NTNU – Trondheim
Norwegian University of
Science and Technology

Localization and Tracking of Intestinal Paths for Wireless Capsule Endoscopy

Anders Sandrib Bjørnevik

Master of Science in Electronics
Submission date: June 2015
Supervisor: Ilangko Balasingham, IET
Co-supervisor: Pål Anders Floor, IET

Norwegian University of Science and Technology
Department of Electronics and Telecommunications

Abstract

Wireless capsule endoscopy (WCE) is a non-invasive technology used for visual inspection of the human gastrointestinal (GI) tract. Localization of the capsule is a vital component of the system, as this enables physicians to identify the position of abnormalities. Several approaches exist that use the received signal strength (RSS) of the radio frequency (RF) signals for localization. However, few of these utilize the sparseness of the signals. Due to intestinal motility, the capsule positions will change with time. The distance travelled by the capsule in the intestine, however, remains more or less constant with time. In this thesis, a compressive sensing (CS) based localization algorithm is presented, that utilize signal sparsity in the RSS measurements. Different ℓ_1-minimization algorithms are used to find the sparse location vector. The performance is evaluated by electromagnetic (EM) simulations performed on a human voxel model, using narrow-band (NB) and ultra wide-band (UWB) signals. From intestinal positions, the distance the capsule has travelled is estimated by use of Kalman- and particle filters. It was found that localization accuracy of a few millimeters is possible under ideal conditions, when the RSS measurements are generated from a path loss model. When using path loss data from the EM simulations, localization accuracy on the order of 20-30 mm was achievable for NB signals. Use of UWB signals resulted in localization errors between 35-60 mm, depending on frequency range and bandwidth. From generated intestinal positions, the travelled distance was estimated with a minimum accuracy of a few millimeters, when using a VNL Kalman filter and moderate amounts of observation noise. The results are found from a limited amount of data. In order to increase the confidence in the presented results, the performance of the localization algorithm and the filters should be evaluated with a larger number of datasets.

Sammendrag

Trådløs kapselendoskopi (WCE) er en banebrytende metode som brukes til å undersøke menneskets fordøyelsessystem ved hjelp av bildeopptak. Posisjonering av kapselen er en kritisk del av systemet, da dette muliggjør å finne tilbake til interessante lokasjoner i tarmen. Det finnes flere metoder som benytter mottatt signalstyrke (RSS) til lokalisering, men få av disse har undersøkt å bruke compressive sensing (CS) til dette formålet. Tarmen beveger seg kontinuerlig, noe som gjør at faste posisjonsdata over tid blir mindre pålitelige. Imidlertid vil distansen kapselen har beveget seg være mer eller mindre konstant. I denne oppgaven blir det presentert en CS-basert lokaliseringsalgoritme. Fra posisjonsdata blir kapselens tilbakelagte distanse beregnet ved hjelp av Kalman- og partikkelfiltre. Systemets ytelse blir vurdert ved hjelp av EM simuleringer på en tredimensjonal modell av menneskekroppen. Signaler med smalt (NB) og bredt frekvensspektrum (UWB) blir benyttet og sammenlignet. Det ble funnet at en posisjonsnøyaktighet mellom 20-30 mm er mulig dersom det brukes NB-signaler. Bruk av UWB-signaler resulterte i posisjonsfeil mellom 35-60 mm, avhengig av frekvensbånd og båndbredde. Den tilbakelagte distansen til kapselendoskopet ble beregnet med noen millimeters feilmargin ved moderate støynivåer ved bruk av multi-mode VNL Kalman filter. Resultatene er basert på en begrenset mengde data. For å få mindre usikkerhet i resultatene, bør ytelsen til lokaliseringsalgoritmen og filtrene vurderes med flere datasett.

Preface

This thesis is submitted as partial fulfillment of the requirements for the degree of Master of Science (M.Sc.) at Norwegian University of Science and Technology (NTNU). The work presented in the thesis has been conducted in the spring semester of 2015, and is an extension of work done in the specialization project performed in the fall of 2014.

The thesis work has been both challenging and exiting. With the area of wireless capsule endoscopy being in rapid development, this has presented many interesting opportunities where little previous research has been done. Due to the human benefit from this technology, it also felt meaningful working with this topic.

I would like to thank my supervisors Ilangko Balasingham and Pål Anders Floor for providing me with insightful discussions and leading me in the right direction during the development of the thesis. They have given valuable advice and been supportive of my work.

Rohit Chandra and Ali Khaleghi deserve to be thanked for providing me with recommendations and hints in the usage of the EM simulation software CST Microwave Studio.

Last, I would like to thank my friends and family for being supportive throughout the development of the thesis. This is especially true for my fiancée Une Sesilie Breines, which has had to live with me though some busy months.

Anders Sandrib Bjørnevik

Trondheim, 22.06.2015

Contents

List of Figures

List of Tables

List of Abbreviations

AD-BCS	Adaptive Bayesian compressive sensing
AOA	Angle of arrival
BCS	Bayesian compressive sensing
BGM	Bayesian graphical model
BP	Basis pursuit
BPDN	Basis pursuit denoising
COM	Component object model
CRLB	Cramer-Rao lower bound
CS	Compressive sensing
CSV	Comma-separated values
CST	Computer Simulation Technology
CT	Computed tomography
EM	Electromagnetic
EMC	Electromagnetic compatibility
FCC	Federal Communication Commission
FDTD	Finite difference time domain
FIT	Finite integration technique
GI	Gastrointestinal tract
GUI	Graphical user interface
HUGO	Visible human project voxel model
IB	In-body
ISM	Industrial, scientific and medical
ITU	International Telecommunication Union
KF	Kalman filter
LE	Localization error

LOS	Line-of-sight
MC	Monte Carlo
MICS	Medical implant communication service
MRI	Magnetic resonance imaging
MT-BCS	Multi-task Bayesian compressive sensing
MWS	Microwave Studio
NB	Narrow-band
NLOS	Non-line-of-sight
NPE	Normalized position error
OB	On-body
PBA	Perfect boundary approximation
PDE	Partial differential equation
PDF	Probability density function
PF	Particle filter
PL	Path loss
RF	Radio frequency
RIP	Restricted isometry property
RL	Return loss
RMSE	Root mean square error
RSSI	Received signal strength intensity
RSS	Received signal strength
RWM	Random waypoint model
SIR	Sequential importance resampling
SIS	Sequential importance sampling
SNR	Signal-to-noise-ratio
SOCP	Second-order cone program
ST-BCS	Single-task Bayesian compressive sensing
TOA	Time of arrival
UWB	Ultra wide-band
VBA	Visual basic for applications
VNL	Variable noise level
VSD	Variable state dimension
WCE	Wireless capsule endoscope
WLAN	Wireless local area network
WSN	Wireless sensor network

Chapter 1

Introduction

Wireless capsule endoscopy (WCE) is an emerging technology for performing examination of the gastrointestinal (GI) system. The patient is examined by swallowing a pill capsule containing a small video camera (Fig. 1.1). The capsule follows the gastrointestinal system from the esophagus to the colon, locomoted by natural contractions in the intestines (peristalsis) [1]. The physician examines the recorded images for abnormalities, thus enabling non-invasive diagnosis of diseases such as gastrointestinal bleeding, cancer, celiac disease and Crohn's disease [2].

There exists multiple commercially available WCEs [4]. The PillCam capsules from Given Imaging have one or more cameras with light sources, a radio frequency (RF) transmitter and a power source. The patient wears a sensor belt, containing an array of sensors that is used to localize and obtain telemetry data from the capsule.

Although commercial WCE products are available, this is a highly active research field. One of the current research objectives is to overcome the limits imposed by the peristaltic locomotion, by adding maneuverability to the capsule. This can enable precise drug delivery and local disease treatment [5]. Other areas receiving attention includes improving visual quality, localization and power usage [1].

Figure 1.1: Two PillCam models from Given Imaging. The upper capsule is designed for inspection of the esophagus, while the one at the bottom is for use in the small intestine. Picture from [3], used with permission.

1.1 Motivation

The WCE models on the market at the moment do not have any means to treat potential diseases found. Thus, if the physician finds any irregularities on the video that needs mechanical treatment, he will have to return to the position with the proper tool. This makes localization of the capsule a vital component of the system [1].

The localization can be performed by different technologies such as ultrasound [6], microwave imaging [7], computer vision [8], radio frequency (RF) signals [9, 10, 11, 12] or by including a permanent magnet in the capsule [13, 14]. Of these methods, the RF and magnet based techniques have received most attention.

In RF and magnet based localization, the patient wears a vest containing an array of sensors sensing the signals. Localization accuracy for the magnet-based methods is on the order of a few millimetres [14]. RF-based methods have a lower accuracy, but comes with the cost-reducing benefit of using the sensors for both communication and localization [12]. Also, as there is no magnet inside the capsule, more space is available for hardware. This space can e.g. be used to increase the battery capacity.

Compressive sensing (CS) is a field that has been extensively studied the latest years, in many sciences and applications [15, 16]. Two such fields are wireless local area networks (WLAN) and sensor networks (WSN), where CS is used for localization [17, 18, 19]. Despite the similarities between WSNs and the WCE localization setup, little work has been done on using CS for WCE localization. To our knowledge, the only article that treats this subject is [20].

Using ultra wide-band signals (UWB) for communication with medical implants have received increased interest in the recent years. Compared to narrow-band (NB) communication, the wide bandwidth allows larger data rates and proposes increased localization accuracy due to higher temporal resolution [21]. However, this promising technology has seen little research for WCE localization usage.

When the physician examines the video footage captured from a WCE, the video is tagged with position information. However, due to intestinal motility, the intestines are constantly moving [22]. This makes a fixed position of little relevance, because the position of the abnormality can have moved significantly at the time of treatment. A better way to approach this problem, is to use the distance travelled from a known point in the intestine, e.g. the entry of the stomach [23]. This distance is not as greatly affected by the intestinal movement as a fixed position.

1.2 Current work

1.2.1 RF localization in wireless capsule endoscopy

The commercial system in [24] is the first reported approach to the WCE localization problem. The patent describes a system in which the patient wears a sensor belt that measures the received signal strength (RSS). Using these measurements, the position is found by triangulation. In [9] the localization error of this system is found experimentally as 37.7 mm.

In [10], the authors use time-of-arrival (TOA) to localize medical implantable devices. The method uses electromagnetic properties from the body tissues found from CT or MRI images, to estimate the propagation speed used for localization by TOA. The position error is reported as 15-60 mm on moderate noise levels.

In [25], Ye et al derives the Cramer-Rao lower bound (CRLB) variance limit for estimators that use RSS measurements for positioning. Number of receiver sensors and number of capsules are also investigated. The presented results show that at least 30 receivers are needed for a guaranteed root-mean square error (RMSE) of 50 mm, with the accuracy gradually increasing for larger number of receivers.

In [26], a WCE localization method based on a Bayesian graphical model (BGM) is presented. The technique uses RSS measurements combined with Gibbs sampling to infer the capsule location, and works in three dimensions. The authors use RSS measurements generated using a statistical path loss model for a WCE following a path along the intestine to test the performance of the algorithm. The results show an average median distance error of 5.7 mm in three dimensions, for a measurement setup having 32 receivers in the 401-406 MHz frequency band.

The article [27] investigates localization of a capsule in the brain. The capsule is localized by measuring angle of arrival (AOA) of UWB pulses, received by a sensor array on the body. The initial results show a maximum localization error of 12 mm, which is considered too poor for further investigation. The authors state that this is caused by having non-line-of-sight (NLOS) conditions and that the channel is inhomogeneous.

In [28], the authors propose a method to localize the WCE using the images captured in the intestine, combined with RF signals received by an array of on-body antennas. The speed and direction of the WCE is estimated by tracking motion vectors of unique features in the captured images. The RF localization is performed by least squares RSS positioning, which is fused with the motion tracking by a Kalman filter.

Chandra et al. [7] propose an algorithm that utilizes microwave imaging for localizing a WCE in the GI tract. Microwave imaging is used to estimate the electrical properties of the tissues at 403 MHz, by using an array of antennas arranged around the body. These antennas are subsequently used as receivers, to measure the phase and amplitude of the electrical field transmitted from the capsule. The position is found as the localization cell that minimizes the field differences between the imaging results and the measurements. The results show a two-dimensional mean error of 3 mm when using a realistic human body phantom.

In [29] the authors propose an in-body localization algorithm where TOA and RSS are employed using CS. The algorithm estimates the tissue properties of the human tissues. Path loss and the estimated time-of-flight data is directly used for the ℓ_1-minimization, which is solved by the regularized basis pursuit denoising (BPDN) algorithm. Using the 400 MHz ISM band, the results show a RMSE of 0.5-1 cm.

In [12], Ito et al. propose a joint TOA/RSS localization algorithm combined with a particle filter for tracking the position. The RSS readings are used to estimate the relative permittivity of the biological tissues. The accuracy of this estimation is validated by EM simulations. The localization and tracking is done by TOA with a particle filter. The particle filter uses the RWM motion model when estimating the next state. The results show a three-dimensional RMSE localization accuracy of 2 mm.

1.2.2 Localization using compressive sensing in WSN/WLAN

Through the last years, there has been an increasing interest in applying compressive sensing for localization in WLAN and WSNs. The first articles that employ CS for localization in these areas are [17, 18, 19], where it is realized that the problem of localizing targets in sensor networks can be formulated as a sparse estimation problem in the two-dimensional spatial domain.

In [17], Feng et al. use CS to localize multiple targets in a WLAN by RSS measurements. Different ℓ_1-minimization algorithms are compared, and it is found that with more than eight measurements, reliable position recovery of the targets are possible. The noise resilience is also investigated, with the results presented having a maximum error of 0.58 m in a 10×10 m grid when the signal-to-noise ratio (SNR) is 25 dB.

In [30], the problem is treated more rigorously, leading to a formalized and validated CS theory for localization usage. The authors propose a sparse minimization greedy matching pursuit (GMP) algorithm and also address the problem of counting and positioning targets from multiple categories.

One of the first works presenting a Bayesian framework for localization in WSNs is the paper [31]. This article uses Bayesian inference to estimate the location, but the framework is not very general, and is designed with a decentralized sensor network in mind.

The conference paper [32] and its following journal paper [33] presents a WSN localization algorithm that uses the Bayesian compressive sensing (BCS) framework developed in [34, 35]. The articles present an algorithm called *AMBL* that utilizes the intra- and inter-signal correlations in the RSS measurements for reliable localization. The algorithm also determines the amount of measurements needed for an accurate position estimate. It is shown that the algorithm require few measurements for accurate recovery of multiple targets in a 20×20 m grid. The algorithm is also found to be resilient to Gaussian noise.

In [36], the BCS framework from [34] is used for localization in a WSN. The performance is compared with the basis pursuit (BP) and greedy matching pursuit (GMP) algorithms. The presented results show an error on the order of 0.25 m in a 128×128 m grid with a SNR of 10 dB.

All the presented articles consider the two-dimensional localization problem. However, in [37], it is shown that localization using CS can be extended to three dimensions by using tensor analysis. The results from a WSN of 12×12 m show a three-dimensional Euclidean error of 1-1.5 m.

1.2.3 Tracking and distance estimation

To our knowledge, the only article that considers the problem of estimating the distance a WCE has travelled in the intestines is [38]. The proposed method use the video-based distance estimation method from [28], with the performance evaluated by generated and clinical data. The results show an average distance error of 2.71 cm, compared to existing methods that have an error of 27 cm, when using generated video data. No claims are made on the accuracy when using clinical data.

In other fields, similar problems are investigated. Pathirana et al. [39] investigate the problem of tracking a mobile user by use of RSS in cellular networks. This is considered for mobile ad-hoc networks, where both users and base stations can be moving. The authors use a non-linear equation of forward link RSSI as the foundation, formulating an uncertain system where the solution is derived to be a PDE. This PDE is solved by use of a robust extended Kalman filter. Promising results are shown, with accurate tracking of a mobile user having a highly non-linear path. The algorithm only needs a single base station to get decent tracking performance.

Shrivastava et al. [40] investigate the problem of tracking a target in a two-dimensional field of binary proximity sensors. The authors develop an algorithm that use a particle filter with the RWM movement model. The results show that the particle filter provides accurate tracking performance.

In [41], tracking of multiple objects from a mobile robot is considered. The technique developed uses the robot's sensors and motion models of the objects being tracked, in combination with particle filters and joint probabilistic data assertion filters (JPDAF), to estimate the trajectory of the tracked objects. The results show good tracking performance compared to Kalman filter approaches, especially when there exist occlusions or obstacles.

1.3 Objective and limitations

One of the aims of this thesis is to address the problem of localizing a WCE using RSS measurements. This is done using sparse signal recovery by the compressive sensing framework, with the performance of the localization algorithm evaluated by in-body numerical EM simulations. The problem will be limited to the two dimensional case. Both NB and UWB signals are considered, and the accuracy will be compared.

The problem of tracking the path of the WCE in the intestines is also addressed. From noisy, intestinal position data, the distance the capsule has travelled is found by use of velocity estimates. The performance is evaluated by the known distance and positions.

1.4 Structure of the thesis

Some background information about compressive sensing, Kalman and particle filters and the EM simulation software will be presented in Ch. 2. Ch. 3 presents the localization and tracking framework, with implementation details following in Ch. 4.

Results from the EM simulations are presented at the beginning of Ch. 5, followed by evaluation of the localization algorithm and tracking performance. Discussion of the results and suggestions for further work follows in Ch. 6. In Ch. 7, a summary of the findings of this thesis will be given.

Appx. A contains figures and tables that is relevant to topics discussed in the thesis, but was considered to be too large or to comprehensive to be included in the main text. The attachments are organized as one section with figures (Sec. A.1) and one section with tables (Sec. A.2). In Appx. B, a selection of the most relevant MATLAB code is

included. Appx. C contains selected parts of a framework developed for scripting EM simulations in CST MWS.

Chapter 2

Background

Some background material will be presented in this chapter. Sec. 2.1 will provide information on localization using RF signals, how path loss models are defined and a short introduction to UWB signals. Sec. 2.2 gives a brief overview of CS theory. In Sec. 2.3, the Kalman and particle filters are presented. The last part of the chapter (Sec. 2.4) contains an introduction to the numerical EM simulation software CST Microwave Studio.

2.1 Localization using RF signals

Localization using RF signals is typically performed using angle-of-arrival (AOA), time-of-arrival (TOA) or received signal strength (RSS) measurements [42]. As only RSS is of interest in this thesis, the presentation will be limited to this technique.

The RSS measurements are used to estimate the distance from a transmitter to a number of receivers. By knowledge of the transmitted and received power, the path loss can be found. From a model relating the distance with loss of power, the distance can be estimated. Once the distances are obtained, the location can be found by

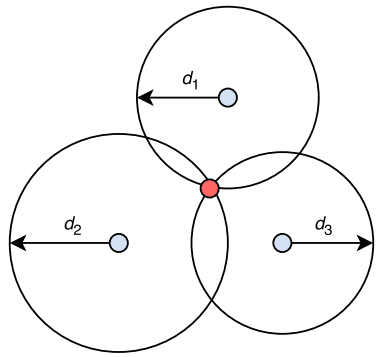


Figure 2.1: Multilateration with distances $d_1 - d_3$, after [43].

multilateration [42]. This is illustrated in Fig. 2.1, where three distances $d_1 - d_3$ are used to locate the red transmitter.

In the following section, the computation of path loss and different path loss models will be presented, with the source being the book [44].

2.1.1 Path loss

The path loss is defined as the ratio of the transmitted power P_t to the received power P_r

$$\mathrm{PL} = \frac{P_t}{P_r}\,. \tag{2.1}$$

For convenience, the path loss and the powers are usually expressed in decibels, which gives an expression of the form

$$\mathrm{PL}_{dB} = 10\log_{10}(\mathrm{PL})\,. \tag{2.2}$$

In this thesis, the explicit decibel notation is omitted for convenience. If the path loss is not in decibels, it will be explicitly mentioned.

Path loss models describe a communication channel, predicting the power loss from the transmitter to the receiver. These models can be separated in large-scale and small-scale models. Large-scale models describe the average power as the receiver moves away from the transmitter, following a gradual decrease in the signal strength. Small-scale models deals with multipath fading, which leads to large differences in received power with only small distance differences.

Free-space path loss models

In free space, with line-of-sight (LOS) between transmitter and receiver, the power decays with the distance squared resulting in a path loss as

$$\mathrm{PL}(d) = \left(\frac{4\pi d}{\lambda}\right)^2, \tag{2.3}$$

where d is the distance and λ is the wavelength. If we also include the effects of the receiver and transmitter antennas, it follows from Friis transmission equation that

$$\mathrm{PL}(d) = G_t G_r \left(\frac{4\pi d}{\lambda}\right)^2, \tag{2.4}$$

where G_t and G_r are the transmitter and receiver antenna gains respectively.

Table 2.1: Path loss exponents for different environments [44].

Environment	Path loss exponent n
Free space	2
Urban area cellular radio	2.7 - 3.5
Shadowed urban cellular radio	3 - 5
Inside a building, LOS	1.6 - 1.8
Obstructed in building, NLOS	4 - 6
Obstructed in factories, NLOS	2 - 3

Log-distance path loss models

From empirical measurements, it has been found that for many environments the path loss follows

$$\mathrm{PL}(d) = \mathrm{PL}_0 + 10n \log_{10}\left(\frac{d}{d_0}\right), \tag{2.5}$$

where PL is the path loss in decibels at distance d, PL_0 is the reference path loss at distance d_0 and n is the path loss exponent. This formula is known as the log-distance path loss model. The path loss exponent is experimentally determined based on measurements. Path loss exponents for different environments are shown in Tab. 2.1.

Extending the model to also include shadowing effects, we get the log-normal path loss model as

$$\mathrm{PL}(d) = \mathrm{PL}_0 + 10n \log_{10}\left(\frac{d}{d_0}\right) + X_\sigma, \tag{2.6}$$

where $X_\sigma \sim \mathcal{N}\left(0, \sigma_s^2\right)$.

2.1.2 Ultra wide-band signals

Ultra wide-band signals are defined by the Federal Communication Commission (FCC) to be signals that have a fractional bandwidth greater than 0.20 or occupies a bandwith of more than 500 MHz [45]. The formula used to calculate the fractional bandwith is

$$\frac{2\left(f_h - f_h\right)}{f_h + f_l}, \tag{2.7}$$

where f_h and f_l are the upper and lower frequency of the -10 dB emission points [45].

2.2 Compressive sensing

In this section the CS framework will be given a short introduction. An in-depth presentation is beyond the scope of this thesis. The reader is referred to some of the

excellent literature on the subject, e.g. the article [15] or the book [46], should more information on the subject be of interest.

Compressive sensing, also known as compressed sensing or compressed sampling, is a framework for recovering compressive signals with fewer measurements than required by the Nyquist-Shannon sampling theorem. First introduced in [47, 48, 49], CS enables direct sensing of a compressive data rather than first perform sampling and subsequently compress the sampled data [46].

The CS theory relies on sparsity in the signal of interest and incoherence in the sensing modality. The definition of sparsity is as follows. A signal $\boldsymbol{x}$ is considered k-sparse if it contains at most k nonzeros, i.e. $\|\boldsymbol{x}\|_0 \leq k$ [46]. Many natural signals are compressible, which can be exploited to give an exact representation of the sparse signal in a proper basis $\boldsymbol{\Psi}$ [15]. Incoherence means that while the original signal is sparse, the representation in $\boldsymbol{\Psi}$ is dense.

The problem of sensing a signal $\boldsymbol{x}$ can be stated as [15]

$$y_k = \langle \boldsymbol{x}, \boldsymbol{\phi}_k \rangle \qquad k = 1, ..., m\,. \tag{2.8}$$

For discrete signals, we have $\boldsymbol{x} \in \mathbb{R}^n$. By denoting $\mathbf{A}$ as a $M \times N$ sensing matrix

$$\mathbf{A} = \begin{bmatrix} \boldsymbol{\phi}_1^* \\ \vdots \\ \boldsymbol{\phi}_m^* \end{bmatrix}, \tag{2.9}$$

where a^* denotes the conjugate transpose of a, the problem can be stated as

$$\boldsymbol{y} = \mathbf{A}\boldsymbol{x} \qquad \boldsymbol{y} \in \mathbb{R}^m\,. \tag{2.10}$$

If signal $\boldsymbol{x}$ is undersampled, with the number of measurements m much less than the dimension of the signal n, the sensing matrix $\mathbf{A}$ reduces the dimensionality of the signal $\boldsymbol{x}$ by mapping it from $\mathbb{R}^n$ to $\mathbb{R}^m$. To recover $\boldsymbol{x}$ from the measurements $\boldsymbol{y}$ one can attempt to solve the optimization problem [46]

$$\boldsymbol{x} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0 \quad s.t. \quad \boldsymbol{y} = \mathbf{A}\boldsymbol{x}\,. \tag{2.11}$$

However, solving Eq. (2.11) require high computational complexity since the objective function is nonconvex [16]. In fact, for the general matrix $\mathbf{A}$, finding a solution approximating the true minimum is NP-hard [46]. A more practical approach to the recovery of $\boldsymbol{x}$ is to consider the convex approximation $\|\cdot\|_1$ of the ℓ_0 norm. Due to convexity, algorithms that estimate the ℓ_1 norm can be implemented as a linear program with polynomial complexity [16].

2.2.1 Coherence

The coherence between a sensing basis $\mathbf{\Phi}$ and the representation basis $\mathbf{\Psi}$ is found by [15]

$$\mu\left(\mathbf{\Phi}, \mathbf{\Psi}\right) = \sqrt{n} \max_{1 \leq k,j \leq n} \left|\langle \phi_k, \psi_j \rangle\right|, \tag{2.12}$$

which measures the largest correlation between elements of $\mathbf{\Phi}$ and $\mathbf{\Psi}$. The value of the coherence will depend on the dimension n of the signal by $\mu\left(\mathbf{\Phi}, \mathbf{\Psi}\right) \in [1, \sqrt{n}]$.

Random matrices have the interesting property of being largely incoherent with any fixed basis $\mathbf{\Psi}$. An orthonormal basis $\mathbf{\Phi}$ has a coherence between $\mathbf{\Phi}$ and $\mathbf{\Psi}$ of approximately $\sqrt{2 \log n}$ [15].

2.2.2 Restricted isometry property

A sensing matrix $\mathbf{A}$ satisfies the resistricted isometry property (RIP) of order k if there exists a $\delta_k \in (0, 1)$ such that [46]

$$\left(1 - \delta_k\right) \|\boldsymbol{x}\|_2^2 \leq \|\mathbf{A}\boldsymbol{x}\|_2^2 \leq \left(1 + \delta_k\right) \leq \|\boldsymbol{x}\|_2^2 . \tag{2.13}$$

For many applications the RIP is fulfilled if number of measurements m obeys

$$m \geq CK \log\left(n/K\right) , \tag{2.14}$$

where K is the sparsity of the signal and C is a constant depending on the matrices [15].

2.3 Dynamic state estimation

A theoretical presentation of the Kalman filter, multi-mode estimation and the particle filter is given in the following sections. A reader familiar with these topics may safely continue with Sec. 2.4.

2.3.1 Kalman filter

The Kalman filter was first introduced by Rudolf Kalman in his famous 1960 paper *A New Approach to Linear Filtering and Prediction Problems* [50]. Over the years, the Kalman filter has evolved to be widely used in practical applications where the state of a process should be estimated. The information in this section is based on the book [51] and the technical report [52].

The Kalman filter considers the problem of estimating the state $\boldsymbol{x}$ of a discrete-time controlled process governed by a linear stochastic difference equation

$$\boldsymbol{x}_k = \mathbf{A}\boldsymbol{x}_{k-1} + \mathbf{B}\boldsymbol{u}_{k-1} + \boldsymbol{w}_{k-1} , \tag{2.15}$$

having measurements $\boldsymbol{z}$ described by

$$\boldsymbol{z}_k = \mathbf{C}\boldsymbol{x}_k + \boldsymbol{v}_k \,. \tag{2.16}$$

$\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are random independent variables representing noise in the process and measurements, with distribution $p(w) \sim \mathcal{N}(0, \mathbf{Q})$ and $p(v) \sim \mathcal{N}(0, \mathbf{R})$ respectively. The covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ can change with time, but will here be assumed constant.

Defining the *a priori* and *a posteriori* estimation errors as

$$\boldsymbol{e}_k^- = \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^- \tag{2.17}$$
$$\boldsymbol{e}_k = \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k \,, \tag{2.18}$$

the estimation error covariances are given by

$$\mathbf{P}_k^- = \mathbb{E}\left[\boldsymbol{e}_k^- \boldsymbol{e}_k^{-\mathrm{T}}\right] \tag{2.19}$$
$$\mathbf{P}_k = \mathbb{E}\left[\boldsymbol{e}_k \boldsymbol{e}_k^{\mathrm{T}}\right] \,. \tag{2.20}$$

The prediction and measurements are related by Eq. (2.26). The factor $\mathbf{K}$ is the Kalman gain which weights the measurements and the predictions according to the covariance error matrices. It can be proven that $\mathbf{K}$ minimizes Eq. (2.20).

The Kalman filter is a two-step process: First, the state of a process at a discrete time is estimated; secondly, the filter gets feedback as noisy measurements. These two steps are known as the *time update* step and the *measurement update* step. The time update step obtains the *a priori* estimate, while the measurement update step uses the current observations to get an improved *a posteriori* estimate. This recursive nature of the Kalman filter makes the implementation compact.

When the matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are constant with time, the Kalman equations are as follows:

Time update step:

$$\text{State estimation prediction:} \quad \hat{\boldsymbol{x}}_k^- = \mathbf{A}\hat{\boldsymbol{x}}_{k-1} + \mathbf{B}\boldsymbol{u}_{k-1} \tag{2.21}$$
$$\text{Error covariance prediction:} \quad \mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^{\mathrm{T}} + \mathbf{Q} \tag{2.22}$$

Measurement update step:

$$\text{Innovation update:} \quad \boldsymbol{\pi}_k = \boldsymbol{z}_k - \mathbf{C}\hat{\boldsymbol{x}}_k^- \tag{2.23}$$
$$\text{Innovation covariance update:} \quad \mathbf{S}_k = \mathbf{C}\mathbf{P}_k^-\mathbf{C}^{\mathrm{T}} + \mathbf{R} \tag{2.24}$$
$$\text{Kalman gain update:} \quad \mathbf{K}_k = \mathbf{P}_k^-\mathbf{C}^{\mathrm{T}}\mathbf{S}_k^{-1} \tag{2.25}$$
$$\text{State estimate observational update:} \quad \hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \mathbf{K}_k\boldsymbol{\pi}_k \tag{2.26}$$
$$\text{Error covariance update:} \quad \mathbf{P}_k = \big(\mathbf{I} - \mathbf{K}_k\mathbf{C}\big)\mathbf{P}_k^- \tag{2.27}$$

It appears to us that there is no consensus on the names of matrices and state vectors in the Kalman literature. We have chosen to use the notation shown in Tab. 2.2.

Table 2.2: Terminology used in this thesis for the Kalman filter.

Symbol	Definition
$\boldsymbol{x}$	State vector
$\boldsymbol{z}$	Observation vector
$\mathbf{A}$	State transition matrix
$\mathbf{B}$	Control input matrix
$\mathbf{C}$	Observation matrix
$\mathbf{P}$	Error estimate covariance matrix
$\mathbf{Q}$	Process noise covariance matrix
$\mathbf{R}$	Observation noise covariance matrix
T	Time step

2.3.2 Variable noise level multi-mode estimation

Hybrid estimation, also known as multi-mode estimation, is defined as estimating the state of a system that has both continuous and discrete components [53], and can e.g. be target tracking. By defining the number of modes as $\mathbb{M} = \{m_1, \ldots, m_n\}$, the movement of the target can be described by choosing the appropriate mode m_i. When tracking a target, the target can perform a maneuver at any given time. The problem then becomes to detect the maneuvers [53].

Two of the simplest methods of multi-mode estimation is variable noise level (VNL) and variable state dimension (VSD) [11]. For a Kalman filter, VNL defines multiple observation noise covariance matrices $\mathbf{Q_1}, \ldots, \mathbf{Q_n}$, corresponding to modes in the filter. Similarly, VSD changes the dimension of the state vector of the system when a mode-change is detected. In this section, the two-mode Kalman VNL filter will be considered, using the presentation from [11].

For a two-mode Kalman filter, we have one observation noise covariance matrix for each mode denoted by $\mathbf{Q}_1$ and $\mathbf{Q}_2$. The filter has to be initialized in one of the modes, for which the corresponding $\mathbf{Q}$ is chosen. The time update step, predicting state and covariance by Eqs. (2.21-2.22), is run first. Then the innovation $\boldsymbol{\pi}_k$ and innovation covariance $\mathbf{S}_k$ is calculated using Eqs. (2.23-2.24).

Having found $\boldsymbol{\pi}_k$ and $\mathbf{S}_k$, the normalized innovation ϵ_k is found if there have been r time steps without mode changes, to check if the mode has changed. ϵ_k is given by

$$\epsilon_k = \boldsymbol{\pi}_k^{\mathrm{T}} \mathbf{S}_k^{-1} \boldsymbol{\pi}_k \,, \tag{2.28}$$

and is affected by the observation noise. Thus, in order to limit the noise influence, a moving average of ϵ_k is used as the test statistic e_k

$$e_k = \sum_{l=0}^{L-1} \epsilon_{k-l} \,. \tag{2.29}$$

The length L of the moving average filter determines how resilient the filter is to noise, but having too large L will cause problems detecting the mode change. The

test statistic e_k is compared to a defined mode-change threshold τ by

$$e_k > \tau \,, \tag{2.30}$$

where a larger e_k indicate that the mode should change. The threshold τ should be chosen with care. Having τ too low will cause constant mode changes, while if τ is too large, the mode will never change.

When a mode change has been detected, the Kalman filter is reset by setting the error covariance matrix $\mathbf{P}$ to

$$\mathbf{P} = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_{q,n}^2 \end{bmatrix} , \tag{2.31}$$

for the case of one-dimensional tracking. $\sigma_{q,n}^2$ is the system noise of the new mode. The observation covariance matrix is also changed to the one corresponding to the new mode, e.g. if we have changed from mode 1 to mode 2 the new covariance matrix is $\mathbf{Q}_2$.

The filter continues in the new mode for r time steps, before reverting back to the other mode and monitoring e_i. r is chosen together with $\mathbf{Q}_2$ to ensure that the filter tracks the maneuver as fast as possible.

2.3.3 Particle filter

In Bayesian non-linear filtering, the particle filter is a relatively new approach. The first traces dates back to the 1950's, but the development of this technique really started with the paper [54] in the 1990's. At the time, the computing power had developed enough to be used for computing complex algorithms. This unleashed a wave of research in this area, especially in the 2000's [55]. The particle filter is also known under the names *bootstrap filtering*, *condensation algorithm*, *interacting particle approximations* and *survival of the fittest* [56].

The particle filter implements a recursive Bayesian filter by use of Monte Carlo (MC) methods. The concept is to represent the posterior density function by a set of random samples, with each of these samples having an associated weight. The estimates are calculated according to the samples and weights. With large number of samples, the estimated posterior probability density function approaches the optimal Bayesian estimate [56].

Compared to the Kalman filter, one of the main differences are that the particle filter is not restricted by assumptions of Gaussian noise. Thus, it may be applied to any state transition and measurement model [54].

Sequential importance sampling (SIS)

The area of particle filtering contains different approaches. The most general algorithm is the sequential importance sampling filter (SIS). The original algorithm, as stated by the authors in the article [54], is shown in the following.

Having a state vector $\boldsymbol{x_k}$ and a set of measurements $D_k = \{\boldsymbol{y_i} : i = 1, \ldots, k\}$ the prior pdf of the state at time step k can be written as

$$p\left(\boldsymbol{x_k} \mid D_{k-1}\right) = \int p\left(\boldsymbol{x_k} \mid \boldsymbol{x_{k-1}}\right) p\left(\boldsymbol{x_{k-1}} \mid D_{k-1}\right) d\boldsymbol{x_{k-1}} \,. \tag{2.32}$$

When a new measurement $\boldsymbol{y_k}$ is available at at time step k, the prior is updated by the Bayes rule

$$p\left(\boldsymbol{x_k} \mid D_k\right) = \frac{p\left(\boldsymbol{y_k} \mid \boldsymbol{x_k}\right) p\left(\boldsymbol{x_k} \mid D_{k-1}\right)}{p\left(\boldsymbol{y_k} \mid D_{k-1}\right)} \,. \tag{2.33}$$

Assuming a set $\{\boldsymbol{x_{k-1}}\,(i) : i = 1, \ldots, N\}$ of random samples from the pdf $p\left(\boldsymbol{x_k} \mid D_{k-1}\right)$ are available, the particle filter obtains a set $\{\boldsymbol{x_k}(i) : i = 1, \ldots, N\}$ approximately distributed as $p\left(\boldsymbol{x_k} \mid D_k\right)$. Thus, the filter approximates the relations in Eq.(2.32-2.33) by the two steps:

Prediction Pass each sample in the set through the system model to obtain samples from the prior at time step k

$$\boldsymbol{x_k^*}(i) = f_{k-1}\big(\boldsymbol{x_{k-1}}(i), \boldsymbol{w_{k-1}}(i)\big) \,, \tag{2.34}$$

where $\boldsymbol{w_{k-1}}(i)$ is a sample drawn from the system noise pdf $p(\boldsymbol{w_{k-1}})$.

Update When a new measurement $\boldsymbol{y_k}$ is available, evaluate the likelihood of each prior sample and obtain a normalised weight

$$q_i = \frac{p\left(\boldsymbol{y_k} \mid \boldsymbol{x_k^*}(i)\right)}{\sum_{j=1}^{N} p\left(\boldsymbol{y_k} \mid \boldsymbol{x_k^*}(j)\right)} \,. \tag{2.35}$$

Sequential importance resampling (SIR)

The SIS particle filter has a common problem called the *degeneracy phenomenon*, where every particle except one has negligible weight. This effect can be solved by resampling the particles, either at each iteration or when a threshold N_{eff} is met [56]. When the resampling is performed in each time step, the filter is known as the sequential importance resampling (SIR) particle filter.

There exist multiple resampling strategies. Some of these will be shown in the following.

Naïve resampling The naive approach is to sample particles at random from the particle weight distribution. Using this approach, there is no guarantee that particles with large weights are sampled [57].

Sampling of cdf A better approach is to generate a cumulative distribution of the particle weights, and sample from this distribuiton [56]. This ensures that particles with large weights have a larger probability of being sampled.

Low variance resampling Yet another solution is to pick a random number to use as the first particle to be resampled. Then the rest of the particles are found from the successive intervals of W/N from this first particle, where W is the sum of particle weights and N is the number of particles [57]. This method is called low variance resampling.

Resampling threshold

Some types of particle filters resample the particles if the number of effective particles N_{eff} is less than a threshold N_{th}. The effective particles can be found by the estimate

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} w_i^2}\,. \tag{2.36}$$

In [56], a reasonable threshold is suggested as $N_{th} = 2N/3$.

2.4 CST Microwave Studio

Computer Simulation Technology (CST) is a German company producing a EM simulation program package called CST Studio Suite. These programs provide numerical simulations for static, low-frequency, RF and charged particles. It also contains design software for producing designs and checking for EMC [58].

Of interest for this thesis is the program CST Microwave Studio (MWS) that has been used for the numerical EM simulations. In this section, a brief description will be given on how the program solves the EM problems numerically. The information is based on the documentation provided with CST MWS [59].

2.4.1 Finite integration technique (FIT)

CST MWS uses a technique called finite integration technique (FIT) to solve Maxwell's equations by discretizing the integral form of the equations

$$\oint_{\partial A} \boldsymbol{E} \cdot d\boldsymbol{s} = -\int_A \frac{\partial \boldsymbol{B}}{\partial t} \cdot d\boldsymbol{A} \tag{2.37}$$

$$\oint_{\partial V} \boldsymbol{D} \cdot d\boldsymbol{A} = -\int_V \rho dV \tag{2.38}$$

$$\oint_{\partial A} \boldsymbol{H} \cdot d\boldsymbol{s} = \int_A \left(\frac{\partial \boldsymbol{D}}{\partial t} + \boldsymbol{J}\right) \cdot d\boldsymbol{A} \tag{2.39}$$

$$\oint_{\partial V} \boldsymbol{B} \cdot d\boldsymbol{A} = 0\,. \tag{2.40}$$

To solve these equations, a finite domain must be defined. This is done by dividing the structure into a suitable primary three dimensional mesh. CST MWS supports

both hexahedral and tetrahedral meshing. In the following the hexahedral mesh will be considered, as this yields the simplest derivation.

In addition to the primary mesh, a secondary mesh grid is defined orthogonal to this grid. By considering Eq. (2.37) on the grid face A_n on one of the hexahedrons, we can write

$$\oint_{\partial A_n} \boldsymbol{E} \cdot d\boldsymbol{s} = -\frac{\partial}{\partial t} \iint_{A_n} \boldsymbol{B} \cdot d\boldsymbol{A} \,. \tag{2.41}$$

The left hand side of Eq. (2.41) can be written as a sum of the four grid voltages on the grid face by Faraday's law

$$e_i + e_j - e_k - e_l = -\frac{\partial}{\partial t} b_n \,. \tag{2.42}$$

Doing this for all the mesh faces, Eq. (2.41) can be rewritten in discrete form as

$$\mathbf{C}\boldsymbol{e} = -\frac{\partial}{\partial t}\boldsymbol{b} \,, \tag{2.43}$$

where the topological matrix $\mathbf{C}$ is equivalent to the curl operator for the discrete domain. Following a similar approach for Eqs. (2.38-2.40), we can obtain the other Maxwell equations in discretized form as

$$\tilde{\mathbf{S}}\boldsymbol{d} = \boldsymbol{q} \tag{2.44}$$

$$\tilde{\mathbf{C}}\boldsymbol{h} = \frac{d}{dt}\boldsymbol{d} + \boldsymbol{j} \tag{2.45}$$

$$\mathbf{S}\boldsymbol{b} = \mathbf{0} \,, \tag{2.46}$$

where $\mathbf{C}$ denotes the discrete curl operator and tilde notes operation on the secondary grid. Eqs. (2.44-2.46) are known as Maxwell's grid equations.

Thus far, no discretization errors have been introduced. But by defining the discrete material relations as

$$\boldsymbol{D} = \epsilon \boldsymbol{E} \quad \Rightarrow \quad \boldsymbol{d} = \mathbf{M}_s \boldsymbol{e} \tag{2.47}$$

$$\boldsymbol{B} = \mu \boldsymbol{H} \quad \Rightarrow \quad \boldsymbol{b} = \mathbf{M}_\mu \boldsymbol{h} \tag{2.48}$$

$$\boldsymbol{J} = \sigma \boldsymbol{E} + \boldsymbol{J}_s \quad \Rightarrow \quad \boldsymbol{j} = \mathbf{M}_\sigma \boldsymbol{e} + \boldsymbol{j}_s \,, \tag{2.49}$$

numerical inaccuracy is introduced due to the spatial discretization. Having obtained all the equations in discrete matrix form, the EM problems can be solved numerically on the discrete grid.

For Cartesian grids, the FIT method can be rewritten in time domain to the finite difference time domain (FDTD) method. However, CST MWS applies a method called perfect boundary approximation (PBA), which gives better accuracy for curved structures than the standard FDTD method.

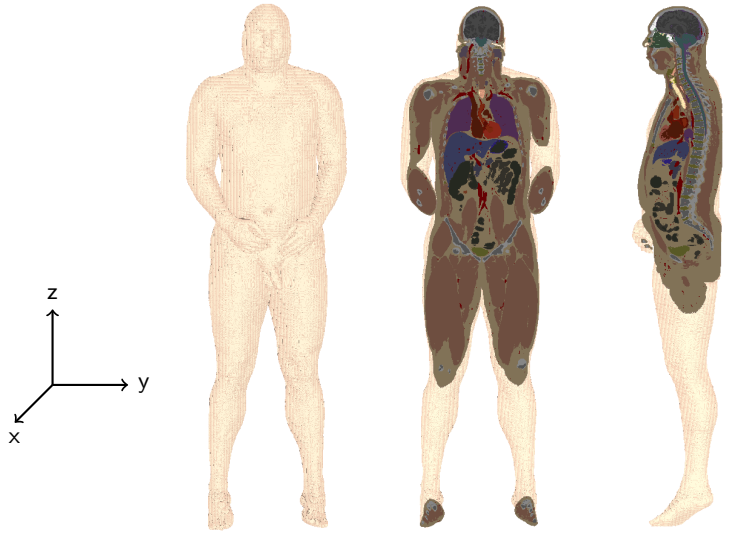


Figure 2.2: The HUGO voxel model, shown from the front and the side.

2.4.2 Transient solver

The transient solver uses Maxwell's grid equations to solve for a wide frequency range with a single simulation run. This is done by replacing the time derivatives with central differences

$$\boldsymbol{e}^{n+1/2} = \boldsymbol{e}^{n-1/2} + \mathbf{M}_{\epsilon}^{-1} \left(\tilde{\mathbf{C}} \mathbf{M}_{\mu}^{-1} \boldsymbol{b}^{n} + \boldsymbol{j}_{s}^{n} \right) \Delta t \tag{2.50}$$

$$\boldsymbol{b}^{n+1/2} = \boldsymbol{b}^{n} - \mathbf{C}^{n+n/2} \Delta t \,, \tag{2.51}$$

where the unknowns are the electric voltages $\boldsymbol{e}$ and the magnetic fluxes $\boldsymbol{b}$. As can be seen from Eqs. (2.50-2.51), the unknowns are located at alternate time steps. Thus, the system is solved for each time step by using the two previous $\boldsymbol{e}$ and $\boldsymbol{b}$.

As the system is conditionally stable, the stability of the system is governed by the stability limit

$$\Delta t \leq \frac{\sqrt{\epsilon \mu}}{\Delta x^{-2} + \Delta y^{-2} + \Delta z^{-2}} \,, \tag{2.52}$$

which has to be fulfilled in each mesh cell.

2.4.3 Visible human project (HUGO model)

CST MWS has functionality for import of 3D models in various formats. One such format is voxel data, which is a three dimensional model built up of volumetric pixels. The only solver that is compatible with voxel data, is the transient solver.

The visible human project has provided a complete anatomical three-dimensional model of the human body [60]. The male model from this project is commonly called HUGO, and will be referred to by this name in this thesis.

The HUGO model is available as a voxel model that can be used with CST MWS, to perform research on human body EM problems. The model is available in resolutions ranging from $8\,\mathrm{mm}^3$ to $1\,\mathrm{mm}^3$, and can be used either partly or as the complete body. In Fig. 2.2, the HUGO model is shown as it appears when being used in CST MWS.

Chapter 3

Method

This chapter consists of the methods used for localization and tracking of the WCE. An overview of the localization and tracking system will be given in Sec. 3.1. The localization framework using compressive sensing will be presented in Sec. 3.2. Sec. 3.3 presents different approaches to the problem of tracking and estimating the distance the WCE has travelled.

Figure 3.1: Overview of the localization and tracking system.

3.1 System overview

The system that is used to solve the problem of this thesis can be characterized as three different modules: a localization module, a tracking module and a distance estimation module. The relationship between these modules are shown in Fig. 3.1.

This illustration shows how a set of RSS measurements $\{\boldsymbol{y^i}\}$ $i = 1, \ldots, P$ is used to estimate the positions $(\hat{x}_i, \hat{y}_i)$. The measurements $\boldsymbol{y^i}$ are typically obtained with sampling rate $T = 1\,\text{s}$, and the localization can either be performed considering each of the measurements in the set separately or simultaneously. The latter approach will be referred to as *multi-localization*.

The tracking system gets position estimates from the localization framework, that is used to refine the positions and estimate the current velocity. In order to perform the tracking, a Kalman and particle filter will be used. The constant velocity model and the RWM movement model are used to represent the capsule movements.

Table 3.1: Notation used for the CS localization framework.

Symbol	Description	Dimensions
N	Number of grid points	
M	Number of measurements	
P	Number of measurement sets	
$\boldsymbol{x}$	Signal of interest	$N \times 1$
$\boldsymbol{y}$	Noisy measurements	$N \times 1$
$\boldsymbol{s}$	Sparse representation of the signal of interest	$N \times 1$
$\boldsymbol{\Phi}$	Measurement matrix	$M \times N$
$\boldsymbol{\Psi}$	Sparsifying matrix	$N \times N$
$\boldsymbol{\Theta}$	$\boldsymbol{\Phi\Psi}$	$M \times N$

The velocity estimates are used to estimate the distance the WCE has travelled from the initial position at $i = 1$ to the final position $i = P$. The distance is found by integrating the velocity.

Having presented an overview of the system components, each module will now be given a detailed presentation in the following sections.

3.2 Localization using compressive sensing

In the following, [17] will be used as the main reference. The terminology that is used in this section is given in Tab. 3.1. An overview of the complete localization system is found in Fig. 3.2.

Localization by compressive sensing is performed by defining a two-dimensional discrete grid of N elements with dimensions $N_x \times N_y$ covering the area of interest. An example grid with $N = 81$ is found in Fig. 3.3. The location of an object in the grid can be modelled as an ideal 1-sparse vector $\boldsymbol{s}$. This $1 \times N$ vector contains a 1 in the index corresponding to the grid where the object is located. The object of the localization algorithm is to estimate this sparse vector $\hat{\boldsymbol{s}}$ through noisy, compressive measurements.

Mathematically the problem is formulated as

$$\boldsymbol{y} = \boldsymbol{\Phi x} = \boldsymbol{\Phi\Psi s} + \boldsymbol{\epsilon} = \boldsymbol{\Theta s} + \boldsymbol{\epsilon} \tag{3.1}$$

where $\boldsymbol{y}$ is a $1 \times M$ vector of noisy measurements, $\boldsymbol{\Phi}$ is a $M \times N$ measurement matrix, $\boldsymbol{\Psi}$ is a $N \times N$ sparsifying basis and $\boldsymbol{\epsilon}$ is measurement noise. The type of measurements in $\boldsymbol{y}$ can for instance be RSS, TOA or AOA, that are measured using M receiving antennas with known spatial positions.

The location of the receivers are mapped to the grid with 1-sparse vectors $\boldsymbol{\phi}_i$ $i = 1, \ldots, M$, where the position of the receiver is defined by a 1. These vectors are combined to the measurement matrix $\boldsymbol{\Phi}$ by

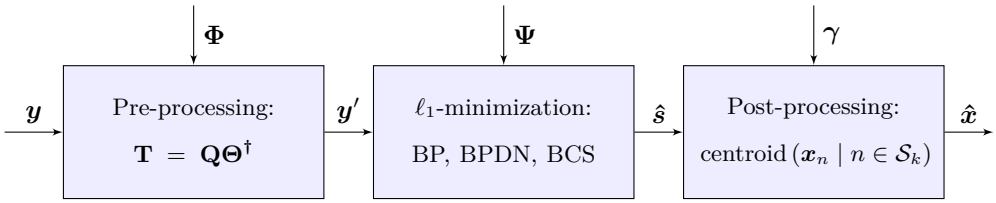


Figure 3.2: Overview of the localization system, after [17].

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}_1 \\ \vdots \\ \boldsymbol{\phi}_M \end{bmatrix}. \tag{3.2}$$

In order to use compressive sensing, a sparsity basis is required in which the location vector $\boldsymbol{s}$ has sparse coefficients, as shown in Sec. 2.2. This matrix will vary depending on the type of measurements performed. For the case of RSS measurements, a sparsity basis $\boldsymbol{\Psi}$ can be found by relating the path loss with distance as

$$\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\psi}_{1,1} & \dots & \boldsymbol{\psi}_{1,N} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\psi}_{N,1} & \dots & \boldsymbol{\psi}_{N,N} \end{bmatrix}, \tag{3.3}$$

where

$$\boldsymbol{\psi}_{i,j} = \mathrm{PL}\left(d_{i,j}\right). \tag{3.4}$$

The notation in Eq. (3.4) describes the path loss experienced from a target located at grid number i to a receiver at grid number j.

3.2.1 Coherence

As presented in Sec. 2.2, the sensing basis $\boldsymbol{\Phi}$ has to have low coherence with the sparsity basis $\boldsymbol{\Psi}$ in order to utilize compressive sensing. As the measurement matrix $\boldsymbol{\Phi}$ and the sparsity matrix $\boldsymbol{\Psi}$ are coherent in the spatial domain, they can not be directly used. One approach to solve this problem, is to explicitly orthogonalize the two matrices. However, in [17] a preprocessing procedure performed on the measurements $\boldsymbol{y}$ is shown to have the same effect as orthogonalizing $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$.

The pre-processing is performed by

$$\boldsymbol{y'} = \mathbf{T}\boldsymbol{y}, \tag{3.5}$$

where the matrix $\mathbf{T}$ is found using

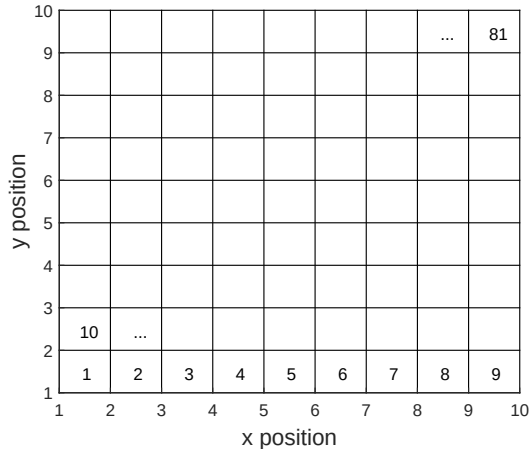


Figure 3.3: An example of a localization grid having $N = 81$.

$$\mathbf{T} = \mathbf{Q}\boldsymbol{\Theta}^{\dagger}. \tag{3.6}$$

In Eq. (3.6), $\mathbf{Q}$ is defined as

$$\mathbf{Q} = \operatorname{orth}\left(\boldsymbol{\Theta}^{\mathrm{T}}\right)^{\mathrm{T}}, \tag{3.7}$$

where the notation $\operatorname{orth}(\mathbf{A})$ denotes an orthogonal basis for the range of $\mathbf{A}$, $\mathbf{A}^{\mathrm{T}}$ is the transpose of $\mathbf{A}$ and † denotes the Moore-Penrose pseudoinverse. Applying $\mathbf{T}$ to the measurements $\boldsymbol{y}$ gives

$$\boldsymbol{y'} = \mathbf{Q}\boldsymbol{\Theta}^{\dagger}\boldsymbol{y} = \mathbf{Q}\boldsymbol{\Theta}^{\dagger}\boldsymbol{\Theta}\boldsymbol{s} + \mathbf{Q}\boldsymbol{\Theta}^{\dagger}\boldsymbol{\epsilon} = \mathbf{Q}\boldsymbol{s} + \boldsymbol{\epsilon'}\,. \tag{3.8}$$

As $\mathbf{Q}$ is an orthonormal matrix, the vector $\boldsymbol{s}$ can be recovered from the processed measurements $\boldsymbol{y'}$ using compressed sensing.

3.2.2 Localization

Different algorithms are used to estimate the sparse location vector $\hat{\boldsymbol{s}}$. These algorithms will be presented in the following. The BCS section is based on [34, 36], with the parts on multi-task BCS found in [32, 35].

Basis pursuit (BP)

The basis pursuit algorithm (BP) is given as [46]

$$\hat{\boldsymbol{s}} = \arg\min_{\boldsymbol{s}} \|\boldsymbol{s}\|_1 \quad s.t. \quad \boldsymbol{y'} = \boldsymbol{\Theta}\boldsymbol{s}\,. \tag{3.9}$$

This algorithm finds the solution with the sparsest ℓ_1 norm.

Basis pursuit denoising (BPDN)

The basis pursuit denoising algorithm (BPDN) is given as [46]

$$\hat{\boldsymbol{s}} = \arg\min_{\boldsymbol{s}} \|\boldsymbol{s}\|_1 \quad s.t. \quad \|\boldsymbol{y'} - \boldsymbol{\Theta s}\|_2 \leq \epsilon\,, \tag{3.10}$$

where ϵ is a scalar value bounding the amount of noise in the data. When $\epsilon = 0$ the algorithm is equal to BP.

Bayesian compressive sensing (BCS)

A regularized formulation of the ℓ_1-minimization problem is stated as

$$\hat{\boldsymbol{s}} = \arg\min_{\boldsymbol{s}} \left\{ \|\boldsymbol{y'} - \boldsymbol{\Theta s}\|_2^2 + \beta \|\boldsymbol{s}\|_1 \right\}, \tag{3.11}$$

where β is a parameter that controls the trade-off between the Euclidean error and sparsity. Assuming that the measurement noise $\boldsymbol{\epsilon}$ from Eq. (3.1) is zero-mean Gaussian noise, we have the Gaussian likelihood as

$$p\left(\boldsymbol{y'} \mid \boldsymbol{s}, \sigma^2\right) = \left(2\pi\sigma^2\right)^{-M/2} \exp\left(-\frac{1}{2\sigma^2} \|\boldsymbol{y'} - \boldsymbol{\Theta s}\|^2\right). \tag{3.12}$$

By establishing a Gaussian mixture to approximate the prior distribution

$$p\left(\boldsymbol{s} \mid \boldsymbol{\lambda}\right) = \prod_{i=1}^{N} \beta_i \mathcal{N}\left(s_i, 0, \lambda_i^{-1}\right), \tag{3.13}$$

the posterior for $\boldsymbol{s}$ can be written as

$$p\left(\boldsymbol{s} \mid \boldsymbol{y'}, \alpha, \boldsymbol{\lambda}\right) = \frac{\left(2\pi\sigma^2\right)^{-M/2}}{\sqrt{\boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2\boldsymbol{\Sigma}} \|\boldsymbol{y'} - \boldsymbol{\mu}\|^2\right), \tag{3.14}$$

with mean and covariance given by

$$\boldsymbol{\mu} = \alpha \boldsymbol{\Sigma} \boldsymbol{\Theta}^{\mathrm{T}} \boldsymbol{y'} \tag{3.15}$$

$$\boldsymbol{\Sigma} = \left(\alpha \boldsymbol{\Theta}^{\mathrm{T}} \boldsymbol{\Theta} + \boldsymbol{\Lambda}\right)^{-1} \tag{3.16}$$

$$\boldsymbol{\Lambda} = \mathrm{diag}\left(\lambda_1, \ldots, \lambda_N\right). \tag{3.17}$$

To use Eqs.(3.15-3.17), estimates of the hyperparameters $\boldsymbol{\Lambda}$ and α must be found. This can be done by the expectation maximization algorithm as

$$\hat{\lambda}_i = \frac{\rho_i}{\mu_i^2} \qquad i \in \{1, \ldots, N\} \tag{3.18}$$

$$\hat{\alpha} = \frac{M - \sum_i \rho_i}{\|\boldsymbol{y'} - \boldsymbol{\Theta \mu}\|_2^2}\,, \tag{3.19}$$

where $\rho_i = 1 - \lambda_i \Sigma_{ii}$. When the expectation maximization algorithm converges, we have found the sparse vector $\hat{\boldsymbol{s}}$ as $\boldsymbol{\mu}$ with covariance $\boldsymbol{\Sigma}$.

Multi-task BCS By extending the localization problem to multi-localization, where we estimate a set of localization vectors $\{\boldsymbol{s}^i\}_{i=1,\ldots,P}$ from a set of measurements $\{\boldsymbol{y}^i\}_{i=1,\ldots,P}$, Eq. (3.1) is expanded to

$$\boldsymbol{y}^i = \boldsymbol{\Theta}^i \boldsymbol{s}^i + \boldsymbol{\epsilon}^i \,. \tag{3.20}$$

Following a similar approach as in Sec. 3.2.2, but where the hierarchical prior is established based on all the measurements in the set, we have

$$\boldsymbol{\mu}^i = \alpha \boldsymbol{\Sigma}^i \boldsymbol{\Theta}^{i\mathrm{T}} \boldsymbol{y'}^i \tag{3.21}$$

$$\boldsymbol{\Sigma}^i = \left(\alpha \boldsymbol{\Theta}^{i\mathrm{T}} \boldsymbol{\Theta}^i + \boldsymbol{\Lambda}\right)^{-1} \tag{3.22}$$

$$\boldsymbol{\Lambda} = \mathrm{diag}\left(\lambda_1, \ldots, \lambda_N\right). \tag{3.23}$$

Estimates for the hyperparameters are found by

$$\hat{\lambda}_j = \frac{P - \lambda_j \sum_{i=1}^{P} \Sigma_{j,j}^i}{\sum_{i=1}^{P} \left(\mu_j^i\right)^2} \qquad j \in \{1, \ldots, N\} \tag{3.24}$$

$$\hat{\alpha} = \frac{\sum_{i=1}^{P} \left(M^i - N + \sum_{j=1}^{N} \lambda_j \Sigma_{j,j}^i\right)}{\sum_{i=1}^{P} \left\|\boldsymbol{y'}^i - \boldsymbol{\Theta}^i \boldsymbol{\mu}^i\right\|_2^2} \,. \tag{3.25}$$

Eqs.(3.21-3.25) are solved iteratively until convergence. The set of estimated localization vectors $\hat{\boldsymbol{s}}^i$ is found as $\boldsymbol{\mu}^i$ with covariance $\boldsymbol{\Sigma}^i$.

3.2.3 Post-processing of sparse vector

Assuming no measurement noise, and having all receivers and the target centered on the grids, the estimated location vector $\hat{\boldsymbol{s}}$ will be 1-sparse. However, this will not always be true for real localization problems. Due to the discretizing operation of the grid, the difference in positions between the true position and the grid position cause the localization vector $\hat{\boldsymbol{s}}$ to contain a few non-zero elements.

To compensate for this error, the centroid of the points can be calculated [17]. This is done by choosing a set $\mathcal{S}_k$ of indices from $\hat{\boldsymbol{s}}$ larger than γ

$$\mathcal{S}_k = \{n \mid \hat{\boldsymbol{s}}(n) > \gamma\} \,. \tag{3.26}$$

Each $n \in \mathcal{S}$ corresponds to a two-dimensional position $\boldsymbol{x}_n = (x_n, y_n)$. The position estimate $\hat{\boldsymbol{x}} = (x, y)$ is then found by

$$\hat{\boldsymbol{x}} = \mathrm{centroid}\left(\boldsymbol{x}_n \mid n \in \mathcal{S}_k\right). \tag{3.27}$$

3.3 Tracking and distance estimation

The problem of estimating the distance $\hat{d}$ the WCE has travelled between a set of two-dimensional positions $\{p_i\}$ $i = 1, \ldots, n$ can be solved by:

1. Integrating the velocity v between the points:

$$\hat{d} = \int_{p_1}^{p_n} v \, dv \,. \tag{3.28}$$

2. Summing the Euclidean distances between each position:

$$\hat{d} = \sum_{i=m}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \,. \tag{3.29}$$

As Eq. (3.29) is only dependent on the positions of the WCE, this is the simplest method to implement, as the positions are already available. However, this solution is not suitable for use with a WCE. The reason for this, is that the capsule stops for a period of time when there are bends in the intestine. Thus, small errors in the position estimates will accumulate to errors in the distance that cause the estimated length $\hat{d}$ to be longer than the true length d.

The capsule velocity is needed in order to use the solution with Eq. (3.28). Due to power consumption, it is not preferable to have sensors measuring the velocity of the WCE. Thus, the velocity has to be estimated from the observed positions. Having available prior information about the previous positions, using the Bayesian approach would be a good choice for such algorithm [61].

3.3.1 Tracking

The general problem of estimating the state $\boldsymbol{x_k}$ using Bayesian dynamic state estimation is given as [52]

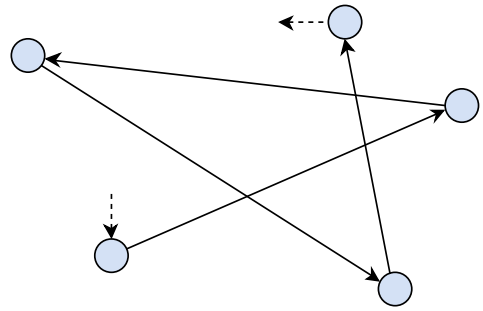
$$\boldsymbol{x_k} = f\left(\boldsymbol{x_{k-1}}, \boldsymbol{u_{k-1}}, \boldsymbol{w_{k-1}}\right) \tag{3.30}$$

$$\boldsymbol{z_k} = h\left(\tilde{\boldsymbol{x}}_{\boldsymbol{k}}, \boldsymbol{v_k}\right), \tag{3.31}$$

where f is a function relating the previous state $\boldsymbol{x_{k-1}}$ with the current state $\boldsymbol{x_k}$ and h is a function that relates the state $\boldsymbol{x_k}$ with the measurements $\boldsymbol{z_k}$. Both these functions can be either linear or non-linear, depending on the dynamic of the system. $\boldsymbol{u_k}$ denotes a vector of known control inputs, while $\boldsymbol{w_k}$ and $\boldsymbol{v_k}$ are process and measurement noise, respectively.

Of the available options for Bayesian dynamic state estimation, the Kalman filter from Sec. 2.3.1 is commonly used for problems with linear dynamics. For problems that are non-linear, the extended Kalman filter and the particle filter (Sec. 2.3.3) are good candidates. The articles presented in Sec. 1.2.3 all used Kalman and particle filters for tracking. Thus, it was decided to use the Kalman and particle filter for the state estimation.

(a) The intestines from the HUGO model. It can be seen that the small intestine has many bends, and exhibits an almost random three-dimensional spatial extent.



(b) The shape of a simplified motion modelling the WCE movement. The blue circles represent areas in the intestine where the capsule is not moving.

Figure 3.4: Three-dimensional voxel model of the intestines compared to a simplified movement model.

3.3.2 Movement models

In order to perform tracking on a system, a model describing the functions f and h needs to be found. In Fig. 3.4a the GI tract of the HUGO model is shown. By simplifying, the two-dimensional movement of a WCE in the intestines can typically be modelled as in Fig. 3.4b. Here we see that the path consist of different segments simplified to be straight lines. Between these segments the capsule stops for a period of time, due to bends in the intestine.

With the described motion, it is apparent that a corresponding motion model needs to handle start/stops and abrupt changes in direction, and that both acceleration and velocity will change with time. However, it seems reasonable to assume that the velocity is constant when the WCE has gained speed and is travelling in areas where the intestine is more or less homogenous.

The problem of modelling the movement of a WCE in the intestines has been considered in [11] and [12]. The author of [11] shows that the capsule movement is totally governed by the stress and strain cycle of the intestine, which can be considered constant under normal conditions, and that the main factors determining the speed is the diameter of the capsule and the intestines. While the capsule diameter is constant, the inner diameter of the gastrointestinal tract was in [62] found to change for different parts of the intestines, causing small changes in velocity. By simplifying the model as having constant inner diameter and that the intestine consists of N segments, the capsule speed at time step k can be approximated as the following Gaussian mixture distribution [11]

$$v_k \approx \sum_{n=-N}^{N} p(n)\,\mathcal{N}\left(v_k \mid v_n, q_n\right), \tag{3.32}$$

where $\mathcal{N}(v_k \mid v_n, q_n)$ is a Gaussian pdf with mean v_n covariance q_n and probability $p(n)$. What Eq. (3.32) represents is that the capsule at timestep k has its velocity drawn from $\mathcal{N}(v_n, q)$, and when the capsule reaches a bend in the intestine the speed drops to zero.

This model can be implemented as a constant velocity dynamic model, having multiple movement states and system noise compensating for the slowly changing velocity.

In [12], the RWM movement model is applied to model the capsule movement. This model describes the movement as a random straight line between the current position at timestep k to a new random position at timestep $k+1$, in which the capsule moves with a random constant velocity. When arriving at the position chosen for $k+1$, a new position and velocity is drawn [63].

Having reviewed these two movement models, it was decided to implement both to test how well they worked for tracking the position datasets used in this thesis. The RWM model do not provide a velocity estimate, and will thus only be used for position estimation with a particle filter. The constant velocity model will be used with both Kalman and particle filters, with the multi-mode model being used with the VNL Kalman filter.

Chapter 4

Implementation

In this chapter, implementation details will be presented. In Sec. 4.1, the setup used to perform the in-body EM simulations is described. Sec. 4.2 presents transmitter and receiver antenna configurations used with four different EM simulation experiments. Implementation details of the localization algorithm is provided in Sec. 4.3. At the end of the chapter, in Sec. 4.4, the implementation of the Kalman and particle filters are shown.

4.1 In-body numerical EM simulations

The purpose of the in-body EM simulations were to obtain power measurements to use with the localization algorithm. The software chosen for this task was CST MWS, as this program supports human voxel models and was found to have the required features. From the different human models available with CST MWS, the HUGO model was chosen due to licensing reasons.

4.1.1 Simulation framework

In order to make the simulation setup reliable, with ability to recreate all simulation results and avoid doing excessive repetitive tasks in the graphical user interface (GUI) of CST MWS, it was decided to generate and control all the simulations by scripting. This can be done to some degree, using macros programmed in Visual Basic for applications (VBA). However, this solution was found to be limited and causing redundant program code.

CST MWS can also be controlled from other applications using the component object model (COM) in Windows. The CST MWS COM-object contains all the functionality of the program. As this solution poses no restriction on the possibilities, it was chosen to control the simulations by designing a simulation framework encapsulating the needed functionality in CST MWS. VBScript was chosen as the language for the simulation framework, as this language is readily compatible with COM-objects. The syntax is also similar to VBA.

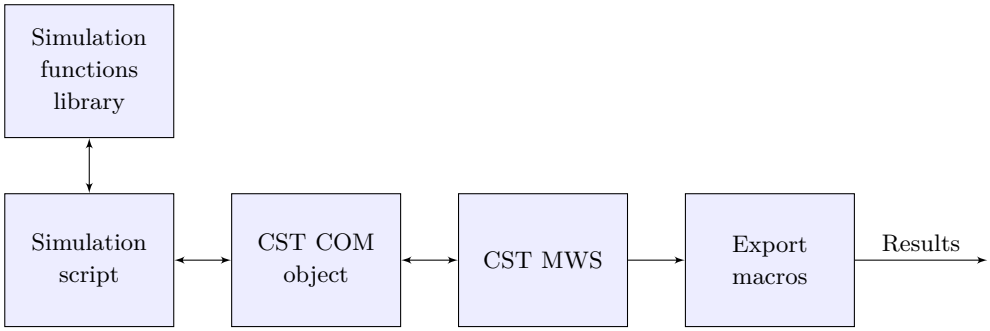


Figure 4.1: Overview of the EM simulation setup.

The simulation framework consists of a library of functions that is used to generate antennas, control simulation settings, import the HUGO model with correct material properties and import antenna coordinates from text files. To design this framework, the extensive program documentation in CST MWS [59] was used, where the objects that can be accessed are described. When a 3D model has been created using the GUI, it is also possible to find the VBA code that was used for the generation of this model. For the functions that produces antennas, all dimensions were parametrized to enable later adjustment and tuning of the performance.

To perform localization using the simulation results, power data had to be exported from CST MWS into MATLAB. This was realized using an export macro to generate a delimited text file with the power data from all the ports in the simulation for the frequencies of interest. Information about the simulation setup was also included in this file, in order to easily access the simulation settings at a later time. Similarly, a macro exporting all S-parameters was created.

A block diagram depicting the simulation setup is shown in Fig. 4.1. In Appx. C, examples showing selected parts of the simulation framework, the export file format and one of the scripts used for generating simulation results can be found.

4.1.2 Frequency-dependent electric properties of tissues

When performing EM simulations, the properties of the materials used in the three-dimensional model are needed in order to provide an accurate result. Of most importance for the simulations conducted in this thesis are the permittivity and conductivity of the material, as these properties will determine the attenuation of the RF power transmitted from the WCE.

For human tissues, the electromagnetic properties depend largely on frequency. For narrow-band simulations, using a constant permittivity and conductivity will give accurate results. But for UWB applications, the model must account for the frequency variation [64]. This can be done using complex permittivity, which is expressed as [65]

$$\epsilon_c = \epsilon + \frac{\sigma}{j\omega} = \epsilon' - j\epsilon'', \tag{4.1}$$

where ϵ is the permittivity, σ is the conductivity and ω is the frequency in radians.

When measuring the the permittivity of materials, often the relative permittivity ϵ_r is used. This parameter is defined as $\epsilon_r = \epsilon_c/\epsilon_0$, where ϵ_0 is the vacuum permittivity. Another useful relation is the loss tangent, defined as [65]

$$\tan \delta_c = \frac{\epsilon''}{\epsilon'} = \frac{\sigma}{\omega\epsilon} \,. \tag{4.2}$$

When specifying the electromagnetic properties of a material, the parameters provided vary from source to source. However, the most commonly used parameters are ϵ_r, $\tan\delta$ and μ_r.

For human tissues, precise frequency dependent electromagnetic properties have been provided by Gabriel in [66] based on four-pole Cole-Cole equations. However, the complexity of the Cole-Cole model will cause long simulation time when used with in EM simulations [67]. Simplified models are used by fitting the Gabriel data in the wanted frequency range [68]. One such model is the two-pole Debye model, which has been adapted to the Gabriel data in [64].

The HUGO model is provided with electromagnetic properties for the materials it contains, such as fat, blood and the other organs. These data originate from Gabriel, but are only valid for a narrow frequency band. In order to get accurate results for the UWB simulations, frequency dependent permittivity and conductivity must to be used.

In CST MWS this can be done either by specifying the two-pole Debye model parameters for each material, or provide a list of frequencies with the corresponding permittivity and loss tangent. The program will then do a curve-fit in the frequency range used in the simulation.

As the authors in [64] only provided their Debye-model parameters for some selected tissues, it was chosen to see if the accuracy using the the curve fitting in CST was sufficient. Using the web page [69] the Gabriel data for all materials can be exported in CSV-format for selected frequencies. This was done for the range 0.5-5 GHz in 500 MHz steps, and imported into CST MWS using the simulation framework. The error between the complex permittivity provided and the 2nd order fit was found to be sufficiently small, with only a few percent deviation.

The names of the tissues used in the HUGO model are not identical with those used by Gabriel. Thus, this matching was done by comparing the names and the permittivity on a common frequency. The mapping used is shown in Tab. A.6 in Appx. A. No match was found for the tissue called IntestinalContents, therefore it was decided to use the same properties as SmallIntestine. This was justified with the difference in permittivity being small, and the fact that when a WCE examination is performed, food has been removed from the intestines by use of ingestible agents [4].

4.1.3 Excitation signal

The 2nd-derivative of the Gaussian pulse was used when performing the simulations to acquire the data sets, which is given as [70]

$$y(t) = -A\left(-\frac{t^2}{\sqrt{2\pi}\sigma^5} - \frac{1}{\sqrt{2\pi}\sigma^3}\right)\exp\left(-\frac{t^2}{2\sigma^2}\right), \tag{4.3}$$

with amplitude spectrum

$$|Y(f)| = A(2\pi f)^2 \exp\left(-\frac{(2\pi f\sigma)^2}{2}\right). \tag{4.4}$$

In Eqs. (4.3) and (4.4), A is the amplitude of the pulse, t is the time in seconds and σ is a time constant that controls the spectral bandwidth [71]. σ was chosen to meet the wanted spectral bandwidth for the simulation.

The Gaussian pulse was chosen due to its properties of having a selectable spectral bandwidth that could be used for both NB and UWB simulations. The 5th derivative pulse is often used for UWB problems in the 3-10 GHz range [70, 71]. However, it was found that the 3nd derivative Gaussian pulse provided better properties in the frequency range between 1-3 GHz.

As the derivative properties of the antenna transforms the transmitted pulse to a higher order derivative, it was chosen to use the 2nd derivative Gaussian pulse for the excitation signal.

4.1.4 Tissue-box model

Due to the lossy tissues in the human body, the effective wavelength λ_{eff} of in-body antennas are described by [72]

$$\lambda_{\text{eff}} = \frac{\lambda_0}{\text{Re}\left(\sqrt{\epsilon_c/\epsilon_0}\right)}, \tag{4.5}$$

where λ_0 is the free-space wavelength, ϵ_c is from Eq. (4.1) and ϵ_0 is the vacuum permittivity. Depending on the tissues, this can cause large changes in the resonating frequency of the antenna. In [73] it is shown that an antenna resonates at 25 GHz in free space, with in-body resonance at 4 GHz.

Thus, in order to simulate in-body antennas, an appropriate surrounding material must be used. For the antenna simulations, this was done by inserting the antenna into a $50\,\text{mm}^3$ box consisting of lossy intestinal tissue, resembling the in-body scenario. The size was chosen as a compromise between simulation speed and the dimensions of the intestine, as simulation performance was critical when the antennas were optimized for best performance.

When proper performance had been obtained, the tissue-box model was extended to also include a $200\,\text{mm}^3$ box of lossy fat tissue surrounding both the antenna and the intestine-box.

4.1.5 Meshing of model

The meshing of a simulation model is critical for the accuracy of the results, because this governs the three-dimensional domain the EM fields are calculated in, as presented in Sec. 2.4. However, the computational complexity scales linearly with number of meshcells [59]. Thus, a compromise between accuracy and complexity must be made.

CST MWS has an option for automatic meshing, where the required mesh resolution is found. This works well for small models, but introduce huge number of meshcells in larger models. Due to this, a local, dense mesh was defined for the antennas in areas with fine details. The required resolution was found by empirically comparing the results using this local mesh with the results found by the automatic meshing. In this process, the results were also compared with the results obtained from the frequency solver. In the end, this procedure was found to yield reliable results, with a number of mesh cells that could be handled on the computer that performed the simulations.

Similarly, in order to get accurate results and realistic simulation times for the simulations performed on the human voxel model, a local mesh surrounding the voxels were defined. The resolution of this mesh was chosen equal to the resolution of the voxel model. As each voxel is homogeneous, it seemed reasonable to not have a denser mesh than the voxel resolution. This meshing was defined with lower priority than the local antenna meshing, to keep the finer mesh setting at the in-body antenna intact.

4.1.6 Simulation time

The simulation time t_s is proportional to the number of simulated timesteps N_t. For a simulation, the number of timesteps can be found by

$$N_t = \frac{t_{\text{end}}}{\Delta t}\,, \tag{4.6}$$

where Δt is the minimum stable timestep from Eq. (2.52) and t_{end} is found by

$$t_{\text{end}} = t_1 + t_2\,. \tag{4.7}$$

In Eq. (4.7), t_1 is the duration of the excitation pulse and t_2 is the duration of the transient field analysis. t_2 is determined by either checking if the field energy in the structure has decreased below a certain level, or if the simulation has lasted for $N_{t,\max}$ time steps. By default the energy limit is -30 dB, while time steps are set to $N_{t,\max} = 20t_1/\Delta t$.

For the simulations that is performed in this thesis, if these criteria are used, the entire signal will not reach the receiver antennas. To ensure that the transmitted signal reaches the receiver antennas, the energy limit was set to "No check" to make t_2 the governing criterion. To find an approximate time for t_2, the phase velocity was found by

$$v_p = \frac{\sqrt{2}c}{\sqrt{\epsilon' + \sqrt{\epsilon'^2 + \epsilon''^2}}}\,. \tag{4.8}$$

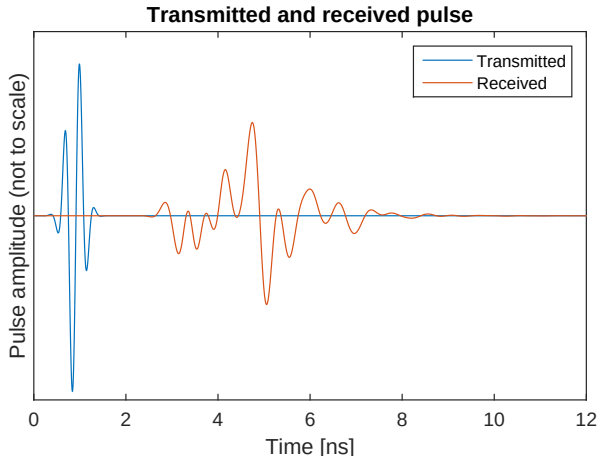


Figure 4.2: The transmitted and received pulse for one of the simulations. The amplitude of the received pulse is in steady state at zero after $t = 10\,\text{ns}$. Note that the scale between the two pulses is modified, as the amplitude of the received pulse is many orders of magnitude smaller.

As the frequency dependent permittivity varies from tissue to tissue, the material having the slowest minimum phase velocity was used together with the maximum transmitter-receiver distance to find an estimate of t_2.

The final t_2 used for the simulations was increased by a small amount, to have a little margin. After the simulation had finished, plots of the received pulses were examined, to ensure that all the energy of the transmitted pulse incident on the receiver antennas had been accepted. This process is visualized in Fig. 4.2, where it can be seen that transmitted pulse is in steady-state at $t = 12\,\text{ns}$ with an amplitude of zero.

4.1.7 Optimization

After modelling of the antennas had been performed, and the results were found to be close to the wanted properties, the optimizer in CST MWS was used to improve the performance. The optimizer was used to adjust the dimensions of the antennas, keeping the number of parameters on a reasonable level.

CST MWS contains multiple optimization algorithms, which can be used for global or local optimization problems. For the initial optimization, the global optimizers *genetic algorithm* and *particle swarm* were used. The fine tuning was performed by the local algorithm *trust region framework*.

The optimization goals were specified as certain return loss level. For all antennas, a minimum return loss of -10 dB was required in the frequency range. For the NB antennas, a goal of $S_{11} \leq -50\,\text{dB}$ for the center frequency of the antenna was employed. For the UWB antennas, due to the wide bandwidth, moderate goals of $S_{11} \leq -20\,\text{dB}$ for a selection of frequencies in the range were used. For most antennas, the required RL of -10 dB was fulfilled, with the performance approaching or exceeding the specified optimization goal.

4.1.8 S-parameter compensation

Ideally, both sender and receiver antennas should have a uniform return loss characteristic over the frequencies of interest. This ensures satisfying power transfer between the antennas. A flat, uniform return loss can be challenging to achieve, especially for UWB antennas. However, if the S-parameters of the antennas are known, it is possible to compensate for this mismatch [74]. For a transmitting antenna, the power $\tilde{P}_t$ coupled from the antenna to the surrounding medium can be expressed as

$$\tilde{P}_t = P_t \left(1 - |S_{11}|^2\right), \tag{4.9}$$

where P_t is the power delivered to the antenna from the feed. With large mismatches in the impedance, $\tilde{P}_t$ will differ significantly from P_t and cause inaccurate description of S_{21}. This can be corrected by using Eq. (4.9), $P_r = P_t \left|S_{21}\right|^2$ and the definition of S_{21}

$$\left|\tilde{S}_{21}\right|^2_{S_{11}} = \frac{P_r}{\tilde{P}_t} = \frac{\left|S_{21}\right|^2}{1 - \left|S_{11}\right|^2}. \tag{4.10}$$

Similarly, the return loss of the receiving antenna can be compensated by

$$\left|\tilde{S}_{21}\right|^2_{S_{22}} = \frac{\tilde{P}_r}{P_t} = \frac{\left|S_{21}\right|^2}{1 - \left|S_{22}\right|^2}, \tag{4.11}$$

where $\tilde{P}_r = P_r / \left(1 - |S_{22}|^2\right)$ has been used. Taking into account both mismatch at the transmitter and receiver, the expression for S_{21} becomes

$$\left|\tilde{S}_{21}\right|^2_{S_{11},S_{22}} = \frac{\tilde{P}_r}{\tilde{P}_t} = \frac{\left|S_{21}\right|^2}{\left(1 - \left|S_{11}\right|^2\right)\left(1 - \left|S_{22}\right|^2\right)}. \tag{4.12}$$

4.2 In-body EM simulation experiments

When deciding the frequency bands to use for the EM simulations, the actual narrowband candidates were limited to the 400 MHz medical implant communication service (MICS) band and the industrial, scientific and medical (ISM) bands of 433 MHz, 915 MHz and 2400 MHz [75]. The MICS band has already been used for WCE localization in [29], while the 433 MHz band is used in the commercial capsules from Given Imaging and Olympus Medical Systems [76]. Thus, it was found of interest to investigate the other bands of 915 MHz and 2400 MHz that has been given less attention.

The IEEE 802.15.6 standard defines two frequency bands for UWB usage [77]. The *low-band* covers the frequency range from 3.25-4.75 GHz in three channels, while the *high-band* covers 6.25-9.75 GHz in seven channels. Of these, the low-band was found to be of most interest, as frequency dependent in-body path loss cause a very large signal attenuation for signals in the high-band spectrum [74]. As even the low-band

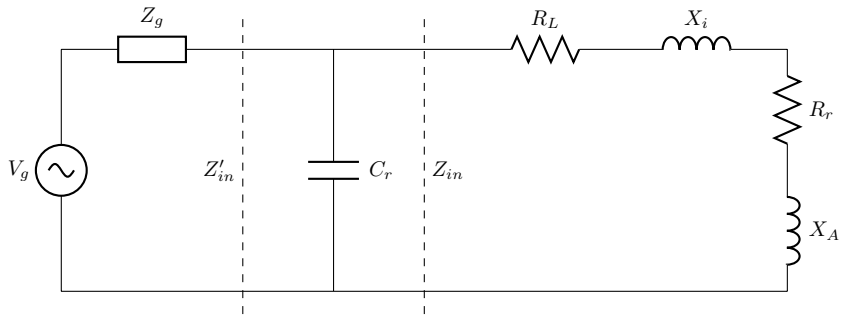


Figure 4.3: Equivalent circuit for a transmitting coil antenna, after [78].

might have too much attenuation, it was decided to also go beyond these bands and explore the spectrum from 1-3 GHz.

For the above reasons, these four simulation setups were chosen:

1. Narrow-band 915 MHz
2. Narrow-band 2400 MHz
3. Ultra wide-band 1-3 GHz
4. Ultra wide-band 3-5 GHz

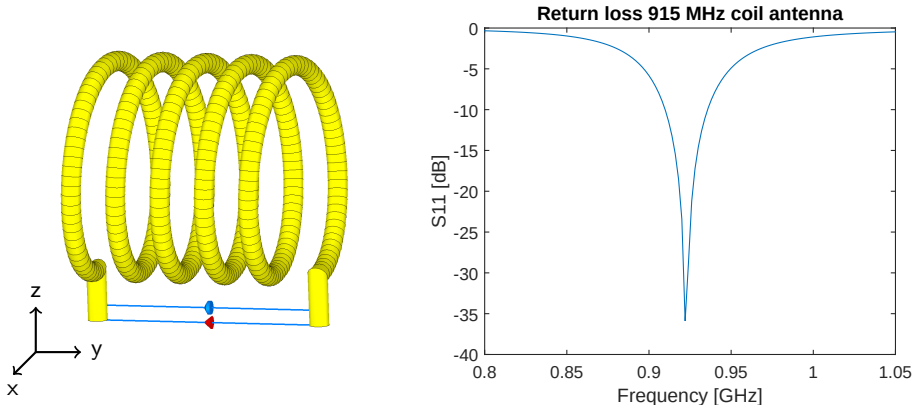
In the following, the transmitter and receiver antennas used for each setup will be explained in detail. With the antennas merely being a part of the simulation setup, existing antenna solutions were found and modified, rather than designed from scratch.

4.2.1 Narrow band 915 MHz

Transmitter antenna: Coil

For the 915 MHz band, multiple options were reviewed for use as the transmitter antenna. The article in [79] presents a planar inverted F-antenna designed for operation between 915-928 MHz. This antenna a general design for in-body devices. The authors discuss how different parameters of the antenna affects the performance, and are able to make a design that has a bandwidth of 8.5 % in simulations. However, no measurements and directivity claims are made.

In [80] a coil antenna for 915 MHz capsule endoscopy operation is presented. The antenna consists of a multi-turn loop in series with a matching network to match the reactive antenna impedance. In simulations, the antenna has a reflection coefficient $|\Gamma|$ of less than 0.2 in the frequency range between 0.8-1.0 GHz. This corresponds to maximum -14 dB return loss. No details about the matching are given, except that the antenna is matched to a 200 Ω circuit. The antenna is measured in a human-body environment, consisting of tissue and skin samples. Measurements with the antenna implanted at a depth of 12.5 cm showed maximum transmission loss of 60.3, 58.4 and 63.7 dB in the capsule oriented along the x-, y- and z-direction.



(a) The antenna design. **(b)** The return loss.

Figure 4.4: Details of the 915 MHz coil antenna design.

The antenna in [80] was chosen as the transmitter, due to the promising performance and the fact that this antenna is similar to antennas used in commercial WCEs. The final antenna design can be seen in Fig. 4.4a, with the return loss shown in Fig. 4.4b.

In order to resonate the loop, a capacitor had to be used in series with the antenna. The authors did not present any information about the matching network used or the capacitor. Thus, the calculations needed to find the proper capacitance and input impedance will be presented in the following, based on [78].

For a loop in transmitting mode, an equivalent circuit can be created consisting of lumped elements. Such a circuit can be seen in Fig. 4.3. Using this circuit, the antenna input impedance Z_{in} can be expressed as

$$Z_{in} = R_{in} + jX_{in} = (R_r + R_L) + j\left(X_A + X_i\right) , \tag{4.13}$$

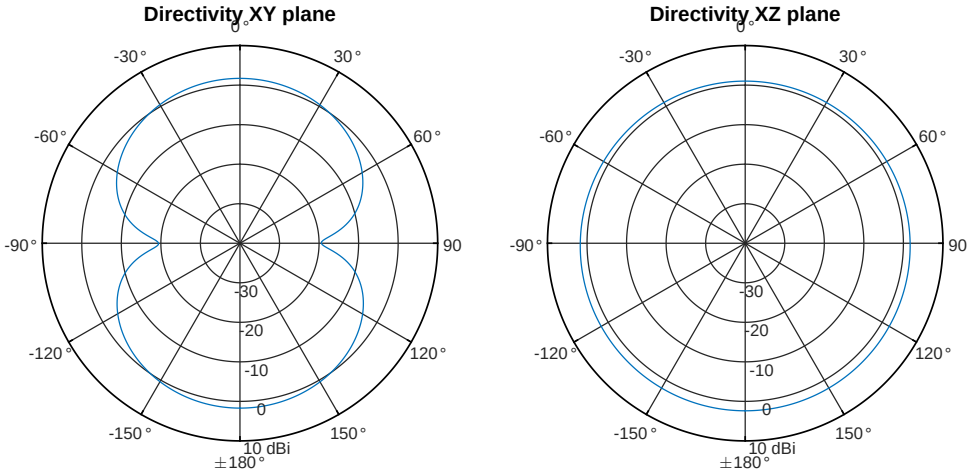
where R_r is the radiation resistance of the antenna, R_L is the loss resistance of the conductor in the loop, X_A is the external inductive reactance of the loop and X_i is the internal reactance of the loop conductor. The capacitor C_r is used to resonate the loop. The capacitance needed for this condition can be found by using the input admittance

$$Y_{in} = G_{in} + jB_{in} = \frac{1}{Z_{in}} = \frac{1}{R_{in} + jX_{in}} , \tag{4.14}$$

where G_{in} and B_{in} are

$$G_{in} = \frac{R_{in}}{R_{in}^2 + X_{in}^2} \tag{4.15}$$

$$B_{in} = -\frac{X_{in}}{R_{in}^2 + X_{in}^2} . \tag{4.16}$$



(a) Directivity in the azimuth plane. **(b)** Directivity in the elevation plane.

Figure 4.5: Directivity for the 915 MHz coil antenna.

At resonance, the suceptance B_r of the capacitor C_r has to eliminate the imaginary part of B_{in}. This is achieved when

$$C_r = \frac{B_r}{2\pi f} = \frac{B_{in}}{2\pi f} = \frac{1}{2\pi f}\frac{X_{in}}{R_{in}^2 + X_{in}^2}\,. \tag{4.17}$$

When resonating, the input impedance becomes

$$Z'_{in} = R'_{in} = \frac{1}{G_{in}} = \frac{R_{in}^2 + X_{in}^2}{R_{in}} = R_{in} + \frac{X_{in}^2}{R_{in}}\,. \tag{4.18}$$

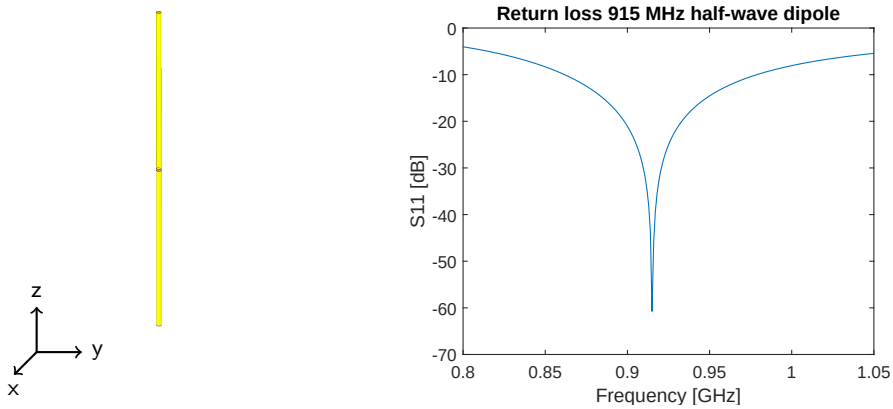
The loss resistance R_l of the loop is found using the formula [78]

$$R_L = \frac{Na}{b} R_s \left(\frac{R_p}{R_0} + 1\right)\,, \tag{4.19}$$

where a is the loop radius, b is the wire radius, $R_s = \sqrt{\omega\mu_0/2\sigma}$ is the surface impedance of the conductor, R_p is the ohmic resistance due to proximity effect and R_0 is ohmic skin effect resistance. The ratio R_p/R_0 can be found from the graph in [78, p.240] which relates the ohmic resistance due to the proximity effect with coil loop spacing.

Using the conductivity of copper as $\sigma = 5.8 \times 10^7\,\mathrm{S\,m^{-1}}$, $f = 915\,\mathrm{MHz}$ and $\mu_0 = 4\pi \times 10^{-7}\,\mathrm{N\,A^{-2}}$, we find the surface impedance as $R_s = 0.0079\,\Omega$. With the loop dimensions $a = 2.1\,\mathrm{mm}$, $b = 0.4\,\mathrm{mm}$ and finding $R_p/R_0 = 0.25$, the loss resistance becomes $R_L = 0.5179\,\Omega$.

The radiation impedance R_r of a N turn loop antenna can be found by the formula [78]



(a) The antenna design. **(b)** Return loss.

Figure 4.6: Details of the 915 MHz half-wave dipole antenna design.

$$R_r = \eta \left(\frac{2\pi}{3}\right) \left(\frac{kS}{\lambda}\right)^2 N^2 \,, \tag{4.20}$$

where η is the wave impedance and S is the area of the coil. η can be found from [65]

$$\eta = \sqrt{\frac{j\omega\mu}{\sigma + j\omega\epsilon}} \,. \tag{4.21}$$

By using the permittivity and conductivity for the intestine at 915 MHz as $\epsilon = 59.47$ and $\sigma = 2.17$ found from the data in [66], we get $\eta = 41.93 + j13.50$. As only an approximate solution is needed, the imaginary part is discarded. Thus, we find the radiation resistance as $R_r = 0.0014\,\Omega$. This gives

$$R_{in} = R_r + R_L = 0.52\,\Omega \,. \tag{4.22}$$

The loop's external inductance can be found by [81]

$$L_A = \frac{K\mu_0 N^2 S}{l} \,, \tag{4.23}$$

where the factor K is known as Nagaoka's constant and is found graphically from the graph in [81] using the relation between coil length and diameter. Using the previous values and $K = 0.7$, we find $L_A = 6.1 \times 10^{-8}\,\mathrm{H}$.

The internal inductance is found by [78]

$$L_i = \frac{Na}{\omega b}\sqrt{\frac{\omega\mu_0}{2\sigma}} \,, \tag{4.24}$$

for which the values give $L_i = 7.20 \times 10^{-11}\,\mathrm{H}$. Thus X_{in} becomes

Table 4.1: Dimensions of the 915 MHz and 2.4 GHz half-wave dipole antennas.

Type	Calculated			Optimized			
	Z_{in} [Ω]	L [mm]	g [mm]	R [mm]	L [mm]	g [mm]	R [mm]
915 MHz OB	73	156.28	0.78	0.33	152.04	0.77	0.34
2.4 GHz OB	73	59.58	0.29	0.12	59.05	0.28	0.12
2.4 GHz IB	61.3				29.01	0.18	0.23

$$X_{in} = 2\pi \left(L_A + L_i\right) = 350.73\,\Omega\,. \tag{4.25}$$

Inserting Eq. (4.22) and Eq. (4.25) into Eq. (4.17), we obtain the value of the parallel capacitor to be $C_r = 4.95 \times 10^{-13}\,\mathrm{H}$. Similarly, the input impedance is found from Eq. (4.18) as $R_{in} = 2.36 \times 10^5\,\Omega$. This impedance may sound high, but will only serve as a starting point to determine the appropriate impedance.

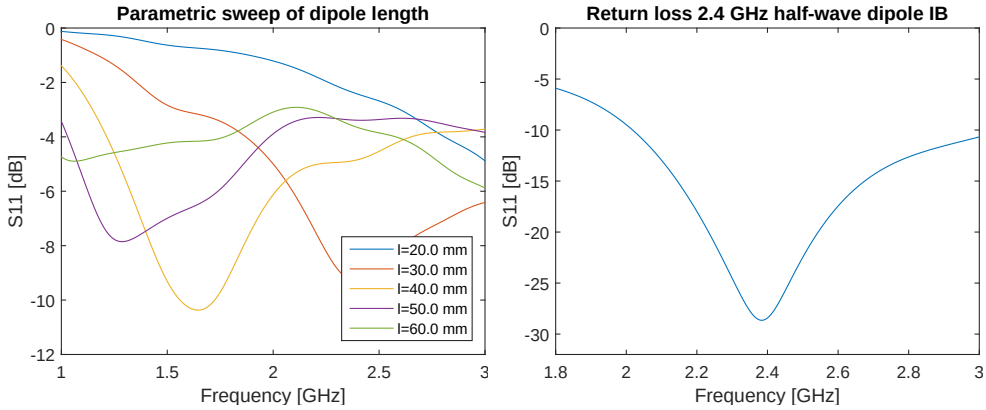
With the high Q-factor of the circuit, minor changes in the resistance and capacitance will cause large changes in the input impedance. In the EM simulations, this is of no problem as the input port can have any real impedance. However, if this antenna should be used in a practical application, the impedance must be matched to the typical $50\,\Omega$ or $200\,\Omega$ circuit. This can be done using a matching circuit. In that case, a T- or π-match would be a good choice, as they allow control of the Q-factor [82].

When the appropriate dimensions and values were found, the coil was made of lossy copper using a solenoidal coil-generating macro in CST MWS. To be able to connect the port and the lumped element, two vertical connectors were added to the coil. To avoid direct contact between the metal and the tissue, the coil and port was covered with teflon as an insulating material.

The structure proved to be somewhat difficult to simulate properly when using the transient solver in CST MWS, as the RLC circuit is highly resonant and stores the excitation energy for a long time compared to more standard antennas. As it was noted in the program documentation [59] that the frequency-domain solver is a good choice for resonant structures, this was used in the initial phase of the design. By using the adaptive meshing, the results became reliable, and the simulation time was shorter than for the transient solver. As the antenna would be used with the transient solver when simulating with the HUGO model, this solver was used after the simulation settings had become reliable. CST MWS has the option of using an online auto regreressive (AR) filter for resonant structures. When this option was enabled, the simulation time was reduced.

It was found that the calculated Z'_{in} was a little high, with a value of $8500\,\Omega$ yielding better matching. The calculated capacitance was not far off, but caused the center frequency to be slighty lower than wanted. By sweeping the parameter, it was found that using $C_r = 1.3 \times 10^{-13}\,\Omega$ gave the wanted center frequency of 915 MHz. A plot of the resulting S_{11} can be seen in Fig. 4.4b, showing a return loss of -25 dB for the center frequency. The bandwidth is quite narrow, i.e. approximately 25 MHz. However, the bandwidth is expected to be slightly wider with more surrounding lossy tissue [83].

The directivity of the antenna can be seen in Fig. 4.5a and Fig. 4.5b. As expected for



(a) Parametric sweep of dipole length. **(b)** Return loss for the antenna.

Figure 4.7: Return loss for the in-body 2.4 GHz half-wave dipole antenna.

a loop antenna, the patterns resemble the directivity of a half-wave dipole antenna, with the azimuth and elevation planes interchanged. It can be seen that the nulls in Fig. 4.5b are not as sharp as on a dipole.

Receiver antenna: Half-wave dipole

For the receiver antenna, the well known half-wave dipole was used. This antenna was chosen for its simple design, omnidirectional radiation pattern and good efficiency. Although known as a half-wave antenna, the performance becomes better if the length is increased slightly [78]. In [84], equations are provided for designing an efficient half-wave dipole with impedance of 73 Ω. These equations are as follows.

The length L of the half-wave dipole is found by the formula

$$L = \frac{1.43 \cdot 10^8}{f}\,, \tag{4.26}$$

where f is the wanted center frequency. The feeding gap g is determined by

$$g = \frac{L}{200}\,. \tag{4.27}$$

The thickness of the wire R is found as

$$R = \frac{\lambda}{1000}\,, \tag{4.28}$$

where λ is the wavelength.

These equations were used to find the dimensions of the dipole for construction in CST MWS, which can be found in Tab. 4.1. A plot of S_{11} can be seen in Fig. 4.6b, with the antenna design shown in Fig. 4.6a. Directivity plots are found in Fig. A.1a and

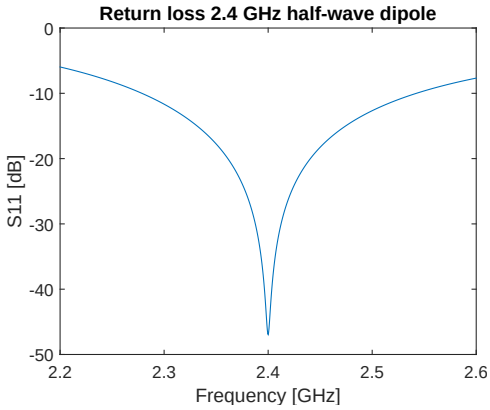


Figure 4.8: Return loss for the 2.4 GHz half-wave dipole antenna.

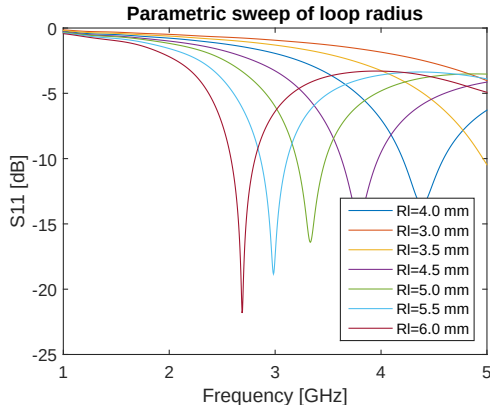


Figure 4.9: Parametric sweep of loop radius for the 1-3 GHz planar loop antenna.

Fig. A.1b in Appx. A. The return loss is approximately -50 dB for the center frequency, with a bandwidth of approximately 100 MHz. The directivity is as expected for a half-wave dipole antenna.

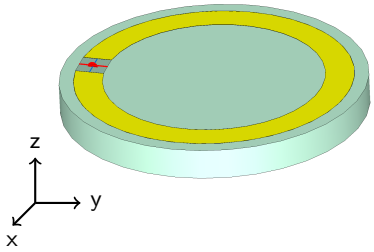
4.2.2 Narrow band 2.4 GHz

For the 2.4 GHz narrow band simulations, dipoles were chosen for both transmitter and receivers. One of the reasons for choosing a half-wave dipole as the transmitter, is the highly omnidirectional radiation pattern. It was thought interesting to compare the performance to the other antenna designs used, that have a less symmetric radiation pattern.

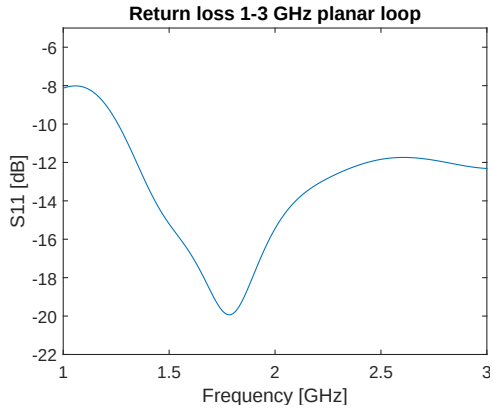
Due to the surrounding tissue, the dimensions of the transmitter antenna could not be found directly from the formulas presented. Using Eq. (4.5) and Gabriel's data, the effective wavelength was found as $\lambda_{\text{eff}} = 0.0166\,\text{m}$ in the intestine. Thus, the length of the dipole had to be reduced. The proper length was found by sweeping the length of the dipole in CST MWS, which is illustrated in Fig. 4.7a. The length 30 mm was found to have $f_c = 2.5\,\text{GHz}$, which was reduced to 2.4 GHz by optimization.

A plot of the return loss can be seen in Fig. 4.7b, with the dimensions used shown in Tab. 4.1. The antenna design is similar to Fig. 4.6a. The directivity plots can be found in Fig. A.3a and Fig. A.3b in Appx. A. Compared to a free space dipole antenna, it can be seen that there is no change in the elevation plane, while the directivity in the azimuth plane is less symmetric.

For the receiver antennas, half-wave dipoles were chosen for the same reasons as mentioned in Sec. 4.2.1. By use of the formulas Eq. (4.26-4.28) appropriate dimensions were obtained. The center frequency was found to be shifted below 2.4 GHz, which was corrected by the optimizer. The return loss of the antenna is shown in Fig. 4.8, with the antenna shape being similar to the design in Fig. 4.6a. In Tab. 4.1, the initial and optimized dimensions of the dipole antenna can be found. The directivity plots are shown in Fig. A.2a and Fig. A.2b in Appx. A.



(a) The antenna design.



(b) The return loss of the antenna.

Figure 4.10: Details of the 1-3 GHz planar loop antenna design.

4.2.3 Ultra wide-band 1-3 GHz

Transmitter: Planar loop

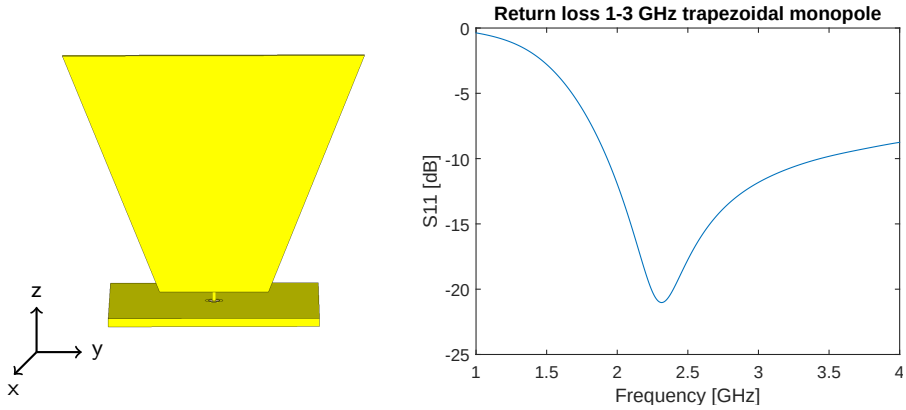
With the 1-3 GHz spectrum being outside the UWB low-band, few antennas exist in the literature that are designed for this frequency range. Thus, it was chosen to modify an existing UWB antenna design. For this purpose, the planar loop WCE antenna in [85] was chosen as a starting point. This antenna is designed for operation between 3.4-4.8 GHz, but has a simple loop design that was thought to be easily modified for another center frequency.

The antenna was modelled following the original specifications in CST MWS, with the design shown in Fig. 4.10a. The simulation results were then compared to the results presented in the article, to check if the modelling had been performed similarly. The results showed similar performance, with a slight shift in center frequency. The difference in the simulation results might originate from different mesh settings or dimensions not specified in the paper.

To adjust the antenna for performance in the lower UWB band, the antenna dimensions were adjusted using the parameter sweep in CST MWS. As the loop antenna is self resonant, the radius of the loop directly determines the resonance frequency. Thus, the first dimension that was swept was the radius. To enable larger loop radius, the diameter of the substrate was increased to 16 mm.

The S_{11} result for different values of R_l can be seen in Fig. 4.9. It was found that the center frequency is lowered when the radius of the loop is increased. As the wanted operation range is between 1-3 GHz, R_l was chosen to be 6.5 mm. The other dimensions (feeding gap, copper thickness) were then swept one at a time. It was found that adjusting these parameters had little impact on the S_{11} performance. However, increasing the width of the loop conductor widened the bandwidth of the loop.

The final dimensions of the planar loop is shown in Tab. 4.3, with the return loss plotted in Fig. 4.10b. The directivity of the antenna can be seen in Fig. 4.14a and Fig. 4.14b. The antenna has good symmetry in both patterns.



(a) The antenna design. **(b)** The return loss.

Figure 4.11: Details of the 1-3 GHz trapezoidal monopole antenna.

Receiver: Trapezoidal monopole

The receiver antenna was chosen to be a trapezoidal monopole design, as this was suggested as a design that had a bandwidth ratio of 11.4:1 and an omnidirectional pattern in [86]. This antenna is also used in the measurement setup in [73] giving good results.

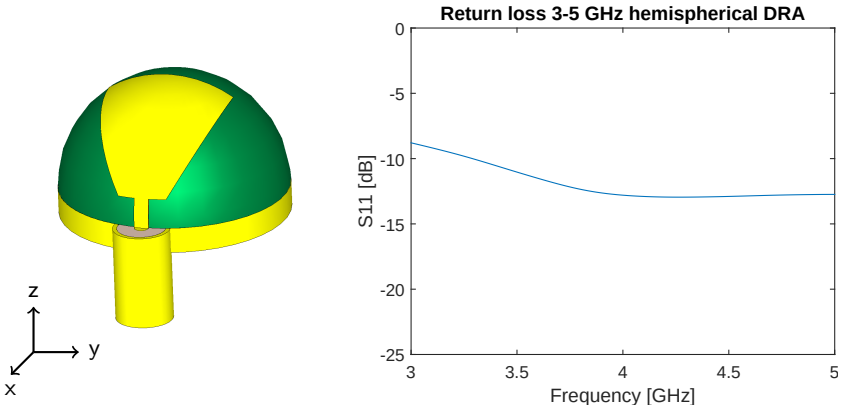
The antenna was designed in CST MWS using the dimensions in [73] as starting point. By sweeping dimensions, it was found that reducing the length and width of the trapezoid, the center frequency was lowered. The dimensions were adjusted until the center frequency was in the range of 2 GHz, and fine tuned using the optimizer. It was found difficult to get a return loss below -10 dB in the spectrum from 1-1.5 GHz. As this is a UWB design, the performance was evaluated as sufficient. The mismatch can be corrected when the S-parameters are known, following Sec. 4.1.8.

The final design is shown in Fig. 4.11a with the return loss plotted in Fig. 4.11b. It was found that the antenna directivity was similar to the pattern of the half-wave dipole, as expected for a monopole design. The directivity plots and the dimensions of the antenna are found in Appx. A as Fig. A.4a, Fig. A.4b and Tab. A.2, respectively.

4.2.4 Ultra wide-band 3-5 GHz

Several antennas have been proposed in the literature in recent years for the UWB low-band from 3.5 to 4.5 GHz for in-body usage. In [87], a printed circuit board patch antenna is presented. The circuit board is dimensioned to fit in the length direction of the WCE capsule, thus differing from the conventional end-design. The manufactured antenna is measured having RL smaller than -10 dB from 3.4-4.6 GHz and gain of 2.23 dB at 4 GHz. However, the antenna is simulated and designed for free space, causing the center frequency to shift from 4 GHz to 1.3 GHz when measured in tissue. No directivity results are shown in the article.

The authors in [85] present three different antenna designs for WCE usage. Two of the designs consist of a dielectric hemisphere, having a conducting layer formed as a loop



(a) The antenna design. **(b)** The return loss.

Figure 4.12: Details of the 3-5 GHz hemispherical DRA antenna design.

and a helix. The performance is analyzed in simulations with 5 cm of surrounding muscle tissue, showing that the loop has best performance with -20 dB RL at the center frequency of 4 GHz. The bandwidth starts at 3 GHz and goes beyond 6 GHz for both the loop and the helix. Considering the directivity, both designs exhibit flat directivity in the xy-plane parallel to the ground plane. In the yz-plane, the helical design has better performance, as the loop has some nulls. Due to manufacturing concerns, a planar version of the loop is produced and measured in a fluid phantom. These results show that the antenna's center frequency has been shifted down to 3.1 GHz, but the bandwidth of 3-5 GHz is still fulfilled.

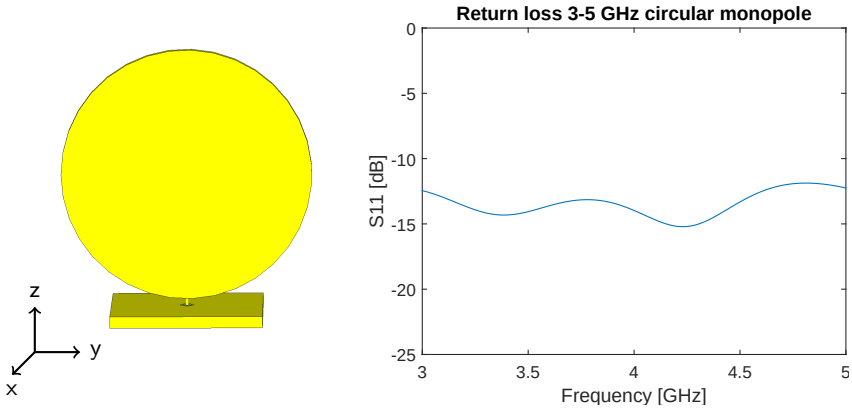
In [73], the authors propose a dielectric resonance antenna. The antenna consists of a dielectric hemisphere excited by a conformal taper strip, and is designed with a surrounding tissue-simulating fluid phantom. The antenna has a free-space center frequency of 25 GHz, which is shifted to 4 GHz with the body tissue affecting the effective wavelength. Simulation results show RL of -25 dB at center frequency, with a bandwidth from 3-5 GHz. The directivity is found to be uniform, and measurements conducted show that the performance is close to the simulations.

Transmitter: Hemispherical DRA

Considering the previous findings, it was chosen to implement the antenna design in [73]. The main motivation for this choice was that it had the best performance, both in simulation as well as measurements. Another point considered, was that it seemed feasible to implement the 3D modelling of the antenna design.

The antenna was modelled according to the description in [73] in CST MWS, with the dimensions presented in Tab. 4.2. The authors did not specify the exact material they used for the dielectric sphere. However, they recommended using a PVC material with relative permittivity around 3. Accordingly, the sphere was modelled with a material having a constant $\epsilon_r = 3$ and $\mu_r = 1$.

The final antenna design can be seen in Fig. 4.12a, with the return loss shown in Fig. 4.12b. It can be seen that the antenna has RL ≤ -9 dB in the 3-5 GHz range. The



(a) The antenna design. **(b)** The return loss.

Figure 4.13: Details of the 3-5 GHz circular monopole antenna design.

directivity of the antenna was found to be frequency dependent, with a less symmetric pattern for higher frequencies, which can be seen in Fig. 4.15a and Fig. 4.15b.

Receiver: Circular monopole

For the receiver antenna, a circular monopole design was employed. At first a trapezoidal design was tested, but this design was found to have a too narrow bandwidth in the 3-5 GHz range. The circular monopole, that has a similar design, gave better performance in this frequency spectrum.

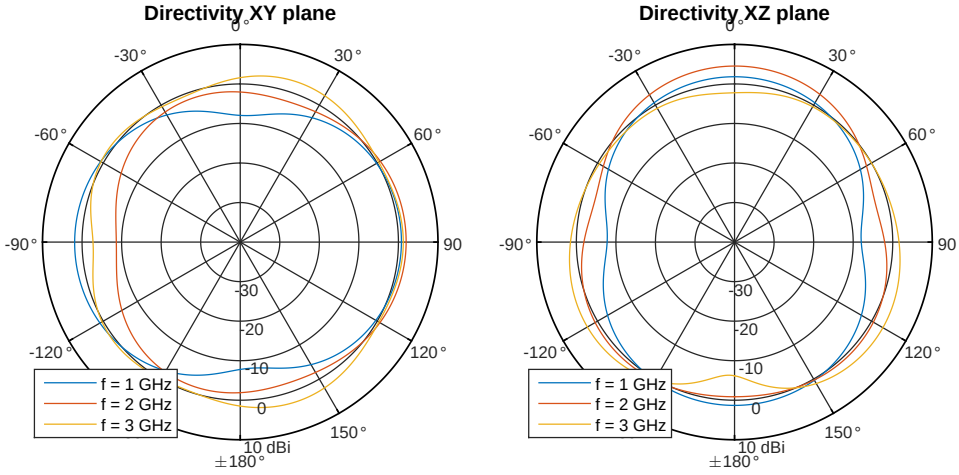
The antenna design is shown in Fig. 4.13a, with the return loss plot found in Fig. 4.13b. The antenna has a flat return loss over the whole frequency band. This, however, comes at the expense of a large size. The antenna directivity can be seen in Fig. A.5a and Fig. A.5b in Appx. A. The directivity is less smooth than some of the other receiver antenna designs presented in this section. The dimensions of the antenna are found in Tab. A.3.

Table 4.2: DRA antenna dimensions.

Description	Symbol	Value
Radius	R	4 mm
Ground plane	t_g	0.9 mm
Strip thickness	t_s	0.009 mm
Coating thickness	t_c	0.18 mm
Strip angle	θ	60 °

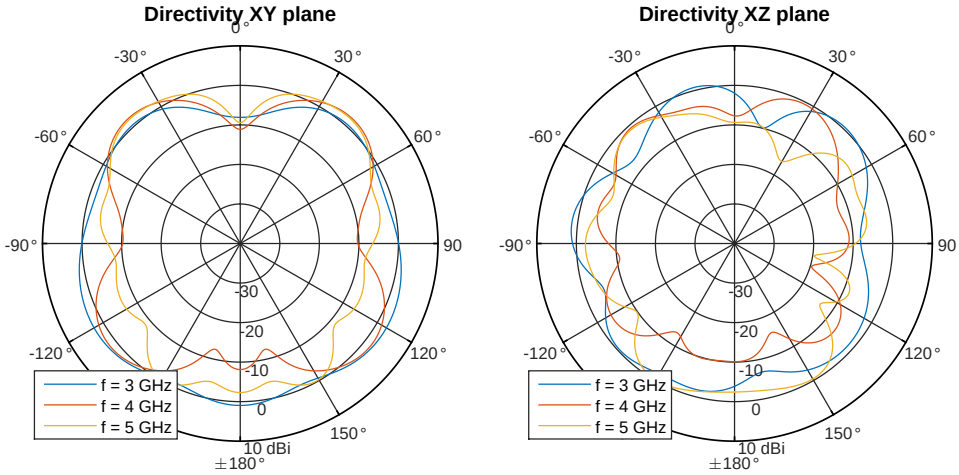
Table 4.3: Planar loop antenna dimensions.

Description		Dimension
Radius substrate	R_s	8 mm
Radius loop	R_l	6.5 mm
Metal height	t_c	0.01 mm
Substrate height	t_s	1.6 mm
Coating	t_i	0.1 mm
Loop gap	g	1.5 mm



(a) Directivity in the azimuth plane. **(b)** Directivity in the elevation plane.

Figure 4.14: Directivity for the 1-3 GHz planar loop antenna, shown for the center frequency and the upper and lower boundary.



(a) Directivity in the azimuth plane. **(b)** Directivity in the elevation plane.

Figure 4.15: Directivity for the 3-5 GHz DRA antenna, shown for the center frequency and the upper and lower boundary.

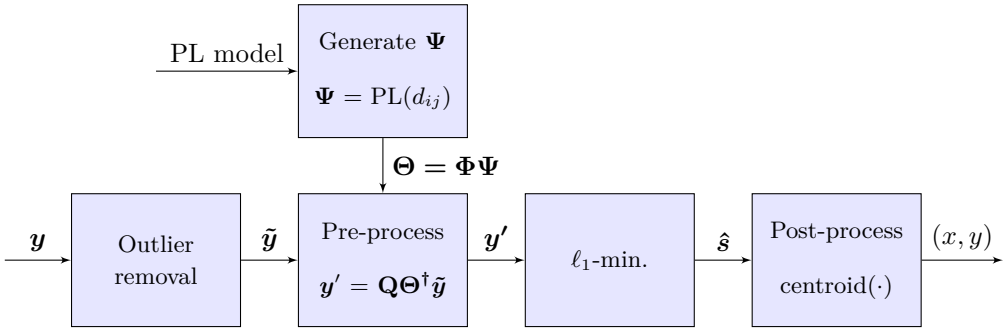


Figure 4.16: Overview of the implementation of the localization system.

4.3 Localization

The implementation of the CS-based localization framework will be presented in this section. A block diagram depicting the implementation is shown in Fig. 4.16, where it can be seen that the main data provided to the system are the path loss measurements $\boldsymbol{y}$ and a path loss model that is used to generate $\boldsymbol{\Psi}$.

The system was implemented by object oriented programming in MATLAB. The main object is the compressive sensing framework called `CompressiveSensing.m`. This object uses the other objects `Grid.m` and `PathLoss.m`, which represents the localization grid and path loss models respectively. Selected parts of the code is found in Appx. B.

4.3.1 ℓ_1-minimization algorithms

From the compressive sensing object, the localization can be performed using the different ℓ_1-minimization algorithms presented in Sec. 3.2.2. The BP and BPDN algorithms are solved using the ℓ_1-MAGIC code package [88] for MATLAB. The Bayesian compressive sensing algorithms ST-BCS, MT-BCS and AD-BCS have been implemented in MATLAB. Implementation details of these algorithms will be presented in the following.

ST-BCS The ST-BCS algorithm is a direct implementation of Eqs. (3.15-3.19) in Sec. 3.2.2.

AD-BCS The AD-BCS algorithm is similar to ST-BCS, except that the required number of measurements are determined dynamically by the covariance estimates in $\boldsymbol{\Sigma}$. The algorithm starts by using four measurements. If the algorithm doesn't converge in the specified number of iterations, one more receiver is added. This procedure is repeated until convergence or all receivers are used, in which the algorithm is equal to the ST-BCS algorithm.

MT-BCS The MT-BCS algorithm was implemented from Eqs.(3.21-3.25) in Sec. 3.2.2.

For all algorithms, the convergence criterion used is to stop iterating if the mean variance in the position estimates $\hat{\boldsymbol{s}}$ are below a certain threshold E_b. This is the same criterion used in [32]. If no convergence is reached for $k_{\max}$ iterations, the algorithms stop and return their current estimate for $\hat{\boldsymbol{s}}$.

Table 4.4: In-body to on-body path loss model parameters [74].

Description	a	b	c
No compensation	-0.813	7.817	-3.235
Compensation	-0.800	7.839	-3.647

4.3.2 In-body path loss models

In order to generate the sparsity basis $\mathbf{\Psi}$, an appropriate in-body path loss model is required. The issue of establishing a body channel model has been addressed in the upcoming IEEE P802.15 standard [89]. For the case of in-body to on-body path loss, however, the only frequency range investigated is 402-405 MHz.

In [90], a path loss model was found from simulations and experiments for the 402, 868 and 2.4 GHz bands. The simulations were done using the HUGO voxel model in CST MWS, with the measurements done using a human phantom. The path loss exponents found can be seen in Tab. A.4 in Appx. A.

For the UWB 1-6 GHz frequency band, the articles [91, 92, 74] all investigate the in-body to on-body path loss. In [91], a model for the in-body channel in the human chest is provided though numerical simulations on a voxel model. Similarly, [92] provides a path loss model for the abdominal region. The article [74] extends the insight obtained through [91, 92], by performing measurements on a living animal. The measurements are performed on a porcine, at depths from 50 to 160 mm in the frequency range 1-6 GHz. From these results, a frequency dependent log-distance path loss model is constructed as

$$\mathrm{PL}(d, f) = \mathrm{PL}_0 + 10N(f)\log_{10}\left(d/d_0\right), \tag{4.29}$$

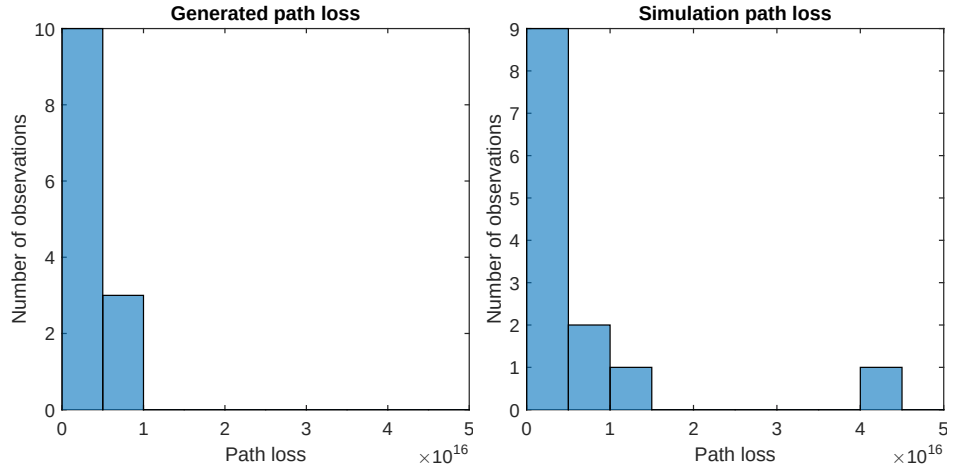
where the path loss exponent $N(f)$ is represented as $N(f) = af^2 + bf + c$. The fitting parameters for $N(f)$ are found in Tab. 4.4.

Models used for localization

From the path loss models presented, the following were chosen to use for localization with the four EM simulation setups:

- **NB 915 MHz**: 868 MHz model from [90]
- **NB 2.4 GHz**: 1-6 GHz model from [74]
- **UWB 1-3 GHz**: 1-6 GHz model from [74]
- **UWB 3-5 GHz**: 1-6 GHz model from [74]

The model [74] was chosen for its frequency range and that it is based on measurements on a living animal. [90] was chosen for being closest to the 915 MHz band. For the UWB setups, an optional shadowing parameter was implemented by using the shadowing distribution from [92].



(a) Path loss data generated from the model in [74].

(b) Path loss data obtained from one of the EM simulations.

Figure 4.17: Distribution of path loss from model compared to path loss from one EM simulation. The same configuration has been used for both plots.

4.3.3 Outlier detection

During testing it was found that the ℓ_1-minimization algorithms were susceptible to outliers. To try to remedy possible outliers, an outlier detection system was developed to find and remove outliers. Some of the outlier removal algorithms from the MATLAB code package [93] were tested, e.g. the generalized ESD procedure. By using datasets with known outliers, the detection was found to be unreliable. This was probably due to the measurements not resembling the Gaussian distribution, which many of the known outlier detection techniques rely on.

As none of the methods in [93] provided usable results, a simple technique that partitioned the path loss measurements into bins were used. With more than a few receivers, it is expected that the path loss varies gradually between the different antennas. Thus, when one of the bins have a deviation from the others, it is likely that the observation is an outlier. It was chosen to detect the observation as an outlier if one or more bins separated the measurement from the other observations, with a limit of maximum three observations in the bin. This procedure is illustrated in Fig. 4.17b, where it can be observed that we have an outlier that will be removed in the measurements. In Fig. 4.17a path loss data generated from the model in Eq. (4.29) for the same transmitter/receiver distances are shown. Comparing the two plots, it seems reasonable to remove this outlier.

4.4 Tracking

Implementation details of the Kalman- and particle filter will be shown in this section. Due to the large similarity between the Kalman filter and the VNL Kalman filter, they will be treated together in the following section.

4.4.1 Kalman filter

The Kalman filter and the VNL Kalman filter was implemented as objects in MATLAB from the recursive estimation equations shown in Sec. 2.3.1 and Sec.2.3.2. Both filters were designed to allow usage of different dynamic models. Code for the VNL Kalman filter is found in Appx. B. Implementation details of the filters will be shown in the following.

System description

As the velocity and positions are tracked using the filter, the state vector is of the form

$$\boldsymbol{x} = \begin{bmatrix} x & \dot{x} & y & \dot{y} \end{bmatrix}^{\mathrm{T}}. \tag{4.30}$$

For a constant velocity model with measurements of positions x and y and timestep T, the state transition matrix $\mathbf{A}$ is

$$\mathbf{A} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.31}$$

and the observation matrix $\mathbf{C}$ is

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{4.32}$$

The control input matrix was $\mathbf{B}$ chosen as zero, with mismatch in the model compensated by increased noise in the process noise covariance matrix $\mathbf{Q}$.

The covariance matrix for observation noise $\mathbf{R}$ was chosen as follows

$$\mathbf{R} = \begin{bmatrix} \sigma_{r,x}^2 & 0 \\ 0 & \sigma_{r,y}^2 \end{bmatrix}, \tag{4.33}$$

where $\sigma_{r,x}^2$ and $\sigma_{r,y}^2$ are the variances of the measured noisy observations. The Kalman filter has a process noise covariance matrix $\mathbf{Q}$

$$\mathbf{Q} = \operatorname{diag}\left(0,\ \sigma_{q,x}^2,\ 0,\ \sigma_{q,y}^2\right), \tag{4.34}$$

while the VNL Kalman filter has two different process noise covariance matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$, corresponding to the two modes of the filter as

$$\mathbf{Q}_1 = \text{diag}\left(0,\ \sigma^2_{q_1,x},\ 0,\ \sigma^2_{q_1,y}\right) \tag{4.35}$$
$$\mathbf{Q}_2 = \text{diag}\left(0,\ \sigma^2_{q_2,x},\ 0,\ \sigma^2_{q_2,y}\right), \tag{4.36}$$

where the variances $\sigma^2_{q_1,x}$ and $\sigma^2_{q_1,y}$ represent unmodelled accelerations when the capsule is moving. $\sigma^2_{q_2,x}$ and $\sigma^2_{q_2,y}$ correspond to process noise that compensates for the transition between movement and the capsule being stopped.

The initial error estimate covariance matrix $\mathbf{P}$ was initialized as

$$\mathbf{P} = \text{diag}\left(\sigma^2_{r,x},\ \sigma^2_{v,x},\ \sigma^2_{r,y},\ \sigma^2_{v,y}\right). \tag{4.37}$$

The variances $\sigma^2_{v,x}$ and $\sigma^2_{v,y}$ regulate the credibility of the initial velocity estimate. They were both chosen as unity when performing the tracking, as the initial velocity is not known. $\sigma^2_{r,x}$ and $\sigma^2_{r,y}$ is found from the observation noise.

4.4.2 Particle filter

The particle filter was implemented as a class in MATLAB from the presentation in Sec. 2.3.3 and information from [55, 56, 94]. The code can be found in Appx. B. As Gaussian noise was added to the positions, the Gaussian distribution was chosen as the particle filter's proposal distribution.

The filter was implemented with the constant velocity and RWM movement models, following the reasoning in Sec. 3.3.2. Both these models will be treated in the following.

Initialization and prediction

The particle filter starts by initializing N particles $\boldsymbol{x}_0^1, ..., \boldsymbol{x}_0^N$ according to Gaussian distribution $\mathcal{N}(\boldsymbol{x_0}, \mathbf{N}_0)$, where $\boldsymbol{x_0}$ is the initial state provided to the filter and $\mathbf{N}_0$ determines the variance of the initial particles. Using the constant velocity dynamic model with the state vector in Eq. (4.30), the state evolution becomes

$$\boldsymbol{x}_k^i = \mathbf{A}\boldsymbol{x}_{k-1}^i + \boldsymbol{v}\,, \tag{4.38}$$

where $\boldsymbol{x}_k^i$ is particle number i representing one proposed state vector at time k. Matrix $\mathbf{A}$ is equal to the matrix in Eq. (4.31) used for the Kalman filter. $\boldsymbol{v}$ is Gaussian noise distributed according to $\mathcal{N}(0, \mathbf{N})$.

For the RWM movement model the state evolution is

$$\boldsymbol{x_k^i} = \boldsymbol{x_{k-1}^i} + v_k \boldsymbol{e_k} T\,, \tag{4.39}$$

where

$$v_k \sim \mathcal{N}(\bar{v}, \sigma_v) \tag{4.40}$$

$$\boldsymbol{e_k} = \frac{\boldsymbol{g_k} - \boldsymbol{g_{k-1}}}{|\boldsymbol{g_k} - \boldsymbol{g_{k-1}}|} \tag{4.41}$$

$$\boldsymbol{g_k} \sim \left[\, \mathcal{U}(0, L_x) \;\; \mathcal{U}(0, L_y) \,\right]^{\mathrm{T}}. \tag{4.42}$$

In Eqs. (4.39-4.42) v denotes the velocity, $\bar{v}$ is the average velocity, σ_v is the standard deviation of the velocity, $\mathcal{U}(a, b)$ is the uniform distribution between a and b and L_x/L_y defines the maximum movement length in x and y direction.

Particle weights

The particles $\boldsymbol{x}_k^i$ are assigned weights w_k^i by the multivariate Gaussian

$$w_k^i = p(\boldsymbol{z}_k \mid \boldsymbol{x}_k^i) = \frac{1}{\sqrt{(2\pi)^M \left|\det(\mathbf{R})\right|}} \exp\left(-\frac{1}{2}(\boldsymbol{\pi_k^i})^T \mathbf{R}^{-1} \boldsymbol{\pi_k^i}\right), \tag{4.43}$$

where

$$\boldsymbol{\pi_k^i} = \boldsymbol{z}_k - \boldsymbol{x}_k^i\,, \tag{4.44}$$

and $\boldsymbol{z}_k$ is the current observations having covariance $\mathbf{R}$ and dimension M.

To give the particle filter that uses the constant velocity model a sense of how well the velocity estimates of each particle fits in the reweighing, the observation vector $\boldsymbol{z_k}$ was chosen as

$$\boldsymbol{z}_k = \left[x_k \quad y_k \quad \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2}/T\right]^{\mathrm{T}}. \tag{4.45}$$

For the RWM movement model, the observation vector is simply $\boldsymbol{z}_k = [x_k \quad y_k]^{\mathrm{T}}$. After all particles have been assigned a weight, the weight distribution is normalized.

Resampling

The resampling strategy chosen was the cumulative distribution sampling, described in Sec. 2.3.3. The resampling process is computationally expensive. In order to get good performance, the resampling code used in the implementation was found in [55] as the *multinomial sampling with sort*.

Estimate

The estimate for timestep k is found by

$$\hat{\boldsymbol{x}}_k = \sum_{i=1}^{N} \boldsymbol{x}_k^i w_k^i \,, \tag{4.46}$$

where $\boldsymbol{x}_k^i$ are the particles after resampling.

Chapter 5

Results

The results from this thesis will be presented in this chapter. The results can be divided into four parts: Sec. 5.2 covers details and the validity of the in-body EM simulations performed, Sec. 5.3 and Sec. 5.4 presents the localization results using generated- and EM simulated path loss data, while Sec. 5.5 contains results from performing tracking and finding distance estimates from intestinal positions.

The following section presents the hardware and software used for the simulations, in addition to performance evaluation metrics that will be employed in later sections.

5.1 Evaluation details

5.1.1 Hardware and software

All results are produced using a PC with a 3.4 GHz 6 core CPU and 32 GB of RAM. The simulations were performed with CST MWS version 2014.00.

The localization, tracking and plotting was performed with MATLAB R2014b. The only external library used is ℓ_1-MAGIC version 1.11.

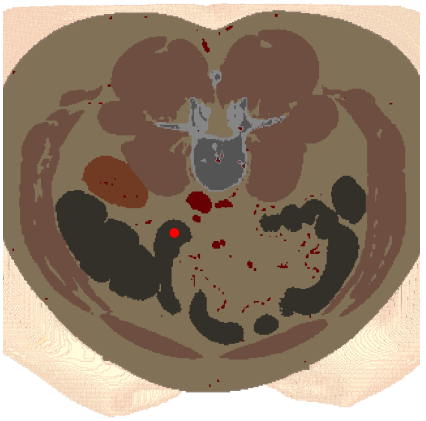
5.1.2 Performance evaluation

The performance of the localization is evaluated using the localization error (LE), defined as Euclidean distance from the true position (x, y) to the estimated position $(\hat{x}, \hat{y})$.

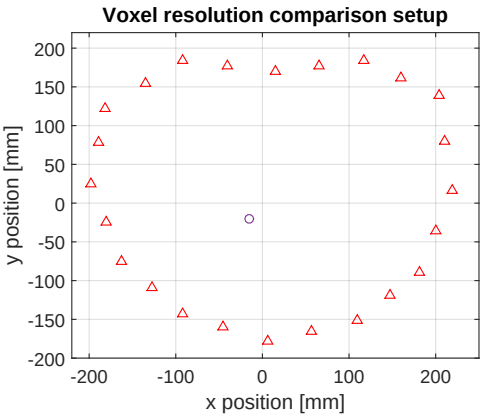
$$\mathrm{LE} = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2}. \tag{5.1}$$

The true position of the transmitter antenna is defined as the center of the antenna structure.

For some of the measurements, the performance under different levels of signal-to-noise ratio (SNR) are considered. The definition of SNR used is [95]



(a) The position of the transmitter in the intestine.



(b) The antenna placement.

Figure 5.1: Simulation setup used for comparison of $1\,\mathrm{mm}^3$ and $2\,\mathrm{mm}^3$ voxel resolution.

$$\mathrm{SNR} = 10\log_{10}\left(\frac{\sum_{n=1}^{N} y_n}{N\sigma_\epsilon^2}\right), \tag{5.2}$$

where y is the noise-free signal of interest and σ_ϵ^2 is the noise variance.

For estimates $\hat{\boldsymbol{x}}$ of a signal $\boldsymbol{x}$, the definition of RMSE used is

$$\mathrm{RMSE} = \sqrt{\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\left(\hat{x}_n^m - x_n\right)^2}, \tag{5.3}$$

where M is the number of simulations and N is the length of the dataset.

The normalized position error (NPE) [94], is defined as the ratio of the mean-square estimation error to the mean-square measurement error over M different simulations

$$\mathrm{NPE}(k) = \sqrt{\frac{\sum_{n=1}^{M}\left(x_n(k) - \hat{x}_n(k)\right)^2 + \left(y_n(k) - \hat{y}_n(k)\right)^2}{\sum_{n=1}^{M}\left(x_n(k) - z_n^x(k)\right)^2 + \left(y_n(k) - z_n^y(k)\right)^2}}, \tag{5.4}$$

where $\boldsymbol{x}, \boldsymbol{y}$ are the true positions, $\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}$ are the position estimates and $\boldsymbol{z}^x, \boldsymbol{z}^y$ are the observations. Lower NPE value implies better estimation, with a value lower than one indicating good estimation. By plotting NPE(k), the estimation performance can be compared for different estimation methods over time.

For the distance estimation problem, the mean difference in length over M simulations is found as

$$\bar{\Delta} = \frac{1}{M}\sum_{n=1}^{M}\left(\hat{d}_n - d\right), \tag{5.5}$$

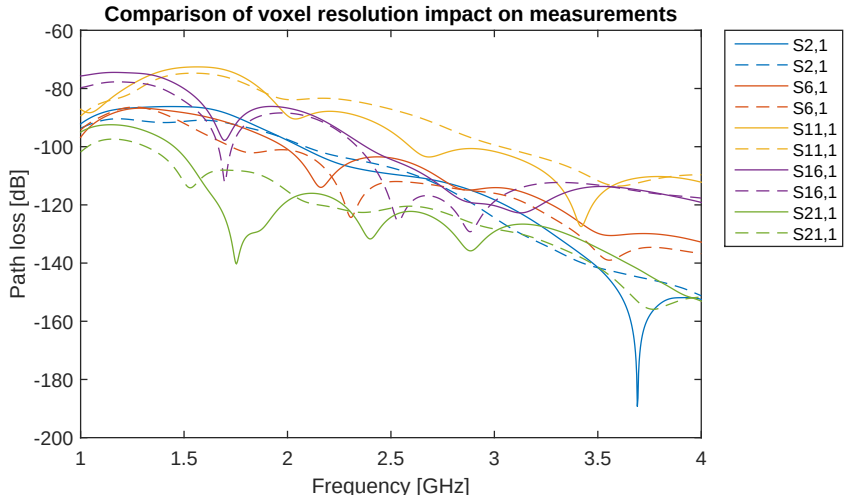


Figure 5.2: Comparison of path loss measurements using different voxel resolutions. The solid lines show results from 1 mm^3 resolution, while the dashed lines are the 2 mm^3 resolution.

where d is the true distance and $\hat{d}$ is the length estimate.

5.2 In-body EM simulation results

5.2.1 Voxel resolution

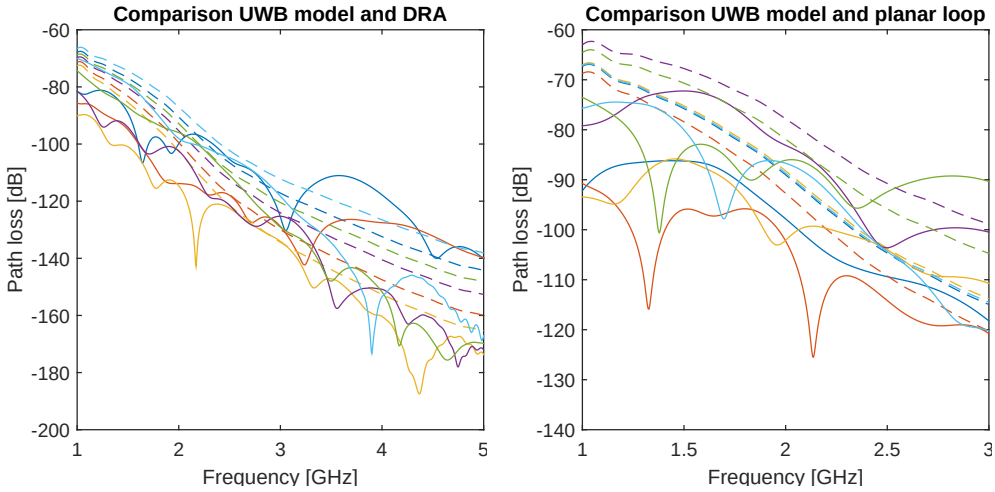
As mentioned in Sec. 2.4.3, the HUGO voxel model is available in different voxel resolutions, which determines the amount of detail in the 3D model. The coarsest resolutions was found to give a resulting model with far too few details to be usable, as it was not possible to distinguish the shape of the intestine. The resolutions that was found to provide enough details were 1 mm^3 and 2 mm^3.

From the point of accuracy, the densest resolution would be the best one to use, as the shape of the organs would be closer to reality. As the material properties is determined in each voxel, the higher resolution could give more realistic results. However, increasing the resolution would warrant a finer simulation mesh and thus cause an increased simulation time.

Before choosing the resolution to be used for the simulations, an experiment was conducted with the HUGO model in 1 mm^3 and 2 mm^3 resolution. This was done to check if the increased voxel density was worth the increased simulation time. The simulation setup used was as follows:

- Transmitter: Planar loop
- Receivers: 24 trapezoidal monopoles
- Frequency range: 1-4 GHz

Two simulations were performed with the transmitter positioned at the same location, and the only difference being different voxel resolution. The simulation setup can be



(a) 3-5 GHz hemispherical DRA antenna. **(b)** 1-3 GHz planar loop antenna.

Figure 5.3: Comparison of the 1-6 GHz UWB path loss model with EM simulations. The solid lines are the simulation results, with the dashed lines representing the model. Note that the y-axis has a different scale on the two plots.

seen in Fig. 5.1a and Fig. 5.1b.

In Fig. 5.2, 5 of the 24 $S_{n,1}$-parameters for receivers evenly spread around the torso are shown. As can be seen, there is clearly a difference for all parameters. For most of the S-parameters, this difference is in the form of a less smooth curve, with small dips for the case of of $1\,\text{mm}^3$. This can easily be seen for the $S_{21,1}$ curves. However, there are also some notable exceptions as e.g. $S_{16,1}$, where fewer dips occurs.

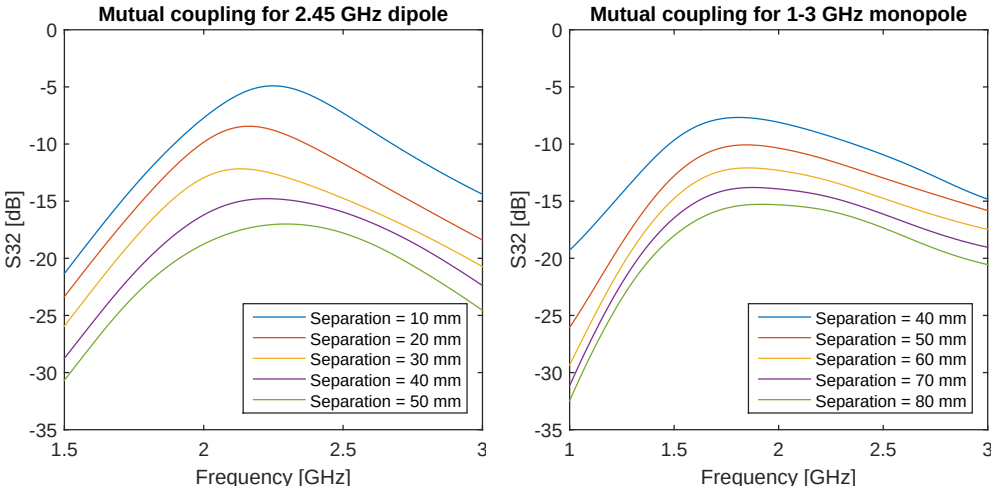
From these results, it was decided to generally use the $1\,\text{mm}^3$ resolution for the simulations. The few exceptions to this decision will be explicitly stated.

5.2.2 Path loss

To validate the simulation results, an experiment was performed comparing the path loss measured as S_{21} to the path loss calculated from the in-body to on-body UWB model in Eq. (4.29). The path loss data were generated by use of the Euclidean distances separating the transmitter and receiver antennas. The data shown in the following plots are an evenly spaced subset of all the measurements, to ease the readability of the plots. For the planar loop, results for all antennas are found in Fig. A.6 in Appx. A.

In Fig. 5.3a, results from one of the simulation runs using the 3-5 GHz DRA antenna are shown, compared to the path loss model. As the DRA antenna is the same one used to obtain measurements for the UWB model, the frequency range of this plot is extended to 1-5 GHz. Note that the path loss measurements from the DRA simulation fits the model quite well. The frequency dependence is also clearly visible, varying between -70 dB at 1 GHz to approximately -150 dB at 5 GHz.

The comparison in Fig. 5.3b is of the 1-3 GHz planar loop and the UWB model. The



(a) 2.4 GHz dipole. **(b)** 1-3 GHz trapezoidal monopole.

Figure 5.4: Mutual coupling comparison for receiver antenna arrays. The coupling is measured as S_{32} between receiver antenna 2 and 3 at different distances. All results were measured at the front of the torso of HUGO, at around navel height.

simulation data from the planar loop has similarity in the range from 2-3 GHz. It is noticed that there are some spreading in the path loss results from the simulation. Between 1-1.5 GHz, the mismatch is quite large for some of the curves, with the model having less loss than the simulations. Some possible explanations for this behaviour will be given in Ch. 6.

From these results it was decided that the 1-6 GHz UWB path loss model from [74] was adequate for localization in the UWB experiments.

5.2.3 Receiver separation

When antennas are placed near each other, the antenna pattern and input impedance will change due to mutual impedance between the antenna elements [78], also known as mutual coupling. To find the minimum antenna distance between the receiver antennas, the antenna isolation was measured for different receiver distance separation. This was done for the half-wave dipole antenna and the trapezoidal monopole.

For the dipole experiment, two 2.45 GHz dipoles were used with separation 10-50 mm. To save some simulation time, only the torso part of HUGO was used with resolution $2\,\text{mm}^3$, as the anatomical details was thought to be of small importance to the mutual coupling. The antenna isolation was measured as S_{32} between port 2 and 3 of the two dipoles. The result can be seen in Fig. 5.4a. The maximum mutual coupling is -5 dB at the center frequency for the 1 cm separation.

In the other experiment, two 1-3 GHz trapezoidal monopoles were used with separation 40-70 mm from center to center. With this antenna, 40 mm is the minimum distance possible without intersection. The mutual coupling can be seen in Fig. 5.4b. The maximal mutual coupling is -7.7 dB close to the center frequency for 40 mm separation,

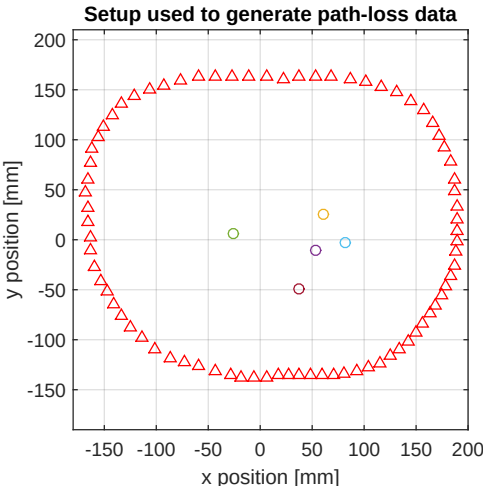


Figure 5.5: The setup used to generate path-loss data to evaluate the performance of the localization algorithm. The circles represent the five transmitter positions used, while the triangles are the receivers.

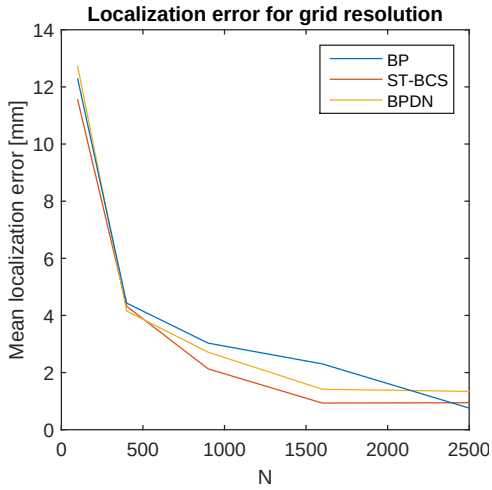


Figure 5.6: Localization error for different grid resolutions, calculated as the mean of the five position estimates shown in Fig. 5.5.

with the value decreasing for higher and lower frequencies.

In [96], an acceptable antenna isolation between two parallel dipoles is suggested as -15 dB at resonance. Following this advice, the minimal distance used for the dipole receiver arrays were chosen as 30-40 mm.[1] Similarly, for the trapezoidal monopoles, a separation of minimum 60 mm was used as the wider bandwidth allows a larger peak isolation.

5.3 Localization using generated data

The performance of the different localization algorithms will first be evaluated using data generated from a path loss model. This was done to measure the performance under ideal conditions, and to check how the accuracy was affected by noise.

The transmitter and receiver setup used for this purpose can be seen in Fig. 5.5. Note that in this figure, the maximum number of receivers are shown; for most of the tests only subset of these receivers will be used. The five transmitter positions were found from the intestine in the HUGO model. The mean localization error is calculated as the mean LE of the five positions. Similarly to Sec. 5.2.2, the path loss was calculated from the 1-6 GHz UWB path-loss model using Eq. (4.29), with an intermediate fixed frequency of 2.4 GHz. Note that for these experiments, the outlier removal procedure from Sec. 4.3.3 was not employed.

In Fig 5.6, the results from changing the resolution of the grid are shown. The accuracy is decent already at the coarsest resolutions, with small improvements up to a resolution of $N = 2000$, where the error stabilizes around 2 mm. As the computational

[1] As the antenna points were picked manually, some deviation in separation was observed.

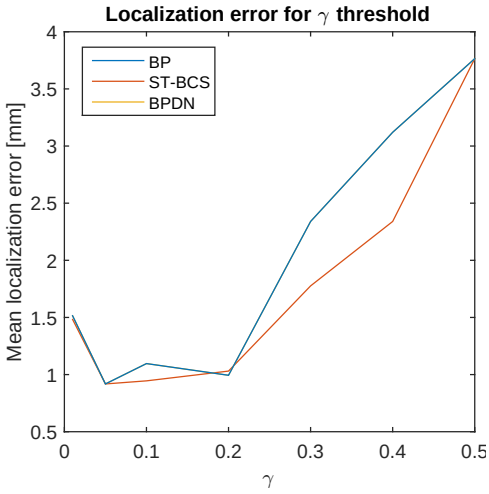


Figure 5.7: Localization error for different choices of the γ threshold. Note that the performance is equal for BP/BPDN.

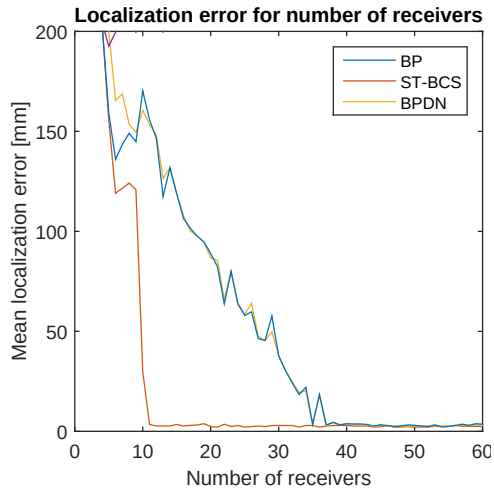


Figure 5.8: Localization error for varying numbers of receivers.

complexity of the Bayesian algorithms follows $\mathcal{O}(N^3)$ with increase in grid resolution, it was chosen to use $N = 900$ as a compromise between speed and accuracy for the rest of the results.

Fig 5.7 show the results of changing the threshold γ that is used when calculating the centroid of the sparse estimated location vector, as explained in Sec. 3.2.3. The plot show that a minimum error is found when having γ between 0.05 and 0.2. From this result, γ was chosen as 0.05.

Fig 5.8 show the location error with increasing number of receivers. It can be observed that increasing the number of receivers increases the chance of successful localization. For the BP and BPDN algorithms, a minimum of 35 receivers are needed to reach a mean error on the order of millimeters, while the ST-BCS algorithm only requires 10.

The results presented up to this point show that an excellent accuracy is possible under perfect theoretical conditions. However, it was found that these settings provided poor results when used with real path loss data. By using the path loss is decibels, the performance increased substantially for real data. Thus, in order to get results representative for the real data, the results in the rest of this section will be from using PL_{dB}. Reasons for this behaviour will be given in Sec. 6.2.

Having established the performance under ideal conditions, Gaussian noise was added to the measurements having different levels of SNR ranging from 0 to 50 dB. The results can be seen in Fig. 5.9, and are found from 50 Monte-Carlo runs. Changing from PL to PL_{dB} decrease the accuracy of the BP and BPDN algorithms to approximately 30 mm under low noise, with large errors occurring for higher noise levels. The performance of the Bayesian CS algorithms are better for all noise levels, with the ST-BCS providing more accurate results under heavy noise corruption. It can be seen that when we have SNR below 30 dB, the position estimates are not really usable due to the low accuracy.

When using real in-body path loss data, the resulting path loss can be larger or smaller

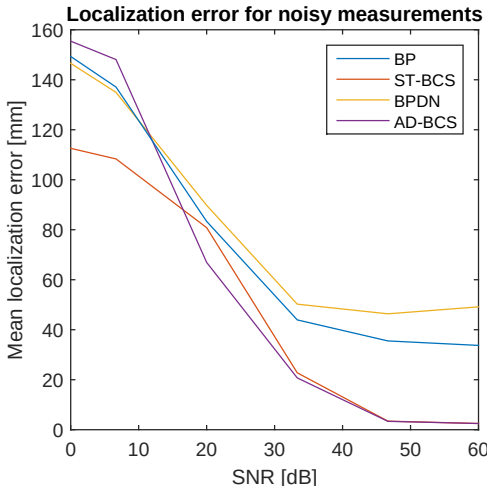


Figure 5.9: Localization error for varying amounts of noise.

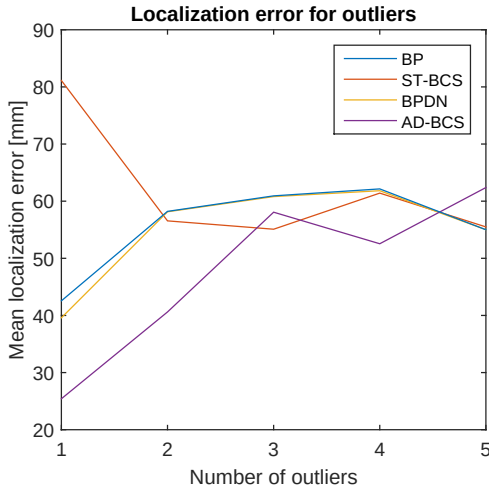


Figure 5.10: Localization error with outliers added to the measurements.

than expected for the distance between transmitter and receiver. Such outliers are typically caused by obstacles and local tissue properties, or non-uniform directivity in the antennas. For this reason, it was decided to test the localization algorithms' resilience to outliers. For this test, 1-5 measurements were randomly chosen from the path loss data. The path loss of these measurements were increased by 50 %, to simulate an outlier. The results are displayed in Fig. 5.10. The performance of the algorithms decrease when facing outliers, with the error stabilizing around 50-60 mm for more than three outliers. It can be seen that AD-BCS performs well when the number of outliers are low, as the algorithm converge without using the affected outlier measurements.

5.3.1 Multi-point localization

In order to evaluate the performance of the MT-BCS algorithm, that utilizes correlation in the path loss measurements, an experiment with ten positions was conducted. The spacing between one point to the next was chosen to be only a few millimeters, to resemble actual WCE measurements from the intestine, and also to try to obtain some correlation in the data. The lower noise boundary was chosen as -20 dB, to determine how the algorithms performed under very large amounts of noise.

The plot in Fig. 5.11 show the results with varying levels of Gaussian noise, compared to the other algorithms that estimate each position separately. The MT-BCS algorithm is seen to outperform the other algorithms below 10 dB SNR, with an error of approximately 50 mm down to -20 dB SNR. The other algorithms behave similarly to the results that was observed in Fig. 5.9, with the exception of the BP and BPDN having a better performance under low amounts of noise. The reason for this behaviour is that the positions used in this experiment have less spatial spreading, and is positioned more ideally, close to the center of the grid.

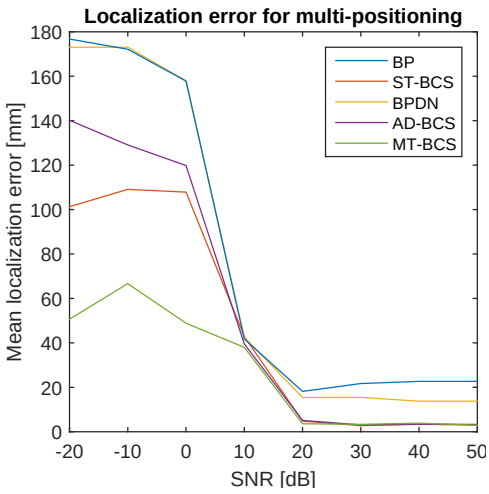


Figure 5.11: Localization error for multi-point localization.

Table 5.1: Settings used for localization using EM simulation data.

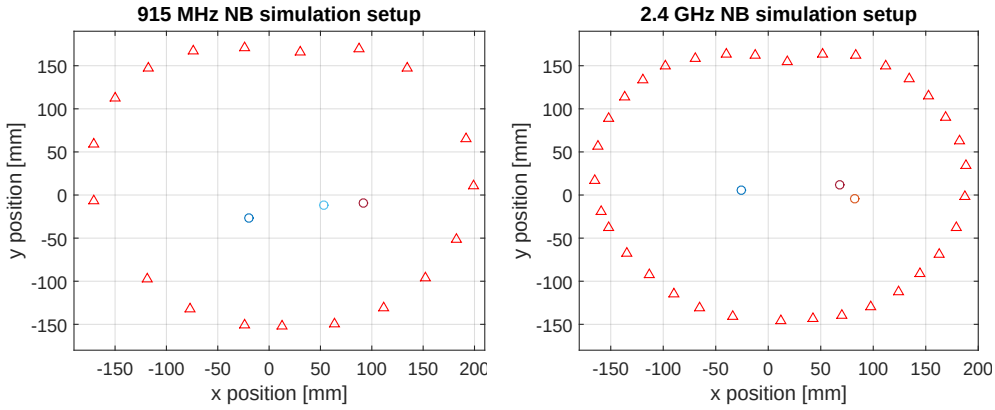
Description	Symbol	Value
Grid resolution	N	900
Gaussian noise	SNR	25 dB
Centroid threshold	γ	0.05
BPDN regularization	ϵ	10^{-3}
Iterations ST-BCS	$k_{\max,\mathrm{ST}}$	30
Iterations MT-BCS	$k_{\max,\mathrm{MT}}$	50
Convergence crit.	E_b	10^{-7}

Table 5.2: Summary of the in-body EM simulation experiments.

Name	Frequency	Transmitter	Receivers
NB0.9	902-928 MHz	Coil	Half-wave dipole
NB2.4	2.35-2.45 GHz	Half-wave dipole	Half-wave dipole
UWB1-3	1-3 GHz	Planar dipole	Trapezoidal monopole
UWB3-5	3-5 GHz	Hemispherical DRA	Circular monopole

Table 5.3: Details of the simulation datasets. Number of meshcells and the simulation time is averaged over all sets.

Name	Simulations	Receivers	Meshcells	Simulation time
NB0.9	3	20	41 mill	29 h
NB2.4	3	34	75 mill	9 h
UWB3-5	3	13	64 mill	7 h
UWB1-3	30	24	48 mill	4 h



(a) The 915 MHz simulation setup. **(b)** The 2.4 GHz simulation setup.

Figure 5.12: The simulation setup used for the narrow-band simulations. The red triangles indicate the receiver antenna positions, while the circles show the transmitter antenna position. Note that the grid shown here has different dimensions than the grid used for localization.

5.4 Localization using EM simulation data

In this section, the performance of the localization algorithms will be evaluated using the narrow-band and wide-band antenna configurations presented in Sec. 4.2. For convenience, these configurations are summarized in Tab. 5.2. The simulations performed are shown in Tab. 5.3, where number of antennas, meshcells and simulation times are listed.

Ideally, all simulations used to compare the different transmitter/receiver combinations would be performed with the exact same number of antenna receivers positioned at the same locations. However, due to the fact that the transmitter and receiver antennas all have a different physical size and shape, this proved to be difficult. Following the results presented in Sec. 5.2.3, the number of receivers were limited by the space needed between them to avoid excessive coupling. Unless explicitly mentioned otherwise, all antennas are oriented as depicted in the antenna design figures presented in Sec. 4.2, with the z-axis aligned with the vertical direction of the HUGO model (the z-axis in Fig. 2.2).

The number of simulations performed with different antenna configurations should ideally be large. Unfortunately, due to the computation time involved with each simulation, this was not possible to accomplish within the given time frame. For the NB0.9, NB2.4 and UWB3-5 three simulations have been performed with different transmitter antenna positions. There have been performed 30 simulations with the UWB1-3 setup. This was done in order to employ the MT-BCS algorithm for multi-positioning, and to have a large number of continuous positions for the tracking- and distance estimation part of the thesis.

To simulate a realistic localization setup, Gaussian noise have been added to the path loss measurements corresponding to receiver noise. Following [17] the amount of noise added was chosen to be SNR = 25 dB. The localization results that have been

Table 5.4: Results from localization using NB0.9, NB2.4 and UWB3-5 simulation setups. All frequencies are in GHz and the localization errors are in mm. $\mathcal{N}$ denote results corrupted with 25 dB SNR Gaussian noise.

Name	BW	f_l	f_h	Mean localization error							
				BP		BPDN		ST-BCS		AD-BCS	
					$\mathcal{N}$		$\mathcal{N}$		$\mathcal{N}$		$\mathcal{N}$
NB0.9	0.03	0.83	0.86	31.0	34.8	30.7	30.8	**22.7**	26.5	83.3	58.1
NB2.4	0.1	2.35	2.45	**19.0**	33.3	19.1	33.6	25.9	32.2	35.3	39.8
UWB3-5	0.2	2.9	3.1	55.4	53.5	61.6	55.1	**33.1**	42.4	65.8	50.8
	0.2	4.9	5.1	94.2	80.5	113.9	77.4	**46.9**	49.9	58.1	56.6
	0.2	3.9	4.1	75.6	74.8	62.3	73.1	59.3	**53.6**	66.3	53.9
	1.0	3.0	4.0	107.2	98.1	108.1	95.6	**40.0**	49.0	52.6	58.1
	1.0	4.0	5.0	83.3	90.2	66.8	93.8	67.0	**63.8**	123.2	65.5
	2.0	3.0	5.0	91.0	85.1	70.3	80.9	71.0	**58.9**	72.7	61.1

corrupted by noise are denoted by $\mathcal{N}$ in the result tables.

From the investigations performed in Sec. 5.3 these parameters were chosen for the localization: $N = 900$, $\gamma = 0.05$. All settings used for the localization algorithms can be found in Tab. 5.1. As outliers were shown to lower the localization accuracy, the outlier removal procedure from Sec. 4.3.3 was employed. All presented results are the mean error of 50 Monte-Carlo simulations for all available datasets for the given setup.

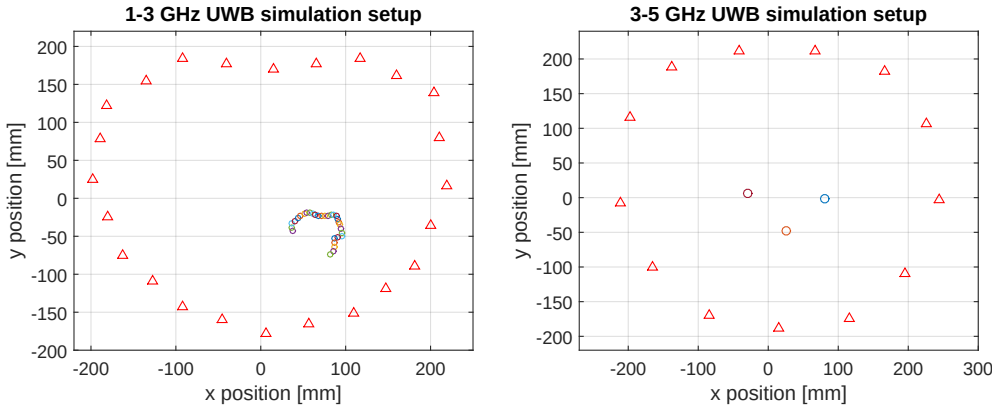
By empirical experiments it was found that including shadowing in the path loss model lowered the performance of the localization algorithms. Compensating for return loss mismatch in the antennas, following Sec. 4.1.8, was also found to provide minimal difference in the localization accuracy. Thus, the results presented in the following are without these compensations.

5.4.1 Narrow-band 915 MHz

The antenna placement used in the NB0.9 simulation setup is visualized in Fig. 5.12a. Due to the coil antenna having reduced directivity for direction $\phi = \{-90°, 90°\}$ in the elevation plane (Fig. 4.5a), the coil was oriented with the y-axis in Fig. 4.4a aligned with the z-axis of HUGO (Fig. 2.2).

A slight frequency shift from the intended f_c of 915 MHz was observed in the S_{11} results. Due to the long simulation time involved with these simulations, it was decided to keep the results with the center frequency being 850 MHz. Due to the wide bandwidth of the receiver antennas, this decision will not cause additional path loss.

The localization results can be seen in Tab. 5.4. The lowest accuracy is found when using the ST-BCS algorithm. For the noiseless situation a mean error of 22.7 mm is observed, increasing to 26.5 mm with noisy measurements. The BP and BPDN algorithms have an error of about 30 mm. The adaptive BCS algorithm is seen to have the lowest performance.



(a) The 1-3 GHz simulation setup. **(b)** The 3-5 GHz simulation setup.

Figure 5.13: The simulation setup used for the UWB simulations. The red triangles indicate the receiver antenna positions, while the circles show the transmitter antenna position. Note that the grid shown here has different dimensions than the grid used for localization.

5.4.2 Narrow-band 2.4 GHz

For the NB2.4 simulation setup, the antenna placement is shown in Fig. 5.12b. The localization results are found in Tab. 5.4. The best performance is found when using the BP algorithm, having an error of 19.0 mm. When adding noise, the ST-BCS algorithm has the lowest error of 32.2 mm.

5.4.3 Ultra-wideband 3-5 GHz

The simulation setup for UWB3-5 can be seen in Fig. 5.13b. Note that the measurements are obtained from fewer receivers than the other experiments performed. The reason for this, was that the size of the receiver antennas required more space between them. The localization was performed in the frequency ranges 3.0-3.1 GHz, 4.0-4.1 GHz, 3-4 GHz, 4-5 GHz and 3-5 GHz. The narrow band ranges were included to determine if there was any difference between NB and UWB on the same simulation setup.

The results from the localization can be seen in Tab. 5.4. The ST-BCS algorithm has the best performance for all frequency ranges. The lowest error is found in the NB range from 2.9-3.1 MHz with 33.1 mm. For the UWB ranges, best performance is found with a bandwidth of 1 GHz between 3-4 GHz with an error of 40 mm. Comparing the different algorithms, BP and BPDN have particularly large errors when using UWB.

5.4.4 Ultra-wideband 1-3 GHz

The simulation setup and points chosen for the UWB1-3 localization can be seen in Fig. 5.13a and Fig. 5.14. The average Euclidean distance between each of the 30 points are 5 mm. With the large number of simulations performed, the resolution of HUGO was reduced to $2\,\mathrm{mm}^3$ for this scenario, to save simulation time.

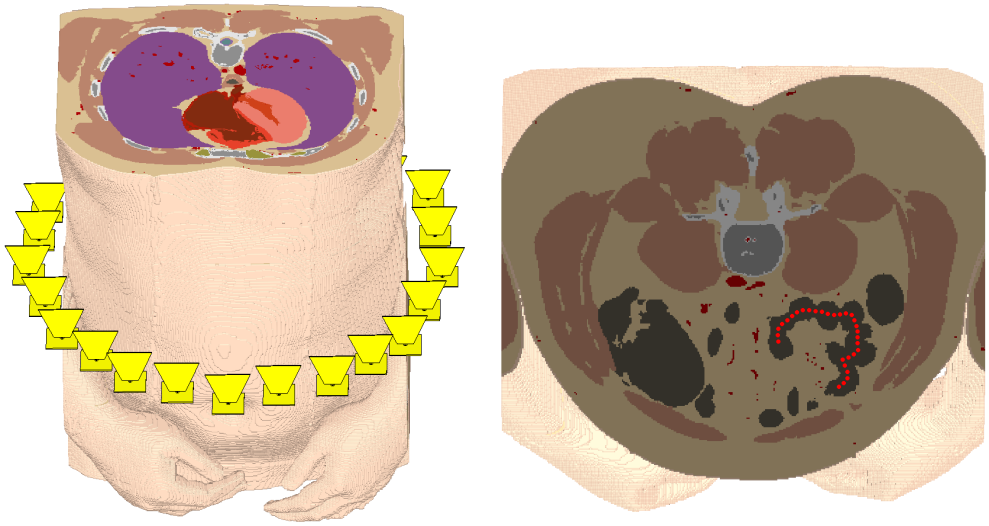


Figure 5.14: The position of the transmitter and the receivers for the multi-point localization.

It was chosen to use the NB frequency ranges 1.0-1.1 GHz, 1.0-1.2 GHz, 2.0-2.1 GHz, 1.9-2.1 GHz. Thus, results from both 100 MHz and 200 MHz bandwidth are available. Also the range 2.9-3.1 GHz was used, so that the results could be compared with the UWB3-5 simulations. For the UWB scenario, the frequency ranges 1.0-2.0 GHz, 2.0-3.0 GHz and 1.0-3.0 GHz were used.

In Tab. 5.5, the localization errors for the different algorithms are shown. For the 100 MHz narrow band experiment, the BP and BPDN algorithms have an error of about 33 mm for the noiseless scenario. Note that there is little difference in error for some of the algorithms when adding noise to the measurements. This could be an effect of the number of MC simulations runs being too low. Extending the bandwidth to 200 MHz, the results from the BP and BPDN algorithms are similar to the performance of the ST-BCS algorithm. The MT-BCS algorithm also performs well, with the lowest error of 47.1 mm from 1.9-2.1 GHz.

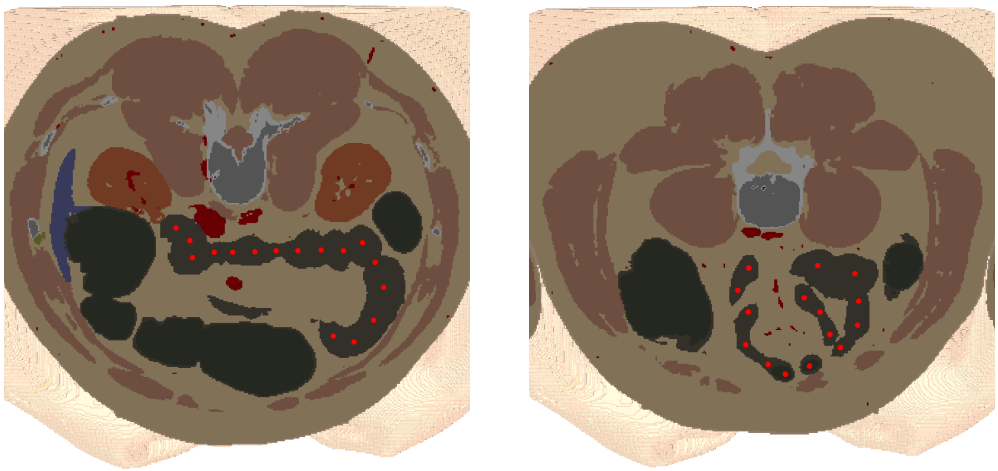
For the UWB bands, the accuracy is highest for the BP algorithm with the lowest error found as 35.9 mm for the 1-3 GHz band. The ST-BCS and BPDN algorithms are close in performance. MT-BCS and AD-BCS have slightly larger errors for all frequency bands.

Comparing the UWB with NB it was found that the latter provides slightly better performance. The accuracy is also more stable for all the algorithms, which could indicate a better robustness. The MT-BCS is found to have a better performance than the ST-BCS algorithm for the NB results. When the frequency band is extended to UWB, the results are opposite. Thus, it seems the MT-BCS has problems establishing the common hierarchical prior when there is more spreading in the data.

In Fig. A.7 in Appx. A, the localization performance is visualized by comparing the estimate from one of the MC simulations to the true positions.

Table 5.5: Results from localization using the UWB1-3 simulation setup. All frequencies are in GHz and the localization errors are in mm. $\mathcal{N}$ denote results corrupted with 25 dB SNR Gaussian noise.

			Mean localization error									
BW	f_l	f_h	BP	$\mathcal{N}$	BPDN	$\mathcal{N}$	ST-BCS	$\mathcal{N}$	AD-BCS	$\mathcal{N}$	MT-BCS	$\mathcal{N}$
0.1	1.0	1.1	34.4	33.9	**33.3**	33.9	46.1	45.9	60.1	55.3	38.7	44.7
0.1	2.0	2.1	**32.4**	48.1	32.7	48.0	46.5	54.1	51.8	55.6	48.4	44.8
0.2	1.0	1.2	43.8	**35.0**	46.1	35.1	54.8	44.2	64.4	54.3	36.0	45.6
0.2	1.9	2.1	49.8	50.5	52.3	50.6	51.7	55.1	58.8	57.1	**47.1**	51.8
0.2	2.9	3.1	41.7	54.1	**41.2**	54.4	44.8	61.1	48.5	58.7	42.4	49.2
1.0	1.0	2.0	49.2	46.8	47.1	**46.3**	53.1	56.9	63.4	60.6	66.6	57.1
1.0	2.0	3.0	46.8	40.7	46.5	**40.3**	44.2	45.6	64.1	53.5	60.6	60.2
2.0	1.0	3.0	**35.9**	52.1	37.7	51.5	43.4	59.6	66.1	61.7	64.5	60.7



(a) Dataset 1. **(b)** Dataset 2.

Figure 5.15: The points in the intestine used as a reference when creating datasets used for tracking.

Table 5.6: Parameters used when generating datasets for tracking, after [11].

Parameter	Value
$\bar{v}$	0.5 mm/s
σ_s	0.05 mm/s
t_s	5 min
σ_s	10 min

Table 5.7: Settings used for the Kalman and particle filters.

Description	Symbol	Value
Noise level	SNR	25 dB
Simulations	S	100
Number of particles	N	10000
Initial particle noise	σ_{N_0}	1
Length of mode 2	r	100
MA filter length	L	2

5.5 Tracking

The tracking performance of the filters was planned to be evaluated both by generated data and data obtained from the localization algorithm. As the points have to be continuous for tracking, the position data obtained from the UWB1-3 experiment is the only candidate for such usage. However, as is visualized in Fig. A.7 in Appx. A, the limited accuracy of the position estimates makes tracking this path difficult. Thus, it was decided to only use generated data sets for the evaluation. This was done in order to properly evaluate the performance of the filters under a more realistic scenario, where the position estimates are more accurate and the filters are able to track the path properly.

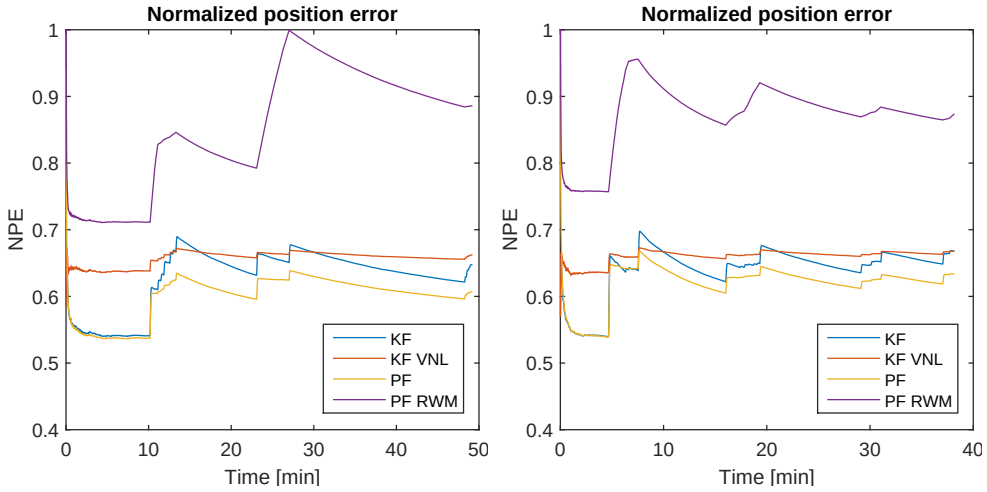
In the dataset generation, it was assumed that capsule positions were obtained every T seconds, with $T = 1\,\text{s}$. This is in line with the existing WCE systems [4]. The velocity was chosen as $\mathcal{N}(\bar{v}, \sigma_v)$, with the stop time modelled as $|\mathcal{N}(t_s, \sigma_s)|$ following the investigations done in [11]. The chosen values are summarized in Tab. 5.7. In Fig. 5.15, points used as reference when generating the two-dimensional datasets can be seen.

The performance of the filters will first be evaluated for the positional accuracy in Sec. 5.5.1, with the velocity and distance estimation performance presented in Sec. 5.5.2. The results are found from 100 Monte-Carlo simulations, where the data sets are generated once. The performance will be evaluated by the root-mean square error (RMSE), normalized positioning error (NPE) and the average difference in estimated length $\bar{\Delta}$.

The amount of noise added to the positions were fixed at SNR = 25 dB, as this was found to be a moderate noise level that is similar to the real tracking scenario if a high-performance localization algorithm is used. At the end of Sec. 5.5.2, distance estimates with varying noise levels will be presented.

5.5.1 Position estimation

To obtain the best accuracy in the position estimates of the filters, the filter parameters were tuned for minimum RMSE for the x and y positions. The RMSE results is found in Tab. 5.8. It can be seen that there are only minor differences between the



(a) Dataset 1. **(b)** Dataset 2.

Figure 5.16: Normalized position error as defined in Eq. (5.4), for tracking of generated datasets. SNR = 25 dB and run over 100 simulations.

Table 5.8: Tracking results with the filters tuned for lowest position RMSE.

Dataset	Filter	RMSE x	RMSE y	RMSE v	$\bar{\Delta}$ length
1	KF	0.11 mm	0.05 mm	0.04 mm/s	-41.88 mm
	KF VNL	0.12 mm	0.05 mm	0.05 mm/s	-79.62 mm
	PF	0.11 mm	0.05 mm	0.04 mm/s	-29.88 mm
	PF RWM	0.15 mm	0.07 mm		
2	KF	0.09 mm	0.11 mm	0.05 mm/s	-32.13 mm
	KF VNL	0.09 mm	0.11 mm	0.06 mm/s	-61.48 mm
	PF	0.08 mm	0.10 mm	0.05 mm/s	-16.96 mm
	PF RWM	0.11 mm	0.14 mm		

Kalman filter, the particle filter and the VNL Kalman filter. The particle filter with RWM model performs worse than the other filters. Looking at the mean difference in estimated length of the path, it can be seen that all the filters estimate a length that is slightly longer than the real length. This is a consequence of the fact that the models used in the filters are not matching the movement perfectly. By adding system noise, the position tracking works well, but with increased noise in the velocity estimates.

Fig. 5.16a and Fig. 5.16b show the normalized position error (NPE) of the two datasets. The NPE visualizes the tracking performance over time. From these plots it is found that the particle filter has slightly better performance than the Kalman filter. It can also be seen that the error increase when the capsule starts moving and when it stops in a bend in the intestine. An example of this behaviour is found in Fig. 5.16b where the capsule is at rest until $t = 5$ min, then starts moving, and stops at $t = 7.5$ min. The filter that is least affected by the starts/stops is the Kalman VNL filter, due to the changes in system noise that better fits the new movement state.

Table 5.9: Tracking results with the filters tuned for best distance accuracy.

Dataset	Filter	RMSE x	RMSE y	RMSE v	$\bar{\Delta}$ length
1	KF	0.84 mm	0.35 mm	0.10 mm	5.76 mm
	KF VNL	0.42 mm	0.16 mm	0.09 mm	1.72 mm
	PF	0.28 mm	0.08 mm	0.07 mm	8.73 mm
2	KF	0.54 mm	0.72 mm	0.10 mm	3.81 mm
	KF VNL	0.30 mm	0.33 mm	0.09 mm	1.86 mm
	PF	0.14 mm	0.21 mm	0.07 mm	10.23 mm

5.5.2 Velocity and distance estimation

In order to get better accuracy in the estimated length, the filters were tuned for the minimum $\bar{\Delta}$. For all filters, it was found that this resulted in less noise added to the system. Thus, the velocity estimates from the movement model are less noisy and provide a better length estimate.

The results from 100 simulations on the datasets can be seen in Tab. 5.9. The Kalman filters have a good performance, with the result being accurate to within a few millimeters. The VNL Kalman filter is shown to be slightly better than the Kalman filter.

Looking at the velocity RMSE it might seem strange that it has increased compared to the results in Tab. 5.8, while the distance estimates are more accurate. This is explained by the fact that the filters track the abrupt changes in velocity slower with less system noise. Thus, the estimated velocity is more accurate and less noisy, but with a slight lag compared to the true velocity that affects the RMSE performance.

An example of varying system noise can be seen in Fig. 5.17, where the velocity of the different filters are compared. The VNL Kalman filter tracks the changes in velocity better than the Kalman filter, due to detecting the maneuver and changing the system noise. The particle filter is even faster in detecting the velocity change, but at the expense of overshoot and more noise in the estimates.

As the filters are tuned at a specified noise level, it was thought interesting to compare the effect of the observation noise on the distance estimates. The result can be seen in Fig. 5.18a for dataset 1 and Fig. 5.18b for dataset 2, where SNR levels between 10-45 dB have been used. For dataset 1, the VNL Kalman filter has the most stable performance. For dataset 2, the Kalman filter performs best for large amounts of noise. The particle filter is most susceptible to high levels of noise for both datasets. As the filters are tuned for operation at 25 dB SNR, the full performance of the VNL Kalman filter is not utilized, as the increased noise cause problems in detecting the real maneuvers.

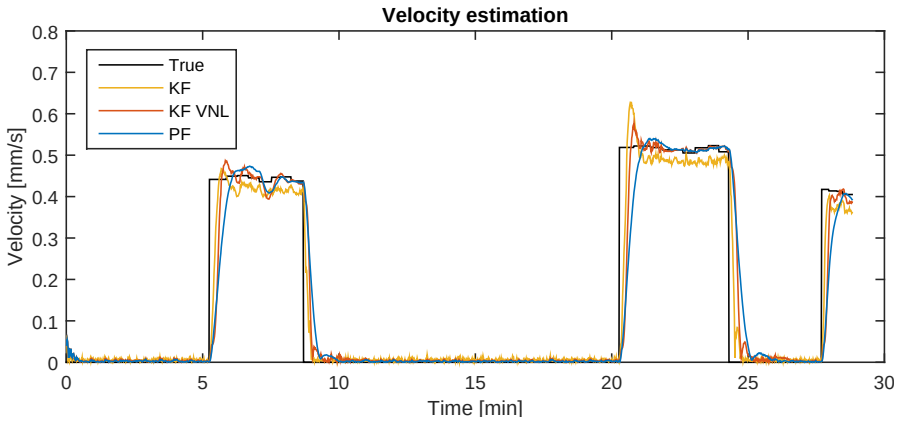
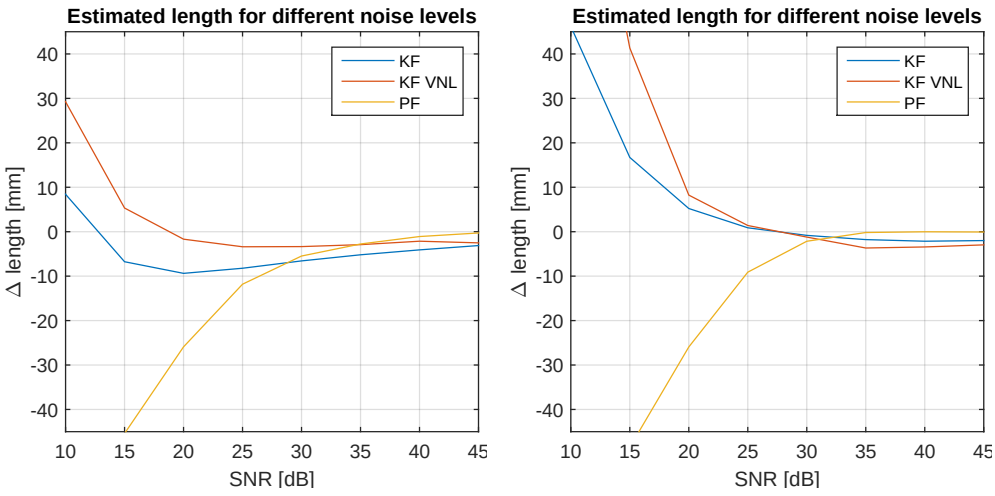


Figure 5.17: Estimated velocity for all filters from one of the simulations with dataset 1.



(a) Dataset 1. **(b)** Dataset 2.

Figure 5.18: $\bar{\Delta}$ length for different SNR levels.

Chapter 6

Discussion

Summarizing the results, it was found that the EM simulation path loss data corresponds with the UWB path loss model from [74]. The compressive sensing localization framework performed well when using generated path loss data, with a localization accuracy of a few millimeters. Adding noise to the path loss measurements increased the errors, but the BCS algorithms were able to provide a decent performance. When using path loss data from the EM simulations, the performance was reduced to a minimum Euclidean error of approximately 20 mm using NB signals at 2.4 GHz. The performance when using UWB signals was found to be slightly lower in accuracy compared to NB, having highest performance with an error of 35.9 mm between 1-3 GHz.

For the tracking problem, the distance estimates had lowest errors when using the VNL Kalman filter. When the observations were corrupted by noise with 25 dB SNR, the error was approximately 2 mm. Under large amounts of noise, with the filters tuned for a lower SNR level, the VNL Kalman had problems detecting the mode changes. This increased the error in the estimated distance. For this specific scenario, the Kalman filter performed better.

6.1 EM simulation results

In the EM simulation results, it was found that the path loss was highly affected by the implant depth. In addition, the results showed a high degree of frequency dependence, with higher attenuation for increasing frequency, and being more prominent for certain frequencies. These results correspond with the path loss investigations performed in [74, 89, 92, 97].

For the experiment comparing the UWB 1-6 GHz path loss model from [74] with simulations, it was expected that the results were similar, considering the fact that the transmitter antennas are equal. However, the model was made from path loss measurements in the range 50-160 mm. In the simulation setup, the average distance between the transmitter and receivers is 227 mm. Thus, the model operates beyond its specification. One other thing to consider is the distance between the body and the receiver antennas. In the measurement setup in [74], the antennas were as close to the body as possible. The antennas used in the simulations are placed a few centimetres off

the body, to allow space for the antenna without touching the body. This additional path loss could have been included in the path loss, but was considered insignificant compared to the large attenuation experienced in-body.

In the same experiment with the 1-3 GHz planar loop, a mismatch was found in the frequency range 1-1.5 GHz. This could partly be explained by the non-flat return loss of the transmitter and receiver antennas (Fig. 4.10b,4.11b). However, by correcting for this power loss following Sec. 4.1.8, a mismatch was still found. This is shown in Fig. 6.2. One reason for this behaviour could be the non-symmetric directivity of the antenna, which affects the radiated power. But looking at Fig. 4.14a and Fig. 4.14b reveals that the variation in antenna directivity is too low to cause this behaviour. The separation between the receiver antennas should not cause problems, following the mutual coupling investigations performed with this receiver antenna. Thus, the best explanation for the differences are factors in the simulation setup, with one possible cause being tissue variations in the path between the transmitter and receiver antennas.

It was found that using the developed scripting framework simplified the process of performing the simulations. New receiver and transmitter simulation setups could be created with ease, once the functions that generated the antennas were made. The traceability provided by the scripts also proved valuable when specific simulation details were needed at a later time. It is worth mentioning that the functionality in the simulation framework that provides frequency dependent tissue properties was found available in CST. Due to poor documentation, this was not realized until after the functionality was implemented. However, it was found that this functionality in CST was not working properly in the version of CST used for the simulations. Thus, it proved to be valuable to have this covered in the framework.

6.1.1 Error sources

Although the EM simulation results correspond with the path loss models, there exist some potential error sources in the measurements. As the EM fields are computed in each mesh cell, the meshing of the model in CST is critical for an accurate result. With the mesh settings used in this thesis, an acceptable accuracy has been found that could be handled by the available computing power. However, by increasing the mesh density, it is certainly possible to get an even better accuracy in the simulation results. The results could also be compared with results obtained from another EM simulation software, e.g. Ansoft HFSS, to verify that the computed data are valid.

One other point to consider is the shape and size of the voxel model. For the simulations, only one male body has been used. In order to get results representable for a larger amount of the human population, simulations should be performed on voxel models with different body shapes, gender and age.

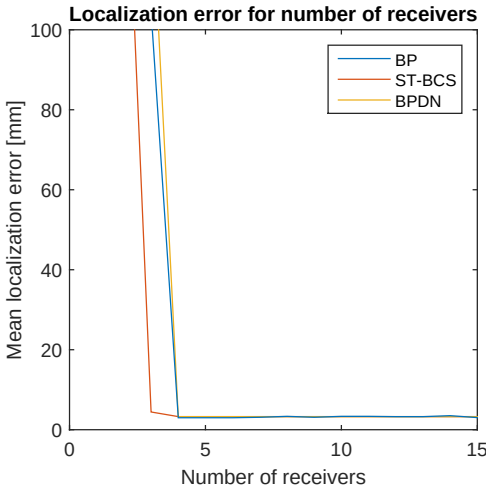


Figure 6.1: Localization error when using the free space log-distance path loss model (Eq. 2.5), with the rest of the setup equal to Fig. 5.8.

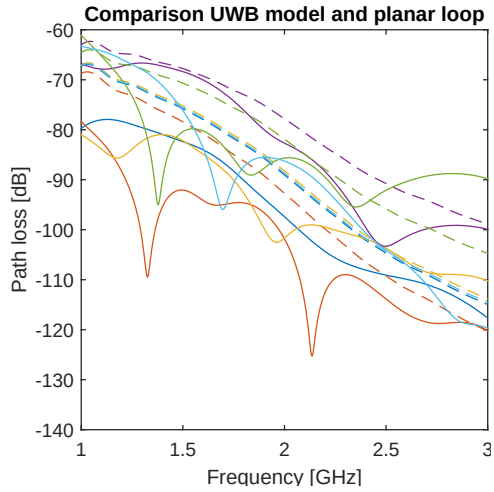


Figure 6.2: Comparison of UWB 1-6 GHz path loss model and simulated path loss for the 1-3 GHz planar loop setup as in Fig. 5.3b, but with the simulated path loss values compensated for the mismatch introduced by S_{11} and S_{22}.

6.2 Localization

6.2.1 Generated path loss data

Localization using path loss data generated from the 1-6 GHz UWB model in Eq. (4.29) showed that recovery of positions on the order of millimeters are possible if the path loss data is noise free and enough measurements are available. The results show that fewer receivers are required for accurate position recovery using Bayesian CS compared to the BP algorithm, which corresponds with findings published in [33, 36, 95].

The result from Fig. 5.8 showing that the BP algorithm needs more than 30 receivers for exact recovery does not correspond with the results from [17], where exact recovery of four targets is achieved with 10 measurements. However, these results were made for the idealized problem when the targets are positioned exactly at the center of the grids. The results in [17] are also different from those presented in [32], where the number of receivers required for perfect recovery of 10 targets in a in 20×20 m grid are found to be more than 100.

The path loss model was also found to be a major factor determining the number of measurements needed for exact recovery, which is shown in Fig. 6.1. These results are obtained by using the log-distance path loss model for free space in [17] with the same setup used in Fig. 5.8. It can be seen that the number of observations satisfy Eq. (2.14) from CS theory. It is assumed that the extreme in-body attenuation cause this behaviour. The sparsity matrix $\mathbf{\Psi}$ contains path loss values ranging between 1 and 1×10^{18} , when treating UWB localization up to 5 GHz. It seems reasonable that additional measurements are needed in order to find the sparsest localization vector under these circumstances.

When noise was added to the measurements, the Bayesian algorithms had a better performance than the BP and BPDN algorithms. This is in line with intuition and is also found in the BCS works [34, 35] and in localization approaches [33, 36, 95]. The results showed that the multi-task BCS algorithm was able to outperform the other algorithms when large amounts of noise was added to the measurements. In [33, 95], the performance difference between MT-BCS, ST-BCS and BP is shown to be even larger than the results presented in this thesis. However, these results were from localizing more than one target, and therefore have more signal correlation to exploit in the localization process.

6.2.2 Simulated path loss data from EM experiments

The localization results when using path loss data from the in-body EM simulations showed an average error of 20-30 mm when using the NB setup. This is similar to the results found in [24] for 2D positioning systems that utilize only RSS measurements. The articles [26, 98] presents accuracies of 1 and 5 mm respectively, but the methods are not validated on actual RSS measurements. For the three-dimensional scenario, [25] presents results with RMSE of 50-120 mm for similar number of receivers used in this thesis. Compared to the WCE localization approaches that use TOA [12, 20], the accuracy of the CS localization is much lower. But these algorithms face practical challenges in maintaining accurate receiver/transmitter clock synchronization.

Few other articles have assessed the performance of the developed algorithms by using EM simulations. As was shown in Fig. 5.8, excellent performance is achievable when using path loss data generated from a model. However, as revealed from the findings in Sec. 5.4, such results doesn't necessarily translate well when using actual in-body path loss data. The main reason for this behaviour is the inhomogeneity of the body tissues, that cause the path loss measurements to have large deviation for equal distances. Thus, the path loss model becomes a coarse indication of the actual loss experienced. This can be compensated to some degree by including shadowing in the model. However, this was not found to have any significant performance increase for the CS localization framework.

As mentioned, the CS algorithms had problems estimating the sparse location vector $\hat{\boldsymbol{s}}$ when using non-decibel in-body path loss measurements. The estimated $\boldsymbol{s}$ was found to contain many large values, rather than a few values in the range 0 to 1. It was considered if it could be a numerical problem, due to the large values in $\boldsymbol{\Psi}$. But scaling or shifting the values in $\boldsymbol{\Psi}$ had no effect. Thus, it seems to us that the poor performance when using non-decibel path loss measurements is the result of the large amount of attenuation experienced in the in-body localization scenario combined with the large spreading found in the data. When using CS for localization in WSN and WLANs, the path loss is more predictable than the in-body scenario. By using decibel measurements, the CS framework has an easier job finding the sparsest solution. However, this comes at the cost of lower accuracy for the BP and BPDN algorithms, as was shown in the results in Sec. 5.3.

6.2.3 Narrow-band and wide-band comparison

Looking at the localization results using path loss data from the in-body simulations, it can be seen that the performance is higher when using NB signals compared to UWB. The lowest mean localization error is found as 19.0 mm when using the 2.4 GHz frequency band with a bandwidth of 100 MHz. However, the NB2.4 results are based on only three datasets, which makes the uncertainty quite high. The 1-3 GHz UWB results are from on 30 datasets, and are thus more reliable. The minimum error experienced for the NB case when using UWB1-3 was found as 32.4 mm between 2.0-2.1 GHz. When using UWB for the same setup, best accuracy was obtained as 35.9 mm from 1-3 GHz.

Comparing the UWB1-3 and UWB3-5 simulation results, it was found that the difference between NB and UWB was larger for the latter. This could be due to the increased path loss experienced for the 4-5 GHz band. As can be seen in the path loss plots in [74], the attenuation is severe for these frequencies when there are large distances separating transmitter and receiver. Due to the large attenuation, there is a larger probability of mismatch between the model and the measurements.

For the frequency band 2.9-3.1 GHz that is used for both UWB1-3 and UWB3-5, the results show that the minimum error was found with the UWB3-5 setup using ST-BCS. However, looking at the results of all the algorithms, the UWB1-3 has a more stable performance, with all the other results being better than UWB3-5. Apart from the UWB1-3 results being based on more data, one explanation for this behaviour could be the directivity of the transmitter and receiver antennas. The UWB1-3 has a smooth, omnidirectional directivity (Fig. 4.14), whereas the DRA antenna has more variation for different directions and frequencies (Fig. 4.15). This is also the case for the receiver antennas, where the directivity is more ideal for the trapezoidal monopole (Fig. A.4), compared to the circular monopole (Fig. A.5).

No studies have been found comparing the performance of UWB and NB for capsule localization. Thus, such results must be found elsewhere. According to [43], using UWB for RSS localization will not increase accuracy, as the distance ranging is not dependent on the signal bandwidth. The only articles found using UWB for RSS localization are [73, 99]. Elsewhere in the literature, the authors that employ UWB for positioning use a hybrid estimation method combining RSS with TOA [43, 100, 101, 102]. This is also the case in [12] for WCE usage. Thus, it seems that the results found in the comparison between NB and UWB in this thesis correspond with the literature.

One benefit of UWB is resilience to frequency dependent shadowing. A system employing few receivers and NB signals will be susceptible to shadowing if a large portion of the small frequency spectrum is affected. Few sensors give rise to low reliability. UWB covers a wider bandwidth, in which the shadowing will be lower in parts of the spectrum. Thus, in theory, the localization will be more reliable under these circumstances.

Another benefit of using UWB is the fact that the wide frequency spectrum enables high data rates compared to NB systems [103]. This is a large benefit for WCE, as this enables better visual quality and higher frame rates on the video, which is one of the WCE subjects that currently receives attention [1].

6.2.4 Error sources

When choosing the receiver positions for the NB and UWB simulation setups, the positions were not chosen to fit with the grid used for the CS localization. This was done to be closest to the real problem of localizing the capsule, where the receiver locations will not be known exactly. This mismatch between the grid and the receiver positions was found to lower the accuracy when using generated path loss data, but the error will likely be minimal with the coarse data provided by the in-body path loss measurements.

In addition to the receiver positions in the grid, also the distance between the body and the receiver antennas will be a possible error source. The properties of the body tissues will affect the antenna when the antennas are close to the body. In the antenna system, one other error source is the non-uniform directivity of both transmitter and receiver antennas. As the WCE orientation will not be known, it is difficult to compensate for this behaviour. Other factors are the influence from the antenna matching not being perfect for all frequencies, and that the exact transmitted power in a real system may be unknown.

Lastly, in the EM simulations the capsule housing of the WCE has not been included. The thin plastic shell will probably cause minor differences in the transmitted power and the directivity of the antenna.

6.3 Tracking and distance estimation

In [38] the proposed video distance estimation algorithm achieves an accuracy of 2.71 cm, for a 41 min WCE path with velocity between 0-4 mm/s. Comparing this to the minimum accuracy result of $\sim$2 mm presented in Sec. 5.5.2, it is apparent that the Kalman VNL filter has a better performance for this specific setup. However, the datasets used for evaluation are not very similar. In [38], it is assumed that the capsule constantly changes velocity between 0-4 mm/s, without stops. As mentioned in Sec. 5.5, the datasets used in this thesis assume that there is only minor deviations in the velocity when the capsule is moving, and that the capsule movement consist of multiple stops.

A better comparison can be made if we assume an average velocity of 2 mm/s in [38], which gives the path a total length of $d = 500\,\text{cm}$. The longest dataset from this thesis has a length of approximately $d = 30\,\text{cm}$. By extrapolating the error of 2 mm to a dataset of length $d = 500\,\text{cm}$, an equivalent error is found as 3.3 cm. Thus, for similar length the performance is a little better when using [38]. However, it is unknown how well this algorithm performs when facing datasets that contain long periods with no capsule movement.

The result that the VNL Kalman filter was found having most accurate distance estimation correspond with [53], where it is presented that multi-mode estimation has better performance for tracking a maneuvering target. However, to track the maneuvers with high accuracy, the problem of detecting changes in mode is critical. It was seen in Fig. 5.18b that the VNL Kalman filter had problems detecting modes properly with changing amounts of observation noise. The performance could probably have been increased if a more robust test statistic was used for the mode detection.

More advanced statistics are investigated in [11], which show better tracking of the maneuvers.

Considering the results presented in this thesis, there is a significant uncertainty in the results since only two datasets are considered. Although the velocity is generated randomly, the movement does not change from one simulation to the next. The main reason for not testing with more datasets, was that it proved difficult to find parts of the intestine with only two-dimensional movement. This is illustrated in Fig. 3.4a, where the intestines of the HUGO model are shown.

6.3.1 Error sources

In the simulations, it has been assumed that the amount of observation noise is known. For clinical use, the noise will not be known exactly. The effect of the unknown observation noise on the distance estimates will depend on the accuracy of the localization algorithm used. If the accuracy is low, poor distance estimates will result regardless of the amount of observation noise on the positions.

The length of the datasets used is also a probable error source. By the extrapolation performed in Sec. 6.3, an approximate distance was found. But, when facing longer datasets, it is possible that larger errors will occur and accumulate to a lower estimation performance.

One other point to consider, is the problem of tuning the performance of the filters. When the distance is known, as in the simulations, this is not a problem. In a clinical setting, the distance will not be known. This also causes problems when evaluating the distance estimation performance clinically.

6.4 Computational complexity

The computational complexity of the BP and BPDN algorithms are low, as the optimization problem is implemented as a linear program. For the BCS algorithms, the major computational bottleneck in the algorithm is the explicit matrix inversion of $\mathbf{\Sigma}$ that is needed in the iterations. In theory this has a complexity of $\mathcal{O}(N^3)$, and the inversion uses some time when N is large. The elapsed time varies with the iteration convergence, but is typically 1-2 sec. for a grid of $N = 900$.

When extending the localization system to three dimensions, the sparsity of the signals will increase, as well as the computational complexity. To utilize the same grid resolution as used in Sec. 5.4, we have $N = 30^3$ in three dimensions. With the complexity being fairly high for the BCS algorithms, other solutions must be employed for effective usage. One such candidate is presented in [34], where an approximate recovery algorithm is shown to have faster performance than the direct approach used in this thesis.

The complexity of the Kalman filters are low, as these filters are run recursively. The particle filter has a higher complexity, increasing with larger number of particles. For the simulations performed in this thesis, the particle filter used approximately 5 sec. to estimate one of the datasets with 10000 particles.

For clinical use, real time performance is not a requirement. The localization could be performed with stored RSS measurements, and the distance estimation is performed by use of the positional data. However, for the localization, the complexity of the system decreases if the positions could be obtained in real time. By reducing the grid resolution and optimizing the performance of the localization algorithms, this could be feasible.

6.5 Future work

Although interesting results have been presented in this thesis, more work must be done in order to use the localization and tracking system clinically. Some suggestions for future work will be presented in the following.

The main problem that has not been considered in this thesis, is the extension to three dimensions. As previously mentioned, both the localization and tracking system can be extended to three dimensions following the same approach as this thesis. It is expected that the performance will decrease when this extension is done due to the increased dimensionality of the system. However, some new possibilities arise when operating in 3D e.g. different receiver constellations. This issue has been addressed to some degree in [25], but it seems likely that more work can be done on this topic.

One weakness with the results presented in this thesis is that the amount of tested datasets could have been higher. The UWB1-3 results are somewhat more reliable, being based on 30 datasets. But the performance of the NB and UWB3-5 setups have only been tested with three different sets, thus lowering the reliability of the results. In order to get a proper conclusion in the comparison between NB and UWB, more simulations should be performed. This is also true for the tracking and distance estimation results, although the random generation of the two datasets used in this thesis limits this problem to some degree.

For the localization algorithm, increased performance can likely be achieved if a better selection of receivers are employed. The rudimentary outlier detection and removal developed in this thesis increased the performance compared to no removal. A more advanced scheme, where only the best measurements are used, should yield better results. One possible starting point for this topic is the usage of the variance estimates of the BCS algorithm.

The tracking method can be developed further by integration with the localization algorithm. It is likely that the performance can be increased by turning this in to a one-step process. Another interesting direction would be to combine RSS localization and video tracking, as in [8], but with more focus on the distance estimation part of the problem.

Chapter 7

Conclusion

This thesis has investigated localization and tracking of a WCE in the intestines. The localization has been performed by RSS measurements using compressive sensing in the frequency bands 915 MHz, 2.4 GHz, 1-3 GHz and 3-5 GHz. The RSS measurements have been obtained from in-body EM simulations on a human voxel model. The tracking has been done by Kalman- and particle filters, and the estimated velocity has been used to determine the distance the WCE has travelled in the intestines.

It was found that localization accuracy of a few millimeters is possible under ideal conditions, when the RSS measurements are generated from a path loss model. When using path loss data from the EM simulations, the accuracy was lowered to a minimum error of 19 mm when using NB signals. Use of UWB signals resulted in localization errors between 35-60 mm, depending on frequency range and bandwidth. Thus, no benefit was found when using a wider bandwidth. The EM simulation path loss results was found to correspond with the UWB 1-6 GHz path loss model in [74].

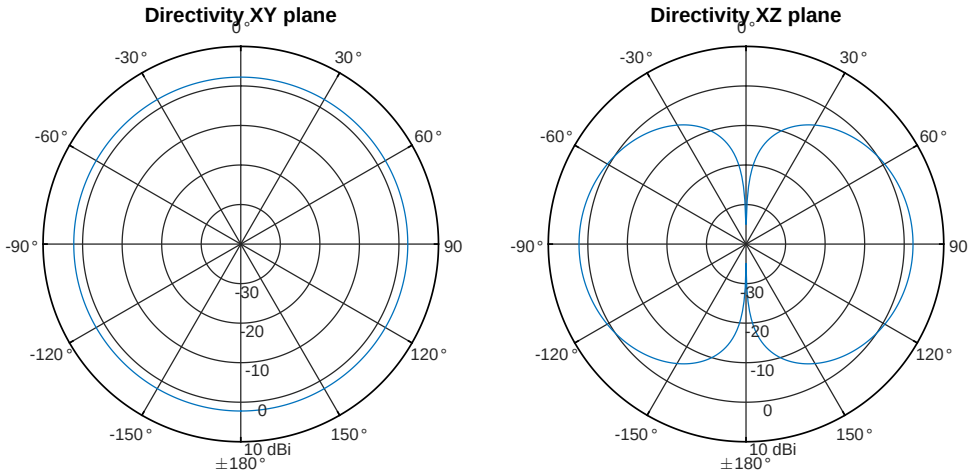
Due to limited accuracy in the position estimates from the localization algorithm, generated datasets were used to evaluate the tracking- and distance estimation performance of the filters. From position data generated from the intestines, it was found that the travelled distance can be found with an error of a few millimeters under moderate levels of observation noise, when using the VNL Kalman filter. Under large amounts of noise, with the filters adjusted for a lower level of observation noise, the VNL Kalman had problems detecting the mode changes. This increased the error in the estimated distance. For this specific scenario, the Kalman filter performed better.

The results are found from a limited amount of data. In order to increase the confidence in the presented results, the performance of the localization and tracking algorithms should be evaluated with more datasets.

Appendix A

Attachments

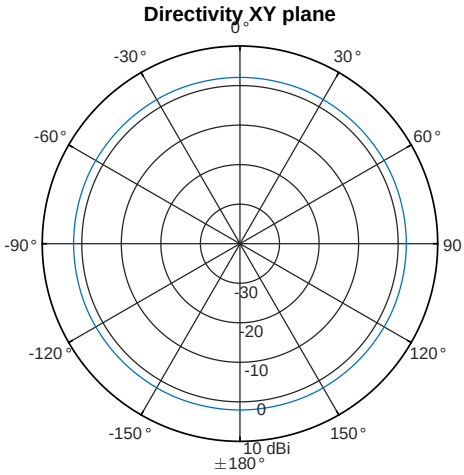
A.1 Plots



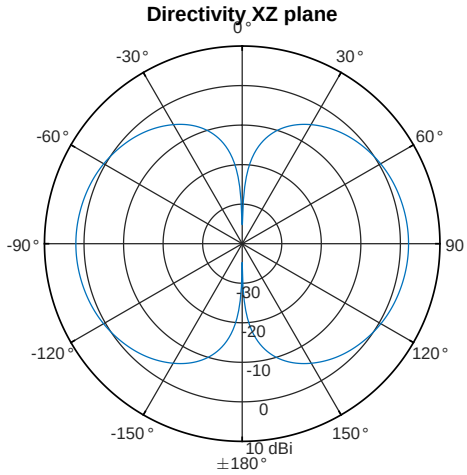
(a) Directivity in the azimuth plane.

(b) Directivity in the elevation plane.

Figure A.1: Directivity for the 915 MHz half-wave dipole antenna.

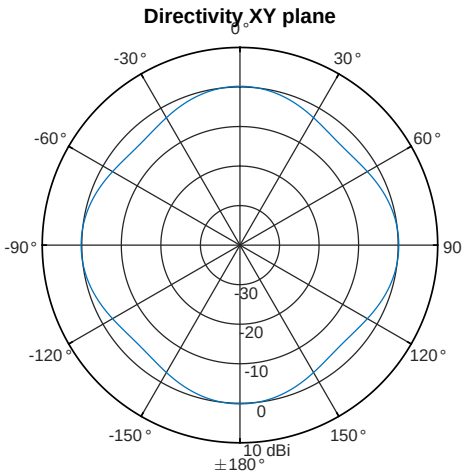


(a) Directivity in the azimuth plane.

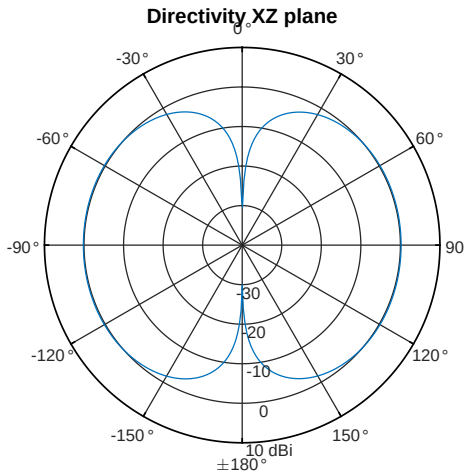


(b) Directivity in the elevation plane.

Figure A.2: Directivity for the 2.4 GHz half-wave dipole antenna.

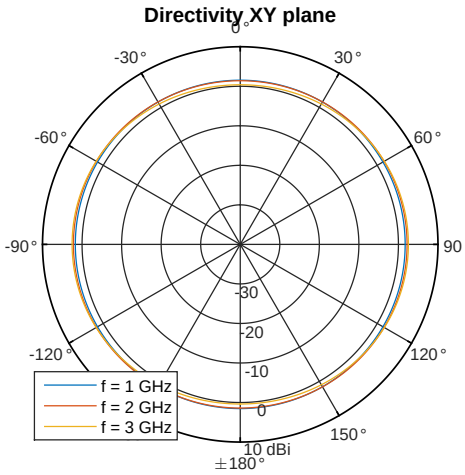


(a) Directivity in the azimuth plane.

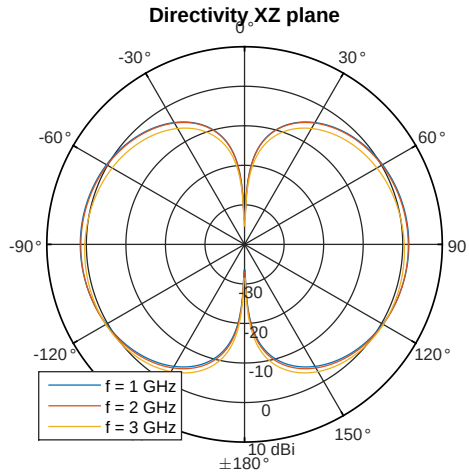


(b) Directivity in the elevation plane.

Figure A.3: Directivity for the in-body 2.4 GHz half-wave dipole antenna.

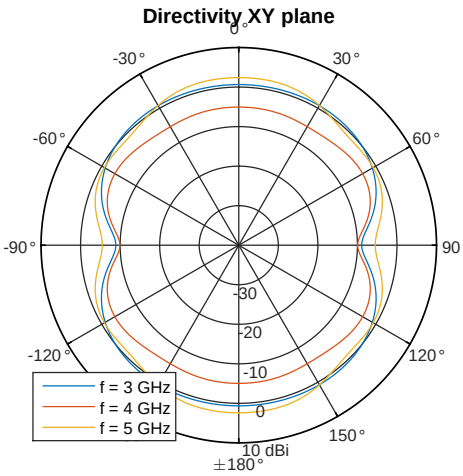


(a) Directivity in the azimuth plane.

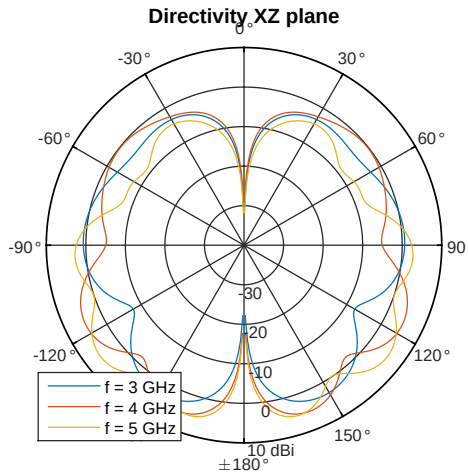


(b) Directivity in the elevation plane.

Figure A.4: Directivity for the 1-3 GHz trapezoidal monopole antenna.



(a) Directivity in the azimuth plane.



(b) Directivity in the elevation plane.

Figure A.5: Directivity for the 3-5 GHz circular monopole, shown for the center frequency and the upper and lower boundary.

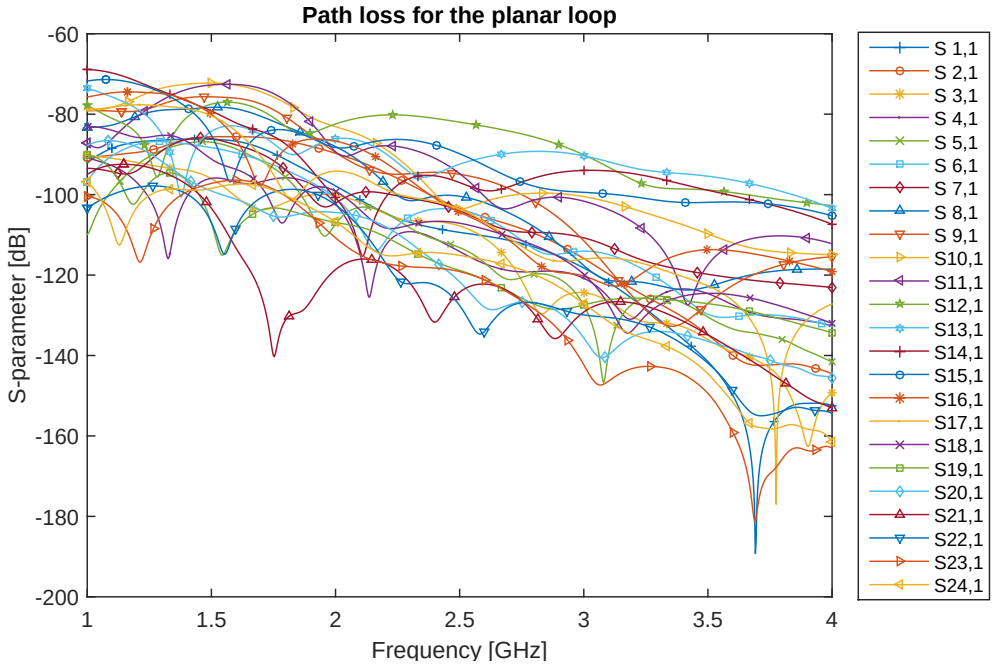


Figure A.6: Path loss results from one EM simulation for the 1-3 GHz planar loop, with all 24 receiver antennas shown.

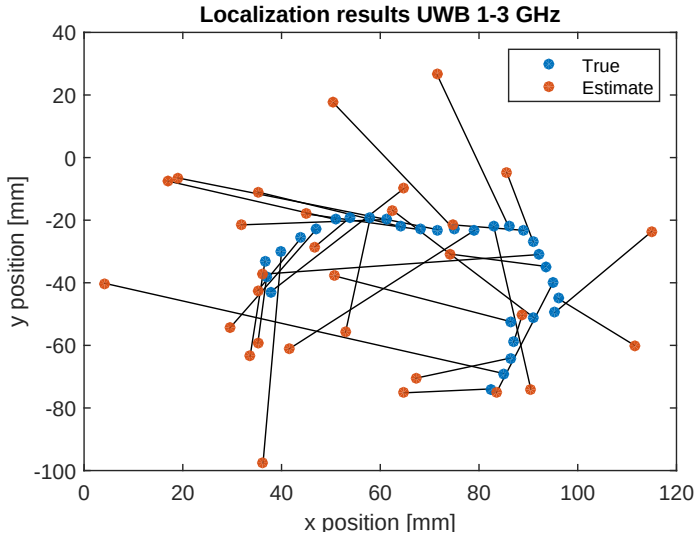


Figure A.7: Localization results visualized for one run using the UWB 1-3 GHz setup with range $f = 1 - 1.1$ GHz and the MT-BCS algorithm. The mean localization error is 35.5 mm.

A.2 Tables

Table A.1: Dimensions of the 915 MHz coil antenna.

Description	Name	Dimension
Length	l	5 mm
Radius	r	2.1 mm
Wire diameter	t	0.4 mm
Turns	N	5

Table A.2: Dimensions of the trapezoidal monopole antenna.

Description	Symbol	Value
Trapezoid width top	l_{t_1}	36.03 mm
Trapezoid width bottom	l_{t_2}	16.37 mm
Trapezoid height	h	36.21 mm
Ground plane depth	l_d	31.55 mm
Ground plane width	t_w	31.55 mm
Ground plane height	t_g	1.25 mm
Gap	h_g	1.38 mm

Table A.3: Dimensions of the circular monopole antenna.

Description	Symbol	Value
Circle radius	R	49.22 mm
Ground plane depth	l_d	59.62 mm
Ground plane width	t_w	59.62 mm
Ground plane height	t_g	4.47 mm
Gap	h_g	2.74 mm

Table A.4: In-body to off-body path loss model parameters for 402, 868 and 2400 MHz [90]. Valid for d = 100-500 mm.

Frequency	Measured n	Simulated n
402 MHz	1.90	1.85
868 MHz	2.00	1.90
2.4 GHz	2.80	2.60

Table A.5: UWB 1-5 GHz in-body to on-body path loss model parameters [92].

d [mm]	$PL_{0,dB}$	n	μ_s	σ_s
10			0.30	1.87
20			1.47	4.25
30			3.39	5.92
40			4.47	7.12
50			4.78	7.49
60			5.74	8.32
70			6.92	9.07
80	1.407	3.397	7.45	9.37
90			7.11	9.08
100			7.14	8.65
110			7.52	8.36
120			7.11	8.15
130			7.67	8.69
140			5.18	7.10
150			6.29	7.18

Table A.6: Mapping of tissues between HUGO and Gabriel's data.

HUGO	Gabriel
Marrow	BoneMarrow
FatTissue	Fat
Bones	BoneCortical
WhiteSubstance	BrainWhiteMatter
GraySubstance	BrainGreyMatter
Skin	SkinDry
Eye	EyeSclera
SkeletonMuscle	Muscle
Blood	Blood
NeuronalFabric	Cerebellum
Lens	Lens
NervusOpticus	Nerve
Cartilages	Cartilage
MucousMembrane	MucousMembrane
Air	Air
Lung	LungInflated
Intestine	SmallIntestine
Kidney	Kidney
Liver	Liver
Glands	Gland
Spleen	Spleen
Stomach	Stomach
Pancreas	Pancreas
Bladder	Bladder
GallBladder	GallBladder
IntestineContents	SmallIntestine
VentriclesRight	Heart
VentriclesLeft	Heart
ForecourtRight	Heart
ForecourtLeft	Heart
BloodV	Blood
BloodA	Blood

Appendix B

Matlab code

Parts of the MATLAB code used in this thesis are shown in this appendix. Only a small portion is included out of relevance and length constraints. Code used for localization is found in Sec. B.1, while filters used for tracking the intestinal paths are shown in Sec. B.2.

B.1 Localization

Listing B.1: The compressive sensing class `CompressiveSensing.m`.

```
classdef CompressedSensing < Localization

properties
    N;
    M;
    O;
    c = 3e8;
    Psi;
    Phi;
    Theta;
    unit;
    grid@Grid
end

methods
  % Create the CS object
  function obj = CompressedSensing(N, unit)
    obj.N = N;
    obj.unit = unit;
  end

  % Localization
  function [x,X,theta,Sigma] = localize(obj,y,gamma,algo,ortho,iter,sigma)
    % Settings
    tolerance = 1e-9;%1e-9;
    epsilon = sigma;
    lambda0 = 1/sigma^2;
    numIt = iter;
    Eb = 0.000001;

    % Orthonormalization
    Theta = obj.Phi*obj.Psi;
    Q = orth(Theta')';

    if ortho
      y_mark = Q*pinv(Theta)*y;
    else
      y_mark = y;
      Q = Theta;
```

```
    end

    % Initial starting point BP
    theta_0 = Q'*y_mark;

    % Use the specified l1 minimization algorithm
    switch algo
      case 'BP'
        theta = l1eq_pd(theta_0,Q,[],y_mark,tolerance);
      case 'BPDN'
        theta = l1qc_logbarrier(theta_0,Q,[],y_mark,epsilon,tolerance);
      case 'ST-BCS'
        [theta,Sigma] = ST_BCS(y_mark,Q,lambda0,0,obj.M,obj.N,numIt,Eb);
      case 'MT-BCS'
        disp('Use the multi-task localization algorithm instead');
      otherwise
        error('Please provide a valid algorithm');
    end

    % Find the position as the centroid of values larger than gamma
    [x,X] = obj.findCentroid(theta,gamma);
  end

  % Multi-task localization
  function [res] = localizeMT(obj,y,Phi,Psi,gamma,algo,ortho,iter)
    % Settings
    P = length(y);
    lambda0 = 1/0.01^2;
    numIt = iter;
    Eb = 0.000001;

    % Orthogonalization
    if iscell(Phi) && iscell(Psi)
      Theta = cellfun(@(a,b) a*b,Phi,Psi,'UniformOutput',0);
      Q = cellfun(@(a) orth(a')',Theta,'UniformOutput',0);
      if ortho
        R = cellfun(@(a,b) a*pinv(b),Q,Theta,'UniformOutput',0);
        y = cellfun(@(a,b) a*b,R,y,'UniformOutput',0)';
      end
    else
      Theta = obj.Phi*obj.Psi;
      Q = orth(Theta')';

      if ortho
        y = cellfun(@(a,b) b*a,y,repmat({Q*pinv(Theta)},1,P),'UniformOutput', 0)';
      end
    end

    % Localization
    switch algo
      case 'BP'
        theta = cell(1,P);
        for i=1:P
          theta_0 = Q'*y{i};
          theta{i} = l1eq_pd(theta_0,Q,[],y{i},1e-9);
        end
        mu = theta;

      case 'ST-BCS'
        mu = cell(1,P);
        sig = cell(1,P);
        for i=1:P
          if i==1
            [mu{i},sig{i},lambda0,lambda] = ST_BCS(y{i},Q{i},lambda0,0,obj.M,obj.N,50);
          else
            [mu{i},sig{i},lambda0,lambda] = ST_BCS(y{i},Q{i},lambda0,lambda,obj.M,obj.N,50);
          end
        end

      case 'MT-BCS'
        [mu,sig] = MT_BCS(y,Q,lambda0,obj.M,obj.N,P,numIt,Eb);

      otherwise
        error('Please provide a valid algorithm');
    end

    % Find the position as the centroid of values larger than gamma
    res = cell(1,P);
    for i=1:P
      [res{i}.x,res{i}.X] = obj.findCentroid(mu{i},gamma);
      if ~isequal(algo,'BP')
        res{i}.sigma = sig{i};
      end
```

```
123         res{i}.mu = mu;
124       end
125     end
126
127     % Adaptive localization
128     function [x,X,theta,Sigma] = localizeAdap(obj,y,gamma,algo,ortho,iter,Eb)
129       % Settings
130       tolerance = 1e-9;%1e-9;
131       epsilon = 1e-6;
132       lambda0 = 1/0.01^2;
133       numIt = iter;
134
135       for i=4:length(y)
136         %[~,idx] = datasample(y,i,'Replace',false); % random sampling
137         idx = chooseAntennas(1,length(y),i);    % even spatial spreading
138         Theta = obj.Phi(idx,:)*obj.Psi;
139
140         % The algorithm to use
141         switch algo
142           case 'ST-BCS'
143             Q = orth(Theta')';
144             y_mark = Q*pinv(Theta)*y(idx);
145             theta_0 = Q'*y_mark;
146             [theta,Sigma,~,~,broken] = ST_BCS(y_mark,Q,lambda0,0,obj.M,obj.N,numIt,Eb);
147           case 'MT-BCS'
148             Theta = cellfun(@(a,b) a*b,Phi,Psi,'UniformOutput',0);
149             Q = cellfun(@(a) orth(a')',Theta,'UniformOutput',0);
150             R = cellfun(@(a,b) a*pinv(b),Q,Theta,'UniformOutput',0);
151             y = cellfun(@(a,b) a*b,R,y,'UniformOutput',0)';
152             [theta,Sigma,broken] = MT_BCS(y,Q,lambda0,obj.M,obj.N,P,numIt,Eb);
153           otherwise
154             error('Wrong algorithm name.');
155         end
156
157         if broken
158           break;
159         end
160       end
161
162       % Find the position as the centroid of values larger than gamma
163       [x,X] = obj.findCentroid(theta,gamma);
164     end
165
166     % Find and remove outliers
167     function [y,Phi] = findOutliers(obj,y_db,th)
168         y = 10.^(y_db./10);
169
170         [a,~,c] = histcounts(y);
171         if (a(end-1)==0 && a(end)~=0)
172           i = length(a);
173           iGlob = c==i;
174         else
175           iGlob = zeros(1,length(y));
176         end
177
178         fprintf('Found %i global outliers\n',sum(iGlob==1));
179
180         Phi = obj.findPhi(iGlob');
181
182         y = y(~iGlob);
183         y = 10*log10(y);
184     end
185
186     % Find the phi matrix when outliers are removed
187     function Phi = findPhi(obj,outlierIdx)
188       numOutliers = sum(outlierIdx);
189       M = length(outlierIdx)-numOutliers;
190       if numOutliers>0
191         Phi = zeros(M,obj.N);
192         iP = sub2ind(size(Phi),1:M,obj.T(~outlierIdx));
193       else
194         Phi = zeros(obj.M,obj.N);
195         iP = sub2ind(size(Phi),1:obj.M,obj.T);
196       end
197       Phi(iP) = 1;
198     end
199
200     % Find the position as the centroid of localization vector
201     function [x,X] = findCentroid(obj,theta,lambda)
202       idx = theta>0.99;
203       idx2 = theta>0.1;
204
205       % Different behaviour depending on number of values larger than gamma
```

```
206       if (isnan(theta))
207         disp('NaNs in theta. Something is wrong');
208         x = NaN;
209         X = NaN;
210       elseif (~isempty(idx) && (sum(idx)==1) && (sum(idx2)==0))
211         disp('Unity value in Theta array found.');
212         x = idx;
213         X = obj.grid.findCenterFromTileIndex(idx);
214       else
215         S = find(theta > lambda);
216         if ~isempty(S)
217           if (length(S) > 1)
218             s = zeros(length(S),2);
219             for i=1:length(S)
220               s(i,:) = obj.grid.findCenterFromTileIndex(S(i,1));
221             end
222             X = sum(s)/length(s);
223             x = obj.grid.findTileIndexFromPos(X);
224             fprintf('Centroid calculated from %i points\n',length(s));
225           else
226             fprintf('Only one value larger than lambda. Centroid not calculated. Theta = %.2f\n',theta(S)
                    );
227             X = obj.grid.findCenterFromTileIndex(S);
228             x = S;
229           end
230           if (sum(idx)>=1) && (sum(idx2)>0)
231             disp('Values in theta are larger than unity. Centrid calculated, but might be bogus.');
232           end
233         else
234           if ~isequal(max(theta),0)
235             fprintf('No values are larger than lambda... Using the maximum value of %E.\n',max(theta));
236             x = find(theta == max(theta));
237             X = obj.grid.findCenterFromTileIndex(x);
238           else
239             fprintf('Theta contains only zero values...\n');
240             x = NaN;
241             X = NaN;
242           end
243         end
244       end
245     end
246
247     % Generate path loss from model
248     function [y,d] = generatePL(obj,X,PL0,d0,freq,fLow,fHigh,dim,centered,shadowing,type,db)
249       pl = PathLoss;
250
251       % Placement of receivers, real pos or exactly on grid center
252       if centered
253         T = obj.T_pos;
254       else
255         T = obj.T_real;
256       end
257       obj = obj.addTargets(T);
258
259       % Extract PL at specified frequencies
260       f = fLow:0.1:fHigh;
261       if (size(PL0,1) > 1 || size(PL0,2) > 1)
262         PLused = zeros(size(f));
263         for i=1:length(f)
264           PLused(i) = PL0(findDec(freq,f(i),1.5E-3));
265         end
266         PL0f = PL0;
267         PL0 = PLused;
268       end
269
270       if isequal(dim,3)
271         d = matNorm(T - repmat(X,size(T,1),1));
272       else
273         d = matNorm(T(:,1:2) - repmat(X(1:2),size(T,1),1));
274       end
275
276       % Choose the path loss model
277       y = zeros(obj.M,size(freq,2));
278       switch type
279         case 'IB_800' % 868 MHz IB model
280           PL0 = 11.9;
281           n = 1.9;
282           d0 = 20;
283           for i=1:length(d)
284             y(i,:) = pl.logDistancePL(n,PL0,d0,d(i));
285           end
286
```

```
287       case 'IB_1000_5000' % 1–6 GHz IB model
288         for i=1:length(d)
289           if (size(f,2) > 1)
290             pathloss = zeros(1,size(f,2));
291             for j=1:size(f,2)
292               pathloss(j) = pl.inBodyPL(PL0(j),d0,f(j),1,d(i));
293             end
294             y(i,:) = meanLog(pathloss,2);
295           else
296             y(i,:) = pl.inBodyPL(PL0,d0,freq,1,d(i));
297           end
298         end
299         if ~db
300           y = 10.^(y./10);
301         end
302       end
303     end
304
305     % Add shadowing to the path loss if wanted
306     if shadowing
307       for i=1:length(d)
308         for j=1:10:length(f)
309           y(i,j) = y(i,j) + pl.findShadowing(d(i));
310         end
311       end
312
313     end
314   end
315
316   % Find the sparsity matrix
317   function [obj,Psi] = sparsityMatrix(obj,type,freq,fLow,fHigh,d0,PL0,shadowing,db)
318     PL = PathLoss();
319
320     f = fLow:0.01:fHigh;
321
322     % Extract PL at specified frequencies
323     if (size(PL0,1) > 1 || size(PL0,2) > 1)
324       PLused = zeros(size(f));
325       for i=1:length(f)
326         PLused(i) = PL0(findDec(freq,f(i),1.7E-3));
327       end
328     else
329       PLused = PL0;
330     end
331
332     %Find distance
333     d = bsxfun(@hypot,bsxfun(@minus,obj.grid.gridPosX',obj.grid.gridPosX),bsxfun(@minus,obj.grid.
          gridPosY',obj.grid.gridPosY));
334
335     % Find sparsity matrix corresponding with the path loss model chosen
336     switch type
337       case 'IB_800'
338         PL0 = 11.9;
339         n = 1.9;
340         d0 = 20;
341         obj.Psi = PL.logDistancePL(n,PL0,d0,d);
342         obj.Psi(1:length(Psi)+1:numel(Psi)) = 0;
343
344       case 'IB_1000_5000'
345         if (size(f,2) > 1 || size(f,1) > 1)
346           pl = zeros(obj.N,obj.N,size(f,2));
347           for i=1:size(f,2)
348             pl(:,:,i) = 10.^(PL.inBodyPL(PLused(i),d0,f(i),1,d)./10);
349           end
350
351           Psi = 10*log10(mean(pl,3));
352         else
353           Psi = PL.inBodyPL(PL0,d0,f,1,d);
354         end
355         if ~db
356           Psi = (10.^(Psi./10));
357         end
358         obj.Psi = Psi;
359       end
360     end
361
362     if shadowing
363       Xs = PL.findShadowing(d);
364       obj.Psi = obj.Psi + Xs;
365     end
366    end
367
368   % Plot grid
```

```
   function plot(obj,text)
      if (obj.N > 2000)
        disp('Grid too large to be plotted');
      else
       obj.grid.plotGrid(obj.T,text,0);
      end
    end

    % Add targets to Phi
    function obj = addTargets(obj,T)
     obj.M = size(T,1);

     for i=1:size(T,1)
       obj.T(i) = obj.grid.findTileIndexFromPos(T(i,:));
       obj.T_pos(i,:) = obj.grid.findCenterFromTileIndex(obj.T(i));
     end
     obj.T_real = T;
     Phi = zeros(obj.M,obj.N);
     iP = sub2ind(size(Phi),1:obj.M,obj.T);
     Phi(iP) = 1;

     obj.Phi = Phi;
    end

    % Make the grid
    function obj = createGrid(obj,pos,T,O)
      obj.O = O;
      obj.T = size(T,1); % added 06.06
     switch O
       case 2
         obj.grid = Grid2(obj.N,pos);
       case 3
         obj.grid = Grid3(obj.N,pos);
     end

     obj = obj.addTargets(T);
    end

end
```

Listing B.2: The path loss class `PathLoss.m`.

```
classdef PathLoss

properties
  c = 3e8;
end

methods
  % 1-6 GHz path loss model
  function P = inBodyPL(obj,PL0,d0,f,comp,d)
     if (comp)
       % Compensation
       a = -0.800;
       b = 7.839;
       c = -3.647;
     else
     % No compensation
       a = -0.813;
       b = 7.817;
       c = -3.235;
     end

     if (f(1) > 0.1e9)
      disp('Converting frequency to GHz');
      f = f./1e9; % Frequency in Hz to GHz
     end

     N = a.*f.^2 + b.*f + c;

     if (size(PL0,1) == 1 && size(PL0,2) == 1)
      P = repmat(PL0,size(d,1),size(d,1)) + 10.*N.*log10(d./d0);
     else
      P =  PL0 + 10.*N.*log10(d./d0);
     end

     % Remove Infs
     P(P==-Inf) = 0;
  end

  % Log-distance path loss model
  function P = logDistancePL(obj,n,PL0,d0,d)
    P = repmat(PL0,size(d,1),size(d,2)) + 10.*n.*log10(d./d0);
```

```
    P(P==-Inf) = 0;
  end

  % Find shadowing distribution UWB
  function Xs = findShadowing(obj,d)
    dMod = 5:10:150;
    mu = [0.30 1.47 3.39 4.47 4.78 5.74 6.92 7.45 7.11 7.14 7.52 7.11 7.67 5.18 6.29];
    sigma = [1.87 4.25 5.92 7.12 7.49 8.32 9.07 9.37 9.08 8.65 8.36 8.15 8.69 7.10 7.18];

    % If a d matrix is provided find shadowing for each element
    if (size(d,1) > 1 || size(d,2) > 1)
      Xs = zeros(size(d,1),size(d,2));
      a = ones(size(d,1),size(d,2));
      for i=2:(length(dMod)-1)
        idx = double(d > dMod(i-1) & d < dMod(i));
        distr = random('normal',mu(i-1),sigma(i-1),size(d,1),size(d,2));
        Xs = Xs + idx.*distr;
      end
    % Shadowing for single d
    else
      idx = find(abs(dMod-d)<5);

      if isempty(idx)
        if d > dMod(end)
          idx = dMod==dMod(end);
        else
          idx = dMod==dMod(1);
        end
      end
      Xs = random('normal',mu(idx),sigma(idx),size(d,1),size(d,2));
    end
  end
end
```

B.2 Tracking

Listing B.3: The particle filter class `ParticleFilter.m`.

```
classdef ParticleFilter < Filter

properties
  LinearModel;
end

methods
  function [data] = FilterDataNoInit(obj,rawData,model,N,sigmaN0,sigmaN)
    if (model.dim == 4)
      X0 = [rawData(1,1) 0 rawData(1,2) 0]';
    elseif (model.dim == 6)
      X0 = [rawData(1,1) 0 rawData(1,2) 0 rawData(1,3) 0]';
    elseif (model.dim == 9)
      X0 = [rawData(1,1) 0 0 rawData(1,2) 0 0 rawData(1,3) 0 0]';
    else
      disp('Model dimensions not supported');
    end

    [data] = obj.filtering(rawData,X0,model,N,sigmaN0,sigmaN);
  end

methods (Access=private)
  function [X_est] = filtering(obj,X,X0,m,N,sigmaN0,sigmaN)
    X_est = zeros(length(X),4);

    % Initial particles
    Xp = repmat(X0,1,N) + sigmaN0.*randn(4,N);

    for i=1:length(X)
      % Update state and add noise for each particle
      Xp = m.A*Xp + sigmaN.*randn(4,N);

      % Find measurement residual and evaluate particle likelihood
      if i>1
        v = norm(m.A*X_est(i-1,:)'-X_est(i-1,:)')/m.T;
        Pi = repmat([X(i,1:2) v]',1,N) - [Xp(1:2:3,:); bsxfun(@hypot,Xp(2,:),Xp(4,:))];
        p = mvnpdf(Pi',[0 0 0],diag([m.R(1,1) m.R(2,2) m.R(3,3)]));
      else
        Pi = repmat(X(i,1:2)',1,N) - m.C*Xp;
        p = mvnpdf(Pi',[0 0],[m.R(1,1) m.R(2,2)]);

      end
```

```
43
44        % Update the weights and normalize
45        w = p./sum(p);
46
47        % Resample
48        Xp = obj.resampling(Xp,w,N);
49
50        % Save estimate
51        X_est(i,:) = sum(Xp.*repmat(w',4,1),2);
52
53        % Break if values are NaN
54        if isnan(X_est(i,:))
55          X_est = NaN(length(X_est),4);
56          disp(['Estimate of state are NaN after ' num2str(i) ' iterations']);
57          break;
58        end
59      end
60      nans = false;
61    end
62
63    % Resampling function
64    function [x] = resampling(obj,x,w,N)
65      u=rand(N, 1);
66      wc=cumsum(w);
67      wc=wc/wc(N);
68      [~,ind1]=sort([u;wc]);
69      ind2=find(ind1<=N);
70      ind=ind2-(0:N-1)';
71      x=x(:,ind);
72    end
73  end
```

Listing B.4: The VNL Kalman filter class `KalmanVNL.m`.

```
1   classdef KalmanVNL < Filter
2
3   properties
4     tau;
5     r;
6     n;
7   end
8
9   methods
10    function [data,modeChanges]= FilterDataNoInit(obj,rawData,model,tau,r,n)
11      if (model.dim == 4)
12        x = [rawData(1,1) 0 rawData(1,2) 0]';
13      elseif (model.dim == 6)
14        x = [rawData(1,1) 0 rawData(1,2) 0 rawData(1,3) 0]';
15      elseif (model.dim == 9)
16        x = [rawData(1,1) 0 0 rawData(1,2) 0 0 rawData(1,3) 0 0]';
17      else
18        disp('Model dimensions not supported');
19      end
20      P = eye(model.dim);
21      obj.tau = tau;
22      obj.r = r;
23      obj.n = n;
24
25      [data,modeChanges] = filtering(mod,rawData,x,u,z,P)
26    end
27  end
28
29  methods (Access=private)
30    function [data,modeChanges] = filtering(obj,mod,rawData,x,u,z,P)
31      m = 1;
32      err = 0;
33      kNoManuver = 0;
34      kMode2 = 0;
35      Q = mod.Q1;
36      modeChanges = 0;
37      data = zeros(length(rawData),mod.dim);
38      eps = zeros(length(rawData),1);
39
40      % Coefficients moving average filter
41      a = 1; b = ones(1,obj.n)/obj.n;
42
43      for i=0:length(rawData)-1
44        x_ = mod.A*x + mod.B*u;
45        P_ = mod.A*P*mod.A'+Q;
46
47        Pi = (z(i+1,:)'-mod.C*x_);
48        S = mod.C*P_*mod.C'+mod.R;
49        K = P_*mod.C'*inv(S);
```

```
50        eps(i+1) = Pi'*inv(S)*Pi;
51
52        % Moving average of eps
53        if (i > obj.n)
54          tmp = filter(b,a,eps(i-1:i,:));
55          err  = tmp(end);
56        elseif (i > 0)
57          tmp = filter(b,a,eps(1:i,:));
58          err  = tmp(end);
59        end
60
61        % Check for maneouver
62        if (err > obj.tau && kNoManuver > obj.r)
63          fprintf('i=%i Changing mode from 1 to 2\n',i);
64          modeChanges = modeChanges + 1;
65          Q = mod.Q2;
66          P = diag([mod.R(1,1) mod.Q2(2,2) mod.R(2,2) mod.Q2(4,4)]);
67
68          m = 2;
69          kNoManuver = 0;
70        else
71          kNoManuver = kNoManuver + 1;
72        end
73
74        % Check if has been in mode 2 long enough
75        if isequal(m,2)
76          kMode2 = kMode2 + 1;
77          if (kMode2 > obj.r)
78            fprintf('i=%i Changing back to mode 1\n',i);
79            Q = mod.Q1;
80            P = diag([mod.R(1,1) mod.Q1(2,2) mod.R(2,2) mod.Q1(2,2)]);
81
82            m = 1;
83            kMode2 = 0;
84            kNoManuver = 0;
85          end
86        end
87
88        if ~isequal(kNoManuver,0)
89          P = P_-K*mod.C*P_;
90        end
91        x = x_+K*Pi;
92
93        data(i+1,:) = x';
94      end
95    end
96  end
```

Appendix C

CST simulation framework

Some examples from the CST simulation framework are shown in this appendix. Due to the large amount of code, only a small portion of the total framework is included here.

In Lst. C.1, one of the simulation scrips is shown. Lst. C.2 shows a function from the simulation framework that was used to generate the planar loop antenna. In Lst. C.3, the functionality for importing the HUGO model with frequency dependent tissue properties is shown. Last, in Lst. C.4, the power data export file format can be seen.

Listing C.1: The script used to produce parts of the UWB1-3 datasets.

```
Option Explicit

Dim studio, mws, fs, funcFile, funcPath, functions, treePath, resID, resComplex
Dim freqLow, freqHigh, addSpace, boundary, receiverPos, exitationPort, status
Dim stepsWaveNear, stepsWaveFar, stepsBoxNear, stepsBoxFar, ratioLimit
Dim X,Y,Z,i
Dim w(15,3)

Sub LoadFunctionsCST
  ' Set English locale used for decimal separator etc.
  SetLocale(1033)

  ' Load functions
  funcPath = "..\functions.vbs"
  Set fs = CreateObject("Scripting.FileSystemObject")
  Set funcFile = fs.OpenTextFile(funcPath,1,False)
  functions = funcFile.ReadAll
  funcFile.Close
  Set funcFile = Nothing
  Set fs = Nothing
  ExecuteGlobal functions

  ' Load CST
  Set studio = CreateObject("CSTStudio.Application")
  Set mws = studio.NewMWS
End Sub

' Receiver positions
posArr = LoadAntennaPositions("103_points_z30_inc_width.dlm")

' Transmitter positions
w(0,0) = "0"  : w(0,1) = "0"  : w(0,2) = "0"
w(1,0) = "96" : w(1,1) = "-45"  : w(1,2) = "-6.500e+001"
w(2,0) = "95" : w(2,1) = "-40"  : w(2,2) = "-6.500e+001"
w(3,0) = "92" : w(3,1) = "-31"  : w(3,2) = "-6.500e+001"
w(4,0) = "91" : w(4,1) = "-27"  : w(4,2) = "-6.500e+001"

' Settings
freqLow     = 1
freqHigh    = 4
addSpace    = 50
```

```
boundary    = "open"
stepsWaveNear = 10
stepsWaveFar  = 10
stepsBoxNear  = 10
stepsBoxFar   = 1
ratioLimit    = 10
exitationPort = 1

' Load functions and set up simulation
Call LoadFunctionsCST()
Call SetupSimulation(freqLow,freqHigh,addSpace,boundary)
Call SetMesh(stepsWaveNear,stepsWaveFar,stepsBoxNear,stepsBoxFar,ratioLimit)
Call LoadPlanarLoopParameters()
Call LoadTrapzParameters()

'Import Hugo
Call ImportHugo(2,freqLow,freqHigh,0,0,-300,0.212121,0.824916,0,1,0.238248,0.471154)
Call MakeMeshgroup("hugo",2,2,2,-1,0)
Call AddToMeshgroup("hugo","voxeldata","Human Model","")

'Make transmitter
Call MakePlanarLoop("loop",1,0,0,0)
Call MakeMeshgroup("loopMesh",0.3,0.3,0.3,0,1)
Call AddToMeshgroup("loopMesh","solid","loop","coating")

'Make receivers
Call MakeMeshgroup("trapzDenseMesh",0.5,0.5,0.5,0,1)
Call MakeMeshgroup("trapzCoarseMesh",2,2,2,0,1)
For i = 0 To UBound(X)
  Call MakeTrapezoidalMonopole("trapz" & CStr(i+1),CStr(i+2),X(i),Y(i),(Z(i)))
  Call AddToMeshgroup("trapzDenseMesh","solid","trapz" & CStr(i+1),"coax_shield")
  Call AddToMeshgroup("trapzCoarseMesh","solid","trapz" & CStr(i+1),"ground_plane")
Next

mws.SaveAs "E:\Master\Prosjekter\Simulations\" & Replace(Wscript.ScriptName,".vbs",".cst"), False

For i = 1 To UBound(W)
  Call MoveVector("loop","Shape",w(i-1,0),w(i-1,1),w(i-1,2),w(i,0),w(i,1),w(i,2))
  Call MoveVector(CInt(exitationPort),"Port",w(i-1,0),w(i-1,1),w(i-1,2),w(i,0),w(i,1),w(i,2))
  status = mws.Solver.Start
  If CInt(status) <> 1 Then
    mws.ReportWarningToWindow("Simulation failed for point i=" & CStr(i))
    Exit For
  Else
    mws.ReportInformationToWindow("Simulation finished for point i=" & CStr(i))
    mws.RunMacro("export_wideband")
    mws.RunMacro("export_wideband_sparam")
  End If
  mws.DeleteResults
Next
```

Listing C.2: Function used to generate the planar loop antenna.

```
Sub MakePlanarLoop(name, portNum, x, y, z)
  Call DefineMaterial("Taconic")
  Call DefineMaterial("Copper")
  Call DefineMaterial("Teflon")
  mws.Component.New CStr(name)

  'Substrate
  With mws.Cylinder
   .Reset
   .Name "substrate"
   .Component CStr(name)
   .Material "Taconic RF-41 (loss free)"
   .OuterRadius "loop_R"
   .InnerRadius "0.0"
   .Axis "z"
   .Zrange "0", "loop_ts"
   .Xcenter "0"
   .Ycenter "0"
   .Segments "0"
   .Create
  End With

  'Loop
  With mws.Cylinder
   .Reset
   .Name "loop"
   .Component CStr(name)
   .Material "Copper (annealed)"
   .OuterRadius "loop_Rl+loop_w/2"
```

```
    .InnerRadius "loop_Rl-loop_w/2"
    .Axis "z"
    .Zrange "loop_ts", "loop_ts+loop_tc"
    .Xcenter "0"
    .Ycenter "0"
    .Segments "0"
    .Create
  End With

  'Gap in loop
  mws.Pick.PickCirclecenterFromId CStr(name) & ":loop", "4"
  mws.Pick.MovePoint "-1", "loop_g/2", "0.0", "0.0", "False"
  mws.WCS.AlignWCSWithSelected "Point"
  mws.WCS.RotateWCS "v", "90.0"
  mws.Solid.SliceShape "loop", CStr(name)
  mws.WCS.ActivateWCS "global"
  mws.Pick.PickCirclecenterFromId CStr(name) & ":substrate", "2"
  mws.Pick.MovePoint "-1", "-loop_g/2", "0.0", "0.0", "False"
  mws.WCS.AlignWCSWithSelected "Point"
  mws.Solid.SliceShape "loop_1", CStr(name)
  mws.WCS.RotateWCS "u", "90.0"
  mws.Solid.SliceShape "loop_1", CStr(name)
  mws.Solid.Delete CStr(name) & ":loop_1"
  mws.Solid.Add CStr(name) & ":loop", CStr(name) & ":loop_1_1"
  mws.Solid.Add CStr(name) & ":loop", CStr(name) & ":loop_1_2"
  mws.WCS.ActivateWCS "global"

  'Port
  mws.Pick.ClearAllPicks
  mws.Pick.PickEdgeFromId CStr(name) & ":loop", "43", "32"
  mws.Pick.PickEdgeFromId CStr(name) & ":loop", "29", "22"
  With mws.DiscreteFacePort
     .Reset
     .PortNumber CStr(portNum)
     .Type "SParameter"
     .Label ""
     .Impedance "50.0"
     .VoltagePortImpedance "0.0"
     .VoltageAmplitude "1.0"
     .SetP1 "True", "0.5", "-3.9681187850687", "1.61"
     .SetP2 "True", "-0.5", "-3.9681187850687", "1.61"
     .LocalCoordinates "False"
     .InvertDirection "False"
     .CenterEdge "True"
     .Monitor "True"
     .UseProjection "False"
     .ReverseProjection "False"
     .Create
  End With

  'Coating
  With mws.Cylinder
    .Reset
    .Name "coating"
    .Component CStr(name)
    .Material "Teflon (PTFE) (lossy)"
    .OuterRadius "loop_R+loop_ti"
    .InnerRadius "0.0"
    .Axis "z"
    .Zrange "-loop_ti", "loop_ts+loop_tc+loop_ti"
    .Xcenter "0"
    .Ycenter "0"
    .Segments "0"
    .Create
  End With
  With mws.Solid
    .Version 9
    .Insert CStr(name) & ":coating", CStr(name) & ":substrate"
    .Version 1
  End With
  With mws.Solid
    .Version 9
    .Insert CStr(name) & ":coating", CStr(name) & ":loop"
    .Version 1
  End With

  'Move component and port
  Call Move(name,"Shape",x,y,z)
  Call Move(portNum,"Port",x,y,z)
End Sub
```

Listing C.3: The functions used to import the HUGO model with frequency dependent material properties.

```

' Function to import frequency dependent material properties for human tissues
Sub ChangeTissueMaterials(fLow, fHigh)
  Dim fs, textFile, files, materials, pathMat
  Dim mapArr(32,2), colorArr(32,3)
  Dim hugo, ifac, strArr, fLowIdx, fHighIdx, i, j

  pathMat = "..\materials\"
  files = Array("tissues_0500.csv","tissues_1000.csv","tissues_1500.csv","tissues_2000.csv","
      tissues_2500.csv","tissues_3000.csv","tissues_3500.csv","tissues_4000.csv","tissues_4500.csv","
      tissues_5000.csv")
  materials = LoadBodyMaterials(pathMat,files)
  Wscript.Echo "Loaded material files"

  Set fs = CreateObject("Scripting.FileSystemObject")
  Set textFile = fs.OpenTextFile(pathMat & "mapping.csv",1)
  hugo = Split(textFile.ReadLine(),",")
  ifac = Split(textFile.ReadLine(),",")
  Set textFile = fs.OpenTextFile(pathMat & "colors.csv",1)

  i = 0
  Do While Not textFile.AtEndOfStream
    strArr = Split(textFile.ReadLine(),",")
    colorArr(i,0) = strArr(0)
    colorArr(i,1) = strArr(1)
    colorArr(i,2) = strArr(2)
    i = i + 1
  Loop

  Set textFile = Nothing
  Set fs = Nothing

  For i = 0 To UBound(hugo)
    mapArr(i,0) = hugo(i)
    mapArr(i,1) = ifac(i)
  Next

  For i = 0 To UBound(materials,2)
    If CStr(materials(mapArr(0,1),i,0)) = CStr(fLow) Then
      fLowIdx = i
    End If
    If CStr(materials(mapArr(0,1),i,0)) = CStr(fHigh) Then
      fHighIdx = i
    End If
  Next

  If Not IsEmpty(fLowIdx) And Not IsEmpty(fHighIdx) Then
    mws.ReportInformationToWindow("Loading frequency dependent dielectric properties for tissues...")
    For i = 0 To UBound(mapArr)
      If mws.Material.Exists("Voxel Data/" & CStr(mapArr(i,0))) Then
        With mws.Material
          .Reset
          .Name CStr(mapArr(i,0))
          .Folder "Voxel Data"
          .FrqType "all"
          .Type "Normal"
          .MaterialUnit "Frequency", "GHz"
          .MaterialUnit "Geometry", "mm"
          .MaterialUnit "Time", "ns"
          .MaterialUnit "Temperature", "Kelvin"
          .Epsilon "1"
          .Mue "1"
          .Sigma "0"
          .TanD "0.0"
          .TanDFreq "0.0"
          .TanDGiven "False"
          .TanDModel "ConstTanD"
          .EnableUserConstTanDModelOrderEps "False"
          .ConstTanDModelOrderEps "1"
          .SetElParametricConductivity "False"
          .ReferenceCoordSystem "Global"
          .CoordSystemType "Cartesian"
          .SigmaM "0"
          .TanDM "0.0"
          .TanDMFreq "0.0"
          .TanDMGiven "False"
          .TanDMModel "ConstTanD"
          .EnableUserConstTanDModelOrderMue "False"
          .ConstTanDModelOrderMue "1"
          .SetMagParametricConductivity "False"
```

```
          .DispModelEps "None"
          .DispModelMue "None"
          .DispersiveFittingSchemeEps "2nd Order"
          .MaximalOrderNthModelFitEps "10"
          .ErrorLimitNthModelFitEps "0.1"
          .UseOnlyDataInSimFreqRangeNthModelEps "False"
          .DispersiveFittingSchemeMue "Nth Order"
          .MaximalOrderNthModelFitMue "10"
          .ErrorLimitNthModelFitMue "0.1"
          .UseOnlyDataInSimFreqRangeNthModelMue "False"
          .DispersiveFittingFormatEps "Real_Tand"
          For j = fLowIdx To fHighIdx
           .AddDispersionFittingValueEps CStr(materials(mapArr(i,1),j,0)), CStr(materials(mapArr(i,1),j
                ,2)), CStr(materials(mapArr(i,1),j,3)), "1.0"
          Next
          .UseGeneralDispersionEps "True"
          .UseGeneralDispersionMue "False"
          .NLAnisotropy "False"
          .NLAStackingFactor "1"
          .NLADirectionX "1"
          .NLADirectionY "0"
          .NLADirectionZ "0"
          .Rho "0"
          .ThermalType "Normal"
          .ThermalConductivity "0"
          .HeatCapacity "0"
          .MetabolicRate "0"
          .BloodFlow "0"
          .VoxelConvection "0"
          .MechanicsType "Unused"
          .Colour CStr(colorArr(i,0)), CStr(colorArr(i,1)), CStr(colorArr(i,2))
          .Wireframe "False"
          .Reflection "False"
          .Allowoutline "True"
          .Transparentoutline "False"
          .Transparency "0"
          .Create
        End With
      End If
    Next
    'mws.ReportInformationToWindow("Using frequency dependent dielectric properties for tissues.")
  Else
    mws.ReportWarningToWindow("The frequency range specified for HUGO is not valid. Using constant
         dielectric properties.")
  End If
End Sub

' Load tissue parameters from csv file
Function LoadBodyMaterials(pathMaterials, ByRef files)
  Dim i
  Dim j
  Dim strLine
  Dim lineNum
  Dim strArr
  Dim freq
  Dim fs
  Dim textFile

  ' material; freq; property: freq,cond,eps,tanD,lambda,penet
  Dim propArr(56,9,6)
  Set fs = CreateObject("Scripting.FileSystemObject")

  For i = 0 To UBound(files)
    Set textFile = fs.OpenTextFile(pathMaterials & files(i),1)
    lineNum = 1

    j = 0
    Do While Not textFile.AtEndOfStream
      strLine = textFile.ReadLine()
      strArr = Split(strLine,",")

      If lineNum = 1 Then
        freq = strArr(3)
      ElseIf lineNum > 3 Then
        j = lineNum-4

        propArr(j,i,0) = CStr(CDbl(strArr(1)/1e9))
        propArr(j,i,1) = strArr(2)
        propArr(j,i,2) = strArr(3)
        propArr(j,i,3) = strArr(4)
        propArr(j,i,4) = strArr(5)
        propArr(j,i,5) = strArr(6)
```

```
        propArr(j,i,6) = strArr(7)
      End If
      lineNum = lineNum + 1
    Loop
    textFile.Close

  Next
  Set textFile = Nothing
  Set fs = Nothing

  LoadBodyMaterials = propArr
End Function
```

Listing C.4: The file format used to export power data from CST.

```
Format        v1
Project       E:\Master\Prosjekter\Simulations\planar_loop_movement
Mesh type     PBA
Min cell      0.3
Max cell      7.14286
Mesh cells      48064956(48064956)
Critical cells    0
Excitation port   1
Frequency range   1–3 GHz
Samples       1001
Time matrix calc  13 min
Time solver     228 min
Data        Port, Port x, Port y, Port z, Freq, Power

1 0 −6.4559898382118  1.61  1 0.458137904424962
1 0 −6.4559898382118  1.61  1.003 0.458100178443079
1 0 −6.4559898382118  1.61  1.006 0.458062193147349
1 0 −6.4559898382118  1.61  1.009 0.458023973112524
1 0 −6.4559898382118  1.61  1.012 0.457985542554547
1 0 −6.4559898382118  1.61  1.015 0.457946925340932
1 0 −6.4559898382118  1.61  1.018 0.457908144995039
1 0 −6.4559898382118  1.61  1.021 0.457869224694647
1 0 −6.4559898382118  1.61  1.024 0.457830187264558
1 0 −6.4559898382118  1.61  1.027 0.457791055164593
1 0 −6.4559898382118  1.61  1.03  0.457751850472851
1 0 −6.4559898382118  1.61  1.033 0.457712594866087
1 0 −6.4559898382118  1.61  1.036 0.457673309597748
1 0 −6.4559898382118  1.61  1.039 0.457634015475153
1 0 −6.4559898382118  1.61  1.042 0.457594732837002
1 0 −6.4559898382118  1.61  1.045 0.457555481532505
1 0 −6.4559898382118  1.61  1.048 0.45751628090331
1 0 −6.4559898382118  1.61  1.051 0.457477149769211
1 0 −6.4559898382118  1.61  1.054 0.457438106418696
1 0 −6.4559898382118  1.61  1.057 0.457399168604744
1 0 −6.4559898382118  1.61  1.06  0.457360353546612
1 0 −6.4559898382118  1.61  1.063 0.457321677937712
1 0 −6.4559898382118  1.61  1.066 0.45728315795941
1 0 −6.4559898382118  1.61  1.069 0.457244809300789
1 0 −6.4559898382118  1.61  1.072 0.457206647183508
1 0 −6.4559898382118  1.61  1.075 0.45716868639127
1 0 −6.4559898382118  1.61  1.078 0.45713094130308
1 0 −6.4559898382118  1.61  1.081 0.457093425929069
1 0 −6.4559898382118  1.61  1.084 0.45705615394813
1 0 −6.4559898382118  1.61  1.087 0.457019138745975
1 0 −6.4559898382118  1.61  1.09  0.456982393452803
1 0 −6.4559898382118  1.61  1.093 0.456945930979422
1 0 −6.4559898382118  1.61  1.096 0.45690976405105
1 0 −6.4559898382118  1.61  1.099 0.456873905237949
1 0 −6.4559898382118  1.61  1.102 0.456838366982408
1 0 −6.4559898382118  1.61  1.105 0.456803161621721
```

Bibliography

[1] G. Ciuti, A. Menciassi, and P. Dario, "Capsule endoscopy: from current achievements to open challenges." *IEEE reviews in biomedical engineering*, vol. 4, pp. 59–72, Jan. 2011.

[2] A. Moglia and A. Menciassi, "Capsule endoscopy: progress update and challenges ahead," *Nature Reviews . . .*, 2009.

[3] Given Imaging - Media Relations. Given Imaging. [Online]. Available: http://www.givenimaging.com/en-int/Who-We-Are/Media-Relations/Pages/Image-Gallery.aspx

[4] J. L. Toennies, G. Tortora, M. Simi, P. Valdastri, and R. J. Webster, "Swallowable medical devices for diagnosis and surgery: the state of the art," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 224, no. 7, pp. 1397–1414, Jan. 2010.

[5] A. Koulaouzidis, D. K. Iakovidis, A. Karargyris, and E. Rondonotti, "Wireless endoscopy in 2020: Will it still be a capsule?" *World journal of gastroenterology*, vol. 21, no. 17, pp. 5119–5130, 2015.

[6] Z. Nagy, M. Fluckiger, O. Ergeneman, S. Pane, M. Probst, and B. Nelson, "A wireless acoustic emitter for passive localization in liquids," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, May 2009, pp. 2593–2598.

[7] R. Chandra, A. J. Johansson, M. Gustafsson, and F. Tufvesson, "A microwave imaging-based technique to localize an in-body RF source for biomedical applications." *IEEE transactions on bio-medical engineering*, vol. 62, no. 5, pp. 1231–41, May 2015.

[8] G. Bao, K. Pahlavan, and L. Mi, "Hybrid Localization of Microrobotic Endoscopic Capsule Inside Small Intestine by Data Fusion of Vision and RF Sensors," *IEEE Sensors Journal*, vol. 15, no. 5, pp. 2669–2678, May 2015.

[9] D. Fischer, R. Schreiber, D. Levi, and R. Eliakim, "Capsule endoscopy: the localization system." *Gastrointestinal endoscopy clinics of North America*, vol. 14, no. 1, pp. 25–31, Jan. 2004.

[10] M. Kawasaki and R. Kohno, "Position Estimation Method of Medical Implanted Devices Using Estimation of Propagation Velocity inside Human Body," *IEICE Transactions on Communications*, vol. E92-B, no. 2, pp. 403–409, Feb. 2009.

[11] B. Moussakhani, "On Localization and Tracking for Wireless Capsule Endoscopy," Ph.D. dissertation, NTNU, 2013.

[12] T. Ito, D. Anzai, and Jianqing Wang, "Novel joint TOA/RSSI-based WCE location tracking method without prior knowledge of biological human body tissues." in *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, Aug. 2014, pp. 6993–6996.

[13] C. Hu, W. Yang, D. Chen, M. Q. H. Meng, and H. Dai, "An improved magnetic localization and orientation algorithm for wireless capsule endoscope." *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, vol. 2008, pp. 2055–8, Jan. 2008.

[14] K. M. Popek, A. W. Mahoney, and J. J. Abbott, "Localization method for a magnetic capsule endoscope propelled by a rotating magnetic dipole field," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, May 2013, pp. 5348–5353.

[15] E. Candes and M. Wakin, "An Introduction To Compressive Sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[16] M. F. Duarte and Y. C. Eldar, "Structured Compressed Sensing: From Theory to Applications," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4053–4085, Sep. 2011.

[17] C. Feng, S. Valaee, and Z. Tan, "Multiple Target Localization Using Compressive Sensing," in *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*. IEEE, Nov. 2009, pp. 1–6.

[18] V. Cevher, M. Duarte, and R. Baraniuk, "Distributed target localization via spatial sparsity," in *Signal Processing Conference, 2008 16th European*. IEEE, 2008, pp. 1–5.

[19] Jia Meng, Husheng Li, and Zhu Han, "Sparse event detection in wireless sensor networks using compressive sensing," in *2009 43rd Annual Conference on Information Sciences and Systems*. IEEE, Mar. 2009, pp. 181–185.

[20] M. Pourhomayoun, Z. Jin, and M. L. Fowler, "Accurate localization of in-body medical implants based on spatial sparsity." *IEEE transactions on bio-medical engineering*, vol. 61, no. 2, pp. 590–7, Feb. 2014.

[21] R. Chavez-Santiago and I. Balasingham, "Ultrawideband Signals in Medicine [Life Sciences]," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 130–136, Nov. 2014.

[22] W. A. Kunze and J. B. Furness, "The enteric nervous system and regulation of intestinal motility." *Annual review of physiology*, vol. 61, pp. 117–42, Jan. 1999.

[23] K. Pahlavan, G. Bao, Y. Ye, S. Makarov, U. Khan, P. Swar, D. Cave, A. Karellas, P. Krishnamurthy, and K. Sayrafian, "RF Localization for Wireless Video Capsule Endoscopy," *International Journal of Wireless Information Networks*, vol. 19, no. 4, pp. 326–340, Oct. 2012.

[24] M. Frisch, A. Glukhovsky, and D. Levy, "Array system and method for locating an in vivo signal source," 2002.

[25] Y. Ye, P. Swar, K. Pahlavan, and K. Ghaboosi, "Accuracy of RSS-Based RF Localization in Multi-capsule Endoscopy," *International Journal of Wireless Information Networks*, vol. 19, no. 3, pp. 229–238, Aug. 2012.

[26] S.-N. Hwang, R. Kim, and H. Lim, "Bayesian inference-based tracking for wireless capsule endoscopes," in *2014 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, Oct. 2014, pp. 277–282.

[27] D. Manteuffel and M. Grimm, "Localization of a functional capsule for wireless neuro-endoscopy," in *2012 IEEE Topical Conference on Biomedical Wireless Technologies, Networks, and Sensing Systems (BioWireleSS)*. IEEE, Jan. 2012, pp. 61–64.

[28] G. Bao, L. Mi, and K. Pahlavan, "A Video Aided RF Localization Technique for the Wireless Capsule Endoscope (WCE) inside Small Intestine," in *Proceedings of the 8th International Conference on Body Area Networks*. ACM, Sep. 2013, pp. 55–61.

[29] M. Pourhomayoun, Z. Jin, and M. L. Fowler, "Accurate localization of in-body medical implants based on spatial sparsity." *IEEE transactions on bio-medical engineering*, vol. 61, no. 2, pp. 590–7, Feb. 2014.

[30] B. Zhang, X. Cheng, N. Zhang, Y. Cui, Y. Li, and Q. Liang, "Sparse target counting and localization in sensor networks based on compressive sensing," in *2011 Proceedings IEEE INFOCOM*. IEEE, Apr. 2011, pp. 2255–2263.

[31] V. Cevher, P. Boufounos, R. Baraniuk, A. Gilbert, and M. Strauss, "Near-optimal Bayesian localization via incoherence and sparsity," pp. 205–216, 2009.

[32] Y. Zhang, Z. Zhao, and H. Zhang, "Adaptive Bayesian Compressed Sensing based localization in wireless networks," in *7th International Conference on Communications and Networking in China*. IEEE, Aug. 2012, pp. 43–48.

[33] H. Zhang, J. Palicot, R. Li, Z. Zhao, and Y. Zhang, "Adaptive multi-task compressive sensing for localisation in wireless local area networks," *IET Communications*, vol. 8, no. 10, pp. 1736–1744, Jul. 2014.

[34] S. Ji, Y. Xue, and L. Carin, "Bayesian Compressive Sensing," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2346–2356, Jun. 2008.

[35] S. Ji, D. Dunson, and L. Carin, "Multitask Compressive Sensing," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 92–106, Jan. 2009.

[36] D. Wang, Y. Zhou, and Y. Wei, "A Bayesian Compressed Sensing Approach to Robust Object Localization in Wireless Sensor Networks," in *2014 IEEE International Conference on Mobile Services*. IEEE, Jun. 2014, pp. 24–30.

[37] M. A. Abid and S. Cherkaoui, "3D compressive sensing for nodes localization in WNs based on RSS," in *2012 IEEE International Conference on Communications (ICC)*. IEEE, Jun. 2012, pp. 5195–5199.

[38] Guanqun Bao, Liang Mi, Yishuang Geng, Mingda Zhou, and K. Pahlavan, "A video-based speed estimation technique for localizing the wireless capsule endoscope inside gastrointestinal tract." in *Engineering in Medicine and Biology*

Society (EMBC), 2014 36th Annual International Conference of the IEEE, Aug. 2014, pp. 5615–5618.

[39] P. Pathirana, A. Savkin, and S. Jha, "Robust extended Kalman filter applied to location tracking and trajectory prediction for PCS networks," in *Proceedings of the 2004 IEEE International Conference on Control Applications, 2004.*, vol. 1. IEEE, 2004, pp. 63–68.

[40] N. Shrivastava, R. M. U. Madhow, and S. Suri, "Target tracking with binary proximity sensors," in *Proceedings of the 4th international conference on Embedded networked sensor systems - SenSys '06*. New York, New York, USA: ACM Press, Oct. 2006, p. 251.

[41] D. Schulz, W. Burgard, D. Fox, and A. Cremers, "Tracking multiple moving targets with a mobile robot using particle filters and statistical data association," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 2. IEEE, 2001, pp. 1665–1670.

[42] G. Mao, B. Fidan, and B. D. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, Jul. 2007.

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

[44] T. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Prentice Hall PTR, Dec. 2001.

[45] FCC, "FCC 02-48," Federal Communications Commission, Tech. Rep., 2002.

[46] Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012.

[47] E. J. Candes and T. Tao, "Near-Optimal Signal Recovery From Random Projections: Universal Encoding Strategies?" *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[48] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[49] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[50] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, p. 35, Mar. 1960.

[51] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice with MATLAB*. John Wiley & Sons, 2008.

[52] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," Nov. 2006.

[53] V. Jilkov, "Survey of maneuvering target tracking. part v: multiple-model methods," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1255–1321, Oct. 2005.

[54] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings For Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, 1993.

[55] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 53–82, Jul. 2010.

[56] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[57] D. Liu, T.; Bagnell, "Particle filters: The good, the bad, the ugly," 2012. [Online]. Available: http://www.cs.cmu.edu/~16831-f14/notes/F11/16831_lecture04_tianyul.pdf

[58] CST - Computer Simulation Technology. CST. [Online]. Available: https://www.cst.com/Products

[59] *CST Studio Suite documentation*, CST.

[60] M. Ackerman, "The Visible Human Project," *Proceedings of the IEEE*, vol. 86, no. 3, pp. 504–511, Mar. 1998.

[61] S. M. Kay, *Fundamentals of Statistical Signal Processing: Practical algorithm development*. Prentice-Hall PTR, 2013.

[62] X. Wang and M. Q.-H. Meng, "An experimental study of resistant properties of the small intestine for an active capsule endoscope." *Proceedings of the Institution of Mechanical Engineers. Part H, Journal of engineering in medicine*, vol. 224, no. 1, pp. 107–18, Jan. 2010.

[63] S. Bandyopadhyay, E. J. Coyle, and T. Falck, "Stochastic Properties of Mobility Models in Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 11, pp. 1218–1229, Nov. 2007.

[64] M. Fujii, R. Fujii, R. Yotsuki, T. Wuren, T. Takai, and I. Sakagami, "Exploration of Whole Human Body and UWB Radiation Interaction by Efficient and Accurate Two-Debye-Pole Tissue Models," *IEEE Transactions on Antennas and Propagation*, vol. 58, no. 2, pp. 515–524, Feb. 2010.

[65] D. K. Cheng, *Field and wave electromagnetics*, 3rd ed. Addison-Wesley Publishing Company, 1989.

[66] C. Gabriel, "Compilation of the Dielectric Properties of Body Tissues At Rf and Microwave Frequencies," Brooks Air Force, Tech. Rep., 2008.

[67] K. Kang, X. Chu, R. Dilmaghani, and M. Ghavami, "Low-complexity Cole-Cole expression for modelling human biological tissues in (FD)2TD method," *Electronics Letters*, vol. 43, no. 3, pp. 143–144, 2007.

[68] A. Khaleghi and I. Balasingham, "Improving in-body ultra wideband communication using near-field coupling of the implanted antenna," *Microwave and Optical Technology Letters*, vol. 51, no. 3, pp. 585–589, Mar. 2009.

[69] Calculation of dielectric properties of body tissues. Italian national research council. [Online]. Available: http://niremf.ifac.cnr.it/tissprop/htmlclie/htmlclie.php

[70] P. Orlik, A. Haimovich, and L. Cimini, "On the spectral and power requirements for ultra-wideband transmission," in *IEEE International Conference on Communications, 2003. ICC '03.*, vol. 1. IEEE, pp. 738–742.

[71] A. Wang, "A varying pulse width 5th-derivative gaussian pulse generator for UWB transceivers in CMOS," in *2008 IEEE Radio and Wireless Symposium.* IEEE, Jan. 2008, pp. 171–174.

[72] P. S. Hall and Y. Hao, *Antennas and Propagation for Body-Centric Wireless Communications, Second Edition.* Artech House, 2012.

[73] Q. Wang, K. Wolf, and D. Plettemeier, "An UWB capsule endoscope antenna design for biomedical communications," in *2010 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL 2010).* IEEE, Nov. 2010, pp. 1–6.

[74] P.-A. Floor, R. Chavez-Santiago, S. Brovoll, O. Aardal, J. Bergsland, O.-J. Grymyr, P. S. Halvorsen, R. Palomar, D. Plettemeier, S.-E. Hamran, T. Ramstad, and I. Balasingham, "In-Body to On-Body Ultra Wideband Propagation Model Derived from Measurements in Living Animals." *IEEE journal of biomedical and health informatics*, vol. PP, no. 99, p. 1, Apr. 2015.

[75] *Radio Regulations - Articles*, 2012th ed., ITU.

[76] D. Bandorski, E. Lotterer, D. Hartmann, R. Jakobs, M. Brück, R. Hoeltgen, M. Wieczorek, A. Brock, T. de Rossi, and M. Keuchel, "Capsule endoscopy in patients with cardiac pacemakers and implantable cardioverter-defibrillators - a retrospective multicenter investigation." *Journal of gastrointestinal and liver diseases : JGLD*, vol. 20, no. 1, pp. 33–7, Mar. 2011.

[77] K. S. Kwak, S. Ullah, and N. Ullah, "An overview of IEEE 802.15.6 standard," in *2010 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL 2010).* IEEE, Nov. 2010, pp. 1–6.

[78] C. A. Balanis, *Antenna Theory: Analysis and Design.* John Wiley & Sons, 2012.

[79] S. M. Islam, K. P. Esselle, D. Bull, and P. M. Pilowsky, "A miniaturized implantable PIFA antenna for indoor wireless telemetry," in *2012 International Conference on Electromagnetics in Advanced Applications.* IEEE, Sep. 2012, pp. 526–530.

[80] T. S. P. See, Z. N. Chen, and X. Qing, "Implanted and external antennas for 915-MHz capsule endoscopy," in *2011 International Workshop on Antenna Technology (iWAT).* IEEE, Mar. 2011, pp. 29–32.

[81] R. C. Johnson, *Antenna Engineering Handbook.* McGraw-Hill, 1993.

[82] D. M. Pozar, *Microwave Engineering, 4th Edition.* Wiley Global Education, 2011, vol. 4.

[83] A. Skrivervik, "Implantable antennas: The challenge of efficiency," pp. 3627–3631, 2013.

[84] M. Tareq, D. Alam, M. Islam, and R. Ahmed, "Simple Half-Wave Dipole Antenna Analysis for Wireless Applications by CST Microwave Studio," *International Journal of Computer Applications*, vol. 94, no. 7, 2014.

[85] Y. Morimoto, D. Anzai, and J. Wang, "Design of ultra wide-band low-band implant antennas for capsule endoscope application," in *2013 7th International Symposium on Medical Information and Communication Technology (ISMICT)*. IEEE, Mar. 2013, pp. 61–65.

[86] M. Abdul Matin, *Ultra Wideband - Current Status and Future Trends*, 2012.

[87] T. Dissanayake and M. Yuce, "Design and Evaluation of a Compact Antenna for Implant-to-Air UWB Communication," *IEEE Antennas and Wireless Propagation Letters*, vol. 8, pp. 153–156, 2009.

[88] E. Candès and J. Romberg, "l1-magic: Recovery of Sparse Signals via Convex Programming," 2005. [Online]. Available: http://statweb.stanford.edu/~candes/l1magic/

[89] K. Yazdandoost and K. Sayrafian-Pour, "Channel model for body area network (BAN)," *IEEE P802.15-08-0780-09-0006*, 2009.

[90] A. Alomainy and Y. Hao, "Modeling and Characterization of Biotelemetric Radio Channel From Ingested Implants Considering Organ Contents," *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 4, pp. 999–1005, Apr. 2009.

[91] A. Khaleghi, R. Chavez-Santiago, X. Liang, I. Balasingham, V. C. M. Leung, and T. A. Ramstad, "On ultra wideband channel modeling for in-body communications," in *IEEE 5th International Symposium on Wireless Pervasive Computing 2010*. IEEE, 2010, pp. 140–145.

[92] S. Stoa, R. Chavez-Santiago, and I. Balasingham, "An ultra wideband communication channel model for the human abdominal region," in *2010 IEEE Globecom Workshops*. IEEE, Dec. 2010, pp. 246–250.

[93] F. A. A. Garcia. MATLAB Central - Tests to Identify Outliers in Data Series. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/28501-tests-to-identify-outliers-in-data-series

[94] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. Springer Science & Business Media, 2013.

[95] Z. Xiahou and X. Zhang, "Adaptive Localization in Wireless Sensor Network through Bayesian Compressive Sensing," *International Journal of Distributed Sensor Networks*, no. Article ID 438638, 2015.

[96] A. Mak, C. Rowell, and R. Murch, "Isolation Enhancement Between Two Closely Packed Antennas," *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 11, pp. 3411–3419, Nov. 2008.

[97] A. Khaleghi, R. Chávez-Santiago, and I. Balasingham, "An improved ultra wideband channel model including the frequency-dependent attenuation for

in-body communications." *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, vol. 2012, pp. 1631–4, Jan. 2012.

[98] E. S. Nadimi, V. Blanes-Vidal, V. Tarokh, and P. M. Johansen, "Bayesian-based localization of wireless capsule endoscope using received signal strength." in *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, Aug. 2014, pp. 5988–5991.

[99] A. Waadt, S. Wang, C. Kocks, A. Burnic, D. Xu, G. Bruck, and P. Jung, "Positioning in multiband OFDM UWB utilizing received signal strength," in *2010 7th Workshop on Positioning, Navigation and Communication*. IEEE, Mar. 2010, pp. 308–312.

[100] N. Bargshady, N. A. Alsindi, K. Pahlavan, Y. Ye, and F. O. Akgul, "Bounds on performance of hybrid WiFi-UWB cooperative RF localization for robotic applications," in *2010 IEEE 21st International Symposium on Personal, Indoor and Mobile Radio Communications Workshops*. IEEE, Sep. 2010, pp. 277–282.

[101] W. Gerok, J. Peissig, and T. Kaiser, "TDOA assisted RSSD localization in UWB," in *2012 9th Workshop on Positioning, Navigation and Communication*. IEEE, Mar. 2012, pp. 196–200.

[102] N. Alsindi and C. Duan, "NLOS channel identification and mitigation in Ultra Wideband ToA-based Wireless Sensor Networks," in *2009 6th Workshop on Positioning, Navigation and Communication*. IEEE, Mar. 2009, pp. 59–66.

[103] R. Chavez-Santiago, J. Wang, and I. Balasingham, "The ultra wideband capsule endoscope," in *2013 IEEE International Conference on Ultra-Wideband (ICUWB)*. IEEE, Sep. 2013, pp. 72–78.