present a comparison between the two systems in different scenarios. In Chapter 6, we will draw conclusions, and recommend future research directions.

Chapter 2

Receiver model

2.1 Receiver blocks

This thesis will compare the synchronization time of DSSS and FODMA radio receivers. A shorter synchronization time is expected to yield considerable energy savings for small data packets (sensor readings) where the synchronization time takes up a significant portion of the total transmission period. Comparing these different systems is complex as a result of different architectures built for different purposes. Only the high-level functionality —to receive data— is the same, while the implementation differs. This chapter provides a framework for comparing the systems. A top-down approach will gradually determine which aspects are fixed for both receivers, and which remain different. The first step will be to seperate both receivers' functionality into parts. A general receiver model is given in Figure 2.1.

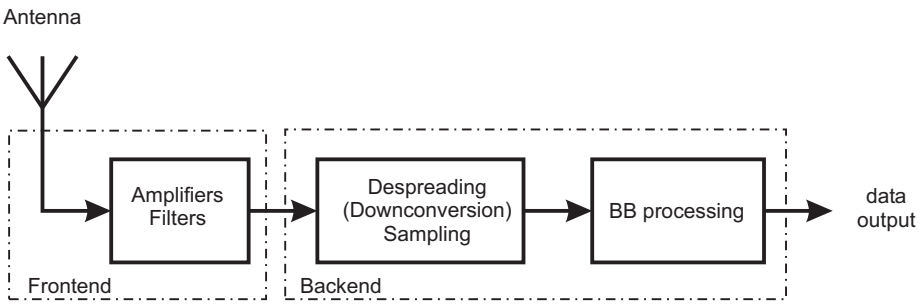

Figure 2.1: Generic receiver model

The first element of the receiver is the antenna. It is followed by the amplification and filtering block. Together, the antenna and the amplification and filtering block are usually called the front-end. Among radio receivers, there are many ways to implement this block. For example, amplification usually happens directly after the antenna element, at RF frequencies, but it is also frequently done after downconversion to an intermediate frequency, to favorably affect the SNR, or both. A filter is usually present

to limit the noise into subsequent blocks, for example the interference produced by other radio sources, or the mirror signals of superheterodyne receivers. The most widely used measure for signal-quality at the front-end is E_b/N_0. It is frequently called the SNR per bit. It represents the fraction between the energy per bit E_b and the noise spectral density N_0. It is a dimensionless measure of the signal quality at the front-end, independent of bandwidth and modulation type. We will not look at the implementation of the amplification and filtering block, but assume that this block consumes the largest amount of power in the receiver, mostly consumed by the low-noise amplifier (LNA), with the antenna being passive. A shorter synchronization time of the subsequent block causes less energy to be consumed by the front-end, because it can be turned off for a longer period of time. A radio that is is in sleep mode most of the time is said to be operating in a low duty-cycle scenario. An internal clock ensures that the node wakes up at given intervals, during which data is sent and received.

We will assume that the energy consumption of the amplification and filtering block is equal for both receivers. This allows us to use the synchronization time of the respective receivers as a direct measure of their energy consumption, since the front-end consumes most energy in the receivers.

The next block is the despreading, downconversion and sampling block. It converts the RF signal to a baseband (BB) signal. The despreading, downconversion and sampling block has to synchronize to the received signal, either despread it directly to baseband or to an intermediate frequency (IF), and perform data detection (sampling). Downconversion is only required if an IF band is used. The implementation of this block is different for the DSSS and FODMA receiver, and it will be subject of Chapter 3 and 4, respectively.

After the despreading and sampling block, we have a BB processing block. In this block, further processing of the detected (discrete) data is done, for example, decoding line codes, higher-level error correcting codes, but also further amplification and filtering. The implementation of this block is not subject of this thesis, they are assumed identical for both the DSSS and FODMA receiver.

The implementation of any of the blocks of the above model can be done in the digital as well as in the analog domain. Processing data digitally has advantages, one of which is better signal quality. Another advantage is the possibility for parralel processing, which can be used to perform quick synchronization. Sampling, however, requires considerable power, especially at high (megabit) rates [24]. Many hybrid analog/digital architectures are possible as well, with some blocks implemented digitally and others in the analog domain. An obvious candidate for a hybrid design is to perform despreading with an analog circuit (where large bandwidths are involved), and subsequent operations digitally, after analog-to-digital conversion. However, sampling costs energy and requires a more complex receiver, so this study will focus on all-analog receivers.

2.2 Synchronization in low-duty-cycle scenarios

The actual energy savings as a result of a reduced synchronization time depends on the reduction itself and the size of the data payload. In Figure 2.2, the payload is large, and the synchronization period relatively small. This is the usual case in communication systems. A twelve-fold reduction in synchronization time reduces the total transmission time of (b) about 60 percent of (a).

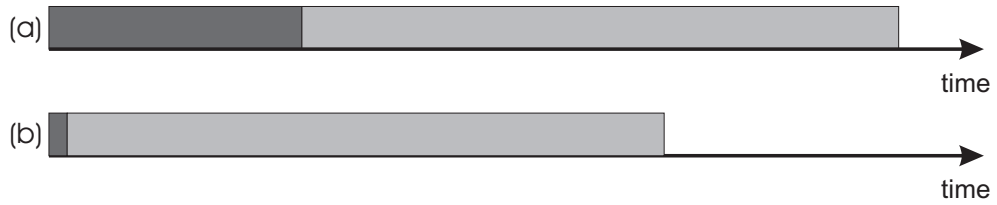

Figure 2.2: A normal scenario, where payload transmission (in grey) takes longer than the synchronization (in dark grey). (a) packet transmission with a 'normal' synchronization period. (b) packet transmission with a reduced synchronization period.

In Figure 2.3, the payload is relatively small, and the synchronization time makes up a relatively large portion of the transmission time. A twelve-fold reduction in synchronization time yields a much larger reduction in transmission time; the total transmission time in scenario (b) is now only 21 percent that of (a). The scenario illustrated in Figure 2.3 is the one prevalent in WSNs, and ultra-low-power WSNs in particular. This is the motivation for research into the duration of synchronization period for both FODMA and DSSS.

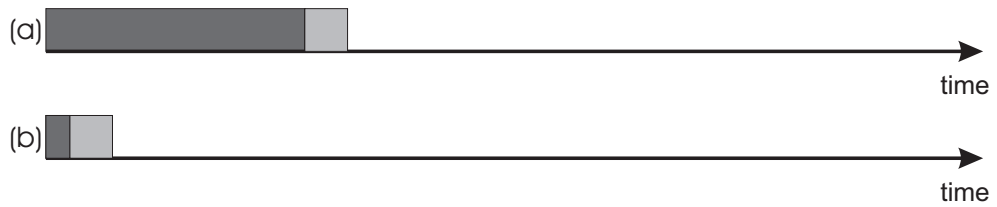

Figure 2.3: Low-duty-cycle scenario where payload transmission (in grey) is a small part of total transmission time, with the synchronization (dark grey) taking up most time. A similar reduction in synchronization time from (a) to (b) as in Figure 2.2 now reduces total transmission time significantly.

2.3 Synchronization of the despreading and sampling block

The implementation of the despreading and sampling block is different for the DSSS and FODMA receiver. Various levels of synchronization are required to be able to perform despreading and sampling.

A coherent narrowband receiver usually performs two types of synchronization: carrier and symbol synchronization. Carrier synchronization concerns the estimation of the phase of the carrier. The phase of the carrier is usually unknown in the receiver because of the propagation through the radio channel. It needs to be recovered to be able to demodulate the signal coherently. When it is not, it induces a performance penalty on the system. The carrier phase recovery is usually performed by a phase-locked loop (PLL) [25]. Symbol synchronization (also called symbol timing, timing recovery) involves the determination of the optimal sampling instant of the signal. It is dependent (among others) on the SNR and the number of baseband symbols 'seen' so far, that is, the number of symbols that can be used in the estimation.

Narrowband systems can recover carrier and symbol timing either seperately or jointly. When recovering them seperately, the carrier phase estimation is performed first, and symbol timing thereafter. The particular PLL used depends on the signal quality, the signal format — for instance, pulse amplitude modulation (PAM) or PSK— and whether the signal on which to lock is unmodulated, modulated with a known sequence, or modulated with an unknown sequence [25]. This thesis will not consider issues related to modulation formats, and assume BPSK modulation for simplicity.

Symbol timing, when recovered seperately from the carrier phase, is, similarly to phase estimation, dependent on the signal quality and whether a known or unknown symbol sequence is transmitted [25]. Several symbols need to be 'seen' by the receiver to estimate the optimal sampling instant with some accuracy. Compared to the PLL, relatively simple expressions were found that describe the behaviour of a symbol timing recovery circuit. For a given SNR, a number of symbols seen (and used in the estimation), bandwidth and pulse shape, a closed expression exists for the accuracy of the estimation of the optimal sampling instant [26]. Symbol timing recovery will be required for FODMA, and will be subject of Section 5.2.

Narrowband systems can also perform carrier and symbol timing estimation jointly. However, this thesis will treat symbol timing recovery seperately because it allows more detailed insight into the greatest contributors to the synchronization time.

For wideband systems, additional synchronization is usually required. In Chapter 3, we will see that the DSSS receiver has to find the right alignment between the chip sequence of the incoming signal and the locally generated chip sequence. In Chapter 4, we will see how the LFO in the FODMA receiver is phase-locked to the received signal.

In our comparison of the synchronization time of both systems, we will assume that the SNR after despreading, γ, is equal. A complication arises here; namely that FODMA performs despreading worse than DSSS, and thus requires a better signal quality at the front-end to arrive at the same SNR after despreading. However, we need to take the SNR after despreading equal because the systems need to be compared operating at the same BER. More observations concerning this issue with the comparison of both systems are made in Chapter 5.

The despreading and sampling blocks for both systems will have different power consumptions. No attempt will be made to characterize power consumptions of these blocks, since it is highly dependent on the specific integrated circuit technology used to implement it. Instead, we will assume it to be small compared to the power required by the front-end, such that the total energy consumption of the receiver is a function of the synchronization and transmission time only.

The assumption that the power consumption of the despreading and sampling block is small relative to the power consumption of the front-end also helps to minimize another problem: the dependency of the required synchronization time for both systems on the power consumption of their respective despreading and sampling blocks, an issue which we will touch upon in Chapter 3 and 4 as well.

2.4 Summary

The most important assumption of the current study is that a receiver consumes an amount of energy roughly proportional to its duty cycle. The required power for the front-end is large compared to the power consumed by the despreading and sampling block. A reduction of its duty cycle through a reduced synchronization period then reduces energy consumption roughly proportionally. The savings should be especially significant for a low-duty-cycle scenario with very short data packets (sensor readings), where the synchronization time is a significant portion of the transmission time, as explained in Section 2.2. Our evaluation measure for both systems will therefore be the synchronization time reduction. Other assumptions of the study include the operation in a single-user, simplex scenario, and assumption of a simple additive white Gaussian noise (AWGN) channel, without multipath fading, narrowband interference, or doppler effects. Furthermore, it is important that we consider synchronization time performance given an equal SNR after despreading. This allows us to compare the systems at the same BER. This does not take into account that FODMA performs despreading inherently worse than DSSS (i.e. it yields a worse SNR given the same E_b/N_0), but this study is solely meant to see whether FODMA synchronizes quicker than DSSS. The E_b/N_0 penalty for FODMA actually occurs at the transmitter, which might not be as power-critical as the receiver, that is, it can be a node with a relatively

large power supply in a heterogeneous network (which are common in WSNs). The energy savings of short synchronization periods might also be large enough to offset the increased transmission power required for a FODMA transceiver, if the data packets are short enough. However, we will not investigate these system issues, but only look into the receiver side, and see whether the synchronization performance is actually better than for DSSS.

Chapter 3 will deal with DSSS-specific topics of code acquisition and tracking. Chapter 4 will deal with acquisition of the frequency-offset phase in a FODMA receiver. Chapter 5 will present a numerical comparison of both systems, highlight to what extent synchronization precision is comparable for both systems, and detail symbol timing estimation, which is required for both systems. Some observations about the requirements for E_b/N_0 for FODMA are also made in Chapter 5.

Chapter 3

Synchronization in direct sequence spread spectrum

3.1 Acquisition and tracking

Synchronization in the DSSS system consists of four steps: coarse code acquisition, code tracking, IF carrier phase synchronization for downconversion, and symbol timing estimation. Coarse code acquisition is the step where the locally generated code is put into coarse (rough) alignment with the received signal, usually to within half a chip time T_c. It searches all possible alignments q for the one that produces the largest correlation. Subsequently, the tracking operation starts.

The tracking operation continuously updates the clock driving the local pseudo-noise (PN) code generator, increasing or decreasing its frequency according to the correlation produced by undelayed and delayed multiplication of the local PN generator and incoming signal. For example, if the incoming signal is slightly delayed, the delayed multiplication in the receiver will produce better correlation, and the clock driving the PN generator will decrease its frequency, delaying the locally generated PN sequence and keeping it in precise synchronization with the incoming signal.

Downconversion is necessary for DSSS receivers which are implemented as IF band systems [15]. The transmitter would then upconvert the baseband information signal to an IF band around a center frequency, after which the DSSS spreading operation is performed. Receiver architectures can either first perform downconversion, or perform despreading first. The latter type is most common and used here.

Symbol timing can be acquired automatically in DSSS once the PN code is acquired. Since one symbol spans many PN code bits, correct acquisition of the PN code will yield the correct symbol timing estimation if they have a fixed relation in the transmitter (that is, when the PN code length multiplied by the chip time T_c is equal to one bit period or an integer multiple of a bit period).

This chapter will be concerned with acquisition and tracking of the DSSS chip

sequence ('code'), and assume that the DSSS system is not implemented as an IF band system. The steady state behaviour of the tracking operation and the mean time to lose lock are important when the quality of the information signal is examined, but this chapter only examines acquisition and tracking since it is the synchronization time that we seek to minimize in order to minimize the power consumption of the receiver, as explained in Chapter 2. Chapter 5 will provide numerical results of the analysis in this chapter.

3.2 Acquisition

Code acquisition in DSSS receivers is a probabilistic process. Given a fixed amount of time, acquisition can only be achieved with limited probability. That probability is directly dependent on the SNR of the incoming signal. For example, if we have an incoming signal of excellent quality, we can achieve synchronization with probability close to one after a search of all the possible code alignments q. However, if our signal is of lesser quality, or if we cannot search the entire space of code alignments due to time constraints, our probability of synchronization drops. Part 4 of the book by Simon, Omura, Scholtz and Levitt [15] contains an extensive treatise on acquisition and tracking for DSSS and FHSS systems. In Section 3.3, we give a description of the work of the authors of [15], on which we base the results for the required acquisition time for the DSSS system. Out of many possible DSSS receiver architectures, we chose the simplest architecture, a single-dwell, serial search architecture. The simplest architecture was chosen because implementation simplicity is a desirable characteristic for a receiver in a wireless sensor network [18]. Implementation simplicity is one of the reasons why we investigate FODMA for wireless sensor networks. This does not mean per se that the simplest transceiver also requires the least synchronization time; a more advanced receiver architecture might require less time in certain scenarios. However, for a first comparison, we will analyze the simplest architecture to acquire a measure of the synchronization time of a DSSS system, which will then be compared to FODMA. The simple single-dwell, serial search DSSS receiver is described in the next section. For this system, the mean acquisition time is described as a function of the properties of the incoming signal.

3.3 Direct sequence spread spectrum acquisition model

The architecture of a single-dwell, serial search DSSS acquisition block is given in Figure 3.1. It represents the acquisition block of the despreading and sampling block, as described in Figure 2.1, the general model of a radio receiver.

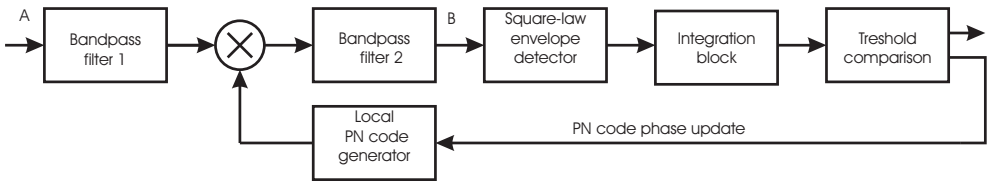

Figure 3.1: DSSS synchronization block for a single-dwell, serial search architecture

To achieve despreading of the incoming wideband signal (at point A), the local PN code generator needs to be in phase with the incoming signal. In other words, the local PN code generator has to find the starting instant of the spreading sequence in the incoming signal. The acquisition block pictured utilizes a sliding correlation mechanism, that is, it discretely shifts the position of the local code against the incoming signal to find the position that yields the greatest correlation. The treshold comparison block determines if the correlation produced is below or above a certain treshold, and thus if the codes are in phase or not. It is not hard to imagine that noise in the incoming signal might cause a false alarm (the treshold is crossed, but the true code alignment is not found), or cause the correct code alignment to be missed (the treshold is not crossed, while the true code alignment was present).

Point A in Figure 3.1 denotes the incoming signal, having a wide bandwidth. Here, the signal is still spread. It is characterized by E_b/N_0, and the bandwidth of the signal on the channel, B_c. To arrive at the SNR γ_DSSS in the bandwidth B_f of bandpass filter 2 at point B, the data rate R needs to be known and several effects need to be taken into account. The first effect is caused by the relation between the power transfer of bandpass filter 1 and the spectral shape of the PN code on the channel. This effect is detailed in Subsection 3.3.1. The second effect concerns the fact that the sliding correlation of the locally generated chip code with the incoming signal may be slightly misaligned, that is, the discrete sliding mechanism might miss the true correlation peak. This effect is described in 3.3.2. The third effect is called modulation distortion, and it is caused by the fact that the spectral envelope of the modulation format may not entirely fit through bandpass filter 2, causing a penalty to the signal power. It is described in Subsection 3.3.3.

In Subsection 3.3.4 we will look at how γ_DSSS and the three effects affect the false alarm probability and the detection probability of the treshold comparison block. In Subsection 3.3.5, we will see how the integration time of the integration block, the detection probability, and the false alarm probability finally affect the acquisition time.

3.3.1 Noise spectral density reduction

The despreading operation causes the noise power into the bandpass filter to be reduced. We will denote the noise spectral density before despreading by N_0. The despreading

operation mixes the incoming signal with the locally generated PN code. Because the spectrum of the white gaussian noise is bandlimited by bandpass filter 1, its convolution with the spectrum of the locally generated spreading code reduces its spectral height slightly. This reduction in noise spectral density is described by the authors of [15] and they derive N_0', the noise spectral density after despreading at point B as

$$N_0' = N_0 M_\mathrm{n} \tag{3.1}$$

where

$$M_\mathrm{n} = \int_{-\infty}^{\infty} T_\mathrm{c} \left(\frac{\sin(\pi f T_\mathrm{c})}{\pi f T_\mathrm{c}} \right)^2 \mid H_1(\mathrm{j}2\pi f) \mid^2 \mathrm{d}f. \tag{3.2}$$

(3.2) expresses the reduction of the noise spectral density N_0 to the noise spectral density after despreading, N_0' as factor M_n. Bandpass filter 1, with transfer function $H_1(\mathrm{j}2\pi f)$, should pass most of the (spread) signal. The spectrum of the spreading code (a square wave) is given by the sinc function.

3.3.2 Chip update misalignment

The second effect to be taken into account relates to a possible misalignment of the half-chip increments of the sliding correlation. We illustrate the sliding correlation operation in Figure 3.2.

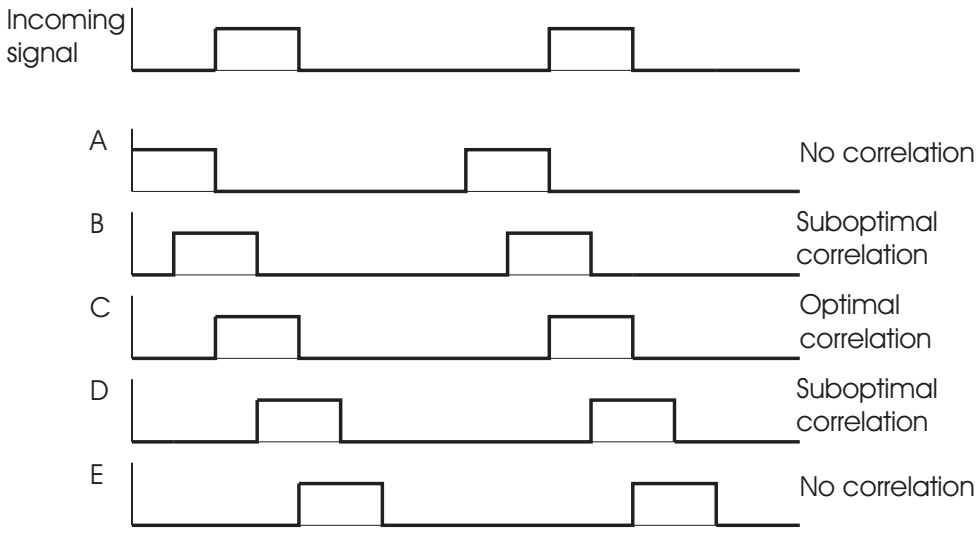

Figure 3.2: Sliding correlation principle

The sliding correlation takes place in discrete increments of half a chip time. This is illustrated in Figure 3.2. The best correlation is achieved when the incoming signal is correlated with waveform C. However, the two waveforms might never be in perfect synchronization like the incoming signal and waveform C. The discrete increments might miss the correlation peak illustrated in Figure 3.2. The worst-case scenario is illustrated in Figure 3.3.

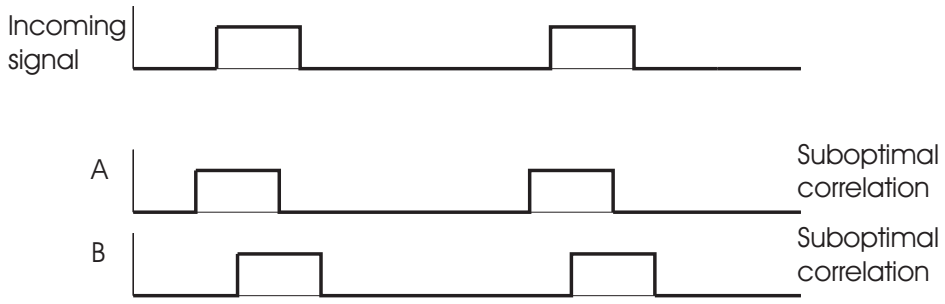

Figure 3.3: Sliding correlation misalignment

We can see that the update of waveform A to B in Figure 3.3 misses the correlation peak. In this worst-case scenario, where the offset is one-half the half-chip increment, $T_c/4$, this results in an (absolute) signal power loss of $L = 0.75^2$. This also means that detection (a treshold crossing) can occur on two positions. This results in an increased detection probability, given by

$$P'_{\mathrm{d}} = P_{\mathrm{d}} + (1 - P_{\mathrm{d}})P_{\mathrm{d}} \tag{3.3}$$

In (3.3), P_{d} represents the detection probability of the first position, while $(1-P_{\mathrm{d}})P_{\mathrm{d}}$ represents the probability that there is no detection at the first position, multiplied by the probability of detection. P'_{d} then represents the 'effective' detection probability, that is, when detection can occur on two positions.

The detection probability of the system is subject of Section 3.3.4, but we refer to it here since the sliding correlation mismatch affects it.

3.3.3 Modulation distortion

The third effect to be taken into account is that of attenuation of the bandpass filter of the information signal. The modulation of the information signal can cause some of the signal power to lie in the range where the bandpass filter attenuates. This is refered to as modulation distortion. The modulation distortion M_{s} is given as a factor between zero (infinite distortion) and one (no distortion). Indeed, the cutoff frequency of the bandpass filter should be chosen with regard to the pulse shape used. The modulation distortion is given by a formula very much like (3.2). It is essentially a similar effect.

$$M_{\mathrm{s}} = \frac{\int_{-\infty}^{\infty} S_{\mathrm{m}}(f) \mid H_2(\mathrm{j}2\pi f) \mid^2 \mathrm{d}f}{\int_{-\infty}^{\infty} S_{\mathrm{m}}(f)\mathrm{d}f} \tag{3.4}$$

In (3.4), $S_{\mathrm{m}}(f)$ is the power spectral density of the modulating signal. $H_2(\mathrm{j}2\pi f)$ represents the transfer function of the bandpass filter 2.

3.3.4 Integration block and treshold comparison

The instantaneous signal power into the integration block can be written as the product of the energy per bit, E_b, and the data rate R. Taking into account the various effects described in the previous sections, the SNR at the input of the integration block in Figure 3.1 can be expressed as

$$\gamma_\mathrm{DSSS} = \frac{E_\mathrm{b}}{N_0}\frac{RLM_\mathrm{s}}{M_\mathrm{n}N_0B_\mathrm{f}} \tag{3.5}$$

The effects of R, L, M_s, and M_n are such that γ_DSSS does not differ by more than an order of magnitude to E_b/N_0. The multiplication with the data rate R is offset by division by B_f, which is usually just a little larger than R. Equivalently, M_s and M_n should be the same order of magnitude if bandpassfilter 1 and 2 are both well-designed. As such, we can write

$$\frac{RLM_\mathrm{s}}{M_\mathrm{n}N_0B_\mathrm{f}} \approx L \approx \frac{1}{2} \tag{3.6}$$

and we can state that

$$\gamma_\mathrm{DSSS} \approx \frac{E_\mathrm{b}}{2N_0} \tag{3.7}$$

Considering all trade-offs would require too much detail here. For example, the choice of the modulation format influences the design of the second bandpass filter, but our interest is solely the synchronization procedure. We therefore simply assume that the system is well-configured and that the SNR γ_DSSS is half E_b/N_0, accounting for L. The integration block is characterized by the integration time ("dwell time") T_d. In the scenario used hereafter, a single information bit spans the entire PN code. Futhermore, the dwelltime T_d will be equal to one bit-time. The result of integration of the correlation during the dwelltime is compared to a treshold η in the treshold comparison block. When the result is higher than the treshold, the locally generated PN code is considered to be in phase with the received signal. When it is not, the systems continues its sliding correlation mechanism, searching for the correct code alignment. The choice of the treshold influences the performance of the system. If it is set too high, a correct code alignment may not be recognized as such if the received signal is too noisy. If the treshold is set too low, accumulated energy of noise may cross the treshold although no alignment is present.

During the correlation of the locally generated PN code with the incoming signal, several things may happen. When the codes are in alignment, this may or may not be detected (due to noise). The detection probability P_d is the probability that the correct chip alignment is indeed identified as such. When the codes are not in alignment, noise may cause the treshold to be crossed anyway. The false alarm probability P_fa is the probability that an incorrect chip alignment is identified as the correct chip alignment. One can also define meaningful probabilities such as $1 - P_\mathrm{d}$, which represents the

probability that correct code alignment is not detected when it is in fact present. $1 - P_{\text{fa}}$ would then represent the probability that the incorrect chip alignment is indeed identified as being incorrect.

When a high detection probability is required, the treshold should be low, or else the correlation produced may not cross the treshold. This consequently also results in a higher false alarm probability, since a lower treshold is also more easily crossed by random fluctuations not produced by the PN code correlation with the incoming signal. Vice versa, a higher treshold means a lower detection probability, and a lower false alarm probability. The detection probability for the serial-search DSSS acquisition system was found to be [15]

$$P_{\text{d}} = 1 - \int_0^{\eta} e^{-Z-\gamma_{\text{DSSS}}} I_0(2\sqrt{\gamma_{\text{DSSS}} Z})\,\text{d}Z. \tag{3.8}$$

In (3.8), Z, the (normalized) integration variable, represents the correlation produced by multiplying the signals, that is, the signal output from the square-law envelope detector. η represents the normalized threshold. This formula, for the case when the dwelltime T_{d} equals one bit period, was derived by the authors of [15], after deriving the probability density function of Z. Integrating over the probability distribution function (PDF) of Z until η then yields the total probability of no detection. I_0 is the modified Bessel function, and γ_{DSSS} the SNR.

The false alarm probability under the same circumstances was found to be [15]

$$P_{\text{fa}} = e^{-\eta} \tag{3.9}$$

Evaluation of (3.8) shows that the detection probability increases with larger γ_{DSSS}, and that it decreases when η is higher. Higher η, however, also decreases the false alarm probability.

We will already reveal that since the threshold η can be written in terms of P_{fa} (its natural logarithm), we can express (3.8) in terms of P_{fa}, eliminating the threshold η.

$$P_{\text{d}} = 1 - \int_0^{-\ln(P_{\text{fa}})} e^{-Z-\gamma_{\text{DSSS}}} I_0(2\sqrt{\gamma_{\text{DSSS}} Z})\,\text{d}Z. \tag{3.10}$$

We have plotted (3.10) in Figure 3.4.

As can be seen from Figure 3.4, the higher the SNR γ_{DSSS} becomes, the easier it is to lower the false alarm probability without decreasing the detection probability too much.

3.3.5 Acquisition time

The time for coarse code acquisition T_{acq} is given by $N\,T_{\text{d}}$, where N is the number of search steps required, and T_{d} is the time required for the DSSS synchronization

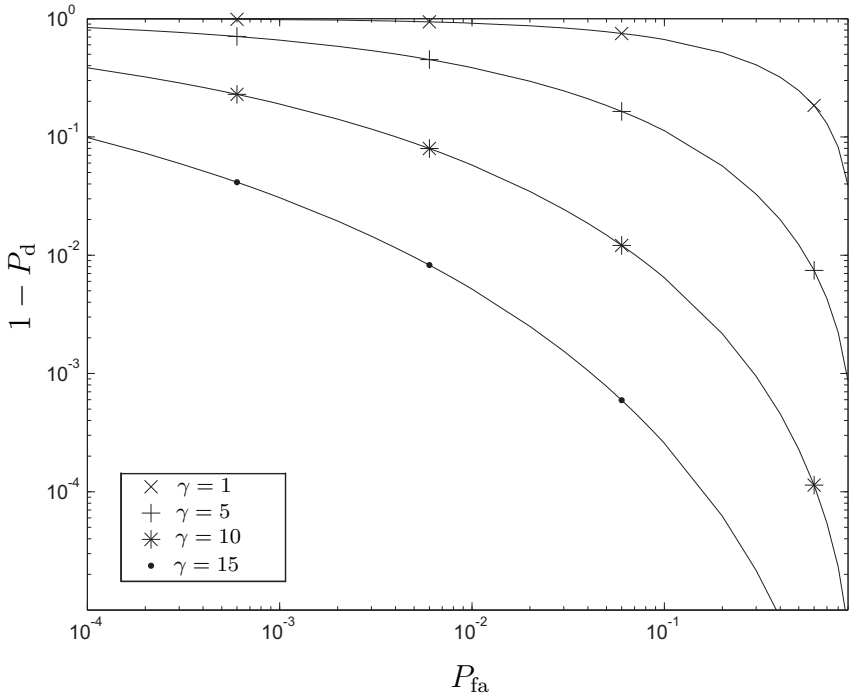

Figure 3.4: The detection probability as a function of the false alarm probability for various SNRs γ_{DSSS}

mechanism to take one such step, that is, evaluate one code alignment position (measure its correlation). T_{d} was assumed to be one bit-period in the previous section. If N is random, the acquisition time is random. In a real scenario, N is random because the starting position of the incoming code is unknown. The system might find the correct alignment after a few increments, or it may cycle through the entire code. It is even possible that multiple cycles through all code positions are required, since the true code alignment might not be detected the first time. It is also possible that the threshold is crossed although the true alignment has not been found. This is expressed in P_{fa}. The synchronization mechanism then considers its job complete and the tracking mechanism will take over. After a while, the tracking mechanism will recognize the error. The time required for the tracking mechanism to recognize a false lock and signal the synchronization system to start over is expressed by an extra number of required steps for the synchronization procedure, K. K depends on system parameters, that is, it is determined by the design parameters of the system. We do not consider the factors that determine K in a real system, but model it as a constant. Example values for K are 4 [15] or 5 [13]. After the system has returned to coarse code acquisition, it may find the true alignment quickly, cycle through the entire code several times (in case of a low detection probability), or generate a new false alarm.

The authors of [15] have modeled the acquisition system as a state machine. The

various scenarios described above correspond to various paths taken by the state machine. The transition probability between states are defined in terms of P_{d} or P_{fa}, and the number of steps taken (multiplied by the dwell time) represents the acquisition time. For large q, that is, long PN code sequences, they found the mean acquisition time to approximate

$$\overline{T}_{\mathrm{acq}} = \frac{(2 - P'_{\mathrm{d}})(1 + KP_{\mathrm{fa}})}{2P'_{\mathrm{d}}} q\, T_{\mathrm{d}} \tag{3.11}$$

The variance and probability distribution were also found, but those are omitted here since they are not relevant to the energy consumption of the receiver in the long term, when many acquisition instances occur.

In (3.11), P'_{d} is the modified detection probability described in Section 3.3.2. When T_{d} is assumed to be one bit-period, we can write the bit-normalized mean acquisition time, that is, the average required number of bit-periods for acquisition, as

$$\overline{T}_{\mathrm{acq}} R = \frac{(2 - P'_{\mathrm{d}})(1 + KP_{\mathrm{fa}})}{2P'_{\mathrm{d}}} q \tag{3.12}$$

If P'_{d} is close to 1, P_{fa} much smaller than 1, and K 4 or 5, as in [13, 15], we can see than the bit-normalized mean acquisition time is close to one-half the size of the search space q, a somewhat intuitive result. From Figure 3.4, we can see this is realistic when γ_{DSSS} is higher than 10. In that case, the size of the search space q (the length of the spreading code) and the chip update precision (usually $T_{\mathrm{c}}/2$) have a large influence on the mean acquisition time. If a single bit spans the entire PN code, it follows that the codelength is proportional to the spreading factor S. This means that for larger spreading factors, long codes are needed, which affect acquisition time proportionally as well.

3.4 Tracking in direct sequence spread spectrum

Once the locally generated code is put into coarse synchronization with the incoming signal, the tracking operation takes control. Tracking ensures that the locally generated code stays in synchronization, and decreases the initial error (at most one half chip time T_{c}) to within a small fraction of a chip time. The delay-locked loop (DLL), which was shown to have similar performance to a tau-dither loop (TDL), is illustrated in the schematic given in Figure 3.5.

The feedback loop shown in Figure 3.5 illustrates the clock driving the PN code generator, which is continuously updated by the error signal, produced by mixing a delayed and undelayed chip sequence with the incoming signal.

The DLL mixes a delayed and undelayed version of the locally generated chip sequence with the incoming signal. Both products from the mixers are bandpass filtered and square-envelope detected. If the delayed code is closest to the incoming code

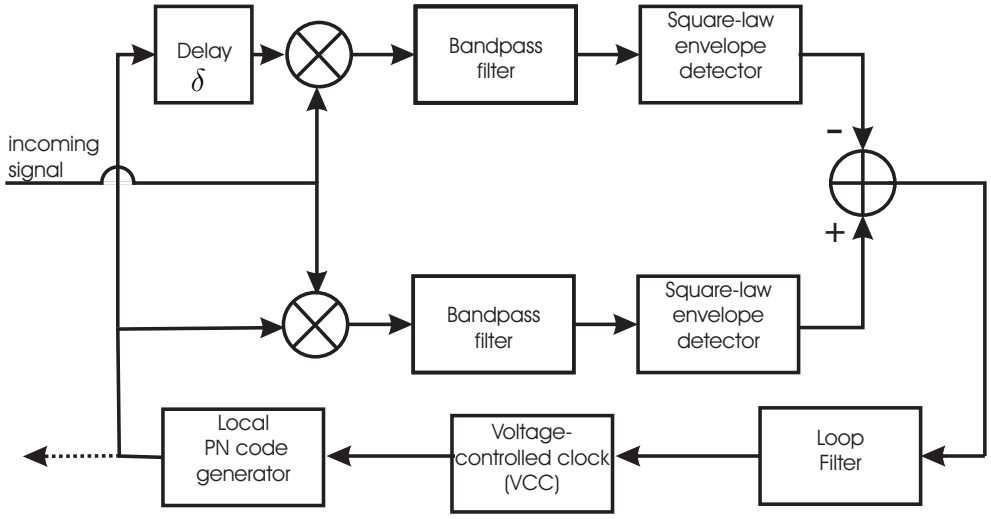

Figure 3.5: Schematic of a delay-locked loop

phase, it generates a larger signal than the undelayed product, causing a negative error signal after the adder, which causes the voltage-controlled clock (VCC) to decrease its frequency, and the error to become smaller.

This feedback loop causes the timing error of the locally generated code with respect to the incoming signal to decrease quickly, after which it keeps the local code in lock with the incoming signal within a time difference of $\delta/2$.

The delay δ is given by T_c/N, with N an integer larger than 1. Common values for N are 2, 4 and 8 [15]. N thus determines how tight the DLL will lock to the code. Alternatively, the delay δ is called the 'correlator spacing', since it determines the time difference between the two correlators (the upper and lower branch). The first observation to make is that the acquisition behaviour of the DLL depends on the multiplier gain. The larger the gain of the multiplier, the larger the control signal becomes, given a certain time offset between the incoming and locally generated signal. Thus, if we make the gain for both multipliers large, acquisition will occur sooner. On the other hand, very large multiplier gains imply a larger sensitivity to noise, which can be amplified and negatively influence tracking and the signal quality. The loop noise bandwidth B_L is related to the multiplier gain. It is the equivalent noise bandwidth of the loop. The loop bandwidth B_L therefore is a measure of the response speed of the DLL to an initial error.

Simulation of the transient reponse of the tracking operation in a noise-free environment obtained in [15] yielded the results results listed in Table 3.1.

Increasing the loop noise bandwidth decreases the response time of the loop, but consequently more noise enters the loop so tracking performance is expected to suffer, causing a penalty to the SNR after tracking. It should be noted that both in [15] and [27], B_l is called the loop noise bandwidth and loop bandwidth interchangedly.

B_l (Hz)	100	200	300	400
$N = 2$	0.0244	0.0122	0.0081	0.0061
$N = 4$	0.0292	0.0146	0.0097	0.0073

Table 3.1: Tracking lock time in seconds

There appears to be no difference in meaning, although Gardner [28] notes that loop bandwidth usually refers to the loop gain B_L for the cases where noise is a significant disturbance.

The results from Table 3.1 serve as an indication of the time required for the DLL to reduce the timing error to within a fraction (one-tenth) of a chip-time T_c. These results are only valid for the case when B_l is much smaller than the data rate R, and when they have a fixed relationship, for example $R/B_l = 100$. The loop noise bandwidth thus cannot be made too high (its ratio to the data rate R should be much larger than one). If one then increases the loop noise bandwidth to favorably affect the time for the tracking loop to enter lock, one has to increase R proportionally to keep the ratio of the loop noise bandwidth to the data rate constant. Since B_l is proportional to the time for the DLL to enter lock, one can then see that the the time to enter lock is independent of the data rate, that is, it always constitutes an equal number of bit periods. The number of bit periods required is indeed quite high, at a ratio of $R/B_l = 100$, the tracking operation requires 244 bit periods to enter lock. This is partially due to the fact that the results from Table 3.1 were acquired for relatively low SNRs, being below 1.

3.5 Summary

In this chapter, we have seen how DSSS requires coarse code acquisition and tracking. Coarse code acquisition is influenced by two distortion effects caused by mismatched spectra and the discrete increments in this system might not be optimal as well. These effects influence the SNR at the input of the integration block. Assuming that one bit spans the entire PN code, and assuming that the correlation produced by mixing the local and incoming signal is integrated over one bit period, expressions were found in [15] that relate the SNR to the probability that the threshold comparison block might generate a false alarm or miss the true alignment. These detection and false alarm probabilities were used by the authors of [15] to approximate the mean acquisition time of the system using a state machine. The bit-normalized mean acquisition time was found to approximate one-half the codelength for high SNRs (> 10). Below this value, the exact SNR plays an important role in determining the false alarm and detection probability.

The time for the tracking operation to enter lock was found to be independent of the data rate R. A limitation of the theory described in [15] was that it was only valid for low SNRs (< 1).

In Chapter 5, we will evaluate whether the tracking operation constitutes a significant portion of the total synchronization time or not, when we will look at specific scenarios specifying data rates, code lengths, and spreading factors. It is interesting to note that the tracking operation is not dependent on the codelength or spreading factor, while the acquisition procedure is.

In the next chapter, we will take a look at the synchronization procedure for FODMA, see how it is influenced by these factors, and prepare for the numerical comparison of Chapter 5.

Chapter 4

Synchronization in noise-based frequency offset modulation

4.1 Introduction

In the previous chapter, the synchronization time for acquisition and tracking of the DSSS code was found. For the DSSS system, the synchronization time depends on the code length (and thus the spreading factor) The SNR γ_{DSSS} only has an influence when it is small (< 10). The goal of our study is to compare the required synchronization time of the new system (FODMA) to the DSSS system. To facilitate this comparison, we look for a measure of synchronization time for FODMA

In the literature, the mean synchronization time for a phase-locked loop, as used in FODMA, was not found. Rather, the probability to achieve synchronization within a certain time period was found. Although these concepts are both useful, they are not equal. However, if both systems are actually implemented and used over a longer period of time, a DSSS receiver with a mean acquisition time of one second would have roughly the same energy consumption as a FODMA receiver with a high probability (0.9-0.99) of acquiring synchronization within one second.

In this chapter, we will first discuss some of the available literature considering synchronization in other TR systems in Section 4.2. Subsequently, we will discuss the architecture of FODMA in Section 4.3. Subsequently, the role of the PLL in this architecture is explained in Section 4.4, including several modeling assumptions and concepts in the relevant subsections. In Section 4.5 the PLL operation in a FODMA system is analyzed, and a connection is made with the work of Shang [19]. Subsection 4.5.1 will detail some additional modifications to the theory in that section to account for the effects of modulation on the PLL. Finally, Section 4.5.2 will discuss some bounds to the optimization of the PLL. FODMA also requires symbol timing estimation, but this part is included in Chapter 5, in Section 5.2.

4.2 Synchronization in transmit-reference systems

Several publications on synchronization in transmit-reference systems are already available. They all consider delay-hopped transmitted-reference (DHTR) systems, as proposed by Hoctor and Tomlinson [17]. Hoctor and Tomlinson are largely responsible for the revived interest in transmit-reference systems, which have been known for a long time [15,16]. DHTR, however, is only a particular flavor of a transmit-reference system. It achieves spreading by transmitting wideband pulses, encoding information through the difference in polarity of consecutive pulses. Varying the time between consecutive pulses allows for multiple access, although multiple access can also be implemented in the MAC layer.

In DHTR, the first pulse represents the reference pulse, and the polarity of the second pulse represents the information bit. The transmission medium is considered constant over the time between pulses, enabling the first pulse to serve as a reference to the second one. DHTR also allows for more than one reference pulse, or multiple information-bearing pulses [29] to achieve better signal quality. Casu and Durisi [29] have analyzed synchronization in DHTR systems. Their work is mainly concerned with analyzing the resources required for realtime or non-realtime (buffering) receivers, all of which operate in the digital domain after sampling the received signal. Another paper analyzing synchronization in DHTR is the one by Feng and Namgoong [13] published in 2005, who propose to split the received pulse into different frequency channels, which (in certain scenarios) results in faster acquisition because of a reduced search space for the start of the received pulse. Yet another paper analyzing DHTR synchronization was published by Djapic et al. [30] in 2006. It analyzes the computational complexity of the search for the starting instant of the received pulse. These three papers all assume Hoctor and Tomlinson's DHTR model. Although Hoctor and Tomlinson have also tested their setup with noise-like carriers instead of pulses [31], and found equal performance, synchronization of the system was not part of their research. The mentioned papers all assume that the signal is sampled after the frontend. Sampling can be an expensive operation, especially at high bandwidths (before despreading). For ultra-low power wireless sensor networks, it would be more attractive to pursue a transmit-reference scheme that does not use sampling after the front-end. Despreading would then be done in the analog domain. Goeckel and Zhang [20] have proposed a TR receiver that performs despreading in the analog domain. Additionally, Goeckel's receiver utilizes a frequency offset instead of a time offset, which results in a simpler receiver, as explained in Chapter 1, Section 1.3. Their proposed system performs synchronization in the digital domain (after despreading). Goeckel and Zhang show that synchronization is possible using this method, but do not perform any quantitative analysis of the synchronization time. A scheme similar to Goeckels' (FODMA), that

also uses a frequency offset instead of a time delay is proposed by Haartsen [18], based on the work of Shang [19]. It further simplifies the transmitter because it does not utilize pulses, but noise as carrier. Moreover, the requirement for digital resources (memory elements, processor) is further minimized using this architecture, although whether this is justified remains to be seen, since digital synchronization might still be faster, requiring less energy. This thesis will pursue the case of analog synchronization (without sampling), since it might still prove to be an attractive and simple option, and since this has not been subject of research as of 2010, to the best of our knowledge.

The next section will detail the operation of the noise-based frequency offset TR modulation scheme, and prepare for the analysis of the synchronization operation.

4.3 Noise-based frequency offset modulation

FODMA, proposed by Haartsen and Shang [18, 19], utilizes a noise source as the carrier. Instead of distinguishing the modulated and non-modulated (reference) signal in time (as in DHTR), they mix the modulated carrier with a low-frequency oscillation to seperate it from the unmodulated (reference) carrier in the frequency domain. The layout of a noise-based frequency offset TR sender and receiver is illustrated in Figure 4.1.

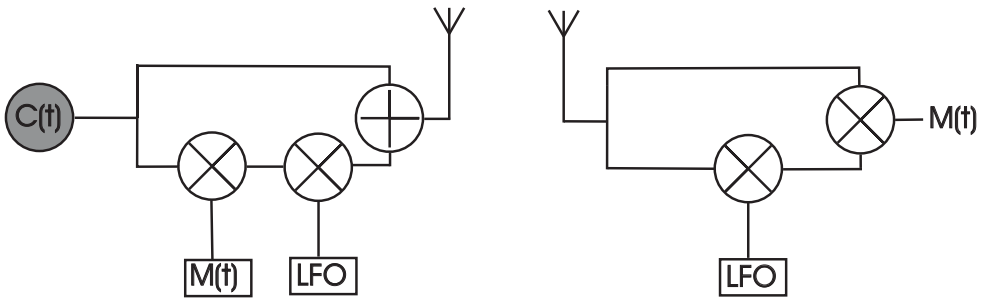

Figure 4.1: FODMA sender and receiver

In Figure 4.1, $C(t)$ represents the carrier, which is noise in our case. $M(t)$ represents the information signal. In the illustration, the reference signal travels through the upper branch in the sender, and the carrier travelling through the lower branch is modulated by the information signal and mixed with the signal from the LFO (causing the 'offset'). In the receiver, the LFO is tuned to the same frequency as the LFO in the sender. Multiplication of the reference signal with the offset signal causes despreading, leaving the information signal $M(t)$ at baseband.

In the receiver, one of the multipliers can be replaced by a squarer. The change in receiver architecture is illustrated in Figure 4.2. Instead of splitting the paths in the receiver and applying a frequency offset to one of the paths and then performing

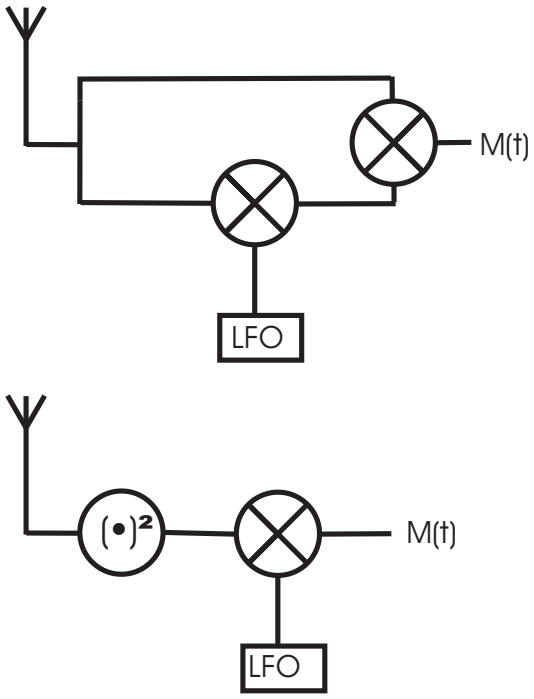

Figure 4.2: noise-based frequency offset receiver architectures

multiplication of the offset and non-offset paths, the received signal is first squared, which despreads the signal to a band around the LFO frequency. Subsequently, it is multiplied with the local LFO signal, which brings the information signal to baseband. This architecture was proposed by Goeckel [20]. Mahrof [22] has done an analysis of a complementary metal oxide semiconductor (CMOS) implementation of the FODMA receiver with a squarer.

For correct operation, the LFO needs to be phase-locked to the oscillation in the received signal. Several approaches are possible for synchronizing the carrier phase. For the case of an unknown carrier phase in the received signal, Shang [19] proposes three different receiver architectures. The first essentially uses a phase-locked loop to achieve phase-lock, and we will study it in detail in Section 4.4 and 4.5. The second and third architecture mix an inphase and quadrature sine with the incoming signal. They differ through the use of on/off keying or differential coding. No PLL is required for these architectures. Because these are not coherent receivers (which demodulate with respect to the carrier phase), they suffer a (small) performance penalty. These architectures also make symbol timing estimation more complicated. We will analyze the first receiver architecture given by Shang, the one utilizing a PLL, since this coherent receiver has the best BER performance in theory, and because it requires less digital resources. For the architecture utilizing a PLL, literature is available [27] which analyzes the time for such a circuit to attain phase synchronization, which is central to our question of whether a FODMA receiver is an interesting option for ultra-low

power wireless sensor networks requiring short synchronization times.

Another option is to perform LFO phase recovery and symbol timing recovery simultaneously. Recovering carrier phase and symbol timing at once is proposed by Goeckel [20] in their analysis of the frequency-shifted reference (FSR) TR receiver, which operates essentially the same as the FODMA receiver, differing only in the use of pulses instead of noise as carrier. Goeckel proposes to set the frequency offset $\omega_0 = 1/T_s$, where T_s is the symbol period. In the transmitter, the LFO signal phase is synchronized to $M(t)$, the information signal. Goeckel then shows that in the receiver, there exists a quantity at the output of the integrator following the coherent mixer which, when minimalized, yields the proper symbol and carrier phase timing. However, in [20], this proposed synchronization mechanism only serves for demonstrating that synchronization can be obtained, and there is no analysis of the time required for the synchronization to complete, which is the current subject of interest.

Another option is to despread the RF signal using the squarer (in the analog domain), and perform carrier estimation digitally, through sampling and algorithms. Because the sampling happens not at the front-end, but at the LFO frequency, the sampling rate is considerably lower, and requires considerably less power. Digital phase-locked loops were analyzed in [32]. Phase acquisition was found to be complete within an impressive 11 cycles of the incoming signal frequency. However, this only holds for a noiseless signal. Furthermore, faster acquisition is attained through using higher frequencies, which once more requires sampling on higher frequencies. Although the digital PLL is promising for the application under consideration (fast acquisition), we chose to focus on analog PLLs, because the analog solution might also provide fast acquisition, and does not require digital resources such as memory elements, making the receiver simpler and more energy-efficient. When the TR receiver first despreads the signal through squaring, and subsequently phase-locks by means of a PLL, it essentially operates as a coherent receiver. In a coherent receiver, the received signal is multiplied with a local oscillator whose phase is synchronized to that of the received signal. The PLL is the circuit that locks the phase (and frequency) of a local oscillator to that of the incoming signal. In the next section, we will detail the operation of a phase-locked loop, and analyze the time required for it to enter phase-lock, which is part of the synchronization time for a FODMA receiver. The second part of synchronization, symbol-timing estimation, will be discussed in Chapter 5, Section 5.2 as it is part of the synchronization time for both systems.

4.4 Phase-locked loop

Phase-locked loops are an essential part of telecommunication systems, and are covered extensively in the literature. Notable publications include those of Gardner [28],

Stensby [33], and Meyr [27]. The book by Meyr also includes several chapters on acquisition for a PLL, and is the book from which most theory is drawn here, but first we will discuss PLL basics. A PLL operates by multiplying the received signal with a locally generated signal, producing an error signal that drives the local oscillator to advance or delay its phase, producing a feedback mechanism, much like the DLL discussed in Section 3.4. The error signal should decrease through feedback, and after a certain period, the incoming and locally generated signals are in phase. The most simple model of a PLL is given in Figure 4.3.

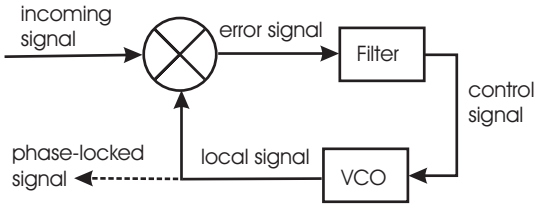

Figure 4.3: Basic PLL model

The basic components of a PLL are the phase comparator (implemented by the multiplier), the filter, and the VCO. The multiplication of the local and incoming signal produces a signal at the sum and difference of the incoming and local frequency. The term at the higher frequency is filtered out by the low-pass filter. The term at the difference of both frequencies is passed by the low-pass filter and serves as the control signal for the VCO. The VCO then adjusts its frequency according to the control signal. This feedback continues until the VCO oscillates in-phase at the same frequency as the incoming signal, which causes the error signal to become small and keep the VCO steady at the frequency and phase of the incoming signal.

4.4.1 Signals in the loop

The incoming signal is characterized by its amplitude, frequency and phase. It can be written as $\sqrt{2}\,A\,\sin(\omega_0\,t+\theta)$. The local signal has the same characteristics, and we write it as $\sqrt{2}\,K_1\,\cos(\omega_0\,t+\hat{\theta})$. Writing the incoming signal as a sine and the locally generated signal as a cosine, the PLL will lock to the incoming signal with a $\pi/2$ offset. In the feedback loop, the multiplier is characterized by a gain, K_m. When analyzing the behaviour of a PLL, the amount of noise at the input clearly has a large influence on the acquisition behaviour of the PLL. However, expressing the quality of the signal in the loop is difficult, since there is no signal in the loop for which we can define an SNR, because the internal signal of the PLL is ideally zero. The goal of the PLL is to bring the phase error $\phi=\theta-\hat{\theta}$ between the incoming and locally generated signal close to zero. Therefore, the SNR in the loop ρ is defined to be the inverse of the variance

of the phase-error ϕ [27,28,33]. In terms of the incoming signal, ρ was found to be [27]

$$\rho = \frac{A^2}{N_0 B_\mathrm{l}} = SNR_\mathrm{i} \frac{B_\mathrm{i}}{B_\mathrm{l}} \tag{4.1}$$

where N_0 is the noise spectral density of the incoming signal, B_i is the bandwidth of the information signal, and SNR_i is the input SNR. B_l is a measure for the bandwidth of the PLL. B_l is that bandwidth which we can multiply with N_0 to get the total amount of noise power passed by the loop. The definition of B_l is subject of the following subsection.

4.4.2 Loop noise bandwidth

The loop noise bandwidth B_l is the equivalent noise bandwidth for the loop. The equivalent noise bandwidth is defined as that bandwidth over which a perfect rectangular filter passes as much power as the actual circuit, given white noise at the input. The amount of power (either noise or signal power) passed by the actual system (the PLL) is determined by the transfer function of the loop. The loop can be cut at an arbitrary point to define an input and output for the loop's transfer function.

For a simple PLL, the transfer function, and thus the loop noise bandwidth B_l, is a function of the multiplier gain and VCO sensitivity. The multiplier gain is given by K_m. The VCO change in frequency given an applied voltage, K_o, is expressed as radian per second per volt (or Hertz per volt, $K_\mathrm{o}/2\pi$). The loop noise bandwidth, or equivalent noise bandwidth of the system, is then known to be [27] $K_\mathrm{m}\,K_\mathrm{o}/4$ Hertz for the case of a simple first-order PLL without a filter.

The expression for B_l here is for a first order loop, that is, a PLL with only one filter with a single pole that only filters out the high-frequency term from the multiplier. More advanced PLL designs include extra filters (implemented as integrators) that establish additional functionality, such as a certain damping factor (of the oscillation around the stable tracking frequency). However, the first-order loop is the only type that allows mathematical approximation of the acquisition time of the PLL [27, 34], which was also found to be a good approximation of a second-order loop, the most common type of PLL.

4.5 PLL operation in FODMA

Now that we have defined the equivalent noise bandwidth B_l for the loop, we continue with analyzing the signal, the low-frequency oscillation, upon which the PLL has to lock. We will consider the receiver architecture with the squarer, and build upon the analysis of Shang. In Figure 4.4, the receiver with squarer is annotated with the relevant quantities.

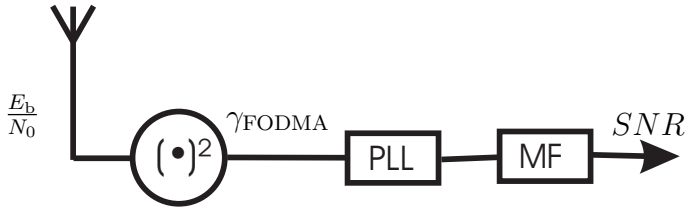

Figure 4.4: Noise-based frequency offset TR receiver

In Figure 4.4, the received signal at the front-end is characterized by the SNR per bit, E_b/N_0. Shang found that the SNR after the matched filter (MF), at the input of the decision block, can be expressed in terms of E_b/N_0, the spreading factor S, and the amplification factor c of the unmodulated reference signal in the transmitter. Shang found that there exist optimal values for both S and c which maximize the SNR at the input of the decision block. The amplification factor c of the unmodulated reference carrier was found to be optimal near $c = 1/\sqrt{2}$. The spreading factor S represents the ratio between the bandwidth of the noise carrier, B_c, and the information signal data rate, R. Larger spreading factors are associated with increased narrowband interference immunity, along with the other advantages discussed in Section 1.2. The SNR after despreading as a function of the spreading factor S and the SNR per bit E_b/N_0 for $c = 1/\sqrt{2}$ was found to be

$$SNR = \frac{16(E_b/N_0)^2}{(50/S)(E_b/N_0)^2 + 40(E_b/N_0) + 16S} \tag{4.2}$$

Shang found that relation (4.2) implies that an optimal value exists for the spreading factor S in terms of the received E_b/N_0 at the front-end. This relation was found to be

$$S_{\text{opt}} = \frac{5}{2\sqrt{2}}\frac{E_b}{N_0} \approx 1.77\frac{E_b}{N_0} \tag{4.3}$$

This somewhat inconvenient fact implies that the system designer should have good knowledge about the range of E_b/N_0, so that a near-optimal value for S can be chosen. Vice versa, the system designer cannot simply change the spreading factor without inducing a performance penalty on the system. Shang found that when both c and S are optimized, the maximum SNR at the decision device (after the matched filter) can be expressed in terms of E_b/N_0 at the front-end, as in relation (4.4).

$$SNR_{\max} = \frac{2}{5(1+\sqrt{2})}\frac{E_b}{N_0} \approx 0.17\frac{E_b}{N_0} \tag{4.4}$$

In Chapter 5, a more detailed interpretation of relation (4.4), (4.3), and (4.2) is made, especially considering the comparison with the DSSS system.

(4.4) holds only when the local oscillator is in perfect lock with the incoming signal. Assuming this perfect lock, γ_{FODMA} at the input of the PLL should be equal to γ_{FODMA}

at the output. The SNR of the signal after the matched filter is known to be $0.17E_\mathrm{b}/N_0$. Since a matched filter yields an SNR of

$$SNR_\mathrm{mf} = 2\gamma_\mathrm{FODMA} \tag{4.5}$$

given additive white noise, we can write γ_FODMA at the input to the PLL as a function of the front-end input,

$$\gamma_\mathrm{FODMA} \approx 0.08E_\mathrm{b}/N_0. \tag{4.6}$$

This shows that FODMA suffers a 11 dB penalty through despreading, and more if E_b/N_0 is not optimal with regard to S. Refering to Subsection 4.4.1, we can now rewrite (4.1), the SNR at the input of the PLL as

$$\rho = \gamma_\mathrm{FODMA}\frac{R}{B_\mathrm{l}} \tag{4.7}$$

The SNR in the PLL is the inverse of the phase-error variance in steady-state (see Subsection 4.4.1), but the quantity ρ is also used as a measure of signal quality during acquisition.

Now that we have derived ρ in terms of the incoming signal and the PLL characteristics, we are ready to take a look at the acquisition behaviour in terms of ρ, B_l and the time t. Let us look ahead and reveal that the acquisition time is linear with respect to the loop gain. Indeed, this holds to such an extent that Meyr introduces the gain-normalized time,

$$\tau = (K_\mathrm{m}\,K_\mathrm{o})\,t = 4\,B_\mathrm{l}t \tag{4.8}$$

when analyzing acquisition behaviour. This makes sense intuitively when one realizes that the $K_\mathrm{m}\,K_\mathrm{o}$ is a measure for how sensitive the PLL is to a phase error between incoming and locally generated signals.

From (4.7), it can be seen that a smaller loop noise bandwidth causes a better SNR in the loop, ρ, because more noise is rejected. However, B_l cannot be too small, because that would imply a small gain, and thus a slow response of the PLL. Furthermore, a small loop bandwidth detoriates the tracking performance of the PLL, as it becomes unable to track instabilities [35]. Meyr and Ascheid [27] have analyzed PLLs in detail. Much of their analysis concerns steady-state (tracking) behaviour of the PLL, where linearizing assumptions are made. The acquisition under noisy conditions, however, is an inherently non-linear phenomenon, which is difficult to evaluate. Meyr and Ascheid were only able to determine the probability that a first-order PLL reduces the phase error ϕ to within a given error ϕ_ϵ after a certain time τ. The results, for different values of ρ, a uniformly distributed initial phase error, and $\phi_\epsilon < \pi/6$, are given in Figure 4.5. According to [36], these results are also a good approximation of the behaviour of second-order PLLs, which are used in practice.

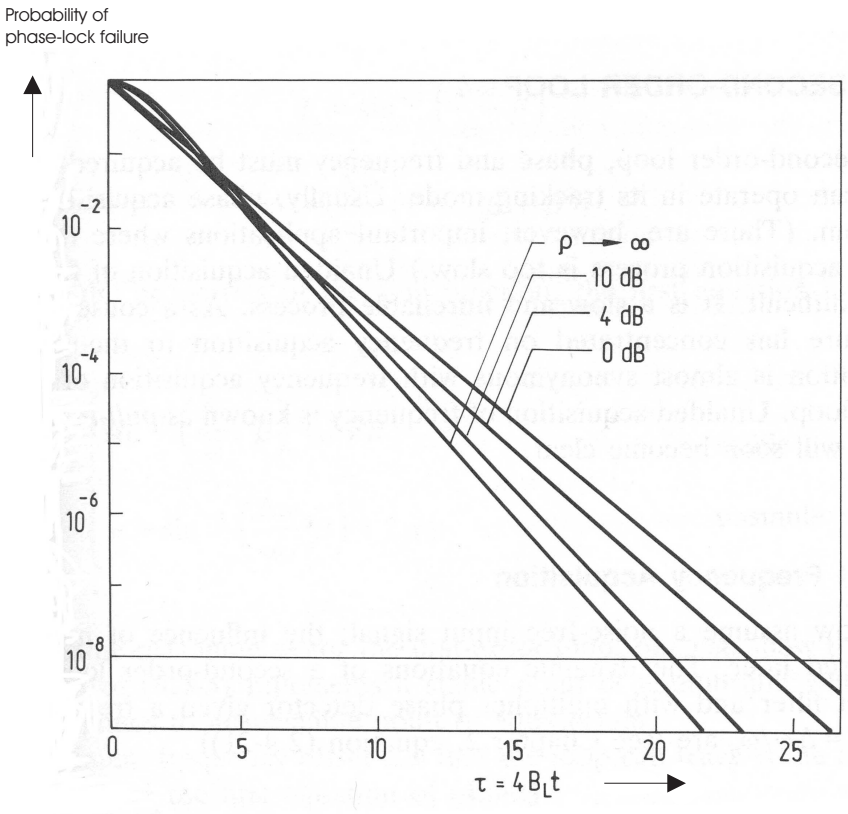

Figure 4.5: Probability of not achieving phase-lock within time τ. From 'Synchronization in Digital Communications' [27].

In Figure 4.5, the probability of not acquiring lock (y-axis) for a given gain-normalized time τ (x-axis) is plotted for different values of ρ. If one requires a 99 percent probability of locking (that is, a probability of not acquiring lock of 10^{-2}), the value of ρ (at least within the range 0–12 dB) is not of a very large influence. The plot for $\rho = \infty$ essentially represents the noise-free case. The question that arises from Figure 4.5 is how large the gain should be made, since making it arbitrarily large would make the acquisition time arbitrarily small. In practice, the gain (and thus the loop noise bandwidth B_l) cannot be made too large, because a small disturbance in the signal could then lead to a loss of phase-lock, as it becomes highly amplified. In Section 4.5.2, we will discuss some theory from [27] about limits to the loop noise bandwidth B_l. Also, the choice of ϕ_ϵ influences the synchronization period. If a small ϕ_ϵ is required, the PLL will require more time to bring the phase error to within that value. Choosing ϕ_ϵ larger would mean that synchronization is considered complete earlier. In [27], no clear method was given to investigate other values of ϕ_ϵ and we thus have to use the results for $\phi_\epsilon < \pi/6$.

The next Section will deal with the required modification of the PLL when the signal that is to be tracked is phase-modulated.

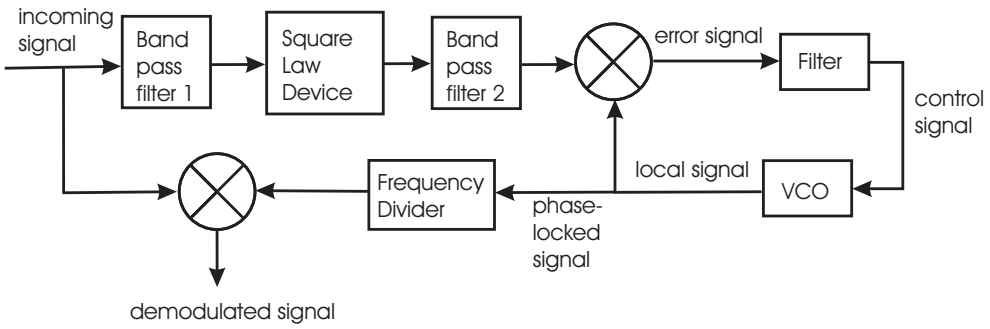

Figure 4.6: Square law device in combination with a PLL allows tracking of a biphase modulated carrier

4.5.1 Squaring loop modification

When the carrier signal to which the PLL has to lock is modulated, a modification to the PLL is required. Assuming BPSK modulation, the carrier can be tracked either by a squaring loop or costas loop [37]. When a squaring device is placed in front of the PLL, the PLL tracks double the frequency of the original signal, and this is called a squaring loop. A squaring device conceals the biphase modulation events. This ensures that a modulation-independent phase-locked signal can be generated in the receiver. Multiplying this phase-locked signal seperately with the incoming signal allows demodulation. The squaring loop was described in [37] as in Figure 4.6.

In Figure 4.6, the PLL locks to the squared signal, which is at twice the frequency of the original signal. This phase-locked signal is subsequently routed through a frequency divider such that it can be mixed with the original signal to perform demodulation. Our interest is now in finding the effect of the added squarer to the PLL. The square law device causes a penalty to the SNR ρ in the loop, called the squaring loss S_{L}. The squaring loss represents the noise that is added because the squarer multiplies two noisy signals. It was derived in [37], being

$$S_{\mathrm{L}} = \frac{1}{1 + 1/(\rho\beta)}. \tag{4.9}$$

Relation (4.9) holds for the case of an ideal (rectangular) bandpass filter before the square law device (bandpass filter 1 in Figure 4.6) with a bandwidth B_{i}. In (4.9), ρ is the original SNR in the loop, and β is given by.

$$\beta = \frac{B_{\mathrm{l}}}{B_{\mathrm{i}}} \tag{4.10}$$

(4.10) states that β is the ratio of the loop bandwidth to the pre-squarer filter bandwidth. The effective SNR in the loop ρ', incorporating the squaring loss, was then found to be [37]

$$\rho' = \rho S_{\mathrm{L}}/4 \tag{4.11}$$

Given (4.11) and (4.7) the SNR in the loop ρ' can be calculated for the case of a BPSK receiver. When β is large, that is, the data rate (and filter bandwidth B_i) is much larger than B_l, the squaring loss S_L will be close to 1 (very little losses), and ρ' will be close to $\rho/4$.

If β is not large, and the squaring loss cannot be neglected, we can write ρ', which incorporates the squaring loss, as a function of γ_FODMA and ρ using (4.11), (4.9) and (4.10):

$$\rho' = \frac{\rho S_\mathrm{L}}{4} = \frac{\rho/4}{1 + 1/(\rho\beta)} = \frac{\rho/4}{1 + \frac{R}{\rho B_\mathrm{l}}} \tag{4.12}$$

if we assume that $B_\mathrm{i} = R$. Using (4.7), we can then write

$$\rho' = \frac{\rho/4}{1 + 1/\gamma_\mathrm{FODMA}} = \frac{\gamma_\mathrm{FODMA}}{1 + \gamma_\mathrm{FODMA}} \frac{\rho}{4}. \tag{4.13}$$

This means that at a given ρ', we can state that

$$\rho = 4\rho' \frac{1 + \gamma_\mathrm{FODMA}}{\gamma_\mathrm{FODMA}}. \tag{4.14}$$

If we require that $\rho' = 10$, that is, we require a small phase error variation, we can see from (4.14) that for SNRs $\gamma_\mathrm{FODMA} > 1$, the difference between ρ and ρ' in (4.14) is about a factor 8 (SNRs near 1) or 4 (SNRs much higher than 1). At a 99 percent probability to achieve phase-lock, this difference between ρ and ρ' then has a neglible effect on the normalized acquisition time τ, which can be seen from Figure 4.5.

If we then write the time to acquire lock t from (4.8) as T_pll

$$T_\mathrm{pll} = \frac{\tau}{4B_\mathrm{l}} \tag{4.15}$$

and rewrite it with (4.7) into

$$T_\mathrm{pll} = \frac{\tau\rho}{4\gamma_\mathrm{FODMA} R} \tag{4.16}$$

we can combine (4.16) and (4.14) into

$$T_\mathrm{pll} = \frac{\tau\rho'}{R} \frac{1 + \gamma_\mathrm{FODMA}}{\gamma_\mathrm{FODMA}^2}. \tag{4.17}$$

which expresses the time for the PLL to enter phase lock as a function of ρ' (incorporating the squaring loss), τ, R and γ_FODMA.

Normalizing for the data rate R

$$T_\mathrm{pll} R = \rho'\tau \frac{1 + \gamma_\mathrm{FODMA}}{\gamma_\mathrm{FODMA}^2}. \tag{4.18}$$

We can already see from (4.18) that the acquisition time is roughly inversely proportional to the SNR γ_FODMA when the latter is much larger than one. We can also see that the spreading factor does not have any influence on the acquisition time for FODMA.

4.5.2 Loss of lock

In Section 4.5 it was found that the acquisition time is linear with the loop noise bandwidth B_l, and thus the PLL gain. This implies that if the acquisition time is to be short, the loop gain needs to be large. However, very large gains make the PLL sensitive to noise; a small deviation could be highly amplified and cause the PLL to lose phase-lock ('loss of lock'). In this section, we will see that there are limitations to how large B_l can be made without causing significant problems, in particular, a loss of phase-lock. Grave disturbances in the PLL, called cycle slips, occur when the disturbance is such that the PLL skips a cycle of the incoming frequency. A single cycle slip, when occuring quickly, should not lead to problems, since a temporary detoriation of signal quality does not have a large effect as the period of one cycle of the LFO is much shorter than one bit time (as was assumed by Shang). However, cycle slips necessarily pass a phase error of π, for which hang-up is known to occur [27]. Hang-up refers to the inability of a PLL to reduce the phase-error quickly if the phase-error is near π. This could mean that the signal-quality detoriation takes much more time; in the order of a bit-period. Statistical occurence of cycle slips as a function of B_l is subject of Chapter 6 of [27]. The mean time between 2 cycle slips $\mathrm{E}(T_s)$ was analyzed experimentally for a second-order PLL and was found to match analytical results for a first-order PLL. The experimental results are given in Figure 4.7.

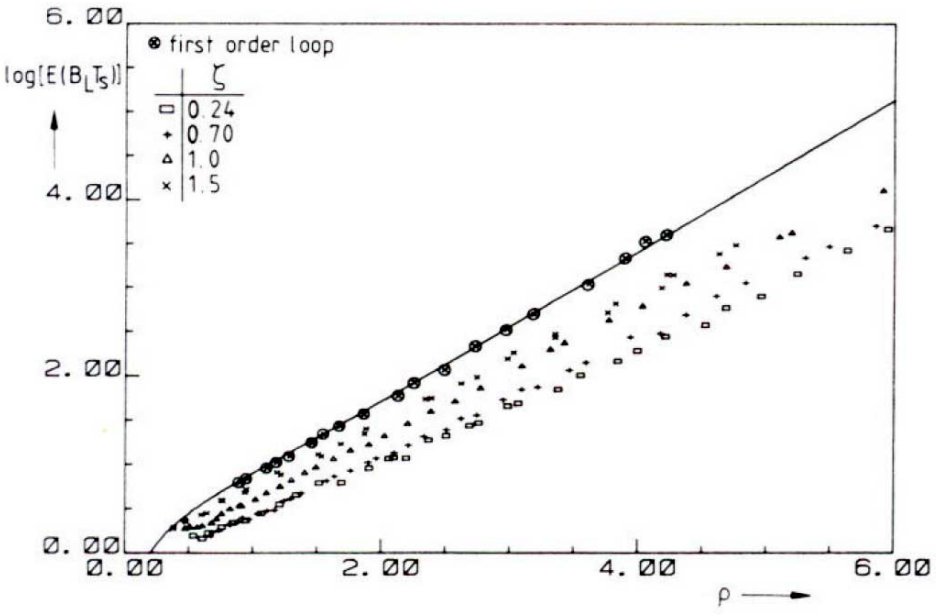

Figure 4.7: Mean time between cycle slips, from 'Synchronization in Digital Communications', [27]. Solid line represents analytical results for a first-order PLL.

In Figure 4.7, the gain-normalized mean time between two cycle slips $\mathrm{E}(B_l T_s)$ is plotted versus the loop SNR ρ. We can write $\mathrm{E}(B_l T_s)$ as $B_l \mathrm{E}(T_s)$. This signifies the

normalization with the loop gain, and that the expectation only refers to T_s.

In Figure 4.7, the solid line represents the analytical results for a first-order loop, obtained in [37,38]. The squares, pluses, triangles and crosses represent different values for the damping factor ζ. The damping factor ζ is a measure for the amount of overshoot the PLL, and it is a function of the loop filter characteristics. Reading the graph we can see that if ρ is 2, and we have a damping factor of 0.70, $\mathrm{E}(B_l T_s)$ is 1, so if B_l is 1000 hertz, we can expect a cycle slip every millisecond. If B_l is smaller, cycle slips are expected to occur less frequently. The results from this graph serve as an indication of the limitations of making B_l arbitrarily large. In Chapter 5, we will see that at the chosen values of ρ' (10), cycle slips are not directly a limiting factor for choosing B_l.

4.6 Summary

In this chapter, we have seen how to establish the required time for the PLL to enter phase-lock, which together with symbol-timing, determines the time required for synchronizing the FODMA receiver. The SNR γ_{FODMA} to the PLL in terms of the front-end input was found to be dependent on the spreading factor S. The SNR γ_{FODMA}, the bandwidth of the information signal B_i and the loop noise bandwidth B_l of the PLL determine the SNR in the loop ρ. The literature yielded the required time for the PLL to enter phase-lock for given values of ρ. Some necessary modifications were described to account for a biphase modulated signal, and some limits were found for B_l.

In Chapter 5, we will compare DSSS and FODMA receivers, setting the SNR after despreading γ equal, along with the data rate R. At fixed values of ρ, this allows us to determine B_l from (4.7). Given certain restrictions on B_l described in Section 4.5.2, this allows us to determine the time required by the PLL to enter phase-lock through Figure 4.5.

Chapter 5

Numerical synchronization comparison

In this chapter, the models and theory from Chapter 3 and Chapter 4 are used to make an approximation of the required synchronization time for the spread spectrum receivers subject of this thesis. Summarizing Chapter 3, DSSS requires coarse code acquisition and a transient period for the code tracking system. Symbol timing estimation for DSSS can be derived automatically if the symbol timing is locked to the PN code in the DSSS transmitter. The FODMA receiver requires the local oscillator to be brought into phase-lock using a PLL and symbol timing estimation. The aim of this chapter is to find out what the largest contributors to the synchronization time are, and provide insight into the numerical comparison itself, most importantly its assumptions and limiting factors.

The first section of this chapter will be concerned with the choice of evaluation criteria and limiting factors of the comparison. Section 5.2 will present some numerical results of the required time for symbol timing estimation. In Section 5.3, we will describe the calculation procedure used for comparing the bit-normalized synchronization periods for the DSSS and FODMA system and analyze to what extent the different synchronization steps add to the total synchronization time.

5.1 Evaluation criteria and limiting factors

The comparison of both systems is neccesarily limited to a certain level of detail. In this thesis, we strive to at least get an approximation of the different contributing factors. An exhaustive analysis of the entire synchronization procedure for either system would require that many factors affecting each other are all evaluated at the same time. For example, the synchronization time for a FODMA receiver depends on the PLL gain, which affects the tracking behaviour of the PLL, which in turn has a large influence on the quality of the demodulated signal, which will subsequently influence symbol timing estimation. It becomes neccesary, then, to treat these issues seperately, which serves not to be exhaustive, but to give an approximation of the contributing factors.

Three quantities will be fixed in the comparison: the SNR after despreading γ, the data rate R, and the spreading factor S; these will be equal for both systems. Setting the SNR after despreading equal is required because the systems will then have the same BER characteristic. If E_b/N_0 would be the fixed evaluation parameter, we would end up with different BERs for both systems. The penalty for FODMA caused by the despreading performance of that system (4.2) is then at the transmitter, which has to transmit with higher power.

We will compare both systems given a range of SNRs after despreading, for various spreading factors S. We stress that a non-optimal spreading factor induces a larger power penalty to the FODMA transmitter, which has to transmit at higher power. At a fixed data rate R, the spreading factor is a measure of the extent to which the system is robust to narrowband interference or multipath fading, discussed in Section 1.2. The effect of fading to FODMA was subject to a semi-analytical study [23]. The effect of narrowband interference to FODMA remains unknown. Fixing the data rate R and SNR after despreading also allows us to consider the required symbol timing estimation period similar for both systems. Furthermore, we would like to consider bit-time normalized synchronization times, that is, express the required synchronization time as a number of bit periods. This is desirable because the packet length and synchronization period then have the same units, such that the advantage of a shorter synchronization period can be expressed for different packet lenghts.

The synchronization precision of both systems is a second issue. How can the synchronization precision for both systems be compared? One system uses a PLL for tracking, the other one a DLL. Ideally, they would incur an equal penalty on the BER. Since we largely have to rely on data from the literature, where a certain error was chosen, it becomes clear that this also complicates the comparison. For the PLL, synchronization was considered complete when the phase-error was within $\pi/6$ [27]. For the DLL, synchronization was considered complete when the timing error was withing one-tenth of a chip-time T_c [15]. The SNR penalty for DSSS at a tracking error of $0.1T_\mathrm{c}$ can be deduced from the picture in Section 3.3.2. In fact, when the step size is T_c/N, that is, at a maximum tracking error of $T_\mathrm{c}/(2N)$, the penalty to the amplitude of the signal is

$$1 - \frac{1}{2N} \tag{5.1}$$

When $N = 2$, as in Section 3.3.2, there is a penalty of 0.75 to the signal amplitude, and 0.75^2 to the signal power, for a maximum error of $T_\mathrm{c}/4$. When the tracking error is $0.1T_\mathrm{c}$, $N = 5$, and the penalty to the signal power is then $0.9^2 = 0.81$, or -0.9 dB.

For the PLL, the allowed error $\phi_\epsilon = \pi/6$ incurs a power penalty of $\cos^2(\pi/6) = 0.75$, or -1.2 dB, to the signal.

Both power penalties are of the same order of magnitude, and both are only small compared to other effects, such as the requirement of at least 10 dB higher E_b/N_0 for

FODMA compared to DSSS.

Some of the more important limiting factors for DSSS include the requirement that $q >> 1$ for DSSS, so we only consider spreading factors $S > 10$, assuming that one bit spans the entire PN code. This fits in our current study, however, since its context is that of wideband communications, for which we consider a spreading factor of at least 10 reasonable. Another important assumption is that the dwelltime for DSSS is one bit period. In [15], the authors mainly consider much longer dwelltimes. These, however, are mainly meant to be able to acquire synchronization in very low (< -10 dB) SNR conditions. Such conditions do not apply to WSNs, and reflect the context of the research on which [15] was based, namely that of low-probability-of-intercept communications. Finally, the assumption that modulation distortion, noise spectral density reduction, and the ratio between the data rate and filter bandwidth only have a small effect (see (3.6)) for DSSS plays an important role in establishing the SNR after despreading γ_{DSSS}.

For FODMA, an important assumption is that no cycle slips will occur during transmission of the data.

5.2 Symbol timing estimation

Symbol timing involves the estimation of the optimal sampling instant at the receiver. The analog signal needs to be sampled, such that afterwards, all further processing can be done digitally. There exists an optimal instant during which to sample the analog signal such that the probability of a bit error is small. The receiver needs to estimate this optimal instant, which is unknown in the receiver because the instant of transmission is unknown, and the channel causes an unknown delay. Various books describe symbol timing estimation [25, 26, 39], and we will use the theory described in Chapter 7 of [26]. The estimation of the optimal instant is done using maximum-likelihood estimation. This is the strategy to use for an unknown, but non-random delay μ. In the estimation process, the (actual) delay μ needs to be estimated. $\hat{\mu}$ is the locally estimated value of μ. The precision of the estimation $\mu - \hat{\mu}$ needs to be within certain bounds if the BER is to be within certain limits. In [26] page 298, acceptable error rates were found to occur when the root mean square (RMS) value of $\hat{\mu} - \mu$ was within a factor 0.3 of the bit time T. The SNR penalty at an RMS tracking error of 0.3 was 1.1 dB, that is, the SNR per bit at the front-end needs to be 1.1 dB higher to achieve the same BER as without an estimation error. In this section, data-aided symbol timing estimation is considered. In data-aided estimation, the transmitted sequence is assumed known in the receiver, and this assumption is justified by the fact that the DSSS system and the FODMA system both already require some knowledge in advance (be it the spreading code or the particular frequency

offset). Data-aided symbol timing estimation is done in a feedback circuit where the incoming data sequence is correlated with the locally generated data sequence, which drives a voltage-controlled-clock. The performance of this feedback circuit depends on the number of symbols used in the estimation process, that is, the length of the sequence that is correlated. The more symbols J that can be used in the estimator, the more precise the symbol timing estimation becomes. The SNR γ of the incoming signal obviously has a large impact on the performance of the estimator. Finally, the shape of the signal is important to the symbol timing recovery effort. Assuming PAM signals are to be recovered, the near-optimal pulse shape $r(t)$ after matched filtering (in terms of timing recovery) was found to be [26]

$$r(t) = T\cos\left(\frac{\pi\alpha t}{T}\right)\operatorname{sinc}\left(\frac{t}{T}\right) \tag{5.2}$$

where T is the bit time, α is the excess-bandwidth factor, and

$$\operatorname{sinc}(x) = \frac{\sin(\pi x)}{\pi x} \tag{5.3}$$

The excess bandwidth factor denotes the amount of bandwidth occupied by the pulse beyond the Nyquist bandwidth $1/(2T)$. This is sometimes referred to as the roll-off factor as well. Without further going into detail regarding the exact performance of the estimator, the bit-time relative root mean square estimator error was found to be [26]

$$\frac{\sigma_\mu}{T} = \sqrt{\frac{3}{\pi^2 J(1+3\alpha^2)}\frac{1}{U} + \frac{9F}{\pi^4 J^2(1+3\alpha^2)}} \tag{5.4}$$

where U is the SNR, α is the roll-off factor, and F a function of J and the pulse shape $r(t)$ only, as

$$F = \sum_{j=0}^{J}\left[\sum_{m=-\infty}^{-1}(r'(mT-jT))^2 + \sum_{m=J}^{\infty}(r'(mT-jT))^2\right] \tag{5.5}$$

where the prime denotes the derivative with respect to time. The SNR U in (5.4) is defined in [26] as

$$U = \frac{A^2}{4N_0} \tag{5.6}$$

which means that $U = \gamma_{\text{FODMA}}/4$, which allows us to rewrite (5.4) as

$$\frac{\sigma_\mu}{T} = \sqrt{\frac{12}{\pi^2 J(1+3\alpha^2)\gamma_{\text{FODMA}}} + \frac{9F}{\pi^4 J^2(1+3\alpha^2)}} \tag{5.7}$$

Concluding now that the error is a function of the number of symbols used, the pulse shape and its bandwidth, and the SNR after despreading γ_{FODMA}, we plot the function (5.7) for some values of J and γ_{FODMA} in Figure 5.1.

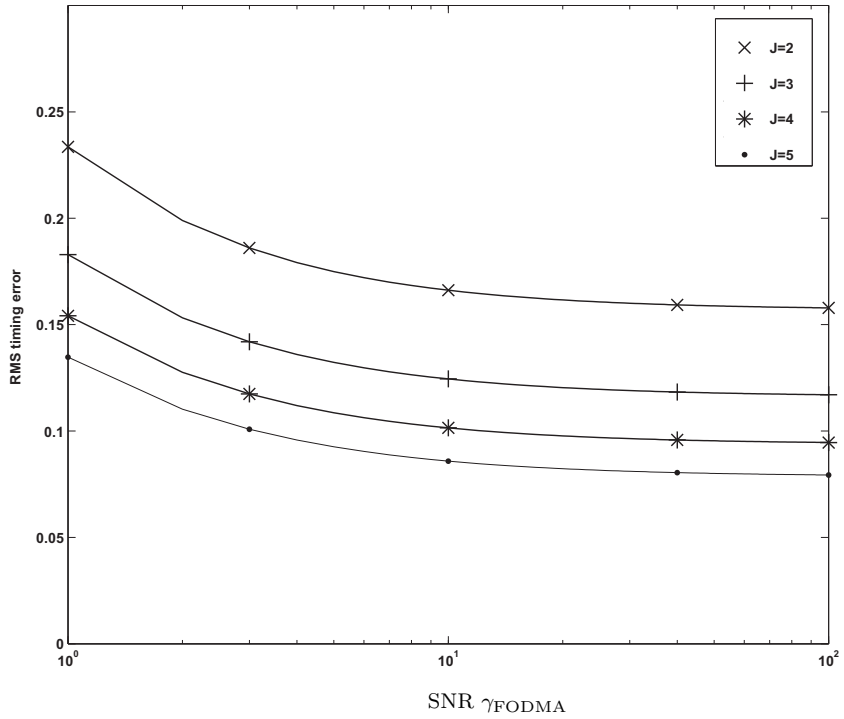

Figure 5.1: Symbol timing estimator error versus SNR γ_{FODMA} for sequence lengths $J = 2$, $J = 3$, $J = 4$, $J = 5$

As we can see in Figure 5.1, the estimator performance increases when more symbols are used, while the second term in (5.4) determines an error floor for high SNRs. We can see that two symbol periods should be enough for SNRs higher than 1, if we assume a timing error of 0.3 symbol periods to be acceptable.

5.3 Numerical comparison

In this section, we will compare the required synchronization period for FODMA and DSSS in terms of bit-periods. The relationship of the synchronization time with the SNR γ and the spreading factor S will be illustrated. As explained in Section 2.1 and Section 5.1, we will consider the SNR after despreading equal for both systems. For FODMA, this parameter is called γ_{FODMA}, and for DSSS, it is called γ_{DSSS}. They were defined in Section 4.5 and Section 3.3.4, respectively. Irrespective of how we arrive at the SNRs of both systems, we apply a penalty of 1/2 to γ_{DSSS} because of the chip-update misalignment described in Section 3.3.2, which is an effect after despreading.

For DSSS, we chose the dwelltime to be one bit time. (3.12) allows us to express express the bit-normalized acquisition time in terms of the false alarm probability P_{fa}, the detection probability P_{d}, the search space q, and the dwelltime. The search space q

is twice the codelength. The codelength is directly related to the spreading factor S when a single bit spans the entire PN code, so once we fix the spreading factor S, the codelength, and thus the search space q, is known. P_{d} and P_{fa} are related through the SNR, through (3.10), which expresses either P_{d} in terms of P_{fa} and γ_{DSSS}, or P_{fa} in terms of P_{d} and γ_{DSSS}. Thus, if we know γ_{DSSS}, we can choose a false alarm probability, which determines a detection probability, which allows us, together with the spreading factor S (determining q), to determine the acquisition time through (3.12), at a false alarm penalty K of 5 [13, 15]. Note that P_{fa} determines P_{d} through the SNR, but that in (3.12), the effective detection probability, P'_{d}, described in Section 3.3.2, is used.

The choice of P_{fa} affects the acquisition time. In our subsequent analysis, P_{fa} was optimized for each SNR value (in terms of total acquisition time). In Figure 5.2, we have plotted the results for DSSS and FODMA (4.18) against the SNR after despreading γ.

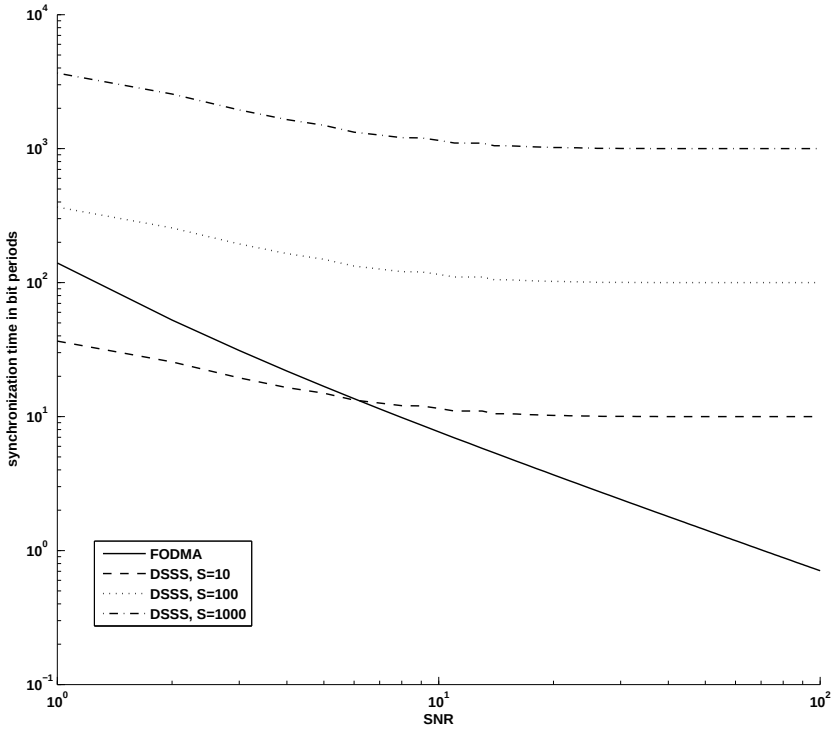

Figure 5.2: Bit-normalized synchronization time as a function of SNR after matched filtering for spreading factors 10, 100, 1000

The acquisition time for DSSS increases with the spreading factor, which is to be expected from inspection of (3.12), where the search space (and thus the spreading factor) is a linear term. The synchronization time of FODMA decreases inversely proportional with the SNR when the SNR is large. When it is not so large, the squaring loss is significant and the relationship between the SNR and time for FODMA

to synchronize becomes less linear. This can be seen from (4.18) as well. For DSSS, increasing the SNR beyond about 8 has little effect. There is a performance floor, determined by the codelength (and thus the spreading factor). At low SNRs, DSSS performance is affected by the SNR because P_{fa} and P_{d} depend on it through (3.10). At higher SNRs, the average required synchronization time for DSSS is close to twice the codelength, since P_{d} approaches one and P_{fa} becomes very small.

Concerning tracking, more then 244 symbol periods (Section 3.4) would be a very large contributor to the synchronization time for DSSS at spreading factors of 100 and lower. The limitations of the model used in [15] (very low SNRs) become obvious here. In practical scenarios (SNRs > 1), tracking should constitute a smaller portion of the total synchronization time, since in the scenario of 244 extra symbol periods, one could also decrease the step size (and increase search space) to achieve a tighter lock for small spreading factors. For spreading factors of 1000 and higher, the tracking operation would not be a significant contributor using the current DSSS model.

Concerning the symbol timing, we can conclude that its contribution to the total synchronization time is small, never more than two bit-times. At low SNRs (< 16 dB), the contribution of at least two symbol periods to the synchronization time is small. At higher SNRs (> 16 dB), the PLL acquisition period becomes small (smaller than a bit-period), such that the contribution of the symbol timing becomes dominant. However, compared to DSSS, FODMA still has a much shorter synchronization period at these SNRs.

Concerning the loss-of-lock condition for FODMA, we can conclude that since the SNR in the loop ρ' is 10, and the SNR in the loop ρ is at least 4 times as high (4.14), the mean time to lose lock $\mathrm{E}(T_{\mathrm{s}})$ is very long. This can be read from Figure 4.7, where $B_{\mathrm{l}}\mathrm{E}(T_{\mathrm{s}})$ is at least 10^5 at quite lower values of ρ. For very short data packets, loss of lock should rarely occur.

Chapter 6

Conclusions and recommendations

6.1 Conclusions

Looking at the results of Section 5.3, we conclude that a FODMA receiver might be a good choice for ultra-low-power, low duty-cycle WSNs. The short synchronization time (on the order of 3–100 bit periods) means that FODMA is a good choice for a wideband radio receiver for packets which are 100 bits or shorter. Only at low spreading factors (in the range of $S = 10$), long packets (> 100), and low SNRs does DSSS perform comparably to FODMA in terms of synchronization time.

One of the most interesting conclusions is that the FODMA synchronization time is independent of the spreading factor, assuming that the SNR after despreading is fixed. The FODMA transmitter, however, has to transmit more energy compared to the DSSS transmitter at high spreading factors, but the receiver can synchronize quickly, independent of the spreading factor. The independence of the synchronization time for FODMA on the spreading factor is due to the independent relationship between the frequency offset used in the transmitter and the bandwidth of the noise carrier (as long as the frequency offset is much lower than the bandwidth of the noise carrier). At higher SNRs, the time for the PLL to attain phase-lock can be made much shorther than one bit-period, and symbol timing estimation becomes the most significant contributor to the synchronization time.

6.2 Recommendations

Many subjects were not explored in this thesis, and several assumptions have been made.

The short synchronization time for FODMA is an advantage, but whether it outweighs the disadvantages of FODMA, namely, the higher required E_b/N_0, should be investigated further. This is depends on several factors, including the packet length,

the implementation of the radio receiver, and whether the WSN has a homogeneous or heterogenous architecture (where some nodes can have a larger power supply).

The other synchronization architectures proposed by Shang are also of interest. Would they outperform the FODMA architecture using a PLL in terms of synchronization time? If so, does the implementation of such a synchronization architecture require more power? Regarding the current synchronization architecture (the one using a PLL), it would be interesting to find out what an acceptable mean time to lose lock is for a given packet-length, and how large the loop noise bandwidth can then be made. Higher loop noise bandwidths imply faster acquisition, but the BER might also suffer under these conditions, along with an increasing probability to lose lock.

Furthermore, a digital PLL might also be an attractive option, since, after despreading, sampling rates are considerably lower, and there are many advantages to a digital approach (such as the reduced quality loss after sampling [24]).

The effect of fading and multipath channels on the synchronization performance was not researched either. Fading to FODMA was subject of a semi-analytical study [23]. However, synchronization could possibly suffer from such conditions as well. Furthermore, the synchronization time of a DSSS Rake receiver could be compared with a FODMA receiver as well, since FODMA does not need a Rake receiver to collect all the energy in the multipath components.

One assumption in Chapter 2 was that the power consumed by the despreading and sampling block was small compared to the power consumption in the front-end. However, the required synchronization time of the despreading and sampling block also depends on its power consumption. For example, the loop gain B_{l} affects the synchronization time, but also changes the power consumption. Some further research into this dependency is therefore recommended.

References

[1] K. Romer and F. Mattern, "The design space of wireless sensor networks," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 54–61, Dec. 2004.

[2] "Smart pill corporation." [Online]. Available: www.smartpillcorp.com

[3] J. M. Kahn, Y. H. Katz, and K. S. J. Pister, "Emerging challenges: Mobile networking for smart dust," *Journal of Communications and Networks*, vol. 2, 2000.

[4] "Electrical engineering, mathematics and computer science faculty website." [Online]. Available: www.ewi.utwente.nl/en/index.html

[5] J. Paradiso and T. Starner, "Energy scavenging for mobile and wireless electronics," *IEEE Pervasive Computing*, vol. 4, no. 1, pp. 18–27, Jan.-March 2005.

[6] T. Schellekens, "Transmit-reference modulation in wireless sensor networks," Literature study report. University of Twente. Department of Electrical Engineering. Telecommunication Engineering Group, January 2009.

[7] K. Cheun, "Performance of direct-sequence spread-spectrum rake receivers with random spreading sequences," *Communications, IEEE Transactions on*, vol. 45, no. 9, pp. 1130–1143, Sep 1997.

[8] S. Gezici, Z. Tian, G. Giannakis, H. Kobayashi, A. Molisch, H. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 70–84, July 2005.

[9] Institute of Electrical and Electronics Engineers, "IEEE 802.15.4: Wireless medium access control and physical layer specifications for low-rate wireless personal area networks."

[10] Bluetooth special interest group, "Bluetooth core specification, version 2.1," 2007.

[11] K. Pahlavan and A. Levesque, *Wireless information networks*. John Wiley and sons, 2005.

[12] M. Win and R. Scholtz, "Impulse radio: how it works," *IEEE Communications Letters*, vol. 2, no. 2, pp. 36–38, Feb 1998.

[13] L. Feng and W. Namgoong, "Fast acquisition for transmitted reference ultra-wideband systems with channelized receiver," *Signals, Systems and Computers, 2005. Conference Record of the Thirty-Ninth Asilomar Conference on*, pp. 1094–1098, 28 October - November 1, 2005.

[14] S. Farahmand, X. Luo, and G. Giannakis, "Demodulation and tracking with dirty templates for UWB impulse radio: algorithms and performance," *Vehicular Technology, IEEE Transactions on*, vol. 54, no. 5, pp. 1595–1608, Sept. 2005.

[15] M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt, *Spread spectrum communications handbook (revised ed.).* New York, NY, USA: McGraw-Hill, Inc., 1994.

[16] E. Chaffee and E. Purington, "Method and means for secret radiosignalling," US Patent 1,690,719, november 1928.

[17] R. Hoctor and H. Tomlinson, "Delay-hopped transmitted-reference RF communications," *2002 IEEE Conference on Ultra Wideband Systems and Technologies. Digest of Papers.*, pp. 265–269, 2002.

[18] J. Haartsen, X. Shang, J. Balkema, A. Meijerink, and J. Tauritz, "A new wireless modulation technique based on frequency-offset," in *12th Symposium on Communications and Vehicular Technology in the Benelux.* Enschede, The Netherlands: University of Twente, 2005. [Online]. Available: http://doc.utwente.nl/53150/

[19] X. Shang, "New radio multiple access technique based on frequency offset division multiple access," Master's thesis, University of Twente. Department of Electrical Engineering. Telecommunication Engineering Group, April 2004.

[20] D. Goeckel and Q. Zhang, "Slightly frequency-shifted reference ultra-wideband (uwb) radio," *IEEE Transactions on Communications*, vol. 55, no. 3, pp. 508–519, March 2007.

[21] J. Balkema, "Realization and characterization of a 2.4 GHz radio system based on frequency offset division multiple access," Master's thesis, University of Twente. Department of Electrical Engineering. Telecommunication Engineering group, August 2004.

[22] D. Mahrof, "Design of a fully integrated rf transceiver using noise modulation," Master's thesis, University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science, April 2008.

[23] A. Meijerink, S. Cotton, M. Bentum, and W. Scanlon, "Noise-based frequency offset modulation in wideband frequency-selective fading channels," in *16th Annual Symposium on Communications and Vehicular Technology in the Benelux*, Louvain-la-Neuve, Belgium, 2009.

[24] J. W. Ulrich Rohde, *Communications receivers: DSP, software radios, and design.* McGraw-Hill, 2001.

[25] J. G. Proakis, *Digital communications.* McGraw-Hill, 2001.

[26] K. Feher, *Digital communications, satellite/earth station engineering.* Prentice-Hall, 1981.

[27] H. Meyr and G. Ascheid, *Synchronization in Digital Communications, volume 1, Phase-, Frequency-Locked Loops and Amplitude Control.* John Wiley and Sons, 1990.

[28] F. M. Gardner, *Phaselock techniques.* Wiley-Interscience, 2005.

[29] M. Casu and G. Durisi, "Implementation aspects of a transmitted-reference UWB receiver," *Wireless Communications and Mobile Computing*, vol. 5, no. 5, pp. 537–549, 2005.

[30] R. Djapic, G. Leus, and A.-J. van der Veen, "Blind synchronization in asynchronous uwb networks based on the transmit-reference scheme," *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on*, vol. 2, pp. 1506–1510, Nov. 2004.

[31] N. van Stralen, A. Dentinger, I. Welles, K., J. Gaus, R., R. Hoctor, and H. Tomlinson, "Delay hopped transmitted reference experimental results," *2002 IEEE Conference on Ultra Wideband Systems and Technologies. Digest of Papers.*, pp. 93–98, 2002.

[32] J. Dunning, G. Garcia, J. Lundberg, and E. Nuckolls, "An all-digital phase-locked loop with 50-cycle lock time suitable for high-performance microprocessors," *Solid-State Circuits, IEEE Journal of*, vol. 30, no. 4, pp. 412–422, Apr 1995.

[33] J. Stensby, *Phase-locked loops: Theory and applications.* CRC Press, 1997.

[34] W. Lindsey and H. Meyr, "Complete statistical description of the phase-error process generated by correlative tracking systems," *Information Theory, IEEE Transactions on*, vol. 23, no. 2, pp. 194–202, Mar 1977.

[35] K. Feher, *Digital communications, satellite/earth station engineering.* Prentice-Hall, 1981.

[36] H. Meyr and L. Popken, "Phase acquisition statistics for phase-locked loops," *Communications, IEEE Transactions on*, vol. 28, no. 8, pp. 1365–1372, Aug 1980.

[37] W. Lindsey and M. Simon, *Telecommunication systems engineering.* Prentice-Hall, Inc., 1973.

[38] G. Ascheid and H. Meyr, "Cycle slips in phase-locked loops: A tutorial survey," *Communications, IEEE Transactions on*, vol. 30, no. 10, pp. 2228–2241, Oct 1982.

[39] K. Zigangirov, *Theory of code division multiple access communication.* Wiley-IEEE, 2004.