Synchronization and Packet Separation in Wireless Ad-hoc Networks

Relja Djapić

Synchronization and Packet Separation

in Wireless Ad-hoc Networks

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.dr.ir. J.T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op dinsdag 5 december 2006 om 10:00 uur
door Relja DJAPIĆ
Electrical Engineer van de Universiteit van Novi Sad, Servië
geboren te Novi Sad, Servië.

Dit proefschrift is goedgekeurd door de promotor:
Prof.dr.ir. A.J. van der Veen

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof.dr.ir. A.J. van der Veen,	Technische Universiteit Delft, promotor
Prof.dr.ir. P.M. Dewilde,	Technische Universiteit Delft
Prof.ir. P. van Genderen,	Technische Universiteit Delft
Prof.dr.ir. J.P. Linnartz,	Technische Universiteit Eindhoven en Philips Research
Prof.dr.ir. J.W.M. Bergmans,	Technische Universiteit Eindhoven
Prof.dr.ir. W. van Etten,	Universiteit Twente
Prof.dr. M.Ghogho,	University of Leeds, UK

ISBN-10: 90-9021356-2
ISBN-13: 978-90-9021356-9

To Katarina, Nikola and Dragana.

Contents

Chapter 1

Introduction

1.1 Spectrum sharing

Over the last two decades, advancements in the area of microelectronics have led to the miniaturization of integrated chips (IC's), increase in IC's processing speed, power efficiency and reduced weight. These advantageous features created a fruitful grounds for expansion of personal wireless (mobile) communication devices for voice, message, data, audio and video distribution. Examples are Bluetooth, WLAN 802.11..., Wi-Fi, digital audio broadcast DAB, digital video broadcast DVB, radio navigation systems (GPS), radio frequency identification (RF-ID) and one of the most widespread services — cellular mobile communications. We are witness of the daily introduction of new wireless gadgets, services and applications.

Evidently, human imagination is the only limiting factor for the number of possible wireless applications. On the other hand, radio spectrum is physically restricted. In order to avoid mutual interference of different services already in the 1920s the Federal Communications Commission (FCC) has been established in the U.S. in order to allocate the spectrum for each one of new services. Looking at the current occupancy of the spectrum one cannot avoid to have the impression that there is very little space for placement of new wireless systems. The extent of the greed for frequency spectrum can best be described by the example of licensing for the third generation (3G) mobile network services that occurred at the beginning of this century. Mobile providers were competing to obtain exclusive rights for deployment of 3G mobile systems (UMTS) and paid billions of dollars for licenses. The winners have obtained the licences but the market demand for mobile broadband services appeared to be insignificant. For this reason, mobile telephony providers suffered enormous losses that led to one of the worst financial collapses in the telecom sector ever.

Nevertheless, deployment of new technologies that exploit frequency spectrum more efficiently and more cooperatively may overcome the spectrum shortage [1]. A solution is found in the introduction of multiple-access schemes that can provide an increase in the network capacity, interference suppression and/or quality of the received signal. To be able to accommodate a large number of simultaneously active users, efficient allocation schemes should exploit extra degrees of freedom such as

1. **TDMA** — time division multiple access, also known as time division multiplexing. TDM allows a number of users to access a single radio-frequency (RF) channel by allocating unique time slots to each user within each channel. This multiplexing scheme, for example, together with frequency division multiplexing (FDM) is deployed in the GSM mobile telephony standard.

2. **FDMA** — frequency division multiple access - also known as Frequency Division Multiplexing (FDM) allocates a single frequency band to one user at a time. In a conventional multicarrier approach a guard interval is introduced between subbands to avoid interchannel interference. As a result, the use of the spectrum becomes highly inefficient. Conversely, each user may be allocated several frequency subbands in order to increase its data rate. Placing subband carrier frequencies closer to each other results in increased efficiency but leads to the problem of the intercarrier interference, the so called *crosstalk*. A solution to this issue is found in deployment of Orthogonal Frequency Division Multiplexing (OFDM), where subbands are spectrally shaped to deliver maximum energy at the subband center frequency while producing no interference at all other carrier frequencies [2]. This scheme is susceptible to synchronization inaccuracy and residual carrier phase rotations but is simple to implement at the transmitter side as multiplexing is achieved by simple and computationally efficient Inverse Fast Fourier Transformations (IFFT). OFDM is deployed in several high data rate communication systems such as Digital Video Broadcasting (DVB), xDSL — digital subscriber line standards, Digital Audio Broadcasting (DAB) in the European market, Wireless Local Area Networks WLAN's, etcetera.

3. **CDMA** — code division multiple access is a "spread spectrum" scheme in which a code spreads the signal energy over a wide frequency band and below the noise level. Each data symbol is multiplied by a code comprised of a number of chips with duration T_c much shorter than the symbol duration T_s, $i.e.$, $T_c \ll T_s$. The spectrum of the coded signal is inversely proportional to the chip duration T_c. Multiple users are allocated different, mutually orthogonal spreading codes. The orthogonality principle implies large code autocorrelation values while the cross-correlation of distinct codes tends to be minimized.

   The orthogonality of the codes facilitates the extraction of the desired user at the receiver side and suppression of multiuser interference. This technique allows simultaneous coexistence of multiple users/services in the same frequency band.

The idea was first deployed in 1960s, by the U.S. army during the Cuban crisis as a means of preventing eavesdropping from the enemy. Currently, the CDMA is deployed in the 3^{rd} generation of celular mobile communications-UMTS.

4. **SDMA** — space division multiple access exploits the ability of a directive antenna or an antenna array to separate multiple active users based on their distinct directions of arrival even if they share the same frequency band. Signals with a particular direction of arrival (DOA) are enhanced while others are suppressed. This process is called *beamforming*. Interference cancellation and extraction of the signal of interest is achieved $e.g.,$ by exploiting the antenna's mechanical properties (directivity) that coherently adds all signal components that arrive from a predefined direction. This principle is exploited by the parabolic dish antennas. Alternative would be to use an array of omnidirectional antennas and to extract the desired source by signal processing algorithms deployed on the signals at the antennas' outputs.

Another solution for a more efficient use of frequency spectrum can be achieved by a different organization of the network structure. Currently, wireless devices (nodes) are accessed via an access point (AP), $i.e.,$ a base station that covers a number of users in a particular area, the so called "cell". Reducing the sizes of cells in a fashion where each user communicates with just its neighboring nodes and applying a multi-hop access scheme, information could be transfered to the destination by hopping via the nodes situated in between the two users. In this manner, the coverage area of each of the devices is reduced, inducing reduced power-consumption. Besides, due to the limited coverage area frequency bands could be reused more often without causing any interference. As stated in [3]: "An important aspect of fourth generation mobile systems may be the possibility that the user will own a part of the network infrastructure". This approach brings us to the self organizable networks also known as ad-hoc systems where devices communicate to other users in their vicinity, $i.e.,$ computers, mobile phones, printers, cameras, without the requirement for a common access point.

The increase in the degrees of freedom due to multiple access schemes leads to an increased number of simultaneously active users, provides higher quality and interference mitigation but at the same time requires development of algorithms that can, in an efficient way, cope with the utilization of the new dimensions.

1.2 System description

In this thesis mobile ad-hoc networks (MANETs) are considered where a number of mobile users are located in the same area and share a common radio channel for data transmission. The environment and the organization of the mobile units can be highly dynamic, $i.e.,$ the network topology is variable and changes in an unpredictable manner. Some users within a MANET are only able to communicate with neighboring

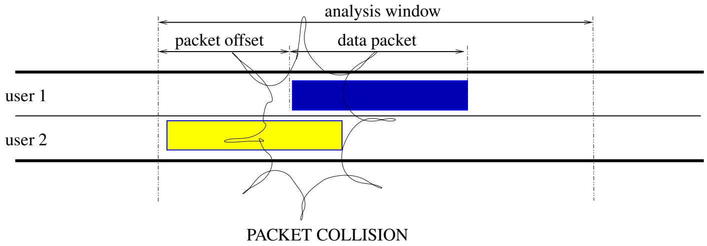


Figure 1.1: In existing wireless networks, the simultaneous presence of multiple users leads to *packet collision*. Consequently, packets may need to be retransmitted inducing reduced system capacity.

users, others may as well be connected to an internet backbone via an access point. A packet-oriented data transmission is presumed where a user, in case it has something to transmit, broadcasts a finite duration data packet and for the rest remains silent. In existing systems, when several users are simultaneously active a packet collision occurs (see figure 1.1), causing the loss of the data content and inducing the retransmission of the packets. As a result, the efficiency and capacity of such systems could be significantly reduced, especially in multiuser environments with dense data traffic. Several protocols have been proposed in the literature to organize the packet transmission. We mention here just the two most common ones:

1. **ALOHA**-[4] is basically a kind of a TDMA transmission scheme where each user transmits a data packet in a completely asynchronous and random fashion and waits for an acknowledgment. If the confirmation of successful reception is not received within a predefined time-out period the occurrence of a packet collision is assumed and the data is retransmitted. A random time offset is introduced to reduce the probability of repeated collisions.

2. **CSMA**- [5] Carrier Sense Multiple Access is a protocol in which each terminal, prior to any transmission, verifies whether the channel is being used by any other user. This process is called *Carrier Sensing* and is performed in order to reduce the packet collision probability. Only if the channel is sensed idle the terminal begins to broadcast the data.

Note that each of the above-mentioned protocols does not overcome the packet collision problem but rather provides data transmission with a predefined failure rate for a fixed number of active users. The packet collision and hence retransmission problem is even more encountered in cases where networks handle a significant amount of

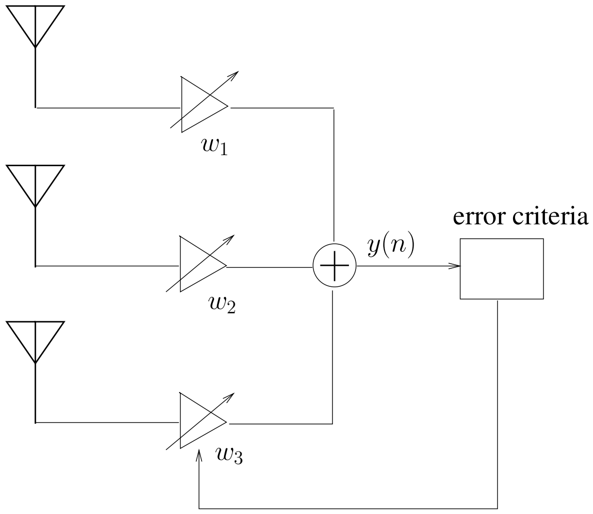


Figure 1.2: In the process of parameter estimation and detection the output of each receive antenna is weighted by w_i (scaling and phase-rotation) in order to reconstruct the transmitted signal.

data traffic, $i.e.$, a large number of simultaneously active users. A more extensive description of the existing protocols is provided in Part I of this thesis.

1.3 Problem statement

The central problem in the proposed multiuser asynchronous MANET scenario is how to extract a data sequence of a user of interest from a linear superposition of signals originating from a number of users. In addition to the multi-user interference, signals may be corrupted by additive white Gaussian noise.

To better conceive our targets we can compare our problem to a similar one in the field of the human speech communications, the so called *cocktail party effect*. This effect represents the ability to focus one's listening attention on a single talker among a cacophony of conversations and background noise [6]. Exploiting features of the speech such as the direction of arrival, color of the voice, talking speed, accent [6] etc. humans are able to separate one particular talker from the mixture of several conversations.

Similarly, in our research, baseband digital signals represent the input to the system on which our signal processing algorithms are deployed in order to estimate parameters that enable subsequent detection of the transmitted data. In an asynchronous system users transmit data packets in a completely unscheduled manner. Therefore as a first step a packet time offset with respect to the beginning of the analysis window has to be determined. At the same time, a beamformer is estimated, $i.e.,$ a weight vector that coherently combines samples of the system's input in order to obtain the signal of interest at the output. Two approaches are possible in estimating the packet offset and a beamforming vector:

1. **Training based**. In addition to the information symbols a number of training symbols are contained in the transmitted data packet. Propagation through the channel may distort the content and the structure of the transmitted packet. Exploiting the knowledge of the training symbols $y_0(n)$ and their position within the packet, the channel impulse response can be estimated and its effect corrected. In the training based schemes the antenna outputs in figure 1.2 are scaled and phase-aligned by w_i such that the distance between the systems output $y(n)$ and the known symbols $y_0(n)$ is minimized, or

$$\min_{\mathbf{w}} \|y(n) - y_0(n)\| \ , \tag{1.1}$$

   where $\mathbf{w} = [w_1, \cdots, w_M]^{\mathrm{T}}$ collects all the weights w_i into a vector and $i = 1, \cdots, M$ is the index of the antenna. Note that the efficiency of the scheme as well as the data rate are reduced by the introduction of training sequence overhead.

2. **Blind** schemes utilize specific signal properties in the process of $e.g.,$ time parameter estimation and detection. Consequently, the signal bandwidth is not reduced. Constant modulus, finite alphabet, cyclostationarity, higher order statistics, superimposed training are examples of signal characteristics that could be used in the estimation process and that do not reduce the system's capacity. In case detection is based on the signal's constant modulus (CM) property the error criteria at the output of the detector (see figure 1.2) becomes

$$\min_{\mathbf{w}} \||y(n)|^2 - 1\| \ \text{ for } \ n = 1, \cdots, N \ , \tag{1.2}$$

   where N denotes the length of the data sequence. The antenna outputs in figure 1.2 are scaled such that after adding them coherently its modulus, $i.e.,$ $|y(n)|^2$ becomes constant over the whole sequence length. A drawback of the constant modulus criteria is that it cannot resolve the ambiguity which is the user of interest in case multiple constant modulus sources are active. To resolve this drawback of the CM signals, in this thesis we propose to superimpose a small, user specific known modulus variation over the CM signal. In such cases the modulus of the system's output $y(n)$ needs to match the known modulus pattern $c(n)$

$$\min_{\mathbf{w}} \||y(n)|^2 - c(n)\| \ \text{ for } \ n = 1, \cdots, N \ . \tag{1.3}$$

Note that the data sequence of interest can be detected from the received mixture of multiple users and noise only in cases where the beginning of the desired data packet is known. For this reason the estimation of the packet offset has crucial importance.

In our work we consider mobile ad hoc networks with finite length data package transmission. For this system we provide deterministic block-based processing schemes in contrast to adaptive estimation algorithms. The latter approach uses each newly received data sample to improve the estimates. This may be time consuming and the final result may not converge to the desired solution.

In this thesis we propose several block-based estimation and detection algorithms that provide answers to the following questions:

1. How can multiple partially overlapping signals be separated in ad hoc networks with packet oriented data transmission?

2. How can relevant synchronization parameters be estimated at the same time?

3. Is it possible to estimate the packet offset and detect the data *blindly*, $i.e.,$ without introducing additional training overhead?

1.4 Outline of the thesis

To overcome the collision and interference problems in MANETS in this thesis we propose transceiver schemes where each user is assigned a specific code. The code is used to identify the user and also to determine the beginning of the data packet within an analysis window. In particular, we consider two different kind of systems:

1. **A narrowband system** is a communication system in which a time delay can be approximated by a phase shift of the signal (see Section 2.1). Exploiting the features of such a system, introducing an antenna array at the receiver side and superimposing the code over the transmitted data packets, facilitates subsequent user separation and suppression of the interfering sources. For such systems we propose combined blind synchronization and detection schemes.

   The scenario we consider is presented in figure 1.3 where different users transmit data packets simultaneously. In order to suppress the interference, an antenna array is introduced at the receiver side. We assume that the packet of the user of interest completely falls within the analysis window. Each user is assumed to generate a constant modulus signal, $i.e.,$ a signal with constant amplitude. Subsequently, a small known amplitude variation (the "color code") is inserted over each symbol within a data packet. The color code is a superimposed training sequence, which multiplies the transmitted signal without

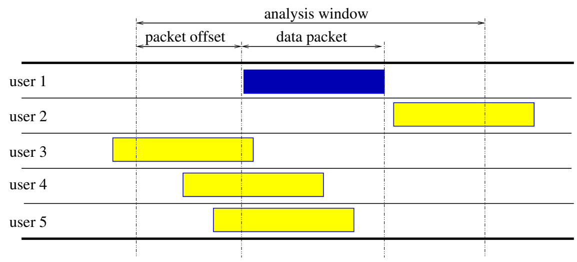


Figure 1.3: An example of the transmitted data packets and the analysis window in the multi-user environment. User 1 (darkened block) represents the user of interest while the rest are interfering users.

increasing the transmission rate. The structure of each data packet is depicted in figure 1.4.

A problem not considered before is that, in an asynchronous packet network, the receiver does not know precisely when the next packet will arrive. Thus, in a given window of collected samples, it has to try to match the code to the data window at every possible offset, and detect the symbols only if it has found a good match. This is obviously rather inefficient. Instead, we consider deterministic algorithms that perform over the block of the received data: after a Fourier transform of the data window, the offset delay becomes a phase shift, which can be estimated using ESPRIT-like [7, 8] techniques (see also Chapter 2). Such algorithms are algebraic and do not contain a combinatorial search. A new aspect compared to ESPRIT is that only a single column in the column span has the expected parametric structure, rather than all columns. This column is computed using a column-span intersection.

In Part I of the thesis we describe some issues arising in mobile ad hoc networks in more detail, namely, packet collision, retransmission and the hidden-exposed terminal problem. In addition the block deterministic Algebraic Constant Modulus Algorithm (ACMA) is introduced as it represents the starting point of our work. ACMA was originally derived by Van der Veen in [9] and performs user separation based on signal's Constant Modulus (CM) property. Eventhough ACMA can simultaneously separate all users in a multiuser environment it is unable to resolve the original ordering of signals as they all possess the same – CM property. To solve this ambiguity, we propose to superimpose a known modulus code at the symbol level over a CM data sequence. In this manner we come to the Known Modulus Algorithm (KMA). In Part I we propose a joint blind synchronization and beamformer estimation algorithms in an asynchronous multiuser scenario. Here, blind refers to the fact that the transmitted data sequence is unknown. The joint synchronization and beamformer estima-

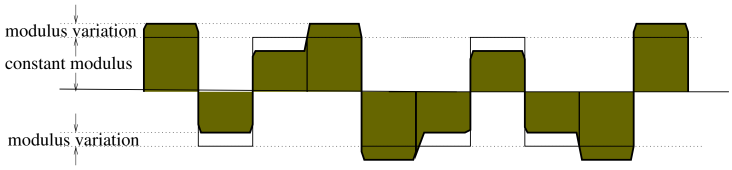


Figure 1.4: A constant modulus (CM) signal with coded amplitude variations.

tion makes our KMA highly efficient. Performance of the KMA is verified through simulations and is tested on a multiple antenna experimental platform - MURX, developed at Delft University of Technology. Furthermore, Cramer-Rao Bounds for blind and training based packet offset estimation are derived.

The Known Modulus algorithm falls in the group of *carrier based* transmissions where a baseband signal is translated to a higher carrier frequency prior to any transmission. In contrast to this approach we have also developed synchronization and detection schemes for a *carrierless* broadband technique known as Ultra Wideband Impulse Radio (UWB-IR).

2. **Ultra wideband systems-UWB.** In general, Ultra wideband (UWB) systems represent broadband transmission techniques in which the signal spectrum is larger than 500MHz. It arose as a solution to ever increasing demand of new services and application for frequency spectrum. The basic idea is to use an ultra wide frequency band for unlicensed transmission in which users broadcast at extremely low power levels and in that manner enable the coexistence with already deployed communication systems.

   As mentioned, in most of the existing broadcasting schemes a baseband signal is shifted to some higher, carrier frequency f_c to acquire improve transmission efficiency. Conversely, in a UWB-IR system information is conveyed by means of pulses of extremely short duration (less than a nanosecond). This results in a large bandwidth of the signal - several GHz. Information bits modulate the pulses by changing their polarization or position within a frame of a known duration (see figure 1.5). The latter is known as pulse position modulation (PPM). A huge bandwidth gives rise to the potentially high data rates of a UWB system. Multiple users could be distinguished by assignng a different CDMA-like spreading code to each of them.

   Generation of narrow pulses and their modulation by information bits can be accomplished in a power efficient way. Additionally, exploiting an autocorrelation receiver structure that overturns the estimation of potentially long channel impulse responses brings about the possibility of creating a simple, low cost, power efficient high data rate UWB transceiver device that affords high data rate communications in multiuser environments.

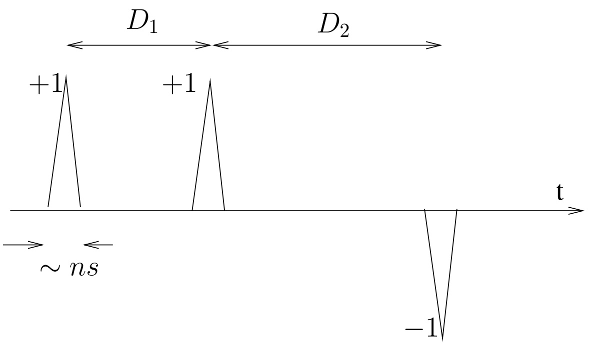


Figure 1.5: Impulse radio principle: A pulse of extremely short duration (smaller than a ns) carries the information. Many modulation schemes are applicable $e.g.$, Pulse Amplitude Modulation (PAM) determines the polarization of the pulses and Pulse Position Modulation (PPM) determines the distance between two pulses.

In Part II of this we propose a data model for an asynchronous Transmit Reference UWB (TR-UWB) scheme in which pulses are transmitted in pairs – the first is a reference used in the detection of the second pulse that carries the information. Such a scheme allows us to side step a computationally complex channel estimation. In UWB systems estimation of the beginning of the received data packets is of crucial importance in subsequent detection process. In Part II we have developed a subspace based, blind, high resolution and low-complexity synchronization scheme applicable in multiuser UWB systems and verified its performance through simulations.

1.5 Context

The research described in this thesis was carried out within two projects at Delft University of Technology:

1. **UbiCom** The Ubiquitous Communications (UbiCom) program has been a multidisciplinary research program at Delft University of Technology (1998 – 2002). The program aimed at carrying out research needed for specifying and developing wearable systems for mobile multimedia communications. The signal processing topics comprised the definition of multiple access synchronization and configuration schemes, and the development and implementation of new multiple antenna algorithms for channel separation and noise suppression.

2. **AirLink** The AIRLINK project ("Ad-hoc Impulse Radio: Local Instantaneous Networks") has been a Freeband Impulse project at Delft University of Technology that investigated the merits of Ultra Wideband (UWB) radio in achieving low power, short range, high data rate communications.

   Development of synchronization and source separation schemes for asynchronous multiuser impulse radio ad-hoc networks have been developed within the signal processing workpackage.

1.6 Publication list

The research topics throughout the PhD research comprised synchronization and source separation schemes for narrowband systems, in particular the Known Modulus Algorithm (KMA), and Transmit Reference Ultra Wideband (TR-UWB) systems. As a result two journal and six conference papers have been prepared

1. R. Djapic, G. Leus, A.-J. van der Veen and A. Trindade, "Blind synchronization in asynchronous ultra wideband (UWB) networks based on the transmit-reference scheme ", in submission to UWB special issue of EURASIP Journal on Wireless Communications and Networking.

2. R. Djapic, A.-J. van der Veen and L. Tong , "Synchronization and packet separation in wireless ad hoc networks by known modulus algorithms", IEEE Journal on Selected Areas in Communications (JSAC), Volume 23, Issue 1, pp 51-64, January 2005.

3. R. Djapic, N. Shi and G. Leus, "Initial code exchange for asynchronous TR-UWB ad hoc networks", Chinacom–International Conference on Communications and Networking, Beijing, October 2006.

4. R. Djapic, G. Leus and A.-J. van der Veen, "Synchronization and detection for transmitted reference UWB systems", In Asilomar Conf. on Signals, Systems, and Computers, IEEE, November 2005.

5. R. Djapic, G. Leus, A. Trindade and A.-J. van der Veen, "Blind synchronization in multiuser transmit-reference ultra wideband systems", In 13th European Signal Processing Conference, September 4-8, 2005, Antalya, Turkey.

6. R. Djapic, G. Leus and A.-J. van der Veen, "Blind synchronization in asynchronous UWB networks based on the transmit-reference scheme", In Asilomar Conf. on Signals, Systems, and Computers, IEEE, November 2004.

7. R Djapic, G. Leus and A.-J. van der Veen, "Cramer-Rao bounds for blind and training based packet offset estimation in wireless ad hoc networks", In

11th IEEE Symposium on Communications and Vehicular Technology (SCVT), November 2004, Gent (Belgium) .

8. R. Djapic and A.-J. van der Veen, "Blind synchronization in asynchronous multiuser packet networks using KMA", In IEEE workshop on Signal Processing Advances in Wireless Communications (SPAWC), Rome (Italy), June 2003.

9. R. Djapic and A.-J. van der Veen, "Packet separation using the Known Modulus Algorithm—experimental results", In 3rd International Symposium on Mobile Multimedia Systems and Applications (MMSA'02), Delft, The Netherlands, December 2002.

Chapter 2

Preliminaries

In this chapter general signal processing aspects needed for a better understanding of this thesis are described. The content of this chapter is derived from [10, 11, 12].

2.1 Signal properties

In a communication system a message to be transfered needs to be preprocessed prior to any transmission. Initially the message is discretized according to Nyquist criterion that provides no loss of useful information and therefore permits the reconstruction of the signal at the receiver side. The Nyquist criterion is met if the sampling frequency is equal to or greater than twice the maximal frequency contained in the message. Such a discretized signal is subsequently quantized and represented by a binary sequence, $i.e.,$ a stream of bits (zeros and ones). A coding step, subsequently, converts the discrete sequence $\{0, 1\}$ into another alphabet, $i.e.,$ constellation. The alphabet in general comprises complex numbers. The smaller the size of the alphabet the higher its resistivity to the effects of the noise due to a larger distance between the symbols. Inversely, larger constellations can provide higher data rates but are highly susceptible to errors in noisy and multi-user environments. At this point, the sequence is convolved with a continuous "pulse shape" in order to obtain an analog signal with good spectrum characteristics, $i.e.,$ finite bandwidth. This process is also known as spectral shaping. The baseband signal representation becomes

$$s(t) = \sum_{k=-\infty}^{+\infty} s_k g(t - kT) \ ,$$

where $g(t)$ represents the pulse shaping signal. Finally, the baseband signal is converted to a bandpass signal $q(t)$ by moving it to the carrier frequency f_c that is subsequently broadcasted,

$$q(t) = \text{real}\{s(t) * e^{j2\pi f_c t}\} = x(t)\cos(2\pi f_c t) - y(t)\sin(2\pi f_c t)\ .$$

Here, $x(t) = \text{real}\{s(t)\}$ and $y(t) = \text{imag}\{s(t)\}$ are defined as inphase and quadrature component of the signal $s(t)$ that represents the complex envelope of the signal $q(t)$.

Narrowband assumption

At the receiver side, transmitted signal is reconstructed by multiplying $q(t)$ with $\cos(2\pi f_c t)$ and $\sin(2\pi f_c t)$ and by subsequent low-pass filtering. This approach is called *coherent demodulation* as the frequency and phase generated at the receiver need to match to the carrier used in the transmission. Consider the simplest case of a wave propagation where the signal at the receiver side represents an attenuated and delayed version of the transmitted signal. Without loss of generality and for the simplicity of exposure we assume no attenuation occurred. A delayed version of a transmitted signal $q(t)$ now becomes

$$q(t-\tau) = q_\tau = \text{real}\{s(t-\tau)e^{j2\pi f_c(t-\tau)}\} = \text{real}\{s(t-\tau)e^{j2\pi f_c t}e^{-j2\pi f_c \tau}\}$$

where the complex envelope of the signal is $s(t-\tau)e^{-j2\pi f_c \tau}$. Let $S(f)$ represent the baseband spectrum and assume $S(f) \neq 0$ for frequency range $f \in [-W/2, W/2]$, represents the baseband spectrum of $s(t)$. Using a Fourier transformation, $s(t-\tau)$ can be written as

$$s(t-\tau) = \int_{-W/2}^{W/2} S(f)e^{j2\pi f t}e^{-j2\pi f \tau}df.$$

In case $|2\pi f \tau| \ll 1$ in the whole baseband $|f| < \frac{W}{2}$ or equivalently $|W\tau| \ll 1$, the last equation yields

$$s(t-\tau) \approx \int_{-W/2}^{W/2} S(f)e^{j2\pi f t}df = s(t).$$

The delayed version of the baseband signal becomes

$$s(t-\tau) \approx s(t)e^{-j2\pi f_c \tau} \ \ for \ \ W\tau \ll 1.$$

This condition $W\tau \ll 1$ is known as a *narrowband assumption* under which a time delay can be approximated by a phase shift of the signal. This assumption is extensively used in case an antenna array is being deployed at the receiver side where a time delay between reception at each antenna is substituted by a phase rotation of the signal. However, τ can also represent differences in path length in case a multipath channel is considered.

Far field assumption

An electromagnetic wave is radially spread in free space around the source. All points in the medium with the same radius will possess the same amplitude and phase. In cases in which a signal impinging on an antenna array originates from a sufficiently distinct source the wave front can be approximated to be planar. This is known as a *far field assumption*.

2.2 Antenna array receiver model

The constant increase in the number of new wireless services and applications, is held backwards by the restricted availability of frequency bands. A solution may be found by exploiting spatial dimension for accommodating new users. Hence, several users could be distinguished based on their dissimilar direction of arrival even though they share the same frequency channel. In this section we describe the aspects of signal enhancement, beamforming and receiver models in case an antenna array is deployed at the receiver side.

2.2.1 Signal enhancement

Assume signal $s(t)$ arrives at an antenna array. At each of the antenna elements the same signal is received affected by additive white Gaussian noise $n(t)$. This can be represented as

$$x_m(t) = s(t) + n_m(t)$$

where $m = [1, \cdots, M]$ depicts antenna's ordinal number. The noise variance is $E\{|n_m|^2\} = \sigma^2$. Averaging a coherently received signal from M antenna outputs yields

$$y(t) \quad = \quad \frac{1}{M}\sum_{i=1}^{M} x_m(t) = s(t) + \frac{1}{M}\sum_{i=1}^{M} n_m(t) \tag{2.1}$$

Presuming that noise has equal variance at all elements of the array and that it is uncorrelated from antenna to antenna, $i.e.,$ $E\{n_i(t) \cdot n_j(t)\} = 0$ for $i \neq j$ the noise variance on $y(t)$ is $E\{|\frac{1}{M}\sum_{i=1}^{M} n_m(t)|^2\} = \sigma^2/M$, while the signal energy $E\{|s(t)|^2\}$ remains unchanged. In this manner a signal enhancement, also known as antenna array gain, of M is achieved.

2.2.2 Antenna array response

Consider an antenna array with M elements placed in a line. A source $s_0(t)$ radiates the signal at carrier frequency f_c. The source is assumed to be distinct enough so

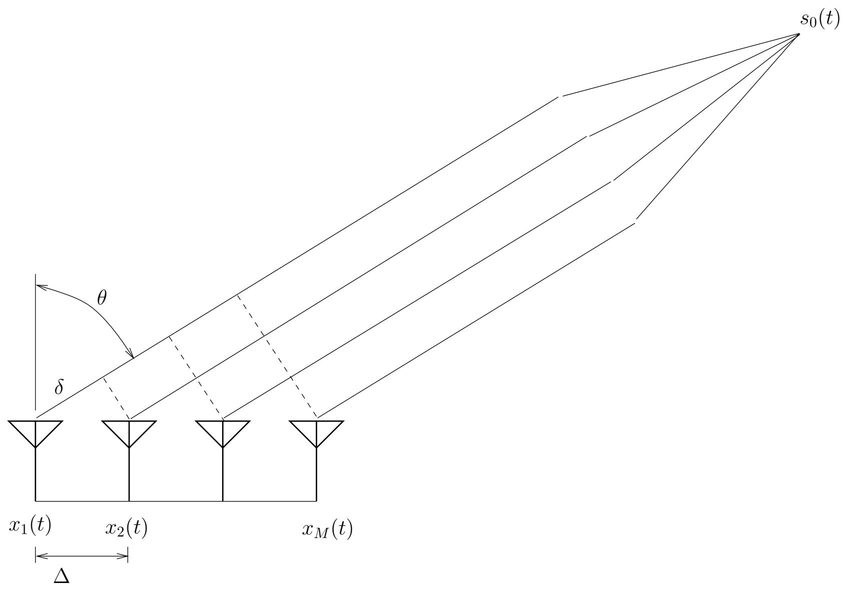


Figure 2.1: A signal from a far field source impinges an antenna array with equal element spacing-uniform linear array (ULA).

that a plane wave impinges onto the antenna array. The received sequence at the i-th antenna is

$$x_i(t) = a_i(t, \theta, f) * s_0(t - T_i)e^{-j2\pi f_c T_i} \; , \tag{2.2}$$

where $*$ denotes convolution, T_i determines the signal propagation time from the source to the ith element of the array and $a_i(t, \theta, f)$ denotes the antenna response as a function of time, angle of arrival and frequency. In case the response does not change in time, is identical in all directions (omnidirectional antenna) and is flat (constant) in the bandwidth of interest $a_i(t, \theta, f) = a_i$ applies. Presuming all antennas are identical yields $a_i = a_0(\theta) = a_0$ for all $i = 1, \cdots, M$. In case only a direct path is considered (2.2) becomes

$$x_i(t) = a_0 s_0(t - T_i)e^{-j2\pi f_c T_i} \; . \tag{2.3}$$

A multipath propagation model can be achieved by a linear extension of (2.2) due to the linearity of the medium.

Let $\mathbf{x}(t) = [x_1(t), \cdots, x_M(t)]^{\mathrm{T}}$ represents a vector that collects outputs of all antennas at time instant t. Under a narrowband assumption for a far field source $s(t)$

the baseband representation of an antenna array can be described as

$$\mathbf{x}(t) = a_0(\theta) \begin{bmatrix} s(t) \\ s(t-\tau_1) \\ \vdots \\ s(t-\tau_{M-1}) \end{bmatrix} = \begin{bmatrix} 1 \\ e^{-j2\pi f_c \tau_1} \\ \vdots \\ e^{-j2\pi f_c \tau_{M-1}} \end{bmatrix} a_0(\theta)s(t) \; . \tag{2.4}$$

In case a uniform linear array is used the delay between two consecutive elements is always τ so the phase rotation between the elements of the antenna can be written as a function of the angle of arrival θ (see figure 2.1)

$$2\pi f_c \tau = -2\pi f_c \frac{\delta \sin(\theta)}{c} = -2\pi \frac{\delta}{\lambda} \sin(\theta) = -2\pi\Delta \sin(\theta),$$

where c denotes the speed of light, λ is the wavelength of the signal and Δ represents the interelement spacing of the antenna array expressed in multiples of the signal wavelength. The array output can now be written as

$$\mathbf{x}(t) = \begin{bmatrix} 1 \\ e^{j2\pi\Delta \sin(\theta)} \\ \vdots \\ e^{j2\pi(M-1)\Delta \sin(\theta)} \end{bmatrix} a_0(\theta)s(t) =: \mathbf{a}(\theta)s(t). \tag{2.5}$$

where $\mathbf{a}(\theta)$ is defined as *array response vector* and represents the response of the array to a planar wave with direction of arrival (DOA) θ. $\mathbf{a}(\theta)$ is also known as spatial signature vector, action vector, array propagation vector and signal replica vector [11]. Sampling the antennas output and stacking the vectors horizontally for N consecutive time instants $\mathbf{X} = [\mathbf{x}(1), \cdots, \mathbf{x}(N)]$ the data model (2.5) becomes

$$\mathbf{X} = \mathbf{a}(\theta)[s(1), \cdots, s(N)] = \mathbf{a}(\theta)\mathbf{s} \; .$$

In a multiple user case this model extends to

$$\mathbf{X} = [\mathbf{a}(\theta_1) \cdots \mathbf{a}(\theta_d)] \begin{bmatrix} s_1(1) & \cdots & s_1(N) \\ \vdots & \vdots & \vdots \\ s_d(1) & \cdots & s_d(N) \end{bmatrix} = \mathbf{AS} \tag{2.6}$$

where $\mathbf{X} : M \times N$, $\mathbf{A} : M \times d$ and $\mathbf{S} : d \times N$, with M representing the number of antennas in an array, d number of sources and N the total number of samples collected.

Adding the contribution of white Gaussian noise (2.6) becomes

$$\mathbf{X} = \mathbf{AS} + \mathbf{N} \; , \tag{2.7}$$

where $\mathbf{N} : M \times N$ collects the samples of the Gaussian noise.

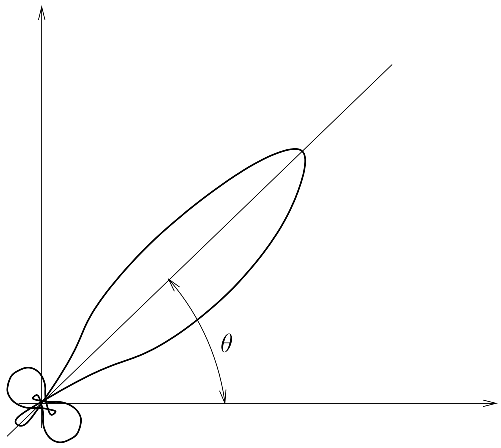


Figure 2.2: Beamforming principle: A source coming from direction θ is amplified while others are weakened.

2.2.3 Beamforming

A beamformer can be viewed as a spatial filtering process that separates distinct signals with a possibly overlapping frequency content that originate from different spatial locations [12]. It is actually a process of "forming a beam" where a source from a desired spatial location is amplified while other sources are attenuated, see figure 2.2. A well known beamformer that exploits the mechanical properties of the antenna is a parabolic dish antenna. The shape of a parabolic antenna provides that each part of the plane electromagnetic wavefront from the look direction is coherently added at the feed of the antenna while the signals from other directions are weakened. The enhancement of the signal from the look direction is achieved due to the fact that all the waves generated by the source pass the same distance to reach the antenna's feed.

A beamforming structure similar to a parabolic antenna can be accomplished by means of an antenna array of omnidirectional elements. Imagine that the parabolic dish is stretched into a plane and antennas are placed at positions denoted by x_1, x_2 and x_3, see figure 2.3. Introducing complex weightings w_i for each of the antenna outputs x_i makes feasible the creation of a beamformer for a specific direction of arrival (DOA). The output of such a discrete system after beamforming can be represented as [12]

$$y(n) = \sum_{i=1}^{M} w_i^* x_i(n) ,$$

where superscript * denotes a complex conjugation. A beamformer that involves only weighting of the sensors output is applicable for narrowband signals, $i.e.,$ in cases where the signal bandwidth is smaller than the inverse of the channel delay spread. In this manner the channel frequency response becomes flat over the whole spectrum of the observed signal. As a result equalization is not required.

In case a broadband sequence is considered, received over L dissimilar multiple propagation paths a space-time equalization needs to be performed. The equalizer recollects as much energy as possible of the original signal arriving from distinct directions in different time instants (figure 2.4). The output of a space-time equalizers yields

$$y(n) = \sum_{l=0}^{L-1} \sum_{i=1}^{M} w_{i,l}^{*} x_i(n-l)$$

where $y(n)$ is an estimated discrete time sequence of the user of interest, L represents the duration of the channel impulse response (in case of line of sight LOS propagation $L = 1$ otherwise $L > 1$), $w_{i,l}$ represents a beamforming coefficient at the i-th sensor at delay tap l and $x_i(n-l)$ is a corresponding discretized i-th antenna output. Let us define

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_L \end{bmatrix} \quad \text{where} \quad \mathbf{w}_i = \begin{bmatrix} w_{i,1} \\ \vdots \\ w_{i,L} \end{bmatrix}$$

and

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_L \end{bmatrix} \quad \text{where} \quad \mathbf{x}_i = \begin{bmatrix} x_{i,1} \\ \vdots \\ x_{i,L} \end{bmatrix} .$$

Using these definitions both beamformer and space-time equalizer can be represented in a vector form

$$y(n) = \mathbf{w}^{\mathrm{H}} \mathbf{x} ,$$

where $\mathbf{w}^{\mathrm{H}}$ is the complex conjugate transpose of $\mathbf{w}$.

2.3 Zero-Forcing and LMMSE receivers

In this section we present two approaches in the receiver design. Starting from a multiuser data model in white Gaussian noise derived in (2.7)

$$\mathbf{X} = \mathbf{AS} + \mathbf{N}$$

two approaches based on the linear least squares minimization can be followed in the design of deterministic receivers.

1. **Deterministic model matching** For the model defined in (2.7), minimizing the model fitting error for the known $\mathbf{A}$ yields

$$\min_{\mathbf{S}} \parallel \mathbf{X} - \mathbf{AS} \parallel_F^2 , \tag{2.8}$$

where $\| \mathbf{Y} \|_F$ denotes the Frobenius norm of the observed matrix $\mathbf{Y}$, $i.e.$, the energy contained in its entries

$$\| \mathbf{Y} \|_F = (\sum |y_{i,j}|^2)^{1/2} \ .$$

From (2.8) we obtain the estimate of $\mathbf{S}$ as

$$\hat{\mathbf{S}} = \underset{\mathbf{S}}{\operatorname{argmin}} \| \mathbf{X} - \mathbf{AS} \|_F^2 \quad \Leftrightarrow \quad \hat{\mathbf{S}} = \mathbf{A}^\dagger \mathbf{X} \tag{2.9}$$

where † represents the Moore-Penrose pseudo inverse operation with

$$\mathbf{A}^\dagger = (\mathbf{A}^{\mathrm{H}} \mathbf{A})^{-1} \mathbf{A}^{\mathrm{H}} \ . \tag{2.10}$$

From (2.9) follows the beamforming matrix $\mathbf{W}^{\mathrm{H}} = \mathbf{A}^\dagger$ known as the Zero-Forcing (ZF) solution because $\mathbf{W}^{\mathrm{H}} \mathbf{A} = \mathbf{I}$ cancels all interfering sources. The ZF solution maximizes the Signal-to-Interference Ratio (SIR) but can produce the undesired noise enhancement in case the inversion of $\mathbf{A}$, $i.e.$, $\mathbf{A}^\dagger$ has large entries.

2. **Deterministic output error minimization**. The second approach in the receiver design minimizes the difference between the output of the receiver and the know transmitted data sequence

$$\mathbf{W}^{\mathrm{H}} = \underset{\mathbf{W}}{\operatorname{argmin}} \| \mathbf{W}^{\mathrm{H}} \mathbf{X} - \mathbf{S} \|_F^2 = \mathbf{S} \mathbf{X}^\dagger \ . \tag{2.11}$$

Using (2.10) we can write

$$\mathbf{W}^{\mathrm{H}} = \frac{1}{N} \mathbf{S} \mathbf{X}^{\mathrm{H}} (\frac{1}{N} \mathbf{X} \mathbf{X}^{\mathrm{H}})^{-1}$$

Under assumption that sources are mutually independent and independent from the noise and for long sequences of collected data (N), beamformer $\mathbf{W}$ converges to

$$\mathbf{W} = \hat{\mathbf{R}}_x^{-1} \mathbf{A} \ , \tag{2.12}$$

where $\hat{\mathbf{R}}_x^{-1} = \frac{1}{N} \mathbf{X} \mathbf{X}^{\mathrm{H}}$ is the sample covariance matrix and $\frac{1}{N} \mathbf{X} \mathbf{S}^{\mathrm{H}}$ converges to $\mathbf{A}$ for large N. The beamformer in (2.12) is known as the Linear Minimum Mean Square Error (LMMSE) or Wiener receiver. Rewriting (2.12) in terms of the noise variance σ^2 and mixing matrix $\mathbf{A}$ yields

$$\mathbf{W} = (\sigma^2 \mathbf{I} + \mathbf{A} \mathbf{A}^{\mathrm{H}})^{-1} \mathbf{A} \ .$$

This receiver minimizes the Signal-to-Interference and Noise Ratio (SINR) and produces the estimates of $\mathbf{S}$ with minimal deviation.

Here we have presented two kinds of beamformers. In case transmitted signal propagates through a multipath channel the mixing matrix $\mathbf{A}$ is replaced by matrix $\mathbf{H}$ that collects the samples of the multipath channel. In such cases space-time equalization is performed to harvest the energy of the signal of interest spread over the antenna array and multiple propagation paths.

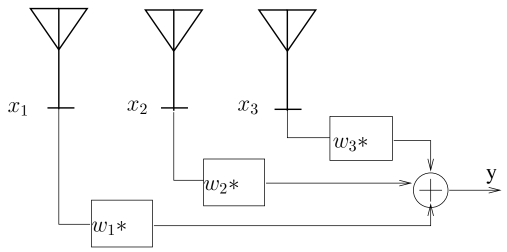


Figure 2.3: Spatial beamformer principle using an antenna array.

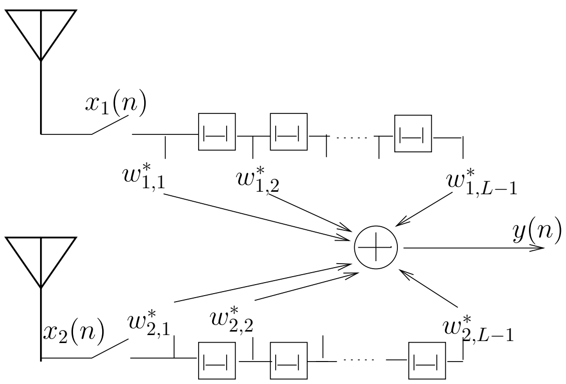


Figure 2.4: Space-time equalization involve weighting of time shifted versions of all antenna outputs.

2.4 Subspace techniques-Singular Value Decomposition

Consider a noiseless data model $\mathbf{X} = \mathbf{AS}$ as in (2.6) where the number of antennas M is larger than the number of sources d and the number of samples N exceeds M, $i.e.,\ d \leq M \leq N$. Matrix $\mathbf{X} : M \times N$ represents the stack of data samples received at M antennas during N consecutive time instants and can be represented as a linear combination of d independent sources (rows of $\mathbf{S}$), thus its rank is d. Similarly, $\mathbf{X}$ has d independent columns determined by the columns of $\mathbf{A}$. Performing an algebraic transformation over $\mathbf{X}$ called *Singular Value Decomposition-SVD* [13, 14, 15, 16] provides a basis for the row space as well as the column space of $\mathbf{X}$. In particular, let

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}} ,$$

where $\mathbf{U} : M \times d$ represents an orthonormal basis for column span of $\mathbf{X}$, $i.e.$, col($\mathbf{X}$) = col($\mathbf{A}$) = col($\mathbf{U}$). In the same manner the rows of $\mathbf{V}^{\mathrm{H}} : d \times N$ provide an orthonormal basis of the space spanned by the rows of $\mathbf{X}$ and $\mathbf{S}$, $i.e.$, row($\mathbf{X}$) = row($\mathbf{S}$) = row($\mathbf{V}^{\mathrm{H}}$). The orthonormality property of the bases $\mathbf{U}$ and $\mathbf{V}$ implies $\mathbf{U}^{\mathrm{H}}\mathbf{U} = \mathbf{I}_d$ and $\mathbf{V}^{\mathrm{H}}\mathbf{V} = \mathbf{I}_d$, where $\mathbf{I}_d$ is identity matrix of size d. Matrix $\mathbf{\Sigma} : d \times d$ is a diagonal matrix with singular values $[\sigma_1, \cdots, \sigma_d]$ on its diagonal. Singular values are positive real and are sorted in non-descending order, $i.e.$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_d > 0$.
In the presence of additive white Gaussian noise (AWGN) in the system the received data sequence becomes

$$\mathbf{X} = \mathbf{AS} + \mathbf{N}, \tag{2.13}$$

and its singular value decomposition yields

$$\mathbf{X} = [\mathbf{U}_s \quad \mathbf{U}_n] \begin{bmatrix} \mathbf{\Sigma}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_n \end{bmatrix} \begin{bmatrix} \mathbf{V}_s^{\mathrm{H}} \\ \mathbf{V}_n^{\mathrm{H}} \end{bmatrix} .$$

Note that at this point $\mathbf{X}$ has a full rank(M) due to the contribution of the noise. $[\mathbf{U}_s \quad \mathbf{U}_n]$ represents a basis of the column span of $\mathbf{X}$ where $\mathbf{U}_s$ and $\mathbf{U}_n$ are signal and noise subspace, respectively. $\mathbf{\Sigma}_s$ and $\mathbf{\Sigma}_n$ are diagonal matrices that collect signal and noise subspace singular values, respectively. In a similar fashion, $\mathbf{V}_s^{\mathrm{H}}$ and $\mathbf{V}_n^{\mathrm{H}}$ denote right signal and noise subspace. At this point we also define the *economy size SVD* that is written in terms of only the signal subspace components

$$\mathbf{X} = \mathbf{U}_s\mathbf{\Sigma}_s\mathbf{V}_s^{\mathrm{H}} .$$

The singular value decomposition is one of the widely used methods in blind source separation and detection algorithms. Based only on the knowledge of the received data matrix $\mathbf{X}$ makes possible the estimation of the transmitted data sequence. Using the knowledge of the signal subspaces and general properties of matrices $\mathbf{A}$ and $\mathbf{S}$ produce the estimates of the transmitted data sequence.

However, without exploiting additional properties the use of the SVD based blind estimation methods always induce the incapability of reconstructing the original ordering of the sources, $e.g.$, in case original sequence was $\mathbf{S} = [\mathbf{s}_1^{\mathrm{T}}\ \mathbf{s}_2^{\mathrm{T}}\ \mathbf{s}_3^{\mathrm{T}}]^{\mathrm{T}}$ an estimated sequences may result in a permuted user ordering, for example $\hat{\mathbf{S}} = [\mathbf{s}_2^{\mathrm{T}}\ \mathbf{s}_3^{\mathrm{T}}\ \mathbf{s}_1^{\mathrm{T}}]^{\mathrm{T}}$. Here $\mathbf{s}_i$ represents a $1 \times N$ row vector containing the transmitted data sequence of source i. The ordering ambiguity arise due to the commutativity of the factorization, $i.e.$,

$$\begin{aligned} \mathbf{AS} &= [\mathbf{a}_1 \quad \mathbf{a}_2] \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix} \\ &= [\mathbf{a}_2 \quad \mathbf{a}_1] \begin{bmatrix} \mathbf{s}_2 \\ \mathbf{s}_1 \end{bmatrix} \\ &= \textstyle\sum_{i=1}^{2} \mathbf{a}_i \mathbf{s}_i \ . \end{aligned}$$

Without the knowledge of specific properties on $\mathbf{a}_1$ versus $\mathbf{a}_2$ or $\mathbf{s}_1$ versus $\mathbf{s}_2$ it is not possible to identify the ordering of the signals.
This is a general problem for blind source estimation schemes based on SVD and constant modulus property of the signal. Note that this ambiguity can be solved by adding some training sequence. In Part I of the thesis we propose a scheme that is able to resolve this ambiguity without reducing the bandwidth of the signal.

2.5 Eigenvalue decomposition

The eigenvalue problem for a matrix $\mathbf{A}$ is

$$\mathbf{Ax} = \lambda \mathbf{x} \quad \Leftrightarrow \quad (\mathbf{A} - \lambda \mathbf{I})\mathbf{x} = \mathbf{0} \ .$$

Any λ that makes $\mathbf{A} - \lambda \mathbf{I}$ singular is called an eigenvalue and the corresponding $\mathbf{x}$ is the eigen vector. Stacking the eigenvectors horizontally yields $\mathbf{T}$ and stacking the eigenvalues on the diagonal produces $\mathbf{\Lambda}$ so that

$$\mathbf{AT} = \mathbf{T\Lambda} \ .$$

At this point we define the eigenvalue decomposition of a regular matrix $\mathbf{A}$

$$\mathbf{A} = \mathbf{T\Lambda T}^{-1} \quad , \mathbf{T} \text{ invertible, } \mathbf{\Lambda} \text{ diagonal}$$

2.6 ESPRIT algorithm

Here we present a general principle used in the Estimation of Signal Parameters via Rotational Invariance Techniques — ESPRIT [8, 7]. In particular, we focus on a uniform linear array (ULA) of M elements with equal interelement spacing Δ (see figure 2.1). Collecting antenna outputs into the data matrix $\mathbf{X} : M \times N$, as in Section 2.2.2 yields

$$\mathbf{X} = [\mathbf{a}(\theta_1) \cdots \mathbf{a}(\theta_d)] \begin{bmatrix} \mathbf{s}_1(1) \\ \vdots \\ \mathbf{s}_d(1) \end{bmatrix} = \mathbf{AS} \ , \tag{2.14}$$

where $\mathbf{s}_i, \ \ i = 1, \cdots, d$ is a $1 \times N$ vector of transmitted data and $\mathbf{a}_i$ denotes a $M \times 1$ array signature vector for direction of arrival θ_i

$$\mathbf{a}(\theta_i) = \begin{bmatrix} 1 \\ e^{j2\pi\Delta \sin(\theta_i)} \\ \vdots \\ e^{j2\pi(M-1)\Delta \sin(\theta_i)} \end{bmatrix} . \tag{2.15}$$

Let us define selection matrices $\mathbf{J}_1 : (M-1) \times M$ and $\mathbf{J}_2 : (M-1) \times M$ which select the first and the last $M-1$ rows, respectively

$$\mathbf{J}_1 = [\mathbf{I}_{M-1} \quad 0] \quad \text{and} \quad \mathbf{J}_2 = [0 \quad \mathbf{I}_{M-1}] \ .$$

Applying the selection matrices to $\mathbf{A}$ yields

$$\begin{array}{rcl} \mathbf{A}_1 & = & \mathbf{J}_1\mathbf{A} \\ \mathbf{A}_2 & = & \mathbf{J}_2\mathbf{A} \end{array} \tag{2.16}$$

where $\mathbf{A}_1$ and $\mathbf{A}_2$ possess the shift invariant property, $i.e.,$ the following relation holds

$$\mathbf{A}_2 = \mathbf{J}_1\mathbf{A}\mathbf{\Phi} = \mathbf{A}_1\mathbf{\Phi} \tag{2.17}$$

where $\mathbf{\Phi}$ is a diagonal $d \times d$ matrix

$$\mathbf{\Phi} = \begin{bmatrix} \phi_1 & & \\ & \ddots & \\ & & \phi_d \end{bmatrix} .$$

Here, $\phi_i = e^{j2\pi\Delta \sin\theta_i}, \ \ i = 1, \cdots, d$ and θ_i denotes the angle of arrival of the ith user.

Under assumption that the number of sources d is known we can compute the economy size SVD of $\mathbf{X}$

$$\mathbf{X} = \mathbf{U}_s\mathbf{\Sigma}_s\mathbf{V}_s^{\mathrm{H}} \ .$$

Note that $\mathbf{U}_s$ contains d columns that span the column space of $\mathbf{X}$ and $\mathbf{A}$, thus there must be a $d \times d$ invertible matrix $\mathbf{T}$ that maps one basis into the other

$$\mathbf{A} = \mathbf{U}_s\mathbf{T} \ . \tag{2.18}$$

Applying selection matrices $\mathbf{J}_1$ and $\mathbf{J}_2$ to $\mathbf{U}_s$ we obtain $\mathbf{U}_1 = \mathbf{J}_1\mathbf{U}_s$ and $\mathbf{U}_2 = \mathbf{J}_2\mathbf{U}_s$. Using these properties and (2.18) we can rewrite (2.16) as

$$\begin{aligned} \mathbf{A}_1 &= \mathbf{U}_1\mathbf{T} \\ \mathbf{A}_2 &= \mathbf{U}_2\mathbf{T} \end{aligned} \tag{2.19}$$

Using (2.17) we obtain

$$\mathbf{A}_1\mathbf{\Phi} = \mathbf{U}_2\mathbf{T} \quad \Leftrightarrow \quad \mathbf{U}_1\mathbf{T}\mathbf{\Phi} = \mathbf{U}_2\mathbf{T}\ . \tag{2.20}$$

If $\mathbf{U}_1$ is "tall", $i.e.$, $M \geq d$ and $\mathbf{A}$ is of full column rank we can find the left-inverse of $\mathbf{U}_1$ as $\mathbf{U}_1^{\dagger} = (\mathbf{U}_1^{\mathrm{H}}\mathbf{U}_1)^{-1}\mathbf{U}_1^{\mathrm{H}}$. Finally we can rewrite 2.20 as

$$\mathbf{U}_1^{\dagger}\mathbf{U}_2 = \mathbf{T}\mathbf{\Phi}\mathbf{T}^{-1}$$

where the matrix on the left hand side is known from the received data. The right hand side of the last equation is recognized as an eigenvalue equation since $\mathbf{\Phi}$ is diagonal. Finding the diagonal elements of $\mathbf{\Phi}$ we obtain ϕ_i for all $i = 1, \cdots, d$ and therefrom we get the estimates of the angle of arrivals θ_i.

2.7 MUSIC algorithm

MUSIC is the acronym of the MUltiple Signal Classification algorithm [17], a subspace method for parameter estimation that we present next.
The singular value decomposition of the received data sequence in white Gaussian noise (2.13) provides the signal and noise subspace $\mathbf{U}_s$ and $\mathbf{U}_n$, respectively. Note that the columns of $\mathbf{U}_s$ span the same subspace as the columns of $\mathbf{A}$. At the same time the noise subspace $\mathbf{U}_n$ is orthogonal to the subspace spanned by the columns of $\mathbf{A}$

$$\mathbf{U}_n^{\mathrm{H}}\mathbf{A} = \mathbf{0}\ . \tag{2.21}$$

This property together with the known structure of column vectors that create $\mathbf{A}$, $i.e.$, $\mathbf{a}(\theta_i),\ i = 1, \cdots, d$ (defined in (2.15)) can be exploited in the estimation of unknown parameters θ_i. In particular we search for the d lowest minima of the cost function

$$J_{MUSIC}(\theta) = \frac{\| \mathbf{U}_n^{\mathrm{H}}\mathbf{a}(\theta) \|^2}{\| \mathbf{a}(\theta) \|^2} = \frac{\mathbf{a}(\theta)^{\mathrm{H}}\mathbf{U}_n\mathbf{U}_n^{\mathrm{H}}\mathbf{a}(\theta)}{\mathbf{a}(\theta)^{\mathrm{H}}\mathbf{a}(\theta)}$$

by an exhaustive search over the range of $\theta = (-\pi, \pi]$.

2.8 Kronecker product

For two matrices $\mathbf{A}$ and $\mathbf{B}$ the Kronecker product, $\otimes$ is defined as

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{1,1}\mathbf{B} & a_{1,2}\mathbf{B} & \\ & \cdots & \\ & & a_{m,n}\mathbf{B} \end{bmatrix} . \tag{2.22}$$

A column wise Kronecker product is named *Khatri-Rao product*

$$\mathbf{A} \circ \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1, \mathbf{a}_2 \otimes \mathbf{b}_2, \cdots] . \tag{2.23}$$

Now we present several Kronecker product properties

$$\begin{array}{rcl} vec(\mathbf{a}\mathbf{b}^{\mathrm{H}}) & = & \mathbf{b}^{*} \otimes \mathbf{a} \\ (\mathbf{A}\mathbf{C}) \otimes (\mathbf{B}\mathbf{D}) & = & (\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) \\ (\mathbf{a}^{\mathrm{H}}\mathbf{c})(\mathbf{b}^{\mathrm{H}}\mathbf{d}) & = & (\mathbf{a} \otimes \mathbf{b})^{\mathrm{H}}(\mathbf{c} \otimes \mathbf{d}) \\ vec(\mathbf{A}\mathbf{B}\mathbf{C}) & = & (\mathbf{C}^{\mathrm{T}} \otimes \mathbf{A})vec(\mathbf{B}) \\ vec(\mathbf{A}\mathrm{diag}(b)\mathbf{C}) & = & (\mathbf{C}^{\mathrm{T}} \circ \mathbf{A})\mathbf{b} \end{array} \tag{2.24}$$

where $^{\mathrm{H}}$ denotes a complex conjugate, $^{\mathrm{T}}$ transpose, * conjugate operation and *vec* stacks the columns of a matrix into a vector.

Part I

Known Modulus Algorithms

Chapter 3

Asynchronous Mobile Ad Hoc Networks

> ad hoc: formed or used for specific or immediate problems or needs
> network: a group of devices connected to allow users to communicate and share information and resources

A wireless ad hoc network consists of a set of wireless devices (nodes) within a particular geographical area that communicate with each other to exchange information. The connection between the nodes is created at the instantaneous need of users and without the presence of a central access point. An appropriate data transmission scheme in a highly dynamic environment is obtained if packets are of finite length and are transmitted in a completely unscheduled manner. To that extent each receiver node needs to recognize any ongoing transmission, detect the beginning of the data packet and estimate its content.

In existing systems, in which all users share the same wireless channel, *packet collision* occurs in case a number of users are simultaneously active. As a result, the content of these packets is destroyed and a *retransmission* needs to be performed. Consequently, the capacity and the throughput of the system may be significantly reduced, in particular within networks that accommodate a large number of users.

In this chapter several basic problems that arise in wireless Mobile Ad hoc Networks (MANET's), such as *packet collision, retransmission, hidden-exposed terminal problem and multiuser co-channel interference suppresion* are described. In the subsequent chapters we will propose a solution to these issues by means of source separation techniques and study their performance.

3.1 Packet collision problem

As mentioned, in wireless networks with unscheduled packet transmissions, in which all users share the same channel, simultaneous transmission by several users leads to a *packet collision* which destroys all data content. In the the existing systems a packet collision problem is treated by several protocols at the Multiple Access Control (MAC) layer. The first such protocol - ALOHA [4] - was introduced by Abramson from the University of Hawaii. In case a user has information to send, it simply broadcasts it and waits for an acknowledgment of reception within a predefined time interval. If a collision occurs, the data packet is destroyed and cannot reach the destinated user. Accordingly, the transmitter does not receive acknowledgment of successful data reception. In such a case, data packets originating from users involved in the collision will be retransmitted after a randomly chosen time offset thus reducing the probability of repeated collisions. As an improvement a *Slotted ALOHA* is described in [18]. Users are allowed to transmit only in predefined time slots so that in case a collision occurs only one, rather than two or more partially overlapping slots is affected. This leads to an increased channel efficiency and system's throughput.

As an alternative to ALOHA, users having data to broadcast can, prior to any transmission, check whether the channel is occupied and in that case postpone the transmission for some later time instant. This approach is named Carrier Sense Multiple Access (CSMA) protocol [5]. As ALOHA, CSMA mode can also be scheduled that leads to a *Slotted CSMA* reducing the interval affected by collided packets.

All mentioned protocols do not resolve the packet collision problem but rather improve the probability of a successful packet reception for a presumed data traffic density and a given number of users. Note that slotted versions of MAC protocols demand some kind of centralized time control that is inappropriate for applications in ad hoc networking.

An ad hoc network such as Bluetooth is formed by having one device act as master (the other are slaves). All communication is done via the master. If a slave wants to transmit it first sends a "request to send" at an appropriate time announced by the master. The master will collect the request and assigns a "clear to send" to each of the slaves in turn which made a request. The formation of such a network carries a significant overhead (may take several seconds) and is not flexible. Thus it is less appropriate for highly dynamic environments.

An alternative way to solve the packet collision problem is to transfer a part of the system's complexity from the MAC to the physical (PHY) layer. By introducing an antenna array at the receiver side and exploiting the spatial diversity (SDMA), the separation of simultaneously active multiple co-channel signals becomes feasible. In that manner packet capture capability of the system is improved and retransmission is no longer required. Accordingly, the data throughput of the system is significantly increased.

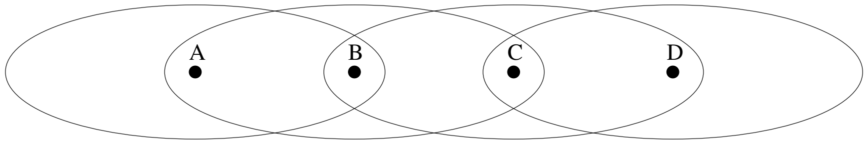


Figure 3.1: Hidden exposed terminal problem.

An example of a *blind* deterministic scheme that is able to resolve packet conflict situations is described in Section 3.3. This method performs separation of all co-channel constant modulus signals by means of an antenna array at the receiver side. This algorithm will further be extended to enable the separation of only the desired user which is presented in Chapter 4 and Chapter 5 for the synchronous and the asynchronous case, respectively.

3.2 Hidden-exposed terminal problem

A hidden-exposed terminal problem [19, 20] may arise in networks in which users share the same wireless channel. Assume a distribution of users within an area as presented in figure 3.1, where each node is in the range of both its neighboring nodes.

1. **Hidden terminal problem**: Suppose A transmits a data packet to B (see figure 3.1). This data transfer cannot be perceived by C, as it is out of the range of A. In case C starts broadcasting a message while a hidden terminal A is already active, a collision occurs at terminal B. Such a situation where a terminal (C) is in the range of a receiver (B) but out of the range of the transmiter (A) is defined as a *hidden terminal problem*.

2. **An Exposed terminal problem** arises in cases in which a terminal is in the vicinity of the transmitting but distant from the receiving node. Imagine (figure 3.1) that node C transmits a packet to node D. During such a data transfer B remains silent even if its transmission to A would not cause any collision.

The design of efficient Multiple Access (MAC) layer protocols may lead to reduced number of collisions in case hidden and/or exposed terminals are present in the network. A *request to send-clear to send* (RTS-CTS) along with busy-tone is an example of a protocol that can mitigate collisions [21], but such protocols are vulnerable to interference from other services [22]. In addition, the collision problem remains unresolved in cases where multiple RTS packets simultaneously target the same node.

3.3 Constant Modulus Algorithms

The Constant Modulus Algorithm (CMA) is a classical example of a blind source separation method. Assume a single user narrowband system (see Chapter 2) in white Gaussian noise where an antenna array is deployed at the receiver side. The data model becomes:

$$\mathbf{x}(k) = \mathbf{a}s(k) + \mathbf{n}(k) \ ,$$

where $\mathbf{x}(k) : [M \times 1]$ is the vector of samples at the output of the M antennas at time instant k, $\mathbf{a} : [M \times 1]$ is the array signature vector and $s(k)$ denotes the transmitted symbol at time instant k where $k = 1, \cdots, N$. $\mathbf{n}(k)$ is a $M \times 1$ vector of white noise samples that are added at the receive antennas. A broadcast sequence $s(k)$ possesses a constant modulus (CM) property, $i.e.,$ its amplitude does not change in time, if $|s(k)| = 1 \ \forall k$. This feature can be used to determine a weight vector $\mathbf{w}$ such that the output of the beamformer is equal to the transmitted CM sequence, $y(k) = \hat{s}(k) = \mathbf{w}^{\mathrm{H}}\mathbf{x}(k)$.

An option to solve the beamforming problem is in the implementation of *adaptive* algorithms that process sample by sample of the received data in the same order as they arrive to the receiver. Adaptive algorithms start from a given initial weight vector $\mathbf{w}_0$ and are updated by each new sample. In order to improve the beamforming vector a suitable cost function is chosen that should be minimized by means of gradient-descent techniques. An example of such a function is

$$J(\mathbf{w}) = E(|y(k)|^2 - 1)^2 \ ,$$

that quantifies the deviation of the beamformer's output from the desired constant modulus. Here, $y(k)$ represents the output of the beamformer at time instant k and $E(\cdot)$ denotes the expectation operator. A simple approach is to iteratively improve vector $\mathbf{w}$ in the direction of the negative gradient of the function $J(\mathbf{w})$,

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \mu \nabla_{\mathbf{w}}(J(\mathbf{w}^{(k)}))$$

where μ represents the step size. Choosing a small μ makes the convergence of the algorithm slow but stable while selecting a larger step sizes results in a faster but more erratic convergence or possibly divergence. The gradient of the cost function in the case of CMA becomes

$$\begin{aligned}\nabla_{(\mathbf{w})} J &= 2E\{(|y(k)|^2 - 1) \cdot \nabla_w(\mathbf{w}^{\mathrm{H}}\mathbf{x}(k)\mathbf{x}(k)^{\mathrm{H}}\mathbf{w})\} \\ &= 4E\{(|y(k)|^2 - 1) \cdot \mathbf{x}(k)\mathbf{x}(k)^{\mathrm{H}}\mathbf{w}\} \\ &= 4E\{(|y(k)|^2 - 1) \cdot y(k)\mathbf{x}(k)\} \ ,\end{aligned}$$

that after absorbing 4 into μ leads to an iterative rule

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \mu\mathbf{x}(k)\mathbf{x}(k)^{\mathrm{H}}\mathbf{w}^{(k)}(|\mathbf{w}^{(k)H}\mathbf{x}(k)|^2 - 1) \ .$$

This algorithm is called the Constant Modulus Algorithm (CMA) [23, 24]. The drawback of an iterative CMA algorithm is that it may need large number of samples to obtain full convergence. Note that the estimated beamformer $\mathbf{w}$ does not provide a unique solution as any other beamformer $\alpha\mathbf{w}$ with $|\alpha| = 1$ also satisfies the CM condition, $|\hat{s}(k)|^2 = |\mathbf{w}^{\mathrm{H}}\mathbf{x}(k)|^2 = |\alpha\mathbf{w}^{\mathrm{H}}\mathbf{x}(k)|^2 = 1$. This phase ambiguity cannot be resolved due to the nature of the algorithm but it can be overcome via a differential modulation scheme. Accordingly, only one training symbol is needed to recover the proper constellation of the transmitted sequence.

In case several CM signals are simultaneously broadcast, the iterative CM algorithm is able to estimate only one of them. It remains unclear whether the user of interest has been estimated as the separation is based on a general (CM) property common for all users. In such a case the solution would be to estimate all sources and subsequently choose the one of interest. This can be achieved using the *CM array* that, at first, extracts the strongest CM signal and subtracts it from the received linear combination. This process is repeated until all CM signals are detected. A drawback of this method is a possible slow convergence for each user. The same source may be estimated twice in case its initial beamformer did not converge sufficiently, thus the signal was not completely subtracted from the received mixture. In [25] the authors propose a multiuser constant modulus algorithm in which the beamformers of all users are estimated simultaneously in an iterative fashion. Clearly, an improperly estimated beamformer of a single user induces the failure in separation of all other sources.

The Algebraic Constant Modulus Algorithm (ACMA), proposed in [26, 27, 28], solves the source separation problem of multiple co-channel CM signals in a non-iterative fashion. In contrast to iterative algorithms, ACMA is able to extract all CM sources from the superposition of the signals received at an antenna array at the receiver side. Moreover, it is a block algorithm that performs well already for a very small number of samples. In the subsequent section we give an overview of ACMA algorithm following the presentation in [26], because the Known Modulus Algorithm presented in this thesis is based on similar principles.

3.4 Algebraic Constant Modulus Algorithm – ACMA

A narrowband system with multiple CM signals sharing the same frequency band is assumed. The sequence received at the antenna array becomes

$$\mathbf{X} = \mathbf{A}\mathbf{S} \; ,$$

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N] : [M \times N]$, $\mathbf{A} : [M \times d]$ and $\mathbf{S} : [d \times N]$ represent received data samples at the antenna array, the array response matrix and the data symbol matrix, respectively, see also (2.6). M denotes the number of receive antennas, N the number of samples (data symbols) and d the number of users. Without loss of generality it is assumed that each transmitted symbol satisfies $|s(k)|^2 = 1$ as any scaling can be incorporated in $\mathbf{A}$. For each user $i \in \{1, \cdots, d\}$ a weight vector $\mathbf{w}_i$

needs to be estimated so that its output satisfies a unit modulus criterion, or

$$\mathbf{w}_i^{\mathrm{H}}\mathbf{X} = \mathbf{s}_i \;\text{ such that }\; |s_i(k)|^2 = 1 \;,\; k = 1, \cdots, N \;. \tag{3.1}$$

The constant modulus condition for each estimated sample is equivalent to

$$|s_i(k)|^2 = \mathbf{w}_i^{\mathrm{H}}\mathbf{x}_k\mathbf{x}_k^{\mathrm{H}}\mathbf{w}_i = 1, \quad k = 1, \cdots, N. \tag{3.2}$$

Exploiting the properties of Kronecker products (2.24) the middle term of (3.2) can be written as

$$\mathbf{w}_i^{\mathrm{H}}\mathbf{x}_k\mathbf{x}_k^{\mathrm{H}}\mathbf{w}_i = (\bar{\mathbf{x}}_k \otimes \mathbf{x}_k)^{\mathrm{H}}(\bar{\mathbf{w}}_i \otimes \mathbf{w}_i) \;.$$

Define

$$\mathbf{P} := \begin{bmatrix} (\bar{\mathbf{x}}_1 \otimes \mathbf{x}_1)^{\mathrm{H}} \\ \vdots \\ (\bar{\mathbf{x}}_N \otimes \mathbf{x}_N)^{\mathrm{H}} \end{bmatrix}$$

of size $\mathbf{P} : N \times d^2$ and

$$\mathbf{y} = \bar{\mathbf{w}}_i \otimes \mathbf{w}_i \;.$$

In this fashion, the problem defined in (3.2) can be written as a linear matrix equation

$$\mathbf{P}\mathbf{y} = \mathbf{1} \;\text{ subject to condition }\; \mathbf{y} = \bar{\mathbf{w}}_i \otimes \mathbf{w}_i \;.$$

The solution to $\mathbf{P}\mathbf{y} = \mathbf{1}$ is of the form $\mathbf{y} = \mathbf{y}_0 + \alpha_1\mathbf{y}_1 + \cdots + \alpha_l\mathbf{y}_l$ that represents an affine space [26]. Here $\mathbf{y}_0$ corresponds to a particular solution while $\{\mathbf{y}_1, \cdots, \mathbf{y}_l\}$ is the basis of a kernel of $\mathbf{P}$. Alternatively, the problem $\mathbf{P}\mathbf{y} = \mathbf{1}$ can be transformed to a system of equations with a solution in a linear space. In that case the basis vectors can be found simply by performing a singular value decomposition. For this purpose a unitary matrix $\mathbf{Q} : N \times N$ is chosen such that

$$\mathbf{Q}\;\mathbf{1} = \sqrt{N}\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \;.$$

Applying $\mathbf{Q}$ to $[\mathbf{1}\;\;\mathbf{P}]$ yields

$$\mathbf{Q}[\mathbf{1}\;\;\mathbf{P}] = \sqrt{N}\begin{bmatrix} 1 & \mathbf{p}^{\mathrm{H}} \\ \mathbf{0} & \mathbf{G} \end{bmatrix} \;.$$

A Discrete Fourier Transform (DFT) matrix and a Householder transformation [26, 13] are examples of matrices that satisfy the condition set on $\mathbf{Q}$. In addition, a QR factorization of $[\mathbf{1}\;\;\mathbf{P}]$ provides the same result. At this point the following holds

$$\begin{aligned} \mathbf{P}\mathbf{y} = \mathbf{1} \quad &\Leftrightarrow \quad \mathbf{Q}[\mathbf{1}\;\;\mathbf{P}]\begin{bmatrix} -1 \\ \mathbf{y} \end{bmatrix} = \mathbf{0} \\ &\Leftrightarrow \quad \begin{cases} \mathbf{p}^{\mathrm{H}}\mathbf{y} = 1 \\ \mathbf{G}\mathbf{y} = \mathbf{0} \end{cases} \;. \end{aligned} \tag{3.3}$$

The first condition ($\mathbf{p}^{\mathrm{H}}\mathbf{y} = 1$) is just a scaling and can always be satisfied (unless $\mathbf{y} = 0$ or $\mathbf{p}^{\mathrm{H}}\mathbf{y} = 0$).

The set of solutions $\{\mathbf{y}_i\}$ of the linear matrix equation $\mathbf{G}\mathbf{y} = \mathbf{0}$ can efficiently be found by means of a singular value decomposition (SVD) of $\mathbf{G}$. The solution vectors $\mathbf{y}_i$ are the right singular vectors corresponding to the d singular values of $\mathbf{G}$ that are equal to 0. On the other hand, there are d independent solutions satisfying the condition $\bar{\mathbf{w}}_i \otimes \mathbf{w}_i$ for $i \in \{1, \cdots, d\}$. Note that any linear combination of these solutions is also a solution to $\mathbf{G}\mathbf{y} = \mathbf{0}$, $i.e.$,

$$\mathbf{y} = \lambda_1(\bar{\mathbf{w}}_1 \otimes \mathbf{w}_1^{\mathrm{H}}) + \cdots + \lambda_d(\bar{\mathbf{w}}_d \otimes \mathbf{w}_d^{\mathrm{H}}) \ .$$

Thus, solutions $\mathbf{y}_i$ for $i \in \{1, \cdots, d\}$ do not necessarily correspond to the structured solutions $\bar{\mathbf{w}}_i \otimes \mathbf{w}_i$ but rather to an arbitrary linear combination thereof.

The second step of ACMA consists of a *decoupling*, $i.e.$, determination of the structured basis $\{\bar{\mathbf{w}}_1 \otimes \mathbf{w}_1, \cdots, \bar{\mathbf{w}}_d \otimes \mathbf{w}_d\}$ exploiting the knowledge of the set of arbitrary solution vectors$\{\mathbf{y}_1, \cdots, \mathbf{y}_d\}$ that both span the same linear subspace. Note that

$$\bar{\mathbf{w}}_i \otimes \mathbf{w}_i := vec(\mathbf{w}_i \mathbf{w}_i^{\mathrm{H}}) \ ,$$

where 'vec' transforms a matrix into a vector by stacking the columns. Define also $\mathbf{y}_i$ by $vec(\mathbf{Y}_i) = \mathbf{y}_i$. According to this definition,

$$\mathbf{y}_i = \lambda_{i1}(\bar{\mathbf{w}}_1 \otimes \mathbf{w}_1) + \cdots + \lambda_{id}(\bar{\mathbf{w}}_d \otimes \mathbf{w}_d)$$

can be written as the sum of rank one matrices

$$\mathbf{Y}_i = \lambda_{i1}(\mathbf{w}_1 \mathbf{w}_1^{\mathrm{H}}) + \cdots + \lambda_{id}(\mathbf{w}_d \mathbf{w}_d^{\mathrm{H}}) \ .$$

Taking into account all $\mathbf{Y}_i$ matrices the decoupling process needs to solve the following system of matrix equations

$$\left\{ \begin{array}{c} \mathbf{Y}_1 = \lambda_{11}(\bar{\mathbf{w}}_1 \otimes \mathbf{w}_1^{\mathrm{H}}) + \cdots + \lambda_{id}(\bar{\mathbf{w}}_d \otimes \mathbf{w}_d^{\mathrm{H}}) \\ \vdots \\ \mathbf{Y}_d = \lambda_{d1}(\bar{\mathbf{w}}_1 \otimes \mathbf{w}_1^{\mathrm{H}}) + \cdots + \lambda_{dd}(\bar{\mathbf{w}_d} \otimes \mathbf{w}_d^{\mathrm{H}}) \end{array} \right. ,$$

or equivalently

$$\left\{ \begin{array}{c} \mathbf{Y}_1 = \mathbf{W}\mathbf{\Lambda}_1\mathbf{W}^{\mathrm{H}} \\ \vdots \\ \mathbf{Y}_d = \mathbf{W}\mathbf{\Lambda}_d\mathbf{W}^{\mathrm{H}} \end{array} \right. ,$$

where $\mathbf{\Lambda}_i = diag[\lambda_{11}, \cdots, \lambda_{1d}]$, $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_d]$ and $diag$ transforms a vector into a diagonal matrix.

[1] $\mathbf{y} = 0$ with $\mathbf{y} = \bar{\mathbf{w}}_i \otimes \mathbf{w}_i$ leads to $\mathbf{w} = 0$ which is not a relevant solution anyway, whereas $\mathbf{p}^{\mathrm{H}}\mathbf{y} = 0$ with $\mathbf{y} = \bar{\mathbf{w}}_i \otimes \mathbf{w}_i$ leads to $\mathbf{w}^{\mathrm{H}}\mathbf{X} = 0$ [26] which is also not relevant.

This is a *joint diagonalization problem* as each $\mathbf{Y}_i$ is diagonalized into $\mathbf{\Lambda}_i$ by the same matrix $\mathbf{W}$ [26, 28]. A simple solution in that case, for invertible $\mathbf{Y}_2$ can be found by eigenvalue decomposition of $\mathbf{Y}_1\mathbf{Y}_2^{-1}$ $i.e.$,

$$\mathbf{Y}_1\mathbf{Y}_2^{-1} = \mathbf{W}(\mathbf{\Lambda}_1\mathbf{\Lambda}_2^{-1})\mathbf{W}^{-1} \ .$$

This approach may be applicable in a noiseless scenario. Each one of the $\mathbf{Y}_i$, $i = \{1, \cdots, d\}$ matrices can be diagonalized by the same $\mathbf{W}$ and all of them should be taken into account in the noisy case in order to obtain a better estimate. A solution is provided by the *joint diagonalization scheme* described in [26, 28]. In this way, the potentially problematic inversion step is avoided.

The condition $\mathbf{p}^{\mathrm{H}}\mathbf{y} = 1$ that arose after the unitary transformation $\mathbf{Q}$ has been performed over $[\mathbf{1} \;\; \mathbf{P}]$ (see 3.3) will determine a proper scaling of the final solution as

$$\begin{aligned}
\mathbf{p}^{\mathrm{H}}\mathbf{y} &= (\tfrac{1}{N}\textstyle\sum_{k=1}^{N} \bar{\mathbf{x}}_k \otimes \mathbf{x}_k)^{\mathrm{H}}\mathbf{y} \\
&= \mathbf{w}^{\mathrm{H}}(\tfrac{1}{N}\textstyle\sum_{k=1}^{N} \mathbf{x}_k\mathbf{x}_k^{\mathrm{H}})\mathbf{w} \\
&= \tfrac{1}{N}\textstyle\sum_{k=1}^{N} \mathbf{w}^{\mathrm{H}}\mathbf{x}_k\mathbf{x}_k^{\mathrm{H}}\mathbf{w} \\
&= \tfrac{1}{N}\textstyle\sum_{k=1}^{N} |s(k)|^2 = 1
\end{aligned}$$

In the derivation of the ACMA we assumed that exactly d independent sources of equation $\mathbf{P}\mathbf{y} = \mathbf{1}$ exist. This assumption does not hold in cases where the number of antennas at the receiver is larger than the number of active users, $M > d$. This problem is solved by means of a *dimension reduction*. Instead of working directly on a rank deficient matrix $\mathbf{X}$, ACMA is applied to the d strongest vectors ($\mathbf{V}_s^{\mathrm{H}}$) in the right signal subspace of matrix $\mathbf{X}$ (for the singular value decomposition see Section 2.4). In this manner we make sure that precisely d beamforming solutions are found.

This finishes the discussion of [26].

3.5 Superimposed training

A fundamental drawback of constant modulus algorithms is the *permutation ambiguity*, $i.e.$, the inability to distinguish which is the user of interest in case several CM sources are simultaneously active. Thus it is possible that the wrong user is retrieved. This is avoided by methods that extract all CM signals and subsequently find the one of the interest based on the data content. Examples are the iterative multiuser CMA [25] or the block structured ACMA [26, 27, 28]. These are considered to be inefficient in case only a single user needs to be estimated. On the other hand, introducing a training sequence would lead to a reduced bandwidth efficiency due to a possibly large overhead.

A so called *superimposed training* scheme represents a possible way to resolve the permutation ambiguity problem of CM signals and is able to extract the user of interest without reduction in bandwidth efficiency. Basically, a known sequence (c_k) is added, multiplied or otherwise imposed on the desired CM source prior to any transmission. The known sequence can vary the phase or the modulus of each data symbol and gives it a known feature without increasing the data rate. At the receiver side, the knowledge of the superimposed training is used to extract the user of interest. This idea was initially introduced by Treichler and Larimore in 1985 [29] where the CM algorithm [23] was extended to signals with known modulus variations. Recently, it has been rediscovered and drawn some new attention so we present an overview of the proposed methods in this context.

In [30] Orozco-Lugo et al. propose to superimpose a known periodically time-varying modulation sequence $c(k)$ over a constant modulus signal $s(k)$ so that the modulated signal becomes

$$z(k) = c(k)s(k) \ .$$

In [30] multiple users and an instantaneous channel model are considered. With an antenna array of M sensors the received signal becomes

$$\mathbf{x}(k) = \mathbf{A}\mathbf{z}(k) + \mathbf{n}(k) \ ,$$

where $\mathbf{z}(k) : d \times 1$ stands for the stack of transmitted samples of d users and $\mathbf{n}(k) : M \times 1$ represent the additive noise. A beamformer vector $\mathbf{w}$ applied to the received data vector $\mathbf{x}(k)$ provides an estimate of the transmitted data sequence at time instant k as

$$y(k) = \hat{s}(k) = \mathbf{w}^{\mathrm{H}}\mathbf{x}(k) \ .$$

The coefficients of the spatial filter $\mathbf{w}$ are computed in an iterative fashion such that the cost function

$$J(\mathbf{w}) = E\{[|y(k)|^2 - c^2(k)]^2\}$$

is minimized.

A drawback of this scheme is that the proposed cost function $J(\mathbf{w})$ is not convex so that the algorithm may be captured in a local minimum. To avoid unwanted solutions, restrictions on the constellation as well as on the modulation sequences have to be imposed. In the asynchronous multiple user case these restrictions need to be satisfied for all mutual code offsets.

The idea from [30] is extended in [31] for a convolutive channel case. The authors propose to use an equalizer with a time span not shorter than the period of the modulating sequences. In this manner, searching for the equalizer with the best performance indirectly solves the synchronization problem. Nevertheless, the restrictions imposed on the choice of the modulating sequences remain.

In [32, 33] Tugnait et al. propose a superimposed training scheme for channel estimation which exploits the first order statistics of the received data sequence. A perfect synchronization between the superimposed training at the transmitter and the known template at the receiver is assumed. A single user case is considered with one transmit and multiple receive antennas. The transmitted sequence at time instant k :

$$z(k) = s(k) + c(k) \ , \tag{3.4}$$

where $s(k)$ and $c(k)$ represent the information and superimposed training sequence, respectively. The information sequence is chosen to possess a constant modulus $E\{|s(k)|^2\} = 1$ and is zero mean, $i.e.$, $E\{s(k)\} = 0$. In addition, the training sequence is periodic with period P, $i.e.$,

$$c(k) = c(k+P) \ . \tag{3.5}$$

After propagation through a convolutive multipath channel with impulse response $\mathbf{h}(l)$ ($l = 0, \cdots, L$ with L the channel length) the received sequence at the antenna array becomes

$$\mathbf{y}(k) = \mathbf{x}(k) + \mathbf{n}(k) = \sum_{l=0}^{L} \mathbf{h}(l)s(k-l) + \mathbf{n}(k)$$

with L being the order of the channel. The expected value of the received sequence

$$\begin{aligned} E\{\mathbf{y}(k)\} &= E\{\textstyle\sum_{l=0}^{L} \mathbf{h}(l)s(k-l) + \mathbf{n}(k)\} \\ &= \textstyle\sum_{l=0}^{L} \mathbf{h}(l)c(k-l) \end{aligned} \tag{3.6}$$

hence this operation completely removes the influence of the information sequence. $E\{\mathbf{y}(k)\}$ is obtained by averaging of the received data over a number of code periods P (see (3.5)). As described in [33] channel impulse response can be estimated from (3.6) in the least square sense.

A synchronization algorithm of the proposed scheme is further treated in [34]. It is based on an exhaustive model fitting for all possible code shifts in the range $[0, \cdots, P]$. A cost function is established and is minimized in case a proper code offset is deployed.

Using the same “induced training” structure as in 3.4, periodicity of the induced training sequence and the property that the expectation of the received data completely removes transmitted data sequence as in (3.6), the authors in [35] derive a channel estimation scheme for a single input single output antenna system. To achieve the synchronization the authors exploit the periodicity of the training sequence and perform an exhaustive search for the best possible match of the code to the received data sequence. Knowledge of the channel and the information of the packet offset makes possible further symbol estimation of the user of interest.

From a source separation point of view, amplitude modulation is just one way to mark users of interest, and several other techniques could play a role. Stochastic techniques such as “transmitter induced cyclostationarity”, initially derived for single user blind equalization [36, 37, 38] have recently been extended to multi-user convolutive channels [39] and OFDM [40]. A deterministic version of such a technique, using phase modulation codes, was proposed for source separation in MANETs [41].

The method in [41] assumes that each user possesses a single transmit antenna, while an antenna array is employed at the receiver side. For source separation purposes a *superimposed training* is inserted over a *real valued* data sequence $s_i(k)$. The modulation code is a polynomial phase sequence (PPS) modeled as

$$c_i(k) = e^{j[2\pi f_i k^2 + 2\pi\alpha_i k]}$$

where f_i and α_i represent user specific code parameters. The transmitted signal is obtained as instantaneous product of the data and the coding sequence $z_i(k) = s_i(k)c_i(k)$. The code sequence $c_i(k)$ exhibits cyclostationary properties to be exploited in the process of source separation in multiuser environments over time-dispersive channels. It is an extension of the method proposed in [42] that exploits a chirp sequence $c_i(k) = e^{j2\pi f_i k^2}$ to introduce cyclostationarity. The latter approach is able to resolve the inter symbol interference (ISI) problem in a single user, time dispersive channel case.

Multiplying the received sequence by the complex conjugate code of the user of interest, the contribution of the induced training is removed. Note that this is valid only in case of perfect synchronization. The equalization of the channel is achieved by forcing its output to reestablish the real value property of the transmitted signal. Synchronization is performed by an exhaustive search over all possible code offsets at the receiver side. Finally, the optimal equalizer is chosen from a set of possible solutions.

To summarize, we give an overview of the possible approaches in the superimposed training schemes

1. **Addition** of the training $c(k)$ over the data symbols $s(k)$ produces

   $$z(k) = s(k) + c(k) \quad \text{subject to} \quad E(s) = 0$$

   where $k = 1, \cdots, N$ is the sample index and N is the length of the data packet. Note that this approach demands a large amount of collected data to satisfy the expectation condition.

2. **Multiplication** of the training $c(k)$ with data symbols $s(k)$ gives

   $$z(k) = s(k)c(k) \quad \begin{cases} s \in \mathcal{R}, & \textit{c(k)} \text{ known phase} \\ |s| = 1, & \textit{c(k)} \text{ known amplitude} \end{cases}$$

The algorithms presented in this section provide the beamformer and channel estimates for instantaneous and convolutive channel cases, respectively. These estimates facilitate subsequent detection of data. Note that in the literature the offset of the received data packet with respect to the beginning of the analysis window is either considered to be known or it is estimated by an exhaustive search for the best possible match of the known code and the received signal. Thus the synchronization part of user separation algorithms for ad hoc networks is neglected in favor of solving the beamforming or channel estimation problem. For this reason, in Part I of this thesis, we focus our attention to the synchronization problem. Synchronization can be based on a training sequence known both at the transmitter and the receiver or only on the known properties of the transmitted signal, $i.e.$, blindly. Regarding the way the beginning of the data packet is obtained we distinguish search methods and block methods. The first scans for all possible solutions and chooses the best outcome while the latter instantaneously provides the best estimate within the observed block of data.

The lack of block oriented synchronization algorithms in packet oriented ad hoc networks was one of the main motivations for our research. We have derived a joint synchronization and beamformer estimation scheme that acts on the block of received data and provides an instantaneous solution.

In the next chapter we propose a data model for a packet oriented MANET under a narrowband assumption. Initially, in Chapter 4 a beamformer estimator for the synchronous case is derived and its performance analyzed. Subsequently, in Chapter 5 we propose a block method for joint synchronization and beamformer estimation.

Chapter 4

Known ModulusAlgorithm – Data Model

In this chapter we consider a multi-user asynchronous ad hoc wireless communication network where users transmit data packets randomly. In the previous chapter we have seen that the risk on the packet collision represents a great problem in ad hoc networks with an increased number of uncoordinated users. These collisions cause the retransmission of data packets reducing the system's efficiency and capacity. It followed that it is relevant to study methods that solve these issues. To this end, in this chapter we will consider a technique in the class of Known Modulus Algorithms.

The system is organized as depicted in figure 4.1. Each user in the system exploits a single transmit antenna. Information to be broadcasted possesses a constant modulus property. Prior to any transmission, small known modulus variations ("color code") are induced over the data sequence at the symbol level as presented in figure 1.4. At the receiver side an antenna array is employed that, together with the knowledge of the user specific code, enables the separation of the user of interest from undesired multiuser interference and additive noise. We set the design criteria in our system to

1. short data packet size
2. simple MAC layer design with no centralized unit, $i.e.,$ ad hoc scenario
3. narrow band

The presented KMA requires neither slot synchronization nor any coordination among transmitters, which makes its application in an uncontrolled environment such as wireless LAN (WLAN) or mobile ad hoc networks (MANETs) particularly attractive. The KMA was introduced in [43, 44, 45] where a small known modulus varia-

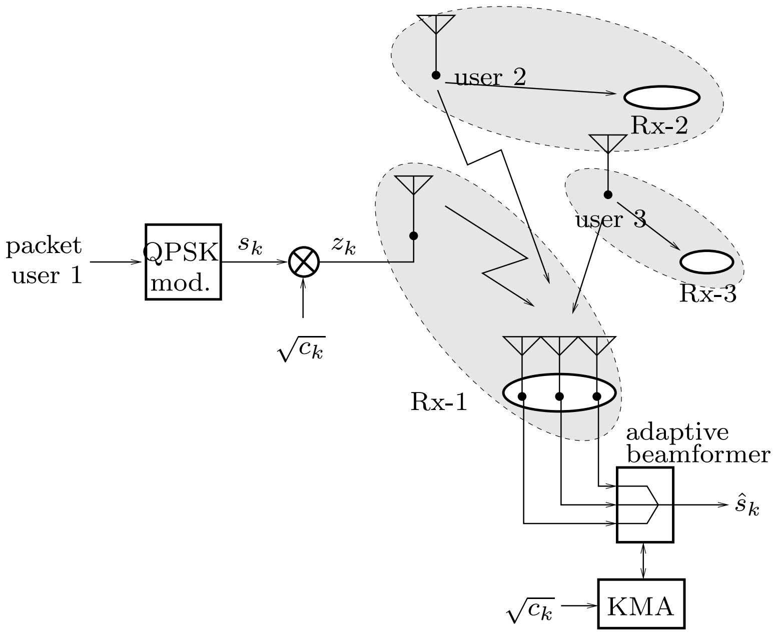


Figure 4.1: Wireless ad hoc communication scenario. $\sqrt{c_k}$ is a known modulus variation used to recognize user-1.

tion, $i.e.,$ the 'color code' is inserted over the CM signal. The code is considered to be known at the receiver side. The antenna array and the known code allow estimation of the beamformer that will select precisely the user of interest.

In contrast to Constant Modulus (CM) algorithms the proposed KMA scheme is able to separate exactly the user of interest that significantly reduces the system's complexity. Users are allowed to broadcast data packets in a completely unscheduled fashion. Further, the size of the packets of the user of interest is assumed to be known at the receiver side, as well as the code of the desired user. The proposed Known Modulus Algorithm performs source separation scheme on blocks of data samples collected within an analysis window.

The results in this chapter were presented at conferences [43, 44, 46, 47] and as a journal paper [45].

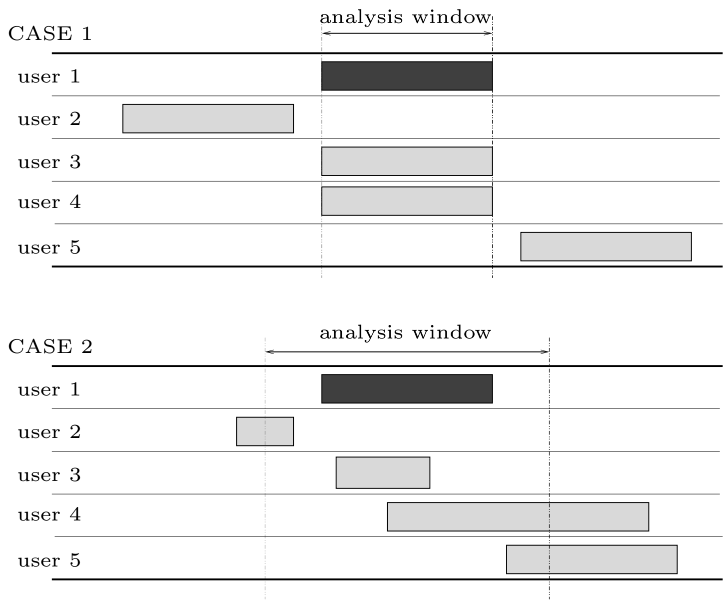


Figure 4.2: Slot structure

4.1 Data model

We assume the situation in figure 4.1 where several users occupy a common wireless channel. The channel is assumed to be narrowband. User 1 is the desired user, it is supposed to be received by receiver 1, but there will be interference from the other users. To suppress the interference, the receiver is equipped with an antenna array of M elements. The potential number of simultaneously active users d is in an analysis window/slot is assumed to be $d \leq M$.

The transmission is modeled by a linear data model of the form

$$\mathbf{x}_k = \sum_{q=1}^{d} \mathbf{a}_q s_k^{(q)} \; + \; \mathbf{n}_k \,, \tag{4.1}$$

where $\mathbf{x}_k \in \mathbb{C}^M$ is a complex data vector received by the array of M antennas at time k, $\mathbf{a}_q$ is the signature vector of source q and $s_k^{(q)} \in \mathbb{C}$ its transmitted symbol at time k, and $\mathbf{n}_k \in \mathbb{C}^M$ an additive noise vector. In this model we assume that the number of users in an analysis window is $d \leq M$, the number of antennas at the receiver.

The modulation of source 1 is assumed to be constant modulus (e.g., QPSK), i.e., $|s_k^{(1)}|^2 = 1$. The modulation of the other users is arbitrary.

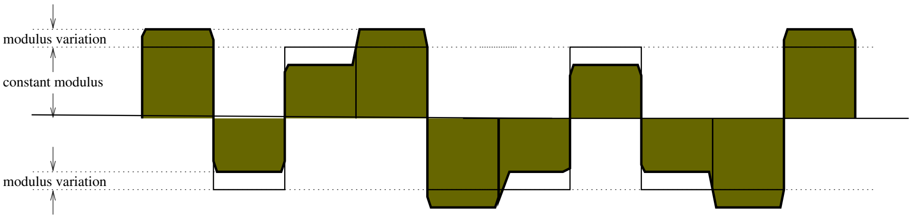


Figure 4.3: Constant modulus signal with coded amplitude variations which are used to identify the user of interest

We will consider two types of transmission scenarios (see figure 4.2):

1. *Slotted, with fixed slot length L.* The situation in a slot is stationary: the number of active users is constant inside a slot, and their spatial signature vectors are constant.

2. *Unslotted, with fixed or variable packet lengths.* Packets can have arbitrary starting times, hence the number of active users changes throughout the slot. The packet length of user 1 is denoted by L.

Initially, we assume that we are synchronized to the user of interest: the start time and length of his packet is known. We collect N samples in a data matrix $\mathbf{X} = [\mathbf{x}_1\,,\ \cdots\,,\ \mathbf{x}_N] : M \times N$. In case 1, we take $N = L$ and $\mathbf{x}_1$ contains the first sample of the packet. In case 2, we take a slightly larger analysis window to avoid certain edge effects, $N \geq L$ samples. In Chapter 5, we consider the estimation of the packet offset.

Let d be the maximal number of active users in the analysis window, and assume for notational simplicity that these are users 1 to d. Defining $\mathbf{A} = [\mathbf{a}_1\,, \cdots\,, \mathbf{a}_d] : M \times d$, $\mathbf{S} = [s_{qk}] = [s_k^{(q)}] : d \times N$ and $\mathbf{N} = [\mathbf{n}_1\,, \cdots\,, \mathbf{n}_N] : M \times N$, we obtain

$$\mathbf{X} = \mathbf{AS} + \mathbf{N}\,. \tag{4.2}$$

$\mathbf{A}$, $\mathbf{S}$ and $\mathbf{N}$ are unknown. The objective is to reconstruct the nonzero part of $\mathbf{s}^{(1)} = [s_1^{(1)}, \cdots, s_N^{(1)}]$ using linear beamforming, $i.e.$, to find a beamformer $\mathbf{w}$ such that $\hat{s}_k = \mathbf{w}^{\mathrm{H}}\mathbf{x}_k$ approximates $s_k^{(1)},\ k = 1, \cdots, N$.

As discussed in Chapter 1 blind algorithms for source separation are applicable at this point (e.g., CMAs), but they have the problem that they cannot distinguish one user from another. To distinguish the desired source, we give it a "color code", in the form of a known pseudo-random modulus variation (figure 4.3). Instead of transmitting s_k, we transmit $z_k = s_k\sqrt{c_k}$, where $c_k = 1 \pm \epsilon$ is a real and positive scaling that induces a small modulus variation, without changing the average transmission power.

For notational convenience, we assume that $c_k = 0$ outside the support of the packet. The data model (4.2) is replaced by

$$\mathbf{X} = \mathbf{A}\mathbf{Z} + \mathbf{N} . \tag{4.3}$$

Recall that we assume that $|s_k|^2 = 1$, so that $|z_k|^2 = c_k$. Similar to CMA, the objective of the beamformer will be to recover z_k based on its modulus, i.e., such that

$$|\mathbf{w}^{\mathrm{H}}\mathbf{x}_k|^2 = |\hat{z}_k|^2 \ = \ c_k \, , \qquad k = 1, \cdots, N \, .$$

With noise, we try to minimize the difference and can obviously recover the source only approximately.

4.2 Known Modulus Algorithms

In this section we consider receiver algorithms, assuming the receiver is synchronized to the user of interest and knows his code.

4.2.1 Iterative solutions

The usual iterative CMA can easily be adapted for the present case: one only has to define the instantaneous modulus error as $e_k = |\mathbf{w}^{\mathrm{H}}\mathbf{x}_k|^2 - c_k$. This leads to the updating step similar to the one presented in Section 3.3

$$\mathbf{w} := \mathbf{w} - \mu(\mathbf{x}_k^{\mathrm{H}}\mathbf{w})\mathbf{x}_k(|\mathbf{w}^{\mathrm{H}}\mathbf{x}_k|^2 - c_k) \, , \qquad k = 1, 2, \cdots, N \, .$$

This is the Known Modulus Algorithm (KMA) introduced by Treichler and Larimore in [48], and used more recently in [30, 31]. Apart from the usual stability and initialization issues, the resulting algorithm would not be very useful for the current purpose since we don't want to track the beamformer; we require a block solution where also the initial symbols are detected correctly. This is provided by an alternating projection algorithm: iterate until convergence

$$\left[\begin{array}{lcl} \mathbf{r} & := & \mathbf{w}^{\mathrm{H}}\mathbf{X} \\ \hat{z}_k & := & \frac{r_k}{|r_k|}\sqrt{c_k} \, , \qquad k = 1, \cdots, N \\ \mathbf{w} & := & (\hat{\mathbf{z}}\mathbf{X}^{\dagger})^{\mathrm{H}} \end{array} \right. \tag{4.4}$$

where $\dagger$ denotes the Moore-Penrose pseudo-inverse. Note that a candidate solution $\hat{\mathbf{z}}$ is alternatingly projected onto the row span of $\mathbf{X}$ (via the projection $\mathbf{X}^{\dagger}\mathbf{X}$), and entry-wise scaled to fit the modulus condition. With a sufficiently accurate initial point of $\mathbf{w}$, this algorithm is stable and converges usually nicely (similar to the LS-CMA, see [49]).

4.2.2 AKMA for slotted packet transmission (case 1)

Assume case 1 in figure 4.2 where all packets in a slot are synchronized. We will derive a closed-form solution to the problem of estimating $\mathbf{w}$, in the style of ACMA [9]. This can be used to obtain an initial point for the iteration (4.4). Given a block of N samples, we try to minimize

$$\begin{aligned}
\hat{\mathbf{w}}_1 &= \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{k=1}^{N} \left| |\mathbf{w}^{\mathrm{H}}\mathbf{x}_k|^2 - c_k \right|^2 \\
&= \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{k=1}^{N} |\mathbf{x}_k^{\mathrm{H}}\mathbf{w}\mathbf{w}^{\mathrm{H}}\mathbf{x}_k - c_k|^2 \\
&= \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{k=1}^{N} |(\bar{\mathbf{x}}_k \otimes \mathbf{x}_k)^{\mathrm{H}}(\bar{\mathbf{w}} \otimes \mathbf{w}) - c_k|^2 \\
&= \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{P}(\bar{\mathbf{w}} \otimes \mathbf{w}) - \mathbf{c}\|^2 \,,
\end{aligned} \tag{4.5}$$

where

$$\mathbf{P} = \begin{bmatrix} (\bar{\mathbf{x}}_1 \otimes \mathbf{x}_1)^{\mathrm{H}} \\ \vdots \\ (\bar{\mathbf{x}}_N \otimes \mathbf{x}_N)^{\mathrm{H}} \end{bmatrix} \quad \text{and} \quad \mathbf{c} = \begin{bmatrix} c_1 \\ \vdots \\ c_N \end{bmatrix} . \tag{4.6}$$

The size of $\mathbf{P}$ is $N \times M^2$. We follow the strategy of ACMA and split this optimization into two steps (hence suboptimal),

$$\begin{aligned}
\hat{\mathbf{y}} &= \operatorname{argmin}_{\mathbf{y}} \|\mathbf{P}\mathbf{y} - \mathbf{c}\|^2 \\
\hat{\mathbf{w}}_1 &= \operatorname{argmin}_{\mathbf{w}} \|\hat{\mathbf{y}} - \bar{\mathbf{w}} \otimes \mathbf{w}\|^2 \,.
\end{aligned}$$

If $\mathbf{P}$ is full column rank, the first problem has a unique solution in terms of the pseudo-inverse $\mathbf{P}^\dagger$:

$$\hat{\mathbf{y}} = \mathbf{P}^\dagger \mathbf{c} \,.$$

With this solution and setting $\hat{\mathbf{Y}} = \operatorname{unvec}(\hat{\mathbf{y}})$, where "unvec" denotes an unstacking of a vector into a square matrix, we can solve the second problem as

$$\hat{\mathbf{w}}_1 = \operatorname{argmin} \|\hat{\mathbf{y}} - \bar{\mathbf{w}} \otimes \mathbf{w}\|^2 = \operatorname{argmin} \|\hat{\mathbf{Y}} - \mathbf{w}\mathbf{w}^{\mathrm{H}}\|^2 \,,$$

the solution of which is given in terms of the dominant eigenvector of $\mathbf{Y}$, scaled by the square root of the corresponding eigenvalue.

We thus see that, if $\mathbf{P}$ is full rank, the algorithm becomes particularly simple, and in the noise-free case will produce the exact separating beamformer to recover the desired packet. If $\mathbf{P}$ is not of full column rank, then there will exist additional solutions $\mathbf{y}_0$ to $\mathbf{P}\mathbf{y}_0 = \mathbf{0}$ (where $\mathbf{0}$ is an all-zero vector) which will add to the desired solution $\mathbf{y} = \bar{\mathbf{w}}_1 \otimes \mathbf{w}_1$, producing a result that cannot be factored. We thus need to study the rank properties of $\mathbf{P}$. This is done in Section 4.2.4.

4.2.3 AKMA for unslotted packet transmission (case 2)

In case 2 in figure 4.2, users are not slotted. We first assume for simplicity that the base station is synchronized to user 1 (known offset). Estimation of the offset is done in Chapter 5.

If the analysis window length N is chosen to be the same as the packet length L, then the algorithm is the same as in Section 4.2.2. If N is larger than L, the packet has leading and/or trailing zeros. This can be modeled by defining

$$\mathbf{c} = [0 \cdots 0 \quad c_1 \quad c_2 \cdots c_L \quad 0 \cdots 0]^T\,, \qquad N \times 1\,,$$

where the total number of zeros is $N - L$ and the offset of the code matches the offset of the packet in the analysis window. After this, the algorithm is as in Section 4.2.2.

4.2.4 Rank of P for case 1

We consider the rank of $\mathbf{P}$ for case 1 (all users synchronized) in the noise-free case, where $\mathbf{X} = \mathbf{AZ}$. To recover $\mathbf{Z}$ using linear beamforming, we need $\mathbf{A}$ to be tall ($d \leq M$) and full rank. In this case, $\mathbf{X}$ has rank d. $\mathbf{P}$ has size $N \times M^2$, and $\mathbf{P}^{\mathrm{H}} = \bar{\mathbf{X}} \circ \mathbf{X} = (\bar{\mathbf{A}} \otimes \mathbf{A})(\bar{\mathbf{Z}} \circ \mathbf{Z})$, where $\circ$ denotes a column-wise Kronecker product: $\bar{\mathbf{X}} \circ \mathbf{X} = [\bar{\mathbf{x}}_1 \otimes \mathbf{x}_1\,,\ \cdots,\ \bar{\mathbf{x}}_N \otimes \mathbf{x}_N]$. Further, $\bar{\mathbf{A}} \otimes \mathbf{A}$ has size $M^2 \times d^2$ and is full rank, and $\bar{\mathbf{Z}} \circ \mathbf{Z}$ has size $d^2 \times N$. Properties of the Kronecker products are described in Section 2.8. The rank of $\mathbf{P}$ is equal to the rank of $\bar{\mathbf{Z}} \circ \mathbf{Z}$, therefore it cannot exceed d^2. A necessary condition for $\mathbf{P}$ to have rank d^2 is $d^2 \leq N$.

If $d = M$ and $d^2 \leq N$, then $\mathbf{P}$ can be full rank. If $d < M$, then $\mathbf{P}$ is not full rank, but can be made full rank by a prefiltering step (cf. [9, 50]). Compute the SVD of $\mathbf{X}$, $i.e.$, $\mathbf{X} = \mathbf{U\Sigma V}^{\mathrm{H}}$, where $\mathbf{U} : M \times d$ is unitary, $\mathbf{\Sigma} : d \times d$ is positive diagonal, and $\mathbf{V}^{\mathrm{H}} : d \times N$ is unitary, then we can replace $\mathbf{X}$ by

$$\underline{\mathbf{X}} := (\sqrt{N})\mathbf{\Sigma}^{-1}\mathbf{U}^{\mathrm{H}}\mathbf{X} = (\sqrt{N})\mathbf{V}^{\mathrm{H}}$$

which has d rows and is of full rank. Note that due to the prewhitening, $\underline{\mathbf{X}}$ satisfies a model $\underline{\mathbf{X}} = \underline{\mathbf{A}}\mathbf{Z}$, where $\underline{\mathbf{A}}$ is $d \times d$ and asymptotically *unitary* (for large N). From now on, we assume that the prewhitening has been performed and that, therefore, $d = M$ (we omit the underscore from the notation).

Even after the prefiltering, there are cases where $\mathbf{P}$ is singular, namely when at least two other sources are constant modulus or have equal modulus variation (equal color codes). Indeed, if $\mathbf{z}^{(2)} = \mathbf{w}_2^{\mathrm{H}}\mathbf{X}$ and $\mathbf{z}^{(3)} = \mathbf{w}_3^{\mathrm{H}}\mathbf{X}$ are constant-modulus, then $\mathbf{P}(\bar{\mathbf{w}}_2 \otimes \mathbf{w}_2) = \mathbf{1}$, $\mathbf{P}(\bar{\mathbf{w}}_3 \otimes \mathbf{w}_3) = \mathbf{1}$, and

$$\mathbf{P}(\bar{\mathbf{w}}_2 \otimes \mathbf{w}_2 - \bar{\mathbf{w}}_3 \otimes \mathbf{w}_3) = \mathbf{0}\,.$$

To avoid this nullspace solution, all sources (except perhaps one) should have amplitude modulations. In case 1, this level of cooperation is reasonable to assume.

We can show that if the sources are statistically independent constant modulus sources, all modulated by binary random power modulations $1 \pm \epsilon$, then as N becomes large, $\frac{1}{N}\mathbf{P}^{\mathrm{H}}\mathbf{P}$ converges to its expected value

$$\mathbf{C}_x := \mathrm{E}\{(\bar{\mathbf{x}}_k \otimes \mathbf{x}_k)(\bar{\mathbf{x}}_k \otimes \mathbf{x}_k)^{\mathrm{H}}\} = (\bar{\mathbf{A}} \otimes \mathbf{A})\mathbf{C}_z(\bar{\mathbf{A}} \otimes \mathbf{A})^{\mathrm{H}}, \tag{4.7}$$

where

$$\begin{aligned} \mathbf{C}_z &:= \mathrm{E}\{(\bar{\mathbf{z}}_k \otimes \mathbf{z}_k)(\bar{\mathbf{z}}_k \otimes \mathbf{z}_k)^{\mathrm{H}}\} \\ &= \mathbf{I} + \mathrm{vec}(\mathbf{I})\mathrm{vec}(\mathbf{I})^{\mathrm{H}} - (\mathbf{I} \circ \mathbf{I})(\mathbf{I} \circ \mathbf{I})^{\mathrm{H}} + \epsilon^2(\mathbf{I} \circ \mathbf{I})(\mathbf{I} \circ \mathbf{I})^{\mathrm{H}}, \end{aligned}$$

where $\mathbf{I}$ is the identity matrix. For the proof, see Appendix at the end of this Chapter (Section 4.3). The eigenvalues of $\mathbf{C}_z$ are

$$\mathrm{eig}(\mathbf{C}_z) = \{d + \epsilon^2,\ \underbrace{1, \cdots, 1}_{d^2-d},\ \underbrace{\epsilon^2, \cdots, \epsilon^2}_{d-1}\}. \tag{4.8}$$

For the proof, see again Appendix (Section 4.3). These are also the eigenvalues of $\mathbf{C}_x$ since $\mathbf{A}$ is asymptotically unitary after prewhitening. Thus, the singular values of $\mathbf{P}$ converge to

$$\mathrm{svd}(\mathbf{P}) = \{\sqrt{N(d+\epsilon^2)},\ \underbrace{\sqrt{N}, \cdots, \sqrt{N}}_{d^2-d},\ \underbrace{\epsilon\sqrt{N}, \cdots, \epsilon\sqrt{N}}_{d-1}\}. \tag{4.9}$$

The smallest singular value of $\mathbf{P}$ is raised by the modulation from 0 to $\epsilon\sqrt{N}$. If ϵ is not too small, $\mathbf{P}$ will be left invertible, so that $\mathbf{y} = \mathbf{P}^{\dagger}\mathbf{c}$ will lead to the correct solution. It can also be verified that these singular values carry important information: truncating them to zero leads to the wrong result.

4.2.5 Rank of P for case 2

In case 2 there are additional situations where $\mathbf{P}$ becomes singular, namely when two sources are non-overlapping in time. Indeed, suppose $\mathbf{z}^{(2)} = \mathbf{w}_2^{\mathrm{H}}\mathbf{X}$, $\mathbf{z}^{(3)} = \mathbf{w}_3^{\mathrm{H}}\mathbf{X}$ are such that $z_k^{(2)} z_k^{(3)} = 0,\ \forall k$. Then $\mathbf{w}_2^{\mathrm{H}}\mathbf{x}_k\mathbf{x}_k^{\mathrm{H}}\mathbf{w}_3^{\mathrm{H}} = 0,\ \forall k$, hence

$$\mathbf{P}(\bar{\mathbf{w}}_2 \otimes \mathbf{w}_3) = \mathbf{0}, \qquad \mathbf{P}(\bar{\mathbf{w}}_3 \otimes \mathbf{w}_2) = \mathbf{0}.$$

Thus, solutions $\mathbf{y}$ to $\mathbf{P}\mathbf{y} = \mathbf{c}$ can be written as

$$\mathbf{y} = \bar{\mathbf{w}}_1 \otimes \mathbf{w}_1 + \lambda_{23}(\bar{\mathbf{w}}_2 \otimes \mathbf{w}_3) + \lambda_{32}(\bar{\mathbf{w}}_3 \otimes \mathbf{w}_2)$$

for arbitrary scalars $\lambda_{23}, \lambda_{32}$, and an arbitrary $\mathbf{y}$ selected from the solution space cannot be factored into $\bar{\mathbf{w}}_1 \otimes \mathbf{w}_1$. We see two solutions for this problem. Firstly, we can write $\mathbf{Y} = \mathrm{unvec}(\mathbf{y})$ as

$$\mathbf{Y} = \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \mathbf{w}_3 \end{bmatrix} \begin{bmatrix} 1 & & \\ & \lambda_{32} & \\ & & \lambda_{23} \end{bmatrix} \begin{bmatrix} \mathbf{w}_1^{\mathrm{H}} \\ \mathbf{w}_3^{\mathrm{H}} \\ \mathbf{w}_2^{\mathrm{H}} \end{bmatrix} = \mathbf{W}\mathbf{\Lambda}_1\mathbf{M}^{\mathrm{H}},$$

1. SVD: $\mathbf{X} =: \mathbf{U\Sigma V}$ (M^2N)
   Estimate rank and truncate to $\mathbf{U}_s\mathbf{\Sigma}_s\mathbf{V}_s$
   Prefiltering: $\underline{\mathbf{X}} := \sqrt{L}\cdot\mathbf{\Sigma}_s^{-1}\mathbf{U}_s^{\mathrm{H}}\mathbf{X} = \sqrt{L}\cdot\mathbf{V}_s$
2. $\mathbf{P} = (\bar{\underline{\mathbf{X}}} \circ \underline{\mathbf{X}})^{\mathrm{H}}$ (d^2N)
   $\mathbf{y} = \mathbf{P}^{\dagger}\mathbf{c}$, with threshold $\epsilon\sqrt{N}$ (d^4N)
3. $\mathbf{Y} = \text{unvec}(\mathbf{y})$
   $\underline{\mathbf{w}} =$ dominant eigenvector of $\mathbf{Y}$ (d^3)
4. $\mathbf{w} = \sqrt{L}\cdot\mathbf{U}_s\mathbf{\Sigma}_s^{-1}\underline{\mathbf{w}}$
   $\hat{\mathbf{z}} = \mathbf{w}^{\mathrm{H}}\mathbf{X}$ (dN)
5. *optional:* alternating projection iterations

$\mathcal{O}(M^2N + d^4N)$

Figure 4.4: Summary of AKMA and the complexity of the algorithm. L is the packet length, N is the analysis window length.

where $\mathbf{M}$ is a permutation of $\mathbf{W}$. Similarly, if we take a basis $\{\mathbf{y}_2, \mathbf{y}_3\}$ of the null space of $\mathbf{P}$, it can be written as

$$\mathbf{Y}_2 = \mathbf{W}\mathbf{\Lambda}_2\mathbf{M}^{\mathrm{H}}\,, \qquad \mathbf{Y}_3 = \mathbf{W}\mathbf{\Lambda}_3\mathbf{M}^{\mathrm{H}}\,,$$

where $\mathbf{\Lambda}_2$, $\mathbf{\Lambda}_3$ are diagonal matrices (with their first entry equal to 0). The problem boils down to a joint diagonalization of unsymmetric matrices, or a joint Schur decomposition, which can be solved using Jacobi iterations [9]. Similar problem raised by the ACMA described in Section 3.4.

Alternatively, we try to avoid the joint diagonalization step. If we have N sufficiently large and do prewhitening, then $\mathbf{A}$ is approximately unitary, and the $\mathbf{w}_i$ are approximately orthogonal to each other. Hence, the desired solution $\bar{\mathbf{w}}_1 \otimes \mathbf{w}_1$ is orthogonal to the null space of $\mathbf{P}$. In this case, we can simply set

$$\mathbf{y} = \mathbf{P}^{\dagger}\mathbf{c} \approx \bar{\mathbf{w}}_1 \otimes \mathbf{w}_1\,.$$

With noise, $\mathbf{P}$ will not be exactly singular, and we will have to set a threshold on the pseudo-inverse: compute the SVD of $\mathbf{P}$ as $\mathbf{P} = \mathbf{U}_P\mathbf{\Sigma}_P\mathbf{V}_P^{\mathrm{H}}$, let $\hat{\mathbf{\Sigma}}_P$ be the submatrix containing the singular values of $\mathbf{P}$ larger than a threshold, and let $\hat{\mathbf{U}}_P$, $\hat{\mathbf{V}}_P$ be the corresponding left and right singular vectors. The solution to $\mathbf{Py} = \mathbf{c}$ which is orthogonal to the (approximate) null space of $\mathbf{P}$ is then given by $\mathbf{y} = \mathbf{V}_P\hat{\mathbf{\Sigma}}^{-1}\hat{\mathbf{U}}_P^{\mathrm{H}}\mathbf{c}$. As is clear from equation (4.9), the threshold on the singular values of $\mathbf{P}$ should be smaller than $\epsilon\sqrt{N}$.

In case 2, it may also happen that only a few samples of the desired packet are disturbed by the head or tail of another source, but with insufficient samples present

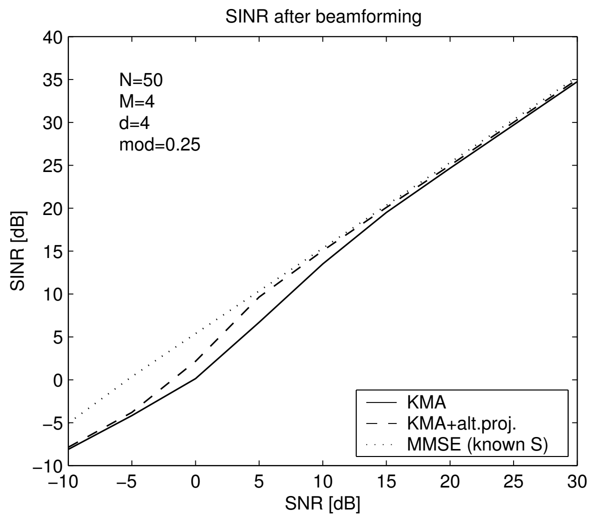


Figure 4.5: Case 1 beamformer performance: SINR of user 1 after beamforming as the function of signal to noise ratio (SNR) at the antenna array.

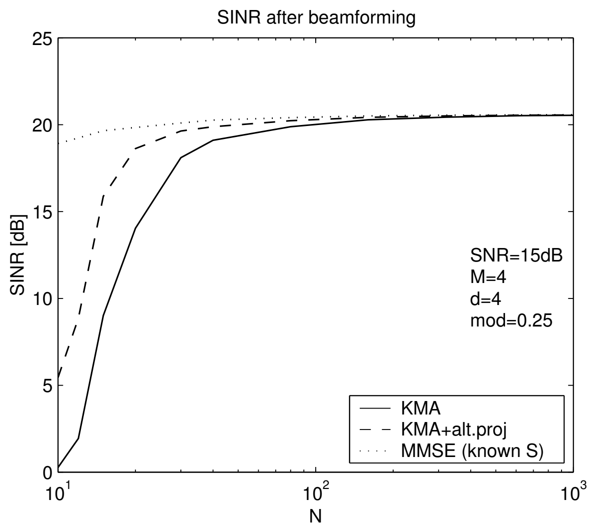


Figure 4.6: Case 1 beamformer performance: SINR of user 1 after beamforming as the function of the data packet size.

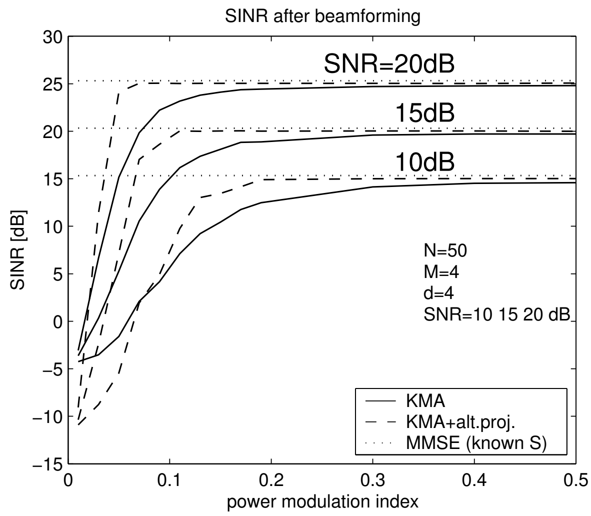


Figure 4.7: Case 1 beamformer performance: SINR of user 1 after beamforming for different levels of the power modulation index.

to estimate that source reliably. A convenient solution that avoids this problem is to increase the analysis window N to be larger than the packet length L, with the desired packet located in the center of the analysis window. In that case, if a disturbing source overlaps the desired packet, it will have at least $(N-L)/2$ samples in the analysis window, sufficient to detect it.

Figure 4.4 lists the algorithm as used in the simulations.

4.2.6 Simulations–known timing

We test the algorithm on simulated data. In these simulations, the receiver knows the code of the desired user, and it knows the timing of this user.

Case 1

Figures 4.5, 4.6 and 4.7 show SINR performance plots of the beamformer of the first source for a simulation with $d = 4$ sources, $M = 4$ antennas in a uniform linear array, equal source powers, and source angles $-20°, 20°, 40°, -40°$, for varying SNR, packet length $L(= N)$, and power modulation index ϵ. All sources are QPSK-modulated constant modulus sources with code-modulated amplitudes. The reference line is the performance of the MMSE receiver with knowledge of $\mathbf{z}^{(1)}$, namely $\mathbf{w} =$

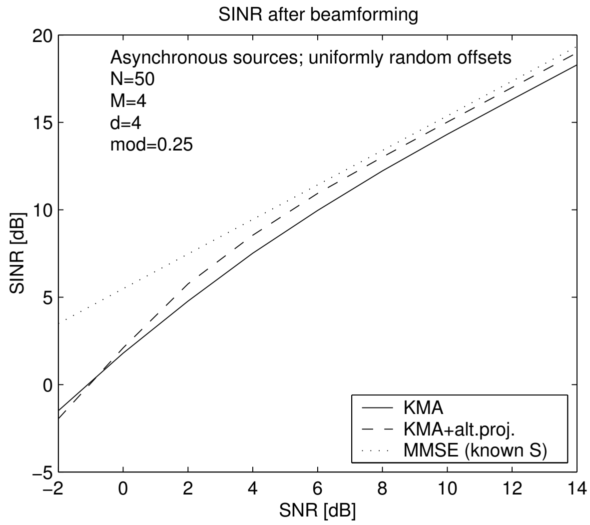


Figure 4.8: Case 2 beamformer performance: SINR of user 1 after beamforming. Asynchronous sources with equal-length packets.

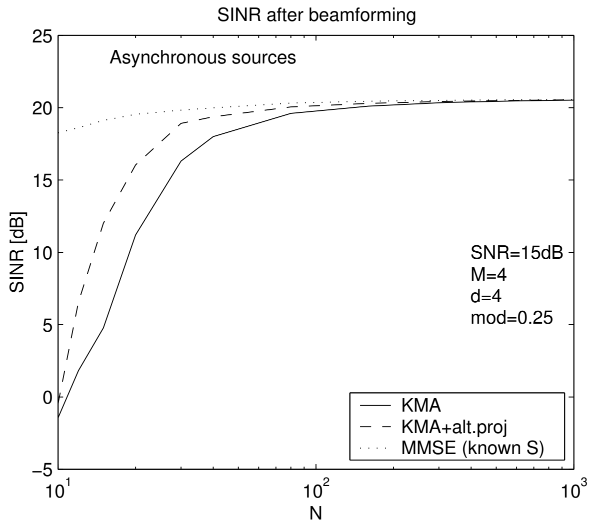


Figure 4.9: Case 2 beamformer performance: SINR of user 1 after beamforming. Asynchronous sources with equal-length packets.

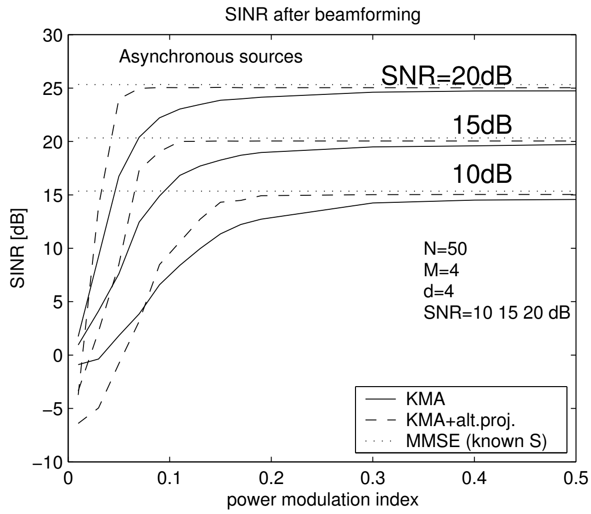


Figure 4.10: Case 2 beamformer performance: SINR of user 1 after beamforming. Asynchronous sources with equal-length packets.

$(\mathbf{z}^{(1)}\mathbf{X}^{\dagger})^{\mathrm{H}}$. The input SNR in dB is defined as $SNR = 10\log(P_1/P_n)$ where $P_n = \sigma^2$ is the power of the AWGN per receiving antenna. The power of the i-th user is defined as $P_i = (\sum_{k=1}^{L} |s_k^{(i)}|^2)/L$. P_1 is the power of user 1 - the user of interest. The SINR after beamforming is computed as $SINR_{out} = 10\log(P_{1,out}/P_{interf+n})$ with $P_{1,out} = \mathbf{w}^{\mathrm{H}}\mathbf{a}_1 P_1 \mathbf{a}_1^{\mathrm{H}}\mathbf{w}$, $P_{interf+n} = \mathbf{w}^{\mathrm{H}}(\mathbf{A}\mathrm{diag}([P_1 \cdots P_d])\mathbf{A}^{\mathrm{H}} - \mathbf{a}_1 P_1 \mathbf{a}_1^{\mathrm{H}} + \sigma^2\mathbf{I})\mathbf{w}$, where $\mathbf{a}_1$ is the first column of the array manifold $\mathbf{A}$. The solid line is the performance of AKMA, the dashed line the performance of 15 iterations of the alternating projection algorithm (4.4), initialized by the AKMA. It is seen that the performance of the AKMA is generally quite good, but that it can be improved for small modulation indices and small N (i.e., $N < 2d^2$). This is due to the squaring involved in the construction of $\mathbf{P}$, and the presence of additional kernel solutions for small modulations.

The convergence of the AKMA to the MMSE is expected: similar to ACMA it is caused by the prewhitening and the fact that the algorithm is unbiased in the noise-free case [50].

Case 2

Figures 4.8, 4.9 and 4.10 show similar performance curves, but for case 2 (unsynchronized users). In these plots, the analysis window N is equal to the packet length L; the packet of user 1 falls completely in the analysis window whereas the three other users have arbitrary arrival times. The performance is virtually identical to that in case 1.

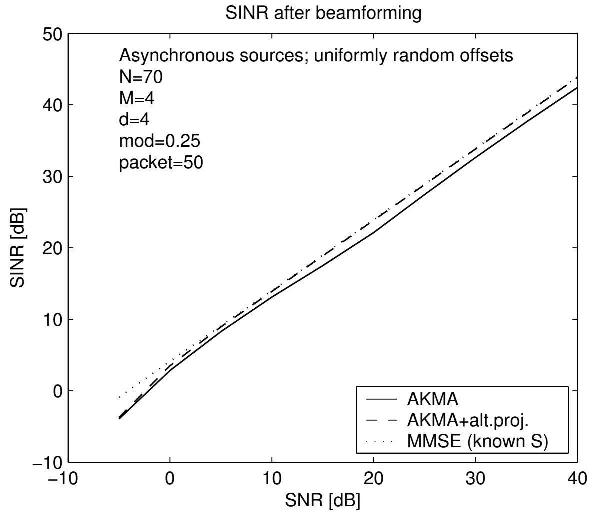


Figure 4.11: Case 2 beamformer performance: SINR of user 1 after beamforming. Asynchronous sources with equal-length packets $L = 50$, and an analysis window $N = 70$.

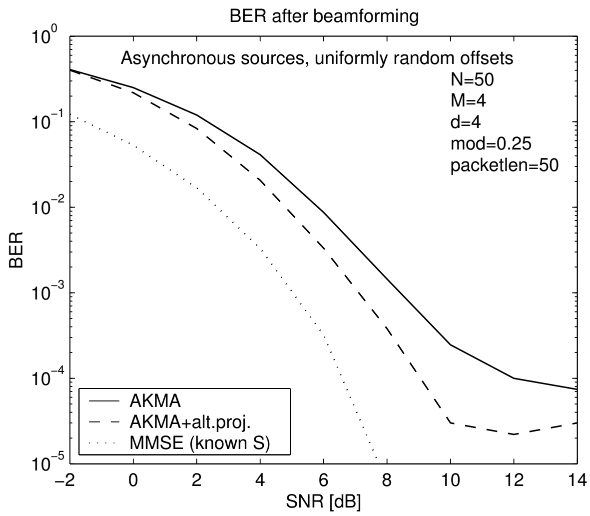


Figure 4.12: Case 2 beamformer performance: BER of user 1 after beamforming. Asynchronous sources with equal-length packets $L = 50$, and an analysis window (b) $N = 50$.

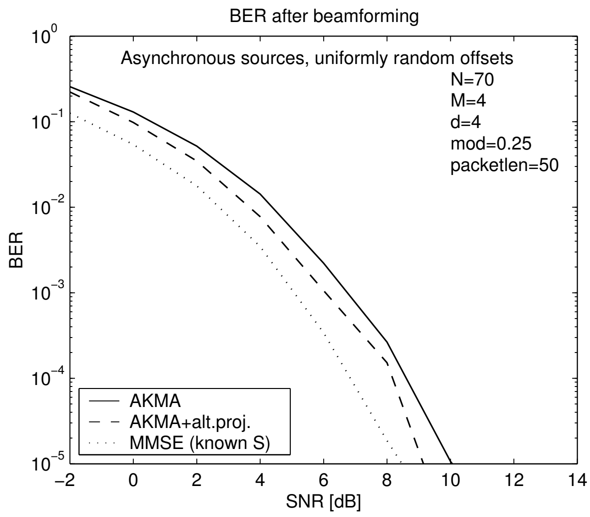


Figure 4.13: Case 2 beamformer performance: BER of user 1 after beamforming. Asynchronous sources with equal-length packets $L = 50$, and an analysis window $N = 70$.

In figures 4.11,4.12 and 4.13, we investigate the choice $N > L$. This is motivated by the bit-error rate curve in fig. 4.12, which (for $N = L = 50$) shows a saturation of the performance for large SNRs. This is because the head and tail of the desired packet can be disturbed by the tail of another source, but with insufficient samples present to estimate that source reliably. Figure 4.13 shows that increasing the analysis window N to $N = 70$ improves the bit error rate (BER) for large SNRs. Figure 4.11 shows the SINR performance for $N = 70$, which can be compared to Figure 4.8 where $N = L = 50$.

4.3 Appendix

Proof of equation (4.7)

First consider a scalar variable, $z = \sqrt{c} \cdot s$, where s is a constant modulus random variable, $c = 1 + e$, and $e = \pm\epsilon$ is a zero mean binary random variable. Then $\mathrm{E}(|z|^4) = \mathrm{E}(1+e)^2 = 1 + \epsilon^2$. The kurtosis of z is $\kappa_z = \mathrm{E}(|z|^4) - 2\mathrm{E}(|z|^2)^2 = -1 + \epsilon^2$.

To prove (4.7), we use some properties of random CM vector signals that are listed in [50]. In particular, if $\mathbf{s}$ is a random vector whose entries are statistically independent and constant modulus, then

$$\begin{aligned}\mathbf{C}_s &:= \mathrm{E}\{(\bar{\mathbf{s}}\otimes\mathbf{s})(\bar{\mathbf{s}}\otimes\mathbf{s})^{\mathrm{H}}\}\\ &= \mathrm{E}\{\mathbf{s}\mathbf{s}^{\mathrm{H}}\}\mathrm{E}\{\bar{\mathbf{s}}\bar{\mathbf{s}}^{\mathrm{H}}\} + \mathrm{E}\{\bar{\mathbf{s}}\otimes\mathbf{s}\}\mathrm{E}\{\bar{\mathbf{s}}\otimes\mathbf{s}\}^{\mathrm{H}} + \mathbf{K}_s\\ &= \mathbf{I} + \mathrm{vec}(\mathbf{I})\mathrm{vec}(\mathbf{I})^{\mathrm{H}} - (\mathbf{I}\circ\mathbf{I})(\mathbf{I}\circ\mathbf{I})^{\mathrm{H}},\end{aligned}$$

where the $d^2 \times d^2$ matrix $\mathbf{K}_s = -(\mathbf{I}\circ\mathbf{I})(\mathbf{I}\circ\mathbf{I})^{\mathrm{H}}$ is the kurtosis of $\mathbf{s}$. It is a diagonal matrix with only d nonzero entries (-1) at selected locations on its main diagonal, which reflects the independence of the entries of $\mathbf{s}$.

Now, define $\mathbf{z} = \mathbf{\Gamma}^{1/2}\mathbf{s}$, where $\mathbf{\Gamma} = \mathbf{I} + \mathbf{E}$ and $\mathbf{E}$ is diagonal with independent entries $\pm\epsilon$, then

$$\mathrm{E}\{\mathbf{z}\mathbf{z}^{\mathrm{H}}\} = \mathrm{E}\{\mathbf{\Gamma}\} = \mathbf{I},$$

and similarly $\mathrm{E}\{\bar{\mathbf{z}}\otimes\mathbf{z}\} = \mathrm{vec}(\mathbf{I})$. We already showed that the kurtosis of a single entry of $\mathbf{z}$ is $-1+\epsilon^2$. Because of independence, the kurtosis of $\mathbf{z}$ is therefore $\mathbf{K}_z = (-1+\epsilon^2)(\mathbf{I}\circ\mathbf{I})(\mathbf{I}\circ\mathbf{I})^{\mathrm{H}}$. It follows that

$$\begin{aligned}\mathbf{C}_z &= \mathrm{E}\{\mathbf{z}\mathbf{z}^{\mathrm{H}}\}\mathrm{E}\{\bar{\mathbf{z}}\bar{\mathbf{z}}^{\mathrm{H}}\} + \mathrm{E}\{\bar{\mathbf{z}}\otimes\mathbf{z}\}\mathrm{E}\{\bar{\mathbf{z}}\otimes\mathbf{z}\}^{\mathrm{H}} + \mathbf{K}_z\\ &= \mathbf{I} + \mathrm{vec}(\mathbf{I})\mathrm{vec}(\mathbf{I})^{\mathrm{H}} + (-1+\epsilon^2)(\mathbf{I}\circ\mathbf{I})(\mathbf{I}\circ\mathbf{I})^{\mathrm{H}}.\end{aligned}$$

Proof of equation (4.8)

Inspection of the structure of $\mathbf{C}_z$ shows that, after column and row permutations, it can be written as

$$\mathbf{C}_z \sim \begin{bmatrix} \mathbf{1}_d\mathbf{1}_d^{\mathrm{H}} + \epsilon^2\mathbf{I}_d & \mathbf{0}\\ \mathbf{0} & \mathbf{I}_{d^2-d}\end{bmatrix}$$

where $\mathbf{1}_d$ is a vector consisting of d entries '1', and $\mathbf{I}_d$ is the $d\times d$ identity matrix. Hence d^2-d eigenvalues are equal to 1. The eigenvalues of $\mathbf{1}_d\mathbf{1}_d^{\mathrm{H}}$ are $\{d, 0, \cdots, 0\}$, hence the eigenvalues of $\mathbf{1}_d\mathbf{1}_d^{\mathrm{H}} + \epsilon^2\mathbf{I}_d$ are $\{d+\epsilon^2, \epsilon^2, \cdots, \epsilon^2\}$.

Chapter 5

Joint offset and beamformer estimation

In the previous chapter, we assumed that the receiver was synchronized to the user of interest. Now, we consider the situation where the receiver is not synchronized, e.g., the users transmit packets at random moments. We assume that the receiver collects a batch of N samples, where $N > L$, and introduce an algorithm to estimate the offset of the packet of the desired user within this analysis window, as well as a beamformer to cancel the interference. Introduce τ as the unknown packet offset. Similar to equation (4.5), we now have to compute a beamformer $\mathbf{w}$ and offset τ such that

$$|\mathbf{w}^{\mathrm{H}}\mathbf{x}_k|^2 = |\hat{z}_k|^2 \; = \; c_{k-\tau} \, , \qquad k = 1, \cdots, N \, . \tag{5.1}$$

For simplicity, we first assume that τ is an integer, and derive corresponding algorithms in Section 5.1. In Section 5.2 we extend this to arbitrary noninteger delays, which can be estimated if oversampling is considered. Results were published in [44, 45].

5.1 Integer offset estimation

5.1.1 Data model

Let user 1 with code $\{c_k, \; k = 1, \cdots, L\}$ be the user of interest. We consider the case where the packet falls completely within the analysis window, $N > L$, and the packet offset τ is an integer. Our aim is to compute

$$\begin{aligned}\{\hat{\mathbf{w}}, \hat{\tau}\} &= \underset{\mathbf{w},\tau}{\operatorname{argmin}} \sum_{k=1}^{N} (|\mathbf{w}^{\mathrm{H}} \mathbf{x}_k|^2 - c_{k-\tau})^2 \\ &= \underset{\mathbf{w},\tau}{\operatorname{argmin}} \|\mathbf{P}(\bar{\mathbf{w}} \otimes \mathbf{w}) - \mathbf{c}_\tau\|^2 ,\end{aligned}$$

Here,

$$\mathbf{c}_\tau = [\underbrace{0, \cdots, 0}_{\tau},\ c_1, \cdots, c_L,\ \underbrace{0, \cdots, 0}_{N-L-\tau}]^T$$

is a vector of length N, and the matrix $\mathbf{P}$ is constructed from the received data in the same way as in Section 4.2.2. As before, we continue with a two-step optimization problem

$$\begin{aligned}\{\hat{\mathbf{y}}, \hat{\tau}\} &= \underset{\mathbf{w},\tau}{\operatorname{argmin}} \|\mathbf{P}\mathbf{y} - \mathbf{c}_\tau\|^2 \\ \hat{\mathbf{w}} &= \underset{\mathbf{w}}{\operatorname{argmin}} \|\hat{\mathbf{y}} - \bar{\mathbf{w}} \otimes \mathbf{w}\|^2 ,\end{aligned}$$

and solve the first problem, which asks for joint estimation of $\mathbf{y}$ and τ. Similar to the derivation of the SI-JADE algorithm [51], we exploit the fact that a delay in time domain corresponds to a phase progression in frequency domain. This can be expressed as

$$\mathbf{F}\mathbf{c}_\tau = \mathbf{F}\mathbf{c}_0 \odot \boldsymbol{\phi}_\tau$$

where $\mathbf{F}$ is the $N \times N$ DFT matrix, $\odot$ represents an entrywise multiplication (Schur-Hadamard product),

$$\boldsymbol{\phi}_\tau := [1 \quad \varphi \quad \varphi^2 \ \cdots \ \varphi^{N-1}]^{\mathrm{T}}$$

and $\varphi = e^{-j\frac{2\pi\tau}{N}}$. The vector $\mathbf{c}_0$ is the unshifted code vector followed by $N - L$ zeros. Our objective will be to estimate φ based on the shift-invariance structure exhibited by the vector $\boldsymbol{\phi}_\tau$. This then immediately determines the offset τ. A similar approach was considered in the SI-JADE algorithm [51] for joint angle-delay estimation.

Thus, apply $\mathbf{F}$ to the equation $\mathbf{P}\mathbf{y} = \mathbf{c}_\tau$ to obtain

$$\mathbf{F}\mathbf{P}\mathbf{y} = \mathbf{F}\mathbf{c}_0 \odot [1\ \varphi\ \varphi^2\ \ldots\ \varphi^{N-1}]^{\mathrm{T}} \quad \Leftrightarrow \quad \mathbf{P}_f\mathbf{y} = \mathbf{g} \odot \boldsymbol{\phi}_\tau \tag{5.2}$$

where $\mathbf{P}_f := \mathbf{F}\mathbf{P}$ and $\mathbf{g} := \mathbf{F}\mathbf{c}_0$. Dividing the rows of $\mathbf{P}_f$ with the corresponding entries of the vector $\mathbf{g}$ we arrive at

$$\tilde{\mathbf{P}}\mathbf{y} = \boldsymbol{\phi}_\tau \tag{5.3}$$

where $\tilde{\mathbf{P}} = (\mathrm{diag}(\mathbf{g}))^{-1}\mathbf{P}_f$ is known and the vector $\boldsymbol{\phi}_\tau$ is a known function of the unknown delay τ. Here, 'diag' maps a vector into a diagonal matrix. The above pointwise division puts a design constraint on the code: it should be chosen such that (after zero padding to length N) it does not contain small values in the DFT domain.

5.1.2 MUSIC-like algorithm

Equation (5.3) can be treated in several different ways. Essentially we have to search for a vector in the column span of $\tilde{\mathbf{P}}$ that has the structure exhibited by $\boldsymbol{\phi}_\tau$, i.e., a shift invariance structure. Obviously a MUSIC-type search is applicable (see Section 2.7): if $\mathbf{U}_s$ is a basis for the dominant column span of $\tilde{\mathbf{P}}$, then

$$\hat{\tau} = \underset{\tau}{\operatorname{argmax}} \ \|\boldsymbol{\phi}_\tau^{\mathrm{H}} \mathbf{U}_s\|^2 \,. \tag{5.4}$$

The optimum is found by searching over a coarse grid, selecting the best interval, and subsequently refining if a higher resolution is required. For integer values of τ, the rows $\boldsymbol{\phi}_\tau^{\mathrm{H}}$ are actually rows of the inverse FFT matrix $\mathbf{F}^{\mathrm{H}}$, and hence we can implement the coarse search over integer values of τ by applying an inverse FFT to $\mathbf{U}_s$. Hence, we simply have to find the row of $\mathbf{F}^{\mathrm{H}}\mathbf{U}_s$ with maximal norm. Essentially, the algorithm has at this point performed a deconvolution with the desired code, implemented in frequency domain. The MUSIC-type algorithm shows good performance in simulations. Besides the processing steps described in figure 4.4, the additional complexity due to the synchronization step is given by: the FFT of $\mathbf{P}$ and IFFT of $\mathbf{U}_s$ $(2d^2 N \log_2 N)$, division by the user's code $(d^2 N)$ and computation of the norm of the first L rows (Ld^2). The complexity of the synchronization part is therefore of order $\mathcal{O}(2d^2 N \log_2 N)$. This is less than the complexity of the other steps in the algorithm, which is dominated by the inversion of $\mathbf{P}(d^4 N)$ as presented in figure 4.4.

5.1.3 ESPRIT-like algorithms

To avoid the search, we can also implement an ESPRIT-like algorithm (see also Section 2.6), where the difference is that, here, we expect only a single column in the column span of $\tilde{\mathbf{P}}$ with shift-invariance structure, whereas in ESPRIT all columns have such a structure.

To this end, split $\tilde{\mathbf{P}}$ into two matrices $\tilde{\mathbf{P}}_x$ and $\tilde{\mathbf{P}}_y$ by taking its first and last $N-1$ rows, respectively. We thus obtain

$$\begin{array}{rcl} \tilde{\mathbf{P}}_x\mathbf{y} & = & [1 \quad \varphi \quad \cdots \quad \varphi^{N-2}]^{\mathrm{T}} \\ \tilde{\mathbf{P}}_y\mathbf{y} & = & [\varphi \quad \varphi^2 \quad \cdots \quad \varphi^{N-1}]^{\mathrm{T}} \,. \end{array} \tag{5.5}$$

This can also be written as

$$\tilde{\mathbf{P}}_x\mathbf{y} = \bar{\varphi}\tilde{\mathbf{P}}_y\mathbf{y} \,, \tag{5.6}$$

which (because $\tilde{\mathbf{P}}_x$ and $\tilde{\mathbf{P}}_y$ are tall) is recognized as a matrix pencil problem. To solve it, we must first find the common column span of $\tilde{\mathbf{P}}_x$ and $\tilde{\mathbf{P}}_y$. Equivalently, we can look at the null space of $[\tilde{\mathbf{P}}_x \ \ \tilde{\mathbf{P}}_y]$.

Algorithm 1

The simplest technique to intersect the column span of $\tilde{\mathbf{P}}_x$ and that of $\tilde{\mathbf{P}}_y$ is to compute the SVD of $[\tilde{\mathbf{P}}_x \;\; \tilde{\mathbf{P}}_y]$. Indeed, from equation (5.5), we see that

$$[\tilde{\mathbf{P}}_x \quad \tilde{\mathbf{P}}_y] \begin{bmatrix} \mathbf{y} \\ -\bar{\varphi}\mathbf{y} \end{bmatrix} = \mathbf{0}\,. \tag{5.7}$$

Now it is clear that after an "economy size" SVD is performed of $[\tilde{\mathbf{P}}_x \;\; \tilde{\mathbf{P}}_y]$, at least one singular value will be zero. The corresponding basis for the null space specifies a set of candidate solutions to (5.6). If $\tilde{\mathbf{P}}_x$ and $\tilde{\mathbf{P}}_y$ each have full column rank, then we can simplify immediately, since we expect only a single solution $\mathbf{v}_n$ in the null space, which then will have the form

$$\mathbf{v}_n =: \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_y \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ -\bar{\varphi}\mathbf{y} \end{bmatrix}. \tag{5.8}$$

After finding $\mathbf{v}_n$, we can estimate the phaseshift φ as $\hat{\varphi} = -(\overline{\mathbf{v}_x^{\dagger}\mathbf{v}_y})$, which directly specifies a delay estimate $\hat{\tau}$. The estimate can be improved by performing a MUSIC-type search (5.4) in the vicinity of this estimate. This will be referred to as "Algorithm 1" in the simulations.

At the same time we can set $\hat{\mathbf{y}} := \mathbf{v}_x$, and since $\mathbf{y} = \bar{\mathbf{w}} \otimes \mathbf{w}$ we can estimate the separating beamformer $\mathbf{w}$ as indicated before: set $\hat{\mathbf{Y}} = \mathrm{unvec}(\hat{\mathbf{y}})$, and let $\hat{\mathbf{w}}$ be the dominant eigenvector of $\mathbf{Y}$, scaled by the square root of the corresponding eigenvalue. This is the estimated beamformer for user 1.

The above algorithm assumed that $\tilde{\mathbf{P}}_x$ and $\tilde{\mathbf{P}}_y$ are full rank. Alternatively we can work with a basis of these subspaces, obtained e.g., after the "economy-size" SVDs

$$\begin{aligned} \tilde{\mathbf{P}}_x &\approx \hat{\mathbf{U}}_x \hat{\mathbf{\Sigma}}_x \hat{\mathbf{V}}_x^{\mathrm{H}} \\ \tilde{\mathbf{P}}_y &\approx \hat{\mathbf{U}}_y \hat{\mathbf{\Sigma}}_y \hat{\mathbf{V}}_y^{\mathrm{H}} \end{aligned} \tag{5.9}$$

where we drop the small singular values and corresponding vectors. Similar to (5.7) we find

$$[\hat{\mathbf{U}}_x \quad \hat{\mathbf{U}}_y] \begin{bmatrix} \hat{\mathbf{\Sigma}}_x \hat{\mathbf{V}}_x^{\mathrm{H}}\mathbf{y} \\ -\bar{\varphi}\hat{\mathbf{\Sigma}}_y \hat{\mathbf{V}}_y^{\mathrm{H}}\mathbf{y} \end{bmatrix} = 0\,. \tag{5.10}$$

We can compute the vector ($\mathbf{v}_n$ say) in the null space of $[\hat{\mathbf{U}}_x \quad \hat{\mathbf{U}}_y]$, which will have the following structure:

$$\mathbf{v}_n =: \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_y \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{\Sigma}}_x \hat{\mathbf{V}}_x^{\mathrm{H}}\mathbf{y} \\ -\bar{\varphi}\hat{\mathbf{\Sigma}}_y \hat{\mathbf{V}}_y^{\mathrm{H}}\mathbf{y} \end{bmatrix}. \tag{5.11}$$

The vector $\mathbf{y}$ can be computed as $\mathbf{y} = \hat{\mathbf{V}}_x \hat{\mathbf{\Sigma}}_x^{-1}\mathbf{v}_x$, and φ follows from $-\bar{\varphi} = (\hat{\mathbf{\Sigma}}_y \hat{\mathbf{V}}_y^{\mathrm{H}}\mathbf{y})^{\dagger}\mathbf{v}_y$.

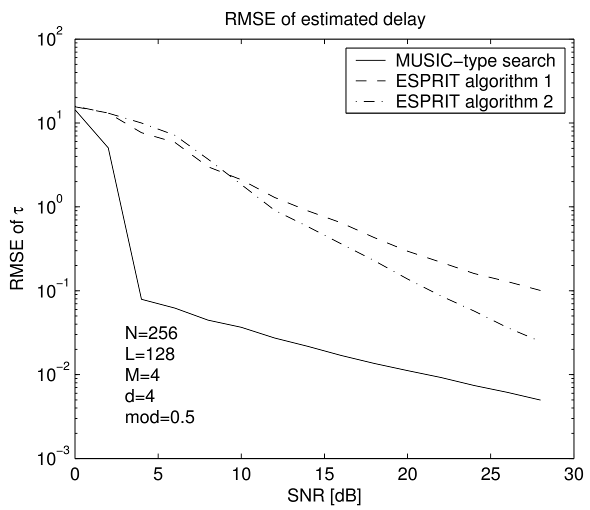


Figure 5.1: Case 2 (asynchronous sources): Root Mean Square Error of the delay estimate.

Algorithm 2

Another algorithm for subspace intersection is mentioned in [52]: the common vector in the column span of $\tilde{\mathbf{P}}_x$ and $\tilde{\mathbf{P}}_y$ is given by the *largest* left singular vector of $[\hat{\mathbf{U}}_x \quad \hat{\mathbf{U}}_y]$, the one corresponding to a singular value $\sqrt{2}$. Interestingly, this vector should have the structure $\boldsymbol{\phi} = [1 \;\; \varphi \;\; \varphi^2 \cdots \varphi^{N-2}]^{\mathrm{T}}$. By computing the vector in the intersection and matching it to this shift-invariance structure, we have another way to compute φ, and hence the offset delay.

Let $\mathbf{u}$ be the largest left singular vector of $[\hat{\mathbf{U}}_x \quad \hat{\mathbf{U}}_y]$. Under noise-free conditions, we have $\mathbf{u} = [1 \;\; \varphi \;\; \varphi^2 \cdots \varphi^{N-2}]^{\mathrm{T}}$. We can estimate φ as in ESPRIT, by constructing $\mathbf{u}_x$ and $\mathbf{u}_y$ consisting of the first and last $N-2$ elements of $\mathbf{u}$, respectively, so that $\hat{\varphi} = \mathbf{u}_x^{\dagger}\mathbf{u}_y$. It is possible to obtain a better estimate of φ by the additional limited MUSIC search using the complete known structure of $\boldsymbol{\phi}$.

In each of the algorithms, the estimated beamformer can be improved by a few iterations of an alternating projection algorithm as already proposed in Section 4.2.1.

5.1.4 Simulations–integer timing estimation

In the following simulations we consider $d = 4$ users and $M = 4$ antennas in a uniform linear array with half-wavelength spacing. Signals are arriving at the array with

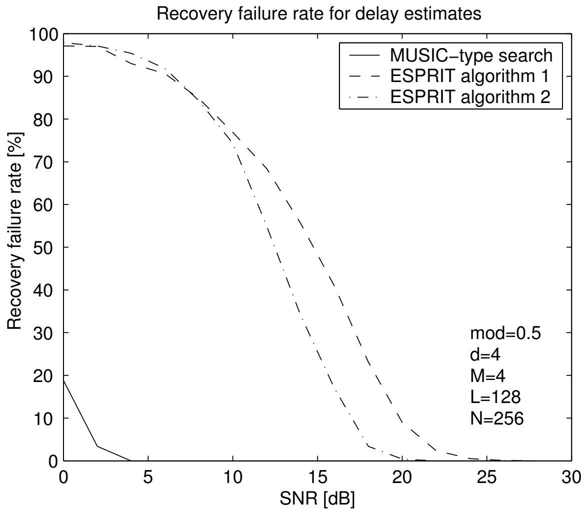


Figure 5.2: Case 2 (asynchronous sources): Percentage of incorrectly estimated packet offsets τ.

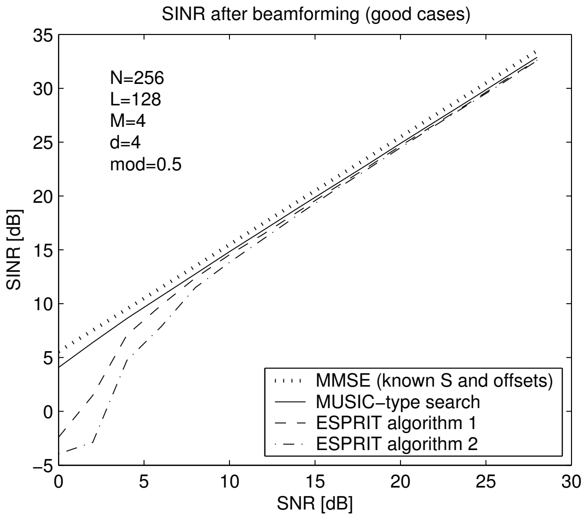


Figure 5.3: Case 2 (asynchronous sources): Output SINR of user 1 after beamforming (cases without failure), all users having equal power.

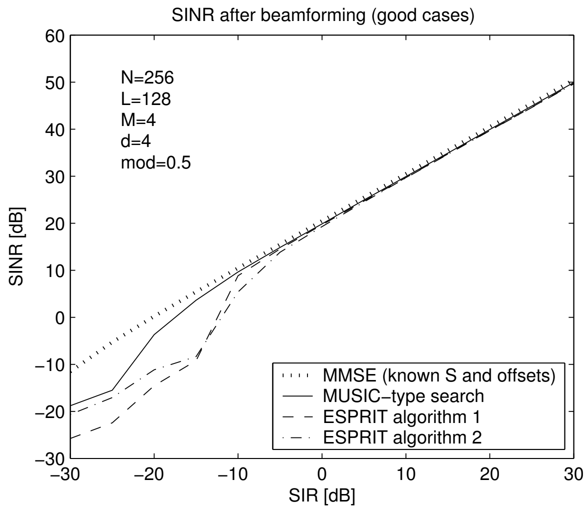


Figure 5.4: Case 2 (asynchronous sources): Output SINR of user 1, where its power is varied with respect to the total power of the interfering users.

angles $[-10°,\ 20°,\ 40°,\ -40°]$ with respect to the array broadside, and with delays $[50,\ 34,\ 65,\ 95]$ samples. The packet length is $L = 128$ while the analysis window size is $N = 256$. All sources are transmitting unit amplitude constant modulus signals modulated by a power modulation with index $\epsilon = 0.5$. The amplitude codes are Gold sequences in order to minimize the cross correlation between the codes of different users. 1000 Monte-Carlo runs for each value of the input SNR were performed.

The power of the i-th user is defined in the region where the signal exists $i.e.$, $P_i = (\sum_{k=\tau_i+1}^{\tau_i+L} |s_k^{(i)}|^2)/L$. P_1 is the power of user 1 - the user of interest. In the simulations presented in figure 5.1, 5.2 and 5.3 all users have the same transmitting power whereas in figure 5.4 the power of the user of interest is varied with respect to the power of the interfering sources. The input SNR in dB is defined as $SNR = 10\log(P_1/P_n)$ where $P_n = \sigma^2$ is the power of the AWGN per receiving antenna. The SINR after beamforming is computed as $SINR_{out} = 10\log(P_{1,out}/P_{interf+n})$ with $P_{1,out} = \mathbf{w}^{\mathrm{H}}\mathbf{a}_1 P_1 \mathbf{a}_1^{\mathrm{H}}\mathbf{w}$, $P_{interf+n} = \mathbf{w}^{\mathrm{H}}(\mathbf{A}\mathrm{diag}([P_1 \cdots P_d])\mathbf{A}^{\mathrm{H}} - \mathbf{a}_1 P_1 \mathbf{a}_1^{\mathrm{H}} + \sigma^2\mathbf{I})\mathbf{w}$, where $\mathbf{a}_1$ is the first column of the array manifold $\mathbf{A}$.

Figure 5.1 presents the root mean square error (RMSE) of the estimated delay for user 1 for each of the proposed algorithms. Similarly, figure 5.2 shows the percentage of cases where the delay offset was not estimated correctly. An estimate is labeled as failure if its rounded value is not equal to the true (integer) delay offset.

From figures 5.1 and 5.2, we see that the MUSIC search performs very reliable, and much better than the closed-form ESPRIT-type algorithms. These algorithms perhaps can be used to obtain coarse estimates of the delays, which can subsequently be refined using a MUSIC-type search over a limited interval. (At the same time, the complexity of the MUSIC-search, when implemented via an inverse FFT, is lower than that of the ESPRIT-type algorithms.)

The "jump" in the performance for low SNR is typical for subspace algorithms: these involve a nonlinear step namely the selection of a subspace which, for low SNR, may exclude the vector of interest.

For the cases without failure, figure 5.3 presents the signal to interference and noise ratio (SINR) at the output, i.e., after beamforming. The dotted line is a theoretical reference line showing the performance of the MMSE beamformer assuming the transmitted signals, codes and offsets of all users are known. The statistics are computed only over the cases without failure. The performance of the algorithms follows that of the MMSE estimator closely, and can be further improved with a few iterations of the alternating projection algorithm.

Figure 5.4 presents the SINR after beamforming as a function of the input signal to interference ratio $SIR = 10\log(P_1/P_{interf})$. All interfering sources have fixed unit power while the power of the user of interest is changed in order to simulate a near/far scenario. The interference to noise ratio is defined as $INR = 10\log(P_{interf}/P_n) = +15$dB where P_{interf} , P_n are the total interfering and noise power respectively. The INR is kept constant for all the SIR values in the simulation.

5.2 Non-integer offset estimation

5.2.1 Algorithm

There are two cases where the DFT property of mapping a delay to a phase progression is accurate: the delay is a multiple of the sampling period, or the signal is sampled at or above Nyquist rate. If the signal was sampled below the Nyquist rate, aliasing occurs which destroys the shift invariance property (cf. [52]). Up to this point, we considered sampling at the symbol rate. Since delays are, in general, noninteger, this leads to inaccuracies when realistic data is considered. Therefore we now consider a case-2 scenario with non-integer delays τ, and where we sample at a rate of P times the symbol rate, assuming that this is above Nyquist. Since there is no need to sample much faster than Nyquist, typically $P = 2$ is sufficient.

The signal $s(t)$ is an analog constant modulus signal, e.g., a phase modulated data sequence. The amplitude code sequence is an analog function $c(t)$, and we assume that $z(t) = s(t)\sqrt{c(t)}$ is sampled above Nyquist, hence $c(t)$ is sufficiently smooth. Under these conditions, we are still in context of the proposed KMAs, hence the joint delay-beamformer estimation algorithms proposed in Section 5.1 are applicable.

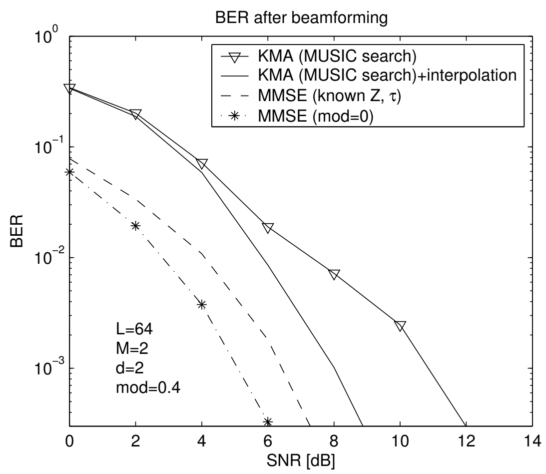


Figure 5.5: Case 2 beamformer performance (asynchronous sources with equal-length packets) where MSK modulation scheme is implemented: BER after beamforming.

Because of oversampling, the spectrum of the 'zero delay code' vector $\mathbf{g} = \mathbf{F}\mathbf{c}_0$ introduced in Section 5.1.1 has most of the power concentrated around the zero frequency, and the spectrum decays fast for higher frequencies. Only the N/P samples centered around the zero frequency are expected to have significant amplitude. In the computation of $\tilde{\mathbf{P}}$, we avoid divisions by small values by taking only the central N/P rows of $\tilde{\mathbf{P}}$ corresponding to the significant part of the vector $\mathbf{g}$, and discarding the other rows.

The estimated delay will be a rational number. For improved accuracy, we propose not to round it to the nearest sample instant, but rather to resample the signal after shifting it over $\hat{\tau}$. Using Shannon's resampling theorem, we can obtain new samples at exactly $\hat{\tau} + k\frac{T}{P}$, $k = 0, \cdots, LP-1$, where T is the symbol period, namely

$$r_k = \hat{z}_{intp}(\hat{\tau} + k\frac{T}{P}) = \sum_{n=0}^{N-1} (\hat{\mathbf{w}}^{\mathrm{H}}\mathbf{x}_n) \cdot \mathrm{sinc}(\frac{\hat{\tau} + (k-n)(T/P)}{T/P}),$$

where $k = 0, \ldots, LP-1$. These samples are used for detection.

5.2.2 Simulations–noninteger timing estimation

Because we now consider oversampling (above Nyquist), the setup of the simulations is a bit more delicate. In the following simulations, we used phase-modulated sources that employ Minimum Shift Keying (MSK) modulation. Even if phase modulation is a non-linear operation in general, an MSK-modulated signal can also be represented as a linearly modulated signal[1]

$$s(t) = d(t) * p(t) = \sum_{k=1}^{L} d_k \cdot p(t - kT) \ , \tag{5.12}$$

where the symbols d_k satisfy $d_k \in \{-1, 1\}$ for even k, $d_k \in \{-j, j\}$ for odd k, and the pulse shape is

$$p(t) = \begin{cases} \cos \frac{\pi}{2} \frac{t}{T} & -T \leq t < T \,, \\ 0 & \text{otherwise} \,. \end{cases}$$

Known modulus variations (the code) are inserted over the signal $s(t)$ as

$$z(t) \quad = \quad \textstyle\sum_{k=1}^{L} \sqrt{c_k} \cdot d_k \cdot p(t-k) = \sum_{k=1}^{L} z_k \cdot p(t-k) \,.$$

We consider $d = 2$ equal-power users transmitting data packets of equal size $L = 64$ symbols. The receiver has $M = 2$ antennas and employs $P = 2$ times oversampling. The analysis window is $N = 256$ samples, or 128 symbol periods. The delays of the two sources were set at $[17.65,\ 35.92]$.

In figure 5.5, the BER versus input SNR performance of the KMA beamformer for the cases with and without resampling via interpolation are presented. In the case without interpolation, the estimated τ is rounded to the nearest sample instant. For reference, we also show the performance of the Linear Minimum Mean Squared Error (MMSE) receiver, where the packet offsets τ_q and the transmitted sequences $z^{(q)}(t)$ are known for all users $q = 1, 2$, and the performance of an MMSE receiver in the case where no amplitude modulation code was inserted by the transmitters.

From the figure, we observe that the performance of KMA improves due to the interpolation; as before, at higher SNRs the performance comes close to that of the MMSE. We also observe a gap between the MMSE receiver for amplitude-modulated signals and unmodulated CM signals. The explanation is that for the modulated signal, the bits that happen to have less power $(1 - \epsilon)$ have less protection against noise than those that have higher power $(1 + \epsilon)$. This motivates the use of a small ϵ. On the other hand, figure 4.7 showed that ϵ should be sufficiently large to guarantee a good estimate of $\mathbf{w}$. Also the packet length L enters into this trade-off because this minimum value is inversely proportional to $\sqrt{L}$.

Since, with oversampling, we require that the pulse shape is a smooth analog function, we can compute the Cramer-Rao bound (CRB) of the estimation of τ for this case. The CRB specifies a lower bound on the variance of any unbiased estimator for τ and is derived later in Chapter 6. To verify the performance of the delay estimation algorithm, we performed a simulation using the same parameters as before. The Cramer-Rao Bound for the estimates of the packet offset is depicted in figure 5.6

[1] The linear modulation is introduced for simplicity, nonlinear modulations are certainly also applicable. By the results of Laurent [53], these can be approximated by a linear modulation.

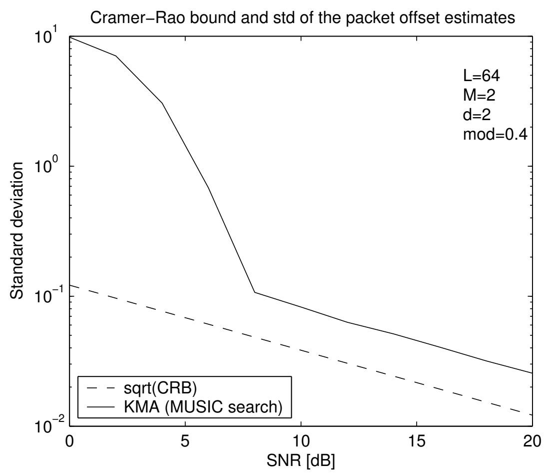


Figure 5.6: Case 2 beamformer performance (asynchronous sources with equal-length packets) where MSK modulation scheme is implemented: The standard deviation of estimated delays $\hat{\tau}$ using the MUSIC-type search, compared to the CRB.

with a dashed line. The solid line represents the standard deviation of the estimates of τ_1 for the user of interest. The gap between the CRB and the standard deviation of the MUSIC-type search algorithm (about a factor 2) shows that the algorithm is not efficient. This can be explained by the fact that the data is squared in the proposed algorithm, which essentially doubles the noise.

In figure 5.7 the packet offset recovery failure rate versus input SNR is presented. A delay estimate $\hat{\tau}$ is considered as good if $\tau - \frac{T}{2} < \hat{\tau} < \tau + \frac{T}{2}$ is satisfied. From the figure we can see that the performance of the synchronization scheme improves by increasing the modulation level ϵ. Increase of ϵ makes some symbols more susceptible to erroneous transmissions, in particular the ones that carry less energy $i.e.,$ have lower amplitude. For this reason, the best choice would be to use modulation levels in the vicinity of $\epsilon = 0.3$.

The conclusions and the discussion of the proposed scheme are given in Section 7.2.

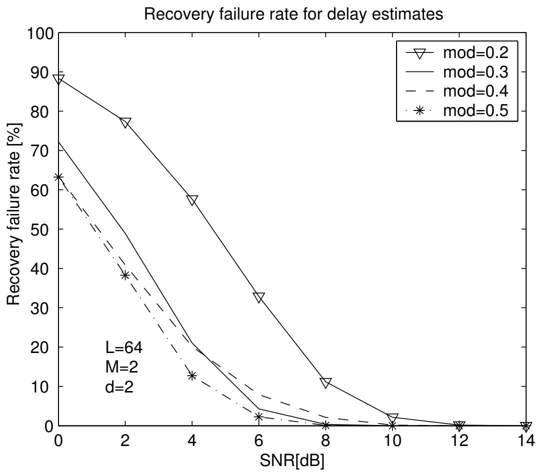


Figure 5.7: Case 2 beamformer performance (asynchronous sources with equal-length packets) where MSK modulation scheme is implemented: The delay offset failure rate for different values of the modulation level $\epsilon = [0.2, 0.3, 0.4, 0.5]$.

Chapter 6

Cramer-Rao bounds

In the previous chapter we have seen the importance of the packet offset estimation in asynchronous multiuser wireless ad hoc networks where users transmit data packets randomly, without any coordination. Synchronizing to the user of interest enables subsequent detection of transmitted data.

In the development of the algorithm we first considered a noiseless case. At that stage we proved that our algorithm provides a unique solution. Subsequently, the algorithm was analyzed and its performance tested in the presence of the white Gaussian noise. The presence of noise causes the deviation of the estimate from the exact solution. A Cramer-Rao bound represents the lower bound of the estimation error for any unbiased estimator. In addition the Cramer-Rao bound does not depend on the deployed algorithm but only on the data model. Comparing the deviation of our estimates to the Cramer-Rao bound demonstrates the quality of the estimator.

In this chapter we derive Cramer-Rao bounds for the packet offset estimation using blind (KMA) and training based algorithms and we compare their performance. Results were published in [46].

6.1 Data model

Similar to Chapter 4, we consider the scenario presented in figure 6.1 where different users transmit data packets simultaneously. In order to suppress the interference, an antenna array of M elements is introduced at the receiver side. We assume that the packet of the user of interest completely falls within the analysis window. If we consider a data packet to be of size L and the size of the analysis window N then the condition $N > L$ should be fulfilled.

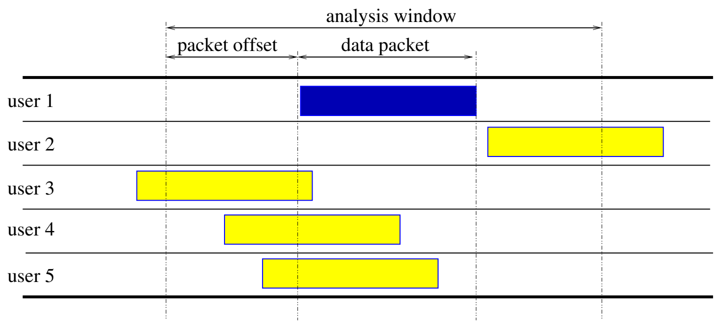


Figure 6.1: Structure of the transmitted data packets and the analysis window.

Here, we repeat the data model for the derivation of Cramer-Rao bound. For this purpose we highlight the assumptions used in our data model:

1. Noise samples are assumed to be white Gaussian with independent identically distribution and known variance σ^2. Noise samples are random variables.

2. The transmitted data symbols and the mixing matrix are considered deterministic but unknown, rather than stochastic.

3. In the blind packet offset estimation the unknown parameters are the array response vectors, the packet delays, and the data symbols of all users. The modulus variations are assumed known in this case.

4. In the training based estimation scheme the complete transmitted signal, $i.e.$, the data sequence modulated by the color code, is considered to be known.

The transmission is modeled by a linear data model of the form

$$\mathbf{x}(t) = \mathbf{A}\mathbf{z}_\tau(t) \; + \; \mathbf{n}(t)\,, \tag{6.1}$$

where as before, $\mathbf{x}(t) \in \mathbb{C}^{M\times 1}$ is the data vector received by the array of M antennas at time instant t, $\mathbf{A} = [\mathbf{a}^{(1)} \cdots \mathbf{a}^{(d)}]$ is the mixing matrix comprised of the signature vectors $\mathbf{a}^{(i)}$ for sources $i = 1, \cdots, d$. $\mathbf{z}_\tau(t) = [z^{(1)}(t-\tau^{(1)}), \cdots, z^{(d)}(t-\tau^{(d)})]^{\mathrm{T}}$ contains the data packets of all d users where each is shifted over $\tau^{(i)}$ with respect to the beginning of the analysis window. Here, (i) stands for the i-th user. In general, $\tau^{(i)}$ has a non-integer value. The transmitted symbol of the i-th user at time t is defined as $z^{(i)}(t) = \sqrt{c^{(i)}(t)}e^{j\phi^{(i)}(t)}$ where $e^{j\phi^{(i)}(t)}$ stands for a constant modulus data signal while $c^{(i)}(t) = 1 \pm \epsilon^{(i)}(t)$ is the known modulation code of the i-th user at time instant t. Define the vector $\boldsymbol{\phi}(t) = [\phi^{(1)}(t), \cdots, \phi^{(d)}(t)]^{\mathrm{T}}$ as the collection of the phase rotations of all users at time t. $\mathbf{n}(t)$ is a column vector of size M representing additive white Gaussian noise with covariance matrix $\mathbf{R}_{nn} = \sigma^2\mathbf{I}$. After sampling, our model becomes

$$\mathbf{x}_k = \mathbf{A}\mathbf{z}_k + \mathbf{n}_k \ , \quad k = 1, \ldots, N \tag{6.2}$$

where $\mathbf{x}_k = \mathbf{x}(kT/P)\,, \mathbf{z}_k = \mathbf{z}_\tau(kT/P)$ and $\mathbf{n}_k = \mathbf{n}(kT/P)$ with T and P representing the signal interval and oversampling rate, respectively. Typically $P = 2$. The number of symbols is N_s. After the oversampling the number of samples becomes $N = PN_s$.

Note that although $\mathbf{z}_k$ is a function of $\boldsymbol{\tau} = [\tau^{(1)}, \cdots, \tau^{(d)}]$, we omit it in the notation for simplicity reasons. Further we define

$$\boldsymbol{\phi}_k = \begin{bmatrix} \phi^{(1)}(kT/P) \\ \vdots \\ \phi^{(d)}(kT/P) \end{bmatrix} ,$$

$$\mathbf{c}_k = \begin{bmatrix} c^{(1)}(kT/P) \\ \vdots \\ c^{(d)}(kT/P) \end{bmatrix} ,$$

and $\mathbf{z}_k = \mathbf{f}(\mathbf{c}_k, \boldsymbol{\phi}_k)$ as

$$\mathbf{z}_k = \begin{bmatrix} z_k^{(1)} \\ \vdots \\ z_k^{(d)} \end{bmatrix} = \begin{bmatrix} c_k^{(1)} e^{j\phi_k^{(1)}} \\ \vdots \\ c_k^{(d)} e^{j\phi_k^{(d)}} \end{bmatrix}$$

6.2 Deterministic Cramer-Rao bound for blind packet offset estimation

The CRB specifies a lower bound on the variance of any unbiased estimator for τ. For simplicity, we assume that the data $\boldsymbol{\phi}_k$ and the mixing matrix $\mathbf{A}$ are deterministic rather than stochastic. Hence the only random variables are the noise samples, whereas the unknown parameters are $\mathbf{A}$, the packet delays $\tau^{(i)}$ of all $i = 1, \ldots, d$ users and $\boldsymbol{\phi}_k$.

6.2.1 Likelihood function

Based on the model and assuming N received samples collected in a matrix $\mathbf{X}$, we can derive the likelihood function. For deterministic Known Modulus (KM) signals (known modulation code c_k) in white Gaussian noise, the likelihood function is given by

$$L(\mathbf{X}|\sigma^2, \mathbf{\Phi}, \mathbf{a}, \boldsymbol{\tau}) = \frac{1}{(\pi\sigma^2)^{MN}} \exp\left\{ -\frac{1}{\sigma^2} \sum_{k=1}^{N} (\mathbf{x}_k - \mathbf{A}\mathbf{z}_k)^{\mathrm{H}} (\mathbf{x}_k - \mathbf{A}\mathbf{z}_k) \right\}, \quad (6.3)$$

where the real valued parameter vectors are

$$\boldsymbol{\phi} = \left[\boldsymbol{\phi}_1^T, \ldots, \boldsymbol{\phi}_N^T\right]^{\mathrm{T}} \quad (6.4)$$

and we define

$$\mathbf{a} := \left[\bar{\mathbf{a}}^{(1)T}, \tilde{\mathbf{a}}^{(1)T}, \cdots, \bar{\mathbf{a}}^{(d)T}, \tilde{\mathbf{a}}^{(d)T}\right]^{\mathrm{T}}, \quad (6.5)$$

where $\mathbf{a}$ stacks vertically the real and the imaginary parts of signatures vectors of $i = 1, \cdots, d$ users, $\bar{\mathbf{a}}^{(i)} = Re(\mathbf{a}^{(i)})$ and $\tilde{\mathbf{a}}^{(i)} = Im(\mathbf{a}^{(i)})$, respectively. Note that this notation differs from the one used up to this point where $\bar{\mathbf{a}}$ denoted the complex conjugate of a vector. As described in [54] the noise as a parameter can be observed separately from other parameters. For this reason we can neglect σ^2 as a parameter in (6.3). Let $\mathcal{L}(\mathbf{X}|\boldsymbol{\phi}, \mathbf{a}, \boldsymbol{\tau}) = \ln L(\mathbf{X}|\boldsymbol{\phi}, \mathbf{a}, \boldsymbol{\tau})$. After omitting constants we obtain the log-likelihood function as

$$\mathcal{L}(\mathbf{X}|\boldsymbol{\phi}, \mathbf{a}, \boldsymbol{\tau}) = -\frac{1}{\sigma^2} \sum_{k=1}^{N} \|\mathbf{x}_k - \mathbf{A}\mathbf{z}_k\|^2 . \quad (6.6)$$

6.2.2 Cramer-Rao Bound

The derivation of the CRB from the log-likelihood function is similar as in [54]. The CRB is determined by the main diagonal of the inverse of the Fisher Information Matrix (FIM). In turn, the FIM specifies the “sensitivity” of the log-likelihood function to changes in the parameters,

$$\mathbf{F}_N = E\left\{ \frac{\partial \mathcal{L}}{\partial \boldsymbol{\rho}} \cdot (\frac{\partial \mathcal{L}}{\partial \boldsymbol{\rho}})^T \right\}$$

where N is the number of samples, $E\{\cdot\}$ represents expectation and $\boldsymbol{\rho}$ is a real valued vector which collects all parameters,

$$\boldsymbol{\rho} = [\boldsymbol{\phi}^{\mathrm{T}},\ \mathbf{a}^{\mathrm{T}},\ \boldsymbol{\tau}^{\mathrm{T}}]^{\mathrm{T}} .$$

For the case we have, this can be worked out in closed form similar as in [55]. Partition the FIM as

$$\mathbf{F}_N = \begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} \\ \mathbf{F}_{21} & \mathbf{F}_{22} \end{bmatrix} \quad (6.7)$$

where the partitioning follows the partitioning of $\boldsymbol{\rho}$ into $\boldsymbol{\phi}$ followed by $[\mathbf{a}^{\mathrm{T}}, \boldsymbol{\tau}^{\mathrm{T}}]^{\mathrm{T}}$. Then

$$\mathbf{F}_{11} = \begin{bmatrix} \mathbf{H}_1 & & 0 \\ & \ddots & \\ 0 & & \mathbf{H}_N \end{bmatrix}, \tag{6.8}$$

$$\mathbf{F}_{21} = \left[\begin{array}{ccc} \bar{\boldsymbol{\Delta}}_1^{(1)} & \cdots & \bar{\boldsymbol{\Delta}}_N^{(1)} \\ \tilde{\boldsymbol{\Delta}}_1^{(1)} & \cdots & \tilde{\boldsymbol{\Delta}}_N^{(1)} \\ \vdots & & \vdots \\ \bar{\boldsymbol{\Delta}}_1^{(d)} & \cdots & \bar{\boldsymbol{\Delta}}_N^{(d)} \\ \tilde{\boldsymbol{\Delta}}_1^{(d)} & \cdots & \tilde{\boldsymbol{\Delta}}_N^{(d)} \\ \hline \mathbf{E}_1 & \cdots & \mathbf{E}_N \end{array}\right], \tag{6.9}$$

where the partitioning follows $[\mathbf{a}^{\mathrm{T}} \quad \boldsymbol{\tau}^{\mathrm{T}}]^{\mathrm{T}}$. Further we define

$$\mathbf{F}_{22} = \left[\begin{array}{cccccc|c} \bar{\boldsymbol{\Gamma}}^{(1)} & -\tilde{\boldsymbol{\Gamma}}^{(1)} & 0 & 0 & 0 & 0 & \bar{\boldsymbol{\Lambda}}^{(1)T} \\ \tilde{\boldsymbol{\Gamma}}^{(1)} & \bar{\boldsymbol{\Gamma}}^{(1)} & 0 & 0 & 0 & 0 & \tilde{\boldsymbol{\Lambda}}^{(1)T} \\ 0 & 0 & \ddots & & 0 & 0 & \vdots \\ 0 & 0 & & \ddots & 0 & 0 & \vdots \\ 0 & 0 & 0 & 0 & \bar{\boldsymbol{\Gamma}}^{(d)} & -\tilde{\boldsymbol{\Gamma}}^{(d)} & \bar{\boldsymbol{\Lambda}}^{(d)T} \\ 0 & 0 & 0 & 0 & \tilde{\boldsymbol{\Gamma}}^{(d)} & \bar{\boldsymbol{\Gamma}}^{(d)} & \tilde{\boldsymbol{\Lambda}}^{(d)T} \\ \hline \bar{\boldsymbol{\Lambda}}^{(1)} & \tilde{\boldsymbol{\Lambda}}^{(1)} & \cdots & \cdots & \bar{\boldsymbol{\Lambda}}^{(d)} & \tilde{\boldsymbol{\Lambda}}^{(d)} & \boldsymbol{\Upsilon} \end{array}\right], \tag{6.10}$$

and $\mathbf{F}_{12} = \mathbf{F}_{21}^{\mathrm{T}}$. The notation in the FIM matrix is such that the upper index (in braces) determines the user while the lower index stands for the discrete time instant. The elements of the FIM become (for the detailed derivation see Appendix at the end of this Chapter, $i.e.,$ Section 6.5)

$$
\begin{array}{llll}
\mathbf{H}_k & := & E\left\{\frac{\partial \mathcal{L}}{\partial \boldsymbol{\phi}_k}(\frac{\partial \mathcal{L}}{\partial \boldsymbol{\phi}_k})^{\mathrm{T}}\right\} & = \quad \frac{2}{\sigma^2} Re(\mathbf{Q}_k^{\mathrm{H}} \mathbf{A}^{\mathrm{H}} \mathbf{A} \mathbf{Q}_k) \\
\bar{\boldsymbol{\Delta}}_k^{(i)} & := & E\left\{\frac{\partial \mathcal{L}}{\partial \bar{\mathbf{a}}^{(i)}}(\frac{\partial \mathcal{L}}{\partial \boldsymbol{\phi}_k})^{\mathrm{T}}\right\} & = \quad -\frac{2}{\sigma^2} Im(z_k^{(i)*} \mathbf{A} \mathbf{Q}_k) \\
\tilde{\boldsymbol{\Delta}}_k^{(i)} & := & E\left\{\frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{a}}^{(i)}}(\frac{\partial \mathcal{L}}{\partial \boldsymbol{\phi}_k})^{\mathrm{T}}\right\} & = \quad \frac{2}{\sigma^2} Re(z_k^{(i)*} \mathbf{A} \mathbf{Q}_k) \\
\mathbf{E}_k & := & E\left\{\frac{\partial \mathcal{L}}{\partial \boldsymbol{\tau}}(\frac{\partial \mathcal{L}}{\partial \boldsymbol{\phi}_k})^{T}\right\} & = \quad -\frac{2}{\sigma^2} Im(\mathcal{Z}_k^{\mathrm{H}} \mathbf{A}^{\mathrm{H}} \mathbf{A} \mathbf{Q}_k) \\
\bar{\boldsymbol{\Gamma}}^{(i)} & := & E\left\{\frac{\partial \mathcal{L}}{\partial \bar{\mathbf{a}}^{(i)}}(\frac{\partial \mathcal{L}}{\partial \bar{\mathbf{a}}^{(i)}})^{T}\right\} & = \quad \frac{2}{\sigma^2} \sum_{k=1}^{N} Re(|z_k^{(i)}|^2 \mathbf{I}) \\
\tilde{\boldsymbol{\Gamma}}^{(i)} & := & E\left\{\frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{a}}^{(i)}}(\frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{a}}^{(i)}})^{T}\right\} & = \quad \frac{2}{\sigma^2} \sum_{k=1}^{N} Im(|z_k^{(i)}|^2 \mathbf{I}) \\
\bar{\boldsymbol{\Lambda}}^{(i)} & := & E\left\{\frac{\partial \mathcal{L}}{\partial \boldsymbol{\tau}}(\frac{\partial \mathcal{L}}{\partial \bar{\mathbf{a}}^{(i)}})^{T}\right\} & = \quad \frac{2}{\sigma^2} \sum_{k=1}^{N} Re(\mathcal{Z}_k^{\mathrm{H}} \mathbf{A}^{\mathrm{H}} z_k^{(i)}) \\
\tilde{\boldsymbol{\Lambda}}^{(i)} & := & E\left\{\frac{\partial \mathcal{L}}{\partial \boldsymbol{\tau}}(\frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{a}}^{(i)}})^{T}\right\} & = \quad -\frac{2}{\sigma^2} \sum_{k=1}^{N} Im(\mathcal{Z}_k^{\mathrm{H}} \mathbf{A}^{\mathrm{H}} z_k^{(i)}) \\
\boldsymbol{\Upsilon} & := & E\left\{\frac{\partial \mathcal{L}}{\partial \boldsymbol{\tau}}(\frac{\partial \mathcal{L}}{\partial \boldsymbol{\tau}})^{T}\right\} & = \quad \frac{2}{\sigma^2} \sum_{k=1}^{N} Re(\mathcal{Z}_k^{\mathrm{H}} \mathbf{A}^{\mathrm{H}} \mathbf{A} \mathcal{Z}_k)\,.
\end{array}
$$

Here * stands for the complex conjugate. The matrix $\mathcal{Z}_k$ is constructed as

$$
\mathcal{Z}_k = \text{diag}([\frac{\partial z^{(1)}(kT/P - \tau(1))}{\partial \tau(1)}, \cdots, \frac{\partial z^{(d)}(kT/P - \tau(d))}{\partial \tau(d)}])
$$

and $\mathbf{Q}_k := diag(\mathbf{z}_k)$. The detailed derivation of the elements of the FIM is presented in the Appendix at the end of this Chapter (Section 6.5). To give closed-form expressions for the inverse of the FIM, we use the following general result for block-partitioned matrices (which can be easily derived by inverting an LDU factorization). Let

$$
\left[\begin{array}{cc} \boldsymbol{\Xi}_{11} & \boldsymbol{\Xi}_{12} \\ \boldsymbol{\Xi}_{21} & \boldsymbol{\Xi}_{22} \end{array}\right] = \left[\begin{array}{cc} \sum_{k=1}^{N} \boldsymbol{\Delta}_k \mathbf{H}_k^{-1} \boldsymbol{\Delta}_k^T & \sum_{k=1}^{N} \boldsymbol{\Delta}_k \mathbf{H}_k^{-1} \mathbf{E}_k^T \\ \sum_{k=1}^{N} \mathbf{E}_k \mathbf{H}_k^{-1} \boldsymbol{\Delta}_k^T & \sum_{k=1}^{N} \mathbf{E}_k \mathbf{H}_k^{-1} \mathbf{E}_k^T \end{array}\right],
$$

where $\boldsymbol{\Delta}_k = [\bar{\boldsymbol{\Delta}}_k^{(1)T} \tilde{\boldsymbol{\Delta}}_k^{(1)T}, \cdots, \bar{\boldsymbol{\Delta}}_k^{(d)T} \tilde{\boldsymbol{\Delta}}_k^{(d)T}]^{\mathrm{T}}$ is of size $2Md \times d$ and let

$$
\boldsymbol{\Psi} = \left[\begin{array}{cc} \boldsymbol{\Gamma} & \boldsymbol{\Lambda}^T \\ \boldsymbol{\Lambda} & \boldsymbol{\Upsilon} \end{array}\right] - \left[\begin{array}{cc} \boldsymbol{\Xi}_{11} & \boldsymbol{\Xi}_{12} \\ \boldsymbol{\Xi}_{21} & \boldsymbol{\Xi}_{22} \end{array}\right], \tag{6.11}
$$

where $\boldsymbol{\Gamma}, \boldsymbol{\Lambda}, \boldsymbol{\Upsilon}$ are the corresponding blocks of the $\mathbf{F}_{22}$ matrix. The CRB on the parameters is given by the diagonal elements of $\mathbf{F}_N^{-1}$. Using the partitioned matrix inversion formula on $\boldsymbol{\Psi}^{-1}$, the CRB for parameter $\boldsymbol{\tau}$ is given by $CRB(\boldsymbol{\tau}) = (\boldsymbol{\Psi}^{-1})_{22}$, $i.e.,$

$$
CRB(\boldsymbol{\tau}) = \text{diag}\left[(\boldsymbol{\Upsilon} - \boldsymbol{\Xi}_{22}) - (\boldsymbol{\Lambda} - \boldsymbol{\Xi}_{21})(\boldsymbol{\Gamma} - \boldsymbol{\Xi}_{11})^{-1}(\boldsymbol{\Lambda}^T - \boldsymbol{\Xi}_{12})\right]^{-1}. \tag{6.12}
$$

6.3 Deterministic Cramer-Rao bound for training based packet offset estimates

6.3.1 Cramer-Rao Bound

In the previous section we derived the Cramer-Rao bound for the case in which only the modulus variations were known while the CM data symbols were considered unknown. In this section we derive the CRB of the estimates of $\boldsymbol{\tau}$ where all the CM data symbols and the superimposed training within a data packet are known and are used for synchronization. This will provide a lower bound on the variance of the training based packet offset estimates. Note that in the previous section only the modulus variations were known while data symbols are considered unknown. Again we discuss the deterministic CRB where noise samples are random variables while the unknowns are delays $\tau^{(i)}$ for all $i = 1, \cdots, d$ users and matrix $\mathbf{A}$.

If we write the model (6.2) in matrix form as $\mathbf{X}^{\mathrm{H}} = \mathbf{Z}_{\tau}^{\mathrm{H}}\mathbf{A}^{\mathrm{H}} + \mathbf{N}^{\mathrm{H}}$ where $\boldsymbol{\tau}$ is the unknown parameter vector that has to be estimated, then this is of the form $\mathbf{X} = \mathbf{A}_{\theta}\mathbf{S} + \mathbf{N}$ as described in direction finding literature [54, 56]. Here $\mathbf{Z}_{\tau}$ is the stack of $\mathbf{z}_k$ for all $k = 1, \cdots, N$. Taking the results from these papers and making the substitutions $\mathbf{A}_{\theta} \to \mathbf{Z}_{\tau}^{\mathrm{H}}$, $\mathbf{S} \to \mathbf{A}^{\mathrm{H}}$, $\frac{d}{d\theta}\mathbf{A} \to \frac{d}{d\boldsymbol{\tau}}\mathbf{Z}_{\tau}^{\mathrm{H}}$, we obtain that the CRB of the parameter vector $\boldsymbol{\tau}$ is given by

$$CRB(\boldsymbol{\tau}) = \frac{\sigma^2}{2}\left\{\sum_{i=1}^{M}\mathrm{Re}[\mathbf{Q}_i\mathbf{D}^{\mathrm{H}}[\mathbf{I}_N - \mathbf{Z}^{\mathrm{H}}(\mathbf{Z}\mathbf{Z}^{\mathrm{H}})^{-1}\mathbf{Z}]\mathbf{D}\mathbf{Q}_i^{\mathrm{H}}]\right\}^{-1},$$

where $\mathbf{A} = [a_{ij}]$, $\mathbf{Q}_i = \mathrm{diag}([a_{i1}, \cdots, a_{id}])$, and $\mathbf{D}$ is a matrix containing derivatives, as follows. Define a row vector $\dot{\mathbf{z}}^{(i)}$ of size N as

$$\dot{\mathbf{z}}^{(i)} = \frac{\partial z^{(i)}(kT/P - \tau^{(i)})}{\partial \tau^{(i)}} \; .$$

Matrix $\mathbf{D}$ now becomes

$$\mathbf{D} = [\dot{\mathbf{z}}^{(1)H}, \cdots, \dot{\mathbf{z}}^{(d)H}] \; : \quad N \times d \, . \tag{6.13}$$

6.3.2 Training based packet offset estimation algorithm

In the training based approach the whole packet of the transmitted data is assumed to be known at the receiver and is used in the estimation of the beginning of the upcoming packet carrying useful data information. The principle is similar to the blind synchronization scheme presented in Section 5.1.2 but now instead of using the matrix $\mathbf{P}$ (defined in (4.6)) we work directly on the matrix of the received data $\mathbf{X}$. We start from the data model $\mathbf{X}^{\mathrm{T}} = \mathbf{Z}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}} + \mathbf{N}^{\mathrm{T}}$. The training data of the user of interest within the analysis window can be modeled as

$$\tilde{\mathbf{z}}_\tau = [\underbrace{0, \cdots, 0}_{\tau},\ z_1^{(1)}, \cdots, z_L^{(1)},\ \underbrace{0, \cdots, 0}_{N-L-\tau}]^T$$

and corresponds to a column of the $\mathbf{Z}^{\mathrm{T}}$ matrix. As in the Section 5.1 we come to $\mathbf{F}\tilde{\mathbf{z}}_\tau = \mathbf{F}\tilde{\mathbf{z}}_0 \odot \boldsymbol{\varphi}_\tau$ where $\mathbf{F}\tilde{\mathbf{z}}_0 = \mathbf{F}\tilde{\mathbf{z}}_{\tau=0} := \mathbf{g}$. We now perform the decorrelation by the known vector $\mathbf{g}$ obtaining $\tilde{\mathbf{X}}^{\mathrm{T}} = (\mathrm{diag}(\mathbf{g}))^{-1}\mathbf{F}\mathbf{X}^{\mathrm{T}} = (\mathrm{diag}(\mathbf{g}))^{-1}\mathbf{X}_f^{\mathrm{T}}$. Note that $\boldsymbol{\varphi}_\tau$ is in the column span of $\tilde{\mathbf{X}}^{\mathrm{T}}$ and has a shift invariant structure. Performing a MUSIC-like search we obtain an estimate of the packet offset by maximizing the expression

$$\hat{\tau} = \underset{\tau}{\mathrm{argmax}}\ \|\boldsymbol{\varphi}_\tau^{\mathrm{H}}\tilde{\mathbf{X}}^{\mathrm{T}}\|^2 .$$

6.4 Simulations

To validate the derivation of the CRB for the packet offset estimates we performed a simulation where we considered the $d = 2$ user case with non-integer delays $[\tau(1) \quad \tau(1)] = [17.65 \quad 35.92]$. The signal period T is normalized to $T = 1$. Both sources are CM where the information is inserted in the phase rotation by means of minimum shift keying (MSK). The code modulus variation is $\epsilon = 0.4$, the data packet is of size $L_s = 64$ symbols, the analysis window is $N = 2PL_s$ wide where $P = 2$ is the oversampling rate. In the training based approach all $L_s = 64$ symbols are used as training. In the simulation plots we present the Cramer-Rao Bound for the estimates of packet offset for both training based (solid-circle line) and blind $i.e.$, Known Modulus (dashed line) approach. Solid and solid-asterisk lines show the standard deviation for the estimates of τ obtained using the algorithms proposed in Sections 5.2 and 6.3.2, respectively. The size of the data packets is kept equal for all simulation runs.

From the simulation plot in figure 6.2 we note that the training based packet delay estimates have a much lower CRB than the case where the 'color code' is used for synchronization. However, the efficiency of the latter scheme is higher as it does not possess the training overhead, $i.e.$, the user code is superimposed over the data sequence. At this point we have to note that our joint blind synchronization and beamformer estimation scheme provides offset estimates that do not reach the Cramer-Rao bound but that are sufficient for a reliable data recovery (see figure 5.7).

In figure 6.3 we present the square root of the estimated Cramer-Rao bound for both blind and training based algorithms for the estimation of the packet offset delay τ. The modulation level is varied in the range $\epsilon = [0.2\ 0.3\ 0.4\ 0.5]$. In the blind estimation case we see that the CRB gets lower for cases in which the code has a larger modulation level. In the case where training is used we can note that the modulation level does not affect the CRB.

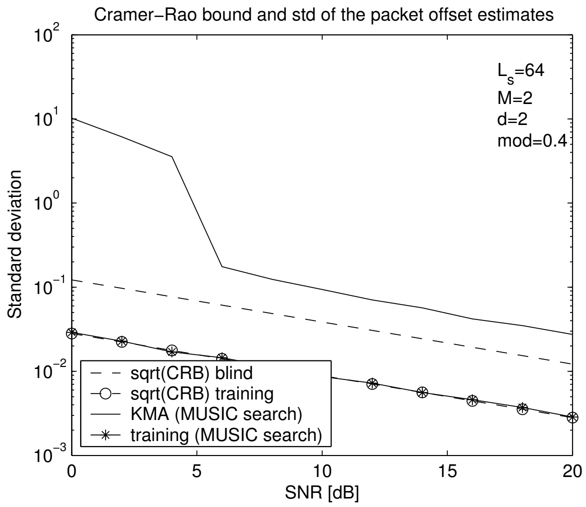


Figure 6.2: The square root of the CRB for blind and training based packet offset estimates and the standard deviation of the estimates of τ obtained using blind and training based algorithms

From the simulations performed we have learned that the training based packet offset estimation scheme provides estimates with smaller deviation. The training based offset estimates reach the Cramer-Rao bound while that is not the case for the KMA approach (figure 6.2). This is explained by the fact that the KMA performs a kind of 'squaring' of the received data samples as in (4.6). This step enhances the noise and thus increases the deviation of the estimates. However, the BER performance of the jointly estimated beamformer (figure 5.7) has only 3dB worse performance than the MMSE receiver. The advantage of the blind synchronization scheme is its efficiency as no data symbols are used for the training overhead.

6.5 Appendix–Derivation of the FIM

In this section we present detailed derivation of the elements of the Fisher Information Matrix (FIM) introduced in Section 6.2.2. Assumptions we make in the derivation of the CRB considering the AWG noise vector $\mathbf{n}_k$ are:

1. $E(\mathbf{n}_k) = \mathbf{0}$,

2. $E(\mathbf{n}_k \mathbf{n}_k^{\mathrm{T}}) = \mathbf{0}$,

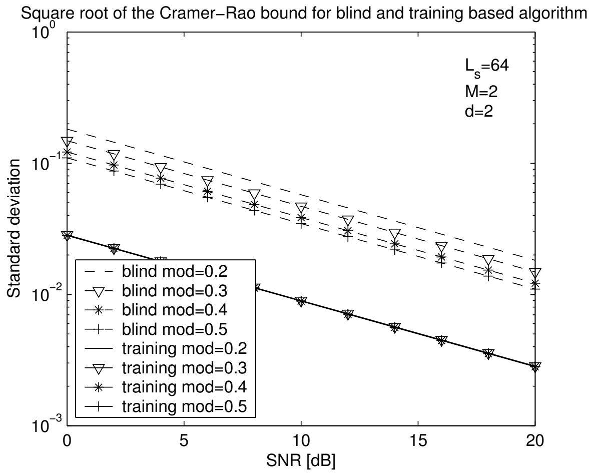


Figure 6.3: The square root of the CRB for blind and training based packet offset estimates involving different modulation levels $\epsilon = [0.2\ 0.3\ 0.4\ 0.5]$. Dashed and solid lines correspond to the blind and training based synchronization schemes.

3. $E(\mathbf{n}_k\mathbf{n}_k^{\mathrm{H}}) = \sigma^2\mathbf{I}$,

4. $E(\mathbf{n}_k\mathbf{n}_p^{\mathrm{H}}) = E(\mathbf{n}_k\mathbf{n}_p^{\mathrm{T}}) = \mathbf{0}$ for $k \neq p$, $i.e.$, the noise vectors are uncorrelated.

The following properties of the complex vector $(\mathbf{x}, \mathbf{y})$ multiplication are used in the computation [54]

$$\begin{array}{rcl} Re(\mathbf{x})Re(\mathbf{y}^{\mathrm{T}}) & = & \frac{1}{2}[Re(\mathbf{x}\mathbf{y}^{\mathrm{T}}) + Re(\mathbf{x}\mathbf{y}^{\mathrm{H}})] \\ Im(\mathbf{x})Im(\mathbf{y}^{\mathrm{T}}) & = & -\frac{1}{2}[Re(\mathbf{x}\mathbf{y}^{\mathrm{T}}) - Re(\mathbf{x}\mathbf{y}^{\mathrm{H}})] \\ Re(\mathbf{x})Im(\mathbf{y}^{\mathrm{T}}) & = & \frac{1}{2}[Im(\mathbf{x}\mathbf{y}^{\mathrm{T}}) - Im(\mathbf{x}\mathbf{y}^{\mathrm{H}})] \end{array}$$

Now we start taking derivatives of the log-likelihood function with respect to all parameters $(\boldsymbol{\phi}, \mathbf{a}, \boldsymbol{\tau})$. First we consider the derivative of log-likelihood function $\mathcal{L}$ with respect to the data vector at time instant k , $i.e.$, $\boldsymbol{\phi}_k$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\phi}_k} = \frac{\partial \bar{\mathbf{z}}_k}{\partial \boldsymbol{\phi}_k}\frac{\partial \mathcal{L}}{\partial \bar{\mathbf{z}}_k} + \frac{\partial \tilde{\mathbf{z}}_k}{\partial \boldsymbol{\phi}_k}\frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{z}}_k}\,,$$

where

$$\frac{\partial \mathcal{L}}{\partial \bar{\mathbf{z}}_k} = \frac{1}{\sigma^2}(\mathbf{A}^{\mathrm{H}}\mathbf{n}_k + \mathbf{A}^{\mathrm{T}}\mathbf{n}_k^*) = \frac{2}{\sigma^2}Re(\mathbf{A}^{\mathrm{H}}\mathbf{n}_k)\,,$$

$$\frac{\partial\mathcal{L}}{\partial\tilde{\mathbf{z}}_k} = \frac{1}{\sigma^2}(-j\mathbf{A}^{\mathrm{H}}\mathbf{n}_k + j\mathbf{A}^{\mathrm{T}}\mathbf{n}_k^*) = \frac{2}{\sigma^2}Im(\mathbf{A}^{\mathrm{H}}\mathbf{n}_k)\,,$$

and $\bar{\mathbf{z}}_k := Re(\mathbf{z}_k)$, $\tilde{\mathbf{z}}_k := Im(\mathbf{z}_k)$ and $j := \sqrt{-1}$.
Further,

$$\frac{\partial\bar{\mathbf{z}}_k}{\partial\boldsymbol{\phi}_k} = \mathrm{diag}[-c_k^{(1)}\sin(\phi_k^{(1)})\,,\cdots,\,-c_k^{(d)}\sin(\phi_k^{(d)})] = -Im(\mathbf{Q}_k)\,,$$

$$\frac{\partial\tilde{\mathbf{z}}_k}{\partial\boldsymbol{\phi}_k} = \mathrm{diag}[c_k^{(1)}\cos(\phi_k^{(1)})\,,\cdots,\,c_k^{(d)}\cos\phi_k^{(d)}] = Re(\mathbf{Q}_k)\,.$$

Finally we obtain:

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial\boldsymbol{\phi}_k} &= -\frac{2}{\sigma^2}Im(\mathbf{Q}_k)Re(\mathbf{A}^{\mathrm{H}}\mathbf{n}_k) + \frac{2}{\sigma^2}Re(\mathbf{Q}_k)Im(\mathbf{A}^{\mathrm{H}}\mathbf{n}_k) \\ &= \frac{2}{\sigma^2}Im(\mathbf{Q}_k^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{n}_k) \end{aligned}$$

Now we define elements of the matrix $\mathbf{F}_{11}$ via

$$\begin{aligned} \mathbf{H}_{kp} &:= E\left\{\frac{\partial\mathcal{L}}{\partial\boldsymbol{\phi}_k}(\frac{\partial\mathcal{L}}{\partial\boldsymbol{\phi}_p})^{\mathrm{T}}\right\} \\ &= \frac{4}{\sigma^4}E\left\{Im(\mathbf{Q}_k^H\mathbf{A}^{\mathrm{H}}\mathbf{n}_k)\left[Im(\mathbf{Q}_p^H\mathbf{A}^{\mathrm{H}}\mathbf{n}_p)\right]^{\mathrm{T}}\right\} = \\ &= \frac{4}{\sigma^4}\frac{1}{2}E\left\{Re(\mathbf{Q}_k^H\mathbf{A}^{\mathrm{H}}\mathbf{n}_k\mathbf{n}_p^H\mathbf{A}\mathbf{Q}_p)\right\} \\ &= \frac{2}{\sigma^2}Re(\mathbf{Q}_k^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{A}\mathbf{Q}_k)\delta_{k,p}\,, \end{aligned}$$

where $\delta_{k,p}$ is a Kronecker delta defined as

$$\delta_{k,p} = \begin{cases} 1 & if\;\; k = p \\ 0 & if\;\; k \neq p \end{cases}\quad.$$

The block matrix comprised of $\mathbf{H}_{kp}$ matrices is a diagonal matrix thus the structure of $\mathbf{F}_{11}$ presented in (6.8) follows from this property. Note that it may happen that some of the matrices on the main diagonal of $\mathbf{F}_{11}$, $i.e.$, $\mathbf{H}_{kk}$ could be zero, thus prevent the inversion of the Fisher information matrix. This arises from the fact that the analysis window is larger than the data packet, allowing some of the samples to be zero. The problem of the singularity of the FIM is solved by cancelling the columns and rows corresponding to zero diagonal elements of $\mathbf{F}_{11}$ leading to a 'constrained CRB' which is still a bound on the remaining parameters [57].

The derivative of the log-likelihood function with respect to $\tau^{(i)}$ for each user $i = 1\,,\ldots,d$ is

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial\tau^{(i)}} &= \frac{1}{\sigma^2}\sum_{k=1}^N\left\{(\mathbf{a}^{(i)}z_k^{(i)'})^{\mathrm{H}}\mathbf{n}_k + \mathbf{n}_k^H\mathbf{a}^{(i)}z_k^{(i)'}\right\} \\ &= \frac{2}{\sigma^2}\sum_{k=1}^N Re(z_k^{(i)'*}\mathbf{a}^{(i)H}\mathbf{n}_k) \end{aligned}$$

where $z_k^{(i)'} = \frac{\partial z^{(i)}(kT/P-\tau(i))}{\partial\tau(i)}$. Using the notation introduced in 6.2.2 yields

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\tau}} = \frac{2}{\sigma^2}\sum_{k=1}^N Re(\mathcal{Z}_k^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{n}_k)\,,$$

and

$$\begin{array}{rcl}\boldsymbol{\Upsilon} & := & E\left\{\frac{\partial\mathcal{L}}{\partial\boldsymbol{\tau}}(\frac{\partial\mathcal{L}}{\partial\boldsymbol{\tau}})^T\right\} \\ & = & \frac{4}{\sigma^4}E\left\{\sum_{k=1}^{N}Re(\mathcal{Z}_k^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{n}_k)\sum_{p=1}^{N}Re(\mathcal{Z}_p^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{n}_p)^{\mathrm{T}}\right\} \\ & = & \frac{4}{\sigma^4}E\left\{\sum_{k=1}^{N}\sum_{p=1}^{N}Re[\mathcal{Z}_k^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{n}_k\mathbf{n}_p\mathbf{A}\mathcal{Z}_p]\right\} \\ & = & \frac{2}{\sigma^2}\sum_{k=1}^{N}Re(\mathcal{Z}_k^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{A}\mathcal{Z}_k)\end{array}$$

The derivatives of $\mathcal{L}$ with respect to the real and imaginary part of the steering vector $\mathbf{a}$ are respectively

$$\frac{\partial\mathcal{L}}{\partial\bar{\mathbf{a}}^{(i)}} = \frac{1}{\sigma^2}\sum_{k=1}^{N}\left\{z_k^{(i)*}\mathbf{n}_k + z_k^{(i)}\mathbf{n}_k^*\right\} = \frac{2}{\sigma^2}\sum_{k=1}^{N}Re[z_k^{(i)*}\mathbf{n}_k]\,,$$

$$\frac{\partial\mathcal{L}}{\partial\tilde{\mathbf{a}}^{(i)}} = -\frac{1}{\sigma^2}\sum_{k=1}^{N}\left\{jz_k^{(i)*}\mathbf{n}_k - jz_k^{(i)}\mathbf{n}_k^*\right\} = \frac{2}{\sigma^2}\sum_{k=1}^{N}Im[z_k^{(i)*}\mathbf{n}_k]\,.$$

Note the fact that

$$\begin{array}{rcl}\bar{\boldsymbol{\Gamma}}^{(ij)} & := & E\left\{\frac{\partial\mathcal{L}}{\partial\bar{\mathbf{a}}^{(i)}}(\frac{\partial\mathcal{L}}{\partial\bar{\mathbf{a}}^{(j)}})^T\right\} \\ & = & \frac{4}{\sigma^4}E\left\{\sum_{k=1}^{N}Re(z_k^{(i)*}\mathbf{n}_k)(\sum_{p=1}^{N}Re(z_p^{(j)*}\mathbf{n}_p))^{\mathrm{T}}\right\} \\ & = & \mathbf{0}\end{array}$$

because $E(z_k^{(i)}z_k^{(j)}) = 0$. This property arises from the fact that the data symbols of two users i and j are uncorrelated. The same is valid for $\tilde{\boldsymbol{\Gamma}}^{(ij)}$, $i.e.$, $\tilde{\boldsymbol{\Gamma}}^{(ij)} = \mathbf{0}$. For this reason we concentrate only on the case $i = j$, $i.e.$, $\bar{\boldsymbol{\Gamma}}^{(i)}$ and $\tilde{\boldsymbol{\Gamma}}^{(i)}$ respectively:

$$\begin{array}{rcl}\bar{\boldsymbol{\Gamma}}^{(i)} & := & E\left\{\frac{\partial\mathcal{L}}{\partial\bar{\mathbf{a}}^{(i)}}(\frac{\partial\mathcal{L}}{\partial\bar{\mathbf{a}}^{(i)}})^T\right\} \\ & = & \frac{4}{\sigma^4}E\left\{\sum_{k=1}^{N}Re(z_k^{(i)*}\mathbf{n}_k)(\sum_{p=1}^{N}Re(z_p^{(i)*}\mathbf{n}_p))^{\mathrm{T}}\right\} \\ & = & \frac{2}{\sigma^2}\sum_{k=1}^{N}Re(|z_k^{(i)}|^2\mathbf{I}) \\ & = & E\left\{\frac{\partial\mathcal{L}}{\partial\tilde{\mathbf{a}}^{(i)}}(\frac{\partial\mathcal{L}}{\partial\tilde{\mathbf{a}}^{(i)}})^{\mathrm{T}}\right\}\end{array}$$

$$\begin{array}{rcl}\tilde{\boldsymbol{\Gamma}}^{(i)} & := & E\left\{\frac{\partial\mathcal{L}}{\partial\tilde{\mathbf{a}}^{(i)}}(\frac{\partial\mathcal{L}}{\partial\bar{\mathbf{a}}^{(i)}})^T\right\} \\ & = & \frac{4}{\sigma^4}E\left\{\sum_{k=1}^{N}Im(z_k^{(i)*}\mathbf{n}_k)(\sum_{p=1}^{N}Re(z_p^{(i)*}\mathbf{n}_p))^{\mathrm{T}}\right\} \\ & = & -\frac{2}{\sigma^2}\sum_{k=1}^{N}Im(|z_k^{(i)}|^2\mathbf{I}) \\ & = & -E\left\{\frac{\partial\mathcal{L}}{\partial\bar{\mathbf{a}}^{(i)}}(\frac{\partial\mathcal{L}}{\partial\tilde{\mathbf{a}}^{(i)}})^T\right\}\end{array}$$

Note that $\tilde{\boldsymbol{\Gamma}}^{(i)}$ is a zero $M \times M$ matrix because $Im(|z_k^{(i)}|^2) = 0$. The off-diagonal elements of the FIM matrix obtained as cross product of the derivatives of $\mathcal{L}$ with respect to different parameters are derived as

$$
\begin{array}{rcl}
\mathbf{E}_k & := & E\left\{\frac{\partial\mathcal{L}}{\partial\boldsymbol{\tau}}(\frac{\partial\mathcal{L}}{\partial\boldsymbol{\phi}_p})^T\right\} \\
& = & \frac{4}{\sigma^4}E\left\{\sum_{k=1}^{N} Re(\mathcal{Z}_k^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{n}_k)(Im(\mathbf{Q}_p^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{n}_p))^{\mathrm{T}}\right\} \\
& = & -\frac{2}{\sigma^2}Im(\mathcal{Z}_k^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{A}\mathbf{Q}_k)
\end{array}
$$

$$
\begin{array}{rcl}
\bar{\boldsymbol{\Delta}}_k^{(i)} & := & E\left\{\frac{\partial\mathcal{L}}{\partial\bar{\mathbf{a}}^{(i)}}(\frac{\partial\mathcal{L}}{\partial\boldsymbol{\phi}_k})^{\mathrm{T}}\right\} \\
& = & \frac{4}{\sigma^4}E\left\{\sum_{k=1}^{N} Re(z_k^{(i)*}\mathbf{n}_k)Im(\mathbf{Q}_p^{H}\mathbf{A}^{\mathrm{H}}\mathbf{n}_p)^{\mathrm{T}}\right\} \\
& = & -\frac{2}{\sigma^2}Im(z_k^{(i)*}\mathbf{A}\mathbf{Q}_k)
\end{array}
$$

$$
\begin{array}{rcl}
\tilde{\boldsymbol{\Delta}}_k^{(i)} & := & E\left\{\frac{\partial\mathcal{L}}{\partial\tilde{\mathbf{a}}^{(i)}}(\frac{\partial\mathcal{L}}{\partial\boldsymbol{\phi}_k})^{\mathrm{T}}\right\} \\
& = & \frac{4}{\sigma^4}E\left\{\sum_{k=1}^{N} Im(z_k^{(i)*}\mathbf{n}_k)Im(\mathbf{Q}_p^{H}\mathbf{A}^{\mathrm{H}}\mathbf{n}_p)^{\mathrm{T}}\right\} \\
& = & \frac{2}{\sigma^2}Re(z_k^{(i)*}\mathbf{A}\mathbf{Q}_k)
\end{array}
$$

$$
\begin{array}{rcl}
\bar{\boldsymbol{\Lambda}}^{(i)} & := & E\left\{\frac{\partial\mathcal{L}}{\partial\boldsymbol{\tau}}(\frac{\partial\mathcal{L}}{\partial\bar{\mathbf{a}}^{(i)}})^T\right\} \\
& = & \frac{4}{\sigma^4}E\left\{\sum_{k=1}^{N} Re(\mathcal{Z}_k^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{n}_k)\sum_{p=1}^{N} Re(z_p^{(i)*}\mathbf{n}_p)^{\mathrm{T}}\right\} \\
& = & \frac{2}{\sigma^2}\sum_{k=1}^{N} Re(\mathcal{Z}_k^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}z_k^{(i)})
\end{array}
$$

$$
\begin{array}{rcl}
\tilde{\boldsymbol{\Lambda}}^{(i)} & := & E\left\{\frac{\partial\mathcal{L}}{\partial\boldsymbol{\tau}}(\frac{\partial\mathcal{L}}{\partial\tilde{\mathbf{a}}^{(i)}})^T\right\} \\
& = & \frac{4}{\sigma^4}E\left\{\sum_{k=1}^{N} Re(\mathcal{Z}_k^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{n}_k)\sum_{p=1}^{N} Im(z_p^{(i)*}\mathbf{n}_p)^{\mathrm{T}}\right\} \\
& = & -\frac{2}{\sigma^2}\sum_{k=1}^{N} Im(\mathcal{Z}_k^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}z_k^{(i)})
\end{array}
$$

To give a closed-form expression for the inverse of the FIM, we use the following general result for block-partitioned matrices (which can be easily derived by inverting an LDU factorization):

$$
\begin{bmatrix}\mathbf{F}_{11} & \mathbf{F}_{12}\\ \mathbf{F}_{21} & \mathbf{F}_{22}\end{bmatrix}^{-1} =
$$

$$
\begin{bmatrix}\mathbf{I} & -\mathbf{F}_{11}^{-1}\mathbf{F}_{12}\\ 0 & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{F}_{11}^{-1} & \\ & (\mathbf{F}_{22}-\mathbf{F}_{21}\mathbf{F}_{11}^{-1}\mathbf{F}_{12})^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{I} & 0\\ -\mathbf{F}_{21}\mathbf{F}_{11}^{-1} & \mathbf{I}\end{bmatrix}.
$$

Thus let

$$
\left[\begin{array}{cc}\boldsymbol{\Xi}_{11} & \boldsymbol{\Xi}_{12}\\ \boldsymbol{\Xi}_{21} & \boldsymbol{\Xi}_{22}\end{array}\right] :=
$$

$$
= \mathbf{F}_{21}\mathbf{F}_{11}^{-1}\mathbf{F}_{12} = \left[\begin{array}{cc}\sum_{k=1}^{N}\boldsymbol{\Delta}_k\mathbf{H}_k^{-1}\boldsymbol{\Delta}_k^T & \sum_{k=1}^{N}\boldsymbol{\Delta}_k\mathbf{H}_k^{-1}\mathbf{E}_k^T\\ \sum_{k=1}^{N}\mathbf{E}_k\mathbf{H}_k^{-1}\boldsymbol{\Delta}_k^T & \sum_{k=1}^{N}\mathbf{E}_k\mathbf{H}_k^{-1}\mathbf{E}_k^T\end{array}\right],
$$

where $\boldsymbol{\Delta}_k = [\bar{\boldsymbol{\Delta}}_k^{(1)T} \tilde{\boldsymbol{\Delta}}_k^{(1)T}, \cdots, \bar{\boldsymbol{\Delta}}_k^{(d)T} \tilde{\boldsymbol{\Delta}}_k^{(d)T}]^{\mathrm{T}}$ is the of size $2Md \times d$ and

$$\boldsymbol{\Psi} := \mathbf{F}_{22} - \mathbf{F}_{21}\mathbf{F}_{11}^{-1}\mathbf{F}_{12} = \begin{bmatrix} \boldsymbol{\Gamma} & \boldsymbol{\Lambda}^T \\ \boldsymbol{\Lambda} & \boldsymbol{\Upsilon} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\Xi}_{11} & \boldsymbol{\Xi}_{12} \\ \boldsymbol{\Xi}_{21} & \boldsymbol{\Xi}_{22} \end{bmatrix}$$

so that

$$\begin{aligned}
(\mathbf{F}_N^{-1})_{11} &= \mathbf{F}_{11}^{-1} + \mathbf{F}_{11}^{-1}\mathbf{F}_{12}(\mathbf{F}_{22} - \mathbf{F}_{21}\mathbf{F}_{11}^{-1}\mathbf{F}_{12})^{-1}\mathbf{F}_{21}\mathbf{F}_{11}^{-1} \\
&= \begin{bmatrix} \mathbf{H}_1^{-1} & & 0 \\ & \ddots & \\ 0 & & \mathbf{H}_N^{-1} \end{bmatrix} + \\
&\quad \begin{bmatrix} \mathbf{H}_1^{-1}\boldsymbol{\Delta}_1^T \, \mathbf{H}_1^{-1}\mathbf{E}_1^T \\ \vdots \quad\quad \vdots \\ \mathbf{H}_N^{-1}\boldsymbol{\Delta}_N^T \mathbf{H}_N^{-1}\mathbf{E}_N^T \end{bmatrix} \boldsymbol{\Psi}^{-1} \begin{bmatrix} \boldsymbol{\Delta}_1\mathbf{H}_1^{-1}, \cdots, \boldsymbol{\Delta}_N\mathbf{H}_N^{-1} \\ \mathbf{E}_1\mathbf{H}_1^{-1}, \cdots, \mathbf{E}_N\mathbf{H}_N^{-1} \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
(\mathbf{F}_N^{-1})_{12} &= -\mathbf{F}_{11}^{-1}\mathbf{F}_{12}(\mathbf{F}_{22} - \mathbf{F}_{21}\mathbf{F}_{11}^{-1}\mathbf{F}_{12})^{-1} \\
&= -\begin{bmatrix} \mathbf{H}_1^{-1}\boldsymbol{\Delta}_1^T & \mathbf{H}_1^{-1}\mathbf{E}_1^T \\ \vdots & \vdots \\ \mathbf{H}_N^{-1}\boldsymbol{\Delta}_N^T & \mathbf{H}_N^{-1}\mathbf{E}_N^T \end{bmatrix} \boldsymbol{\Psi}^{-1} \\
(\mathbf{F}_N^{-1})_{21} &= -(\mathbf{F}_{22} - \mathbf{F}_{21}\mathbf{F}_{11}^{-1}\mathbf{F}_{12})^{-1}\mathbf{F}_{21}\mathbf{F}_{11}^{-1} \\
&= -\boldsymbol{\Psi}^{-1} \begin{bmatrix} \boldsymbol{\Delta}_1\mathbf{H}_1^{-1} & ,\cdots, & \boldsymbol{\Delta}_N\mathbf{H}_N^{-1} \\ \mathbf{E}_1\mathbf{H}_1^{-1} & ,\cdots, & \mathbf{E}_N\mathbf{H}_N^{-1} \end{bmatrix} \\
(\mathbf{F}_N^{-1})_{22} &= (\mathbf{F}_{22} - \mathbf{F}_{21}\mathbf{F}_{11}^{-1}\mathbf{F}_{12})^{-1} = \boldsymbol{\Psi}^{-1} \\
&= \left(\begin{bmatrix} \boldsymbol{\Gamma} & \boldsymbol{\Lambda}^T \\ \boldsymbol{\Lambda} & \boldsymbol{\Upsilon} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\Xi}_{11} & \boldsymbol{\Xi}_{12} \\ \boldsymbol{\Xi}_{21} & \boldsymbol{\Xi}_{22} \end{bmatrix} \right)^{-1} \qquad (6.14)
\end{aligned}$$

The CRB on the parameters is given by the diagonal elements of $\mathbf{F}_N^{-1}$. Using the partitioned matrix inversion formula again on (6.14), the CRB for parameters $\mathbf{a}$ and $\boldsymbol{\tau}$ follows as

$$CRB_N(\mathbf{a}) = \text{diag}(\boldsymbol{\Psi}^{-1})_{11}$$

$$= \text{diag}\left[(\boldsymbol{\Gamma} - \boldsymbol{\Xi}_{11}) - (\boldsymbol{\Lambda}^T - \boldsymbol{\Xi}_{12})(\boldsymbol{\Upsilon} - \boldsymbol{\Xi}_{22})^{-1}(\boldsymbol{\Lambda} - \boldsymbol{\Xi}_{21})\right]^{-1} ,$$

and

$$CRB_N(\boldsymbol{\tau}) = \text{diag}(\boldsymbol{\Psi}^{-1})_{22}$$

$$= \text{diag}\left[(\boldsymbol{\Upsilon} - \boldsymbol{\Xi}_{22}) - (\boldsymbol{\Lambda} - \boldsymbol{\Xi}_{21})(\boldsymbol{\Gamma} - \boldsymbol{\Xi}_{11})^{-1}(\boldsymbol{\Lambda}^T - \boldsymbol{\Xi}_{12})\right]^{-1} .$$

Similarly, the bound on the estimation variance of the signal phases follows as

$$CRB_N(\boldsymbol{\phi}_k) = \mathrm{diag}\left\{\mathbf{H}_k^{-1}\left[\mathbf{I} + [\boldsymbol{\Delta}_k^T \quad \mathbf{E}_k^T]\boldsymbol{\Psi}^{-1}\left[\begin{array}{c}\boldsymbol{\Delta}_k \\ \mathbf{E}_k\end{array}\right]\mathbf{H}_k^{-1}\right]\right\} .$$

Chapter 7

Experimental results and conclusions

7.1 TUDelft MIMO experimental platform

We illustrate the performance of AKMA on realistic channels using a MIMO testbed. The experimental MIMO platform "MURX" developed at TU Delft by Gerrard Janssen consists of three transmit and four receive antennas. MURX is not a real-time processing system but modulates, transmits and receives data frames that are stored for offline processing. The structure of the transceiver is presented in figure 7.1.

Data frames generated in Matlab are transferred to DACs with 10 bit resolution and then modulated to RF at a carrier of 813 MHz. On the receiver side the signal is down-modulated to baseband (25 MHz bandwidth), sampled at a rate of 50 MSPS (mega samples per second) with an accuracy of 10 bits and stored in memory for offline processing. Frequency up and down converters are coupled with each other and run on one central oscillator (813 MHz). This reduces the frequency deviation and phase errors to a minimum. On the other hand this is the point where the experimental setup differs from real-life as the cases with residual carriers are not taken into account. Transmit as well as receive antennas are placed on a movable platform which allows different propagation environments to be studied.

7.1.1 Structure of transmitted data bursts

Constant modulus (QPSK) data symbols s_k^q of user $q = 1, \cdots, d$ are modulated by a small known modulus amplitude modulation producing $z_k^{(q)}$. Although MURX sup-

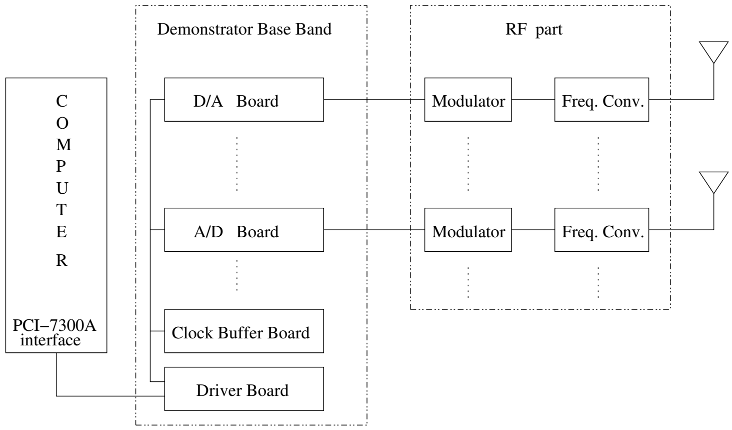


Figure 7.1: The transceiver structure of the experimental MIMO platform–MURX.

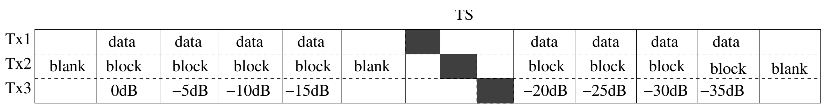


Figure 7.2: Structure of transmitted data frame.

ports higher data rates the symbols are transmitted at a rate of 1MSPS by extending each symbol using a rectangular pulse shape consisting of 50 chips. This time domain "extension" is used for all transmitted symbols. The structure of the transmitted frame is presented in figure 7.2, where each row in figure 7.2 determines one user. $Blank$ blocks (no data) of length 60 symbols at the beginning, in the middle and at the end of the transmitted frame are used in the estimation of the noise covariance matrix $\mathbf{R}_n$. Good estimation of $\mathbf{R}_n$ is important because it is used in the prewhitening step (7.1) shown later. Block TS contains training sequences used in the estimation of the array response matrix $\mathbf{A}$ as in (7.2). Each dark-colored training segment represents 100 QPSK symbols while in the $blank$ parts zeros are transmitted. In a $data\ block$ all three users transmit their data synchronously. The first data block has unit average power while the average power of each following block is decreased with 5dB so that we achieve a range of SNRs for the same channel realization. Altogether there will be eight data blocks. Every data block consists of five random data packets whose lengths are 50 symbols each. All Mean Square Error results are obtained after averaging over these five data packets. The sampling frequency at the receiver and transmitter side is the same. The receiver samples at 50 MSPS, and averages over blocks of 50 samples to reduce the sampling rate to the symbol rate (this implements a matched filter over the rectangular pulse). This choice is made to avoid the crosstalk of oscillator signals at 25MHz. In addition we want to obtain a narrowband model that is achieved for the propagation delays lower than 100ns. To achieve this the bandwidth of the signal needs to be lower than 10MHz. The resulting samples are collected in the data matrix $\mathbf{X}$.

7.1.2 Data model in the experiment

The experiment took place in a typical TU Delft office (4×6m). Considering the carrier frequency, bandwidth, duration of the transmitted frame (around 2ms), as well as the size of the office itself, it is reasonable to model the channel as instantaneous and stationary over the frame. The general data model used throughout the experiment was

$$\mathbf{X} = \mathbf{A}\mathbf{Z} + \mathbf{N} \,,$$

where $\mathbf{X} : M \times N$ represents N received data samples on the antenna array of M elements. $\mathbf{A} : M \times d$ denotes array signature matrix for d sources, $\mathbf{Z} : d \times N$ denotes d CM transmitted data sequences over which the known modulus code is superimposed and $\mathbf{N} : M \times N$ collects the noise samples at the antenna array.

Initially we have estimated the noise covariance matrix $\mathbf{R}_n$ from the blocks denoted by "blanked" (see figure 7.2) that do not carry any data. From the estimate of $\mathbf{R}_n$ we observed that the noise is not white, $i.e.$, $\hat{\mathbf{R}}_n \neq \sigma_n^2\mathbf{I}$. Recall that the rank detection using the eigenvalues of the data covariance matrix (or the SVD of $\mathbf{X}$) is only valid if the noise is white. Thus, a prewhitening step is performed, such that

$$\hat{\mathbf{R}}_n^{-\frac{1}{2}}\mathbf{X} = (\hat{\mathbf{R}}_n^{-\frac{1}{2}}\mathbf{A})\mathbf{Z} + \hat{\mathbf{R}}_n^{-\frac{1}{2}}\mathbf{N} \Leftrightarrow \underline{\mathbf{X}} = \underline{\mathbf{A}\mathbf{Z}} + \underline{\mathbf{N}} \,. \tag{7.1}$$

After this step our model fits the model defined in (4.3) and the processing steps in the source separation algorithms proposed in chapter 4 are followed (we drop the underscore from the notation for simplicity). The "true" values of each of the d columns of the $\mathbf{A}$ matrix after prewhitening are estimated based on the corresponding non-overlapping training sequences as

$$\mathbf{A}_{est} = \mathbf{X}_{TS}\mathbf{Z}_{TS}^{\dagger} \tag{7.2}$$

where $\mathbf{X}_{TS}$ and $\mathbf{Z}_{TS}$ represent the received and transmitted data, respectively in the regions that contain the training only (dark blocks in figure 7.2).

Experimental performance

The performance of the KMA algorithm is determined by the computation of the output SINR,

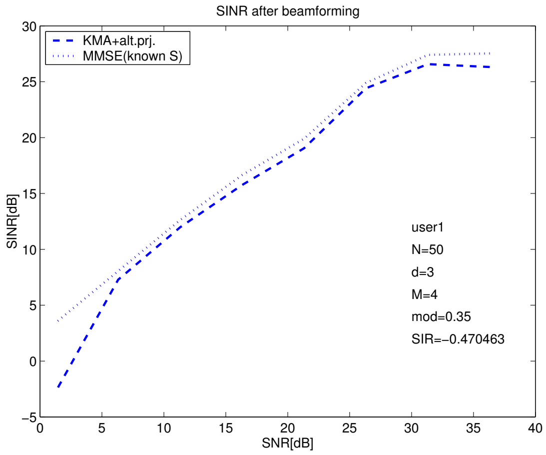
$$SINR = 10\log(\frac{P_s}{P_{in}}) = 10\log(\frac{\mathbf{w}^{\mathrm{H}}\mathbf{A}_{est}\sigma_s^2\mathbf{A}_{est}^{\mathrm{H}}\mathbf{w}}{\mathbf{w}^{\mathrm{H}}(\mathbf{A}_{est}\sigma_s^2\mathbf{A}_{est}^{\mathrm{H}} - \mathbf{a}_{est,j}\sigma_s^2\mathbf{a}_{est,j}^{\mathrm{H}} + \sigma_n^2\mathbf{I})\mathbf{w}}) \tag{7.3}$$

where σ_s^2, σ_n^2 denote the power of the transmitted signal and the noise power, respectively. A column vector of M elements $\mathbf{a}_{est,j}$ denotes the jth column of the estimate of the mixing matrix $\mathbf{A}_{est}$. P_s and P_{in} stands for the power of the user of interest and interference plus noise power, respectively. The Signal to Interference Ratio (SIR) is defined as power of the signal of interest (P_s) divided by the sum of all the other users' powers (P_i) at the output of the receiver

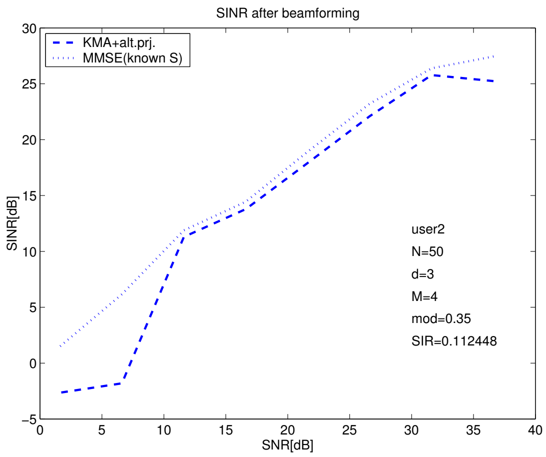
$$SIR = 10\log(\frac{P_s}{P_i}) = 10\log(\frac{\mathbf{a}_{est,j}^{\mathrm{H}}\mathbf{a}_{est,j}}{\sum_{i=1,i\neq j}^{d}\mathbf{a}_{est,i}^{\mathrm{H}}\mathbf{a}_{est,i}}) \ .$$

Figures 7.3 (a) and 7.3 (b) present the SINR of users 1 and 2 after beamforming, respectively. The algorithms which are tested are the KMA described in figure 4.4 followed by alternating projections described in (4.4). Note that the performance of the KMA beamformer is close to that of the MMSE beamformer. In fact the Matlab simulations presented in figure 4.5 are remarkably similar to the experimental results for the SINR after beamforming presented in figure 7.3.

Another algorithm performance indicator is the output Mean Squared Error, computed as $MSE^{(q)} = \frac{1}{N}\|\mathbf{s}^q - \hat{\mathbf{s}}^q\|_F^2$, where $\hat{\mathbf{s}}^q$ is the estimated data vector $(1 \times N)$ of user q at the output of the beamformer. The advantage of this criterion is that it does not depend on the estimated spatial signature matrix. Figures 7.4 (a) and 7.4 (b) show that the MSE is decreasing as the input SNR decreases, with an error floor at higher SNRs. The third user turned out to have $SIR = -16$dB, and was to weak to be extracted using this experimental set-up (10 bit accuracy, or 1024 levels, is too small to detect the small amplitude variations in a signal with small power). The low resolution of the ADCs is also responsible for the saturation of the output SINR and MSE for higher input SNR.

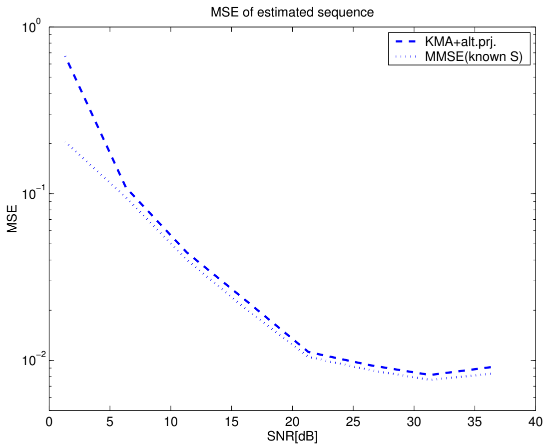


(a)

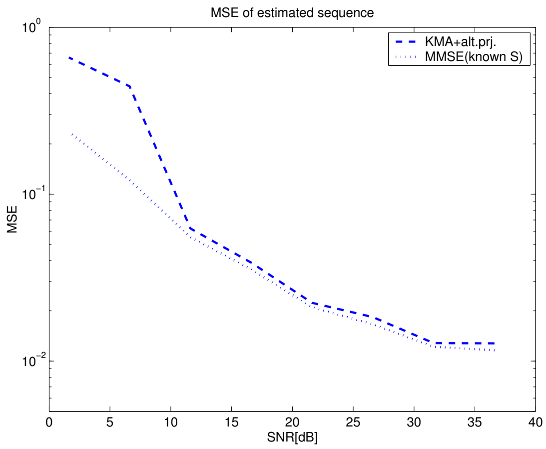


(b)

Figure 7.3: Beamformer performance (experimental): (a) SINR of user 1 after beamforming; (b) SINR of user 2 after beamforming.



(a)



(b)

Figure 7.4: Beamformer performance (experimental): (a) MSE of user 1 after beamforming; (b) MSE of user 2 after beamforming.

7.2 Concluding remarks

In the first part of the thesis, we considered an ad hoc network where users are unsynchronized and send packets randomly. An antenna array is used by the receiver to focus on the user of interest and suppress unwanted interfering packets. We introduced the KMA algorithms to jointly estimate the timing offset and separating beamformer of the packet of a user of interest. The scheme instantaneously provides high resolution packet offset estimates by processing the block of data in the frequency domain. The algorithm recognizes the user of interest from his 'color code', an amplitude modulation of the constant modulus data signal. This color code does not increase the bandwidth of the signal, and the amplitude modulation can be small (e.g., 25% or 1 dB) if the packet length is sufficiently large (e.g., 50 symbols). The complexity of our joint synchronization and detection scheme is reduced compared to the existing superimposed training schemes as the exhaustive search for the packet offset is avoided.

An advantage of the scheme is that it can be used as an upgrade to existing WLAN services or ad hoc networks (provided they use constant modulus signals). A transmitter employing a color code will have an advantageous reception at a KMA receiver, but will also be compatible with legacy receivers since the modulation index is small (around the noise level for sufficiently large packets). The algorithms can tolerate frequency-hopping schemes, since non-stationary interfering signals $i.e.,$ randomly appearing packets are admitted.

Some open issues are the following. The algorithm assumed flat-fading channels. It may be extended to OFDM systems, but we did not investigate this. We rather focused on the development of a joint packet offset estimation and detection scheme– a topic not considered by other researchers up to this moment. Another issue is that the ESPRIT-based timing estimators do not have so good performance and are rather complicated. Further, the algorithm that we propose varies the modulus of the CM signal. In this manner symbols with the reduced modulus are more susceptible to errors than the symbols with the increased modulus. However, from figure 5.5 we can see that the BER of our joint synchronization and detection scheme deviates only 3dB from the training based scheme without the variation of symbols' modulus.

Part II

Ultra Wideband Systems

Chapter 8

Ultra Wideband Communication Systems

In the first part of the thesis we have developed a narrowband multiple-access synchronization and detection scheme able to resolve the packet collision and retransmission problem in ad hoc networks. This is achieved by deployment of an antenna array at the receiver side. In contrast to this approach, in Part II we consider a broadband communication system that can accommodate a large number of simultaneously active users due to its extreme bandwidth. Each user is assigned a code that spreads the signal to a very low power level. This enables the coexistence of such a broadband system with existing narrrowband systems. In this part we will develop a synchronization and detection scheme for such a broadband system.

8.1 Ultra wideband systems

Ultra wideband (UWB) system is any broadband system which bandwidth exceeds 500MHz. For its great bandwidth UWB has recently generated interest as a candidate of a system that can provide high data rate communications (up to 100Mbps) on short distances (order of 10m). The original idea was to define an extremely wide frequency band for unlicensed transmission. To provide the coexistence with already deployed narrowband systems UWB systems need to transmit at extremely low power levels. This issue was first regulated by the Federal Communications Commition (FCC) in the United States. On February 14, 2002 FCC approved a spectral mask for operation of UWB devices. The major part of it lies between 3.1 and 10.6GHz with allowed effective isotropically radiated power (EIRP) of -41.3 dBm/MHz (see figure 8.1). The decision of the European Communication Commission (ECC) regarding the spectral

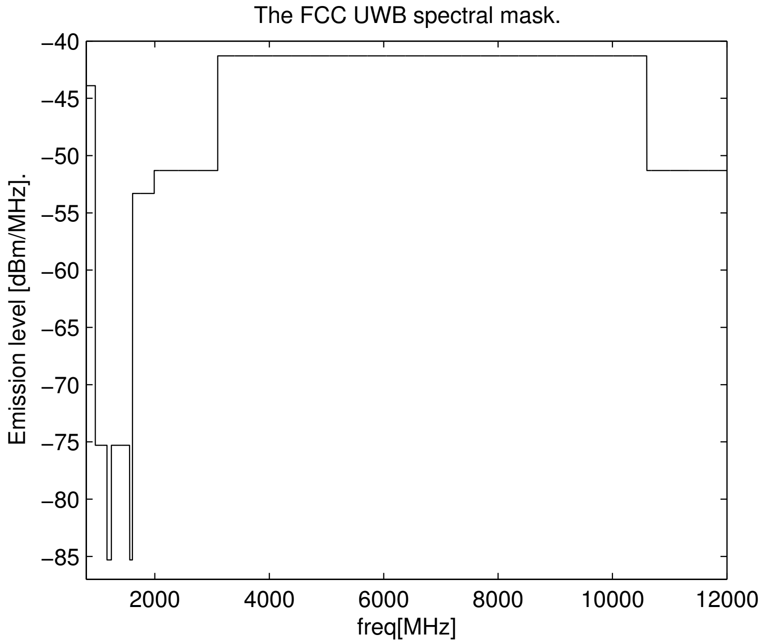


Figure 8.1: The spectral mask for UWB communication systems defined by the FCC in the USA.

management dated on $24.03.2006$ has kept even lower levels below the 4.8GHz for devices using Ultra-Wideband (UWB) technology (figure 8.2).

With respect to the communication techniques applicable in UWB systems two main directions can be distinguished. The first approach is known as Impulse-Radio (IR) and it is based on the transmission of extremely short pulses (shorter than a ns). Such short pulses cover the complete UWB spectrum. The activities of industrial organizations and academia in the field of IR-UWB resulted in the IEEE $802.15.4$a UWB standard.

The second approach supported by the Multiband OFDM Alliance (MBOA) combines the OFDM modulation technique with a multiband approach [58, 59]. The 7.5GHz band is divided into the subbands of approximately 500MHz that is further divided into a number of simultaneously transmitted sub-carriers. In addition, a hopping of OFDM symbols across the subbands is deployed in order to enable the coexistence of multiple users, $i.e.,$ reduce the possibility of catastrophic collisions. The development of this UWB design technique evolved in a IEEE $802.15.3$a standard.

One of the biggest advantages of the IR-UWB systems is that the transceiver de-

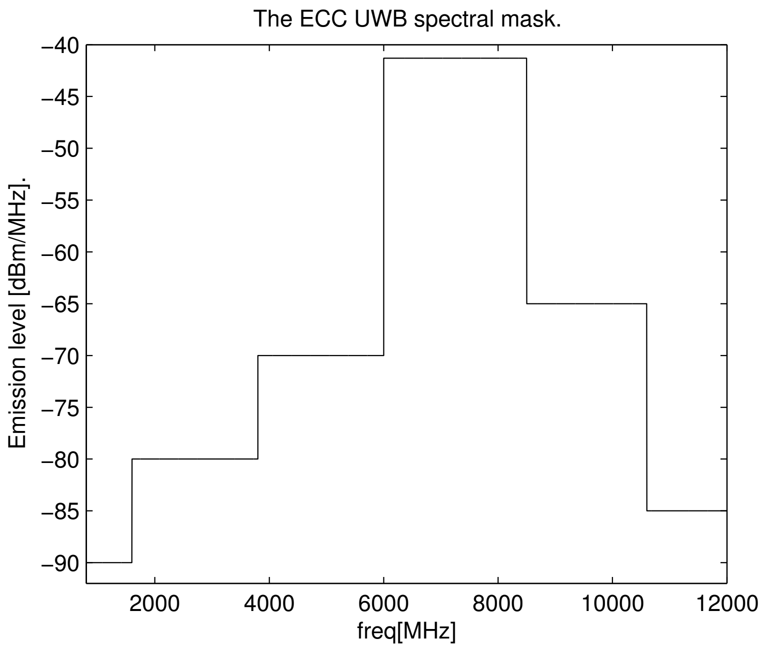


Figure 8.2: The actual proposal for the spectral management of UWB systems in Europe.

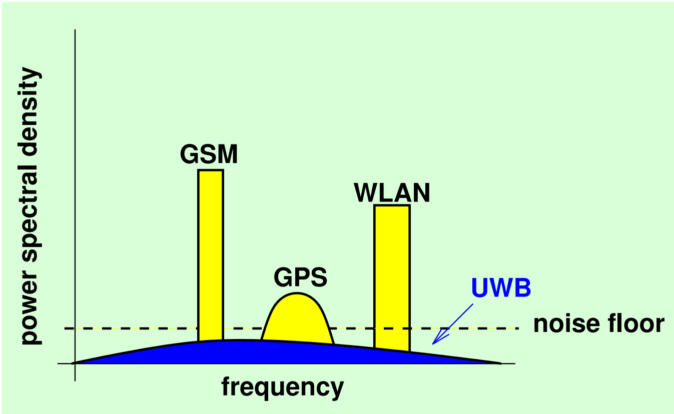


Figure 8.3: The difference in the allowed transmitted power for UWB and narrowband systems is substantial.

sign is much simpler than in the case of MBOA. Nevertheless, the large bandwidth or short pulse duration introduces a large number of signal processing issues and challenges such as synchronization, channel and symbol estimation, noise and narrowband interference suppression [60, 61]. In our further work we will focus our attention only to the IR-UWB systems.

8.2 Impulse radio UWB

Classical transceiver schemes use the data signal to modulate a carrier, $i.e.$, the spectrum of the data sequence is shifted from baseband to a higher carrier frequency. In contrast, in IR-UWB systems a carrier-less approach is employed. The information is conveyed by modulation of temporal pulses of extremely short duration — less than a nanosecond. As a consequence, the spectrum of a UWB signal covers an extremely large frequency band. To allow coexistence with already deployed narrowband communication systems such as GSM, GPS and WLAN, the energy of the emitted UWB pulses needs to be very low as indicated in figure 8.3.

From the circuit design aspect generation of the pulses is a relatively low complexity and low power consuming operation [62] and therefore facilitates the accomplishment of low cost transmitter devices. All these features make impulse radio attractive for high data rate, short distance, multi-user Wireless Local Area Networks (WLANs) and Personal Area Networks (PANs).

Propagation of temporally narrow pulses (order of ns), also known as *monocycles*

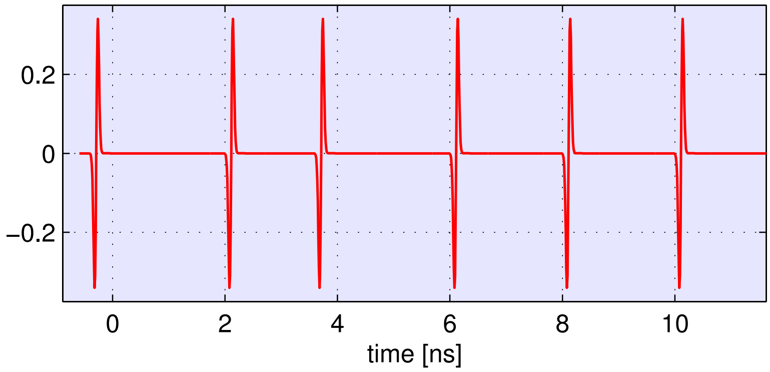


Figure 8.4: The principle of the impulse radio: Information is conveyed by a stream of pulses of extremely short duration (inferior to a nanosecond).

(see figure 8.4) is affected by a channel impulse response with a large number of delay taps [63] that is much longer than the duration of the pulse itself. As the channel resolution is inversely proportional to the bandwidth of the signal, differences in path delays or path lengths of 1ns and 30 cm, respectively, can be resolved [64, 65]. In figure 8.5 and figure 8.6 two examples of the measured channel impulse responses (CIR) of UWB system [66, 67] are presented. The shown channel data was collected in a measurement campaign performed within the AirLink project at Delft University of Technology. The first figure shows the CIR of a non-line of site (NLOS), corridor-to-office scenario in a typical university building. The latter is an example of a typical UWB channel in an industrial environment containing a large amount of metal reflectors. Note that the length of the presented high resolution CIRs is several hundred times larger than the pulse duration itself. This fact might limit the maximal achievable data rate in case transceiver is designed such that the spacing between two consecutive pulses needs to be larger than the length of the CIR.

8.3 Signalling schemes for UWB systems

A monocycle $g(t)$ of duration T_p is considered to be a part of a much longer time interval defined as a *frame* of duration T_f. In that manner, the ratio between the pulse and the frame duration, the so called *duty cycle*, has a very low value. This design parameter is introduced in order to reduce the possibility of catastrophic collisions between a number of active users [68].

One modulation scheme that facilitates coexistence of multiple users in UWB systems is a Time-Hopping Pulse Position Modulation (TH-PPM) [68]. To avoid colli-

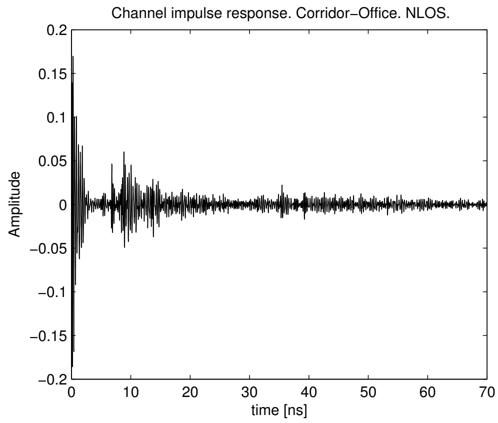


Figure 8.5: Corridor-to-office scenario(NLOS): An example of a UWB channel measured in a typical university building.

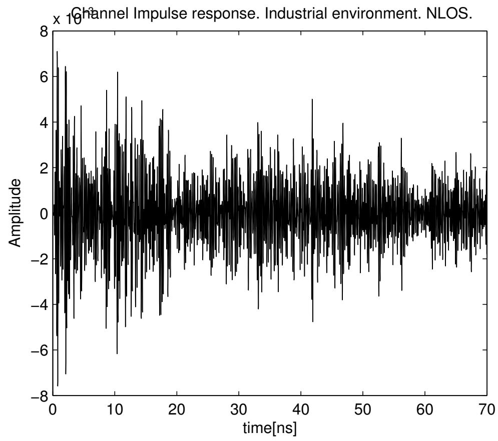


Figure 8.6: Industrial environment scenario(NLOS): A large amount of metal components introduces extremely large number of multipath components.

sions due to multiple access, each user is assigned a random *time hopping code* and shifts his monocycles within N_f subsequent frames according to it. This leads to a user specific sequence of pulses $g_c(t)$

$$g_c(t) = \sum_{d=0}^{N_f-1} g(t - dT_f - \delta_d) \ ,$$

where $\delta_d \in \{D_1, D_2, ..., D_M\}$ determines the position of the monocycle $g(t)$ with respect to the beginning of the d-th frame and $T_p < D_1 < D_2 < \cdots < D_M \ll T_f$. Over the sequence $g_c(t)$ coded by a time hopping code, a TH-PPM modulation can be performed (see figure 8.7) using data symbols $s_n = \{-1, +1\}$ to produce the broadcasted signal

$$t_x(t) = \sum_n g_c(t - nT_s - s_n\Delta) \ , \tag{8.1}$$

where $T_s = N_f T_f$ and Δ represent the symbol duration and the modulation offset [68], respectively. To avoid interframe interference the parameters δ_d and Δ should be chosen such that the transmitted pulse after being affected by the channel remains within the boundaries of a single frame. Note that in a multiuser case different time hopping patterns δ_d^q for $d = 0, \cdots N_f - 1$ are assigned to each of $q = 1, \cdots, Q$ users. One purpose of the time hopping is to avoid spectral lines that would occur if a fixed interpulse distance is used.

Another modulation scheme varies the polarity $i.e.,$ the sign of the monocycles within the pulse stream in accordance to the binary data sequence (see figure 8.8). It may be also viewed as a Pulse Amplitude Modulation (PAM) with alphabet $s_n \in \{-1, +1\}$. This approach is very similar to a CDMA modulation, now for pulses with much lower duty cycle. The transmitted PAM modulated signal now becomes

$$t_x(t) = \sum_n s_n g_c(t - nT_s) \ , \tag{8.2}$$

It is also possible to combine the above mentioned coding and modulation schemes $e.g.,$ a time hopping code (THC) and a PAM modulation. In such a case the TH-PPM $s_n\Delta$ from (8.1) is considered excessive.

To comply to the prescribed spectral mask, pulses are broadcasted at low power level. Consequently, to be able to extract the useful information at the receiver side, some kind of coding gain needs to be introduced. In UWB systems transmission of pulses of duration T_p is organized as follows. The basic unit is called a *frame* and is of length T_f. Depending on the deployed modulation scheme one (coherent) or two pulses (non-coherent) may be transmitted within a frame [69]. To prevent the mutual interference of adjacent frames, T_f is chosen such that the nonzero support of a doublet after the propagation through the channel of length T_h remains within the boundaries of the frame. Simplified, this condition may be written as $T_f > T_h + T_p$ and $T_f > T_h + 2T_p + \max\{\delta_d\}$ for coherent and non-coherent systems, respectively.

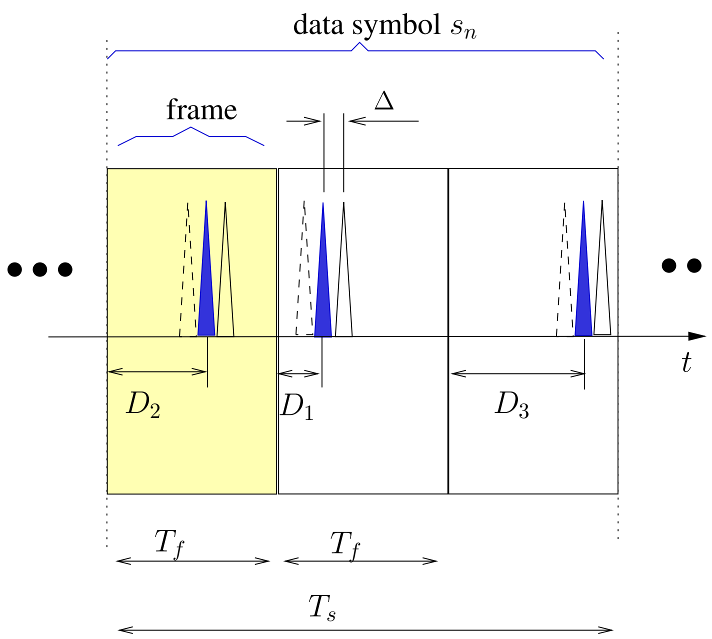


Figure 8.7: Time hopping code (THC) moves the monocycle (dark pulse) within each frame ($N_f = 3$) in accordance with the user specific time hopping code pattern $[\delta_1, \delta_2, \delta_3] = [D_2, D_1, D_3]$. Time-hopping pulse position modulation TH-PPM moves the monocylce to the left (dashed line) and to the right (solid line) for data symbol $s_n = -1$ and $s_n = +1$, respectively. Modulation offset is depicted by Δ.

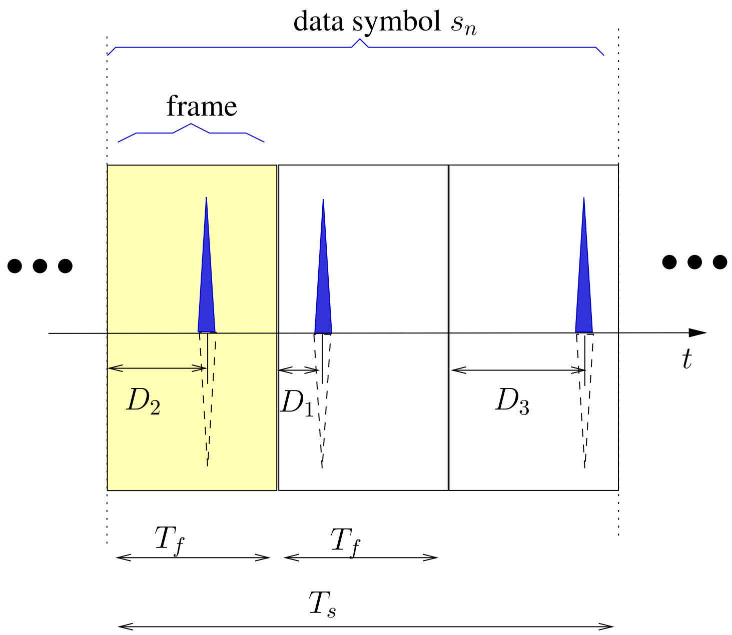


Figure 8.8: Pulse amplitude modulation changes the sign of the monocycle in accordance with data symbol. Dark and dashed line pulse correspond to the symbols $s_n = +1$ and $s_n = -1$, respectively. In the figure, next to the PAM a time hopping code is deployed.

Here $\max\{\delta_d\} = D_M$ represents the maximal spacing between two pulses of the transmitted doublets, see (8.3). To obtain a certain coding gain, a frame may be repeated several times to create a *chip* of duration T_c. Finally, several chips form a *data symbol*.

8.4 Synchronization issues and transceiver design

Two problems faced in UWB communication systems are the burst-like nature of transmissions and the sharing of the spectrum among own-system interferers and narrowband interferers. In ad-hoc communication systems, data packets are transmitted asynchronously, and it is of great importance to estimate the beginning of the data packet of interest in order to be able to perform subsequent symbol estimation. Most of the proposed signalling schemes exploit a combination of amplitude and time-hopping codes, where information is being transmitted in individual data packets that have a random time offset. The offset estimation needs to be performed prior to any symbol estimation and therefore it is of crucial importance in UWB systems to develop high resolution, efficient synchronization schemes.

To be able to detect the signal of interest, different transceiver modulation and coding schemes put different demands on the level of the synchronization. Several levels can be defined according to the synchronization accuracy, namely pulse-, frame-, chip- and symbol-level synchronization, where the synchronization error is in the order of a fraction of T_p, T_f, T_c and T_s, respectively. In subsequent two sections we analyze two different kind of receivers (correlation and autocorrelation receivers) and how they treat the synchronization issues.

8.4.1 Correlation receiver

The correlation receiver (RAKE receiver) is considered to be the optimal receiver in case a time hopping modulation scheme is deployed in the presence of noise. For this kind of receiver, the channel impulse response has to be estimated and convolved with the known user code to obtain a template. The template is correlated to the received data to detect the signal. The synchronization may be achieved by performing an exhaustive search over different delays, averaging over several data symbols and finally searching for the maximum of the collected energy function. The principle of the correlation based synchronization process is presented in figure 8.9. Note that the RAKE receiver is highly susceptible to timing jitter as the synchronization needs to be achieved on the sub-pulse resolution level.

As the process of estimating, storing and filtering the high resolution channel impulse response/template in an UWB system is highly complex, a number of authors searched for a way to reduce the computational complexity of the template offset based

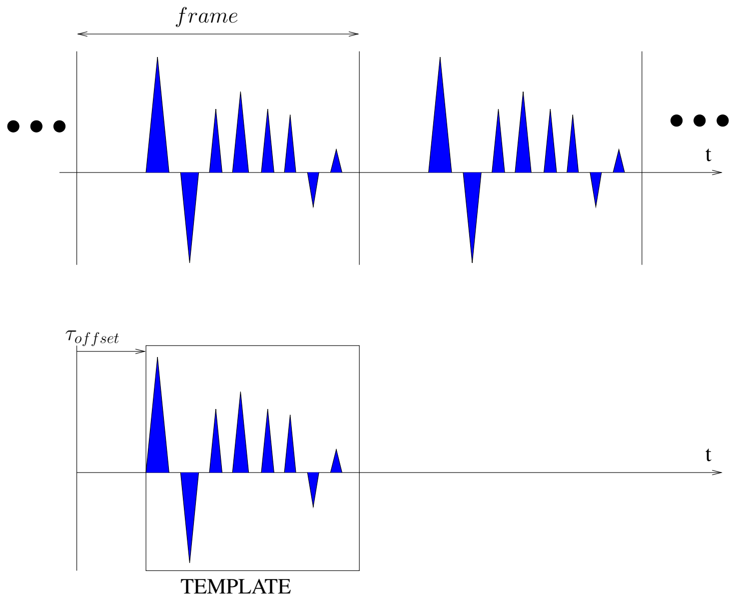


Figure 8.9: The principle of a correlation receiver. Initially the channel impulse response needs to be estimated to create a copy of a received pulse $i.e.,$ a template. Discrete points in which the channel is evaluated are defined as channel taps. The synchronization can be achieved by shifting the template over time (serial search) and correlating it with the received signal. The maximum of the correlation process is achieved when the template matches, $i.e.,$ becomes aligned with the received data.

on the serial search approach. The authors in [70] propose to use just a copy of the transmitted pulse (single tap RAKE receiver) as a template and examine several search (correlation) patterns in order to exceed an energy trashold and achieve the frame level synchronization. As shown in [70] a proper choice of the search pattern may reduce the total number of iterations by two orders of magnitude. However, the performance of the algorithm is analyzed for large signal to noise ratio, namely $SNR = 50$ dB. The presence of a significant amount of noise would considerably increase the complexity of this approach as substantial iterations would be needed to average out the effects of the noise in an UWB system.

In [71] Blazquez et al. tend to reduce the complexity of the correlation receiver by using a rectangular integration (correlation) window instead of the exact replica of the transmitted pulse to achieve the pulse level synchronization. To relax the computational burden of the receiver the authors propose to place two consecutive correlations as far apart as possible with a predetermined probability of detection as the constraint. In addition, the performance is analyzed for different types of transmitted pulses. As in [70] the synchronization process is treated as a Markov chain.

An UWB timing acquisition scheme based on the maximum-likelihood approach is treated in [72]. The proposed algorithm is based on the RAKE receiver and does not reconstruct the channel delay taps and corresponding gains but rather strive to capture the energy of the received signal projected onto a reduced number (L_c) of fixed time locations. Varying the size of L_c the desired trade-off between the complexity and the synchronization accuracy can be achieved. In this manner the sampling rate may range from a several samples per frame to several samples per pulse. Accordingly, the level of the synchronization varies from the frame to the pulse level.

The authors in [73] propose a transceiver scheme that exploit cyclostationarity of a periodic non-zero mean transmitted sequence for synchronization and subsequent estimation of the *continuous time synchronized aggregate template* (SAT). SAT incorporates the effects of the transmit filter together with the ISI channel impulse response. Knowing the SAT demodulation is performed by means of a SAT-correlator. Note that this algorithm estimates the template in the continuous time so that the long analog sequences need to be stored for processing. This is however very hard to achieve in an efficient way. Further, the use of a non-zero mean sequence in the synchronization process limits the number of users that simultaneously perform the synchronization to one. In that sense this scheme is applicable only to ad-hoc networks with a centralized structure, $i.e.,$ ones with star or clustered topologies.

All the algorithms presented above exploit the correlation of a template and the received data in order to achieve synchronization. To generate the template the knowledge of the complete or partial channel impulse response is required. However, the estimation of a high resolution UWB CIR and the process of collecting its energy leads to a huge computational complexity and may be time consuming. For this reason correlation receiver is unsuitable for the targeted low power, low cost UWB transceiver devices.

8.4.2 Transmit reference UWB (TR-UWB) scheme

A way to relax the synchronization burden and to side-step the estimation of a high resolution UWB channel is found in the implementation of a Transmit-Reference (TR) transmission scheme introduced already in 1964 [74] and revived by Hoctor and Tomlinson in [75, 76]. The idea consists of the transmission of two pulses, one after another, within a single frame, where the first pulse is used as the reference for the second pulse that is modulated with a data symbol. The polarity of the second pulse corresponds to data symbols $s_n \in \{-1, +1\}$. The transmitted signal is modeled as

$$t_x(t) = \sum_{d=0}^{N_f-1} \left[g(t - dT_f) + s_n \cdot g(t - dT_f - \delta_d) \right] .$$

Note that in this case $\delta_d = \{D_1, D_2, \cdots, D_M\}$ represents the spacing between the reference and the data modulated pulse. However, the sign of the information pulse may also be varied within the symbol interval increasing the randomness of the sequence. For this purpose a CDMA kind of (polarity) code $c_d \in \{+1, -1\}$ is introduced leading to the structure depicted in figure 8.10 and described by the equation

$$t_x(t) = \sum_{d=0}^{N_f-1} \left[g(t - dT_f) + s_n \cdot c_d \cdot g(t - dT_f - \delta_d) \right] . \tag{8.3}$$

Note that both pulses undergo the same multipath channel. At the receiver side an autocorrelation receiver is deployed in which the reference pulse is delayed and correlated with the data pulse. By subsequent integration, the energy that is smeared out by the channel is collected. At this point sampling is performed at a rate much lower than what is needed for the correlation receiver. With the proposed receiver design, a part of the complexity is transfered from the digital to the analog domain making the scheme realistic to implement by the current state of technology. The effective data rate is reduced up to 50% but the receiver sampling rate and complexity are highly reduced because the correlation and integration steps are done in the analog part of the receiver.

Even if TR-UWB system is suboptimal compared to the matched filter approach, due to its low complexity it is considered practical to be used in UWB systems. The main advantage is that the estimation of the channel impulse response is avoided by simultaneous transmission of a reference and information pulse. For the autocorrelation receivers used in the TR-UWB systems chip level synchronization (CLS) proved to be sufficient to achieve acceptable BER performance. Being synchronized to the chip level allows for improvements in the signal detection by further splitting of the chip intervals at the integrators output into the amplitude or time sections [77]. In this manner, and due to a low duty cycle of UWB systems, the regions within a chip that contain most of the energy may be allocated in a blind fashion. Accordingly, a

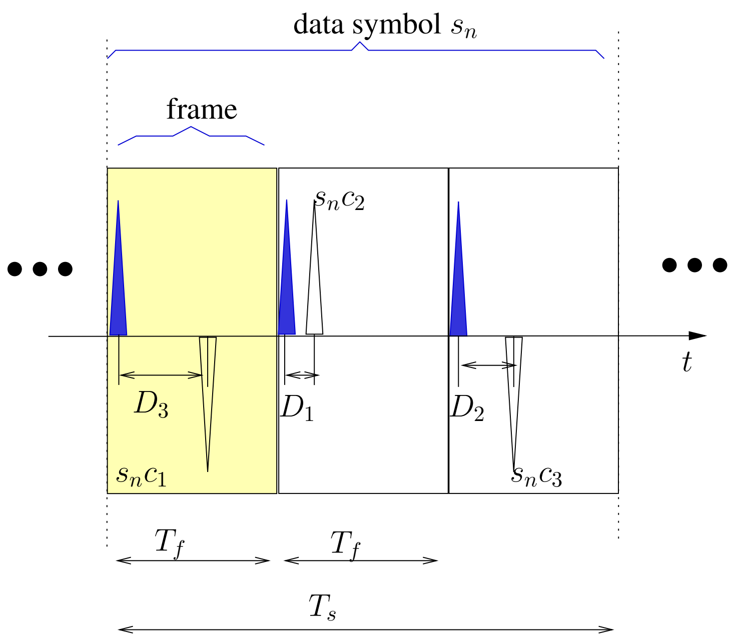


Figure 8.10: A transmit-reference UWB modulation scheme. Dark pulses represent the reference for the second (information) pulse within a frame. The time hopping code (THC) is deployed where the spacing between the two pulses is varied in accordance to a user specific code. In the figure the THC is $[\delta_1, \delta_2, \delta_3] = [D_3, D_1, D_2]$, where $D_1 < D_2 < D_3$. In addition, the second pulse is modulated by a combination of the polarity code $c_d = \{\pm 1\}$ and data $s_n = \{\pm 1\}$. At the receiver the reference pulse is aligned and correlated with the information pulse in order to recollect the energy smeared by the channel, hence the channel estimation is side-stepped. In the figure the data symbol is $s_n = 1$ and the polarity code is $[c_1,\ c_2,\ c_3] = [-1,\ 1,\ -1]$

reduced amount of noise energy is collected at the receiver enabling improvements in the BER performance.

8.5 Outline of Part II

In Part II of this thesis we propose a synchronization scheme for a TR-UWB system. The proposed algorithm exploits the fact that a shift in time corresponds to a phase rotation in the frequency domain. Accordingly, a blind, chip level computationally-efficient synchronization algorithm that takes advantage of the shift invariance structure in the frequency domain is developed. Integer and fractional delay estimation for the synchronization on the frame level are considered, along with subsequent symbol estimation step. This results in a multi-user algorithm, readily applicable to a fast acquisition procedure in an UWB ad-hoc network. The complexity of the system is greatly reduced as the sampling needs to be performed only once per channel (frame) duration. For typical channels that range from about 10ns to 100ns this leads to sampling rates of $f_s = 10$ to 100Msps. Nevertheless, the signal energy recollected by frame level of synchronization is sufficient for a proper symbol estimation.

This Part is organized as follows. In the Chapter 9 the data model for a single user synchronous transmit-reference UWB system is described. Chapter 10 extends this model to the asynchronous case. In Chapter 10 we present the single user synchronization scheme algorithm for TR-UWB systems. Chapter 12 extends the proposed scheme to the multi-user case. The synchronization algorithm is derived in chapter 11. The performance of the synchronization algorithm is verified by means of simulations at the end of Chapter 10 and Chapter 12 for single and multiuser case, respectively.

Chapter 9

TR-UWB Data Model–Synchronous case

In the previous chapter we have seen that an correlation UWB receiver has to have a template of the received pulse including the numerous multipath effects to be able to perform detection. In TR-UWB the idea is to transmit pilot (reference) pulses so that this template is directly available.

In this chapter the structure of the transmitted data burst as well as the parameters that arise as the result of a transmitted reference (TR) transceiver design and the deployment of an autocorrelation receiver are described.

In a TR-UWB system information is conveyed by pair of pulses (*doublets*) where the first pulse represents a reference while the second pulse is data modulated (TH-PPM, PAM). Both pulses undergo the same multipath channel. At the receiver side an autocorrelation receiver is deployed – the reference pulse is delayed and aligned with the second pulse, subsequently correlated (multiplied) and further integrated by a sliding window integrator. In this manner the energy of a transmitted data symbol is collected without the need of estimating a high resolution channel impulse response that may be several hundred times longer than the pulse itself.

The deployment of an autocorrelation receiver in a TR-UWB system where the channel impulse response is much longer than the duration of the monocycle results in a specific data model derived in [78] for a synchronous single symbol. This model is further extended by us to the asynchronous single user case in [79] and Chapter 10 and represents the starting point of our synchronization scheme. To define all parameters used in the data model, in the subsequent section we follow the mainline of the derivation from [78, 80] and extend it to cover more details.

9.1 Single data symbol transmission

9.1.1 Single doublet transmission

In the TR-UWB scheme pulses are transmitted in pairs (doublets). The first pulse is fixed and represents the reference, whereas the second one is modulated with the data. This approach is described in [75] where the authors consider the case in which the channel taps are mutually uncorrelated. This is in general not true for the impulse responses in UWB systems.

Let us first consider the transmission of a single doublet $d(t)$

$$d(t) = g(t) + c \cdot s \cdot g(t - D_i) \tag{9.1}$$

where $g(t)$ represents a reference pulse while $c \cdot s \cdot g(t - D_i)$ is a data modulated pulse, with scalars $c \in \{\pm 1\}$ and $s \in \{\pm 1\}$ representing a polarity code and a data symbol, respectively. Accordingly the sign of the data modulated pulse is defined by the value of $c \cdot s \in \{\pm 1\}$. The position of the second pulse with respect to the reference pulse is determined by a delay D_i, $i = 1, 2, \ldots, M$, where $D_1 < D_2 < \ldots < D_M$. We assume that a doublet is placed within a frame of length T_f and that a constraint

$$T_f > T_h + 2 \max_i \{D_i\} = T_h + 2D_M \tag{9.2}$$

holds. The length of the channel impulse response is in the range from $T_h =$ 2,ns$\cdots$, 200ns for distances between users of $\sim$ 1cm to $\sim$ 10m. The spacing between the reference and the information pulses is typically in the range $D_i = 1, \cdots, 5$ns. Note that most of the authors consider the case in which the spacing between two subsequent pulses $i.e.,$ frame size, needs to be larger than the maximal channel spread. In our case however, condition (9.2) needs to be satisfied. It implies that the pulses of a doublet affected by the channel fade out completely within a single frame. However, authors in [81] loose this constraint and allow for the interframe interference.

After the propagation through a long convolutive channel the signal at the output of the receiver antenna can be written as

$$r(t) = h(t) + c \cdot s \cdot h(t - D_i)$$

where $h(t) = g(t) \star h_p(t)$ represents the total channel impulse response obtained as the convolution of the transmitted pulse $g(t)$ and the channel impulse response $h_p(t)$. Note that h_p collects the effects of the transmit and receive antenna together with the propagation over the wireless channel. Subsequently, the received signal $r(t)$ is passed through a bank of receive delays D_m with $m = 1, \cdots M$, followed by the correlators. Here, index M represents the number of receiver chains that is equal to the number of possible offsets (see figure 9.1). At the output of the m-th correlator the resulting signal $y_m(t)$ is given by

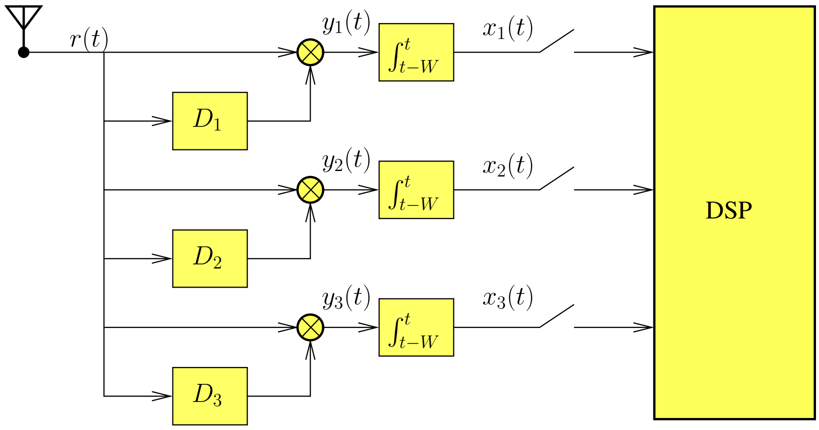


Figure 9.1: The structure of the autocorrelation receiver for $M = 3$ delay branches.

$$
\begin{aligned}
y_m(t) &= r(t)r(t-D_m) \\
&= [h(t) + c \cdot s \cdot h(t-D_i)][h(t-D_m) + c \cdot s \cdot h(t-D_i-D_m)] \\
&= h(t)h(t-D_m) + h(t-D_i)h(t-D_i-D_m) \\
&\quad +c \cdot s \cdot [h(t-D_i)h(t-D_m) + h(t)h(t-D_i-D_m)] \; .
\end{aligned}
$$

The output of each correlator is subsequently integrated by a sliding window integrator of width W, resulting in a signal

$$x_m(t) = \int_{t-W}^{t} y_m(\tau)d\tau \; , \quad m = 1, \cdots, M \; .$$

As only a single doublet is observed, the integration window is chosen to be larger than the frame duration, *i.e.*, $W > T_f$. Introducing a new function $\kappa(t, \Delta)$ that depicts the channel correlation

$$\kappa(t, \Delta) = \int_{t-W}^{t} h(\tau)h(\tau - \Delta)d\tau \tag{9.3}$$

the integrator's output can be written as

$$x_m(t) = \kappa(t, D_m) + \kappa(t-D_i, D_m) + c \cdot s \cdot \kappa(t-D_i, D_m - D_i) + c \cdot s \cdot \kappa(t, D_i + D_m). \tag{9.4}$$

Considering a channel impulse response with uncorrelated taps as in [76, 75] yields a non-zero channel correlation function $\kappa(t, \Delta) \neq 0$ only for matching delays $\Delta = 0$ (perfectly aligned reference and data modulated pulse at the receiver side)

$$\kappa(t, \Delta) = \begin{cases} P_h b(t) & \text{for} \quad \Delta = 0 \\ 0 & \text{for} \quad \Delta \neq 0 \end{cases} \tag{9.5}$$

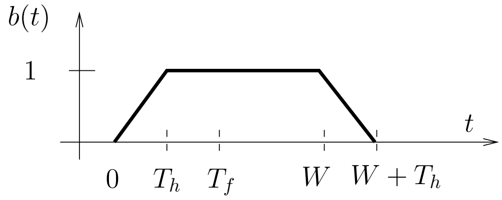


Figure 9.2: The approximate shape of the brick function $b(t)$ for integration window sizes larger than the duration of the channel, $W > T_h$. Linear slopes are used to portray the channel dependant shape in the transient regions $0 \leq t < T_h$ and $W < t \leq W + T_h$.

where $P_h \approx \int h^2(\tau)d\tau$ denotes the energy collected from the channel impulse response and $b(t)$ stands for the pulse shape obtained as a result of the sliding window integration:

$$b(t) = \begin{cases} 0 & t < 0 \\ 1 & T_h < t \leq W \\ 0 & t > T_h + W \\ \text{linear slope} & 0 \leq t < T_h \text{ or } W < t \leq W + T_h \end{cases} . \tag{9.6}$$

Note that in the regions $0 \leq t < T_h$ and $W < t \leq W + T_h$, $b(t)$ depends on a particular channel impulse response but it is approximated by a linear rising and decaying slope, respectively. The approximate shape of $b(t)$ is depicted in figure 9.2 for the integration period $W \geq T_h$, where T_h depicts the interval of the channel's non-zero support. An integration period larger than the channel duration ($W > T_h$) creates a flat region of the 'brick shape'.

For channels with uncorrelated taps [76, 75] the third term in (9.4) becomes dominant for matching transmit and receive delays ($D_i = D_m$). To simplify the notation we assume $\kappa(t - D_i, \Delta) \approx \kappa(t, \Delta)$ which is a realistic approximation in cases W is much larger than D_i, $\forall i$. The signal after integration $x_m(t)$ now becomes

$$x_m(t) \approx \begin{cases} \kappa(t, \Delta = 0) \approx \alpha_i b(t) \cdot c \cdot s & \text{for} \quad D_i = D_m \\ 0 & \text{for} \quad D_m \neq D_i \end{cases}$$

where $\alpha_i = P_h$ is a scaling coefficient. Nevertheless, the paths of *measured* UWB channel impulse responses proved to be correlated over short lags up to a few ns – the same order as D_i [78, 80]. Note: most other authors consider much longer D_i, typically $D_i > T_h$. As a result of the correlated channel taps, the expression in (9.5) does not hold for $\Delta \neq 0$. As presented in [80] the channel correlation function exhibits a 'sinc'-kind of shape (see figure 9.3). The fact that the channel autocorrelation function

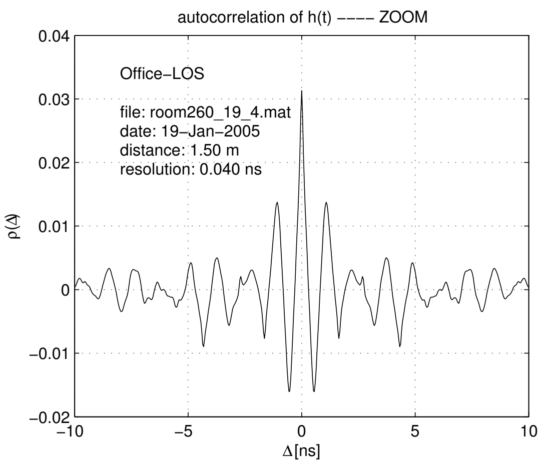


Figure 9.3: A measured channel correlation function versus the delay mismatch Δ for a measured channel impulse response in a typical university building.

differs from zero for non-matching delays modifies (9.5) such that

$$\kappa(t,\Delta) = \begin{cases} 0\ , & t \leq 0 \text{ or } t > W + T_h \\ \rho(\Delta)\ , & T_h < t < W \\ \int_0^t h(\tau)h(\tau-\Delta)d\tau & 0 \leq t < T_h \\ \int_{t-W}^{T_h} h(\tau)h(\tau-\Delta)d\tau & W < t \leq W + T_h\ , \end{cases} \tag{9.7}$$

where

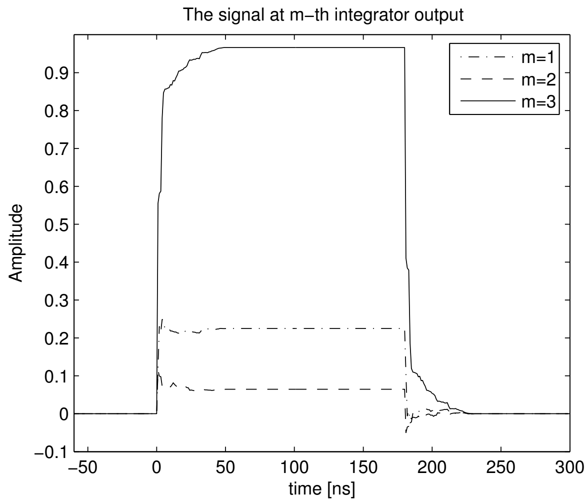
$$\rho(\Delta) = \int_0^\infty h(\tau)h(\tau-\Delta)d\tau \tag{9.8}$$

is the channel autocorrelation function.

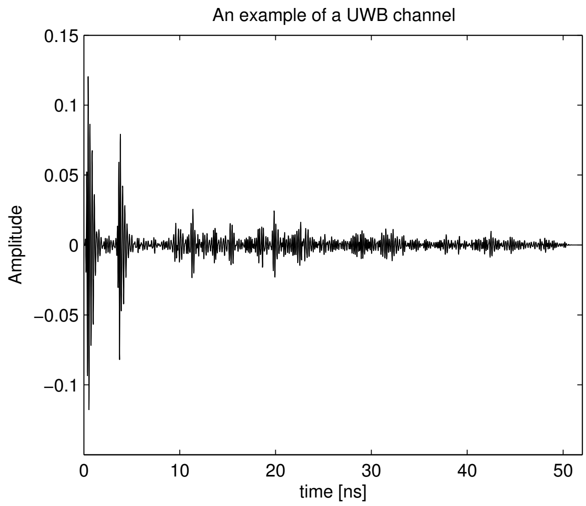
Note that P_h used in (9.5) is actually $P_h = \rho(\Delta = 0)$. Comparing (9.7) and (9.6) we see that the correlation function (9.7) can be approximated as

$$\kappa(t,\Delta) \approx b(t)\rho(\Delta) \tag{9.9}$$

The reason for this is that the **shape** of $\kappa(t,\Delta)$ in the transient regions $0 \leq t < T_h$ and $W < t \leq W + T_h$ depends on particular channel realization and its correlation properties (function of the delay mismatch Δ) as shown in figure 9.4. All these shapes may be approximated by (9.9) with a single 'brick shape' introduced in (9.6). In this manner the complexity of the data model is significantly reduced.



(a)



(b)

Figure 9.4: (a): The signal at each of $m = 1, 2, 3$ integrator outputs as the response to a single transmitted doublet. The integration window size is three times larger than the frame size, $W = 3T_f$. The maximum energy is recovered in the third delay branch whose delay ($D_m = D_3$) matches the one used at the transmitter side ($D_i = D_3$). (b): A measured UWB channel impulse in a typical university building used to generate subplot(a).

At this point the output of the m-th integrator (9.4) in case delay D_i is used at the transmission site is modeled as (see [80])

$$
\begin{aligned}
x_m(t) &= b(t)\{2\rho(D_m) + c \cdot s \cdot [\rho(D_m - D_i) + \rho(D_m + D_i)]\} \\
&= b(t)(\alpha_{m,i} \cdot c \cdot s + \beta_m) \ ,
\end{aligned} \tag{9.10}
$$

where

$$
\begin{aligned}
\alpha_{m,i} &= \rho(D_m - D_i) + \rho(D_m + D_i) \\
\beta_m &= 2\rho(D_m)
\end{aligned} \tag{9.11}
$$

Here $\alpha_{m,i}$ and β_m are real numbers that correspond to a ”virtual channel” gain and a DC offset, respectively. $\alpha_{m,i}$ is a function of both transmit delay D_i and receive delay D_m, whereas β_m depends only on the delay at the receiver D_m. From figure 9.3, in which the channel autocorrelation function of a measured channel impulse response is depicted, it becomes evident that $\rho(\Delta)$ is an even function implying $\alpha_{m,i} = \alpha_{i,m}$. Note that the latter is only valid in cases when the set of transmit delays exactly matches the set of the receive delays $\{D_i\} \equiv \{D_m\}$. In general, this condition is not satisfied, $e.g.$, in the production of a delay line a small mismatch occurs so that $D'_m = D_i + \tau_{\text{offset}}$. In that case $\alpha_{m,i} \neq \alpha_{i,m}$.

The top part of figure 9.5 shows a single transmitted doublet where the pulse separation equals D_i. Both pulses are affected by the same multipath channel depicted by dashed lines. The bottom part of the figure shows a general shape of the signal at the m-th integrator output as defined in (9.10). In the latter case the *linear slope approximation* of $b(t)$ introduced in (9.6) is used. In reality the signal shapes will insignificantly differ at each integrator output as shown in figure 9.4. Ignoring this differences and using the ’brick shape’ approximation, $i.e.$, $b(t)$ we are able to simplify the data model without losing any useful information.

In case of the matching delays ($D_m = D_i$, for $i = m$), $\alpha_{m,m}$ becomes a dominant component in (9.10), $i.e.$, $\alpha_{m,m} \gg \beta_m$ whilst for the unmatching delays ($D_m \neq D_i$, for $i \neq m$ or $i = m$) β_m and $\alpha_{m,i}$ become comparable [82]. Let us define a matrix $[\mathbf{A}]_{m,i} = \alpha_{m,i}$ and a column vector $[\mathbf{b}]_m = \beta_m$. In figures 9.6, 9.7, 9.8 and 9.9 typical values of the channel coefficients for matching and non-matching delay case are presented. These matrices were computed based on measured impulse responses [66].

The symmetrical and diagonally dominant structure of $\mathbf{A}$ for matched delay cases may be exploited in the design of a simple TR-UWB receiver that utilizes only the diagonal elements, as proposed in [75] and section ’symbol estimation’ in Section 11.3. In general, due to a possible mismatch in the production of the delay components $\mathbf{A}$ might not be structured.

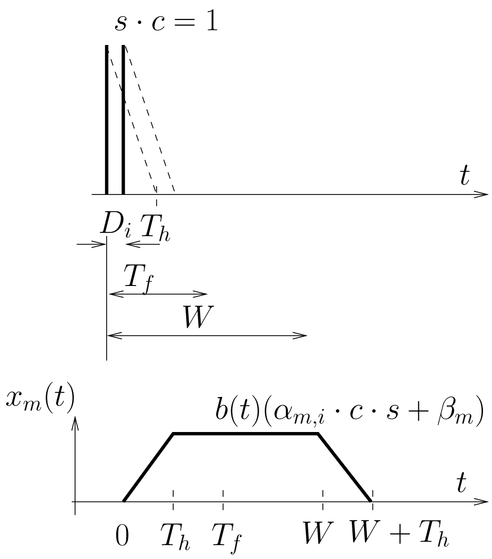


Figure 9.5: A single transmitted doublet (top) and the shape of the m-th integrator output $x_m(t)$.

$$\mathbf{A}_1 = \begin{bmatrix} 0.90 & 0.21 & -0.17 & -0.04 \\ 0.21 & 0.92 & 0.21 & -0.11 \\ -0.17 & 0.21 & 0.98 & 0.24 \\ -0.04 & -0.11 & 0.24 & 1.05 \end{bmatrix}, \ \mathbf{b}_1 = \begin{bmatrix} 0.47 \\ -0.19 \\ -0.05 \\ -0.14 \end{bmatrix}$$

Figure 9.6: Matching delays $D_m = D_i$, LOS case: Values of $\alpha_{m,i} = [\mathbf{A}]_{m,i}$ and $\beta_m = [\mathbf{b}]_m$ computed using a single outcome of a measured channel impulse response.

$$\mathbf{A}_2 = \begin{bmatrix} 1.08 & 0.37 & 0.17 & 0.12 \\ 0.37 & 1.08 & 0.28 & 0.05 \\ 0.17 & 0.28 & 0.97 & 0.29 \\ 0.12 & 0.05 & 0.29 & 1.08 \end{bmatrix}, \ \mathbf{b}_2 = \begin{bmatrix} 0.53 \\ 0.17 \\ 0.20 \\ 0.17 \end{bmatrix}$$

Figure 9.7: Matching delays $D_m = D_i$, NLOS case: Values of $\alpha_{m,i} = [\mathbf{A}]_{m,i}$ and $\beta_m = [\mathbf{b}]_m$ computed using a single outcome of a measured channel impulse response.

$$\mathbf{A}_1 = \begin{bmatrix} 0.08 & 0.12 & -0.16 & -0.05 \\ 0.01 & 0.09 & 0.15 & -0.08 \\ -0.03 & 0.04 & 0.17 & 0.21 \\ 0.05 & 0.04 & 0.10 & 0.25 \end{bmatrix}, \mathbf{b}_1 = \begin{bmatrix} 0.33 \\ -0.17 \\ -0.08 \\ -0.15 \end{bmatrix}$$

Figure 9.8: Non-matching delays $D_m \neq D_i$, LOS case: Values of $\alpha_{m,i} = [\mathbf{A}]_{m,i}$ and $\beta_m = [\mathbf{b}]_m$ computed using a single outcome of a measured channel impulse response.

$$\mathbf{A}_2 = \begin{bmatrix} 0.38 & 0.31 & 0.14 & -0.10 \\ -0.09 & 0.09 & 0.31 & 0.18 \\ -0.33 & -0.09 & 0.13 & 0.36 \\ -0.17 & -0.29 & -0.04 & 0.16 \end{bmatrix}, \mathbf{b}_2 = \begin{bmatrix} 0.73 \\ 0.43 \\ -0.10 \\ -0.14 \end{bmatrix}$$

Figure 9.9: Non-matching delays $D_m \neq D_i$, NLOS case: Values of $\alpha_{m,i} = [\mathbf{A}]_{m,i}$ and $\beta_m = [\mathbf{b}]_m$ computed using a single outcome of a measured channel impulse response.

9.1.2 Multiple doublets

To enable the coexistence with already deployed narrowband systems and due to its extremely wide band - a TR-UWB system needs to be broadcasted at a very low power level. Accordingly, a coding gain needs to be introduced so that the useful information at the receiver side may be extracted. Repetitive transmission of the same sequence is a simple way to supply the receiver with a sufficient amount of energy needed for the reconstruction of the desired signal. We examine now, how the energy of multiple transmitted frames per single information symbol can be recovered. The distance of pulses for all observed doublets, N_f in total, and the polarity $(c \cdot s)$ of the data modulated pulses are kept unchanged for all N_f frames. Such a block of frames forms *a chip* of length $T_c = N_f T_f$, where N_f and T_f denote the number of repeated frames and their duration, respectively (see top part of figure 9.10). To summarize, the transmitted sequence can be written as

$$t_x(t) = \sum_{d=0}^{N_f-1} \left[g(t - dT_f) + s \cdot c \cdot g(t - dT_f - D_i) \right], \tag{9.12}$$

where D_i is the spacing between the pulses of a doublet and remains unchanged within a single chip.

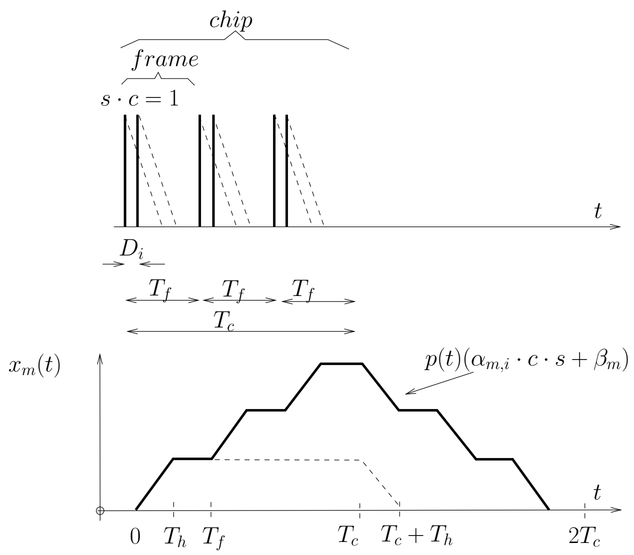


Figure 9.10: The shape at the m-th integrator output $x_m(t)$ in case three frames (doublets) per symbol s are transmitted. The size of the integration window is $W = T_c > T_h$.

The output of the m-th integrator is now modeled as a superposition of contributions of each individual doublet. Define a 'tent' shape function $p(t)$ that sums "brick" shapes of all doublets within a chip as

$$p(t) = \sum_{d=0}^{N_f-1} b(t - dT_f - D_i) \; . \tag{9.13}$$

The tent shape of the $p(t)$ is the same as presented in the bottom part of figure 9.10 that assumes $W = T_c > T_h$. The amplitude of $p(t)$ equals the total number of frames N_f.

Following the derivation in the previous section where the effects of correlated channel taps are introduced by a scaling $\alpha_{m,i}$ and a DC offset β_m the resulting signal $x_m(t)$ at the output of the m-th integrator becomes

$$x_m(t) = p(t)(\alpha_{m,i} \cdot c \cdot s + \beta_m) \; , \tag{9.14}$$

Note that the 'linear slope' assumption introduced in (9.6) is also applied in the definition of the $p(t)$ in (9.13). Under this assumption the shape of the signal at the m-th integrator output is shown in figure 9.10. The exact signal that arise at the output of the integrators is shown in figure 9.11.

Let the length of the integration window be equal to the chip duration, $i.e.$, $W = T_c$. Note that the nonzero support of each chip at the receiver side will be

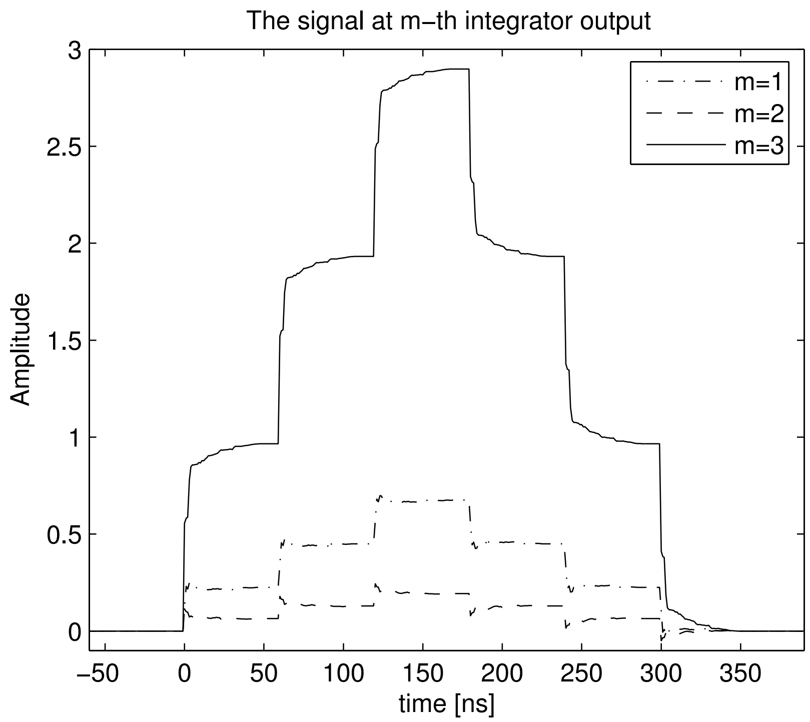


Figure 9.11: The signal $x_m(t)$ as it appears at the output of the m-th integrators. Three frames (doublets) per single symbol s are transmitted. The length of the channel is $T_h = 50$ns, frame duration $T_f = 60$ns and the chip duration $T_c = 3T_f = 180$ns. The size of the integration window is $W = T_c > T_h$. The channel impulse response used to generate this plot is presented in the bottom part of figure 9.4.

extended to the range $0 < t < 2W = 2T_c$, due to a sliding window integration step. This effect will lead to interchip (intersymbol) interference in cases of a multiple chip transmission.

9.1.3 Multiple chips

We consider now the transmission of multiple chips. As described in Section 9.1.1 the basic information unit is a *frame* of duration T_f. Further, N_f frames represent a chip of duration $T_c = N_f T_f$, and N_c chips represent a data symbol of duration $T_s = N_c T_c$. The j-th chip of the data symbol is modulated by sc_j, where $s \in \{-1, +1\}$ is the data symbol sequence and $c_j \in \{-1, +1\}$, $j = 0, 1, \ldots, N_c - 1$, represents the polarity code (CDMA-like code). The value of the delay D_i is constant within the j-th chip but changes from chip to chip according to the so-called time-hopping code $J_{i,j}$, $i = 1, 2, \ldots, M$, $j = 0, 1, \ldots, N_c - 1$, which is 1 if the delay D_i is used for the j-th chip and 0 otherwise:

$$J_{i,j} = \begin{cases} 1, & \text{if transmit at delay } D_i \text{ for chip } j \\ 0, & \text{elsewhere} \end{cases} \tag{9.15}$$

The transmitted signal for a single data symbol s is now modelled as

$$t_x(t) = \sum_{j=0}^{N_c-1} \sum_{d=0}^{N_f-1} \sum_{i=1}^{M} [g(t - jT_c - dT_f) + s \cdot c_j \cdot g(t - jT_c - dT_f - D_i)] J_{i,j}, \tag{9.16}$$

and is presented in figure 9.12. The top part of figure 9.12 represents the structure of a transmitted burst of a single symbol s that consists of two chips with corresponding CDMA codes c_0 and c_1. Each chip is further divided into three frames. The spacing between doublets D_i and the CDMA code is kept unchanged for the duration of a chip - T_c.

At the bottom part of figure 9.12 the signals at the integrators' output are presented, for the case in which two frames form a chip and two chips form a data symbol. The solid line depicts the signal in case of matching delays $D_i = D_m$. Due to the channel correlation property an attenuated copy of the latter signal is present at the outputs of all integrators and is proportional to channel coefficients $\alpha_{m,i}$ and β_m (dashed line). The overall signal is obtained as the sum of the signals for matched and non-matched delay cases.

Note that a scaled version of a signal transmitted on delay D_i emerges on all M receiver branches due to the channel correlation properties. In that manner a TR-UWB system resembles a single transmit multiple receive (SIMO) antenna system. Accordingly, the outputs of the M integrators may be viewed as a system with M receive antennas.

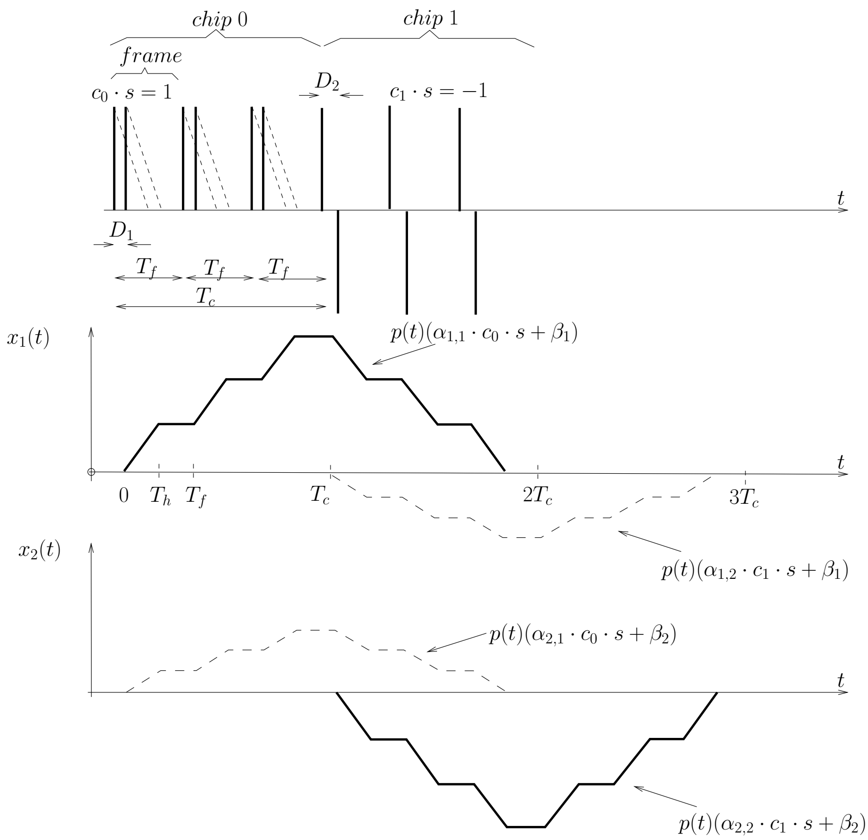


Figure 9.12: Top: The structure of pulses for transmission of three doublets per chip $N_f = 3$ and two chips per symbol $N_c = 2$. The data symbol is set to $s = +1$. The polarity (CDMA) code vector comprises two chips $[c_0, c_1] = [+1, -1]$ and the delay code is $[J_{1,0}, J_{2,1}]$. The latter means that the transmit delays D_1 and D_2 are used for the 1-st and the 2-nd chip, respectively. Bottom: The outputs of integrators for matching delays $D_m = D_i$ are depicted by a solid line; the dashed line depicts the contribution for non-matching delay cases. The overall signal at the output of an integrator is the superposition of these sequences ($D_m = D_i$ and $D_m \neq D_i$).

9.2 Matrix formulation–synchronous case

The matrix formulation for a single symbol case was originally derived in [82]. To make the derivation of our asynchronous multiple symbol data model self-contained, we present it here integrally.

As we have already seen, we consider the data model after transmission of N_c consecutive chips $\mathbf{c} = [c_0 \cdots c_{N_c-1}]^T$ for a single data symbol s. A perfect synchronization is presumed. Suppose we transmit a chip using one of the delays $D_1, \cdots, D_M$ and receive with a bank of receivers with delays $D_1, \cdots, D_M$. The next chip may be transmitted with a different delay.

Let $\alpha_{m,i}$ be the gain coefficient of the effective channel $p(t)$ for a transmitter delay D_i and a receiver delay D_m, and $\beta_{m,i}$ the corresponding gain offset. We also define matrices $\mathbf{A} = [\alpha_{m,i}]$, $\mathbf{B} = [\beta_{m,i}]$ of size $M \times M$. If a channel does not have temporal correlations, then $\mathbf{A} = \alpha\mathbf{I}$ (only a response at matching delays) and $\mathbf{B} = 0$, but in general the matrices can be arbitrary, although $\mathbf{A}$ is expected to be diagonally dominant. Note that due to (9.11) the following property holds: $\mathbf{B} = \mathbf{b}\mathbf{1}_M^{\mathrm{T}}$, where $\mathbf{b} = [\beta_1, \cdots, \beta_M]^{\mathrm{T}}$ is a column vector with elements $\beta_m = 2\rho(D_m)$ as defined in (9.11).

In terms of these coefficients, the model of the received data at the output of the integrator with delay D_m becomes

$$x_m(t) = \sum_{j=0}^{N_c-1} \sum_{i=1}^{M} p(t - jT_c)(\alpha_{m,i} J_{i,j} c_j s + \beta_m J_{i,j}) \ , \tag{9.17}$$

where $J_{i,j}$ is defined in (9.15). This data is sampled at instances $t = k\frac{T_c}{P}$, where the integer P is the oversampling factor. Define a channel matrix $\mathbf{P}_{k,j+1} = [p_{k,j}]$ of size $(N_c+1)P \times N_c$, where $p_{k,j+1} = p(k\frac{T_c}{P} - jT_c)$, for $k = 0, \cdots, (N_c+1)P - 1$ samples and $j = 0, \cdots, N_c - 1$ chips. The structure of $\mathbf{P}$ is illustrated in fig. 9.13. In case sampling is performed once per frame duration, *i.e.*, $P = N_f$, and assuming a perfect synchronization, each darkened part of $\mathbf{P}$ matrix (figure 9.13) comprise $2P$ samples depicted by small squares in figure 9.14. Under the mentioned assumptions these samples may be observed as samples of a known triangular pulse shape (dashed line in figure 9.14). Consequently $\mathbf{P}$ is completely known in a synchronous case.

The sampled data $x_{m,k} = x_m(k\frac{T_c}{P})$ has the model

$$x_{m,k} = \sum_{j=0}^{N_c-1} \sum_{i=1}^{M} p_{k,j}(\alpha_{m,i} J_{i,j} c_j s + \beta_m J_{i,j})$$

Collecting the samples of the m-th integrator output $x_{m,k}$ into a vector $\underset{\sim}{\mathbf{x}}_m$ of size

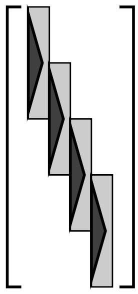
$\mathbf{P}_{(N_c+1)P\times N_c} =$ 

Figure 9.13: Structure of the matrix $\mathbf{P}$.

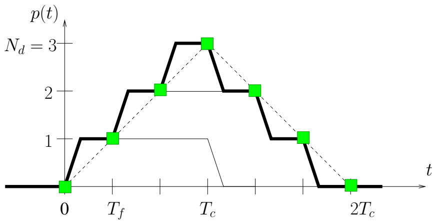


Figure 9.14: The shape of a sampled function $p(t)$ in a synchronous case. The shaded squares represent the sampling instants.

$(N_c + 1)P \times 1$, we obtain

$$\begin{aligned}
\underset{\sim}{\mathbf{x}}_m &= \begin{bmatrix} x_{m,0} \\ \vdots \\ x_{m,N-1} \end{bmatrix} \\
&= \sum_{j=0}^{N_c-1} \sum_{i=1}^{M} \mathbf{p}_j (\alpha_{m,i} J_{i,j} c_j s + \beta_m J_{i,j}) \\
&= \sum_{j=0}^{N_c-1} \mathbf{p}_j [\mathbf{a}_m^T \mathbf{j}_j c_j s + \mathbf{b}_m^T \mathbf{j}_j] \,,
\end{aligned}$$

where $\mathbf{j}_j$ is a column vector of size $M \times 1$ that collects all elements of $J_{i,j}$ for $i = 1, \cdots, M$ and a given j. Note that $\mathbf{j}_j$ has only one nonzero entry, corresponding to the transmitted delay, according to (9.15). We further define a 'code delay' matrix

$$\mathbf{J} = [\mathbf{j}_0, \cdots, \mathbf{j}_{N_c-1}]$$

of size $M \times N_c$. Vector $\mathbf{p}_j \,:\, (N_c + 1)P \times 1$ is the j-th columns of $\mathbf{P}$ (see also figure 9.13).

The received data vector $\underset{\sim}{\mathbf{x}}_m$ now becomes

$$\begin{aligned}
\underset{\sim}{\mathbf{x}}_m &= \sum_{j=0}^{N_c-1} (\mathbf{a}_m^T \mathbf{j}_j) \otimes (\mathbf{P} \cdot \mathbf{e}_j) c_j s + (\mathbf{b}_m^T \mathbf{j}_j) \otimes (\mathbf{P} \cdot \mathbf{e}_j) \\
&= \sum_{j=0}^{N_c-1} (\mathbf{a}_m^T \otimes \mathbf{P})(\mathbf{j}_j \otimes \mathbf{e}_j) c_j + (\mathbf{b}_m^T \otimes \mathbf{P})(\mathbf{j}_j \otimes \mathbf{e}_j) \\
&= (\mathbf{a}_m^T \otimes \mathbf{P})(\mathbf{J} \circ \mathbf{I}_{N_c})\mathbf{c}s + (\mathbf{b}_m^T \otimes \mathbf{P})(\mathbf{J} \circ \mathbf{I}_{N_c})\mathbf{1}
\end{aligned}$$

where $\mathbf{a}_m^T$ and $\mathbf{b}_m^T$ are the m-th rows of $\mathbf{A}$ and $\mathbf{B}$ respectively, and $\mathbf{e}_j$ is the j-th column of the identity matrix. Here, we denote by $\otimes$ the Kronecker product, and by $\circ$ the Khatri-Rao product (column-wise Kronecker product). For the definition of Kronecker and Khatri-Rao products and their properties see Chapter 2.

Let us at this point, without loss of generality, introduce a subscript index k. In this manner we can relate the transmitted symbol s_k to the received data block x_k. This notation will further be exploited in the derivation of the asynchronous case data model. Stacking the $(N_c+1)P$ received samples at the output of all $m = 1, \cdots, M$ integrators into a vector $\underset{\sim}{\mathbf{x}}_k$ of size $M(N_c+1)P \times 1$, we obtain for the transmitted symbol s_k

$$\underset{\sim}{\mathbf{x}}_k = \begin{bmatrix} \underset{\sim}{\mathbf{x}}_1 \\ \vdots \\ \underset{\sim}{\mathbf{x}}_M \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1^T \otimes \mathbf{P} \\ \vdots \\ \mathbf{a}_M^T \otimes \mathbf{P} \end{bmatrix} (\mathbf{J} \circ \mathbf{I}_{N_c})\mathbf{c}s_k + \begin{bmatrix} \mathbf{b}_1^T \otimes \mathbf{P} \\ \vdots \\ \mathbf{b}_M^T \otimes \mathbf{P} \end{bmatrix} (\mathbf{J} \circ \mathbf{I}_{N_c})\mathbf{1}$$

This can be written compactly as

$$\begin{aligned}
\underset{\sim}{\mathbf{x}}_k &= (\mathbf{A} \otimes \mathbf{P})(\mathbf{J} \circ \mathbf{I}_{N_c})\mathbf{c}s_k + (\mathbf{B} \otimes \mathbf{P})(\mathbf{J} \circ \mathbf{I}_{N_c})\mathbf{1} \\
&= (\mathbf{AJ} \circ \mathbf{P})\mathbf{c}s_k + (\mathbf{BJ} \circ \mathbf{P})\mathbf{1}
\end{aligned} \tag{9.18}$$

In this model, $\mathbf{J}$, $\mathbf{P}$ and $\mathbf{c}$ are known, while $\mathbf{A}$ and $\mathbf{B}$ are unknown $M \times M$ matrices, and s_k is the unknown data symbol.

An equivalent alternative model is obtained by restacking the received signal $\underset{\sim}{\mathbf{x}}_k$ given by eq. (9.18) into a matrix $\underset{\sim}{\mathbf{X}}_k : (N_c+1)P \times M$, such that $\text{vec}(\underset{\sim}{\mathbf{X}}_k) = \underset{\sim}{\mathbf{x}}_k$. The latter means that the samples from M integrator outputs are placed into M columns. This is modeled as

$$\begin{aligned}
\underset{\sim}{\mathbf{X}}_k &= \mathbf{P}\text{diag}(\mathbf{c})\mathbf{J}^T\mathbf{A}^T s_k + \mathbf{PIJ}^T\mathbf{B}^T \\
&= \mathbf{P}[\text{diag}(\mathbf{c})\mathbf{J}^T \quad \mathbf{J}^T][\mathbf{A}s_k \quad \mathbf{B}]^T
\end{aligned} \tag{9.19}$$

Define $\mathbf{Z} = \mathbf{P}[\text{diag}(\mathbf{c}) \quad \mathbf{J}^T]$, $i.e.,$ a known matrix of size $(N_c+1)P \times M$. The rank of $\mathbf{B}$ is 1 and it can be written as $\mathbf{B} = \mathbf{b}\mathbf{1}^{\mathrm{T}}$. Accordingly, (9.19) can be written as

$$\begin{aligned}
\underset{\sim}{\mathbf{X}}_k &= [\mathbf{Z} \quad \mathbf{PJ}^T][\mathbf{A}s_k \quad \mathbf{b}\mathbf{1}^{\mathrm{T}}]^T \\
&= [\mathbf{Z} \quad \mathbf{PJ}^T\mathbf{1}][\mathbf{A}s_k \quad \mathbf{b}]^T \\
&= [\mathbf{Z} \quad \mathbf{q}][\mathbf{A}s_k \quad \mathbf{b}]^T \ .
\end{aligned}$$

where the column vector $\mathbf{q}$ is defined as $\mathbf{q} = \mathbf{P}\mathbf{J}^{\mathrm{T}}\mathbf{1} = [q_1, \cdots, q_{(N_c+1)P}]^{\mathrm{T}}$. Note that $\mathbf{q}$ merges all the columns of the $\mathbf{P}$ into a single vector. Let q_i represent the i-th element of $\mathbf{q}$. The columns of $\mathbf{P}$ represent the shifts of the discretized tent pulse shape $p(t)$ with amplitude N_f, therefore $q_i = N_f$ for $i = P+1, \cdots, N_cP$. Further, q_i collects the samples of the rising and falling edge of the $p(t)$ for $i = 1, \cdots P$ and $i = N_cP+1, \cdots, (N_c+1)P$, respectively. As a result $\mathbf{q}$ may be approximated by a vector whose entries are all equal to N_f, *i.e.*, $\mathbf{q} = N_f\mathbf{1}$. This brings us to a final, single symbol synchronous block data model that collects the samples at the output of M integrators and collects them columnwise

$$\underset{\sim}{\mathbf{X}}_k = [\mathbf{Z} \quad \mathbf{1}][\mathbf{A}s_k \quad \mathbf{b}]^T \, . \tag{9.20}$$

In the last equation, to simplify the notation and without loss of generality we have moved the scaling N_f from $\mathbf{q}$ to $\mathbf{b}$. $\underset{\sim}{\mathbf{X}}_k$ is of size $(N_c+1)P \times M$

Remark: *A single symbol at the transmission side covers the interval* $t = [0, N_cT_c]$ *while the non-zero support of the signal at the integrators output extends to the interval* $t = [0, (N_c+1)T_c]$*, due to the sliding window integration. This phenomenon introduces intersymbol interference in cases of multiple symbol transmissions and will be studied in the next chapter.*

In this chapter we have presented the data model for a synchronous transmission of a single data symbol in TR-UWB systems. This model represents the starting point in the derivation of a single user data model in asynchronous transmission of multiple symbols that is studied in the next chapter. In chapter 11 we further derive the synchronization scheme for the given asynchronous data model.

Chapter 10

Single user data model–Asynchronous case

In this Chapter the data model for a synchronous case, presented in the previous chapter is extended to the asynchronous single user case. We will analyze transmission of multiple data symbols that creates intersymbol interference due to the deployment of an autocorrelation receiver.

For a better insight in the data model, in the following we recall briefly in the structure of the transmitted pulses in a TR-UWB system, presented in figure 10.1.

- *Monocycles* $g(t)$ are pulses of duration shorter than a nanosecond. In the transmit reference UWB scheme monocycles are transmitted in pairs (doublets) which are mutually separated by a delay D_i, $i = 1, 2, \dots, M$, where M represents the total number of delays used. We assume that $D_1 < D_2 < \dots < D_M$. The first pulse is fixed and represents the reference, whereas the second one is modulated with the data.

- *A frame* is considered to be the basic unit of the transmitted sequence. The reference pulse is placed at the beginning of each frame followed by a pulse that carries the information. The distance between the pulses is D_i. The size of the frame $T_f > T_h + 2\max\{D_i\}$ defined in (9.2) is chosen such that a doublet affected by the channel completely fades out within a single frame (see the first frame in figure 10.1).

- *A chip* of duration $T_c = N_f T_f$ is comprised of N_f consecutive frames. The j-th chip of the k-th data symbol is modulated by $s_k c_j$, where $s_k \in \{-1, +1\}$ is the data symbol sequence and $c_j \in \{-1, +1\}$, $j = 0, 1, \dots, N_c - 1$, represents the polarity code. The value of the delay D_i is constant within the j-th chip but

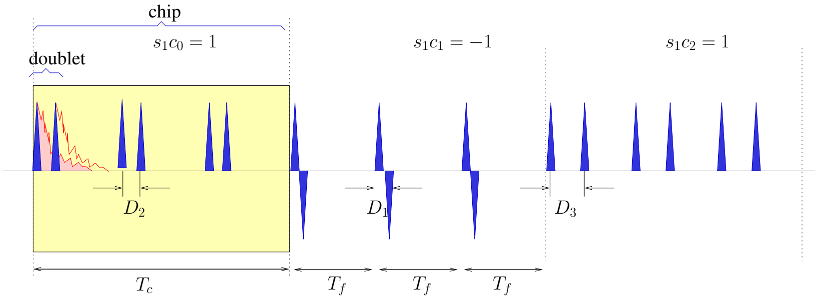


Figure 10.1: The structure of transmitted UWB pulses. The data symbol is set to $s_1 = +1$. The polarity (CDMA) code vector comprise three chips $\mathbf{c} = [c_0, c_1, c_2] = [+1, -1, +1]$, the delay code is $\mathbf{J} = [J_{2,0}, J_{1,1}, J_{3,2}]$. The latter means the transmit delays D_2,D_1,D_3 are used for the 1^{st},2^{nd} and 3^{rd} chip, respectively.

changes from chip to chip according to the so-called time-hopping code $J_{i,j}$, $i = 1, 2, \ldots, M$, $j = 0, 1, \ldots, N_c - 1$, which is 1 if the delay D_i is used for the j-th chip and 0 otherwise.

- *A symbol* is composed of N_c chips and is of duration $T_s = N_c T_c$.

Generalizing (9.16) to the multiple symbol case, the transmitted sequence becomes (see figure 10.1)

$$t_x(t) = \sum_k \sum_{j=0}^{N_c-1} \sum_{d=0}^{N_f-1} \sum_{i=1}^{M} [g(t - kT_s - jT_c - dT_f) + s_k c_j g(t - kT_s - jT_c - dT_f - D_i)] J_{i,j} \ ,$$

where k depicts the k-th data symbol.

Assuming the convolution of the pulse $g(t)$ with the propagation channel $h_p(t)$ is denoted by $h(t) = g(t) \star h_p(t)$, we can write the received sequence as

$$r(t) = \sum_k \sum_{j=0}^{N_c-1} \sum_{d=0}^{N_f-1} \sum_{i=1}^{M} [h(t - kT_s - jT_c - dT_f) + s_k c_j h(t - kT_s - jT_c - dT_f - D_i)] J_{i,j} . \tag{10.1}$$

We make the assumption that the channel duration T_h satisfies the condition $T_h < T_f - D_i - \max\{D_m\}$ in order to avoid uncontrolled interference between adjacent doublets [78] caused by the unknown channel realization. Note that we consider no additive noise throughout this work, in order to simplify the presentation. However, all the simulations will be carried out in the presence of noise. Since both pulses of a doublet undergo the same channel, one can be used as a "matched filter" for the other one at the receiver. This is the principle behind the autocorrelation receivers presented in [75, 78] and Chapter 9 (see figure 9.1). The autocorrelation receiver includes several steps: the received data $r(t)$ is first delayed over all possible delays

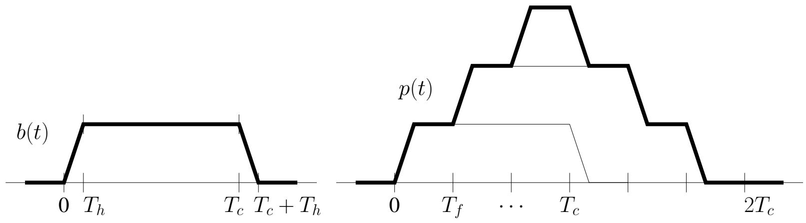


Figure 10.2: The shape of the signal at the integrators output for cases a single ($N_f = 1$) and triple ($N_f = 3$) doublets are broadcasted - $b(t)$ and $p(t)$, respectively.

$D_m, m = 1, 2, \ldots, M$, correlated with the original non-delayed signal, and integrated over a sliding window $W = T_c$. Just at this point the sampling is performed. In this manner the energy of the channel is collected in the analog domain avoiding the channel estimation step.

In an asynchronous system the receiver may start collecting the data before the moment any useful signal appears or the received data blocks might not be synchronized to the transmitted blocks carrying the information. To model this mismatch we introduce a parameter τ that is unknown and needs to be estimated to enable the estimation of the transmitted data sequence. Further, in the same manner as in Chapter 9 the output of the m-th integrator in the asynchronous noiseless case can be modeled as [79]

$$x_m(t) = \sum_k \sum_{j=0}^{N_c-1} \sum_{i=1}^{M} p(t - kT_s - jT_c - \tau)[\alpha_{m,i} c_j s_k + \beta_m] J_{i,j} \;, \tag{10.2}$$

The function $p(t)$ arises at the output of integrators as the typical response to a single chip and generally has a unit amplitude staircase tent shape, which is unknown and depends on the channel impulse response $h(t)$. In figure 10.2 the "brick" function $b(t)$ represents the recollected energy of the channel normalized to unity in case a chip contains just a single doublet, while $p(t) = \sum_{d=0}^{N_f-1} b(t - dT_f - D_i)$ arises in the case a chip contains several doublets. Accordingly, the summation over d in (10.1) is now 'hidden' within $p(t)$. T_h depicts the channel duration. The effect of the channel will be a scaling $\alpha_{m,i}$ and an offset β_m as presented in Chapter 9. While the scaling depends on both the transmitter delay D_i and the receiver delay D_m, the offset only depends on the receiver delay D_m. Moreover, both $\alpha_{m,i}$ and β_m depend on the correlation properties of the channel, as indicated in [78] and Chapter 9. Note that although $\alpha_{m,i}$ is generally maximal if $m = i$, some residual information remains when $m \neq i$, as an effect of the channel correlation.

Finally, τ is an unknown delay, which we try to estimate in our work. The received sequence is collected and stacked for further processing into blocks of data that equals the length of a transmitted symbol ($T_s = N_c T_c$). Several blocks, in general, form a complete analysis window. Due to the sliding window integration a single transmitted symbol becomes $(N_c+1)T_c$ wide. Accordingly, two data blocks are needed to collect all the useful information of the symbol. The unknown delay τ corresponds to the offset of the response to a transmitted symbol with respect to the beginning of a block of the analysis window. Hence, τ is restricted to the interval $\tau = [0, N_cT_c)$. An example of an asynchronous signal at the output of the integrators and the way it is collected in an analysis window for a single transmitted data symbol is presented in figure 10.3.

The bandwidth of $x_m(t)$ is at the same order of magnitude as the chip rate, which is significantly smaller than the transmission bandwidth. Hence, at this point, it is realistic to introduce sampling and switch to the digital domain. Let us sample $x_m(t)$ at rate P/T_c, where the oversampling rate per chip is $P = N_f$, $i.e.,$ equals the number of doublets per chip. Accordingly, the sampling interval is T_f. The sampled signal can then be written as

$$x_{m,n} = x_m(nT_c/P) = \sum_{k=-\infty}^{\infty} \sum_{j=0}^{N_c-1} \sum_{i=1}^{M} p_{n,j+kN_c}[\alpha_{m,i}J_{i,j}c_j s_k + \beta_m J_{i,j}] \ ,$$

where $p_{n,j} = p(nT_c/P - jT_c - \tau)$. The crucial observation now is that if we sample once per frame, $i.e.,$ at the rate $P = N_f$ per chip, $p_{n,j}$ may be observed as a sequence of samples of a perfectly known triangular pulse shape (see dashed line in figure 10.4). As a result $p_{n,j}$ is completely known if τ is an integer multiple of T_c/P. This fact is exploited in the process of estimating an arbitrary integer offset τ as presented in Chapter 11 (Chapter 11 also considers general, noninteger, offsets).

The samples at the integrators' output are collected in a block of data of the same size as the duration of a transmitted symbol $i.e.,$ $T_s = N_cP$ (the length of a *data block* in figure 10.3 equals T_s). In other words, we stack the N_cP samples $x_{m,n}$, $n = kN_cP, kN_cP+1, \ldots, (k+1)N_cP-1$ together in the $N_cP \times 1$ vector

$$\mathbf{x}_{m,k} = [x_{m,kN_cP}, \ldots, x_{m,(k+1)N_cP-1}]^T. \tag{10.3}$$

and stack the M vectors $\mathbf{x}_{m,k}$, $m = 1, 2, \ldots, M$, together in the $MN_cP \times 1$ column vector

$$\mathbf{x}_k = [\mathbf{x}_{1,k}^{\mathrm{T}} \cdots \mathbf{x}_{M,k}^{\mathrm{T}}]^{\mathrm{T}} \ , \tag{10.4}$$

where index m denotes the ordinal number of the integrator. Note that $\underset{\sim}{\mathbf{x}}_k$ introduced in the data model (9.18) for the transmission of an isolated data symbol collects the samples over the whole interval in which the integrators' outputs have non-zero support $i.e.,$ $(N_c+1)P$. However, $\mathbf{x}_k$ in (10.4) stacks the samples collected over a single symbol interval $i.e.,$ $T_s = N_cP$.

We further define a matrix $N_cP \times M$

$$\mathbf{X}_k = [\mathbf{x}_{1,k}, \ldots, \mathbf{x}_{M,k}]. \tag{10.5}$$

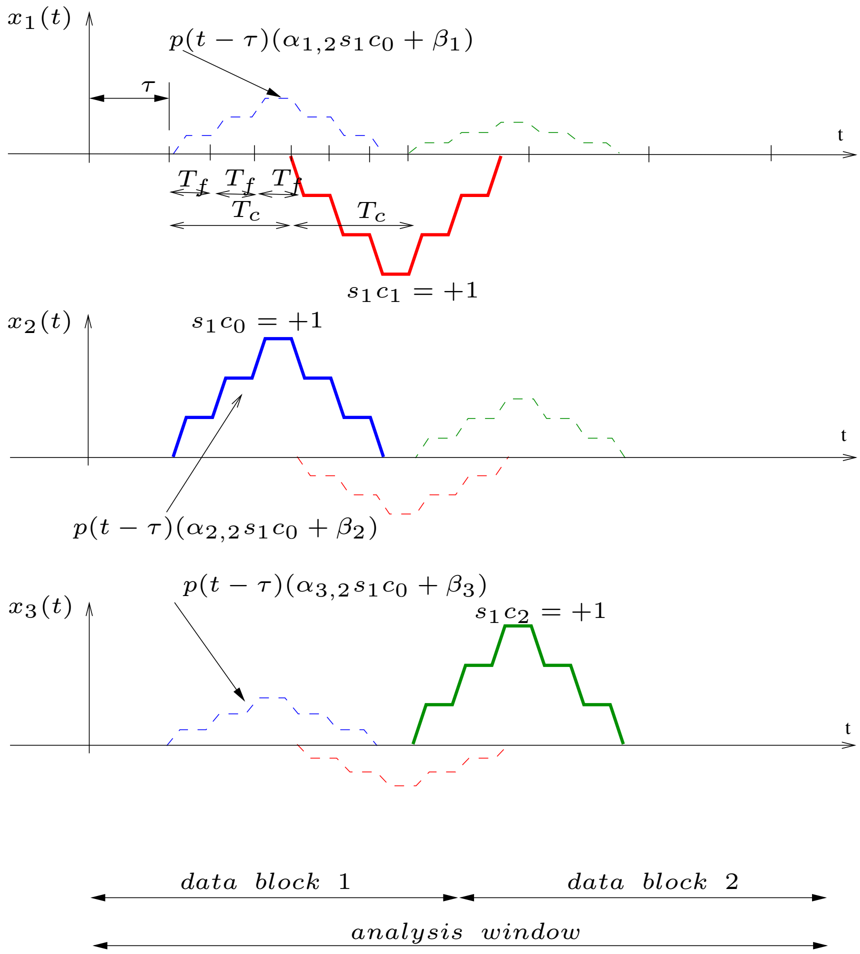


Figure 10.3: Signal at the integrators' output for the single transmitted data symbol in figure 10.1. The solid line represents the integrators output for matched delays ($D_m = D_i$) while dashed lines depicts the residual information in cases of non-matched delays $D_m \neq D_i$. The total signal per integrator is the superposition of these two components. τ is the offset of the observed sequence with respect to the beginning of the analysis window.

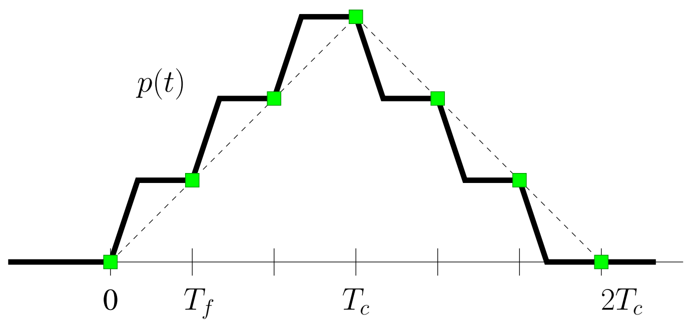


Figure 10.4: Sampling $p(t)$ once per frame duration $P = N_f$ for τ being an integer multiple of T_c/P creates a sample sequence that equals the samples of a known triangular pulse (dashed line).

The relation between $\mathbf{x}_k$ and $\mathbf{X}_k$ is defined by $\mathbf{x}_k = vec(\mathbf{X}_k)$, where operator vec stacks columnwise the elements of the matrix into a vector. We now first introduce a matrix model for a single transmitted data symbol, and then generalize this to multiple transmitted data symbols.

10.1 Single transmitted symbol – Matrix data model

In this section we derive a data model for an asynchronous single data symbol transmission. For the simplicity of exposure for the moment we consider only integer delay $i.e.,$ the offset of the received data is an integer multiple of the sampling interval T_c/P.

Suppose only the k-th symbol is transmitted. Let $\mathbf{x}_{m,k}$ be a vector of samples of $x_m(t)$ corresponding to the k-th symbol period. The sequences of length N_cP samples from the outputs of $m = 1, \cdots, M$ are stacked columnwise to create a matrix $\mathbf{X}_k$ of size $N_cP \times M$ as in (10.5). A single transmitted symbol s_k covers the interval $t = [0, N_cT_c)$. Nevertheless, due to the sliding window integration, at the receiver side it is extended to the interval $t = [0, (N_c + 1)P)$ that corresponds to two consecutive symbol periods, $\mathbf{X}_k$ and $\mathbf{X}_{k+1}$. In figure 10.3 these symbol periods are denoted as *data blocks*. Based on (10.3) and similar as in Chapter 9 and [78], a data model for an asynchronous single transmitted symbol now becomes

$$\begin{bmatrix} \mathbf{X}_k \\ \mathbf{X}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_\tau \\ \tilde{\mathbf{X}}_k \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_\tau \\ \mathbf{P}\mathrm{diag}(\mathbf{c})\mathbf{J}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}} \\ \mathbf{0} \end{bmatrix} s_k + \begin{bmatrix} \mathbf{0}_\tau \\ \mathbf{P}\mathbf{J}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}} \\ \mathbf{0} \end{bmatrix} . \tag{10.6}$$

Here, $\underset{\sim}{\mathbf{X}}_k$ is of size $[(N_c + 1)P \times M]$ collects the outputs of M integrators and represents the response of the system to a single synchronized transmitted symbol (see Chapter 9, equation (9.18)). The unknown time offset is represented by zero matrices.

$\mathbf{A}$ and $\mathbf{B}$, both of size $[M \times M]$, are unknown matrices which depend on the correlation properties of the multipath channel. It is shown in Chapter 9 that for matching delay cases ($D_i = D_m$) $\mathbf{A}$ is symmetric and diagonally dominant with positive entries on its main diagonal. $\mathbf{B}$ is a rank one matrix that can be written as $\mathbf{B} = \mathbf{b}\mathbf{1}^{\mathrm{T}}$ where $\mathbf{b}$ collects the channel 'offset' coefficients of all M receiver branches $i.e.,\ \mathbf{b} = [\beta_1, \cdots, \beta_M]^{\mathrm{T}}$ (see (9.11)). $\mathbf{1}$ is a vector with all entries equal to 1. $\mathbf{c} = [c_0\ , \cdots\ , c_{N_c-1}]^T$ is the known spreading code vector of length N_c chips. $\mathbf{P}\mathrm{diag}(\mathbf{c})\mathbf{J}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}$ and $\mathbf{P}\mathbf{J}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}$ are both of size $[(N_c + 1)P \times M]$ while zero matrices $\mathbf{0}_\tau$ and $\mathbf{0}$ have τ and $N_cP - P - \tau$ rows, respectively. For the moment assume that the packet delay with respect to the beginning of the received data block is an integer multiple of $T_c/P\ i.e.,\ \tau \in \{0, \ldots, N_cP - 1\}$. For such chosen τ, $\mathbf{P}$ is a completely known $[(N_c + 1)P \times N_c]$ block-Sylvester matrix whose columns are shifts of $p(t)$ as presented in figure 9.13.

Moreover, $[\mathbf{J}]_{i,j} = J_{i,j},\ i = 1, \cdots, M,\ j = 1, \cdots, N_c$ of size $[M \times N_c]$ is a known large selector matrix which has a single unit element per column that determines the time-hopping pattern for the chip that corresponds to that particular column. This means that $J_{i,j}$ is 1 in case that the transmit delay D_i is used for the j-th chip and 0 otherwise. Note that matrices $\mathbf{P}\mathrm{diag}(\mathbf{c})\mathbf{J}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}$ and $\mathbf{P}\mathbf{J}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}$ are both of size $(N_c + 1)P \times M$ and are P samples longer than a symbol period covered in $\mathbf{X}_k$. For this reason inter-symbol interference (ISI) arises in the case of multiple transmitted symbols.

We can rewrite 10.6 as

$$\begin{bmatrix} \mathbf{X}_k \\ \mathbf{X}_{k+1} \end{bmatrix} = \mathbf{P}_\tau \mathrm{diag}(\mathbf{c})\mathbf{J}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}} s_k + \mathbf{P}_\tau \mathbf{J}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}} \tag{10.7}$$

where $\mathbf{P}_\tau$ represents the shifted version of the P matrix $i.e.,$

$$\mathbf{P}_\tau = \begin{bmatrix} \mathbf{0}_\tau \\ \mathbf{P} \\ \mathbf{0} \end{bmatrix}. \tag{10.8}$$

The structure of $\mathbf{P}_\tau$ is presented in detail in figure 10.5.

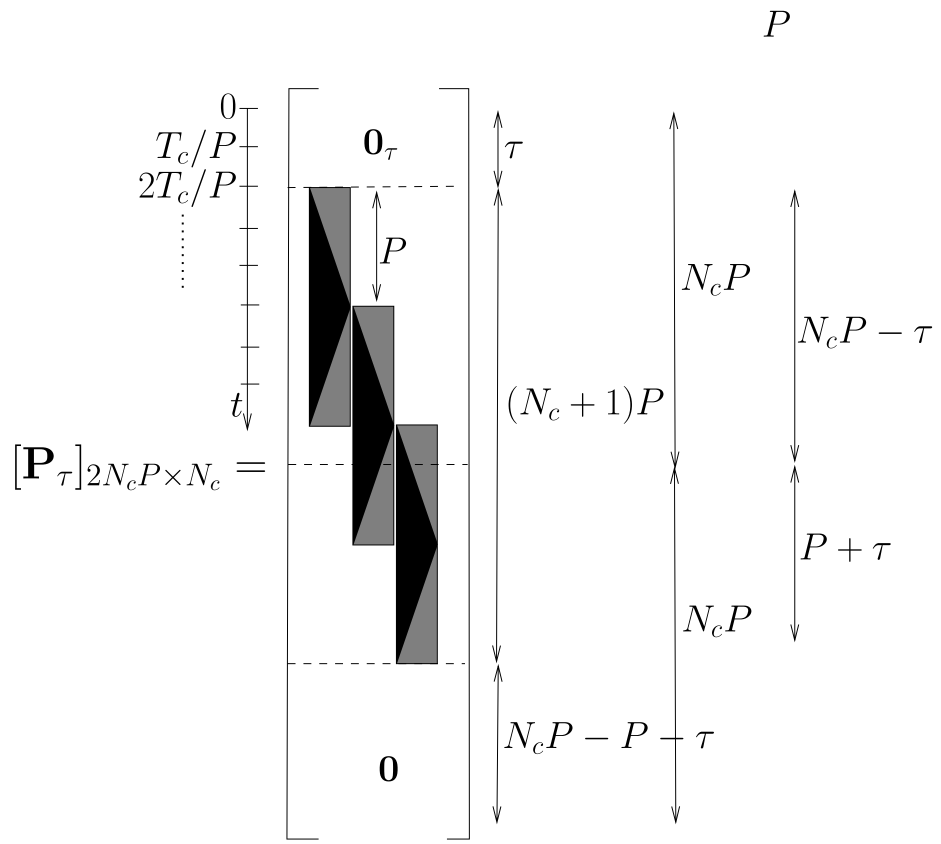


Figure 10.5: The structure of $\mathbf{P}_\tau$. The middle block part comprised of rectangles correspond to $\mathbf{P}$. Black triangles corresponds to the triangular pulse shape from figure 10.4. Sampling this triangular signal at the rate of T_c/P provides the elements of the $\mathbf{P}$ *i.e.*, $\mathbf{P}_\tau$. In case of an integer τ, samples of $\mathbf{P}$ are known. As a result matrix $\mathbf{P}_\tau$ is known up to the offset τ.

Equation (10.7) may be written as

$$\begin{aligned}
\begin{bmatrix} \mathbf{X}_k \\ \mathbf{X}_{k+1} \end{bmatrix} &= \mathbf{P}_\tau \text{diag}(\mathbf{c})\mathbf{J}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}} s_k + \mathbf{P}_\tau \mathbf{J}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}} \\
&= \mathbf{P}_\tau [\text{diag}(\mathbf{c})\mathbf{J}^{\mathrm{T}} \;\; \mathbf{J}^{\mathrm{T}}][\mathbf{A}s_k \;\; \mathbf{B}]^{\mathrm{T}} \\
&= \mathbf{P}_\tau [\text{diag}(\mathbf{c})\mathbf{J}^{\mathrm{T}} \;\; \mathbf{J}^{\mathrm{T}}\mathbf{1}][\mathbf{A}s_k \;\; \mathbf{b}]^{\mathrm{T}} \\
&= \begin{bmatrix} \mathbf{0}_\tau & \mathbf{0}_\tau \\ \mathbf{Z} & \mathbf{q} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} [\mathbf{A}s_k \;\; \mathbf{b}]^{\mathrm{T}} \qquad (10.9) \\
&\approx \begin{bmatrix} \mathbf{0}_\tau & \mathbf{0}_\tau \\ \mathbf{Z} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} [\mathbf{A}s_k \;\; \mathbf{b}]^{\mathrm{T}}
\end{aligned}$$

where $\mathbf{Z} := \mathbf{P}\text{diag}(\mathbf{c})\mathbf{J}^{\mathrm{T}}$ is a known $[(N_c+1)P \times M]$ "code matrix" while

$$\mathbf{q} := \mathbf{P}\mathbf{J}^{\mathrm{T}}\mathbf{1} \approx \mathbf{1}_{(N_c+1)P} \qquad (10.10)$$

The approximation $\mathbf{q} \approx \mathbf{1}_{(N_c+1)P}$ follows from the structure of the $\mathbf{P}$ and $\mathbf{J}$ matrix (see also figure 10.6). $\mathbf{X}_k$ and $\mathbf{X}_{k+1}$ are blocks of received sampled data and are both of size $[N_c P \times M]$ while matrix $\mathbf{0}_\tau$ and vector $\mathbf{0}$ have τ and $N_c P - P - \tau$ rows, respectively, and an appropriate number of columns. The delay takes any integer value in the interval $\tau \in \{0\ ,\dots,\ N_c P - 1\}$. The channel parameters $\mathbf{A}$, $\mathbf{b}$ and the data symbol s_k are unknown.

Let us define $\mathbf{Z}_\tau$ of size $2N_c P \times M$ as

$$\mathbf{Z}_\tau = \begin{bmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{0}_\tau \\ \mathbf{Z} \\ \mathbf{0} \end{bmatrix}, \qquad (10.11)$$

where $\mathbf{Z}_1$ and $\mathbf{Z}_2$ are both of size $N_c P \times M$ and depict the upper and lower half of $\mathbf{Z}_\tau$. The structure of $\mathbf{Z}$ is presented in figure 10.7. Following the approach from (10.11) we define

$$\mathbf{1}_\tau = \begin{bmatrix} \mathbf{0}_\tau \\ \mathbf{1} \\ \mathbf{0} \end{bmatrix}. \qquad (10.12)$$

We can now rewrite (10.9) as

$$\begin{bmatrix} \mathbf{X}_k \\ \mathbf{X}_{k+1} \end{bmatrix} = [\mathbf{Z}_\tau \quad \mathbf{1}_\tau][\mathbf{A}s_k \quad \mathbf{b}]^{\mathrm{T}} \quad . \qquad (10.13)$$

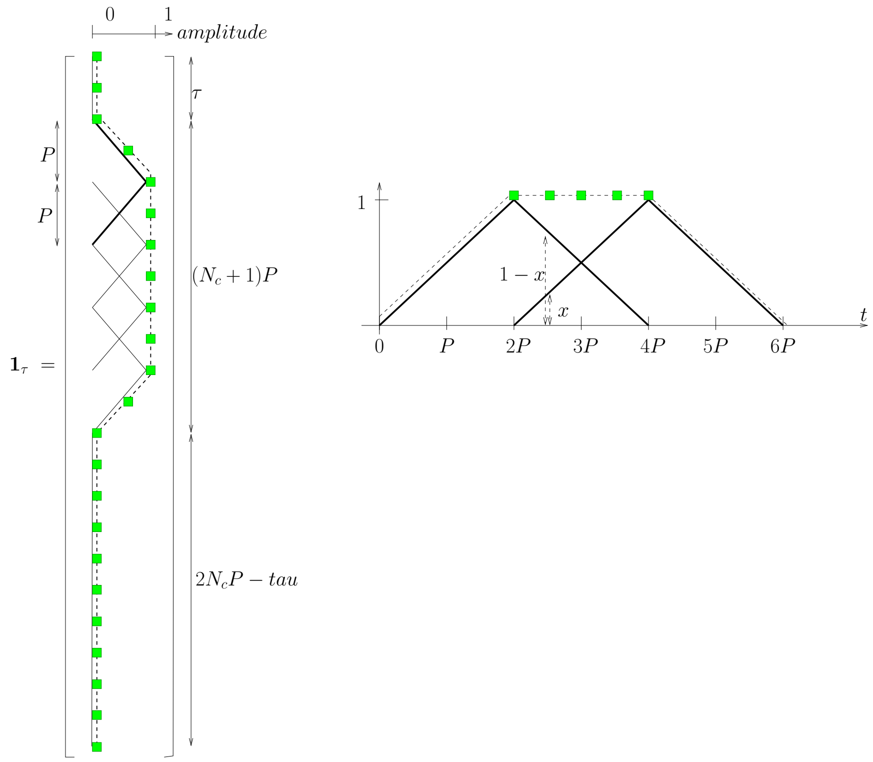


Figure 10.6: The structure of $\mathbf{1}_\tau$ vector from (10.12). Left: Triangles represent the approximated 'tent shape' signal that arise at integrators' output. Dashed line of amplitude 1 represents the shape of the resulting signal that collects the contributions of partially overlapping unit amplitude triangles. Consequently, we can approximate the resulting signal to have unit amplitude in the region $[\tau, \tau + (N_c + 1)P]$ as in (10.10). Right: The unit amplitude of the resulting signal in the 'overlap region' is not a function of the sampling instant.

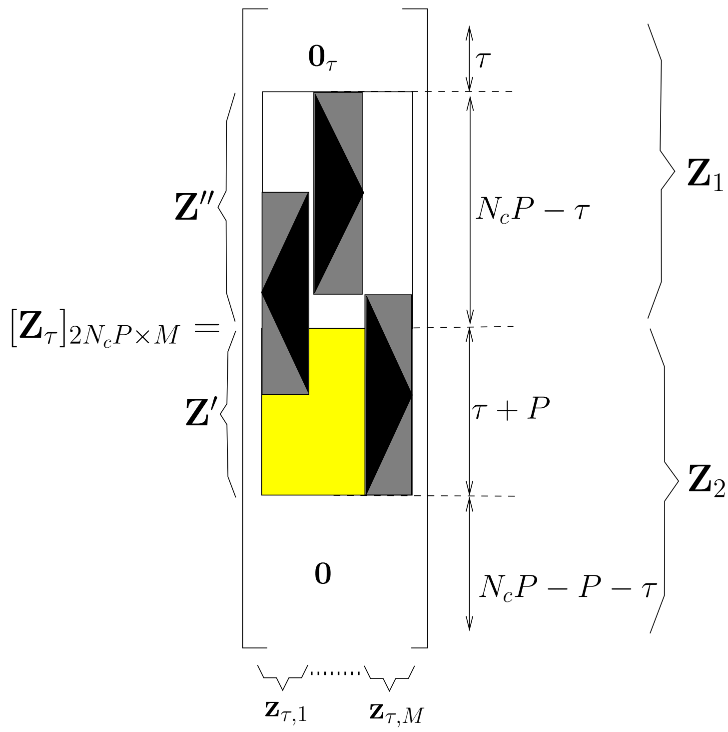


Figure 10.7: The structure of $\mathbf{Z}_\tau = [\mathbf{Z}_1^{\mathrm{T}} \quad \mathbf{Z}_2^{\mathrm{T}}]^{\mathrm{T}}$. $\mathbf{Z}''$ and $\mathbf{Z}'$ represent the parts of $\mathbf{Z}$ contained in the upper and lower half of $\mathbf{Z}$, respectively. Note that $\mathbf{Z} = [\mathbf{Z}''^T \quad \mathbf{Z}'^T]^{\mathrm{T}}$. $\mathbf{z}_{\tau,i}$ denotes the i-th column of $\mathbf{Z}_\tau$. In this figure we can clearly see how $\mathbf{Z} := \mathbf{P}\mathrm{diag}(\mathbf{c})\mathbf{J}^{\mathrm{T}}$ is created – initially, each of the N_c triangular shapes contained in $\mathbf{P}$ are scaled by $\{\pm 1\}$, $i.e.$, the corresponding element of the polarity code $\mathbf{c}$. Subsequently, $\mathbf{J}^{\mathrm{T}}$ places the triangles in one of the M columns in accordance to the time hopping code. In our example $M = 3$, $N_c = 3$, $\mathbf{c} = [+1, -1, +1]$ and the non-zero elements of $\mathbf{J}$:$M \times N_c$ are $J_{2,0}, J_{1,1}, J_{3,2}$.

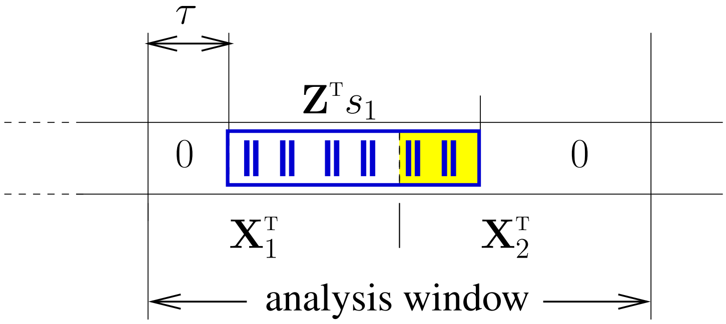


Figure 10.8: A single symbol spread by the block code, shifted for τ with respect to the beginning of a data packet.

The schematic representation of the received data blocks for a single transmitted data symbol s is depicted in figure 10.8. For clarity of the figure the part containing the product of $\mathbf{1b}^{\mathrm{T}}$ as well as $\mathbf{A}$ are omitted.

Observe that the presented data model resembles a conventional data model for DS-CDMA, up to the scaling ($\mathbf{A}$) and offset ($\mathbf{b}$) term of the channel correlation. This will allow us to use synchronization methods similar in spirit to the DS-CDMA synchronization methods. But before we introduce these synchronization methods, we first generalize the above model to a data model for multiple transmitted data symbols.

10.2 Asynchronous single user, multiple symbols data model

We now consider a model for an asynchronous single user and several transmitted symbols. When transmission of multiple data symbols is considered, inter-symbol interference (ISI) arises due to the implementation of the sliding window integration. In particular, two neighboring symbols have an overlap region of P samples. Generally two data symbols affect a single block of received data $\mathbf{X}_k$. Therefore, stacking $\mathbf{X}_k$ and $\mathbf{X}_{k+1}$ vertically, we can modify (10.13) to the following matrix model

$$\begin{bmatrix} \mathbf{X}_k \\ \mathbf{X}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_2 & \mathbf{Z}_1 & \mathbf{0} & \mathbf{1}' \\ \mathbf{0} & \mathbf{Z}_2 & \mathbf{Z}_1 & \mathbf{1}'' \end{bmatrix} \begin{bmatrix} \mathbf{A}^{\mathrm{T}} s_{k-1} \\ \mathbf{A}^{\mathrm{T}} s_k \\ \mathbf{A}^{\mathrm{T}} s_{k+1} \\ \mathbf{b}^{\mathrm{T}} \end{bmatrix} = [\mathbf{Z}'_\tau \;\; \mathbf{Z}_\tau \;\; \mathbf{Z}''_\tau \;\; \mathbf{1}] \begin{bmatrix} \mathbf{A}^{\mathrm{T}} s_{k-1} \\ \mathbf{A}^{\mathrm{T}} s_k \\ \mathbf{A}^{\mathrm{T}} s_{k+1} \\ \mathbf{b}^{\mathrm{T}} \end{bmatrix} . \tag{10.14}$$

The structure of the matrix $[\mathbf{Z}'_\tau \quad \mathbf{Z}_\tau \quad \mathbf{Z}''_\tau \quad \mathbf{1}]$ is presented in figure 10.9. Here, $\mathbf{1} = [\mathbf{1}'^{\mathrm{T}} \quad \mathbf{1}''^{\mathrm{T}}]^{\mathrm{T}}$ is a vector of size $2N_c P \times 1$ whose all elements are equal to 1 and

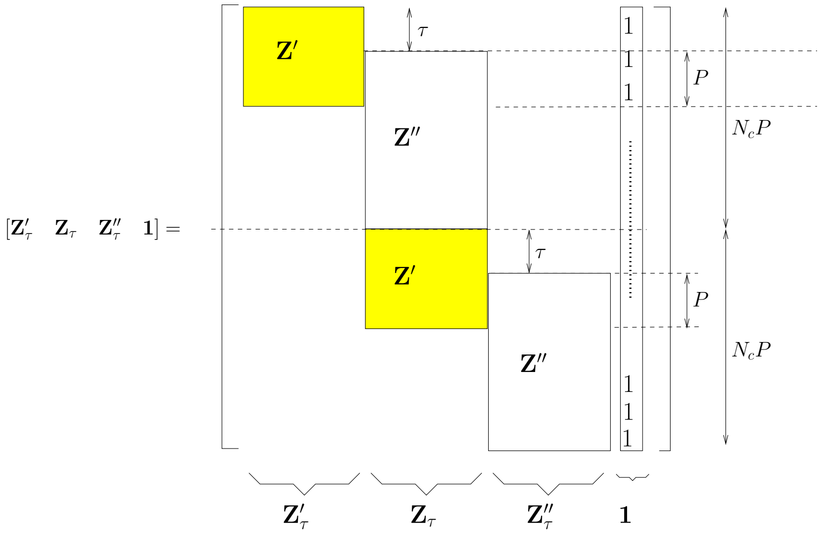


Figure 10.9: The structure of the block code matrix for two vertically stacked blocks of received data $\mathbf{X}_k$ and $\mathbf{X}_{k+1}$. $\mathbf{Z}'$ and $\mathbf{Z}''$ depict the 'tail' and the 'head' of the previous and subsequent data symbols, respectively. Note the existence of the intersymbol interference of P samples among two neighboring data symbols s.

where $\mathbf{1}'$ and $\mathbf{1}''$ are both vectors of size $N_cP \times 1$. This is obtained following the same strategy presented in figure 10.6. Note that in this case the triangles that create a unit amplitude resulting signal, cover the complete observation period ($2N_cP$) as the contributions of multiple data symbols that are added together.

Since short polarity and time-hopping codes (c_j and $J_{i,j}$) are considered and symbols are assumed unknown in a first stage, we may restrict τ to the interval $\tau \in [0, T_s)$. In this manner, τ depicts the offset of each data symbol s_k with respect to the corresponding block of received data $\mathbf{X}_k$ (see figure 10.10). The block columns $\mathbf{Z}'_\tau$, $\mathbf{Z}_\tau$, and $\mathbf{Z}''_\tau$, all of size $2N_cP \times M$, comprise the effects of the polarity and time-hopping codes as well as the effect of the pulse shape $p(t)$. For the moment we assume τ is an integer in the range $\tau \in \{0, \cdots, N_cP - 1\}$.

The complete structure of $[\mathbf{Z}'_\tau \quad \mathbf{Z}_\tau \quad \mathbf{Z}''_\tau \quad \mathbf{1}]$ is depicted in figure 10.9. For completeness of exposure we give their tedious mathematical description. $\mathbf{Z}_\tau$ shown in figure 10.7 has the following structure: $\mathbf{Z}_\tau = [\mathbf{0}^{\mathrm{T}}_{\tau\times M}, \mathbf{Z}^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}_{[(N_c-1)P-\tau]\times M}]^{\mathrm{T}}$. This block column comprises the complete version of a user specific code matrix $\mathbf{Z} = \mathbf{P}\mathrm{diag}(\mathbf{c})\mathbf{J}^{\mathrm{T}}$, which is known in case τ is an integer in the range $\tau = [0, \cdots, N_cP - 1]$. The other two block columns $\mathbf{Z}'_\tau$ and $\mathbf{Z}''_\tau$ can be defined as $\mathbf{Z}'_\tau = [\mathbf{Z}'^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}_{[(2N_c-1)P-\tau]\times M}]^{\mathrm{T}}$ and $\mathbf{Z}''_\tau = [\mathbf{0}^{\mathrm{T}}_{(N_cP+\tau)\times M}, \mathbf{Z}''^{\mathrm{T}}]^{\mathrm{T}}$. They contain only part of the user block code $\mathbf{Z}$. $\mathbf{Z}'$, with size $(P+\tau) \times M$, and $\mathbf{Z}''$, with size

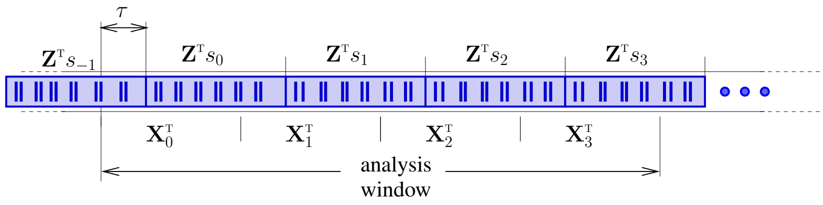


Figure 10.10: The structure of analysis window for the asynchronous TR-UWB scheme. Synchronization is achieved based on the repetitive structure of the 'short' CDMA-like code ($\mathbf{Z}$). For this reason the analysis window does not necessarily need to fatch the complete data packet to obtain the offset estimate. Knowing the size of the transmitted packet data may be reconstructed from several adjacent analysis windows. However, the proposed synchronization scheme is also applicable to cases where the analysis window is larger than the data packet.

$(N_cP - \tau) \times M$, depict the effect of a 'previous' and a 'subsequent' data symbol, respectively. It is thereby crucial to observe that $\mathbf{Z} = [\mathbf{Z}''^{\mathrm{T}}\ \mathbf{Z}'^{\mathrm{T}}]^{\mathrm{T}}$, see figure 10.7.

Writing (10.14) in a more compact form, we obtain

$$\begin{bmatrix} \mathbf{X}_k \\ \mathbf{X}_{k+1} \end{bmatrix} = [\mathbf{G}\ \ \mathbf{1}] \begin{bmatrix} \mathbf{S}_k \\ \mathbf{b}^{\mathrm{T}} \end{bmatrix}, \tag{10.15}$$

where $\mathbf{G} = [\mathbf{Z}'_\tau,\ \mathbf{Z}_\tau,\ \mathbf{Z}''_\tau]$ and $\mathbf{S}_k = [\mathbf{A}s_{k-1},\ \mathbf{A}s_k,\ \mathbf{A}s_{k+1}]^{\mathrm{T}}$ (see also figure 10.11).

Let us now define a received data matrix $\mathbf{X}$ that collect n received data blocks, as

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_0 & \mathbf{X}_1 & \cdots & \mathbf{X}_{n-1} \\ \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n \end{bmatrix}, \tag{10.16}$$

where n denotes the number of transmitted data symbols are transmitted. Using (10.15), we can write this matrix as

$$\mathbf{X} = [\mathbf{G}\ \ \mathbf{1}] \begin{bmatrix} \mathbf{S} \\ \mathbf{1}_n^{\mathrm{T}} \otimes \mathbf{b}^{\mathrm{T}} \end{bmatrix}, \tag{10.17}$$

where $\mathbf{S} = [\mathbf{S}_0, \ldots, \mathbf{S}_{n-1}]$. The structure of the received data blocks for multiple transmitted symbols is depicted in figure 10.11.

In the case where the analysis window is not within the transmitted packet, we can use the same model but allow some of the symbols s_k to be zero.

In this Chapter we have developed a single user asynchronous data model for a TR-UWB system. For simplicity of presentation ($e.g.$, matrix structure definitions) but without loss of generality we have consider only integer delays. In general, the

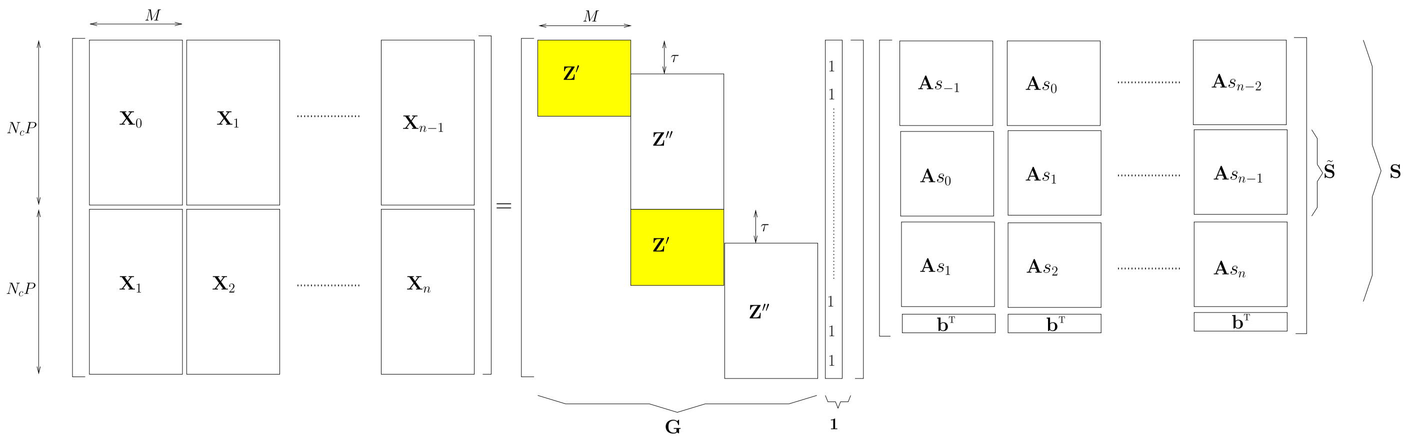


Figure 10.11: Block data model for the asynchronous single user case using a TR UWB scheme.

data model is also applicable for non-integer packet offsets. In the subsequent section we derive a subspace based synchronization schemes that performs over a block of received data. The proposed scheme provides a high resolution chip-level packet offset estimates and facilitates subsequent symbol detection. The performance of the algorithm will be verified for integer and non-integer delays.

Chapter 11

Blind synchronization algorithm – single user

In this Chapter we derive a blind, single user, block-based synchronization algorithm that provides high resolution packet offset estimates at the chip level along with the estimates of data symbols. The starting point is the asynchronous single user data model presented in figure 10.11. Note that the synchronization scheme doesnot demand the knowledge of channel correlation coefficients $\mathbf{A},\ \mathbf{b}$.

The aim is to estimate τ and the data $\{s_k\}$. The algorithm is an extension of the algorithm of Torlak and Xu [83], who considered blind channel estimation for CDMA using subspace techniques (see Appendix at the end of this Chapter).

11.1 Integer packet offset

At this point we recall the single user asynchronous data model introduced in equations (10.17) and (10.15):

$$\mathbf{X} = [\mathbf{G}\ \ \mathbf{1}] \begin{bmatrix} \mathbf{S} \\ \mathbf{1}_n^{\mathrm{T}} \otimes \mathbf{b}^{\mathrm{T}} \end{bmatrix},$$

where $\mathbf{G}$ represents the first $3M$ columns of the "block code matrix" as presented in figure 10.11. Further, $\mathbf{X}$ is decomposed by means of a singular value decomposition as (see Chapter 2)

$$\mathbf{X} = [\mathbf{U}_s \quad \mathbf{U}_0] \begin{bmatrix} \mathbf{\Sigma}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_0 \end{bmatrix} \begin{bmatrix} \mathbf{V}_s^{\mathrm{H}} \\ \mathbf{V}_0^{\mathrm{H}} \end{bmatrix}$$

where $\mathbf{U}_s$ denotes the left signal subspace and is of size $2N_cP \times 3M+1$, $\mathbf{U}_0$ is the left null space of size $2N_cP \times 2N_cP - 3M - 1$. The synchronization algorithm exploits the following properties of $\mathbf{G}$

1. $\mathbf{G}$ is a known function of the packet offset τ
2. $\mathbf{G}$ is tall matrix of size $2N_cP \times 3M$ so that $2N_cP > 3M$. Typically the number of delays $M \leq 4$, oversampling rate $P = 2$ or $P = 3$ and the number of chips $N_c < 20$. Note that we define the oversampling rate per chip, $i.e.,$ we perform sampling P times per chip duration.
3. $\mathbf{G}$ is orthogonal to the left nullspace of the matrix $\mathbf{X}$, $i.e.,$ $\mathbf{U}_0^{\mathrm{H}}\mathbf{G} = \mathbf{0}$.

We further exploit these properties to find an estimate of τ. More specifically, we solve

$$\begin{aligned} &\operatorname{argmin}_\tau \parallel \mathbf{U}_0^{\mathrm{H}}\mathbf{G} \parallel^2 \\ &= \operatorname{argmin}_\tau \textstyle\sum_i \parallel \begin{bmatrix} \mathbf{u}_{1,i} \\ \mathbf{u}_{2,i} \end{bmatrix}^{\mathrm{H}} \begin{bmatrix} \mathbf{Z}_2 & \mathbf{Z}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_2 & \mathbf{Z}_1 \end{bmatrix} \parallel^2 , \end{aligned} \tag{11.1}$$

where $\mathbf{u}_{1,i}$ and $\mathbf{u}_{2,i}$ are both of size $N_cP \times 1$ and depict the first and the second half of the i-th column of $\mathbf{U}_0$, respectively. $\mathbf{Z}_1$ and $\mathbf{Z}_2$ are of size $[N_cP \times M]$ and represent the upper and lower half of $\mathbf{Z}_\tau$ as defined in (10.11) and shown in figure 10.7. Accordingly $\mathbf{Z}_\tau = [\mathbf{Z}_1^{\mathrm{T}} \quad \mathbf{Z}_2^{\mathrm{T}}]^{\mathrm{T}}$. Note that $\mathbf{Z}_\tau$ is known up to the integer packet offset delay τ.

We now aim to transform (11.1) without changing the criterion, in order to bring out the block column $\mathbf{Z}_\tau$, containing the user specific code matrix $\mathbf{Z}$, which is known for integer delays τ. Restacking (11.1) as in [83] yields

$$\begin{aligned} \hat{\tau} &= \operatorname{argmin}_\tau \textstyle\sum_i \parallel [\mathbf{Z}_1^{\mathrm{H}}\ \mathbf{Z}_2^{\mathrm{H}}]^{\mathrm{H}} \begin{bmatrix} \mathbf{0} & \mathbf{u}_{1,i} & \mathbf{u}_{2,i} \\ \mathbf{u}_{1,i} & \mathbf{u}_{2,i} & \mathbf{0} \end{bmatrix} \parallel^2 \\ &= \operatorname{argmin}_\tau \textstyle\sum_i \parallel \mathbf{Z}_\tau^{\mathrm{H}} \begin{bmatrix} \mathbf{0} & \mathbf{u}_{1,i} & \mathbf{u}_{2,i} \\ \mathbf{u}_{1,i} & \mathbf{u}_{2,i} & \mathbf{0} \end{bmatrix} \parallel^2 \\ &= \operatorname{argmin}_\tau \textstyle\sum_i \parallel \mathbf{Z}_\tau^{\mathrm{H}}\mathcal{U}_i \parallel^2 \quad . \end{aligned} \tag{11.2}$$

Here, $i = 1, \cdots, 2N_cP - 3M - 1$ sweeps all the vectors from the left null space of $\mathbf{X}$ $i.e.,$ $\mathbf{U}_0$. By stacking horizontally $\mathcal{U}_i$ for all possible i-s we get the matrix $\mathcal{U}_0$ of size $2N_cP \times 3(2N_cP - 3M - 1)$

$$\mathcal{U}_0 = [\mathcal{U}_1, \mathcal{U}_2, \cdots, \mathcal{U}_{2N_cP-3M-1}] \ . \tag{11.3}$$

Now (11.2) can be written as:

$$\hat{\tau} = \operatorname*{argmin}_\tau \parallel \mathbf{Z}_\tau^{\mathrm{H}}\mathcal{U}_0 \parallel^2 \quad . \tag{11.4}$$

For integer offsets τ, this can be solved by performing the enumeration over $\tau \in \{0, \cdots, N_cP - 1\}$, since we know $\mathbf{Z}_\tau$ up to the integer packet offset delay τ. The resolution of this algorithm is limited by the oversampling rate $1/P$. Nevertheless, in the following exposure a computationally more efficient way of computing a packet offset τ is proposed. Note that the scheme is not limited by the sampling resolution T_c/P and can provide a high resolution packet offset estimate.

11.2 High resolution synchronization scheme

At this point, let us make a distinction between an integer and non-integer delay estimation. In the latter case the packet offset may take any value in the range $[0, \cdots, N_cT_c)$. The scheme we propose now is able to provide a high resolution delay estimates, not limited to integer multiples of T_c.

The idea presented in Section 11.1 can be made computationally more efficient and can also be implemented for the estimation of non-integer data packet offsets by use of the Fast Fourier Transformation (FFT). The FFT transforms a delay into a phase progression, and it will be shown that the problem transforms into a MUSIC-like estimation problem. We start by rewriting (11.4) as

$$\begin{aligned} \hat{\tau} &= \operatorname{argmin}_\tau \parallel \mathbf{Z}_\tau^{\mathrm{H}} \mathcal{U}_0 \parallel^2 \\ &= \operatorname{argmin}_\tau \parallel [\mathbf{z}_{\tau,1}^{\mathrm{H}} \mathcal{U}_0 | \mathbf{z}_{\tau,2}^{\mathrm{H}} \mathcal{U}_0 | \cdots | \mathbf{z}_{\tau,M}^{\mathrm{H}} \mathcal{U}_0] \parallel^2 , \end{aligned} \tag{11.5}$$

where $\mathbf{z}_{\tau,l}^{\mathrm{H}}$ represents the l-th row of $\mathbf{Z}_\tau^{\mathrm{H}}$ and subscript τ is a real number indicating the time offset of the data packet and takes any value from the interval $\tau \in [0\ ,\ N_cT_c)$. In general, we can write τ as

$$\tau = \tau_{int} + \tau_{frac} ,$$

where τ_{int} corresponds to the integer multiple of the sampling interval T_c/P and τ_{frac} represents a fractional delay in the range $0 \leq \tau_{frac} < T_c/P$.

As defined before (see figure 10.7) $\mathbf{Z}_\tau$ in (11.5) contains a shifted version of the user specific code $\mathbf{Z}$ that is a function of $\mathbf{P}$ $i.e.$, ,

$$\begin{aligned} \mathbf{Z}_\tau &= [\mathbf{0}_{\tau_{int}}^{\mathrm{T}} \quad \mathbf{Z}_{\tau_{frac}}^{\mathrm{T}} \quad \mathbf{0}^{\mathrm{T}}]^{\mathrm{T}} \\ &= [\mathbf{0}_{\tau_{int}}^{\mathrm{T}} \quad \mathbf{P}_{\tau_{frac}} \mathrm{diag}(\mathbf{c}) \mathbf{J}^{\mathrm{T}} \quad \mathbf{0}^{\mathrm{T}}]^{\mathrm{T}} , \end{aligned}$$

where, $\mathbf{0}_{\tau_{int}}$ of size $\tau_{int} \times M$ consists of τ_{int} zero columns. The fractional delay τ_{frac} will be incorporated in $\mathbf{Z}$. Now we discuss the cases for integer and non-integer packet offsets τ.

Recall the Section 11.1 where the packet offset τ is an integer multiple of the sampling interval T_c/P. Function $p(t)$ is being sampled in the known time instants

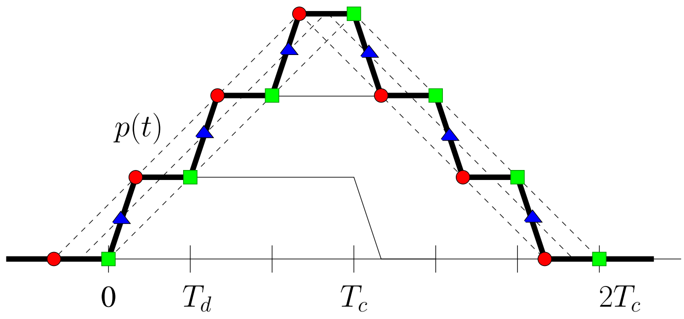


Figure 11.1: The sampling instants of the tent pulse shape $p(t)$ determine the structure of $\mathbf{P}_{\tau_{frac}}$. In all cases a triangular structure is obtained at an unknown but unimportant amplitude scaling.

as presented in figure 9.14. As a result the sampled values of $p(t)$ are known and accordingly $\mathbf{P}_{\tau_{frac}=0}$ (see figure 9.13) and $\mathbf{Z}_{\tau_{frac}=0}$ become completely known. In such a case $\mathbf{Z}_{\tau}$ has a known structure up to the unknown integer offset τ.

For non-integer offsets the sampling instants of function $p(t)$ and corresponding sample values are the function of an unknown offset τ_{frac}. Each black triangle of $\mathbf{P}$ (see figure 9.13) now corresponds to a stack of samples of the function $p(t - \tau_{frac})$ as indicated in figure 11.1. The important fact to note is that, in case sampling is performed once per frame, the obtained sample sequence of function $p(t - \tau_{frac})$ can be seen as the sampled version of a triangular pulse shape shifted over the offset τ_{frac}. Hence, $\mathbf{P}_{\tau_{frac}}$ and $\mathbf{Z}_{\tau_{frac}}$ will possess a known structure up to an unknown fractional offset τ_{frac} and a scaling. Consequently, $\mathbf{Z}_{\tau}$ possesses a known structure up to the unknown non-integer offset and a scaling. Note however, that we can completely reconstruct $\mathbf{Z}_{\tau}$ in the case the packet offset is zero, $i.e.,\ \tau = \tau_{int} + \tau_{frac} = 0$ as both $\mathbf{P}_{\tau_{frac}}$ and $\mathbf{Z}_{\tau_{frac}}$ become completely known. This fact will be used in the derivation of our synchronization scheme that we present next.

Taking the advantage of the last note and the properties of the Fourier transformation the problem caused by the structure of $\mathbf{P}$ that is in general, a function of τ_{frac} may be solved. In addition the scheme we propose provides a high resolution packet offset estimate.

Recognizing the property that a shift (under a Nyquist assumption that prevents the aliasing of the repeated spectral components) in the time domain corresponds to a phase progression in the frequency domain $e.g.,\ x(n - n_0) \rightarrow X(k)e^{j2\pi k n_0}$, where $n = 0, 1, \cdots, N-1$ and $k = 0, 1, \cdots, N-1$ correspond to sample index in time and frequency domain, respectively, we have for $\mathbf{z}_{\tau,l}$ in (11.5)

$$\mathbf{F}\mathbf{z}_{\tau,l} \quad = \quad \mathbf{D}_\tau \mathbf{F}\mathbf{z}_{0,l} \ . \tag{11.6}$$

Note that sampling at rate $P = 3$ (used in our simulations) only approximately fulfills the Nyquist criterion (similar as in [51]). The spectrum of a triangular pulse shape is known – $(sinc)^2$, and the influence of the sidelobes that create the aliasing effect when $P = 3$ is insignificant. We define $\mathbf{z}_{0,l}$ as the l-th column of $\mathbf{Z}_\tau$ (see figure 10.7) for which the offset $\tau = 0$

$$\mathbf{z}_{0,l} := \mathbf{z}_{\tau=0,l} \ .$$

Note that $\mathbf{z}_{0,l}$ is known for all $l \in \{1, \cdots, M\}$. Further, $\mathbf{F}$ stands for the discrete Fourier transform matrix and

$$\begin{aligned} \mathbf{D}_\tau &= \mathrm{diag}(\boldsymbol{\phi}_\tau) \\ \boldsymbol{\phi}_\tau &= [1, e^{j2\pi\tau/(2N_cP)}, \ldots, e^{j2\pi(2N_cP-1)\tau/(2N_cP)}]^{\mathrm{T}} \ , \end{aligned}$$

where $\boldsymbol{\phi}_\tau$ is a column vector of size $2N_cP \times 1$ and diag$(\cdot)$ converts a vector to a diagonal matrix and vice versa. We also define $\tilde{\mathbf{z}}_l$ as the Fourier transformation of the unshifted version of the l-th column of $\mathbf{Z}_\tau$, $i.e.$, $\mathbf{z}_{0,l}$:

$$\tilde{\mathbf{z}}_l := \mathbf{F}\mathbf{z}_{0,l} \ . \tag{11.7}$$

Applying these definitions to (11.6) yields

$$\mathbf{F}\mathbf{z}_{\tau,l} = \mathrm{diag}(\tilde{\mathbf{z}}_l)\boldsymbol{\phi}_\tau \tag{11.8}$$

Using the steps defined in (11.8) and a unitary transformation $\mathbf{F}^{\mathrm{H}}\mathbf{F} = \mathbf{I}$, we can rewrite (11.5) as

$$\begin{aligned} \hat{\tau} &= \mathrm{argmin}_\tau \parallel \mathbf{z}_{\tau,l}^{\mathrm{H}}\mathcal{U}_0 | \cdots | \mathbf{z}_{\tau,l}^{\mathrm{H}}\mathcal{U}_0 \parallel^2 \\ &= \mathrm{argmin}_\tau \parallel \mathbf{z}_{\tau,1}^{\mathrm{H}}\mathbf{F}^{\mathrm{H}}\mathbf{F}\mathcal{U}_0 | \cdots | \mathbf{z}_{\tau,M}^{\mathrm{H}}\mathbf{F}^{\mathrm{H}}\mathbf{F}\mathcal{U}_0 \parallel^2 \end{aligned} \tag{11.9}$$

Similar as in (11.7) the Fourier transform of $\mathcal{U}_0$ defined in (11.3) becomes

$$\tilde{\mathcal{U}}_0 := \mathbf{F}\mathcal{U}_0 \ .$$

We further rewrite (11.9) as

$$\begin{aligned} \hat{\tau} &= \mathrm{argmin}_\tau \parallel \mathbf{z}_{0,1}^{\mathrm{H}}\mathbf{F}^{\mathrm{H}}\mathbf{D}_\tau^*\tilde{\mathcal{U}}_0 | \cdots | \mathbf{z}_{0,M}^{\mathrm{H}}\mathbf{F}^{\mathrm{H}}\mathbf{D}_\tau^*\tilde{\mathcal{U}}_0 \parallel^2 \\ &= \mathrm{argmin}_\tau \parallel \tilde{\mathbf{z}}_1^{\mathrm{H}}\mathbf{D}_\tau^*\tilde{\mathcal{U}}_0 | \cdots | \tilde{\mathbf{z}}_M^{\mathrm{H}}\mathbf{D}_\tau^*\tilde{\mathcal{U}}_0 \parallel^2 \end{aligned} \tag{11.10}$$

where * denotes the complex conjugate operation.

At this point we aim to separate the knowns $\tilde{\mathbf{z}}_l^{\mathrm{H}}$ for $l = 1, \cdots, M$ and $\tilde{\mathcal{U}}_0$ on the right hand side and move the unknown $\mathbf{D}_\tau$ to the front. The multiplication of a vector and a diagonal matrix may be written as

$$\tilde{\mathbf{z}}_l^{\mathrm{H}} \mathbf{D}_\tau^* = \boldsymbol{\phi}_\tau^{\mathrm{H}} \mathrm{diag}(\tilde{\mathbf{z}}_l^{\mathrm{H}}) \ .$$

Using this property, $\boldsymbol{\phi}_\tau^{\mathrm{H}}$ that possesses a known structure but with an unknown parameter τ can be moved into the front of (11.10)

$$\begin{aligned} \hat{\tau} &= \mathrm{argmin}_\tau \parallel \tilde{\mathbf{z}}_1^{\mathrm{H}} \mathbf{D}_\tau^* \tilde{\mathcal{U}}_0 | \cdots | \tilde{\mathbf{z}}_M^{\mathrm{H}} \mathbf{D}_\tau^* \tilde{\mathcal{U}}_0 \parallel^2 \\ &= \mathrm{argmin}_\tau \textstyle\sum_{l=1}^{M} \parallel \boldsymbol{\phi}_\tau^{\mathrm{H}} \mathrm{diag}(\tilde{\mathbf{z}}_l^{\mathrm{H}}) \tilde{\mathcal{U}}_0 \parallel^2 \\ &= \mathrm{argmin}_\tau \parallel \boldsymbol{\phi}_\tau^{\mathrm{H}} [\mathrm{diag}(\tilde{\mathbf{z}}_1^{\mathrm{H}}) \tilde{\mathcal{U}}_0 | \cdots | \mathrm{diag}(\tilde{\mathbf{z}}_M^{\mathrm{H}}) \tilde{\mathcal{U}}_0] \parallel^2 \\ &= \mathrm{argmin}_\tau \parallel \boldsymbol{\phi}_\tau^{\mathrm{H}} \mathcal{K} \parallel^2 \end{aligned} \tag{11.11}$$

Here $\mathcal{K}$ is a known $2N_cP \times 3M(2N_cP - 3M - 1)$ matrix.

Due to the structure of $\boldsymbol{\phi}_\tau$, searching for the $\boldsymbol{\phi}_\tau$ that minimizes the last expression is equivalent to performing an inverse Fourier transform (IFFT) on the matrix $\mathcal{K}$ and searching for the row of the resulting matrix that has the lowest norm:

$$\hat{k} = \underset{k}{\mathrm{argmin}} \parallel [\mathbf{F}^{\mathrm{H}} \mathcal{K}]_k \parallel^2 \ \text{ for } k = \{1, \cdots, N_cP\} \ , \tag{11.12}$$

where k denotes the row index of the given matrix. The index k of the row with the lowest norm determines the part of the delay offset that corresponds to an integer multiple of T_c/P, *i.e.*, $\hat{\tau}_{int} = \hat{k} - 1$.

An additional fine grid MUSIC-kind search

$$\hat{\tau}_{frac} = \underset{\tau}{\mathrm{argmin}} \parallel \boldsymbol{\phi}_{\hat{\tau}_{int}+\tau}^{\mathrm{H}} \mathcal{K} \parallel^2 \tag{11.13}$$

performed around $\hat{\tau}_{int}$ provides a non-integer delay $\hat{\tau}_{frac}$ that takes a value in the interval $[-1/2, 1/2)$. The overall delay estimate expressed as a fraction of the chip duration T_c now becomes

$$\hat{\tau} = (\hat{\tau}_{int} + \hat{\tau}_{frac}) \frac{T_c}{P} \ , \tag{11.14}$$

where P is the oversampling rate and T_c is the chip duration.

11.3 Symbol estimation

At this point we provide a summary of the asynchronous single user data model used in symbol estimation. The model was introduced in (10.17)

$$\mathbf{X} = [\mathbf{G} \ \ \mathbf{1}] \begin{bmatrix} \mathbf{S} \\ \mathbf{1}_n^{\mathrm{T}} \otimes \mathbf{b}^{\mathrm{T}} \end{bmatrix} ,$$

where $\mathbf{S} = [\mathbf{S}_0, \ldots, \mathbf{S}_{n-1}]$ of size $3M \times nM$, where n denotes the number of transmitted data symbols. Matrix $\mathbf{S}_k : 3M \times M$, for $k = 1, \cdots n$ stacks vertically M consecutive data symbols affected by a channel matrix $\mathbf{A}$ $i.e.,$ $\mathbf{S}_k = [\mathbf{A}s_{k-1},\ \mathbf{A}s_k,\ \mathbf{A}s_{k+1}]^{\mathrm{T}}$ (for better overview of the data model structure we refer reader to figure 10.11). Matrix $\mathbf{A}$ of size $M \times M$ contains the channel correlation coefficients and is introduced in Section 9.1.1. It is channel dependant and unknown but constant over the whole analysis window and diagonally dominant with positive entries on the main diagonal. Vector $\mathbf{b} : M \times 1$ represents a DC shift of the equivalent channel (see Section 9.1.1). Unknown data symbols are represented by s_k for $k = 1, \cdots, n$.

After estimating the packet offset τ, we can reconstruct the complete $\mathbf{G}$ matrix. Estimation of the transmitted data symbols is now possible by performing a deconvolution of the matrix $\mathbf{X}$ using the known user code. A zero-forcing equalizer is obtained as

$$\begin{bmatrix} \mathbf{S}_{est} \\ \mathbf{1}_n^{\mathrm{T}} \otimes \mathbf{b}^{\mathrm{T}} \end{bmatrix} = [\mathbf{G} \quad \mathbf{1}]^{\dagger}\mathbf{X}\ , \tag{11.15}$$

where $\dagger$ denotes the pseudo-inverse and index est indicates the estimated value. Considering $\mathbf{1}_n^{\mathrm{T}} \otimes \mathbf{b}^{\mathrm{T}}$ as a nuisance part we simply chop it obtaining $\mathbf{S}_{est}$. We now stick to the middle block row of $\mathbf{S}$ (see figure 10.11), name it $\tilde{\mathbf{S}}$ of size $M \times nM$ as the part that carries most of the energy. The estimates of the transmitted data symbols s_k for $k = 1, \cdots, n$ can be estimated from $\tilde{\mathbf{S}}_{est}$ given by

$$\tilde{\mathbf{S}}_{est} = [\mathbf{A}s_0 \quad \mathbf{A}s_1 \cdots \mathbf{A}s_n]\ .$$

The symbol estimation can be performed in two different ways:

1. by computation of the trace of the $[M \times M]$ data blocks that have the $\mathbf{A}s_k$ structure. This approach exploits the fact that $\mathbf{A}$ possesses a dominant diagonal structure with positive elements on the main diagonal.

2. by vectorizing the $[M \times M]$ blocks of the matrix $\tilde{\mathbf{S}}_{est}$ such that we get a rank one matrix whose row span corresponds to the data symbols. Vectorizing is performed over subsequent estimated $\mathbf{A}\hat{s}_k$ blocks. Vectors are stacked in a new matrix whose row span represents a scaled version of the estimated data symbols $\hat{s}_k$

$$\begin{aligned} [\mathrm{vec}(\mathbf{A}\hat{s}_0), \cdots, \mathrm{vec}(\mathbf{A}\hat{s}_{n-1})] &= \mathbf{a}[\hat{s}_0 \quad \hat{s}_1 \quad \cdots \hat{s}_{n-1}] \\ &= \mathbf{a}\hat{\mathbf{s}} \end{aligned} \tag{11.16}$$

   where $\mathbf{a} = \mathrm{vec}(\mathbf{A})$ is a stacking of the columns of a matrix $\mathbf{A}$ into a vector of size $M^2 \times 1$. Vector $\mathbf{s} : 1 \times n$ and $\hat{\mathbf{s}} : 1 \times n$ are row vectors of transmitted data symbols and their estimates.

Here we presented a simple estimator that neglects the unknown channel parameters contained in $\mathbf{A}$ and $\mathbf{b}$.

11.4 Simulations

The performance of the proposed algorithm is first tested for a single user in noise scenario.

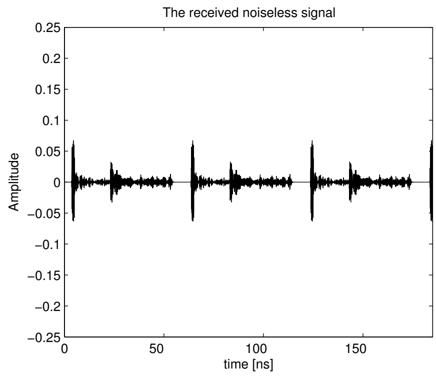
Signals are generated in accordance to the description provided in chapter 10. 250 Monte-Carlo runs are performed for fixed polarity- and time hopping-codes. Data symbols and noise samples are varied in each run as well as the packet offset that is randomly chosen from the interval $[0, N_cT_c)$. We consider the transmission of $N_s = n = 30$ data symbols, the polarity code length of $N_c = 15$ chips and $M = 3$ possible delays, $D_1 = 0.7$ns, $D_2 = 1.4$ns and $D_3 = 2.1$ns. Data pulses are chosen to be Gaussian pulses of duration 10ps, the same as in the measurement setup described in [66]. The transmitted monocycles are convolved with the channel impulse responses measured for different scenarios in a typical university building. The antenna responses at the transmit and the receive side are included in the channel impulse response. The following scenarios are taken into account: 1) office, 2) corridor, 3) corridor-to-office, 4) library, and 5) office-to-office. Both line of sight and non line of sight channel impulse responses are covered in this fashion (see [66]). Using stroboscopic sampling, a sampling rate of 10ps is used in the channel measurements.

After the receive antenna, where the white Gaussian noise is added, a bandpass filtering is performed to limit the bandwidth of the observed signal to the interval $4 - 10$GHz. The filtering step reduces the impact of the noise and the low frequency interference. We limit the measured channel impulse responses to the interval $[0, 50]$ ns, as the contribution of the channel components that fall outside this interval is insignificant (see for example subplot (b) in figure 9.4). The duration of the frame is chosen to be $T_f = 60$ns. The energy of a single transmitted data symbol is defined as $E_b = 2N_dN_c \int [h(t)]^2 dt$ where h denotes the total channel impulse response including the pulse shape and antenna responses, before the band pass filtering at the receiver. In the computation of the E_b both reference and information pulses carry the energy and therefore coefficient 2 is included in the expression. Noise power N_0 is computed in the frequency band $4 - 10$GHz. The E_b/N_0 is changed in steps of 1dB.

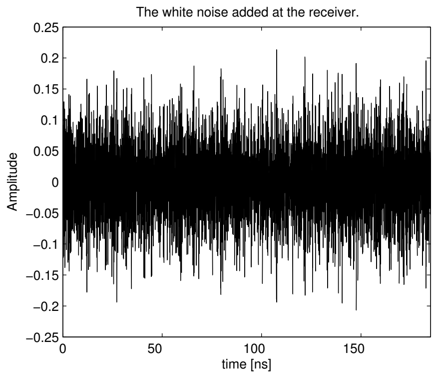
At the output of the autocorrelation receiver with three receiver branches the oversampling is performed at rate equal to the number of frames per chip $i.e.$, $P = N_d = 3$. Note that the oversampling rate $P = 3$ is still lower than the Nyquist rate for the expected pulse shapes at the integrators' output but is sufficient as the distortions in the estimates of τ are negligible compared to sampling at the Nyquist rate [51].

An example of a received single user signal and noise (after bandpass filtering) is presented in figure 11.2. In this figure $E_b/N_0 = 34$dB. At this E_b/N_0, the useful signal will (for the chosen parameters N_c and N_d) clearly drown in the noise, yet the proposed method can synchronize, as will be illustrated next.

Figure 11.3 shows the percentage of cases where the packet offset is estimated



(a)



(b)

Figure 11.2: Received single user signal (a) and noise (b) ($E_b/N_0 = 34$dB). Note that the given value of $E_b/N_0 = 34$dB includes the effects of integration over 1 data symbol (N_c chips and N_d frames). However, each doublet is dieply buried in the noise.

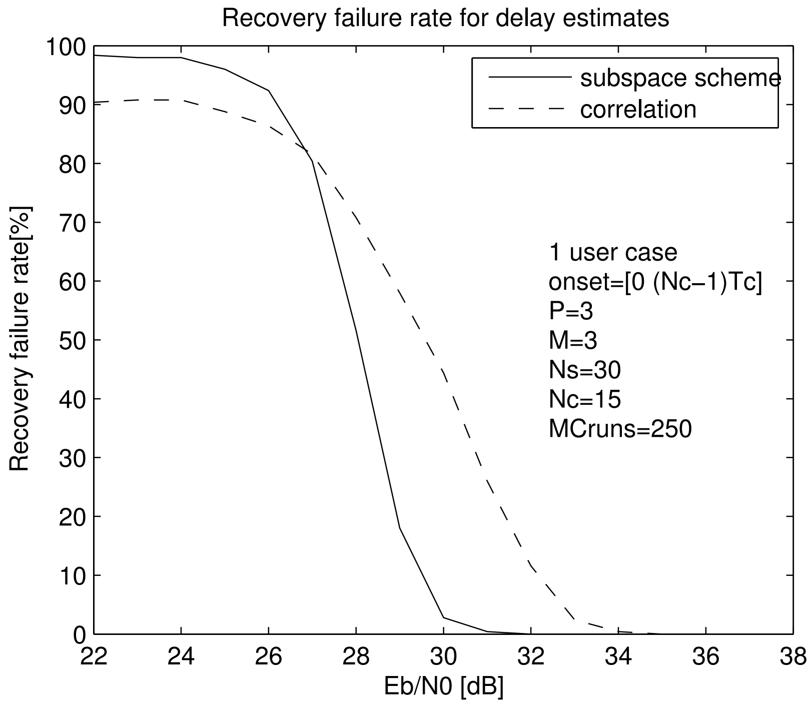


Figure 11.3: The percentage of incorrectly estimated packet offsets using the proposed subspace based (solid) and correlation of the template with the received signal approach (dashed).

wrongly. An estimate is considered to be wrong if it does not fall into the interval $\tau - T_c/2 \le \hat{\tau} < \tau + T_c/2$. The solid line denotes the performance of the delay estimation based on the subspace based frequency-domain search (as we presented in this chapter), while the dashed line shows the performance of the correlation-based scheme

$$\underset{\tau}{\operatorname{argmin}} \parallel \mathbf{X}^{\mathrm{H}} \mathbf{G}_\tau \parallel^2 ,$$

which can be solved in a similar fashion as the subspace-based scheme, but which does not recquire a subspace decomposition.

Figure 11.4 shows the standard deviation of the 'good' estimates of τ for subspace based (solid line) and correlation estimation schemes(dashed). In figure 11.5 the BER of the symbol estimates is shown for all estimates of τ. We deployed a decorrelating receiver and the vectorization approach described in Section 11.3.

From the simulations we can conclude that the subspace-based synchronization

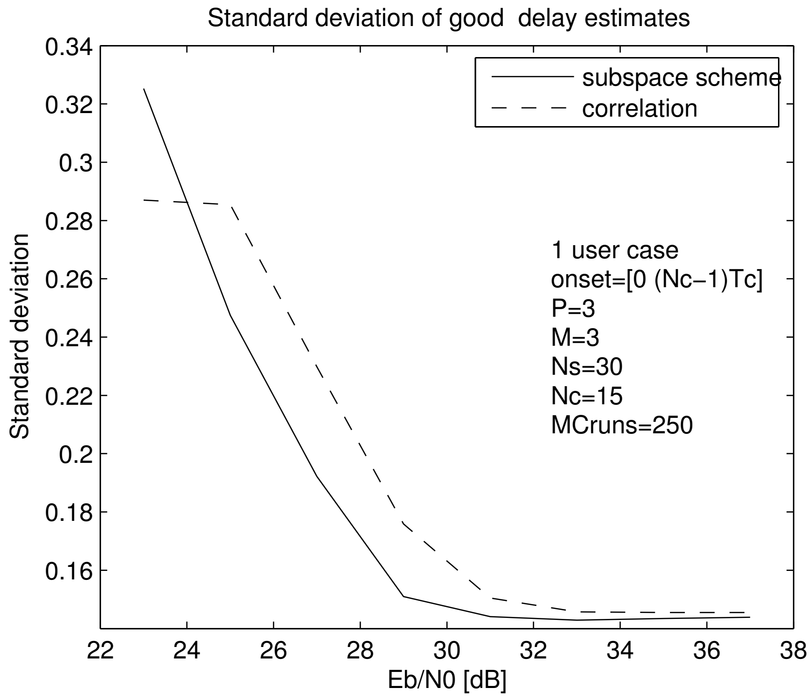


Figure 11.4: Standard deviation of the correctly estimated packet offset delays.

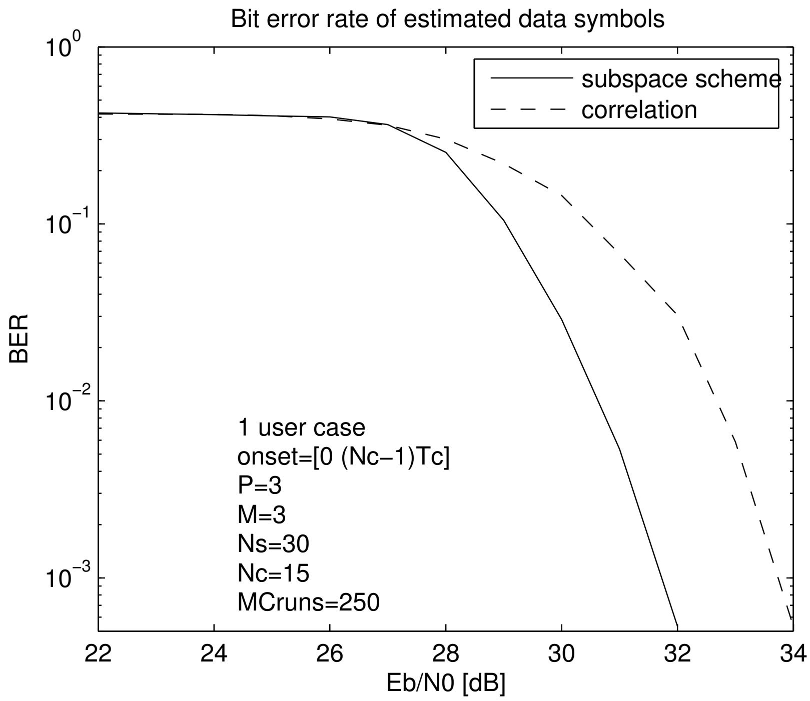


Figure 11.5: BER of the estimated data symbols computed for all estimates of τ.

scheme outperforms the correlation-based approach. However the complexity of the first scheme is dominated by the singular value decomposition which complexity is of order $n(2N_cP)^2$, where n denotes the number of received data blocks $\mathbf{X}_i$ collected in the received data matrix $\mathbf{X}$ (see 10.16).

In this chapter we have derived a deterministic, block based synchronization and detection scheme and verified its performance in the presence of white noise. Exploiting the property that a time shift in time domain corresponds to the phase progression in the frequency domain we developed the algorithm that provides high resolution (fraction of the chip precision) packet offset estimates. Exploiting the properties of fast Fourier transformation (FFT) the complexity of the algorithm is reduced comparing to the correlation based approaches. Moreover, the simulation proved that our subspace-based synchronization scheme outperforms the correlation based estimator.

In the next chapter we extend our data model to the multi-user case. The effects and the performance of the autocorrelation receiver in such a scenario are studied in detail.

Appendix

11.5 Blind Multiuser Channel Estimation in Asynchronous CDMA Systems

In this appendix we present a subspace based, blind channel estimation scheme for asynchronous multiuser CDMA systems proposed in [83] by Torlak and Xu. The algorithm provides the estimates of users' delays that are subsequently used to estimate the multiuser channels. One of the processing steps of this algorithm is used in the derivation of our TR-UWB synchronization scheme. For this reason and for the consistency of exposure the Torlak's algorithm from [83] is presented in brief.

The algorithm is developed for a no-noise case. An asynchronous CDMA (A-CDMA) baseband model for P active users at the receiver side becomes

$$x(t) = \sum_{i=1}^{P} \sum_{n=-\infty}^{\infty} s^i(n) g^i(t - nT_s) \tag{11.17}$$

where $s^i(n) \in \{\pm 1\}$ represents the ith user data symbol of duration T_s. Define $[c^i(1),\ c^i(2), \cdots, c^i(L_c)]$ as the spreading code of the ith user, where $c^i(k) \in \{\pm 1\}$. The signature waveform of the ith user now becomes

$$g^i(t) = \sum_{k=1}^{L_g^i L_c} c^i(k - k^i) h^i(t - kT) \ , \tag{11.18}$$

where $h^i(t)$ denotes the realizations of the channels between transmitters $i = [1, \cdots, P]$ and the receiver. Further, k^i and T are the duration and the delay index of a chip. The multipath channel impulse response is modeled as

$$h(t) = \sum_{q=1}^{L_d} \alpha_q p(t - \tau_q) \ , \tag{11.19}$$

where L_d denotes the total number of paths, τ_q and α_q are the delay and the complex gain of the ith path, respectively. $p(t)$ is a pulse shaping function that provides the desired spectral shape of the signal. Assume that an observation block at the receiver side corresponds to the symbol length $T_s = L_c T$. The relative offset between the beginning of such a block and the leading channel component that falls into it, can take any value from interval $[0, T_s]$. The channel is assumed to have a finite, non-zero support much shorter than one symbol interval, $LT \ll L_c T$. Accordingly, the signature waveform is, in general, spread over three symbol periods but for simplicity of exposure and without loss of generality the authors assume it is restricted to two symbol intervals $L_g = 2$.

Sampling the $x(t)$ from (11.17) at the chip rate yields

$$x(l) = \sum_{i=1}^{P} \sum_{n=-\infty}^{\infty} s^i(n) g^i(l - nL_c) \ . \tag{11.20}$$

The discretized version of the signature waveform in (11.18) becomes

$$\begin{aligned} g^i(l) &= \textstyle\sum_{k=1}^{2L_c} c^i(k-k^i)h^i(l-k) \\ &= \textstyle\sum_{k=1}^{L} h^i(k)c^i(l-k+k^i) \ , \quad l = 1, \cdots, 2L_c \ , \end{aligned} \tag{11.21}$$

where k^i denotes the chip delay index. This can be written in matrix form as

$$\begin{aligned} \bar{\mathbf{G}}^i &= \bar{\mathbf{C}}^i \mathbf{h}^i \\ \begin{bmatrix} \bar{\mathbf{G}}_1^i \\ \bar{\mathbf{G}}_2^i \end{bmatrix} &= \begin{bmatrix} \bar{\mathbf{C}}_1^i \\ \bar{\mathbf{C}}_2^i \end{bmatrix} \mathbf{h}^i \end{aligned} \tag{11.22}$$

where the *kernel* matrix $\bar{\mathbf{C}}^i$ is of size $2L_c \times L$, $\bar{\mathbf{G}}^i : 2L_c \times L$, $\bar{\mathbf{C}}_j^i : L_c \times L$ and $\bar{\mathbf{G}}_j^i = \bar{\mathbf{C}}_j^i \mathbf{h}^i$ for $j = 1, 2$. The channel multipath components for the ith user are contained in the vector $\mathbf{h}^i : L \times 1$. In more detail the structure of the introduced matrices are as follows:

$$\begin{bmatrix} \bar{\mathbf{G}}^i(1) \\ \bar{\mathbf{G}}^i(2) \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0_{k^i} & 0 & \cdots & 0 \\ c^i(1) & 0 & \ddots & 0 \\ c^i(2) & c^i(1) & \ddots & 0 \\ \vdots & c^i(2) & \ddots & c^i(1) \\ c^i(L_c) & \vdots & \ddots & c^i(2) \\ 0 & c^i(L_c) & \ddots & \vdots \\ 0 & 0 & \ddots & c^i(L_c) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} h^i(1) \\ h^i(2) \\ \vdots \\ h^i(L) \end{bmatrix} \tag{11.23}$$

The structure of the $\bar{\mathbf{G}}^i$ shows effect of spreading a coded data symbol at the receiver side over two consecutive symbol intervals $i.e.,\ 2T_s = 2L_cT$, where T represents the chip duration.

The received data samples in the rth symbol interval are stacked in a vector $\mathbf{x}(r)$

$$\mathbf{x}(r) = \begin{bmatrix} x[rL_c+1] \\ x[rL_c+2] \\ \vdots \\ x[rL_c+L_c] \end{bmatrix} . \tag{11.24}$$

At this point a block Hankel matrix that collects the contribution of $N+K-1$ subsequent symbol intervals of user i - $\mathbf{X}^{(i)}$ is created as

$$\mathbf{X}^i = \begin{bmatrix} \mathbf{x}(0) & \mathbf{x}(1) & \cdots & \mathbf{x}(N-1) \\ \mathbf{x}(1) & \mathbf{x}(2) & \cdots & \mathbf{x}(N) \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{x}(K-1) & \mathbf{x}(K) & \cdots & \mathbf{x}(N+K-2) \end{bmatrix} , \tag{11.25}$$

where K is defined as a *smoothing factor*. For simplicity of exposure, in future we stack vertically only two $\mathbf{x}(r)$ vectors providing the presence of a complete signature waveform matrix $\bar{\mathbf{G}}^i$ in the left signal subspace of $\mathbf{X}^i$. The authors in [83], however, generalize the approach for any K.

The relation between the received data blocks - $\mathbf{x}(r)^i$, the *kernel* matrix $\bar{\mathbf{C}}^i$, the channel impulse response $\mathbf{h}^i$ and the transmitted data symbols $s^i(r)$ of each user i is given by

$$\mathbf{X}^i = \mathbf{G}^i \mathbf{S}^i \; ,$$

where $\mathbf{G}^i = \mathbf{C}^i \mathbf{H}^i$ with $\mathbf{C}^i$ of size $2L_c \times 3L$

$$\mathbf{C}^i = \begin{bmatrix} \bar{\mathbf{C}}_2^i & \bar{\mathbf{C}}_1^i & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{C}}_2^i & \bar{\mathbf{C}}_1^i \end{bmatrix} ,$$

and $\mathbf{H}^i$ of size $3L \times 3$ defined as

$$\mathbf{H}^i = \begin{bmatrix} \mathbf{h}^i & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{h}^i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{h}^i \end{bmatrix} .$$

The transmitted data symbols are collected in a $3 \times N$ matrix $\mathbf{S}^i$

$$\mathbf{S}^i = \begin{bmatrix} s^i(1) & s^i(2) & \cdots & s^i(N) \\ s^i(2) & s^i(3) & \cdots & s^i(N+1) \\ s^i(3) & s^i(4) & \cdots & s^i(N+2) \end{bmatrix} .$$

The superposition of received data signals from all $i = 1, \cdots, Q$ users yields a multiuser data model

$$\mathbf{X} = \sum_{i=1}^{Q} \mathbf{X}^i = \mathbf{G}\mathbf{S} \tag{11.26}$$

where $\mathbf{G} = [\mathbf{G}^1 \cdots \mathbf{G}^P]$ and $\mathbf{S} = [\mathbf{S}^{1T} \cdots \mathbf{S}^{QT}]^{\mathrm{T}}$.

11.6 Multiuser channel estimation

The multiuser data model in (11.26) represents the starting point for the channel estimation algorithm in [83]. The goal is to estimate the channel impulse response $\mathbf{h}^i$ in a blind fashion, using only the knowledge of the user spreading code but not the knowledge of the transmitted data symbols. In a noisy case (11.26) extends to

$$\mathbf{X} = \mathbf{G}\mathbf{S} + \mathbf{N} \; .$$

Matrix $\mathbf{G}$ is in the left signal subspace of $\mathbf{X}$ as well as any of the signature matrices $\mathbf{G}^i$. Using the property that the signal subspace is orthogonal to the nullspace yields

$$\mathbf{U}_0 \perp \mathbf{G} \quad \text{so that in particular} \quad \mathbf{U}_0^{\mathrm{H}}\mathbf{G}^i = \mathbf{0} \; ,$$

that is valid for any user $i = 1, \cdots, Q$. The complex conjugate transpose operation is depicted by $(\cdot)^{\mathrm{H}}$. Employing the known structure of $\mathbf{G}^i$ gives

$$\mathbf{U}_0^{\mathrm{H}}\mathbf{C}^i\mathbf{H}^i = \mathbf{0} \; . \tag{11.27}$$

The authors now propose to transform the latter expression such that it becomes a function of an unknown vector $\bar{\mathbf{C}}^i\mathbf{h}^i$ rather than matrix $\mathbf{C}^i\mathbf{H}^i$. This step facilitates the estimation of the channel impulse response. Now (11.27) becomes

$$\hat{\mathbf{h}}^i = \arg \min_{\|\mathbf{h}^i\|=1} \mathbf{h}^{iH}\mathbf{c}^{iH}[\sum_l \tilde{\mathbf{U}}_l\tilde{\mathbf{U}}_l^{\mathrm{H}}]\mathbf{c}^i\mathbf{h}^i \; , \tag{11.28}$$

where l sweeps all the columns from the left zero subspace. Further, the vectors of $\mathbf{U}_0$ are transformed in accordance with

$$\tilde{\mathbf{U}}_l = \begin{bmatrix} \mathbf{u}_{l,1} & \mathbf{u}_{l,2} & \mathbf{0} \\ \mathbf{0} & \mathbf{u}_{l,1} & \mathbf{u}_{l,2} \end{bmatrix} \; ,$$

where $\mathbf{u}_{l,1} \; : \; L_c \times 1$ and $\mathbf{u}_{l,2} \; : \; L_c \times 1$ are the first and the second half of the lth column of the zero subspace $i.e.$,

$$\mathbf{U}_l = \begin{bmatrix} \mathbf{u}_{l,1} \\ \mathbf{u}_{l,2} \end{bmatrix} \; .$$

Note that in the proposed scheme the relative delay of the channel impulse response was considered to be known. However, the initial synchronization is crucial for the performance of the algorithm. The approach proposed by Torlak and Xu is based on the search for the best possible $\mathbf{g}^i$ for each user. Defining

$$\mathbf{Q}_k^i = \bar{\mathbf{C}}_k^{iH}[\sum_l \tilde{\mathbf{U}}_l\tilde{\mathbf{U}}_l^{\mathrm{H}}]\bar{\mathbf{C}}_k^i$$

where l sweeps all the vectors from the nullspace and $k = 0, \cdots, L_c$ denotes the possible shifts of the *kernel* matrix defined in (11.22). The channel and delay estimation are further performed by optimizing

$$J(\hat{\underline{\mathbf{h}}}^i, \hat{\underline{k}}^i) = \underline{\mathbf{h}}^{iH} \begin{bmatrix} \mathbf{Q}_1^i & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_2^i & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{Q}_{L_c}^i \end{bmatrix} \underline{\mathbf{h}}^i \tag{11.29}$$

where

$$\underline{\mathbf{h}}^i = [\underbrace{0 \cdots 0}_{Lk^i} \ \ \mathbf{h}^{iH} \ \ \underbrace{0 \cdots 0}_{L(L_c - k^i - 1)} \ ]^{\mathrm{H}} .$$

A power constraint $\|\mathbf{h}^i\| = 1$ is introduced to avoid a trivial all-zero solution.

The authors in [83] proposed a blind subspace based channel estimation and synchronization scheme for CDMA systems. The authors collect a block of the received data in $\mathbf{X}$ such that a user specific signature vector comprised of the channel impulse response $\mathbf{h}$ and a CDMA code $\mathbf{c}$ becomes orthogonal to the null space of $\mathbf{X}$. This fact is further used to estimate the channel impulse response and its unknown offset by means of a singular value decomposition.

In this chapter we have used a similar approach from [83] now for a different data model specific for a TR-UWB system. By deployment of an autocorrelation receiver in a TR-UWB system channel estimation is avoided but a specific data model arises. Realizing a specific form of a user specific code we were able to exploit the approach authors used in [83], namely the fact that the code is orthogonal to the null space of the received block matrix. In this manner we developed a blind, high resolution synchronization scheme presented in this chapter.

Chapter 12

Multiple users data model and blind synchronization algorithm – Asynchronous case

In this chapter we extend the asynchronous single user data model presented in Chapter 10 to the case of multiple active users. The deployment of the autocorrelation receiver in the transmit-reference UWB systems prevents us to simply extend the data model (10.17) in a linear fashion. This is due to the fact that an autocorrelation receiver introduces a number of crosscorrelation terms in a multiuser scenario. In this chapter we describe these crosscorrelation terms and extend the data model (10.17) to the case of asynchronous simultaneously active multiple users. Further, a practical aspect of the proposed scheme is presented, namely the ability of avoiding collisions of data packets even in case multiple users deploy the same spreading code. At the end of the chapter the performance of the synchronization scheme in a multiuser scenario is validated.

12.1 Cross correlation terms

The extension of the single user data model given in (10.17) to the multiuser case can in general not be achieved by simple linear extension. This is due to the deployment of the autocorrelation receiver. We will show this in the following example.

Similar to the derivation for a single user case presented in Section 9.1.1 we now consider two simultaneously active users that transmit a single doublet,

$$\begin{aligned} d^1(t) &= g^1(t) + c^1 \cdot s^1 \cdot g^1(t - D_i^1) \\ d^2(t) &= g^2(t) + c^2 \cdot s^2 \cdot g^2(t - D_i^2) , \end{aligned} \tag{12.1}$$

where the upper index denotes the ordinal number of the user ($p = 1, 2$). Each user broadcasts a random data symbol (s^p) and uses a specific polarity (c^p) and transmit delay D_i^p code for $i = 1, \cdots, M$. Here M denotes the maximal number of the transmit as well as receive delays. Let $h^1(t)$ and $h^2(t)$ represent the propagation channel of user 1 and user 2, respectively. The signal at the receiver antenna now becomes

$$r(t) = r^1(t) + r^2(t) = h^1(t) + c^1 \cdot s^1 \cdot h(t - D_i^1) + h^2(t) + c^2 \cdot s^2 \cdot h(t - D_i^2) \ .$$

To be able to perceive the mutual impact of multiple users in case an autocorrelation receiver is deployed, we assume that their contributions at the receiver side ($r^1(t)$ and $r^2(t)$) overlap.

In the first part of the autocorrelation receiver the received signal is passed through a delay bank with delays D_m, $m = 1, \cdots, M$. Subsequently, the delayed signal is multiplied with the non-delayed received signal to produce

$$y_m(t) = r(t)r(t - D_m)$$

where m denotes the ordinal number of the receiver branch (see figure 9.1). Similar as in (9.1.1) we can write $y_m(t)$ as

$$y_m(t) = y_m^{1,1} + y_m^{2,2} + y_m^{1,2} + y_m^{2,1} \ .$$

Here $y_m^{1,1}$ collects the products of unshifted and shifted version of the user 1 propagation channel, let us call them the *self-product* terms

$$y_m^{1,1} = [h^1(t) + c^1 \cdot s^1 h^1(t - D_i^1)][h^1(t - D_m) + c^1 \cdot s^1 \cdot h^1(t - D_i^1 - D_m)] \ . \tag{12.2}$$

In the same manner for the propagation channel of user 2 $i.e.,$ h^2 we write

$$y_m^{2,2} = [h^2(t) + c^2 \cdot s^2 h^2(t - D_i^2)][h^2(t - D_m) + c^2 \cdot s^2 \cdot h^2(t - D_i^2 - D_m)] \ . \tag{12.3}$$

The cross-product terms of the received signal of user 1 and user 2 are collected in the following terms

$$y_m^{1,2} = [h^1(t) + c^1 \cdot s^1 h^1(t - D_i^1)][h^2(t - D_m) + c^2 \cdot s^2 \cdot h^2(t - D_i^2 - D_m)] \ . \tag{12.4}$$

and

$$y_m^{2,1} = [h^2(t) + c^2 \cdot s^2 h^2(t - D_i^2)][h^1(t - D_m) + c^1 \cdot s^1 \cdot h^1(t - D_i^1 - D_m)] \tag{12.5}$$

These signals are further passed through the sliding window integrator of width W

$$x_m(t) = \int_{t-W}^{t} y_m(\tau)d\tau \quad m = 1, \cdots, M \ .$$

Following the strategy described in Section 9.1.1 we can write the signal at the output of the m-th integrator as

$$x_m(t) = x_m^{1,1}(t) + x_m^{2,2}(t) + x_m^{1,2}(t) + x_m^{2,1}(t) \ , \tag{12.6}$$

where the terms that contain the contribution of the self-products, $i.e.$, $x_m^{1,1}$ and $x_m^{2,2}$, respectively can be written as

$$\begin{aligned} x_m^{1,1}(t) &= b(t)\{2\rho^{1,1}(D_m) + c^1 \cdot s^1 \cdot [\rho^{1,1}(D_m - D_i) + \rho^{1,1}(D_m + D_i)]\} \\ &= b(t)(\alpha_{m,i}^{1,1} \cdot c^1 \cdot s^1 + \beta_m^{1,1}) \ , \end{aligned}$$

and

$$\begin{aligned} x_m^{2,2}(t) &= b(t)\{2\rho^{2,2}(D_m) + c^2 \cdot s^2 \cdot [\rho^{2,2}(D_m - D_i) + \rho^{2,2}(D_m + D_i)]\} \\ &= b(t)(\alpha_{m,i}^{2,2} \cdot c^2 \cdot s^2 + \beta_m^{2,2}) \ . \end{aligned}$$

where $\alpha_{m,i}^{p,p}$ and $\beta_m^{p,p}$ are the same as defined in (9.11) now with the upper index ^{p,p} that corresponds to the the user index p. Here we define

$$\rho^{p,q}(\Delta) = \int_0^\infty h^p(\tau)h^q(\tau - \Delta)d\tau \ \text{ for } \ p, q = 1, \cdots, Q \ . \tag{12.7}$$

as the auto- and cross-correlation of the channel impulse response(s) for $p = q$ and $p \neq q$, respectively. $Q = 2$ denotes the total number of users and $h^p(t)$ and $h^q(t)$ represent the channel impulse responses of user p and q, respectively.

The terms in (12.6) that contain the cross-products of the two channels now become

$$\begin{aligned} x_m^{1,2}(t) = \ & b(t)\{\rho^{1,2}(D_m) + c^1c^2s^1s^2\rho^{1,2}(D_m + D_i^2 - D_i^1) \\ & + c^2s^2\rho^{1,2}(D_m + D_i^2) + c^1s^1\rho^{1,2}(D_m - D_i^1)\} \end{aligned}$$

and

$$\begin{aligned} x_m^{2,1}(t) = \ & b(t)\{\rho^{2,1}(D_m) + c^1c^2s^1s^2\rho^{2,1}(D_m + D_i^1 - D_i^2) \\ & + c^1s^1\rho^{2,1}(D_m + D_i^1) + c^2s^2\rho^{2,1}(D_m + D_i^2)\} \end{aligned}$$

Note that the number of cross-correlation terms increases quadratically with the number of users. Hence, an increased amount of terms that we consider as interference will appear.

However, since different users employ distinct time-hopping and polarity codes, propagate through different channels, and arrive at the receiver at random time instants, we can assume that the the cross-correlation terms are small compared to the

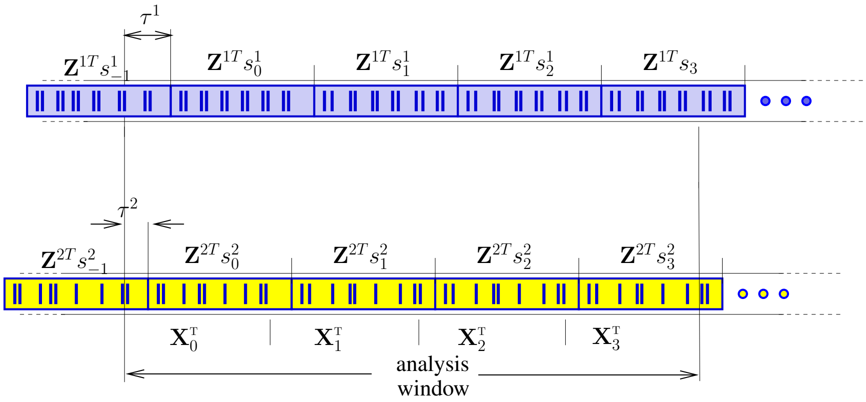


Figure 12.1: The structure of the analysis window and the received data blocks for the two user asynchronous case.

auto-correlation. This approximation is set for purpose of designing a simple multiuser receiver. In other words we treat these cross terms as additive white noise, and add them to the other noise terms that might be present. As before, we do not take the additive noise terms into account in our derivations, but we do include them in our simulations further in this chapter.

This assumption allow us to extend the single user data model from figure 10.11 in a linear fashion. Accordingly the algorithm used to synchronize to a single user described in Chapter 10 can be deployed for the synchronization of multiple users.

12.2 Multi-user matrix model

In this section, we extend the previous ideas developed for a single user, to multiple users. In figure 12.1 the structure of the received data blocks for the case of two asynchronous users is presented. The offset with respect to the beginning of the analysis window for the first and second user are denoted by τ^1 and τ^2, respectively.

As a result, indicating the user index by means of a superscript q ($q = 1, 2, \ldots, Q$) we define a received data block of user q as

$$\mathbf{X}^q = \begin{bmatrix} \mathbf{X}_0^q & \mathbf{X}_1^q & \cdots & \mathbf{X}_{n-1}^q \\ \mathbf{X}_1^q & \mathbf{X}_2^q & \cdots & \mathbf{X}_n^q \end{bmatrix},$$

where $n+1$ denotes the number of consecutive data blocks $\mathbf{X}_k^q$ of an analysis windows (see figure 12.1). The overall received data block matrix $\mathbf{X}$ collects the contributions

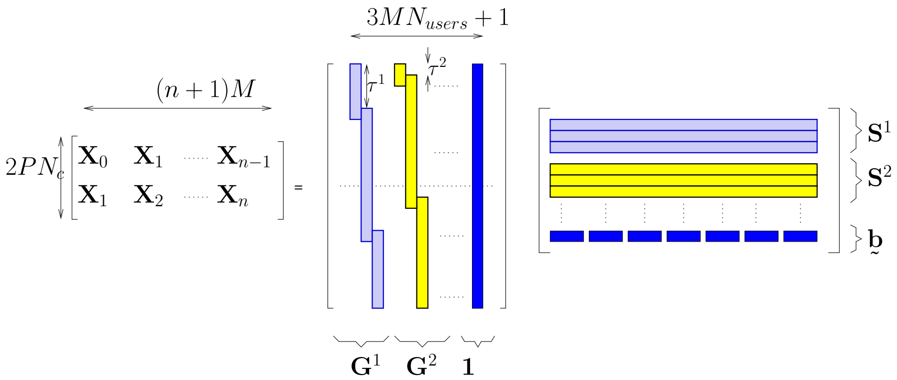


Figure 12.2: The structure of the asynchronous multiuser matrix data model. Different users possess different packet offsets.

of each individual user

$$\mathbf{X} = \sum_{q=1}^{Q} \mathbf{X}^q \; .$$

Based on the assumptions from the previous section we start by extending the data model of Section 10.2 equation 10.17 in a linear fashion to the multiple user case

$$\begin{aligned} \mathbf{X} &= \textstyle\sum_{q=1}^{Q} [\mathbf{G}^q \quad \mathbf{1}] \begin{bmatrix} \mathbf{S}^q \\ \mathbf{1}_n^{\mathrm{T}} \otimes \mathbf{b}^{q\,\mathrm{T}} \end{bmatrix} \\ &= [\mathbf{G}^1 | \cdots | \mathbf{G}^Q \mid \mathbf{1}] \begin{bmatrix} \mathbf{S}^1 \\ \vdots \\ \mathbf{S}^Q \\ \mathbf{1}_n^{\mathrm{T}} \otimes \sum_{q=1}^{Q} \mathbf{b}^{q\,\mathrm{T}} \end{bmatrix} \; , \end{aligned} \tag{12.8}$$

where the structures of $\mathbf{S}^q = [\mathbf{S}_0^q, \ldots, \mathbf{S}_{n-1}^q]$ and $\mathbf{G}^q$ are defined in Section 10.2. Further, $\mathbf{S}_k^q = \mathbf{A}^q s_k^q : M \times M$ represents a data symbol of user q affected by the correlation matrix $\mathbf{A}^q$ that corresponds to the channel impulse response of the qth user. Note that in the case some users are not active for the duration of the whole analysis window, several $\mathbf{S}_k^q$ matrices will be zero and some small changes in the structure of $\underset{\sim}{\mathbf{b}}^{\mathrm{T}} = \mathbf{1}_n^{\mathrm{T}} \otimes \sum_{q=1}^{Q} \mathbf{b}^{q\,\mathrm{T}}$ will occur. Consequently, a few additional vectors with low energy may emerge in the left signal subspace.

Identifiability for multi user case

In the multiuser (MU) case as presented in (12.8), the matrix $\mathbf{G}_{MU} = [\mathbf{G}^1|\cdots|\mathbf{G}^Q \mid \mathbf{1}]$ of size $[2N_cP \times (3MQ+1)]$ needs to be tall to be able to subsequently estimate data symbols. Moreover the matrix comprising all data blocks and the effect of offset vectors becomes $\mathbf{S}_{MU} = [\mathbf{S}^{1T},\ \cdots\ \mathbf{S}^{QT},\ \mathbf{1}_n \otimes \sum_{q=1}^{Q} \mathbf{b}^{(q)}]^{\mathrm{T}}$ is of size$[(3MQ+1) \times nM$. $\mathbf{G}_{MU}$ should be tall and of full rank in order to be invertible, $i.e.,\ 2N_cP > 3MQ+1$, while $\mathbf{S}_{MU}$ should be wide with linearly independent rows, $i.e.,\ 3MQ+1 < Mn$. From these conditions we see the limitations on the code size:$N_c > \frac{3MQ}{2P}$ that for typical values of $P=3$ and $M=4$ yields $N_c > 2Q$. The conditioning on the size of $\mathbf{S}_{MU}$ gives the relation between the number of users Q and the lowest number of symbols transmitted n, $i.e.,\ Q < \frac{Mn-1}{3M}$.

12.3 Synchronization and detection in multiuser scenario

For the estimation of the packet offset of the user of interest in a multiuser scenario the synchronization scheme presented in Chapter 11. In particular, we tend to minimize the expression in (11.1), now for the block code $\mathbf{G}^q$ of the user of interest q

$$\begin{aligned} &\operatorname{argmin}_\tau \parallel \mathbf{U}_0^{\mathrm{H}} \mathbf{G}^q \parallel^2 \\ = \operatorname{argmin}_\tau \textstyle\sum_i \parallel &\begin{bmatrix} \mathbf{u}_{1,i} \\ \mathbf{u}_{2,i} \end{bmatrix}^{\mathrm{H}} \begin{bmatrix} \mathbf{Z}_2^q & \mathbf{Z}_1^q & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_2^q & \mathbf{Z}_1^q \end{bmatrix} \parallel^2 , \end{aligned}$$

Following the strategy described in Sections 11.1 and 11.2 we can obtain a high resolution packet offset estimates up to the fraction of the oversampled chip interval, $i.e.,$ a fraction of T_c/P. Here T_c and P denote the chip interval and the oversampling rate, respectively.

After obtaining a packet offset estimate of user q we are able to reconstruct $\mathbf{G}^q$ and use it in the deconvolution of the received data block $\mathbf{X}$. A zero-forcing equalizer $[\mathbf{G}^q \quad \mathbf{1}]^\dagger$ applied to (12.8) yields

$$\begin{bmatrix} \mathbf{S}_{est}^q \\ \mathbf{1}_n^{\mathrm{T}} \otimes \mathbf{b}^{\mathrm{T}} \end{bmatrix} = [\mathbf{G}^q \quad \mathbf{1}]^\dagger \mathbf{X} ,$$

where † denotes a pseudo inverse. Chopping out the part $\mathbf{1}_n^{\mathrm{T}} \otimes \mathbf{b}^{\mathrm{T}}$ we can estimate the data symbols from $\mathbf{S}_{est}^q$ using one of the strategies presented in Section 11.3.

12.4 Algorithm remarks

In this section we present the ability of our synchronization and detection scheme to cope with the packet collision problem that in general occurs when two or more users employ the same spreading code and have packets that arrive at the receiver at close or even the same time instants.

Consider a two-user system where both users adopt the same spreading code. The data model (12.8) then becomes:

$$\mathbf{X} = [\mathbf{G}_{\tau_1}\ \ \mathbf{G}_{\tau_2}\ \ \mathbf{1}] \begin{bmatrix} \mathbf{S}^1 \\ \mathbf{S}^2 \\ (\mathbf{1}_n \otimes \sum_{q=1}^{Q} \mathbf{b}^q)] \end{bmatrix} \tag{12.9}$$

The synchronization to the common code used by several users simultaneously and subsequent data estimation is in general possible if $\tau_i \neq \tau_j$ for $i \neq j$ and by implementation of a common code that has a low autocorrelation property.

Note that even if two users utilize the same spreading code and completely overlap in time it is still possible to separate both overlapping users and detect their data sequences. In that case the linear dependency between $\mathbf{G}_{\tau_1}$ and $\mathbf{G}_{\tau_2}$ reduces the rank of the code matrix, *i.e.*, $[\mathbf{G}_{\tau_1}\ \ \mathbf{G}_{\tau_2}] \rightarrow [\mathbf{G}_{\tau_{1,2}}]$. As a consequence, data blocks $\mathbf{S}^1$ and $\mathbf{S}^2$ merge into a single block $\mathbf{S} = \mathbf{S}^1 + \mathbf{S}^2$ changing (12.9) to

$$\mathbf{X} = [\mathbf{G}_{\tau_1=\tau_2}\ \mathbf{1}][\mathbf{S}^{\mathrm{T}}\ \ (\mathbf{1}_n \otimes \sum_{q=1}^{2} \mathbf{b}^q)]^{\mathrm{T}} .$$

Estimating the packet offset delay $\tau_1 = \tau_2$ we can reconstruct $\hat{\mathbf{G}}_{\tau_1}$ and subsequently, as in (Section 11.3) obtain an estimate of the data matrix $\mathbf{S}_{est}$. Considering only the mid block row of $\mathbf{S}_{est}$ (as in figure 10.11) we get $\tilde{\mathbf{S}}_{est}$. Finally, $\tilde{\mathbf{S}}_{est}$ can be modeled as

$$\tilde{\mathbf{S}}_{est} = [\mathbf{A}^1\hat{s}_1^1 + \mathbf{A}^2\hat{s}_1^2, \cdots, \mathbf{A}^1\hat{s}_n^1 + \mathbf{A}^2\hat{s}_n^2] , \tag{12.10}$$

where $\mathbf{A}^q : M \times M$ collects the channel coefficients of user $q = 1, 2$. Performing the vectorization of each $M \times M$ block of $\tilde{\mathbf{S}}_{est}$ yields

$$\begin{aligned} \tilde{\mathbf{S}}_{est,vec} &= [\mathrm{vec}(\mathbf{A}^1\hat{s}_1^1 + \mathbf{A}^2\hat{s}_1^2), \quad \cdots, \quad \mathrm{vec}(\mathbf{A}^1\hat{s}_n^1 + \mathbf{A}^2\hat{s}_n^2)] \\ &= [\mathbf{a}^1\ \mathbf{a}^2] \begin{bmatrix} \hat{s}_1^1 & \hat{s}_2^1 & \cdots & \hat{s}_n^1 \\ \hat{s}_1^2 & \hat{s}_2^2 & \cdots & \hat{s}_n^2 \end{bmatrix} \\ &= [\mathbf{a}^1\ \mathbf{a}^2] \begin{bmatrix} \hat{\mathbf{s}}^1 \\ \hat{\mathbf{s}}^2 \end{bmatrix} , \end{aligned}$$

where $\hat{\mathrm{s}}^1$ and $\hat{\mathrm{s}}^2$ are of size $1 \times n$ and collect data symbol estimates of user 1 and 2, respectively. Further, $\mathbf{a}^1$ is obtained by stacking vertically the columns of $\mathbf{A}^i$.

After performing the singular value decomposition of $\tilde{\mathbf{S}}_{est,vec}$ vectors $\hat{\mathbf{a}}^1$, $\hat{\mathbf{a}}^2$ and data symbols vectors $\hat{\mathbf{s}}^1$ and $\hat{\mathbf{s}}^2$ can be estimated from the column and row span of $\tilde{\mathbf{S}}_{est,vec}$. This approach fails only in the case when $\mathbf{A}^1 = \alpha\mathbf{A}^2$ where α is a scalar, but this has an extremely low probability of occurrence as each user propagates through a different channel. Matrix $[\mathbf{a}^1\ \mathbf{a}^2]$ should be tall and full rank which put the constraint on the maximal number of users $R < M^2$.

12.5 Application in UWB networking

The ability to achieve high resolution packet offset estimation in a multi-user environment in a fast and computationally simple way is of crucial interest for the subsequent step of data symbol estimation. Imagine the scenario of a UWB ad hoc network where users need to exchange their codes at the moment they join the network. The simplest way to solve this problem is to implement a *common code* known to all the users in the initialization phase as presented in the previous section. In existing wireless network protocols a data packet is considered to be lost if several users simultaneously use the same code which is known as the packet collision problem. Nevertheless, the structure and the design of the TR-UWB scheme will allow us to avoid the collision problem. In TR-UWB systems different users propagate through different channels creating distinct correlation matrices $\mathbf{A}^q$. The impact of the correlation matrices can be viewed as an additional coding introduced by the channel itself and is applicable to solving the collision problem in a simple way, as presented in Section 12.4.

12.6 Simulations

In this section we test the resistivity of the proposed synchronization scheme to the multiuser interference. In accordance to this approach we consider a noiseless scenario. We define the signal to interference ratio (SIR) as $SIR = 10\log(P_1/P_I)$. P_1 represents the energy of a single data symbol of the user of interest $i = 1$ at the receiver after the filtering with a bandpass filter of $4-10$GHz step at the receiver side. As each data symbol is spread over N_c chips and further over N_d frames containing two pulses (a doublet) the signal energy can be expressed as $P_1 = 2N_dN_c\int[h^1(t)]^2dt$, where $h^1(t)$ is the channel corresponding to the user of interest $i = 1$. $P_I = \sum_{i=2}^{Q} P_i$ collects the energy P_i of all interfering sources $i = 2, \cdots, Q$. An example of a single user signal in the noiseless case, after propagation through the channel and subsequent filtering at the receiver is presented in figure 12.3(a). Further, in figure 12.3(b), a mixture of three users corresponding to $SIR = -10$dB is depicted. Note that for both figures, the x-axis represents the number of samples, where sampling is performed at a rate of 10ps. From figure 12.3 we can see that the signal of the user of interest $i = 1$ is completely buried in the superposition of two interfering signals.

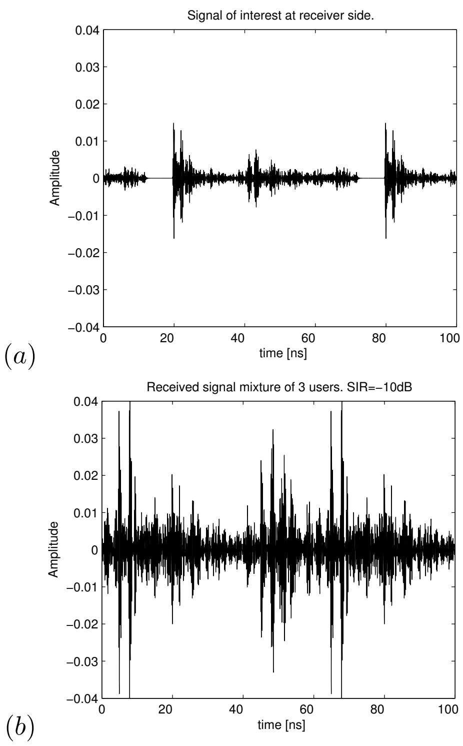


Figure 12.3: Received signal of the user of interest (a) and the mixture of three users at the receiver side (b) for the signal to noise ratio $SIR = -10$dB.

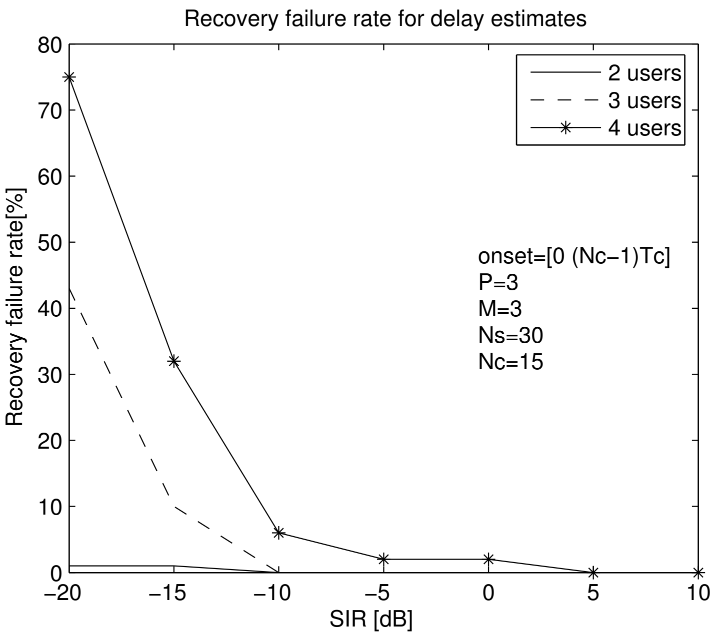


Figure 12.4: The percentage of incorrectly estimated packet offsets.

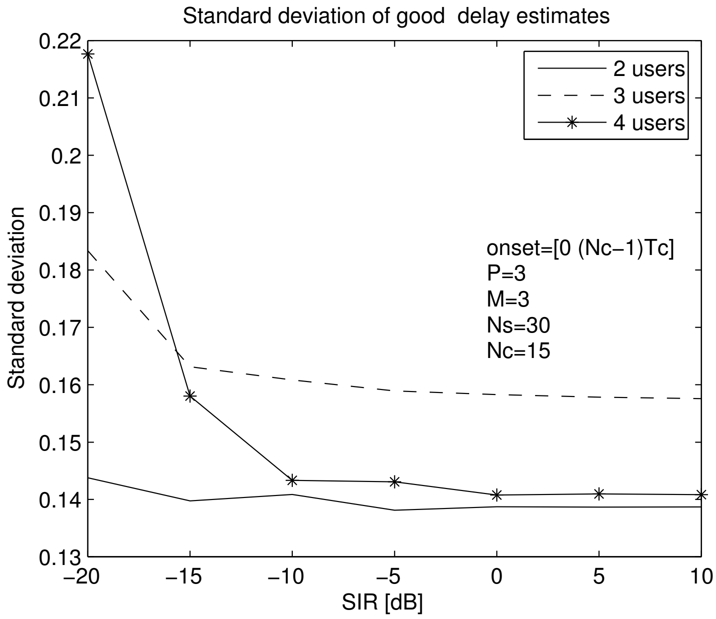


Figure 12.5: Std of the correctly estimated packet offset delays.

In figure 12.4, we present the recovery failure rate versus the signal to interference ratio. Any packet offset estimate $\hat{\tau}$ that does not fit into the range $\tau - T_c/2 < \hat{\tau} < \tau + T_c/2$ is considered to be a failure. The chosen interval is considered to provide sufficient recovery of the energy after the deployment of a decorrelating receiver. Initially, we choose a TH and CDMA code. A Gold code sequence of length N_c is chosen for the latter one. Both codes remain unchanged in all Monte-Carlo runs. In each run, a new set of channels as well as packet offsets is assigned to each of the users and is kept the same for all the values of the SIR changed in steps of 5dB. $N_s = 30$ stands for the number of data symbols within a transmitted packet. The oversampling rate $P = 3$ equals the number of frames (doublets) per chip. Delays used in the time hopping scheme are chosen to be $D_1 = 1$ ns, $D_2 = 2$ ns, and $D_3 = 3$ ns. In figure 12.4, the solid, dashed and dash-asterisked line corresponds to the two, three and four user case, respectively. The performance of the algorithm drops by increasing the total number of users. This can be explained by an augmented influence of the cross-correlation terms as the number of users increases. However, in the four user case, the algorithm exhibits a low failure rate even in the event of $SIR = 0$dB, $i.e.,$ in the case the energy of the signal of interest equals the energy of all interfering sources. This issue could be improved by a selection of user codes that would have low cross-correlation properties for any code offset.

Figure 12.5 describes the standard deviation of the estimates of τ, $i.e.,$ $\hat{\tau}$. The y-axis depicts the deviation expressed as a fraction of the chip duration T_c. Due to the fact that packet offsets of all users are randomly chosen in each Monte-Carlo run and to the unresolvable ambiguity related to the initial sampling point, the '3 user' scenario has a slightly degraded performance compared to the other scenarios.

In this chapter we have developed a multiuser data model based on the assumption that the cross-correlation terms that arise at the output of the autocorrelation receiver may be neglected (treated as noise). For the given multiuser data model we have verified the ability of our high resolution synchronization scheme to estimate the offset of the user of interest. The algorithm has shown the excellent performance in the presence of multiuser interference. In addition, we have shown the capability of our synchronization and detection scheme to combat the packet collision problem. Therefore it can be used for fast initial code exchange in multiuser asynchronous TR-UWB ad hoc networks.

12.7 Conclusions

In Part II of the thesis we considered a transmit-reference UWB system where a reference pulse is transmitted next to the information pulse. Both pulses propagate through the same multipath channel. At the receiver side the received signal is passed through a bank of delays and subsequently integrated. In this manner computationally complex channel estimation process is avoided but the energy smeared by the channel is collected. Accomplishing the correlation and integration steps in analog domain the need of sampling the received signal at Nyquist rate ($\sim (10 - 100)$GHz) is side-stepped. The sampling of the signal is performed just after the integration step, now at a rate much lower that Nyquist $\sim (10 - 100)$MHz that can easily be achieved with the current state of technology. Thus, transceiver properties of the presented TR-UWB system (low power consumption, low-complexity, sampling below the Nyquist, data rates up to 100Mbps) gives it great potential in the actual telecom market.

Appart from an advantageous TR-UWB transceiver design signal processing issues raised questions whether a synchronization and detection can be achieved in an efficient way. In Part II we have developed a data model for a packet oriented asynchronous TR-UWB system. For this model we have designed a subspace-based synchronization scheme that performs over a block of received data and provides a single-shot solution. Thus, a widely spread (and computationally demanding) approach of an exhaustive search for the best match of the user's template to the received data sequence is avoided. In the simulations we have seen that our synchronization scheme provides high resolution packet offset estimates and that it outperforms the exhaustive search approach. It also proved the superior synchronization capabilities in a multiuser environment. In the last Chapter we discussed the ability of our synchronization and detection scheme to resolve a packet collision problem even in cases multiple users simultaneously deploy the same spreading code.
Simplicity, computational efficiency, excellent performace in asynchronous multiuser environments and the ability to resolve the packet collision problem make our synchronization and detection scheme highly applicable in the networks without any centralized control, $i.e.,$ ad hoc Personal Area Networks and Wireless Local Area Networks.

Summary

A mobile asynchronous ad hoc network should be organized such that it enables connection of mobile users without the use of a base station. Multiple users (devices) randomly join the network and may transmit data packets at the same time. In current systems, with multiple co-channel users if two packets are overlapping both are lost and have to be retransmitted. Obviously, this transceiver scheme will break down if the number of users becomes too large. To overcome the collision and interference problems we propose transceiver schemes where each user is assigned a specific code. The code is used to identify the user and also to determine the beginning of the data packet within an analysis window.

In this work, we consider two cases:

- **A narrowband system** – To combat interference in an uncontrolled environment, we propose to introduce a small antenna array at the receiver side. The desired transmitter can then be separated from the competing signals by combining antenna outputs (beamforming or interference nulling). To distinguish desired user from the interfering sources a small, known modulus variation (color code) is inserted over a constant modulus (CM) signal. The code is a superimposed training sequence, which multiplies the transmitted signal without increasing the transmission rate. These features facilitate subsequent user separation and suppression of the interfering sources. We propose combined blind synchronization and detection schemes for a narrowband system that can comply with existing Bluetooth and WLAN standards. Here, blind refers to the fact that the transmitted data sequence is unknown.
- **An ultra wideband (UWB) system** – is a carrier-less wireless communication system where the information is transmitted using narrow pulses (each shorter than a ns). These pulses have extremely wide bandwidth and thus can be used to establish high data rate ($\sim$100Mb/s) communications at short distances (order of 10m). The system studied in this thesis is a ”transmit reference impulse system, where data is transmitted using pairs of pulses which captures the energy of a rich multipath channel by means of correlation and integration at the receiver side. There is no need for channel estimation and the sampling rate is

highly reduced facilitating synchronization and symbol estimation. Two types of codes are combined to distinguish multiple users – *time-hopping* defined by the distance between the pulses and a *polarity code*– similar to the codes used in CDMA.

In the second part of the thesis we have derived the algorithm for combined blind synchronization and data symbol estimation in transmit-reference UWB (TR-UWB) systems. This synchronization scheme allows for fast data exchange between users in an ad-hoc UWB network.

Samenvatting

Een mobiel asynchroon ad hoc netwerk moet zo georganiseerd worden dat het meerdere simultane gebruikers toelaat zonder centrale sturing door een basisstation. Meerdere gebruikers (apparaten) kunnen op hetzelfde moment data-pakketten versturen. In huidige systemen heeft in het geval van twee overlappende pakketten het verlies van beide pakketten tot gevolg—zij zullen opnieuw verzonden moeten worden. Als het aantal gebruikers te groot wordt bezwijkt dit systeem. Om dit probleem op te lossen stelt dit proefschrift een systeem voor waar iedere gebruiker een specifieke code toegewezen krijgt. De code identificeert de gebruiker en markeert ook het begin van een datapakket in een observatie-interval.

In dit proefschrift bestuderen we twee gevallen:

- **Een smalbandig systeem** – Om storing in een niet-gecontroleerde omgeving te verminderen stellen we voor om een klein antenne-array bij de ontvanger te installeren. De gewenste zender kan dan van de overige signalen gescheiden worden door de antennes te combineren (bundelvorming of nulsturing). Om de gewenste gebruiker te onderscheiden van de overige signalen geven we zijn constante-amplitude signaal een kleine, bekende, amplitudevariatie ("kleurcode"). De multiplicatieve code is een vorm van training, zonder dat de benodigde datasnelheid groter wordt. Deze code maakt het mogelijk en makkelijk om signalen van meerdere gebruikers te scheiden en hierop te synchroniseren. We stellen een blind synchronisatieschema voor dat compatibel is met bestaande standaarden zoals Bluetooth en WLAN. "Blind" betekent hier dat de verstuurde datareeks onbekend is.
- **Een ultra wideband (UWB) systeem** – Dit is een draadloos communicatiesysteem dat zonder draaggolf werkt, de informatie wordt verstuurd door middel van korte pulsen (korter dan een ns). Hierdoor hebben de pulsen een extreem brede bandbreedte, waardoor het mogelijk is om hoge transmissiesnelheden (orde 100 Mb/s) te halen over korte afstanden (orde 10 m). Het systeem dat in het proefschrift bestudeerd wordt is een "transmit-reference impuls" systeem, waarbij data verstuurd wordt door middel van puls-paren. De energie van een rijk multipad kanaal kan verzameld worden door correlatie en integratie aan de ont-

vangstzijde. We beschouwen twee soorten codes om meerdere gebruikers te onderscheiden – *time hopping*, waarbij de afstand tussen de twee pulsen wordt gevarieerd, en *polarity code*, vergelijkbaar met de codes in CDMA.
In het tweede deel van het proefschrift leiden we een algoritme af voor gecombineerde blinde synchronisatie en data-detectie voor transmit-reference UWB (TR-UWB) systemen. Het algoritme maakt hoge transmissiesnelheden mogelijk tussen gebruikers in een ad hoc UWB netwerk.

Acknowledgements

Above all, I would like to express my gratitudes to prof. Alle-Jan van der Veen for providing me the opportunity to carry out PhD research at Circuits and Systems (CAS) Group at Delft University of Technology. I wish to thank him for everyday guidance and encouragement that have lead to the accomplishment of this thesis.

Thanks to the committee members, J. T. Fokkema(Rector Magnificus), P. M. Dewilde, P. van Genderen, J. P. Linnartz, J. W. M. Bergmans, W. van Etten, M. Ghogho for reviewing the thesis and providing me with a number of meaningful remarks.

I highly appreciated numerous fruitfull discussions with Geert Leus that enhanced the quality of this thesis. Special thanks to Laura Zondervan for all the letters written in "Algemeen Beschaafd Nederlands". Thanks to Zoubir Irahhauten for providing me with measured UWB channels. Thanks to Antonio for being a good officemate, Giusseppe for teaching me brilliant Italian expressions, Filip for Linux hints, Eelco, Edin, Hieu, Tang, Bas, Laura Bruns and all the members of CAS group, Ubicom and Airlink project that made TU Delft a decent place to work.
Thanks to all the colleagues from TNO-ICT for helping me finishing the thesis.
Special thanks to Donna Laura, Don Michelle and Don Cicio for teaching me Sicilian.

Thanks to Aca, Adam, Aleksandra, Ana, Andrea, Arnaud, Beti, Biki, Bojan, Bojanica, Bas, Bram, Cecile, Čeda, Ćela, Ćići, Danica, Djapić–Vladimir and Nenad, Dragana, Duca, Dušan, Erik, Federiko, Friedrich, Gian-Luca, Goca, Goga, Gile, Hernan, Ida, Ivan, Ivana, Jan, Jasmina, Jeroen, Joos, Klaus, Livija, Ljuba, Maarten, Maja, Marco, Marijana,Marion, Marjan, Mark, Mika, Mike, Miki, Milena, Miške, Nada, Nebojša, Nenad, Nicolas, Nikos, Ozren, Paja, Pyrrhos, Peter-Paul, Rob, Satish, Silva, Sloba, Speedy, Stanči, Saša, Tanja, Tanjica, Teun, Tijana, Trifa, Vesna, Vitke, Vlaja, Wojtek, Yannis, Zak, Zloki, Zoki, Zoubir, Žika.
I apologize to whoever, unintentionally, I forgot to mention.

Figure 12.6: Creation of the painting used at the cover page of this thesis – Woezelhuis creche, 'Muizen' group, summer 2006, Delft, The Netherlands.

I highly appreciate the effort that Leoni, Nikola, Levy, Nickey, Laura, Faith and Mink made to create a colorful painting at the cover of this thesis.
Thanks to Martien van Gaalen and the members of TNO-ICT 'tekenkamer' for designing the cover page.

Thanks to my parents and sister.
Thanks to my mother in law.

Thanks to Dragana, Nikola and Katarina for support and inspiration.

Curriculum Vitae

Relja Djapic was born on February 12, 1975, in Novi Sad, Serbia. In June 1994, he graduated from the electrotechnical secondary school ”Mihaijlo Pupin”, Novi Sad with highest honours. He began his studies at Faculty of Technical Sciences, University of Novi Sad in September 1994. In July 2000 he obtained the Graduate Electrical Engineer degree with major in electronics and telecommunications. For his final thesis project he accomplished a permutation speech scrambler using the digital signal processing unit TMS DSP 320C50.
In 1997, he carried out a three month internship in Ecole National d'Ingenieurs de Tunis (ENIT), Tunisia where he investigated the properties of thin film gas sensors together with measurement and numerical methods used in gas detection.

In 2001 he started his research towards a Ph.D. degree at Delft University of Technology, The Netherlands in the Circuits and Systems Group under the supervision of professor Alle-Jan van der Veen. His research interests include signal processing for communication systems, blind source separation and synchronization schemes in wireless ad-hoc networks and ultra wideband systems. He obtained the best poster award at the Third International Symposium on Mobile Multimedia Systems and Applications (MMSA 2002), Delft, The Netherlands.

Since 2006 Relja is employed as a scientist in the Broadband Access and Home Networks department of TNO Information and Communication Technology, Delft, The Netherlands. Currently, his work focuses on the wireline broadband communication techniques over copper (xDSL) and coax.

References

[1] G. Staple and K. Werbach, “The end of spectrum scarcity[spectrum allocation and utilization],” *IEEE Spectrum*, vol. 41, pp. 48–52, mar 2004.

[2] R. van Nee and R. Prasad, *OFDM for wireless multimedia communications*. Artech House Publishers, 1st ed., 2000.

[3] K. Witrisal, *OFDM air-interface design for multimedia communications*. 2002.

[4] N. Abramson, “The ALOHA system–another alternative for computer communications,” in *Proc. AFIPS Conf.*, vol. 37, (Montvale, N.J.), pp. 281–285, 1970.

[5] L. Kleinrock and F. Tobagi, “Packet switching in radio channels: Part I– carrier sense multiple-access modes and their throughput-delay characteristics,” *IEEE Transactions on Communications*, vol. 23, pp. 1400–1416, dec 1975.

[6] B. Arons, “A review of the cocktail party effect,” *Journal of the American Voice I/O Society*, pp. 35–50, July 1992.

[7] A.Paulraj, R.Roy, and T.Kailath, “Estimation of signal parameters via rotational invariance techniques—ESPRIT,” in *Asilomar Conference on Signals, Systems, and Computers*, (Pacific Grove, CA), pp. 83–89, July 1985.

[8] R.Roy and T.Kailath, “ESPRIT—estimation of signal parameters via rotational invariance techniques,” *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 37, pp. 984–995, July 1989.

[9] A. J. van der Veen and A. Paulraj, “An analytical constant modulus algorithm,” *IEEE Trans.Signal Processing*, vol. 44, pp. 1136–1155, May 1996.

[10] A. J. van der Veen, “Signal processing for communications,” *Lecture notes, TUD elft*, 1998.

[11] H. Krim and M. Viberg, “Two decades of array signal processing research,” *IEEE Signal Processing Magazine*, pp. 67–94, jul 1996.

[12] B. V. Veen and K. Buckley, “Beamforming: A versatile approach to spatial filtering,” *IEEE ASSP Magazine*, pp. 4–24, apr 1988.

[13] G. Golub, *Matrix Computations*. The John Hopkings University Press, 2nd ed., 1989.

[14] T. Moon and W.C.Stirling, *Mathematical Methods and Algorithms for Signal Processing*. Prentice Hall, 2000.

[15] G. Strang, *Linear Algebra and its Applications*. Thomson Learning, 3rd ed., 1988.

[16] A. J. van der Veen, “Algebraic methods for deterministic blind beamforming,” *Proceedings of the IEEE*, vol. 86, pp. 1987–2008, Oct. 1998.

[17] P.Stoica and A.Nehorai, “MUSIC, maximum likelihood and cramer-rao bound,” *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 37, pp. 720–741, May 1989.

[18] N. Abramson, “Packet switching with satellites,” in *in NAtional Computer Conference, AFIPS Conf. Proc.*, vol. 42, pp. 695–702, 1973.

[19] V. Bharghavan, “Performance evaluation of algorithms for wireless medium access,” in *IEEE International Computer Performance and Dependability Symposium*, pp. 86–95, Sept. 1998.

[20] M. Joa-Ng and I.-T. Lu, “Spread spectrum medium access protocol with collision avoidance in mobile ad-hoc wireless network,” in *INFOCOM ’99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, pp. 776 – 783, Mar. 1999.

[21] P. Karn, “MACA—a new channel access method for packet radio,” in *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pp. 134–140, ARRL, 1990.

[22] L. Kleinrock and F. Tobagi, “Packet switching in radio channels: Part II—the hidden terminal problem in carrier sense multiple access and the busy-tone solution,” *IEEE Tr. Communications*, vol. 23, pp. 1417–1433, Dec. 1975.

[23] J. Treichler and B. Agee, “A new approach to multipath correction of constant modulus signals,” *IEEE Transactions Acoustics, Speech, Signal Processing*, vol. 31, pp. 459–471, apr 1983.

[24] J. Treichler and B. Agee, “Convergence behaviour of the constant modulus algorithm,” in *IEEE ICCASP*, vol. 1, pp. 13–16, 1983.

[25] C. Papadias and A. Paulraj, “A constant modulus algorithm for multiuser signal separation in presence of delay spread using antenna arrays,” *IEEE Signal Processing Letters*, vol. 4, pp. 178–181, June 1997.

[26] A. J. van der Veen and A. Paulraj, “An analytical constant modulus algorithm,” *IEEE Transactions on Signal Processing*, vol. 44, pp. 1–19, may 1996.

[27] A. van der Veen, “Analytical method for blind binary signal separation,” *IEEE Trans. Signal Proc.*, vol. 45, pp. 1078–1082, Apr. 1997.

[28] A. J. van der Veen, “Algebraic constant modulus algorithms,” *in Signal Processing Advances in Wireless and Mobile Communications, G. Giannakis e.a., editors*, 2000.

[29] J. Treichler and M. Larimore, “New processing techniques based on the constant modulus adaptive algorithm,” *IEEE Transactions Acoustics, Speech, Signal Processing*, vol. 33, pp. 420–431, apr 1985.

[30] A. Orozco-Lugo and D. McLernon, “Blind signal separation for SDMA based on periodically time varying modulation,” in *IEE National conference on antennas and propagation*, (York, U.K.), pp. 182–186, Apr. 1999.

[31] A. Orozco-Lugo and D. McLernon, “Blind ISI and MAI cancellation based on periodically time varying transmitted power,” *Electronics letters*, vol. 37, July 2001.

[32] J. Tugnait and W.Luo, “On channel estimation using superimposed training and first-order statistics,” in *Proc. IEEE ICASSP*, (Hong Kong), 2003.

[33] X. Meng and J. Tugnait, “Semi-blind channel estimation and detection using superimposed training,” in *Proc. IEEE ICASSP*, (Montreal, Canada), 2004.

[34] J. Tugnait and X. Meng, “Synchronization of superimposed training for channel estimation,” in *Proc. IEEE ICASSP*, (Montreal, Canada), 2004.

[35] E.Alameda-Hernandez, D. McLernon, A. Orozco-Lugo, M.Lara, and M. Ghogho, “Synchronization for superimposed training based channel estimation,” *Electronic letters*, vol. 41, Apr. 2005.

[36] M. Tsatsanis and G. Giannakis, “Transmitter induced cyclostationarity for blind channel equalization,” *IEEE Trans.Signal Processing*, vol. 45, pp. 1785–1794, July 1997.

[37] E. Serpedin and G. Giannakis, “Blind channel identification and equalization with modulation-induced cyclostationarity,” *IEEE Tr. Signal Proc.*, vol. 46, pp. 1930–1944, July 1998.

[38] A. Chevreuil, P. Loubaton, and L. Vandendorpe, “Performance of general transmitter induced cyclostationarity precoders: analysis based on a mmse-df receiver,” *IEEE Tr. Signal Proc.*, vol. 48, pp. 3072–3086, Nov. 2000.

[39] A. Chevreuil and P. Loubaton, “MIMO blind second-order identification method and conjugate cyclostationarity,” *IEEE Trans.Signal Processing*, vol. 47, pp. 572–578, feb 1999.

[40] R. H. H. Bölcskei and A. Paulraj, "Blind channel estimation in spatial multiplexing systems using nonredundant antenna precoding," in *Proc. IEEE 33rd Asilomar conference on SSC*, vol. 2, pp. 1127–1132, 1999.

[41] A. Orozco-Lugo, M. Lara, D. McLernon, and H. Muro-Lemus, "Multiple packet reception in wireless ad hoc networks using polynomial phase modulating sequences," *IEEE Trans. Signal Processing*, vol. 51, pp. 2093–2110, Aug. 2003.

[42] A. Orozco-Lugo and D. McLernon, "Blind channel equalization using chirp modulating signals," *IEEE Trans. Signal Processing*, vol. 52, pp. 1364–1375, May 2004.

[43] A. J. van der Veen and L. Tong, "Packet separation in wireless ad-hoc networks by known modulus algorithms," in *Proc. IEEE ICASSP*, (Orlando, FL, USA), May 2002.

[44] R. Djapic and A. J. van der Veen, "Blind synchronization in asynchronous multiuser packet networks using KMA," in *IEEE workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, (Rome (Italy)), June 2003.

[45] R. Djapic, A. J. van der Veen, and L. Tong, "Synchronization and packet separation in wireless ad hoc networks by known modulus algorithms," *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 51–64, 2005.

[46] R. Djapic, G. Leus, and A. J. van der Veen, "Cramer-Rao bounds for blind and training based packet offset estimation in wireless ad hoc networks," in *In 11th IEEE Symposium on Communications and Vehicular Technology (SCVT)*, (Gent(Belgium)), Nov. 2004.

[47] R. Djapic and A. J. van der Veen, "Packet separation using the known modulus algorithm– experimental results," in *In 3rd International Symposium on Mobile Multimedia Systems and Applications (MMSA'02)*, (Delft(The Netherlands)), Dec. 2002.

[48] J. Treichler and M. Larimore, "New processing techniques based on constant modulus adaptive algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 33, pp. 420–431, Apr. 1985.

[49] B. Agee, "The least-squares CMA: A new technique for rapid correction of constant modulus signals," in *Proc. IEEE ICASSP*, (Tokyo), pp. 953–956, 1986.

[50] A. J. van der Veen, "Asymptotic properties of the algebraic constant modulus algorithm," *IEEE Trans. Signal Processing*, vol. 49, pp. 1796–1807, Aug. 2001.

[51] A. J. van der Veen, M. Vanderveen, and A. Paulraj, "SI-JADE: An algorithm for joint angle and delay estimation using shift-invariance properties," *IEEE Signal Processing Letters*, vol. 4, pp. 142–145, May 1997.

[52] A. J. van der Veen, S. Talwar, and A. Paulraj, "A subspace approach to blind space-time signal processing for wireless communication systems," *IEEE Trans.Signal Processing*, vol. 45, pp. 173–190, Jan. 1997.

[53] P.Laurent, "Exact and approximate construction of digital phase modulations by superposition of amplitude modulated pulses (amp)," *IEEE Transactions on Communications*, vol. 5, pp. 150–160, feb 1986.

[54] P. Stoica and A. Nehorai, "MUSIC, Maximum Likelihood and Cramér-Rao bound," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. 37, pp. 720–741, May 1989.

[55] A. Leshem and A. van der Veen, "Direction of arrival estimation for constant modulus signals," *IEEE Tr. Signal Proc.*, vol. 47, Nov. 1999.

[56] P. Stoica and A. Nehorai, "MUSIC, Maximum Likelihood, and Cramér-Rao bound: Further results and comparisons," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. 38, pp. 2140–2150, Dec. 1990.

[57] P. Stoica and B. C. Ng, "On the Cramer-Rao bound under parametric constraints," *IEEE Signal Processing Letters*, vol. 5, pp. 177–179, July 1998.

[58] A. Batra, J. Balakrishnan, R. Aiello, J. R. Forester, and A. Dabak, "Design of a multiband ofdm system for realistic uwb channel environments," *Transactions on microwave theory and techniques*, vol. 52, sep 2004.

[59] V.S.Somayazulu, J. R. Forester, and S. Roy, "Design challenges for very high data rate uwb systems," in *Asilomar Conference on Signals, Systems, and Computers*, nov 2002.

[60] L. Yang and G. Giannakis, "Ultra-wideband communications: an idea whose time has come," *Signal Processing Magazine, IEEE*, vol. 21, pp. 26–54, Nov. 2004.

[61] R. A. Scholtz, D. M. Pozar, and W. Namgoong, "Ultra-wideband radio," *EURASIP Journal on Applied Signal Processing*, no. 3, pp. 252–272, 2005.

[62] C. Carbonelli, U. Mengali, and U. Mitra, "A PPM gaussian monocycle tansmitter for ultra-wideband communications," in *IEEE Conference on Ultra Wideband Systems and Technologies*, (Japan), pp. 130–134, May 2004.

[63] D. Cassioli, M. Win, and A. Molisch, "The ultra-wide bandwidth indoor channel: From statistical model to simulations," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 1247–1257, Aug. 2002.

[64] M. Win and R. Scholtz, "On the robustness of ultra-wide bandwidth signals in dense multipath environments," *IEEE Communications Letters*, Feb. 1998.

[65] M. Win and R. Scholtz, “Ultra-wide bandwidth time-hopping spread spectrum impulse radio for wireless multiple-access communications,” *IEEE Transactions on Signal Processing*, vol. 48, pp. 679–691, Apr. 2000.

[66] Z. Irahhauten, G. J. Janssen, H. Nikookar, A. Yarovoy, and L. P. Ligthart, “UWB channel measurements and results for office and industrial environments,” in *submitted to International Conference on UWB (ICUWB)*, (Boston,USA), 2006.

[67] Z. Irahhauten, G. J. Janssen, H. Nikookar, A. Yarovoy, and L. P. Ligthart, “Statistical analysis of ultra-wideband channel measurements in indoor office environments,” in *submitted to International Conference on UWB (ICUWB)*, (Boston,USA), 2006.

[68] M. Win and R. Scholtz, “Impulse radio: How it works,” *IEEE Communications Letters*, Feb. 1998.

[69] N. He and C. Tepedelenlioglu, “Fast and low-complexity synchronization for non-coherent UWB receivers,” *Submitted to Transactions on Communications*, 2006.

[70] E. Homier and R. Scholtz, “Rapid acquisition of ultra-wideband signals in the dense multipath channel,” in *IEEE Conference on Ultra Wideband Systems and Technologies*, vol. 2, (Orlando, FL), pp. 105–109, 2002.

[71] R. Blazquez, P.Newaskar, and A.Chandrakasan, “Coarse acquisition for ultra wideband digital receivers,” in *International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, 2003.

[72] Z. Tian and V. Lottici, “Efficient timing acquisition in dense multipath for uwb communications,” in *IEEE Vehicular Technology Conference, VTC*, vol. 2, (Orlando, FL), pp. 1318–1322, oct 2003.

[73] X.Luo and G.B.Giannakis, “Blind timing acquisition for ultra-wideband multiuser ad hoc access,” in *International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, 2005.

[74] R. Gagliardi, “A geometrical study of transmitted reference communication systems,” *IEEE Transactions on Communication Technology*, pp. 118–123, Dec. 1964.

[75] R. Hoctor and H. Tomlinson, “Delay-hopped transmitted-reference RF communications,” in *IEEE Conference on Ultra Wideband Systems and Technologies*, pp. 265–270, 2002.

[76] N. van Stralen, A. Dentinger, K. Welles, R. Gauss, R. Hoctor, and H. Tomlinson, “Dealy hopped transmitted reference experimental results,” in *IEEE Conference on Ultra Wideband Systems and Technologies*, pp. 93–98, 2002.

[77] G. Leus and A. J. van der Veen, "Noise suppression in UWB transmitted reference system," in *IEEE Signal Processing Workshop on Signal Processing Advances in Wireless Communications, SPAWC*, (Lisboa, Portugal), July 2004.

[78] A. Trindade, Q. H. Dang, and A. J. van der Veen, "Signal processing model for a transmit-reference UWB wireless communication system," in *IEEE Conference on Ultra Wideband Systems and Technologies*, (Reston, Virginia), Oct. 2003.

[79] R. Djapic, A. J. van der Veen, and G. Leus, "Blind synchronization in asynchronous UWB networks based on the transmit-reference scheme," in *Asilomar Conference on Signals, Systems, and Computers*, Nov. 2004.

[80] A. J. van der Veen, A. Trindade, Q. Dang, and G. Leus, "Signal model and receiver algorithms for a transmit-reference ultra-wideband communication system," *JSAC(Special issue on ultra wideband)*, vol. 24, pp. 773–779, Apr. 2006.

[81] Q. Dang, A. J. van der Veen, and A. Trindade, "Resolving inter-frame interference in a transmit-reference ultra-wideband communication system," in *International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, (Toulouse), May 2006.

[82] A. Trindade, Q. Dang, and A. J. van der Veen, "Signal processing model for a transmit-reference UWB wireless communication system," in *IEEE Conference on Ultra Wideband Systems and Technologies*, (Reston, Virginia), Oct. 2003.

[83] M. Torlak and G. Xu, "Blind multiuser channel estimation in asynchronous CDMA systems," *IEEE Transactions on Signal Processing*, vol. 45, pp. 137–147, Jan. 1997.