Delft University of Technology
Master's Thesis in Embedded Systems

Sensor-Assisted Movement Identification and Prediction for Beamformed 60 GHz Links

Arjan Doff

embedded software

TU Delft Delft University of Technology

Sensor-Assisted Movement Identification and Prediction for Beamformed 60 GHz Links

Master's Thesis in Embedded Systems

Embedded Software Group
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands

Arjan Doff
a.w.doff@student.tudelft.nl

19th January 2015

Author
Arjan Doff (a.w.doff@student.tudelft.nl)
Title
Sensor-Assisted Movement Identification and Prediction for Beamformed 60 GHz Links
MSc presentation
26th January 2015

Graduation Committee

prof. dr. ir. K. Langendoen (chair)	Delft University of Technology
dr. ir. R. Venkatesha Prasad	Delft University of Technology
ir. K. Chandra	Delft University of Technology
dr. ir. H. Wang	Delft University of Technology

Abstract

The 60 GHz frequency band promises very high data rates – in the order of Gb/s – due to the availability of large amounts of bandwidth. High free-space path loss at the 60 GHz frequency band makes it necessary to employ beamforming capable directional antennas to confine signal power in the desired direction. When beamforming is used, the links are sensitive to misalignment in antenna directionality, due to the movement of devices. To identify and circumvent the misalignments, we propose to use motion sensors (i.e., accelerometer and gyroscope) which are already present in most modern mobile devices. By finding the extent of misaligned beams, corrective actions are carried out to reconfigure the antennas. Motion sensors in mobile devices provide means to estimate the extent of misalignments. We collected real data from motion sensors and steered the beams appropriately. The results from our study show that the sensors are capable of detecting the cause of the error as translational or rotational movements. Furthermore it is also shown that sensor data can be used to predict the next location of the user. This can be used to reconfigure the directional antenna to switch the antenna beam and hence avoid frequent link disruptions. This decreases the number of beam searches, thus lowering the MAC overhead.

Preface

During my Embedded Systems master I always tried to achieve a broad understanding in many topics related to embedded systems. My graduation project has been no different, it was yet another unexplored topic; wireless communications. This past year I worked not only on wireless communications, but also learnt the ups and downs of being on the forefront of academic research. The result of which is not only this M.Sc. thesis, but also two published papers at the IEEE Consumer Communications and Networking Conference (CCNC).

First of all I would like to thank my daily supervisor Kishor Chandra for his encouragements and reassurances throughout this year. He was always willing to help and give me the extra nudge to keep learning and exploring. My thanks also go out to RangaRao Venkatesha Prasad (VP) who gave me insights into problem solutions I otherwise could not have solved. Furthermore I would like to thank Koen Langendoen for hosting me at the Embedded Software group and Huijuan Wang for being a member of my committee. And finally, I would like to thank my friends and family who have supported me through and through.

Arjan Doff

Delft, The Netherlands
19th January 2015

Contents

Glossary

A-BFT association beamforming training. 7

ACK acknowledgement. 8

AGC automatic gain control. 6

AP access point. 3, 5, 7, 8, 15, 16, 27, 29–33, 36, 37, 41, 44

AT announcement time. 7

AWV antenna weight vector. 8, 9

BC beam combining. 9–11

BI beacon interval. 7

BRP beam refinement protocol. 7–10

BTI beacon time interval. 7

CBAP contention based access period. 7

CE channel estimation. 5

CSMA/CA carrier sense multiple access with collision avoidance. 7

DTI data transmission interval. 7

FCC federal communications commission. 1

HMM hidden markov model. 19–26, 31, 43

HPBW half-power beamwidth. 16, 32

ISS initiator sector sweep. 8

k-NN k-nearest neighbors. 19, 20, 22, 24–26, 31, 34, 39, 40, 43

LOS line of sight. 2

MAC medium access control. 2, 5–7, 43

MCS modulation and coding scheme. 6, 31

MID multiple sector ID detection. 9, 10

MIDC multiple sector ID detection capture. 8, 9

MM markov model. 19, 20

NLOS non line of sight. 2, 3, 35, 44

OFDM orthogonal frequency division multiplexing. 6

PBSS personal basic service set. 5, 7

PCP PBSS central point. 5, 7, 8

PHY physical. 2, 5, 6, 9, 10, 43

RPS Radiowave Propagation Simulator. 31–34, 38

RSS responder sector sweep. 8

RWPM random waypoint model. 30, 31, 34–37, 39, 40

RX receive. 3, 4, 7–9, 31, 34

SC single carrier. 6, 31

SINR signal-to-interference-plus-noise ratio. 3, 20

SLS sector level sweep. 7–10

SP service period. 7

SSW sector sweep. 8

STA station. 3, 5, 7, 8, 11, 15, 16, 27, 29–31, 34, 41

STF short training field. 5

TDMA time division multiple access. 7

TX transmit. 3, 4, 7–9, 31–34

WLAN wireless local area network. 1, 2, 5, 7

WPAN wireless personal area network. 2

Chapter 1

Introduction

1.1 Motivation

The unprecedented increase of mobile devices together with a rapid emergence of bandwidth hungry applications require wireless communication technologies which can provide data rates in the order of multi-Gb/s. Presently, Wi-Fi (IEEE 802.11) operating at 2.4/5 GHz is the most favored communication technology for wireless local area networks (WLANs). There have been many efforts such as higher order modulation schemes (64 QAM, 256 QAM, etc.), MIMO, MU-MIMO, channel bonding, frame aggregation, etc. at PHY and MAC layers to enhance the data rates of Wi-Fi. However, the limited bandwidth present in the 2.4/5 GHz frequency band is not sufficient to provide the desired multi-Gb/s connectivity. Large unused bandwidths are available in the millimeter-wave band, which ranges from 30 to 300 GHz. Specifically a lot of interest has been shown in the 60 GHz band for short-range communications after the federal communications commission (FCC) has allocated 4-9 GHz in the 57-64 GHz band for unlicensed use [11,22,33,34], as shown in Figure 1.1.

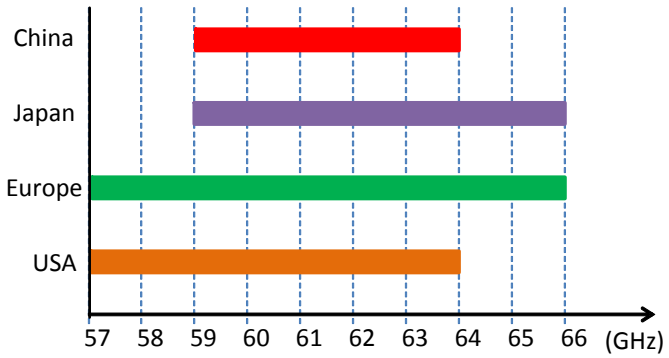


Figure 1.1: The unlicensed bands in the 60 GHz frequency range.

The IEEE organization has come up with IEEE 802.15.3c and IEEE 802.11ad – 60 GHz standards, to define medium access control (MAC) and physical (PHY) specifications for short range communications in the 60 GHz band. IEEE 802.15.3c [1] published in 2009 targets wireless personal area network (WPAN) applications and specifies a mandatory data rate of 1.6 Gbps and maximum data rate of 4.6 Gbps at a distance of 10-20 m. In 2012, IEEE has proposed IEEE 802.11ad [2] as a WLAN alternative at the 60 GHz frequency band which is back compatible with the popular IEEE 802.11b/g/n/ac. IEEE 802.11ad promises data rates up to 6.7 Gbps.

1.2 Challenges

The physical properties of 60 GHz propagation significantly differs from 2.4/5 GHz signal propagation, throwing several challenges. A major challenge in the 60 GHz band is the very high free-space path loss following from Friis transmission equation [12]:

$$P_r = P_t G_r G_t \left(\frac{\lambda}{4\pi R}\right)^2, \tag{1.1}$$

where P_r, P_t, G_t, G_r, λ and R are the received/transmitted power, receiver/transmit antenna gain, wavelength and distance between the antennas, respectively. Using $\lambda = \frac{c}{f}$, it can easily be seen that the attenuation at the 60 GHz band is 22 dB higher than at the 5 GHz band. In addition to free-space path losses, the 60 GHz band is also subject to oxygen losses with a peak value of about 10 dB/km. However current 60 GHz standards are mainly focused on indoor usage [1, 2], thus oxygen losses are negligible at shorter ranges. Even though the high attenuation is a major challenge, consequently it offers less interference between devices. To compensate for the additional path loss at 60 GHz, directional antennas are used to confine signal power in the desired direction. Use of directional antennas also offers excellent spatial reuse. Another major challenge is the necessity of line of sight (LOS) paths due to the inability to diffract around obstacles and penetrate through walls [14,27,28]. In indoor environments, presence of furniture, walls and humans can greatly affect the 60 GHz links. The loss due to human shadowing is around 20-30 dB [13]. To solve this issue, intelligent beamforming antennas are needed which can select alternative non line of sight (NLOS) paths, in case the LOS path is obstructed.

1.3 Problem statement

As already stated, to compensate the high free-space path loss at 60 GHz, narrow-beamwidth directional antennas are used. Highly directional antennas help to improve link quality and extend the communication range. IEEE

802.15.3c and IEEE 802.11ad provide multi-level beamforming protocols to setup directional links. However, in case of device movement, it can be very difficult to maintain the desired link quality if narrow-beamwidth directional antennas are used. A slight displacement in device position/orientation can result in link misalignment. This thesis focuses on the challenge of device movement in 60 GHz networks, which can limit the maximum achievable link quality [3, 9, 18, 20]. Different causes are attributed to link degradation due to device movement:

- **Linear motion**: The first cause is linear motion, also referred to as a translational movement, which usually happens when the user walks with the device.
- **Angular motion**: Secondly, angular movement is caused when the user is turning around or changing the orientation of the mobile device.
- **Blockage**: Lastly, blockage occurs when the links are interrupted by other users or obstacles while the user is moving around. In this case NLOS paths have to be used.

It is usually difficult to determine the cause of link degradation. In [32] the rate at which the received signal power changes is monitored to identify the error type affecting the link. It is important to note that the solution to reclaim the lost communication link because of one of these errors is not compatible with the other errors. Each error requires different compensation. For example, in a traditional setup if the device is moving in a linear direction, the station (STA)/access point (AP) does not know what causes the link degradation. This means it might switch to a NLOS beam-pair, which has a worse signal-to-interference-plus-noise ratio (SINR). Hence it is important to correctly identify the communication errors before proceeding to solve them.

1.4 Contributions

This work proposes to use the motion sensors, which are already embedded into most modern devices, to help with the identification of the error. When the device employs its sensors, it can estimate using simple classification techniques if the error is caused by rotation, translation or beam blockage. Once the error is identified, it is possible to go a step further and also try to resolve this error. This means that we can predict the next beam-pair and shift to them before the link quality drops significantly. If the link quality drops too much, the beamformed link has to be re-established. To re-establish the communication link, transmit (TX) and receive (RX) stations have to restart searching for the best beam-pair; we refer to this as *re-beamforming*. However, with the help of sensor data, device movement

can be predicted and appropriate antenna weight vectors can be calculated to maintain a stable link. So the overheads of a beam searching procedure can be avoided. Thus this work mainly solves two problems: (i) determining the error that caused the link degradation using activity recognition; and (ii) predict the next beam-pair such that the link quality remains stable while minimizing the number of re-beamforming procedures.

An overview of these objectives is shown in Figure 1.2. The sensor data

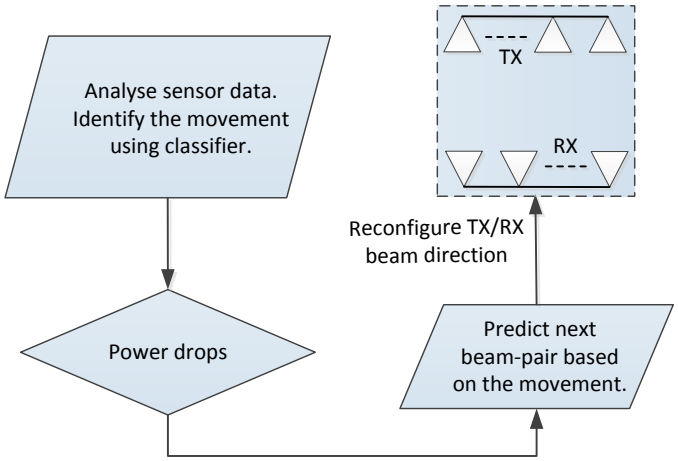


Figure 1.2: Flowchart of the error identification and prediction phases.

is analyzed to identify the user's movement by means of a classifier. When the received signal strength drops by a certain power threshold, the next beam-pair is predicted based on the movement. Consequently the TX/RX antenna weights are adapted such that an improved link is established.

1.5 Thesis organisation

The rest of the thesis is organised as follows. In Chapter 2 we describe the architecture of the IEEE 802.11ad standard to position relevance of our work. Following that, in Chapter 3 we discuss the used sensors and generalize major movements that occur. In Chapter 4 the activity recognition phase is explained and results are shown. This is followed by the movement prediction phase and its results in Chapter 5. Finally we conclude with general remarks, limitations and future work in Chapter 6.

Chapter 2

IEEE 802.11ad

As stated in Chapter 1, IEEE 802.11ad standard provides MAC and PHY specifications for 60 GHz WLANs. It aims to achieve data rates up to 6.7 Gbps needed for applications, such as wireless HD displays, 3D gaming, and rapid upload/download. IEEE 802.11ad is back compatible to IEEE 802.11 series, and in absence of 60 GHz link, fallback options to 2.4/5 GHz bands are provided.

Similar to the IEEE 802.11b/g/n basic service set, IEEE 802.11ad uses a personal basic service set (PBSS), which is the operating area of IEEE 802.11ad networks. To provide basic timing to the STAs, one STA assumes the role of AP/PBSS central point (PCP), as shown in Figure 2.1.

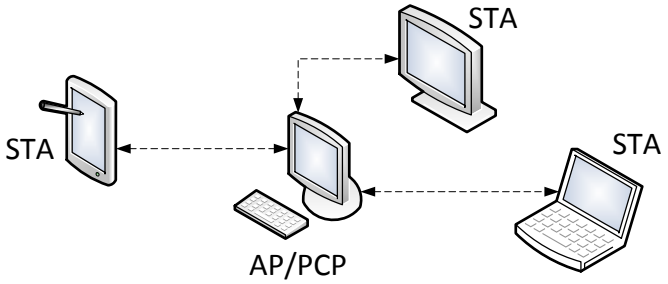


Figure 2.1: An example of the IEEE 802.11ad WLAN architecture.

This chapter will explain in detail how the IEEE 802.11ad standard operates. First the PHY and MAC layers will be discussed. After that the focus will be put on the beamforming process and the beam training duration.

2.1 WLAN PHY layer

A packet in the PHY layer has a structure as shown in Figure 2.2. The first frames are the short training field (STF) and channel estimation (CE)

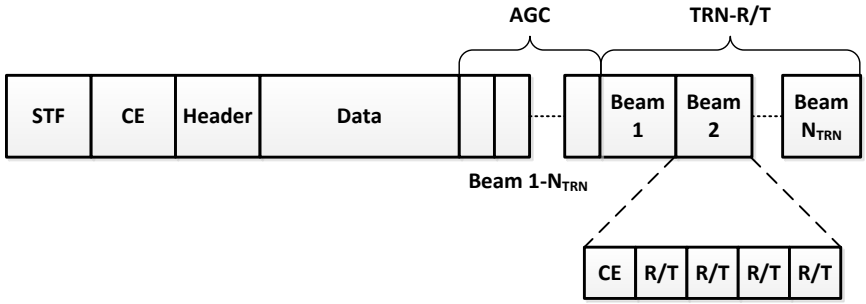


Figure 2.2: IEEE 802.11ad PHY in-packet training structure [2].

fields. They help among others with signal acquisition and synchronization [26]. The header contains general information about the packet, such as the modulation and coding scheme (MCS), the size of the packet and also if optional training fields are appended. The data field contains the MAC header and MAC data. The beam training fields for both receiver and transmitter (TRN-R/T) can be appended to the data packet as part of the in-packet beamforming training. The automatic gain control (AGC) fields are added to account for the variation in signal strength when transmitting and receiving beam training fields.

The PHY layer supports three modulation methods: (i) control modulation, (ii) single carrier (SC) modulation + low-power SC modulation and (iii) orthogonal frequency division multiplexing (OFDM) modulation. These modulation methods are summarized in Table 2.1.

Table 2.1: Modulation and coding schemes for IEEE 802.11ad [2].

Modulation method	MCS index	Modulation	Data rate (Mbps)
Control PHY	0	DBPSK	27.5
SC PHY	1-12	$\pi/2$-BPSK, $\pi/2$-QPSK, $\pi/2$-16QAM	385 - 4620
OFDM PHY	13-24	SQPSK, QPSK, 16-QAM, 64-QAM	693 - 6757
Low-power SC PHY	25-31	$\pi/2$-BPSK, $\pi/2$-QPSK	626 - 2503

Each modulation method has its own purpose. The control PHY is used for basic communication before a high resolution beamformed link is established between devices. SC PHY allows for lower receive sensitivity compared to OFDM PHY, where MCSs 1 to 4 are mandatory to ensure interoperability. OFDM PHY offers the highest data rate of 6.757 Gbps with MCS 24 using 64-QAM. The low-power SC PHY is optional and provides lower

processing power requirements.

2.2 WLAN MAC layer

The access methods used in WLAN comprises of both carrier sense multiple access with collision avoidance (CSMA/CA) and time division multiple access (TDMA) [2, 8]. A frame in WLANs is referred to as a beacon interval (BI), the structure of which can be seen in Figure 2.3. The BI consists of

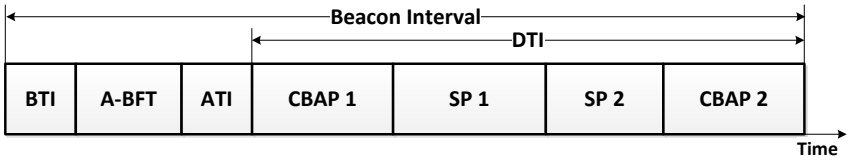


Figure 2.3: A superframe of the MAC layer of the IEEE 802.11ad protocol.

different access periods. The first period is the beacon time interval (BTI), in which the AP/PCP transmits one or more beacons in different directions. STAs wanting to join the PBSS, can be trained in the association beamforming training (A-BFT) stage of the BI. During announcement time (AT) the AP/PCP can transmit management information to the STAs in a request/response fashion. The main data transmission interval is the data transmission interval (DTI), in which two periods are present. The contention based access period (CBAP) and service period (SP) allows any frame exchange, including data transmissions. It is possible to use any combination in the number and order of SPs and CBAPs in the DTI.

Thanks to the directional nature of the antennas, 60 GHz communication provides excellent spatial reuse capability. Thus it is possible to employ spatial sharing, where different stations use different SPs at the same time. For example, STA A and B can have a scheduled SP at the same time as a SP between STA C and D, as directional links will not interfere with each other.

2.3 Beamforming training

IEEE 802.11ad uses a multi-level beamforming scheme similar to that of IEEE 802.15.3c [2]. The IEEE 802.11ad beamforming process consists of two phases. The first phase is the sector level sweep (SLS). Its purpose is to allow communications between two STAs. The SLS is followed by the beam refinement protocol (BRP). In general the BRP is used to further train RX and TX antennas of a STA after the SLS phase, in a request/response based manner. An example of a beamforming training sequence can be seen in Figure 2.4.

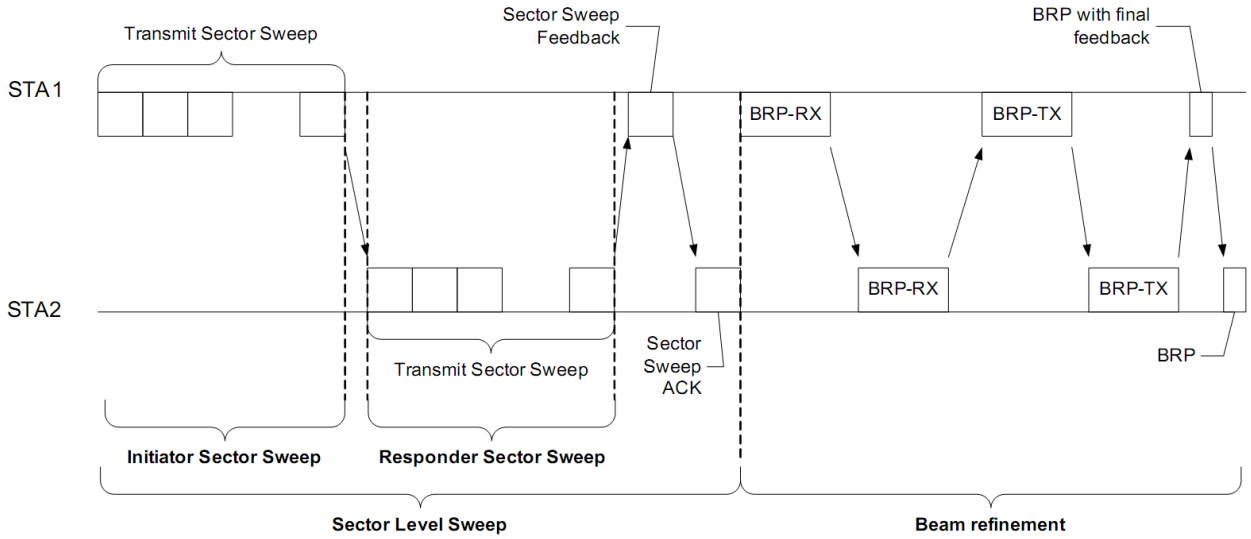


Figure 2.4: An example of beamforming training in IEEE 802.11ad [2].

The STA requesting beamforming is referred to as the initiator, while the receiving STA is referred to as the responder. In order to track the beams/channel a beam tracking phase can be used after the SLS/BRP.

2.3.1 SLS

The SLS is started by the initiator. Its purpose is to allow communications between two STAs. In turn, the SLS can consist of as many as 4 subphases. The first is the initiator sector sweep (ISS), to train the initiator link. This is followed by the responder sector sweep (RSS), to subsequently train the responder link. The last two subphases are the sector sweep (SSW) feedback and the SSW acknowledgement (ACK).

During the ISS an initial connection is made between STA and AP/PCP. One of them receives with a quasi-omni directional antenna, while the other transmits a sequence of training frames covering different TX sectors. The roles are reversed in the RSS, after at least one sector sweep frame is successfully received. The optimal antenna weight vector (AWV) of both stations are exchanged during the SSW feedback and SSW ACK.

2.3.2 BRP

In general the BRP is used to further train a STA's RX and TX antennas after the SLS phase. This phase is a request/response based process. The STA requesting beamforming training is referred to as the initiator, while the receiving STA is referred to as the responder. The BRP phase consists of a BRP setup subphase, and a multiple sector ID detection capture (MIDC)

phase. In turn the MIDC phase can either consist of a multiple sector ID detection (MID) subphase and beam combining (BC) subphase combined, or only the MID subphase.

The BRP setup subphase is used to exchange information regarding the next subphases. During the MID subphase, a quasi-omni TX beam is used while the receiver sweeps its beams to determine the set of RX AWVs with the best link quality. The MID subphase is followed by the BC subphase. The BC subphase tests the best B RX AWVs together with the best B TX AWVs, to determine the best beam-pair. The IEEE 802.11ad standard suggest a maximum number of $B = 7$ AWVs to be tested in the BC subphase.

2.4 Training duration

This section will try to get a grasp for the training duration involved in the beamforming process. Since there are many variations in the sequence of possible beam training phases, we assume the following generic sequence. First a full SLS is performed (both initiator and responder), followed by a BRP setup subphase and a full MID phase, and finally a full BC phase. All the phases in our sequence are performed using Control PHY.

We first take a look at the time required to send a packet, with and without beamtraining data appended. Combining this with the beam training sequence, an approximate time required to establish a beamformed link is calculated.

Training duration for a single packet

The minimal time required to send one packet using Control PHY without any additional training fields is given by T_{PNT} (packet no training):

$$T_{PNT} = T_{STF} + T_{CE} + T_{Header} + T_{Data}. \tag{2.1}$$

Using Figure 2.2 the time for a single in-packet beamtraining packet T_{IPT} (in-packet training) can be defined as follows:

$$\begin{aligned} T_{IPT} &= T_{PNT} + (\overbrace{4 \times 320}^{\text{AGC}} + \overbrace{1152}^{\text{CE}} + \overbrace{4 \times 640}^{\text{R/T}}) \times N_{TRN} \times T_C \\ &= T_{PNT} + 4992 \times N_{TRN} \times T_C, \end{aligned} \tag{2.2}$$

where N_{TRN} is the amount of AWVs that need to be trained in a packet.

The time parameters for Control PHY are listed in Table 2.2.

Different stages involved in training

Let N_{QO}^I and N_{QO}^R be the number of quasi-omni beams at the initiator (transmitter) and responder (receiver) respectively. And let N_b^I and N_b^R be

Parameter	Description	Value
T_{STF}	Short training field duration	$3.636\,\mu$s
T_{CE}	Channel estimation duration	$0.656\,\mu$s
T_C	Chip time	0.57 ns
T_{Header}	Header duration	$1.6051\,\mu$s
T_{Data}	Minimal data duration	$7.2960\,\mu$s
SBIFS	Short beamforming interframe spacing	$1\,\mu$s
BRPIFS	Beam refinement protocol interframe spacing	$40\,\mu$s
MBIFS	Medium beamforming interframe spacing	$3\,\mu$s
LBIFS	Long beamforming interframe spacing	$6\,\mu$s

Table 2.2: IEEE 802.11ad Control PHY time parameters.

the number of beam sectors available at the initiator and responder respectively. During the SLS the antenna setting should remain constant for the duration of the entire packet, thus no in-packet training is possible during the SLS phase. Using this, the training duration for the SLS becomes:

$$\begin{aligned} T_{SLS} &= T_{ISS} + \text{MBIFS} + T_{RSS} + \text{MBIFS} \\ &\quad + T_{SSW-FB} + \text{MBIFS} + T_{SSW-ACK}, && (2.3) \\ T_{ISS} &= T_{PNT} \times N_b^I \times N_{QO}^R + \text{SBIFS} \times (N_b^I - 1) \\ &\quad + \text{LBIFS} \times (N_{QO}^R - 1), && (2.4) \\ T_{RSS} &= T_{PNT} \times N_b^R \times N_{QO}^I + \text{SBIFS} \times (N_b^R - 1) \\ &\quad + \text{LBIFS} \times (N_{QO}^I - 1), && (2.5) \\ T_{SSW-FB} &= T_{SSW-ACK} = T_{PNT}. && (2.6) \end{aligned}$$

After the SLS phase, the BRP setup subphase is performed which consists of a single packet sending the information for the subsequent MID and BC phase.

$$T_{BRP-setup} = T_{PNT}. \tag{2.7}$$

Next is the MID phase, where in-packet training is possible:

$$\begin{aligned} T_{MID} &= T_{I-MID} + \text{BRPIFS} + T_{R-MID} + \text{BRPIFS} \\ &\quad + T_{I-MID-FB} + \text{BRPIFS} + T_{R-MID-FB}, && (2.8) \\ T_{I-MID} &= (T_{PNT} + 4992 \times N_b^R \times T_C) \times N_{QO}^I \\ &\quad + \text{SBIFS} \times (N_{QO}^I - 1), && (2.9) \\ T_{R-MID} &= (T_{PNT} + 4992 \times N_b^I \times T_C) \times N_{QO}^R \\ &\quad + \text{SBIFS} \times (N_{QO}^R - 1), && (2.10) \\ T_{I-MID-FB} &= T_{R-MID-FB} = T_{PNT}. && (2.11) \end{aligned}$$

And finally the BC phase, where in-packet training is also used:

$$\begin{aligned} T_{BC} &= T_{I-BC} + \text{BRPIFS} + T_{R-BC} + \text{BRPIFS} \\ &\quad + T_{I-BC-FB} + \text{BRPIFS} + T_{R-BC-FB}, && (2.12) \\ T_{I-BC} &= (T_{PNT} + 4992 \times B \times T_C) \times B \\ &\quad + \text{SBIFS} \times (B-1), && (2.13) \\ T_{R-BC} &= (T_{PNT} + 4992 \times B \times T_C) \times B \\ &\quad + \text{SBIFS} \times (B-1), && (2.14) \\ T_{I-BC-FB} &= T_{R-BC-FB} = T_{PNT}. && (2.15) \end{aligned}$$

Combining all this results in the following beamforming training time:

$$\begin{aligned} T_{training} &= T_{SLS} + \text{SBIFS} + T_{BRP-setup} + \text{BRPIFS} \\ &\quad + T_{MID} + \text{SBIFS} + T_{BC}. && (2.16) \end{aligned}$$

If we assume $N_b^I = N_b^R = 64$, $N_{QO}^I = N_{QO}^R = 4$ and $B = 5$ then the beamforming process for both the initiator and the responder takes $T_{training} = 9.2\,\text{ms}$. If we furthermore assume that 4 STAs are present in the network and re-beamforming needs to be done every 100 ms; which is a real possibility if rotations are involved as explained in Chapter 3. Then in this situation throughput is decreased by 37%, which otherwise could have been spent on transmitting data.

This chapter gave an overview of the IEEE 802.11ad standard, which uses the 60 GHz band to achieve data rates up to 6.7 Gbps necessary for wireless HD displays, 3D gaming, and rapid upload/download. The beamforming protocol and required beam training times were also explored and it was shown that throughput is quickly decreased by 37% if re-beamformings are not taken care of. The next chapter will explore what causes link degradation and what means are available to detect them.

Chapter 3

Sensors and movement

Narrow-beamwidth directional antennas are employed in 60 GHz communication to compensate for the high free-space path loss. However, device movement can result in frequent link outage due to an error in the alignment of transmit and receive antenna beams. To resolve this error and prevent a total loss of the communication link, the movement first has to be detected, which is possible with the use of motion sensors.

This chapter first discusses the possibilities of sensors embedded in modern devices. After that the possible movements and corresponding breakdown frequencies are explained.

3.1 Sensors

Smartphones have many sensors, among which accelerometers, gyroscopes and magnetometers can be seen as the most relevant to detect motion. Both the accelerometer and the gyroscope data are used in the error identification phase. Furthermore we make use of a sensor fused orientation sensor in the movement prediction phase. All the used sensors will be elaborated upon below.

Accelerometers

Accelerometers can be found in every smartphone. Thanks to accelerometers the earlier smartphones were able to distinguish between up and down and thus implement an auto-tilt feature. The accelerometer measures, as its name suggests, acceleration along the X, Y and Z axis:

$$\boldsymbol{a} = \begin{bmatrix} a_x \\ a_y \\ a_z. \end{bmatrix} \tag{3.1}$$

Note that gravitational accelerations are also measured by the accelerometer. This means that when the device is at rest $||\boldsymbol{a}|| \approx 9.81\,\mathrm{m/s^2}$, and when the

device is in free fall $||\boldsymbol{a}|| \approx 0\,\mathrm{m/s^2}$. The linear acceleration (without the force of gravity) can still be obtained by applying a high-pass filter to extract the gravitational component from the raw data. Apart from not directly being able to measure the linear acceleration, accelerometers are also very noisy. Usually this noise is diminished by applying a low-pass filter.

The actual accelerometers found in smartphones have quite a high sampling rate, which varies from 1 kHz to 2 kHz [25,29,31]. However the sampling rate is limited by the smartphone manufacturers to a maximum of 100-200 Hz depending on the smartphone model. This is most likely done to limit the power consumption.

Gyroscopes

Gyroscopes measure angular velocity based on the Coriolis force along the X, Y and Z axis:

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z. \end{bmatrix} \tag{3.2}$$

A challenge inherent to gyroscopes is a bias causing a linearly increasing error when the gyroscope data is integrated over time, called drift. The bias can be substantially reduced by using a calibration period in which the bias error is estimated and subtracted from later readings. This process will need to be repeated periodically such that the error remains small.

Gyroscopes also have quite a high sampling rate, varying from 800 Hz to 8 kHz depending on the gyroscope model [30, 31]. As is the case with the accelerometers, the smartphone manufacturers also limit the sample rate to 100-200 Hz.

Sensor fused orientation sensor

Sensor fusion is based on combining sensory data, such that the result is better than using the sensors individually. The result of sensor fusion is a virtual sensor, which combines one or more physical sensors by exploiting their strengths while trying to diminish their weaknesses. In [5] it is investigated how accelerometers, gyroscopes and magnetometers can be combined to obtain the orientation of the smartphone. The result is an orientation sensor which measures the orientation of the device relative to the East-North-Up coordinates, and is represented as

$$\boldsymbol{\epsilon} = \begin{bmatrix} \alpha \\ \beta \\ \psi \end{bmatrix} \tag{3.3}$$

Here α, β and ψ denote the azimuth angle, elevation angle and skew angle of the device, respectively. For simplicity, it is assumed that the device is

held horizontally in the hands of the user. This means only the azimuth angle α will be used in the movement prediction phase to determine the direction of movement. This is of course an abstraction of reality where the user may hold the device in a tilted manner. There are methods that deal with this problem such that an estimate of the direction of the user can still be found [24], however these methods were not implemented here.

Specifications

A summary of the sensor specifications from the used Samsung Galaxy SIII is shown in Table 3.1 [31]. Since the orientation sensor is sensor fused, it's

Table 3.1: Accelerometer, gyroscope and orientation sensor specifications [31].

Sensor	Sample rate	Scale	Sensitivity
Accelerometer	100 Hz	$\pm 2/\pm 4/\pm 8$ g	$\pm 1/\pm 2/\pm 4/\pm 7.8$ mg/digit
Gyroscope	200 Hz	$\pm 250/\ \pm 500/$ ± 2000 dps	8.75/17.50/70 mdps/digit
Orientation	200 Hz	-	-

scale and sensitivity are not known and only the sampling rate is known through experimentation.

3.2 Translational movement

We assume the beams of both the STA and the AP are aligned after a successful beamforming procedure. When the STA starts to move in a linear direction, both the STA and the AP need to change beam direction, as can be seen in Figure 3.1. This means that if the STA starts to move, it needs

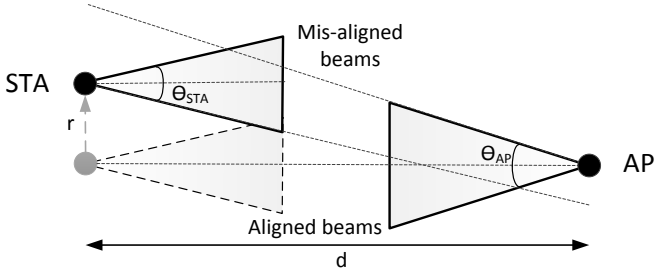


Figure 3.1: A movement in a linear direction, also called translational movement.

to notify the AP such that both devices know that their beam-pairs need to

be realigned when the signal power drops.

The distance r after which the link begins to deteriorate can be calculated as follows [18]:

$$r = d \tan\left(\frac{\Theta_{min}}{2}\right), \tag{3.4}$$

where d is the distance between the STA and the AP, and $\Theta_{min} = \min(\Theta_{STA}, \Theta_{AP})$. Θ_{STA} and Θ_{AP} are the half-power beamwidth (HPBW) of the STA and AP, respectively. If the user is moving in a linear direction as shown in Figure 3.1, the time after which the beam needs to be switched is calculated with:

$$\tau_{tran} = \frac{r}{v}, \tag{3.5}$$

where v is the velocity of the STA. As an example; if the stations are $d = 3\,\text{m}$ apart, the STA is moving at $v = 1.5\,\text{m/s}$ and $\Theta_{min} = 30°$, the beams will need to switch beams after $\tau_{tran} = 536\,\text{ms}$.

3.3 Rotational movement

When the STA experiences a turn, only the beam direction of the STA needs to be changed, as can be seen in Figure 3.2. Furthermore, from Figure 3.2 it

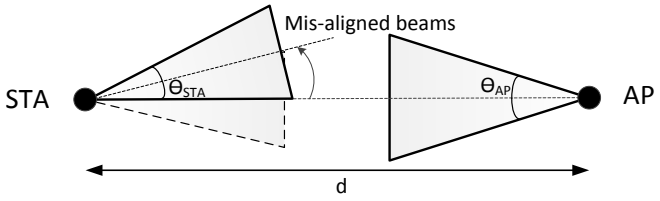


Figure 3.2: A rotational movement.

can quickly be seen that the STA should switch its beam after the following time:

$$\tau_{rot} = \frac{\Theta_{STA}}{2\omega}. \tag{3.6}$$

Here ω denotes the angular speed of the device. An interesting thing to note is that τ_{rot} is not dependent on the distance between the STA and the AP.

To get an understanding of how often beam switching needs to be done, angular speeds were measured [32]. Table 3.2 shows the necessity of performing frequent re-beamforming due to rotations if no additional information is known. This was also explored and shown in [18] where beam switching needed to be done every 14-54 ms during rotations, depending on the antenna setup.

Thus if the user is rotating the device at $\omega = 800°/\text{s}$ and $\Theta_{STA} = 30°$, the STA beam should switch after $\tau_{rot} = 18.7\,\text{ms}$. Comparing this to the

Table 3.2: Angular displacement of a smartphone under different activities [32].

Activities	Angular displacement in 100 ms
Reading, web browsing (no change of orientations)	$6° - 11°$
Reading, web browsing (horizontal from/to vertical orientation changes)	$30° - 36°$
Playing games	$72° - 80°$

example switching time of $\tau_{tran} = 536$ ms for translational movement, it is clear that timing restrictions due to rotations are a lot more severe than timing restrictions due to translations.

This chapter describes the available sensors; accelerometers, gyroscopes and orientation sensors. Both the accelerometer and gyroscope can be used to detect motion, while the orientation sensor gives an indication of the movement direction. Angular and translational movements were discussed and it was shown that rotations can cause faster link degradation than translations. In the next two chapters we discuss and show the results for our two objectives. The first objective is to see if we can accurately identify what caused the error: translation, rotation or blockage. The second objective consists of checking whether or not we can accurately predict the next beam-pair when we know the origin of the error.

Chapter 4

Activity Recognition

As explained in the previous chapter, two major types of movement can occur: (i) translational and (ii) rotational movement. Both types of movement approach beam switching in different ways, which is why we need to differentiate between them. This can be done by performing activity recognition to identify whether a user is moving in a straight line or turning around.

In this chapter two activity recognition algorithms are explored. The first algorithm, k-nearest neighbors (k-NN) is a simple, yet very effective algorithm used in activity recognition [4, 23], which uses a feature vector to identify features specific to a certain activity. The second algorithm is an extension of the well-known markov model (MM) and is called a Hidden Markov Model [7, 21]. In a MM every state corresponds to a (physical) observation, where the next state is only dependent on the current state, this is often referred to as "memoryless". The hidden markov model (HMM) is different from the standard MM in one unique way; the observation is a probabilistic function of the state. Thus the actual (hidden) states are not known, only the observed states emitted at every hidden state are known. This allows for more complex and flexible models to be created.

4.1 k-NN

The k-NN algorithm is well known and can be described as follows [19]. Let k be the number of nearest neighbours and $\boldsymbol{T} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ be the training samples. $\boldsymbol{x}_i = (\boldsymbol{f}_i, c_i)$, where $\boldsymbol{f}_i$ is the feature vector of the training samples $\boldsymbol{x}_i$ and c_i is the class that $\boldsymbol{x}_i$ belongs to. The feature vector was chosen to contain the mean and standard deviation of $\boldsymbol{a}$ and $\boldsymbol{\omega}$, which have shown good results as reported in [6]. The maximal autocorrelation of $\boldsymbol{a}$ was also used as a feature to better detect the steps taken by a user. Resulting in $\boldsymbol{f} = [\mu_{\boldsymbol{a}}, \mu_{\boldsymbol{\omega}}, \sigma_{\boldsymbol{a}}, \sigma_{\boldsymbol{\omega}}, \max(R_{\boldsymbol{a}})]$.

A new sample $\hat{\boldsymbol{x}} = (\hat{\boldsymbol{f}}, \hat{c})$ is classified as shown in Algorithm 1. Multiple measures of distances can be used to calculate the distance $d(\hat{\boldsymbol{f}}, \boldsymbol{f}_i)$

Algorithm 1 k-NN

1: **for each** $\hat{\boldsymbol{x}} = (\hat{\boldsymbol{f}}, \hat{c})$ **do**
2: Calculate the distance $d(\hat{\boldsymbol{f}}, \boldsymbol{f}_i)$ between $\hat{\boldsymbol{x}}$ and all $\boldsymbol{x}_i$ in $\boldsymbol{T}$.
3: Sort $\boldsymbol{T}$ ascending based on the distance $d(\hat{\boldsymbol{f}}, \boldsymbol{f}_i)$.
4: Select the first k samples from $\boldsymbol{T}$, these are the k points closest to $\hat{\boldsymbol{x}}$.
5: Assign a class to $\hat{c}$ based on the majority vote of the k classes.
6: **end for**

between the feature vector of the input sample and the training samples. We chose to use the l^2 norm, where the distance d is calculated as $d(\hat{\boldsymbol{f}}, \boldsymbol{f}) = \sqrt{\sum_{j=1}^{n} |\hat{f}_j - f_j|^2}$. Here $n = 15$ is the length of the feature vector.

Blockage can also indirectly be detected using sensors. If the SINR drops and the device does not measure any movement, it is apparent that the beam was blocked. Note that this conclusion is only valid if we assume movement and blockage do not occur at the same time. If they do occur at the same time, it will be difficult to identify the source of the link disruption.

4.2 HMM

In a MM every state can directly be observed, whereas in a HMM the actual states are hidden. The only thing that can be observed are the observable states emitted at every hidden state, as shown in Figure 4.1.

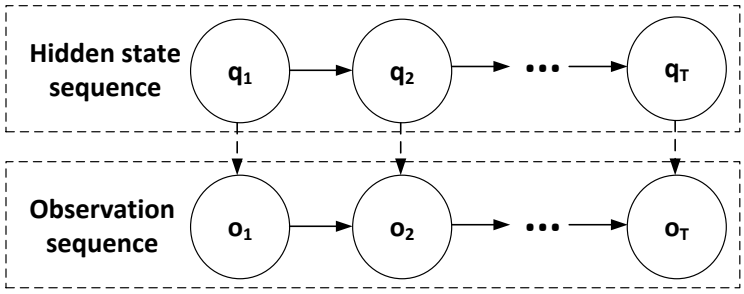


Figure 4.1: The hidden and observed states in the HMM.

The HMM can be characterized as follows:

- A set of N states: $S = \{s_1, s_2, \ldots, s_N\}$.
- A set of M observable states: $V = \{v_1, v_2, \ldots, v_M\}$.
- A sequence of hidden states: $Q = (q_1, q_2, \ldots, q_T)$, where the hidden state at time t is denoted as q_t.

- A sequence of observed states: $O = (o_1, o_2, \ldots, o_T)$, where the observed state at time t is denoted as o_t.
- The state transition matrix: $A = [a_{i,j}]$, where $a_{i,j}$ is the probability of a state transaction from state s_i to state s_j:

$$a_{i,j} = P(q_t = s_j | q_{t-1} = s_i), \quad 1 \leq i, j \leq N. \tag{4.1}$$

- The observation symbol probability distribution in state j and for observation k is $B = [b_j(k)]$ where $b_j(k)$ is

$$b_j(k) = P(o_t = v_k | q_t = s_j), \quad 1 \leq j \leq N, 1 \leq k \leq M. \tag{4.2}$$

- The initial state distribution: $\pi = [\pi_i]$ where π_i is

$$\pi_i = P(q_1 = s_i), \quad 1 \leq i \leq N. \tag{4.3}$$

The model parameters of a HMM are represented as the tuple $\lambda = (A, B, \pi)$.

4.2.1 Compute observation probability

The HMM based activity recognition uses multiple HMMs to determine which underlying activity has the highest probability of happening. To determine this probability the following question needs to be answered: "How do we efficiently compute the probability of an observation sequence given a model, $P(O|\lambda)$, where O is the observation sequence $O = o_1 o_2 \ldots o_T$ and λ is the model $\lambda = (A, B, \pi)$?". The probability of an observation sequence O for a given hidden state sequence Q is [7, 21]:

$$P(O|Q, \lambda) = \prod_{t=1}^{T} = P(o_t|q_t, \lambda) = b_{q_1}(o_1) b_{q_2}(o_2) \ldots b_{q_T}(o_T), \tag{4.4}$$

and the probability of the hidden state sequence is:

$$P(Q|\lambda) = \pi_{q_1} a_{q_1,q_2} a_{q_2,q_3} \ldots a_{q_{T-1},q_T}. \tag{4.5}$$

To calculate $P(O|\lambda)$ we multiply Equation (4.4) and Equation (4.5) to obtain:

$$P(O|\lambda) = \sum_{Q} P(O|Q, \lambda) P(Q|\lambda) \tag{4.6}$$

$$= \sum_{q_1, q_2, \ldots, q_T} \pi_{q_1} b_{q_1}(o_1) a_{q_1,q_2} b_{q_2}(o_2), \ldots a_{q_{T-1},q_T} b_{q_T}(o_T). \tag{4.7}$$

However, using this approach to calculate $P(O|\lambda)$ we would need in the order of $2T \cdot N^T$ calculations [21].

Thus to efficiently find $P(O|\lambda)$ the forward-backward algorithm is used. For the forward procedure, let us define the forward variable $\alpha_t(i)$ as the probability of the observation sequence $O = o_1 o_2 \dots o_t$ and state s_i at time t given the model λ:

$$\alpha_t(i) = P(o_1 o_2 \dots o_t, q_t = s_i|\lambda). \tag{4.8}$$

$\alpha_t(i)$ can be solved using the following algorithm:

1. Initilization:
$$\alpha_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N. \tag{4.9}$$

2. Induction:
$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{i,j}\right] b_j(o_{t+1}), \quad 1 \leq t \leq T-1, 1 \leq j \leq N. \tag{4.10}$$

3. Termination:
$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i). \tag{4.11}$$

Using this procedure the calculations are now in the order of N^2T, substantially less than $2T \cdot N^T$.

4.2.2 Activity recognition using HMM

This section covers how we apply HMMs for activity recognition, inspired by [16]. First we describe how the HMM is trained. After that it is shown how an activity is identified based on an observation sequence.

Training the HMM

To classify the data in different categories the model parameters $\lambda = (A, B, \pi)$ of the HMM need to be trained. One can use either supervised or unsupervised training. Supervised training assumes the training data is labeled such that the parameters can easily be estimated. For unsupervised training the training data is not labeled and the Baum-Welch algorithm can be used [21]. Since labeled training data is available (and necessary) for the k-NN algorithm we will focus on supervised training. The state transition matrix $A = [a_{i,j}]$ can be estimated as follows:

$$a_{i,j} = P(s_i|s_j) = \frac{\text{Count}(s_i, s_j)}{\text{Count}(s_i)}, \tag{4.12}$$

where $\text{Count}(s_i, s_j)$ is the number of transitions from s_i to s_j in the training data.

The observation matrix $B = [b_j(k)]$ follows from:

$$b_{j,k} = P(v_j|s_k) = \frac{\text{Count}(v_k, s_j)}{\text{Count}(s_j)}, \tag{4.13}$$

where $\text{Count}(v_k, s_j)$ is the number of observed emissions of v_k at state s_j in the training data.

And the initial state $\pi = [\pi_i]$:

$$\pi_i = P(q_1 = s_i) = \frac{\text{Count}(q_1 = s_i)}{\text{Count}(q_1)}. \tag{4.14}$$

The model parameters A, B and π, now completely define the HMM.

Identifying the observation sequences

An activity is classified as one of four activities: (i) standing still, (ii) turning, (iii) walking or (iv) both turning and walking. We use two sensors (accelerometer + gyroscope) that are capable of measuring along three axes, thus we have six different observation sequences. The six sensor-based HMMs are trained for each activity, according to the procedure described in Section 4.2.2, resulting in a total of 24 HMMs.

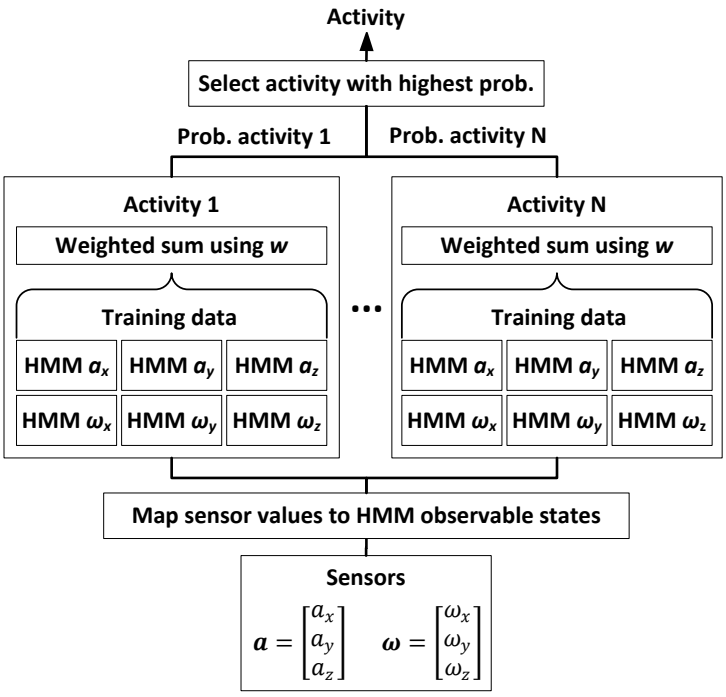


Figure 4.2: The HMM structure used for activity recognition.

To identify an observation sequence we use the procedure as shown in Figure 4.2. The input is a sequence of sensor values $\mathrm{a} = [\mathrm{a}_x, \mathrm{a}_y, \mathrm{a}_z]^T$ and $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$, which are observed for a certain duration.

The first step is to map these values to observable states for the HMMs to use. The HMMs in our setup use discrete hidden and observed states, thus to convert the sensor values to discrete symbols we round the raw sensor values. It is possible to model the observed symbols as Gaussian and thus create continuous states; however this increases the computational complexity of the setup. Also the discretization acts as a filter to limit small noise variations present in the sensor data.

The second step is to calculate the probability of the observed sequence for every activity. This means we need to calculate $P(O|\lambda)$, as described in Section 4.2.1. Here O is the mapped sequence of sensor values and λ are the model parameters of the trained HMMs. The probabilities are combined by using a weighted sum $\boldsymbol{w} = [w_{ax}, w_{ay}, w_{az}, w_{\omega x}, w_{\omega y}, w_{\omega z}]$.

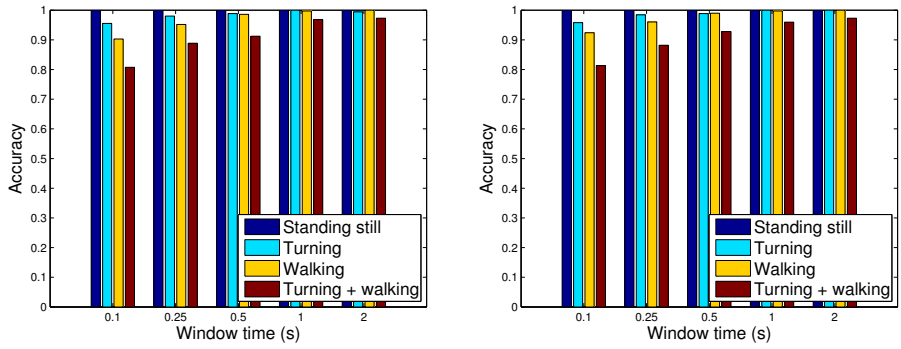
Finally, the identified activity is based on the highest probability coming from each individual activity. The different explicit features from the activities can be strengthened by assigning different weights $\boldsymbol{w}$ to the probabilities calculated from the HMMs, which will be shown in the next section.

4.3 Results

We perform activity recognition using the sensors in the devices, such that we can detect if the user is (i) standing still, (ii) turning, (iii) walking or (iv) both turning and walking. The activity recognition was done by sampling the sensors for 10 min at the highest sampling frequency, as shown in Table 3.1, while doing the above activities. The training data was gathered from the first 5 min of the recorded samples. Both activity recognition algorithms were run on the remaining 5 min to determine their accuracy.

4.3.1 k-NN results

The k-NN search was tested with k = 1 and k = 3. The accuracy of the k-NN search for different window sizes can be seen in Figure 4.3. A window size of 0.1 s means we are trying to detect the activity from the last 0.1 s. From Figure 4.3 it can be seen that there is almost no discernible difference between using k = 1 and k = 3. Both achieve an accuracy of 100% when the user is standing still, meaning we were always able to identify if the user was standing still. Furthermore, turning could be identified with an accuracy of 96-100%, and linear movement was detected with an accuracy of 92-100%. It can be observed that the combination of angular and linear movement was the hardest to detect with only 81% to 97% accuracy. As was also explored in [6] the window size has a big impact on the results. Basically there is a trade-off between accuracy and speed. When the window size decreases, the



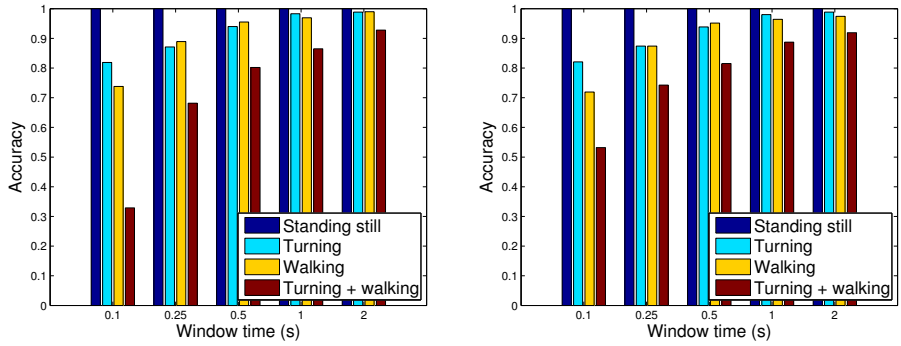
(a) (b)

Figure 4.3: The accuracy of the k-NN based search for different window sizes with (a) k = 1 and (b) k = 3.

activity will be recognized faster, however the results will be less accurate and vice versa.

4.3.2 HMM results

Using different weights the HMM based activity recognition algorithm also produced two sets of results, as shown in Figure 4.4. Figure 4.4a shows the accuracy of activity recognition with uniform weights $\boldsymbol{w} = [\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}]$. Next to that in Figure 4.4b the weights were changed to $\boldsymbol{w} = [\frac{1}{8}, \frac{1}{8}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}, \frac{1}{4}]$, such that more emphasis was put on the Z-axis of both sensors. The reasoning behind this is that specific features of the movement can best be seen along the Z-axis.



(a) (b)

Figure 4.4: The accuracy of the HMM based algorithm for different window sizes with (a) uniform weights and (b) higher weight on the Z-axis.

In both Figure 4.4b and Figure 4.4a standing still achieved an accuracy

of 100%. Also turning and walking individually give similar results for both set of weights - in the range of 70% to 99% accuracy. The real difference is observed when the user is both turning and walking at the same time. Especially for a short window time of 0.1 s the accuracy rises from 33% to 54% when higher weights are applied to the Z-axis, as compared to the uniform weights.

Comparing HMM to k-NN it can be seen that activity recognition using HMM does not perform nearly as well as k-NN. We suspect this is due to two factors. The first is that for the k-NN algorithm specific features were selected that could best identify each activity. Such as μ_a, σ_a for standing still, $\mu_a, \sigma_a, \max(R_a)$ for walking and $\mu_\omega, \sigma_\omega$ for turning. This inherently gives the k-NN algorithm an edge over HMM, where no features were selected and only raw sensor data was used as input for the model. The second factor comes from using the short window times. We suspect that short window times of 0.1 to 0.5 s are not sufficient for the HMM, which needs longer observation sequences to differentiate between the activities. This can also be seen from Figure 4.4 where the HMM algorithm performs almost, if not as well as k-NN with a window time of 2 s.

4.4 Improvements for activity recognition

From Table 3.2 we observe that angular movement can happen rapidly. This means that the window size of 0.1 s is not enough when the movement is faster. Table 3.1 shows that he accelerometer is sampling at roughly 100 Hz, which means 10 samples are taken every 0.1 s. To further increase the reaction time for activity recognition, the window size needs to be decreased further. If we strive for a reaction time of 0.01 s, consequently the window time should be decreased to 0.01 s. This means we would observe barely 1 sample in a window, which is not sufficient to recognize an activity. To overcome this limitation the sampling rate of the sensors needs to be increased. If the sampling rate is increased, faster and more accurate results can be obtained.

In this chapter two activity recognition algorithms were explored that use motion sensor data as input. The first algorithm uses k-NN, which is a simple algorithm focused on features specific for certain activities. The second algorithm uses multiple HMMs to determine which activity has the highest likelihood of happening. Results show that k-NN outperforms the HMMs, especially for short window times. The next chapter describes the second phase of this thesis, where the emphasis is on movement prediction.

Chapter 5

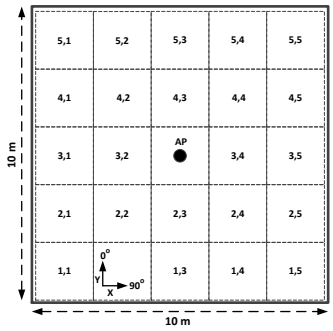
Movement prediction

In the previous chapter, we presented activity recognition as a means to identify the cause of link degradation, i.e., translational or rotational movement, in 60 GHz communication using directional antennas. The next logical step is to steer the AP beam in an appropriate direction such that the link quality can be maintained.

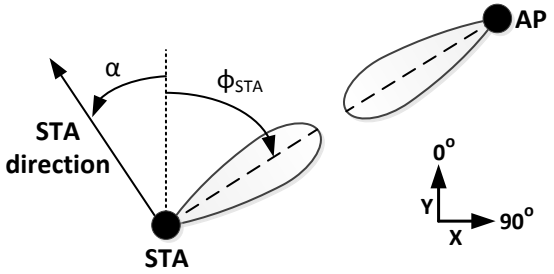
According to IEEE 802.11ad, if the directional link between a pair of devices is lost, the re-beamforming procedure is invoked, which is nothing else than an exhaustive search for the best beam-pairs. Re-beamforming consumes a significant amount of time to align the beam-pair, which reduces the channel usage for transmitting data, as shown in Section 2.4. In this approach we propose next location/angle prediction of devices based on the observed sensor data. Instead of invoking the re-beamforming procedure, the device can switch to the predicted antenna beam-pair, based on the activity of the user. If the predicted beams are correct the STA/AP are able to maintain the link quality. In case the predicted beam-pair is not correct and the link is lost, the re-beamforming procedure still needs to be performed. Thus our goal is to minimize the number of re-beamformings by switching to predicted beam-pairs.

We assume a 60 GHz AP is located on the ceiling of the room. The room in the test setup is divided into separate sectors, each with their own beam directed towards them, as shown in Figure 5.1a. The STA is also able to beam to the AP. To simplify the simulation only the azimuth plane is considered for the STA beam, as shown in Figure 5.1b. This is a simple 2D case, where only clockwise and anti-clockwise rotations occur, thus the beam elevation angle of the STA θ_{STA} will not be considered.

This chapter will first explain the prediction methods that will be evaluated in Section 5.1. After this the test setup and the movement prediction results will be discussed in Section 5.2.



(a) The 5 × 5 sector grid in the room.



(b) Top-down view which shows the azimuth beam angle ϕ_{STA} and the direction α of the device.

Figure 5.1: The (a) room and (b) directional beams.

5.1 Prediction methods

Apart from using no prediction to switch the beam pairs, two prediction methods are investigated: (i) simple prediction and (ii) sensor prediction. All three methods are activated only when the received power drops by a certain threshold P_{Dth} – the drop-off threshold. A forced re-beamforming is done if the received power reaches the re-beamforming threshold P_{Rth}, as shown in Figure 5.2.

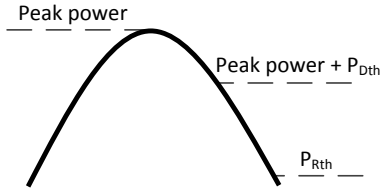


Figure 5.2: A power curve showing the drop-off threshold P_{Dth} and re-beamforming threshold P_{Rth}.

Using no prediction is easily explained; every time the signal power drops by P_{Dth}, re-beamforming is performed. This method is the baseline where no intelligence is used.

The other two prediction methods do require some sort of intelligence and will be explained below. Both methods make use of the signed angle distance between two angles $\Delta(a, b)$, expressed in degrees:

$$\Delta(a, b) = \text{mod}(a - b + 180, 360) - 180, \tag{5.1}$$

where $\text{mod}(m, n)$, a and b are the modulo operation, target angle and source/starting angle, respectively.

5.1.1 Simple prediction

This prediction method is based on extrapolating the next beam sector from the previous beam sector. This is done differently for both the AP and the STA, both of which will be explained below. There is no distinction between rotations or translations, since the simple prediction method is unable to differentiate between the movements.

AP beam prediction

Using Figure 5.1a, if we take $\boldsymbol{S_c} = [x_c, y_c]^T$ as the current sector and $\boldsymbol{S_p} = [x_p, y_p]^T$ as the previous sector. The AP predicts the next sector $\boldsymbol{S_n}$ as follows:

$$\boldsymbol{S_n} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x_c + \text{sgn}(x_c - x_p) \\ y_c + \text{sgn}(y_c - y_p) \end{bmatrix},$$

where sgn$(*)$ is the signum operator. For example, if the user moves from beam sector $[x_p, y_p]^T = [3, 1]^T$ to $[x_c, y_c]^T = [3, 2]^T$, the next beam sector is predicted to be $[x_n, y_n]^T = [3, 3]^T$. This method works well if the user is always walking in one direction, however if turns are made this method wrongly predicts the next beam-pair.

STA beam prediction

The STA predicts the next beam based on the previous and the current beam as follows:

$$\phi_{STA}(k+1) = \phi_{STA}(k) + \text{sgn}(\Delta(\phi_{STA}(k), \phi_{STA}(k-1)))\phi_{STA\,as}, \quad (5.2)$$

where $\phi_{STA}(k)$ is the beam direction of the STA at time k, and $\phi_{STA\,as}$ is the angle spacing between two adjacent beam directions of the STA.

5.1.2 Sensor prediction

Besides differentiating the beam switching prediction for the AP and the STA, the sensor prediction also separates between rotational and translational movement. Before the beam prediction is done, the movement is first identified using the sensors as explained in Chapter 4. The prediction method employs the orientation sensor of the device. It is assumed that the azimuth angle α from this sensor can be used as an indication of the direction of the user.

AP beam prediction

Rotation prediction: Beam predictions for the AP in case of device rotations are not applicable as seen in Chapter 3. In this case only the STA needs to reorientate its beam in the direction of the AP.

Translation prediction: If we again take $\boldsymbol{S_c} = [x_c, y_c]^T$ as the current sector. Then the AP predicts the next beam sector $\boldsymbol{S_n}$ in the case of translations as follows:

$$\boldsymbol{S_n} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x_c + \text{nint}(\sin\alpha) \\ y_c + \text{nint}(\cos\alpha) \end{bmatrix}, \tag{5.3}$$

where nint$(*)$ is the nearest integer, or round function and α represents the azimuth angle of the device orientation.

STA beam prediction

Rotation prediction: If the user is turning, the next beam of the STA can be found with:

$$\phi_{STA}(k+1) = \phi_{STA}(k) + \text{sgn}(\Delta(\alpha(k), \alpha(k-1)))\phi_{STA\,as}. \tag{5.4}$$

Note that this equation is very similar to that of simple prediction; however we are now using the direction of the device to predict the next beam, instead of the previous beams.

Translation prediction: If the user is walking, the next beam of the STA is predicted with:

$$\phi_{STA}(k+1) = \phi_{STA}(k) + \text{sgn}(\Delta(\phi_{STA}(k), \alpha(k)))\phi_{STA\,as}. \tag{5.5}$$

The function $\text{sgn}(\Delta(\phi_{STA}(k), \alpha(k)))$ should evaluate to 0 in the special case where $\Delta(\phi_{STA}(k), \alpha(k)) = 0^\circ$ or $\Delta(\phi_{STA}(k), \alpha(k)) = -180^\circ$.

5.2 Evaluation

To evaluate the 60 GHz network with mobility we first need to create a test setup, as described in Section 5.2.1. Secondly, we evaluate the different prediction methods by exploring the statistical properties in simulation using a random waypoint model (RWPM), the results of which are shown in Section 5.2.2. Lastly, we also try to verify the simulations by illustrating what happens along a single route, recorded with real sensor data. These routes and their results are shown in Section 5.2.3.

In both the RWPM simulation and the single route verification we explore two scenarios.

Scenario 1 The first scenario is where the user is standing/sitting in a location in the room and rotating their device at the same time. Since the only activity the user (STA) will be performing are turns, the AP does not need to change beams. This scenario will often happen when users operate their device.

Scenario 2 The second scenario is less common where the user is walking around in the room. In this scenario both translations and rotations occur, and thus this scenario has the extra challenge of identifying the activity, since different activities require different beam switching strategies. To identify the activity the k-NN algorithm was selected instead of the HMM algorithm since it achieved better performance, as shown in Chapter 4.

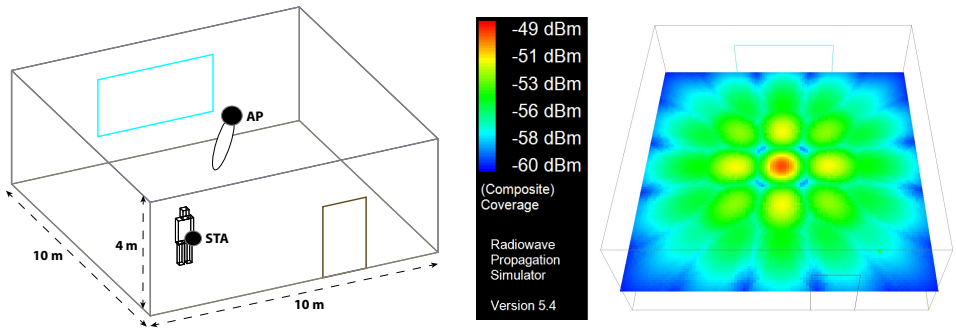
In both scenarios the re-beamforming threshold P_{Rth} is set to -70 dBm, since the receiver sensitivity for the most robust SC PHY MCS is -68 dBm according to the IEEE 802.11ad [2] standard.

5.2.1 Test setup

The test setup is divided into two parts which are combined to create a simulation environment. The first part consists of creating the room, the TX antennas (AP) and a measurement grid. This environment was created using a verified radio frequency propagation simulator, called Radiowave Propagation Simulator (RPS) [10]. RPS provides close to real 60 GHz signal strength at various locations on a floor plan. The second part happens in Matlab, which collects the signal strength data from RPS. In Matlab a directional RX (STA) antenna is added, and the RWPM and single route are created for scenario 1 and 2. Both parts will be explained in more detail below.

RPS setup

In RPS a room is created as shown in Figure 5.3a, with material properties of glass, concrete and wood as shown in Table 5.1.



(a) (b)

Figure 5.3: The test setup in the room, where (a) shows the RPS simulation environment and (b) the received power.

Table 5.1: Dielectric properties of materials in 60 GHz [13].

Material	Thickness (m)	ϵ_{Re}	ϵ_{Im}
Concrete wall/ceiling	0.3	6.14	-0.3015
Wooden floor	0.2	2.81	-0.0964
Wooden door	0.04	2.81	-0.0964
Glass window	0.02	4.58	-0.0458

A 100×100 grid is placed at a height of 1.5 m to simulate possible positions of users when holding a mobile device. RPS is able to measure the signal power at every position in this grid. A directional TX antenna is placed in the middle of the room at a height of 4 m, which will act as the AP. This antenna can be directed at one of 25 different sectors, such that a 5×5 grid is created as shown in Figures 5.1a and 5.3b. The TX antenna pattern is Gaussian shaped, as is also assumed in the IEEE 802.11ad task group channel model [17], and represented by:

$$G(\theta, \phi) = G_0 \exp(-\gamma\theta^2), \tag{5.6}$$

where G_0, θ and ϕ are the maximum antenna gain, elevation and azimuth angle of the beams, respectively. The coefficient γ is determined by the HPBW as follows:

$$\gamma = \frac{4\ln(2)}{\text{HPBW}}. \tag{5.7}$$

We chose a HPBW of 30° for the TX (AP) antenna, such that some overlap is present among the different sectors.

The antenna gain is calculated using the directivity of the antenna [15]:

$$G_0 = kD, \tag{5.8}$$

where k is the efficiency factor and D the directivity. When the HPBW is known, the directivity can be approximated by [15]:

$$D = \frac{40000}{\Theta_{az}\Theta_{el}}, \tag{5.9}$$

where Θ_{az} and Θ_{el} are the HPBW in the azimuth and elevation plane, respectively. Thus for $\Theta_{az} = 30°$ and $\Theta_{el} = 30°$, the antenna directivity is $D = 16.48$ dB. The antenna efficiency k is chosen such that the resulting maximum antenna gain is $G_0 = 16$ dB.

In order to beamform to the 25 different sectors with center coordinates $[x, y]$, the TX beams and corresponding sectors were all given an azimuth angle ϕ and elevation angle θ. These angles were obtained using the following

simple geometric formula:

$$\phi_{x,y} = \text{arctan2}\left(x - x_{AP}, y - y_{AP}\right) \tag{5.10}$$

$$\theta_{x,y} = \text{arctan2}\left(h, \sqrt{(x - x_{AP})^2 + (y - y_{AP})^2}\right) \tag{5.11}$$

where $h = 2.5\,\text{m}$ is the height difference between AP and user, $x_{AP} = 5\,\text{m}$, $y_{AP} = 5\,\text{m}$ are the x and y coordinate of the AP. The x and y coordinates represent the center of the sector, where x and y both range from 1 m to 9 m in steps of 2 m. arctan2(y, x) is the four-quadrant inverse tangent function, which gives the angle between the positive x-axis of a plane and the point given by the coordinates $[x,\ y]$ on it. The angle is positive for counter-clockwise angles (upper half-plane, $y > 0$), and negative for clockwise angles (lower half-plane, $y < 0$).

Using Equations (5.10) and (5.11) the TX antenna beams were assigned beaming directions ϕ and θ as depicted in Table 5.2.

Table 5.2: Angles used in the RPS test setup where each cell denotes the $[\phi, \theta]$ of that sector with center coordinates $[x,y]$.

9	−45°, 24°	−27°, 29°	0°, 32°	27°, 29°	45°, 24°
7	−63°, 29°	−45°, 41°	0°, 51°	45°, 41°	63°, 29°
5	−90°, 32°	−90°, 51°	0°, 90°	90°, 51°	90°, 32°
3	−117°, 29°	−135°, 41°	180°, 51°	135°, 41°	117°, 29°
1	−135°, 24°	−153°, 29°	180°, 32°	153°, 29°	135°, 24°
y (m) / x (m)	1	3	5	7	9

The discussed parameters and others for the RPS setup are shown in Table 5.3.

Table 5.3: RPS parameters.

Room dimensions	$10 \times 10 \times 4\,\text{m}$
Carrier frequency	60 GHz
TX Θ_{az}	30°
TX Θ_{el}	30°
TX power	10 dBm
TX antenna gain (G_0)	16 dB
Noise figure	10 dB
Antenna polarization	Left hand circular

Matlab setup

In Matlab the signal strength values from the RPS grid are collected and used for further simulations. To simulate the directionality of the RX (STA) antenna we didn't use RPS, instead Matlab was used. The same Gaussian antenna pattern is used as for the TX antenna in RPS. To simplify our analysis the RX antenna beams are only able to beam in a 2D plane, which are uniformly spaced by $\phi_{STA\,as}$. This means $\Theta_{az} = 30°$ and $\Theta_{el} = 180°$, which (using Equation (5.9)) results in $D = 8.70\,\text{dB}$. Again the antenna efficiency k is chosen such that the maximum RX antenna gain is $G_0 = 8\,\text{dB}$. These and other parameters for the Matlab setup are shown in Table 5.4.

Table 5.4: Matlab parameters.

RX Θ_{az}	30°
RX Θ_{el}	180°
RX antenna gain	8 dB
$\phi_{STA\,as}$	22.5°
Scenario 1 location	$[x, y] = [2, 2]$ m
Single route P_{Dth}	-3 dBm
P_{Rth}	-70 dBm

5.2.2 RWPM simulation results

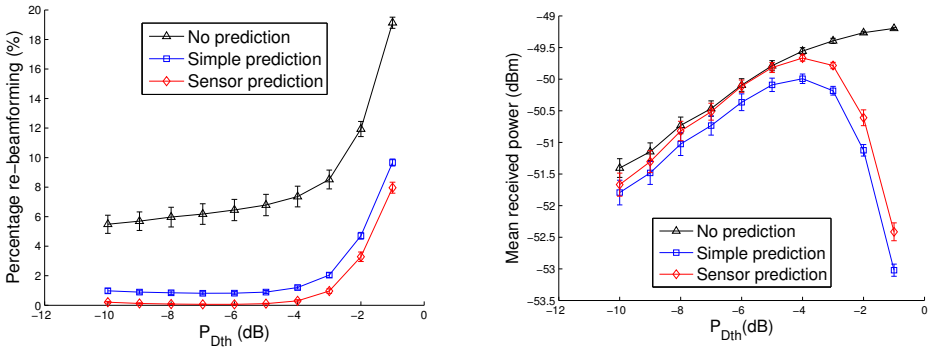
We use a RWPM such that we can recreate multiple instances and compare multiple types of movement prediction statistically. It is not possible to match real sensor data to a random waypoint model, thus in the tests with the RWPM the activity and the orientation of the user are assumed to be known. However as we've seen in Chapter 4 the accuracy is not 100%, thus the recognition accuracy obtained from using k-NN with $k = 3$ and a window time of 0.1 s is applied in the RWPM. Furthermore, the orientation data is also not perfect in the real world, thus an orientation error is added in the simulated RWPM orientation data. The error was set to be zero mean Gaussian with a standard deviation of 5°, such as to simulate sensor and environmental noise.

Scenario 1: stationary and turning

The waypoints in this scenario were 100 randomly generated angles to which the device turns. Turning to these waypoints was done at random speeds in the range of 6° to 80° per 100 ms. These values are taken from Table 3.2 to represent different user activities.

For different drop-off thresholds the re-beamforming percentage and the mean received power were calculated along the route. We use the total

number of points along the angular route to normalize the number of re-beamformings, such that a re-beamforming percentage is obtained. This was run 100 times to obtain a statistically significant mean and standard deviation, as shown in Figure 5.4.



(a) (b)

Figure 5.4: RWPM of scenario 1 *without* blockage effects and $P_{Rth} = -70\,\text{dBm}$. Different drop-off thresholds are plotted against (a) the re-beamforming percentage and (b) the mean received power.

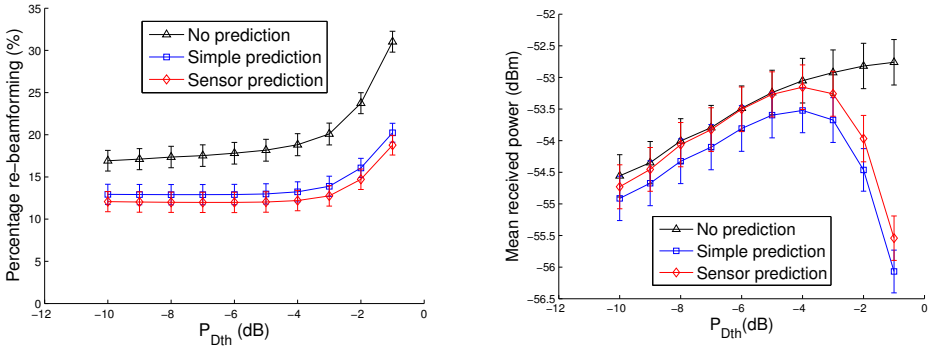
At every point along the route it is evaluated if the power has dropped by P_{Dth}. In case it has dropped by P_{Dth}, the prediction method is initiated and the next beam is predicted, or re-beamforming is applied when using no prediction. If the power still reaches the re-beamforming threshold P_{Rth}, the re-beamforming procedure is applied anyways. This may happen due to either switching to an incorrect beam, blockage or the fact that the current power combined with P_{Dth} is simply lower than P_{Rth}.

From Figure 5.4a it can be seen that only 0.1% re-beamforming is needed, using sensor prediction at $P_{Dth} = -5\,\text{dB}$. Using no prediction the re-beamforming percentage increases to 6.8% for the same P_{Dth}. Thus in the RWPM for scenario 1, the overhead due to re-beamforming is up to 68 times lower if sensors are used, as compared to using no predictions. For higher drop-off thresholds ($P_{Dth} = -3\,\text{dBm}$ to $-1\,\text{dBm}$) beam switching is triggered too often, increasing the re-beamforming percentage for all prediction methods.

The trade-off can be seen when looking at the mean received power along the route, as shown in Figure 5.4b. The mean received power of sensor prediction is almost equal with using no prediction for $P_{Dth} = -10\,\text{dBm}$ to $-4\,\text{dBm}$, whereas the signal power for simple prediction is always below no prediction and sensor prediction.

The previous results only show what happens when no blockage occurs, or if the blockage is immediately resolved by using NLOS beam paths and reflective surfaces. The results differ under the influence of blockage, since blocked signals encounter 20-30 dB more attenuation [13]. Thus these same

simulations were done, however now with the effects of body blockage. When the user's back is turned to the AP, we assume the user is standing in between the device and the AP, resulting in a blocked link and thus lowering the received power by 30 dBm. The width of this body blockage was assumed to be 30° in angular terms. The results of the RWPM simulation with blockage are shown in Figure 5.5.



(a) (b)

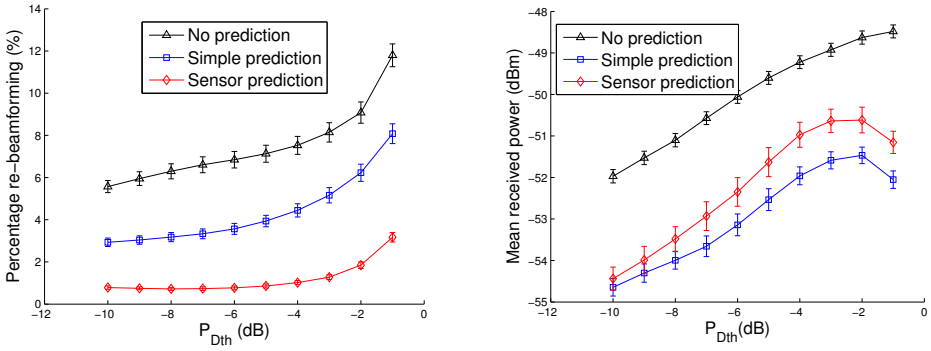
Figure 5.5: RWPM of scenario 1 *with* blockage effects and $P_{Rth} = -70$ dBm. Different drop-off thresholds are plotted against (a) the re-beamforming percentage and (b) the mean received power.

Comparing Figure 5.4 to Figure 5.5 it can be seen that blockage has a substantial effect on both graphs. At every instance, whenever the signal is blocked, re-beamforming is attempted. Since no appropriate beam link can be found in the case of blockage the power remains low. At the next instance re-beamforming is attempted again until the signal is no longer blocked. This results in an increase of the re-beamforming percentage by 12-13%, while the mean received power is lowered by 3-3.5 dBm. Interestingly enough the curves of both graphs do not change, only the vertical alignment is changed.

Scenario 2: walking and turning

This scenario also generated 100 random waypoints in the room, where the drop-off threshold was plotted against the re-beamforming percentage and the mean received power. No blockage effects were added, as the results would only differ by a vertical displacement, as was the case for scenario 1. The re-beamforming percentage was again obtained by normalizing the number of re-beamformings by the total number of points along the walked and angular route. This too was run 100 times to obtain a statistically significant mean and standard deviation as shown in Figure 5.6.

Figure 5.6a shows the same trend as can be seen in scenario 1, where sensor prediction has the lowest re-beamforming percentage. The minimum



(a) (b)

Figure 5.6: RWPM of scenario 2 without blockage effects and $P_{Rth} = -70\,\text{dBm}$. Different drop-off thresholds are plotted against (a) the re-beamforming percentage and (b) the mean received power.

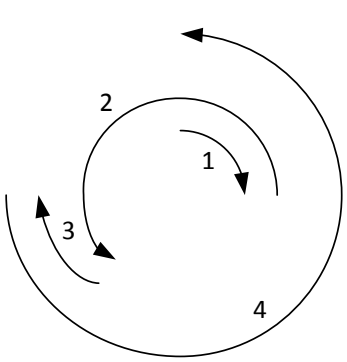
re-beamforming percentage of 0.8% is obtained at $P_{Dth} = -10$ to $-5\,\text{dBm}$ using sensor based prediction. As it was with scenario 1, for higher drop-off thresholds ($P_{Dth} = -4\,\text{dBm}$ to $-1\,\text{dBm}$) beam switching is triggered too often, increasing the re-beamforming percentage.

A noticeable difference between scenario 1 and 2 is seen in Figure 5.6b, where no prediction and sensor prediction no longer have the same mean received power for lower drop-off thresholds. Instead a difference of $\pm 2\,\text{dBm}$ mean received power can be observed between no prediction and sensor prediction. This is probably due to sensor prediction not choosing the optimal beam-pair, but still staying above the re-beamforming threshold P_{Rth}. Whereas no prediction continuously re-beamforms, which resets the current beam-pairs to the optimal beam-pairs, thus achieving a higher mean received power.

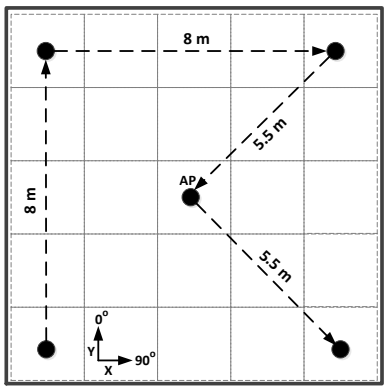
5.2.3 Single route verification

To verify the simulations, both scenarios are also performed in the real world, only on a smaller scale as compared to the RWPM. For both scenarios a simple route is created, as shown in Figure 5.7. In Figure 5.7a the rotations are shown for scenario 1, where the user is standing still and turning. The involved rotations are: $[0° \rightarrow 90° \rightarrow -135° \rightarrow -90° \rightarrow 0°]$. For scenario 2, the user is walking along a route, starting at $[x, y] = [1, 1]$, as shown in Figure 5.7b.

As already stated, to predict the next beam sector, it is assumed that the direction of the user with respect to the AP can be measured directly using the azimuth angle of the device given by the orientation sensor. The sensors were all set to the fastest sampling rate, as is shown in Table 3.1. The sensor data from the user and the received signal strength values from



(a) The angular route.



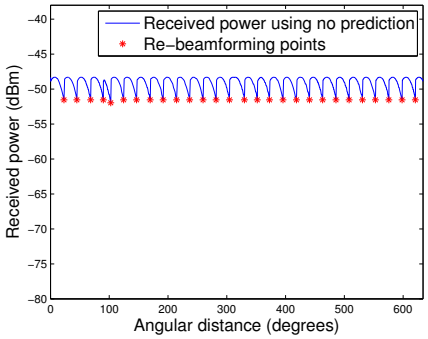
(b) The walking route.

Figure 5.7: The single routes for (a) scenario 1 and (b) scenario 2.

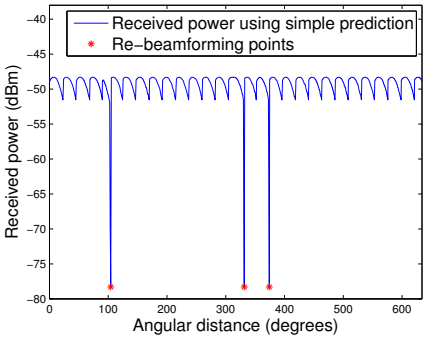
RPS are then combined in Matlab.

Scenario 1: stationary and turning

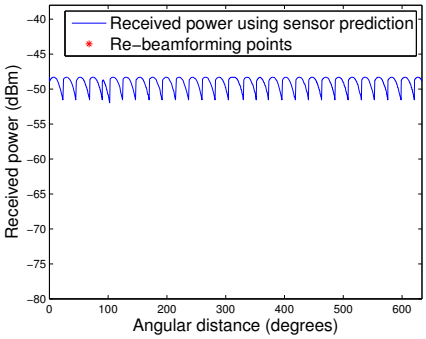
This route is defined as standing in one spot ($[x, y] = [2, 2]$) and turning to multiple different directions as shown in Figure 5.7a. For this scenario we chose $P_{Dth} = -3\,\text{dBm}$, without any body blockage. The received power for the three prediction methods is shown in Figure 5.8. Re-beamforming



(a) No prediction.



(b) Simple prediction.



(c) Prediction using sensors.

Figure 5.8: The received power for the three prediction methods, along the route of Figure 5.7a, without blockage and using $P_{Dth} = -3\,\text{dB}$ and $P_{Rth} = -70\,\text{dBm}$.

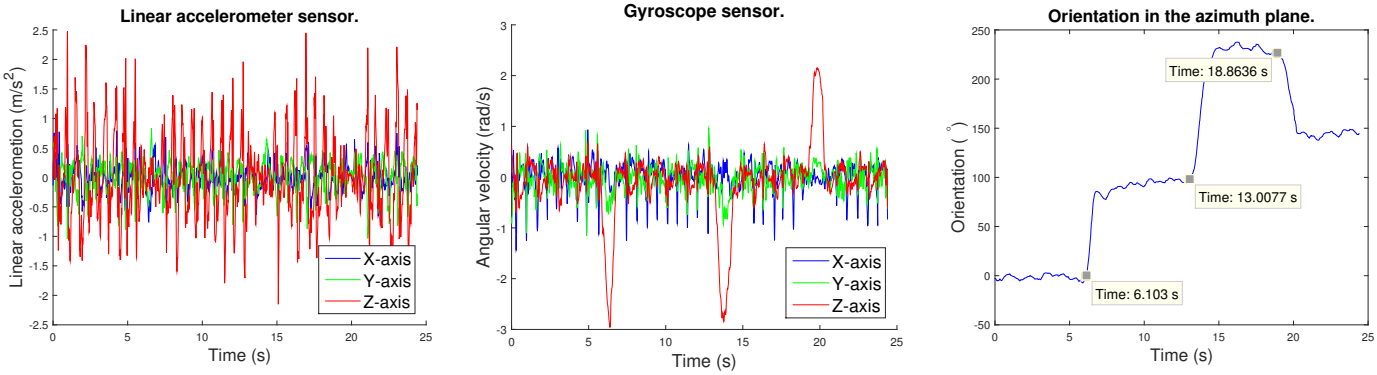
needs to be done if a switch is made to a wrong beam sector or if the re-beamforming threshold is reached. These locations are indicated with a red star in Figures 5.8a to 5.8c.

Using no prediction, re-beamforming is performed every time the power drops by P_{Dth}, resulting in 28 re-beamformings as shown in Figure 5.8a.

Next to that in Figure 5.8b, three re-beamformings are required to sustain the beamformed link, once for every time the turning direction changed (not including the start). And finally Figure 5.8c shows no re-beamforming is needed when sensor prediction is employed. This confirms the trend of the RWPM simulations, that the re-beamforming percentage is lowest for sensor prediction, and the mean signal strength is lowest for simple prediction due to the sudden power drops.

Scenario 2: walking and turning

Using real measured sensor data along the route shown in Figure 5.7b, the three sensors (accelerometer, gyroscope and azimuth orientation) are plotted in Figure 5.9. Figure 5.9a shows the accelerometer data, where the steps

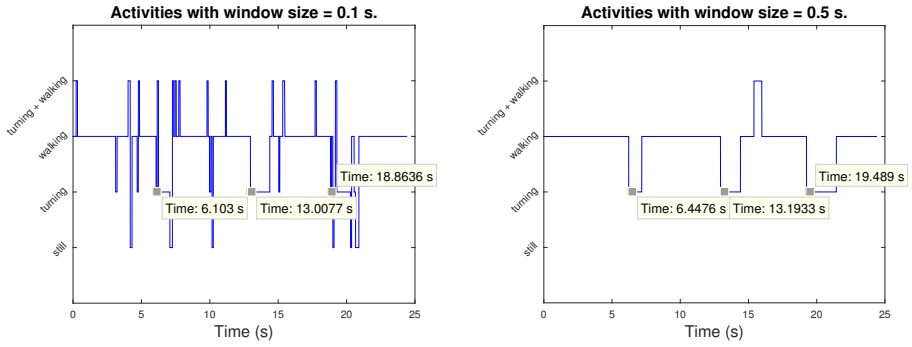


(a) (b) (c)

Figure 5.9: Measured sensor data along the route of Figure 5.7b, where (a) shows the accelerometer data, (b) the gyroscope data and (c) the azimuth orientation.

of the user correspond to the pulses from the accelerometer's Z-axis. The turning points of the route can be seen from the gyroscope's pulses along the Z-axis in Figure 5.9b, and also directly from the device's azimuth orientation in Figure 5.9c.

The k-NN activity recognition algorithm is applied using these sensor values, the results of which are shown in Figure 5.10. The parameters of the k-NN algorithm were chosen to be $k = 3$ with a window size of 0.1 s to achieve the fastest reaction. Again the importance of window size is shown in both Figures 5.10a and 5.10b. Figure 5.10a uses a window size of 0.1 s, and shows that walking and turning are both generally identified correctly, however some erroneous hops are made between the recognized activities. In contrast, Figure 5.10b uses a window size of 0.5 s and does not show these small hops. Thus a window size of 0.5 s may be more accurate, however there is an increased delay before the activity is recognized. The time stamps on

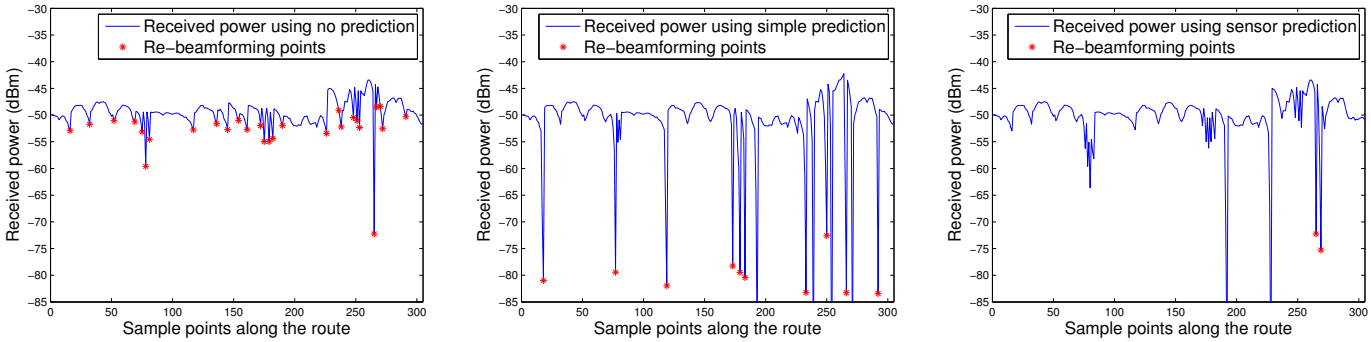


(a) (b)

Figure 5.10: The identified activities using k-NN with $k = 3$ and a window size of (a) 0.1 s and (b) 0.5 s, using the sensor values from Figure 5.9.

the graphs indicate that using a window size of 0.5 s increases the delay in the range of 0.2 s to 0.5 s. To see how quickly a window size of 0.1 s is able to identify the angular movement, time stamps are also placed in Figure 5.9c.

As was done for scenario 1, the received power of the three prediction methods was also calculated for the route of scenario 2, as shown in Figure 5.11. Again we chose $P_{Dth} = -3$ dBm, without any body blockage. Using no prediction as seen in Figure 5.11a, re-beamforming has to be per-



(a) No prediction. (b) Simple prediction. (c) Prediction using sensors.

Figure 5.11: The received power along the route of Figure 5.7b, without blockage and using $P_{Dth} = -3$ dBm and $P_{Rth} = -70$ dBm.

formed 28 times along the route. In contrast Figure 5.11b shows only 14 re-beamformings are required along the route when using the simple prediction method. The number of re-beamformings along the route is reduced to 4 if the sensor prediction method is employed, as shown in Figure 5.11c. These results also confirm the trends which were obtained from the RWPM setup.

Three methods for predicting the next beam direction of both STA and AP were analyzed. The first method was using no prediction, which usually achieved the highest average received power, at the high cost of re-beamforming often. The second method consisted of using the previous and current beam/sector to predict the next beam/sector. Results indicated that it performed worse than no prediction w.r.t. average received power, but that it performed better w.r.t. number of re-beamformings. The last prediction method uses sensor data available to modern devices to predict the next beam-pair. The sensor prediction method has the lowest number of re-beamformings and still manages to maintain a good average received power.

Chapter 6

Conclusion

The unprecedented increase in the number of wireless communication devices and the emergence of new bandwidth hungry multimedia applications require multi-Gb/s connectivity. The availability of large amounts of bandwidth in the 60 GHz frequency band has made it a promising candidate for short-range high-speed communication. To combat the high free-space path loss at 60 GHz, directional antennas are used to confine signal power in the desired direction. Unfortunately, directional links are prone to link outage due to user movement. User movement introduces translational, rotational or blockage errors in 60 GHz networks which can cause beam misalignment when directional antennas are used. Using commonly available sensors in mobile devices, it is possible to identify and predict the movement and quickly realign the beams without disrupting the connection. This work takes the first steps to incorporate sensor data as a means of improving network performance.

To identify the movement causing the error, two classification algorithms were tried; one based on k-NN and one based on HMMs. A trade-off between window sampling time and accuracy was observed for both classification algorithms, where most success was achieved with the k-NN algorithm.

Sensors were employed to predict the movement of users in a 60 GHz network before the communication link is lost. Simulation results showed sensor-based prediction can significantly reduce the number of beam searches and thus lower the MAC overhead. This was verified by using real-life sensor data applied to a generic route.

6.1 Limitations

As is the case with many works, this thesis too has its limitations. First of all, the IEEE 802.11ad standard has no fields wherein the sensor data can be transmitted. This means either the PHY or MAC data fields need to include the sensor data.

Furthermore, we assumed the user's device only has an antenna pattern in the azimuth plane, thus the beam stretched the whole elevation plane. It was also assumed the user's movement direction could be measured by the azimuth angle of the device. Both these assumptions were made for simplicity's sake, and thus might not be an accurate representation of reality.

A lot of parameters were used during this research. Depending on the parameter, it can impact the results either drastically or almost not at all. As such this work does not list all the possible outcomes for employing sensor based movement identification and prediction in 60 GHz networks. However an attempt was made to create a generic environment, covering many possible scenarios.

As was already discussed the sampling rate of current smartphone sensors are limited by manufacturers, presumably to limit energy consumption. However, faster sensor sampling allows for faster and more accurate activity recognition, lowering the number of re-beamformings, and thus improves link stability.

6.2 Future work

Many research directions are possible within this work, hence many areas were not explored. Some suggestions for future work are given below.

Sudden drops in power were observed when switching to an incorrect beam-pair. We believe this can be improved upon by using a quick beam-pair test, to rapidly see if the predicted beam-pair is capable of sustaining the link. Furthermore, both the AP and the user's device switch beams at the same time when a dip in signal power is observed. This might not be ideal and can be improved upon by using some sort of alternating algorithm such that the change in signal power is more subtle.

The effects of blockage were investigated; however this was not done in detail, due to time restrictions. Future work may focus on blockage effects, other than blockage due to the user's body. Also an emphasis may be put on how sensors can assist in using NLOS paths to help recover a blocked link.

Lastly, energy consumptions were not measured. It is reasonable to assume that sensor-assisted movement identification and prediction will consume additional energy, especially if the sensors are sampling at high frequencies. Thus further research is required for sensor-assisted 60 GHz communication on the area of energy consumption.

Bibliography

[1] IEEE Standard for Information technology - Telecommunications and information exchange between systems. *IEEE Std 802.15.3c-2009 (Amendment to IEEE Std 802.15.3-2003)*, pages c1–187, Oct 2009.

[2] IEEE Standard for Information technology–Telecommunications and information exchange between systems. *IEEE Std 802.11ad-2012 (Amendment to IEEE Std 802.11-2012, as amended by IEEE Std 802.11ae-2012 and IEEE Std 802.11aa-2012)*, pages 1–628, 2012.

[3] Xueli An, Chin-Sean Sum, R.V. Prasad, Junyi Wang, Zhou Lan, Jing Wang, R. Hekmat, H. Harada, and I. Niemegeers. Beam switching support to resolve link-blockage problem in 60 GHz WPANs. In *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on*, pages 390–394, 2009.

[4] A. Anjum and M.U. Ilyas. Activity recognition using smartphone sensors. In *Consumer Communications and Networking Conference (CCNC), 2013 IEEE*, pages 914–919, Jan 2013.

[5] Shahid Ayub, Alireza Bahraminasab, and Bahram Honary. *A Sensor Fusion Method for Smart phone Orientation Estimation.* 2012.

[6] Oresti Banos, Juan-Manuel Galvez, Miguel Damas, Hector Pomares, and Ignacio Rojas. Window Size Impact in Human Activity Recognition. *Sensors*, 14(4):6474–6499, 2014.

[7] Phil Blunsom. Hidden markov models. *Lecture notes, August*, 15:18–19, 2004.

[8] Carlos Cordeiro, Dmitry Akhmetov, and Minyoung Park. IEEE 802.11Ad: Introduction and Performance Evaluation of the First Multi-gbps Wifi Technology. In *Proceedings of the 2010 ACM International Workshop on mmWave Communications: From Circuits to Networks*, mmCom '10, pages 3–8, New York, NY, USA, 2010. ACM.

[9] Bao Linh Dang, R. Venkatesha Prasad, Ignas G. Niemegeers, M. Garcia Larrode, and A. M. J. Koonen. Toward a Seamless Communication Architecture for In-building Networks at the 60 GHz band. In *LCN'06*, pages 300–307, 2006.

[10] J. Deissne and et al. RPS Radiowave Propagation Simulator User Manual-Version 5.4. *Actix GmbH*, 2008.

[11] FCC. Code of Federal Regulation, title 47 Telecommunication, chapter 1, part 15.255, October 2010.

[12] H.T. Friis. A Note on a Simple Transmission Formula. *Proceedings of the IRE*, 34(5):254–256, May 1946.

[13] Z. Genc, U.H. Rizvi, E. Onur, and I. Niemegeers. Robust 60 GHz Indoor Connectivity: Is It Possible with Reflections? In *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*, pages 1–5, May 2010.

[14] M. Jacob, S. Priebe, R. Dickhoff, T. Kleine-Ostmann, T. Schrader, and T. Kurner. Diffraction in mm and Sub-mm Wave Indoor Propagation Channels. *Microwave Theory and Techniques, IEEE Transactions on*, 60(3):833–844, 2012.

[15] John D Kraus. Antennas. 1988.

[16] Young-Seol Lee and Sung-Bae Cho. Activity Recognition Using Hierarchical Hidden Markov Models on a Smartphone with 3D Accelerometer. In *Proceedings of the 6th International Conference on Hybrid Artificial Intelligent Systems - Volume Part I*, HAIS'11, pages 460–467, Berlin, Heidelberg, 2011. Springer-Verlag.

[17] Alexander Maltsev and et. al. Channel Models for 60 GHz WLAN Systems, May 2010.

[18] Minyoung Park and Helen K. Pan. Effect of Device Mobility and Phased Array Antennas on 60 GHz Wireless Networks. In *Proceedings of the 2010 ACM International Workshop on mmWave Communications: From Circuits to Networks*, mmCom '10, pages 51–56, New York, NY, USA, 2010. ACM.

[19] L. E. Peterson. K-nearest neighbor. 4(2):1883, 2009. revision #136646.

[20] Bien Van Quang, R. Venkatesha Prasad, and I Niemegeers. A Survey on Handoffs #x2014; Lessons for 60 GHz Based Wireless Systems. *Communications Surveys Tutorials, IEEE*, 14(1):64–86, Jan 2012.

[21] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb 1989.

[22] T.S. Rappaport, J.N. Murdock, and F. Gutierrez. State of the Art in 60-GHz Integrated Circuits and Systems for Wireless Communications. *Proceedings of the IEEE*, 99(8):1390–1436, Aug 2011.

[23] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L Littman. Activity recognition from accelerometer data. In *AAAI*, volume 5, pages 1541–1546, 2005.

[24] Nirupam Roy, He Wang, and Romit Roy Choudhury. Demo: I am a smartphone and i can tell my user's walking direction. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, pages 354–354. ACM, 2014.

[25] Bosch Sensortec. BMA250 Digital, triaxial acceleration sensor, May 2012. Datasheet.

[26] S. Shankar N., D. Dash, H. El Madi, and G. Gopalakrishnan. WiGig and IEEE 802.11ad - For multi-gigabyte-per-second WPAN and WLAN. *ArXiv e-prints*, Nov. 2012.

[27] S. Singh, F. Ziliotto, and U. Madhow. Blockage and directivity in 60 GHz Wireless Personal Area Networks: From Cross Layer Model to Multi-hop MAC Design. *IEEE J. Sel. Areas Commun., Special Issue on Realizing Gbps Wireless Personal Area Networks*, 27:1400–1413, Oct. 2009.

[28] P. Smulders. Exploiting the 60 GHz band for local wireless multimedia access: prospects and future directions. *Communications Magazine, IEEE*, 40(1):140–147, 2002.

[29] STMicroelectronics. LIS331DLH, MEMS digital output motion sensor ultra low-power high performance 3-axes "nano" accelerometer, July 2009. Datasheet.

[30] STMicroelectronics. L3G4200D, MEMS motion sensor: ultra-stable three-axis digital output gyroscope, December 2010. Datasheet.

[31] STMicroelectronics. iNEMO inertial module: 3D accelerometer and 3D gyroscope, September 2012. Datasheet.

[32] Y.M. Tsang and A.S.Y. Poon. Detecting Human Blockage and Device Movement in mmWave Communication System. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1–6, Dec 2011.

[33] Su Khiong Yong and Chia-Chin Chong. An Overview of Multigigabit Wireless Through Millimeter Wave Technology: Potentials and Technical Challenges. *EURASIP J. Wirel. Commun. Netw.*, 2007(1):50–50, Jan. 2007.

[34] Su-Khiong (SK) Yong, Pengfei Xia, and Alberto Valdes Garcia. *60 GHz Technology for Gbps, WLAN, WPAN: From Theory to Practice*. John Wiley & Sons, Ltd, 2011.

List of publications

[1] Arjan W. Doff, Kishor Chandra and R. Venkatesha Prasad. Sensor Assisted Movement Identification and Prediction for Beamformed 60 GHz Links. In *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC) (CCNC 2015)*, pages 648-653, Jan. 2015.

[2] Kishor Chandra, Arjan W. Doff, Zizheng Cao, R. Venkatesha Prasad and Ignas G.M.M. Niemegeers. 60 GHz MAC Standardization: Progress and Way Forward. In *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC) (CCNC 2015)*, pages 184-189, Jan. 2015.