Vorlesung:
Collaborative transmission in wireless sensor networks

Wintersemester 2010/2011

Version: October 25, 2010 ($v0.0.3 + \varepsilon$)

Veranstalter: Stephan Sigg

Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund
Verteilte und Ubiquitäre Systeme

D-38106 Braunschweig

Abstract

In wireless communication, beamforming techniques are utilised to constructively superimpose carrier-signal-components from distributed transmit antennas at a receive antenna. In centralised beamforming applications, the transmit antennas are tightly synchronised and the signal flow transmitted over these antennas is controlled by a single entity.

In distributed beamforming, however, we assume that transmit antennas are distributed with various mobile devices. This scenario of spatially distributed, wirelessly interconnected and probably mobile devices is prominent in many ubiquitous and wireless network settings. With distributed adaptive beamforming, the transmission range of such a system can be increased or the robustness of a signal can be improved. Alternatively, it becomes feasible to reduce the transmission power and energy consumption of individual nodes and therefore to improve the network lifetime.

Each of the distributed transmit devices has its own local oscillator. For beamforming, an exactly synchronised phase and frequency of the transmit signal is required for all devices that participate in the transmission. As the oscillation frequency is not synchronised among nodes, the synchronisation of phase and frequency constitutes the main challenge for distributed beamforming.

Several approaches for carrier-synchronisation of distributed beamforming devices are proposed in the literature. Generally, we distinguish between open-loop carrier- synchronisation techniques and closed-loop carrier-synchronisation. In open-loop approaches, the node that receives the transmission controls the synchronisation of transmit nodes by broadcasting a training signal or by estimating the relative phase-offset of all individual nodes.

In closed-loop carrier-synchronisation, transmit and receive nodes achieve carrier- synchronisation by communication in both directions.

From the methods discussed in the literature, the computationally cheapest and therefore best suited approach for resource restricted nodes is an iterative feedback-based closed-loop carrier-synchronisation approach. In the scope of this lecture we analyse this approach and provide algorithmic improvements.

We describe this method analytically and observe that the approach generally implements a randomised search mechanism. In particular, the search mechanism can be modelled as an evolutionary algorithm. With this knowledge, we derive asymptotic upper and lower bounds on the expected synchronisation-time that are in the same order.

Based on this result we propose several modifications on the basic synchronisation approach to further improve the synchronisation-time and derive upper and lower bounds on the asymptotic synchronisation-time of a local random search approach.

Finally, an asymptotically optimal carrier phase-synchronisation algorithm is proposed and evaluated.

After carrier-synchronisation, data is transmitted collaboratively by the remote nodes. The quality of such a transmission is dependent on the synchronisation quality among carrier-signals of nodes.

We study the impact of various environmental impacts as noise, transmission-distance and network size on the carrier-synchronisation quality. It is derived that the synchronisation mechanism can be optimised for given environmental impacts.

We propose an adaptive protocol for transmission by distributed adaptive beamforming and study several learning approaches to adapt the synchronisation method for distinct environments. In particular, a binary search method and a learning classifier system are considered.

Overall, this lecture provides a comprehensive discussion of distributed adaptive beamforming by feedback-based carrier-synchronisation approaches. Starting with a concise discussion of synchronisation algorithms and their asymptotic bounds, we analyse environmental effects and finally propose transmission protocols for distributed adaptive transmit beamforming that are able to adapt to changing environments.

Acknowledgements

Several people have contributed to this lecture in different ways.

During the last three years I worked together closely with students completing practical training courses and theses on similar topics. The interesting and deep discussions have often led to refinements or clarifications of this work. In particular, the insightful discussions with Jialin Wang and Weiwei Liang have helped to shape the scope of this lecture in its early stages.

Section 7.3.3 was predominantly influenced by the work of Rayan Merched El Masri. Rayan brought in some new ideas on the optimisation process and did a great job questioning existing assumptions which helped to find the relevant optimisation parameters. The experimental results in section 7.4.2 have been deduced by Julian Ristau. Julian has been the first to implement the iterative synchronisation scheme for the USRP software radios and has had a great impact on the implementation.

Several students in practical training courses in the summer terms of 2009 and 2010 have tweaked the possibilities of the software radios and helped with the implementation. In particular, Aaron Israel, Georg von Zengen, Gerrit Bagschik and Toni Günther have done a great job in preparing very useful tools and demonstrating the capabilities of the USRPs. Also, Johannes Starosta, Sebastian Schwarzl and Markus Reschke coupled the USRP with the classification capabilities of the Orange framework nicely.

I would like to thank the students of my lecture 'Collaborative transmission in wireless sensor networks' at the Technische Universität Braunschweig in the winter term 2009/2010 and in the summer term 2010 for their patience, insightful questions and constructive feedback and discussions on the topic. Timo Schulz and Sascha Lity, in particular, have spotted many spelling errors in prior versions of this document.

Contents

Abbreviations and Notation

The following notations are utilised throughout this document. The attempt has been made to keep the standard notation from the literature whenever possible. However, since diverse scientific areas are covered, the notation had to be adapted in order to provide an unambiguous notation. The page number given in the table refers to the first occurrence of the mentioned term.

Notation	Explanation	Page
A	Region where nodes of a sensor network are placed	32
α	Path-loss exponent	46
B	Bandwidth	50
CDMA	Code division multiple access	16
c	Speed of light $(3 \cdot 10^8 \frac{m}{s})$	43
d	Distance	45
$E[x]$	The expectation of a random variable x	62
η	Density of a sensor network	32
f	Frequency	43
$\mathcal{F}$	Fitness-function	79
G_{RX}	Gain of the receive antenna	45
G_{TX}	Gain of the transmit antenna	45
γ	Phase offset of a transmit signal	43
$\Im(s)$	Imaginary part of a complex signal s	
IAC	inquiry access codes	52
ISM	Industrial, Scientific, Medical band	52
i.i.d.	Identically and independently distributed	16
J	Joule	50
K	Kalvin	50
κ	Boltzmann constant	50
λ	Wavelength of a transmit signal	43
MIMO	Multiple input multiple output	15
MISO	Multiple input single output	55

Notation	Explanation	Page
μ	Population size of an evolutionary algorithm	
$\mu(R)$	Density of a sensor network with transmission range R	
N	Network size	32
ν	Offspring population size of an evolutionary algorithm. Note: In the literature, typically λ denotes the offspring population size	68
P_N	Thermal noise-power	50
P_{RX}	Received signal power	45
P_{TX}	Transmission power	43
$P(x)$	Probability of an event x	60
$P(\chi_1\|\chi_2)$	Conditional probability of 2 events χ_1, χ_2 with $P(\chi_2) > 0$	62
$\mathcal{P}$	An optimisation problem	79
Π	Sample space	60
$\Re(s)$	Real part of a complex signal s	44
$R_k(t)$	The reliability or fault tolerance of a sensor node	32
R	Transmission range of a sensor network	32
RF-	Radio frequency	16
RMSE	Root of the Mean Squared Error	104
RSS	Received Signal Strength	45
S	A search-space	67
SIMO	Single input multiple output	55
SINR	Signal to interference and noise ratio	51
SISO	Single input single output	56
SNR	Signal to noise ratio	16
s^*	Complex conjugate of s	
T	Temperature in Kalvin	50
υ	Failure rate	32
$var[\chi]$	The variance of a random variable χ: $E[(\chi - E[\chi])^2]$	63
$\overrightarrow{v}$	A vector $v = (v_1, \ldots, v_k)$	
WSN	Wireless sensor network	16
x	A sample-point for a random experiment	58

1 Introduction

> *In the long history of humankind (and animal kind, too) those who learned to collaborate and improvise most effectively have prevailed*
>
> (C. Darwin)

In recent years, sensor nodes of an extremely tiny size have been envisioned [1, 2, 3]. In [4], for example, applications for square-millimetre sized nodes that integrate seamlessly into an environment are detailed. Because of these small form-factors the transmission power of wireless nodes is restricted to several microwatts. Communication between a single node and a remote receiver is therefore only feasible at short distances. Nonetheless it is possible to increase the maximum transmission range by cooperatively transmitting information from distinct nodes of a network [5, 6]. The basic idea is to superimpose identical RF-carrier-signal-components from various transmitters that function as a distributed beamformer. When the relative phase-offset of these carrier-signal-components at a remote receiver is small, the signal strength of the received sum-signal is improved. Cooperation can improve the capacity and robustness of a network of transmitters [7, 8] and decreases the average energy consumption per node [9, 10, 11].

Related branches of research are cooperative transmission [12], collaborative transmission [13, 14], distributed adaptive beamforming [15, 16, 17, 18], collaborative beamforming [19] or cooperative/virtual MIMO for wireless sensor networks [20, 21, 22, 23]. One approach is utilising neighbouring nodes as relays [24, 25, 26] as proposed by Cover and El Gamal in [27]. Cooperative transmission is then achieved by multi-hop [28, 29, 30] or data flooding [31, 32, 33, 34] approaches. The general idea of multi-hop relaying based on the physical channel is to retransmit received messages by a relay node so that the destination will receive not only the message from the source destination but also from the relay. In data flooding approaches a node retransmits a received message. It has been shown that the approach outperforms non-cooperative multi-hop schemes significantly. It was derived that the average energy consumption of nodes is decreased [9, 10] and the transmission time is reduced compared to traditional transmission protocols in wireless sensor networks [35].

In these approaches, nodes are not synchronised tightly and transmission may be asynchronous. This, however, is achieved by virtual MIMO techniques. In virtual MIMO for wireless sensor networks, single antenna nodes are cooperating to establish a multiple antenna wireless sensor network [21, 20, 22]. Virtual MIMO is capable of adjusting to different frequencies and is highly energy-efficient [23, 11]. However, the implementation of MIMO capabilities in WSNs requires accurate time synchronisation, complex transceiver circuits and signal processing that might surpass the power consumption and processing capabilities of simple sensor nodes.

Other solutions proposed are open-loop synchronisation methods such as round-trip synchronisation-based [36, 37, 38]. In this transmission scheme, the destination sends beacons in opposed directions along a multi-hop circle in which each of the nodes appends its part of the overall message to the beacons. Beamforming is achieved when the processing time along the multi-hop chain is identical in both directions. This approach, however, does not scale with the size of a network.

Closed-loop feedback-based approaches include full-feedback techniques, in which carrier-synchronisation is achieved in a master-slave manner. The phase-offset between the destination and a source node is corrected by the receiver node. Diversity between RF-transmit signal-components is achieved over CDMA channels [39]. This approach is only applicable to small network sizes and requires sophisticated processing capabilities of the source nodes.

A more simple and less resource demanding implementation is the one-bit feedback-based closed-loop synchronisation considered in [39, 40]. The authors describe an iterative process in which n source nodes $i \in [1, \ldots, n]$ randomly adapt the phases γ_i of their carrier-signal $\Re\left(m(t)e^{j(2\pi(f_c+f_i)t+\gamma_i)}\right)$. Here, f_i denotes the frequency-offset of node i to a common carrier frequency f_c. Initially, i.i.d. phase-offsets γ_i of carrier-signals are assumed. When a receiver requests a transmission from the network, carrier phases are synchronised in an iterative process (cf. figure 1.1).

1. Each source node i adjusts its carrier phase-offset γ_i and frequency-offset f_i randomly.
2. The source nodes transmit simultaneously as a distributed beamformer.
3. The receiver estimates the level of phase-synchronisation of the received sum-signal (e.g. by the signal to noise ratio (SNR)).
4. This value is broadcast as a feedback to the network. Nodes interpret this feedback and adapt their phase adjustments accordingly.

These four steps are iterated repeatedly until a stop criteria is met (e.g. maximum iteration count or sufficient synchronisation). This process has been studied by us and other authors [41, 42, 43, 13] where the approaches proposed differ in the implementation of the first and the fourth step specified above. The authors of [43] show that it is possible to reduce the number of transmitting nodes in a random process and still achieve synchronisation among all nodes.

In [41, 42, 43] a process is described in which each node alters its carrier phase-offset γ_i according to a normal distribution with a small variance in step one. In [13] a uniform distribution is utilised, but the probability for one node to alter its phase-offset is low. Only in [42] not only the phase but also frequency is adapted.

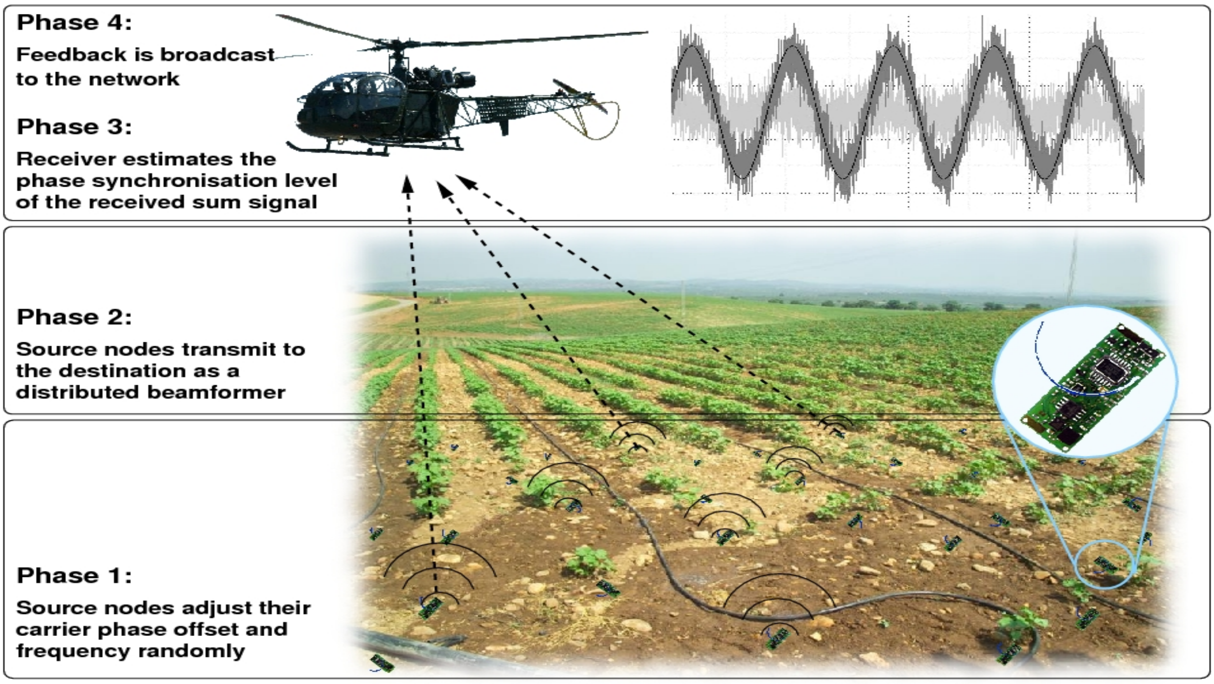

Phase 4:
Feedback is broadcast to the network
Phase 3:
Receiver estimates the phase synchronisation level of the received sum signal
Phase 2:
Source nodes transmit to the destination as a distributed beamformer
Phase 1:
Source nodes adjust their carrier phase offset and frequency randomly

Figure 1.1: Schematic illustration of feedback-based distributed adaptive beamforming in wireless sensor networks

This lecture focuses on cooperative transmission schemes in wireless sensor networks. Distinct nodes in a network of nodes cooperate in their transmission of data. As detailed above, this cooperation may differ in its exact implementation for various approaches. Some approaches require inter-node communication while others don't. For some approaches the aim is to reduce the failure probability through multiple transmissions while others aim to improve the signal strength.

We discuss these communication schemes and the theory required for the analysis. Then, a computationally cheap approach is analysed analytically and bound in its synchronisation-time. We present simulations and experimental studies for several algorithmic improvements .

Finally, an adaptive transmission protocol is proposed and the learning mechanism is analysed.

1.1 Application scenarios of carrier phase-synchronisation

A typical scenario for the application of distributed adaptive beamforming was depicted in figure 1.1. In this scenario the transmission range of distributed wireless devices to a remote receiver is extended by distributed adaptive transmit beamforming.

In a similar way, the technique can also be utilised to sustain connectivity between two parts of a disconnected wireless network.

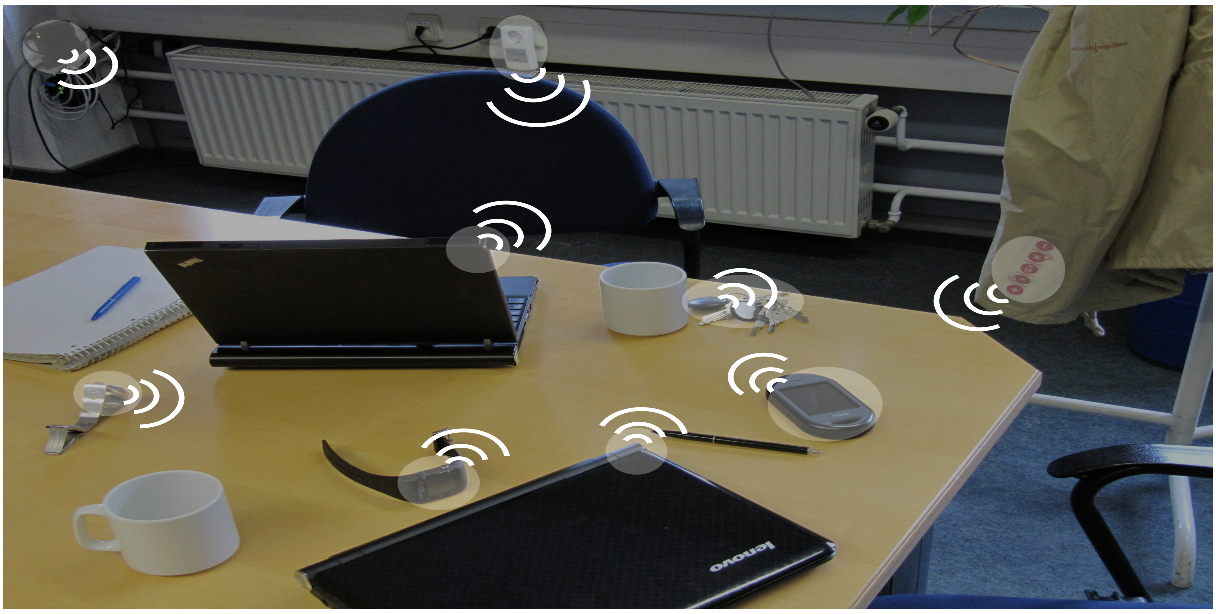


Figure 1.2: A meeting situation that is enriched with communication and sensing devices

Instead of increasing the range of wireless nodes, however, we could also decrease the power consumption of devices by reducing the transmission power. Low power nodes would then reach a remote target collaboratively, regardless of their restricted individual transmission range. We detail the benefits of this concept as follows.

We have seen the implementation of sensor-enriched and computing-power-enriched devices developed at research labs. Popular examples are the Media Cup developed at the TecO [44] or also the aware home project [45] of the Georgia Institute of Technology. These prototype applications have served as proof-of-concept implementation that show the immense potential of enriched Internet of Things (IoT) environments [46]. However, sensing, computation and, most importantly, communication require a vast amount of energy when incorporated by a high number of objects in an Internet of Things. When nearly every object is enriched by communication and sensing capabilities, a high share of devices that utilise energy harvesting techniques is required in order to keep the energy demands at a reasonable level.

This means that the energy consumption of individual devices has to be reduced dramatically to enable an internet of things. In particular, very low-power transmission schemes are required.

In an internet of things distributed adaptive transmit beamforming can be utilised to enable very low-power transmitters in communication and sensing-enriched devices. In figure 1.2 an exemplary meeting environment is depicted. In this setting computation is embedded in multiple objects. For instance, in order to communicate with remote devices such as an office worker's smartphone the neighbouring devices can collaborate in their signal transmission to reach the remote device despite their low individual transmission power.

Although the overall power required for this transmission of multiple distributed devices is higher than a single link transmission, the maximum (peak) transmission power required by a single device can be reduced in exchange for a slightly increased average power consumption. By this approach it might be feasible to lower the average energy consumption of devices to an amount provided from energy harvesting sources such as, for instance, light, temperature, pressure or movement.

1.2 Challenges

Challenges for distributed adaptive beamforming result from a problem that is central to many applications of wireless sensor nodes. The hardware available often is very cheap so that a low accuracy and a high probability of error can be expected. In particular, this also affects the local oscillator of nodes.

A transmit signal from an individual node is created via individual local oscillators in these nodes. These oscillators are not synchronised and differ in their phase and frequency. Therefore, a carrier-signal of identical frequency from distinct nodes is only possible when the local oscillators of these nodes are synchronised. A main challenge in distributed adaptive transmit beamforming is this synchronisation.

When an identical frequency is established among carrier-signal-components, the signals have to be synchronised for their phase-offset to achieve a collaborative transmission. At this point several approaches to distributed carrier-synchronisation can be applied. Which approach is applicable is dependent on the restricted computational power and resources of these devices. For mobile devices it is generally important to design a synchronisation approach that has low computational complexity and memory requirements. For the iterative synchronisation process that we identify as well-suited in mobile environments a concise problem description and an analytic model of the synchronisation algorithm is missing. In particular, the impact of environmental effects and sharp bounds on the synchronisation-time are not provided. This hinders the development of optimal synchronisation approaches and the educated reaction to environmental settings.

In networks of distributed devices the mobility of devices can often be expected. Devices can move out of the transmission range of others or sources of interference might appear or disappear. Therefore, another aspect is the carrier-synchronisation-time, especially when the mobility of nodes is regarded. Since the synchronisation of carrier-signal-components is always respecting a given spatial location, mobility of nodes decreases the synchronisation quality over time. More seriously even, when transmit nodes are mobile, the synchronisation may drift apart even faster.

Finally, the environment in which the nodes are located may change. Persons moving or a different arrangement of furniture (e.g. opened or closed doors) alter the channel characteristic of individual links and therefore impact the synchronisation quality.

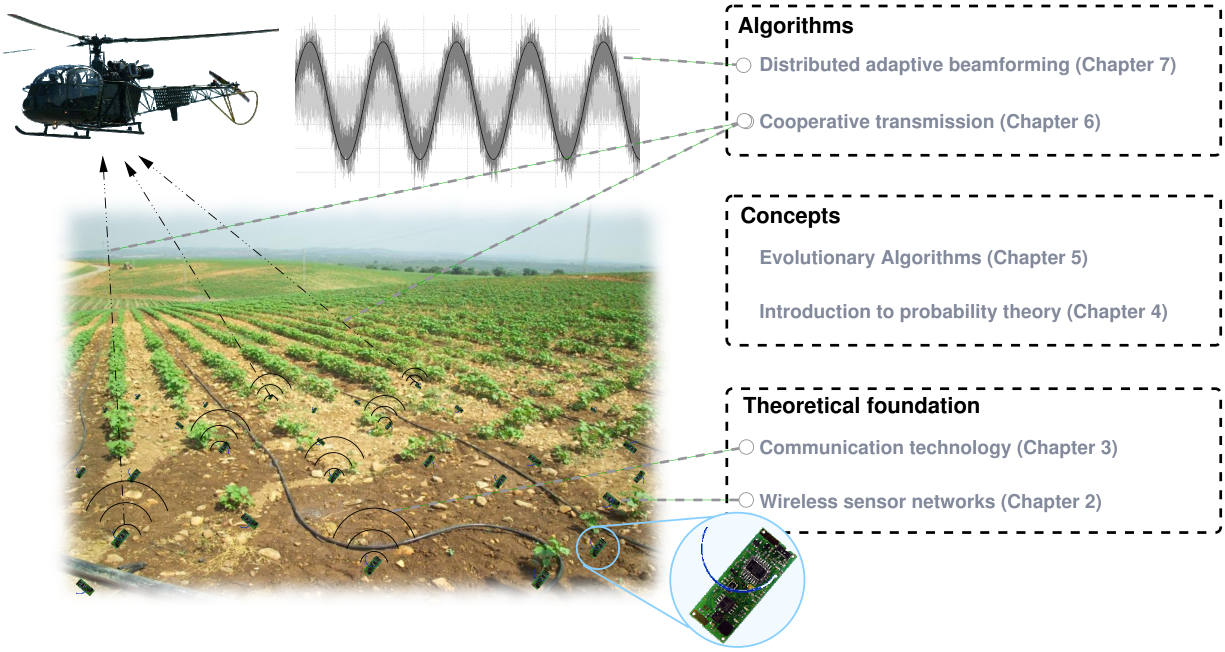

Algorithms
Distributed adaptive beamforming (Chapter 7)
Cooperative transmission (Chapter 6)
Concepts
Evolutionary Algorithms (Chapter 5)
Introduction to probability theory (Chapter 4)
Theoretical foundation
Communication technology (Chapter 3)
Wireless sensor networks (Chapter 2)

Figure 1.3: Possible scenario for distributed adaptive transmit beamforming

1.3 Outline of the lecture

This lecture is structured in three main blocks, namely 'theoretical foundation', 'concepts' and 'algorithms' as depicted in figure 1.3. From these blocks, the largest part and our main contribution is described in chapter 7.

Keeping in mind the theoretical foundation of this discussion, first the general notation of wireless sensor networks are introduced in section 2. After beginning with the general design and hardware of an individual sensor node (section 2.1), we continue with a discussion about the cooperation of nodes in networks in section 2.2 and present MAC protocols for wireless sensor networks in section 2.3.

As a second very basic foundation required for the discussion in this lecture aspects of wireless communication are detailed in chapter 3. After introducing very general effects such as superimposition of signals, pathloss, fading and the Doppler effect in section 3.1, we lay the foundation for distributed beamforming with a discussion of multiple antenna techniques and centralised beamforming in section 3.2 and section 3.3.

The concepts required for the analysis of the distributed adaptive beamforming scheme are discussed in chapter 4 and chapter 5. Chapter 4 closes with the discussion of the Markov and the Chernoff inequalities regarded as two important probability theoretic results for the estimation of expectation. These results are required to derive expected synchronisation-performances of the approaches. The following chapter 5 then introduces evolutionary algorithms. We will utilise this optimisation scheme to model the feedback-based distributed adaptive beamforming and thereby introduce basic principles, restrictions and design aspects of evolutionary algorithms in the sections 5.1, 5.2 and section 5.3. The chapter closes with the discussion of three general methods to determine asymptotic bounds on the optimisation-time of evolutionary algorithms in section 5.4. Specifically, we present the method of the fitness-based partitions to compute an upper bound, a general lower bound for evolutionary algorithms and the method of the expected progress.

In the following two chapters algorithmic solutions are investigated and are differentiated into cooperative and collaborative transmission techniques. Several cooperative techniques such as network coding, multi-hop and data flooding are introduced in section 6.1. Open-loop and closed-loop distributed carrier-synchronisation are introduced in section 6.2.

From the multi-antenna techniques discussed in section 6.2 a computationally cheap approach is introduced and discussed in chapter 7. In section 7.1 we analyse the problem-scenario analytically and derive a problem representation as an evolutionary algorithm. In section 7.2, the convergence time of this approach is analysed and upper and lower bounds on the expected synchronisation-performance are derived in section 7.2.1 and section 7.2.2. After the presentation of simulation results for this method in section 7.2.3 the impact of environmental settings and algorithmic configurations are discussed in section 7.2.4 and section 7.2.5. This discussion closes with the consideration of several algorithmic modifications to improve the synchronisation-performance (section 7.2.6).

In section 7.3 algorithmic improvements for distributed carrier-synchronisation are proposed. A hierarchical cluster scheme is investigated in section 7.3.1. Section 7.3.2 details a local random search method and presents asymptotic upper and lower bounds on the expected synchronisation-time of this algorithm. In Section 7.3.3 we present an asymptotically optimal solution for distributed carrier-synchronisation in wireless networks.

The performance of all approaches for distributed carrier-synchronisation is dependent on environmental impacts. Several environmental impacts are studied in section 7.4. In section 7.4.1 the velocity of transmit and receive nodes is studied. Section 7.4.2 covers multiple receiver nodes.

After synchronisation of carrier-signal-components, data is transmitted collaboratively by distributed nodes. A protocol for this transmission scheme is detailed in section 7.5 and the impact of changing environments on the performance of this protocol is taken into consideration. To account for these changes, adaptive protocols are proposed in section 7.6 and section 7.7.

1.4 Publications

Parts of the work undertaken for this lecture have already been submitted to journals, conferences or workshops. The related publications are as follows:

1. Stephan Sigg, Michael Beigl: An adaptive for distributed beamforming, in Proceedings of the MSN 2010 conference, December 2010 (to appear)

2. Stephan Sigg, Michael Beigl: A sharp asymptotic bound for feedback-based closed-loop distributed adaptive beamforming in wireless sensor networks (minor revision submitted to IEEE Transactions on Mobile Computing in July 2010)

3. Behnam Banitalebi, Stephan Sigg and Michael Beigl: Performance analysis of receive collaboration in TDMA-based Wireless Sensor Networks, in Proceedings of the fourth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (Ubicomm), October 2010 (to appear)

4. Stephan Sigg and Michael Beigl: Expectation aware in-network context processing, in Proceedings of the 4th ACM International Workshop on Context-Awareness for Self-Managing Systems in conjunction with Ubicomp 2010, September 2010 (to appear)

5. Behnam Banitalebi, Stephan Sigg and Michael Beigl: On the Feasibility of Receive Collaboration in Wireless Sensor Networks, in Proceedings of the 21st annual IEEE Symposium on Personal, Indoor and Mobile Radio Communications, September 2010 (to appear)

6. Rayan Merched El Masri, Stephan Sigg and Michael Beigl: An asymptotically optimal approach to distributed adaptive transmit beamforming in wireless sensor networks, Proceedings of the 16th European wireless conference, April 2010

7. Stephan Sigg, Rayan Merched El Masri, Julian Ristau and Michael Beigl: Limitations, performance and instrumentation of closed-loop feedback-based distributed adaptive transmit beamforming in WSNs, in Fifth International Conference on Intelligent Sensors, Sensor Networks and Information Processing - Symposium on Adaptive Sensing, Control, and Optimisation in Sensor Networks, December 2009

8. Stephan Sigg and Michael Beigl: Algorithmic approaches to distributed adaptive transmit beamforming, in Fifth International Conference on Intelligent Sensors, Sensor Networks and Information Processing - Symposium on Adaptive Sensing, Control, and Optimisation in Sensor Networks, December 2009

9. Stephan Sigg and Michael Beigl: Algorithms for closed-loop feedback-based distributed adaptive beamforming in wireless sensor networks, in Fifth International Conference on Intelligent Sensors, Sensor Networks and Information Processing - Symposium on Adaptive Sensing, Control, and Optimisation in Sensor Networks, December 2009

10. Stephan Sigg and Michael Beigl: Randomised Collaborative Transmission of Smart Objects, in 2nd International Workshop on Design and Integration Principles for Smart Objects (DIPSO2008) in conjunction with Ubicomp 2008, September 2008

11. Stephan Sigg and Michael Beigl: Collaborative Transmission in Wireless Sensor Networks by a (1+1)-EA, in Proceedings of the 8th International Workshop on Applications and Services in Wireless Networks (ASWN'08), 2008

2 Wireless sensor networks

Adding [...] sensors to everyday objects will create an Internet of Things, and lay the foundations of a new age of machine perception.

(K. Ashton, co-founder and executive director of the Massachusetts Institute of Technology's Auto-ID Center in a presentation in spring 1998)

The term wireless sensor network commonly depicts a network of distributed devices that consists of sensing and communication entities which collaborate to achieve a common objective. A wireless sensor network typically has one or more sinks to collect data from all sensing devices. These sinks might, for instance, constitute the interface of the sensor network to the outside world. Sensor networks can be applied to monitor environmental stimuli and provide feedback on a monitored area. In particular, sensor networks can be implemented in areas that are seldom or even hard to access. Examples are the monitoring of underwater scenarios, animal monitoring by sensor networks, the application in emergency situations such as an avalanche warning or the monitoring of earthquake regions. Further applications include smart sensors for the monitoring and control of manufacturing processes, bio-sensors for health monitoring, weather or habitat monitoring and smart environments for home entertainment [47]. Popular features of sensor networks are that these networks are easy to deploy, cheap, and require low maintenance effort in the optimum case. The following sections detail aspects of sensor networks that are related to cooperative and collaborative transmission strategies in these networks. These sections represent a comprised introduction to wireless sensor networks. An in-depth discussion of these topics can be found in [8, 48].

2.1 The sensor node

Sensor networks are constituted of sensor nodes that are deployed densely either very close to or directly inside a phenomenon to be observed. Therefore, sensor nodes are routinely required to work unattended in remote geographic areas.

These nodes typically consist of sensing, data processing, storage and communication components. Figure 2.1 depicts typical components of a sensor node schematically. Due to advances in integrated cirquit design the size, weight and cost of nodes is constantly decreasing while the accuracy and the resolution are improved [47].

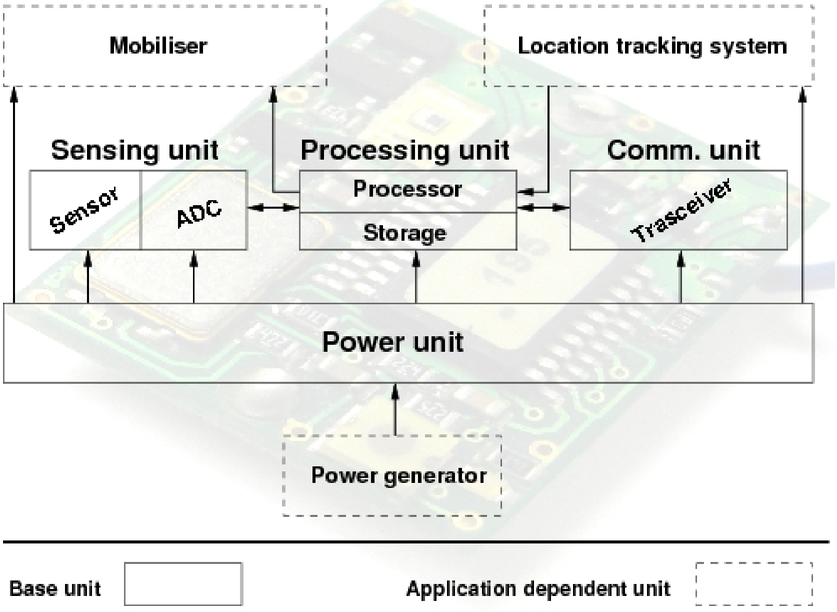


Figure 2.1: Components that typically constitute a sensor node [8]

A sensor node consists of some base units and application-dependent units as a mobiliser to accomplish some spatial movement of the node, a location tracking system such as GPS or relative positioning techniques (e.g. ultrasound) or a power generator (e.g. solar cells). The following section details the base units depicted in the figure.

2.1.1 Power unit

some of the most important design aspects for sensor networks are related to the power unit. As sensor networks are often designed to operate unattended over long time spans and operate disconnected from an external power source, the energy consumption and capacity of the power unit is a critical aspect.

Since the wireless sensor node is a microelectronic device, it can only be equipped with a very limited power source in the order of less than 0.5Ah, 1.2V (Another estimation from [48] is 2.2 to 2.5Ah at 1.5V). In a multi-hop sensor network, each node plays the dual role of a data originator and a data router. The malfunctioning of few nodes can cause significant topological changes and might require rerouting of packets and the reorganisation of the network. Therefore, power conservation and power management are of heightened importance.

Power consumption in a sensor network is divided into the categories sensing, communication and data processing. The most energy-consuming task among these is the communication (reception and transmission of data).

Power units can also be combined with by power harvesting units such as, for instance, solar cells.

2.1.2 Sensing unit

Sensing units are usually composed of two sub-units: sensors and analog-to-digital converters (ADCs). A sensor is a device that transforms physical parameters such as temperature, pressure or light intensity into analog electrical signals. The ADC transforms the analog signals produced by the sensors into digital signals that are then processed further in the processing unit and transmitted by the transceiver in the communication unit if required. A digital signal is a voltage signal that is either on or off but that does not allow for any intermediate value.

These digital signals can be used to control different components or parameters of the sensor nodes via actuators. An actuator is an electromechanical device such as a relay that can control other parts of an electromechanical system.

Typical sensors are, for example, audio, light or humidity sensors as well as ball switches or acceleration sensors.

2.1.3 Processing unit

The processing unit contains a processor and typically a small memory. It is responsible for pre-processing the information obtained by the sensing unit and for implementing the communication-related logic regarding the applications, the transport layer (e.g. TCP) and the network layer (e.g. routing).

Typically, a sensor node will not transmit each data item obtained from the sensing unit directly, but convert the raw data to a common representation or possibly discard redundant or highly error-prone data in order to save energy for communication. The processor decides when to transmit and which data via the communication unit.

The task of the processing unit can be summarised as enabling cooperation between nodes in a network. As a rule microcontrollers are utilised in the processing unit of sensor nodes due to their low-power consumption.

2.1.4 Communication unit

The communication unit consists of a transceiver to connect the node to the sensor network. A transceiver combines a transmitter and a receiver in a single entity. The communication unit might employ radio, infrared or optical technologies. Typical sensor nodes implement RF-technology. However, for low-lying antennas the RF-signal propagation is influenced by surface roughness, reflecting and obstructing objects and antenna elevation. The signal intensity may drop then as the fourth power of distance due to partial cancellation by a ground-reflected ray [49, 50]. Table 2.1 details some path-loss measurements obtained in various environments that were observed in [50].

For this reason it is more energy-efficient to implement a multi-hop topology with short hops instead of utilising high-power transmission nodes to cover greater distances. Alternative cooperative and collaborative communication paradigms that can be utilised when the transmission range of one individual node is not sufficient to reach a distant receiver. This

Location	Mean Path-loss exponent	Shadowing variance σ^2 (dB)
Apartment Hallway	2.0	8.0
Parking structure	3.0	7.9
One-sided corridor	1.9	8.0
One-sided patio	3.2	3.7
Concrete Canyon	2.7	10.2
Plant fence	4.9	9.4
Small boulders	3.5	12.8
Sandy flat beach	4.2	4.0
Dense bamboo	5.0	11.6
Dry tall underbrush	3.6	8.4

Table 2.1: Mean power loss and shadowing variance obtained over the range 800-1000 MHz and with an antenna height of about 15cm [50].

subject is introduced in chapter 6 and chapter 7. When the distance to a remote receiver exceeds the transmission range of a single node, nodes may cooperate during transmission to combine their transmit signal-components and improve the transmission range.

Another possible communication mechanism is infrared. Infrared communication is license free and robust against interference from electrical devices which is one major concern in RF-based systems. RF-based transceivers are cheap and easy build solutions accordingly. However, IR is only feasible with direct line of sight between nodes.

An alternative approach is detailed in [51]. The authors utilise the optical medium for transmission. Of course wired communication is also possible and in fact often implemented as a communication medium in sensor networks.

In the following sections sensor networks that utilise RF-transmission technology are assumed.

Transceiver design

A transceiver is an entity that combines transmission and reception capabilities. A typical transceiver consists of a Radio Frequency (RF) frontend and a baseband processor. The RF-frontend processes the analog signal when it is actually being transmitted or received through the wireless channel. The signals in the radio frequency band are typically of a high frequency like, for instance, in the 2.4 GHz ISM (Industrial, Scientific and Medical) band.

Processing at these high frequencies would overtax the limited processing capabilities of sensor nodes. It is, indeed even uncommon to employ oscilloscopes that are capable of visualising frequencies higher than about 1 GHz, for example. Therefore, the radio frequency signals are transformed into digital signals by analog-digital converters (ADC) in an intermediate frequency (IF) section and are downconverted by digital downconverters (DDC) to lower frequencies of several kHz (cf. figure 2.2). These lower frequency signals

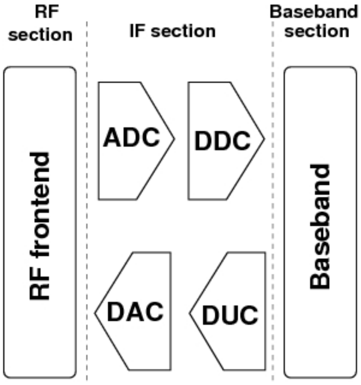


Figure 2.2: Schematic layout of a transceiver design.

are referred to as the baseband signals which are then processed by the sensor node's processor.

Transceiver states

Normally, a transceiver is in one of four states:

- Transmit
- Receive
- Idle
- Sleep

In the transmit and receive states, the transceiver transmits or receives RF-signals. These two states are typically the most energy-consuming for a sensor node. Consequently, the time in which the transceiver is in transmit or receive states should be minimised to reduce the energy-consumption of a node.

In the idle state, the transceiver is not actually receiving or transmitting but ready to receive an RF-signal. In this state, some parts of the receive circuitry are switched off to reduce the energy consumption.

In the sleep state major parts of the transceiver are switched off. Many architectures distinguish between several sleep states depending on the amount of parts that are switched off. For these sleep states the recovery times and energy required to wake up differ.

2.2 Sensor networks

Sensor networks are constituted from sets of sensor nodes that build up a wireless network and accomplish one or more tasks collaboratively. For sensor networks to be dispatchable in inaccessible terrains and to work unattended over a long time span, it is desired to

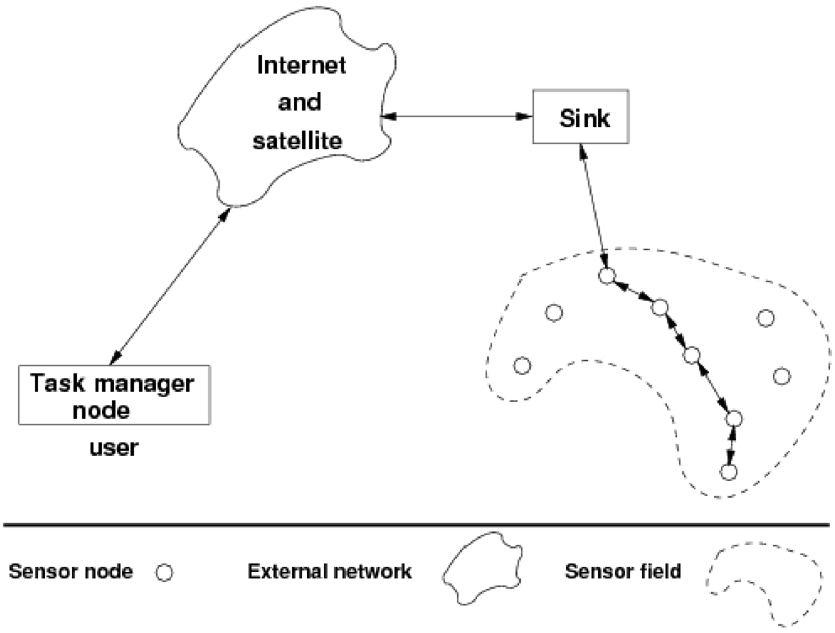


Figure 2.3: Schematic illustration of a sensor network implementation [8]

incorporate self-organising capabilities. These enable the network to establish a wireless connection among nodes as well as routing paths along the network among other things.

Applications for sensor networks range from health care to military or disaster monitoring to home appliances.

Figure 2.3 illustrates a possible implementation of a wireless sensor network. The sensor nodes might be scattered in a field. The terminology of sensor networks refers to sources and sinks. For any communication between nodes in a network the source is the sender-node of the information and the sink is the recipient. Each node has the capability to collect data and to route data from a source to a sink either directly by a single hop or by a multi-hop protocol. The sink may communicate via an external communication network with a task-manager-node that is controlled and/or read out by a user.

In order to accomplish these tasks, wireless sensor networks require ad-hoc networking techniques. However, common ad-hoc networking protocols are usually not well-suited to wireless sensor networks since typically they differ in various aspects from wireless ad-hoc networks. Some of the differences are

- The number of nodes in a sensor network might be higher by several orders of magnitude than in ad-hoc networks
- Sensor nodes are deployed densely
- Sensor nodes are prone to failures
- Due to node failures or environmental impacts, the topology of a sensor network changes frequently

- Sensor nodes mainly use a broadcast communication paradigm, whereas most ad-hoc networks are based on point-to-point communications
- Sensor nodes are limited in power, computational complexity and memory
- Sensor nodes typically do not utilise global identification because of the large amount of overhead for the naming scheme and the large number of nodes

As we will see in section 3 the transmission range of a node is restricted due to growing signal decay with increasing transmission-distance. Consequently, the maximum distance for a single hop is restricted. The classical approach to reach a distant receiver is to apply a multi-hop protocol for transmission. In such a protocol, intermediate nodes are utilised as relay nodes that forward received packages along the network.

It is sometimes claimed that multi-hop transmission schemes are more energy-efficient than direct transmission schemes since the radiated energy for a direct connection is higher than the sum of the radiated energy of relay nodes along the transmission path. The intuition behind this assumption is that the decay in signal strength is more than linear with increasing transmission-distance. This property of path-loss is detailed in section 3.1.2).

Nonetheless this assertion generally is not correct. We can easily reduce this assumption to absurdity by shortening the distance between relay nodes to a small ε so that the nodes are placed next to each other. Then the overall energy consumed by the extra nodes is then much higher than the gain in transmission energy. For a more comprehensive discussion on this effect we refer to [51].

Multi-hop transmission is only feasible when a chain of intermediate nodes exists so that the distance is smaller than the transmission range for every pair of adjacent nodes in this chain. If this is not the case, multi-hop transmission is not possible. Cooperative strategies to reach a distant receiver for which no intermediate nodes are required are introduced in section 6.. Alternatively, relay stations with increased transmission range can be utilised to bridge longer distances. Since the latter approach contradicts the WSN principle of utilising standard, low-power and low-cost nodes, this idea will not be taken into consideration in the following chapters.

2.2.1 Metrics to measure the quality of a WSN

Different wireless sensor network installations are utilised for distinct applications. Each application has other requirements for a wireless sensor network. In order to quantify the suitability of a concrete wireless sensor network instrumentation for a given application, special quality metrics are developed. Some typical metrics to derive the quality of a given wireless sensor network instrumentation are being discussed in the following paragraphs.

Fault tolerance

As nodes of a sensor network are cheap and low-power devices, they might fail in the course of the operation of a sensor network due to power shortage, physical damage or environmental interference, for instance. Ergo, a network is required to provide some means of fault tolerance, for example, such as an ability to sustain network functionality in the presence of node failures. The reliability $R_k(t)$ or fault tolerance of a sensor node is modelled in [52] by using the Poisson distribution. The probability of not having a failure within a time interval $(0, t)$ is then

$$R_k(t) = e^{-\upsilon_k t} \tag{2.1}$$

where υ_k is the failure-rate of sensor node k and t is the time period.

Scalability

Another relevant aspect for sensor networks is the scalability of the network. As sensor networks consist of densely deployed nodes, the number of nodes in a network may easily reach extreme values. The density might range to hundreds of nodes in a region of no more than 10 m^2. In order to be able to support such huge and dense networks, the development of new protocols is necessary. Not all nodes are equally well-suited for a given network density as the transmission range my be different. In order to be able to classify how well a given node is suited for a predefined network installation, a quantitative metric describing the network density is required.

In [53] the density of a sensor network is calculated as

$$\eta(R) = \frac{N \cdot \pi k^2}{A} \tag{2.2}$$

where N is the number of nodes, A is the region and R is the transmission range.

2.2.2 Mobility in wireless sensor networks

Often mobility of some kind is involved in practical WSN environments. Generally, three types of mobility may apply. These are the mobility of nodes, the mobility of sinks or mobility of objects that are monitored by the network.

When the wireless nodes themselves are mobile we speak of node mobility. This might be the case when sensor nodes are attached to moving vehicles or also cattle. In the presence of node mobility, the network is required to frequently reorganise, for example, connection tables and routing paths.

Generally, the trade-off for this reorganisation is to achieve sufficient organisation of the network at reasonable overhead in communication and energy consumption.

When sinks are mobile such as a user with a PDA walking alongside the network, the same challenges occur as in the scenario of mobile nodes, but the effects are restricted to nodes near the mobile sinks [54].

A common case in wireless sensor networks is that a monitored object is mobile. In this case mobility mainly affects the activity of nodes. While nodes that are far away from the moving object can remain in a sleep mode, nodes near the object are required to wake up from sleep. In this context algorithms to ascertain the nodes that wake up next with regard to the movement of the object are especially interesting. In [55] this has been introduced as the Frisbee-model along with algorithms to handle the wake-up front of nodes.

2.3 MAC protocols

Medium access control (MAC) protocols coordinate the communication among nodes or, more precisely, they coordinate the times when a number of nodes access the shared medium.

Since energy consumption is one of the key problems in wireless sensor networks and communication is typically the most energy consuming task, an important aspect of a MAC protocol is its ability to reduce the energy consumption of a node. Accordingly these protocols might switch the transceiver into a sleep mode.

Medium Access Control (MAC) protocols are the first protocol layer above the physical layer and thereby are impacted by the physical layer properties.

2.3.1 Requirements and design constraints

Requirements for MAC protocols are a high throughput, stability, fairness, low access delay (time between packet arrival and first attempt to transmit it), low transmission delay (delay between reception of a packet and first approach to further forward the packet over the wireless channel) and a low protocol overhead.

The respective overhead may result from collisions, from the exchange of extra control packets but also from protocol overhead, such as large header or checksum fields.

Typically, as nodes communicate over a wireless channel a high robustness against bit-errors is also required. Furthermore, a constant problem is the energy consumption due to protocol design. Popular protocol-related energy problems result from collisions, overhearing and idle listening.

Collisions: Whenever a collision occurs the transmission energy spent for all packets involved in the collision is wasted. Collisions can be avoided by design, for instance, through fixed assignment schemes such as time division multiple access (TDMA) or by demand assignment protocols. Naturally, another way to reduce the probability of a collision is to reduce the number of packets transmitted in the network. Therefore, when the expected load in a sensor network is sufficiently low, collision avoidance schemes can be omitted as they also require channel resources.

Overhearing: Due to the broadcast nature of the wireless channel a transmitted packet is received by all neighbouring nodes and not only by the intended recipient. Although the non-destination nodes might discard the packages received, they would have spent energy on their reception in the first hand. Especially in dense sensor networks, schemes that reduce the frequency of overhearing may have a significant gain in energy resources [56, 57]. However, it must be observed that overhearing is wanted

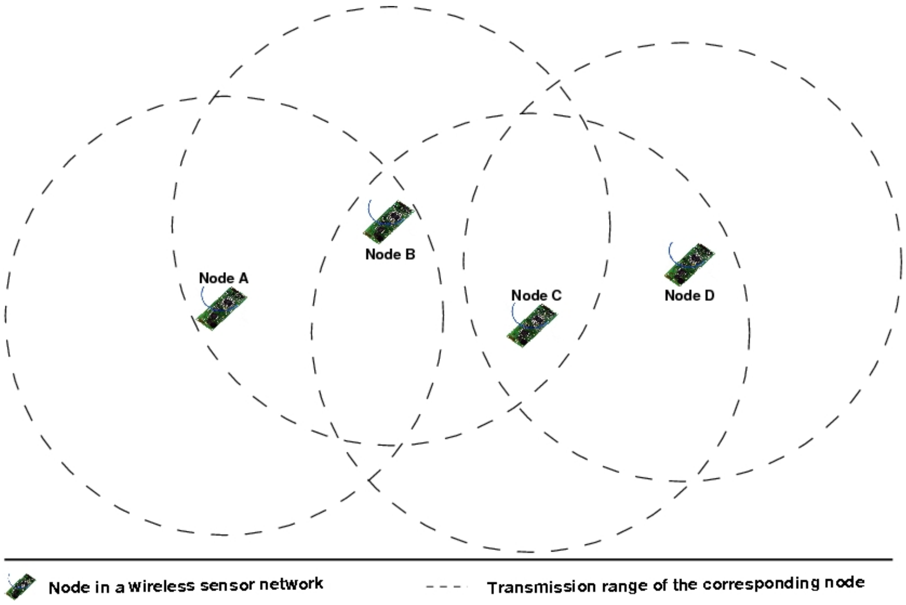

Node A
Node B
Node C
Node D
Node in a wireless sensor network
Transmission range of the corresponding node

Figure 2.4: Schematic illustration of the hidden node problem and the exposed node scenario

at times. Multi-hop-routing, for instance, is only possible when nodes overhear and forward packets that are not addressed to them.

Idle listening: A node in its idle state is ready to receive a packet but not actually transmitting or receiving information. Although the energy consumption in this state is less resource-demanding than for the transmit or receive states, the energy consumption might be significant for longer idle periods. That is why it is desirable not only to reduce the time a node remains in transmit and receive states but also to reduce the time a node resides in its idle state. One viable solution is to switch to sleep mode more often. However, since mode-changes cost additional energy, the frequency of these also leads to a trade-off which needs to be considered in the design of protocols.

A prominent problem in wireless sensor networks is the hidden node problem. It occurs when the nodes transmit in a carrier sense multiple access (CSMA) protocol Consider the sensor placement in figure 2.4. In the figure dashed lines represent transmission ranges. When node A is transmitting information to node B and node C is transmitting information to node B at the same time, both nodes, A and C are not capable of sensing the transmission of the respective other node and thus are not capable of preventing the collision occurring at node B when both, node A and node C transmit simultaneously.

Another aspect is the exposed node scenario. In this scenario, node B transmits a packet to node A and a short time later, node C is ready to transmit a packet to node D. Both transmissions are possible simultaneously; but since node D is out of reach of node B and node A cannot be reached by node , node C would not transmit its data because it

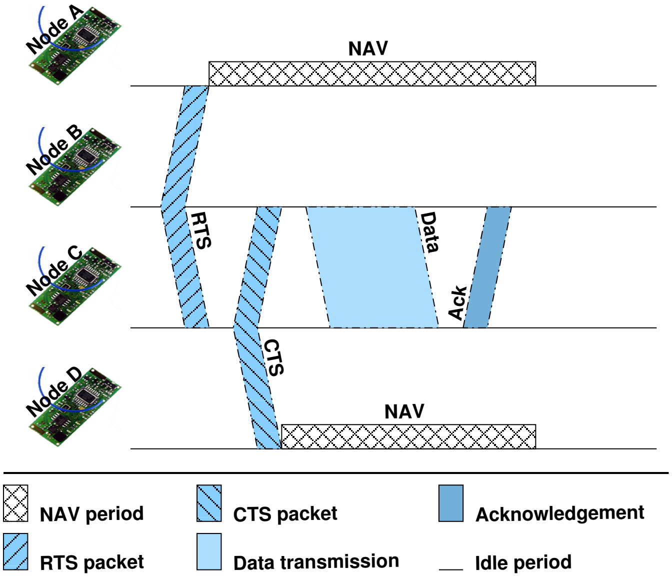


Figure 2.5: Schematic illustration of IEEE 802.11 RTS/CTS handshake

senses that the channel is already in use and concludes that a transmission would lead to a collision. In this way, the transmission is delayed unnecessarily.

2.3.2 A standard communication protocol for wireless sensor networks

Classical, wired communication protocols can, of course, also be utilised for wireless sensor networks; but the special requirements of the wireless communication in a dense network of resource-restricted nodes may lead to a low efficiency of these protocols. For the purpose of illustration the application of the IEEE 802.11 protocol [58] with its RTS/CTS handshake will be taken as an example. RTS and CTS packets are special control packets. The abbreviations stand for Ready to Send and Clear to Send. Figure 2.5 illustrates the RTS/CTS handshake in IEEE 802.11 protocols.

In this example node B wants to transmit data to node C. In order to announce its intention, node B sends an RTS packet. This packet contains the addressee of the communication and a probable transmission time for the transmission. Both neighbouring nodes A and C receive the RTS message and therefore the estimated transmission time. Based on this information, node A allocates an internal timer called Network Allocation Vector (NAV). The NAV indicates a busy medium so that node A will not try to access the medium. Node C, the addressee of the RTS packet, answers with a CTS packet that contains an estimation of the time required to transmit the data once again. Apart from node B all other nodes neighbouring C receive this CTS packet and can calculate a NAV again. Since the NAV was calculated by nodes receiving both the CTS and the RTS packets all nodes neighbouring both communicating nodes B and C calculate a NAV and free the medium in order to avoid collisions (it will be shown later that collisions can

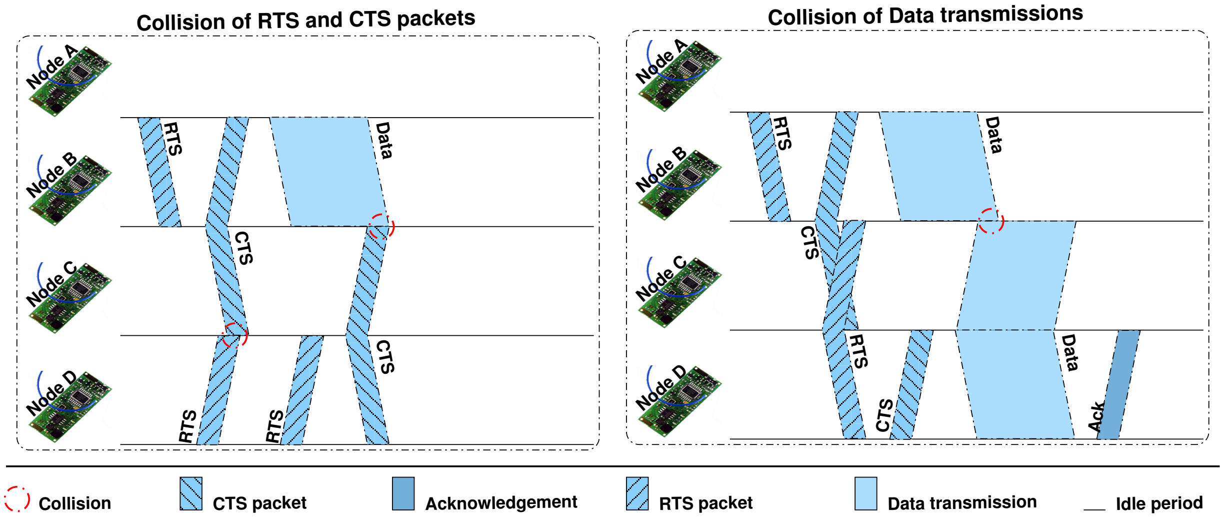


Figure 2.6: Schematic illustration of the hidden node problem and the exposed node scenario

occur anyway). After receiving the CTS packet, node B transmits the data and awaits an acknowledgement from node C.

Problems with RTC/CTS handshake

While this RTS/CTS handshake scheme is applied successfully in the IEEE 802.11 protocol, in dense wireless sensor networks, the frequent transmission of control packets (RTS, CTS, Ack) may lead to congestion and collision. Figure 2.6 illustrates two problems of the RTS/CTS handshake procedure in dense wireless networks.

One problem may occur when one node is not able to receive a CTS packet due to a collision with an RTS packet. In this case (see figure 2.6, left), some node A sends an RTS packet to a node B that in turn answers with a CTS packet. However, this CTS packet is not received by all neighbouring nodes of B. In particular, node D transmits an RTS packet to node C so that the RTS and CTS packets collide at node C. Consequently, node C neither answers the RTS request from D nor does it know of the CTS transmission by node B. Node D, after not receiving a CTS response retransmits its RTS request. This time it is acknowledged by node C. Unfortunately, now the CTS transmitted by node C collides at node B with the data transmitted by node A.

Another problem is the collision of data transmissions as illustrated in figure 2.6, right. Node A requests a transmission to node B again. This time the CTS transmitted by node B overlaps with an RTS request sent by node C to node D. As both, node A and node D did not know of this problem, the protocol is continued as normal until a collision in the data transmission from node C and node A occur.

In both cases, the transmission was not successful and has to be repeated at the cost of further energy for the transmission of control messages and data between nodes. For long

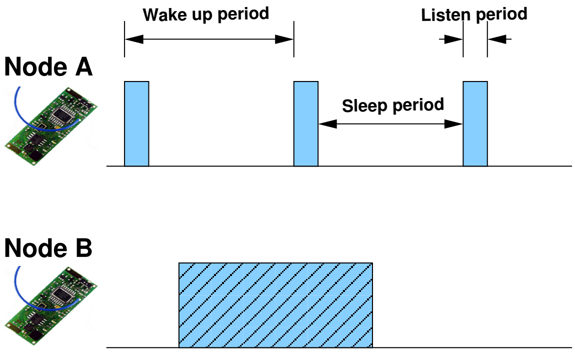


Figure 2.7: A generic Wake Up Radio scheme

data packets this overhead can be neglected. However, in wireless sensor networks, the data to be transmitted is typically small. It is arguable whether this high control overhead of the RTS/CTS handshake is worthwile when only small data packets are transmitted. Also, longer data packets are more likely to suffer from bit-errors and must be retransmitted more often so that in wireless sensor networks long data packets are typically fragmented into smaller ones.

2.3.3 Wake-up radio

Since transmission and receive states are the most energy consuming tasks and the idle task is also quite resource consuming, one might think of building a radio that has a very low energy consumption or that stays in the sleep mode for a maximum amount of time. Currently the optimum feasible solution known includes a special radio with very low energy consumption which is able to wake up the standard radio of a node on demand. The main idea of a wake-up radio is to implement a very low-power receiver that has no other task than to wake up the regular transceiver of a node. In this manner, a node can be in the receiving state only when it is requested to listen to the channel. Unfortunately, such a radio has not yet been designed, although various current projects are aiming to build it as part of their project[1].

Therefore, a popular implementation is for the radio to wake up periodically from the sleep mode to listen to the channel for potential communication requests. Figure 2.7 illustrates this concept.

Here the wake-up period denotes the time between two consecutive listen periods of one node. In a listen period the node listens to the channel for communication requests. In the sleep period the node does not listen. We denote the duty cycle of a node by the fraction of a listen period to wake-up periods:

$$\text{Duty Cycle} = \frac{\text{Length of the listen period}}{\text{Length of the wake-up period}}. \tag{2.3}$$

[1] One example is the FP7-ICT-2007-2 project CHOSeN (http://www.chosen.eu)

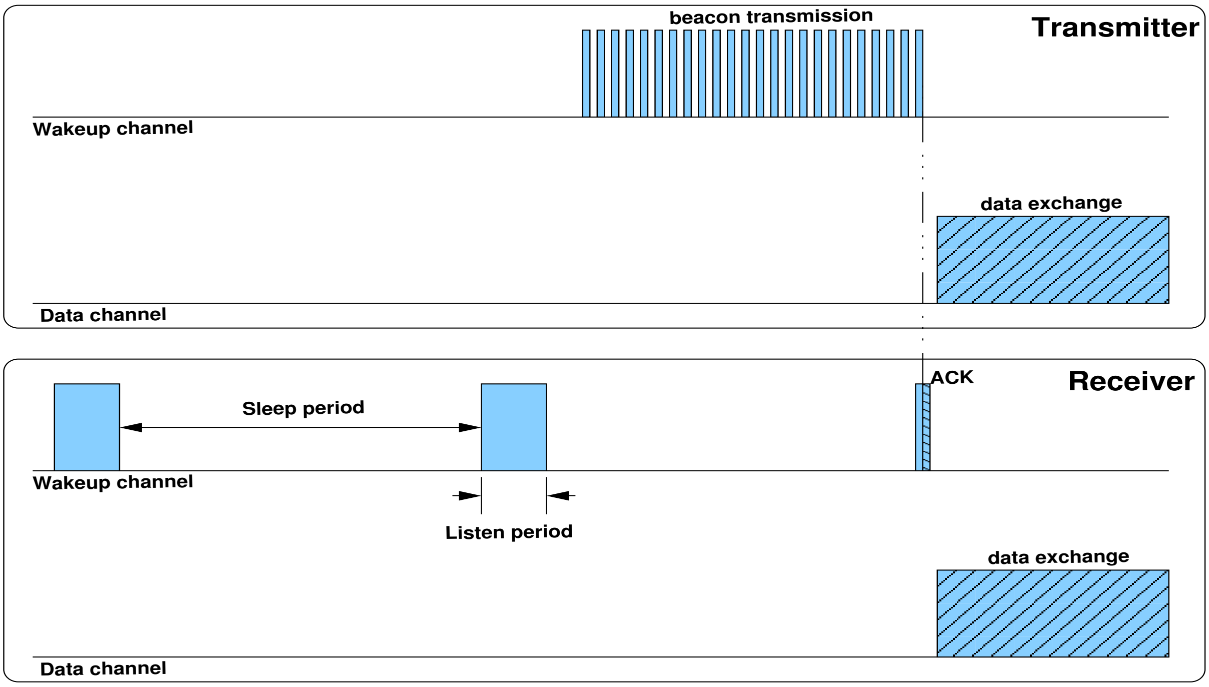


Figure 2.8: Schematic illustration of the STEM-B MAC protocol

A second node, willing to communicate has to transmit a preamble that lasts not more than one wake-up period so that the listen period of the receiving node is covered. Two often cited protocols that implement this basic idea are the STEM protocol and the S-MAC protocol. For the STEM protocol an extra channel is utilised for data transmission and notification of sleeping nodes. In the case of the S-MAC protocol, the same channel is utilised.

Sparse topology and energy management protocol

The sparse topology and energy management protocol (STEM) provides a solution for the idle listening problem [59]. The protocol is suited best to networks in which the usual data traffic is low and in which traffic is mainly in response to events or external stimuli. The network states are then logically divided into a monitor state and a transfer state. In the former nodes are idle, while in the latter significant sensing and communication activity can be observed.

In STEM idle listening in the monitor state is minimised. Figure 2.8 depicts the operation of STEM schematically.

Two different channels are utilised for STEM: The data channel and the wake-up channel. The time on the wake-up channel is divided into sleep periods and listen periods. The listen period is considerably shorter than the sleep period. A node in the listen period switches on its receiver for the wake-up channel and waits for signals on the channel. In positive case, the node starts a packet transfer on the data channel with the transmitter.

Two options to wake up the receiver for the transmitter are discussed in the available literature.

STEM-T Instead of transmitting packets the transmitter sends out a continuous busy-tone on the wake-up channel that lasts at least one wake-up period. The tone does not contain any address information. A transceiver capable of transmitting and sensing busy-tones can be significantly cheaper and less energy consuming than a normal transceiver.

All receivers that observe this busy-tone wake up and wait until the end of this tone. No acknowledgement is transmitted by the receive nodes.

Afterwards, the transmitter starts the communication on the data channel. Receive nodes then deduce from the packet exchange on the data channel whether they can go back to sleep or participate in the communication.

STEM-B The transmitter transmits beacon signals on the wake-up channel periodically. A beacon indicates the MAC addresses of transmitter and receiver. When the receiver receives such a beacon, it sends an acknowledgement packet on the wake-up channel. This causes the transmitter to stop in its beacon transmission. Then both nodes exchange data on the data channel following an arbitrary MAC protocol. The transmitter sends the beacon packets for at least one wake-up period so the wake-up period of the remote receiver is definitely hit.

Note that in STEM-B no carrier-sensing takes place before the transmitter starts with its beacon packages. But when a collision occurs, a node experiencing energy on the channel does not go to the sleep state again and then behaves as if it would follow the STEM-T protocol. It is to be noted that in such a case no acknowledgement is transmitted.

Observe that, in STEM a node entering the listen period does not transmit a packet. This implementation differs, for instance, from the mediation device protocol detailed in section 2.3.3.

Sensor-MAC

The sensor-MAC (S-MAC) protocol provides mechanism to circumvent idle listening, collisions and overhearing [57, 60]. This is a single channel protocol as well.

In S-MAC, each node alternates between a listen period and a sleep period (cf. figure 2.9). In the sleep period, the transceiver of the node is idle while in the listen period the node can transmit and receive packets. The listen period is divided into the three distinct phases, namely SYNCH, RTS and CTS.

SYNCH The SYNCH-phase is divided into time slots. In these time slots nodes contend with each other according to a CSMA scheme with additional backoff time. In this phase, a node accepts SYNCH-packets from its neighbours. In each of these packets, the sleep-listen schedule of an individual neighbour is detailed.

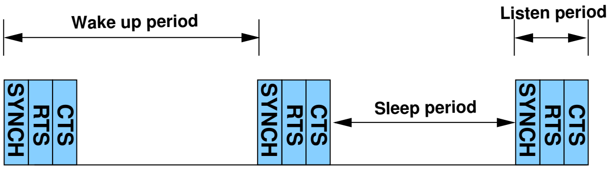


Figure 2.9: Schematic illustration of the Sensor-MAC protocol

RTS In the RTS-phase, a node expects RTS-packets from its neighbours. All neighbours again contend with each other in this phase according to a CSMA scheme with additional backoff time.

CTS In the CTS-phase, a node transmits a CTS-packet when a RTS-packet was received in the previous phase. Afterwards the packet exchange is extended to the normal sleep phase of the node for data transmission.

The schedules received in the SYNCH-phase are stored in a schedule table by the receiving nodes. Since a node then knows the schedule of neighbouring nodes, it wakes up at appropriate times to broadcast its own schedule.

This process of schedule-broadcasting is repeated periodically in order to keep nodes in synch and to provide schedule-information to newcomers.

One benefit of this protocol is that we can arrange the sleep and listen periods of neighbouring nodes so that their listen periods coincide. In such a case a node can reach all its neighbours in each listen period.

This scheme is called virtual clustering. Neighbouring nodes then agree on a schedule to build these clusters. Generally, when a node receives a SYNCH-message, it adapts the received schedule and roadcasts it in one of the next SYNCH-periods. Otherwise, the node randomly generates a new schedule and broadcasts this schedule periodically. When a node that already has its own schedule receives a new schedule, it either discars its own schedule in favour of the new one or follows both schedules. This decision depends on wether it is the only node in its current virtual cluster or not.

The drawback of this solution is that the per-hop-latency of the scheme generally deteriorates towards the sleep period. In [57], however, an adaptive listening scheme was presented that roughly halves the per-hop-latency of S-MAC.

Another drawback of the S-MAC protocol is that it is not easy to adapt the length of the wake-up period when the network load changes as many nodes in a virtual cluster would have to be synchronised then. To overcome this problem the T-MAC protocol was presented in [61]. In principle, this is an S-MAC implementation that shortens the listen period adaptively when no activity is sensed on the medium for some time.

Mediation device protocol

Another solution that deviates from the generic approach depicted in figure 2.7 is the mediation device protocol. In this protocol a mediation device is utilised as an intermediate

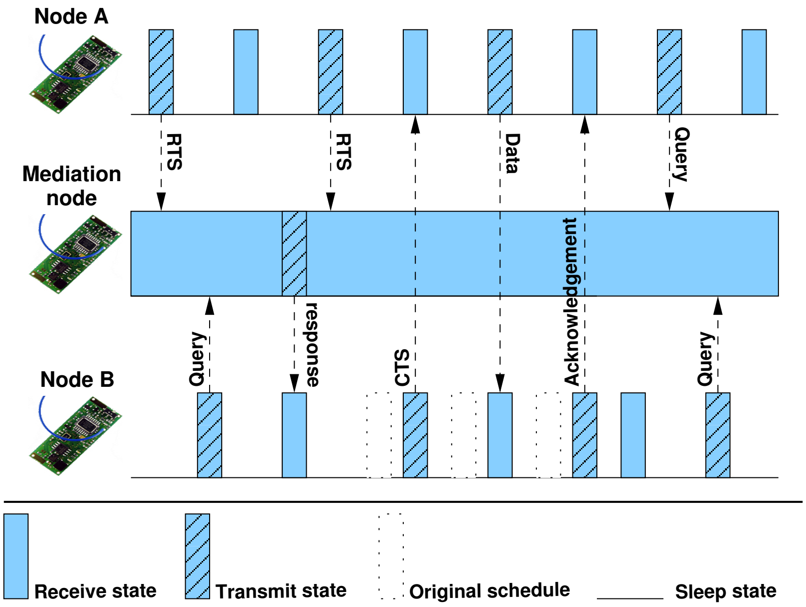


Figure 2.10: A schematic illustration of the mediation protocol

node to coordinate the communication between other nodes. Figure 2.10 schematically illustrates the idea of this communication protocol.

All but the mediation node have their own, not synchronised sleep schedules. The mediation node is never in the sleep mode. When a node wakes up and wants to send data, it informs the mediation node of the address of the receiver repeatedly until an answer is received. A Node that has data to send transmits an RTS packet to the mediation node that also contains the address of the addressee. A node that wakes up and has no data to transmit, requests data from the mediation node via a Query packet. If no packet is received in reply to this Query request, the node returns to the sleep mode.

But when the Query of the node is answered by the mediation node in a Query response, it sends a CTS-packet directly to the node willing to transmit. Synchronisation of transmit and receive periods is achieved since the mediation node obverses the frequency of RTS-packets from the transmitting node. This transmission timing is included in the CTS-packet sent to the receiving node together with the address of the transmitter node. The receiver can thus adapt its timing. Then the transmitter transmits the data and awaits an acknowledgement from the receiving node. Afterwards the timing of the receiving node is changed back to the original schedule and both nodes query the mediation node for transmission requests again.

3 Wireless communications

> *Communication technologies are necessary, but not sufficient, for us humans to get along with each other. [...]Technology tools help us to gather and disseminate information, but we also need qualities like tolerance and compassion to achieve greater understanding between peoples and nations.*
>
> (Sir Arthur Charles Clarke – British author, inventor and futurist.[62])

Similar to wired communication, wireless communication is about transmitting information through a medium – in this case the air. An electromagnetic waveform is transmitted between communication partners and information is modulated on top of this signal wave. Basically, some phenomena observed in a wireless channel such as path-loss are similar in a wired channel. However, wireless communication is more challenging than wired communication as the medium utilised is shared between multiple communication partners. Therefore, new effects are observed such as slow-fading and fast-fading as well as interference between transmitted signal-components. We will introduce basic properties of the wireless radio channel in the following sections.

3.1 Aspects of the mobile radio channel

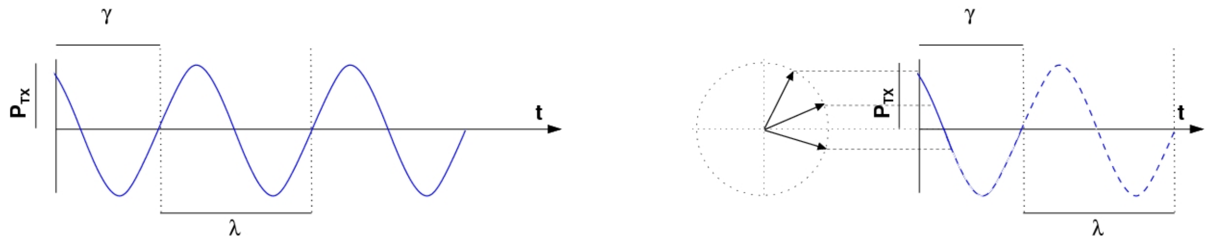
Electromagnetic signals, utilised in wireless communication are radiated in wave-form. These waves propagate from a transmitter omnidirectionally (in all directions/dimensions) at the speed

$$c = 3 \cdot 10^8 \frac{m}{s}. \tag{3.1}$$

c is the speed of light and approximates the propagation speed of electromagnetic waves. An intuitive analogy (in two dimensions) for the propagation of electromagnetic waves are the waves on the water surface created by a single drop into the water.

The signal is characterised by the transmission power P_{TX}, by the frequency f and by a phase-offset γ. The wavelength

$$\lambda = \frac{c}{f} \tag{3.2}$$



(a) Signal wave with phase-offset γ and wavelength λ

(b) Analogy between the wave form of an analog signal and a rotating vector

Figure 3.1: Schematic illustration of a signal wave

of a signal describes the length (in meters) of one signal period. Figure 3.1(a) schematically depicts an electromagnetic wave with frequency $f = \frac{c}{\lambda}$, phase-offset γ and transmit power P_{TX}.

We can imagine that this wave-form is created by the real part of a rotating vector $\zeta = \Re\left(e^{j(ft+\gamma)}\right)$ as depicted in figure 3.1(b). The height $\cos(\zeta)$ of the vector $\zeta = \Re\left(e^{j(ft+\gamma)}\right)$ equals the signal strength P_{TX} and the rotation speed determines the frequency of the signal.

3.1.1 Superimposition of electromagnetic signals

Since electromagnetic signals are generally transmitted undirected via a broadcast channel, several signal-components originating from the same signal source may reach the receiver on multiple signal paths. Signals are reflected from obstacles or also might change their propagation direction by diffraction at edges of objects. Since the propagation speed of all signal-components is identical, signals propagated along indirect signal paths take longer than a signal-component along the direct line of sight (LOS).

At a receiver all incoming electromagnetic waves add up to one superimposed sum-signal. In general, this is true for signals of arbitrary frequency and from an arbitrary signal source. However, in practice, a receiver is capable of filtering a restricted frequency spectrum so that only signal-components falling into this spectrum are taken into account by the receiver. Figure 3.2 illustrates how a single sum-signal is composed of individual signal-components. The interference of other signal-components can be constructive or destructive. Constructive interference occurs when identical signal-components arrive at a remote receiver in phase so that the sum-signal reflects an amplification of the individual signal-components. Destructive interference occurs when signal-components arrive out of phase. Naturally, destructive interference is the rule and might lead to a heavily distorted sum-signal that is not distinguishable by a receiver.

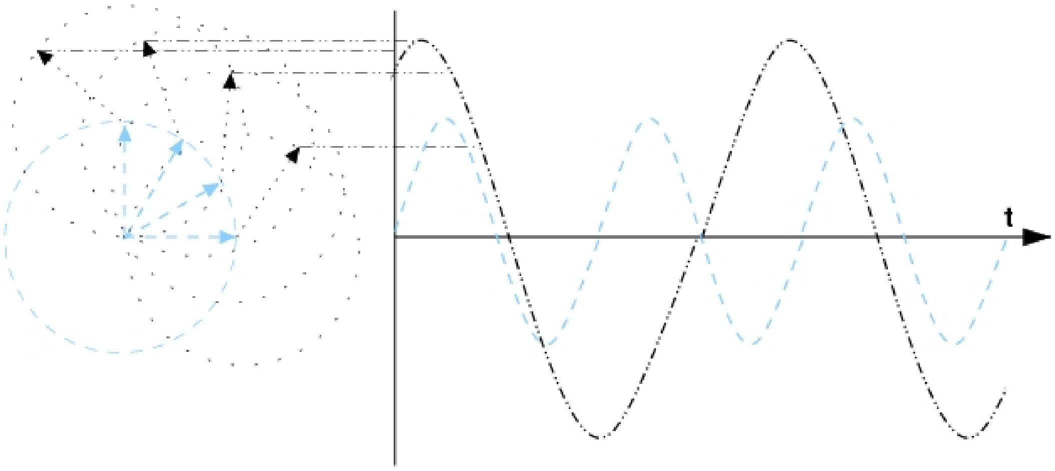


Figure 3.2: Composition of a superimposed sum-signal from two individual signal-components

3.1.2 Path-loss

Electromagnetic waves decrease in signal strength when propagating over the wireless channel. The order of this decay varies in different environments. With higher frequencies the effect is greater but it can also be reduced with improved antennas (directed ones, for example). The actual path-loss in a given environment is heavily dependent on the environmental characteristics. For analytic consideration, the path-loss in a given scenario can therefore be approximated at most. For various scenario classes, different formulae are proposed to approximate this path-loss.

A general formula to calculate the received signal strength (RSS) in free space is the Friis free-space equation [49]:

$$P_{RX} = P_{TX} \cdot \left(\frac{\lambda}{2\pi d}\right)^2 \cdot G_{TX} \cdot G_{RX} \tag{3.3}$$

In this formula, P_{TX} and P_{RX} denote the transmit and receive power (i.e. the signal strength), d is the distance in meters between sender and receiver and G_{TX}, G_{RX} are the gain of the transmit and receive antennas. Observe that the distance d impacts the received signal strength quadratically. This is due to the decrease of the electric field, which decreases linearly in the distance d so that the power per square meter decreases as d^{-2}.

The free-space equation can be utilised for outdoor-scenarios in which no reflections of the signal-components are considered and in which only a single direct signal-component from each transmitter is assumed then. This assumption holds in wireless networks approximately, as long as the receiver is sufficiently far away from the transmitter in the so-called far-field distance. For cellular systems like GSM or UMTS this distance is in the range of 1 km while for short-range systems like WLAN it is in the range of 1 m [49].

The path-loss is defined as the fraction of the transmitted signal strength P_{TX} to the received signal strength P_{RX}:

$$PL^{FS}(\zeta_i) = \frac{P_{TX}(\zeta_i)}{P_{RX}(\zeta_i)} \tag{3.4}$$

A path-loss model suited in buildings or densely populated areas is the log-distance path-loss model [63]:

$$PL^{LD}(\zeta_i) = \frac{P_{TX}(\zeta_i)}{P_{RX}(\zeta_i)} = 10^{\frac{L_0}{10}} \cdot d^{\alpha} \cdot 10^{\frac{x_g}{10}} \tag{3.5}$$

or in dB:

$$PL^{LD}(\zeta_i) = P_{TX}(\zeta_i) - P_{RX}(\zeta_i) = L_0 + 10 \cdot \alpha \cdot \log_{10}\left(\frac{d}{d_0}\right) + X_g[dB] \tag{3.6}$$

In these formulae, L_0 represents the path-loss at a reference distance d_0 in dB, while d represents the length of the path and α is the path-loss exponent. X_g is a random variable with zero mean that represents the attenuation due to fading in dB. When only slow-fading occurs, this variable may follow a Gaussian distribution with standard deviation σ in dB (or a log-normal distribution of the received power in Watt). When only fast-fading occurs, this gain may be modelled as a variable following a Rician or Rayleigh distribution. Both these distributions are introduced in section 3.1.4.

Often no direct signal component exists in urban scenarios; there reflection of signals and multipath propagation are common and have to be considered. Effectively, the received strength of the signal is decreased at a higher pace in such scenarios. A diverse set of distinct models to account for this increased path-loss are proposed in the literature. For the purpose of our lecture these models will not be detailed since generality is not achieved by any of these. It is, however, important to note that the path-loss differs in various scenarios and that this impact can be expressed by the path-loss exponent α. In the Friis equation, α was set to 2. In other environments than in free space, it may also take other values up to 6 (for instance shadowed areas and obstructed in-building scenarios) [49] but also $\alpha < 2$ is possible. Table 2.1 depicts various path-loss exponents for distinct wireless scenarios.

3.1.3 The Doppler effect

Due to the relative movement of the transmitter and the receiver the frequency of the received signal differs from the frequency of the signal as it was transmitted by the transmitter. This is called the Doppler effect. It can be observed when the relative distance between a transmitter and a sender is altered during transmission. The received frequency is calculated as $f_c + f_d$ then, where f_c denotes the original carrier frequency at the transmitter and f_d denotes the frequency-shift experienced due to the Doppler effect. For a wavelength λ a relative movement between a receiver and a transmitter of v and an angle between the movement direction of the transmitter and the direction of the signal transmission of α, it is

$$f_d = \frac{v}{\lambda} \cos\alpha \tag{3.7}$$

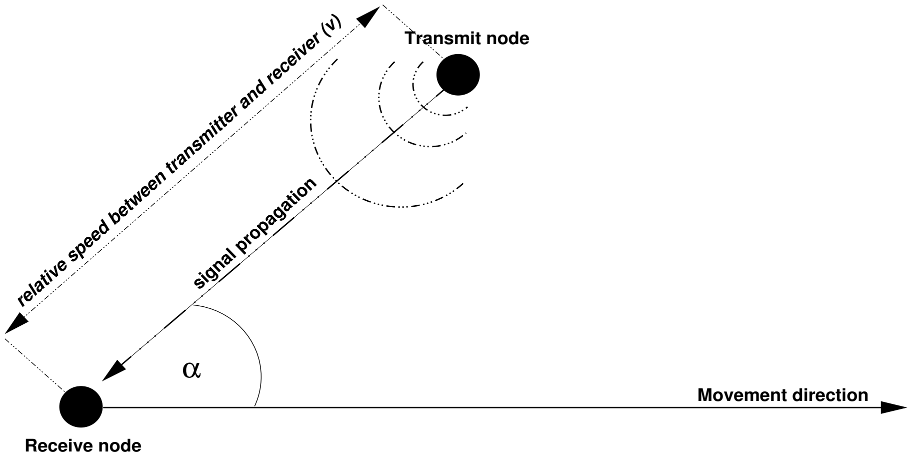


Figure 3.3: Schematic illustration of the Doppler effect

Figure 3.3 illustrates this interdependency.

Clearly, the Doppler effect becomes greater with increasing relative velocity v. Note that not the actual velocity of transmitter or receiver with respect to any reference point are relevant, but that the relative velocity with which the transceiver approaches the transmitter is. In particular, when transmitter and receiver move at identical speed and in an identical direction, no Doppler effect is observed. Observe that, due to the factor $\cos \alpha$, the Doppler effect is at its greatest, when the angle α is 0. For the same reason no Doppler effect is observed with $\alpha = 90°$.

This means that the greatest Doppler effect is observed when the transmitter is moving on a direct line towards (or away from) the receiver. The Doppler effect is also 0 when a transmitter is circling around a receiver at constant distance, regardless of its speed v. The reason for this property is the omnidirectional propagation of a signal from a transmitter to a receiver. Thereby from a receiver-point-of-view the signal received is identical wether the transmitter is circling around the receiver or sustains its position during transmission.

3.1.4 Fading

As could be shown in the last section, the signal strength generally deteriorates with greater transmission-distance. Furthermore, the signal quality fluctuates with location and with time, even when the distance to the receiver is kept identical. There are two effects that account for this problem and which are referred to as slow-fading and fast-fading.

Slow-fading occurs when the coherence time of the channel is large relative to the delay constraint of the channel. This is the result of environmental changes so that signal paths are blocked or changed. Examples are trees moving in the wind, moving vehicles, open or closed doors or also pedestrians. The amplitude change caused by these effects can be modeled by a log-normal distribution with a standard deviation according to the log-distance path-loss model.

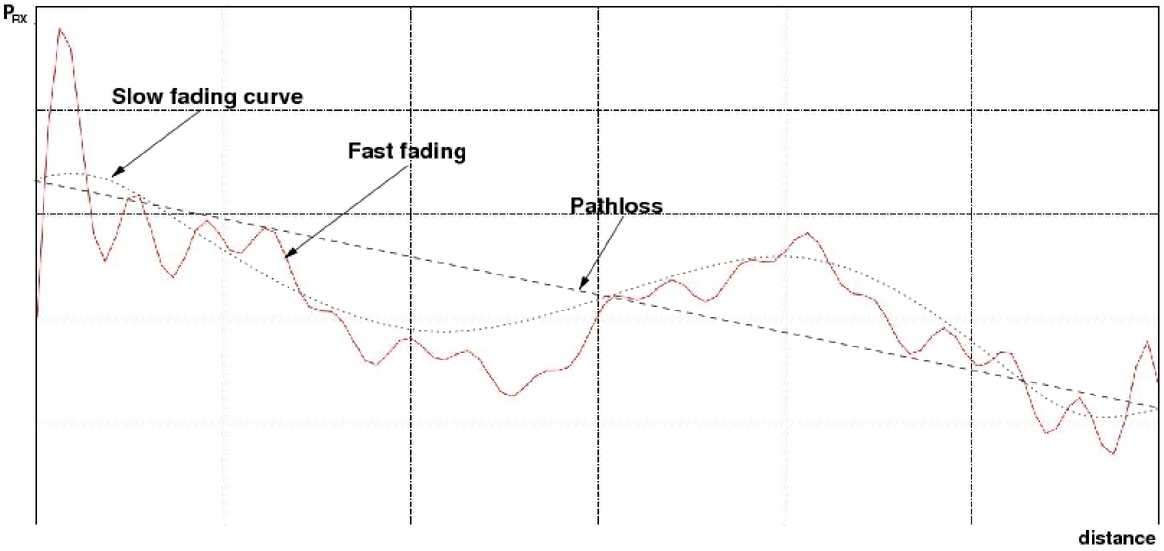


Figure 3.4: Schematic illustration of fading effects on the RF-signal

In contrast, fast-fading describes a phenomenon that several signal-components on multiple paths cancel each other out on the RF-level or also add up to a stronger signal. Figure 3.4 depicts these effects schematically. Fading incursions of a fast-fading profile are to be expected in the distance of $\frac{\lambda}{2}$ so that the quality of the channel changes drastically even when the receiver is moved for only a short distance. Typically statistical measures are utilised to model the fading profile for simulation-studies and to model the channel in a given environment. Popular probability-distributions to estimate the probability of fading incursions are, for example, the Rice and Rayleigh distributions.

When a direct line of sight is exists between receiver and transmitter, a dominating signal-component exists. Then we can assume that the fast-fading is weakened because of the dominance of the main signal. This case can be modelled by the Rice distribution which is defined as

$$f(A) = \frac{A}{\sigma^2} e^{-\frac{A^2+s^2}{2\sigma^2}} I_0\left(\frac{As}{\sigma^2}\right) \tag{3.8}$$

In this formula, $I_0(x)$ is the modified Bessel function with the order 0. It is

$$I_0(x) = \frac{1}{2\pi} \int_0^{2\pi} e^{x\cos(\Psi)} d\Psi \tag{3.9}$$

The value s in equation (3.8) describes the dominant component of the received signal and σ is the standard deviation. We define the Rice factor as

$$K = \frac{s^2}{2\sigma^2}. \tag{3.10}$$

This factor impacts the probability-density-function of equation (3.8) or impacts, intuitively, the most probable outcome for the probability function.

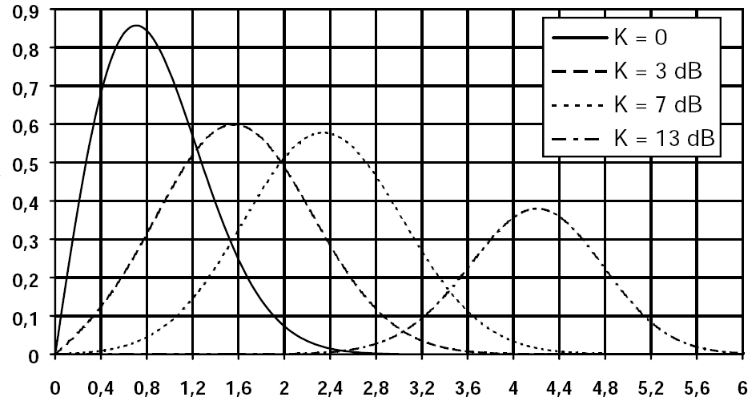


Figure 3.5: Probability-density-functions for various values of K

For $K = 0$, the Rice distribution migrates to the Rayleigh distribution:

$$\begin{aligned}
\lim_{K\to 0} f(A) &= \lim_{K\to 0} \frac{A}{\sigma^2} e^{-\frac{A^2}{2\sigma^2}-K} I_0\left(\frac{A\sqrt{2K}}{\sigma}\right) \\
&= \lim_{K\to 0} \frac{A}{\sigma^2} e^{-\frac{A^2}{2\sigma^2}-K} \frac{1}{2\pi}\int_0^{2\pi} e^{\frac{A\sqrt{2K}}{\sigma}\cos(\Psi)} d\Psi \\
&= \frac{A}{\sigma^2} e^{-\frac{A^2}{2\sigma^2}-0} \frac{1}{2\pi}\int_0^{2\pi} e^{\frac{A\sqrt{2\cdot 0}}{\sigma}\cos(\Psi)} d\Psi \\
&= \frac{A}{\sigma^2} e^{-\frac{A^2}{2\sigma^2}} \qquad (3.11)
\end{aligned}$$

Figure 3.5 illustrates probability density functions for various values of K.

The Rayleigh distribution

$$\frac{A}{\sigma^2} e^{-\frac{A^2}{2\sigma^2}} \qquad (3.12)$$

describes the probability-density-function of the amplitude of the received signal if we consider $n \gg 1$ signals of equal strength. It is used when all components of a received signal are indirect and approximately equal in power as, for example, in urban scenarios with dense blocks of houses.

With a large K (very small dispersion component) the rice distribution evolves into the

Gauss distribution:

$$\begin{aligned}
I_0(x) &\rightarrow_{x\gg 1}\rightarrow \frac{e^x}{\sqrt{2\pi}} \\
\Rightarrow f(A) &\rightarrow_{x\gg 1}\rightarrow \frac{A}{\sigma^2}e^{-\frac{A^2}{2\sigma^2}-K}\frac{e^{\frac{A\sqrt{2K}}{\sigma}}}{\sqrt{2\pi\frac{A\sqrt{2K}}{\sigma}}} \\
f(A) &= \frac{A}{\sigma^2\sqrt{\frac{2\pi}{\sigma}}\sqrt{A\sqrt{2K}}}e^{-\frac{A^2}{2\sigma^2}-\frac{s^2}{2\sigma^2}}e^{\frac{A\sqrt{2K}}{\sigma}} \\
&= \frac{A}{\sigma^2\sqrt{\frac{2\pi}{\sigma}}\sqrt{A\sqrt{2K}}}e^{-\frac{A^2+s^2-2As}{2\sigma^2}} \\
&= \sqrt{\frac{A}{s}}\frac{1}{\sigma 2\pi}e^{-\frac{1}{2}\left(\frac{A-s}{\sigma}\right)^2} \qquad (3.13)
\end{aligned}$$

This expression differs only in the term $\sqrt{\frac{A}{s}}$ from the Gauss distribution:

$$f_{Gauss}(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{A-s}{\sigma}\right)^2} \qquad (3.14)$$

With $\sqrt{\frac{A}{s}} \approx 1$ the Rice distribution can be approximated by the Gauss distribution.

3.1.5 Noise, interference and spread-spectrum systems

Electromagnetic waves are transmitted from multiple sources at various frequencies. Signal components are reflected at buildings and other obstacles and are scattered in arbitrary directions. Therefore, a certain signal power is always present at a given frequency range. It is referred to as thermal noise. The exact noise value fluctuates heavily and depends on the encountered environment. In measurements taken in [64] a typical noise-power of

$$P_N = -103dBm \qquad (3.15)$$

was observed.

The thermal noise-power can also be estimated analytically as

$$P_N = \kappa \cdot T \cdot B, \qquad (3.16)$$

where $\kappa = 1.3807 \cdot 10^{-23}\frac{J}{K}$ is the Boltzmann constant, T is the temperature in Kalvin and B is the bandwith of the signal. For example, in a GSM system with $200kHz$ bands and an average temperature of $300K$, the noise-power is estimated to be $P_N = -120.82dBm$.

In networks in which various parties communicate over the same wireless channel neighbouring nodes cause interference as discussed in section 3.1.1. Electromagnetic waves that are received simultaneously are combined and accumulate. Figure 3.6 illustrates this property schematically for three distinct sinusoid signal-components.

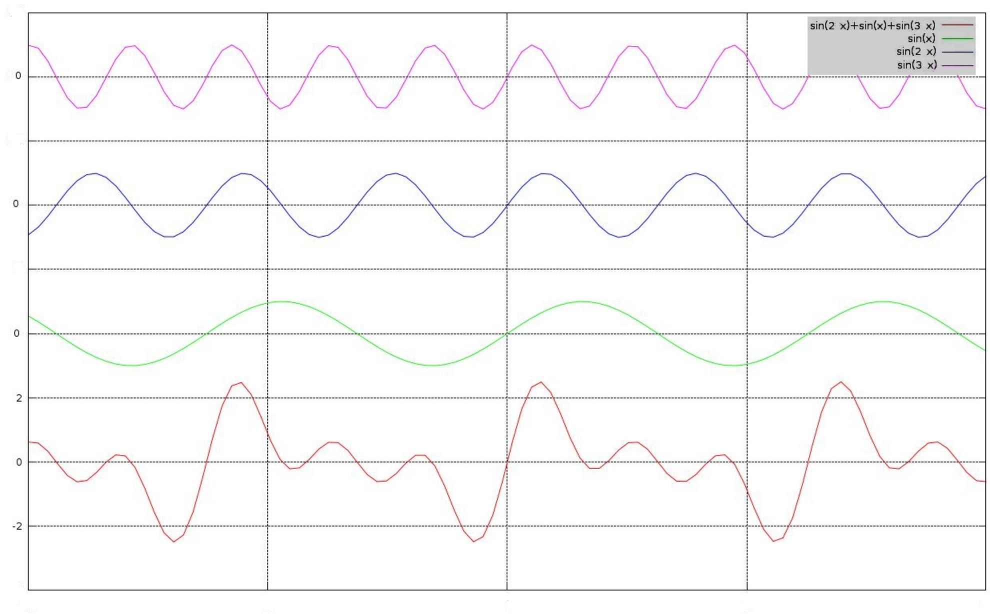


Figure 3.6: Superimposition of signal waves

For ι signal-components $\Re\left(e^{j(f_i t+\gamma_i)}\right)$ that are received simultaneously, the sum-signal is

$$\zeta_{\text{sum}} = \sum_{i=1}^{\iota} \Re\left(e^{j(f_i t+\gamma_i)}\right). \tag{3.17}$$

To cope with interference and noise, concepts like clustering or CDMA are introduced. A radio system typically requires a specific minimum signal-power beyond the interference and noise-level. This is referred to as the signal to interference and noise ratio (SINR) which is defined as

$$SINR = \frac{P_{\text{signal}}}{P_{\text{noise}} + P_{\text{interference}}} \tag{3.18}$$

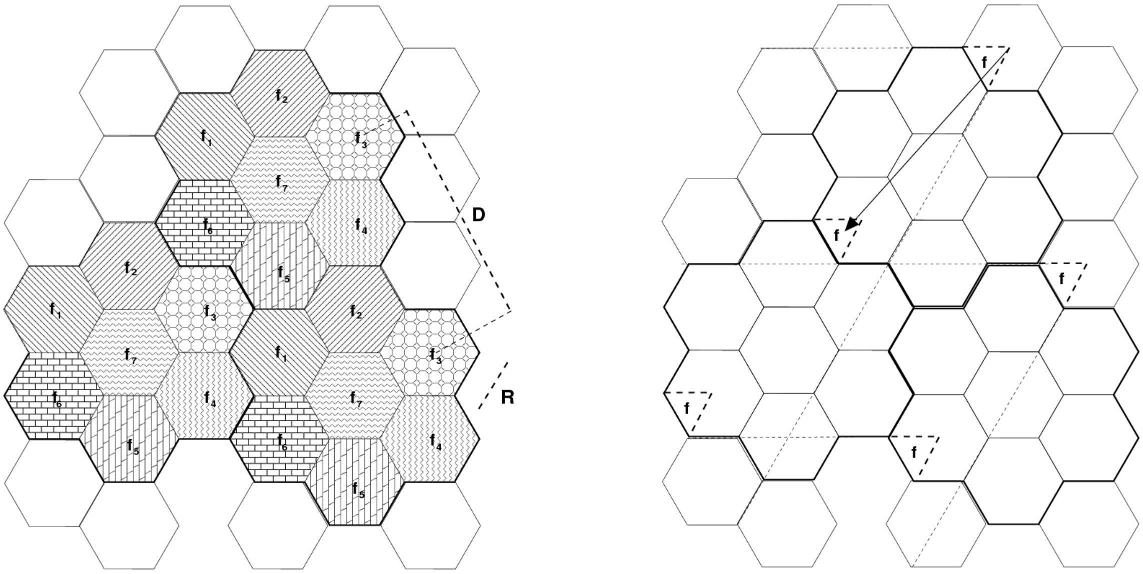
To reduce the inter-cell interference from cells that utilise the same frequency-range in cellular networks, cells can be clustered into groups of neighbouring cells that operate at different sub-frequencies within the whole available frequency band.

Figure 3.7(a) illustrates this procedure schematically with cells depicted as octagons.

Since cells with identical frequencies are separated at a considerable distance, the interference level is reduced as signal strength fades with transmission-distance. The concept of clustering is utilised in GSM instrumentation, for example. The number of interfering cells can be reduced further by utilising sectored antennas as depicted in figure 3.7(b)

Clustering is typically not implemented for sensor networks since relative locations of sensors are seldom known so that the organisation of a cluster-structure is not straightforward.

Instead, spread-spectrum systems are utilised in some sensor network instrumentation. The general idea of these techniques is to spread the spectrum utilised for transmission to



(a) schematic illustration of a cellular network

(b) Interfering cells with sectored antennas

Figure 3.7: Cellular network structures

reduce the impact and/or probability of interference for a transmission.

One possibility is to utilise frequency-hopping techniques like in current Bluetooth-networks. Bluetooth-devices work in the ISM band between 2.402 and 2.480 GHz. The utilised frequency-band is then divided into 79 sub-frequency-bands that are spaced by 1 MHz. A Bluetooth-device changes the transmission frequency at a high pace up to 1600 times per second.

The hop-sequence is defined in a pseudo-random manner by the hardware-address of a master node. When a new node wants to enter the network, it listens to the master and waits for the master-node to send inquiry access codes (IAC). On hearing such a message, the device sends a package containing its device-address and timing-information. Then the master node controls the hopping of the device. Figure 3.8 illustrates this communication-paradigm.

Due to this fast hopping-frequency interferences that affect only small frequency bands are reduced. However, the processing required for frequency-hopping schemes would typically surpass the processing capabilities of sensor nodes.

A spread-spectrum technique utilised in WSNs is code division (for instance, CDMA). In these approaches carrier-signals are transmitted over a very wide spectrum (in contrast to the frequency-hopping approach) so that interference in limited frequency-bands will not have great impact on the overall transmitted information. Additionally, various devices share the same frequency and transmit simultaneously. The overlayed received signal at a receiver is then extracted again and allocated to the distinct transmitters by utilisation of code division.

Basically, the transmit sequence is combined by a pseudo-noise sequence at the trans-

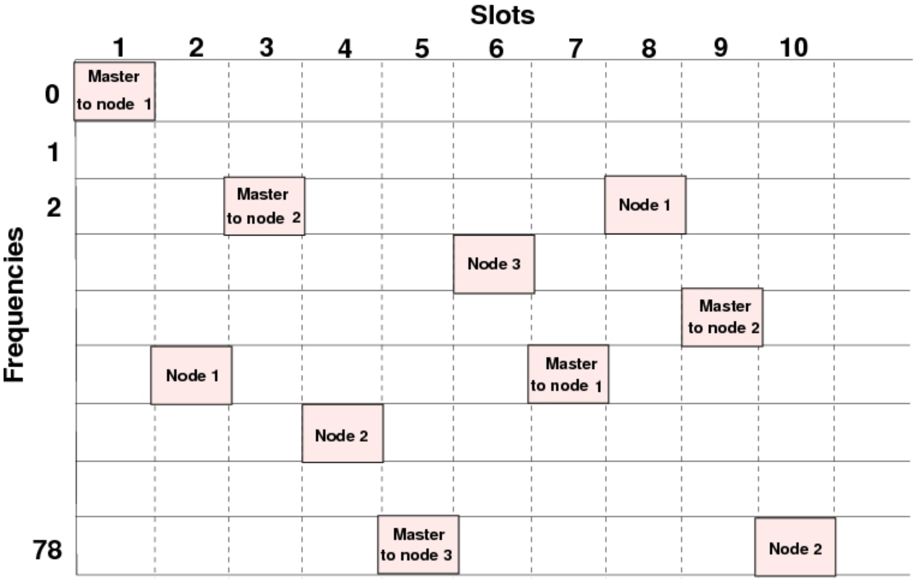


Figure 3.8: Frequency hopping communication in Bluetooth

mitter and then transmitted through the channel. Each transmitter has a unique pseudo-noise-sequence. At the receiver, the original signal is obtained by decoding the transmitted signal with the correct pseudo-noise sequence for a given transmitter.

Figure 3.9 illustrates the generation of a modified signal from the original signal through XOR combination with a code-sequence.

For CDMA systems code-sequences for distinct nodes are required to be orthogonal to each other. This means that it shall not be possible for one code sequence to occur in any other code sequence. Therefore, the number of code sequences of a given length is strictly limited. Such orthogonal codes are created by various algorithms. One example is Orthogonal Variable Spreading Factor (OVSF).

OVSF guarantees that spreading codes of different length remain orthogonal. Starting with a root spreading code $c_{i,j} \in \{0,1\}^i; i,j \in \mathbb{N}$, new spreading codes are created as $c_{2i,2j-1} = (c_{i,j}c_{i,j})$ and $c_{2i,2j} = (c_{i,j}\overline{c_{i,j}})$. In this creation rule, $\overline{x}$ is the binary complement of x. Figure 3.10 illustrates all OVSF codes with a length of 16, starting with $c_{1,1} = (1)$.

3.2 MIMO

Traditionally one transmitter and one receiver each with a single antenna are assumed for wireless communication. For such systems, capacity is increased by utilising diversity-schemes; for instance, time diversity (several nodes are scheduled in transmit and receive time slots), frequency diversity (several communication channels are implemented on various frequencies) or diversity by modulation with different random sequences such as CDMA.

Another possibility is to utilise spatial diversity. In section 3.1.5 a frequency and spatial

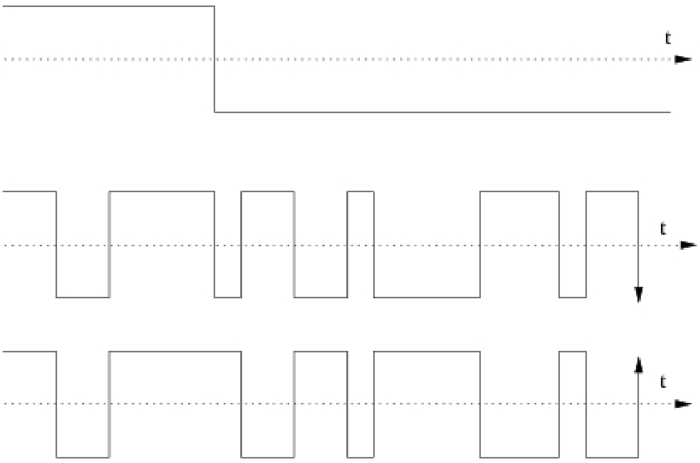


Figure 3.9: DS-CDMA scheme

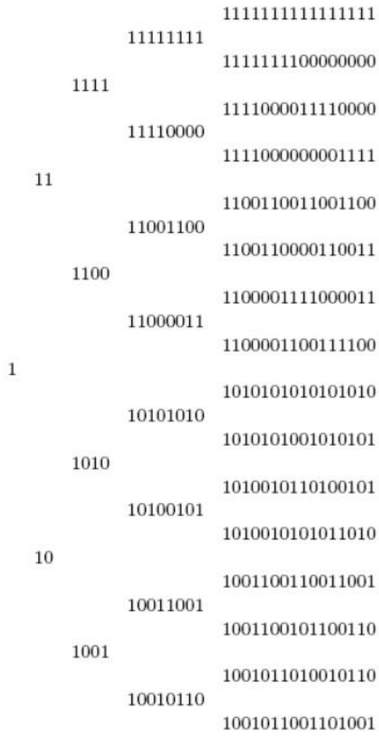


Figure 3.10: Orthogonal OVSF spreading codes of length 16 with $c_{1,1} = (1)$

diversity scheme was presented: Clustering.. In this scheme the same frequency is utilised in distinct cells that have a low interference with each other so that one area of the communication system is served by several base-stations in these cells. This is more energy-efficient than the alternative of having a single base-station since the radiated energy decreases at least quadratically (cf. section 3.1.2)

Another approach for utilising spatial diversity is by using multiple transmit antennas for a single communication-link that are separated spatially. When the distance between antennas is sufficiently large (a typical estimation is $\frac{\lambda}{2}$), the communication channels established by these antennas are considered independent. This means that for two distinct communication channels the quality of the channels fluctuates independently from each other. It follows that with a sufficient amount of distinct channels the probability of all channels experiencing inferior channel quality is low.

Communication channels are differentiated by the number of transmit and receive antennas. The traditional and straight forward implementation is a system with one transmit and one receive antenna, i.e. a single input single output system (SISO). In a SISO system, diversity is only utilised in the form of several independent parallel channels.

Other channel models are SIMO or MISO channels, where multiple transmitters but only one receiver or vice versa are utilised.

When multiple transmitters and receivers reside on both ends of the communication channels multiple transmitters/receivers reside MIMO systems are spoken of.

The transmission behaviour of a MIMO-System is characterised by the following Vector-Matrix:

$$\overrightarrow{\zeta^{RX}} = \begin{bmatrix} \zeta_1^{RX} \\ \zeta_2^{RX} \\ \vdots \\ \zeta_M^{RX} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1L} \\ h_{21} & \ddots & & h_{2L} \\ \vdots & & \ddots & \vdots \\ h_{M1} & h_{M2} & \cdots & h_{ML} \end{bmatrix} \begin{bmatrix} \zeta_1^{TX} \\ \zeta_2^{TX} \\ \vdots \\ \zeta_L^{TX} \end{bmatrix} + \begin{bmatrix} \zeta_1^{\text{noise}} \\ \zeta_2^{\text{noise}} \\ \vdots \\ \zeta_M^{\text{noise}} \end{bmatrix} \tag{3.19}$$

In this formula,

$$\overrightarrow{\zeta^{RX}} = (\zeta_1^{RX}, \zeta_2^{RX}, \ldots, \zeta_M^{RX})^T \tag{3.20}$$

is the vector of received signal-components ζ_i^{RX}. The receiver has M inputs and the transmitter L outputs. The channel matrix H defines how each input is connected to each output. The vector

$$\overrightarrow{\zeta^{\text{noise}}} = (\zeta_1^{\text{noise}}, \zeta_2^{\text{noise}}, \ldots, \zeta_M^{\text{noise}})^T \tag{3.21}$$

is the vector of noise signals that is added onto the individual transmit channels for each receiver input.

Typically the channel matrix can be altered only in particular boundaries e.g. by reordering of the antennas. It is, however, mainly dependent on the environmental situation of the wireless system.

These individual signal-components can be utilised to improve the overall quality of the communication channel. We can, for instance, improve the communication speed when symbols are transmitted over these spatially separated channels in parallel instead of sequential transmission in a SISO-system. Alternatively, identical symbols can be transmitted on all channels synchronised to improve the robustness against interference and signal fading [65].

3.3 Beamforming

The general idea of (centralised) beamforming in wireless communication systems is to create an antenna-beam that is focused on a restricted area. Intuitively, signal-components from various transmit antennas are coherently overlaid to create constructive interference in this area, while signal-components interfere constructively and destructively to create a general low noise-level outside. For a fixed antenna array in which the exact relative location of all antennas is known, the signals sent from each antenna element are suitably weighted to focus and controt the beam on a target location.

Identical symbols are transmitted from each antenna element. When these symbols from n such antenna elements are perfectly aligned at a receiver location, the SINR is amplified in the best case by factor n.

We can see this easily: When $y_i(t) = \Re(e^{j(2\pi f_i t + \gamma_i)})$ is the received signal-component of transmitter $i, i \in [1...n]$,

$$\sum_{i=1}^{n} \Re(e^{j(2\pi f_i t + \gamma_i)}) \tag{3.22}$$

defines the received sum-signal. Consequently, when all signal-components are in phase, the signal strength is increased linear to the number of signal-components. For centralised beamforming, all antenna elements are tightly synchronised so that the signal-beam can be controlled by the transmitter node.

4 Basics on probability theory

> *The generation of random numbers is too important to be left to chance.*
>
> (Robert R. Coveyou – Oak Ridge National Laboratory.)

The methods discussed in later chapters require some basic knowledge of the notion of probability. We can differentiate between probability-spaces with a finite or infinite number of events. For this lecture it suffices to consider only probability spaces with finite events. This section is designated to provide these basics on the theory of probability. Some of the examples and illustrative explanations are borrowed from [66, 67].

4.1 Discussion

Historically, probability theory was designed to describe phenomena observed in games of chance that could not be described by classical mathematical theory. A simple example is the description of the outcome of tossing a coin.

Today we find probability in many aspects of everyday life. An omnipresent example is the weather forecast. What we can learn from it is typically wether it will definitely rain or not, but that a certain probability of rain was derived from distinct measurements at various places all over the world and over a considerable amount of time. Although this information provides no guarantee that it will actually rain or will not, we have developed an intuitive understanding of probability. In this way the information is useful to us although it includes the chance of being incorrect. Other examples are insurance policies, where probability is used to calculate the probability of ruin, or also the lottery. The latter example is fascinating as people regularly ignore the infinitesimal probability to win a considerable amount of money.

A further instance where we regularly stumble across probability are quiz shows on TV. Consider, for instance, the following setting (see figure 4.1). A quiz master confronts a candidate with three doors A, B and C and explains that behind one of these the candidate will find (and win) a treasure, while the candidate will win nothing if he opens one of the other doors. The candidate then makes a decision for one of these doors, but before it is opened, the quiz master opens one of the remaining two doors and proves that this door does not hide the treasure. The candidate is now given the opportunity to rethink his decision and vote for the closed door he didn't vote for initially.

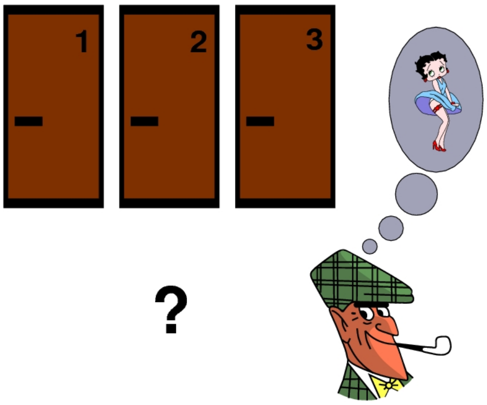


Figure 4.1: Which door hides the treasure?

What should the candidate do in order to maximise the probability to win the treasure? We can show that it is better to alter the initial choice since the remaining closed door contains the treasure with probability $\frac{2}{3}$.

4.2 Preliminaries

In order to calculate with probabilities we have to define an idealised model. We generally assume that probabilities can be verified by conceptual experiments. This means that the probability for an event represents the quotient of the number of occurrences of this event to the number of repetitions of the experiment provided that the experiment is repeated very often. An event with probability 0.4 should be expected to occur forty times out of one hundred in the long run. An often quoted example is the tossing of a coin. As an outcome of the experiment one of the two events (head or tail) are expected with an equal probability $\frac{1}{2}$. Note that this assumption is idealised in that we assume a 'fair' coin and disregard some possible but unlikely events such as the case of the coin falling on neither side but might roll away or stand on its edge, for example.

We have already introduced some of the typical notation. The results of experiments or observations are called events. Events are sets of sample-points. We denote events by capital letters. The fact that a sample-point x is contained in event E is denoted by $x \in E$. We write $E_1 = E_2$ when no point $x \in E_1 \wedge x \notin E_2$ or $x \in E_2 \wedge x \notin E_1$ exists. The sample-space of an experiment is the set of all possible events. The sample-space for the experiment of tossing a coin twice is *head-head, head-tail, tail-head, tail-tail.*

The following examples shall illustrate these concepts

Example 4.2.1 :

Consider the following experiment: Three distinct balls (a,b,c) are to be placed in three distinct bins. The following illustration depicts all possible outcomes of this experiment which form the sample-space together

Event Bin	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	abc			ab	ab	c		c		ac	ac	b		b
2		abc		c		ab	ab		c	b		ac	ac	
3			abc		c		c	ab	ab		b		b	ac

Event Bin	15	16	17	18	19	20	21	22	23	24	25	26	27
1		bc	bc	a		a		a	a	b	b	c	c
2	b	a		bc	bc		a	b	c	a	c	a	b
3	ac		a		a	bc	bc	c	b	c	a	b	a

However, the sample-space is altered when the conditions of the experiment are altered. Supposing the three balls are not distinguishable. Consequently the following sample-space belongs to this experiment.

Event Bin	1	2	3	4	5	6	7	8	9	10
1	***			**	**	*		*		*
2		***		*		**	**		*	*
3			***		*		*	**	**	*

When we also consider indistinguishable bins, the sample-space becomes

Event Bin	1	2	3
1	***	**	*
2		*	*
3			*

4.3 Relation between events

An arbitrary but fixed sample-space is assumed for the remainder of this lecture. The following definitions are essential for our notion of events.

Definition 4.3.1 : Impossible event

> *With $\chi = \{\}$ we denote the fact that event χ contains no sample-points. It is impossible to observe event χ as an outcome of the experiment.*

For every event χ there is an event $\neg\chi$ that is defined as 'χ does not occur'.

Definition 4.3.2 : Negation of events

> *The event consisting of all sample-points x with $x \notin \chi$ is the complementary event (or negation) of χ and is denoted by $\neg\chi$.*

For two events χ_1 and χ_2 we can also denote new events with the conditions that 'both χ_1 and χ_2 occur' or that 'either χ_1 or χ_2 occur'. These events are denoted by $\chi_1 \cap \chi_2$ and $\chi_1 \cup \chi_2$ respectively. These events are defined by

$$\chi_1 \cap \chi_2 = \{x | x \in \chi_1 \wedge x \in \chi_2\} \tag{4.1}$$

and

$$\chi_1 \cup \chi_2 = \{x | x \in \chi_1 \vee x \in \chi_2\} \tag{4.2}$$

and can be generalised to arbitrary many events. When the events χ_1 and χ_2 have no sample-point x in common, the event $\chi_1 \cap \chi_2$ is impossible: $\chi_1 \cap \chi_2 = \{\}$. Events χ_1 and χ_2 are referred to as mutually exclusive.

4.4 Basic definitions and rules

All events χ that might possibly occur are summarised in a sample-space.

Definition 4.4.3 : Sample space

> *A sample-space Π is a set of elementary events χ_i*

These events, together with an occurrence probability constitute the probability space.

Definition 4.4.4 : Probability space

> *A probability space (Π, P) consists of a sample-space Π and a probability measure $P : \Pi \rightarrow \mathbb{R}$. This function satisfies the following conditions*
>
> - *For each subset $\Pi' \subseteq \Pi, 0 \leq P(\Pi') \leq 1$*
> - $P(\Pi) = 1$
> - *For each $\Pi' \subseteq \Pi, P(\Pi') = \sum_{\chi \in \Pi'} P(\chi)$*

Given a sample-space Π and sample-points $x_i \in \Pi$, we denote the probability that x_i is observed by $P(x_i)$ with $P : \Pi \rightarrow [0, 1]$ and $P(x_1) + Px_2 + \cdots = 1$.

Definition 4.4.5 : Probability of events

> *Given a sample-space* Π *and an event* $\chi \in \Pi$*, the occurrence probability* $P(\chi)$ *of event* χ *is the sum of the probability of all sample-points from* χ*:*
>
> $$P(\chi) = \sum_{x \in \chi} P(x). \tag{4.3}$$

Since all probabilities of the sample-space sum up to 1 it follows that

$$0 \leq P(\chi) \leq 1 \tag{4.4}$$

for any event χ.

Consider two arbitrary events χ_1 and χ_2. In order to compute the probability $P(\chi_1 \cup \chi_2)$ that either χ_1 or χ_2 or both occur we add the occurrence probabilities that a sample-point either in χ_1 or in χ_2 is observed.

$$P(\chi_1 \cup \chi_2) \leq P(\chi_1) + P(\chi_2) \tag{4.5}$$

The '$\leq$'-relation is correct since sample-points might be contained in both events. Thereby we obtain the exact probability by

$$P(\chi_1 \cup \chi_2) = P(\chi_1) + P(\chi_2) - P(\chi_1 \cap \chi_2). \tag{4.6}$$

Example 4.4.2 :

> *A coin is tossed twice so that the sample-space contains the four sample-points head-head, head-tail, tail-head and tail-tail that are associated with probability* $\frac{1}{4}$ *each. Consider the two events* χ_1 *– head occurs first – and* χ_2 *– tail occurs second.* χ_1 *then contains head-head and head-tail while* χ_2 *contains head-tail and tail-tail. Consequently,* $\chi_1 \cup \chi_2$ *consists of the sample-points head-head, head-tail, tail-tail while* $\chi_1 \cap \chi_2$ *consists of the single sample-point head-tail. We therefore obtain the probability that either* χ_1 *or* χ_2 *occurs as*
>
> $$P(\chi_1 \cup \chi_2) = \frac{1}{2} + \frac{1}{2} - \frac{1}{4}. \tag{4.7}$$

This can be generalised to higher event counts. For arbitrary events $\chi_1, \chi_2, \dots$ this is expressed by the inequality

$$P(\chi_1 \cup \chi_2 \cup \dots) \leq P(\chi_1) + P(\chi_2) + \dots - P(\chi_1 \cap \chi_2) - P(\chi_1 \cap \chi_3) \dots \tag{4.8}$$

In the special case that all events $\chi_1, \chi_2, \dots$ are mutually exclusive, we obtain

$$P(\chi_1 \cup \chi_2 \cup \dots) = P(\chi_1) + P(\chi_1) + \dots \tag{4.9}$$

since $P(\chi_i) \cap P(\chi_j) = \{\}$ for any two mutually exclusive events χ_i and χ_j.

In some cases the probability of a specific event occurring in the presence of another event is of interest. This can be expressed by the conditional probability.

Definition 4.4.6 : Conditional probability

> *The conditional probability of two events χ_1 and χ_2 with $P(\chi_2) > 0$ is denoted by $P(\chi_1|\chi_2)$ and is calculated by*
>
> $$\frac{P(\chi_1 \cap \chi_2)}{P(\chi_2)} \tag{4.10}$$

$P(\chi_1|\chi_2)$ describes the probability that event χ_1 occurs in the presence of event χ_2 (read: The probability of χ_1 given χ_2). With rewriting and some simple algebra we obtain the Bayes rule that states

$$P(\chi_1|\chi_2) = \frac{P(\chi_2|\chi_1) \cdot P(\chi_1)}{\sum_i P(\chi_2|\chi_i) \cdot P(\chi_i)}. \tag{4.11}$$

This equation is useful in many statistical applications. Note that the denominator is the sum of all probabilities for all possible events. This means that everything on the right-hand side of the equation is conditioned on the events χ_i. When we say that χ_i is the important variable, the shape of the distribution $P(\chi_1|\chi_2)$ depends on the numerator $P(\chi_2|\chi_1) \cdot P(\chi_1)$ with the denominator as a normalising factor that ensures that the $P(\chi_1|\chi_2)$ sum to 1. The Bayes rule then is interpreted in such a way that it inverts $P(\chi_1|\chi_2)$ to $P(\chi_2|\chi_1)$. This is useful when it is easy to calculate $P(\chi_2|\chi_1)$ but not to calculate $P(\chi_1|\chi_2)$. With the Bayes rule it is easy to calculate $P(\chi_1|\chi_2)$ then, provided that we know $P(\chi_2|\chi_1)$ and $P(\chi_1)$.

Definition 4.4.7 : Independence

> *A collection of events χ_i that form the sample-space Π is independent if for all subsets $S \subseteq \Pi$*
>
> $$P\left(\bigcap_{\chi_i \in S} \chi_i\right) = \prod_{\chi_i \in S} P(\chi_i). \tag{4.12}$$

Statistical independence is required for many useful results in probability theory. This means, on the other hand, that we have to be careful not to apply such results in cases where independence between sample-points is not provided.

Definition 4.4.8 : Expectation

> *The expectation of an event χ is defined as*
>
> $$E[\chi] = \sum_{x \in \mathbb{R}} x \cdot P(\chi = x) \tag{4.13}$$

Although the expectation of an event represents the expected outcome of the event intuitively, the expectation is not necessary equal to one of the possible sample-points.

Consider, for instance, the event χ of throwing a dice. The Sample space is given by $S_\chi = \{1, 2, 3, 4, 5, 6\}$. However, the expectation of this event is

$$E[\chi] = \frac{1}{6} \cdot (1 + 2 + 3 + 4 + 5 + 6) = 3.5. \tag{4.14}$$

It is also possible to perform calculations with expectations of events.

Definition 4.4.9 : Linearity of expectation

For any two random variables χ_1 and χ_2,

$$E[\chi_1 + \chi_2] = E[\chi_1] + E[\chi_2]. \tag{4.15}$$

For independent random variables χ_1 and χ_2,

$$E[\chi_1 \cdot \chi_2] = E[\chi_1] \cdot E[\chi_2]. \tag{4.16}$$

Definition 4.4.10 : Variance

The variance of a random variable χ is defined as

$$var[\chi] = E[(\chi - E[\chi])^2]. \tag{4.17}$$

For any random variable χ

$$var[\chi] = E[\chi^2] - E[\chi]^2 \tag{4.18}$$

For any independent random variables χ_1 and χ_2

$$var[\chi_1 + \chi_2] = var[\chi_1] + var[\chi_2]. \tag{4.19}$$

For any random variable χ and any $c \in \mathbb{R}$,

$$var[c\chi] = c^2 var[\chi]. \tag{4.20}$$

4.4.1 The Markov inequality

To estimate the deviation of a random variable from its expectation, the following bound is useful.

Definition 4.4.11 : Markov inequality

Let (Π, P) be a probability space and $x : \Pi \rightarrow \mathbb{R}^+$ a non-negative random variable. For $t \in \mathbb{R}^*$ the following inequality holds:

$$P(x \geq t \cdot E[x]) \leq \frac{1}{t} \tag{4.21}$$

Proof. For an arbitrary $s \in \mathbb{R}^+$ we define an indicator variable $Y_s : \Omega \to \{0, 1\}$ with

$$Y_s := \begin{cases} 1 & \text{if } x \geq s \\ 0 & \text{else} \end{cases} \tag{4.22}$$

We have $x \geq s \cdot Y_s$ for all s. Therefore we obtain

$$\begin{aligned} E[x] &\geq E[s \cdot Y_s] \\ &= s \cdot E[Y_s] \\ &= s \cdot 1 \cdot P(Y_s = 1) + 0 \cdot P(Y_s = 0) \\ &= s \cdot P(x \geq s) \end{aligned} \tag{4.23}$$

Observe that we have shown $P(x \geq s) \leq \frac{E[x]}{s}$. When we choose $s := t \cdot E[x]$ the assertion follows. □

4.4.2 The Chernoff bound

A stronger bound useful to estimate the deviation of a random variable from its mean is the Chernoff bound.

Definition 4.4.12 : Chernoff bound

> *Let* (Π, P) *be a probability space and* $x_1, x_2, \ldots, x_n : \Pi \to \{0, 1\}$ *independent random variables with* $0 < P(x_i = 1) < 1$ *for all* $i \in \{1, 2, \ldots, n\}$. *For* $X := \sum_{1 \leq i \leq n} x_i$ *and* $\delta > 0$ *the following inequality holds:*
>
> $$P(X < (1 + \delta)E[X]) < \left(\frac{e^\delta}{(1 + \delta)^{1+\delta}}\right)^{E[X]} \tag{4.24}$$
>
> *and for all* δ *with* $0 < \delta < 1$
>
> $$P(X < (1 - \delta)E[X]) < e^{-\frac{E[X]\delta^2}{2}} \tag{4.25}$$

Proof. First we prove the former estimation. Basically we apply the Markov inequality. From there we arrive at a much stronger result by utilising the random variable $e^{\varsigma X}$ for a suitable constant $k\varsigma$ instead of utilising X. Observe that

$$P(X < (1 + \delta)E[X]) = P(e^{\varsigma X} < e^{\varsigma(1+\delta)E[X]}) \tag{4.26}$$

holds for all $\varsigma \in \mathbb{R}^+$. We now apply the Markov inequality and obtain

$$P(X > (1 + \delta)E[X]) < \frac{E[e^{\varsigma X}]}{e^{\varsigma(1+\delta)E[X]}} \tag{4.27}$$

We utilise the definition of X and can therefore write

$$E[e^{\varsigma X}] = E\left[e^{\varsigma \sum_{i=1}^n x_i}\right] = E\left[\prod_{i=1}^n e^{\varsigma x_i}\right] \tag{4.28}$$

Following the precondition, the random variables x_i are independent and therefore

$$E\left[\prod_{i=1}^{n} e^{\varsigma x_i}\right] = \prod_{i=1}^{n} E\left[e^{\varsigma x_i}\right] \tag{4.29}$$

Altogether we have now

$$P(X > (1+\delta)E[X]) < \frac{\prod_{i=1}^{n} E\left[e^{\varsigma x_i}\right]}{e^{\varsigma(1+\delta)E[X]}} \tag{4.30}$$

Since X are binary random variables we obtain

$$\begin{aligned} E\left[e^{\varsigma x_i}\right] &= P(x_i = 1)e^{\varsigma \cdot 1} + (1 - P(x_i = 1))e^{\varsigma \cdot 0} \\ &= 1 + P(x_i = 1)(e^{\varsigma} - 1) \end{aligned} \tag{4.31}$$

We estimate $e^x \geq 1 + x$ with $x := P(x_i = 1)(e^{\varsigma} - 1)$ to obtain

$$\begin{aligned} P(X > (1+\delta)E[X]) &< \frac{\prod_{i=1}^{n} e^{P(x_i=1)(e^{\varsigma}-1)}}{e^{\varsigma(1+\delta)E[X]}} \\ &= \frac{e^{(e^{\varsigma}-1)\sum_{i=1}^{n} P(x_i=1)}}{e^{\varsigma(1+\delta)E[X]}} \\ &= \frac{e^{(e^{\varsigma}-1)E[X]}}{e^{\varsigma(1+\delta)E[X]}} \end{aligned} \tag{4.32}$$

Since $\varsigma \in \mathbb{R}^+$ and $\delta > 0$ we choose $\varsigma := \ln(1+\delta)$ and obtain

$$P(X > (1+\delta)E[X]) < \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{E[X]} \tag{4.33}$$

as claimed.

To prove the second estimation we proceed similarly. It is for all $\varsigma \in \mathbb{R}^+$

$$\begin{aligned} P(X < (1-\delta)E[X]) &= P(-X > -(1-\delta)E[X]) \\ &= P(e^{-\varsigma X} > e^{-(1-\delta)E[X]}) \end{aligned} \tag{4.34}$$

With the Markov inequality the following is obtained

$$P(X < (1-\delta)E[X]) < \frac{\prod_{i=1}^{n} E\left[e^{-\varsigma x_i}\right]}{e^{-\varsigma(1-\delta)E[X]}} \tag{4.35}$$

With $E\left[e^{-\varsigma x_i}\right] = P(x_i = 1)e^{-\varsigma} + 1 - P(x_i = 1)$ we obtain

$$P(X < (1-\delta)E[X]) < \frac{e^{E[X]\left(e^{-\varsigma}-1\right)}}{e^{-\varsigma(1-\delta)E[X]}} \tag{4.36}$$

for all $\varsigma \in \mathbb{R}+$. We choose $\varsigma = \ln\left(\frac{1}{1-\delta}\right)$ and obtain

$$P(X < (1-\delta)E[X]) < \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{E[X]} \tag{4.37}$$

Additionally we can assume $0 < \delta \leq 1$ and obtain $(1-\delta)^{1-\delta} > e^{-\delta+\frac{\delta^2}{2}}$ and obtain

$$P(X < (1-\delta)E[X]) < e^{-E[X]\frac{\delta^2}{2}} \tag{4.38}$$

as claimed. □

5 Evolutionary algorithms

> *The theory of evolution by cumulative natural selection is the only theory we know of that is in principle capable of explaining the existence of organised complexity.*
>
> (Richard Dawkins)

Evolutionary algorithms have been developed in the 1960s. At that time the primary aim was to model phenomena observed in evolution of nature and to use these models to solve algorithmic problems [68]. The two main operators to guide the optimisation process are mutation and crossover. While crossover combines distinct search-points to generate a new one, mutation is designed to apply small changes to a single search-point. We can distinguish four main branches.

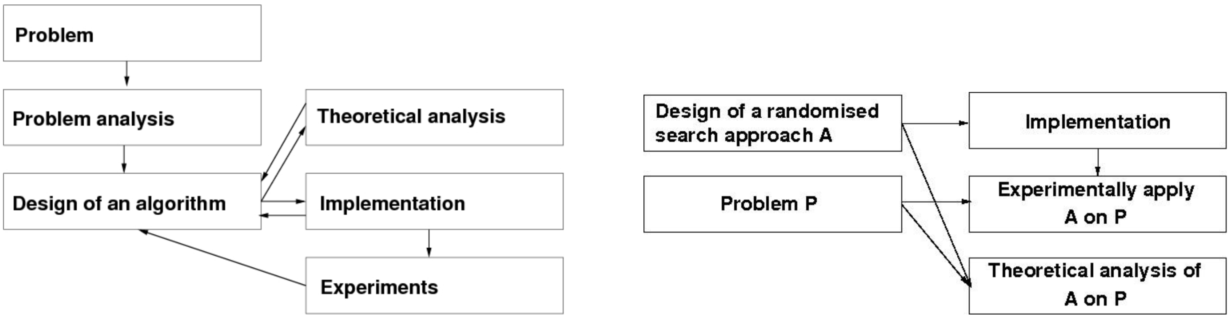
Genetic algorithm (GA): John Holland developed genetic algorithms [69] that operate on the search-space $\mathbb{B}^n$ and mainly utilise k-point-crossover.

Evolution strategy (ES): Evolution strategies have been developed by Hans-Paul Schwefel and Ingo Rechenberg [70]. These strategies are applied first in the search-space $\mathbb{R}^n$ and mainly utilised mutation as recombination operator.

Evolutionary programming (EP): Evolutionary programming has been developed by Larry Fogel [71]. Fogel considers the space of finite automata as search-space and mainly utilises mutation as search operator.

Genetic programming (GP): Finally, John Koza developed genetic programming [72]. For this approach, the search-space is mostly represented by a set of graphs that represent logical expressions or programs.

For several years now these approaches are less clearly separated from each other so that we speak of evolutionary approaches or evolutionary algorithms. Evolutionary algorithms operate on a search-space S. All points in this search-space are associated with a fitness-value that is provided by a fitness-function f. For the optimisation and application in practical settings usually the time until a search-point is found that reaches or exceeds a given fitness threshold or at least that could not been improved for some while is crucial. An evolutionary algorithm in a population of search-points maintains a set of μ search-points



(a) The classical approach to developing an algorithm

(b) The development and application of evolutionary algorithms

Figure 5.1: Approaches to algorithmic development: A comparison

(individuals) from the search-space S. It iterates an ever identical sequence of operations (generations) until a stop criteria is reached. In every generation all members of the populations are evaluated initially with a non-monotonously-decreasing fitness-function $f : S \rightarrow \mathbb{R}$. In a random process, individuals are elected for mutation or crossover. This leads to a set of ν individuals of an offspring population. Depending on the strategy implemented, either μ individuals with the highest fitness score of all $\mu + \nu$ individuals ($(\mu + \nu)$-strategy) or the μ individuals with highest fitness-value are picked from the offspring population of size ν ((μ, ν)-strategy).

The research field of evolutionary algorithms has also been studied in the scope of theoretical computer science in recent years [73, 74, 75, 76]. Most relevant in our context are the results on $(1 + 1)$-strategies, [77, 78, 79, 80, 81]. We will see in chapter 7 that this is the natural scenario for distributed adaptive transmit beamforming in wireless sensor networks.

5.1 Basic principle and notations

Evolutionary algorithms differ from classical algorithms firstly in the way they are developed. In the classical approach a problem is the origin of the development or optimisation of an algorithm. The problem is analysed and, based on this analysis an algorithm for this very problem is developed. After the algorithm has been designed a theoretical analysis might help to improve the algorithmic design. Also, the algorithm might be implemented and applied to experimental data or in problem-scenarios. In both cases, the experience gained during the implementation or application of the algorithm might affect the design of the algorithm (cf. figure 5.1(a)).

For evolutionary algorithms, the procedure is different. There the design and optimisation of the algorithm is independent from the concrete problem. When the algorithm is developed, it is applied to one or many distinct problems. These problems might have been existent beforehand and it is not proven that the evolutionary algorithm is especially well-suited to solve these problems. So the question arises, why should this be a reason-

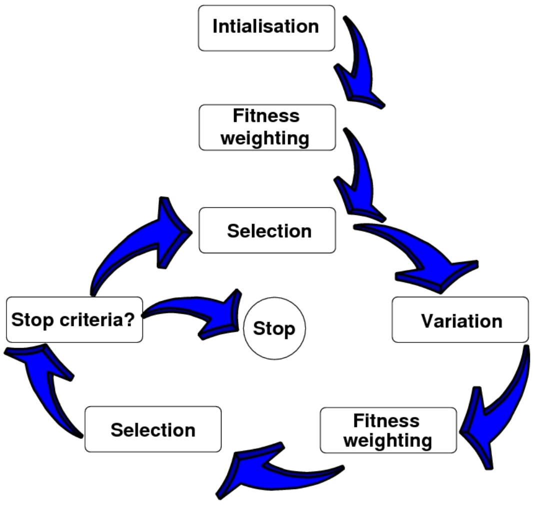


Figure 5.2: Modules of an evolutionary algorithms

able approach. Normally, the problems to which evolutionary algorithms are applied to are not understood well or too complex to describe them analytically. Therefore it might be very complex or even impossible to state a deterministic algorithm that is able to achieve sufficient results on the given problem. Evolutionary algorithms are designed for specific search-spaces and might inherit properties that enable them to cope with these search-spaces. A theoretical analysis of evolutionary algorithms therefore is commonly applied for a given algorithm on a specific problem. In turn, these results might impact the design and optimisation of the algorithm, but this regression is not always applied so that the algorithm might be taken as an as-is implementation that is applied to arbitrary problems (cf. figure 5.1(b)).

Evolutionary algorithms are random search heuristics that operate in ever identical iterations. Figure 5.2 depicts the various modules/phases that constitute an evolutionary algorithm.

5.1.1 Initialisation

Before the start of the optimisation the population of the algorithm is to be initialised. Typically, the μ individuals from the search-space S that constitute the initial population are drawn uniformly at random. Typical search-spaces are $S = \mathbb{R}^n$ or $S = \mathbb{B}^n$. In order to achieve sufficient coverage of the search-space, we can also define a distance measure d and populate the initial population with points that are at least at a distance d to each other. Also, in order to improve the optimisation-time and the quality of the solution, fast heuristics can be utilised to find the individual population.

5.1.2 Fitness-function – Weighting of the population

At the start of the optimisation the individuals of the population are weighted for their fitness-value. For each individual in the population, a fitness-function $f : S \to \mathbb{R}$ is applied. The fitness-function is a monotonous function that maps individuals from the search-space onto $\mathbb{R}$.

5.1.3 Selection for reproduction

Dependent on the fitness-values reached by the individuals in the population, individuals are chosen to produce the offspring population. The underlying assumption is that individuals with a good fitness-value have a higher probability of producing individuals for the offspring population - which in turn also inherit a higher fitness-value - than individuals with lower fitness-values. Typical selection mechanisms are

Uniform selection: An individual is chosen uniformly at random from all individuals in the population.

Fitnessproportional selection: An individual $\mathcal{I}$ is chosen from the population $\mathcal{P}$ with probability $\frac{f(\mathcal{I})}{\sum_{x \in \mathcal{P}} f(x)}$. A problem with this scheme is that the selection behaviour is considerably different for two fitness-functions f and $f + c$. Furthermore, when fitness-values are sufficiently separated, the selection is nearly deterministic while it is very similar to the uniform selection when the differences between fitness-values are small relative to their absolute values.

Tournament selection: The intuition for this selection method is to find the best individuals from a randomly drawn subset of the whole population. This selection method also allows individuals with non-optimal fitness-values to be chosen for the variation.

Apart from these approaches it is also possible, of course, to choose the k highest rated individuals for the selection deterministically.

5.1.4 Variation

For the selected individuals, the offspring population is created by mutation and/or crossover. While typically mutation is a local search operator, crossover allows finding search-points in regions of the search-space that are currently not populated. Dependent on the stage/ progress of the optimisation, some implementation adapt the probability to apply crossover and/or mutation.

Mutation

Mutation alters the representation of an individual, e.g. by flipping of bits for individuals from $\mathbb{B}^n$. The underlying assumption for the mutation is that it produces individuals for the offspring-population that differ only slightly from the parent-individuals. For the mutation operator, one parent-individual produces one offspring-individual. It is clear that possible mutation operators differ for different search-spaces.

Mutation operators for individuals from $\mathbb{B}^n$:

Standard bit mutation: The offspring-individual is created bit-wise from the parent individual. Every bit is 'flipped' with probability p_m and sustained with probability $(1 - p_m)$. A common choice is $p_m = \frac{1}{n}$ as this represents the case that one bit is flipped in one mutation on average on average.

1 bit mutation: The offspring individual is identical to the parent individual in all but one bit. This bit is chosen uniformly at random from all n bits in the individual representation.

Mutation operators for $\mathbb{R}^n$: An offspring individual is, in general, generated by adding a vector $m \in \mathbb{R}^n$ to the parent individual. For this operation we can distinguish between restricted or unrestricted mutation. For restricted mutation the vector is chosen form a restricted interval (e.g. all vector entries are chosen from $[-a, a]$).

Crossover

Crossover is an alternative technique that produces one or more offspring individuals from two or more parent individuals.

Crossover operators for $\mathbb{B}^n$:

k-point crossover: From the possible n positions in one individual representation, $k \leq n$ positions are chosen uniformly at random. The offspring individuals are created such that the representation is concatenated from sub-sequences of the parent individuals when the origin of the sub-sequence is altered at every one of the k chosen positions. For two parent individuals x_1, x_2 and 2-point crossover with k_1, k_2, an offspring individual y can be created as follows:

$$\begin{array}{rl}
x_1 = & x_{11}, x_{1,2}, \ldots, x_{1,k_1} | x_{1k_1+1}, \ldots, x_{1k_2} | x_{1k_2+1}, \ldots, x_{1n} \\
x_2 = & x_{21}, x_{2,2}, \ldots, x_{2,k_1} | x_{2k_1+1}, \ldots, x_{2k_2} | x_{2k_2+1}, \ldots, x_{2n} \\
\hline
y_1 = & x_{11}, x_{1,2}, \ldots, x_{1,k_1} | x_{2k_1+1}, \ldots, x_{2k_2} | x_{1k_2+1}, \ldots, x_{1n} \\
y_2 = & x_{21}, x_{2,2}, \ldots, x_{2,k_1} | x_{1k_1+1}, \ldots, x_{1k_2} | x_{2k_2+1}, \ldots, x_{2n}
\end{array}$$

Uniform crossover: Each bit in the individual representation is chosen with uniform probability from one of the parent individuals.

Crossover operators for $\mathbb{R}^n$:

k-point crossover: Analogous to k-point crossover in $\mathbb{B}^n$

Uniform crossover: Analogous to uniform crossover in $\mathbb{B}^n$

Arithmetic crossover: An individual $\mathcal{I} \in \mathbb{R}^n$ for the offspring population is created as a weighted sum from k parents $x_1, \ldots, x_k$:

$$\mathcal{I} = \sum_{i=1}^{k} \alpha_i x_i; \text{ with } \sum_{i=1}^{k} \alpha_i = 1 \tag{5.1}$$

5.1.5 Fitness-function – Weighting of the offspring population

After the variation, all newly generated offspring individuals are weighted by a fitness-function f.

The structure of this fitness-function also impacts the performance of random search approaches applied to the problem. A fitness-function with many local optima, for example, may require that the search approach also considers search-points in greater distance to the current best-rated search-point in order to avoid that the algorithm converges in a local optimum while the global optimum may still be far away.

Definition 5.1.1 : Optima

Let $f : G \rightarrow P$ be a real valued fitness-function. $x^* \in G$ is an optimum point of for $\varepsilon > 0$ with $|x - x^*| < \varepsilon$ the inequality $f(x^*) \geq f(x)$ ($f(x^*) \leq f(x)$) holds.

Global optimum *An optimum point x^* is called global optimum, if $f(x^*) \geq f(x)$ ($f(x^*) \leq f(x)$) for all $x \in G$.*

Local optimum *An optimum point which is not globally optimal is called local optimum.*

We distinguish between multimodal and unimodal functions dependent on the count of optima.

Definition 5.1.2 : Multimodality and Unimodality

A function f is called unimodal when only one global optimum exists. Otherwise it is called multimodal.
An unimodal or multimodal function f with no local optima is called strong multimodal (unimodal). Otherwise it is called weak multimodal (unimodal).

Figure 5.3 depicts an exemplary weak multimodal function.

5.1.6 Selection for substitution

Since the number of search-points has increased due to the variation, we have to reduce the population size to μ again for the next iteration. Commonly, individuals with higher fitness-values have a higher probability of staying in the population. Sometimes, also the age of individuals is taken into account so that older individuals have a higher probability

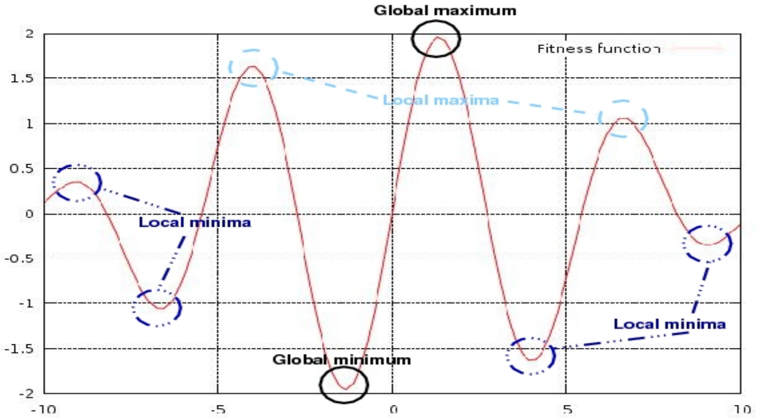


Figure 5.3: Illustration of an exemplary function that is weak multimodal

of being replaced. The selection operators are basically the same as for the selection for reproduction.

Regarding the pool from which the individuals are drawn, we distinguish between plus and comma strategies. In $(\mu + \nu)$ strategies, the offspring population is chosen from the μ individuals of the old population 'plus' the ν offspring individuals created in the variation phase.

In (μ, ν) strategies, the μ individuals for the next population are drawn from the ν individuals created in the variation phase while the μ individuals of the old population are discarded.

5.2 Restrictions of evolutionary algorithms

In the early days of evolutionary algorithms, assumptions were formulated that these algorithms are superior to classical algorithms in the sense that the performance of evolutionary algorithms is higher on average for all possible problems although a specialised algorithm may outperform an evolutionary algorithm on a small set of problems. Figure 5.4 illustrates this intuition.

After contemplating this assumption it can be easily seen that it is somewhat absurd or at least not sufficiently concrete. The question wether one algorithm can be applied to all problems must be asked. Clearly, the input format might differ among problems. Can an algorithm suited to solve the travelling salesman-problem also be applied to the task of recognising patterns in image databases or to the task of computing prime factors of an integer?

It can be seen that we have to be more concrete in order to qualify the assumption illustrated in figure 5.4.

We consider algorithms that operate on the search-space S which is mapped onto an ordered set W by a fitness-function $f : S \rightarrow W$ from the set F of all possible fitness-functions. We assume that both, S and W are finite. This is a reasonable assumption, as

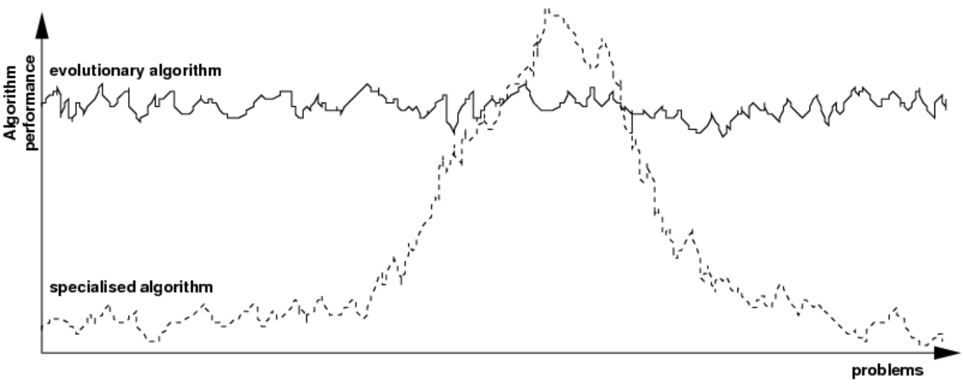


Figure 5.4: Illustration of the assumption that evolutionary algorithms have a better average performance than classical algorithms

digital computers, which are utilised to host the algorithms can only store finite sets and address a finite number of points.

The assumption is that no search-point is requested twice. This is a simplifying assumption for technical reasons in favour of randomised search approaches as especially random approaches might request a search-point more than once. This means that this assumption does not impact the performance of evolutionary algorithms negatively.

For $f \in F$ and an algorithm A we denote the number of search-points requested until an optimum is found by this algorithm with $A(f)$.

Theorem 5.2.3 : No free lunch theorem (NFL)

For two algorithms A and A' the relation $A_{S,W} = A'_{S,W}$ holds.

Proof. We show this assertion by induction over the size of the search-space: $s := |S|$. We consider all functions $F_{s,i,N}$ on a search-space S with $|S| = s$ and a fitness domain of $\{1, \dots, N\}$ with at least one $x \in S$ with $f(x) > i$. Initially, we have the situation $F_{s,0,N}$. It can be observed that, since we are talking about all possible functions, i.e. all possible mappings from S to W , it is not required to consider all search-spaces of size $|S| = s$. Since another search-space is just another permutation of S on W this would only reorder the functions f. For each function f and each permutation π on S also $f_\pi(s) = f(\pi(s))$ is a member of $F_{s,i,N}$.

We denote the average performance of a search strategy A in $F_{s,i,N}$ with $A_{s,i,N}$ and claim $A_{s,i,N} = A'_{s,i,N}$ for arbitrary search strategies A and A' and $s \in \mathbb{N}, N \in \mathbb{N}, 0 \leq i < N$

Induction start: $s = 1$. Each algorithm has to request the only search-point and finds the optimum with the first request.

Induction step: $s - 1 \rightarrow s$. Since for every f and permutations π also f_π belongs to $F_{s,i,N}$, a function $a : S \rightarrow \mathbb{N}$ exists, so that the share of functions with $f(x) = j$ is $a(j)$ and thus $a(j)$ is the probability to choose a search-point with the fitness-value j. With probability $a(j)$, every algorithm observes a search-point $x \in S$ with $f(x) = j$.

If $j \leq i$, x is not optimal and the new scenario is $F_{s-1,i,N}$. For each $f' \in F_{s-1,i,N}$ with $S' = \{1, \ldots, s-1\}$ exactly one function $f \in F_{s,i,N}$ exists with $f(x)$ in $f'(x)$ for $x \in S'$ and $f(s) = j$. Therefore, all functions from $F_{s-1,i,N}$ are equally probable.

Let $j > i$. Since with f also f_π belongs to $F_{s,i,N}$, the number of functions with $f(x) = j$ and x optimal is equal to the number of functions f' with $f'(x) = j$. Consequently, the probability to find an optimal point is not dependent on the algorithm in this scenario.

□

5.3 Design aspects

In the following sections some general design aspects for the implementation of evolutionary algorithms will be discussed.

5.3.1 Search-space

For problems that do not have their origin in computer science, the problem has to be represented as a search problem and, in particular, the representation of individuals and the search-space have to be designed.

It is often sufficient to represent search-points as vectors of a constant length for a given set (e.g. $\mathbb{R}, \mathbb{Z}, \mathbb{N}, \mathbb{B}$). The decision taken on the modelling of the search-space also impacts the mutation and crossover operators applicable and therefore can impact the optimisation performance.

Generally, it is favourable to choose an individual representation that reflects the problem characteristics well. Otherwise the representation might change the actual problem and possibly increase its complexity. Assume, for example, that numbers in their binary representation are to be encoded. The hamming-distance between 2^n and $2^n + 1$ is 1 while the hamming-distance between 2^n and $2^n - 1$ is $n + 1$! A solution to this problem might be the utilisation of Gray codes. A Gray code is defined recursively. The numbers 0 and 1 are encoded as 0 and 1. When the numbers $0, \ldots, 2^n - 1$ are encoded by $a_0, \ldots, a_{2^n-1}$ we encode $0, \ldots, 2^{n+1} - 1$ as $0a_0, \ldots, 0a_{2^n-1}, 1a_{2^n-1}, \ldots, 1a_0$. Obviously, the distance of neighbouring numbers is 1. However, also higher numbers have a smaller distance: $\text{hammingDist}(0a_0, 1a_0) = 1$.

5.3.2 Selection principles and population structure

Selection is an important method for evolutionary algorithms. Without selection, the choice of search-points would be unguided and thus merely random. Of course, it is intended to keep individuals with good fitness scores but this also contains the danger that the population might degenerate to an isolated region in the search-space that possibly constitutes only a local optimum.

To circumvent this problem we can consider further measures in the calculation of the selection probability such as the distance of search-points or the count of points in a defined neighbourhood.

In order to sustain a diverse population that is spread widely across the search-space, we strive to keep isolated individuals with respectable fitness score. On the other hand, keeping many similar individuals with similar fitness values is not desirable.

In order to achieve this aim we can replace the fitness score $f(x)$ of an individual x by $f'(x) = \frac{f(x)}{d(x,\mathcal{P})}$ where $d(x,\mathcal{P})$ measures the singularity of x in a population $\mathcal{P}$.

In the case that the problem structure is such that multiple global optima exist (multimodal fitness-function), a crossover of search-points from distinct global optima may lead to a point which has a weak fitness value somewhere in-between the two parents in the search-space. In such a case it may be beneficial to restrict the crossover parents to individuals from a bounded region in the search-space.

5.3.3 Comments on the implementation of evolutionary algorithms

Evolutionary search approaches are very simple schemes that are straightforward to implement. However, evolutionary algorithms are time-consuming optimisation methods when compared to specialised optimisation approaches since the optimisation progress is guided by a random process.

Most of the time is spent for the calculation of fitness-values. Therefore, a straightforward approach is to prevent a fitness-value being calculated multiple times by storing search-points that were already visited.

Another aspect is the calculation of random bits. As the calculation of good random bits is expensive and weak random generators can create dependencies between search-points, strategies to reduce the number of random bits are required. This is possible, for instance, by calculating the next flipping bit in an individual representation instead of taking a random decision for each bit separately.

5.4 Asymptotic bounds and techniques

To foster the theoretical analysis of evolutionary algorithms, some methods have been derived that are especially well-suited for the analysis of evolutionary approaches. Generally, as the dimensions and structure of the problems to which evolutionary approaches are applied are typically too complex for a complete analytical consideration, most analytical methods consider evolutionary algorithms with very restricted population size and offspring population size.

5.4.1 A simple upper bound

In this section the method of the fitness based partition is described. It is a comparatively simple method that is suitable for providing respectable upper bounds on the expected optimisation-time of evolutionary algorithms with 'plus'-selection. We describe the method for the (1 + 1)-EA. However, it can be easily adapted for other random search schemes with 'plus'-selection.

Preparatory, the notion of a fitness-based partition is required.

Definition 5.4.4 : Fitness-based partition

> *Let $f : \mathbb{B}^n \to \mathbb{R}$ be a fitness-function. A partition $L_0, L_1, \ldots, L_k \subseteq \mathbb{B}^n$ with $\mathbb{B}^n = L_0 \cup L_1 \cup \cdots \cup L_k$ is a fitness based partition of f when*
>
> 1. $\forall i, j \in \{0, \ldots, k\}, \forall x \in L_i, y \in L_j : (i < j \Rightarrow f(x) < f(y))$ *and*
> 2. $L_k = \{x \in \mathbb{B}^n | f(x) = \max\{f(y) | y \in \mathbb{B}^n\}\}$
>
> *hold.*

Since the algorithm in question utilises plus-selection, the population of the algorithm can follow these partitions only in ascending order until the optimum is reached with L_k. The question is then, how long it takes to leave a partition L_i.

Definition 5.4.5 : Vacation probability

> *Let $f : \mathbb{B}^n \to \mathbb{R}$ be a fitness-function and $L_0, \ldots, L_k$ be a fitness based partition of f. For a standard bit-mutation-probability of p and $i \in \{0, 1, \ldots, k-1\}$*
>
> $$s_i := \min_{x \in L_i} \sum_{j=i+1}^{k} \sum_{y \in L_j} p^{H(x,y)}(1-p)^{n-H(x,y)} \tag{5.2}$$
>
> *defines the vacation-probability of L_i. In this formula, $H(x, y)$ describes the hamming-distance from x to y.*

The definition can be understood as follows. The term $p^{H(x,y)}(1-p)^{n-H(x,y)}$ describes the probability to mutate from x to y. When we fixate x for several y and sum up these probabilities, we obtain the probability for mutating from x to one of these y. Since we have, for an $x \in L_i$ summed up the y of all L_j with $i < j$, we obtain the probability for leaving L_i. This probability is minimised over all possible x so that we achieve the worst-case-probability. s_i is a lower bound for the probability to leave L_i with one mutation. Consequently, the expected count of mutations until this happens is bound from above by s_i^{-1}. This leads to the following result.

Assertion 5.4.1 : A simple Upper bound

Let $f : \mathbb{B}^n \to \mathbb{R}$ be a fitness-function and $L_0, \ldots, L_k$ a fitness-based partition of f. The expected optimisation-time of an $(1+1)$-EA is then bound from above by

$$E[T_{\mathcal{P}}] \leq \sum_{i=0}^{k-1} s_i^{-1}. \tag{5.3}$$

5.4.2 A simple lower bound

We discuss a simple general lower bound for evolutionary algorithms that utilise only mutation as a variation operator with standard bit mutation and a mutation-probability $p = \frac{1}{n}$ on functions $f : \mathbb{B}^n \to \mathbb{R}$ with exactly one global optimum.

Assertion 5.4.2 : A simple lower bound

> *Let $f : \mathbb{B}^n \to \mathbb{R}$ be a function with exactly one global optimum x^* and A an evolutionary algorithm that initialises its population uniformly at random and utilises only standard bit mutation with mutation probability $p = \frac{1}{n}$. The expected optimisation-time of this algorithm then is*
>
> $$E[T_{\mathcal{P}}] = \Omega(n \log(n)) \tag{5.4}$$

Proof. Let μ be the population size of A. For $\mu = \Omega(n \log(n))$ the algorithm requires already $\Omega(n \log(n))$ evaluations of fitness-values for search-points prior to finding x^* for the random initialisation of the population with probability $1 - 2^{-\Omega(n)}$. When $\mu = \mathcal{O}(n \log(n))$, we can see by applying Chernoff bounds that the probability that the hamming-distance of a search-point x to the optimum x^* is smaller than $\frac{n}{3}$ is

$$P(H(x, x^*) < \frac{n}{3}) = 2^{-\Omega(n)}. \tag{5.5}$$

We can therefore assume that at least $\frac{n}{3}$ bits have to be flipped in order to reach the optimum. The probability to flip one bit is $p = \frac{1}{n}$. The probability of not flipping the bit in t mutations is $(1 - \frac{1}{n})^t \geq e^{-\frac{t}{n-1}}$. With $t = (n-1)\ln(n)$ we obtain $e^{-\frac{t}{n-1}} = \frac{1}{n}$. Consequently, the probability that from $\frac{n}{3}$ bits in t mutations at least one does not mutate is at least $1 - (1 - \frac{1}{n})^{\frac{n}{3}} \geq 1 - e^{-\frac{1}{3}}$. This leads to

$$E_{T_{\mathcal{P}}} = (1 - 2^{-\Omega(n)}) \cdot (1 - e^{-\frac{1}{3}}) \cdot (n-1)\ln(n) = \Omega(n \log(n)). \tag{5.6}$$

□

5.4.3 The method of the expected progress

For some problems the optimisation process is similar during the whole optimisation run. In such a case we can keep to the idea that an algorithm does not deviate much from the expectation in most cases.

This can be used to derive a lower bound on the optimisation-time. The idea is to identify steps that are required for the optimisation and which are to be applied often in order to reach a global optimum. Then when we then bound the probability to achieve such a step from above, a lower bound can be derived.

When the expected count of these steps can be determined, we can typically bound the probability to obtain many of these steps fast from above. If the steps are also independent of each other, we know the expected number of these steps in a fixed number of generations.

Through Chernoff bounds it can be shown that the probability to deviate from this expected number is very small.

We describe the method for the $(1+1)$-EA. We denote the optimisation problem with $\mathcal{P}$ and assume a progress measure $\mathcal{F} : \mathbb{B}^m \to \mathbb{R}_0^+$ such that $\mathcal{F}(s_t) < \Delta$ represents the case that a global optimum was not found in the first t iterations. Let $T_\mathcal{P}$ denote the count of iterations required by the optimisation algorithm to reach one of the optimum values for the problem $\mathcal{P}$.

For every $t \in \mathbb{N}$ we have

$$\begin{aligned} E[T_\mathcal{P}] &\geq t \cdot P[T_\mathcal{P} > t] \\ &= t \cdot P[\mathcal{F}(s_t) < \Delta] \\ &= t \cdot (1 - P[\mathcal{F}(s_t) \geq \Delta]) \end{aligned} \tag{5.7}$$

With the help of the Markov-inequality we obtain $P[\mathcal{F}(s_t) \geq \Delta] \leq \frac{E[\mathcal{F}(s_t)]}{\Delta}$ and therefore $E[T_\mathcal{P}] \geq t \cdot \left(1 - \frac{E[\mathcal{F}(s_t)]}{\Delta}\right)$ This means that we can obtain a lower bound on the optimisation-time by providing the expected progress after t iterations.

6 Cooperative transmission schemes

> *Cooperative transmission is a cross-layer approach bridging the network and physical layers with the objective of forwarding information that is available at multiple terminals more reliably.*
>
> (Birsen Sirkeci-Mergen and Anna Scaglione: Randomised cooperative Transmission in Large-Scale Sensor Networks [82])

Cooperation is one of the major challenges of research on sensor networks. It covers many aspects in sensor networks such as energy consumption, sharing of resources for computation or finding of routing paths. Cooperation can also be utilised to improve aspects of data transmission in wireless networks. These approaches generally make use of spatial diversity of noise or achieve redundancy of transmission.

The utilisation of cooperative diversity in sensor networks has been studied by various research institutes during the past years [83, 84, 7, 85]. These approaches have in common that nodes in a network are utilised as relays [24, 25, 26], as it was first proposed by Cover and El Gamal in [27]. Neighbouring nodes repeat a signal sequence once it was received and achieve – sufficient temporal synchronisation presumed – constructive interference between superimposed transmit signals. The major complexity of these approaches lies in the temporal synchronisation of transmit signals that is commonly preconditioned. In these schemes, the capacity and robustness of a network of sensor nodes [7, 8] as well as the maximum transmission range [6] can be increased. Furthermore, the average energy consumption per node [9] and the time to propagate messages in sensor networks can be reduced [35] by these approaches. Superimposing signals was studied in [24] with regard to the probability of packet loss and in [86] with regard to the Shannon-capacity.

Another approach to obtain a transmission gain is to utilise the spatial diversity of distributed nodes and to combine the RF-transmit signal-components of distinct nodes on the channel.

This problem is addressed by the approach of virtual MIMO in wireless sensor networks [21, 20, 22]. In virtual MIMO, single antenna nodes are cooperating to establish a multiple antenna wireless sensor network. Virtual MIMO has capabilities to adjust to different frequencies and is highly energy-efficient [23, 11]. However, the implementation of MIMO capabilities in WSNs requires accurate time synchronisation, complex transceiver circuity and signal processing that might surcharge the power consumption and processing capabilities of simple sensor nodes.

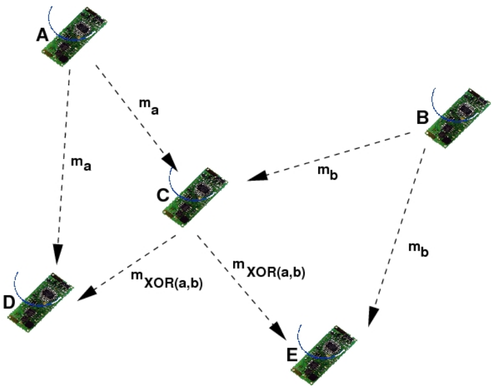


Figure 6.1: An example for network coding

Other solutions proposed are open-loop synchronisation methods as round-trip synchronisation based [36, 37, 38]. In this scheme, the destination sends beacon signals in opposed directions along a multi-hop circle. Beamforming is achieved when the processing time along the multi-hop chain is identical in both directions. This approach, however, does not scale with the size of a network.

Closed-loop feedback approaches include full-feedback techniques, that achieve carrier-synchronisation in a master-slave manner. The phase-offset between the destination and a source node is corrected by the receiver node. Diversity between transmit signals is achieved over CDMA channels [39]. This approach is applicable only to small network sizes and requires sophisticated processing capabilities at the source nodes.

6.1 Cooperative transmission

In cooperative networks, nodes relay each others messages to provide spatial diversity and to increase the spectral efficiency by combining a received signal at the physical layer.

6.1.1 Network coding

In traditional approaches, relay nodes forward messages they receive unmodified to reach a next hop. Another approach, however, is to allow relay nodes to modify incoming messages before further transmission. The essential idea of network coding is that nodes combine several incoming messages so that the cost for transmission is reduced [87, 88]. Figure 6.1 illustrates a network coding scenario.

Assume that nodes A and B both transmit messages m_a and m_b. Clearly, nodes D and E receive the messages directly from these nodes. Since wireless transmission is inherently a broadcast scheme, also node C overhears both messages. In traditional broadcast schemes, node C would transmit messages m_a and m_b separately so that both nodes D and E received both messages m_a and m_b in the end (one message is even received twice by each of the nodes). In network coding, node C could combine both messages m_a and m_b by, for

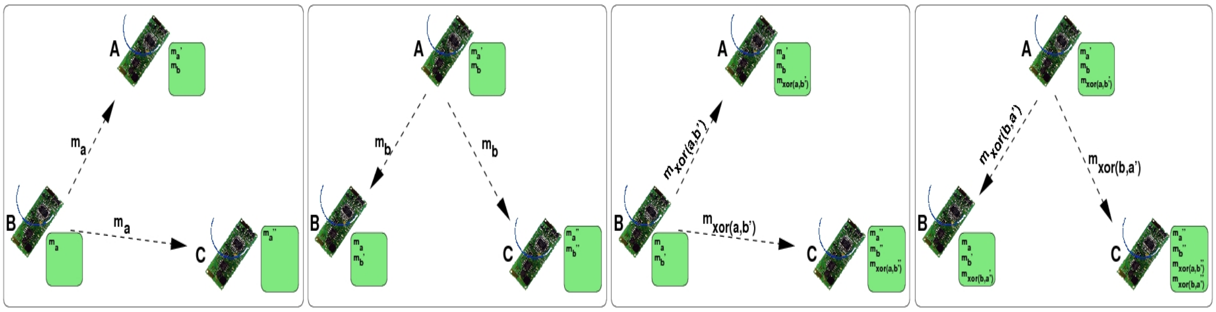


Figure 6.2: Error reduction by network coding

instance, XOR to receive a combined message $m_{XOR(a,b)}$. After receiving $m_{XOR(a,b)}$, both nodes D and E obtain the messing message (m_a or m_b) by computing the logical XOR between the messages they received.

Regarding the overall energy consumption, one transmission is omitted in this network coding example, so that this scheme is more energy-efficient compared to the traditional broadcast scheme.

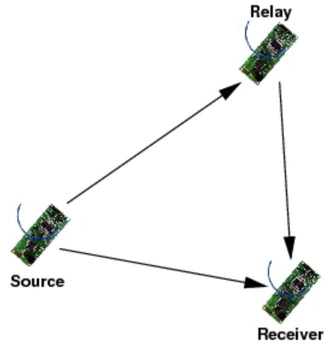
A similar situation is depicted in figure 6.2.

Both nodes A and B transmit a message (m_a, m_b) to a remote node C. Also, after receiving the message from the respective other source node (A or B), both nodes transmit $m_{XOR(a,b')}$ and $m_{XOR(b,a')}$. The notion a' and b' indicates that the message received over the wireless channel might be subject to transmission errors. Now, node C holds $m''_a, m''_b, m''_{XOR(a,b')}, m''_{XOR(b,a')}$. It can consequently obtain three copies of m_a and m_b by combining the received messages. When the probability for bit-errors is low, this scheme can be utilised to detect and correct errors. Consider, for example one bit position i for three copies of message m_a:

- $m_a(i)'$
- $XOR(m'_b, m_{XOR(b,a')}(i)'$
- $XOR(m'_b, m_{XOR(a,b')}(i)'$

Assume that one bit is erroneous with probability $\frac{1}{m}$ for all transmissions. Then, $m_a(i)'$, $m_b(i)'$ are incorrect with probability $\frac{1}{m}$ each and $m_{XOR(b,a')}(i)'$, $m_{XOR(a,b')}(i)'$ are incorrect with probability $1 - \left(1 - \frac{1}{m}\right)^2$ each. Consequently, the probability that more than one of the received messages is incorrect at position i is at most $\frac{1}{m} \cdot \left(1 - \left(1 - \frac{1}{m}\right)^2\right)^2$, which is smaller than $\frac{1}{m}$.

A more detailed discussion that also considers the coverage area, pathloss and fading is found in [89].



	Block 1	Block 2	Block 3	Block 4
Source	$c_1(1, w_1)$	$c_1(w_1, w_2)$	$c_1(w_2, w_3)$	$c_1(w_3, 1)$
Relay	$c_2(1)$	$c_2(w_1)$	$c_2(w_2)$	$c_2(w_3)$

Figure 6.3: A two-hop strategy for multi-hop relaying

6.1.2 Multi-Hop approaches

In the case of multi-hop-relaying based on the physical channel, multi-hop is interpreted as a multi-dimensional relay channel. Communication is allowed between all nodes [28]. It was shown that this approach optimally divides the network resources regarding information theoretic metrics [12]. In larger multi-hop scenarios it is, however, not well-suited since the count of successfully transmitted bits per square metre decreases quadratically with the size of the network [29, 30].

The general idea of this approach is to retransmit received messages by a relay node so that the destination will receive not only the message from the source destination but also from the relay. Figure 6.3 illustrates this scheme for a two-hop strategy.

A message w is divided into B blocks $w_1, \ldots, w_B$. The transmission is performed in $B+1$ blocks using the code words $c_1(w_i, w_j)$ and $c_2(w_i)$. The relay node will always transmit the word w_i recently observed from the source node while the source still transmits both, w_i and w_{i+1}. Although omitting further details, the general idea of this approach is to encode redundancy in $c_1(w_i, w_j)$ and $c_2(w_i)$ that both depend on w_i so that the destination node can reliably decode w_i.

For the destination it can be shown that by sufficiently combining the blocks received from source and relay the data-rate is improved [28].

6.1.3 Data flooding

An alternative approach attempts to 'flood' the network with a message to be transmitted [31, 32]. At Cornell University 'Opportunistic Large Arrays' (OLAs) have been proposed for cooperative transmission [90, 91, 82]. A node will retransmit a received message at its reception. It has been shown that the approach outperforms non-cooperative multi-hop schemes significantly. This approach floods the network with nodes that retransmit a desired transmit signal. Basically, in this scheme, an avalanche of signals is proceeded through the network. When the network is sufficiently dense, the distinct transmit signals are superimposed with special OLA modulations it is then even possible to encode information into this signal wave. This transmission scheme is robust to environmental noise but not capable of coping with moving receivers due to the inherent randomness of the protocol.

It was derived that the average energy consumption of nodes is decreased [9, 10] and the transmission time is reduced compared to traditional transmission protocols in wireless sensor networks [35].

6.2 Multiple antenna techniques for networks of single antenna nodes

Multi-antenna systems have been studied for their potential to increase the channel capacity in fading channels [92]. MIMO systems can achieve higher data-rates under identical transmit power and bit-error-rate as SISO systems. However, a key requirement for these systems is that the distinct antennas are separated by at least $\frac{\lambda}{2}$. As sensor nodes are typically of very restricted dimensions, this requirement might easily lead to the conclusion that single sensor nodes with multiple antennas are absurd. When, however, single nodes are allowed to cooperate for their transmission so that a set of nodes is treated as a multi-antenna transmitter or receiver, a cooperative MIMO system can be constructed. This approach to cooperative transmission was proposed in [93]. The idea is to create clusters of collaboratively transmitting nodes in a network [12]. Figure 6.4 illustrates this transmission scheme. It is denoted as virtual MIMO or cooperative MIMO.

In [93] the general idea to apply MIMO transmission schemes to a scenario of distributed transmitters and receivers was proposed by using Alamouti diversity codes [92]. The Alamouti code with two transmit antennas proposed in [94]. The extension of the Alamouti code to more than two antennas is discussed in [92]. We will discuss Alamouti codes with two transmit antennas in the following.

The Alamouti diversity scheme for two receivers is depicted in figure 6.5

In the figure, two transmit nodes transmit to a single receive node. At the receiver, the system is constituted by a channel estimator, a combiner and a maximum likelihood detector.

Both transmit nodes will, at a given symbol period t simultaneously transmit two signals. The signal transmitted from node A is denoted as s_0 and the signal transmitted from node B is denoted s_1. In the following symbol period $t+T$, node A transmits signal $(-s_1^*)$ and node B transmits s_0^*.[1] Here, s^* denotes the complex conjugate of s.

[1] In this description, distinct signals are separated in time. Therefore, this scheme is referred to space-time coding. It is, however, also possible to separate signals in frequency by utilising distinct transmit channels. In this case, the coding scheme is denoted as space-frequency coding

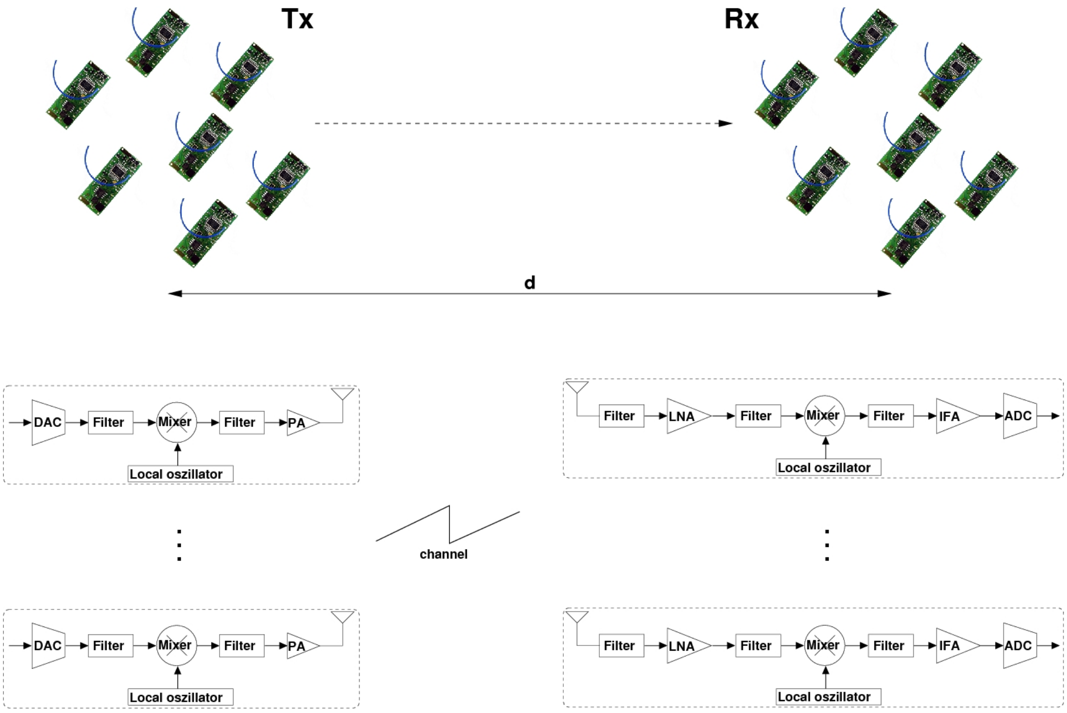


Figure 6.4: Illustration of virtual MIMO in wireless sensor networks

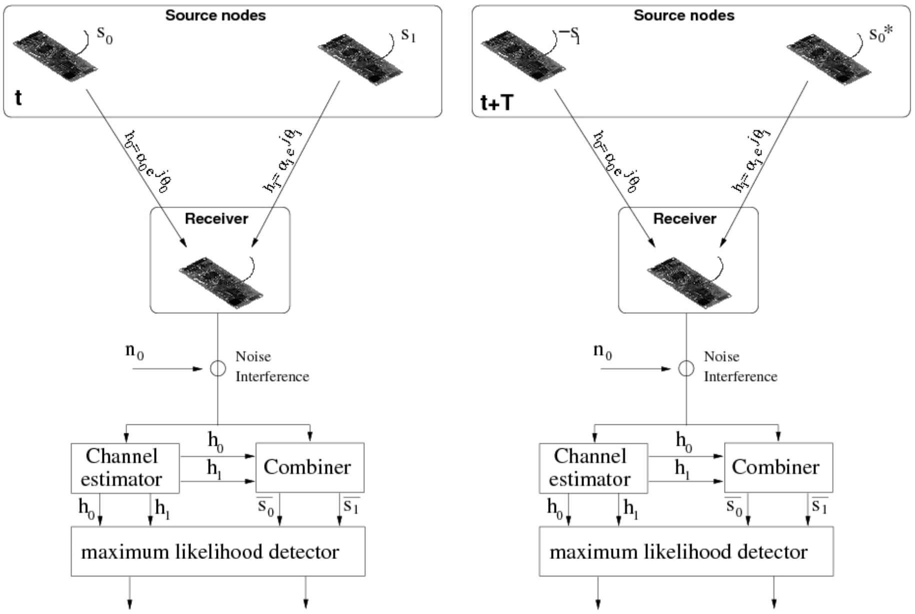


Figure 6.5: Illustration of the Alamouti transmit diversity scheme with two receivers

Assume that the channel at time t is modelled by a complex multiplicative distortion $H_A(t)$ for node A and $H_B(t)$ for node B. When we further assume that fading is constant over one symbol period, we can write

$$\begin{aligned} H_A(t) &= H_A(t+T) = H_A = \alpha_A e^{j\Theta_A} \\ H_B(t) &= H_B(t+T) = H_B = \alpha_B e^{j\Theta_B} \end{aligned} \tag{6.1}$$

The received signals are then expressed as

$$\begin{aligned} r_0 &= r(t) = H_A s_0 + H_B s_1 + n_0 \\ r_1 &= r(t+T) = -H_A s_1^* + H_B s_0^* + n_1 \end{aligned} \tag{6.2}$$

with r_0 and r_1 representing the received signals at t and $t+T$. n_0 and n_1 represent receiver noise and interference.

The combiner module depicted in figure 6.5 creates the signals

$$\begin{aligned} \overline{s_0} = H_A^* r_0 + H_B r_1^* = (\alpha_A^2 + \alpha_B^2) s_0 + H_A^* n_0 + H_B n_1^* \\ \overline{s_1} = H_B^* r_0 - H_A r_1^* = (\alpha_A^2 + \alpha_B^2) s_1 - H_A n_1^* + H_B^* n_0 \end{aligned} \tag{6.3}$$

and forward these to the maximum likelihood detector. The maximum likelihood detector uses for s_0 and s_1 the decision rules

- Choose s_i iff

$$\begin{aligned} & (\alpha_0^2 + \alpha_1^2 - 1)|s_i|^2 + d^2(\overline{s_0}, s_i) \\ \leq & \ (\alpha_0^2 + \alpha_1^2 - 1|s_k|^2 + d^2(\overline{s_0}, s_k), \\ & \forall i \neq k \end{aligned} \tag{6.4}$$

- Choose s_i iff

$$d^2(\overline{s_0}, s_i) \leq d^2(\overline{s_0}, s_k), \forall i \neq k \tag{6.5}$$

where $d^2(s_i, s_j)$ is the squared Euclidean distance between signals s_i and s_j:

$$d^2(s_i, s_j) = (s_i - s_j)(s_i^* - s_j^*). \tag{6.6}$$

In virtual MIMO schemes, it is assumed that each node has a pre-assigned index i and will transmit the transmission sequence that the i-th antenna would transmit in an Alamouti MIMO system. At the receiver side, the receiver nodes join the received sum-signals cooperatively.

All nodes in a cluster cooperate when receiving or transmitting data from and to neighbouring nodes. Basically, a cluster is seen as a single multiple antenna device so that MIMO, SIMO, or MISO transmission is possible in wireless sensor networks. By grouping of nodes, the complexity of synchronising nodes in a cluster is reduced. [12, 5] derive results regarding the energy efficiency and the optimum cluster design. It was shown that this scheme can be more energy-efficient than traditional SISO transmission between nodes of a sensor network. The approach allows the utilisation of existing routing algorithms and multi-hop theory when a cluster is understood as minimum entity. The capacity of a sensor network that is organised by this approach is, however, decreased compared to other approaches of cooperative transmission [30, 95].

It was presumed that the local oscillators of all transmit and receive nodes are synchronised. However, this synchronisation is typically not easy to establish among distributed wireless nodes.

Several solutions to this requirement are proposed by distributed transmit beamforming. A key distinguishing feature of distributed transmit beamforming with respect to centralised beamforming is that each source node in a distributed beamformer has an independent local oscillator. These typically multiply the frequency of a crystal oscillator up to a fixed nominal frequency. Carrier frequencies generated in this manner, however, typically exhibit variations in the order of 10-100 parts per million (ppm) with respect to the nominal. If uncorrected, these frequency variations among sources are catastrophic for transmit beamforming since the phases of the signals may drift out of alignment over the duration of the transmission and may even result in destructive combining at the destination. The first goal, therefore, is to synchronise the carrier frequencies for the different sources to minimise or eliminate frequency-offset.

One approach to frequency synchronisation is to employ a master-slave architecture [18] where slave source nodes use phase-locked loops (PLLs) to lock their phase to a reference carrier-signal broadcast by a master node.

A simple form of a PLL consists of three components:

- A phase detector that compares the phase-offset of an input signal $Y(s)$ with the phase-offset of a controlled oscillator. It creates an output signal $E(s)$ – the error signal – that is proportional to the phase-offset
- A filter that computes with the function $F(s)$ from the error signal $E(s)$ the control signal $C(s)$.
- A variable electronic oscillator. This module might, for instance be implemented by a Voltage Controlled Oscillator (VCO) that might be altered in its frequency by, for instance, a capacity diode. Other possible implementation are Numerically Controlled Oscillators (NCO) that take numerical inputs to alter the oscillation frequency.

Alternatively, the destination node could broadcast a reference carrier to facilitate frequency synchronisation among the source nodes [36, 37, 39]. A source node that estimates its frequency-offset to be Δf can multiply its complex baseband transmitted signal by $e^{-j2\pi\Delta ft}$. When frequency synchronisation is achieved, the phase of the transmission from distinct transmitters must be synchronised to achieve reasonable phase coherence at a receiver.

In the following sections open-loop and closed-loop distributed carrier-synchronisation are detailed.

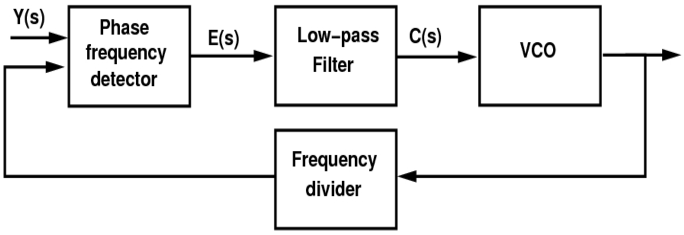


Figure 6.6: Schematic illustration of a phase locked loop

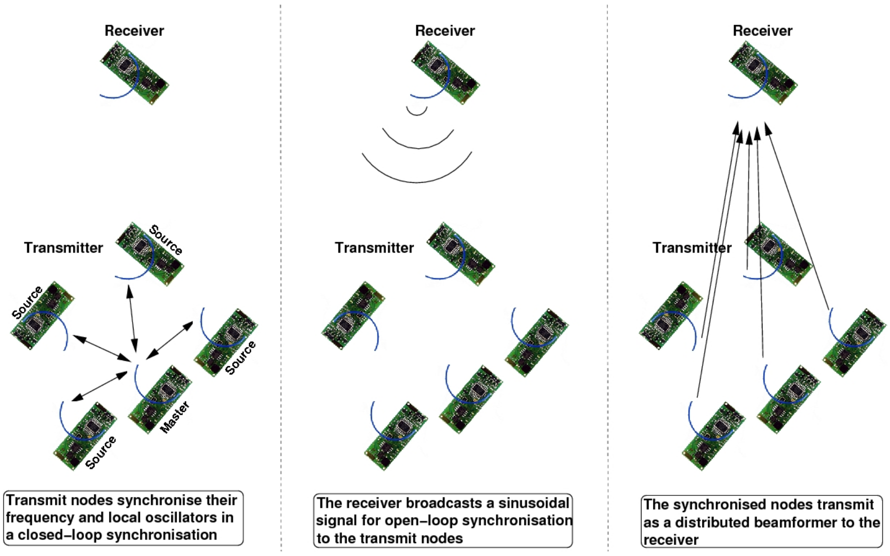

Receiver
Transmitter
Source
Source
Source
Source
Master
Transmit nodes synchronise their frequency and local oscillators in a closed–loop synchronisation
Receiver
Transmitter
The receiver broadcasts a sinusoidal signal for open–loop synchronisation to the transmit nodes
Receiver
Transmitter
The synchronised nodes transmit as a distributed beamformer to the receiver

Figure 6.7: Illustration of the master-slave open-loop distributed adaptive carrier-synchronisation scheme

6.2.1 Open-loop distributed carrier-synchronisation

In open-loop phase-synchronisation schemes, transmit nodes communicate to cooperatively achieve phase-synchronisation at a remote receiver. Interaction with the receiver is minimised in these approaches. The receiver may, for instance, transmit a sinusoidal signal. Transmitters utilise this signal together with the signals from other transmitters to achieve sufficient phase-synchronisation.

Master-slave open-loop distributed carrier-synchronisation

The master-slave open-loop carrier-synchronisation approach was described in [18]. The general principle is illustrated in figure 6.7

Initially, among the transmitters one is identified as master node, while the others are considered slaves. These master and slave nodes in the sensor network communicate to synchronise their frequency and local oscillators. For frequency synchronisation, the master node broadcasts a sinusoidal signal to the slave nodes that in turn estimate and correct

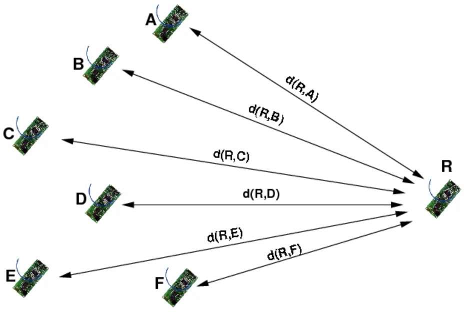


Figure 6.8: Illustration of the open-loop distributed beamforming scenario with known relative node locations

their relative frequency-offset to this signal. Phase synchronisation is then achieved in a closed-loop method.

In order to achieve beamforming to a receiver node, all transmitters then estimate their channel response to the destination. This can be achieved by the destination broadcasting a sinusoidal signal on the carrier frequency. As transmitters are already synchronised, they can estimate their individual complex channel gain to the destination using their phase and frequency synchronised local oscillators. The transmitters then transmit as a distributed beamformer by applying the complex conjugate of the gains to their transmitted signals.

Carrier synchronisation when locations of distributed nodes are known

An approach to synchronise the carrier phase-offsets of a set of distributed nodes was described in [17]. The assumption in this approach is that the distance between the receiver node and the transmit nodes is known and that a Line-of-sight (LOS) scenario is considered. The receiver node serves as a master node broadcasting both, carrier and timing signals. Assume that a slave node i is located at distance $d(i) = d_0(i) + d_e(i)$ from a receiver node. In this formula, $d_0(i)$ denotes the distance between the receiver and node i and $d_e(i)$ is an additional placement error. Figure 6.8 illustrates this scenario.

The receiver node broadcasts a carrier-signal $\Re\left(m(t)e^{j(2\pi f_0 t)}\right)$. A transmit node i receives a noisy signal

$$\Re\left(n_i(t)m(t)e^{j(2\pi f_0 t+\gamma_0(i)+\gamma_e(i))}\right) \tag{6.7}$$

where

$$\gamma_0(i) = \frac{2\pi f_0 d_0}{c} = \frac{2\pi d_0}{\lambda_0} \tag{6.8}$$

is the normal phase-offset from the transmitted carrier and

$$\gamma_e(i) = \frac{2\pi f_0 d_i(i)}{c} = \frac{2\pi d_i(i)}{\lambda_0} \tag{6.9}$$

represents the phase error resulting from the placement error $d_e(i)$. $n_i(t)$ is the noise signal. Each transmit node applies a PLL to lock on to the carrier with the result

$$\Re\left(n_i(t)m(t)e^{j(2\pi f_0 t+\gamma_\Delta(i))}\right) \tag{6.10}$$

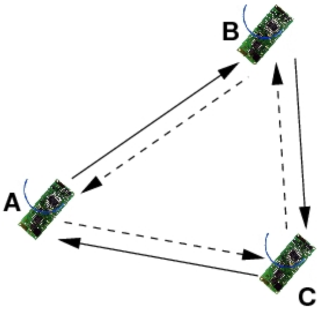


Figure 6.9: Illustration of the Round-trip open-loop distributed adaptive carrier-synchronisation scheme

In this equation, $\gamma_{\Delta}(i) = \gamma_0(i) + \gamma_e(i) - \gamma_{pll}(i)$. The phase variations across different slaves originate from placement errors and PLL errors. The PLL error can be reduced by increasing the SNR. When the locations of nodes are sufficiently well known and fixed, phase-synchronisation among distributed carrier-signals is therefore possible with this approach.

Carrier synchronisation when locations of distributed nodes are not known

In a scenario where distances among nodes are not known, the same approach as above can be applied when relative distances among nodes are calculated and the slave clocks are updated accordingly. The authors of [39] describe this approach. In advance of synchronising carrier phase-offsets, clock-offsets are estimated by a standard approach [96].

When nodes are moving, this estimation of clock offsets is a constant procedure. However, when the velocity is too high, it is not feasible to synchronise a sufficiently large number of nodes by this approach. Furthermore, the overhead for the time synchronisation is energy demanding.

Round-trip open-loop distributed carrier-synchronisation

An approach to open-loop phase-synchronisation between two source nodes and one receiver node that allows for high mobility of source and destination nodes was presented in [36, 37, 38]. Figure 6.9 illustrates this scenario. The destination node C will in this system transmit a sinusoidal beacon signal $\Re\left(m(t)e^{j(2\pi f_0 t+\gamma_0)}\right)$ to both source nodes. At the nodes the signals $\Re\left(m(t)e^{j(2\pi f_0 t+\gamma_0^A)}\right)$ and $\Re\left(m(t)e^{j(2\pi f_0 t+\gamma_0^B)}\right)$ are received. Both source nodes employ a primary frequency synthesis PLL [97] tuned onto the beacon frequency f_0 and generate a low-power secondary sinusoidal beacon that is phase locked to the received beacon but of frequency $f_1^A = \frac{N_1^A}{M_1^A}f_0^A$ and $f_1^B = \frac{N_1^B}{M_1^B}f_0^B$. N_1 and M_1 are integers. The signals $\Re\left(m(t)e^{j(2\pi f_1^A t+\gamma_1^A)}\right)$ and $\Re\left(m(t)e^{j(2\pi f_1^B t+\gamma_1^B)}\right)$ are then transmitted among the source nodes. At receiving a secondary beacon signal, a source node again generates a carrier-signal at frequency $f_c^A = \frac{N_1^A}{M_1^A} \cdot f_1^B = f_c^B = \frac{N_1^B}{M_1^B} \cdot f_1^A$ that is phase locked to the secondary beacon signal. These carrier-signals are then utilised to transmit to the destination node. The received signal at the destination can be written as

$$\Re\left(m(t)e^{j(2\pi f_c^A t+\gamma_2^A)} + m(t)e^{j(2\pi f_c^B t+\gamma_2^B)}\right). \tag{6.11}$$

Observe that frequencies from both source nodes are identical. Also, when the round trip time along $C \rightarrow A \rightarrow B \rightarrow C$ and $C \rightarrow B \rightarrow A \rightarrow C$ are similar, the phase-offset $\gamma_\Delta = \gamma_2^A - \gamma_2^B$ is small at the destination node.

Since the round-trip signal propagation has low delays, this method can be applied also at high velocities of nodes. It is, however, only feasible for exactly two source nodes. The expected maximum gain is therefore limited.

6.2.2 Closed-loop distributed carrier-synchronisation

In closed-loop phase-synchronisation schemes, the destination directly controls the relative phase-offset between RF-transmit signal-components. This is achieved by measuring a function of the received phases of the source transmissions and then transmitting feedback signals that guide the transmitter's phase adjustment process. Interaction between transmit nodes is minimal in these approaches.

Full feedback closed-loop synchronisation

A full feedback-based closed-loop carrier-synchronisation technique suitable for distributed beamforming is described in [39]. Carrier frequency synchronisation is achieved over a master-slave approach. The Receiver node acts as the master node. The phase-offset between the master and the i-th transmitter is corrected over a closed-loop protocol:

- The master broadcasts a common beacon to all source nodes.
- Source nodes 'bounce' this beacon back to the master utilising a different frequency. Transmission from source nodes to the master nodes is achieved in a direct-sequence code division multiple access (DS-CDMA) scheme in which all source nodes utilise distinct codes. By this approach, the master is then able to distinguish the received signals.
- After receiving the bounced beacons, the master estimates the phase of each source relative to its originally transmitted beacon. These estimates are divided by two, quantised and then transmitted to the source nodes via DS-CDMA. This message is utilised by the source nodes for phase compensation and may also contain clock correction information to facilitate symbol timing synchronisation.
- Source nodes extract their phase compensation estimate, and adjusts their carrier phase.

When phase-offsets did not change significantly between synchronisation and beamforming intervals, the bandpass transmissions from all source nodes will then combine coherently. The authors of [39] also considered the impact of energy allocation between synchronisation and information transmission on the error probability of signals transmitted in this distributed beamforming manner. It was derived that an optimal energy trade-off exists and that allocating too much or too little energy to carrier-synchronisation is inefficient.

1-bit feedback-based closed loop synchronisation

In [15] an iterative process to synchronise signal phases of transmit signals at a remote receiver without inter-node communication is described. It is assumed that, for a network of size n, initially the carrier phase-offsets γ_i of transmit signals $e^{j(2\pi(f+f_i)t+\gamma_i)}$; $i \in \{1..n\}$ are arbitrarily distributed. When a receiver requests a transmission from the network, carrier phases are synchronised in an iterative process.

1. Each source node i adjusts its carrier phase-offset γ_i and frequency-offset f_i randomly
2. The source nodes transmit to the destination simultaneously as a distributed beamformer.
3. The receiver estimates the level of phase-synchronisation of the received sum-signal (e.g. by SNR or comparison to an expected signal).
4. A feedback on the level of synchronisation is broadcast to the network. Nodes sustain their phase adjustments if the feedback has improved or else reverse them.

These four steps are iterated repeatedly until a stop criteria is met (e.g. maximum iteration count or sufficient synchronisation). Figure 6.10 illustrates this procedure. The computational complexity for transmit nodes in this approach is low as only the phase and frequency of the transmit signal is adapted according to a random process. Further processing is not required for synchronisation.

This process was successfully applied in [13, 14, 40, 42].

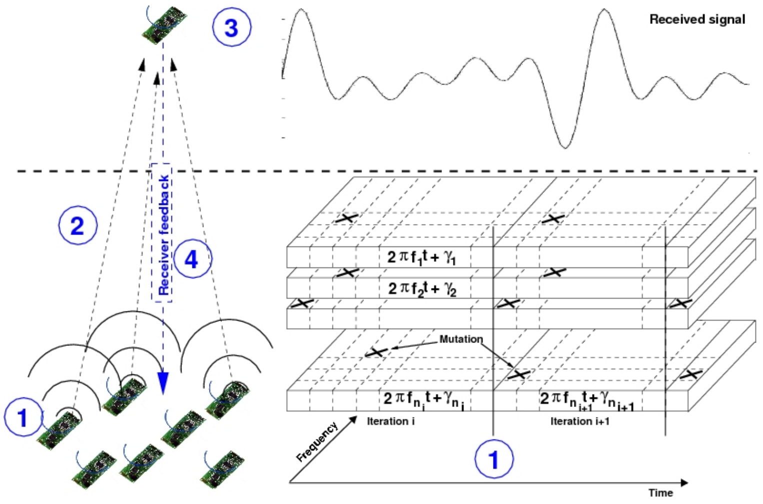


Figure 6.10: An iterative approach to closed-loop distributed adaptive transmit beamforming

7 Analysis of a simple closed loop synchronisation approach

> *[...] distributed adaptive beamforming is on the cusp of feasibility. The prototypes reported in the literature thus far have focused on demonstrating that the critical task of aligning carrier phases at the intended destination is feasible. The next step is to investigate and demonstrate distributed beamforming in a networked context.*
>
> (Raghuraman Mudumbai et al: Distributed Transmit Beamforming: Challenges and Recent Progress [15])

As detailed in section 6.2.2, the 1-bit feedback-based closed loop synchronisation is probably the slowest approach to synchronise carrier phases from the methods presented. However, the computational load for a single node is also by far the smallest among all methods. Beside applying a random decision on the carrier phase-offset of single nodes, in each iteration only an additional binary decision is taken whether to keep the carrier phase based on the feedback received from a master node.

The process has been studied by different authors [41, 42, 43, 13]. The distinct approaches proposed differ in the implementation of the first and the fourth step specified above. The authors of [43] show that it is possible to reduce the number of transmitting nodes in a random process and still achieve sufficient synchronisation among all nodes.

In [41, 42, 43] a process is described in which each node alters its carrier phase-offset γ_i according to a normal distribution with small variance in each iteration. In [13] a uniform distribution is utilised instead but the probability for one node to alter the phase offest of it's carrier-signal is low. Only in [42] not only the phase, but also frequency is adapted.

Significant differences among these approaches also apply to the feedback and the reactions of nodes in the iterative process. In [15, 42, 43] a one-bit feedback is utilised. Nodes sustain their phase modifications when the feedback has improved and otherwise reverse them. In [43] it was shown that the optimisation-time is improved by factor of two when a node as response to a negative feedback from the receiver applies a complementary phase-offset instead of simply reversing its modification. In [13], authors propose to utilise more than one bit as feedback so that parameters of the optimisation can be adapted with regard to the optimisation progress.

The strength of feedback-based closed-loop distributed adaptive beamforming in wireless sensor networks is its simplicity and low processing requirements that make it feasible for the application in networks of tiny sized, low-power and computationally restricted sensor nodes.

The first step to problem analysis is always a very precise understanding of the problem itself. A reasonable general understanding of the optimisation process was already gained by the definition of the iterated process detailed in section 6.2.2. In this chapter we will describe this process analytically to be able to provide some quantitative results on the optimisation performance as well as impacts of parameter and problem settings. This discussion will detail design decisions as, for example, the random phase alteration method or the feedback function. After describing these aspects in a mathematical sense, we are able to provide and prove sound results on the synchronisation performance.

In particular, we derive asymptotic bounds on the synchronisation performance of the feedback-based distributed adaptive beamforming process for several proposed optimisation algorithms. This completes the studies in the literature where, for instance, Mudumbai et al show in [18] a linear upper bound on the convergence time of the one-bit feedback based closed loop synchronisation approach when the probability distribution is altered during synchronisation so that always an optimal distribution for the phase adaptation is chosen. In our case, we derive an asymptotically sharp bound on the expected convergence time when a fixed uniform probability distribution is chosen.

Additionally, we propose algorithms and algorithmic modifications to improve the synchronisation performance further. Among these, an asymptotically optimal algorithm is presented.

7.1 Analysis of the problem-scenario

In this section we discuss algorithmic aspects of the 1-bit feedback-based closed loop phase-synchronisation approach for distributed adaptive transmit beamforming in wireless sensor networks. Figure 7.1 illustrates a general scenario for this approach.

As a first observation, note that the iterative optimisation approach basically represents an evolutionary random search method. By altering the phase-offset of carrier-signals, new search-points are requested. When the fitness-value feedback by the remote receiver is decreasing, the modifications of one iteration are discarded.

Generally, this is the behaviour of a $(1+1)$-evolutionary algorithm. Naturally, the population size and the offspring population size are 1, since at one time instant, each node can only transmit a carrier-signal at one distinct carrier phase-offset (and not multiple phase-offsets at once). Although also greater population sizes may be modelled by subsuming several iterations, the standard optimisation approach proposed can be modelled by a standard $(1+1)$-EA. A suitable algorithmic representation of the optimisation problem is discussed in the following.

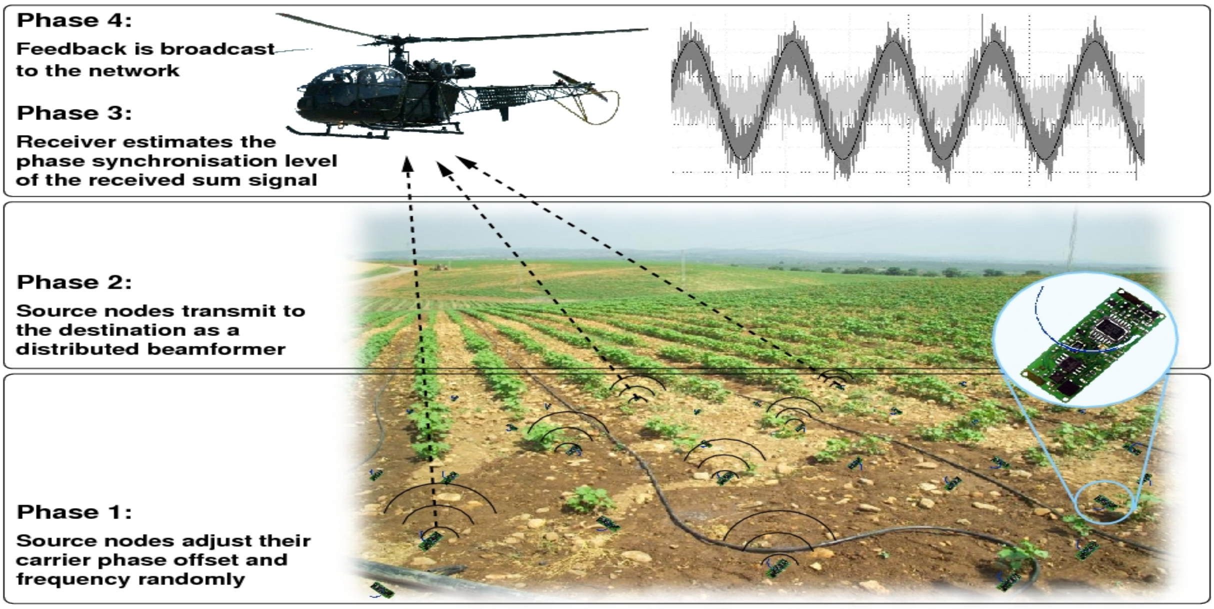

Phase 4:
Feedback is broadcast to the network
Phase 3:
Receiver estimates the phase synchronisation level of the received sum signal
Phase 2:
Source nodes transmit to the destination as a distributed beamformer
Phase 1:
Source nodes adjust their carrier phase offset and frequency randomly

Figure 7.1: A schematic overview on feedback-based closed-loop distributed adaptive transmit beamforming in wireless sensor networks

7.1.1 Representation of individuals

The optimisation process described consists of several iterations. During each iteration (e.g. after step 1) the phases and frequencies of all transmit signals are fixed while they might change from one iteration to another. We therefore model the iteration-wide fixed phase and frequency configurations of all distinct transmit signals as one individual in the sense of the optimisation process. In other words, an individual in the optimisation process is the set of all phase and frequency modifications (γ_i, f_i) of the base band signal for all source nodes $i \in \{1, \dots, n\}$.

How shall we represent an individual for the analytic consideration in the following sections? Possible representations are

- As ordered set of phase and frequency pairs (γ_i, f_i). A drawback of this solution is that similarity measures between individuals are not straightforward so that an intuitive understanding of a neighbourhood of individuals is not provided.

- A vector $V = v_1, \dots, v_{2n}$ of phases and/or frequencies (e.g. $v_{2i-1} = \gamma_i; v_{2i} = f_i$). This representation has the favourable property that we can then represent configurations as points in multi-dimensional search-spaces so that simple distance (similarity) measures between configurations are straightforward (e.g. Euclidean distance). Therefore, a neighbourhood measure for the configuration is implicitly given.

- Binary representations in which possible phase and/or frequency-shifts are encoded for each source node and source node sub-configurations are concatenated to one

$$\mathcal{I} = \underbrace{\overset{\substack{\text{Phase offset}\\ \text{of node 1}\\ |}}{0\dots1}}_{\log(k)\text{ bits}} \; \underbrace{\overset{\substack{\text{Frequency}\\ \text{offset for}\\ \text{node 1}\\ |}}{0\dots1}}_{\log(\varphi)\text{ bits}} \dots \underbrace{\overset{\substack{\text{Phase offset}\\ \text{for node } i\\ |}}{0100\dots1001}}_{\substack{\log(k)\text{ bits encode } k\\ \text{distinct possible}\\ \text{phase-offsets of node}\\ i}} \; \underbrace{\overset{\substack{\text{Frequency}\\ \text{offset for}\\ \text{node } i\\ |}}{0100\dots1001}}_{\substack{\log(\varphi)\text{ bits encode } \varphi\\ \text{distinct}\\ \text{frequency-offsets of}\\ \text{node } i}} \dots \underbrace{\overset{\substack{\text{Phase offset}\\ \text{for node } n\\ |}}{0\dots1}}_{\log(k)\text{ bits}} \; \underbrace{\overset{\substack{\text{Frequency}\\ \text{offset for}\\ \text{node } n\\ |}}{0\dots1}}_{\log(\varphi)\text{ bits}}$$

Figure 7.2: Possible binary representation of individuals

global binary representation of individuals (Figure 7.2 illustrates this representation scheme exemplary). Basically, all possible phase shift values for a single source node are enumerated and represented by a binary value. Although the number of possible phase shifts is infinite, any practical sensor node implementation can only realise a finite number of possible phase shifts. We therefore assume that this abstraction from an infinite number of possibilities does represent the problem-scenario well.

This solution is favourable since various results to binary search-spaces are known and can be applied. In order to respect neighbourhood relationships, similar phase-offsets are to be mapped onto similar binary representations. One problem with a binary configuration is the neighbourhood function. The hamming-distance between neighbouring points (with similar phase shift for a single source node) might be very large when possible phase-offsets are simply enumerated. We can overcome his problem by using Gray codes (cf. section 5.3.1). A drawback of Gray Codes is that numbers that are very far apart also have hamming-distance 1.

For our analytic consideration, Grey Codes will be sufficient. We therefore assume in the following that individuals are encoded by Grey Codes as described above.

7.1.2 Feedback function

At receiving the superimposed sum-signal

$$\zeta_{\text{sum}} = \Re\left(m(t)e^{j2\pi f_c t}\sum_{i=1}^{n}\text{RSS}_i e^{j(\gamma_i+\phi_i+\psi_i)}\right) \tag{7.1}$$

the receiver estimates the quality of the synchronisation. A possible, simple implementation for this calculation is to utilise the signal power or SNR of the superimposed signal ζ_{sum} at the receiver. Figure 7.3 illustrates the superimposition of received signals components $\Re\left(m(t)e^{j(2\pi f_i t+\gamma_i+\phi_i+\psi_i)}\right)$ and $\Re\left(m(t)e^{j(2\pi f_{\bar{i}} t+\gamma_{\bar{i}}+\phi_{\bar{i}}+\psi_{\bar{i}})}\right)$ from two nodes i and $\bar{i}$. For ease of presentation, the phase-offset due to the signal propagation, ϕ_i and the phase-offset due to distinct phases of the local oscillator ψ_i are omitted in the figure. At the receiver, both signals combine to a new, overlayed signal

$$\zeta_{i,\bar{i}} = \Re\left(m(t)e^{j2\pi f_c t}\sum_{\varsigma\in[i,\bar{i}]}\text{RSS}_i e^{j(\gamma_\varsigma+\phi_\varsigma+\psi_\varsigma)}\right) \tag{7.2}$$

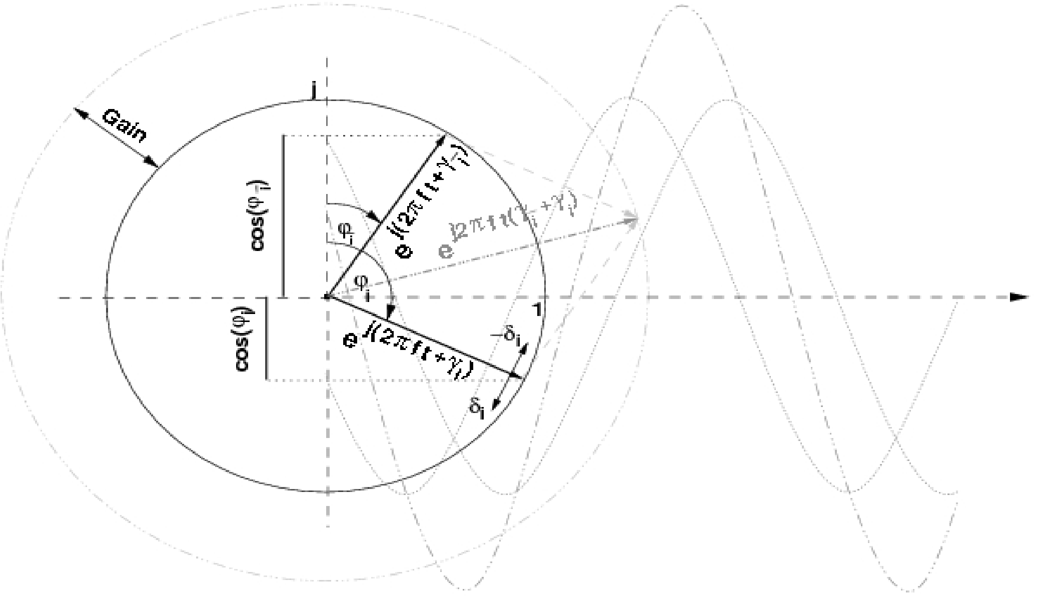

Gain
j
$\cos(\varphi_{\bar{i}})$
$\cos(\varphi_i)$
$\varphi_{\bar{i}}$
φ_i
$e^{j(2\pi f t+\gamma_{\bar{i}})}$
$e^{j(2\pi f t+\gamma_i)}$
$e^{j2\pi f t(\gamma_{\bar{i}}+\gamma_i)}$
1
$-\delta_i$
δ_i

Figure 7.3: Superimposition of received signal-components from nodes i and $\bar{i}$.

After the quality of the synchronisation is calculated at the receiver, a feedback is broadcast to the sensor nodes to guide the optimisation process. The authors of [40, 41] propose to utilise a one bit feedback. A '0' in this feedback scheme means that the synchronisation quality has deteriorated while a feedback of '1' means that the quality is at least as good as in the last iteration. The advantage of this approach is that the amount of data transmitted between network and receiver is minimised. This efficiency can be 'invested' into higher redundancy schemes for the transmission in order to guard the information from transmission errors. A drawback of this solution is, however, that the expected progress of the optimisation is not known at the source nodes, although it might be known at the receiver node. The receiver might, for instance, require a minimum SNR so that the distance to the currently reached SNR represents the progress of the approach. When only a binary feedback is provided, the transmit nodes have restricted means to estimate the progress of the optimisation process. When the progress is not known, only less advanced optimisation schemes are possible. In section 7.3.3 we discuss a synchronisation scheme that greatly improves the synchronisation-performance but requires more than binary feedback information.

Another approach not feasible with binary feedback-values is an adaptive random search operator. The idea is that early in the optimisation process the reached search-point is not well synchronised with high probability. In this case, many search-points have a higher fitness-value than the current one. In particular, we expect the search-points in the neighbourhood of the current one to differ only slightly so that it is probable to reach a search-point with improved fitness-value by applying greater changes to the current search-point. When, however, later in the optimisation process the optimum is nearly reached,

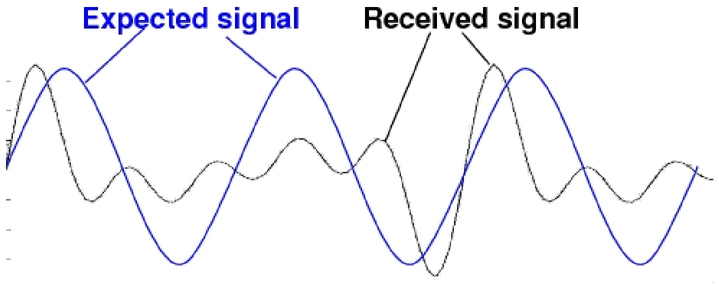


Figure 7.4: Schematic illustration of a calculated expected signal and a received superimposed sum-signal.

most other search-points have a fitness-value worse than the current one and one global optimum is expected to be near, it is more beneficial to apply only slight modifications to the search-points (i.e. to search in its direct proximity).

In [40, 41, 42] this was achieved by changing the standard deviation of the Gaussian process to alter the signal phases over the course of the optimisation. Since a progress was not reported, in these implementation the modification scheme was purely time based. With progress information, more ambiguous optimisation schemes are possible.

One solution to this problem is to transmit the (possibly normalised) output of the quality function to the network e.g. as integer value. When maximum and minimum values are known by the sensor nodes, an estimation on the optimisation progress and therefore a more exact control of the optimisation process is possible.

In order to calculate the synchronisation quality of the received signals at a receiver, various approaches are feasible. Apart from very ambiguous (and computationally expensive) methods, less accurate and computationally cheaper methods are detailed in the following.

SNR

One possible approach is to calculate the SNR of the received signal. The SNR specifies the strength of the received sum-signal above thermal noise. When the SNR increases, the transmit signals are probably better synchronised than before so that the amount of constructive vs. destructive interference is increased.

A higher SNR is therefore interpreted as an improved synchronisation of transmit signals.

Simple distance between received signals

Another possible solution is to estimate an expected sum-signal and to calculate the difference (basically the surface) between the received superimposed sum-signal and the (possibly phase shifted) expected signal (see figure 7.4).

When the surface becomes smaller, this can be interpreted as an improved phase-synchronisation. In order to estimate the expected signal when all signals are perfectly synchronised we would like to specify the signal strength of each single signal-component, the frequency and the transmit sequence.

Regarding the transmit sequence utilised for synchronisation purposes we assume that this is predefined among all nodes and known by the receiver. The same is assumed for the transmit power of each single node. Although the receiver is not able to communicate with distinct nodes separately, it is possible to estimate a median distance to the sensor network over the round trip time between the network and the receiver.

Together with the transmit power of distinct nodes, this directly leads to an estimated signal strength of a received and optimally synchronised superimposed sum-signal. The missing value, is now the count of transmitting nodes. In [98] an approach was detailed by which the count of simultaneously transmitting sensor nodes can be estimated from signal statistics of the received signal. Basically, the absolute energy received from simultaneously transmitting unsynchronised nodes is related to the number of nodes transmitting.

7.1.3 Search-space

The performance of the optimisation algorithm is heavily dependent on the underlying search-space. In particular, we would like to know if the search-space is unimodal or multimodal, i.e. if local optima exist. In this case, the optimisation approach could converge in a local optimum. We easily see that the feedback function is not unimodal. The reason is that, given the search-point corresponding to an optimum sum-signal ζ_{opt} we can state another optimum by adding the same phase-offset γ' to all carrier-signals. In particular, the fitness-function is weak multimodal which means that no local optimum exists. We detail this observation in the following.

Identical transmit frequencies

When carrier frequencies among nodes are identical, i.e.

$$e^{j(2\pi ft+\gamma_i)}; \forall i \in \{1,\ldots,n\} \tag{7.3}$$

a local optimum exists if we can identify at least one search-point $s_{\overline{\zeta}}$ for which all small phase modifications decrease the fitness-value, while some larger modifications increase it. The smallest possible modification is realised when the transmit phase is altered for exactly one carrier-signal ζ_i. Figure 7.5 illustrates that the fitness of a signal is given by the distance between the rotation angles φ_{opt} and φ_i of an optimal configuration $s_{\zeta_{\text{opt}}}$ and s_{ζ_i} as

$$\left|\cos(\varphi_{\text{opt}}) - \cos(\varphi_i)\right|. \tag{7.4}$$

A phase shift of $\delta_i \neq 0$ alters the fitness-value. For some t the fitness increases while for others it decreases. W.l.o.g. assume $(\varphi_i + \delta_i) - \varphi_{\text{opt}} < 180°$ and $\varphi_i > \varphi_{\text{opt}}$. As depicted in figure 7.5, for $\varphi_i > 180° \wedge \varphi_{\text{opt}} < 180°$ (or $\varphi_i > 360° \wedge \varphi_{\text{opt}} < 360°$, respectively) the contribution to the fitness-function $\mathcal{F}$ is zero overall, while in all other cases a phase-offset of δ_i will have either always positive or always negative impact on the fitness-value.

Compared to s_{opt} no configuration short of the optimum configuration $s_i = s_{\text{opt}}$ exists for which the distance between signal-components is increased for phase-offset δ_i regardless of the sign of δ_i (cf. figure 7.5).

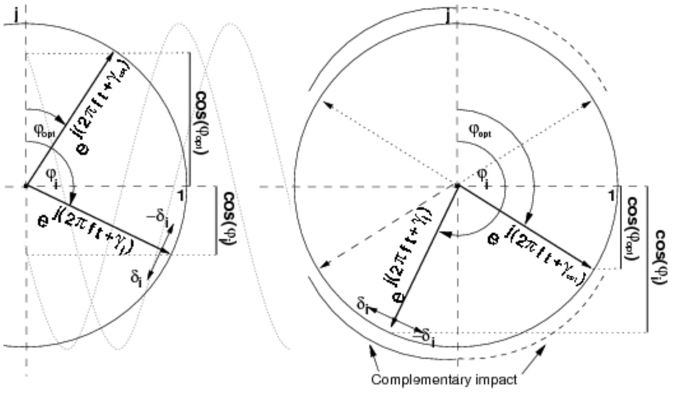


Figure 7.5: Fitness calculation of signal-components. The fitness of the superimposed sum-signal is impacted by the relative phase-offset of an optimally aligned signal and a carrier-signal i.

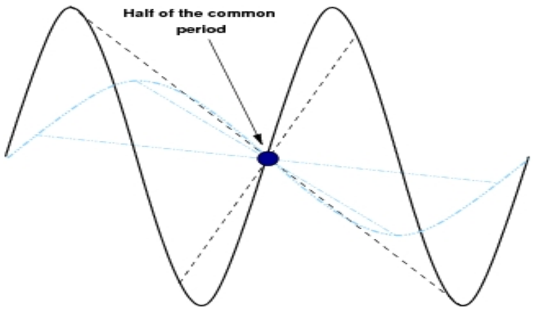


Figure 7.6: Pattern of periodic signals.

Distinct transmit frequencies

Although no local optima exist when all transmit frequencies are identical, this might differ when frequencies are also subject to change. For this configuration it is sufficient to consider the phase-offset between two signals only: An individual carrier-signal ζ_i that is to be modified in order to improve the amplitude of the sum-signal ζ_{sum} and the nearest global optimum ζ_{opt} to the superimposed sum-signal.

When signal frequencies differ, the feedback function is not affected by phase modifications only. The reason for this is that we can for every positive contribution to the fitness-function also find a negative contribution of the same amount in the common period of ζ_{opt} and ζ_i. The periodic function of the second half of the common period $\varPhi$ is a reflection of the first half of the function at the point $(\varPhi, 0)$. This property is illustrated in figure 7.6.

Consequently, for any time t relative to the common period $\varPhi$, when the relative distance between a point of the individual signal $\zeta_i(t \mod \varPhi)$ and the optimum signal $\zeta_{\text{opt}}(t \mod \varPhi)$ is decreased by an arbitrary phase-offset γ_i' applied to the carrier phase of ζ_i, we can state a value t' for at which the distance is increased by the same amount at $\zeta_i(t'$

mod Φ). This means that, in between these two configurations, the paths of the vectors are symmetrical so that for each $t \mod \Phi$ a corresponding $t' \mod \Phi$ exists with

$$\begin{aligned} & e^{j(2\pi(f_1)t \mod \Phi + \gamma_1)} - e^{j(2\pi f t \mod \Phi)} \\ = & -\left(e^{j(2\pi(f_1)t' \mod \Phi + \gamma_1)} - e^{j(2\pi f t' \mod \Phi)}\right) \end{aligned} \tag{7.5}$$

When the phase of one singal is now shifted, this property is not impacted. Also, when the frequency is changed, the length of the overall period is impacted but within the period the property described remains unchanged. The important points to note here are that

1. signal quality is not affected by phase adaptations when frequencies are unsynchronised
2. without frequency synchronisation, phase-synchronisation alone is useless in order to improve the signal quality

In summary, in both cases no local but several global optima exist in the search-space so that local random search approaches will always converge a global optimum.

7.1.4 Variation operators

Generally, for revolutionary algorithms, mutation and crossover can be applied in order to proceed to new search-points in the search-space. We discuss both in the following.

Mutation

Mutation of individuals shall apply small modifications on the individual in general so that a target individual with small distance to the current individual is more probable than individuals that are farther away.

For feedback-based phase adaptation of nodes for distributed adaptive beamforming in wireless sensor networks, mutation constitutes phase modifications of one or more carrier-signal-components. Design parameters are therefore the number of altered carrier-signal-components and the alteration method. An example for a possible mutation operator is detailed in [41].

Example 7.1.1 :

> *The authors apply normal distributed phase modifications and propose to restrict the standard deviation in order to receive a local search module. The probability that a signal phase is adapted by a source node was chosen as* 1. *This means that each node alters the phase of its carrier-signal in each iteration. Later in the optimisation this might lead to a small convergence speed and, when the optimum is near, many transmit signal-components already have a nearly optimal phase-offset and need not be modified.*
> *It was proposed in [41] to adapt the mutation speed purely by changing the standard deviation of the phase modification and not by the probability to apply phase modifications of single transmit signals.*

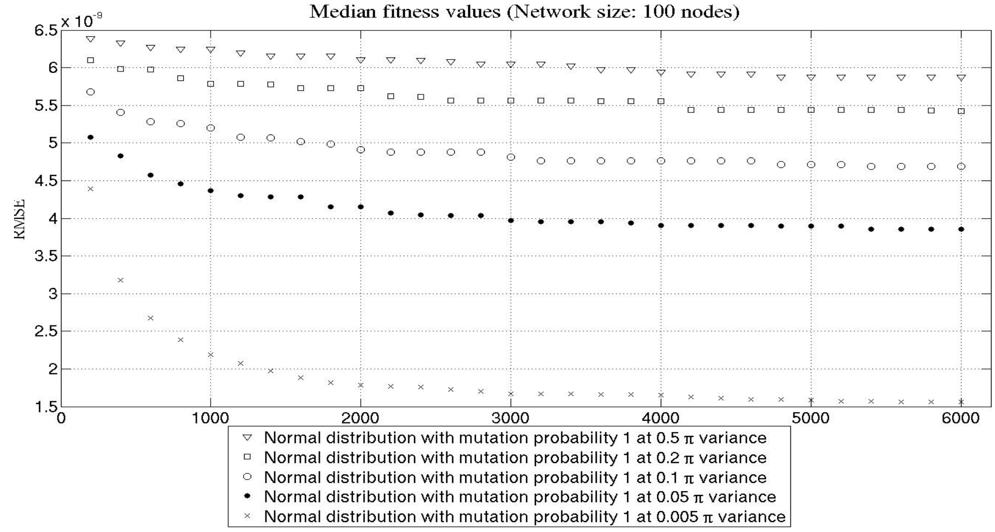


Figure 7.7: Uniform distribution of phase mutations

Figure 7.7 illustrates the impact of the standard deviation of the normal distributed phase modification on the synchronisation-performance. These results have been achieved in a Matlab-based simulation-environment in which 100 nodes are randomly distributed on a $30m \times 30m$ square area. The receiver node is located $30m$ above the centre of this area (see figure 7.8) For each one configuration, 10 simulation runs are completed to obtain the median values depicted in the figure. In the figure, the Root of the mean square error

$$RMSE = \sqrt{\sum_{t=0}^{\tau} \frac{\left(\zeta_{\text{sum}} + \zeta_{\text{noise}} - \zeta_{\text{opt}}\right)^2}{n}}. \tag{7.6}$$

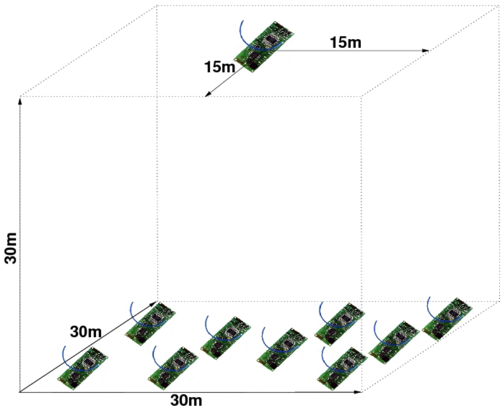
between the optimum signal ζ_{opt} and the received superimposed sum-signal ζ_{sum} is illustrated. In this formula, τ is chosen to cover several signal periods.

We can observe that better synchronisation-performances are achieved with small variances (i.e. greater probability to proceed to similar individuals) for the normal distributed phase alteration operation. This property is further discussed in section 7.2.4.

Of course, also a uniformly distributed phase-offset is possible for the mutation operator. For the uniform distributed operator we can only modify the number of nodes altering the phase-offset of their carrier phase in each iteration but not the deviation of the distribution. The uniform distribution leads to a global random search approach since all points in the search-space have a positive probability to be elected as next search-point.

Since the uniform distribution is better suited for our individual representation, we will assume a uniformly distributed phase alteration procedure in our theoretical analysis in section 7.2. In section 7.2.4 we will observe that the performance of uniform and normal distributed phase modification operators is similar.

For both distributions it is of course possible to restrict the neighbourhood for the search by applying sharp neighbourhood boundaries. This approach leads to a local



Property	Value
Node distribution area	$30m \times 30m$
Location of the receiver	$(15m, 15m, 30m)$
Mobility	stationary nodes
Base band frequency	$f_{base} = 2.4$ GHz
Transmission power of nodes	$P_{tx} = 1$ mW
Gain of the transmit antenna	$G_{tx} = 0$ dB
Gain of the receive antenna	$G_{rx} = 0$ dB
Iterations per simulations	6000
Identical simulation runs	10
Random noise-power [64]	-103 dBm
Pathloss calculation (P_{rx})	$P_{tx}\left(\frac{\lambda}{2\pi d}\right)^2 G_{tx}G_{rx}$

Figure 7.8: Configuration of the simulation-environment. P_{rx} is the the received signal power, d is the distance between transmitter and receiver and λ is the wavelength of the signal

random search mechanism and is detailed in section 7.3.2. Figure 7.9 illustrates the effect of various distributions for the phase-offset on the neighbourhood-size.

Crossover

Until now, we considered the basic scenario to be solved by a $(1+1)$-EA. This approach is straightforward in that it considered only one individual at a time. As an individual is constituted by one configuration (phase- and frequency changes applied to the carrier-signal of transmit nodes) of all signal-components, multiple individuals are achieved artificially at most. Also, when only mutation is applied, a multi-start strategy seems to be equally beneficial as an increased population size. With population size 1, however, no recombination (crossover) is possible.

In general, two approaches might enable crossover in collaborative transmission scenarios.

1. Simultaneous transmission of distinct transmit signals by source nodes
2. Time-shifted transmission of transmit signals related to distinct individuals

The first solution, however, would at least require more ambiguous source nodes, and can therefore be disregarded in wireless sensor networks. For the second solution and population size of $\mu > 1$, one iteration of the algorithm is then extended to μ transmission periods each of which is associated to one individual. This scenario has not yet been considered in the literature.

7.1.5 Discussion

Does collaborative transmission always converge? In [41] it was proved that this is the case when all transmit signals are of identical frequency. We have discussed the problem-

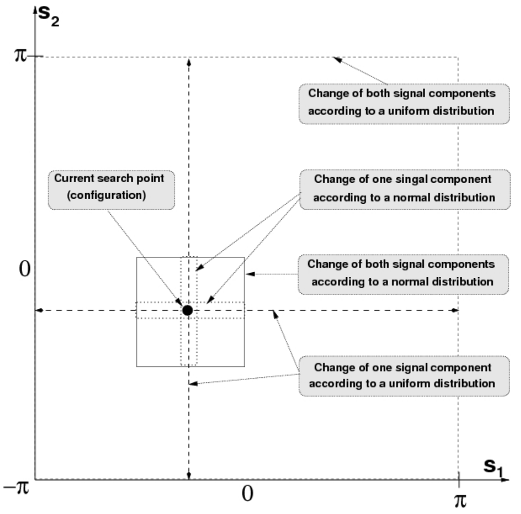

s2
π
0
−π
0
π
s1
Change of both signal components according to a uniform distribution
Current search point (configuration)
Change of one singal component according to a normal distribution
Change of both signal components according to a normal distribution
Change of one signal component according to a uniform distribution

Figure 7.9: A point in the search-space (configuration of transmit nodes) spanned by the phase-offsets of the carrier-signals s_1 and s_2

scenario thoroughly regarding several aspects so that we can easily come to the same result and even disregard the frequency constraint.

As observed in section 5.3.1 that no local optima exist in the search-space. Also, the random uniform and normal distributed search methods are not restricted to any local neighbourhood in the search-space (although more similar points have greater probability to be reached). Furthermore, the algorithm never accepts search-points with a worse fitness-value. These results together are already sufficient to see that one of the global optima is reached with probability 1. Even if the mutation operator were a local search operator, an optimum would be reached with probability 1 since the fitness-function is only weak multimodal, i.e. no local optima exist.

In the following section we will derive the expected time until a global optimum is reached for this phase-synchronisation approach.

7.2 Analysis of the convergence time of 1-bit feedback-based distributed adaptive transmit beamforming in wireless sensor networks

We analyse the process of distributed adaptive beamforming in wireless sensor networks as described above and assume that each one of the n nodes in step 1) decides with probability $\frac{1}{n}$ to change the phase of its carrier-signal uniformly at random in the interval $[0, 2\pi]$. On obtaining the feedback of the receiver, nodes that recently updated the phase of their carrier-signal either sustain this decision or reverse it, depending on whether the feedback has improved or not. A feedback function $\mathcal{F} : \zeta^*_{\text{sum}} \to \mathbb{R}$ maps the superimposed received carrier-signal

$$\zeta_{\text{sum}} = \Re\left(m(t)e^{j2\pi f_c t}\sum_{i=1}^{n}\text{RSS}_i e^{j(\gamma_i+\phi_i+\psi_i)}\right) \tag{7.7}$$

to a real-valued fitness score. A possible feedback function that is proportional to the distance between an observed superimposed carrier ζ_{sum} and an optimum sum carrier-signal

$$\zeta_{\text{opt}} = \Re\left(m(t)\text{RSS}_{\text{opt}}e^{j(2\pi f_c t+\gamma_{\text{opt}}+\phi_{\text{opt}}+\psi_{\text{opt}})}\right) \tag{7.8}$$

is $\mathcal{F}(\zeta_{\text{sum}}) = \int_{t=0}^{2\pi}\left|\zeta_{\text{sum}} - \zeta_{\text{opt}}\right|$. Since this function can be mapped onto other fitness measures as, for instance, the signal to noise ratio (SNR) or the received signal strength (RSS), the following discussion remains valid for these fitness measures. While the multimodality of this fitness-function is straightforward, observe that it is also weak multimodal so that no local optima exist (The weak multimodality of the fitness-function is shown in section 5.3.1).

Distributed adaptive beamforming in wireless sensor networks constitutes a search problem. The search-space $\mathcal{S}$ is given by the set of possible combinations of phase and frequency-offsets γ_i and f_i for all n carrier-signals. A global optimum is a configuration of individual carrier phases that result in identical phase and frequency-offset of all received direct signal-components. For the analysis, we assume that the optimisation aim is to achieve for an arbitrary k a maximum relative phase-offset of $\frac{2\pi}{k}$ between any two carrier-signals. We therefore divide the phase space for a single carrier-signal into k intervals of width $\frac{2\pi}{k}$.

For a specific superimposed carrier-signal ζ at a receiver we model the corresponding search-point $s_\zeta = (\Gamma_t, F_t)_\zeta \in \mathcal{S}$ at iteration t by a specific combination of phase and frequency-offsets with $\Gamma_t = (\gamma_{t,1},\dots,\gamma_{t,n})$ and $F_t = (f_{t,1},\dots,f_{t,n})$. In order to respect neighbourhood similarities we represent search-points as Gray encoded binary strings $s_\zeta \in \mathbb{B}^{n\cdot\log(k)}$ so that similar search-points have a small hamming-distance [99]. A search-point is then composed from n sections of $\log(k)$ bits each. Every block of length $\log(k)$ describes one of the k intervals for the phase-offset of one carrier-signal. For the analysis, we assume that the frequency-offset f_i is zero for all carriers. The optimisation problem is denoted as $\mathcal{P}$ and $T_{\mathcal{P}}$ describes the count of iterations required to reach one of the optimum values for the problem $\mathcal{P}$.

7.2.1 An upper bound on the expected optimisation-time

The value of the fitness-function increases with the number of carrier-signals ζ_i that share the same interval for their phase-offset γ_i at the receiver. We can roughly divide the values of the fitness-function into $n\cdot\log(k)$ partitions $L_1,\dots,L_{n\cdot\log(k)}$ depending on the number of bits in the individual representation that are identical with an individual global optimum. For each single transmitter, the probability to adapt its phase to one specific interval is $\frac{1}{k}$. The probability to increase the fitness-value so that at least the next partition is reached is then at least

$$\frac{1}{k}\cdot\frac{1}{n\cdot\log(k)} \tag{7.9}$$

since each carrier-signal ζ_i is altered with probability $\frac{1}{n \cdot \log(k)}$ and the probability to reach any particular of the partitions that would increase the fitness-value is $\frac{1}{k}$. In partition i, a total of

$$\begin{pmatrix} n \cdot \log(k) - i \\ 1 \end{pmatrix} = n \cdot \log(k) - i \tag{7.10}$$

carrier-signals each suffice to improve the fitness-value with probability $\frac{1}{n \cdot \log(k)} \cdot \frac{1}{k}$. We therefore require that at least one of the not synchronised carrier-signals is correctly altered in phase while all other $n - 1$ signals remain unchanged. This happens with probability

$$\begin{aligned} & \begin{pmatrix} n \cdot \log(k) - i \\ 1 \end{pmatrix} \cdot \frac{1}{n \cdot \log(k)} \cdot \left(1 - \frac{1}{n \cdot \log(k)}\right)^{n \cdot \log(k) - 1} \\ = & \left(\frac{n \cdot \log(k) - i}{n \cdot \log(k)}\right) \cdot \left(1 - \frac{1}{n \cdot \log(k)}\right)^{n \cdot \log(k) - 1}. \end{aligned} \tag{7.11}$$

Since

$$\left(1 - \frac{1}{n}\right)^{n} < \frac{1}{e} < \left(1 - \frac{1}{n}\right)^{n-1} \tag{7.12}$$

We obtain the probability $P[L_i]$ that L_i is left and a partition j with $j > i$ is reached as

$$P[L_i] \geq \frac{n \cdot \log(k) - i}{n \cdot \log(k) \cdot e}. \tag{7.13}$$

The expected number of iterations to change the layer is bound from above by $P[L_i]^{-1}$. We consequently obtain the overall expected optimisation-time as

$$\begin{aligned} E[T_{\mathcal{P}}] & \leq \sum_{i=0}^{n \cdot \log(k) - 1} \frac{e \cdot n \cdot \log(k)}{n \cdot \log(k) - i} \\ & = e \cdot n \cdot \log(k) \cdot \sum_{i=1}^{n \cdot \log(k)} \frac{1}{i} \\ & < e \cdot n \cdot \log(k) \cdot (\ln(n \cdot \log(k)) + 1) \\ & = \mathcal{O}\left(n \cdot \log\left(n \cdot \log(k) + k\right)\right). \end{aligned} \tag{7.14}$$

7.2.2 A lower bound on the expected optimisation-time

After the initialisation, the phases of the carrier-signals are identically and independently distributed. Consequently for a superimposed received sum-signal ζ, each bit in the binary string s_ζ that represents the corresponding search-point has an equal probability to be 1 or 0. The probability to start from a search-point s_ζ with hamming-distance $h(s_{\text{opt}}, s_\zeta)$ not larger than $l \in \mathbb{N}$; $l \ll n \cdot \log(k)$ to one of the global optima s_{opt} directly after the random initialisation is at most

$$\begin{aligned} P[h(s_{\text{opt}}, s_\zeta) \leq l] &= \sum_{i=0}^{l} \left(\begin{array}{c} n \cdot \log(k) \\ n \cdot \log(k) - i \end{array} \right) \cdot \frac{k}{2^{n \cdot \log(k) - i}} \\ &\leq \frac{(n \cdot \log(k))^{l+2}}{2^{n \cdot \log(k) - l}} \end{aligned}$$

In this formula,

$$\left(\begin{array}{c} n \cdot \log(k) \\ n \cdot \log(k) - i \end{array} \right) \tag{7.15}$$

is the count of possible configurations with i bit-errors to a given global optimum, $\frac{1}{2^{n \cdot \log(k) - i}}$ represents the probability for all these bits to be correct and k is the count of global optima.

This means that with high probability (w.h.p.) the hamming-distance to the nearest global optimum is at least l. We will use the method of the expected progress to calculate a lower bound on the optimisation-time. The general idea is the following.

Let (s_ζ, t) denote the situation that search-point s_ζ was achieved after t iterations of the algorithm. We assume a progress measure $\Lambda : \mathbb{B}^{n \cdot \log(k)} \to \mathbb{R}_0^+$ such that $\Lambda(s_\zeta, t) < \Delta$ represents the case that a global optimum was not found in the first t iterations. For every $t \in \mathbb{N}$ we have

$$\begin{aligned} E[T_{\mathcal{P}}] &\geq t \cdot P[T_{\mathcal{P}} > t] \\ &= t \cdot P[\Lambda(s_\zeta, t) < \Delta] \\ &= t \cdot (1 - P[\Lambda(s_\zeta, t) \geq \Delta]). \end{aligned} \tag{7.16}$$

With the help of the Markov-inequality we obtain

$$P[\Lambda(s_\zeta, t) \geq \Delta] \leq \frac{E[\Lambda(s_\zeta, t)]}{\Delta} \tag{7.17}$$

and therefore

$$E[T_{\mathcal{P}}] \geq t \cdot \left(1 - \frac{E[\Lambda(s_\zeta, t)]}{\Delta}\right). \tag{7.18}$$

This means that we can obtain a lower bound on the optimisation-time by providing the expected progress after t iterations. The probability for l bits to correctly flip is at most

$$\begin{aligned} &\left(1 - \frac{1}{n \cdot \log(k)}\right)^{n \cdot \log(k) - l} \cdot \left(\frac{1}{n \cdot \log(k)}\right)^{l} \\ \leq\ &\frac{1}{(n \cdot \log(k))^{l}}. \end{aligned} \tag{7.19}$$

In this formula, $\left(1 - \frac{1}{n \cdot \log(k)}\right)^{n \cdot \log(k) - l}$ describes the probability that all 'correct' bits do not flip while the remaining l bits mutate with probability $\left(\frac{1}{n \cdot \log(k)}\right)^{l}$. The expected progress in one iteration is therefore

$$\begin{aligned} E[\Lambda(s_\zeta,t),\Lambda(s_{\zeta'},t+1)] &\leq \sum_{i=1}^{l}\frac{i}{(n\cdot\log(k))^i} \\ &< \frac{2}{n\cdot\log(k)} \end{aligned} \tag{7.20}$$

and the expected progress in t iterations is consequently not greater than $\frac{2t}{n\cdot\log(k)}$.

When we choose $t=\frac{n\cdot\log(k)\cdot\Delta}{4}-1$, the double of the expected progress is still smaller than Δ. With the Markov inequality we can show that this progress is not achieved with probability $\frac{1}{2}$. Altogether we conclude that the expected optimisation-time is bound from below by

$$\begin{aligned} E[T_{\mathcal{P}}] &\geq t\cdot\left(1-\frac{E[\Lambda(s_\zeta,t)]}{\Delta}\right) \\ &\geq \frac{n\cdot\log(k)\cdot\Delta}{4}\cdot\left(1-\frac{\frac{2\cdot n\cdot\log(k)}{4\cdot n\cdot\log(k)}\cdot\Delta}{\Delta}\right) \\ &= \Omega(n\cdot\log(k)\cdot\Delta) \end{aligned} \tag{7.21}$$

With $\Delta=\log(n\cdot\log(k))$ we obtain a lower bound in the same order as the upper bound derived in section 7.2.1 and consequently an asymptotically sharp bound of

$$E[T_{\mathcal{P}}]=\Theta\left(n\cdot\log\left(n\cdot\log\left(k\right)+k\right)\right). \tag{7.22}$$

7.2.3 Simulation and experimental results for the basic scenario

In the analytic consideration in section 7.2 we could derive asymptotic bounds on the expected synchronisation-time of a (1+1) evolutionary algorithm with standard bit mutation and mutation probability $\frac{1}{n}$ in the scenario of distributed adaptive transmit beamforming. In the analysis we had to abstract from several physical properties. These are, for instance, the noise of the receiver, multipath propagation as well as distinct transmit and receiver powers of nodes or phase and frequency instability of local oscillators.

The following sections detail the performance of a $(1+1)$ evolutionary algorithm for distributed adaptive beamforming achieved in simulations and experimental settings.

Simulation results

We have implemented the scenario of distributed adaptive beamforming in Matlab to obtain simulation results for greater scale sensor networks. In these simulations, 100 transmit nodes are placed uniformly at random on a $30m\times30m$ square area. The receiver is located $30m$ above the centre of this area. Receiver and transmit nodes are stationary. Simulation parameters are summarised in Table 7.1.

Property	Value
Node distribution area	$30m \times 30m$
Location of the receiver	$(15m, 15m, 30m)$
Mobility	stationary nodes
Base band frequency	$f_{base} = 2.4$ GHz
Transmission power of nodes	$P_{tx} = 1$ mW
Gain of the transmit antenna	$G_{tx} = 0$ dB
Gain of the receive antenna	$G_{rx} = 0$ dB
Iterations per simulations	6000
Identical simulation runs	10
Random noise-power [64]	-103 dBm
Pathloss calculation (P_{rx})	$P_{tx}\left(\frac{\lambda}{2\pi d}\right)^2 G_{tx}G_{rx}$

Table 7.1: Configuration of the simulations. P_{rx} is the the received signal power, d is the distance between transmitter and receiver and λ is the wavelength of the signal

Frequency and phase stability are considered perfect. We derived the median and standard deviation from 10 simulation runs. One iteration consists of the nodes transmitting, feedback computation, feedback transmission and feedback interpretation at the network side. It is possible to perform these steps within few singal periods so that the time consumed for a synchronisation of 6000 iterations is in the order of milliseconds for a base band signal frequency of 2.4 GHz. Signal quality is measured by the RMSE of the received signal to an expected optimum signal.

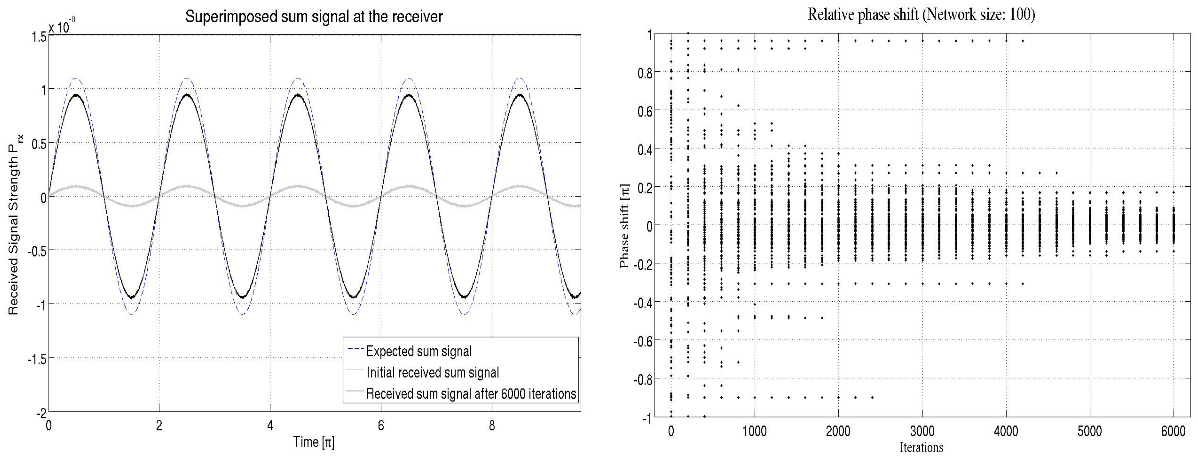
$$RMSE = \sqrt{\sum_{t=0}^{\tau} \frac{\left(\sum_{i=1}^{n} s_i + s_{noise} - s_{opt}\right)^2}{n}} \tag{7.23}$$

In this formula, τ was chosen to cover several signal periods. The terms $s_i = RSS_i \Re\left(m(t)e^{j(2\pi(f+f_i)t+\overline{\gamma_i})}\right)$ and $s_{opt} = nRSS_{sum}\Re\left(m(t)e^{j(2\pi f_{opt}t+\gamma_{opt})}\right)$ represent the i-th signal-component of the received sum-signal and the expected optimum signal, respectively. The optimum signal is calculated as the perfectly aligned and properly phase shifted received sum-signal from all transmit sources. For the optimum signal, noise is disregarded.

Figure 7.10(a) depicts the optimum carrier-signal, the initial received sum-signal and the synchronised carrier after 6000 iterations when carrier phases are altered with probability $\frac{1}{n}$ in each iteration according to a uniform distribution. In figure 7.10(b), the phase-offset of received signal-components for an exemplary simulation run with the same parameters are illustrated. We observe that after 6000 iterations about 98% of all carrier-signals converge to a relative phase-offset of about +/- 0.1π. The median of all variances of the phase-offsets for simulation runs with this configuration is 0.2301 after 6000 iterations.

Experimental results

We have utilised USRP software radios (http://www.ettus.com) to model a sensor network capable of distributed adaptive transmit beamforming. Figure 7.11(b) depicts an exper-



(a) Received sum-signal from 100 transmit nodes without synchronisation and after 6000 iterations

(b) Evolution of the phase adaptation process

Figure 7.10: Simulation results for a simulation with 100 transmit over 6000 iterations of the random optimisation approach to distributed adaptive beamforming in wireless sensor networks.

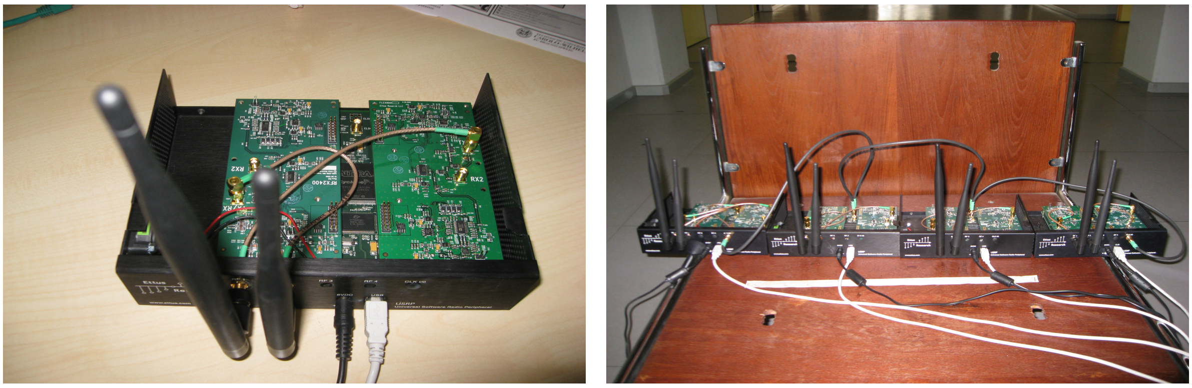
imental setting for this scenario. The software radios are controlled via the GNU radio framework (http://gnuradio.org). GnuRadio is a software to control the USRP software radios from a personal computer. It enables processing, analysis and visualisation of RF-signals from standard hardware (cf. figure 7.12(a)) The signal flow graph can be assembled graphically (cf. figure 7.12(b)).

The transmitter and receiver modules implement the feedback-based distributed adaptive beamforming. For the superimposed transmit channel and for the feedback channel we used widely separated frequencies so that the feedback did not impact the synchronisation-performance. Two experiments have been conducted with low and high RF-transmit frequencies of 27MHz and 2.4GHz, respectively. Table 7.2.3 summarises the configuration and results of both experiments.

After 10 experiments at an RF-transmit frequency of 27MHz we achieved a median gain in the received signal strength of 3.72dB for three independent transmit nodes after 200 iterations. In 14 experiments with 4 independent nodes that transmit at 2.4GHz the achieved median gain of the received RF-sum-signal was 2.19dB after 500 iterations. In figure 7.13(a) the synchronisation-performance is depicted over 600 iterations of the synchronisation algorithm.

While the amplitude achieved in these synchronisations differs for various positions of the receiver nodes relative to the transmit nodes, figure 7.13(b) shows that the general synchronisation run is similar in all cases.

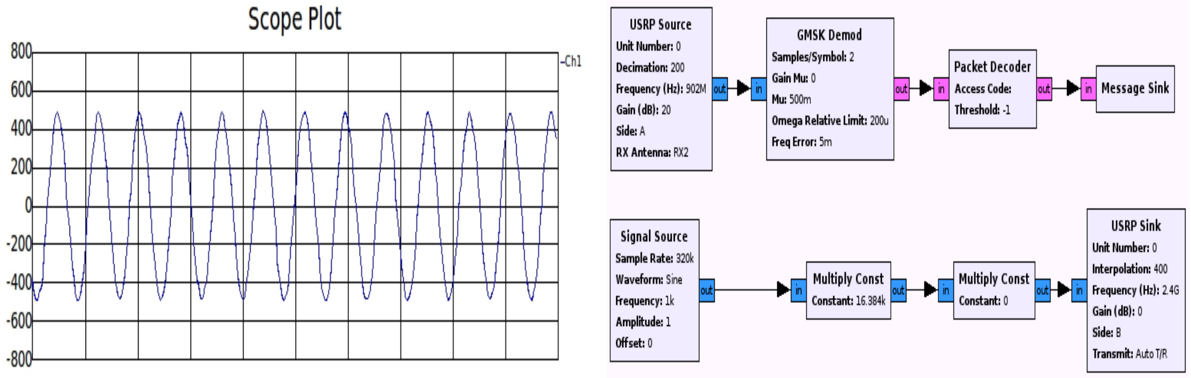
The transmit nodes have been synchronised in their clock but no other communication or synchronisation between nodes was allowed. The clock synchronisation is necessary to ensure that the frequency of transmit nodes is identical. Otherwise, as derived in section 5.3.1 a phase-synchronisation is not feasible since signals would quickly run out



(a) A single USRP software radio with two transceiver boards

(b) Experimental setting with four clock synchronised USRP software radios

Figure 7.11: An experimental setting for a distributed adaptive beamforming scenario with USRP software radios



USRP Source
Unit Number: 0
Decimation: 200
Frequency (Hz): 902M
Gain (dB): 20
Side: A
RX Antenna: RX2

GMSK Demod
Samples/Symbol: 2
Gain Mu: 0
Mu: 500m
Omega Relative Limit: 200u
Freq Error: 5m

Packet Decoder
Access Code:
Threshold: -1

Message Sink

Signal Source
Sample Rate: 320k
Waveform: Sine
Frequency: 1k
Amplitude: 1
Offset: 0

Multiply Const
Constant: 16.384k

Multiply Const
Constant: 0

USRP Sink
Unit Number: 0
Interpolation: 400
Frequency (Hz): 2.4G
Gain (dB): 0
Side: B
Transmit: Auto T/R

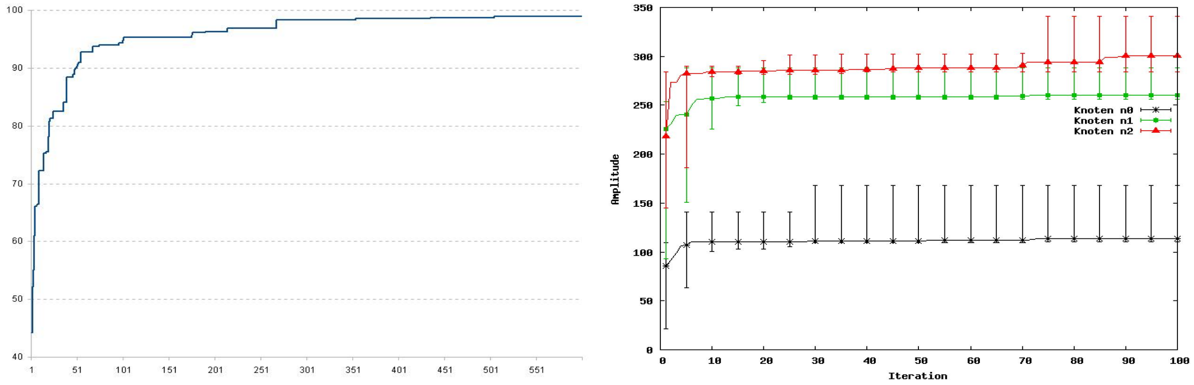
(a) Scope plot of Gnuradio while synchronising four USRP software radios at 2.4 GHz

(b) A GnuRadio flow graph

Figure 7.12: GnuRadio is utilised to control the USRP software radios

	Experiment 1	Experiment 2
Sender	4	3
Mobility	stationary	stationary
Distance to receiver [m]	≈ 0.75	≈ 4
Separation of TX antennas [m]	≈ 0.21	≈ 0.3
Transmit RF-Frequency [MHz]	$f_{TX} = 2400$	$f_{TX} = 27$
Receive RF-Frequency [MHz]	$f_{RX} = 902$	$f_{RX} = 902$
Gain of receive antenna [dBi]	$G_{RX} = 3$	$G_{RX} = 3$
Gain of transmit antenna [dBi]	$G_{TX} = 3$	$G_{TX} = 1.5$
Iterations per experiment	500	200
Identical experiments	14	10
Median gain (P_{RX}) [dB]	2.19	3.72

Table 7.2: Experimental results of software radio instrumentation



(a) Experimental measurement of distributed adaptive transmit beamforming by USRP software radios

(b) Optimisation performance for nodes at distinct relative locations to a receiver

Figure 7.13: Performance of distributed adaptive beamforming in an experimental setting with USRP software radios

of phase. In future implementation GPS for the clock synchronisation of nodes could be utilised.

7.2.4 Impact of distinct parameter configurations

In distributed adaptive beamforming in wireless sensor networks an individual node i in a network of size n cooperates with other nodes to reach a distant receiver by superimposing its individual carrier-signal $\Re\left(m(t)e^{j(2\pi(f+f_i)t+\gamma_i)}\right)$ with carrier-signals transmitted from other nodes. The approach constitutes a closed-loop feedback technique that is attained in an iterative process. During each iteration, following a random distribution, each transmit node i adds random phase and frequency-offsets γ_i', f_i' to its carrier phase γ_i and frequency f_i. According to the random distribution it is possible that all, some or no node adjusts the phase or frequency of its carrier-signal. The nodes then transmit to the destination simultaneously. From this received superimposed sum-signal $\Re\left(RSS_{sum}m(t)\sum_{i=1}^{n}e^{j(2\pi(f+f_i)t+\overline{\gamma_i})}\right)$ the receiver estimates the level of phase-synchronisation achieved (e.g. by SNR or comparison to an expected signal). At the end of each iteration, the receiver broadcasts this estimation as a channel quality indicator to the network. Nodes then sustain their (adapted) carrier phases $\gamma_i + \gamma_i'$ and frequencies $f_i + f_i'$ when the feedback has improved to the feedback obtained during the last iteration or else reverse them.

These steps are repeated until a stop criteria is met. Possible stop criteria are, for example, the maximum iteration count or sufficient quality of the estimated channel.

The exact implementation of the phase alteration process in each iteration impacts the performance of the optimisation approach. Generally, two parameters are of importance. One is the number of nodes altering their carrier phase and frequency in each iteration and the other is the probability distribution applied on the alteration of phases and frequencies of one carrier-signal.

When we illustrate the search-space of the algorithm with respect to these parameters we observe that they effectively impact the neighbourhood-size of the search approach (see figure 7.14). When the probability to alter the carrier phase of a single node is low, few carrier-signals are altered in one iteration. In the figure this might translate to the situation that the phase-offset of only one carrier-signal is altered. The new search-point then differs from the recent one in only one dimension. When a uniform distribution is implemented, all points along this dimension are equally probable while for a normal distribution the most probable points are near to the recent search-point as specified by the variance of the probability distribution utilised to alter phase and frequency-offsets. When, however, the mutation probability is increased, so that more often several carrier-signals (in the figure, for example, both carrier-signals s_1 and s_2) are altered, the new search-point is drawn from a region in the search-space. In the case of the uniform distribution the whole search-space might constitute this region. Consequently, a smaller probability for nodes to alter their phase and frequency or a smaller variance for the random process to alter these parameters increases the probability to draw a search-point that is near to the recent one.

We expect that a moderate size of the neighbourhood (i.e. probability to alter phase or frequency of one carrier-signal and variance for the random process to alter these values)

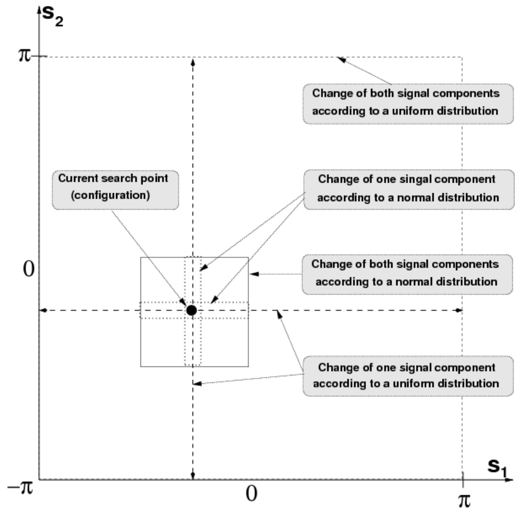

s2
π
Change of both signal components according to a uniform distribution
Current search point (configuration)
Change of one singal component according to a normal distribution
0
Change of both signal components according to a normal distribution
Change of one signal component according to a uniform distribution
s1
−π
0
π

Figure 7.14: A point in the search-space (configuration of transmit nodes) spanned by the phase-offsets of the carrier-signals s_1 and s_2

is beneficial at the start of the synchronisation. The initial search-point or the initial synchronisation of carrier phases is typically far away from the optimum. Therefore, many search-points have a higher fitness-value (are better synchronised) than the initial one. Consequently, since the probability to improve the fitness-value is high, it might be beneficial to increase the neighbourhood-size so that also bigger improvements are possible. Later in the optimisation process, however, when the optimum is near, most search-points reduce the fitness-value so that the probability to increase it is best for search-points drawn from a small neighbourhood.

In this section we consider various different parameter configurations for feedback-based distributed adaptive transmit beamforming in wireless sensor networks. These are the probability $P_{mut,i}$ for individual nodes i to alter the phase-offset as well as the distribution $P_{dist,i}$ with which the phase is altered. We distinguish between a uniformly distributed and a normal distributed offset. Both are considered in the following.

Uniformly distributed phase-offset

In the first set of simulations we consider the impact of the mutation probability on the synchronisation-performance when new transmit phase-offsets are drawn according to a uniform distribution.

When the average number of carrier-signals that are modified in one iteration is altered, this affects the performance of the optimisation approach, as the stepwidth of the random process is changed. We are interested in the optimum percentage of nodes that alter their phase per iteration (mutation probability) when the phase-alteration for each mutating node is drawn according to a uniform distribution. Figure 7.2.4 depicts simulation results for several mutation probabilities.

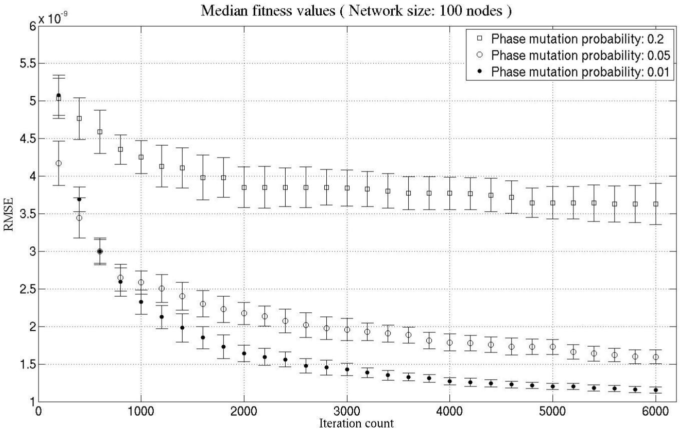


Figure 7.15: Performance of distributed adaptive beamforming in a wireless sensor network of 100 nodes and normal and uniform probability distributions on the phase mutation probability. Uniform distribution of phase mutations.

A mutation probability of p represents the case that on average $p \cdot n$ out of n transmit signal-components mutate (randomly alter their phase-offset) in one iteration.

We observe that with a low mutation probability the overall performance of the synchronisation process is improved. At the start of the optimisation, however, higher mutation probabilities are more beneficial. This accounts for the fact that initially many configurations exist that would improve the fitness-value as the initial point is not well synchronised with high probability (cf. section 7.2).

Since a mutation probability of 0.01 translates to 1 node to adapt the phase-offset of its carrier-signal per iteration on average, smaller mutation probabilities are not beneficial.

Normal distributed phase-offset

We also considered a normal distribution for the phase alteration as it was applied in [42, 16, 41, 18] and study the impact of the mutation probability and the variance utilised.

Figure 7.16 shows the performance of this approach with several values for the mutation variance applied to the phase mutation process when all nodes mutate in every iteration (mutation probability of 1). This configuration is identical to the simulations conducted in [42, 16, 41, 18]. For ease of presentation, error bars are omitted in this figure.

We observe that a smaller variance is beneficial for the performance of the synchronisation process. The reason for this is again the reduced neighbourhood-size of the search method at smaller variances. In particular, by means of the mutation variance the neighbourhood-size can be adapted to an optimum for a given mutation probability.

However, when both, the variance and the mutation probability are small, the simulation performance deteriorates as the improvements in the fitness space are minor. Figure 7.17 depicts this property for various combinations of p_γ and σ_γ^2.

We observe that the mutation performance can be improved by altering the mutation

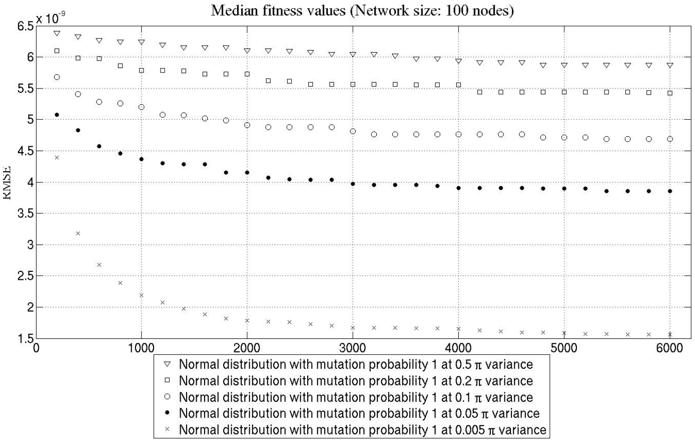
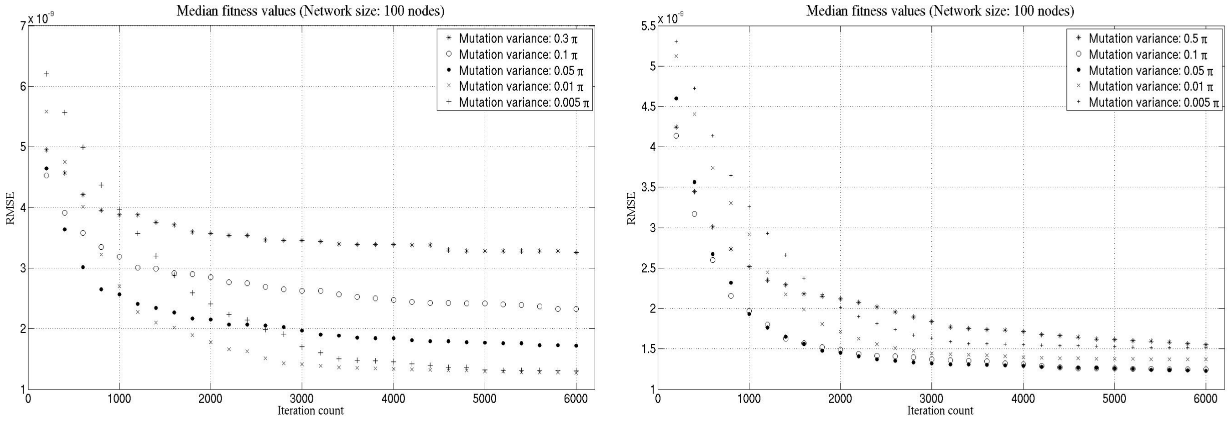


Figure 7.16: Normal distribution of phase mutations with mutation probability 1



(a) Normal distribution of phase mutations with mutation probability 0.2 and various values for the mutation variance σ_γ^2

(b) Normal distribution of phase mutations with mutation probability 0.05 and various values for the mutation variance σ_γ^2

Figure 7.17: Normal distribution of phase mutations with a fixed mutation probability and various values for the mutation variance σ_γ^2

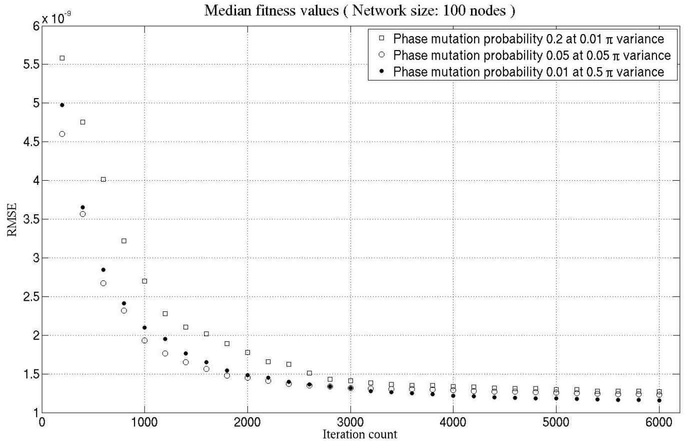


Figure 7.18: Performance of distributed adaptive beamforming in a wireless sensor network of 100 nodes and normal and uniform probability distributions on the phase mutation probability. Normal distribution of phase mutations.

variance only to a particular extent. When the variance is decreased further, the overall synchronisation-performance deteriorates.

We approximated the optimum variance for several mutation probabilities. Generally, a higher mutation probability necessitates a smaller variance to achieve the optimum performance. Figure 7.18 depicts the performance for mutation probabilities of 0.2, 0.05 and 0.01 and near optimum variance, respectively.

For these configurations the performance is similar to the performance of the uniform distribution with mutation probability 0.01.

Discussion

Summarising, we conclude that the best performance was achieved for small modifications of a search-point in each iteration. This is achieved for the uniformly distributed phase-offset by a small probability to alter the phase-offset of an individual node. For the normal distributed process, small steps in the search-space can be achieved either by a small variance or by a small probability to alter the phase-offset of an individual node.

This optimisation performance can be further improved at the start of the synchronisation process when a moderate mutation probability is implemented that is gradually adapted in the course of the synchronisation process.

For the normal distribution the performance can be improved by adopting the variance of the process.

Overall, results for near optimum configurations are similar for normal and uniform distributed phase alteration processes. A straightforward implementation would, for instance, constitute an adaptive phase mutation process with uniform distribution.

Since in [100] the search-space of distributed adaptive beamforming was classified as weak unimodal, we expect a nearest neighbour search approach to be best suited to achieve

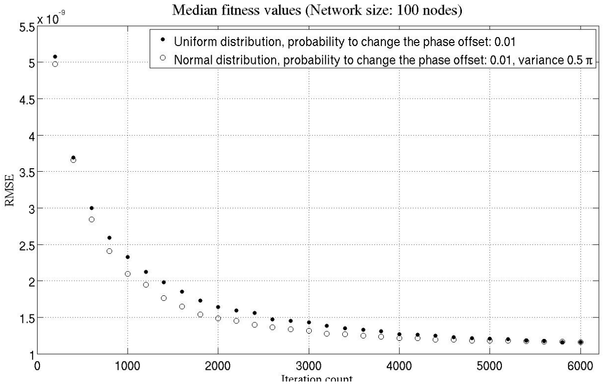


Figure 7.19: Performance of normal and uniform distributions for a network size of 100 nodes and $p_\gamma = 0.01, \sigma_\gamma^2 = 0.5\pi$.

fast synchronisation. We implemented and tested a local random search heuristic for this scenario with various neighbourhood-sizes. The implementation differs from the normal distribution mainly since it utilises a fixed size neighbourhood instead of a variance on the phase alteration process. Figure 7.19 shows that the performance is similar to the implementation with uniformly distributed phase alteration probabilities. We therefore conclude that the global random search approaches utilised are already near optimal solutions for distributed adaptive beamforming in wireless sensor networks.

Distributed adaptive beamforming in wireless sensor networks has been studied in the literature according to various random phase alteration processes. The authors in [15, 16, 17] report good results when the probability p_γ to alter the phase of a single carrier-signal in one iteration is 1 for all nodes and the phase-offset is chosen according to a normal distribution. The variance σ_γ^2 applied is not reported. In [13, 14] p_γ was set to $\frac{1}{n}$ for each one of the n nodes while the phase is altered according to a uniform distribution.

For both, uniform and normal distributed processes, we consider several values for p_γ and σ_γ^2. Generally, we achieved good performance when modifications in one iteration were small. For the uniform distribution this translates to $p_\gamma = \frac{1}{n}$. For the normal distribution, good results are achieved when σ_γ^2 and p_γ are balanced so that the modification to the overall sum-signal is small. With increasing p_γ good results are achieved with decreasing σ_γ^2. Figure 7.19 depicts the simulation results for $p_\gamma = \frac{1}{n}$ and $\sigma_\gamma^2 = 0.5\pi$

The figure shows the median RMSE value achieved in 10 simulations by normal and uniform distributed processes over the course of 600 iterations. For ease of presentation, error bars are omitted in this figure. However, the standard deviation is low for both processes (the standard deviation of this normal distributed process is depicted in figure 7.29(b)).

The normal distributed process has a slightly improved synchronisation-performance. The optimum fitness-value reached is, however, identical to the uniformly distributed process.

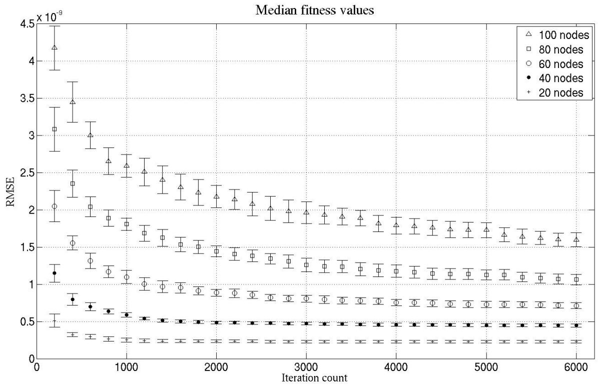


Figure 7.20: The synchronisation-performance for various network sizes in a uniformly distributed process with $p_\gamma = 0.05$.

7.2.5 Impact of environmental parameters

Apart from parameters of the optimisation algorithm, also the environmental setting might impact the synchronisation-performance of distributed adaptive beamforming in wireless sensor networks. We consider the impact of the network size and the distance between a receiver and the network.

Impact of the network size

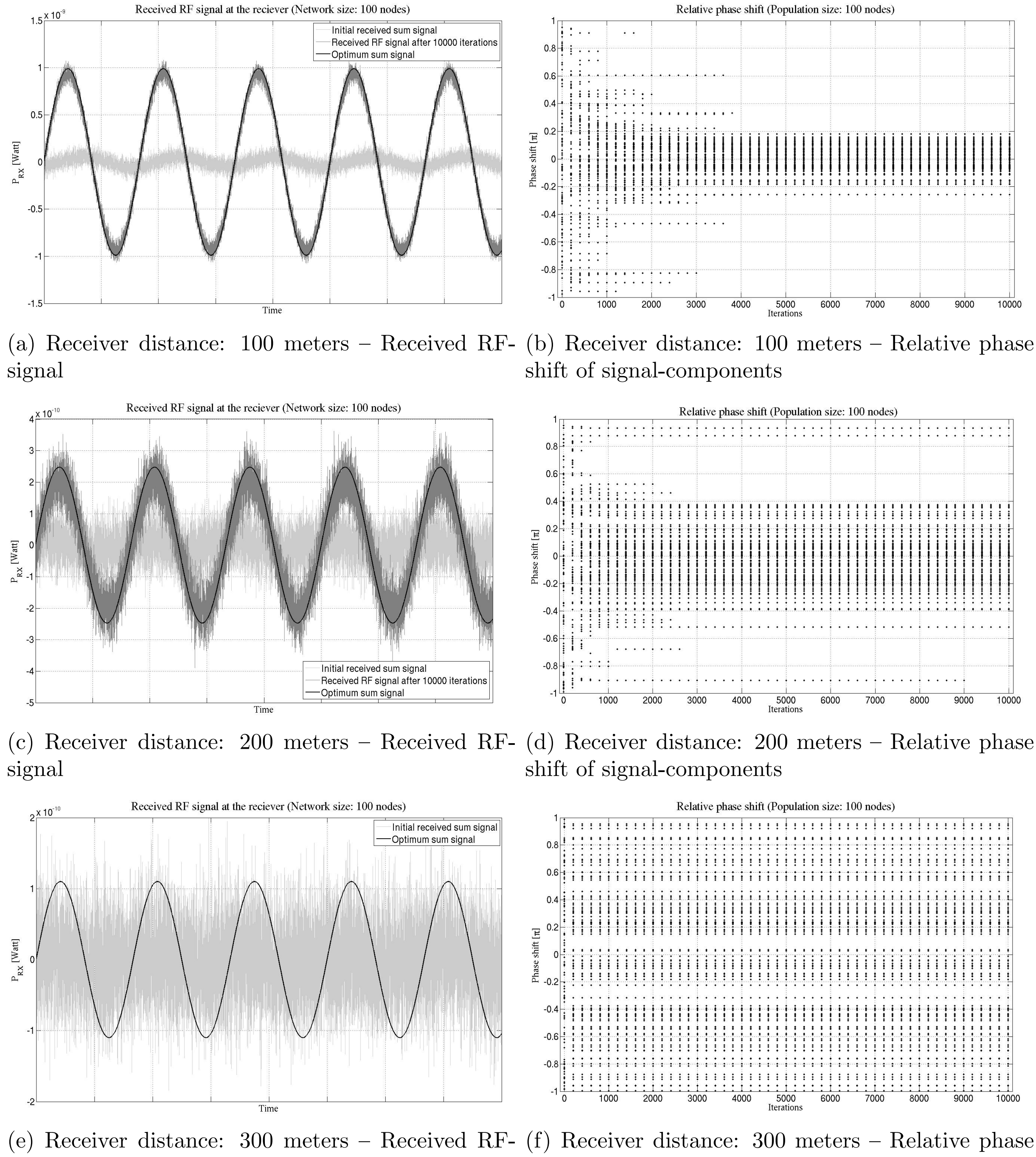
When the count of nodes that participate in the synchronisation is altered, this also impacts the performance of this process (cf. section 7.2). We conducted several simulations with network sizes ranging from 20 to 100 nodes. Figure 7.20 depicts the performance of several synchronisation processes with varying network sizes.

In these simulations, we set $p_\gamma = 0.05$ and utilised a uniformly distributed phase alteration process. We see that the maximum fitness-value achieved is lower for smaller network sizes. This is due to the RMSE measure that compares the achieved sum-signal to an expected optimum superimposed signal. As the count of participating nodes diminishes, also the amplitude of the optimum signal decreases. As expected, the optimum value is reached earlier for smaller network sizes.

Distance between receiver and network

We are also interested in the performance of distributed adaptive beamforming when the distance between the network and a receiver is increased. For a uniformly distributed phase alteration process with $p_\gamma = \frac{1}{n}$ we increase the transmission-distance successively. Figure 7.21 depicts the phase coherency achieved and the received sum-signal for various transmission-distances.

Although the noise-power relative to the sum-signal increases, synchronisation is possible at about 200 meters distance. Observe that in our model with $P_{tx} = 1mW$ we expect a



(a) Receiver distance: 100 meters – Received RF-signal

(b) Receiver distance: 100 meters – Relative phase shift of signal-components

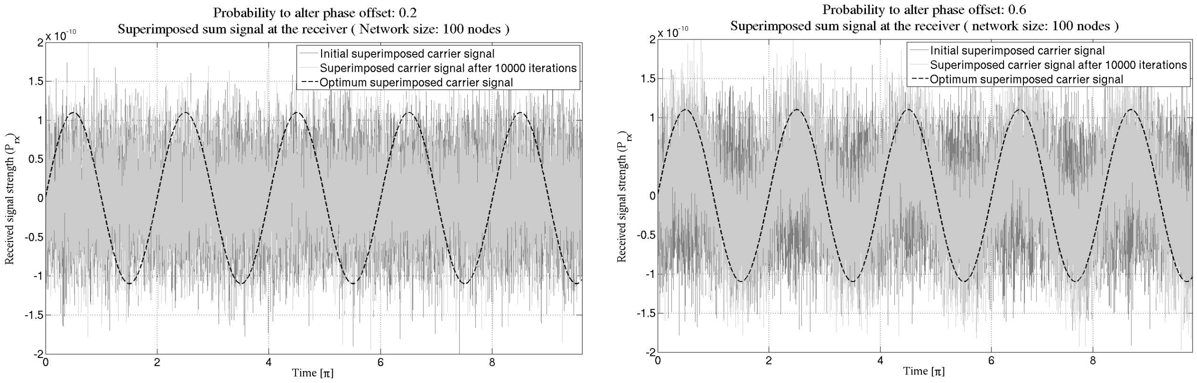
(c) Receiver distance: 200 meters – Received RF-signal

(d) Receiver distance: 200 meters – Relative phase shift of signal-components

(e) Receiver distance: 300 meters – Received RF-signal

(f) Receiver distance: 300 meters – Relative phase shift of signal-components

Figure 7.21: RF-signal strength and relative phase shift of received signal-components for a network size of 100 nodes after 10000 iterations. Nodes are distributed uniformly at random on a $30m \times 30m$ square area and transmit at $P_{TX} = 1mW$ with $p_\gamma = \frac{1}{n}$.



(a) Receiver distance: 300 meters, $p_\gamma = 0.2$

(b) Receiver distance: 300 meters, $p_\gamma = 0.6$

Figure 7.22: RF-signal strength and relative phase shift of received signal-components for a network size of 100 nodes after 10000 iterations. Nodes are distributed uniformly at random on a $30m \times 30m$ square area and transmit at $P_{TX} = 1mW$.

signal strength at the receiver of $0.1\mu W$ or $-40dBm$ for each single carrier at this distance.

When the distance is further increased to 300 meters, however, synchronisation is not possible with this configuration. This is due to the high impact of the noise fluctuation on the received signal. The fluctuation of the noise-figure has a higher impact on the signal than the alteration of single carrier-signals.

However, when more carrier-signals are altered simultaneously, a weak synchronisation is still possible. Figure 7.22 depicts the received carrier-signal after 100 iterations for the uniformly distributed process with $p_\gamma = 0.2$ and $p_\gamma = 0.6$. We see that the synchronisation quality is improved with increasing p_γ. While the superimposed signal is indistinguishable for $p_\gamma = 0.2$, the synchronisation quality increases with $p_\gamma = 0.6$. Although the signal is heavily distorted, the carrier can be extracted.

7.2.6 Impact of algorithmic modifications

The iterative feedback-based distributed adaptive beamforming in wireless sensor networks was analysed by various authors in the literature. Some of them have proposed small algorithmic alterations of the basic approach in order to improve the synchronisation-performance. We present three of these approaches in the following sections.

Reelection of unsuccessful nodes

The author of [43] observed that when nodes discard their last alteration to their carrier phases at the end of an iteration, the information available from the receiver feedback is not optimally utilised.

In particular, when a set of nodes alters their individual phase-offsets simultaneously, this constitutes a distinct shift in the carrier phase of the received sum-signal.

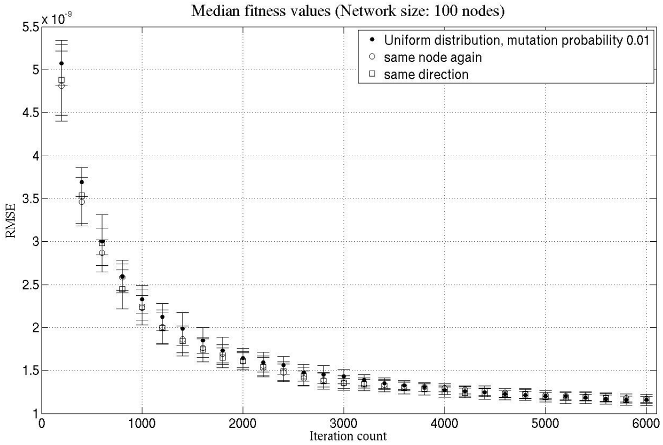


Figure 7.23: Performance of distributed adaptive beamforming in WSNs when successful nodes are re-considered for mutation

When the fitness-value reached after this operation is decreased, however, it means that the distance between the phase of an optimal configuration and the current superimposition was increased. Consequently, when all nodes would apply the inverse of their last applied phase-offset, this distance would be decreased and the fitness improved.

In [43] it was proposed to apply the inverse of the last applied phase modulation each time a fitness-value is decreased. The author could show that the optimisation-time is reduced by factor $\frac{1}{2}$ by this approach. The reason is, clearly, that the fitness decrease which is achieved with a probability of about $\frac{1}{2}$ (an improved fitness-value in in every second iteration on average) is now converted into an increase of the fitness-value.

Reelection of successful nodes

For distributed adaptive beamforming, an optimum is reached when all received transmit signal-components are in phase. In the random synchronisation process, this situation is approached iteratively, which means that several steps are required to attain an optimum. In particular, as long as the optimum phase-offset for one transmit signal-component is not reached, the fitness-value can be improved by sufficiently altering the phase-offset for this transmit signal-component again. To exploit this property, we modify the implementation of the original approach in such a way that transmit signal-components that were altered in an iteration in which the fitness-value was increased, are altered again in the next iteration. The intuition behind this approach is that for a transmit signal-component which was successfully altered we expect that the optimum phase-offset is not yet reached so that further benefit is possible. We implemented two distinct approaches. The first approach is to alter the phase-offset of the same transmit signal-component again uniformly at random. For the second implementation also the direction in which the phase was altered is sustained. Figure 7.23 depicts simulation results for these two approaches when compared to the standard uniformly distributed phase alteration approach. We observe that both

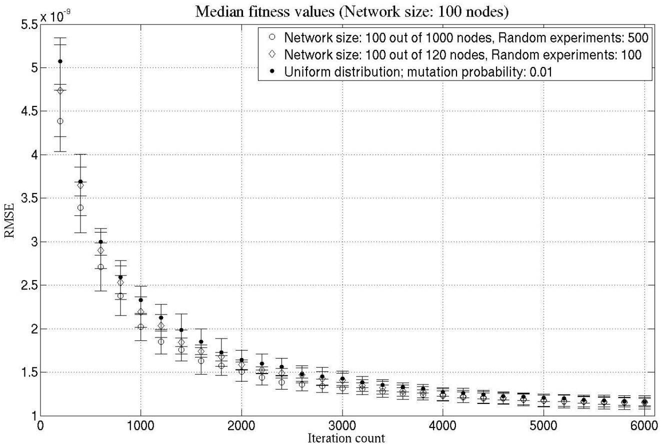


Figure 7.24: Performance of distributed adaptive beamforming in WSNs when participating nodes are chosen based on random experiments

approaches have similar performance and improve the fitness early in the synchronisation process. With increasing iteration count, however, the standard approach caught up.

Preconfigured nodes

In distributed adaptive beamforming, several nodes in a network collaboratively reach a distant receiver. The synchronisation-time for these nodes is dependent on the number of nodes that participate in the synchronisation (cf. [14, 15]). Consequently, when not all nodes are required to reach a given distance, it is beneficial to utilise only a subset of nodes. In particular, we would like to pick those nodes for the synchronisation process that are well pre-synchronised so that the initial fitness-value is already high. We conducted several simulation runs in which a set of pre-synchronised nodes is identified in random experiments prior to the synchronisation. In these random experiments a set of 100 randomly chosen nodes transmit simultaneously. The fitness-value reached in this simultaneous transmission is stored for each experiment. After all random experiments are completed the synchronisation is conducted with the nodeset that scored the best fitness-value during the random experiments. Figure 7.24 details the simulation results. For a network size of 1000 and 120 nodes, 100 nodes are picked in 500 and 100 random experiments, respectively. These simulations are compared to simulation results in which a network of 100 nodes is synchronised with uniformly distributed phase alterations and a mutation probability of 0.01 but without pre-synchronisation of nodes. We see that the pre-synchronised node-sets reach better fitness-values earlier in the simulation. In particular, an improvement is already visible for 100 nodes chosen in 100 random experiments out of 120 nodes. This shows that we can benefit from the node choice already with few random experiments and with network sizes that deviate from the required network size only slightly. However, observe also that the lead in fitness-value is shrinking with increasing iteration count.

7.3 Alternative algorithmic approaches

We have so far discussed the feasibility of a simple iterative approach to achieve distributed adaptive beamforming in wireless sensor networks. Based on these findings we will discuss three modifications to this general iterative method that are suited to improve the synchronisation-performance.

7.3.1 Hierarchical clustering

Since this bound on the synchronisation-time is more than linear in n (cf. section 7.2), it is striking that the $\mathrm{RSS}_{\mathrm{sum}}$ changes in a linear way with the network size n. Therefore, the overall energy consumption and synchronisation-time might benefit from a reduced number of nodes transmitting at increased power level.

We propose the following hierarchical clustering scheme that synchronises all transmit nodes iteratively in clusters of reduced size.

1. Determine clusters (e.g. by a random process initialised by the receiver node)

2. Guided by the receiver node, synchronise clusters successively as described above with possibly increased transmit power for nodes. When cluster ι is sufficiently synchronised, all nodes in this cluster sustain their carrier-signal and stop transmitting until all other clusters are synchronised.

3. At this stage, carrier-signals in each one of the clusters are in phase but carrier phases of distinct clusters might differ. To achieve overall synchronisation, we build an overlay-cluster of representative nodes from all clusters. These nodes are then synchronised.

4. Nodes in all clusters alter the phase of their carrier-signal by the phase-offset experienced by the corresponding representative node. Let $\zeta_i = \Re\left(m(t)\mathrm{RSS}_i e^{j2\pi f_c t(\gamma_i+\phi_i+\psi_i)}\right)$ and $\zeta_i' = \Re\left(m(t)\mathrm{RSS}_i e^{j2\pi f_c t(\gamma_i'+\phi_i+\psi_i)}\right)$ be the carrier-signals of representative node i from cluster ι before and after synchronisation between representative nodes was achieved. A node h from the same cluster ι will then alter its carrier-signal $\zeta_h = \Re\left(m(t)\mathrm{RSS}_h e^{j2\pi f_c t(\gamma_h+\phi_h+\psi_h)}\right)$ to $\zeta_h' = \Re\left(m(t)\mathrm{RSS}_h e^{j2\pi f_c t(\gamma_h+\phi_h+\psi_h+\gamma_i-\gamma_i')}\right)$. Under ideal conditions, all nodes should be in phase after this step although an overall synchronisation was not applied.

5. To account for synchronisation errors a final synchronisation phase in which all nodes participate concludes the overall synchronisation process.

Figure 7.25 illustrates this procedure. A potential problem for this approach is phase noise. As only one cluster is synchronised at a time, due to practical properties of oscillators, phases of nodes in the inactive clusters experience phase noise and start drifting

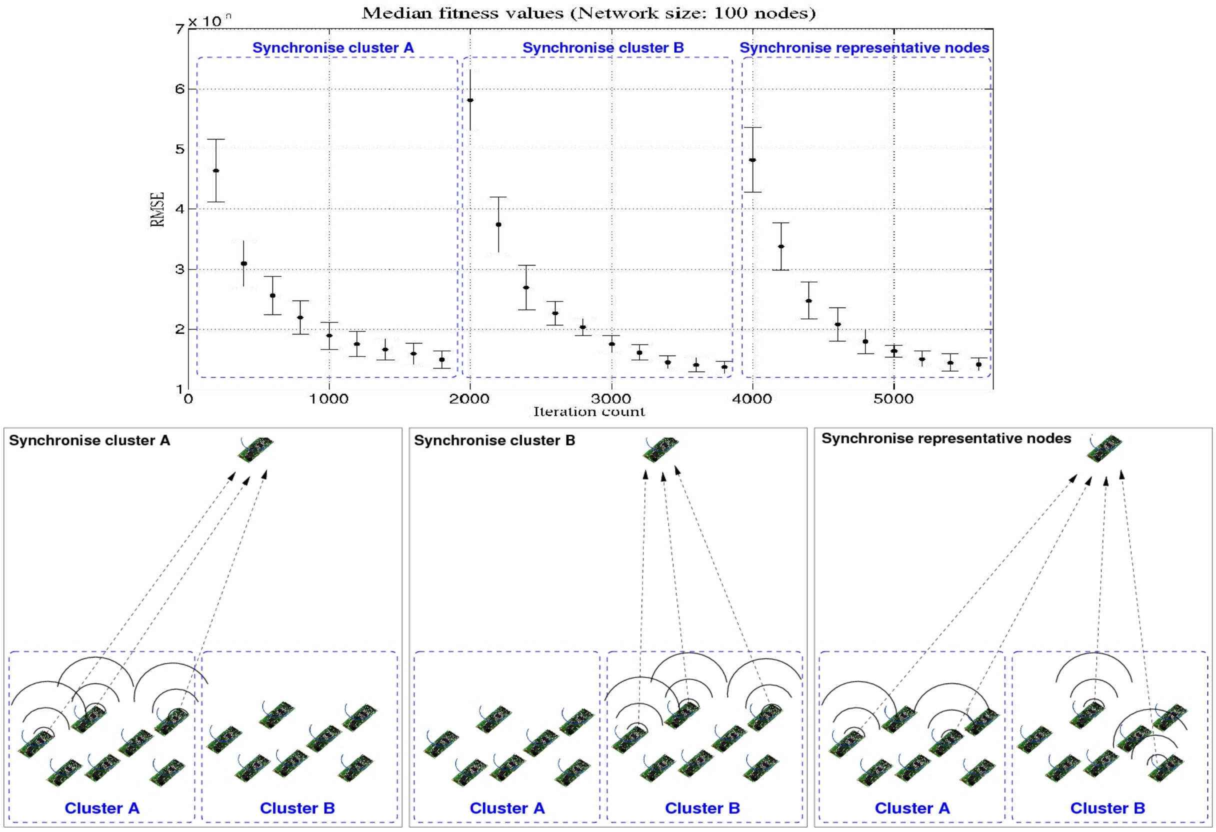


Figure 7.25: Illustration of the approach to cluster the network of nodes in order to improve the synchronisation-time of feedback-based closed-loop distributed adaptive beamforming.

out of phase. However, sufficient synchronisation is possible in the order of milliseconds. [1] Therefore, we do not consider phase noise an important issue for the hierarchical clustering approach.

Observe that all coordination is initiated by the receiver node so that no inter-node communication is required for coordination. Clusters are formed by a random process and synchronisation between and within clusters is achieved by the distributed adaptive beamforming approach described above.

Depending on the size of the network, more than one hierarchy stage might be optimal for the optimisation-time and the energy consumption. In order to estimate the optimal hierarchy depth and the optimum cluster size, the count of nodes participating in the synchronisation process must be computed. We assume that the nodes themselves do not know of the network size. This means that the remote receiver derives the network size, calculates optimal cluster sizes and hierarchy depths and transmits this information to the nodes in the network. In [98] it was demonstrated that the superimposed sum-signal from arbitrarily synchronised nodes over some time interval is sufficient to compute an estimation on the number of transmitting nodes. We can derive the optimum hierarchy depth and cluster size by integer programming in time $\mathbb{O}(n^2)$.

We estimate the expected optimisation-time for a network of size n by $E[T_{\mathcal{P}n}] = c \cdot k \cdot n \cdot \log(n)$ (cf. section 7.2) for a suitable constant c and the expected energy consumption by $E[\mathcal{E}_{\mathcal{P}n}] = c \cdot k \cdot n \cdot \log(n) \cdot \overline{\mathcal{E}_{\mathcal{P}n}}$ where $\overline{\mathcal{E}_{\mathcal{P}n}}$ is the energy consumption of all n nodes in one iteration [14]. A hierarchy and cluster structure that minimises these formulae when added up over all hierarchy stages is optimal in our sense. We derive the optimum cluster sizes and hierarchy depths by integer programming. Observe that for a cluster size of m the above formulae have the property

$$E[T_{\mathcal{P}n}] = E[T_{\mathcal{P}\frac{n}{m}}] \cdot \frac{n}{m} \cdot E[T_{\mathcal{P}m}]$$
$$E[\mathcal{E}_{\mathcal{P}n}] = E[\mathcal{E}_{\mathcal{P}\frac{n}{m}}] \cdot \frac{n}{m} \cdot E[\mathcal{E}_{\mathcal{P}m}].$$

We define the recursion by

$$\begin{aligned} E_{\text{opt}}[T_{\mathcal{P}n}] &= min_m \left[E_{\text{opt}}[T_{\mathcal{P}\frac{n}{m}}] \cdot \frac{n}{m} \cdot E_{\text{opt}}[T_{\mathcal{P}m}] \right] \\ E_{\text{opt}}[\mathcal{E}_{\mathcal{P}n}] &= min_m \left[E_{\text{opt}}[\mathcal{E}_{\mathcal{P}\frac{n}{m}}] \cdot \frac{n}{m} \cdot E_{\text{opt}}[\mathcal{E}_{\mathcal{P}m}] \right] \end{aligned}$$

and the start of the recursion by $E_{\text{opt}}[T_{\mathcal{P}\eta}]$ and $E_{\text{opt}}[\mathcal{E}_{\mathcal{P}\eta}]$ with η being the minimum feasible cluster size for distributed adaptive beamforming when the maximum transmission power and distance are given. Since η is dependent on the distance to the receiver, it can be calculated over the round trip time between the sensor network and the receiver.

[1] Consider, for instance, that the synchronisation sequence requires several ten signal periods and that the RSS estimation and application of phase shift is also possible in similar order. A simple calculation (e.g. frequency $f = 2.4$GHz, transmission-distance $d = 100m$, $c = 3 \cdot 10^8 \frac{m}{s}$) reveals that a single iteration requires only microseconds.

The time required for the calculation of the optimum hierarchy depth and cluster sizes is quadratic. With a network of n nodes, at most n^2 distinct terms $E_{\text{opt}}[T_{\mathcal{P}i}]$ and $E_{\text{opt}}[\mathcal{E}_{\mathcal{P}i}]$ with $i \in \{1, \dots, n\}$ are of relevance. We can start by calculating $E_{\text{opt}}[\mathcal{E}_{\mathcal{P}\eta}]$ and $E_{\text{opt}}[T_{\mathcal{P}\eta}]$ and obtain all other values by table look-up according to $E_{\text{opt}}[T_{\mathcal{P}n}]$ and $E_{\text{opt}}[\mathcal{E}_{\mathcal{P}n}]$ in time $\mathcal{O}(n^2)$ since every one of the (at most) n entries has not more than n possible predecessors.

7.3.2 A local random search approach

Recent approaches to distributed adaptive transmit beamforming in wireless sensor networks utilise a random search approach that could reach any point in the search-space with a positive probability in each iteration. We define a search-point as one combination of phase-offsets for all RF-signal-components and a neighbourhood-relation by the difference in phase-offsets in these configurations. As the search-space is multimodal, no local optima exist so that in an arbitrary neighbourhood around a given search-point the fitness-value is either optimal or search-points with an improved fitness-value exist.

Therefore, a local random search cannot converge in a local minima. Additionally, the optimisation landscape for a local search approach is simple then. As long as it manages to follow a path with increasing fitness-values, it will find an optimum with a probability of 1.

It can also be shown that, when the distance to the optimum and to the worst point is greater than the neighbourhood radius, the algorithm has an equal chance to improve or make the fitness score worse. This can be demonstrated as follows. Assume that the fitness of the sum-signal is proportional to the amplitude of the signal. When $n-1$ signal-components are received simultaneously at a given frequency f the resulting sum-signal is of the same frequency as the individual RF-signal-components. When the phase-offset of the n-th signal-component is identical or complementary to the phase-offset of the sum-signal, the amplitude of the signal created by the superimposition of these two signals is at its maximum or minimum, respectively. In between these two values it deteriorates symmetrically. Consequently, when the minimum or maximum is not inside the neighbourhood, an equal number of points incorporate better or worse fitness-values (cf. figure 7.26) – although the slope of the fitness landscape might differ among search-points that improve or decrease the fitness-value.

This means that an algorithm with restricted neighbourhood-size has a probability of 0.5 to increase the fitness-value for a long time during the optimisation until the optimum point is within the neighbourhood. The price for this high probability to improve the fitness-value in each iteration is that the chance to reach the optimum in one step or, generally to achieve great progress in one step (as possible with an unrestricted neighbourhood-size) is lost. Since this event is significantly less probable, we are prepared to pay this price. An upper bound on the optimisation-time of a local random search approach for distributed adaptive beamforming in wireless sensor networks is provided accordingly.

To simplify the analysis we model the optimisation problem in a binary representation. We assume that each one of the sensor nodes is able to apply k distinct phase-offsets of transmit RF-signals. For each of the n nodes, we encode the distinct phase-offsets by $\log(k)$

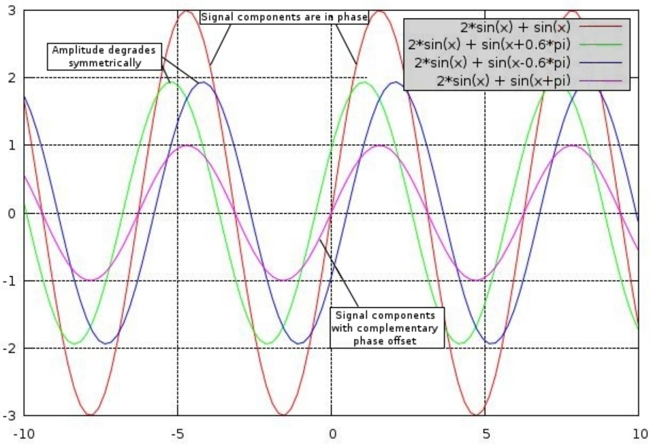


Figure 7.26: Example of sinusoid sum-signals. The amplitude of the sum-signal degrades symmetrically when the phase-offset between the two signal-components increases

bits binary. To summarise, a binary string of length $n \cdot \log(k)$ describes the contribution of phase-offsets in one iteration. The neighbourhood of a given configuration is defined by the maximum phase-offset applicable to an individual node altering the phase of its carrier-signal. An individual node i alters the phase-offset γ_i of its carrier-signal uniformly at random within a range of $[\gamma_i - \varrho, \gamma_i + \varrho]$ for suitable ϱ. We assume that configurations are encoded so that the hamming-distance between two configurations increases with an increasing difference in phase-offsets.

We analyse the count of bit mutations of this representative bitstring until a bitstring is found that encodes the phase configurations of a global optimum. We define a global optimum as a superimposition of signal-components in which all signal-components are within $\frac{2\pi}{k}$ of a superimposition with perfect phase coherency. We choose the mutation probability as $\frac{1}{n}$ for a network of n nodes. In this case, the $\log(k)$ bits that represent the phase-offset of the corresponding node are altered uniformly at random inside the neighbourhood boundaries. With Chernoff bounds we can show that w.h.p. the hamming-distance to an optimum is not much smaller than $\frac{n \cdot \log(k)}{2}$.

An asymptotic upper bound

We calculate an upper bound for the local random search approach similarly to the upper bound derived in section 7.2. We define the neighbourhood of size N for each binary representation $\mathcal{I}$ of an individual carrier-signal ζ_i (remember that the individual representation was concatenated from all n binary representations of individual carrier-signals of length $\log(k)$ (cf. figure 7.2)) as all individual representations with hamming-distance at most N to $\mathcal{I}$. As long as the optimum is far away (i.e. outside the neighbourhood boundaries of all carrier-signals), the probability to reach a new search-point with better fitness-value is at least $\frac{1}{2}$ as detailed above. The expected time until all carrier-signals have the optimum inside their neighbourhood boundaries then is

$$E[\mathcal{P}_1] = \mathcal{O}(n \cdot \log(k)). \tag{7.24}$$

Afterwards, the analysis of the optimisation-time is similar to the upper bound derived for the global random search approach in section 7.2. The difference for the local random search approach is, that the distance to the optimum is $n \cdot N$ at most. Consequently, the estimation of the upper bound on the synchronisation-performance in equation (7.14) then is altered to

$$\begin{aligned} E[T_{\mathcal{P}_2}] &\leq \sum_{i=n\cdot\log(k)-n\cdot N}^{n\cdot\log(k)-1} \frac{e \cdot n \cdot \log(k)}{n \cdot \log(k) - i} \\ &= e \cdot n \cdot \log(k) \cdot \sum_{i=n\cdot\log(k)-n\cdot N+1}^{n\cdot\log(k)} \frac{1}{i} \\ &< e \cdot n \cdot \log(k) \cdot (\ln(n \cdot \log(k)) + 1 - \ln(n \cdot \log(k) - n \cdot N) + 1) \\ &= \mathcal{O}\left(n \cdot \log\left(\frac{n \cdot \log(k)}{n \cdot \log(k) - n \cdot N} + k\right)\right). \end{aligned} \tag{7.25}$$

The upper bound on the expected overall optimisation-time is then composed from the upper bound on the optimisation-times of both phases as

$$\begin{aligned} E[T_{\mathcal{P}}] &= E[T_{\mathcal{P}_1}] + E[T_{\mathcal{P}_2}] \\ &= \mathcal{O}\left(n \cdot \log\left(\frac{n \cdot \log(k)}{n \cdot \log(k) - n \cdot N} + k\right)\right). \end{aligned} \tag{7.26}$$

An asymptotic lower bound

For the local random search approach we can obtain a lower bound on the synchronisation-performance by the method of the expected progress similarly to the approximation in section 7.2. With similar arguments as in section 7.2 this leads to a lower bound of

$$\Omega\left(n \cdot \log\left(\frac{n \cdot \log(k)}{n \cdot \log(k) - n \cdot N} + k\right)\right). \tag{7.27}$$

Simulation results

For our implementation of the local random search algorithm we modify the phase alteration mechanism applied by individual nodes. While the phase-offset is chosen uniformly at random from all possible phase-offsets in $[-\pi, \pi]$, for the original random search approach, we restrict the set of possible future phase-offsets to a subset of this range: $\gamma \in [\sigma_1, \sigma_2]$ with $-\pi \leq \sigma_1 < \sigma_2 \leq \pi$. Figure 7.27 shows the simulation results of the algorithm. We observe that the hill-climbing approach reaches lower fitness-values faster than the original approach. In particular, the nearly optimal fitness-value reached by the original algorithm after 6000 iterations is achieved by the hill-climber after about 3000 iterations already.

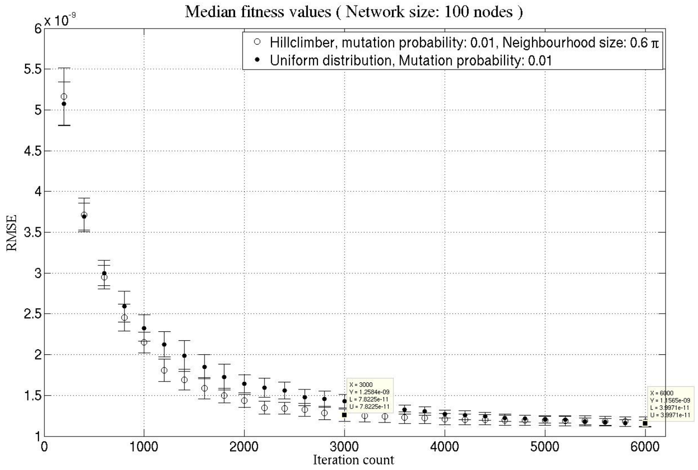


Figure 7.27: Performance of the local random search implementation for distributed adaptive beamforming in wireless sensor networks

7.3.3 Multivariable equations

The recent approaches discussed for distributed adaptive transmit beamforming in wireless sensor networks all rely heavily on random experiments. This was necessary since distinct nodes in the scenario do not communicate with each other. Therefore, each node has access only to very limited information at the current search-point. Consequently, as a concise view on the current synchronisation progress is not given, a random search-heuristic can be seen as the natural approach for this problem-scenario.

The introduction of randomness into the search approach, however, also decreases the synchronisation-performance. First approaches to reduce the amount of randomness have been introduced in section 7.3.1 and section 7.3.2. In the former approach, the problem was divided into sub-problems of a smaller size. For these smaller problem-instances the synchronisation-performance improved since the search-space is drastically smaller for smaller network sizes. In particular, it decreases exponentially in n as each node corresponds to a further dimension in the search-space.

For the latter approach, randomness was reduced by restricting the neighbourhood-size of the search approach. As long as the optimum is not inside the neighbourhood, the probability to improve the fitness-value was $\frac{1}{2}$.

In the following section we present an approach by which a single node is able to estimate the optimum phase-offset of its carrier-signal relative to the current phase-offsets of all other nodes. Due to the distributed nature of this scenario, we are still bound to a small amount of randomness. However, we can show that the approach achieves an asymptotically optimal synchronisation-performance.

The idea to reach this synchronisation-performance is to utilise the receiver feedback optimally. In the approaches discussed above, the feedback required is binary. However, when more information is encoded in the feedback-value so that information on the actual feedback function can be derived, transmitters can utilise this additional information to apply improved synchronisation-methods.

Consequently, a vital requirement on the design of the algorithm is that a rich feedback is provided by the receiver node. In particular, the approach described is not feasible with binary feedback-values.

Algorithmic modelling

Since no local optimum exists in the search-space due to its weak multimodality (cf. section 5.3.1), the performance of $\Theta(n \cdot \log(n \cdot \log(k) + k))$ we derived for the random search approach seems weak. In every iteration the receiver provides additional information over a feedback-value so that a node i can learn the optimum phase-offset of its own carrier

$$\zeta_i = \Re\left(m(t)\mathrm{RSS}_i e^{j2\pi f_c t(\gamma_i+\phi_i+\psi_i)}\right)$$

relative to the superimposed sum-signal

$$\zeta_{\mathrm{sum}\setminus i} = \Re\left(m(t)e^{j2\pi f_c t}\sum_{o\in[1,n];o\neq i}\mathrm{RSS}_o e^{j(\gamma_o+\phi_o+\psi_o)}\right)$$

of all other nodes, provided that the latter does not change significantly. $\zeta_{\mathrm{sum}\setminus i}$ is a sinusoid signal. The fitness-value is at its maximum when ζ_i and $\zeta_{\mathrm{sum}\setminus i}$ have an identical phase-offset at a receiver. With increasing phase-offset

$$|(\gamma_i+\phi_i+\psi_i)-(\gamma_{\mathrm{sum}\setminus i}+\phi_{\mathrm{sum}\setminus i}+\psi_{\mathrm{sum}\setminus i})|$$

the fitness-value decreases symmetrically. Consequently, the fitness-function (note that we are now considering the fitness-function $\mathcal{F}$ and not the received superimposed sum-signal ζ_{sum}) is of the form

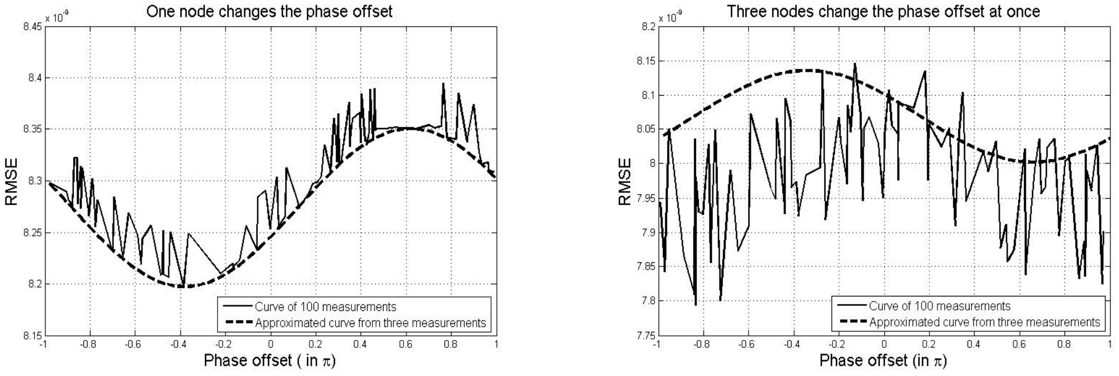
$$\mathcal{F}(\gamma_i) = A\sin(\gamma_i + \varPhi) + c. \tag{7.28}$$

This is an equation with the three unknowns A (amplitude), $\varPhi$ (phase-offset of $\mathcal{F}$) and the additive term c so that a node i can calculate it with three distinct measurements.

These three function-values for three distinct phase-offsets $\gamma_1, \gamma_2, \gamma_3$ of a carrier-signal can be calculated when the one node with a non-fixed carrier-signal acquires the corresponding feedback-values from the remote receiver. In figure 7.28(a), we depict the accuracy at which the RMSE fitness-function can be estimated by this procedure. We calculate the root of the mean square error (RMSE) in this figure as

$$RMSE = \sqrt{\sum_{t=0}^{\tau}\frac{\left(\zeta_{\mathrm{sum}}+\zeta_{\mathrm{noise}}-\zeta_{\mathrm{opt}}\right)^2}{n}}. \tag{7.29}$$

In this formula, τ is chosen to cover several signal periods.



(a) RMSE-γ-relationship when only one sender node changes the phase-offset

(b) RMSE-γ-relationship when three sender nodes change the phase-offset at once

Figure 7.28: Approximation of the RMSE- phase-offset- relationship

The dashed line in the figure depicts the fitness-function estimated from three distinct feedback-values while the solid line is created from 100 feedback-calculations in a Matlab-based simulation-environment. The 100 nodes are placed randomly in a field with a minimal distance of 30 meters from the receiver. The transmit nodes send with 2.4 GHz and a transmission power of 0.1 mW. From the calculated expected fitness-function we can easily determine the optimum phase-offset for this carrier-signal.

We experienced a maximum deviation between the approximated and the measured RMSE values of less than 1% when all but one node keep their phase-offset constant. In our measurements, the deviation of the calculated fitness curve did not exceed 0.6% when only one node adapts its phase-offset.

However, since inter-node communication is not assumed in the scenario of distributed adaptive transmit beamforming in wireless sensor networks, more than one node might alter its phase-offset at a time. In this case, we can show that the calculated fitness curve deviates more significantly from the actual fitness-function. With two nodes adapting their phase-offset simultaneously we already experienced a deviation of approximately 1.5%. Figure 7.28(b) represents the approximation of the fitness-function when three nodes change their phase-offset simultaneously. The deviation between the approximated and the measured values is greater than that in the previous case and reaches values of more than 3%.

This observation leads to two important conclusions. The first is that a precise calculation of the optimum phase-offset for a single node is possible when only one sender changes the phase-offset of its carrier-signal during a single iteration. The second fact is, that verification of the correctness of the results is possible by measuring the significance of the deviation between the calculated and the actual fitness-function.

In the following discussion we describe the steps of the synchronisation procedure. Four iterations each are aggregated logically into one action. During an action, a node may either participate by calculating it's optimum phase-offset γ_i^* (active node) or it may transmit its carrier-signal unmodified (passive node).

With a probability $p_i = \frac{1}{n}$ (for a network of n nodes) a node i becomes an active participant and stays passive otherways.

An active node will :

1. Transmit it's carrier-signal with three distinct phase-offsets $\gamma_1 \neq \gamma_2 \neq \gamma_3$ and measure the feedback generated by the remote receiver. Feedback value and corresponding phase-offset are stored by the node.
2. From these three feedback-values and phase-offsets, it estimates the feedback function and calculates the optimum phase-offset γ_i^*.
3. Transmit a fourth time with $\gamma_4 = \gamma_i^*$.
4. If the deviation is less than 1% save γ_i^* as the optimal phase-offset, otherwise discard it.

A passive node will :

1. Transmit the carrier-signal four times with identical phase-offset γ_i.

As nodes are chosen according to a random process, it is possible that more than one node simultaneously alters its phase-offset. In this case, the node's conclusions on the impact of their phase-alteration on the fitness-value are biased. Therefore, when the measured value deviates significantly from the expected fitness-value in the fourth measurement a node concludes that it was not the only one to alter its phase and reverses its decision.

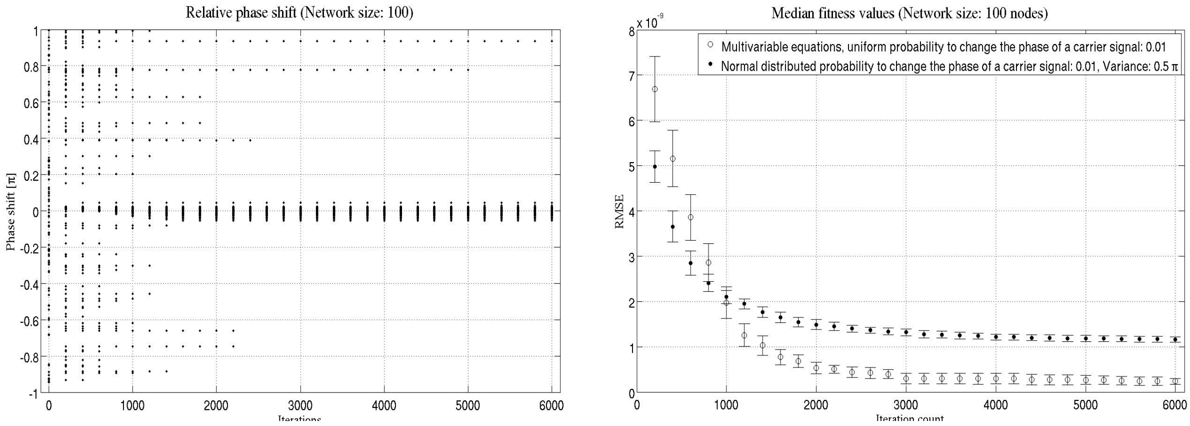
In order to decrease the count of idle actions at which either more than one node or no node actively participate, the nodes may adjust the value of p_i. After receiving the fourth feedback message, an active node i that has calculated γ_i^* successfully, becomes a passive node for a certain number of iterations. The node sets $p_i = 0$ to reduce the interference for other active nodes. All passive nodes that register an improvement of the feedback-value after the fourth transmission assume that a node has calculated its γ_i^* successfully and alter their p_i-value to $p_i = \frac{1}{n - \text{successful actions}}$. The probability that a node successfully calculates γ_i^* during a time slot is 0.35.

As this procedure is guided purely by the feedback of the receiver node, inter-node communication is not required as the feedback is broadcast to all nodes in the network.

Further performance improvements can be achieved when nodes utilise only three subsequent iterations and acquire the first measurement from the last transmission of the preceding three subsequent iterations.

Analytic consideration

In our current implementation, we require about $12n$ iterations for all nodes to finally find and set the optimum phase-offset of their carrier-signal. This is asymptotically optimal, since the optimum phase-offset of the carrier-signal has to be calculated for each single node. Asymptotically, the synchronisation-time of this algorithm is $\Theta(n)$ since on average the count of carrier-signals that are in phase increases by 1 in each iteration.



(a) Phase offset achieved by the proposed optimisation algorithm for distributed adaptive beamforming in WSNs

(b) Performance of the proposed optimisation algorithm for distributed adaptive beamforming in WSNs

Figure 7.29: Distributed adaptive beamforming with a network size of 100 nodes where phase-alterations are drawn uniformly at random. Each node adapts its carrier phase-offset with the probability of 0.01 in one iteration. In this case, multivariable equations are solved to determine the optimum phase-offset of the carrier-signal.

Simulation results

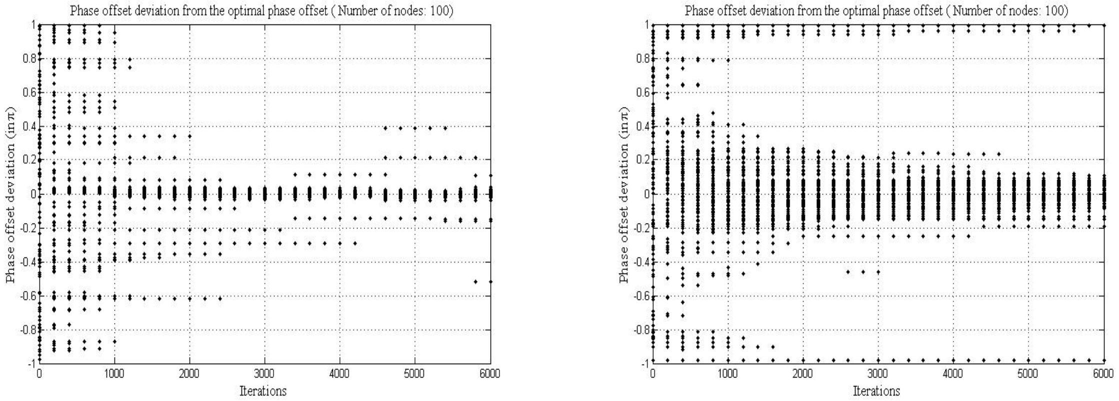
We also conducted simulation runs in which our implementation of the numerical algorithm described in section 7.3.3 is compared to the classical process with normal distributed phase alterations. When optimum phase-offsets are calculated by solving multivariable equations at the transmit nodes, the synchronisation-performance can be greatly improved as detailed in section 7.3.3. Figure 7.29 depicts the performance improvement achieved by solving multivariable equations to determine the fitness-function compared to a global random search approach. We observe that the global random search heuristic is outperformed already after about 1000 iterations and the fitness-value reached is greatly improved.

The phase-offset of distinct nodes is within $+/-0.05\pi$ for up to 99% of all nodes.

Figure 7.30(a) shows the relative deviation of the phase-offsets γ_i among 100 nodes. The results have been obtained in a Matlab-based simulation-environment. After about 1500 iterations most of the nodes (about 90 %) have near optimum phase-offsets. In comparison, the global random search approach typically utilised for distributed adaptive beamforming in wireless sensor networks has a greatly degraded performance (cf. figure 7.30(b)).

7.4 Environmental changes

For distributed adaptive beamforming in wireless sensor networks not only the synchronisation algorithm applied to the problem-scenario but also the environmental aspects impact the synchronisation-performance. In section 7.2.5 we have already observed that a higher



(a) Results using the numeric algorithm

(b) Results using a global random search approach

Figure 7.30: Deviation of the phase-offsets from the optimal phase-offsets using the numerical and the random method

noise-figure impacts the synchronisation-performance negatively. Also, we learned that this effect can be partly compensated with an increased mutation probability.

In addition, the velocity of nodes might impact the synchronisation-performance of distributed adaptive transmit beamforming in wireless sensor networks. Since the transmission beam that is sought after is focused on a restricted area, the synchronisation of nodes might be biased if the receiver traverses out of this area. This problem is being examined in section 7.4.1.

Another natural scenario has a bearing on the consideration of more than one receiver node. Typically the nodes in a sensor network are identical and a special exposed node with extended capabilities does not exist. When a sub-network is, however, separated from another part of the network, distributed adaptive transmit beamforming might help to establish a connection between both sub-networks. The central question in this scenario, however, is how communication between both sub-networks is feasible when nodes are not able to overhear messages from nodes of the other sub-network directly. This problem is considered in section 7.4.2.

So far, a population size of 1 was maintained for the evolutionary optimiser since this seemed to be the natural choice.

7.4.1 Velocity of nodes

In current studies on distributed adaptive beamforming in wireless sensor networks, all nodes are considered static. An interesting case to be studied is the mobility of nodes. The next section presents results from a Matlab-based simulation-environment where mobility is applied to transmitter or receiver nodes.

All nodes in the following simulations transmit at a frequency of 2.4 GHz. The sender nodes utilise a transmission power of 0.1 mW. Ambient white Gaussian noise (AWGN) with a noise-power of $-103dBm$ is applied as proposed in [64]. 100 nodes are placed in a $30m \times 30m \times 30m$ field. The transmit nodes are distributed randomly at the bottom of the field and the receiver is placed initially at the centre of the field's top, so that the minimum possible distance between a sender and receiver is 30 meters.

In the simulation we considered the Doppler effect due to the mobility of nodes. Individual signal-components are summed up at the receiver node to generate the superimposed sum-signal ζ_{sum}. Path-loss was calculated by the Friis free space equation [49]

$$P_{tx}\left(\frac{\lambda}{2\pi d}\right)^2 G_{tx}G_{rx} \tag{7.30}$$

with $G_{tx} = G_{rx} = 0$. We disregarded shadowing and signal reflection so that only the direct signal-component is utilised but not multipath propagation. We implemented a global random search approach and the asymptotically optimal algorithm introduced in section 7.3.3 for this scenario as well as two mobility models. The first mobility model is a random-walk model and the second a linear model.

Nodes in the random-walk model have no specific constant direction. After every iteration the direction of the movement is altered uniformly at random. The distance travelled between two consecutive iterations is constant and depends on the speed of the motion specified. In the linear model, the sender or the receiver move linearly in a constant direction and with a constant speed.

In order to quantify the maximum speed at which a synchronisation is possible, we define a synchronisation as successful when the signal strength achieved is at minimum 75% of the signal strength possible with perfect synchronisation. All simulation results obtained are depicted in figure 7.31. These results are discussed in the following sections.

Performance of the global random search approach

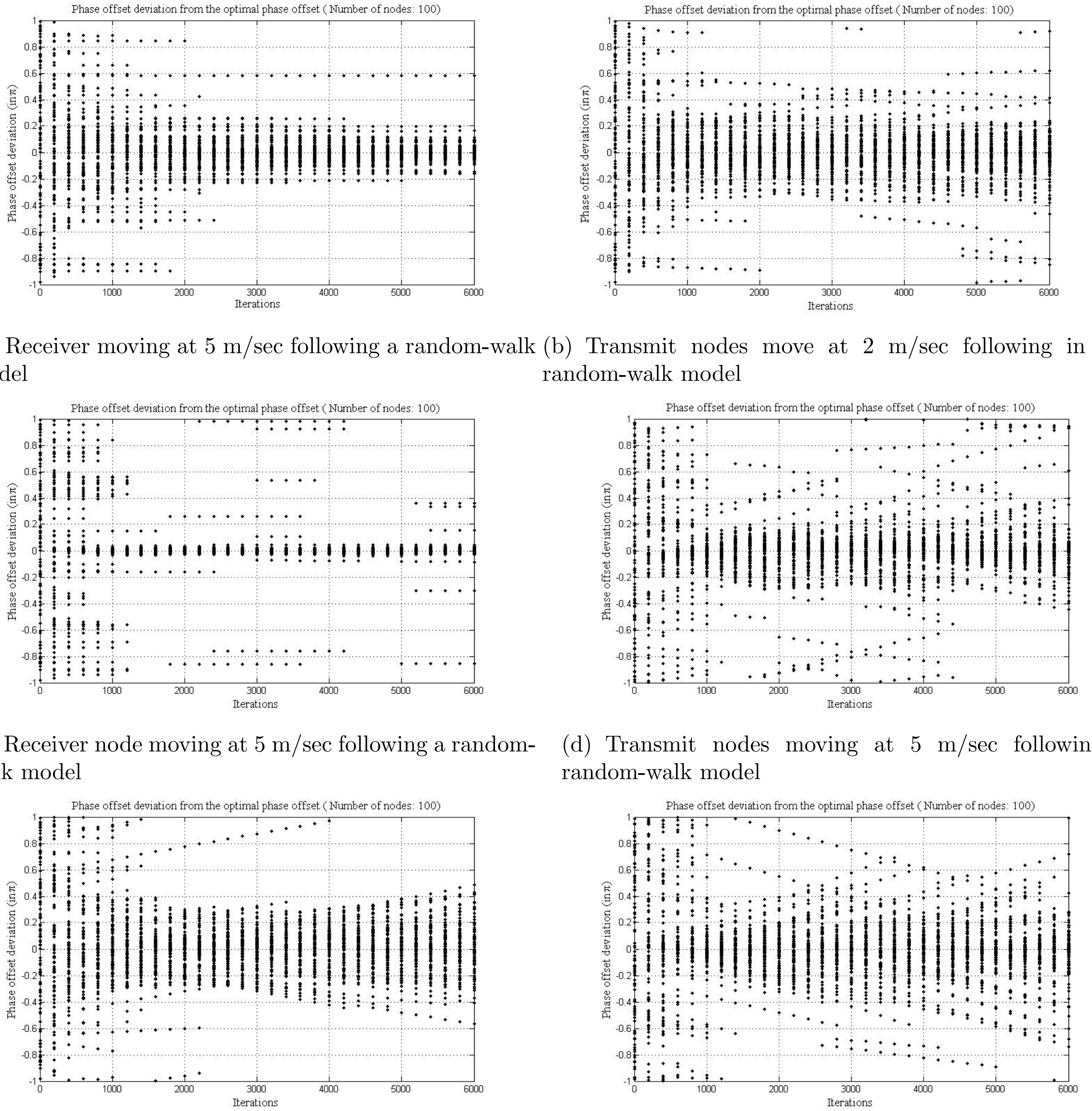
In our first scenario, the receiver node moves in a random walk mode. All transmit nodes remain static. We observed that a successful synchronisation in this scenario is possible with a movement speed of 5m/sec at most. Figure 7.31(a) shows the relative phase-offset of individual carrier-signals.

The standard deviation σ is about 0.1π for about 95% of all nodes after 6000 iterations. When, however, transmitters follow a random-walk movement, while the receiver is not moving, the maximum speed is about 2 m/sec (cf. figure 7.31(b)).

For the linear movement, the maximum relative speed between transmit and receive nodes with the global random search implementation is 30 m/sec regardless of whether the receive or the transmit nodes are moving (cf. figure 7.31(e).

Performance of the numeric algorithm

For the proposed numerical algorithm we applied the same settings as in section 7.4.1. When the receiver moves in a random-walk model while transmitters are static, the movement speed of 5 m/sec is easily supported (cf. figure 7.31(c)).



(a) Receiver moving at 5 m/sec following a random-walk model

(b) Transmit nodes move at 2 m/sec following in a random-walk model

(c) Receiver node moving at 5 m/sec following a random-walk model

(d) Transmit nodes moving at 5 m/sec following a random-walk model

(e) Nodes move with 30 m/s in a linear mode

(f) Nodes move with 60 m/s in a linear mode

Figure 7.31: Performance of two approaches to distributed adaptive transmit beamforming for wireless sensor networks in a Matlab-based simulation-environment

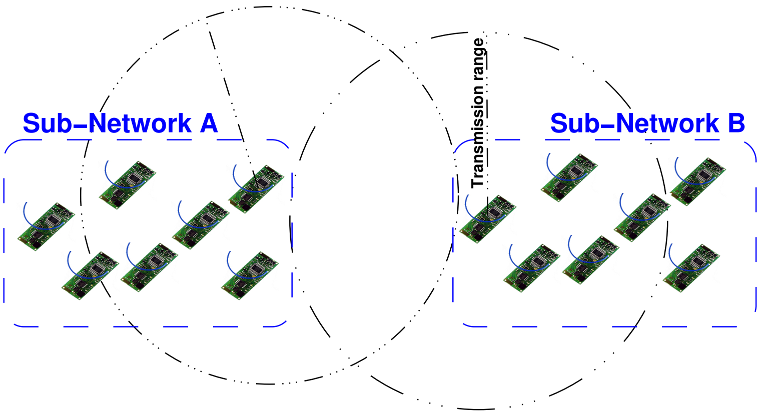


Figure 7.32: Schematic illustration of a disconnected network

In the figure, the standard deviation σ is about 0.03 π. Figure 7.31(d) depicts the phase-deviations when the receiver is static and the transmit nodes are moving. In this case a standard deviation of $\sigma = 0.22\pi$ is achieved after 6000 iterations.

In conclusion, we remark that the numeric method enables higher movement speeds as well as an improved synchronisation-performance. The maximum relative movement speed for the linear movement model is about 60 m/sec. Figure 7.31(f) depicts deviations of phase-offsets.

The standard deviation in this case is $\sigma = 0.18\pi$

7.4.2 Consideration of multiple receivers

In practical settings, networks of nodes may become disconnected when nodes die or become unavailable. In a disconnected network, two nodes, A and B, exist, so that no communication path can be established from A to B or from B to A through any number of intermediate nodes. As depicted in figure 7.32 a network is then divided into several sub-networks.

In order to establish a link between two sub-networks, although a single communication-link is not possible, collaborative communication can be utilised.

When several nodes from one sub-network collaboratively transmit to the remote sub-network, a communication link between these networks might be established. Until now, it was assumed that the receiver node is somehow more potent than the transmit nodes so that it is able to reach the transmit nodes on its own. When a network of identical nodes is separated, however, this assumption does not hold. None of the nodes in the receiving sub-network is capable of reaching the nodes in the remote sub-network by definition (otherwise, the network would not be disconnected). Therefore, a feedback transmitted by a single node will not reach any nodes in the other sub-network.

With a small modification of the approach detailed in section 6.2.2, however, it is possible for the nodes in the receiving network to create a feedback collaboratively that is overheard by the transmit nodes.

1. Each source node i adjusts its carrier phase-offset γ_i and frequency-offset f_i randomly
2. The source nodes transmit to the destination nodes simultaneously as a distributed beamformer.
3. The receiver nodes estimate the level of phase-synchronisation of the received sum-signal (e.g. by SNR, RSS or comparison to an expected signal).
4. A feedback-beacon is broadcast simultaneously by the receive nodes when the amount of synchronisation has improved to the last iteration. Otherwise, receive nodes do not transmit any data to the transmit nodes.
5. Transmit nodes sustain their phase adjustments if they sense a significant increase in the energy on the channel or else discard them.

Generally, compared to the approach detailed in section 6.2.2, we replaced the 1-bit feedback of a single receiver node by a transmit/non-transmit feedback of several simultaneously transmitting nodes. When a feedback is collaboratively transmitted, the energy on the channel increases significantly. This is interpreted as a positive feedback by the receive nodes. Otherwise, the feedback is interpreted as negative.

Observe, that it is not possible with this implementation to apply the asymptotically optimal algorithm detailed in section 7.3.3 to the scenario of multiple receivers, since this approach requires a quantitative feedback in order to estimate the fitness-function.

Since remote antennas of receive nodes are spatially distributed, the degree of phase-coherency differs among receiver nodes for any signal collaboratively transmitted by distributed transmitters.

Due to this property, distinct receive nodes may observe distinct changes to the signal-quality. While one node observes an increased signal quality, another node may observe a decreased signal quality. Therefore, the decision about the transmitted feedback might be ambiguous. This might lead to the situation that an eve- changing set of nodes transmits a positive feedback message. In the worst case, this set of nodes is either always sufficiently large, so that a positive feedback is observed at the transmit nodes, or it is always too small, so that always a negative feedback is observed. Due to this property, an optimisation may become infeasible in such a scenario.

Clearly, the distance between receive nodes, the transmit frequency utilised as well as the distance between transmit and receive nodes impact the similarity of superimposed signals received by distinct nodes. The following section discusses the impact of these parameters on the difference in phase-synchronisation of distributed remote receivers.

Practical problems for multiple receivers

We examine the scenario depicted in figure 7.33

A set $\mathcal{A} = \{A_1, \dots, A_n\}$ of transmit nodes simultaneously transmits its transmit signals

$$\zeta_i = \Re\left(m(t)\mathrm{RSS}_i e^{j2\pi f_c t(\gamma_i + \phi_i + \psi_i)}\right) \tag{7.31}$$

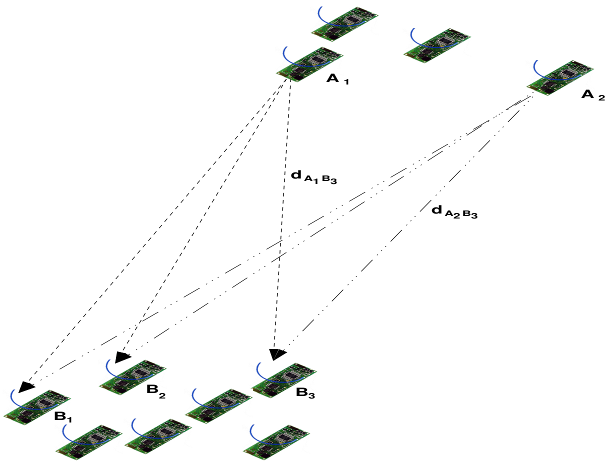


Figure 7.33: Distributed adaptive beamforming with multiple transmit and receive nodes

so that a set $\mathcal{B} = \{B_1, \dots, B_m\}$ of receive nodes observes the superimposition

$$\zeta_{\text{sum}\smallsetminus i} = \Re\left(m(t)e^{j2\pi f_c t}\sum_{o\in[1,n];o\neq i}\text{RSS}_o e^{j(\gamma_o+\phi_o+\psi_o)}\right) \tag{7.32}$$

of these nodes on the channel.

Since the transmission-distances $d_{A_iB_j}$ between node $A_i \in \mathcal{A}$ and node $B_j \in \mathcal{B}$ are distinct for all receive nodes B_j relative to each individual transmit node $A_i \in \mathcal{A}$, the received superimposition

$$\zeta_{\text{sum}\smallsetminus i} = \Re\left(m(t)e^{j2\pi f_c t}\sum_{o\in[1,n];o\neq i}\text{RSS}_o e^{j(\gamma_o+\phi_o+\psi_o)}\right) \tag{7.33}$$

at set $\mathcal{B}$ differs among nodes $B_j \in \mathcal{B}$.

Consequently, the relative phase-offset among received transmit signal-components

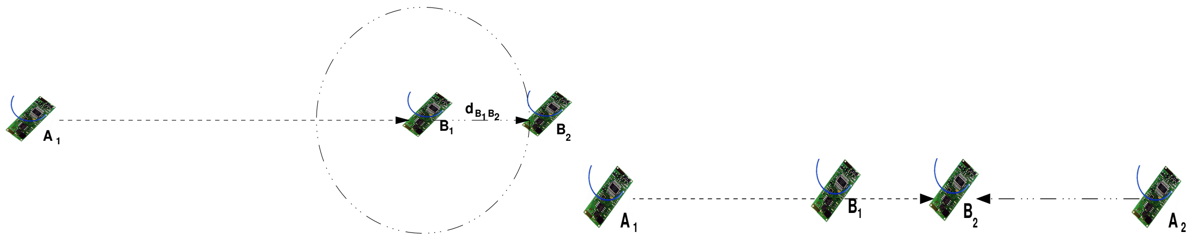
$$\zeta_i = \Re\left(m(t)\text{RSS}_i e^{j2\pi f_c t(\gamma_i+\phi_i+\psi_i)}\right) \tag{7.34}$$

i.e. the amount of synchronisation among received signal-components differs for distinct receive nodes.

The amount of this offset is determined by the location of transmit nodes, the transmission-distance and the distance between receive nodes and the transmit frequency.

Most importantly, the transmit frequency f and the distance among receive nodes limit the difference in the amount of synchronisation.

For two nodes the maximum phase-offset among these nodes is achieved when both nodes and a transmit node that alters the phase-offset are placed on a straight line (cf. figure 7.34(a)) since the difference in the transmission-distance is then at its maximum.



(a) When two receive nodes are located in a straight line, the difference in the phase-offset observed by these nodes of the received signal to a reference signal is maximal

(b) The greatest possible phase difference at two receive nodes is maximal when all nodes are placed on a straight line with transmit nodes maximally separated

Figure 7.34: Transmission among nodes in a straight line

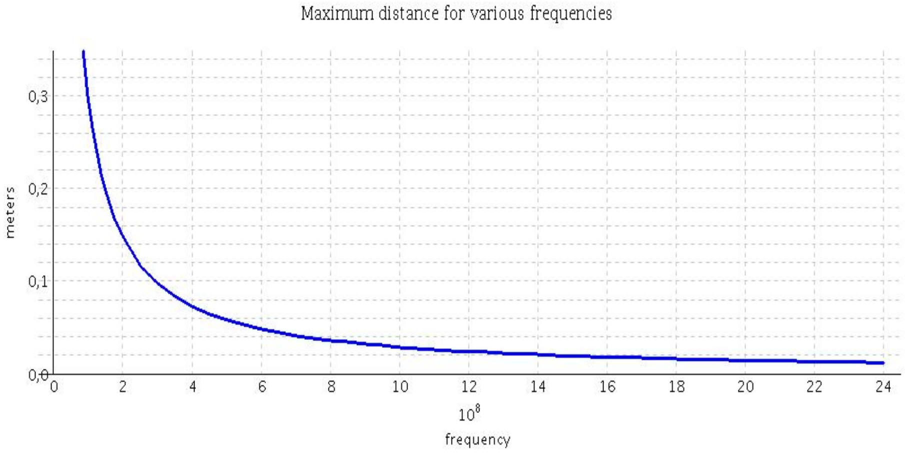


Figure 7.35: Maximum distance among nodes for various transmit frequencies ($\frac{a\text{max}}{4} \cdot \lambda$)

For a given distance d between nodes, the maximum relative phase-offset for two transmit signals is observed when transmit nodes are located at opposite directions of receive nodes in a straight line (cf. figure 7.34(b)).

Therefore, when a maximum relative phase-offset $\gamma_{\max} = a_{\max} \cdot \pi$ among any two nodes in a set of receive nodes $\mathcal{B}$ is allowed, a simple bound on the maximum distance allowed between any two nodes is given as $\frac{a_{\max}}{4} \cdot \lambda$. We can see this as follows. Since the wavelength $\lambda = \frac{c}{f}$ is proportional to a whole signal period of 2π, the maximum allowed phase-offset of $a_{\max}\pi$ translates to $\frac{a_{\max}}{2}\lambda$. In the worst case receive signals propagate in opposite directions so that the maximum distance allowed is at most $\frac{a_{\max}}{4}\lambda$. For several common frequencies figure 7.35 calculates the maximum distance among any two nodes in a set of receiving nodes.

As we observe from the table, this multiple receiver approach is only feasible for networks of a high density and for low transmit frequencies.

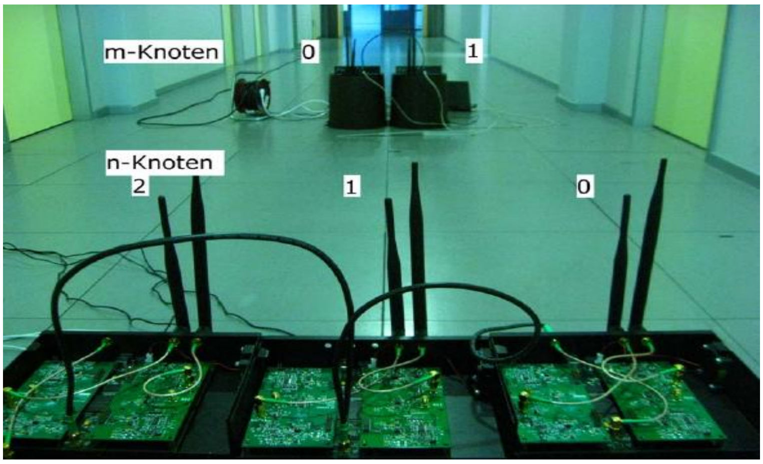


Figure 7.36: Setting for the Experiment with multiple receivers at two sets of nodes.

A protocol for distributed adaptive beamforming with multiple receivers

A more practical but also more energy-consuming approach therefore is to determine a coordinator node among a set $\mathcal{B}$ of receive nodes that publishes the signal quality it received to all other nodes in the set of receive nodes. Receive nodes can then provide a common feedback as the superimposition is then taken from one distinct position. However, since additional communication among nodes is required, this approach is more energy- and time-consuming than the original one. The protocol to achieve phase-synchronisation among two sets of distributed nodes is depicted in table 7.3

With this approach it is also possible to establish a duplex beamforming-channel between two sets of nodes, $\mathcal{A}$ and $\mathcal{B}$, when this synchronisation is repeated for set $\mathcal{B}$ then.

Experimental evaluation of 1-bit feedback-based beamforming with multiple receive nodes

We conducted experiments in a distributed adaptive beamforming scenario with multiple receivers. For the experiment we utilised a purely random search approach to improve the synchronisation among nodes. For the experiments, 5 USRP software radios[2] have been utilised with two nodes in one set and three nodes in the other set (cf. figure 7.36).

The initialisation signal was transmitted at 2.4 GHz. Feedback is exchanged at 900 MHz.

All antennas are placed in a distance of 0.21 m. Transmit and receive node have been placed in distances of 3m, 12m and 24m. As expected in wireless sensor netwok settings, The radiated power of the software radios among nodes differs due to hardware-imperfections. For each experimental setting, 10 experiments were conducted.

Experimental results are depicted in table 7.4 We observe that the gain in the signal strength greatly differs for distinct nodes. Generally, the gain achieved in set m is higher

[2] More details on the universal software radio peripheral (USRP) is available from Ettus research at http://www.ettus.com

Step	Operations in set $\mathcal{A}$	Operations in set $\mathcal{B}$
1	Node A_i decides that it is willing to transmit data to a node in subset $\mathcal{B}$	
2	Node A_i gathers nodes in set $\mathcal{A}$ for distributed beamforming	
3	Nodes A_i from set $\mathcal{A}$ superimpose a synchronisation signal $\zeta_i = \Re\left(m(t)\text{RSS}_i e^{j2\pi f_c t(\gamma_i+\phi_i+\psi_i)}\right)$	
4		A node B_j from set $\mathcal{B}$ detects the increased energy level and measures the synchronisation quality of the received signal $\zeta_{\text{sum}} = \Re\left(m(t)e^{j2\pi f_c t}\sum_{i=1}^{n}\text{RSS}_i e^{j(\gamma_i+\phi_i+\psi_i)}\right)$
5		Node B_j gathers nodes in set $\mathcal{B}$ for distributed beamforming
6		Node B_j informs all participating nodes of the feedback to transmit
7		Participating nodes in $\mathcal{B}$ transmit their feedback-value • When signal quality was improved, nodes transmit for Δt • When signal quality is sufficient, nodes transmit for $\alpha\Delta t$. • Otherwise, nodes stay idle
8	Nodes in $\mathcal{A}$ measure the energy on the channel to interpret the feedback-value • When the energy level increases significantly for about Δt, this is interpreted as a positive feedback (go to 3) • When the energy level increases significantly for about $\alpha\Delta t$, the synchronisation is assumed completed (go to 9) • Otherwise it is interpreted as negative feedback (go to 3)	
9	Nodes in set $\mathcal{A}$ transmit data as a distributed beamformer	
10		Signals are received by the coordinator node of set $\mathcal{B}$ and forwarded to all nodes in the set.

Table 7.3: Protocol for distributed adaptive beamforming with multiple receive nodes.

	n0	n1	n2	m0	m1
Distance : 3 meters					
Median gain (to initial superimposed signal) [dB]	0.96	2.39	1.40	1.46	1.10
Median gain (to individual signal) [dB]	2.33	2.32	2.37	3.50	4.05
Distance : 12 meters					
Median gain (to initial superimposed signal) [dB]	1.24	0.63	1.39	2.06	1.47
Median gain (to individual signal) [dB]	2.53	1.09	2.00	2.74	4.18
Distance : 24 meters					
Median gain (to initial superimposed signal) [dB]	1.12	2.33	2.76	3.61	1.67
Median gain (to individual signal) [dB]	1.2	2.54	2.03	5.15	3.76

Table 7.4: Experimental results for random synchronisation of 1-bit feedback-based distributed adaptive beamforming with multiple receive nodes

than the gain achieved in set n. The reason for this is that set m receives superimposed signals from 3 nodes, while set m itself consists only of 2 nodes. Therefore, the margin for potential gain is higher for set m.

For all nodes the median gain to the initial amplitude, i.e. to a superimposition of various signals is depicted as well as the median gain to the amplitude achieved when an individual node is transmitting. The gain to the amplitude of an individual signal is up to 5 dB for the experiment with a distance of 24 meters between transmit and receive nodes.

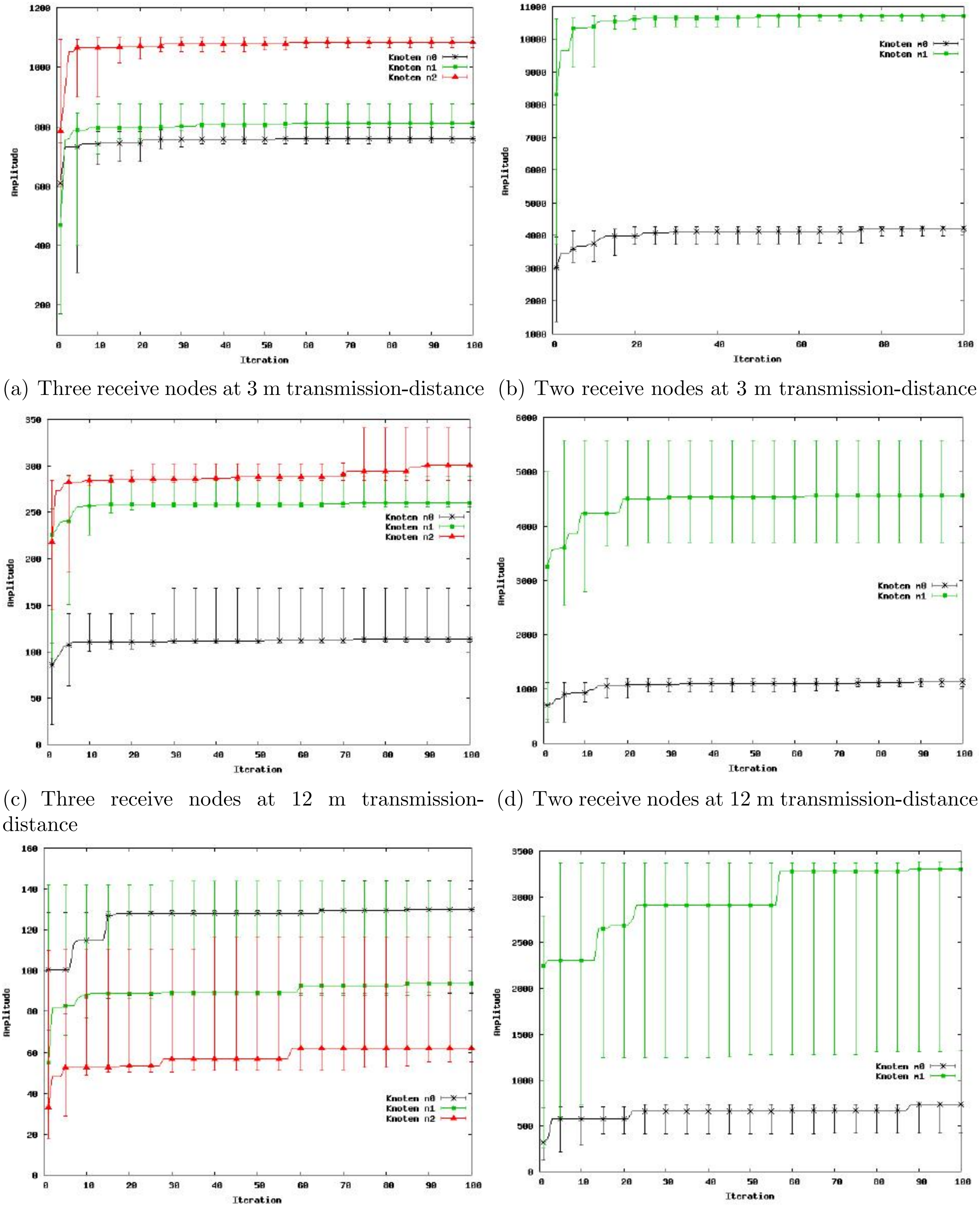
Generally, the gain compared to the initial amplitude of a superimposed signal is higher for greater distance between transmit and receive nodes. The reason is that with an increased distance the environmental noise has a greater impact on this value due to the increasing relative noise-level. Generally, the same effect can be observed for the gain of the signal amplitude compared to an individual signal. The maximum median gain achieved becomes greater with increasing distance.

Observe that for distinct nodes the maximum gain achieved differs among nodes in one set. The reason for this is that the antennas far separated too far from each other so that the signal-components received at distinct antennas are uncorrelated. The synchronisation conducted is therefore centred on one antenna. We also can observe these properties from the optimisation progress as depicted in figure 7.37.

7.5 Data transmission in collaborative beamforming

As we have observed, carrier-phase-adaptation among a high number of transmit signals is feasible with feedback-based carrier-synchronisation approaches. After carriers are synchronised, nodes then are to transmit a message signal $m(t)$ collaboratively. This means that all nodes modulate an identical signal-sequence onto their carrier and transmit this signal to the remote receiver simultaneously. In the optimum case, all signals are perfectly phase-aligned and the received signal is demodulated by the receiver.

Since, however, phases are typically not perfectly aligned, we expect errors in the mod-



(a) Three receive nodes at 3 m transmission-distance

(b) Two receive nodes at 3 m transmission-distance

(c) Three receive nodes at 12 m transmission-distance

(d) Two receive nodes at 12 m transmission-distance

(e) Three receive nodes at 24 m transmission-distance

(f) Two receive nodes at 24 m transmission-distance

Figure 7.37: Optimisation-progress for random synchronisation of 1-bit feedback-based distributed adaptive beamforming with multiple receiver nodes.

ulation so that the transmitted message $m(t)$ and the received message $m(t)'$ might differ even when noise and interference are disregarded.

The following discussion details a protocol to collaboratively transmit a message signal to a remote receiver.

The proposed protocol incorporates the synchronisation of multiple carriers as well as the transmission of a message $m(t)$.

All communicating devices utilise the iterative distributed carrier-synchronisation approach detailed in section 7 [16, 18]. We assume that each device maintains the following set of synchronisation parameters.

P_{mut} Probability to alter the phase-offset of an individual device ($P_{mut} \in [0, 1]$)

P_{dist} Probability distribution for the random process ($P_{dist} \in \{\text{normal, uniform}\}$)

var Variance for the random process ($var \in [0, \pi]$)

Each device i maintains its own synchronisation parameters $P_{mut,i}$, $P_{dist,i}$ and var_i.

The transmission protocol consists of the following steps that are executed in this order.

1. a single device willing to transmit a data sequence s_d broadcasts this sequence among devices in its proximity in order to initialise a collaborative transmission.

2. Devices decide whether to participate in the transmission. This decision may be dependent on decision parameters such as the energy level of devices, the required count of collaborating devices or the current computational load.

3. Carrier synchronisation is achieved by following the iterative approach proposed in [16]. Each device i utilises its synchronisation parameters $P_{mut,i}$, $P_{dist,i}$ and var_i for this process.

4. The receiver broadcasts an acknowledgement message which indicates that carrier-synchronisation is sufficient.

5. Directly after receiving the acknowledgement message, devices collaboratively transmit the data sequence s_d with the phase and frequency derived during step 3.

We implemented the protocol described for transmission via distributed adaptive beamforming for the simulation-environment detailed in table 7.8 on page 105. In this study we altered the count of devices that participate in the synchronisation and transmission process, the distance between the transmitting devices and the receiver as well as the transmit bit-rate at which the information was transmitted. In each simulation run, first the carrier phases were synchronised for 6000 iterations of the iterative random synchronisation algorithm also utilised in section 7.6.1. After synchronisation, devices simultaneously transmitted their data sequence following the proposed transmission protocol.

Figure 7.38 visualises the encoding and decoding process for an exemplary configuration. As you can observe from the figure, the data sequence utilised was an alternating 0-1-

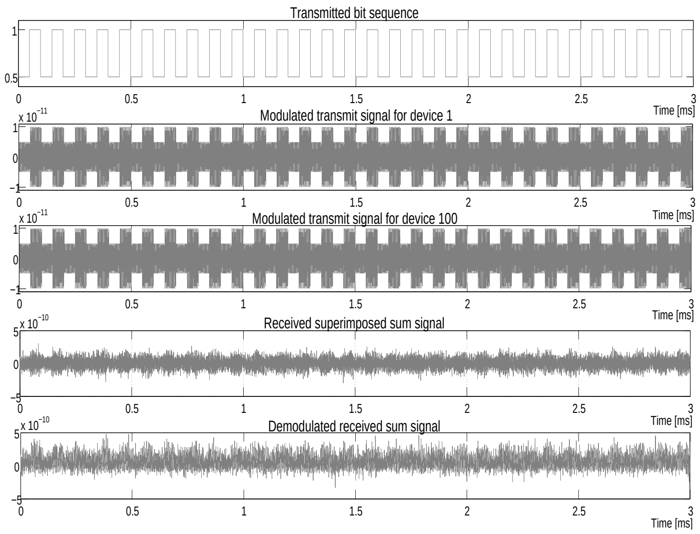


Figure 7.38: Exemplary encoding and decoding process of the protocol for a total of 100 participating devices at at transmission-distance of 1 meter and a data-rate of 20 kbps

sequence. The data-rate of this data-sequence was altered in different simulation runs. We utilised a simple amplitude modulation in which each symbol represents one bit. The individual modulated transmit signals are superimposed at the receiver. On receiving this sum-signal, the receiver demodulates the received sum-signal and retrieves the transmitted bit-sequence by a simple integration of the demodulated signal.

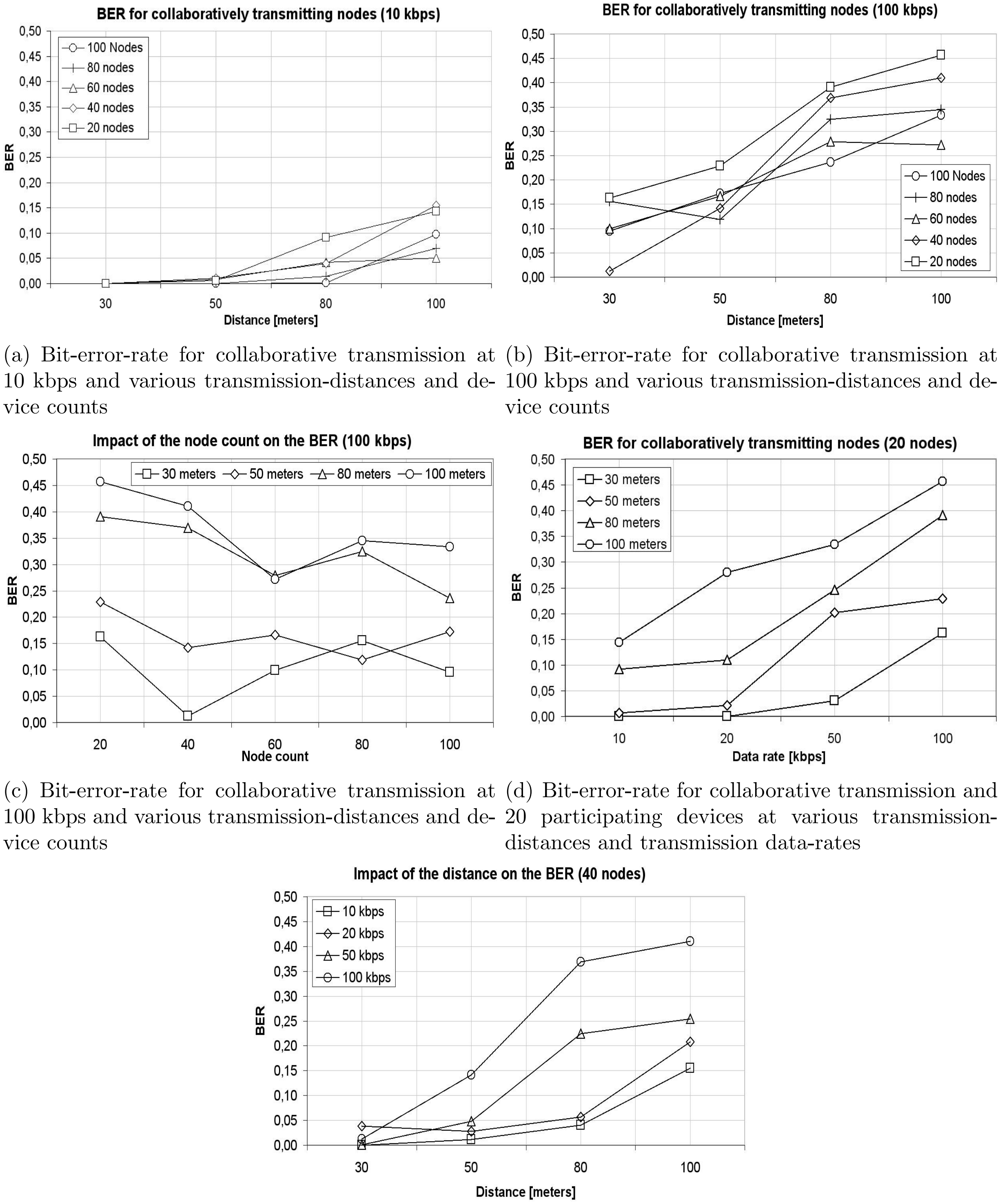
Figure 7.39 shows the observed median performance of the protocol for various simulation settings. In these figures, the median bit-error rate (BER) for 10 independent synchronisation and transmission processes is calculated. The BER is derived from a binary sequence $s \in \mathbb{B}^n$ and a received binary sequence $s' \in \mathbb{B}^n$ as

$$BER = \frac{ham(s, s')}{n}. \tag{7.35}$$

In this equation, $ham(s, s')$ denotes the hamming-distance between the sequences s and s'.

We observe that the BER is mainly impacted by the transmission-distance and the transmission data-rate. As we can observe from figure 7.39(a) and figure 7.39(b) the BER increases for data-rates of 10 kbps and 100 kbps with the transmission-distance for all numbers of participating devices considered.

However, the lines in these figures cross for several network sizes. In general, with increasing distance, also the noise-level rises relative to the strength of the superimposed received sum-signal. Therefore, when perfect carrier phase-synchronisation were achieved, we would expect a lower BER when more nodes participate in the transmission of a signal, since the received signal strength is then increased. However, due to noise and interference



(a) Bit-error-rate for collaborative transmission at 10 kbps and various transmission-distances and device counts

(b) Bit-error-rate for collaborative transmission at 100 kbps and various transmission-distances and device counts

(c) Bit-error-rate for collaborative transmission at 100 kbps and various transmission-distances and device counts

(d) Bit-error-rate for collaborative transmission and 20 participating devices at various transmission-distances and transmission data-rates

(e) Bit-error-rate for collaborative transmission and 20 participating devices at various transmission-distances and transmission data-rates

Figure 7.39: Bit-error-rates for a distributed adaptive beamforming protocol among a set of distributed wireless devices and various transmission-distances, transmission data-rates and device counts

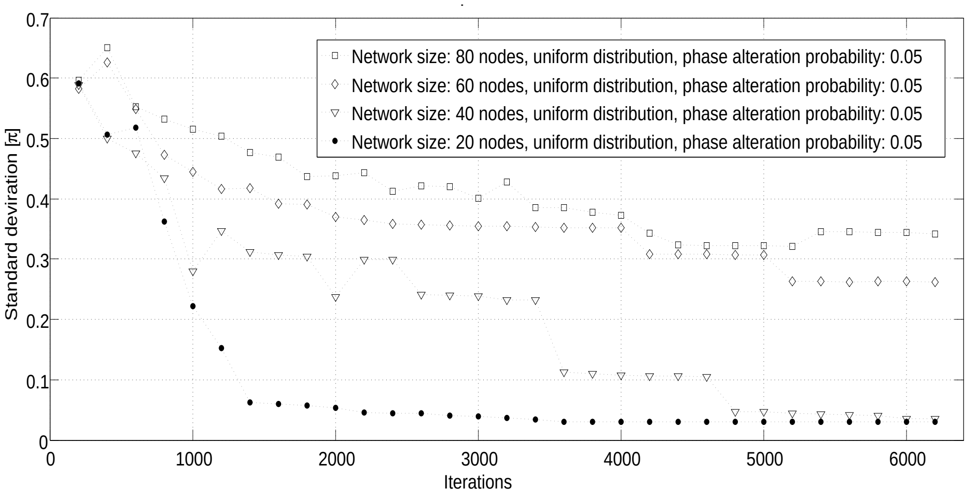


Figure 7.40: Standard deviation achieved for several network sizes with identical synchronisation parameters

effects we cannot expect that an optimum synchronisation is achieved. Also, it is more difficult to synchronise a higher number of transmit signal-components so that we can expect the synchronisation-quality to deteriorate with an increasing node count. We can see this from figure 7.40.

In the figure, synchronisation with identical environmental and algorithmic parameters is achieved with a differing count of nodes participating in the synchronisation. The median standard deviation from every ten simulations is plotted for these synchronisations. Clearly, the deviation in phases is smallest for small network sizes. Consequently, we conclude that the synchronisation of an increased number of devices can be expected to be deteriorate in accuracy.

Due to this ambiguity, the results are not decisive regarding this effect. When the distance is increased and the network size is kept constant, we can generally observe from figure 7.39(a) and figure 7.39(b) that the BER increases with increasing distance since the relative noise-power increases.

However, when both, network size and distance are altered simultaneously, we cannot observe a clear tendency since the increased signal strength for greater network sizes acts adversely to the reduced synchronisation quality for increasing network sizes (cf. figure 7.39(c)).

When the transmit data-rate is increased, we observe that also the BER increases as more bits are represented by one symbol (cf. figure 7.39(a) and figure 7.39(b)). We can also observe this correlation from figure 7.39(d).

Figure 7.39(e) summarises these observations in that the BER increases with increasing transmission-distance and data-rate.

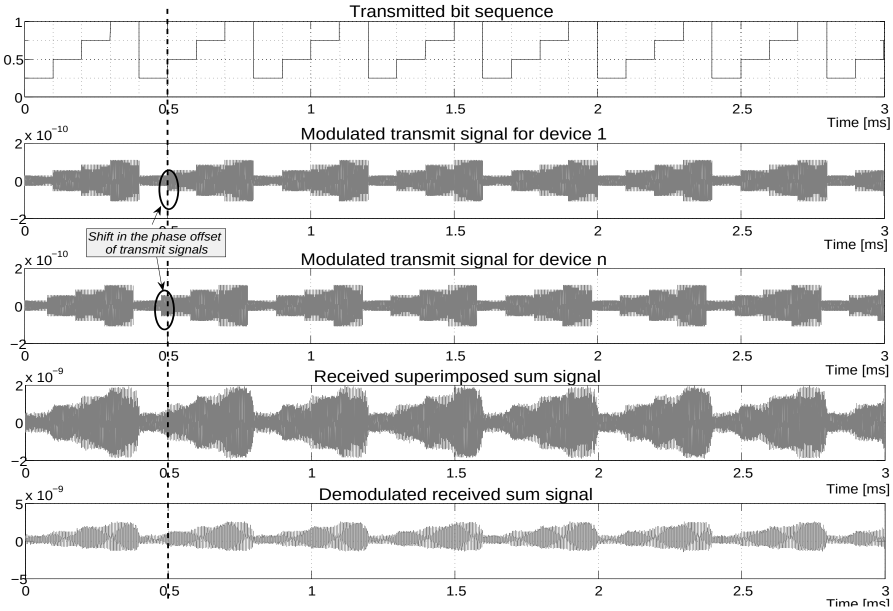


Figure 7.41: Modulation and demodulation of a simple symbol-sequence

This observation is remarkable for collaborative transmission among IoT-devices. Since the synchronisation-performance is directly impacted by the count of participating devices [101], it is beneficial when this count is just large enough to collaboratively reach a remote receiver. Greater numbers of transmitters do not lead to an improved BER but impair the synchronisation-performance.

Finally, we implemented higher-order-modulation schemes. In particular, we encoded two or three bits per symbol.

Figure 7.41 illustrates the process of modulation, transmission and demodulation for an amplitude modulation scheme that represents two bits per transmitted symbol.

We modulated a simple periodical symbol-sequence on the transmit-carrier of each single node. In the figure, we observe a considerable carrier phase-offset of the two nodes depicted. For this environmental setting we observe that the received superimposed signal is improved in its signal strength compared to a single transmit signal. Also, the symbol-sequence is clearly visible from the received superimposed signal. Consequently, the bit-error rate (BER) for this configuration is low. Figure 7.42 depicts the BER for two amplitude modulation schemes and for various transmission-distances.

In all simulations, 100 transmit-devices are utilised to superimpose their carrier-signals. In the second modulation scheme, three bits are represented by one transmit symbol. As expected, we observe that the BER is higher with the higher modulation scheme and with increasing distance. While it is neglectable for the weaker modulation scheme at a distance of 30 meters, the BER becomes significant with increasing distance for both modulation schemes.

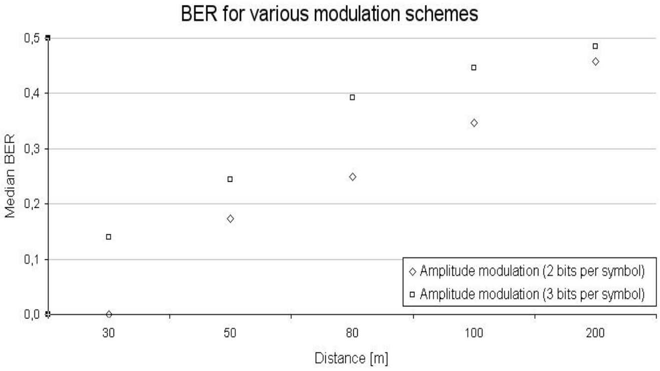


Figure 7.42: Bit-error-rate for two amplitude modulation schemes

7.6 On the impact of environment on carrier-synchronisation

In the last section we have discussed a protocol for the collaborative transmission of distributed devices. We have also observed that the BER for transmission with this protocol is dependent on the quality of synchronisation among transmit signal-components. However, since the synchronisation quality is dependent on environmental settings we propose an adaptive protocol for distributed beamforming that also incorporates an interactive optimisation of synchronisation parameters to improve the synchronisation quality at the instant of data transmission.

Section 7.6.1 will discuss impacts on the synchronisation quality and section 7.6.2 proposes and verifies an adaptive protocol for distributed adaptive beamforming.

7.6.1 Impacts on the carrier-synchronisation quality

Several parameters impact the carrier-synchronisation quality. The probably most obvious event for incomplete carrier-synchronisation is when the RSSI value received from simultaneously transmitting nodes is not saturated (i.e. the feedback from the receiver also is frequently improved at the end of the synchronisation process). This might be an indication that the synchronisation was not yet complete (cf. figure 7.2.4 on page 117). A possible reason for this characteristic is that the algorithm adapts the carrier phase of individual devices too frequently. When many devices in each iteration alter their carrier phase-offset, a steady-state optimisation is not achieved. Therefore, the synchronisation process is hindered. By decreasing the mutation variance var or the mutation probability P_{mut}, an improved synchronisation-performance and BER can be achieved (cf. figure 7.21(e)).

If, however, a saturation is achieved very early in the synchronisation process and the achieved gain in the RSSI is low, this might be an indication that the noise-figure at the receiver is higher than the impact achieved by altering few carrier phases. For such a case, it was shown in [102] that it is beneficial to increase the probability P_{mut} to alter the

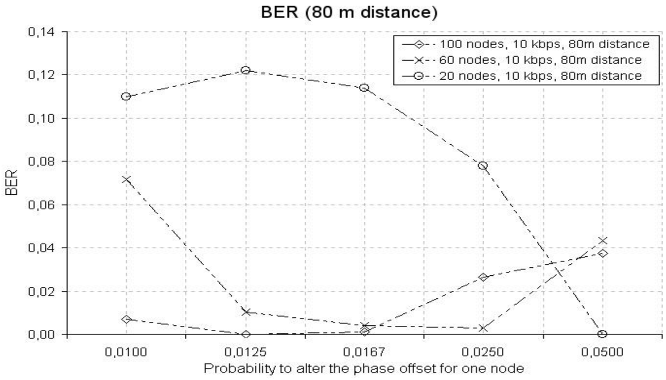


Figure 7.43: Impact of the network size on the BER at distinct phase alteration probabilities for one bit feedback-based distributed phase-synchronisation and adaptive transmit beamforming

phase-offset of a single device.

Another parameter that might impact the synchronisation-performance is the number of nods participating in synchronisation (cf. figure 7.20). Figure 7.43 shows the impact of the network size on the synchronisation process exemplarily. We observe that the best phase alteration probability to achieve an optimum bit-error rate (BER) for collaborative transmission differs among various network sizes. Consequently, optimum parameters have to be derived individually for each concrete scenario.

Other environmental parameters that impact the carrier-synchronisation quality are

1. The relative noise and interference-level at the receiver [102]

2. The distance between a network of nodes and a receiver [101]

3. The networks size [103]

4. The transmission-power of nodes [100]

In order to achieve an optimum synchronisation-performance, the following parameters can be adapted for feedback-based distributed adaptive transmit beamforming..

1. The mutation probability $P_{mut,i}$

2. Its distribution $P_{dist,i}$

3. The variance var_i

In the next sections we study the impact of various of these environmental parameters on the carrier-synchronisation quality in simulations and experiments.

Simulations

Again we utilise the simulation-environment detailed in Table 7.8 on page 105.

Each simulation run lasts for 6000 iterations and one iteration consists of the devices transmitting, feedback computation, feedback transmission and feedback interpretation by the distributed devices. It is possible to perform these steps within a few singal periods, so that the time consumed for a synchronisation of 6000 iterations is in the order of milliseconds for a base band signal frequency of 2.4 GHz. We expect environmental fluctuations to be minor in this short time-period. The quality of a signal is measured by the RMSE of the received signal (see equation (7.6)) on page 104.

Figure 7.44 depicts exemplary results from our simulations that show the impact of parameter settings in changing environments.

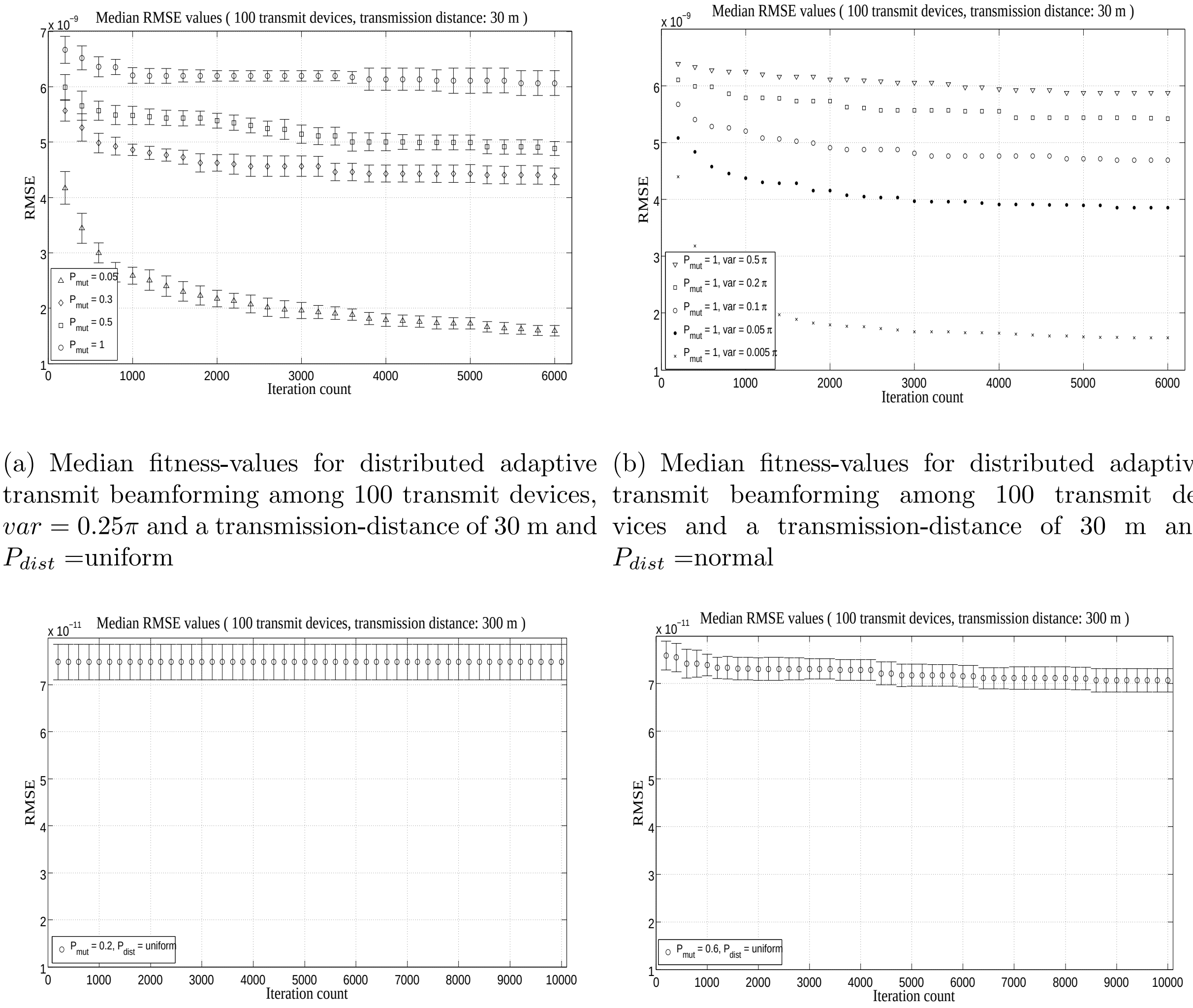
As we can observe from figure 7.44(a) the optimisation parameters such as the phase alteration probability P_{mut} impact the synchronisation-performance. With a lower probability P_{mut}, the simulation performance is improved. A similar impact can also be observed for the variance var of the phase alteration process as depicted in figure 7.44(b). By decreasing the variance of the phase alteration process the synchronisation-performance is improved for the scenario considered.

So far, we observed that it was beneficial to decrease the stepwidth of the iterative algorithm by reducing the variance and the probability applied for the phase alteration process. However, in different environments, a high stepwidth may also be beneficial. In figure 7.44(c) and figure 7.44(d) we depict the synchronisation-performance of the iterative algorithm in a scenario in which the noise-level is reasonably higher than the signal strength of an individual received signal-component. In such a scenario, a low mutation probability leads to a bad synchronisation-performance since the alterations in one iteration are often too small to have a significant impact on the signal (cf. figure 7.44(c)). With an increased probability to alter the phase-offset, however, the impact of several devices altering their carrier phase-offsets simultaneously is sufficient to achieve a weak synchronisation in the same environmental setting (cf. figure 7.44(d)).

Experimental study

In order to approximate realistic conditions in an experimental setting we utilised USRP software radios (http://www.ettus.com) to represent distributed adaptive beamforming devices. Three USRP devices have been utilised as collaborative transmitters and one device as a receiver. In order to achieve identical transmit frequencies among devices, the clock of the first transmit device was utilised as a reference clock for the two remaining transmitters as depicted in figure 7.45. This is a simplification compared to an actual instrumentation in which clocks could be synchronised via GPS or, for instance, by the iterative frequency synchronisation approach described in [42]. Table 7.6.1 summarises our experimental configuration.

In these experiments we repeatedly synchronised the carrier phases of the three transmit devices with the help of the 1-bit feedback-based algorithm described in [16].



(a) Median fitness-values for distributed adaptive transmit beamforming among 100 transmit devices, $var = 0.25\pi$ and a transmission-distance of 30 m and P_{dist} =uniform

(b) Median fitness-values for distributed adaptive transmit beamforming among 100 transmit devices and a transmission-distance of 30 m and P_{dist} =normal

(c) Median fitness-values for distributed adaptive transmit beamforming among 100 transmit devices, $P_{mut} = 0.2$, uniform probability distribution and a transmission-distance of 300 m

(d) Median fitness-values for distributed adaptive transmit beamforming among 100 transmit devices, $P_{mut} = 0.6$, uniform probability distribution and a transmission-distance of 300 m

Figure 7.44: Performance of distributed adaptive beamforming in an Internet of Things for various environmental settings

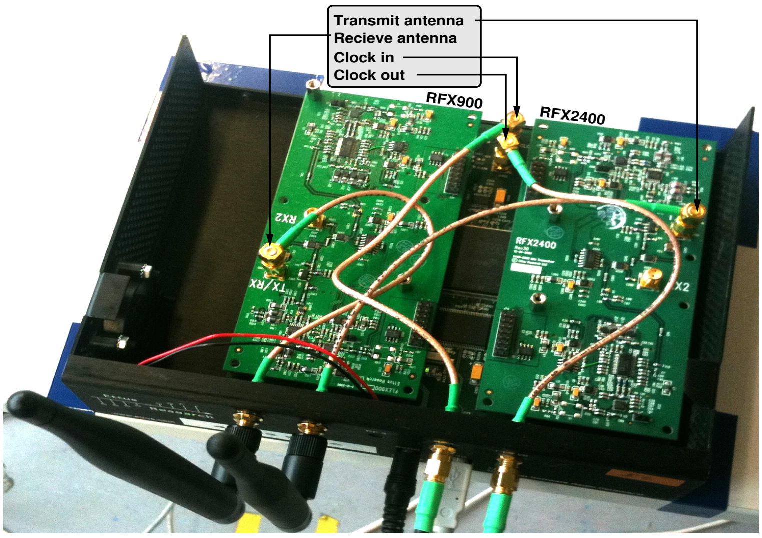


Figure 7.45: USRP transmit devices are synchronised in their clock

Experimental setting	
Separation of transmit antennas [m]	0.44
Distance to receive antenna [m]	5.5 / 11 / 16.4
Transmit frequency [MHz]	$f_{TX} = 2400$
Receive frequency [MHz]	$f_{RX} = 902$
Iterations per experiment	400
Mobility	stationary
Identical experiments	12
Transmit devices	3
Receive devices	1
Algorithmic configuration	
Random distribution of the phase alteration	normal distribution
Phase alteration probability	0.33 / 0.66 / 1.00
Variance for normal distributed phase-offset [π]	0.25 / 1
Hardware	
Transmit board	RFX2400
Receive board	RFX900
Transmit antenna	VERT2450
Receive antenna	VERT900
Gain of receive antenna [dBi]	$G_{RX} = 3$
Gain of transmit antenna [dBi]	$G_{TX} = 3$

Table 7.5: Configuration of the USRP experiment

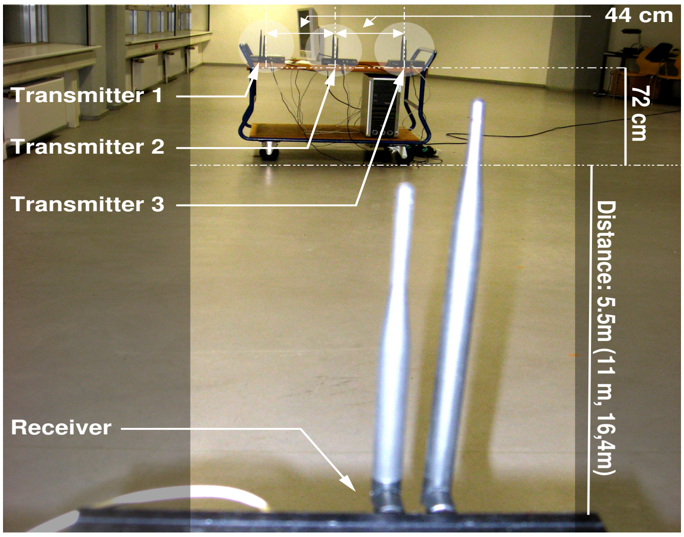


Figure 7.46: Experimental instrumentation of distributed adaptive beamforming among three transmit USRP devices and one receive USRP device.

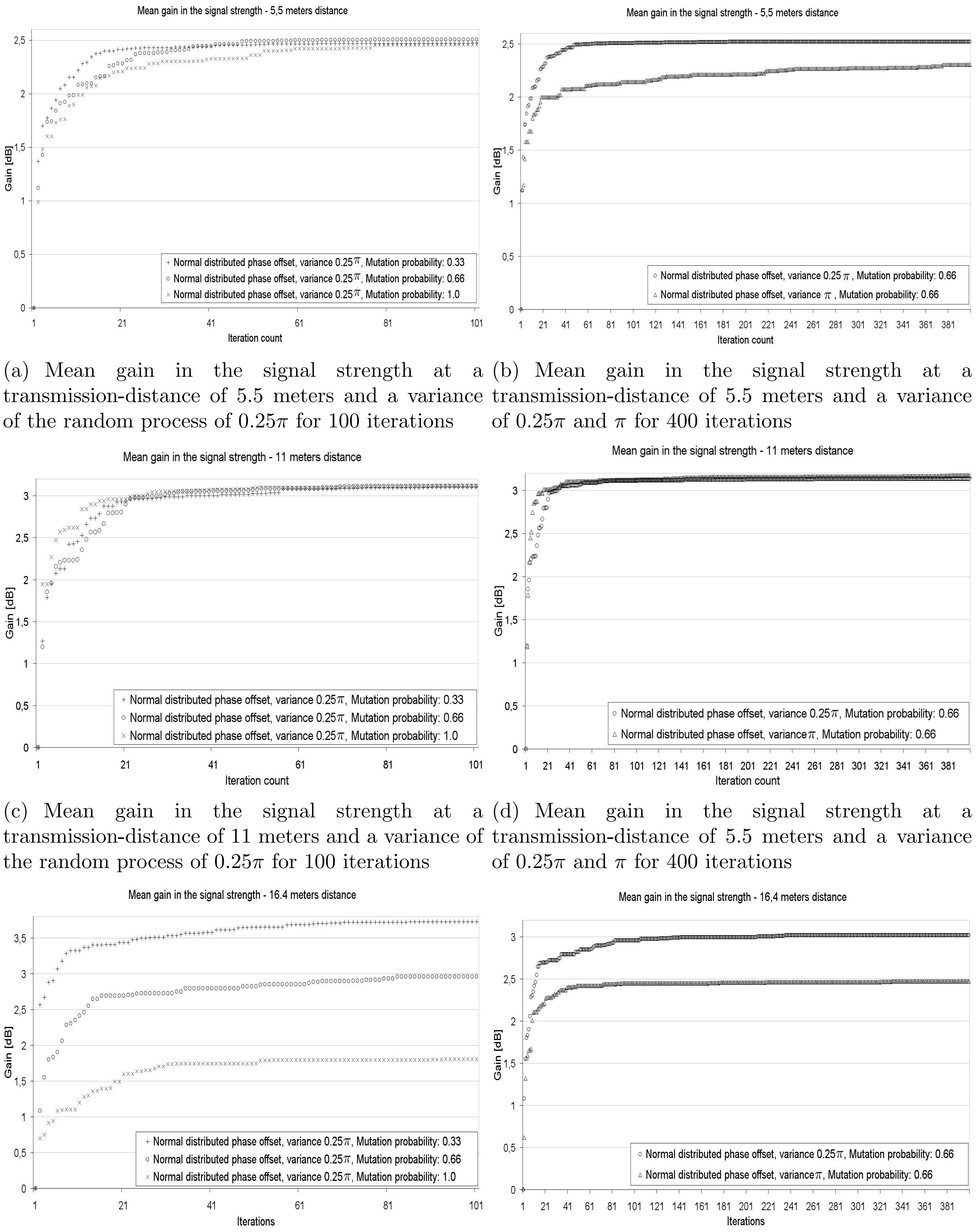
Carrier phases have been adapted for each transmit device independently following a normal distributed random process. We modified the probability of the alteration of the phase-offset of one device and the variance for its normal distributed random process as well as the distance between transmit and receive devices to account for distinct environments of distributed installations. Figure 7.46 depicts our experimental setting.

Results derived in these experiments are detailed in figure 7.47. We can observe from the figures that the synchronisation process differs for different environmental situations. When the transmission-distance increases, the best synchronisation is generally reached later in the synchronisation process. For instance, in figure 7.47(a), at a transmission-distance of 5.5 meters, the best value is reached after about 40 iterations. In figure 7.47(c) and figure 7.47(e) (11 meters and 16.4 meters) we observe that the optimum synchronisation is reached after about 50 and 60 iterations. Also, the choice of the optimum configuration differs dependent on the scenario. While in figure 7.47(e) at a distance of 16.4 meters a variance of 0.25π with a probability to alter the phase-offset of $P_{mut} = 0.33$ achieves the best results, at shorter distances, the configuration with $P_{mut} = 0.66$ results in a slightly better synchronisation-performance.

Consequently, dependent on the environmental situation in which a synchronisation takes place, different parameter configurations lead to an optimum synchronisation-performance. While for installations in one room a higher mutation probability might be beneficial, this might change in another room where communicating devices are arranged differently.

7.6.2 Adaptive distributed beamforming protocol

We have observed that environmental changes impact the synchronisation-performance of 1-bit-feedback-based distributed adaptive transmit beamforming. These changes may



(a) Mean gain in the signal strength at a transmission-distance of 5.5 meters and a variance of the random process of 0.25π for 100 iterations

(b) Mean gain in the signal strength at a transmission-distance of 5.5 meters and a variance of 0.25π and π for 400 iterations

(c) Mean gain in the signal strength at a transmission-distance of 11 meters and a variance of the random process of 0.25π for 100 iterations

(d) Mean gain in the signal strength at a transmission-distance of 5.5 meters and a variance of 0.25π and π for 400 iterations

(e) Mean gain in the signal strength at a transmission-distance of 16.4 meters and a variance of the random process of 0.25π for 100 iterations

(f) Mean gain in the signal strength at a transmission-distance of 5.5 meters and a variance of 0.25π and π for 400 iterations

Figure 7.47: Mean gain in the signal strength of three collaboratively transmitting devices

originate from the presence or absence of persons, opened or closed windows or doors, changes in temperature, moving nodes, nearby interference sources or also moved furniture. An algorithm can react to these changes by altering the parameters P_{mut}, P_{dist} and var.

In order to account for these environmental impacts, we propose a protocol for distributed adaptive beamforming that incorporates self-adaptation and self-optimisation features. The protocol adapts the parameters P_{mut}, P_{dist} and var of the iterative synchronisation to a given environment so that an optimum synchronisation-performance is achieved. We extend the protocol presented in section 7.5 accordingly

1. An individual device broadcasts a data sequence s_d to devices in its proximity.
2. Devices decide whether to participate in the transmission. Possible decision parameters are, for instance, the energy level, a required count of participating devices or the current computational load.
3. Closed-loop one bit feedback-based carrier-synchronisation is achieved. Devices utilise $P_{mut,i}$, $P_{dist,i}$, var_i.
4. Upon sufficient synchronisation the receiver broadcasts an acknowledgement.
5. Optimisation parameters $P_{mut,i}$, $P_{dist,i}$ and var_i are adapted.
6. Devices collaboratively transmit s_d.

For a given environment, a set of devices can improve its synchronisation-performance after several transmissions with this protocol.

The protocol is self-adaptive to a given environment and self-healing as it automatically adapts the optimisation parameters to changing numbers of participating devices or to communication topologies. Several learning methods can be applied in step 5 to alter the optimisation approach.

We evaluate this protocol in mathematical simulations in a network of 100 devices. The scenario of distributed adaptive beamforming in an environment of distributed devices was implemented in Matlab. Devices are distributed uniformly at random on a $30m \times 30m$ square area with a receiver located up to $200m$ above the centre of this area. All devices are stationary and frequency and phase stability are considered perfect (cf. Table 7.8) on page 105.

Each simulation lasts for 6000 iterations and one iteration consists of the devices transmitting, feedback computation, feedback transmission and feedback interpretation.

The signal quality of a signal during the synchronisation phase is measured by the RMSE as defined in equation (7.6) on page 104.

We utilise an optimisation approach that implements a uniform distribution to alter the phase-offset of distributed carrier-signals. Consequently, only the mutation probability $P_{mut,i}$ of devices $i \in [1..n]$ is altered. After each 10 successful synchronisations, the mean achieved RMSE is compared to recently achieved RMSE values. The phase alteration

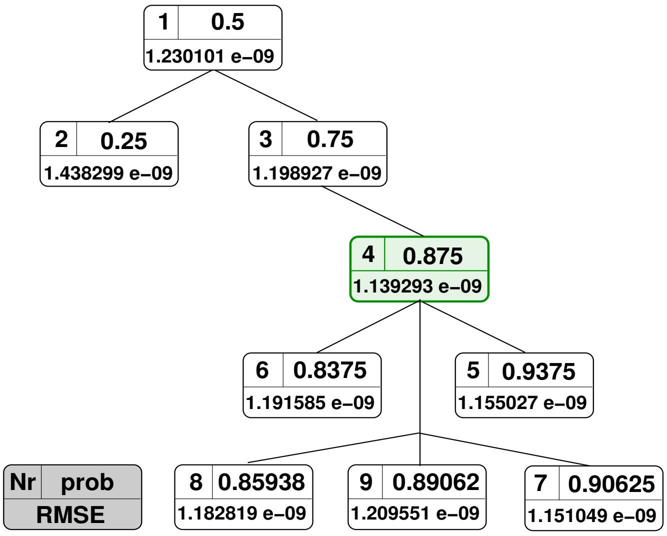


Figure 7.48: Schematic of the optimisation process of the proposed protocol. RMSE values depicted denote the mean RMSE after 10 synchronisations with identical $P_{mut,i}$

probability is adapted accordingly. As all nodes receive identical feedback from the receiver, this adaptation process is identical among devices.

We implement the search for the optimum mutation probability as a simple divide-and-conquer approach. Nodes start with a mutation probability of $P_{mut,i} = 0.5$ and then subsequently approximate the optimum probability by testing those parts of the search-space with lower and higher probability. The process always follows the phase alteration probability with the best achieved RMSE after 10 synchronisations.

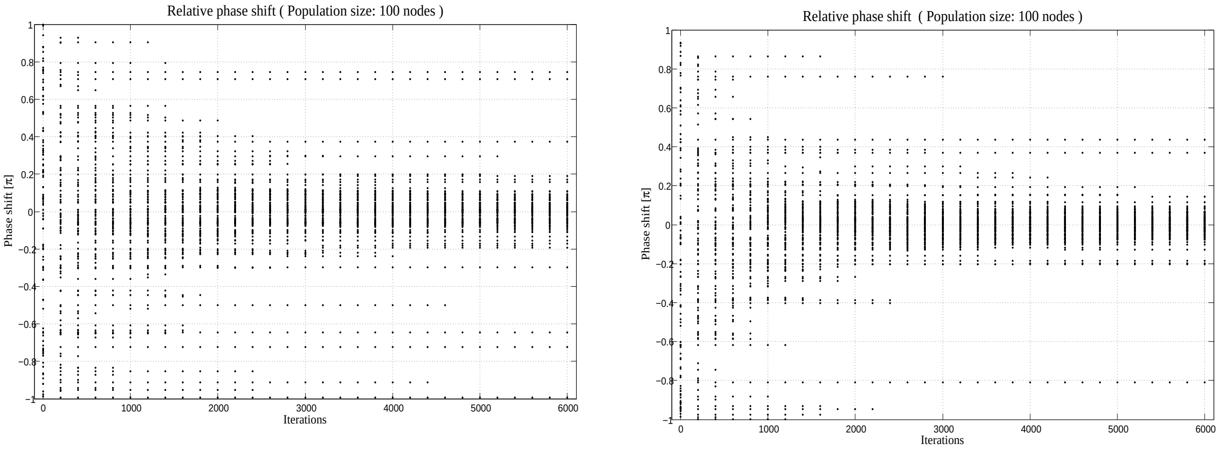
Figure 7.48 schematically illustrates this optimisation process in a network of 100 devices that are located approximately 30 meters from a remote receiver.

All devices first complete synchronisations with $P_{mut,i} = 0.5$ and then derive the mean RMSE values for $P_{mut,i} = 0.25$ and $P_{mut,i} = 0.75$. Since the latter probability achieves a better RMSE in this simulation, the lower half of the probability space ($P_{mut,i} \in [0, 0.5]$) is disregarded in the synchronisation process. With 0.875 a probability is reached for which no further improvement is found. In order to derive an optimum value, the algorithm tests three additional probability values in the proximity of the best value reached so far and then exits with the derived optimum probability of $P_{mut,i} = 0.875$ in this case.

Figure 7.49 depicts the relative phase-offset of all carrier-signal-components during the course of the synchronisation for $P_{mut,i} = 0.5$ and $P_{mut,i} = 0.875$. Clearly, the synchronisation is improved as a result of the search for an optimum probability $P_{mut,i}$.

We can see this result also from the sequence of RMSE values observed by the remote receiver as depicted in figure 7.50. The synchronisation with $P_{mut,i} = 0.875$ is faster and achieves an improved RMSE during the synchronisation.

In general, the algorithm searches the probability space in a binary search fashion in order



(a) Distributed carrier phase-synchronisation with $P_{mut,i} = 0.5$ and a transmission-distance of 30 m

(b) Distributed carrier phase-synchronisation with $P_{mut,i} = 0.875$ and a transmission-distance of 30 m

Figure 7.49: Relative phase-offset achieved during distributed carrier phase-synchronisation processes

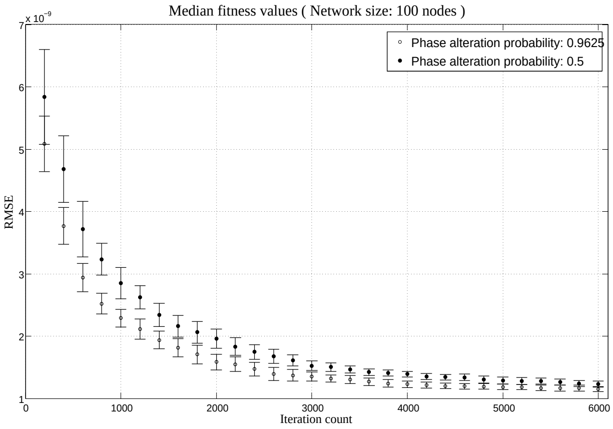


Figure 7.50: Median RMSE values achieved in the course of the synchronisation

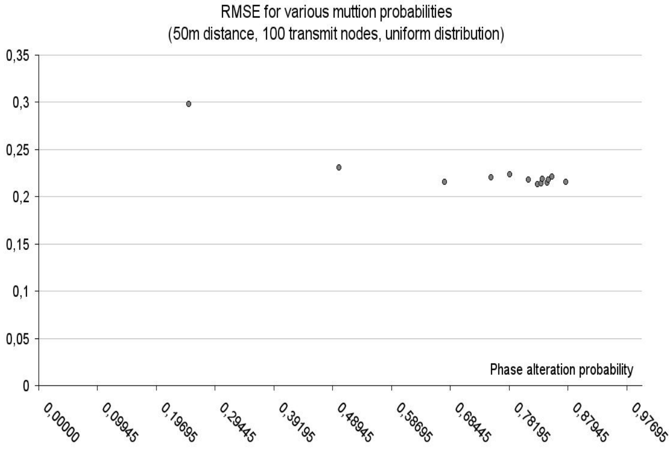


Figure 7.51: Mean RMSE values after each 10 synchronisations for various phase alteration probabilities $P_{mut,i}$

to bound the optimum mutation probability. For a second set of simulations we derived the optimisation process of this algorithm when transmitters are located 50m apart the receiver. Figure 7.51 depicts the RMSE values achieved in this process in an environment where the receiver is located 50 meters away from the transmit devices.

Algorithm 1 depicts a pseudo-code of this algorithm.

At the beginning, the algorithm starts with a probability of 0.5 (line 2) and no restrictions on P_{mut} (line 3-4).

In each iteration, the carrier phase-offset for one parameter configuration is derived ten times (lines 7-9). This is a design parameter. When increased, the confidence on the median RMSE increases at the cost of decreased search speed. If decreased, however, the search speed increases but the search might be more vulnerable to process noise.

After the median RMSE is calculated for $P_{mut} = 0.5$, it is then computed for $P_{mut} = 0.25$ (lines 11- 13). If this value is improved compared to $P_{mut} = 0.5$, the search is continued in the lower half of the probability space $[0, 0.5]$ (lines 14-18). Otherwise, $P_{mut} = 0.75$ is computed (lines 19-21). This process is continued until the median RMSE could not be improved for three distinct consecutive values of P_{mut} (lines 28-33). This value of three is, of course, a design parameter. If greater accuracy is required, it can be increased at the cost of an increased overall search time.

7.7 Alternative learning approaches

The simple binary search approach presented in the last section is one possible method for distributed nodes to learn about environmental settings and adapt optimisation parameters accordingly. This simple approach can be applied to adapt parameters to a fixed environment. Clearly, arbitrary learning approaches can be used instead. As one class of learning methods capable of doing so we propose learning classifier systems.

Algorithm 1 Adaptive distributed beamforming with a binary search approach

```
$n = 0$ % count of negative search attempts
probability = 0.5
lowerBorder = 0
upperBorder = 1
currentValue = init
while $n \leq 3$ do
    for i=1:10 do
        synchronise nodes and calculate median RMSE fitness achieved
    end for
    switch currentValue
    case init:
    currentValue = lower
    probability = (probability-lowerBorder)/2
    case lower:
    if RMSE value better than recent RMSE value then
        currentValue := lower
        upperBorder = probability
        probability = (probability-lowerBorder)/2
    else
        currentValue := upper
        probability = probability + (upperBorder - probability)/2
    end if
    case upper:
    if RMSE value better than recent RMSE value then
        currentBound = lower
        lowerBorder = probability
        probability = probability + (upperBorder - probability)/2
    else
        n++
        currentBound = lower
        lowerBorder = probability - (probability - lowerBorder)/2
        upperBorder = probability + (upperBorder - probability)/2
        probability = probability - (probability - lowerBorder)/2
    end if
    end switch
end while
```

The principle idea of a learning classifier system (LCS) is to combine a discovery and a learning approach [104]. In the discovery part, a search algorithm is implemented in order to iteratively improve the parameter settings applied. In the learning part, patterns in the observed performance and parameters chosen in the discovery part are analysed in order to adapt to a given environment. With these approaches is is, in particular, possible to implement a constant learning method to constantly adapt to a changing environment. We implemented a simple LCS approach for the same scenario as above. Our implementation is detailed in algorithm 2.

Algorithm 2 Adaptive distributed beamforming with an LCS search approach

```
 1: populationSize = $\mu$
 2: offspringPopulationSize = $\lambda$
 3: Initialise population of $\mu$ individual, each with $P_{mut}$, $P_{dist}$ and $var$
 4: synchronise nodes for each individual and calculate median RMSE fitness achieved
 5: while maximum population count not reached do
 6:    create $\lambda$ individuals from the $\mu$ individuals by applying $P_{mut}$, $P_{dist}$ and $var$
 7:    for all individuals in the offspring population do
 8:       synchronise nodes and calculate median RMSE fitness achieved
 9:       % might be repeated to derive median RMSE values for each individual
10:    end for
11:    select $\mu$ best rated individuals for the offspring population
12: end while
```

For the discovery part, we utilise an evolutionary algorithm. The learning part is constituted of storing the best parameter setting reached so far. Clearly, this implementation is only feasible in a static environment. In a changing environment, a more ambiguous learning method is required in order to learn and update the parameter settings continuously. After the initialisation of the population in line 3, carrier phases are synchronised (line 4) and the evolutionary optimisation is executed in lines 5-12.

For an exemplary optimisation process we detail the first three generations of this algorithm with a population size and offspring population size of 5 and plus selection in figure 7.52.

Clearly, we can implement arbitrary search approaches for the LCS system. In algorithm 3, for instance, we exemplarily detail a learning classifier system that features a metropolis search approach.

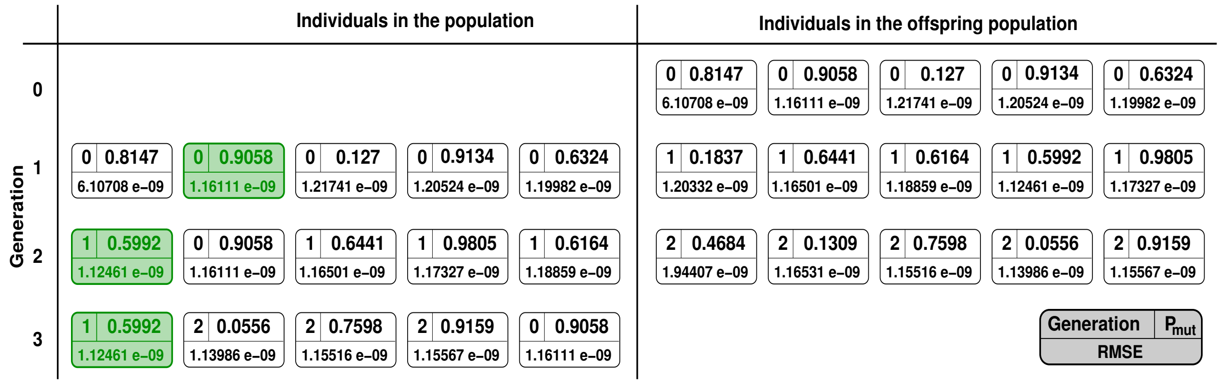


Figure 7.52: Schematic of the optimisation process of the protocol utilising the LCS learning approach with an evolutionary algorithm. RMSE values depicted denote the mean RMSE after 10 synchronisations with identical $P_{mut,i}$

Algorithm 3 Adaptive distributed beamforming with an M-LCS search approach

```
% This algorithm is repeated in a multistart strategy
initialise search-point with $P_{mut}$, $P_{dist}$ and $var$ % random search-point
synchronise nodes and calculate median RMSE fitness achieved % might be repeated to
  derive median RMSE values
while stop criterion not met do
    create new search-point by applying $P_{mut}$, $P_{dist}$ and $var$
    synchronise nodes and calculate median RMSE fitness achieved for this search-point
      % might be repeated to derive median RMSE values
    if RMSE of new search-point is better than RMSE of old search-point then
        discard old search-point and continue with new search-point
    else
        if random decision positive then
            % probability to accept worse points inverse to decrease in fitness-value
            discard old search-point and continue with new search-point
        else
            discard new search-point
        end if
    end if
end while
```

8 Conclusion

> *Throughout these lectures I have delighted in showing you that the price of gaining such an accurate theory has been the erosion of our common sense.*
>
> (Richard P. Feynman, QED, Chapter 3 [105])

We have studied approaches to transmit a data sequence from various distributed nodes collaboratively. In particular, our studies have focused on a computationally not complex and resource-efficient approach that implements an interactive carrier-synchronisation-mechanism and is guided by feedback from a remote receiver.

This one-bit-feedback-based carrier-synchronisation was analysed analytically in section 7. We observed that this carrier-synchronisation approach incorporated a random evolutionary search approach. We derived a sharp asymptotic bound on the expected synchronisation-time of $\Theta\left(n \cdot \log\left(n \cdot \log\left(k\right) + k\right)\right)$ in section 7.2. In this formula, n denotes the number of transmitting nodes and k represents the granularity with which carrier phases are approximated by distinct nodes. This means that the synchronisation-time depends on the network size and the count of distinct phases a transmitter can distinguish between.

Also, we considered several modifications of this basic algorithm in order to improve synchronisation-time. For a local random search mechanism with a neighbourhood-size N we bound the asymptotic synchronisation-time by $\Theta\left(n \cdot \log\left(\frac{n\cdot\log(k)}{n\cdot\log(k)-n\cdot N} + k\right)\right)$ in section 7.3.2.

Apart from these two general search approaches, the synchronisation-performance can also be improved by utilising a set of $m < n$ pre-synchronised nodes for transmission (see section 7.4) or by re-electing successful nodes (see section 7.4). For all algorithms and modifications proposed, we derived simulation results in a mathematical simulation-environment. In these simulations the impacts of environmental settings and algorithmic parameters were studied. Also, the velocity of receive and transmit nodes was taken into consideration in these simulations. The basic optimisation approaches were also applied in experimental settings utilising USRP software radios.

Furthermore, we examined a situation in which carrier phase-synchronisation is required both at transmit and receive side in section 7.4.2. For this scenario we proposed a modification of the one-bit-feedback-based carrier-synchronisation approach in which the binary feedback modulated onto a standard carrier is replaced by the measurement of energy on the channel. This setting was implemented in an experimental set-up with five USRP software radios.

Regarding the environmental parameters node count, transmission-distance, transmission power, noise and interference as well as the algorithmic parameters P_{int_i} (probability to alter the phase-offset of node i), $P_{dist,i}$ (distribution of this probability) and var (variance of the phase alteration probability), we presented simulation results that show the impact of these parameters on the synchronisation-performance. Generally, we observed that for environments with a higher relative noise-figure it is beneficial to increase the stepwidth in each iteration. This means that the probability $P_{mut,i}$ and the variance var should be higher in environments with a higher relative noise-figure (e.g. increased transmission-distance, reduced transmission power, increased interference from other nodes, urban scenarios).

Finally, we presented an asymptotically optimal synchronisation algorithm for distributed adaptive transmit beamforming in section 7.3.3. For this method, a one-bit feedback from the receive node is not sufficient so that a richer feedback is required. The general approach is to measure the feedback-value of several phase-offsets for each individual node. With these measurements it is possible to estimate the optimum phase-offset directly. The expected optimisation-time of this method was calculated as $\mathcal{O}(n)$ Although several phase-offsets have to be calculated repeatedly since the simultaneous calculation of an optimum phase-offset by distinct nodes leads to a bias in the calculation we achieved an average synchronisation-time of about $12n$ in mathematical simulations for this method.

For all these approaches, a hierarchical clustering can be applied in order to optimise the synchronisation-performance further. We propose this mechanism and derived the time required to find an optimum cluster hierarchy and cluster size as $\mathcal{O}(n^2)$ in section 7.3.1. This method took advantage of the observation that the synchronisation-performance rises more than linearly with the network size while the transmission power decreases only linearly with the network size. It is therefore beneficial to synchronise smaller clusters of nodes that transmit with an increased transmission power for a shorter period of time.

After carrier-signal-components are sufficiently synchronised, data is transmitted between the network and a receiver node. In section 7.5 we propose a protocol for transmission via distributed adaptive beamforming that incorporates the synchronisation of carrier-signals and collaborative transmission of nodes. For this protocol the achieved bit-error-rate (BER) for distinct environmental settings and modulation schemes was examined. Remarkably, the BER increases with increasing transmission-distance and data-rate. Since the synchronisation-performance is directly impacted by the count of participating devices, we conclude that it is beneficial when this number is just large enough to reach a remote receiver collaboratively. Greater numbers of transmitters do not lead to an improved BER but make the synchronisation-performance worse.

Since different algorithmic parameters lead to an optimum synchronisation-performance for distinct environments, we proposed an adaptive protocol in section 7.6.2 that finds optimum parameters for a given environment.

In section 7.7 we discussed the application of learning classifier systems to support environments with frequently changing settings.

To summarise, a concise investigation of distributed adaptive transmit beamforming with feedback-based carrier-synchronisation was presented. Starting from a basic analytic discussion on the algorithmic and environmental structure, asymptotic bounds were calculated for several of the proposed synchronisation algorithms. With the modifications of these algorithms and with studies on the impact of environmental and algorithmic parameter settings in simulations and experimental instrumentation, a comprehensive understanding of the optimisation scenario and environmental impacts was gained. To react to these impacts optimally in mobile and changing environments, we proposed adaptive protocols and derived the performance of the protocols in quantitative mathematical simulations.

Index

List of Tables

List of Figures

Bibliography

[1] Culler, D., Estrin, D., Srivastava, M.: Overview of sensor networks. IEEE Computer **37**(8) (2004) 41–49

[2] Zhao, F., Guibas, L.: Wireless Sensor Networks: An Information Processing Approach. Morgan Kaufmann, Los Altos, CA (2004)

[3] Norman, D.: The invisible computer. MIT press (1999)

[4] Butera, W.J.: Programming a paintable computer. PhD thesis, Massachusetts Institute of Technology (2002)

[5] Pillutla, L., Krishnamurthy, V.: Joint rate and cluster optimisation in cooperative mimo sensor networks. In: Proceedings of the 6th IEEE Workshop on signal Processing Advances in Wireless Communications. (2005) 265–269

[6] Scaglione, A., Hong, Y.W.: Opportunistic large arrays: Cooperative transmission in wireless multihop ad hoc networks to reach far distances. IEEE Transactions on Signal Processing **51**(8) (2003) 2082–2092

[7] Sendonaris, A., Erkop, E., Aazhang, B.: Increasing uplink capacity via user cooperation diversity. In: IEEE Proceedins of the International Symposium on Information Theory (ISIT). (2001) 156

[8] Laneman, J., Wornell, G., Tse, D.: An efficient protocol for realising cooperative diversity in wireless networks. In: Proceedings of the IEEE International Symposium on Information Theory. (2001) 294

[9] Hong, Y.W., Scaglione, A.: Critical power for connectivity with cooperative transmission in wireless ad hoc sensor networks. In: IEEE Workshop on Statistical Signal Processing. (2003)

[10] Hong, Y.W., Scaglione, A.: Energy-efficient broadcasting with cooperative transmission in wireless sensor networks. IEEE Transactions on Wireless communications (2005)

[11] Jayaweera, S.K.: Energy analysis of mimo techniques in wireless sensor networks. In: 38th conference on information sciences and systems. (2004)

[12] del Coso, A., Sagnolini, U., Ibars, C.: Cooperative distributed mimo channels in wireless sensor networks. IEEE Journal on Selected Areas in Communications **25**(2) (2007) 402–414

[13] Sigg, S., Beigl, M.: Collaborative transmission in wsns by a (1+1)-ea. In: Proceedings of the 8th International Workshop on Applications and Services in Wireless Networks (ASWN'08). (2008)

[14] Sigg, S., Beigl, M.: Randomised collaborative transmission of smart objects. In: 2nd International Workshop on Design and Integration principles for smart objects (DIPSO2008) in conjunction with Ubicomp 2008. (2008)

[15] Mudumbai, R., Brown, D.R., Madhow, U., Poor, H.V.: Distributed transmit beamforming: Challenges and recent progress. IEEE Communications Magazine (2009) 102–110

[16] Mudumbai, R., Wild, B., Madhow, U., Ramchandran, K.: Distributed beamforming using 1 bit feedback: from concept to realization. In: Proceedings of the 44th Allerton conference on communication, control and computation. (2006) 1020–1027

[17] Barriac, G., Mudumbai, R., Madhow, U.: Distributed beamforming for information transfer in sensor networks. In: Proceedings of the third International Workshop on Information Processing in Sensor Networks. (2004)

[18] Mudumbai, R., Barriac, G., Madhow, U.: On the feasibility of distributed beamforming in wireless networks. IEEE Transactions on Wireless communications **6** (2007) 1754–1763

[19] Ochiai, H., Mitran, P., Poor, H.V., Tarokh, V.: Collaborative beamforming for distributed wireless ad hoc sensor networks. IEEE Transactions on Signal Processing **53**(11) (2005) 4110 – 4124

[20] Chen, W., Yuan, Y., Xu, C., Liu, K., Yang, Z.: Virtual mimo protocol based on clustering for wireless sensor networks. In: Proceedings of the 10th IEEE Symposium on Computers and Commmunications. (2005)

[21] Youssef, M., Yousif, A., El-Sheimy, N., Noureldin, A.: A novel earthquake warning system based on virtual mimo wireless sensor netwroks. In: Canadian conference on electrical and computer engineering. (2007) 932–935

[22] del Coso, A., Savazzi, S., Spagnolini, U., Ibars, C.: Virtual mimo channels in cooperative multi-hop wireless sensor networks. In: 40th annual conference on information sciences and systems. (2006) 75–80

[23] Jayaweera, S.K.: Energy efficient virtual mimo based cooperative communications for wireless sensor networks. IEEE Transactions on Wireless communications **5**(5) (2006) 984–989

[24] Laneman, J., Wornell, G.: Distributed space-time coded protocols for exploiting cooperative diversity in wireless networks. IEEE Transactions on Information theory **49**(10) (2003) 2415–2425

[25] Sendonaris, A., Erkip, E., Aazhang, B.: User cooperation diversity – part i: System description. IEEE Transactions on Communications **51**(11) (2003) 1927–1938

[26] Zimmermann, E., Herhold, P., Fettweis, G.: On the performance of cooperative relaying protocols in wireless networks. European Transactions on Telecommunications **16**(1) (2005) 5–16

[27] Cover, T.M., Gamal, A.A.E.: Capacity theorems for the relay channel. IEEE Transactions on Information Theory **525**(5) (1979) 572–584

[28] Kramer, G., Gastpar, M., Gupta, P.: Cooperative strategies and capacity theorems for relay networks. IEEE Transactions on Information Theory **51**(9) (2005) 3037–3063

[29] Scaglione, A., Hong, Y.W.: Cooperative models for synchronization, scheduling and transmission in large scale sensor networks: An overview. In: 1st IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing. (2005) 60–63

[30] Gupta, P., Kumar, R.P.: The capacity of wireless networks. IEEE Transactions on Information Theory **46**(2) (2000) 388–404

[31] Mitran, P., Ochiai, H., Tarokh, V.: Space-time diversity enhancements using collaborative communications. IEEE Transactions on Information Theory **51**(6) (2005) 2041–2057

[32] Simeone, O., Spagnolini, U.: Capacity region of wireless ad hoc networks using opportunistic collaborative communications. In: Proceedings of the International Conference on Communications (ICC). (2006)

[33] Krohn, A., Beigl, M., Decker, C., Varona, D.G.: Increasing connectivity in wireless sensor network using cooperative transmission. In: 3rd International Conference on Networked Sensing Systems (INSS). (2006)

[34] Krohn, A.: Optimal non-coherent m-ary energy shift keying for cooperative transmission in sensor networks. In: 31st IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). (2006)

[35] Hong, Y.W., Scaglione, A.: Cooperative transmission in wireless multi-hop ad hoc networks using opportunistic large arrays (ola). In: SPAWC. (2003)

[36] Brown, D.R., Prince, G., McNeill, J.: A method for carrier frequency and phase synchronization of two autonomous cooperative transmitters. In: sixth IEEE workshop on signal processing advances in wireless communications. (2005)

[37] Brown, D.R., Poor, H.V.: Time-slotted round-trip carrier synchronisation for distributed beamforming. IEEE Transactions on Signal Processing **56** (2008) 5630–5643

[38] Ozil, I., Brown, D.R.: Time-slotted round-trip carrier synchronisation. In: Proceedings of the 41st Asilomar conference on signals, signals and computers. (2007) 1781–1785

[39] Tu, Y., Pottie, G.: Coherent cooperative transmission from multiple adjacent antennas to a distant stationary antenna through awgn channels. In: Proceedings of the IEEE Vehicular Technology Conference. (2002) 130–134

[40] Mudumbai, R., Hespanha, J., Madhow, U., Barriac, G.: Scalable feedback control for distributed beamforming in sensor networks. In: Proceedings of the IEEE International Symposium on Information Theory. (2005) 137–141

[41] Mudumbai, R., Hespanha, J., Madhow, U., Barriac, G.: Distributed transmit beamforming using feedback control. IEEE Transactions on Information Theory ((In review))

[42] Seo, M., Rodwell, M., Madhow, U.: A feedback-based distributed phased array technique and its application to 60-ghz wireless sensor network. In: IEEE MTT-S International Microwave Symposium Digest. (2008) 683–686

[43] Bucklew, J.A., Sethares, W.A.: Convergence of a class of decentralised beamforming algorithms. IEEE Transactions on Signal Processing **56**(6) (2008) 2280–2288

[44] Gellersen, H.W., Beigl, M., Krull, H.: The mediacup: Awareness technology embedded in an everyday object. In Gellersen, H.W., ed.: 1th International Symposium on Handheld and Ubiquitous Computing (HUC99). Volume 1707 of Lecture notes in computer science., Springer (1999) 308–310

[45] Kidd, C.D., Orr, R., Abowd, G.D., Atkeson, C.G., Essa, I.A., MacIntyre, B., Mynatt, E., Starner, T.E., Newstetter, W.: The aware home: A living laboratory for ubiquitous computing research. In: Lecture in Computer Science; Proceedings of the Cooperative Buildings Integrating Information, Organisation and Architecture (CoBuild'99). Volume 1670. (1999)

[46] ITU: ITU Internet Reports 2005: The Internet of Things. 7th edn. International telecommunication union (2005)

[47] Li, Y., Thai, M., Wu, W.: Wireless sensor networks and applications. Signals and Communication Technology. Springer (2008)

[48] Karl, H., Willig, A.: Protocols and Architectures for Wireless Sensor Networks. Wiley (2005)

[49] Rappaport, T.: Wireless Communications: Principles and Practice. Prentice Hall (2002)

[50] Sohrabi, K., Manriquez, B., Pottie, G.: Near-ground wideband channel measurements. In: Proceedings of the 49thvehicular technology conference. (1999) 571–574

[51] Kahn, J.M., Katz, R.H., Pister, K.S.J.: Next century challenges: Mobile networking for smart dust. In: Proceedings of the ACM MobiCom. (1999) 271–278

[52] Hoblos, G., Staroswiecki, M., Aitouche, A.: Optimal design of fault tolerant sensor networks. In: Proceedings of the IEEE International Conference on Control Applications. (2000) 467–472

[53] Bulusu, N., Estrin, D., Girod, L., Heidemann, J.: Scalable coordination for wireless sensor netwroks: Self-configuring localisation systems. In: Proceedings of the 6th IEEE International Symposium on Communication Theory and Application. (2001)

[54] Shen, C.C., Srisathapornphat, C., Jaikaeo, C.: Sensor information networking architecture and applications. IEEE Personal Communications **8**(4) (2001) 52–59

[55] Cerpa, A., Elson, J., Estrin, D., Girod, L., Hamilton, M., Zhao, J.: Habitat monitoring: Application driver for wireless communications technology. In: Proceedings of the ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean. (2001)

[56] Raghavendra, C.S., Singh, S.: Pamas - power aware multi-access protocol with signalling for ad hoc networks. ACM Computer Communication Review **27** (1998) 5–26

[57] Ye, W., Heidemann, J., Estrin, D.: Medium access control with coordinated, adaptive sleeping for wireless sensor networks. IEEE/ACM Transactions on Networking (2004)

[58] of IEEE 802.11, T.E.: Ieee standard for wireless lan medium access control (mac) and physical layer (phy) specifications (1997)

[59] Schurgers, C., Tsiatsis, V., Ganeriwal, S., Srivastava, M.: Optimising sensor networks in the energy-latency-density design space. IEEE Transactions on Mobile Computing **1**(1) (2002) 70–80

[60] Ye, W., Heidemann, J., Estrin, D.: An energy efficient mac protocol for wireless sensor networks. In: Proceedings of the INFOCOM 2002, IEEE Press (2002)

[61] v. Dam, T., Langedoen, K.: An adaptive energy-efficient mac protocol for wireless sensor networks. In: Proceedings of the first international conference on embedded networked sensor systems (SenSys). (2003)

[62] Jerome, J.K.: Three men in a boat. Collector's Library (2005)

[63] Seybold, J.S.: Introduction to RF propagation. Wiley (2005)

[64] 3GPP: 3rd generation partnership project; technical specification group radio access networks; 3g home nodeb study item technical report (release 8). Technical Report 3GPP TR 25.820 V8.0.0 (2008-03) (March)

[65] Ohm, J.R., Lüke, H.D.: Signalübertragung – Grundlagen der digitalen und analogen Nachrichtenübertragungssysteme. Volume 10. Springer (2007)

[66] Feller, W.: An Introduction to Probability Theory and its Applications. Wiley (1968)

[67] Duda, R., Hart, P., Stork, D.: Pattern Classification. 2nd edn. Wiley Interscience (2001)

[68] Schwefel, H.P.: Direct search for optimal parameters within simulation models. Proceedings of the twelfth annual simulation symposium (1979) 91–102

[69] Holland, J.: Genetic algorithms and the optimal allocation of trials. SIAM, Journal of computing **3**(4) (1974) 88–105

[70] Schwefel, H.P.: Evolution and optimum seeking. Wiley-Interscience (1995)

[71] Fogel, L.J., Fogel, D.B.: A preliminary investigation on extending evolutionary programming to include self-adaptation on finite state. Informatica **18**(4) (1994)

[72] Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press (1992)

[73] Bäck, T.: An overview of parameter control methods by self-adaptation in evolutionary algorithms. Fundamenta Informaticae **35** (1998) 51–66

[74] Droste, S.: Efficient genetic programming for finding good generalizing boolean functions. In Koza, J.R., Deb, K., Dorigo, M., Fogel, D.B., Garzon, M., Iba, H., Riolo, R.L., eds.: Proceedings of the second Genetic Programming conference (GP 97), Morgan Kaufmann (1997) 82–87

[75] Droste, S., Heutelbeck, D., Wegener, I.: Distributed hybrid genetic programming for learning boolean functions. In Schoenauer, M., ed.: Proceedings of the 6th Parallel Problem Solving from Nature (PPSN VI). Volume 1917 of Lecture Notes in Computer Science (LNCS)., Springer (2000) 181–190

[76] Jansen, T., Wegener, I.: Evolutionary algorithms – how to cope with plateaus of constant fitness and when to reject strings of the same fitness. IEEE Transactions on Evolutionary Computation **5**(6) (2001) 589–599

[77] Wegener, I., Witt, C.: On the optimization of monotone polynomials by the (1+1)ea and randomized local search. In: Genetic and Evolutionary Computation Conference (GECCO). Number 2723 in Lecture notes in computer sciences (LNCS) (2005) 622–633

[78] Wegener, I., Witt, C.: On the analysis of a simple evolutionary algorithm on quadratic pseudo-boolean functions. Journal of Discrete Algorithms (3) (2005) 61–78

[79] Droste, S., Jansen, T., Wegener, I.: A rigorous complexity analysis of the (1+1) evolutionary algorithm for linear functions with boolean inputs. In: Proceedings of the third IEEE International conference on Evolutionary computation (ICEC 98), Piscataway, NJ, IEEE press (1998) 499–504

[80] Droste, S., Jansen, T., Wegener, I.: A rigorous complexity analysis of the (1+1) evolutionary algorithm for separable functions with boolean inputs. Evolutionary Computation **6**(2) (1998) 185–196

[81] Droste, S., Jansen, T., Wegener, I.: On the analysis of the (1+1) evolutionary algorithm. Theoretical Computer Science **276**(1-2) (2002) 51 – 81

[82] Sirkeci-Mergen, B., Scaglione, A.: 10. In: Randomized cooperative transmission in large scale sensor networks. A. Swami, Q. Zhao, Y. Hong, and L. Tong, Wiley (2007)

[83] Laneman, J., Wornell, G.: Energy-efficient antenna sharing and relaying for wireless networks. In: Proceedings of the IEEE Wireless Communication Networking Conference. (2000) 294

[84] Lanemann, J., Tse, D., Wornell, G.: Cooperative diversity in wireless networks: Efficient protocols and outage behaviour. Transactions on inforamtion Theory **50**(12) (2004)

[85] Nabar, R., Bölscskei, H., Wornell, G.: Fading relay channels: Performance limits and space-time signal design. IEEE Journal on Selected Areas in Communications **22**(6) (2004) 1099–1109

[86] Gastpar, M., Vetterli, M.: On the capacity of large gaussian relay networks. IEEE Transactions on Information Theory **51**(3) (2005) 765–779

[87] Ahlswede, R., Cai, N., Li, S.Y., Yeung, R.: Network information flow. IEEE Transactions on Information Theory **46**(4) (2000) 1204–1216

[88] Li, S.Y., Son, S., Stankovic, J.: Linear network coding. IEEE Transactions on Information Theory **49**(2) (2003) 371–381

[89] Woldegebreal, D.H., Karl, H. In: Network-Coding-Based Cooperative Transmission in Wireless Sensor Networks: Diversity-Multiplexing Tradeoff and Coverage Area Extension. Volume 4913 of Lecture Notes in Computer Science (LNCS). Springer-Verlag (2008) 141–155

[90] Scaglione, A., Hong, Y.W.: Opportunistic large arrays. In: IEEE International Symposium on Advances in Wireless Communications. (2002)

[91] Salhotra, A., Scaglione, A.: Multiple access in connectionless networks using cooperative transmission. In: Allerton Conference. (2003)

[92] Paulraj, A., Nabar, R., Gore, D.: Introduction to Space-Time Wireless communications. Cambridge University Press (2003)

[93] Shuguang, C., Goldsmith, A.: Energy-efficiency of mimo and cooperative mimo techniques in sensor networks. IEEE Journal on Selected Areas in Communications **22**(6) (2004) 1089–1098

[94] Alamouti, S.M.: A simple transmit diversity technique for wireless communications. IEEE Journal on select areas in communications **16**(8) (1998)

[95] Gastpar, M., Vetterli, M.: On the capacity of wireless networks: the relay case. In: Proceedings of the IEEE Infocom. (2002) 1577–1586

[96] Hagmann, W.: Network synchronisation techniques for satellite communication systems. PhD thesis, USC, Los Angeles (1981)

[97] Best, R.: Phase-Locked Loops: Design, Simulation and Applications. McGraw-Hill (2003)

[98] Krohn, A.: Superimposed Radio Signals for Wireless Sensor Networks. PhD thesis, Technical University of Braunschweig (2007)

[99] Bennett, W.: Introduction to signal transmission. McGraw-Hill (1971)

[100] Sigg, S., Masri, R.M.E., Beigl, M.: A sharp asymptotic bound for feedback based closed-loop distributed adaptive beamforming in wireless sensor networks. Transactions on mobile computing (2010 (submitted))

[101] Sigg, S., Beigl, M.: Algorithms for closed-loop feedback based distributed adaptive beamforming in wireless sensor networks. In: Proceedings of the fifth International Conference on Intelligent Sensors, Sensor Networks and Information Processing - Symposium on Adaptive Sensing, Control, and Optimization in Sensor Networks. (2009)

[102] Sigg, S., Masri, R., Ristau, J., Beigl, M.: Limitations, performance and instrumentation of closed-loop feedback based distributed adaptive transmit beamforming in wsns. In: Fifth International Conference on Intelligent Sensors, Sensor Networks and Information Processing - Symposium on Theoretical and Practical Aspects of Large-scale Wireless Sensor Networks. (2009)

[103] Sigg, S., Beigl, M.: Algorithmic approaches to distributed adaptive transmit beamforming. In: Fifth International Conference on Intelligent Sensors, Sensor Networks and Information Processing - Symposium on Theoretical and Practical Aspects of Large-scale Wireless Sensor Networks. (2009)

[104] Urbanowicz, R.J., Moore, J.H.: Learning classifier systems: A complete introduction, review, and roadmap. Journal of Artificial Evolution and Applications **2009** (2009)

[105] Feynman, R.P.: QED, The strange theory of light and matter. Princeton University Press (1985)