NTNU

Innovation and Creativity

WISA vs. WLAN: Co-existence challenges

Analysis of frequency-hopping sequences

Erik Skarstein Sandnes

Master of Science in Communication Technology

Submission date: June 2007

Supervisor: Geir Egil Øien, IET

Co-supervisor: Anne Elisabeth Vallestad, ABB Corporate Research

Norwegian University of Science and Technology

Department of Electronics and Telecommunications

Problem Description

WISA is ABB's proprietary 1Mbit/s wireless protocol for industrial automation. It operates within the 2.4GHz ISM band. WISA targets manufacturing \& robotics applications (i.e. not process automation).

WLANs - which occupy a fixed portion of the 2.4GHz band - are becoming more and more common on the factory floor, e.g. bar-code scanners. Customers need confirmation that WISA can co-exist with other systems - particularly WLAN - without degrading their performance 'too much'. Preferably they would like WISA to have some form of adaptive frequency-hopping. That is, detect any other equipment that occupies part(s) of the 2.4GHz band - and stay away from those frequencies.

WISA today has frequency hopping which spreads traffic uniformly over 77MHz of the 83MHz wide band. However it cannot - adaptively or otherwise - avoid any part of the band.

The problem of implementing adaptive frequency hopping in WISA is at least threefold: a) How to detect which frequencies to avoid; b) Re-map the FH sequences to a smaller part of the band, while keeping the nice properties of the original FH sequences; and c) How much degradation is 'too much' – and how does it depend on the WLAN and WISA traffic parameters?

This thesis will focus on subproblem b). The objective of the diploma project is to create a Matlab based simulation tool that can (i) analyze cross-correlation between FH sequences in two closely spaced WISA cells, and (ii) generate new FH sequences which avoid a user-selectable portion of the frequency band.

The module is handed over to ABB when this thesis is finished and must therefore be user friendly and well documented. The module will also be used in this thesis to analyze the existing FH sequences and any new possible sequences that can be derived during this work.

Assignment given: 15. January 2007
Supervisor: Geir Egil Øien, IET

Abstract

Wireless Interface for Sensors and Actuators (WISA) is ABB's proprietary wireless protocol for industrial automation. It operates in the 2.4 GHz ISM band, as do nearly all Wireless LAN systems. WISA does frequency-hopping (FH) over most of the ISM band, but has currently no means of avoiding parts of the band occupied by other wireless systems.

The objective of the diploma project was to create a Matlab based simulation tool that can (i) analyze cross-correlation between FH sequences in two closely spaced WISA cells, and (ii) generate new FH sequences which avoid a user-selectable portion of the frequency band.

New frequency-hopping sequences were designed using Galois field computations for creating periodic sequences with minimum correlation.

The developed Matlab simulation module did indeed meet the objectives. However the algorithm for subband-allocation is not optimal and will for some cases not give maximum utilization of the available frequency band.

Analysis of the existing FH algorithm confirmed that some sequence pairs are non-ideal in the sense that their inevitable frequency collisions are not spread evenly over all relative shifts between the sequences, but concentrated to a few of these shifts. It was also pointed out that not all cell ids met the desired requirement of large separation between transmissions occurring on consecutive frames. Analysis of the new FH sequences, which avoid a user-selected portion of the frequency band, showed that these had many of the same properties as the existing algorithm. It was possible to find sequence pairs with low correlation and thus allow multiple cells to operate in the same radio space.

Preface

This thesis was completed as a part of the 5-year Master's Degree Programme and is the final examination for the Master's Degree in Communications Technology at Norwegian University of Science and Technology (NTNU), Trondheim, Norway. The thesis was completed in conjunction with ABB Corporate Research Center, Billingstad, Norway.

I would like to thank Dr. Dagfin Brodtkorb, Research Director of ABB Corporate Research Center, Billingstad, Norway, for giving me the opportunity to work with this project. I would also like to thank my supervisors MSc Anne Elisabeth Vallestad at ABB Corporate Research Center, Norway, and Dr. Dacfey Dzung at ABB Corporate Research Center, Switzerland, for guiding me through the process and helping me understand the WISA system. A special thanks goes to Anne for her dedication and commitment as my supervisor. Her helpful comments and follow-ups has been vital in the process of completing this thesis. Finally I would like to thank my local supervisor Prof. Geir Øien for accepting the thesis proposal from ABB and thus providing me with the opportunity to work with a project outside the institute.

Erik Skarstein Sandnes, June 2007

Contents

Abbreviations

BS	Base station
CCF	Cross-correlation function
DL	Downlink
Dslot	Double-slot
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
FH	Frequency-hopping
FN	Frame number
GF	Galois field
GUI	Graphical User Interface
IF	Intermediate frequency
ISM	Industrial, Scientific and Medical
Rx	Receiver
SA	Sensor/actuator
SAW	Surface Acoustic Wave
Sslot	Single-slot
TDMA	Time Division Multiple Access
Tx	Transmitter
UL	Uplink
WISA	Wireless Interface for Sensors and Actuators
WLAN	Wireless Local Area Network

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Wireless Interface for Sensors and Actuators (WISA) is ABB's proprietary wireless protocol for industrial automation. It operates in the 2.4 GHz ISM band, as do nearly all Wireless LAN systems. WLANs are becoming more and more common on the factory floor. The chance of co-located WISA and WLAN installations is therefore growing.

WISA does frequency-hopping (FH) over most of the ISM band, but has currently no means of avoiding parts of the band occupied by other wireless systems.

A WISA communications cell consists of a base station and a number of devices. Each cell is given a unique ID which determines its fixed-length FH sequence. There is no synchronism between cells. Interference between cells is undesirable and not easy to analyze. Generating new FH sequences which avoid certain portions of the band is not straightforward.

ABB will greatly benefit from a tool that can analyze cross-correlation between sequences as well as generate new sequences.

1.2 Objective

The objective of the diploma project was to create a Matlab based simulation tool that can:

a) **Analyze** existing (and new) WISA frequency-hopping sequences according to a given set of metrics; the most important being self-interference, i.e. cross-correlation between the FH sequences of two closely spaced WISA cells.

b) **Generate** new FH sequences which avoid a user-selectable portion of the frequency band; while keeping cross-correlation low, and also complying with a number of extra constraints.

Both a) and b) require an understanding of the theoretical properties of the FH sequences used. See Chapter 3.

a) also demands an understanding of what happens to the relative timeshift between the FH sequences of two cells which are not syncronised, and how this can be translated to a tractable 'Matlab-style' problem.

1.3 Scope

Task a) - Analysis of FH sequences The scope of a) is limited to the case where two WISA cells interfer with each other. Furthermore, interference is defined as collision in frequency and time, i.e. simultaneous operation on the same frequency.[1] Distances between interfering devices/base stations, i.e. varying signal-to-interference ratio, has not been considered.

A WISA cell uses five different frequencies at any given time - one 'downlink' channel from the base station to the devices, and 4 'uplink' channels in the other direction. This has been included in the analysis. The downlink is always transmitting, whereas uplink transmissions occur only a small portion of the time, as a result of sensor events. In the analysis, the probability of an uplink frequency actually being used has been set according to a typical scenario given in WISA documentation.

[1] Interference between neighbour frequencies - 'adjacent channels' - was not included in this project due to lack of time. In practice, such interference indeed occurs in any radio system; channel definition is never ideal, i.e. the spectrum of a transmitter will always spread into adjacent channels, although it will be attenuated.

Various metrics for the properties of the FH sequences are included in the new tool. Graphical and textual presentation of these metrics is an important part of the work. User input has been included in the tool.

Task b) - Generating new FH sequences Existing and new algorithms for generation of FH sequences are described, together with the desired properties of the sequences. In the end, two algorithms have been implemented in the tool:

- **FH02** - The existing frequency-hopping algorithm implemented in WISA.
- **FH07-1** - New frequency-hopping algorithm avoiding user-selected WLAN frequencies. This algorithm is an extension of the existing frequency-hopping algorithm, FH02.

In addition to the cross-correlation properties, a number of 'extra constraints' have been described and taken into consideration. These are:

- The base stations FH sequence should be uniquely selected by the cell id.
- All 4 uplinks must be located in the same contiguous subband.
- The uplinks must be spaced 2 MHz apart to avoid adjacent channel interference.
- Subbands are limited to a width of minimum 7 MHz and maximum 17 MHz.
- Downlink subband and uplink subbands must have a duplex spacing of 22 MHz.

All these constraint will be explained in Chapter 2.

User input is enabled where necessary.

The sequences generated using the tool are presented graphically and can be stored in a file for later use.

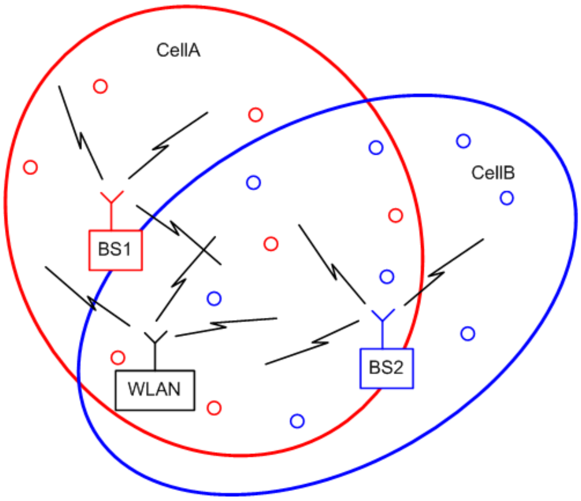


Figure 1.1: Illustration of the simulation environment. Two WISA cells and a WLAN operating within the same radio space.

1.4 Simulation environment

Fig. 1.1 shows a setup where two WISA base stations - BS1 and BS2 - and their allocated devices are operating within the same radio space. Each base station and its allocated devices are defined as a cell, CellA (BS1) and CellB (BS2). These cells will will interfer with each other whenever they use the same frequency. The frequency-hopping sequences in the cells are selected by the base station's cell ids. Base stations are always transmitting[2], i.e. downlink is always on. The uplinks are used less frequently, and are therefore not as likely to introduce interference in the other cell. The WLAN will occupy a fixed portion of the available ISM band and should not be interfered by the WISA system.

1.5 Thesis outline

Chapter 2 gives a more thorough explanation of the problem studied in this thesis. The constraints that follow from the existing hardware will be explained in regard to how they affect new FH sequences when it is necessary to avoid part of the frequency band.

Chapter 3 introduces ABB's WISA system. It will also explain the frequency-hopping algorithm used in WISA and the co-existing

[2] Except for the one Dslot used for switching frequencies, see Section 3.1

WLAN's fixed frequency occupation. Finally it will look at how the FH sequences used in WISA are derived using Galois field computations.

Chapter 4 will present the Matlab simulation module and explain the use of this tool.

Chapter 5 analyzes the existing FH sequences and the interference between two base stations.

Chapter 6 introduces new FH sequences and analyzes the performance of these sequences.

Chapter 7 concludes on the analysis presented in this thesis and presents recommendations for further work.

Chapter 2

Problem description

The objective of this thesis was to create a tool in Matlab for simulation of interference (correlation) between two WISA cells. This tool was then to be used for analysis of the existing frequency-hopping (FH) sequences used in the WISA system and any possible new frequency-hopping algorithms designed in this project. The following sections will look individually at these parts.

2.1 Simulation module

There are two elements in the simulation module: simulation and presentation. The module will simulate the interference between two base stations operating in parallel. The two base stations have local oscillators controlling timing and these are not synchronized. In time, an offset between the two base stations will be introduced and the frame number[1], FN, will no longer be the same. For simplicity, the offset can be introduced to one of the base stations frame number counters and thus incremented as complete frame offsets. The simulation module will calculate the cross-correlation between the two base stations FH sequences for all possible frame offsets. A crucial issue of the WISA system is to ensure a reliable transmission with minimum delay to make sure information from the sensor/actuators reaches its destination in time. So a worst-case scenario must also be included in the module. The worst-case scenario is the maximum number of consecutive collisions that

[1] Explained in Section 3.1.2

can occur for any offset between the two base stations.

The simulation module will have to present results in well arranged manner so that ABB can use this tool in their work. The tool is to be used for finding satisfying combinations of cell ids, which select the FH sequences[2], for base stations operating in the same cluster. The cross-correlation can be shown as (i) an average cross-correlation between a selected FH sequence in base station 1 vs. all possible FH sequences, equivalent to all possible cell ids, in base station 2 or (ii) the cross-correlation between two selected FH sequences vs. the frame offset between the two base stations.

2.2 Analysis of the existing FH sequences

ABB have through thousands of well monitored hours of WISA operation observed that some combinations of FH sequences, uniquely defined by certain cell ids, can co-exist with lower interference than others. Particularly base stations with equal first or second digit cell ids seem to have periods of high interference between them. This thesis will study this observation. It will also study the frequency separation between consecutive frames.

2.3 New algorithm for generation of FH sequences avoiding selectable portions of the ISM band

If a WLAN is operating in the same environment as a WISA system, the customers want to be sure that WISA introduces no interference to the WLAN. This means that WISA will have to avoid the portion of the band occupied by the WLAN. A new frequency-hopping algorithm avoiding the frquencies used by WLAN must be derived. As specified in Section 1.2, this project is based on avoiding user-selected frequencies (WLAN channels) and will not focus on how a WLAN is detected.

Defining allowed WISA frequencies The ISM band ranges from 2400 MHz to 2483,5 MHz and WISA uses 79 channels of 1 MHz bandwidth with center frequencies from 2402 MHz to 2480 MHz. The existing FH algorithm uses 77 of these channels ranging from 2403 MHz to 2479 MHz. The two remaining channels are used for control. When a new system which uses a

[2] Explained in Section 3.1.2

fixed portion of the available band, like WLAN, is introduced to the environment the set of allowed frequencies must be re-mapped to a subset of the original 77 frequencies. From this subset the system has to utilize as much as possible of the remaining bandwidth.

Derivation of new FH sequences There are several constraints connected to the derivation of new FH sequences. ABB wish to keep the good properties of the existing configuration. The cell id should select a unique FH sequence. Re-transmission occurring in consecutive frames should be on different frequencies with wide separation. The 4 uplink groups must be in the same contiguous subband. Uplinks must be spaced 2 MHz apart due to co-channel interference.[3] Since up to 4 uplinks must be supported the subband is limited to a minimum of 7 MHz. It is also limited to a maximum of 17 MHz due to RX IF bandwidth in the receiver. The filter used in the receiver is a SAW filter designed for processing WLAN channels of 18-22 MHz. This means that the uplink and downlink subbands must have a duplex spacing of minimum 22 MHz. [2]

Analysis The analysis of the new FH sequences will be similar to the analysis of the existing sequences. The new cross-correlation will be studied together with worst-case scenarios.

[3] At 0,5 MHz from the center frequency the signal is only attenuated 30 dB at the receiver and a signal from a sensor/actuator close to the base station could drown the signal from a device spaced further apart.

Chapter 3

Theory

3.1 WISA

In this section a brief summary of the motivation and history behind WISA will be presented as well as the basic principles of the system. The theory presented in this section is an abstract of [3] and [4].

In a factory environment cables to sensors and actuators are one of the most frequent sources of failure, especially in harsh environment conditions and where nodes[1] are installed on moving parts in a factory environment. When the source of failure lies in a cable the costs associated with trouble-shooting, materials and repairing are high. Wireless technology reduces overall costs thanks to its easy installation and simpler engineering. When ABB started their planning of a wireless interface to sensors and actuators in a factory environment they found that none of the existing wireless solutions met the necessary requirements in latency and data rates. Bluetooth and WLAN had higher latency than acceptable and ZigBee did not provide sufficient data rates. This led to the concept of WISA - *W*ireless *I*nterface to *S*ensors and *A*ctuators - in 1998 and the first prototype in 2001. The first products went on the market in 2004 followed by new and improved additions. Fig. 3.1 compares the latency vs data rates for WISA and other standards.

The WISA system consists of two central modules. One is WISA-COM which defines the wireless communication between nodes and base station.

[1] Sensors/actuators installed on different applications in a factory environment, e.g. transport band and robotic arms.

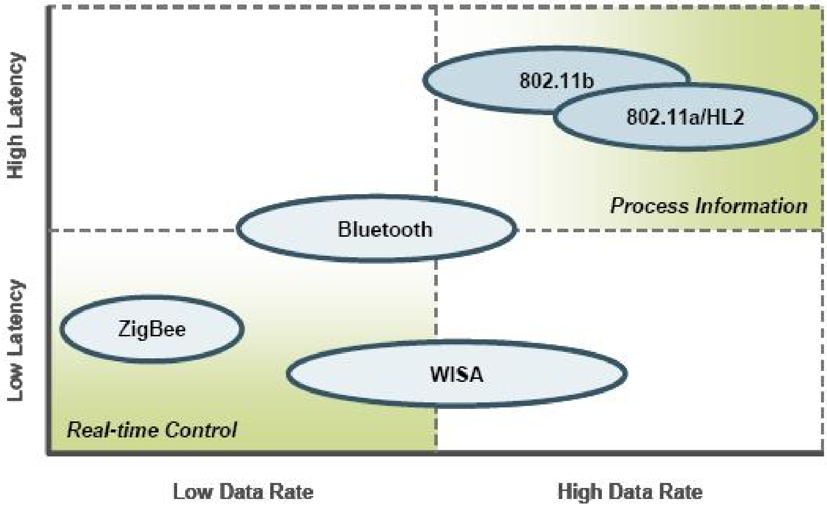


Figure 3.1: Latency vs. data rate of different communication systems at higher node densities.

The other is WISA-POWER which ensures wireless transmission of power to the nodes.

WISA-COM WISA-COM covers 77+2 channels in the 2.4 GHz ISM band. Each channel is 1 MHz wide, as given by the 802.15.1 PHY. Transmit power is limited to 0 dBm, and the practical range with today's technology is 5-10 m.

WISA-COM links a WISA base station with its devices, i.e. Proximity Switches, I/O Pads, or Sensor Pads. A base station and its devices constitute a WISA cell.

4 uplink frequencies and 1 downlink frequency are used concurrently at any given instant. (Uplink = from device to base station; downlink = from base station to device.) Frequency-hopping is done every 2ms frame. The FH sequences are deterministic and 77 frames long. There are 60 unique sequences with low cross-correlation. The idea is that nearby cells are given different FH sequences such that they can coexist with limited degradation.

TDMA + FDD: There is room for 120 proximity switches or 60 pads per base station cell, each being allocated a dedicated combination of uplink frequency and sub-frame time slot.

Unlike most comparable wireless communication standards, the WISA base station transmits continuously on the downlink, while listening continously on the four uplinks; it pauses only for 1/16 frame while it changes frequencies. The remaining 15/16 of the frame is used for communication.

In comparison, the devices transmit sporadically. The devices have one radio and cannot transmit and receive at the same time.

The reason why the base station transmits continuously is partly power-saving in the devices and partly latency requirements: when a device wakes up due to an event, it should not need to search for long to 'see' the base station and get its transmission overwith.

WISA-POWER In order to provide a real benefit to the user, also a "wireless power supply" has to be provided for the critical field devices. Batteries/energy storage is normally not an option in industrial factory automation with its tens of thousands of nodes and often 24h/day operation; even today's WISA power consumption is not low enough.

Alternative ways of energy harvesting from the environment of the application are not reliable, when considering the widely varying applications. Varying magnetic fields can be set up in a limited volume, fitting well with typical manufacturing applications, modular lines or stand-alone machines for reasonable expenditure.

The so-called WISA-POWER system operates with such magnetic fields (1̃20 kHz), similar to RFID and anti-theft devices, and provides power in a similar fashion to a transformer, but without a core and with a huge air-gap. Typically, current loops are installed around the application. These are fed by "primary" power supplies that set up an alternating current in the loops, producing a magnetic field throughout the application/machine.

Wireless devices within the machine/application each has a small "secondary" coil that picks up the energy from the magnetic field and converts it to electric power.

A typical power value achieved under worst-case conditions (e.g. partly shielding) on the receiver side in the Wireless Proximity Sensor, is 10mW.

The primary power supply system is able to serve typical applications up to 6x6x3 m in size.

Fig. 3.2 shows a typical installment of a WISA system on a robotic type manufacturing application.

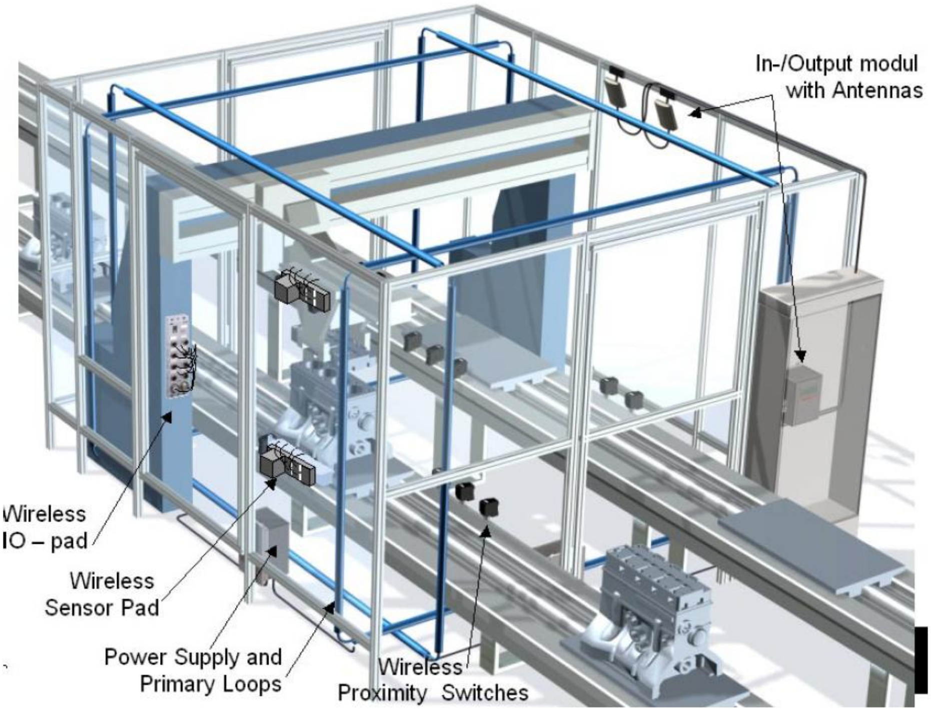

In-/Output modul with Antennas
Wireless IO – pad
Wireless Sensor Pad
Power Supply and Primary Loops
Wireless Proximity Switches

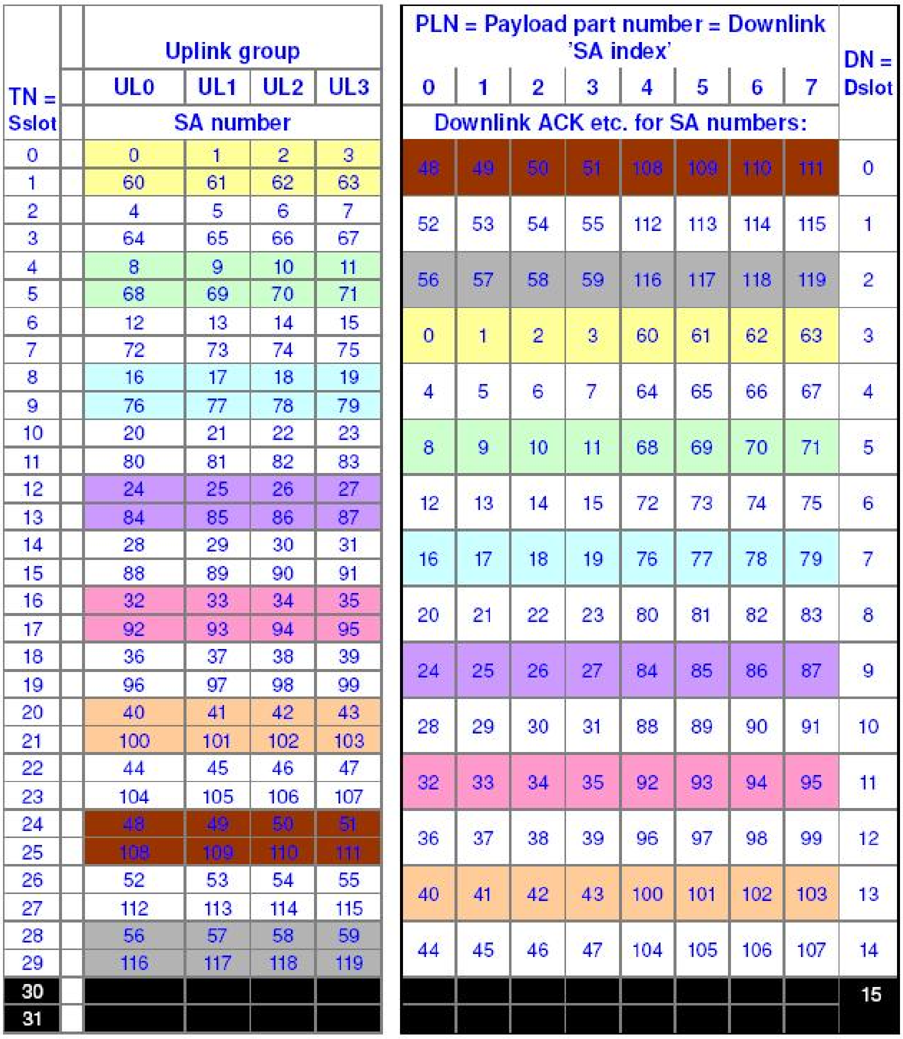
Figure 3.2: WISA devices installed in a typical factory environment.

3.1.1 Media access

In WISA the medium access used is time division multiple access TDMA/FDD/FH. Different formats are used in the uplink and downlink.

The TDMA frame length is T_{frame} 2048 μs. A downlink frame is divided into 16 Dslots (double-slot) and each slot last 128 μs and contains payload for 8 sensor/actuators. This allows for a maximum number of 120 SAs to be connected to one base station (one slot is used for frequency switching). An uplink frame is divided into 32 Sslots (single-slot) à 64 μs. Each SA can transmit an uplink burst in at most one slot per frame.

In order to support 120 SAs per base station each SA is part of one of four uplink groups. The uplinks are separated in frequency (FDMA) within the same subband. This subband is different from the downlink subband (FDD). These 5 frequencies will hop on a per frame basis. WISA supports more than 120 SAs by running multiple cells in parallel using different FH sequences with minimum correlation. [4]



TN = Sslot		Uplink group			
		UL0	UL1	UL2	UL3
		SA number			
0		0	1	2	3
1		60	61	62	63
2		4	5	6	7
3		64	65	66	67
4		8	9	10	11
5		68	69	70	71
6		12	13	14	15
7		72	73	74	75
8		16	17	18	19
9		76	77	78	79
10		20	21	22	23
11		80	81	82	83
12		24	25	26	27
13		84	85	86	87
14		28	29	30	31
15		88	89	90	91
16		32	33	34	35
17		92	93	94	95
18		36	37	38	39
19		96	97	98	99
20		40	41	42	43
21		100	101	102	103
22		44	45	46	47
23		104	105	106	107
24		48	49	50	51
25		108	109	110	111
26		52	53	54	55
27		112	113	114	115
28		56	57	58	59
29		116	117	118	119
30					
31					

PLN = Payload part number = Downlink 'SA index'								DN = Dslot
0	1	2	3	4	5	6	7	
Downlink ACK etc. for SA numbers:								
48	49	50	51	108	109	110	111	0
52	53	54	55	112	113	114	115	1
56	57	58	59	116	117	118	119	2
0	1	2	3	60	61	62	63	3
4	5	6	7	64	65	66	67	4
8	9	10	11	68	69	70	71	5
12	13	14	15	72	73	74	75	6
16	17	18	19	76	77	78	79	7
20	21	22	23	80	81	82	83	8
24	25	26	27	84	85	86	87	9
28	29	30	31	88	89	90	91	10
32	33	34	35	92	93	94	95	11
36	37	38	39	96	97	98	99	12
40	41	42	43	100	101	102	103	13
44	45	46	47	104	105	106	107	14
								15

Table 3.1: Single- and double-slots vs. SA numbers; uplink and downlink. [1]

Tab. 3.1 shows the time and channel allocation for uplinks and downlink in a WISA system. The downlink time slot for a SA is located three double-slots after the uplink time-slot for base station acknowledgment of a transmission. TN (Sslot) 30 and 31 and DN (Dslot) 15 are used for switching frequencies. [4]

3.1.2 Generation of frequency-hopping sequences

The hopping frequency is uniquely determined by (i) the frame number FN, (ii) the cell identifier, *cell_id*, and (iii) the transmission direction $\{DL, UL_0, UL_1, UL_2, UL_3\}$. The FH sequences can be represented by periodic multilevel integer sequences. The construction of these sequences is explained in Section 3.3. This construction ensures low correlation between sequences so multiple base stations can operate in parallel. To enforce high separation of consecutive frames the construction is applied in two stages (outer and inner).

The available frequency band is divided into $n_B = 7$ non-overlapping subbands. Each subband contains $n_I = 11$ frequencies spread 1 MHz apart. This gives that a total of $7 \times 11 = 77$ MHz out of the available 83,5 MHz is utilized.

$W_0(\cdot) =$	[	0	1	2	3	4	5	6	]
$W_1(\cdot) =$	[	0	2	4	6	1	3	5	]
$W_2(\cdot) =$	[	0	3	6	2	5	1	4	]
$W_3(\cdot) =$	[	0	4	1	5	2	6	3	]
$W_4(\cdot) =$	[	0	5	3	1	6	4	2	]
$W_5(\cdot) =$	[	0	6	5	4	3	2	1	]

Table 3.2: Subband hopping sequences.

$X_0(\cdot) =$	[	0	1	2	3	4	5	6	7	8	9	10	]
$X_1(\cdot) =$	[	0	2	4	6	8	10	1	3	5	7	9	]
$X_2(\cdot) =$	[	0	3	6	9	1	4	7	10	2	5	8	]
$X_3(\cdot) =$	[	0	4	8	1	5	9	2	6	10	3	7	]
$X_4(\cdot) =$	[	0	5	10	4	9	3	8	2	7	1	6	]
$X_5(\cdot) =$	[	0	6	1	7	2	8	3	9	4	10	5	]
$X_6(\cdot) =$	[	0	7	3	10	6	2	9	5	1	8	4	]
$X_7(\cdot) =$	[	0	8	5	2	10	7	4	1	9	6	3	]
$X_8(\cdot) =$	[	0	9	7	5	3	1	10	8	6	4	2	]
$X_9(\cdot) =$	[	0	10	9	8	7	6	5	4	3	2	1	]

Table 3.3: Index hopping sequences.

The outer subband-hopping sequence $W_{c_o}(\cdot)$, Tab. 3.2, determines the subband. The inner index-hopping sequence $X_{c_i}(\cdot)$, Tab. 3.3, determines the actual frequency within the selected subband. For a given *cell_id*, where *cell_id* $\in [0..59]$, these sequences are selected in a unique manner and determine the actual frequency-hopping sequence. The outer subband sequence

$W_{c_o}(\cdot)$ is selected by c_o = (*cell_id* div 10) and inner index sequence X_{c_i} by c_i = (*cell_id* mod 10). These sequences are read out at each increment of the frame number FN, where FN count cyclically mod 77, according to the rules: [4]

$$i = FN \bmod 11 \;\rightarrow\; \text{index } X_{c_i}(i) \tag{3.1}$$

$$j = FN \bmod 7 \;\rightarrow\; \text{subband } W_{c_o}(j) \tag{3.2}$$

These indicies are used to determine the frequency sequences for downlink and uplinks:

$$\begin{array}{lllll}
\mathrm{F}_{\mathrm{cell\_id}}\mathrm{DL}(FN) & = [\text{subband} = W_{c_o}(j), & & \text{index} = X_{c_i}(i) &] \\
\mathrm{F}_{\mathrm{cell\_id}}\mathrm{UL}_0(FN) & = [\text{subband} = (W_{c_o}(j)+3) \bmod 7, & & \text{index} = X_{c_i}(i) &] \\
\mathrm{F}_{\mathrm{cell\_id}}\mathrm{UL}_1(FN) & = [\text{subband} = (W_{c_o}(j)+3) \bmod 7, & & \text{index} = (X_{c_i}(i)+3) \bmod 11 &] \\
\mathrm{F}_{\mathrm{cell\_id}}\mathrm{UL}_2(FN) & = [\text{subband} = (W_{c_o}(j)+3) \bmod 7, & & \text{index} = (X_{c_i}(i)+6) \bmod 11 &] \\
\mathrm{F}_{\mathrm{cell\_id}}\mathrm{UL}_3(FN) & = [\text{subband} = (W_{c_o}(j)+3) \bmod 7, & & \text{index} = (X_{c_i}(i)+9) \bmod 11 &]
\end{array}$$

The actual frequency is finally mapped from the subband and index with the formula:

$$\mathrm{f}_{\mathrm{cell\_id}}(FN)/\mathrm{MHz} = 2403 + 11 \times \text{subband} + \text{index} \tag{3.3}$$

Fig. 3.3 illustrates the algorithm for selecting hopping frequency determining the subband from W_{c_o} and index from X_{c_i}, given the cyclically incremented frame number FN.

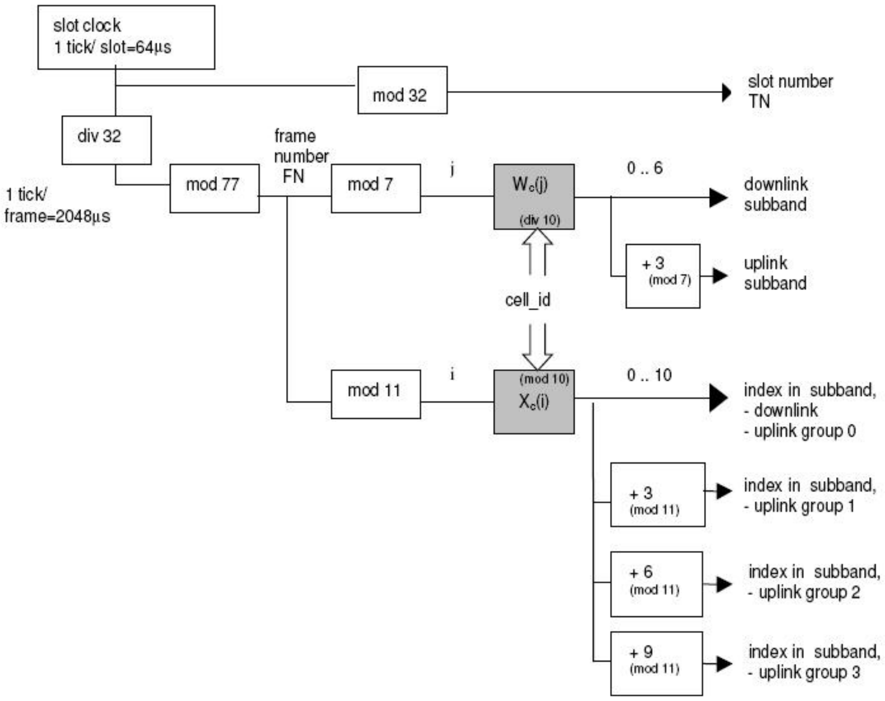


Figure 3.3: Selection of Tx frequencies from the base stations frequency-hopping sequence.

3.2 WLAN frequency usage

A WLAN normally occupies a fixed portion of the available ISM frequency band. This is the portion of the band that must be avoided by WISA so WLAN can exist without interference or degradation of performance. The IEEE 802.11g WLAN specifications define the carrier frequencies to be 2407 MHz + CHNL_ID×5 MHz, where the CHNL_ID is between 1 and 13. There are three relevant bandwidths to consider [5], these are

- specified TX bandwidth of 22 MHz, at -20 dBc (802.11g)
- typical TX bandwidth of 11 MHz, at -10 dBc. This is the dominant contributor of interference from WLAN to other systems such as WISA.
- RX bandwidth of 17 MHz. This is the bandwidth that is susceptible of interference from WISA to WLAN.

Since typical TX bandwidth is smaller than the RX bandwidth and the main objective is to avoid disturbing the WLAN, the RX bandwidth is the relevant bandwidth for WISA to avoid.

3.3 Using Galois field computations to obtain frequency-hopping sequences

This section will explain how Galois field computations are used to produce one-coincidence hopping sequences for WISA.

The performance of hopping sequences can be measured using the periodic Hamming cross-correlation function $H_{XY}(\cdot)$, defined as [6]

$$H_{XY}(\tau) = \sum_{i=0}^{S-1} h(X_i,\ Y_{i+\tau}) \tag{3.4}$$

where

a) $h(a,\ b) = \begin{cases} 0 & \text{if} \quad a \neq b \\ 1 & \text{if} \quad a = b \end{cases}$

b) $X = (X_0,\ X_1,\ ...,\ X_{S-1})$ and $Y = (Y_0,\ Y_1,\ ...,\ Y_{S-1})$ denote two hopping sequences of period S

c) the integer τ describes the shift between the two sequences

d) the sum $(i + \tau)$ is taken modulo S

e) X_i and $Y_i \in \{f_1,\ f_2,\ ...,\ f_q\}$, where f_i is one of the q frequency slots $(q \geq S)$.

The average Hamming cross-correlation function $\overline{H}_{XY}$ is defined to be

$$\overline{H}_{XY} = \frac{1}{S} \sum_{\tau=0}^{S-1} H_{XY}(\tau) \tag{3.5}$$

One-coincidence sequences are defined as non-repeating sequences producing a peak of at most one for the Hamming cross-correlation function for any offset τ. This means that there is only one hit between any two sequences in a period, S, for any given shift between these two sequences.

3.3.1 Sequences constructed from $GF(P)$

This is construction 2 in [6].

1. Select a prime number P
2. Write down the elements of $GF(P)$ in ascending (or descending) order, such that
$$J = \{0,\ 1,\ 2,\ ...,\ P-1\} \tag{3.6}$$
3. Generate a sequence S^i by multiplying the elements of J by a field element $\alpha_i \in GF(P)$, such that
$$S^i = \{0\alpha_i,\ 1\alpha_i,\ 2\alpha_i,\ ...,\ (P-1)\alpha_i\} \tag{3.7}$$
Operations in eqn. (3.7) are to be done mod-P.

Example 1: Let $P = 7$, then $GF(7)$ in ascending order will be $J = \{0, 1, 2, 3, 4, 5, 6\}$. The sequences S^i are shown in Tab. 3.4.

J	0	1	2	3	4	5	6
$S^0 = 0 \cdot J$	0	0	0	0	0	0	0
$S^1 = 1 \cdot J$	0	1	2	3	4	5	6
$S^2 = 2 \cdot J$	0	2	4	6	1	3	5
$S^3 = 3 \cdot J$	0	3	6	2	5	1	4
$S^4 = 4 \cdot J$	0	4	1	5	2	6	3
$S^5 = 5 \cdot J$	0	5	3	1	6	4	2
$S^6 = 6 \cdot J$	0	6	5	4	3	2	1

Table 3.4: Full set of one-coincidence sequences, $P = 7$.

This is the constructor used to derive the existing FH sequence used in the WISA system with $P = 7$ for the subband hopping sequences, and $P = 11$ for hopping within the subbands. The sequence S^0 is not used because it gives no frequency diversity.

3.3.2 Sequences constructed from $GF(P^N)$

This is construction 4 in [6].

1. Select a prime number P
2. Select a primitive polynomial $h(x)$ of degree N over $GF(P)$
3. Select a primitive element α of $GF(P^N)$
4. Write down the nonzero field elements of $GF(P^N)$ as the powers of α, such that
$$K = \{\alpha^0,\ \alpha^1,\ \alpha^2,\ ...,\ \alpha^{P^N-2}\} \tag{3.8}$$
5. Generate a distinct sequence by adding a distinct element α_j of $GF(P^N)$ to the elements of K, such that
$$S^j = \{\alpha^0 + \alpha_j,\ \alpha^1 + \alpha_j,\ ...,\ \alpha^{P^N-2} + \alpha_j\} \tag{3.9}$$

The the sequence K is derived by using the remainder from polynomial division with the primitive polynomial $h(x)$. This is shown in the appendix of [6]. Example 2 gives a simple demonstration of polynomial division.

Example 2: Let $P^N = 2^3 = 8$, $h(x) = x^3 + x + 1$ and $\alpha = 2$. Using polynomial division on all elements in K. Shown here for α^4.

$$\begin{array}{rcccccl}
 & x^4 & & & & & : \; x^3 + x + 1 = x \\
- & (x^4 & + & x^2 & + & x) & \\
\hline
 & & & x^2 & + & x &
\end{array}$$

Subtraction here is performed elementwise modulo P, therefore $(-x) \bmod 2 = x$. The complete set will then be $K = \{1,\ \alpha,\ \alpha^2,\ \alpha+1,\ \alpha^2+\alpha,\ \alpha^2+\alpha+1,\ \alpha^2+1\}$. All derived sequences are shown in Tab. 3.5.

	$\alpha^0 =$	$\alpha^1 =$	$\alpha^2 =$	$\alpha^3 =$	$\alpha^4 =$	$\alpha^5 =$	$\alpha^6 =$
K	001	010	100	011	110	111	101
	$\equiv 1$	$\equiv 2$	$\equiv 4$	$\equiv 3$	$\equiv 6$	$\equiv 7$	$\equiv 5$
$S^0 = K + 0$	1	2	4	3	6	7	5
$S^1 = K + \alpha^0$	0	3	5	2	7	6	4
$S^2 = K + \alpha^1$	3	0	6	1	4	5	7
$S^3 = K + \alpha^2$	5	6	0	7	2	3	1
$S^4 = K + \alpha^3$	2	1	7	0	5	4	6
$S^5 = K + \alpha^4$	7	4	2	5	0	1	3
$S^6 = K + \alpha^5$	6	5	3	4	1	0	2
$S^7 = K + \alpha^6$	4	7	1	6	3	2	0

Table 3.5: Full set of one-coincidence sequences, $P^N = 2^3 = 8$.

3.3.3 Properties of the selected one-coincidence sequence-constructors

Tab. 3.6 shows the number of sequences and the length of each sequence that can be derived using each constructor. The number of sequences produced by the $GF(P)$ constructor is a bit misleading as the S^0 sequence cannot be used. This is because it introduces no frequency diversity as demanded by WISA. Thus the actual number of sequences in set used by WISA is $P - 1$.

Constructor	Peak of periodic CCF	Peak of aperiodic CCF	Average periodic CCF	Number of sequences in set	Sequence length
$GF(P)$	1	1	1	P	P
$GF(P^N)$	1	1	$\frac{P^N-2}{P^N-1}$	P^N	$P^N - 1$

Table 3.6: Comparison of the two selected constructors of one-coincidence sequences. It shows that the peak of the periodic, aperiodic and average Hamming cross-correlation function (CCF) is at most one, thus defined as one-coincidence sequences.

Chapter 4

Simulation module

This simulation module is programmed in Matlab 7.0 and uses the Communications Toolbox. The matlab code for simulation of the environment illustrated in Fig.1.1, is included in Appendix A.

4.1 The main simulation script - mainSim.m

The `mainSim.m` script performs a simulation of collisions between all possible combinations of FH sequences based on the input parameters set by the user. The simulation script can be summarized into three main parts:

- **Parameters.** Parameters that must be set prior to execution of `mainSim.m`. These parameters are set by graphical user input, explained and shown in Section 4.2.1.
- **Computation of FH sequences.** This part is only performed if the frequency-hopping algorithm FH07-1 is selected.
  - *Remapping allowed WISA frequencies.* The allowed WISA frequencies is remapped to a subset of the original WISA frequencies depending on the selected WLAN channels.
  - *Find contiguous*[1] *sequences in the allowed WISA frequencies.* Calculate the number and length of these contiguous sequences.

[1] Contiguous sequences refers to contiguous frequency bands in the set of allowed WISA frequencies, i.e. contiguous bands not divided by any interfering WLAN frequencies.

  - *Select subband configuration.* Checks how much of the available band that can be utilized with the different subband widths. The user then selects the wanted configuration, explained and shown in Section 4.2.1.
  - *Calculate hopping sequences.* Calculate FH sequences using Galois field computations.[2]

- **Calculation of interference.** For all possible pairs of cell IDs the number of frequency collisions are calculated for all possible frame shifts between the two corresponding FH sequences.

The rest of this section will look more closely at each of these main parts.

4.1.1 Parameters

There are some parameters that have to be set by the user before `mainSim.m` can be executed. These are:

- **P_UPLINK** The probability of an uplink slot being used by the device.
- **WLAN_channels** One or more WLAN channels being used in coexistence with the WISA system. If none is selected, the parameter is set to zero.
- **WLAN_BW** The bandwidth of the WLAN channel that has to be avoided. Set to either 11, 17 or 22 MHz.[3]
- **fhAlgorithm** The selected FH algorithm. Set to either FH02 or FH07-1. FH02 is the frequency-hopping algorithm currently implemented in the WISA system. FH07-1 is the new frequency-hopping algorithm proposed in Chapter 6.
- **filename** The simulation data will be saved in a mat-file with this name.

[2] Explained in Section 3.3

[3] See Section 3.2

4.1.2 Computation of FH sequences

If the frequency-hopping algorithm FH02 is selected the script loads the existing FH sequences from the file `fh_02.mat`. When the algorithm FH07-1 is selected the script performs four operations. If a WLAN channel is selected the allowed WISA frequencies are remapped to a subset of the original frequencies depending on WLAN channels and the bandwidth that must be avoided. The second step is to calculate the length of contiguous sequences in the set of allowed WISA frequencies. The script then calculates the maximum number of subbands that can be allocated within these contiguous sequences for all relevant subband widths. Then it calculates the minimum distance between these subbands and the maximum number of downlink-uplink subband pairs that can be used. This maximum number of downlink-uplink pairs is reduced to the closest equal or lower prime, or prime-extension, for Galois field computations. Finally the user is asked to select one of these subband configurations, i.e. the width of subbands and the corresponding number of realizable subbands, before the new FH sequences are derived using Galois field computations. The FH sequences are then saved in the file `calcHS.mat`. These procedures are the steps in the new frequency-hopping algorithm FH07-1 explained in Chapter 6.

4.1.3 Calculation of interference

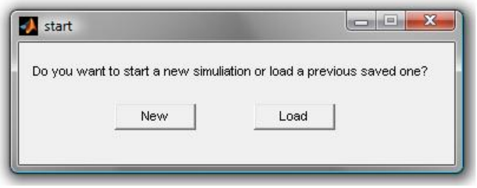
Calculation of interference is performed for all possible pairs of cell ids, i.e. all pairs of FH sequences, in the two base stations. The pseudo code below illustrates how the interference is calculated.

```
for all cell ids in BS1
    load BS1_FHsequence
    for all cell ids in BS2
        load BS2_FHsequence
        for all possible frame offsets
            calculate interference between BS1 and BS2
        end
    end
end
```

The results from this calculation are stored in a matrix and saved under the filename provided by the user.

4.2 Graphical user input

The GUI in the simulation module is made for simplifying the use of the `mainSim.m` script and presenting the simulation data in a well-arranged manner. To initialize the simulation module, make sure the working directory is set to the path of the module and type `init` in the Matlab command window. The user will then be given a choice either to start a new simulation or load data from a saved simulation, Fig. 4.1.



Do you want to start a new simuliation or load a previous saved one?

Figure 4.1: Initialization interface.

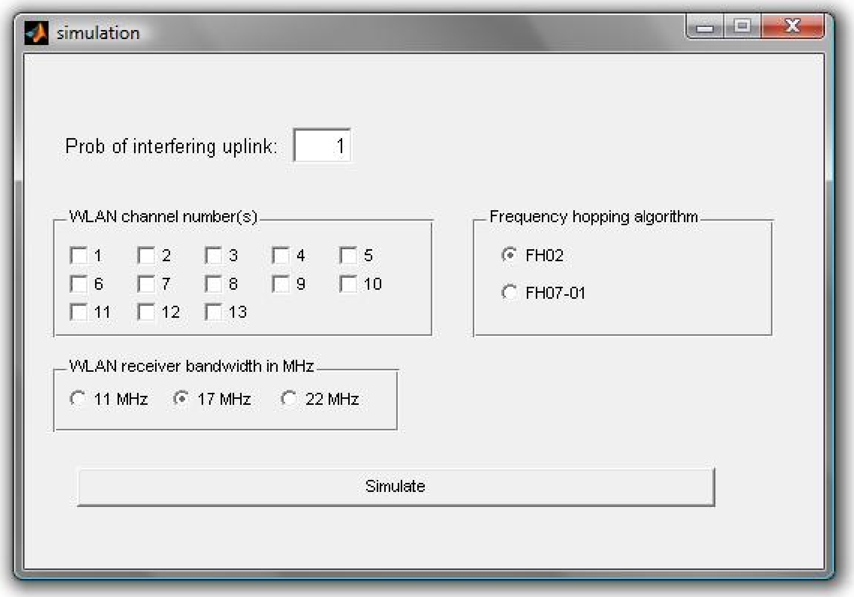
If the `new` option is selected the user will be directed to a new window where the parameters for the new simulation are set. This is explained further in the next section. The `load` option will ask the user to select a simulation file from the specified folder and open a window for analyzing the data. The presentation interface is shown in Section 4.2.2.

4.2.1 Simulation

The simulation interface, Fig. 4.2, provides the user with a simple interface to set the input parameters, defined in Section 4.1.1, for the simulation. The WLAN channels selected will not have any effect unless the algorithm FH07-1 is selected. When the desired parameters are set the user clicks the `Simulate` button and is asked to provide a filename for the data to be saved in. It is important that this file is saved in the given folder, `root\simulations\`, as this is the folder where the GUI will look for the given filename when it presents the simulation data.

After the button is clicked the `mainSim.m` script is executed. The script will calculate the number of realizable subbands in the set of allowed WISA frequencies. The user will be asked to select the wanted subband configuration with the interface shown in Fig. 4.3.

When the wanted configuration is selected, the script will finish the simulation. The text in the `Simulate` button will change and show the progress,



Prob of interfering uplink: 1

WLAN channel number(s)

□ 1 □ 2 □ 3 □ 4 □ 5
□ 6 □ 7 □ 8 □ 9 □ 10
□ 11 □ 12 □ 13

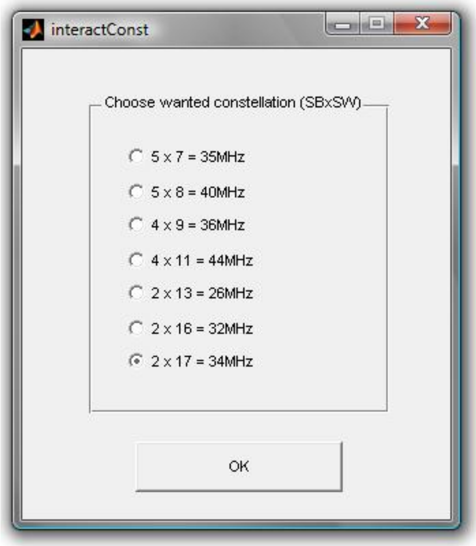
Frequency hopping algorithm

● FH02
○ FH07-01

WLAN receiver bandwidth in MHz

○ 11 MHz ● 17 MHz ○ 22 MHz

Figure 4.2: Simulation interface.



Choose wanted constellation (SBxSW)

○ 5 x 7 = 35MHz
○ 5 x 8 = 40MHz
○ 4 x 9 = 36MHz
○ 4 x 11 = 44MHz
○ 2 x 13 = 26MHz
○ 2 x 16 = 32MHz
● 2 x 17 = 34MHz

Figure 4.3: Interface for selection of subband configuration. Options are calculated based on the user-selected WLAN channels in Fig. 4.2.

in percentage, of the simulation. When the script is finished the results will be presented in a new window shown in the next section.

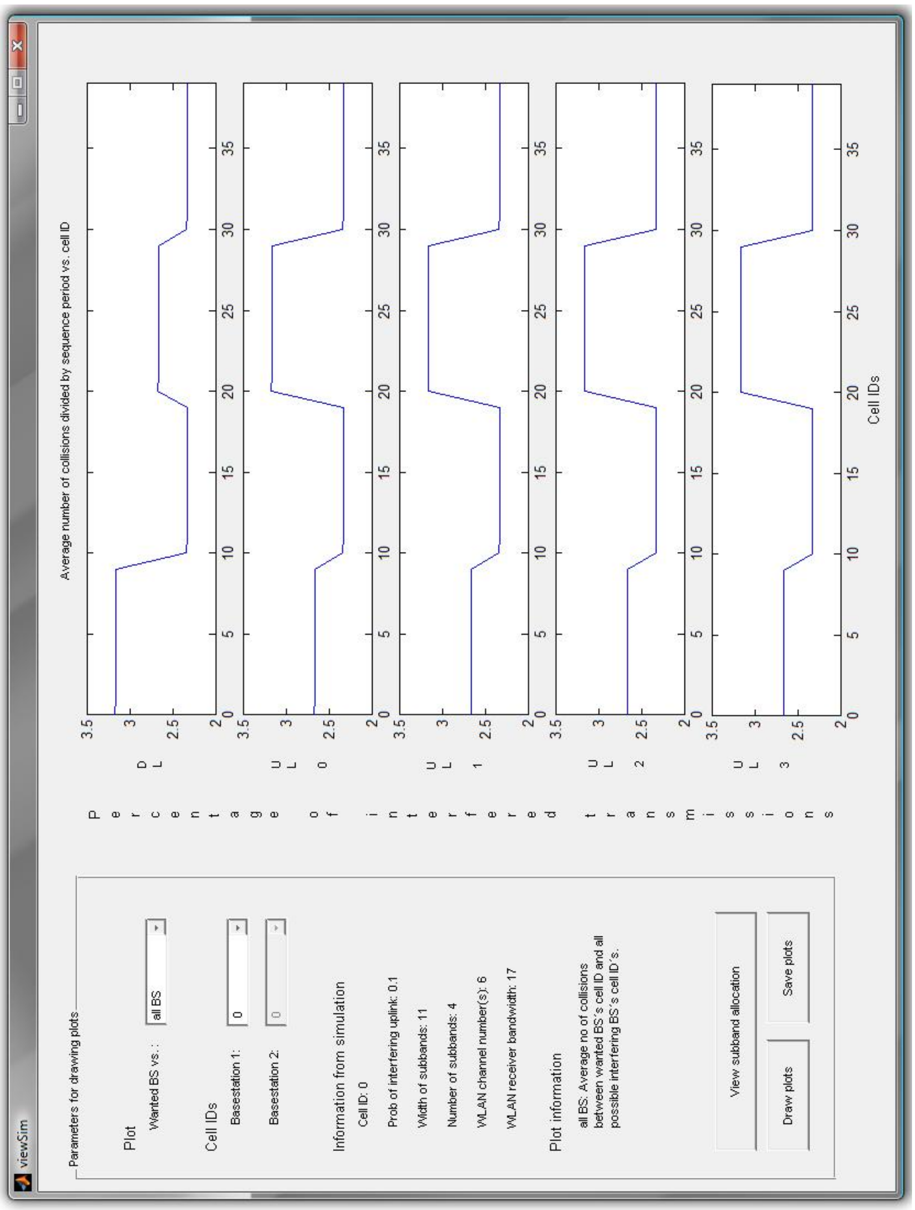
4.2.2 Presentation of results

The presentation interface, shown in Fig. 4.4, gives the user a series of plots so the results from the simulation can be presented and analyzed. The user selects the type of presentation plot and one or two cell ids depending on the plot type. The plots can be saved in `.fig`-files by clicking the `Save plots` button. The filename of these files will have a prefix of the simulation's filename and the rest of the filename will consist of information like cell ids and plot type. The plot types are:

- **all BS** Shows the average percentage of interfered transmissions in CellA[4] for all possible cell ids in base station 2. Option: Cell id in base station 1.
- **one BS** Shows the percentage of interfered transmissions per frame offset in Cell A for all possible frame offset between the two base stations FH sequences. Options: Cell ids in base station 1 and 2.
- **conCol** Shows the maximum number of consecutive interfered transmissions in CellA for all possible frame offsets between the two base stations FH sequences. Options: Cell ids in base station 1 and 2.
- **varCol** Shows the variation of interfered collisions over the possible frame offsets in CellA for all possible cell ids in base station 2. Option: Cell id in base station 1.
- **worCas** Shows the maximum number of consecutive interfered transmissions in CellA for all possible cell ids in base station 2. Option: Cell id in base station 1.

The user can view the subband configuration for the current simulation by clicking the `View subband allocation` button. This will present the user with a window showing the ISM frequency band and the WISA downlink subbands with corresponding uplink subbands, Fig. 4.5. The frequencies used by a WLAN will also be shown.

[4] Base station 1 is the base station in CellA and base station 2 is the base station in CellB. Transmissions occur inside a cell. Cell id is always given to the base station. The notation is shown in Fig. 1.1.


viewSim
Average number of collisions divided by sequence period vs. cell ID
Percentage of interfered transmissions
DL
UL 0
UL 1
UL 2
UL 3
3.5
3
2.5
2
0
5
10
15
20
25
30
35
Cell IDs
View subband allocation
Draw plots
Save plots

Parameters for drawing plots

Plot

Wanted BS vs.: : all BS

Cell IDs

Basestation 1: 0

Basestation 2: 0

Information from simulation

Cell ID: 0

Prob of interfering uplink: 0.1

Width of subbands: 11

Number of subbands: 4

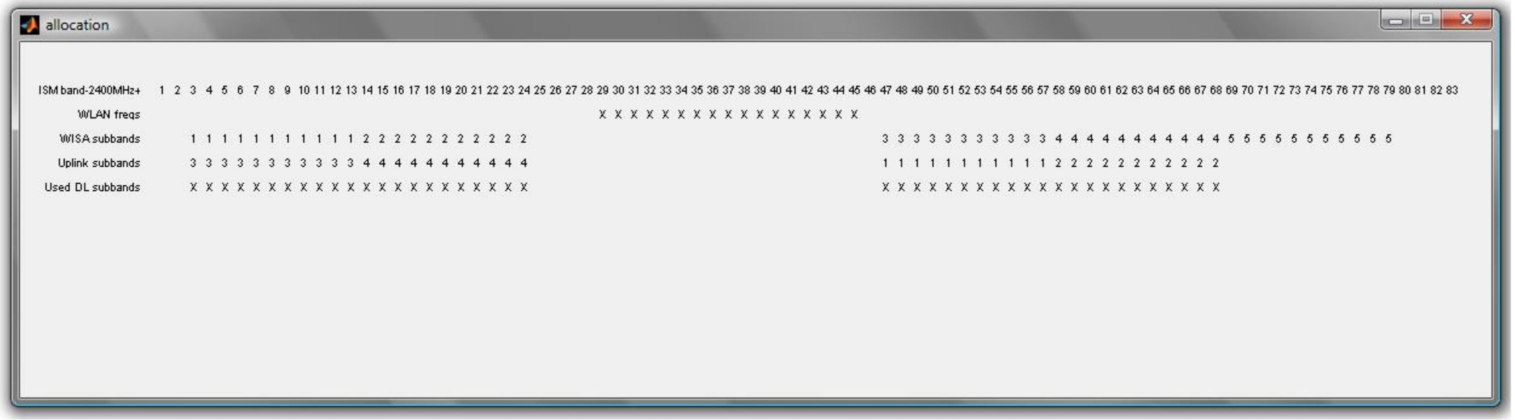
WLAN channel number(s): 6

WLAN receiver bandwidth: 17

Plot information

all BS: Average no of collisions between wanted BS´s cell ID and all possible interfering BS´s cell ID´s.

Figure 4.4: Interface for presentation of simulation data. The information presented in this window will be explained in Chapters 5 and 6.



ISM band-2400MHz+	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
WLAN freqs																					
WISA subbands			1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2
Uplink subbands			3	3	3	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4
Used DL subbands			X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

ISM band-2400MHz+	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42
WLAN freqs								X	X	X	X	X	X	X	X	X	X	X	X	X	X
WISA subbands	2	2	2																		
Uplink subbands	4	4	4																		
Used DL subbands	X	X	X																		

ISM band-2400MHz+	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
WLAN freqs	X	X	X																		
WISA subbands					3	3	3	3	3	3	3	3	3	3	3	4	4	4	4	4	4
Uplink subbands					1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2
Used DL subbands					X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

ISM band-2400MHz+	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83
WLAN freqs																				
WISA subbands	4	4	4	4	4	5	5	5	5	5	5	5	5	5	5	5				
Uplink subbands	2	2	2	2	2															
Used DL subbands	X	X	X	X	X															

Figure 4.5: GUI-window for presentation of subband configuration.

Chapter 5

Analysis of existing Frequency-Hopping sequences

As explained in Section 3.1.2 its cell id uniquely determines the base station's FH sequence. For simplicity, the notation cell id will be used in this chapter when referring to the corresponding FH sequence.

The results presented in this chapter will look at how the downlink and uplinks in CellA[1] are interfered by the downlink and uplinks in cellB. The cell id can be linked to both the cell, CellA or CellB, and the cell's base station, BS1 or BS2. This is because the cell id is given to the base station which controls the cell's frequency-hopping sequence. Interfered transmissions, on the other hand, will always be linked to the cell, CellA or CellB.

5.1 Average interference

The average percentage of interfered transmissions is the average Hamming cross-correlation function divided by the sequence length and multiplied by 100.[2] This will produce results that can be compared with the results from the new FH sequences in Chapter 6. The interference from uplinks are

[1] The simulation evironment is explained in Section 1.4.

[2] This will result in the equation $\frac{100}{S^2}\sum_{\tau=0}^{S-1}\sum_{i=0}^{S-1} h(X_i,\ Y_{i+\tau})$. And this will calculate the average percentage of interfered transmissions between the two FH sequences.

weighted with the probability of the interfering uplink being used.

Analysis of the existing frequency-hopping algorithm shows that the percentage of interfered transmissions in CellA is equal for all possible combinations of cell ids in the two cells. There is a higher level of interference between sequences when uplinks are introduced[3]. The increase depends on the probability of an interfering uplink being used, but it will still be equal for all cell id combinations. Fig. 5.1 shows the percentage of interfered transmissions in CellA with cell id 0 and all possible cell ids in CellB with P(uplink) = 0.1. These plots are equal for all possible values of the cell id in base station 2. When the base stations have been operating for a long time it can be assumed that they have been at all frame offsets the same amount of time and the percentage of interfered transmissions will then be ≈ 1.82%.

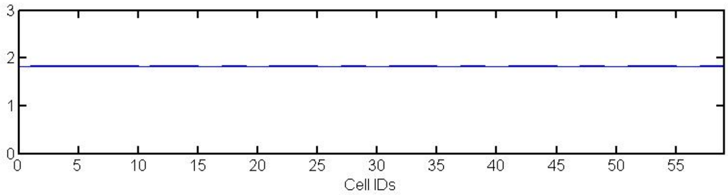


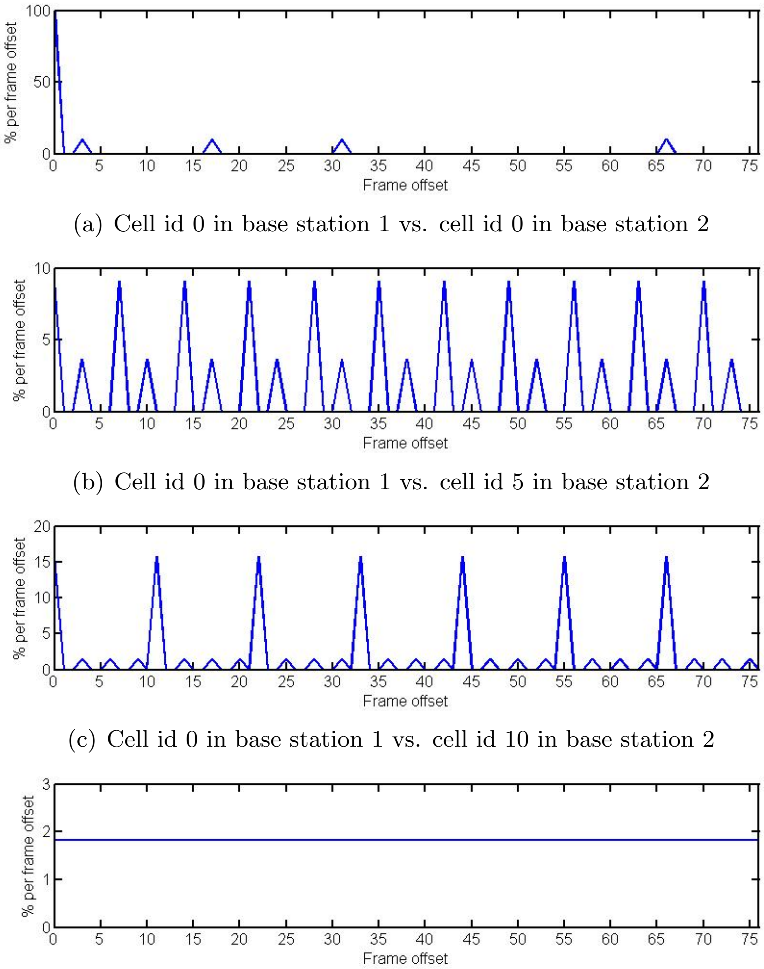
Figure 5.1: Percentage of interfered transmissions in CellA, cell id 0, vs. all possible cell ids in base station 2.

5.2 Distribution of interference

The average number of interfered transmissions in the downlink and uplinks in CellA are the same for all combinations of cell ids. However the distribution of these interferences depends on the cell id in base station 2. Fig. 5.2 shows the percentage of interfered transmissions per frame offset in the downlink in CellA, cell id 0, for 4 different cell ids in CellB. As expected, if the cell id in base station 2 is 0 all transmissions in the downlink in CellA will be interfered when no frame offset is introduced as the two FH sequences are identical. The small peaks are interference from the uplinks in CellB. When the cell id in CellB is 5, same subband hopping sequence as CellA, or 10, same index hopping sequence as CellA, there are multiple peaks with high interference in the downlink. The last plot shows the interference when the cell id in base station 2 is 32, here we see that the interference is evenly distributed over all possible frame offsets between the two cells.

Since the average number of interfered transmission for all frame offsets

[3] P(uplink) > 0



(a) Cell id 0 in base station 1 vs. cell id 0 in base station 2

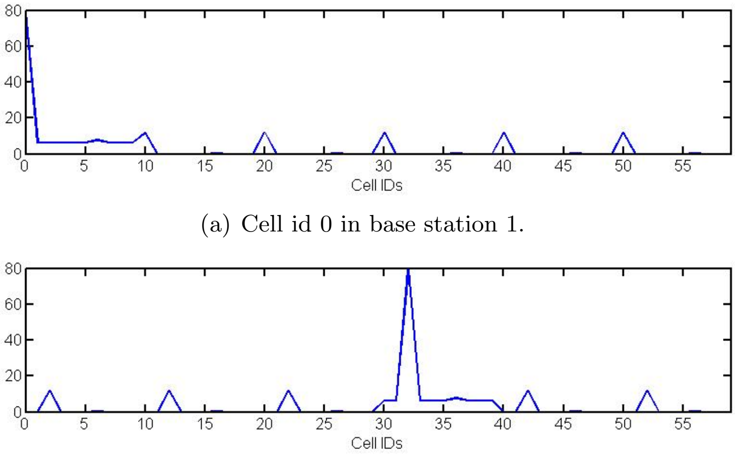
(b) Cell id 0 in base station 1 vs. cell id 5 in base station 2

(c) Cell id 0 in base station 1 vs. cell id 10 in base station 2

(d) Cell id 0 in base station 1 vs. cell id 32 in base station 2

Figure 5.2: Percentage of interfered transmission in CellA per frame offset. The plots for downlink and uplinks are equal.

is the same for all possible combinations of cell ids in the two cells, it is necessary to look at the statistical properties of the distribution of these collisions. The mean is always the same but the variation will vary depending on the cell id combination. The variation of the collisions is shown in Fig. 5.3 for cell id 0 and 32 in CellA and all possible cell ids in CellB. These plots confirm ABBs observation of periods with higher interference when the two cells have equal first or second digit in their cell ids. Self interference naturally gives the highest variation since this is the combination that gives the highest deviation from the mean with 100% interference at zero frame offset. However it is clear that the interference is not equally distributed in time for the combinations where the cell id in CellB has one equal digit with the cell id in CellA. The equal digit problem is related to the *div 10* and *mod 10* functions selecting the subband and index hopping sequences.[4]



(a) Cell id 0 in base station 1.

(b) Cell id 32 in base station 1.

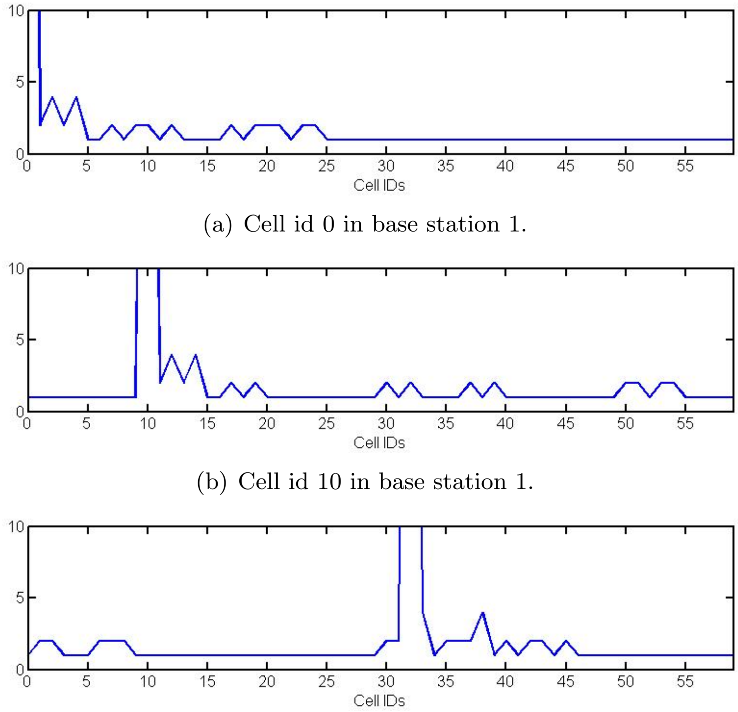
Figure 5.3: The variation of interfered transmissions in the downlink in CellA vs. from all possible cell ids in base station 2.

5.3 Worst-case scenarios

For the WISA system the worst case scenario is the loss of a message from a sensor/actuator. This could lead to a failure in the factory environment and even worse, an accident. A message from a device to a base station will be retransmitted 9 times before the message is lost. This means that there should not be more than 8 consecutive interfered transmissions in a cell. The worst-case scenarios are calculated with the probability of an interfering uplink being used equal to 1. Fig. 5.4 shows the maximum number

[4] Explained in Section 3.1.2

of consecutive interfered transmissions for three different cell ids in CellA. It shows that the worst cases are when the cell id in CellB uses the same subband hopping sequence, i.e. same first digit.



(a) Cell id 0 in base station 1.

(b) Cell id 10 in base station 1.

(c) Cell id 32 in base station 1.

Figure 5.4: Maximum number of consecutive interfered transmissions in the downlink in CellA vs. all possible cell ids in base station 2.

5.4 Frequency separation of transmissions occurring on consecutive frames

As explained in Section 3.1.2 the reason for using a two stage construction of FH sequences is to enforce high separation of consecutive frames. From Fig. 5.5 it is clear that this property is not fulfilled by sequences selected by cell ids from 0 to 9 and 50 to 59. The cell ids from 10 to 49 enforces a separation of at least 11 MHz between consecutive frames. Cell ids 0 and 59 do not have more than 6 occurrences of frame separation less than 11 MHz, but they include an occurrence of two frames only separated by 1 MHz. For cell ids located closer to 9 and 50 the minimum distance is higher and so is also the number of occurrences of frame separation less than 11 MHz.

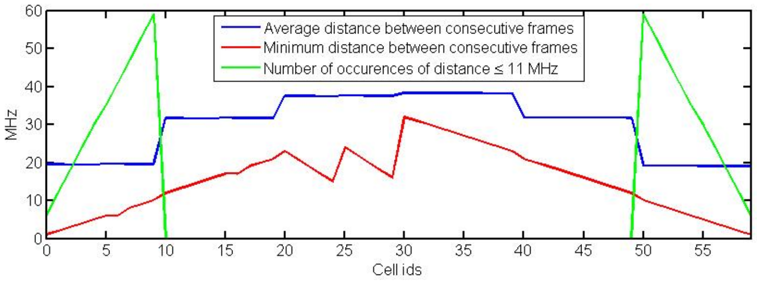


Figure 5.5: Freguency separation of transmissions occurring on consecutive frames for original WISA frequency-hopping sequences.

Chapter 6

Design and Analysis of new Frequency-Hopping sequences

The new frequency-hopping algorithms proposed in this chapter are designed in collaboration with Dacfey Dzung at ABB Corporate Research, Segelhof, Switzerland. [5]

6.1 FH07-1 - Extension of existing frequency-hopping algorithm

FH07-1 generalizes the existing algorithm for generation of FH sequences, FH02[1]. n_B pairs of subbands, with at least D duplex spacing, are identified, where subbands are non-overlapping and have width n_I MHz. The set of allowed WISA frequencies are remapped to a smaller set of the original 77 frequencies where the WLAN frequency band is excluded.

6.1.1 Generation of FH sequences

The sequence construction of [6] is applied in two stages, inner and outer. First a set of n_B-ary sequences W is determined using a suitable construction method in Section 3.3, $G(p)$ or $G(p^N)$. The sequence W is used to select

[1] Explained in Section 3.1.2

the subband on each hop. Given the selected subband, a separate n_I-ary FH sequence X determines the frequency index within the subband. The sequences X is also found by a suitable construction method in Section 3.3.

Algorithm FH07-1

1. Select n_I width of subbands. n_I should be either a prime, or a prime extension[2], in order to use the constructors presented in Section 3.3. From this and the restrictions given in Section 2.3 the allowed values of n_I are $\{7,\ 8,\ 9,\ 11,\ 13,\ 16,\ 17\}$. Larger n_I allows larger variation in the uplink assignments (less correlation between the two cell's uplinks), but will lower the number n_B of available subbands (lower variation in subband hopping sequences).
2. Non-overlapping subbands in the set of allowed frequencies are identified.
3. All pairs of subbands which are at least $D = 22$ MHz apart are enumerated. The number of such pairs is designated as n_B.
4. If n_B is not a prime, or a prime extension, n_B is reduced to the next lower integer satisfying that property, i.e. $n_B \in \{2,\ 3,\ 4,\ 5,\ 7,\ 8,\ 9,\ 11\}$.
5. The set of subband hopping sequences W_k is generated according to a suitable method in Section 3.3, using $GF(n_B)$.
6. The set of index hopping sequences X_i is generated by suitable constructor in Section 3.3, using $GF(n_I)$.
7. W_k and X_i are combined as in FH02, explained in Section 3.1.2.

This will produce $K \times I = S$ new hopping sequences, where K is the number of sequences in W_k and I is the number of sequences in X_i. The same method used in FH02 can be used for FH07-1; For a given *cell_id*, where *cell_id* $\in [0..S-1]$, these sequences can be uniquely selected by the functions $c_o =$ *cell_id* div I and $c_i =$ *cell_id* mod I. The functions selects the outer subband sequence $W_{c_o}(\cdot)$ and the inner index sequence $X_{c_i}(\cdot)$.

Summary The new frequency-hopping algorithm uses the same two stage hopping sequences as the existing FH02 algorithm. Every subband must be “paired” with a another subband spaced at least 22 MHz above or below in

[2] A prime extension the same as an integer power of a prime.

the set of allowed WISA frequencies for duplex communication. The number of such possible pairs are designated as the number of realizable subbands. For computation of FH sequences this number must either be prime, or a prime extension. If this is not the case the number is reduced to the closest number fulfilling this requirement. Different widths of the subband can be selected for maximum utilization of the available frequency band. Cell ids are used for unique selection of a FH sequence in a cell.

6.1.2 Analysis

There are 7 × 13 possible combinations of width of subband, n_I, and WLAN channels. It is not possible to analyze all these combinations in regard to all possible combinations of cell ids for each of them. Hence the results presented will focus on the algorithm's performance when one of the three standard WLAN channels are used, channel 1, 6 or 11.

	WLAN channels					
	1		6		11	
n_I	n_B	Total MHz	n_B	Total MHz	n_B	Total MHz
7	8	56	5	35	8	56
8	5	40	5	40	5	40
9	5	45	4	36	5	45
11	4	44	4	44	2	22
13	2	26	4	52	2	26
16	0	0	2	32	0	0
17	0	0	2	34	0	0

Table 6.1: The achievable number of subbands, n_B, and the total of used MHz possible for a given width of the subband, n_I, and a collocated WLAN channel. The WLAN bandwidth that must be avoided is set to be 17 MHz.

The algorithm for allocating the non-overlapping subbands in the set of allowed WISA frequencies is not optimal. The algorithm places the subbands in contiguous band as far to both sides of the WLAN channel as possible.

Fig. 6.1(a) shows a non-optimal allocation of subbands in the ISM band. In this case all 6 subbands could have been utilized by shifting the subbands 1 to 3 up to the edge of the WLAN band and thus creating the necessary duplex spacing, $D = 22$ MHz, between subband 1 and 4. The new uplink-downlink pairing would be 1-4, 2-5, 3-6, 4-1, 5-2 and 6-3. This would have utilized $n_B = 6$ subbands, but since 6 is not a prime, or prime extension, the actual realizable number of subbands would be $n_B = 5$.

In Tab. 6.1 three such cases of non-optimal subband allocation has been located. In the case of $n_I = 13$ and WLAN channel 11 n_B is corrected from 0 to 2, and for the cases $n_I = 9$ and WLAN channels 1 and 11, Fig. 6.1(c), n_B has been corrected from 4 to 5.

There may be other occurrences of non-optimal values of n_B since all corrections are made manually by visual studies of figures like the the ones in Fig. 6.1. Results presented later in this chapter are from configurations where n_B is found to be the maximum realizable number of subbands, like Fig. 6.1(b).

Average percentage of interfered transmissions There are two properties of the new FH algorithm that counteract the equal average percentage of interfered transmissions property of the FH02 algorithm. First is the use of sequences derived from prime extensions, $GF(P^N)$ in Section 3.3. In each of these sequences one of the available integers are left out[3], e.g. S^0 does not include 0. This means that one subband or index is left out of the hopping sequence, depending on where the $GF(P^N)$ construction is applied. The second property is the irregular mapping of downlink-uplink subband pairs. For FH02 the uplink is always in the third subband after the subband where the downlink is located, counting modulo-7. In FH07-1 the subbands may not be located in the same regular manner because pairs are allocated using a brute-force algorithm checking all possible combinations to ensure that the maximum number of pairs are found. This could result in a subband being paired as an uplink subband but not as a downlink subband when n_B is reduced to a prime or prime extension.

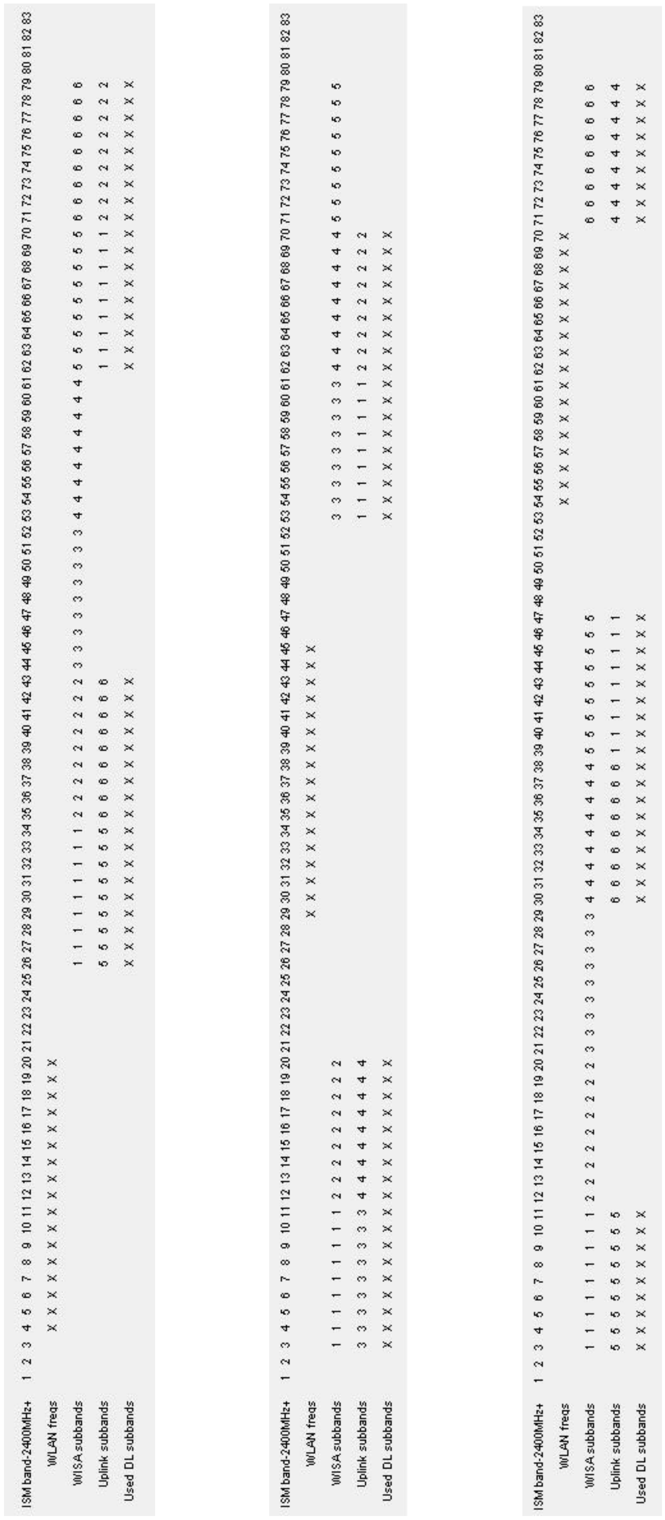
$\mathbf{n_I = 7}$ Because of the minimum of 2 MHz spacing between uplinks[4], there is only one way to allocate four uplink groups in a 7 MHz wide subband. This introduces less diversity between uplinks in the two cells, i.e. every time the two cells use the same uplink subband, their respective uplinks will be at the same frequencies. The index hopping sequences will never be used for uplinks. The result is that if the two cells use the same subband hopping sequence, i.e. equal result from the *cell_id* div I calculation, the worst-case scenarios are inadequate. The plot for uplinks 0 to 3 in Fig. 6.3 confirms these effects.

Fig. 6.2 shows the average percentage of interfered transmissions in CellA[5]

[3] See table 3.5 in Section 3.3.2.

[4] Explained in Section 2.3.

[5] The simulation evironment is illustrated in Fig. 1.1



ISM band-2400MHz+	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83
WLAN freqs				X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X																																																														
WISA subbands																										1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	6				
Uplink subbands																										5	5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	6																			1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2				
Used DL subbands																										X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X																			X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X				

(a) WLAN channel 1.

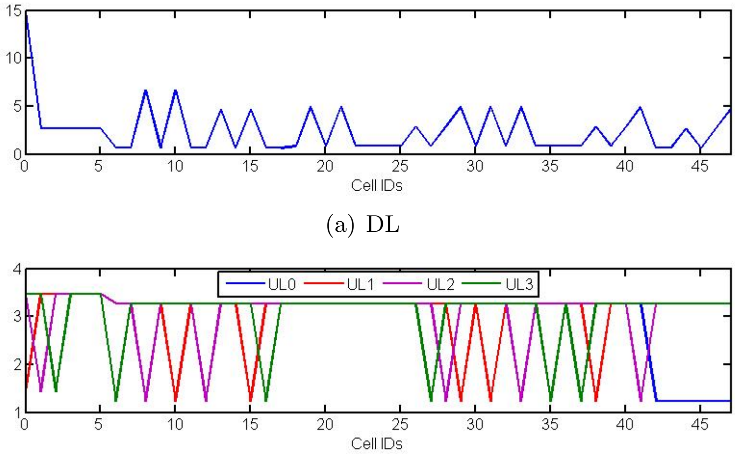
ISM band-2400MHz+	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83
WLAN freqs																													X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X																																												
WISA subbands			1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2																																	3	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5				
Uplink subbands			3	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4																																	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2													
Used DL subbands			X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X																																	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X													

(b) WLAN channel 6.

ISM band-2400MHz+	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83
WLAN freqs																																																						X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X													
WISA subbands			1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5																									6	6	6	6	6	6	6	6	6				
Uplink subbands			5	5	5	5	5	5	5	5	5																			6	6	6	6	6	6	6	6	6	1	1	1	1	1	1	1	1	1																									4	4	4	4	4	4	4	4	4				
Used DL subbands			X	X	X	X	X	X	X	X	X																			X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X																									X	X	X	X	X	X	X	X	X				

(c) WLAN channel 11.

Figure 6.1: Subband allocation for $n_I = 9$ and each of the three standard WLAN channels, 1, 6 and 11.

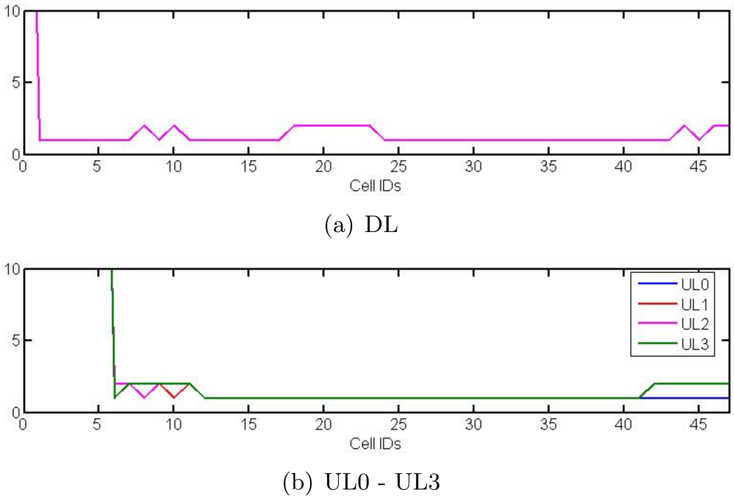


(a) DL

(b) UL0 - UL3

Figure 6.2: Average percentage of interfered transmissions in CellA with cell id 0 vs. all possible cell ids in CellB. $n_I = 7$, $n_B = 8$ and WLAN channel 1.

with cell id 0. The interference at the downlink appears random and it is clear that there are some combinations of cell ids that are better than others. The uplinks also vary, but none of the uplinks are "good" for the same cell id combination. As long as the worst-case scenario is acceptable the average percentage of interfered transmissions should not introduce any problem. Fig. 6.3 shows that the worst-case scenarios are acceptable outside the area of cell ids using the same subband hopping sequences, cell ids 0 to 6.

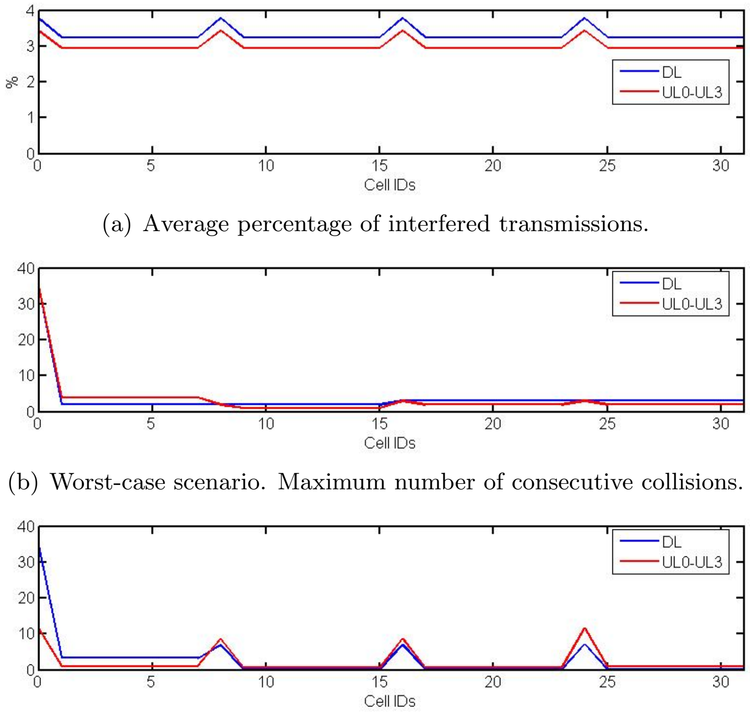


(a) DL

(b) UL0 - UL3

Figure 6.3: Worst-case scenarios. Maximum number of consecutive interfered transmissions in CellA with cell id 0 vs. all possible cell ids in CellB. $n_I = 7$, $n_B = 8$ and WLAN channel 1.

Other values of $\mathbf{n_I}$ To produce a basis for analysis the next figures will show the average percentage of interference, worst-case scenario and variation plots for a few of the configurations from Tab. 6.1.

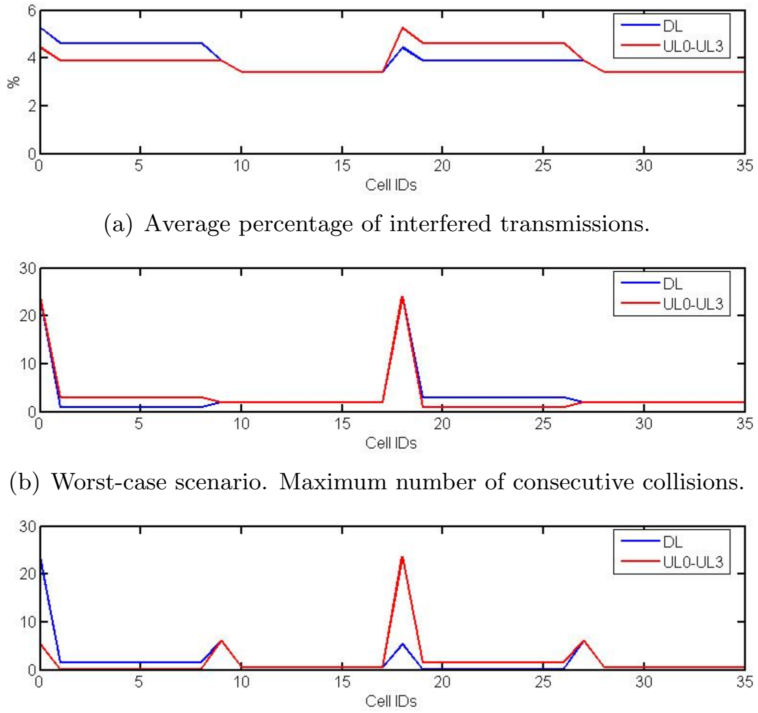


(a) Average percentage of interfered transmissions.

(b) Worst-case scenario. Maximum number of consecutive collisions.

(c) Variation of interference.

Figure 6.4: Average interference, worst-case scenario and variation of interference plots for cell id 0 in base station 1, $n_I = 8$, $n_B = 5$ and WLAN channel 1.

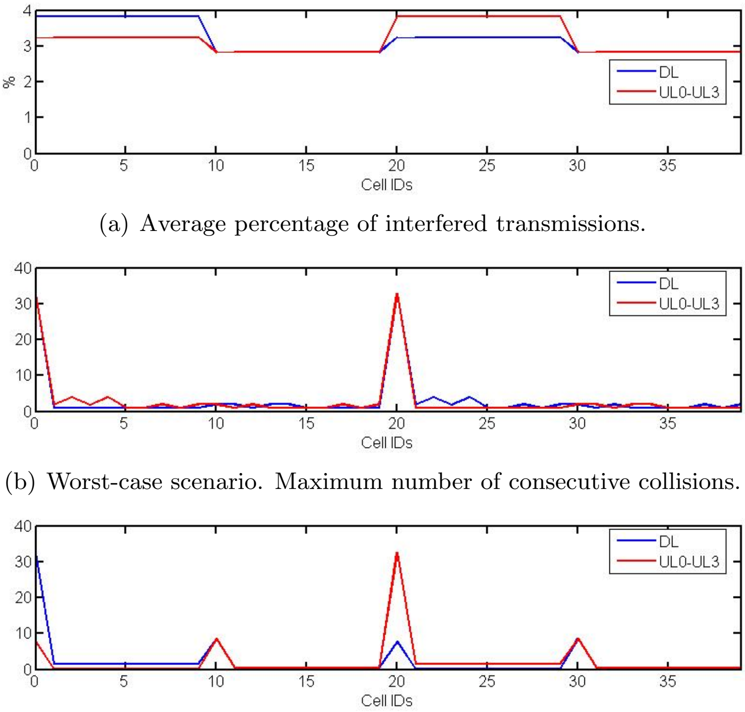


(a) Average percentage of interfered transmissions.

(b) Worst-case scenario. Maximum number of consecutive collisions.

(c) Variation of interference.

Figure 6.5: Average interference, worst-case scenario and variation of interference plots for cell id 0 in base station 1, $n_I = 9$, $n_B = 4$ and WLAN channel 6.

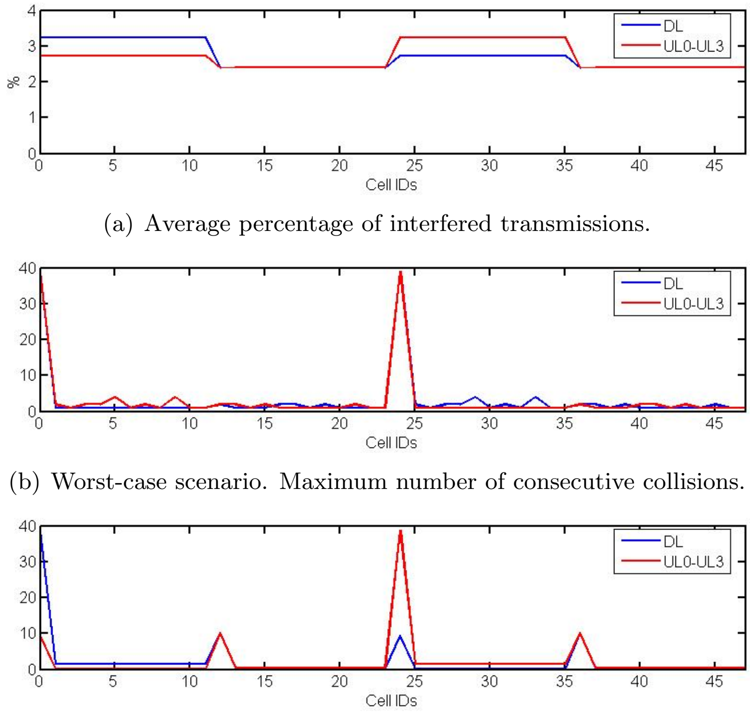


(a) Average percentage of interfered transmissions.

(b) Worst-case scenario. Maximum number of consecutive collisions.

(c) Variation of interference.

Figure 6.6: Average interference, worst-case scenario and variation of interference plots for cell id 0 in base station 1, $n_I = 11$, $n_B = 4$ and WLAN channel 6.



(a) Average percentage of interfered transmissions.

(b) Worst-case scenario. Maximum number of consecutive collisions.

(c) Variation of interference.

Figure 6.7: Average interference, worst-case scenario and variation of interference plots for cell id 0 in base station 1, $n_I = 13$, $n_B = 4$ and WLAN channel 6.

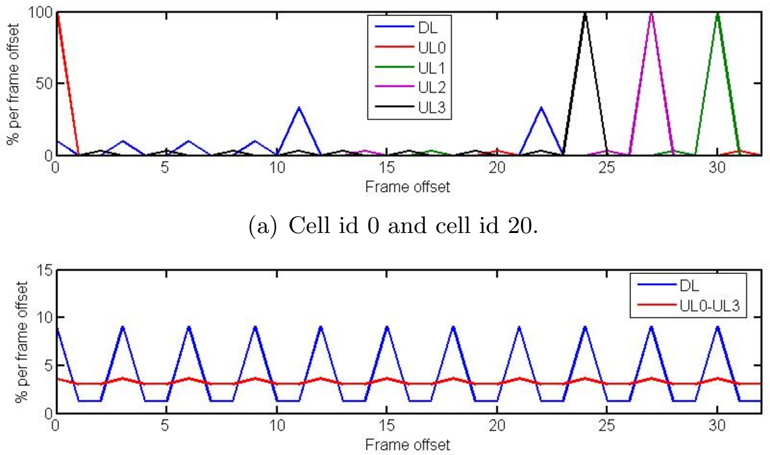
The plots show that the new sequences still can operate with at least two base stations in parallel with low interference. It is possible to find combinations of cell ids for $n_I \in \{7, 8, 9, 11, 13\}$ and all three analyzed combinations of WLAN channels. Which of the possible values of n_I that utilizes the available bandwidth the most depends on the selected WLAN channel.

The same effects as those noticed for the existing FH sequences with equal first or second digit in the cell id are also apparent with the new sequences. Of course the equal digit phenomenon is a special case of the div 10 and modulo 10 functions for allocation of inner and outer hopping sequences. In the new algorithm this can be summarized by periods of higher interference when the two parallel cells use the same subband- or index-hopping sequences, div I and mod I.[6]

There seems to be a special case of interference when $n_B = 4$. When the cell id in CellB is exactly two times the number of unique index-hopping sequences, X_i, higher or lower than the cell id in CellA, the maximum number of consecutive interfered transmissions is the same as when the the two cell ids are equal. This is shown in figures 6.5, 6.6 and 6.7. Fig. 6.8(a) shows the percentage of interfered transmissions between two cells with cell id 0 and cell id 20 for $n_I = 11$, $n_B = 4$ and WLAN channel 6. This plot shows that this combination has 100% interference at different frame offsets for the uplinks and two points of ≈40% interference for the downlink.

There are also occurences of high interference when cells are using the same subband-hopping sequence, and then particularly for the downlink. This is shown in Fig. 6.8(b) for the same subband configuration as above for cell ids 0 and 5. These effects has been confirmed in simulations of other combinations of cell ids and are not random occurrences in the selected plots.

[6] See Section 6.1.1



(a) Cell id 0 and cell id 20.

(b) Cell id 0 and cell id 5.

Figure 6.8: Percentage of interfered transmissions per offset for all possible frame offsets between the two given cell ids. $n_I = 11$, $n_B = 4$ and WLAN channel 6.

6.2 FH07-2 - Single stage generation of frequency

This is a proposed frequency-hopping algorithm for further work and will not be analyzed in this thesis.

6.2.1 Generation of FH sequences

FH07-2 uses a one-stage generation of FH sequences where downlink frequencies do not use subbands. Hence the downlink frequency can be any of the allowed WISA frequencies. The paired uplink subband is chosen to satisfy the duplex spacing requirement within the set of allowed WISA frequencies. This allows for a greater diversity in the frequency-hopping sequences of the constantly transmitting base station.

Algorithm FH07-2

1. Select n_I width of subbands. Same aspects as in step 1 of algorithm FH07-1 are relevant.
2. In the given set of allowed WISA frequencies, pick the next allowed downlink frequency f_{DL}.

3. For this given f_{DL}, find a “paired” uplink subband in the set of allowed WISA frequencies, which is at least $D = 22$ MHz away, above or below f_{DL}. This “pairing” is in general not unique.

4. Continue from step 2 until all possibilities are considered. The result is non-unique in general. The number found of such pairs is designated as n_F.

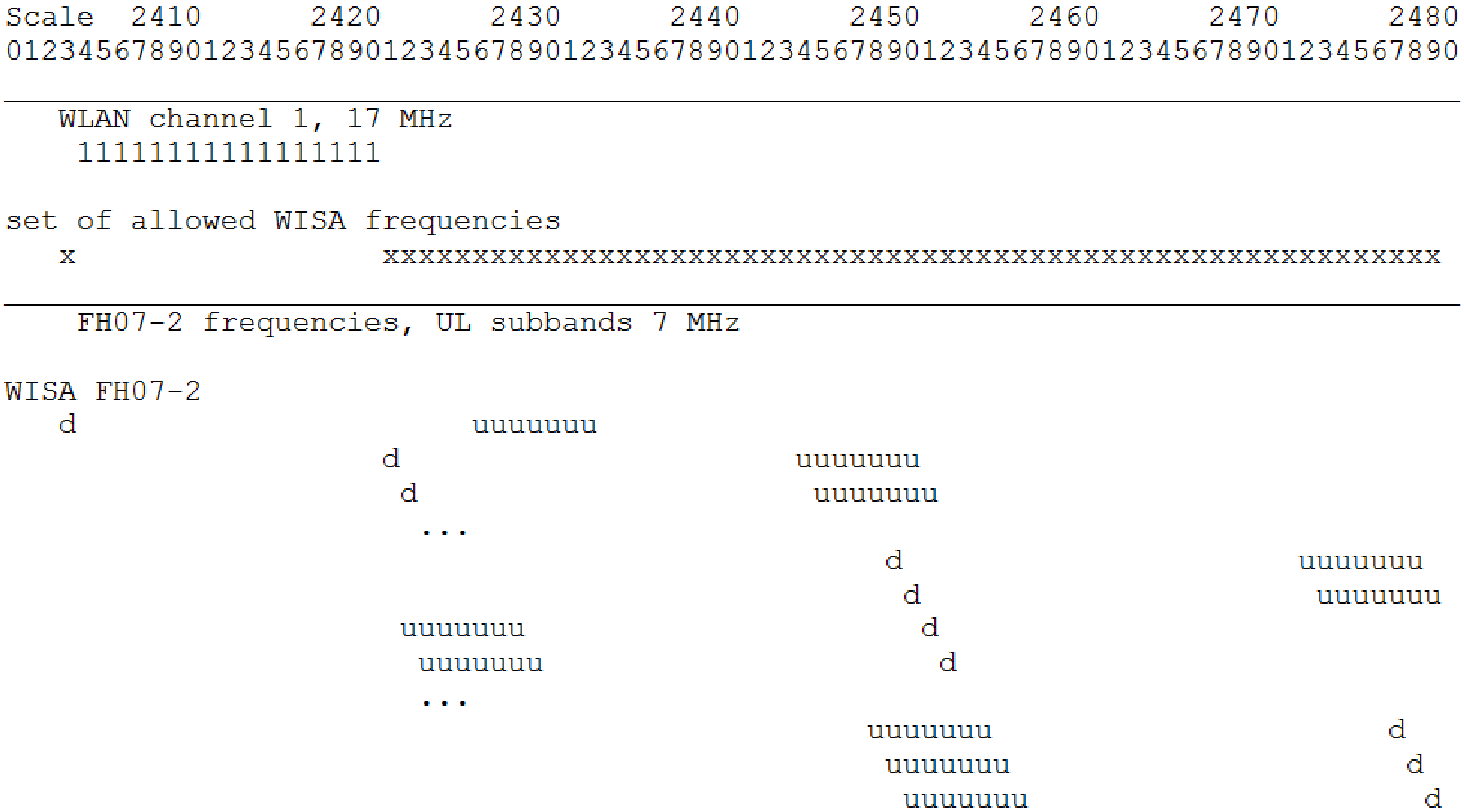
5. If n_F is not a prime, or a power of a prime, reduce n_F to the next lower integer satisfying that property.

6. Generate the set of hopping sequences X_i, according to a suitable method of Section 3.3, using $GF(n_F)$.

7. Map X_i to the actual frequency in the set of allowed WISA frequencies. f_{DL} is mapped from $X_i(n_F)$ and the corresponding uplink is given by the “pairing” in step 3. The index of the u-th uplink frequency $f_{UL,u}$ is determined as in step 7 of FH07-1.

Fig. 6.9 illustrates how a FH07-2 sequence might look. The base stations downlink frequency hops 1 MHz for every increment of the frame number and the 7 MHz wide uplink subband is spaced 22 MHz from the downlink frequency. It is important to emphasize that this sequence is made by hand and may not represent an actual FH sequence derived by the FH07-2 algorithm.

Comparison of FH07-1 and FH07-2 The FH07-2 algorithm will have a finer quantization, since it does not rely on non-overlapping subbands, and can thus often make use of more frequencies than FH07-1. FH07-2 can however for some sequences be in the same area of the band for several consecutive frames. Large spacing of consecutive frames was desirable in the FH02 algorithm to avoid consecutive frames in the same “bad” bands, e.g. interfered by a WLAN. Since the new frequency-hopping algorithm avoids the WLAN band, this is no longer such an important requirement. The sequences derived by $GF(n_F)$ will not have the problem of closely spaced cells using the same subband- or index-hopping sequences.



```
Scale  2410      2420      2430      2440      2450      2460      2470      2480
012345678901234567890123456789012345678901234567890123456789012345678901234567890

   WLAN channel 1, 17 MHz
    11111111111111111

set of allowed WISA frequencies
   x                 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

    FH07-2 frequencies, UL subbands 7 MHz

WISA FH07-2
   d                      uuuuuuu
                     d                      uuuuuuu
                      d                      uuuuuuu
                       ...
                                                 d                      uuuuuuu
                                                  d                      uuuuuuu
                      uuuuuuu                      d
                       uuuuuuu                      d
                       ...
                                                uuuuuuu                      d
                                                 uuuuuuu                      d
                                                  uuuuuuu                      d
```

Figure 6.9: Illustration of FH07-2's utilization of the allowed WISA frequencies when WLAN channel 1 is selected.

Chapter 7

Conclusions

The objective of the project resulting in this thesis was to create a Matlab module for simulation of the correlation between frequency-hopping sequences in two closely spaced WISA cells. This tool was then to be used for analysis of the existing frequency-hopping algorithm as well as implementation and analysis of new algorithms.

The developed simulation module does indeed meet the objectives. One weak aspect is the non-optimal algorithm allocating subbands in the set of allowed WISA frequencies for the FH07-1 algorithm. There are also a couple of GUI-related issues. However, the most important features are working and the simulation module will provide the user with the ability to analyze cross-correlation between FH sequences.

Analysis of the existing FH sequences confirmed ABB's observation of higher interference between cells with equal first or second digit cell ids. The reason for this is that these cell id combinations use either the same subband- or the same index-hopping sequences. It was also pointed out that not all cell ids met the requirement of large frequency separation between consecutive frames. Cell ids 0 to 9 and 50 to 59 can introduce as little as 1 MHz spacing between consecutive frames.

The new frequency-hopping algorithm FH07-1 is able to provide secure transmission for at least two closely spaced cells. In the special case of subband width = 7 MHz the FH sequences in the two cells cannot be using the same subband hopping sequence as this will ensure interference of all transmissions in all four uplink groups. Even so, the large number of possible subbands, and thus a large number of subband hopping sequences, still makes this an interesting configuration. It is hard to conclude on which subband widths are the best since the number of closely spaced cells able to operate with acceptable self-interference is an important issue and has not been investigated in this thesis. Two conclusions can still be drawn; First, interference introduced by using the same subband- or index-hopping sequences in the two cells becomes less of an issue for large subband widths. Second, smaller subband widths will give a larger number of realizable subbands and thus provide a larger number of subband hopping sequences. Analysis shows that is likely to enable a larger number of collocated cells.

7.1 Recommendations for further work

As mentioned above, the algorithm for subband-allocation is not optimal. This algorithm should be replaced with one checking the number of realizable subbands for possible allocations of subbands. This will give ABB the opportunity of analyzing the FH07-1 algorithm for all possible subband widths.

The new FH algorithm FH07-2 has some interesting properties, and analysis of these properties will be interesting for ABB. This algorithm can in most cases utilize more of the available frequency band and thus produce longer FH sequences. For further investigation of other new frequency-hopping algorithms, the paper [7] may be of interest.

Appendix A

Matlab code

A.1 mainSim.m

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% mainSim.m
%
% The main simulation script. Performs simulation of
% correlation between FH sequences in two closely spaced
% WISA cells. Computes new FH sequences avoiding
% user selected frequencies in the ISM band.
%
% Erik Sandnes, June 2007
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Global variables can be used in any function used by this
% script.
global P_UPLINK
global NoOfUplinks
global Period
global fhAlgorithm
global clicked
global selected

% Default values of the parameters.
P_UPLINK = 0.1; % Probability of an interfering UL being used
WLAN_channels = 0; % [1..13], 0 is off.
WLAN_BW = 17; % typical 11, 17 or 22MHz.
fhAlgorithm = 2; % 1: FH_02, 2: FH_07_01
NoOfUplinks = 4; % No of uplinks in the two cells
Period = 77; % Length of FH sequences

```

```
% Loads the user selected variables.
load('matfiles\simVars')
P_UPLINK = sim_P_UPLINK;
WLAN_channels = sim_WLAN_channels;
WLAN_BW = sim_WLAN_BW;
fhAlgorithm = sim_fhAlgorithm;
filename = sim_filename;

% Defing FH sequences. Loads from file if FH02 is selected,
% computes new if FH07−1 is selected.
if fhAlgorithm == 1
    hsFile = 'matfiles\fh_02';
    cellIds = 0:59;
    Period = 77;
    WLAN_usedFreqs = [];
else
    % Remapping the allowed frequencies for WISA to subset
    % of the original 79 frequency bands. Removing the
    % frequencies used by WLAN, decided by the WLAN_channel
    % and the WLAN_BW.
    WISA_allowedfreqs = remapWISAfreqs(WLAN_channels, WLAN_BW);

    % Calculation of new Hopping Sequences and CellIDs. Galois
        Field.
    cellIds = computeHoppingSequences(WISA_allowedfreqs);
    hsFile = 'matfiles\calcHS';
end

% Matrices for saving simulation data.
collisionMatrix = zeros(length(cellIds)^2*(NoOfUplinks+1),
    Period);
maxSucCollsMatrix = zeros(length(cellIds)^2*(NoOfUplinks+1),
    Period);
cell_cell_Interference = zeros(length(cellIds)*(NoOfUplinks+1),
    length(cellIds));

% Calculation of corrolation between sequences and maximum
% number of consecutive interfered transmissions
for cellid_1 = cellIds
    % "Loads" the subband− and index−hopping sequences for
        CellA.
    [cell1_sbHS, cell1_indexHS] = cellHSAllocation(cellid_1,
        hsFile);
    for cellid_2 = cellIds
        % "Loads" the subband− and index−hopping sequences for
            CellB.
        [cell2_sbHS, cell2_indexHS] = cellHSAllocation(cellid_2
            , hsFile);

        % Calculates the complete sequence for each cell.
        [cell1_sequence, cell2_sequence] = sequenceAllocation(
            cell1_sbHS, cell1_indexHS, cell2_sbHS,
            cell2_indexHS);

        % Temporary simulation data files.
        collisions = zeros(Period,NoOfUplinks+1);
        maxSucColls = zeros(Period,NoOfUplinks+1);

        % Calculation of the correlation and maximum number of
        % consecutive collision for all possible cyclic shifts between
        % the two sequences.
        for FNshift=0:Period-1
            cell2_sequence_shifted = circshift(cell2_sequence,[0,FNshift]);
            [collisions(FNshift+1,:), maxSucColls(FNshift+1,:)] = collisionCounter_v2(cell1_sequence, cell2_sequence_shifted);
        end

        % Saaving of data in to the matrices.
        for k = 1:NoOfUplinks+1
            collisionMatrix(cellid_1*length(cellIds)*(NoOfUplinks+1)+cellid_2*(NoOfUplinks+1)+k,:) = collisions(:,k);
            maxSucCollsMatrix(cellid_1*length(cellIds)*(NoOfUplinks+1)+cellid_2*(NoOfUplinks+1)+k,:) = maxSucColls(:,k);
            cell_cell_Interference(cellid_1*(NoOfUplinks+1)+k,cellid_2+1) = sum(collisions(:,k));
        end
    end
end

% Saves the filename for the presentation GUI, viewSim.m
save('matfiles\simFile', 'filename')

% Loading/defining variables needed to present the necessary information
% in viewSim.m (GUI).
if fhAlgorithm ≠ 1
    load('matfiles\calcHS')
else
    pairs = [4 5 6 7 1 2 3];
    noOfSubbands = 7;
    widthOfSubbands = 11;
    sbMatrix = [2403:2413; 2414:2424; 2425:2435; 2436:2446; 2447:2457; 2458:2468; 2469:2479];
end

% Saves the simulation data and necessary variables in the filename
% specified by the user.
save(filename, 'pairs', 'fhAlgorithm','noOfSubbands', 'widthOfSubbands', 'fhAlgText', 'WLANchText', 'WLAN_BWText', 'P_UPLINK', 'Period', 'cellIds', 'collisionMatrix', 'maxSucCollsMatrix', 'cell_cell_Interference', 'sbMatrix')
```

A.2 Function remapWISAfreqs.m

```
function WISA_remapped = remapWISAfreqs(WLAN_channel, WLAN_BW)
% Returns the set of allowed WISA frequencies which is a subset of the 79 original
% freqs.
    WLAN_centerfreqs = 2412:5:2472;
    WISA_centerfreqs = 2403:1:2479;

    % No WLAN channels selected
    if (WLAN_channel == 0)
       WISA_remapped = WISA_centerfreqs;
       WLAN_usedFreqs = [];
    else
        % widht of WLAN sideband
        WLAN_SB = floor(WLAN_BW/2);
        WLAN_usedFreqs = 0;

        % For all selected WLAN channels, the used frquencies are added to a matrirx.
        for i = 1:length(WLAN_channel)
            WLAN_cf = WLAN_centerfreqs(WLAN_channel(i));
            WLAN_usedFreqs = [WLAN_usedFreqs, (WLAN_cf-WLAN_SB):1:(WLAN_cf+WLAN_SB)];
        end

        % Remapping the set of allowed WISA freguencies.
        remapLength = length(WISA_centerfreqs)-length(WLAN_usedFreqs);
        WISA_remapped_temp = zeros(1,length(WISA_centerfreqs));
        counter = 0;
        for i = 1:length(WISA_centerfreqs)
            if(¬isempty(find(WLAN_usedFreqs == WISA_centerfreqs(i))))
            else
                counter = counter+1;
                WISA_remapped_temp(counter)=WISA_centerfreqs(i);
            end
        end
        WISA_remapped = WISA_remapped_temp(1:counter);
    end
return
```

A.3 Function computeHoppingSequences.m

```
function cellIds = computeHoppingSequences(WISA_allowedfreqs)
    global fhAlgorithm
    global clicked
    global selected
    global Period

    [seqLengths, startIndexes] = calcConnSeqs(WISA_allowedfreqs
        );

    noOfAvFreqs = sum(seqLengths);

    primeNr = [2 3 2 5 7 2 3 11 13 2 17 19 23 5 3 29 31 2 37 41
         43 47 7 53 59 61 2 67 71 73 79];
    powers = [1 1 2 1 1 3 2 1 1 4 1 1 1 2 3 1 1 5 1 1 1 1 2 1 1
         1 6 1 1 1 1];
    allNumbers = primeNr.^powers;

    relNumbers = allNumbers(5:11);

    sbCalc = zeros(1, length(relNumbers));
    NoOfSbInSeq = zeros(length(seqLengths), length(relNumbers))
        ;

% ################################################
% Calculate nr of subbands given a subband width.
% ################################################
    for i = 1:length(seqLengths)
        NoOfSbInSeq(i,:) = floor(seqLengths(i)./relNumbers);
        sbCalc = sbCalc + NoOfSbInSeq(i,:);
    end
    possSB = sbCalc;

% ################################################
% Calulate the number of realizable subbands.
% ################################################

    % Allocate subbands in the set of alloed WISA frequencies
    for i = 1:length(sbCalc)
       sbMatrix = zeros(sbCalc(i),relNumbers(i));
       counter = 1;
       for j = 1:length(seqLengths)
           if NoOfSbInSeq(j,i) ≠ 0
               for k = 1:NoOfSbInSeq(j,i)
                   if startIndexes(j)>1
                       shift = seqLengths(j)−NoOfSbInSeq(j,i)*
                           relNumbers(i);
                   else
                       shift = 0;
                   end
```

```
                    sbMatrix(counter,:) = WISA_allowedfreqs((startIndexes(j)+shift+(k−1)*relNumbers(i)):(startIndexes(j)+shift+k*relNumbers(i)−1));
                    counter = counter+1;
                end
            end
        end

        counter=0;
        % Check minimum distance between subband.
        sbDist = zeros(sbCalc(i),sbCalc(i));
        for j = 1:sbCalc(i)−1
            for k = j+1:sbCalc(i)
                sbDist(j,k) = min(sbMatrix(k,:))−max(sbMatrix(j,:));
                sbDist(k,j) = sbDist(j,k);
            end
        end

        % Find subband pairs with at least 22 MHz spacing.
        sbAllowed=sbDist≥22;

        % Find number of realizable downlik−uplink paris.
        pairs = zeros(1,sbCalc(i));
        subbands = 1:sbCalc(i);
        level = 1;
        maxConst = 0;
        [pairs, maxConst] = pairsComp_v2(pairs, maxConst, level, subbands, sbAllowed, pairs, pairs);
        possSB(i) = maxConst;
    end

% ################################################
% Remap to nearest(lower) prime or prime extension
% ################################################
    for i = 1:length(possSB)
        if possSB(i) > 1
            possSB(i) = allNumbers(find(allNumbers≤possSB(i),1,'last'));
        end
    end

% ################################################
% User interface to choose wanted constellation
% ################################################
    usedFreqsMatrix = relNumbers.*possSB;
    save('matfiles\constellation','usedFreqsMatrix', 'relNumbers', 'possSB')
    clicked = 0;
    interactConst
    pause(3)
    while ¬clicked
        pause(1)
    end

    noOfSubbands = possSB(selected);
    widthOfSubbands = relNumbers(selected);

    % Calculate hopping sequences for the selected configuration
    subbandHS = galoisCalc(noOfSubbands);
    indexHS = galoisCalc(widthOfSubbands);

% ################################################
% Redoing previous calculations for selected
% configuration of subbands to save the data
% for viewSim.m
% ################################################
    i = selected;
    sbMatrix = zeros(sbCalc(i),relNumbers(i));
    counter = 1;
    for j = 1:length(seqLengths)
        if NoOfSbInSeq(j,i) ≠ 0
            for k = 1:NoOfSbInSeq(j,i)
                if j>1
                       shift = seqLengths(j)-NoOfSbInSeq(j,i)*relNumbers(i);
                   else
                       shift = 0;
                   end
                   sbMatrix(counter,:) = WISA_allowedfreqs((startIndexes(j)+shift+(k-1)*relNumbers(i)):(startIndexes(j)+shift+k*relNumbers(i)-1));
                   counter = counter+1;
            end
        end
    end
    counter=0;

    sbDist = zeros(sbCalc(i),sbCalc(i));
    for j = 1:sbCalc(i)-1
        for k = j+1:sbCalc(i)
            sbDist(j,k) = min(sbMatrix(k,:))-max(sbMatrix(j,:));
            sbDist(k,j) = sbDist(j,k);
        end
    end

    sbAllowed=sbDist≥22;
    pairs = zeros(1,sbCalc(i));
    subbands = 1:sbCalc(i);
    level = 1;
    maxConst = 0;
    [pairs, maxConst] = pairsComp_v2(pairs, maxConst, level, subbands, sbAllowed, pairs, pairs);
    if maxConst>0
```

```
        for i = 1:length(pairs)
            if sbAllowed(i,pairs(i)) == 0
                pairs(i) = 0;
            end
        end
    end

    availableSubbands = sbCalc(selected);

% ################################################
% Save variables to file
% ################################################
    cellIds = 0:size(subbandHS,1)*size(indexHS,1)-1;
    Period = size(subbandHS, 2)*size(indexHS,2);
    save( 'matfiles\calcHS', 'availableSubbands', 'noOfSubbands', 'widthOfSubbands', 'sbMatrix', 'pairs', 'subbandHS', 'indexHS');
```

A.3.1 Function calcConnSeqs.m

```
function [seqLengths, startIndexes] = calcConnSeqs(WISA_allowedfreqs)
    % This function calulates contiguous frequency bands
    % in the set of allowed WISA frequencies. The script
    % will return the length of such sequences and their
    % startindeices in the WISA_allowedfreqs vector.

    nextFreq = WISA_allowedfreqs(1)+1;
    seq_counter = 1;
    connection_counter = 1;
    tempIndexList = zeros(1, length(WISA_allowedfreqs));
    tempSeqList = zeros(1, length(WISA_allowedfreqs));
    tempIndexList(1) = 1;
    startIndex = 1;
    for i=2:length(WISA_allowedfreqs)
        if (nextFreq == WISA_allowedfreqs(i))
            connection_counter = connection_counter + 1;
            nextFreq = nextFreq + 1;
            if (i == length(WISA_allowedfreqs))
                tempIndexList(seq_counter) = startIndex;
                tempSeqList(seq_counter) = connection_counter;
            end
        else
            tempIndexList(seq_counter) = startIndex;
            startIndex = i;
            tempSeqList(seq_counter) = connection_counter;
            connection_counter = 1;
            seq_counter = seq_counter + 1;
            nextFreq = WISA_allowedfreqs(i) + 1;
            if (i == length(WISA_allowedfreqs))
                tempIndexList(seq_counter) = startIndex;
                tempSeqList(seq_counter) = connection_counter;
            end
        end

    end
    startIndexes = tempIndexList(1:seq_counter);
    seqLengths = tempSeqList(1:seq_counter);
```

A.3.2 Function galoisCalc.m

```matlab
function hs = galoisCalc(inPrime)
    % Calulates one-coincidence hopping sequences.

    % All relevant primes, or prime extension.
    primeNr = [2 3 2 5 7 2 3 11 13 2 17 19 23 5 3 29 31 2 37 41 43 47 7 53 59 61 2 67 71 73 79];
    powers = [1 1 2 1 1 3 2 1 1 4 1 1 1 2 3 1 1 5 1 1 1 1 2 1 1 1 6 1 1 1 1];
    allNumbers = primeNr.^powers;

    index = find(allNumbers==inPrime);

    p=primeNr(index);
    m=powers(index);

    % GF(P^N) calculations. Uses gftuple and gfadd from Communications Toolbox
    if m>1
        tuple = gftuple([0:p^m-2]',m,p);
        matrisa = zeros(p^m-1, p^m-1);
        matrisa(1,:) = bi2de(tuple,p,'right-msb');
        for i = 2:p^m
            for j = 1:p^m-1
                matrisa(i,j) = bi2de(gfadd(tuple(j,:), tuple(i-1,:),p),p,'right-msb');
            end
        end
    % GF(P) calculations. Uses gfmul from Communications Toolbox
    else
        row = 0:p-1;
        table = ones(p,1)*row;
        matrisaTemp = gfmul(table,table',p);
        matrisa = matrisaTemp(2:end,:);
    end

    hs = matrisa;
return
```

A.3.3 Function pairsComp_v2.m

```
function [pairs, maxConst]= pairsComp_v2(pairs, maxConst, level
    , subbands, sbAllowed, pairs_temp, pairs_value)
    % A brute—force algorithm using its own function
    % to check all possible combinations of downlink—
    % uplink pairs to ensure the maximum number of
    % realizable subbands is found for the given
    % subband allocation.
    localLevel = level;
    level = localLevel+1;
    localSubbands = subbands;
    if ¬isempty(localSubbands)
        for i = localSubbands
            pairs_temp(localLevel) = i;
            pairs_value(localLevel) = sbAllowed(localLevel, i);
            subbands = localSubbands(find(localSubbands ≠ i));
            [pairs, maxConst] = pairsComp_v2(pairs, maxConst,
                level, subbands, sbAllowed, pairs_temp,
                pairs_value);
        end
    else
        if maxConst<sum(pairs_value)
            pairs = pairs_temp;
            maxConst = sum(pairs_value);
        end
    end
```

A.4 Function cellHSAllocation.m

```
function [sbHS, indexHS] = cellHSAllocation(cellid, hsFile)
    % Assigns the correct subband— and index—hopping
    % sequence from the cellID and filename where
    % the HS sequences aresaved.
    load(hsFile);
    size(indexHS,1);
    floor(cellid/size(indexHS,1));
    mod(cellid,size(indexHS,1));
    sbHS = subbandHS(floor(cellid/size(indexHS,1))+1, :);
    indexHS = indexHS(mod(cellid,size(indexHS,1))+1, :);
return
```

A.5 Function secuenceAllocation.m

```
function [cell1_sequence, cell2_sequence] = sequenceAllocation(cell1sbHS, cell1indexHS, cell2sbHS, cell2indexHS)
    % Function for loading the cells hopping sequences.
    % Returns the sequences in matrix with four rows.
    % Row 1: DL, Row 2-5: UL0-UL3.

    global NoOfUplinks
    global Period
    global fhAlgorithm

    cell1_sequence = zeros(NoOfUplinks+1,Period);
    cell2_sequence = zeros(NoOfUplinks+1,Period);
    HSLength = length(cell1sbHS);
    IndexLength = length(cell1indexHS);

    % Hopping sequence for existing FH02 algorithm.
    if fhAlgorithm == 1
        for FN = 0:Period-1
            %DL
            cell1_sequence(1,FN+1) = cell1sbHS(mod(FN,HSLength)+1)*IndexLength + cell1indexHS(mod(FN,IndexLength)+1);
            cell2_sequence(1,FN+1) = cell2sbHS(mod(FN,HSLength)+1)*IndexLength + cell2indexHS(mod(FN,IndexLength)+1);

            % UL0-UL3
            if NoOfUplinks > 0
                for i = 1:NoOfUplinks
                    cell1_sequence(i+1,FN+1) = cell1sbHS(mod(FN+3,HSLength)+1)*IndexLength + cell1indexHS(mod(FN+3*(i-1),IndexLength)+1);
                    cell2_sequence(i+1,FN+1) = cell2sbHS(mod(FN+3,HSLength)+1)*IndexLength + cell2indexHS(mod(FN+3*(i-1),IndexLength)+1);
                end
            end
        end

    % Hopping sequence for new FH07-1 algorithm
    elseif fhAlgorithm == 2
        load('matfiles\calcHS')
        temp = find(pairs≠0);
        dlSubbands = temp(1:noOfSubbands);
        ulSubbands = pairs(dlSubbands);

        for FN = 0:Period-1
            % DL
            cell1_sequence(1,FN+1) = sbMatrix(dlSubbands(
                cell1sbHS(mod(FN,HSLength)+1)+1), cell1indexHS(
                mod(FN,IndexLength)+1)+1);
            cell2_sequence(1,FN+1) = sbMatrix(dlSubbands(
                cell2sbHS(mod(FN,HSLength)+1)+1), cell2indexHS(
                mod(FN,IndexLength)+1)+1);

            % UL0-UL3. Uses different distance between uplink
                groups depending on the width of the subband.
            if NoOfUplinks > 0
                for i = 1:NoOfUplinks
                    if widthOfSubbands == 7
                        cell1_sequence(i+1,FN+1) = sbMatrix(
                            ulSubbands(cell1sbHS(mod(FN,
                            HSLength)+1)+1), (i-1)*2+1);
                        cell2_sequence(i+1,FN+1) = sbMatrix(
                            ulSubbands(cell2sbHS(mod(FN,
                            HSLength)+1)+1), (i-1)*2+1);
                    else
                        if widthOfSubbands == 8 |
                            widthOfSubbands == 9
                           ulSpread = 2;
                        else
                           ulSpread = 3;
                        end
                        cell1_sequence(i+1,FN+1) = sbMatrix(
                            ulSubbands(cell1sbHS(mod(FN,
                            HSLength)+1)+1), cell1indexHS(mod(
                            FN+ulSpread*(i-1),IndexLength)+1)
                            +1);
                        cell2_sequence(i+1,FN+1) = sbMatrix(
                            ulSubbands(cell2sbHS(mod(FN,
                            HSLength)+1)+1), cell2indexHS(mod(
                            FN+ulSpread*(i-1),IndexLength)+1)
                            +1);
                    end
                end
            end
        end
    end
return
```

A.6 Function collisionCounter_v2.m

```matlab
function [collisions, maxSucColls] = collisionCounter_v2(cell1_frequencies, cell2_frequencies)

    global P_UPLINK
    global NoOfUplinks
    global Period

    collisions = zeros(1,NoOfUplinks+1);
    maxSucColls = zeros(1,NoOfUplinks+1);

    for i = 1:NoOfUplinks+1
        sucColls = zeros(1,Period);
        for j = 1:NoOfUplinks+1
            % Calculate where the collisions are
            collisionsIndicator = (cell1_frequencies(i,:) == cell2_frequencies(j,:));

            % Collisions from an interfering downlink
            if j==1
                collisions(i) = collisions(i)+sum(collisionsIndicator);

            % Collisions from an interfering uplink. Weighted with the probability of the interfering uplink being used.
            else
                collisions(i) = collisions(i)+sum(collisionsIndicator)*P_UPLINK;
            end

            % Marking all collision whether its from a downlink or an uplink
            if (P_UPLINK > 0) | (j == 1)
                sucColls = sucColls | collisionsIndicator;
            end
        end

        % Calculation of the maximum number of consecutive collisions.
        sucColls = int8(sucColls);
        I = find(collisionsIndicator==0,1,'first');
        % No zeors found. All transmissions have been interfered
        if isempty(I)
            maxSucColls(i) = maxSucColls(i) + Period;

        % Checking the maximum number of consecutive collisions, also for end to start of the sequence.
        else
            for k = 2:Period
```

```
                sucColls(k) = sucColls(k-1)*sucColls(k)+
                    sucColls(k);
            end
            if I ≠ 1
                sucColls(I-1) = sucColls(end)+sucColls(I-1);
            end
            maxSucColls(i) = max(sucColls);
        end
    end
return
```

Bibliography

[1] Jan Endresen, Anne E Vallestad, and Harald Vefling. WISA profile spesification. Technical report, ABB Corporate Research, Spe 2005, V1.03.

[2] Anne E Vallestad and Harald Vefling. WPS basestation RF board documentation. Technical report, ABB Corporate Research, September 2005, V1.11.

[3] Jan Endresen Richard Steigmann. Introduction to WISA. Technical Report 2.0, ABB, July 2006.

[4] Dacfey Dzung, Christoffer Apneseth, Jan Endresen, Jimmy Kjellson, Anne E Vallestad, and Harald Vefling. Air interface spesification. Technical report, ABB Corporate Research, September 2005, V1.11.

[5] Dacfey Dzung. Generation of new WISA FH sequences. Technical report, ABB Corporate Research, March 2007, V2.

[6] A.A. Shaar and P.A. Davies. A survey of one-coincidence sequences for frequency-hopped spread-spectrum systems. *IEE Proceedings*, December 1984.

[7] Li Bin. One-coincidence sequences with specified distance between adjacent symbols for frequency-hopping multiple access. *IEE Transactions on communication*, April 1997.