DISS. ETH NO. 26089

Robust Global Localization Using GPS and Aircraft Signals

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES

(Dr. sc. ETH Zurich)

presented by

MANUEL EICHELBERGER

MSc ETH CS, ETH Zurich, Switzerland
born on 01.06.1990
citizen of Sumiswald BE

accepted on the recommendation of

Prof. Dr. Roger Wattenhofer, examiner
Prof. Dr. Penina Axelrad, co-examiner

2019

First Edition 2019

Series in Distributed Computing, Volume 31
TIK-Schriftenreihe-Nr. 177
Edited by Roger Wattenhofer
Cover image credits: ESA/Science Office/Herschel/NASA/JPL-Caltech/R. Hurt, CC BY-SA 3.0 IGO (https://creativecommons.org/licenses/by-sa/3.0/igo/legalcode)

Free Space Publishing
ISBN 9781095664292

Abstract

This book introduces several techniques to make global localization methods, most importantly GPS, more robust. The main topic is the *collective detection (CD)* maximum likelihood localization technique for *Global Navigation Satellite Systems (GNSS)*, such as GPS.

CD has several benefits compared to classical receiver techniques. It improves the signal-to-noise ratio after processing of the received satellite signals. This leads to more certain and robust localization even when multipath signals are present, such as in urban canyons. With a rough initial estimate of the receiver state of the order of 100 km and one minute, the improved received signal power can also be used to make localization feasible with received signals as short as one millisecond. *Snapshot receivers* which build on this idea have several benefits. One is the short *time to first fix (TTFF)*, compared to classical receivers which need approximately six or even 30 seconds for such a first localization. Also, the few milliseconds of signal captured by snapshot receivers amount to only a few kilobytes of data, which permits offloading the location computation to the cloud. Therefore, GPS data loggers can be built without correlators, which are the hardware components that consume the most power in classical GPS receivers. Like this, also the area of the receiver can be reduced. But most importantly, snapshot receivers can run for months or even years on a small battery, like a coin cell. Also, snapshot receivers allow for arbitrary duty cycles, as opposed to classical GPS receivers, which need to track the satellite signals continuously. Further, current GPS receivers are susceptible to rogue signals sent by attackers to mislead the receiver. Using CD helps detecting and mitigating such spoofing attacks.

As an alternative to GPS for situations with low received signal power, such as in urban environments or even indoors, an experimental localiza-

tion method is presented. This method localizes a receiver handset using *ADS-B* signals sent by aircraft for air traffic control purposes. Aircraft are comparatively close to the ground and send strong signals. Hence, these signals reach the Earth surface with signal power several orders of magnitude stronger than GNSS satellite signals and are thus easier to recognize.

The first chapter gives an overview of current and potential future localization applications, which are more widespread than many people may be aware of. Thereafter, the book introduces the concepts of GNSS and localization in general, and then builds on these to discuss the robustness improvements. As such, the book is suitable for readers with a technical background, but no prior knowledge of localization systems. Readers familiar with GNSS can skip the introductory chapters and directly dive into the advanced material provided in the subsequent chapters.

Zusammenfassung

Dieses Buch präsentiert verschiedene Möglichkeiten, um globale Lokalisierungsmethoden, insbesondere GPS, robuster zu machen. Das Hauptthema ist die *Collective Detection (CD)* Methode für globale Navigationssatellitensysteme (GNSS) wie GPS.

CD hat mehrere Vorteile gegenüber herkömmlichen Empfängertechniken. Es verbessert das Signal-Rausch-Verhältnis nach dem Verarbeiten der empfangenen Satellitensignale. Das führt zu zuverlässigerer und robusterer Lokalisierung, auch wenn Signalreflexionen empfangen werden, wie häufig zwischen Hochhäusern. Mit einer ungefähren ersten Schätzung des Empfängerzustands, in der Grössenordnung von 100 km und einer Minute, kann die verbesserte Signalstärke auch genutzt werden, um Lokalisierung mit nur einer Millisekunde empfangener Signale zu ermöglichen. Sogenannte *Schnappschuss-Empfänger*, die auf diesem Prinzip beruhen, haben mehrere Vorteile. Ein Vorteil ist die kurze Zeit bis zur ersten Lokalisierung, verglichen mit konventionellen Empfängern, welche dazu ungefähr sechs oder sogar 30 Sekunden brauchen. Ausserdem resultieren die wenigen aufgezeichneten Millisekunden Signale in nur wenigen Kilobytes Datenvolumen, was es erlaubt, die Positionsberechnung in die Cloud auszulagern. Deshalb können GPS-Datenlogger ohne Korrelatoren gebaut werden, welche in herkömmlichen GPS-Empfängern die Hardwarekomponenten mit dem höchsten Energieverbrauch sind. So kann auch die Fläche des Empfängers reduziert werden. Aber am wichtigsten ist, dass Schnappschuss-Empfänger monatelang oder sogar jahrelang laufen können mit nur einer kleinen Batterie, wie einer Knopfzelle. Ausserdem können Schnappschuss-Empfänger mit beliebigem Arbeitszyklus betrieben werden, im Gegensatz zu herkömmlichen GPS-Empfängern, welche die Satellitensignale fortwährend verfolgen müssen. Ferner sind gängige GPS-Empfänger anfällig für arglistige Signale von Angreifern, die den

Empfänger irrezuführen versuchen. Mit CD können solche Angriffe besser entdeckt und entschärft werden.

Als Alternative zu GPS für Situationen mit schwachen Signalen, wie zwischen Gebäuden und in Innenräumen, wird eine experimentelle Lokalisierungsmethode vorgestellt. Diese Methode lokalisiert einen mobilen Empfänger mithilfe von *ADS-B*-Signalen, die von Luftfahrzeugen zu Flugsicherungszwecken ausgesendet werden. Luftfahrzeuge verkehren vergleichsweise nah am Boden und senden starke Signale aus. Daher erreichen diese Signale die Erdoberfläche mehrere Grössenordnungen stärker als GNSS-Satellitensignale und können somit einfacher erkannt werden.

Acknowledgments

> *"Good actions give strength to ourselves and inspire good actions in others."*
>
> — Plato

This book is a result of my PhD studies in the *Distributed Computing Group* at *ETH Zurich*. These studies were a creative, instructive, intensive and often fun time. Frequently together with colleagues, master's and bachelor's students, I explored many different topics of research. This diversity of collaborators and research questions, as well as regular teaching activities, provided plenty of opportunities to improve my skills and knowledge. Numerous people contributed to this great experience and I would like to thank some of them here.

First of all, Roger Wattenhofer, my advisor, or *doctoral father* in German: His straightforward, creative and venturous nature enabled the pursuit of those many projects. I am grateful for the opportunity to join his group and for his continuous encouraging and benevolent support. Among many other things, I profited from Roger's valuable marketing ideas to spark interest in our research results. Plus, he granted me an office with a magnificent view, including snowy mountains, Lake Zurich, the city and colorful sunsets.

Also, I am grateful to Penina Axelrad for taking the time to review my thesis and serving on my examination committee. Her thorough proofreading and valuable comments led to numerous improvements of this book.

One of ETH's strengths is the diverse and open community of colleagues. It is no secret that the best jobs are those with amazing co-workers. I appreciate the time that I could spend with so many of you and express my gratitude to[1] former group members

[1] in chronological order of leaving or joining the group during my presence

Jochen Seidel for being a great example how to capitalize on a PhD in computer science to become an outstanding computer game player; David Stolz for sharing his philosophies and teaching visions; Christian “Dr. Bitcoin” Decker for always being happy and welcoming and for his seemingly unbounded patience; Laura Peer for filling the whiteboards in our office and for teaching students how to download data automatically, such as *all* papers from a large publisher; Philipp Brandes for co-supervising my first student with me, for making sure that everyone participates in student project testing procedures and for introducing Reddit marketing for those who “have been to the Internet before”; Klaus-Tycho Förster for making the office his home and with his kindness making everyone else feel at home, too, further for bringing extraordinary cakes and for never forgetting to send a postcard from his increasingly frequent trips; Ladislas Jacobe “Ladi” de Naurois for his enthusiasm, his fountain of crazy ideas and for showing us how to make a fortune without a PhD, but by betting his car on the cryptocurrency bubble; Benny Gächter for his shell wizardry, for fighting through a broken-server Whac-A-Mole and for organizing regular C++ meetings; Pascal Bissig for his brilliant, innovative ideas, for keeping our GPS project focused and for sharing my start-up ambitions; Michael “The King” König for proving that bread and ketchup are sufficient to survive and even do world-class research, for demonstrating how to live cheap in an expensive city, for sharing his self-made games and most importantly for his puns and Easter eggs, including a correct implementation of the hamburger button; Friederike “Rike” Brütsch for bearing stacks of receipts and for sweetening our days; Thomas Ulrich for keeping up the mood at our regular “Töggeli” games; Sebastian “Hair Guy” Brandt for his massive hair, for playing cops and robbers for a living and for showing that skill and emotions make all the difference at table soccer; Georg Bachmeier for digging to the bottom of the Internet and sharing the discovered stories with us, for adhering to a strict physical training schedule, for clearing away the food in our offices and for regularly baking cakes for us; Conrad Burchert for not taking anything seriously, for his achievement of “writing” a paper without a major contribution and for escaping the blockchain rabbit hole; Julian Steger for his positive attitude towards everything and being passionate about a great many things; Stefan Schindler for increasing the “cyberness” of our institute with that yellow duct tape, for consistently labeling our institute’s hardware, for his Rust evangelism, for a tenfold cut in our physical server count while keeping the functionality, for improving our infrastructure documentation and for creative discussions about hovercraft;

and current group members

Pankaj Khanchandani for providing me with some training data to learn understanding Indian accents, for being a tough guy, trying new sports despite regular (minor) injuries and for placing a little joke in every sentence; Yuyi Wang for collecting PhD titles, for figuring out how to bring sweets without owning an oven and for scoring a multi-month sabbatical in China; Darya Melnyk for attempting to teach us proper swimming techniques, for initiating some weekend activities and for entrusting me with her (supposedly) fragile plants despite her skepticism that I could care for them adequately; Gino Brunner for organizing social events, for being "allergic [to] too little alcohol in ... cakes" and for the most artistic portrait photograph on our webpage; Simon Tanner for enhancing my signal processing knowledge, for supervising numerous students together with me and for braving completely wet shoes while hiking; Georgia "Zeta" Avarikioti for bringing some Greek sociability along, for transforming her office into a living room and for the courage of switching her subject area quite a bit when joining us; Susann Arreghini for her cheerfulness and for her office decorations, making administration quite pleasant; Pál András Papp for being dependable and committed and for correcting our Hungarian pronounciation; Oliver "Oli" Richter for his entrepreneur character, persevering and taking some risks to score grand rewards and for solving mazes professionally; Jakub "Kobi" Sliwinski for being here regardless of the closed office door suggesting otherwise and for his enduring surprise about Swiss habits; Roland Schmid for his helpfulness, for his forthrightness and for bringing back the "Töggeli" spirit; Tejaswi Nadahalli for his efficient handling of duties and for recognizing the appeal of research and waiving a high salary for it; Niklaus "Niggi" Kappeler for his quick handling of our support requests and for bringing a guitar to his office, but never playing it; Aryaz Eghbali for introducing us to new exotic flavors with tasty Iranian cookies; Damián Pascual Ortiz for helping to save lives with his research and for filling the empty spot in our office; Annina Ziltener for coping with the exam review sessions during her first weeks at our institute.

A few people have been around and helping me during the whole time that I spent here: Beat Futterknecht who has the ability to solve any employee's problems, including finding a flat, and who prefers spending his time with us instead of taking holidays; Jan Beutel, Tonio Gsell, Reto Da Forno and Thomas Kleier who enabled our hardware building efforts with laboratory space, equipment and occasionally their own help; René Widmer and Melanie Johnson from *ETH transfer* who actively and passionately supported our technology commercialization efforts.

Further, I would like to thank all the students who contributed their time towards our research in over fifty(!) projects. Noteworthy is Ferdinand "Ferdi" von Hagen who executed a record three student projects mentored by me.

Although I spent a lot of time at the office, my family and friends supported me in having some of that “real life”, too. Also, they showed interest in my work and challenged my ideas, nurturing my creativity. I appreciate the time that I could spend with my university friends, playing board games and having fun. And my deepest thanks go to my parents, Ursula and Heinrich, and my aunt and uncle Regula and Peter, for their cordial support during all the years of work and study culminating in this book. Their kindness fostered my productivity by helping me dedicate extra time and energy to work. Also, I believe that my father significantly influenced my choice of pursuing a career in computer science. He provided me with plenty of computer hardware during my childhood, which must have sparked my interest in this field.

Finally, I thank all the other people who facilitated my work in some way, but who are not named here explicitly.

Thanks to all of you! I hope this book is worth your efforts.

Collaborations and Contributions

Some of the chapters in this book are the result of collaborations with fellow researchers. Below is a list of the co-authors per chapter, whom I would like to thank for their contributions and especially for the interesting meetings that we had during these projects.

Chapters 3 and 4 are based on Chapter 14 of the lecture notes for the course *Computer Systems* taught at ETH Zurich since Autumn 2018 [107]. The same material also appears in Chapter 15 of the book *Blockchain Science – Distributed Ledger Technology* [129]. Co-author is Roger Wattenhofer.

Chapter 5 is based on the publication *Fast and Robust GPS Fix Using One Millisecond of Data* [14]. That joint work has also been published by my colleague Pascal Bissig in the book *Mobile Sensing: GPS Localization, WiFi Mapping, Applications, and Risks* [13]. The chapter is included in this book for completeness, since the software and hardware discussed in the subsequent two chapters make use of the branch-and-bound technique presented therein. Co-authors are Pascal Bissig and Roger Wattenhofer.

Chapter 6 is based on the publication draft *A Spoof-Proof GPS Receiver.* Co-authors are Ferdinand von Hagen and Roger Wattenhofer.

Chapter 7 is based on the publication *Multi-Year GPS Tracking Using a*

Coin Cell [40]. Co-authors are Ferdinand von Hagen and Roger Wattenhofer.

Chapter 8 is based on the publication *Indoor Localization with Aircraft Signals* [39]. Co-authors are Kevin Luchsinger, Simon Tanner and Roger Wattenhofer.

Contents

1

Introduction

> *"The idea that a robot will become more aware of its environment, that telling it to 'go to the kitchen' means something - navigation and understanding of the environment is a robot problem. Those are the technological frontiers of the robotics industry."*
>
> — Colin Angle, CEO of *iRobot* Company

Global localization is a driver for so many applications that it is often considered to be a key technology of our time. For instance car navigation systems and smartphones assist our navigation needs in unfamiliar places and are therefore well-known. However, many people are unaware that localization technology pervades the infrastructure that serves our lives every day. That includes critical services such as communication, electricity distribution and emergency services.

Due to their ubiquitous availability and high accuracy, *Global Navigation Satellite Systems (GNSS)* such as the *Global Positioning System (GPS)* are a universal localization technology that is widely and diversely used. Also, GNSS receivers are cheap, which accelerates their dispersion and integration into all sorts of consumer products. However, GNSS receivers currently have two major problems:

The localization is not robust. Obstructions such as buildings or foliage may render localization infeasible; attackers are able to spoof signals that deceive receivers; and signal reflections can lead to unacceptably large localization errors.

The energy consumption is high. GNSS receivers drain mobile devices' batteries quickly. Even smartphones, which have relatively large batteries, run out of power after several hours of continuous GNSS operation. In contrast, low power sensors such as accelerometers or compasses do not have a significant effect on the total power draw of a smartphone and can be active at all times. GNSS tracking devices with more constrained size, such as for cats or dogs, opt for low duty cycles, determining and sending their location only every few hours. Still, their batteries drain in several days. Therefore, GNSS receivers necessitate large batteries or frequent recharging.

This book addresses both shortcomings. The robustness issue can be tackled with a maximum likelihood localization technique called *collective detection (CD)*. CD processes all satellite signals jointly, instead of acquiring each satellite signal individually like the classical least-squares method. This improves the resulting *signal-to-noise ratio (SNR)*, making the localization more robust, especially with weak or obstructed signals. Also, the computed receiver location likelihood distribution opens up new possibilities to detect spoofed signals and easily integrate the GPS observations with sensor data. For instance accelerometer readings can be used for enhancing localization certainty and spoofing rejection.

Although with the increased SNR, CD enables localization in some areas without direct line-of-sight of the GPS satellites, indoor GPS localization is still out of reach unless many seconds or even minutes of signals are combined like in existing high-sensitivity GPS receivers. In Chapter 8, we explore an alternative localization method using signals transmitted by aircraft. Aircraft repeatedly send so-called *ADS-B* messages containing their location for airspace control purposes. ADS-B signals are transmitted with similar power as GPS signals, but are of the order of ten thousand times stronger than GPS signals when arriving on Earth. This is due to the hundredfold closer distance of aircraft to the Earth surface than GPS satellites – usually between 10 and 400 km instead of about 20.000 km – and due to the inverse square law of the signal power decrease with distance. With a set of ground stations at known locations, ADS-B messages can be used instead of GPS signals for localizing a receiver handset. This may improve the localization robustness in situations with weak GPS signals such as urban areas and might even render indoor localization feasible in some situations.

GPS and ADS-B localization complement each other well. GPS localization is unrivaled in uninhabited spaces where signal obstructions are rare and due to the independence from ground infrastructure. Meanwhile, densely populated regions are rife with aircraft signals, due to the usual proximity of airports.

The energy consumption of GPS localization can be reduced by three orders of magnitude through *snapshot* receivers, which capture only a few milliseconds of satellite signals. Such a signal snapshot suffices to compute the corresponding location, assuming a coarse knowledge of the receiver location and time with an accuracy of the order of 100 km and one minute, respectively. The energy savings of a snapshot GPS receiver are twofold: The active time of the receiver is reduced from seconds to milliseconds; and due to the low amount of data, the processing can be offloaded to another device, saving processing power on the receiver itself. The latter also means that processing hardware can be omitted, leading to smaller receiver size and cost. Snapshot receivers also allow for arbitrary and dynamic duty cycles. Whenever an application requires a so-called location *fix*, a snapshot can be immediately taken. Classical receivers need about six or even 30 seconds at startup, depending on the prior availability of satellite orbit data, before being able to compute their location. For some workloads, this is not acceptable. For instance, when taking holiday pictures, many people turn on their camera, take a picture and turn it off again, in order to save battery energy. If the time is too short to determine the location with an integrated GPS receiver, the photographs can either be tagged with an outdated location or none at all. Therefore, with classical GPS receivers, dynamic location requests can often only be satisfied if the receiver is continuously running. Also, due to the energy overhead of the heavy processing for finding the satellite signals at startup, even low duty cycles are often better served by continuously tracking the signals instead of switching the GPS receiver on and off for each location fix.

So far, snapshot GPS receivers have mostly been a concept, without major commercial implementations. In practice, the low signal power accumulated during just a few milliseconds poses a challenge to extract the signals from the snapshots. Again, CD helps, because the signals from individual satellites do not need to be found. Instead, the signal power of all available signals is united for the receiver location determination. Classical receivers only sum the signal power of each satellite over time (coherent and non-coherent integration). CD therefore adds another power accumulation dimension by summing the signal power also over the satellites. This increases the localization robustness with equal snapshot duration. Apart from this theoretical foundation, we also present a snapshot GPS receiver hardware design in Chapter 7. It uses a small number of components and consumes so little power that it can capture GPS snapshots in 15 minute intervals for about two years while being powered by a coin cell. Our prototype implementation has a size of 23 mm × 14 mm and weighs 1.3 grams. It exhibits the potential of snapshot GPS receivers for enabling new applications, for instance tracking small birds for years. Also, it multiplies the operating time of cameras tagging photographs with their location and fitness trackers and the receiver can be hidden in valuable belongings like wallets and bags. Real-world measurements with our receiver using only one millisecond of data show that the localization is robust and has an accuracy of the order of ten meters while a single snapshot consumes only 0.74 mJ. With longer snapshots, the localization accuracy can be improved while trading off the longevity of the receiver, which is currently constrained by the used flash chip's data capacity.

GPS Applications

To illustrate the importance of global localization to the modern world, we start by revealing a number of current and potential future GPS applications. We also discuss some challenges that these applications face.

Because a GPS receiver measures arrival times of electromagnetic signals that travel at the speed of light, the receiver's location and time are linked tightly. Each nanosecond of error corresponds to a distance offset of about 30 cm. Thus, the GPS receiver needs to be synchronized with the satellites' time, which is the reason why the receiver location and time are resolved jointly. Due to the resulting time accuracy of the order of tens of nanoseconds, GPS is widely used for time synchronization, besides the self-evident localization. Therefore, GPS applications can generally be divided in two categories, actual localization applications and timing applications.

Localization

Satellite Navigation Starting with consumer products, satellite navigation systems spring to mind. These come in the form of dedicated devices or just as smartphone applications. Their uses are for pedestrians and vehicles such as cars, trucks, ships, aircraft or bikes. Especially for ships and aircraft, GPS may often be the only absolute source of localization and navigation information, as landmarks for orientation are missing on many parts of the oceans. GPS navigation devices are not just a replacement for reading a map or following directions. Integrating external real-time information, for instance about traffic jams, can be used to adapt paths to current conditions.

Fitness Tracking Fitness trackers have gained some popularity and some include GPS receivers to log one's workout routes and compute some statistics like running speed and distance and estimate burned calories. Many smartwatches also include this functionality, nowadays.

Holiday Logging Many people like to log their travel routes and tag their holiday photographs with the corresponding location. Since commercial GPS receivers take about 30 seconds from startup to being able to compute their location, they need to run continuously to support tagging photographs. Otherwise, the tag might include an old location or none at all. Due to this problem annoying users, many middle class cameras do not feature GPS receivers any more. Cameras which do, like many professional cameras, need frequent battery replacements. Some travelers and photographers use standalone GPS loggers, which commonly run for a few days on a single battery charge. Logged tracks can later be combined with timestamped photographs, in order to tag them with their location.

Asset Tracking Some people use GPS receivers to keep track of their pets, such as dogs and cats. Not only due to the high power draw of current commercial GPS receivers, but also because the location needs to be transmitted wirelessly to the device owner, such trackers need to be recharged every week and still report the pet's location only every few hours. If the power of such devices can be improved substantially, this may result in a more general class of asset trackers, for instance for bags, bikes or other valuable belongings to retrieve them in the case of loss or theft. Both energy-consuming parts are seeing some development: On the GPS side, snapshot receivers are promising huge energy savings, although these are reduced when the data has to be sent out wirelessly, since snapshots are several kilobytes large, compared to only a few bytes for coordinates only. At the same time, narrowband cellular communication standards such as LTE-M and NB-IoT are deployed worldwide. These are optimized for relatively low data rates of a few hundred kilobytes per second but use significantly reduced transmission power compared to standard LTE.

Logistics In the industrial and commercial domain, goods and fleet tracking in logistics is an important application. Taxi and emergency services use their cars' location for real-time management and dispatch. Parcels are tracked to inform recipients of estimated arrival dates and times and medical goods like drugs are traced for supply chain verification and to control the appropriate handling, for example by a certified party. Aircraft and trains use GPS as an additional and more accurate localization method besides radar or sensors between rail tracks (so-called balises), respectively.

Services Related to the examples above, taxi passenger locations are used to pick those passengers up and calculate fares, and emergency requests from cell phones are pinpointed, since people in need of help are not always aware where they are exactly or able to describe their location. In the digital realm, online advertisements, often personalized and localized for users are an important driver for some of the world's largest businesses, for instance Google. However, such ads are not undisputed. Third parties tracking users' location continuously raises privacy concerns. While privacy is an important aspect of ubiquitous user localization, this book focuses on the underlying technology.

Robotics Automation is changing most industry branches. This transformation is so important that the emerging automated industry processes are called "Industry 4.0". Localization is an integral part of many of those processes. In agriculture, tractors and planes use GPS to guide their path through fields. In automated storage spaces, robots may use GPS to find goods and drop them off in the right places. While autonomous cars are mostly being developed using lidars for localization, radar, ultrasound and GPS systems may be required in situations when lidar fails, for instance in fog. Also during normal operation, these secondary systems my improve the localization certainty. Even for more mundane tasks, there is a substantial automation potential. Examples include painting line markings on soccer fields. Such lines need to be drawn precisely, which can be facilitated by GPS [89].

Science Bird tracking is one research application that was already mentioned above. Recently, an eagle was found in Saudi Arabia twenty years after it was equipped with a GPS receiver and solar cells in Russia [117]. Interestingly, the reconstructed track shows that the eagle never crossed over wide waters, but instead stayed above land at all times. Large birds like this eagle can carry relatively heavy loads, allowing them to carry sizeable batteries and solar cells. However, small songbirds can usually carry only a few grams [17]. Due to the high power draw of GPS receivers, multi-year GPS tracking has been infeasible so far. Instead, researchers equip birds with light sensors and a clock, and hope to catch some birds again after a year to retrieve this package with the logged data. From the day length and start, determined from the (sun) light intensity, the latitude and longitude can be reconstructed, respectively. However, the error is of the order of 200 km, which cannot give insights for instance on swarm behavior [17]. With our receiver that is presented in Chapter 7, the accuracy is improved to some ten meters while preserving the multi-year battery life. At the other end of the spectrum, GPS is also used for tracking millimeter geological ground shifts to predict land slides [12]. This application needs high precision, but is less energy constrained, as large solar panels and batteries can be deployed with each sensor.

Time Synchronization

Database Synchronization Instead of using classical synchronization algorithms, distributed databases can also achieve consistency by tagging events with accurate timestamps for correct ordering later when messages about these events are exchanged [33]. This is especially useful for datacenters with large latencies due to long distances, such as those spanning the globe and if high throughput must be tolerated, making it impractical to wait for one or several message round trips before accepting new transactions.

Communication Cellular networks use microsecond-level time synchronization to minimize guard periods in time-division multiple access (TDMA). Like this, short time slots can be allocated to provide low latencies to individual handsets while keeping the utilization of the available radio spectrum high for good throughput. For this purpose, cellular base stations are normally equipped with GPS receivers, since GPS provides the most accurate global timing service. As a result, mobile devices like smartphones, deployed sensors and increasingly even cars (for software updates or real-time traffic data) all depend on GPS.

Electricity Distribution In electricity grids, the sequence, types and locations of unexpected perturbations such as power plant failures or tripped lines can be reconstructed using power measurements along the grid. Since electricity propagates at the speed of light, such measurements must be accurately synchronized. This is done using GPS. Such measurements are important for maintaining the integrity of electric grids, which always need to have balanced power production and consumption.

Stock Markets The transition from trading floors to electronic stock exchanges enabled automatic, algorithmic trading using computers. Especially the high-frequency trading (HFT) branch now generates more than half of the stock market volume [78]. HFT is for instance used for arbitrage, that is for leveraging price differences between multiple exchange markets. Such price differences are short-lived, meaning that traders have to be fast to make a profit. Therefore, orders also have a short expiration time. Orders with an expired deadline are not executed. Thus, it is important that traders and markets are synchronized. Otherwise, if a trader's time lags behind the market, none of its orders might be executed [34]. Since the time synchronization uses GPS, an attacker might push a trader out of the market by spoofing GPS signals. Also, since markets and traders check the integrity of their data, transaction sequences with out-of-order timestamps or other time anomalies due to forged GPS signals might halt trading activities. Even if only individual traders are affected, liquidity is removed from the market, which leads to more price volatility and loss of trust in the market [105]. Even more attacks misusing timing properties exist: Attackers which get market order information with lower latency than the granularity of timestamps used in the market can cheat as follows. When a long-term investor places an order to buy shares, a high-frequency trader can place a large order to buy shares of the same company so quick that the market treats both orders as incoming concurrently. Subsequently, the high-frequency trader's order might be executed first, making the shares more expensive for the investor. Also the investor's buy increases the share price, so the trader can sell his position immediately at a profit. The resulting decrease in the share price leaves the investor with an instant loss [78]. To prevent such fraud, stock markets need to use precise and synchronized timestamps, which can be provided by GPS.

The above examples illustrate that GPS localization and timing are key drivers for distributed systems and smart devices. As localization technology becomes cheaper, more available, smaller and more power efficient, it can be included in a growing number of products like for instance smart clothing or other wearables. To progress into this direction, a few challenges remain to be solved: Currently, the most widely available localization technology are *Global Navigation Satellite Systems (GNSS)*, such as the US *Global Positioning System (GPS)*. However, GNSS receivers require a substantial amount of energy to produce a location *fix*, that is, compute their location. This means that 1) relatively large and 2) heavy batteries need to be used and that 3) receivers run for a few hours at most on a single battery charge. Also, due to the weakness of the satellite signals when they reach the Earth, GNSS receivers work poorly or not at all indoors and in environments such as urban canyons and under foliage.

In this book, we explore possibilities to compute GNSS location fixes from only a few milliseconds of the received satellite signals, instead of using several seconds like classical receivers. Due to the shorter recorded signals, the required energy per fix is reduced significantly, and more importantly, the location computation can be offloaded to the cloud. While the reduced signal power trades off some accuracy, the localization error increases by only a few meters, as we will see in Chapter 5.

Our discussion includes theory, software and also hardware aspects of GPS receivers. Chapter 4 gives an overview of classical GPS localization and introduces assisted GPS techniques such as *Assisted GPS (A-GPS)*, which is widely used in smartphones, and snapshot GPS, which is the main focus of this book. Namely, Chapter 6 shows how robust maximum likelihood methods, that are used for snapshot receivers, can also help protecting receivers from fake satellite signals sent by attackers and Chapter 7 elaborates how a snapshot GPS receiver can be built. The latter is important since no commercial snapshot receivers exist and therefore, the real-world advantages of such a receiver design cannot easily be tested and further research is hindered.

Unfortunately, GNSS work poorly indoors due to signal attenuation in walls and multipath effects. Many different other indoor localization approaches exist. While some use dedicated hardware beacons, other use existing hardware such as Wi-Fi base stations. Also the techniques vary. Most common are fingerprinting approaches and time-of-flight based methods. The drawback of all these methods is that they have to be set up or initialized locally and this process may need to be repeated each time the physical setup changes. Ideally, we would like a worldwide system such as a GNSS for indoor localization. For this purpose, we propose an approach that is based on signals regularly sent by aircraft. These signals are so-called *automatic dependent surveillance – broadcast (ADS-B)* messages, required by flight traffic control authorities to be sent by each aircraft twice per second. As such, those signals are available in most urban areas, where GNSS often do not work well, as mentioned above. Those aircraft messages contain the location of the aircraft and can be used in a similar manner as GPS satellites. In fact, the aircraft determine their location using GPS and then relay this information to the ground. Since aircraft are in the sky, their view of the GPS satellites is unobstructed. Therefore, aircraft always receive GPS signals with good quality. With some efforts explained in Chapter 8, like using ground stations to determine the send time of the ADS-B signals, the aircraft sending ADS-B messages can be leveraged as a sort of "satellite signal amplifiers". This is meant in the sense that the same time-of-flight (ToF) localization method that is the standard for GPS receivers can be used with the aircraft signals.

2

Localization

> *"Ubi materia, ibi geometria. — Where there is matter, there is geometry."*
>
> — Johannes Kepler

First, we study some geodesy basics and global coordinate systems. Then, an overview and classification of localization systems and methods is given.

Definition 2.1 (Localization)**. Localization** *is the process of determining an object's place with respect to some reference, usually a coordinate system.*

Positioning is an alternative expression for *localization.* We stick to the term *localization* in this book, except for proper names like *Global Positioning System.* Note that in other domains, *localization* can also mean the adaptation of products for different regions, for instance the translation of text in a smartphone app [57]. Meanwhile, *positioning* is a term used in marketing for the formation of a product or brand identity. Instead of using a coordinate system, one could also just resolve the relative ordering of objects, find the closest object in a given set or decide which region an object is situated in.

2.1 Coordinate Systems

Definition 2.2 (Coordinate System)**.** *A* **coordinate system** *uses an ordered list of* **coordinates***, to uniquely describe the location of points in space. The meaning of the coordinates is defined with respect to some anchor points. The point with all coordinates being zero is called* **origin***.*

Often, coordinates are just numbers, but they can also include letters or symbols, such as in *47°22′38.1″N 8°33′11.7″E.* Depending on the application, different anchors are used. In astronomy, celestial coordinate systems are used, which are fixed with respect to galaxies, stars and other distant objects. Meanwhile, virtual and augmented reality (VR and AR) systems use anchor points in a room. For global localization, we are mostly concerned with *terrestrial* coordinate systems, used to locate places on and around Earth.

Definition 2.3 (Earth-Centered Coordinate System)**.** *An* **Earth-centered** *or* **geocentric** *coordinate system has its origin at Earth's center of mass.*

The center of mass is also called the *geocenter* or *barycenter* [86].

Definition 2.4 (Earth-Fixed Coordinate System)**.** *An* **Earth-fixed** *coordinate system rotates with Earth's surface, that is, the coordinates of a point on Earth are time-invariant.*

In contrast, *celestial* coordinate systems do not rotate with respect to the stars.

Definition 2.5 (Pole)**.** *The (geographic)* **poles** *are the two points where Earth's axis of rotation meets Earth's surface.*

Note that Earth's *magnetic* poles differ from the geographic poles. The choice between the North and South Poles is an arbitrary convention. Actually, Earth exhibits polar motion, moving its rotational axis several meters relative to its crust [21]. Due to the rotational forces, Earth is not a perfect sphere, but resembles an ellipsoid.

Definition 2.6 (Ellipsoid)**.** *An ellipsoid is the surface obtained by rotating an ellipse about one of its axes.*

For the Earth, the rotational axis is the shorter (semi-minor) axis of the ellipse. Such a shape is called an *oblate spheroid.* The fraction by which the rotational (semi-minor) axis is shorter than the equatorial (semi-major) axis is called *flattening.* Earth's diameter ranges from 12,714 km between the poles to the equatorial diameter of 12,756 km [131].

Definition 2.7 (Equator)**.** *The* **equator** *is the circle formed by the intersection of the ellipsoid with the plane containing all points equidistant from both poles.*

The equatorial plane is perpendicular to the rotational axis. Earth's circumference, or the length of the equator, is 40,075 km.

Definition 2.8 (Meridian)**.** *A* **meridian** *is a (curved) line segment on the ellipsoid connecting the poles.*

A meridian connects points of equal longitude. The *prime meridian*, defining the 0° longitude, can be chosen arbitrarily. The historically popular Greenwich prime meridian passes through the Royal Observatory, Greenwich, United Kingdom.

Definition 2.9 (Longitude)**.** *The* **longitude** *is a coordinate indicating the angle corresponding to the horizontal (east-west) location of a point on Earth. The angle is formed between the plane through the meridian containing the point and the plane through the prime meridian. The longitude is zero at the prime meridian and* $\pm 180°$ *opposite of it. Positive longitudes are east of the prime meridian, while negative longitudes are west.*

Definition 2.10 (Latitude)**.** *The* **latitude** *is a coordinate indicating the angle corresponding to the vertical (north-south) location of a point on Earth. The angle is formed between the equatorial plane and a line passing through the point. In the case of* **geodetic latitude***, this line is perpendicular to the ellipsoid. For* **geocentric latitude***, the line passes through the geocenter. The latitude is zero at the equator and ranges from -90° at the South Pole to +90° at the North Pole.*

Often, the type of latitude is not indicated. By convention, this means that latitudes are *geodetic*. Still, without specifying the parameters of the used ellipsoid or other surface shape model, the geodetic latitude is underdetermined. Different coordinate systems may use different reference ellipsoids to get a better fit of the ellipsoid to the surface in some geographic region, like a certain country.

Definition 2.11 (Geoid)**.** *The* **geoid** *is a model defining the mean sea level over the whole Earth according to Earth's gravity and rotation [137, Sec. 2.2].*

The geoid does not consider winds or tides. Due to the Earth's uneven mass distribution, the geoid has an irregular shape. Therefore, the height above sea level generally differs from the height above the ellipsoid. The difference is called *geoid separation*.

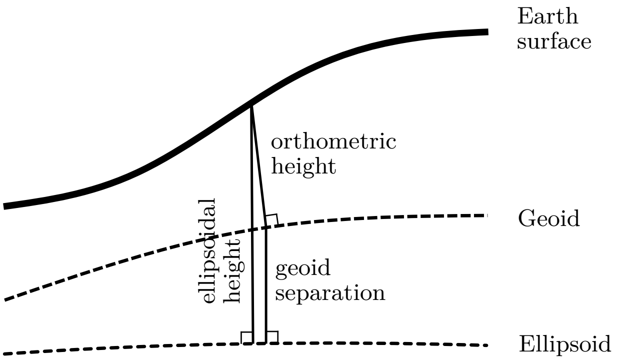


Figure 2.12: Relation between different height definitions and the geoid separation.

Definition 2.13 (Height)**.** *A point's* **height** *is its distance to the employed Earth model, perpendicular to that model's shape.*

In common use, height refers to the height *above mean sea level (AMSL)*, that is, above the *geoid* (orthometric height) [137, Sec. 2]. GPS height is measured with respect to the ellipsoid (ellipsoidal height) [137, Sec. 2]. These two heights differ approximately by the geoid separation – usually less than 100 m. The concepts are illustrated in Figure 2.12. *Elevation* refers to the height of a point on the ground, while *altitude* is used for points above ground.

Definition 2.14 (Terrestrial Coordinate System)**.** *A* **terrestrial** *coordinate system is used to locate places on or near Earth's surface. It uses anchor points on Earth's surface.*

The Royal Observatory in Greenwich, United Kingdom is a well-known anchor point, defining the Greenwich prime meridian.The two most important types of terrestrial coordinate systems are Cartesian and geodetic coordinate systems. Both are Earth-centered and Earth-fixed.

Definition 2.15 (Cartesian Coordinate System)**.** *A* **Cartesian** *coordinate system represents points in space as their projection onto perpendicular axes.*

Definition 2.16 (International Terrestrial Reference System)**.** *The* **International Terrestrial Reference System (ITRS)** *is a standard defining an Earth-centered, Earth-fixed Cartesian coordinate system. Its realizations are the* **International Terrestrial Reference Frames (ITRF)**.

The ITRF positive z-direction crosses the north pole. The x-axis points towards the intersection of the equator with the prime meridian passing through Greenwich. The y-axis completes a right-handed coordinate system, surfacing in the Indian Ocean between Sri Lanka and Sumatra [67].

The newest ITRF, ITRF2014, was published in 2016 [2]. The ITRF is updated every few years to account for changes in Earth's crust motion and improved surveying technology [90]. The term *ECEF* (for *E*arth-*c*entered, *E*arth-*f*ixed) usually stands for an ITRS coordinate system.

Definition 2.17 (Geodetic Coordinate System)**.** *A* **geodetic** *coordinate system represents each point on or near Earth's surface by two angular coordinates, latitude and longitude, and by a height coordinate.*

As an example, the main building of ETH Zurich has the coordinates *(47°22′35.2″N, 8°32′53.1″E, 412 m)* or *(47.376442°, 8.548075°, 412 m)* in the WGS 84 geodetic coordinate system. N and S indicate north and south of the equator, respectively, while E and W indicate east and west of Greenwich (longitude 0), respectively. In the absence of such a letter, negative coordinates correspond to south and west, respectively. Like in the example above, coordinates can be indicated in decimal degrees or in degrees (°), arcminutes (′) and arcseconds (″). These units are related by $1° = 60' = 3600''$. Conceptually, it does not matter if Cartesian or geodetic coordinate systems are used, as coordinates can be converted between different systems. In practice, using Cartesian coordinates results in simpler computations, while geodetic coordinates are more easily interpretable.

Definition 2.18 (World Geodetic System)**.** *The* **World Geodetic System (WGS)** *is a standard defining a geodetic coordinate system [32]. The coordinate system is geocentric and Earth-fixed. The current version, WGS 84, uses an ellipsoid with equatorial radius* $a = 6378137$ *m and flattening* $f = 1/298.257223563$ *[55]. The WGS 84 prime meridian passes 102 m east of the Royal Observatory, Greenwich, United Kingdom [84, 98]. WGS 84 also defines a geoid based on the* **Earth Gravitational Model 2008 (EGM2008)***. The geoid separation ranges from -106 to 85 meters [99].*

WGS is maintained by the U.S. Department of Defense and used by GPS. The WGS 84 prime meridian does not pass exactly through the Royal Observatory because the observations for the historic Greenwich Meridian were based on plumb line observations. Due to local gravitational effects, this plumb line does not pass through the geocenter, which is called *deflection of the vertical.* In WSG 84, that effect is accounted for and the plane containing the prime meridian contains the geocenter. The current version, *WGS 84*, was established in 1984, with updates having been made regularly since then.

2.2 Setup

Definition 2.19 (Handset)**.** *In localization systems, a* **handset** *is a receiver and/or transmitter with unknown location.*

In robotics, handsets are sometimes called *rovers*. A handset's location can either be computed by the handset itself or by the localization infrastructure.

Definition 2.20 (Base Station)**.** *In localization systems, a* **base station** *is a transmitter and/or receiver with known location.*

Base stations do not need to have a fixed location, like for instance GPS satellites which orbit around the Earth. Base stations located on the ground are sometimes called *ground stations*. Base stations range from simple beacons, which repeat a fixed signal over and over again, to more elaborate transceivers which execute a protocol with a handset or a network of base stations which compute the location of the handset together.

Definition 2.21 (Attenuation)**.** *Signal* **attenuation** *is the reduction of a signal's power during transmission. In free space, the attenuation is proportional to the square of the distance from the emitter.*

The fact that the signal power decreases inversely proportional with distance in free space is also referred to as the *inverse-square law*. The reason for the inverse-square law is that the signal's waves cover a spherical sector. Such a sector's surface area grows with the square of the signal's traveled distance. The signal power is distributed over that surface area. Thus, there is a tradeoff for localization system base stations: Using a low number of long-range transceivers keeps the infrastructure cost low, while using many short-range transceivers provides high signal power to handsets. While the inverse-square law holds in free space, objects like walls further attenuate signals. Therefore, most indoor localization systems employ base stations in each room.

Definition 2.22 (Localization Signal)**.** *A* **localization signal** *is a signal used for localization.*

Localization signals need not always be specifically designed and generated for localization. With some effort, localization can also be performed using existing signals such as Wi-Fi, LTE or DVB-T signals. Wi-Fi, Bluetooth and many other signals use the *industrial, scientific and medical (ISM)* radio bands, which can be used without a license in many countries as long as the emitted power is kept below some thresholds [132]. Apart from radio waves, ultrasound and light are commonly used for localization. For instance *lidar* (from *light radar* or *light detection and ranging*) is the prime localization technology of most current autonomous car designs.

2.3 Methods

While in most localization systems the handset receives signals from the base stations to compute its location, a handset can also send signals that enable the base stations to collaboratively determine the handset location. Since transmitting wireless signals consumes more power than receiving such signals, the former option is more common.

Definition 2.23 (Fingerprinting)**. Fingerprinting** *is a localization method which matches observed signals with a previously compiled set of signals observed at different locations. The handset location is estimated based on the best-matching signal observations from the precompiled set and those observations' locations.*

To achieve fine-grained localization, extensive data sets have to be collected before a fingerprinting-based localization system can be used. Since signals usually vary over time and with infrastructure changes, like obstacles being moved around, databases should be updated regularly. For coarse localization, smartphones match available Wi-Fi or cellular base stations against previously known base station locations. That base station data can be collected and updated by all users of this method together.

Definition 2.24 (Triangulation)**. Triangulation** *is a localization method based on measuring* **angles of arrival (AoA)***, that is, angles between base stations and a handset.*

Geometrically, the handset is located at the intersection of the straight lines with the measured angles, passing through the anchor points, which are the base stations. In practice, measurement errors have to be accounted for. This is also the case for the localization methods below. Theodolites are common instruments used to optically measure AoA for construction site and land surveying.

Definition 2.25 (Trilateration)**. Trilateration** *is a localization method based on measured ranges between a handset and base stations.*

Often, distances are determined through *time-of-flight (ToF)* measurements of signals sent at a known time and location. Geometrically, the handset location can be found by intersecting the spheres (or circles in 2D) around the base stations with the radii being the measured distances. Through the use of an additional independent measurement, a common distance bias of all measurements can be compensated. This is for instance needed when the handset is not synchronized with the base stations, like in GPS. Some alternative terminology exists: Trilateration is also called *multilateration* and ToF measurements are equivalent to *time of arrival (ToA)* measurements when the signal transmission times at the base stations are known.

3 Time

> *"Nanosecond precision matters for worldwide communications systems. It matters for navigation by Global Positioning System satellite signals: an error of a billionth of a second means an error of just about a foot, the distance light travels in that time."*
>
> — James Gleick, Author and Science Historian

Location and time are tightly related in GNSS. The distance and ToF of the satellite signals have a direct correspondence with the proportionality coefficient being the speed of light. While the previous chapter puts GPS into context from the localization perspective, this chapter gives an overview of the time and clock synchronization side.

3.1 Time & Clocks

The standard unit of time is the second, which is officially defined by the *Bureau International des Poids et Mesures.* A slightly simplified definition is the following.

Definition 3.1 (Second)**.** *A* **second** *is the time that passes during 9,192,631,770 oscillation cycles of a caesium-133 atom [50, Résolution 1].*

Historically, a second was defined as one in 86,400 parts of a day, dividing the day into 24 hours, 60 minutes and 60 seconds [63]. Since the duration of a day depends on the unsteady rotation cycle of the Earth, the novel oscillation-based definition has been adopted. Leap seconds are used to keep time synchronized to Earth's rotation.

Definition 3.2 (Wall-Clock Time)**.** *The* **wall-clock time** t^* *is the true time (a perfectly accurate clock would show).*

Definition 3.3 (Clock)**.** *A* **clock** *is a device which tracks and indicates time.*

A clock's time t is a function of the wall-clock time t^*, that is, $t = f(t^*)$. Ideally, $t = t^*$, but in reality there are often errors.

Definition 3.4 (Clock Error)**.** *The* **clock error** *or clock skew is the difference between two clocks, for instance,* $t - t^*$ *or* $t - t'$*.*

The importance of accurate timekeeping and clock synchronization is reflected in the following statement by physicist Steven Jefferts: "We've learned that every time we build a better clock, somebody comes up with a use for it that you couldn't have foreseen." [63]

In practice the clock error is often modeled as $t = (1+\delta)t^* + \xi(t^*)$. Therefore, the variation of the clock error over time is divided into a predictable and a random part.

Definition 3.5 (Drift)**.** *The* **drift** δ *is the predictable clock error.*

Drift is relatively constant over time, but may change with supply voltage, temperature and age of an oscillator. It is possible to fit higher-order models, capturing for instance quadratic errors. But in practice, this is rarely done. Stable clock sources, which offer a low drift, are generally preferred, but also more expensive, larger and more power hungry. Thus, many consumer products feature inaccurate clocks.

Definition 3.6 (Parts Per Million)**.** *Clock drift is indicated in* **parts per million (ppm)***. One ppm corresponds to a time error growth of one microsecond per second.*

In PCs, the so-called *real-time clock (RTC)* normally is a crystal oscillator with a maximum drift between 5 and 100 ppm. Applications in signal processing, for instance GPS, need more accurate clocks. Common drift values are 0.5 to 2 ppm.

Definition 3.7 (Jitter)**.** *The* **jitter** ξ *is the unpredictable, random noise of the clock error.*

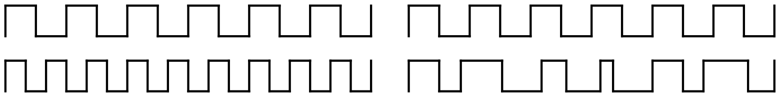


Figure 3.8: Drift (left) and Jitter (right). On top is a square wave, the wall-clock time t^*.

In other words, jitter is the irregularity of the clock. Unlike drift, jitter can vary fast. Jitter captures all the errors that are not explained by drift. Figure 3.8 visualizes the concepts.

3.2 Clock Synchronization

In this section, we study several clock synchronization protocols, including NTP, which is used in most computers and smartphones.

Definition 3.9 (Clock Synchronization). **Clock synchronization** *is the process of matching multiple clocks (nodes) to have a common time.*

A trade-off exists between synchronization accuracy, convergence time, and cost. Different clock synchronization variants may tolerate crashing, erroneous or byzantine nodes.

Algorithm 3.10 Network Time Protocol (NTP) [87]

```
Two nodes, client $u$ and server $v$

while true do
    Node $u$ sends request to $v$ at time $t_u$
    Node $v$ receives request at time $t_v$
    Node $v$ processes the request and replies at time $t'_v$
    Node $u$ receives the response at time $t'_u$

    Propagation delay $\delta = \frac{(t'_u - t_u) - (t'_v - t_v)}{2}$ (assumption: symmetric)
    Clock skew $\theta = \frac{(t_v - (t_u + \delta)) - (t'_u - (t'_v + \delta))}{2} = \frac{(t_v - t_u) + (t'_v - t'_u)}{2}$
    Node $u$ adjusts clock by $+\theta$
    Sleep before next synchronization
end while
```

NTP estimates the packet delay to reduce clock skew. Unbalanced propagation path delays lead to estimation errors. The regular synchronization of NTP limits the maximum error despite unpredictable clock errors. Synchronizing clocks just once is only sufficient for a short time period.

Many NTP servers are public, answering to UDP packets. The most accurate NTP servers derive their time from atomic clocks, synchronized to UTC. To reduce those server's load, a hierarchy of NTP servers is available in a forest (multiple trees) structure.

Definition 3.11 (PTP)**.** *The* **Precision Time Protocol (PTP)** *is a clock synchronization protocol similar to NTP, but which uses* **medium access control (MAC)** *layer timestamps [9].*

MAC layer timestamping removes the unknown time delay incurred through messages passing through the software stack. PTP can achieve sub-microsecond accuracy in local networks.

Definition 3.12 (Global Synchronization)**.** **Global** *synchronization establishes a common time between* **any** *two nodes in the system.*

For example, email needs global timestamps. Also, event detection for power grid control and earthquake localization need global timestamps. Earthquake localization does not need real-time synchronization; it is sufficient if a common time can be reconstructed when needed, also known as "post factum" synchronization. NTP and PTP are both examples of clock synchronization algorithms that optimize for global synchronization. However, two nodes that constantly communicate may receive their timestamps through different paths of the NTP forest, and hence they may accumulate different errors. Because of the clock skew, a message sent by node u might arrive at node v with a timestamp in the future.

Algorithm 3.13 Local Synchronization

```
while true do
    Exchange current time with neighbors
    Adapt time to neighbors, e.g., to average or median
    Sleep before next synchronization
end while
```

Local synchronization is the method of choice to establish *time-division multiple access (TDMA)* and coordination of wake-up and sleeping times in wireless networks. Only close-by nodes matter as far-away nodes will not interfere with their transmissions. Local synchronization is also relevant for precise event localization. For instance, using the speed of sound, measured sound arrival times from co-located sensors can be used to localize a shooter. While global synchronization algorithm such as NTP usually synchronize to an external time standard, local algorithms often just synchronize among themselves, that is, the notion of time does not reflect any time standards. In wireless networks with fixed device locations, one can simplify and improve synchronization.

Algorithm 3.14 Wireless Clock Synchronization with Known Delays

```
Given: transmitter $s$, receivers $u, v$, with known transmission delays
    $d_u, d_v$ from transmitter $s$, respectively.

$s$ sends signal at time $t_s$
$u$ receives signal at time $t_u$
$v$ receives signal at time $t_v$

$\Delta_u = t_u - (t_s + d_u)$
$\Delta_v = t_v - (t_s + d_v)$

Clock skew between $u$ and $v$: $\theta = \Delta_v - \Delta_u = t_v - d_v + d_u - t_u$
```

3.3 Time Standards

Definition 3.15 (TAI)**.** *The* **International Atomic Time (TAI)** *is a time standard derived from over 400 atomic clocks distributed worldwide [43, Résolution 4].*

The involved clocks are synchronized using simultaneous observations of GPS or geostationary satellite transmissions using Algorithm 3.14. Using a weighted average of all involved clocks, TAI is an order of magnitude more stable than the best clock. While a single satellite measurement has a time uncertainty on the order of nanoseconds, averaging over a month improves the accuracy by several orders of magnitude [7].

Definition 3.16 (Leap Second)**.** *A leap second is an extra second added to a minute to make it irregularly 61 instead of 60 seconds long.*

Time standards use leap seconds to compensate for the slowing of the Earth's rotation. In theory, also negative leap seconds can be used to make some minutes only 59 seconds long. But so far, this was never necessary. For easy implementation, not all time standards use leap seconds, for instance TAI and GPS time do not.

Definition 3.17 (UTC)**.** *The* **Coordinated Universal Time (UTC)** *is a time standard based on TAI with leap seconds added at irregular intervals to keep it close to mean solar time at* 0° *longitude [43, Résolution 5].*

The global time standard *Greenwich Mean Time (GMT)* was already established in 1884 [106]. With the invention of caesium atomic clocks and the subsequent redefinition of the SI second, UTC replaced GMT in 1967. Before time standards existed, each city set their own time according to the local mean solar time, which is difficult to measure exactly. This was changed by the upcoming rail and communication networks.

Different notations for time and date are in use. A standardized format for timestamps, mostly used for processing by computers, is the ISO 8601 standard. According to this standard, a UTC timestamp looks like this: `1712-02-30T07:39:52Z`. `T` separates the date and time parts, while `Z` indicates the time zone with zero offset from UTC. Why UTC and not "CUT"? Because France insisted. Same for other abbreviations in this domain, like TAI.

Definition 3.18 (Time Zone)**.** *A* **time zone** *is a geographical region in which the same time offset from UTC is officially used.*

Time zones serve to roughly synchronize noon with the sun reaching the day's highest apparent elevation angle. Some time zones' offset is not a whole number of hours. For instance the *Indian Standard Time* is five and a half hours ahead of UTC.

3.4 Clock Sources

Definition 3.19 (Atomic Clock)**.** *An* **atomic clock** *is a clock which keeps time by counting oscillations of atoms.*

Atomic clocks are the most accurate clocks known. They can have a drift of only about one second in 150 million years, about 2e-10 ppm! Many atomic clocks are based on caesium atoms, which led to the current definition of a second. Others use hydrogen-1 or rubidium-87. In the future, atoms with higher frequency oscillations could yield even more accurate clocks. Atomic clocks are getting smaller and more energy efficient. Chip-scale atomic clocks (CSAC) are currently being produced for space applications and may eventually find their way into consumer electronics.

Atomic clocks can serve as a GPS fallback for data center synchronization [33].

Definition 3.20 (System Clock)**.** *The* **system clock** *in a computer is an oscillator used to synchronize all components on the motherboard.*

Usually, a quartz crystal oscillator with a frequency of some tens to hundreds MHz is used. Therefore, the system clock can achieve a precision of some ns! The *CPU clock* is usually a multiple of the system clock, generated from the system clock through a clock multiplier. To guarantee nominal operation of the computer, the system clock must have low jitter. Otherwise, some components might not get enough time to complete their operation before the next (early) clock pulse arrives. Drift however is not critical for system stability. Applications of the system clock include thread scheduling and ensuring smooth media playback. If a computer is shut down, the system clock is not running; it is reinitialized when starting the computer.

Definition 3.21 (RTC)**.** *The* **real-time clock (RTC)** *in a computer is a battery backed oscillator which is running even if the computer is shut down or unplugged.*

The RTC is read at system startup to initialize the system clock. This keeps the computer's time close to UTC even when the time cannot be synchronized over a network. RTCs are relatively inaccurate, with a common maximum drift of 5, 20 or even 100 ppm, depending on quality and temperature. In many cases, the RTC frequency is 32.768 kHz, which allows for simple timekeeping based on binary counter circuits because the frequency is exactly 2^{15} Hz.

Definition 3.22 (Radio Time Signal)**.** *A* **Radio Time Signal** *is a time code transmitted via radio waves by a time signal station, referring to a time in a given standard such as UTC.*

Time signal stations use atomic clocks to send as accurate time codes as possible. Radio-controlled clocks are an example application of radio signal time synchronization. In Europe, most radio-controlled clocks use the signal transmitted by the *DCF77* station near Frankfurt, Germany. Radio time signals can be received much farther than the horizon of the transmitter due to signal reflections at the ionosphere. DCF77 for instance has an official range of 2,000 km. The time synchronization accuracy when using radio time signals is limited by the (unknown) ToF (same as propagation delay) of the signal. For instance the delay Frankfurt-Zurich is about 1 ms.

Definition 3.23 (Power Line Clock)**.** *A* **power line clock** *measures the oscillations from electric AC power lines, for instance 50 Hz.*

Clocks in kitchen ovens are usually driven by power line oscillations. AC power line oscillations drift about 10 ppm, which is remarkably stable. The magnetic field radiating from power lines is strong enough that power line clocks can work wirelessly. Power line clocks can be synchronized by matching the observed noisy power line oscillation patterns. Power line clocks operate with as little as a few ten µW.

Definition 3.24 (Sunlight Time Synchronization)**. Sunlight time synchronization** *is a method of reconstructing global timestamps by correlating annual solar patterns from light sensors' length of day measurements.*

Sunlight time synchronization is relatively inaccurate. Due to low data rates from length of day measurements, sunlight time synchronization is well-suited for long-time measurements with data storage and post-processing, requiring no communication at the time of measurement. Historically, sun and lunar observations were the first measurements used for time determination [4, 5]. Some clock towers still feature sun dials. ... but today, the most popular source of time is probably GPS!

4

GPS

> *"The satellite-based Global Positioning System (GPS) is perhaps the most significant civil spinoff of the cold war."*
>
> — Bradford W. Parkinson, "Father of GPS" [96]

The *Global Positioning System (GPS)* is one of several *Global Navigation Satellite Systems (GNSS)*. GPS has been such a technological breakthrough that even though it dates back to the 1970s, the new GNSS still use essentially the same techniques, differing mainly in used signal frequencies and modulations. Therefore, we discuss GPS as an example which will enable the reader to understand all GNSS. While we only cover the basic concepts here, the details of the GPS signals, models and system parameters can be found in the official interface document [91].

Definition 4.1 (Global Positioning System)**.** *The* **Global Positioning System (GPS)** *is a* **Global Navigation Satellite System (GNSS)***, consisting of at least 24 satellites orbiting around the Earth, each continuously transmitting its location and time code [38].*

Localization is done in space and *time*! GPS provides location and time information to receivers anywhere on Earth where at least four satellite signals can be received. Line of sight (LOS) between satellite and receiver is advantageous. GPS works poorly indoors, or with reflections. Besides the US GPS, three other GNSS exist: the European Galileo, the Russian GLONASS and the Chinese BeiDou. GPS satellites orbit around Earth approximately 20,000 km above the surface, circling Earth twice a day. The signals take between 64 and 89 ms to reach Earth. The orbits are precisely determined by ground control stations, optimized for a high number of satellites being concurrently above the horizon at any place on Earth.

4.1 Satellites

Algorithm 4.2 GPS Satellite Data Transmission [56]

The code below is a bit simplified, concentrating on the digital aspects, ignoring that the data is sent on a carrier frequency of 1575.42 MHz.
Input: Each satellite has a unique 1023 bit (±1, see below) PRN sequence, plus some current navigation data D (also ±1).

```
while true do
    for all bits $D_i \in D$ do
        for $j = 0 \ldots 19$ do
            for $k = 0 \ldots 1022$ [this loop takes exactly 1 ms] do
                Send bit $PRN_k \cdot D_i$
            end for
        end for
    end for
end while
```

Definition 4.3 (PRN)**. Pseudo-Random Noise (PRN)** *sequences are pseudo-random bit strings. Each GPS satellite uses a unique PRN sequence with a length of 1023 bits for its signal transmissions [56, Sec. 3.2.1.3].*

The GPS PRN sequences are so-called *Gold codes*, which have low cross-correlation with each other. To simplify our math (abstract from modulation), each PRN bit is either 1 or -1.

Definition 4.4 (Navigation Data)**. Navigation Data** *is the data transmitted from satellites, which includes orbit parameters to determine satellite locations, timestamps of signal transmission, atmospheric delay estimations and status information of the satellites and GPS as a whole, such as the accuracy and validity of the data [56, Sec. 3.2.2].*

As seen in Algorithm 4.2, each bit is repeated 20 times for better robustness. Thus, the navigation data rate is only 50 bit/s. Due to this limited data rate, timestamps are sent every 6 seconds, satellite orbit parameters (function of the satellite location over time) only every 30 seconds. As a result, the latency of a first location estimate after turning on a receiver, which is called *time to first fix (TTFF)*, can be high.

4.2 Classical Receivers

Definition 4.5 (Circular Cross-Correlation). *The* **circular cross-correlation** *is a similarity measure between two vectors of length N,* **circularly** *shifted by a given displacement d:*

$$cxcorr(\boldsymbol{a}, \boldsymbol{b}, d) = \sum_{i=0}^{N-1} a_i \cdot b_{i+d \bmod N}$$

The two vectors are most similar at the displacement d where the sum (cross-correlation value) is maximum. The vector of cross-correlation values with all N displacements can efficiently be computed using a fast Fourier transform (FFT) in $\mathcal{O}(N \log N)$ instead of $\mathcal{O}(N^2)$ time.

Algorithm 4.6 Acquisition [125, Ch. 7]

```
Input:
Received 1 ms signal $\boldsymbol{s}$ with sampling rate $r \cdot 1{,}023$ kHz
Possible Doppler shifts $F$, e.g. {-10 kHz, -9.8 kHz, ..., +10 kHz}

Tensor $A = 0$: satellite × carrier frequency × time
for all satellites $i$ do
    $PRN'_i = PRN_i$ stretched with ratio $r$
    for all Doppler shifts $f \in F$ do
        Build modulated $PRN''_i$ with $PRN'_i$ and Doppler frequency $f$
        for all delays $d \in \{0, 1, \ldots, 1{,}023 \cdot r - 1\}$ do
            $A_i(f, d) = |cxcorr(\boldsymbol{s}, \boldsymbol{PRN''_i}, d)|$
        end for
    end for
    Select $d^*, f^*$ that maximize $\max_d \max_f A_i(f, d)$
    Signal arrival time $r_i = d^* / (r \cdot 1{,}023$ kHz$)$
end for
Output: For all $i$: $d^*, f^*$
```

Alternatively, the algorithm may output values only for those satellites with $A_i(f^*, d^*)$ above some threshold, or it may output only the signal arrival times r_i for those satellites, or it may output the full tensor A. Multiple milliseconds of acquisition can be summed up to average out noise and therefore improve the arrival time detection probability. This is called *non-coherent integration.* Meanwhile, in *coherent integration* PRN sequences are extended through repetition before doing the acquisition. Thus, the correlation uses longer signals. Coherent integration results in an improved signal-to-noise ratio (SNR) compared to non-coherent integration, but only when navigation bit flips are accounted for and the Doppler frequency is close enough. Since this means that more fine-grained Doppler shifts have to be tested and because the correlation has superlinear complexity in the length of the input sequences, the computational effort of coherent integration is higher than that of non-coherent integration.

Definition 4.7 (Acquisition)**. Acquisition** *is the process in a GPS receiver that finds the visible satellite signals and detects the delays of the PRN sequences and the Doppler shifts of the signals.*

The relative speed between satellite and receiver introduces a significant Doppler shift to the carrier frequency. In order to decode the signal, a frequency search for the Doppler shift is necessary. The nested loops make acquisition the computationally most intensive part of a GPS receiver.

Algorithm 4.8 Classical GPS Receiver Localization [15]

```
Input:
$h$: Unknown receiver handset location
$\theta$: Unknown handset time offset to GPS system time
$r_i$: measured signal arrival time in handset time system
$c$: signal propagation speed (GPS: speed of light)

Perform Acquisition (Algorithm 4.6)
Track signals and decode navigation data
for all satellites $i$ do
    Using navigation data, determine signal transmission time $s_i$ and location $p_i$
    Measured satellite ToF $d_i = r_i - s_i$
end for
Solve the following system of equations for $h$ and $\theta$:
$||p_i - h||/c = d_i - \theta$, for all $i$
Output: $h^*$ minimizing the sum of the squared residuals
```

GPS satellites carry precise atomic clocks, but the receiver is not synchronized with the satellites. The arrival times of the signals at the receiver are determined in the receiver's local time. Therefore, even though the satellite signals include transmission timestamps, the exact distance between satellites and receiver is unknown. In total, the localization problem contains four unknown variables, three for the handset's spatial location and one for its time offset from the system time. Therefore, signals from at least four transmitters are needed to find the correct solution. Since the equations are quadratic (distance), with as many observations as variables, the system of equations has two solutions in principle. For GPS however, in practice one of the solutions is far from the Earth surface, so the correct solution can always be identified without a fifth satellite. More received signals help reducing the measurement noise and thus improving the accuracy. Since the localization solution, which is also called location fix, includes the handset's time offset Δ, this establishes a global time for all handsets. Thus, GPS is useful for global time synchronization. For a handset with unknown location, GPS timing is more accurate than time synchronization with a single transmitter, like a time signal station (cf. Definition 3.22). With the latter, the unknown signal ToFs cannot be accounted for.

Definition 4.9 (A-GPS)**.** *An* **Assisted GPS (A-GPS)** *receiver fetches the satellite orbit parameters and other navigation data from the Internet, for instance via a cellular network [127].*

A-GPS reduces the data transmission time, and thus the TTFF, from a maximum of 30 seconds per satellite to a maximum of 6 seconds. Smartphones regularly use A-GPS. However, coarse localization is usually done based on nearby Wi-Fi base stations only, which saves energy compared to GPS. Another GPS improvement is *Differential GPS (DGPS)*: A receiver with a fixed location within a few kilometers of a mobile receiver compares the observed and actual satellite distances. This error is then subtracted at the mobile receiver. DGPS achieves accuracies in the order of 10 cm.

4.3 Snapshot Receivers

Definition 4.10 (Snapshot GPS Receiver)**.** *A* **snapshot receiver** *is a GPS receiver that captures one or a few milliseconds of raw GPS signal for a location fix [127, Ch. 4].*

Snapshot receivers aim at the remaining latency that results from the transmission of timestamps from the satellites every six seconds. Since time changes continuously, timestamps cannot be fetched together with the satellite orbit parameters that are valid for two hours. A snapshot receiver can determine the ranges to the satellites modulo 1 ms, which corresponds to 300 km. An approximate time and location of the receiver is used to resolve these ambiguities without a timestamp from the satellite signals themselves.

Definition 4.11 (CTN)**. Coarse-Time Navigation (CTN)** *is a snapshot receiver localization technique measuring sub-millisecond satellite ranges from correlation peaks, like classical GPS receivers [127, Ch. 4].*

A CTN receiver determines the signal transmission times and satellite locations from its own approximate location by subtracting the signal ToF from the receive time. The receiver location and time is not exactly known, but since signals are transmitted exactly at whole milliseconds, rounding to the nearest whole millisecond gives the signal transmission time. With only a few milliseconds of signal, noise cannot be averaged out well and may lead to wrong signal arrival time estimates. Such wrong measurements usually render the system of equations unsolvable, making localization infeasible.

Algorithm 4.12 Collective Detection Receiver Localization [11]

```
Input: A raw 1 ms GPS sample $\boldsymbol{s}$, a set $H$ of location/time hypotheses
In addition, the receiver learned all navigation and atmospheric data

for all hypotheses $h \in H$ do
    Vector $\boldsymbol{r} = 0$
    Set $V$ = satellites that should be visible with hypothesis $h$
    for all satellites $i$ in $V$ do
        $\boldsymbol{r} = \boldsymbol{r} + \boldsymbol{r_i}$, where $\boldsymbol{r_i}$ is expected signal of satellite $i$. The data of vector $\boldsymbol{r_i}$ incorporates all available information: distance and atmospheric delay between satellite and receiver, frequency shift because of Doppler shift due to satellite movement, current navigation data bit of satellite, etc.
    end for
    Probability $P_h = cxcorr(\boldsymbol{s}, \boldsymbol{r}, 0)$
end for
Output: hypothesis $h \in H$ maximizing $P_h$
```

Definition 4.13 (Collective Detection)**. Collective detection (CD)** *is a maximum likelihood snapshot receiver localization method, which does not determine an arrival time for each satellite, but rather combine all the available information and take a decision only at the end of the computation [11].*

CD can tolerate a few low quality satellite signals and is thus more robust than CTN. In essence, CD tests how well location hypotheses match the received signal. For large location and time uncertainties, the high number of hypotheses require a lot of computation power. CD can be sped up by a branch-and-bound approach, which reduces the computation per location fix to the order of one second even for uncertainties of 100 km and a minute.

Several people worked on snapshot GPS receivers, but the technique has not penetrated into commercial receivers yet. Liu et al. [75] presented a practical CTN receiver and reduced the solution space by eliminating solutions not lying on the ground. CD receivers are studied since at least 2011 [11] and have recently been made practically feasible through branch and bound [14].

Another method which considers all satellite signals together, like CD, is *Direct Position Estimation (DPE)* [29]. In addition to the signal amplitude, like CD, DPE also considers the signal phase for the receiver localization. However, an efficient implementation, like the branch-and-bound technique for CD, has not been found, yet.

5

Fast and Robust GPS Fix Using One Millisecond of Data

> *"In this age of the rule of brute force, it is almost impossible for anyone to believe that any one else could possibly reject the law of the final supremacy of brute force."*
>
> — Mahatma Gandhi

Location sensing has proven to be an important prerequisite for many applications. Examples are navigation, tracking, life-logging, research such as animal tracking, and rescue services. Many classes of battery powered devices are more useful when location information is available, such as smartphones, cameras, fitness trackers, smart watches and sensor nodes. For most outdoor scenarios, GPS is the localization system of choice, mainly due to its global coverage and accuracy.

However, continuous GPS receiver operation still consumes too much energy for mobile devices such as fitness trackers or even smartphones, since current receivers cannot be efficiently duty-cycled. When the receiver is switched off for a few minutes to conserve power, it takes a lot of time and energy to compute a new location fix once it is turned back on again. This has far-reaching consequences for many application scenarios. For example, today's GPS receivers make us wait for a first fix, which can be annoying if one wants to navigate an unknown place. Also, geo-tagging photos is not instant and energy consuming. Due to the energy consumption issues, many applications, such as long term tracking, are still out of reach.

In this chapter, we present a receiver which requires only a single millisecond of GPS signal to compute its location. This means that the signal can be recorded and stored locally for later processing. The signal recording can be sent to a remote server which can perform the energy consuming location computation. This translates to a reduction in power consumption as well as an increase in convenience for many applications. For example, the initial location when navigating with your phone can be found within a few milliseconds depending on network latency. A smartwatch or fitness tracker may be able to track its location every few seconds for weeks at a time. When the duty cycle is further reduced, a tracking device that only requires one location fix per hour may run for years on a single coin cell battery. Geo-tagging photos can be simplified to adding a one ms signal recording to the photo which is stored and the location can be computed later on.

The GPS signal that reaches the surface of the earth is weak due to the path loss. To reduce the effects of noise, current receivers track GPS signals over extended periods of time. Since we want to be able to store the recorded signals for later processing, this is not a feasible solution for us. To still increase the noise tolerance of our approach, our solution yields the location fix that best explains the given signal measurement. This means that we do not need to detect satellite ranges which easily can throw off current GPS receivers as well as CTN receivers.

The problem of finding the location that best explains a given signal measurement is non-convex. Hence, the solution cannot be found by iteratively improving a candidate solution in all cases. If the location is approximately known, finding the most likely location can be achieved by computing the likelihoods of all close-by locations and selecting the most likely one. The more uncertain the initial guess about the location and time, the larger the search space (location and time) becomes. Computing all the likelihoods presents a computationally expensive maximization problem in this case. However, we show how the global maximum can be found efficiently using a branch-and-bound approach. The runtime of the algorithm is correlated with signal quality: In good signal conditions, the computational load is low. The worse the signal conditions become, the higher the computational burden. However, the best location and time fix is found in any case. The branch and bound implementation speeds up the acquisition time and hence also the *time to first fix (TTFF).*

We exploit the shape of the likelihood function to achieve higher localization accuracy and robustness. As a result, under similar conditions (signal duration and sampling rate), our method leads to more accurate localization compared to previous approaches. Furthermore, we show that there is a trade-off between the amount of sampled signal used and the accuracy of the localization solution. If we average over two consecutive location fixes from one millisecond of data each, the median error is reduced from 25 to 15 meters. Averaging over 30 fixes (0.03 s of signal), the median error is as low as 6 meters. Tracking a user's location decreases the computational complexity of each consecutive fix as the search space (space and time) is much smaller.

5.1 Related Work

Van Diggelen [127] has introduced the idea of *Coarse-Time Navigation (CTN)*. Using CTN, a location fix can be found from only a few milliseconds of data without decoding any data from the GPS signal. The requirement for this is prior knowledge of the receiver time and location to within a few seconds and 150 kilometers, respectively. Liu et al. [75] showed that since CTN only requires a few milliseconds of data, the raw signal can be stored and the computation can be outsourced or postponed until power is available. This mitigates the problem of high energy consumption for acquisition by not acquiring the satellites on the receiver, enabling duty cycling. However, due to the short signal duration, accuracy and robustness is worse than in classic receiver designs relying on acquisition and tracking stages. Our GPS receiver design extends this idea and can compute a location from a *single* millisecond of signal. Our localization method counteracts the effect of the short signal duration and improves the localization accuracy compared to existing work on CTN. Also, we show how accurate location fixes can be computed from inaccurate time estimates. This allows us to drop the heavy and power consuming DCF-77 clock receiver required by Liu et al. [75]. As a result, our receiver can be miniaturized and can function for years even when there is no clock synchronization except at the very beginning.

A second branch of research is concerned with improving the robustness of GPS receivers. In classical GPS, the receiver location is determined based on signal parameters. The most important ones being Doppler shift and code delay for each satellite. From these parameters, a location in space is computed. Clearly, signal parameters may be erroneously detected which leads to unusable location estimates.

Instead of estimating the signal parameters, Closas et al. [28] showed how the receiver location can be estimated directly and how this can improve the robustness of GPS receivers. We refer to the basic idea as *collective detection (CD)*, but it is also called *direct positioning* or *combined detection* in the literature. Evaluations of CD have been performed in both simulation and practice [11, 26, 28]. The main concern is the computational complexity introduced by the high-dimensional search space. Also, the likelihood function is generally non convex, prohibiting standard greedy maximization methods. Optimizations such as the one proposed by Axelrad et al. [11] reduce the computational complexity but cannot guarantee that the best possible location is found. We improve the robustness of our approach by applying CD. Especially so in multipath environments because CD finds the globally best solution whereas classical receiver designs depend on correct pseudorange estimates for each individual satellite. Hence, one bad pseudorange estimate can throw off the classical solution whereas the most likely location (in CD) may still remain unaffected. However, CD is expensive in terms of computation. To alleviate this drawback of CD we introduce a branch-and-bound algorithm which yields reduced computational complexity while still guaranteeing that the best possible solution is found.

5.2 GPS Fundamentals

The GPS system conceptually consists of three parts: the control segment, the space segment and the user segment. The space segment nominally consists of 24 satellites orbiting the Earth [38]. A network of monitor stations and ground antennas makes up the control segment. It is primarily used to monitor the satellites' state and keep track of their ever-changing orbits. The orbits need to be known accurately for good localization accuracy [38].[1] The third – and for our discussion most important – part of GPS are the receivers, making up the user segment.

5.2.1 GPS Signal

The satellites transmit signals in different frequency bands. These include at least the so-called L1 and L2 frequency bands at 1.57542 GHz and 1.2276 GHz [38]. The signals are transmitted through a helix array antenna which right-hand circularly polarizes the signals [38]. This helps suppressing multipath signals at a receiver because a reflection of the signal polarizes it in the opposite direction. In order to distinguish the signals from different satellites and to extract the signals from the background noise, code division multiple access (CDMA) is used.

[1] Further information can be found at `http://www.gps.gov/systems/gps/control/`

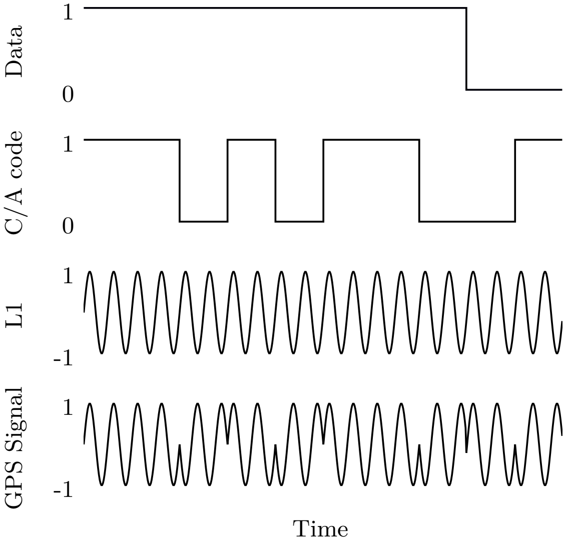


Figure 5.1: The structure and modulation of the GPS Signal. The binary data and C/A code are mixed with the carrier frequency (L1) using the BPSK modulation scheme.

Figure 5.1 shows the modulation scheme utilized in GPS. The Coarse/Acquisition code (C/A code) is a sequence of 1023 bits which is unique for each satellite. Specifically, Gold codes are used to achieve favorable correlation and cross-correlation properties [127]. Because Gold codes look like random bit strings, C/A codes are also called pseudo-random noise (PRN) sequences. The C/A code is transmitted at 10.23 MHz which means it repeats every millisecond. The data is transmitted at $50\frac{\text{bit}}{\text{s}}$ and hence, each bit contains 20 complete C/A cycles. The data and C/A code are merged using an XOR before being mixed with the L1 or L2 carrier. Figure 5.1 shows how the GPS signal is generated. Note that for better readability, the C/A frequency and the L1 frequency do not have the correct ratio. The data that is broadcast contains a timestamp (called HOW) which can be used to compute the location of the satellite when the packet was transmitted. However, to do this, the receiver needs accurate orbital information (called ephemeris) about the satellite which changes over time. While the HOW timestamp is broadcast every six seconds, the ephemeris data can only be received if the receiver can decode at least 30 seconds of signal.

5.2.2 Localization

Classical GPS receivers use three stages when obtaining a location fix.

Acquisition. First, the set of available satellites has to be found. This can be achieved by correlating the received signal with the known C/A codes from the satellites. Since the satellites move at considerable speeds, the signal frequency is affected by a Doppler shift. Hence, receivers usually correlate the received signal with C/A codes with different Doppler shifts.

Tracking. After a set of satellites has been acquired, the data contained in the broadcast signal is decoded. Doppler shifts and C/A code phase are tracked using tracking loops. After the receiver obtained the ephemeris data and HOW timestamps from at least four satellites, it can start to compute its location.

Localization. Localization in GPS is achieved using signal *time of flight (ToF)* measurements. Specifically, the ToFs are the difference between the arrival times of the HOW timestamps decoded in the tracking stage of the receiver and those signal transmission timestamps themselves. Due to missing time synchronization of GPS receivers with the system time, the corresponding satellite distances contain a common bias and are therefore called *pseudoranges*.

Assuming the devices are synchronized, the localization is geometrically simple: The location of the mobile handset lies at the intersection of the spheres around the stations with the radii corresponding to the measured ToFs. But, while the GPS satellites operate on an atomic frequency standard, the receivers are not synchronized to the GPS time. Therefore, the local time at a receiver is unknown and the localization is done using the pseudoranges. That problem formulation just contains one more variable, which is the receiver time. Hence, measurements from at least four instead of three satellites are needed for the problem to be well-defined. The receiver location is usually found through a least-squares optimization.

5.2.3 Assisted GPS

A disadvantage of GPS is the low bit rate of the navigation data encoded in the signals transmitted by the satellites. The minimal data necessary to compute a location fix, which includes the ephemerides of the satellites, repeats only every 30 seconds. In order to decode all that data, the receiver has to continuously track and process the satellite signals, which induces a high energy consumption. Furthermore, upon starting up a receiver, a location will not be instantly available. To overcome this drawback, receivers can run continuously, but this consumes even more power.

Assisted GPS (A-GPS) drastically reduces the start-up time by fetching the navigation data over the Internet, commonly by connecting via a cellular network. Data transmission over cellular networks is faster than decoding the GPS signals and normally only takes a few seconds. The ephemeris data is valid for at least 30 minutes. Using that data, also the acquisition time can be reduced since the set of available satellites can be estimated along with their expected Doppler shift. With A-GPS, the receiver still needs to extract the HOW timestamps from the signal. But since those timestamps are transmitted every six seconds, that is roughly how much time it takes an A-GPS receiver to compute a location fix.

5.2.4 Coarse-Time Navigation

Coarse-Time Navigation (CTN) is an A-GPS technique which drops the requirement to decode the HOW timestamps from the GPS signals. Van Diggelen [127] describes the concept in detail. The only information used from the GPS signals are the phases of the C/A code sequences which are detected using a matched filter. Those C/A code arrival times are directly related to the sub-millisecond parts of the corresponding ToFs. The number of whole milliseconds of the signal ToF are resolved with a known approximate location and time. Because the signals travel at the speed of light, which is about 300 km per millisecond, in order to be able to resolve the number of whole milliseconds unambiguously, the deviation may at most be 150 km from the correct values. Here, the deviation is defined as the time offset multiplied by the speed of light plus the location distance. Since the PRN sequences repeat every millisecond, without considering navigation data bit flips in the signal, CTN can in theory compute a location from one millisecond of the sampled signal. But since bit flips can happen, to make sure all visible satellites can be used, two milliseconds are necessary. With such short signal recordings, clearly noise becomes a major issue, because noise cannot be filtered out as easily as with much longer recordings of several seconds. But the advantage of this extremely short recording period is that the signal processing is fast and power-efficient and thus also the latency of a first fix. Also, since no metadata has to be extracted from the GPS signal, CTN may be able to compute a location even if the GPS signal cannot be decoded anymore due to noise or attenuation.

5.2.5 Collective Detection

Collective detection builds upon the observation that detecting peaks in the correlation functions of individual satellites might yield sets of pseudoranges which are not consistent with the laws of physics. By searching a solution in space and time directly, this can be avoided. The problem then consists of finding the most likely location given the received signal. From a given hypothetical location and time (referred to as hypothesis in the following), the corresponding ranges of the satellites and therefore the ToFs can be inferred. Figure 5.2 shows how the correlation functions of the received signal with PRN codes of different satellites on the left. On the right, the same correlation functions are circularly shifted by the expected ToF at the correct location. That makes the correlation peaks of all four satellites align. A receiver can exploit this by combining corresponding correlation values from all the satellites to compute a likelihood measure. This is essentially what our receiver does. Erroneous peaks in the correlation function most likely never align which improves noise resistance. Commonly, the hypothesis pseudo-likelihood is defined as the sum of the satellite pseudo-likelihoods, but one could also use other measures, for instance the product.

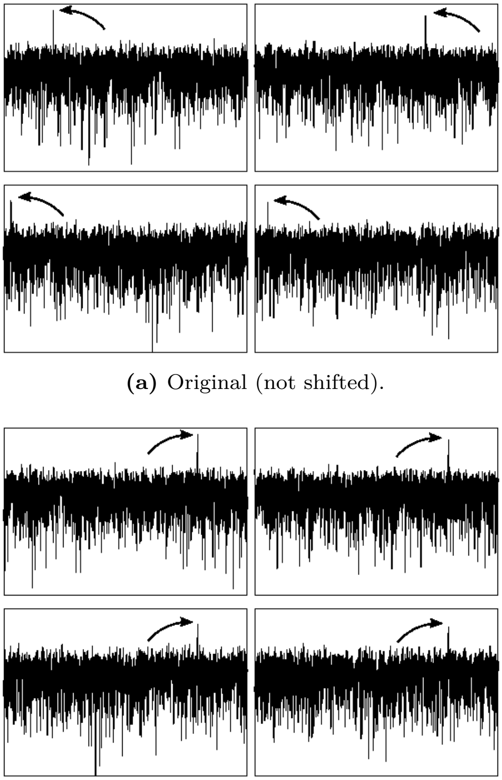
5.3 Localization Method

The basic idea of our method is to asses the quality of many hypothetical receiver states $h = (h^p, h^t)$ which consist of the receiver location h^p and time h^t. The quality of a hypothesis is determined through a likelihood function which assigns a pseudo-likelihood to the hypothesis given external information and the observed signal. This likelihood $\mathcal{L}(h)$ is a measure of how well the observed signal matches the signal expected at a hypothesis h.

5.3.1 Likelihood

Given a hypothesis h, we can use the knowledge about the satellites' signal transmission times and orbits (from the navigation data) to compute the expected signal phase $\phi_i(h)$ arriving at the receiver from the i^{th} satellite. This is discussed in detail in Section 5.3.2. Hence, for any hypothesis h we can expect a C/A code with phase $\phi_i(h)$ from satellite i in the arriving signal. We can check how well the received signal $r(t)$ matches this expectation by computing a single correlation value with satellite i's C/A code $\mathrm{ca}_i(t)$.

$$c_i(h) = \sum_{\tau=0}^{1\text{ms}} |r(\tau) \cdot \mathrm{ca}_i(\tau - \phi_i(h))| \tag{5.1}$$



(a) Original (not shifted).

(b) Shifted (circularly) according to the distance from the receiver to the corresponding satellite.

Figure 5.2: Correlation functions for four satellites. Above are the correlations of the received signal with the PRN sequences of four different satellites. The spikes indicating the beginning of the PRN codes in the received signal are marked with arrows. If we shift the correlation vectors according to the true distance to the satellites, we see below that the peaks all align.

If our hypothesis h is correct, we expect large correlation values c_i for satellites whose signal can be received, because the code phase of the C/A code in the received signal match the expected code phase $\phi_i(h)$. For satellites that are heavily attenuated or reflected, c_i will be almost completely random. We define our likelihood function as the sum of the correlation values for a given hypothesis over all *visible* satellites, whose indices are denoted by the set V.

$$\mathcal{L}(h) = \sum_{i \in V} c_i(h) \tag{5.2}$$

The receiver location and time are estimated by selecting the hypothesis h^* which maximizes the likelihood measure:

$$h^* = \underset{h \in F}{\arg\max}\, \mathcal{L}(h)$$

where F is a set of *feasible* (location, time) tuples.

5.3.2 Computing the C/A Code Phase

To compute the likelihood of a hypothesis h, we need to know the C/A code phases $\phi_i(h)$ of the visible satellites. In the following, we assume that the signal ToF $d_i(h)$ is mainly determined by the distance between receiver and satellite. Note that the maximum signal ToF to a receiver on Earth is 87 ms [125]. During such a short time, a receiver's movement does not have a significant effect on the signal ToF. However, the fast satellite movement has. Therefore, we compute the ToF at the transmission time t_i of a signal even though the receiver may still travel for an additional 87 ms.

The code phase $\phi_i(h)$ relates to the transmission time $t_i(h)$ of the received signal as follows:

$$\phi_i = t_i(h) \bmod 1 \text{ ms}$$

The transmission time $t_i(h)$ of the received signal at time h^t are related by the ToF $d_i(h)$ between the hypothetical location and the satellite location.

$$t_i(h) = h^t - d_i(h) \tag{5.3}$$

The ToF can be found by dividing the spatial distance between the hypothetical location h^p and the satellite location p_i by the speed of light C:

$$d_i(h) = \frac{||h^p - p_i(t_i(h))||}{C} \tag{5.4}$$

The ToF $d_i(h)$ depends on the distance between the satellite location p_i at the transmission time $t_i(h)$ and the hypothetical location h^p. The satellite location $p_i(t_i(h))$ at a given time can be computed from the ephemeris.

So, the ToF $d_i(h)$ can be found knowing the transmission time $t_i(h)$ which itself can be found knowing the satellite location $p_i(t_i(h))$ which in turn can only be found knowing the transmission time $t_i(h)$ for which the ToF $d_i(h)$ needs to be known. This circular dependency can be resolved by a short fixed-point iteration which exploits the difference between the speed of light and the satellite movement speed.

Namely, the signal ToFs from a satellite to a receiver on Earth range between 67 and 86 ms [125]. If we compute the signal transmission time using Equation 5.3 and this crude estimate, we get $t_i \approx h^t - (67 + 86)/2$ ms $\approx h^t - 76.5$ ms. The estimation error in the transmission time $t_i(h)$ is at most 9.5 ms. The maximum satellite speed relative to a receiver on Earth is 929 m/s [125]. This means that our estimate for $t_i(h)$ of 9.5 ms leads to a worst case satellite location estimation error of 9.5 ms $\cdot$ 929 m/s $= 8.83$ m. Using this new satellite location error, the second iteration starts with a new estimate of the transmission time $t_i(h)$, based on a satellite location error which is at most 8.83 m. Hence, the ToF estimation error is at most 8.83 m$/C = 19.4$ ns. The satellite location estimate that can be achieved using this ToF estimate already has a negligible error of 19.4 ns$\cdot$929 m/s $=$ 18 µm.

5.3.3 Search Region

To guarantee the uniqueness of the solution, we limit the search region in which the set F of feasible hypotheses is contained. As GPS signals travel at the speed of light C, the C/A code phase of a satellite are the same for two hypotheses if their distances to the satellite differ by $k \cdot C \cdot 1\text{ms} \approx 300\text{km}$ for integer values for k. To avoid this affecting our results, we bound the search region in which the set F of feasible hypotheses is contained to a diameter of 300 km. Most likely the correct solution can still be found in larger areas, especially when more than four satellites are visible. Note that the correspondence between time error and range error is given by the maximum relative satellite speed against a receiver, which is less than $1\,\frac{\text{km}}{\text{s}}$ on the Earth surface [125]. For instance, a location range of 100 km and a time range of 50 km / $1\frac{\text{km}}{\text{s}} = 50$ s are guaranteed to deliver a unique solution.

For bounding the solution domain, one can use the antenna location of a cellular network as a reference. When the signal of the satellites is strong enough, we can also find the approximate receiver location with an idea presented by Liu et al. [75]. The authors show how the measured

Doppler shift of a signal limits the receiver location to a cone. The receiver location is then at the intersection of the cones from each satellite. If we do not compute an initial fix, we can use the last computed location as an approximation for the new location.

5.3.4 Visible Satellites

The set V contains the indices of all potentially visible satellites. It is assumed to be the same for all hypotheses $h \in F$ and is determined as all the satellites with an elevation above the horizon larger than five degrees, as seen from the center of the search region. In theory, V is a function of a hypothesis h and the ephemerides, from which the satellite elevation angles can be computed. However, it is safe to assume V is fixed with respect to all hypotheses since the elevation angles barely change within the search regions we consider. Also the Earth's rotation during the signal transmission can be neglected when computing the elevation angle of a satellite.

5.3.5 Space Discretization

The computation of the correlation values given in Equation 5.1 shifts the locally generated C/A code by its expected phase. In our case, the expected phase is rounded such that we shift by an integer value corresponding to one sampling interval T_s of the receiver. This helps to simplify the computation of the likelihood function as no signal interpolation is required. Due to the rounding, the likelihood of two hypotheses that are close may lead to the exact same set of C/A code phases ϕ_i for all visible satellites.

Ideally, we spread hypotheses in the search range such that no two hypotheses correspond to the same set of C/A codes to conserve computation resources. Also, we would like to have one hypothesis for every set of C/A code ranges which can be achieved within the search region.

Depending on the sampling interval T_s, we can compute the range difference that is required to change the value of the rounded C/A code phase ϕ_i. Namely, the corresponding "length" of a sample is $\lambda_s = cT_s$, where c is the speed of light ($\lambda_s \approx 37$ m for $T_s = \frac{1}{8\text{MHz}}$). Thus, for each satellite, the solution space is sliced into *spherical shells* with a slice width of λ_s. Each hypothesis in such a slice produces the same rounded expected C/A code phase ϕ_i.

With multiple satellites, the space is sliced in several directions as shown in Figure 5.3. This divides the solution space into volumes in which all the hypotheses correspond to the same rounded C/A code phase and therefore equal likelihoods. Figure 5.3 shows a two-dimensional example.

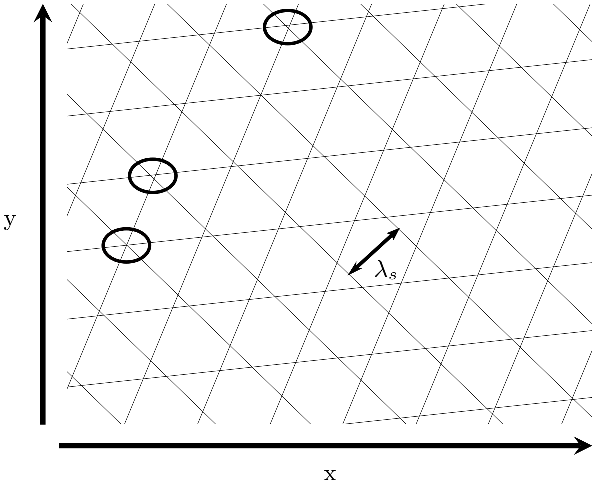


Figure 5.3: Two-dimensional search space discretization example. In this example, three satellites are visible, which cause three groups of parallel lines slicing the search space. When crossing a line, the expected C/A code phase ϕ_i for the corresponding satellite is rounded to the previous or next sampling period T_s. All locations inside a bounded area have the same likelihood. Extremely small regions may exist (indicated by circles).

Since we do not know the exact shape of the division of the search space in the volumes of equal observations, we sample the space with a regular grid. Ideally, this grid would be dense enough to "capture" all these volumes. However, some of these volumes can be infinitely small and thus, with any fixed grid density, we might not sample some volumes and therefore not find the most likely hypothesis. This means that we cannot guarantee that we sample the volume which corresponds to the highest likelihood that is achievable given the observations.

Luckily, we can make sure that we do not miss the correct solution completely because no hypothesis is close enough. We do this by selecting the grid such that neighboring points are λ_s apart. Like this, each hypothesis represents a cube of side length λ_s. Such a cube has a diameter of $\sqrt{3}\lambda_s \approx 1.7\lambda_s < 2\lambda_s$. Since the space is divided into those cubes, an uncovered area can at most be half a diameter apart from the nearest hypothesis, that is the distance to the nearest hypothesis is less than λ_s. Note that a distance smaller than λ_s can at most cross one slice boundary for each satellite. This means that for an uncovered area and its nearest hypothesis, the expected code phases ϕ_i are at most one sampling interval T_s apart.

The key observation is that our (and also common) GPS receivers oversample the GPS signals. For the correlation, this means that the peaks are not confined to a single sample of length T_s. Rather, their neighboring values are quite high as well and form a triangle-like pattern. Without noise, the correlation values at a distance of k samples from the peak have a value of at least $(1 - k \cdot 2 \cdot f_{\mathrm{PRN}}/f_s)$ times the value of the peak. f_{PRN} is the rate of the PRN sequences (1.023 MHz). f_{PRN}/f_s is the fraction of the locally generated PRN sequence which does not match the correct part of the PRN in the signal. For a sampling rate of 8 MHz ($f_s = \frac{1}{T_s}$) for instance, the directly neighboring values of the peak are at least 74 % as high as the peak itself, for a sampling rate of 56 MHz at least 96 %. Assuming the used sampling rate is at least 8 MHz, the found correlation values may at most be 26 % smaller compared to the largest one. Alternatively, we could filter the correlation values such that the correlation value at an index contains the highest correlation values amongst its direct neighbors. In this case, we are guaranteed to find the highest achievable likelihood, but the likelihood function is less sharp. The trade-off that we make here is a decrease in the likelihood at the correct location.

5.3.6 Time Discretization

The hypotheses also have to be spread in the time domain. As in the spatial discretization above, we have to make sure that we sample densely enough, such that we do not miss the most likely location. If the hypothetical time for the correct location h^p is off by as few as $10 \cdot T_s$, its likelihood will be completely random (assuming $T_s = 8\mathrm{MHz}$). This follows from the same argument about the shape of the PRN autocorrelation function above. In order to allow for more coarse sampling in the time domain, we exploit the fact that the expected C/A code phase $\phi_i(h)$ is approximately constant when varying the hypothetical time h^t by less than one ms. Hence, we simplify the computation of the correlation values $c_i(h)$ for hypotheses that are identical up to a difference in time t_μ which is smaller than 1 ms.

$$c_i(h, t_\mu) = \sum_{\tau=0}^{1\mathrm{ms}} |r(\tau) \cdot \mathrm{ca}_i(\tau - \phi_i(h) - t_\mu)| \tag{5.5}$$

We can simplify the computation of $c_i(h, t_\mu)$ for all $t_\mu \in [0, T_s, 2 \cdot T_s, \ldots, 1\text{ms}]$ using the correlation function C_i:

$$C_i(t_\mu) = \sum_{\tau=0}^{1\text{ms}} |r(\tau) \cdot \text{ca}_i(\tau - t_\mu)| \tag{5.6}$$

Note that the correlation function $C_i(t_\mu)$ can be computed independent of the hypothesis. By shifting the correlation function $C_i(t_\mu)$ of the received signal with the C/A code according to the expected phase $\phi_i(h)$, we can simplify the computation of the likelihood as follows:

$$\mathcal{L}(h) = \max_{t_\mu} \sum_{i \in V} C_i(t_\mu - \phi_i) \tag{5.7}$$

This allows us to choose the time domain to be sampled at up to 1 ms intervals without leaving a good solution undetected.

In the worst case, an inaccurate time hypothesis shifts the most likely location by the maximal speed of the satellites relative to the earth's surface ($1\,\frac{\text{km}}{\text{s}}$). This means that the localization error is expected to increase less than 1 m if the hypothetical time is off by 1 ms. Hence, we can further increase the intervals at which the time domain is sampled. This does not negatively affect the observations about the spatial discretization. We are still guaranteed to observe hypotheses that are close to the highest achievable likelihood.

5.3.7 Averaging Over Likely Hypotheses

So far, we only discussed choosing the hypothesis with the largest likelihood as the solution. As described in Section 5.3.5, hypotheses that are near the correct solution should get a high likelihood as well, because the PRN is oversampled and therefore its auto-correlation function has a triangular shape around the peak. To improve localization accuracy, we consider the set of hypotheses H with the highest likelihoods. The set of most likely hypotheses is then combined using a weighted average.

$$\bar{h^p} = \sum_{h \in H} \mathcal{L}(h) \cdot h^p$$

This averaging allows estimating locations more fine-grained than the granularity of the hypotheses grid. Also, the averaging over multiple hypotheses should make the localization more robust to measurement noise because a hypothesis with a slightly higher likelihood than the one at the correct location will not completely move the final location estimate to that

Algorithm 5.5 Finding the n most likely points given a search space defined by a hypothesis h.

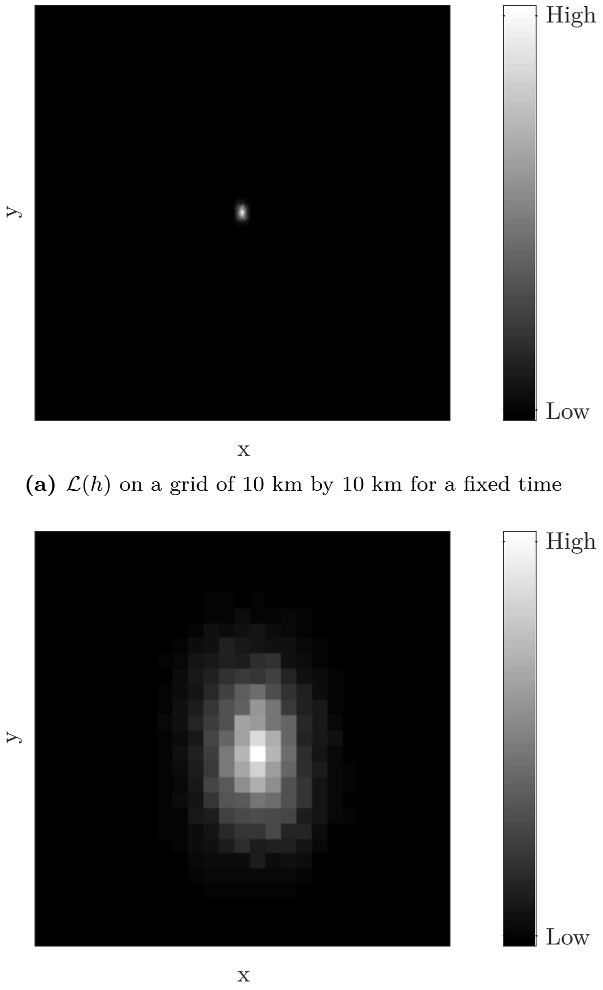
```
procedure S = GetMostLikelyPoints($n$,h)
    $n$: the number of likely points contained in S.
    h: the initial hypothesis defining the search space.

    $\mathrm{h}_{\mathrm{lmax}}$ = maxLikelihood( h )
    queue.add(h)
    S = $\emptyset$
    while queue.hasElement() do
        h = queue.popMostLikely()
        if $\mathrm{h}_{\mathrm{lmax}} \leq \min_n(\mathrm{h}_{\mathrm{lmin}} \in S)$ then
            continue
        end if
        $\mathrm{h}_{\mathrm{lmin}}$ = likelihood( h )
        $\mathrm{h}_{\mathrm{lmax}}$ = maxLikelihood( h )
        S.add(h)
        h[1] ... h[16] = splitHypothesis( h )
        for h[i] = h[1] ... h[16] do
            $\mathrm{h[i]}_{\mathrm{lmax}} := \mathrm{h}_{\mathrm{lmax}}$
            queue.add(h[i])
        end for
    end while
end procedure
```

wrong location. In Section 5.4 we discuss the performance impact of the averaging as opposed to only selecting the most likely hypothesis.

5.3.8 Efficient Implementation with Branch and Bound

Figure 5.4 shows horizontal cuts of example distributions of our likelihood computed from a one millisecond window of samples in good signal conditions. Our branch-and-bound method exploits this shape of the likelihood function under clear signal conditions, avoiding the computation of all likelihoods in the search space. The search space as discussed in Sections 5.3.3 and 5.3.5 is large as the hypotheses are spread at a distance of 37 m from each other and the search space spans 200 km × 200 km × 30 km. In addition to this, the time domain is searched within 10 s at intervals of 40 ms. This means that there are roughly $2 \cdot 10^{12}$ hypotheses which need to be tested. To reduce the number of hypotheses for which we need to com-



(a) $\mathcal{L}(h)$ on a grid of 10 km by 10 km for a fixed time

(b) $\mathcal{L}(h)$ on a grid of 1 km by 1 km for a fixed time

Figure 5.4: In situations with line of sight between receiver and satellites, the likelihood function is smooth and unambiguous. The figures shown are a cut through the search space where the time and height of the receiver have been fixed at the values corresponding to the most likely hypothesis. The distance between two points in the grid is approximately 37 meters.

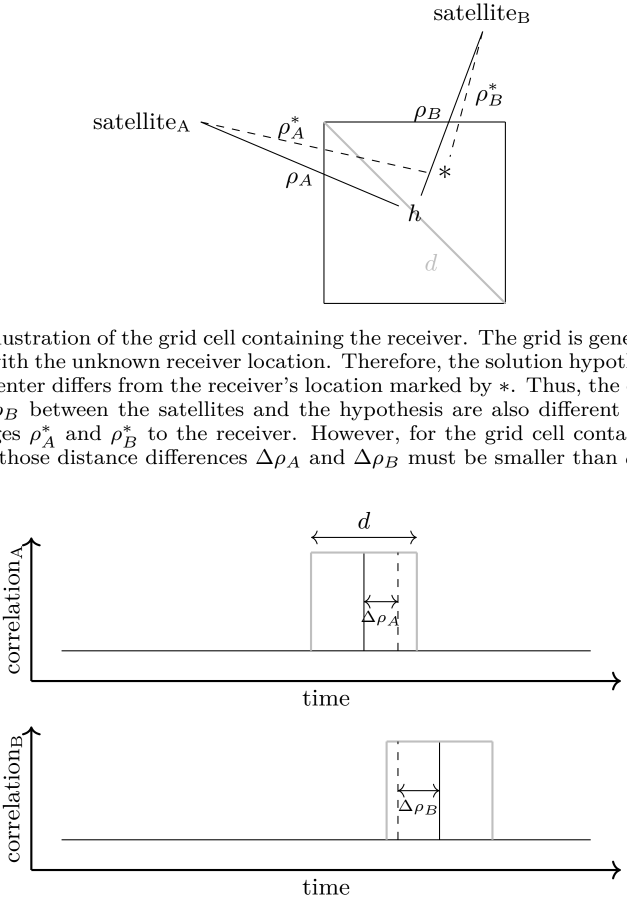
pute the likelihood, we employ a branch-and-bound method as described in Algorithm 5.5. To do so, we need a method to compute both an upper- and lower-bound on the achievable likelihood (indicated by $\mathrm{h_{lmax}}$ and $\mathrm{h_{lmin}}$) within an area defined by a hypothesis. Note that in the algorithm, a hypothesis h contains the center location in x,y,z, and t and also the size of the search space around it in all dimensions (x,y,z,t). The initial hypothesis covers the entire search space, that is, it extends over 200 km × 200 km × 30 km × 10 s. We approximate the lower bound of achievable likelihoods within an area as the likelihood of the hypothesis itself (likelihood(h) in the Algorithm). For the upper bound (maxLikelihood(h) in the Algorithm), we use the expected code phases ϕ_i along with the size of the area covered by the hypothesis. Note that the larger the area covered by a hypothesis, the larger the uncertainty about the possible code phases ϕ_i. The uncertainty is given by the diagonal of the area covered divided by the speed of light. An illustration is shown in Figure 5.6a. For a hypothesis with a diagonal of 10 km, the uncertainty is roughly 33 microseconds which corresponds to roughly 270 sample intervals T_s at 8 MHz. This can efficiently be taken into account when computing the likelihood as described in Equation 5.7. Instead of utilizing the correlation function as described in Equation 5.6, we apply a max-filter first.

$$C_i'(t_\mu) = \max_{\tau \in R} C_i(t_\mu) \tag{5.8}$$

R is the set of possible shifts that can be expected within the region covered by a hypothesis. In the example above with 10 km diagonal, $R = [-16.5\ \mu\text{s}, 16.5\ \mu\text{s}]$. The likelihood computation stays the same as in Equation 5.7 but using $C_i'(t_\mu)$ instead of $C_i(t_\mu)$. This yields the highest possible likelihood, as detailed in Figure 5.6b. To further speed up the computation, the max-filtered correlation functions can be pre-computed as it is the same for all hypotheses covering areas of the same size.

Hypotheses in the queue are processed according to their maximum achievable likelihood $\mathrm{h_{lmax}}$ (popMostLikely()). This is crucial as areas with great potential are explored first, making it more likely that bad areas are not further explored. Each processed hypothesis is split in two in all dimensions (x,y,z,t) which leads to 16 new hypotheses, covering a smaller volume of the search space each. As soon as a hypothesis cannot achieve a higher likelihood than the n best hypotheses already observed, it is not further split up and discarded. The method guarantees that the n most likely points are found as only hypotheses are discarded which cannot possibly achieve a high enough likelihood.

The performance of the algorithm depends on the quality of the received signal as the bounds will be more accurate for a smooth likelihood function.



(a) 2D illustration of the grid cell containing the receiver. The grid is generally not aligned with the unknown receiver location. Therefore, the solution hypothesis h in the cell center differs from the receiver's location marked by $*$. Thus, the distances ρ_A and ρ_B between the satellites and the hypothesis are also different from the true ranges ρ_A^* and ρ_B^* to the receiver. However, for the grid cell containing the receiver, those distance differences $\Delta\rho_A$ and $\Delta\rho_B$ must be smaller than $d/2$.

(b) Idealized correlation vectors of two satellites, *shifted* according to the estimated satellite distances from the grid cell center. Length d is the diagonal of a grid cell. The widening of the correlation peak due to the sliding window maximum is shown in gray. The dashed peaks indicate the shift, or code phase, for the true receiver location. For the cell containing the true receiver location, even if the correlation peaks do not exactly align, the sliding window maximum with window width d ensures that the computed upper likelihood bound is at least as high as the likelihood at the point where all peaks overlap. (With low enough noise, that point should be the true receiver location.) Therefore, that correct cell is never dropped in the branch-and-bound process, as its likelihood (upper bound) must lie above the highest likelihood of any tested hypothesis so far.

Figure 5.6: A sliding-window maximum over the correlation vectors can be used to compute an upper bound on the likelihoods of all hypotheses within a grid cell. For simplicity, this example is given in a two-dimensional space instead of the actual four dimensions, location and time.

We analyze the performance degradation as the signal quality deteriorates in Section 5.4.

5.3.9 Local Oscillator Frequency Bias

In practice, one of the problems we have to deal with is the frequency error of the local oscillator (LO) in the front end. The LO is not only used for the generation of the reference frequency for the frequency down-conversion, but also as the clock of the ADC. Therefore, the LO error influences two parameters. First, the observed frequencies of the signals from the satellites change. Second, the effective sampling rate or the time that passes per sample changes. Akos [1] states that the frequency for the locally generated C/A code should match the actual frequency with an accuracy better than 250 Hz. Otherwise, correlation peaks are hard to find even under good signal conditions. To get an SNR close to the optimum possible, the accuracy of the frequency should be much better.

During the acquisition phase in classical receivers, the Doppler shift of each satellite is estimated by correlating the received signal with multiple frequency shifted versions of the C/A code. The frequency shifted C/A code which matches the received signal the best is used to estimate the ToF and also gives information about the sum of the LO offset and the Doppler shift. After the acquisition, the Doppler shift, and hence the LO, is known only approximately to reduce the computational complexity during acquisition. This approach could be replicated in our solution to estimate the LO offset.

Similar to the search performed in classic receivers, we could track the LO offset by computing the C/A code correlation functions for different frequency offsets. Note that since we do compensate for the Doppler shifts using our prior knowledge, we only need to estimate the LO offset instead of the sum of the LO offset and the Doppler shifts of each satellite.

In our test setup described in Section 5.4, we measured the LO offset initially using the classical GPS approach. We observed that the offset stayed almost constant even over more than a year. Therefore, careful calibration of the LO can reduce the impact of its errors to an extent that is acceptable. Over the course of 1.5 years, all experiments were performed with the same, constant LO offset (+1.9 ppm).

For an oscillator which does not exhibit such a stable frequency offset over a long time, it would be possible to regularly update the frequency error estimate by correlating with a local signal with slightly lower and higher frequency – similar to the early-late tracking of the code phase in classical receivers – in situations with good SNR. Since the frequency error will not change quickly, a low SNR of the received signal can be tolerated for an extended period of time without significant performance degradation.

5.4 Evaluation

For the evaluation of our method, we used an Ettus USRP B200 software radio with a standard GPS patch antenna from Trimble Navigation. Samples were recorded as 8 bit I/Q samples with 8 MHz sampling frequency. We made recordings of several minutes and cut out windows with one millisecond length every 0.999 seconds. We did not choose exactly one second, since bit flips in the navigation signal, which severely degrade the signal quality, can occur every 20 milliseconds. To prevent these to always have an influence on the same satellites' signal, we chose a slightly shorter interval. Samples with 8 bits were used since this is the lowest number of bits supported by the board's driver. However, we expect that the performance does not significantly vary when only 2 bit samples are used, because using 2 bit samples degrades the SNR by only 0.55 dB [125] (Section 6.12).

We used navigation data originally broadcast from the satellites, which we downloaded from NASA's archive of space geodesy data[2] [94]. For the time synchronization, we determined the time of the first sample received from the RF front-end with the Network Time Protocol (NTP). The start time of subsequent one millisecond windows was estimated by counting the number of elapsed samples in the recorded data stream.

To evaluate the accuracy of our algorithm, we placed the receiver antenna on a survey point located on our university building. The location of this point is known accurately. We expect errors in its location to affect our results negatively giving us a lower bound of the performance.

Unless otherwise indicated, experiments were performed under good signal conditions (direct line of sight to most satellites above the horizon) and the search space size was 200 km $\times$ 200 km $\times$ 30 km $\times$ 10 s. The reason for the size of the search in the time dimension is that with a low energy oscillator with maximum drift of 5 ppm and an initial time error of 50 ms (easily achievable with NTP), a range of ± 5 seconds covers a duty-cycle interval of more than 11 days. So, ± 5 seconds are a large bound on the time search especially since time inaccuracies can be compensated when a fix is computed.[3]

For each processed one millisecond window of signal, we varied the grid of hypotheses uniformly at random in each dimension, up to half the distance between two points. This eliminates possible bias from a specific localization

[2] We used the "Daily GPS Broadcast Ephemeris Files" data set that can be found at `http://cddis.nasa.gov/Data_and_Derived_Products/GNSS/broadcast_ephemeris_data.html`

[3] We think a more reasonable upper bound on the duty-cycle interval would be one day, which means that with such a large time search, we could tolerate many failed localizations between two successful ones, for instance when the receiver is indoor for a long time period.

of the grid. For instance, if one hypothesis always matched the correct receiver location and time exactly, the results might look much better than if the correct location and time lie in the center between the closest hypotheses in each dimension.

5.4.1 Averaging Over Likely Hypotheses

First, we evaluate how the accuracy depends on the number of most likely points used to compute the weighted average as described in Section 5.3.7. Figure 5.7 shows the cumulative distribution functions of 501 fixes covering approximately 500 seconds with the duty cycle of 0.999 s. The shown numbers of points are $\{1, \ldots, 7\}$ to the power of four. Since the correlation peaks are triangular due to oversampled C/A codes, we expect the points around the correct location to have the highest likelihoods. Therefore, our intuitive idea is that the curves in the plot show the results when averaging over the hypercubes in four dimensions with side lengths of one to seven hypotheses around the correct location. This roughly corresponds to averaging over all points influenced by the given number of samples before and after each peak in a correlation vector.

The best accuracies are achieved with 81 or 256 points. Since lower number of points correspond to a higher likelihood threshold to eliminate regions of hypotheses with low maximum likelihood (see Section 5.3.8), we use 81 points in the following, as this will save more computation time.

Note that existing CD methods search for the best point only, which is clearly suboptimal. The median location error with 81 points is 23.5 m, which is almost twice as good as the solution with the best point only, which has a median error of 44.3 m. The standard deviation is 17.3 m with 81 points and 27.6 m with the best point only. This shows that our weighted averaging is a substantial improvement over standard CD, substantially improving accuracy.

5.4.2 Location Averaging over Time

To understand the trade-off between accuracy and the amount of data used, we tested the influence of averaging multiple locations computed from different one millisecond long windows (sliding window average). The results – obtained again from 501 windows – are shown in Figure 5.8. With just a few more milliseconds, we can gain significant accuracy. For instance, with two milliseconds of data, the median localization error drops from 23.5 m to 17.4 m. With 10 ms, it even drops to 9.2 m. And with 30 ms, *all* locations are within 13.9 m.

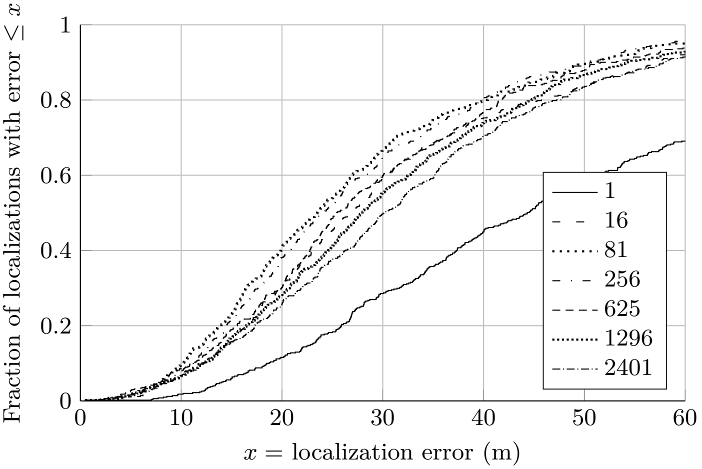


Figure 5.7: Accuracy of our method with different numbers of most likely points used for the weighted averaging. Cumulative distribution functions of localization error (distance to ground truth).

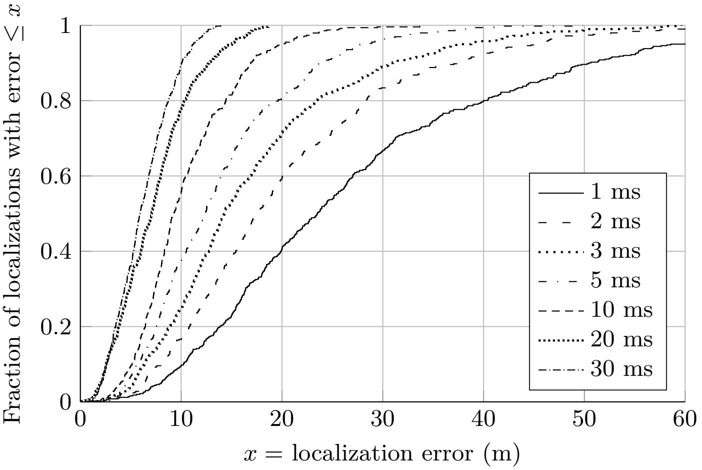


Figure 5.8: Comparison of localization accuracy when averaging over different numbers of consecutive fixes.

5.4.3 Horizontal Localization

To evaluate the accuracy when searching in space only horizontally, we fixed the altitude for the search to that of the ground truth. This emulates scenarios where the receiver is 1) on the Earth surface, so the height can be determined using an Earth elevation model (for example the United States Geological Survey (USGS) elevation model[4]) or 2) the receiver has a barometer, whose measurements can be used together with meteorological data to determine the altitude. The benefit of such an approach is not only better accuracy, as can be seen in Figure 5.9, but the search space is reduced by one dimension, resulting in less hypotheses to test, which translates to faster and less energy consuming processing. For the localization with fixed height, we also first determined the best number of points for weighted averaging with the same procedure as explained in Section 5.4.1, although with numbers to the power of three, because the search space is three-dimensional. The best number of points turned out to be 64. Also for this experiment, the number of one milliseconds windows processed was 501.

The idea of using an Earth elevation model to restrict the possible solutions has also been used by Liu et al. [75]. Because we do not have an implementation of CTN available, we cannot directly compare our results to theirs. However, the box plots in their paper show a median error of approximately 40 m with 2 ms of data used. Our median error when using 2 ms of signal and fixing the height of the solution is 12.1 m. This suggests that our approach is competitive.

5.4.4 Computation Time

To show how the performance of our method using branch and bound depends on the signal conditions, we conducted two experiments capturing both good signal conditions (rooftop) as well as bad signal conditions (inside a multistory university building). We reduce the search space to 10 km × 10 km × 1 km × 4 s for this experiment, to also be able to test the brute force implementation which tests every single hypothesis.

Figure 5.10 shows the cumulative distribution functions in both indoor and outdoor scenarios as described in the last paragraph. It clearly shows that the indoor scenario does not allow for a meaningful localization. For the indoor data, the computation takes 240 s per location, whereas in the outdoor conditions, it takes only 18.6 s. Note that the indoor test presents a worst case scenario in both computation time and localization accuracy.

[4] More information about the USGS elevation model can be found at the “The National Map” website: `http://nationalmap.gov/elevation.html`

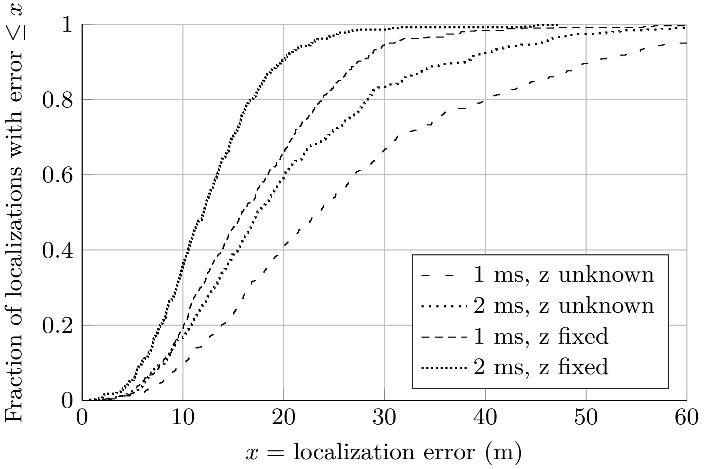


Figure 5.9: Localization accuracy with and without fixed height.

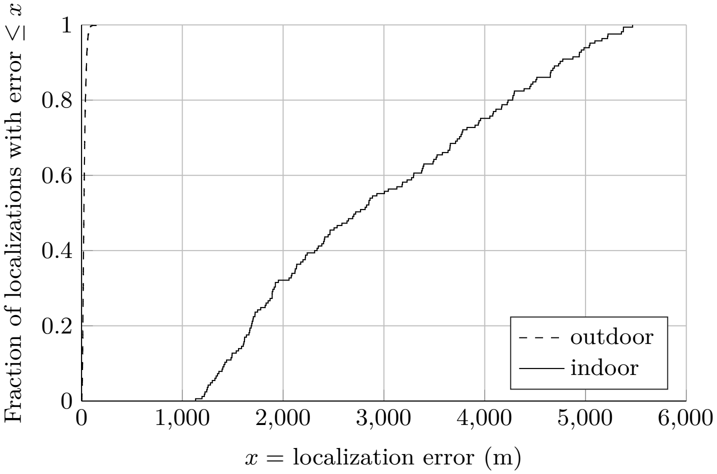


Figure 5.10: Localization accuracy with different signal qualities. Outdoors the solution is much more accurate than indoors.

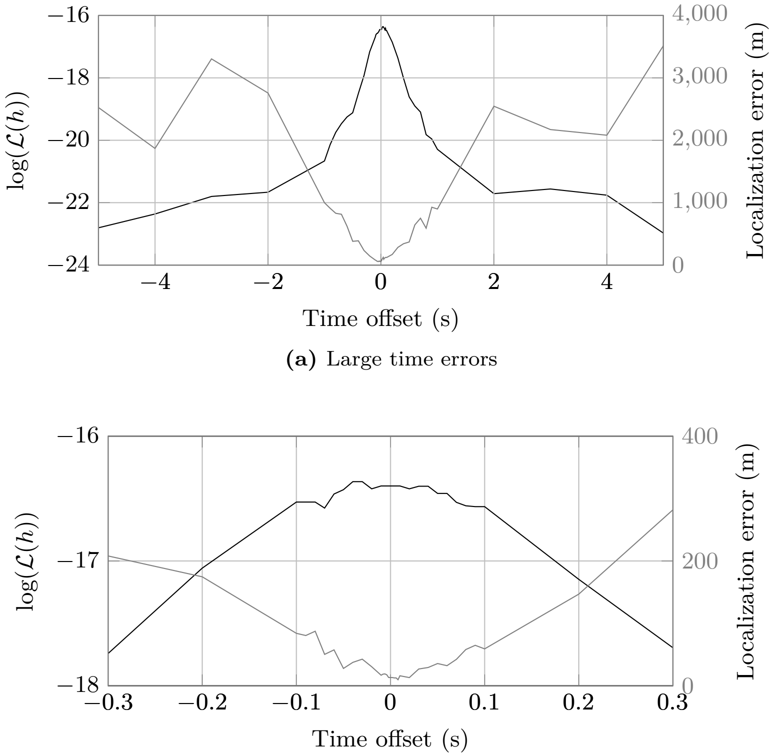
The brute force implementation takes more than two hours to compute the whole likelihood distribution in any scenario.

This means that even in situations that make it difficult to find a fix, we find the most likely location in reasonable time compared to a brute force implementation. For the previous experiments with the larger grid in good signal conditions, our method takes 31 s of computation time.

The performance corresponds to the execution on a current Intel i7 mobile processor with a single thread. The runtimes are not indicative of an optimized implementation of our method, since it could easily be parallelized because the computation of the likelihood is independent for each hypothesis. In all the above experiments about computation time, roughly $2 \cdot 10^4$ hypotheses are evaluated each second. A working CUDA implementation of the brute force method revealed that on a Nvidia GTX 1080, roughly $2 \cdot 10^6$ hypotheses can be evaluated each second which indicates that the search can be sped up 100 times. Therefore, an initial fix can be computed in significantly less than a second under good signal conditions. Note that *tracking* a receiver uses less computation because the search space is smaller.

5.4.5 Time Dependence of the Likelihood Function

To test the influence of the time parameter in our likelihood function, we picked a random one millisecond long window of the sampled signal and searched the location which maximizes the likelihood given different receiver times. The results are shown in Figure 5.11, other ms windows exhibit the same properties as described below. The plot to the left shows that, at least for signals with good quality, our likelihood function (in blue) is roughly convex in the time dimension. However, we cannot reconstruct the correct time precisely because the probability of the best point does not change significantly when the time is within a second of the correct time (blue curve in the right hand side plot). However, the localization quality varies significantly inside this time range (orange curve in the right hand side plot). This is due to the fact that within the search space, there are points which still match the received signal well. Judging from the localization error, the most likely location passes the correct location in a linear fashion as the time error is varied from negative to positive. This suggests that the likelihood function is quite flat in the time domain which is one of the reasons why the averaging over the most likely hypotheses helps to increase the accuracy of our method. As a side note, this last observation could lead one to think that the correct hypothesis can also be found at the center of the likelihood plateau. However, the localization error scales in Figure 5.11 are relatively large, meaning that the center would have to be found accurately. In fact,



(a) Large time errors

(b) Small time errors

Figure 5.11: Shape of $\mathcal{L}(h)$ for different time errors as well as the corresponding localization error. For each time offset, the values of the hypothesis h with maximum $\mathcal{L}(h)$ are indicated.

we experimented with fitting different shapes to the likelihood function, but this yielded results with a large variance.

5.5 Conclusion

We showed how collective detection can be optimized to achieve performance that allows for coarse initial guesses for both location and time. Our branch-and-bound method scales well in both good as well as bad signal conditions. The localization performance is superior to similar approaches due to the averaging which greatly reduces the effect of the flatness of the likelihood function. When utilizing more than one millisecond of signal, the performance is competitive even with classical GPS receivers consuming much more energy.

Our method allows for a snapshot GPS receiver design, which only samples one millisecond of signal per fix and can run on a low power 5 ppm oscillator. Therefore, the energy per fix will be extremely low - since the computation can be done in the cloud - and the receiver can also support small duty cycles, for instance 10 ms per fix once every hour. This gives our method an advantage in terms of energy usage over the classical approach, which samples the signal continuously. Note that such a snapshot receiver does also not need a large and heavy radio time signal antenna like the design presented by Liu et al. [75].

6

Spoof-Proof GPS

> *"Most people stop looking when they find the proverbial needle in the haystack. I would continue looking to see if there were other needles."*
>
> — Albert Einstein

Today, many applications rely on the Global Positioning System (GPS). This makes GPS interesting for attacks which spoof a receiver's perceived location or time. Wrong information in time or space can have severe consequences, as we highlight in the following examples.

Aircraft Navigation Air traffic control is partially transitioning from radar to a scheme in which aircraft transmit their current location twice per second, through so-called *ADS-B* messages. This system is already mandatory for most airliners in Europe and will become so in the US starting in the year 2020 [30, 49]. The aircraft determine their own location using GPS. If a wrong location is estimated by the on-board GPS receiver, for instance due to signal spoofing, this may have fatal consequences. For instance, wrong routing instructions might be given due to a wrong reported aircraft location, leading to aircraft collisions.

Ship Navigation Like aircraft, ships may have little reference points to localize themselves apart from GPS. Trusting a wrong location indication can strand a ship or alter its course. A GPS spoofing incident in 2017, when several ships were placed inland although they were actually on the Black Sea, shows that spoofing attacks against ships already happen in the wild [22].

Car/Truck Navigation Drivers rely more and more on GPS navigation alone rather than orienting themselves. Too often, directions given by car navigation systems are not validated at all and followed blindly. This emerging dependence on GPS is dangerous: Even without spoofers being present, people get stuck in remote places. This may be due to errors in the given directions or simply because of typing errors. In the worst cases, consequences are fatal [88]. Attackers can use this combined weakness of GPS and car drivers to reroute cars and cause a traffic chaos, for instance.

Train Routing and Control Emerging train control systems such as the ETCS may employ GPS localization for each train instead of placing a large number of balises along tracks [92]. Wrong train location estimations could wreak havoc: Collisions between trains might not be anticipated early enough or barriers may not be lowered in time. Also track switches could be triggered while a train is passing through.

Cellular Network Synchronization While the examples above are in the domain of *location* spoofing, an attacker can also try to change the perceived *time* of a GPS receiver. Cellular networks rely on accurate time synchronization for exchanging communication data packets between ground antennas and mobile handsets in the same network cell. Also, *all* neighboring cells of the network need to be time-synchronized for seamless call handoffs of handsets switching cells and for coordinating data transmissions in overlapping coverage areas [6, 58]. Because most cellular ground stations get their timing information from GPS, a signal-spoofing attacker could decouple cells from the common network time. Overlapping cells might then send data at the same time and frequencies, leading to message collisions and losses [6]. Failing communication networks can disrupt emergency services, as people in need of help lose the means of requesting assistance. Also, many businesses relying on mobile phones to coordinate their work with customers, like taxi services and transport companies, could not carry out their work.

Stock Market Synchronization Audit rules mandate that financial markets record trading activities with accurate timestamps [31]. Such timing is often accomplished through GPS receivers set up on the roof of those market places [79]. Accurate timestamps help revealing illegal trading activities, which can sometimes be detected by trading discontinuities, arising for instance when market orders are not executed immediately. Also, with too coarse timestamps, it is possible to observe new market orders and then place own orders "at the same time", so that the latter might be executed before the former orders [77].

Power Grid Synchronization The operation of power grid assets is coordinated with GPS-based precision timing. Also, grid operators use GPS-synchronized observations for disturbance monitoring and fault localization, to maintain grid stability [44]. For many nations, a power grid outage is one of the worst imaginable scenarios. Problems include water pumps that stop working and food and medicines which cannot be delivered due to failing communication.

6.0.1 Spoofing

The threats and weaknesses above show that large damages can be caused by transmitting forged GPS signals. Such signals can nowadays be generated with only a few hundred dollars worth of hardware. While stock market manipulation might cause monetary damages only, GPS signal spoofing attacks targeting air traffic control endanger numerous lives. These threats are well acknowledged. In 2013, a US government study concluded that critical infrastructures rely on GPS, but are not prepared for signal disruptions [48].

A GPS receiver computing its location wrongly or even fail to estimate any location at all can have different causes. Wrong localization solutions come from (i) a low signal-to-noise ratio (SNR) of the signal, for instance when inside a building, below trees or in urban canyons, (ii) reflected signals in multipath scenarios or (iii) deliberately spoofed signals. The first two cases are challenging, but various ideas help mitigating their effects: For low SNR (i), it is possible to use a longer recorded signal in order to increase the total received signal energy. There are some challenges associated with that, for instance phase changes in the signal due to data modulation on top of the carrier signal. Also, the latency of the localization solution increases because the amount of signal used can be on the order of minutes. Multipath signals (ii) can often be discarded by selecting only the strongest signals and those which are consistent in the sense that the localization solution fits well with all the chosen signals. Signal spoofing (iii) is the most difficult case, since an attacker can freely choose the signal power and delays for each satellite individually.

In this work, we not only *detect* spoofing attacks, but also *mitigate* them. We present a robust spoofing mitigation algorithm based on the *collective detection* maximum likelihood localization approach. Our method can differentiate closer distances between correct and spoofed locations than previously known approaches. While information about the internals of commercial receivers is scarce, to the best of our knowledge, consumer products currently have at most simple spoofing mitigation integrated [113]. Military receivers use symmetrically encrypted GPS signals which are not available to the public. Like this, the signals are unknown to attackers in advance. Still, an attacker could replay even these encrypted signals with a small delay to confuse receivers. Academically, some anti-spoofing methods have been studied, but the spatial resolution of those methods is hundreds of meters, which means that attacks spoofing a closer location cannot be detected. Details are given in the next section. Our method achieves median errors under 19 m on the TEXBAT dataset, which is the de facto reference dataset for testing GPS anti-spoofing algorithms [104, 130].

Apart from spoofing fake signals, an attacker can simply *jam* the frequency band of the GPS signals with strong random signals, increasing the noise level at receivers. Jamming is the least sophisticated kind of attack and has a result equivalent to Challenge (i) above: a low SNR at the receiver. Therefore, using longer signal recordings also helps against jamming. Apart from taking measures against special types of jamming attacks, like using directional antennas to exclude ground-based jammers, one cannot do much against jamming [100]. Like jammers, spoofers sending strong signals can be detected by measuring the received signal power and also decrease the SNR. And weak spoofing signals do not have much of an influence on a receiver's location estimation, as the stronger authentic signals can be detected without any problem. We focus on the toughest type of GPS spoofing attack which consists of spoofed signals with power levels similar to the authentic signals.

A specialty of our method is that it uses only a few milliseconds worth of raw GPS signals, so-called *snapshots*, for each location fix. This enables offloading the computation into the cloud, which allows combining knowledge of observed attacks. Measurements from enough receivers may even permit finding spoofers' locations. Cloud offloading also makes our technique suitable for energy-constrained sensors. Existing spoofing mitigation methods require a constant stream of the GPS signals and track those signals over time. Generally, spoofing mitigation is computationally more demanding than normal localization, since fake signals have to be detected, removed or different solutions compared. Therefore, spoofing mitigation is even a computational challenge on smartphones, which nowadays have reasonably large batteries.

6.1 Related Work

Three tracks of research are most relevant to our work, maximum likelihood GPS localization, GPS spoofing mitigation algorithms and successive signal interference cancellation.

6.1.1 Maximum Likelihood Localization

Our work is based on *collective detection (CD)*, which is a maximum likelihood GPS localization technique. Maximum likelihood GPS localization was already proposed in 1996 [118], but was computationally infeasible at that time. Collective detection has first been implemented by Axelrad et al. in 2011 [11]. Due to search spaces containing millions or more location hypotheses that have to be searched through, subsequent work focused on improving the computational burden through various heuristics [26,60]. Recently, a branch-and-bound algorithm has been proposed that finds the optimal solution within some ten seconds running on a single CPU thread [14]. Our method is an adaptation of this branch-and-bound algorithm to mitigate GPS signal spoofing attacks. Another maximum likelihood approach by Closas et al. models the signal observations as a function of the receiver state [28]. Due to a high-dimensional and non-linear cost function, it remains unclear how the optimal receiver location can efficiently be computed in that framework.

6.1.2 Spoofing Mitigation

GPS spoofing defenses have intensively been studied. However, while most research focuses on *detecting* spoofing attacks, there is a lack of ideas for spoofing *mitigation* and *recovering* from successful attacks by finding and authenticating the correct signals [101]. Our work helps in this area, as the technique presented in this chapter inherently mitigates spoofing attacks.

A lot of research focuses on tracking multiple signals per satellite instead of at most one [23, 104]. This is a useful approach for *detecting* spoofing attacks. However, given multiple signals per satellite, it is a challenge to select the correct signal from each satellite. Another method for detecting spoofing attacks is hypothesis testing [136].

Whether sophisticated spoofing attacks are practical is subject to debate [113]. Still, spoofing hardware performing a relatively sophisticated *seamless satellite-lock takeover attack* has already been built, although it has only been tested in a lab environment [53]. Challenges associated with spoofing are for instance matching the spoofed and authentic signals' amplitudes at the receiver, which might not be in line of sight and moving [110]. Despite that, it is even practically feasible for a spoofer to erase the authentic signals with signals at a 180° phase offset [101]. This is one of the strongest attacks and can only be detected with multiple receiver antennas or by a moving receiver [101]. Thus, a cooperative victim, like a convicted criminal with an ankle monitor, could use this technique to deceive authorities [101, 110]. For signal erasure to be feasible, the spoofer needs to know the receiver location more accurately than the GPS L1 wavelength, which is 19 cm. Receivers with only a single antenna cannot withstand such an erasure attack. Our method targets single-antenna receivers and we therefore do not deal with signal erasure. In basically all other types of spoofing attacks (cf. Section 6.3), including signal replay and even spoofers with multiple transmission antennas, the original signals are still present and our algorithm remains robust.

Due to the limitations of receivers with a single antenna, some research focuses on receivers with multiple antennas or even multiple receivers combining their information [76]. Coordinated spoofing attacks with multiple antennas can circumvent *some* defenses using multiple receiver antennas like detecting signal timing inconsistencies [124]. Also, size requirements and a high price sensitivity for consumer GPS receivers make multi-antenna receivers impractical for many applications. Single-antenna receivers cannot differentiate between spoofing signals sent from one or more locations. Our algorithm is aimed at those single-antenna receivers and is therefore indifferent to multi-antenna attackers.

One approach against erasing spoofers with a single transmitting antenna focuses on moving receivers [19]. Signals are classified into spoofed and non-spoofed signals by moving the receiver around and observing the spatial correlation of signals sent from a single source. The method does not cover stationary applications like the introductory time synchronization examples and time periods during which a mobile receiver is not moving.

The GPS anti-spoofing work most relevant to this chapter is that based on joint processing of satellite signals and maximum likelihood localization. One method is able to *mitigate* a limited number of spoofed signals by vector tracking of all satellite signals [59]. A similar technique is shown to be relatively robust against jamming and signal replay [93]. Another idea is to combine all satellite signals in a Bayesian estimation algorithm [66]. Compared to our snapshot receiver, this technique uses a continuous stream of received signals for the sequential parameter estimation. Extensions of aforesaid maximum likelihood method by Closas [28] for countering spoofing have also been proposed. One assumes a spoofer which sends unsynchronized spoofing signals that do not consistently point to a spoofed location [128] and the other tries to solve the global convergence problem with an initial grid search and subsequent iterative refinement [133]. Our method can tolerate consistent spoofing signals, even in case the spoofing signal is already present when the receiver starts.

We could not find any anti-spoofing methods for GPS snapshot receivers. Since our method yields robust location fixes from signal snapshots, there is no need for recovery like in classical receivers. The latter may lock onto spoofed signals without noticing a drift from the authentic satellite signals over time.

Interestingly, in contrast to the vast research on GPS spoofing, there is a lack of commercial, civil receivers with anti-spoofing capabilities.

6.1.3 Successive Interference Cancellation

Our iterative signal dampening technique to deal with spoofing signals is similar to successive interference cancellation (SIC). SIC removes the strongest received signals one by one in order to find weaker signals and has been used with GPS signals before [80, 82]. That work is based on a classical receiver architecture which only keeps a signal's timing, amplitude and phase. Our receiver is based on CD, which directly operates in the localization domain and does not identify individual signals in an intermediate stage. As it is impossible to differentiate between authentic and spoofed signals a priori, we do not remove signals from the received sampled data. Otherwise, the localization algorithm might lose the information from authentic signals. Instead, we dampen strong signals in order to reveal weaker signals. This can reveal localization solutions with lower CD likelihood.

6.2 GPS Localization

The Global Positioning System (GPS) is a Global Navigation Satellite System (GNSS) operated by the United States Air Force. It provides location and time information to receivers anywhere on Earth where signals from at least four satellites can be received. The GPS satellites are located in a non-stationary medium Earth orbit and circle the Earth about twice a day.

GPS satellites transmit multiple signals in different frequency bands. Some of the signals are encrypted and reserved for military use. We focus on the signal most commonly used in civilian receivers, which is located in the L1 frequency band at 1.57542 GHz. To distinguish the satellites, *code division multiple access (CDMA)* is used. The employed Gold codes, one for each satellite, with 1023 bits length, achieve good correlation and cross-correlation properties [91]. Those signals are also called *pseudo-random* *noise (PRN)* sequences due to their noise-like nature. Sent with a data rate of 1.023 MHz, the codes repeat every millisecond. The satellites further transmit navigation data. The navigation data contains satellite orbit information, called *ephemeris*, and transmission timestamps, which allow calculating the exact location of the satellites at the time of signal transmission. The data is modulo-2 added to the Gold codes at a rate of 50 bit/s. Hence, each data bit is transmitted through 20 subsequent Gold codes. The generated signal is sent with *binary phase shift keying (BPSK)* on the L1 frequency band.

6.2.1 Localization

For localization, GPS receivers measure times of flight of received satellite signals. Using an orbit model whose parameters are received with the navigation messages of the previous step, the location of the satellites at the time of signal transmission is determined. Unlike the satellites, the receiver does not carry an atomic clock and is thus not synchronized with the satellites. Therefore, the localization problem has four unknowns, namely three spatial coordinates and the receiver's time offset from the GPS system time. The classical way of computing a solution to the localization problem, a so-called *fix*, consists of setting up a system of equations from the measured satellite distances and solving it in a least-squares sense.

Classical GPS receivers consist of three stages, acquisition of the satellite signals, decoding of the satellite data and finally, calculation of a location solution based on the received data.

The acquisition finds the visible satellites and detects the code phase of the Gold codes and the Doppler shifts of the signals. A strong correlation marks the code phase of the Gold code for a given satellite. An example can be seen in Figure 6.1. The code phase is determined by the time of flight of the signal between the satellite and the receiver and therefore by their distance. The relative speed between satellite and receiver introduces a significant Doppler shift to the carrier frequency. This Doppler shift has to be found during acquisition to allow decoding of the signal.

Classical receivers use the information gathered during acquisition and start using a feedback loop to track the satellite signals to decode the contained navigation message. After a receiver obtains that information from at least four satellites, the receiver can compute its location.

6.2.2 Snapshot Receivers

Assisted GPS (A-GPS) addresses a weakness of the basic GPS system: Due to a limited data rate, arising from the large satellite distance and therefore

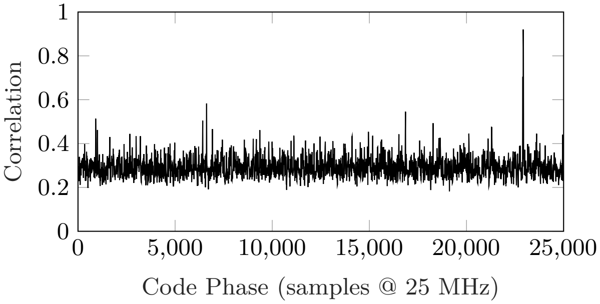


Figure 6.1: Acquisition result for a satellite signal with average SNR. The correlation peak indicates the signal's receive time. The length of the correlation vector with 25,000 samples corresponds to 1 ms of signal.

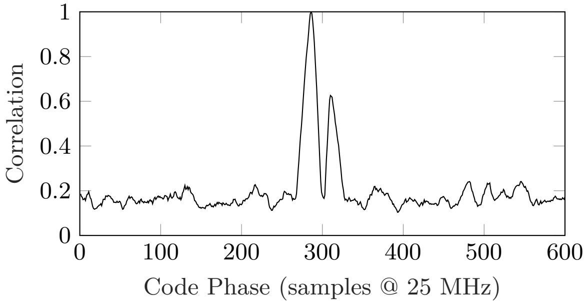


Figure 6.2: Acquisition result for a satellite signal with good SNR but two matching signals. The two peaks are 24 samples apart, which corresponds to a measured distance difference of 288 m. Two possible interpretations are that the first signal is the authentic signal and the second is a signal reflection (multipath) or that one of the signals is spoofed. Only the relevant part of the acquisition vector is shown. The full vector is 25,000 samples or 1 ms long.

weak received signal power, satellite orbit parameters are only transmitted every 30 seconds. Thus, the latency of a first fix after turning on a receiver, the so-called *time to first fix (TTFF)*, can be high. With A-GPS instead, these orbit parameters are fetched over the Internet, for instance via a cellular network, which reduces the data transmission time, and thus the TTFF, drastically [127].

While the satellite orbit parameters are usually valid for two hours, classical receivers also need to receive a current timestamp. Timestamps are transmitted from satellites every six seconds. Threrefore, receiving timestamps still causes a relatively high latency and high energy consumption in GPS receivers. Snapshot GPS receiver techniques such as *Coarse-Time Navigation (CTN)* or *collective detection (CD)* allow computing the receiver location even if no timestamp is received. GPS signals repeat every millisecond and the signals propagate 300 km during that time. Therefore, only the remainder of the satellite distances modulo 300 km can be measured without receiving a timestamp. If the initial estimate of the receiver's location and time is equivalent to less than 150 km, the measurement's full-millisecond ambiguity vanishes. For this purpose, an offset of one second is approximately equivalent to an error of 1 km since the satellites' relative speed to an observer on the Earth surface is about 1 km/s. With such an approximate initial receiver state, one can estimate the satellite locations and signal times of flight and the localization can be executed [127]. With longer code periods, such as Galileo's 4 ms long signals, the receiver state estimate's required accuracy can be relaxed proportionately.

With this insight, snapshot receivers are able to compute a location fix from as little as one millisecond of data if the signal quality is good. However, the influence of noise is often too large to make localization viable from one millisecond of signal only. Combining several milliseconds of signal is more robust [75]. Due to only a few milliseconds activation to receive enough signal power for a fix, snapshot receivers use low power. Therefore, snapshot receivers are suitable for multi-year tracking of battery-powered or even energy-harvesting sensors [40]. In comparison, classical GPS receivers drain a smartphone battery in a few hours. Therefore, it can be expected that snapshot receiver will be deployed extensively in the future. However, snapshot receivers cannot be protected by existing GPS anti-spoofing methods that track signals over time. Our present work is designed for signal snapshots and therefore helps protecting snapshot GPS receivers.

6.2.3 Collective Detection

In recent years, maximum likelihood (ML) localization methods have been proposed, promising more robust localization solutions. That is, ML methods are more tolerant to low SNR, multipath effects and spoofing than the classical least-squares localization methods. Since the arrival time of a satellite signal cannot always be determined with certainty, a wrong signal time of flight might be estimated. This renders the system of equations unsolvable. The information from the rest of the satellites could still be enough to compute the location fix, but eliminating "bad" measurements is not always easy. Maximum likelihood methods [28] and in particular collective detection (CD) [11, 14, 26, 60] do not pick an arrival time for each satellite signal, but rather combine all the available information and take a decision only at the end of the computation. This uses more computation power, but is less prone to errors than solving a system of equations in the least-squares sense like in CTN and classical GPS localization.

Since the GPS localization scheme is based on satellite signal time of flight measurements, the main challenge is determining the signal arrival times despite low received signal power. In the methods presented so far, the arrival times are detected based on the amplitude in the correlation with the corresponding satellite's PRN. This requires the presence of a clear peak in the correlation vector. With bad signal conditions, for instance under a tree, in an urban canyon or even indoors, there may be several or no such correlation peaks. The problem is particularly pronounced when only a few milliseconds of signal are used as in CTN, because the received signal power is less than with multiple seconds of signal.

To mitigate this problem, CD does not only "accumulate" the captured signal over time, but also over all available satellites. Combined, the signal energy of multiple satellites gives a higher chance to detect the signal arrival times correctly. The gain in the signal-to-noise ratio (SNR) of CD compared to CTN means that CD is more robust to noise. Therefore, CD is suited for bad signal conditions such as in spoofing scenarios.

Our method in this paper is based on an efficient implementation of CD [14]. A given four-dimensional (location and time) search space is discretized as a regular grid of solution hypotheses. The expected distance between satellite and receiver and therefore the expected code phase of the received signal is calculated for each grid point. The satellite acquisition results are aligned by the expected code phase and a pseudo-likelihood of the point is calculated. By searching over all possible solutions in the grid, the algorithm is guaranteed to deliver the most likely location given the observed signals. A branch-and-bound implementation delivers the same result with reduced computational effort.

6.3 GPS Signal Attacks

The easiest way to prevent a receiver from finding a GPS location is jamming the GPS frequency band. GPS signals are weak and require sophisticated processing to be found. Satellite signal jamming considerably worsens the signal-to-noise ratio (SNR) of the satellite signal acquisition results. CD algorithms achieve a better SNR than classical receivers and are thus able to tolerate more noise or stronger jamming [11].

A jammed receiver is also less likely to detect spoofing, since the original signals cannot be found any more. The receiver tries to acquire any satellite signals it can find. Thus, the attacker only needs to send a set of valid GPS satellite signals stronger than the noise floor, without any synchronization with the authentic signals.

As jamming is detectable by observing the noise floor, in-band power level and loss of satellite signal lock, a more subtle attack may be performed. The spoofer can send the set of satellite signals with adjusted power levels and synchronized to the authentic signals to successfully spoof the receiver.

Seamless Satellite-Lock Takeover The most powerful attack is a seamless satellite-lock takeover. In such an attack, the original and counterfeit signals are nearly identical with respect to the satellite code, navigation data, code phase, transmission frequency and received power. This requires the attacker to know the location of the spoofed device precisely, so that time of flight and power losses over distance can be factored in. After matching the spoofed signals with the authentic ones, the spoofer can send its own signals with a small power advantage to trick the receiver into tracking those instead of the authentic signals. A classical receiver without spoofing countermeasures, like tracking multiple peaks, is unable to mitigate or detect this attack, as there is no indicative interruption of the receiver's signal tracking.

Navigation Data Modification An attacker basically has two attack vectors: modifying the signal's code phase or altering the navigation data. Misaligning the code phase leads to changes in the signal arrival time measurements, which results in different localization results. And by changing the navigation data, the attacker displaces the perceived satellite locations and therefore also influences the calculated receiver location. In comparison to classical receivers, assisted or snapshot GPS receivers like CTN and CD are not vulnerable to navigation data changes in the satellite signals as they fetch that information from other sources like the Internet. An attacker could tamper with such data sources, but this shall not be our concern in this chapter. Rather, we deal with modified, wireless GPS signals.

6.4 Algorithm Design

Our method is aimed at single-antenna receivers. Therefore, we do deal with signal erasure attacks (cf. Section 6.1.2). Instead, given a combination of authentic and spoofed signals observed at a receiver, our goal is to identify all likely localization solutions. Based on external knowledge, the receiver can then decide which of these solutions must be the correct one. For instance, using sensor data from an accelerometer, a motion model can be matched with the sequences of likely localization solutions. Or only smooth receiver paths can be accepted, based on the receiver's maximum de- and acceleration. Further, the location hypotheses can be reconciled with a map, for instance eliminating all locations not on a road.

Our GPS spoofing mitigation algorithm is based on collective detection (CD). CD is a good choice for several reasons: 1) CD has improved noise tolerance compared to classical receivers, 2) CD is not susceptible to navigation data modifications, 3) CD is suitable for snapshot receivers, and 4) CD computes a location likelihood distribution which can reveal all likely receiver locations including the actual location, independent of the number of spoofed and multipath signals. Related to the last point, spoofing and multipath signals are actually similar from a receiver's perspective. Both result in several observed signals from the same satellite. The difference is that multipath signals have a delay dependent on the environment while spoofing signals can be crafted to yield a consistent localization solution at the receiver. In order to detect spoofing and multipath signals, classical GPS receivers can be modified to track an arbitrary number of signals per satellite, instead of only one [104]. In such a receiver, the set of authentic signals—one signal from each satellite—would have to be correctly identified. Any selection of signals can be checked for consistency by verifying that the resulting residual error of the localization algorithm is small enough. Consistent solutions are either the actual receiver location or a spoofed one. However, already finding sets of signals which are consistent for one receiver location, is combinatorially difficult. For n satellites and m transmitted sets of spoofed signals, there are $(m+1)^n$ possibilities for the receiver to select a set of signals. Only $m+1$ of those will result in a consistent localization solution, namely the actual location and m spoofed locations. Even if running a least-squares optimization for each signal combination may be feasible, in practice one additionally needs to identify and exclude multipath signals, which further enlarges the search. Therefore, tracking multiple signals per satellite helps detecting spoofing and multipath events by raising a warning if multiple signals per satellite are received, but it is impractical to mitigate spoofing. CD avoids this signal selection problem by joining and transforming all signals into a location likelihood distribution.

CD only shows consistent signals, since just few signals “overlapping” for some location hypothesis do not accumulate a significant likelihood. In the CD likelihood measure, all plausible receiver locations—given the observed signals—have high likelihoods. However, finding those likely locations efficiently is challenging. The basic version of CD computes a likelihood for *all* localization hypotheses in a given bounded and discretized search region. Since the search is usually performed in four dimensions, space and time, the basic CD is computationally expensive. We discuss computation performance subsequently, in Subsection 6.4.1.

First, let us investigate how spoofing mitigation is possible based on the basic CD algorithm. Among all location hypotheses, the basic CD algorithm simply selects the most likely location. However, this may be a spoofed location, so it is necessary to also consider less likely locations to be sure that the true receiver location is included in the results. A straightforward idea is to select all points with a likelihood above some threshold. To understand why this does not work well, we have to dig into the definition of the likelihood measure. The (pseudo-)likelihood of a point is computed by shifting and adding signal correlation vectors and selecting the maximum value of the resulting summed vector as the likelihood. For close points, the correlation vectors being added are shifted by only a few entries. There are two reasons, why such small shifts result in only marginally lower likelihoods: 1) the correlation peaks form triangular shapes due to usual oversampling of the received signal, and 2) small timing estimation errors between signals from different satellites may misalign the correlation vectors by a few entries. Therefore, locations close to local maxima all have high likelihoods. Thus, there will be a large number of points above some threshold likelihood, clustered around points with local likelihood maxima. Pursuing this insight, we would like to pick only such local maxima as potential localization solutions. This can for instance be done with some clustering or by simply excluding points in some radius around a local maximum. We also strive to find local likelihood maxima, but at the same time improve the runtime performance of the algorithm. We see next that our proposed, faster branch-and-bound approach iteratively searches such local maxima.

6.4.1 Branch and Bound

To reduce the computational load compared to exhaustively enumerating all the location hypotheses in the search space, like the basic CD does, we rely on a fast CD algorithm leveraging branch and bound [14]. Branch and bound CD does not compute the whole location likelihood distribution. Instead, it finds the most likely location orders of magnitude faster. However, the most likely point may be the spoofer-induced localization solution, not the actual receiver location. Therefore, as discussed above, we want to find *all* locations with a likelihood above some threshold and being a local maximum, since we assume that the receiver also observes the authentic signals. To achieve this, we run the branch and bound algorithm repeatedly. Our idea is to find the next likely location in each iteration. The most likely location must be formed by some high peaks in the individual satellite signal correlation vectors. In order to remove the most likely location, one could therefore try removing the highest correlation peak for each satellite before proceeding to the next iteration. This is similar to the classical receivers tracking multiple peaks, as described above. The classical receivers select some number of highest peaks, which is equivalent to iteratively removing the highest peak before each subsequent peak selection [104]. However, it need not be *the*, but only *some*, of the highest correlation peaks forming the most likely locations. For instance, for some satellites, the highest peak may result from an authentic signal, while for other satellites the highest peak may be from a spoofed signal. This might for instance be the case when the attacker sends the spoofed signals with different power levels in order to thwart our strategy. In such a case, it can happen that the most likely location is a spoofed location or a location getting a high likelihood as a result of a combination of authentic and spoofed signals. If the highest peaks are removed, also some authentic signals are removed and the actual receiver location may not be found in any later algorithm iteration. In essence, this is the same problem that classical receivers face: If multiple peaks per satellite are present, it is unclear a priori, which peaks belong together, that is, yield a consistent localization solution. Instead of completely removing the highest correlation peaks for each satellite, we exploit the advantage of CD that we do not need to take a hard decision whether the strongest acquisition peak is authentic or not. Instead, in every algorithm iteration, we attenuate the strongest peak of each satellite by some factor. Like this, that peak has less influence on the next iteration, but it can still aggregate with signals from other satellites. For instance, if the peak is formed by an authentic signal, it can still reinforce the likelihood of a somewhat weak correct localization solution. Also, not completely removing signals prevents a collapse of the problem in the sense that the solution becomes underdetermined due to too little signals being available. Generally, signals from at least four satellites are needed to resolve the location and time of the receiver. (For simplicity, we just write *location* in this chapter, but actually mean *location and time*.) In the end, the dampening of peaks emulates the selection of local maxima in the complete location likelihood distribution, as outlined above. By dampening the strongest peaks iteratively, the strongest local maxima are eventually dampened as well, letting other local maxima stand out and be found in subsequent iterations. This iterative dampening process is repeated until the likelihood of the found, most likely local maximum has a likelihood below some threshold. For example, this threshold can be selected as a multiple of the noise floor. The "noise floor" can be determined as the median or average likelihood of random locations.

6.5 Implementation

Our implementation follows a branch-and-bound algorithm for collective detection (CD) [14] with our modifications to find several likely points in iterations, as described in the previous section. In this section, we describe the settings and optimizations of the algorithm that we used to obtain the results presented hereinafter.

Generally, the employed data types should be carefully selected. While calculations like the computation of the signal time of flight require double precision floating point numbers, smaller data types are used wherever possible for best performance and memory consumption. For instance, due to the large search space of the algorithm, 16-bit indices for the localization solution hypotheses grid enable more hypotheses to be cached, yielding faster computation speeds. Next, we account for atmospheric delays of the received signals to improve the localization accuracy. Further, though not strictly necessary for the function of the algorithm, we found that more consistent results can be achieved when scaling raw input data and satellite signal acquisition results to the value range $[-1, 1]$.

6.5.1 Acquisition

Through testing with low SNR recordings, we found that an optimized satellite signal acquisition implementation can improve the results significantly. For the Doppler shift search, a bin width of 500 to 667 Hz suffices for most applications [47]. Since the runtime impact of the acquisition is negligible in our implementation, we reduced our frequency search bin width to a finer resolution of 200 Hz. The correlation length is the most important factor in tuning the acquisition. At least one millisecond is required to fit a whole code period. Longer correlations can significantly increase the SNR. Unfortunately, we cannot choose an arbitrarily long correlation: Every 20 ms a navigation bit flip can happen, which reduces the correlation, yielding worse results than with shorter a correlation. To make sure that at least one millisecond of data without a bit flip is available for each satellite, CTN receivers capture at least two consecutive milliseconds of signals [75]. In our experiments (cf. Sec. 6.6), a correlation length of 3 milliseconds often provides the best results.

The resulting correlation vectors are combined to form one vector with one millisecond length. We add the vectors within one frequency bin pointwise to form a one millisecond long vector and then combine all frequency bins by selecting the maximum value at each array location. We found that this yields consistent results with good SNR.

6.5.2 Receiver Implementation

The most significant change to the receiver is the elimination or dampening of the strongest satellite signals prior to running subsequent iterations. Typically, the satellite signal is reconstructed and then subtracted from the raw signal before running a new acquisition. Because the datasets we primarily used for testing offered good signal quality and clearly visible signals, we opted for removing or dampening the signals directly in the acquisition vectors.

In the acquisition stage of a GPS receiver, the received signal is correlated with C/A codes. The highest correlation is (theoretically) achieved when the C/A code in the received signal is aligned with the reference C/A code. Due to the pseudo-random nature of the C/A codes, a shift larger than one code chip from the correct location result in a low correlation value. Since one C/A code chip has a duration of 1/1023 ms, the the width of the peaks found in the acquisition vector is less than 2 ‰ of the total vector size. We reduce the maximum peak by 60 % in each vector. A detection for partially overlapping peaks prevents changes to those peaks. While it is possible to remove the signals completely, this has a negative impact on the resulting localization accuracy. For example during a seamless satellite-lock takeover, it is impossible to detect two separate signals, which means that both signals would be removed at once. Only reducing the signal level instead has no or little negative impact on accuracy in general, while it improves accuracy in such cases.

Before using these vectors for the next iteration of the algorithm, the acquisition result vectors are normalized again. For improved computation performance, subsequent iterations can be run with a reduced search space based on the results of the first iteration.

6.6 Experiments

In 2012, Humphreys et al. [52] presented *TEXBAT*, the first public database of scenarios with spoofing attacks. So far, it has been the de facto standard for any GPS spoofing research. TEXBAT contains a total of 8 different spoofing scenario recordings and two “clean” recordings without any spoofing. All scenarios are constructed based on the clean recordings. The first

Table 6.3: Median, average and maximum errors and the variance of the localization solutions to all TEXBAT scenarios computed with our algorithm in two iterations. Units are in meters and for the variance in m^2. Location-push scenarios are marked in bold.

Scenario	med 1	avg 1	max 1	var 1	med 2	avg 2	max 2	var 2
1	4	5	32	14	10	13	182	165
2	3	3	9	3	3	4	25	12
3	5	8	78	104	19	51	552	5135
4	**9**	**23**	**214**	**911**	**155**	**190**	**572**	**28443**
5	7	21	348	1659	7	14	156	548
6	**405**	**385**	**559**	**21142**	**19**	**34**	**222**	**1315**
7	3	4	10	4	15	46	438	5138
8	4	5	45	30	19	63	630	10596

clean recording is in a stationary setting with an antenna placed on top of the university building. The second clean recording is a dynamic recording from an antenna mounted on a car driving across the city. The spoofing scenarios are produced by replaying one of the clean datasets and adding counterfeit signals from a signal generator. Those combined signals are recorded using signal capture hardware. The counterfeit signals are generated with appropriate characteristics to be as representative as possible for all currently known attack techniques.

Scenarios 1 to 4, as well as Scenarios 7 and 8, are derived from the static dataset, while Scenarios 5 and 6 are derived from the dynamic dataset. Scenarios 5 and 6 are most difficult, as no ground truth is available and environmental effects like multipath signals can affect the recording. Such effects could modify authentic signals in a way that they might be mistaken for spoofed signals. Another difference between the scenarios is the dimension in which the spoofing attack is happening. While in Scenarios 4 and 6 a *location* error of approximately 600 meters to the north is introduced, all other scenarios introduce a *time* error of approximately 2 microseconds.

For our experiments, we extract snapshots from the TEXBAT scenarios. For every second of a recording, five windows of 9 ms length are extracted and the localization results are averaged over those five windows. So, each localization uses a total of 45 ms of signal data. Average, median, and maximum errors as well as the variance of the location estimates, compared to the respective clean scenarios, are summarized in Table 6.3. As the TEXBAT dataset contains at most one set of authentic satellite signals and at most one set of spoofed signals, we show the results of two algorithm iterations. The first iteration is equivalent to a run of the basic branch-and-bound CD algorithm. The second iteration uses the modified signals with the dampened high-power signal components. So, since at most two sets of signals are present, either our first or second algorithm iteration should find the correct receiver location. Which iteration that is, depends on the relative signal power of the authentic versus the spoofed signals. The first iteration finds the location pertaining to the stronger set of signals.

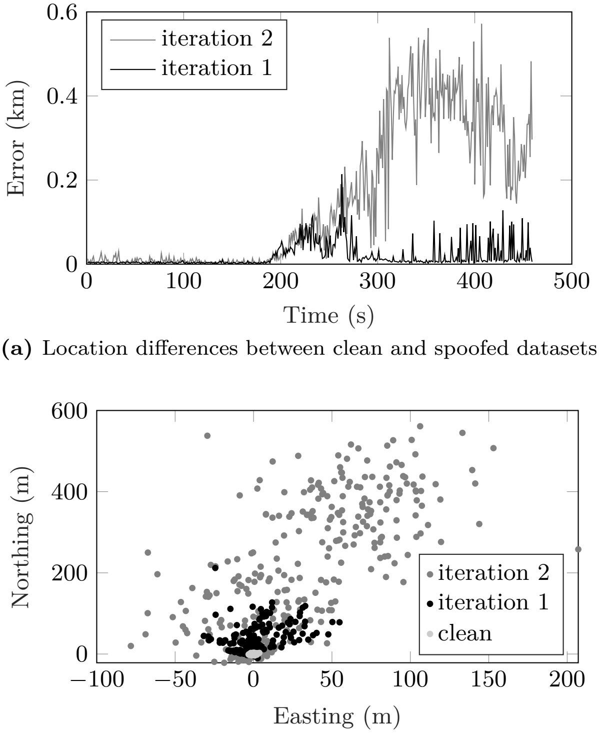
6.6.1 TEXBAT Time-Push Scenarios

Scenario 1 contains a *switch* attack in which the original signals are switched for counterfeit signals. In this scenario, while it might be possible to detect whether spoofing is happening or not by analyzing the raw data, it is impossible to recover the original signals as they are not present once the spoofing starts. Scenario 2 contains an *overpowered* attack in which the adversary adds the spoofing signals with a 10 dB power advantage over the authentic signals. Scenarios 5 is similar to Scenario 2, but based on the dynamic dataset. While in Scenario 2, the spoofing can be detected by the in-band power increase, Scenario 3 represents a case in which the spoofer attempts to match the authentic signals' power. *Scenario 3* contains a *matched-power* attack in which the adversary signals have 1.3 dB power advantage. The spoofer also locks the spoofed signals at some fixed phase angle to the authentic signals, which makes detection more difficult. Scenario 7 contains a *matched-power* attack comparable to Scenario 3, but aligns the carrier phase between the spoofing and authentic signals. Scenario 8 is identical to Scenario 7 except that received navigation data is treated as an unpredictable low-rate security code that is guessed by the spoofer with a delay of some tens of microseconds.

The time push scenarios from the TEXBAT dataset are of limited use for testing our algorithm. The time resolution of our algorithm for the TEXBAT dataset is around 10 ms and therefore does not detect the induced time errors of 2 ms. We can however see that the algorithm produces stable results and finds the correct location with high accuracy. The average and median errors for the static time-push scenarios stay below 7.6 m, which is even better than the results achieved with the branch-and-bound algorithm for benign scenarios [14].

The second iteration of the algorithm produces worse results and especially the maximum error increases significantly. This is probably due to the decreased signal-to-noise ratio in the second iteration.

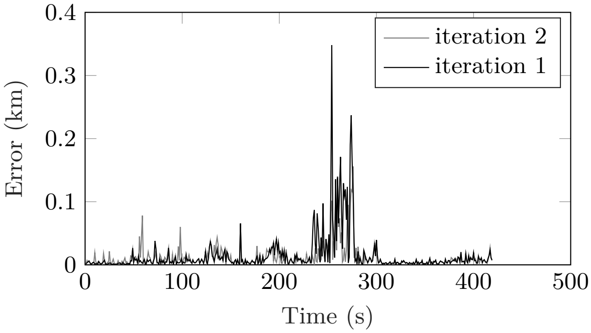
An interesting outlier is Scenario 5, in which the second iteration's result is better. As can be seen in Figure 6.5b, both iterations show almost identical results apart from one section at the beginning of the last third of the track. In the clean dataset it seems like the car has stopped multiple times to wait for example at a red light in traffic. One possible explanation is the



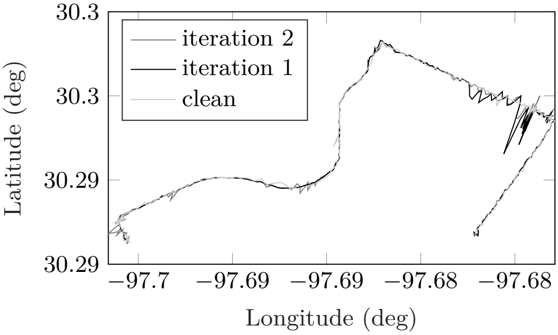
(a) Location differences between clean and spoofed datasets.

(b) Computed ground tracks.

Figure 6.4: Results for the TEXBAT Scenario 4. *Static matched-power location-push* scenario with 0.4 dB spoofing power advantage. At 200 to 300 s, the takeover attack has a negative impact on the accuracy. Afterwards, the first iteration finds the authentic location with little error. In this scenario, the second iteration tracks the spoofed location as the spoofer only has a temporary power advantage.

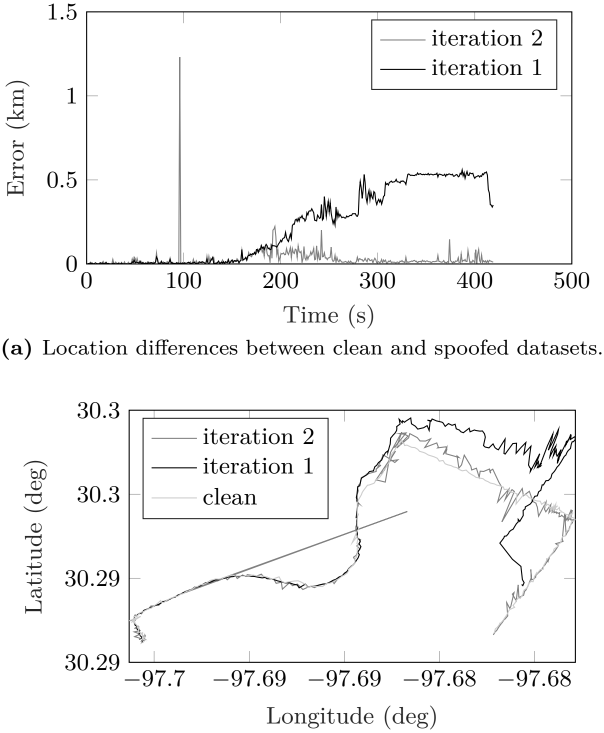


(a) Location differences between clean and spoofed datasets.



(b) Computed ground tracks.

Figure 6.5: Results for the TEXBAT Scenario 5. *Dynamic overpowered time-push* scenario similar to Scenario 2. The moving receiver makes defense more difficult as signals can be misinterpreted as environmental influences and multipath signals. The higher error at around 250 to 300 seconds likely comes from a traffic light stop where noise is induced by nearby cars. The second iteration shows a limited induced error.



(a) Location differences between clean and spoofed datasets.

(b) Computed ground tracks.

Figure 6.6: Results for the TEXBAT Scenario 6. *Dynamic matched-power location-push* scenario comparable to Scenario 4 but with 0.8 dB spoofing power advantage and based on the dynamic dataset. Iteration 1 tracks the spoofed location while iteration 2 tracks the authentic location. The high error at 99 seconds is likely a data artifact from the TEXBAT dataset and vanishes with other correlation lengths.

presence of multipath signals or environmental impacts which influence the signal.

6.6.2 TEXBAT Location-Push Scenarios

As collective detection algorithm such as ours are indifferent to small time offsets, the location-push scenarios are more interesting. Only two TEXBAT scenarios contain location spoofing. Scenario 4 contains a *matched-power* attack with a spoofing power advantage of 0.4 dB and frequency locking of the spoofed to the authentic signals. Scenario 6 is similar, but based on the dynamic dataset. For the first 100 s in case of the static datasets and 180 s in case of the dynamic datasets, no spoofing signals are present. This allows classical receivers to acquire a location first. With these benign signals, the location estimates of our algorithm are accurate as can be seen in Figures 6.4 and 6.6.

When the spoofer starts to introduce the location error, the correlation peaks of the satellite signal acquisition get broader which increases location error and variance. This happens approximately in between 180 s and 280 s for Scenario 4 (Figure 6.4) and in between 150 s and 250 s for Scenario 6 (Figure 6.6).

When the spoofed and authentic location differ enough and the spoofed and authentic satellite signals become visible individually, the algorithm is able to distinguish between the authentic and counterfeit signals and the location estimations start to diverge. One location starts to track the spoofer signals while the other location recovers the original location. Due to the increased signal-to-noise ratio, the original location is not recovered perfectly, but the error remains small.

The average, maximum and median error, as well as variance for both location estimations compared to the clean recordings can be found in Table 6.3. A maximum location error of 222.4 m and a median error of 18.8 m are not exceeded. The best anti-spoofing work that we are aware of can only *detect* spoofing attacks, but may not find the actual receiver location and has a maximum location offset without detection greater than 1.5 km [104].

It can be observed that the results from the first iteration of our algorithm achieves far better results for Scenario 4. A spoofer with constant power advantage should lead to worse results for the first iteration compared to the second, since the spoofed location is found first. Figure 19 of [52] shows indeed that the spoofer has a power advantage only during the beginning of the spoofing attack and looses it within a couple of seconds. A classical receiver whose tracking loops have already been acquired by the spoofer would continue to track the spoofed signal whereas our algorithm falls back to the stronger signals which in this case are the authentic signals. Scenario 6 has no such behaviour as the spoofer has a continuous power advantage.

6.6.3 Correlation Length

For the presented results, a correlation length of 3 ms length is used. Figure 6.7 shows the average, median and maximum error of Scenario 6 for different correlation lengths. The best results are achieved with a correlation length of 3 ms, which is the length used during our algorithm development and testing. Foucras et al. present that 5 milliseconds should be the ideal trade-off between long correlation length for high SNR and low probability of bit flips [45]. In our case, the accuracy is similar with 3 and 5 ms. One reason for the good performance of 3 vs. 5 ms might be that longer correlation lengths increase the absolute SNR advantage of the spoofed signals, because those are slightly stronger than the authentic signals. This could worsen the results for the authentic location found in the second iteration of our algorithm. Also, some parameter tuning, like post-processing of the acquisition results, may change the results somewhat.

6.6.4 Computation Time

Currently, the algorithm is optimized for robustness rather than speed. However, depending on the required update rate it is possible to use the algorithm in real-time applications. Computation speed is mainly dependent on the size of the search grid, number of visible satellites in the signal, the SNR and sampling rate of the recording.

Currently, two limitations impact the performance directly. Computing the likelihood of each point is bounded by memory speed. This means that doubling the sampling frequency, and thus the amount of data, also doubles the computation time. Currently, at least half of the computation time is used for computing the likelihoods. The second performance limitation is the calculation of grid points and code phases. This accounts for about one third of the computation time. Experiments with pre-calculated satellite orbits show that this could be reduced significantly. Sorting and filtering the points consumes the remaining computation time. Calculating the satellite acquisition results is negligible.

A tracking feature which feeds back the location and time information of the previous location estimation allows the receiver to reduce the search space significantly. But due to branch and bound, the computation time is reduced by only 30 % in the tested scenarios, while the search space is two magnitudes smaller. This allows us to compute a location estimation in around 1.0 seconds. As our algorithm computes two location estimations

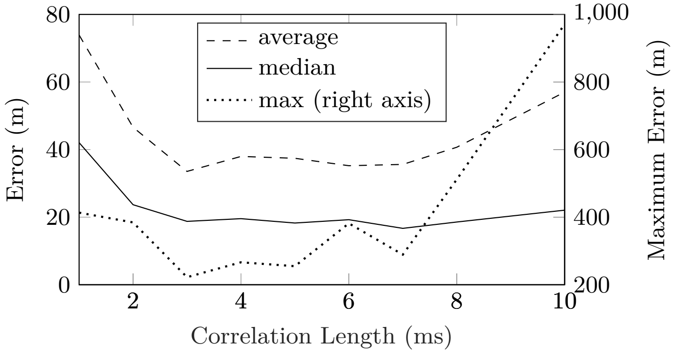


Figure 6.7: Average, median and maximum error for different correlation lengths in Scenario 6. The algorithm shows best results with 3 ms correlation length. Good results are also achieved with 4 and 5 ms correlation length. Note that the maximum error is shown at a different scale.

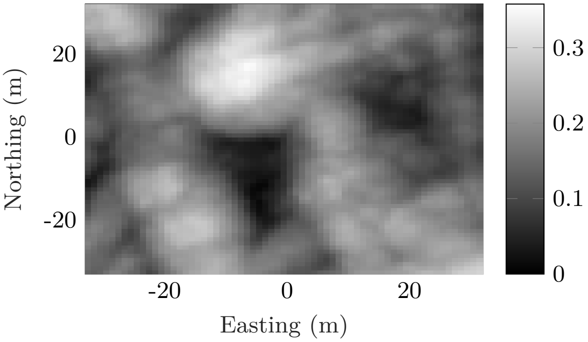


Figure 6.8: Two-dimensional likelihood distribution computed from a signal snapshot of a spoofing scenario. Higher values indicate higher likelihood of the point being the receiver location. The actual receiver location in the middle is invisible while the spoofed location slightly northwest dominates the likelihood distribution. This means that the first iteration of our algorithm would find this spoofed location. Other points with high likelihood result from a combination of spoofed and authentic signals.

per point it takes 2.0 seconds per point to calculate. We calculated 2349 points for the static scenarios and 2090 points for the dynamic scenarios taking roughly 80 minutes and 70 minutes respectively.

6.7 Conclusion

GPS spoofing is a broad topic and many methods have been proposed to detect and mitigate spoofing. Most research focuses on the detection of spoofing attacks. Methods for spoofing mitigation are often specialized or only work for certain scenarios.

Our implementation and evaluation shows that with some modifications, the robustness of collective detection can be exploited to mitigate spoofing attacks. We show that multiple locations, including the actual one, can be recovered from scenarios in which several signals are present. Experiments based on the TEXBAT dataset show that a wide variety of attacks can be mitigated. In the TEXBAT scenarios, an attacker can introduce a maximum error of 222 m and a median error under 19 m. This is less than a sixth of the maximum unnoticed location offset reported in previous work that only *detects* spoofing attacks [104].

Since our method does not track signals, but works with signal snapshots, our spoofing mitigation method is suitable for snapshot receivers, which are a new class of low-power GPS receivers [40, 75].

7

Snapshot GPS Hardware

> *"Energy and persistence conquers all things."*
>
> — Benjamin Franklin, 6th President of Pennsylvania

Global localization is a driver for so many applications that it is often considered to be a key technology of our time. However, all GPS receivers today have a high energy consumption. Mobile phones and smart watches can run days or even weeks on a single battery charge, but with GPS enabled, they barely make it through a single day. While personal devices such as smartphones can be recharged regularly, GPS trackers cannot.

Applications for GPS tracking include animal tracking, both wildlife and domestic animals. In addition, one may like to track personal items such as wallets or keys. More generally, we believe that the availability of a low energy GPS receiver will open up a unforeseen number of surprising applications, in tracking and beyond. Many of these applications also need a small footprint in terms of size and weight.

Current commercial GPS receivers include a lot of signal processing hardware, mostly so-called correlators, which are used to find and track satellite signals. These correlators collectively consume much power and the hardware is active continuously, because receivers constantly decode timing and satellite orbit information from the satellite signals.

In this chapter, we present a novel GPS tracker hardware design. Our design is a snapshot GPS receiver which captures only a few milliseconds of satellite signals for each location computation. The active time of snapshot receivers for a single location request is three orders of magnitude lower than that of classical receivers. The latter require about six or even 30 seconds of data at startup, depending on available prior satellite orbit information. Snapshot receivers can be designed either with a storage or a wireless communication component. We choose the first option, as it consumes less power and space. Loggers such as our device can be used for applications which do not need real-time localization, like wildlife tracking, collecting workout statistics or geotagging photographs.

Besides the hardware design, we present a corresponding prototype implementation using a suitable selection of components. An evaluation of the actual energy consumption shows that such a tracker, powered from a single coin cell, is not limited by the energy consumption, but rather by the size of the storage for the recorded signals. The used 2 Gb flash storage can hold 65600 signal snapshots of one millisecond length. This corresponds to a lifetime of 683 days with quarter-hourly localization. Our prototype GPS tracker weighs a mere 1.3 grams and its dimensions are 23 x 14 mm. This makes it suitable for weight-constrained applications like bird tracking and enables it to be concealed for instance in valuable belongings like wallets, handbags or bicycles.

With our hardware receiver design and the insights we gained when designing and testing our receiver, we want to provide the GPS research community with a platform to test and build snapshot receiver algorithms. Furthermore, our receiver design is a step towards a practically usable hardware building block, which can be integrated into a real product, like for instance a low-power, long-term animal tracking device.

7.0.1 Related Work

Commercially, no GPS hardware is available to implement snapshot receivers. One way to test snapshot GPS is to use a *software defined radio (SDR)*. SDRs are relatively large, heavy and consume orders of magnitude more power than a dedicated GPS receiver. Therefore, SDRs are most useful for static testing, but not for mobile scenarios. The same holds for the only alternative, which is using a *SiGe GN3S*[1] USB GPS sampling dongle together with a laptop. This is a problem for the GPS tracking research community, because snapshot receiver algorithms cannot be tested in their intended application environment. So far, mostly simulations or data cut

[1] The product is not available any more: `https://www.sparkfun.com/products/retired/10981`

out from longer recorded signal sequences have been used to show the performance of these methods [11, 14, 74].

In research, snapshot GPS receivers are known for several years [11, 14, 74]. They drastically reduce the power consumption of a GPS receiver, because signal processing can be offloaded to a web service [74]. This simplifies the hardware design and moves the most energy consuming part of the receiver into the cloud. However, most proposals focus on the software of such a receiver. While Liu et al. [74] propose a snapshot receiver hardware design, their first version used additional, large hardware for time synchronization. We use the same MAX2769 GPS front-end chip. However, our prototype implementation is almost 12 times smaller than Liu et al.'s second version, *CLEON*, that drops the time synchronization hardware. Also, our hardware draws a standby power of 9.6 µW instead of 9.25 mW [74], effectively increasing its lifetime for long duty cycles by a factor of almost 1000. And our receiver's active energy during a signal capture is reduced by a factor of 84, from 62 mW s [74] to 0.74 mW s, while capturing only 10 times less data, namely one millisecond instead of 10, and while improving the localization accuracy.

7.0.2 Applications

We give two example applications which can directly benefit from the availability of snapshot GPS receivers and do not require real-time localization. One has to keep in mind that due to the drastic improvements in size, weight and power consumption, snapshot receivers may spark a variety of unexpected applications.

Bird Tracking Ornithologists use tracking devices to study bird behavior. Large birds like geese or birds of prey can be equipped with classical GPS tracking devices. Due to weight constraints, batteries can only be small and will thus last for a short time only, limiting the usefulness of such trackers. Small songbirds can only carry additional weights of less than two grams [17], which is not enough for a classical GPS receiver and a battery. A current technique is to equip such small birds with small and low-power light-level sensors and a real-time clock. Reading the light levels and matching them with timestamps from the clock allows determining the length of the day at a bird's location and thus determining its latitude approximately. Errors are on the order of 200 km or more [17]. This allows for a limited set of studies like observing approximate migratory bird movements and their timing. Our receiver, which weighs 1.3 grams only, fits into the weight budget for equipping such small birds with GPS, while providing several months long observation times. Due to our receiver's accuracy in the range of tens of meters (see Section 7.3.3), our hardware enables more detailed studies on bird behavior.

Holiday Logging Many travelers like to tag their holiday photographs with the location where those were taken. Due to the high energy consumption and the multi-second latency from activating a receiver to getting the first location estimate, many cameras do not include a GPS receiver. Therefore, some people buy stand-alone GPS trackers which run a day or two on a single charge and whose computed locations can be combined with the holiday pictures afterwards on a computer. Our receiver eliminates recharging. After initial setup, our tracker can be forgotten about, even for a world tour! In the end, one can extract all locations with 15 minute resolution and has a log of the complete holiday journey.

Summarizing, our work lays the foundation for inexpensive, accurate and low-power GPS localization. It enables a new range of objects and animals to be equipped with global localization.

7.1 Hardware Design

A snapshot GPS receiver samples a few milliseconds of GPS signal and stores or transmits this data for computing the receiver location from it. *Raw* data is needed, meaning I/Q or real samples of the signal and not processed data that commercial GPS chips provide.

The goals of our snapshot receiver design are:

- Capture raw GPS signal samples
- Store them on the device
- Keep track of the current time
- Allow simple configuration and data transfer
- While consuming minimum power

Our design addresses all of these goals and allows for large duty cycles with minimum sleep power.

7.1.1 Sampling

The frequency of the L1 GPS code modulation is 1.023 MHz. Therefore, by the Nyquist-Shannon sampling theorem, a sufficient minimum sampling rate with a single channel (*real*) receiver is 2.046 MHz and half of that for a dual channel (I/Q) receiver. Using a higher sampling frequency will usually yield a better quality of the received signal. But more importantly, the Galileo GNSS is also transmitting ranging signals at the L1 frequency, but with a sub-carrier rate of 6.138 MHz. Therefore, it is beneficial if a sampling rate of at least 6.138 MHz (I/Q) or 12.276 (real) is used. In our design, we settle for a real receiver and a sampling frequency of 16.368 MHz, which allows for the simultaneous reception of GPS and Galileo signals, increasing accuracy and robustness.

As seen in Table 7.1, most GPS front ends use 2-bit quantization levels. Using such low sampling precision degrades the signal-to-noise ratio by only 0.55 to 0.72 dB [125, Section 6.12], while the reduced data size allows capturing more snapshots with the same energy and storage space.

7.1.2 Component Selection

The main parts required for our hardware design are:

- GPS Front End
- Microcontroller
- Flash Storage
- Battery
- Power Converters

GPS Front End The front end is the circuit converting the received RF signals into digital samples. Although we spent quite some time searching GPS front-end chips, there seem to be only a dozen manufacturers producing standalone chips. All the models we could find are listed in Table 7.1. Note that the reason for this short list is probably due to the fact that most GPS receivers in commercial products integrate the location computation and are not designed to output the raw signal samples. Even though some commercial receivers seem to offer “raw” data output, they actually only report computed values like pseudoranges to all satellites or navigation data is provided, which is insufficient for our application, which requires raw GPS signal samples. For the front end we selected the Maxim MAX2769 GPS front end as it allows testing a wide range of RF, data format and

Table 7.1: List of all commercial GPS front-end chips that we found through an extensive search (web, books, emails, phone calls). The following abbreviations are used to denote the different GNSS systems. US: GPS, R: GLONASS, E: Galileo, C: BeiDou.

Manufacturer	Model	GNSS	Sampling rate	Sample format	Max. power	Min. quantity	Price
Analog Devices	ADSST-GPSRF01	US	max. 32 MHz	2 bit real	195 mW	unavailable?	on request
IMST	[unnamed]	US/E	*unknown*	2 bit real	*unknown*	unavailable	*unknown*
Maxim	MAX2769	US/R/E	max. 50 MHz	2 bit I/Q or 3 bit real	62.7 mW	2500?	on request
Maxim	MAX2769B	US/R/E/C	max. 50 MHz	2 bit I/Q or 3 bit real	88.4 mW	2500?	on request
Navika	AST-GPSRF	US/E	16.368 MHz	2 bit real	> 48.9 mW	discontinued	*unknown*
NTLab	NT1065	US,R,E,C	≥ 99.231 MHz	4 x 2 bit real	> 306 mW	*unknown*	*unknown*
SAPHYRION	SM1027U	US/R/E/C	50 MHz	3 bit I/Q	≥ 76.9 mW	*unknown*	*unknown*
SiGe	SE4110L	US	max. 19.5 MHz	2 bit real	> 28.4 mW	1	$ 3.59
Skyworks	SE4150L	US	16.368 MHz	2 bit real	59.4 mW	1	$ 3.20
STA	STA5620	US	16.368 MHz	2 bit real	51.3 mW	221 (obsolete)	$ 4.78
STA	STA5630	US/E	16.368 MHz	3 bit real	25 mW	3000	$ 1.05
Zarlink	GP2015	US	5.71 MHz?	2 bit real	254.1 mW	discontinued	*unknown*

filter settings. The raw GPS signals are output at 16.368 MHz with two bit precision. Both active and passive antennas can be connected. It uses less than 59.4 mW. Unfortunately, shutdown power is about 54 µW which requires using an external switch to not exhaust our power budget. An alternative would be the SE4150L GPS front end, which requires less external components and therefore is easier to integrate. It also has fewer necessary settings and slightly lower power consumption.

Microcontroller The microcontroller needs to fulfill two important constraints: 1) It must read incoming samples, two bits at a time, at the designed rate of 16 MHz and 2) it should have a low standby power consumption to allow for long tracking periods with large duty cycles. During inactive times of the receiver without signal sampling, the microcontroller needs to keep track of the current time while all other receiver components can be cut off from the power source to save energy. Thus, the microcontroller's standby power consumption is one of the factors limiting the battery longevity. For our design, we select the Atmel SAM4L, as it offers a *parallel input capture interface (PARC)*, which reads up to 8 bits concurrently at a maximum rate of 24 MHz. The PARC is a perfect interface for reading the data of the GPS front end. At less than 5.6 µW, the SAM4L offers low standby power consumption with activated *real-time clock (RTC)*.

Flash Storage The flash storage must offer large storage size while consuming little power during write operations. The NAND flash memory MT29F2G01 offers 2 Gb storage and a maximum power during data write operations of 45 mW. At our sampling rate of 16.368 MHz, the flash memory size allows collecting one GPS signal snapshot every fifteen minutes during 683 days. Combined GPS and Galileo snapshots could be captured hourly during the same time period.

Battery Many batteries are heavy (cylindrical batteries) or have significant self discharge (LiPo cells, supercapacitors), making them unsuitable for our purpose. Coin cells offer high power density, low weight and are cheap, which makes them ideal for our requirements. We use the CR2032 [41] which has an energy capacity of 653 mW h. With a target run time of 2 years, an average power consumption of 37.2 µW cannot be exceeded. Accounting for battery degradation, the power consumption has to be restricted to significantly lower values. Real-world evaluations of coin cells show that high peak currents or low quality can lower a coin cell's actual capacity to only half their rated capacity [123]. To ensure stable operation even with temperature variations, bad coin cell quality or high peak currents, the average power consumption of our receiver should therefore remain below 18.6 µW. Our average power consumption of 9.6 µW (cf. Section 7.3) is well within this soft limit and our tracker requires only a quarter of the coin cell's rated capacity for a two year run time.

Power Converter The coin cell offers an initial voltage of 3 V, which will drop to 2 V during high load or towards the end of the cell's lifetime. We use two controlled power domains. One domain with 1.8 V for the microcontroller and one domain for the GPS front end with at least 2.7 V. While it would be possible to design the entire system for 2.7 V, 1.8 V reduces the external components required for the processor and minimizes standby power consumption because an efficient step-down converter can be used. The TPS63743 step-down converter powers the 1.8 V domain and its quiescent power is only 1.1 µW, which allows it to be active continuously. The TPS61098 provides 2.7 V to the GPS front end even when the coin cell voltage drops.

Standby Power Consumption Both the flash memory and GPS front end consume a couple of µW during shutdown. While this might be negligible in most applications, the shutdown power consumption of these chips combined will empty the coin cell in less than our targeted two years lifetime. Using a controlled high-side load switch like the ADP199, the power consumption during shutdown can be reduced to below 300 nW.

Ceramic capacitors can have leakage powers in the range of nanowatts to a few milliwatts. Therefore, it is crucial to remove and minimize all capacitors wherever possible, to stay within our power budget. Resistors have to be planned carefully, too, to not waste any energy. An alternative is to use different switched power domains, decoupled by the aforementioned load switch, for example.

7.2 Implementation

An overview of the final architecture can be seen in Figure 7.2. The microcontroller is powered continuously by the step-down converter. To save energy, the NAND flash is connected through a load switch to the power domain of the microcontroller. The switch is controlled by the microcontroller and only enabled when the flash is necessary. Communication with the flash is performed via the *Serial Peripheral Interface (SPI)* at data rates of up to 24 Mbps. The GPS data is transferred in parallel with 16.368 MHz to the capture interface of the microcontroller, where it is first cached in RAM and then written to memory.

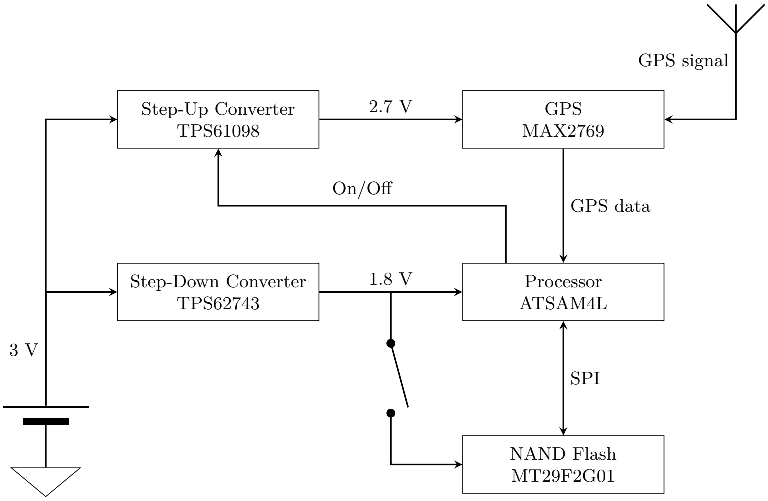


Figure 7.2: Overview of the main components in the system. Arrows indicate the energy and data "flows".

7.2.1 PCB Design

To minimize PCB size and allow using small parts, the board is designed with four layers and a minimum design width of 100 µm. To reduce PCB cost, only vias through the whole stack are used, but no blind or buried vias. Passive components are reduced to the smallest size available, often 01005 (0.4 mm × 0.2 mm). The resulting board is only 23 mm × 14 mm big and weighs 1.3 g. Figure 7.3 shows a close-up view and a size comparison with a wristwatch is shown in Figure 7.4. GPS front end, power management and microcontroller are mounted on the (visible) top side of the board while the flash chip and USB connector are mounted on the bottom side.

Two challenges are matching the impedance of the antenna connection and reducing the electromagnetic interference of digital signals with the RF signals. The antenna RF connector is mounted on the bottom through an impedance-controlled via buildup. The GPS front end is prepared for a shielding enclosure and RF traces are shielded by surrounding grounded vias and ground layers.

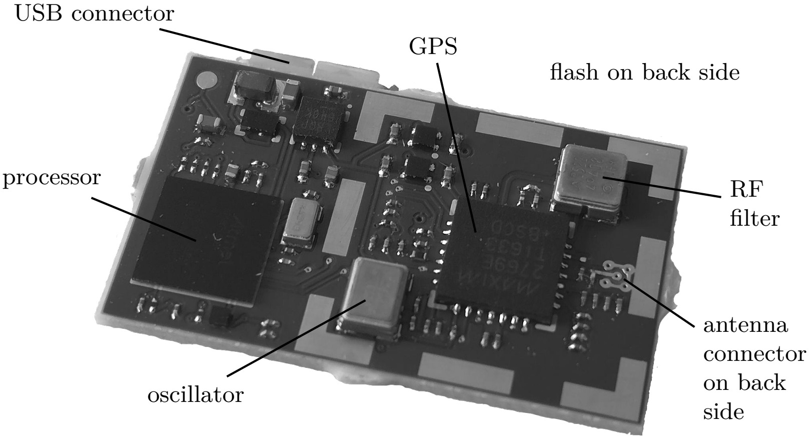


Figure 7.3: Close-up view of our GPS tracking hardware.

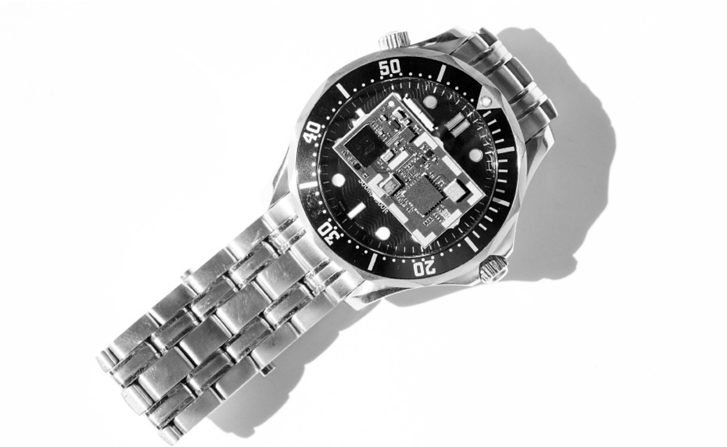


Figure 7.4: Size comparison of our GPS tracking hardware with a wristwatch.

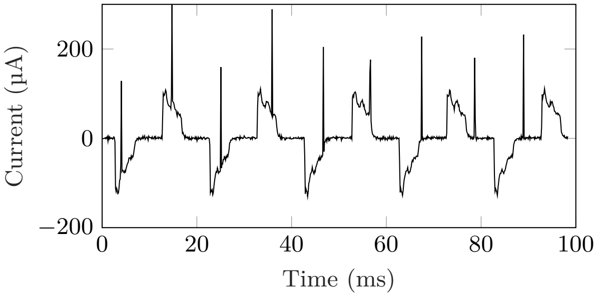


Figure 7.5: Total standby power consumption of the GPS tracking device. The processor is in deep-sleep mode with only the *real-time clock (RTC)* running, the step-up converter for the GPS front end disabled and the flash disconnected by the load switch. The variability of the current is a characteristic of the switching voltage converter powering the microcontroller. The average current is 3.2 µA which corresponds to a power of 9.6 µW.

7.3 Evaluation

This section gives an evaluation of the power consumption and signal reception. Power measurements are difficult to obtain due to the high dynamic range between the active and standby currents. Even attached debugging circuits can have much higher leakage currents than our device's standby power consumption. Special care was taken during the hardware development to make all debugging circuits detachable for measuring the power consumption. Furthermore, the GPS data is analyzed to verify proper operation of the receiver.

7.3.1 Standby Power Consumption

During standby, only the microcontroller is powered. The boost converter is in low-power mode and the flash memory switched off, which should consume less than 3 µW combined. The microcontroller is powered by the TPS62743 and should consume less than 10.5 µW combined. Therefore, a total standby power consumption of under 13.5 µW can be expected which is well below our maximum power budget of 18.6 µW.

Our measurements (Figure 7.5) show that the current during standby varies between -125 and 300 µA over time due to the switching voltage converter powering the processor. The average current measured with multi-

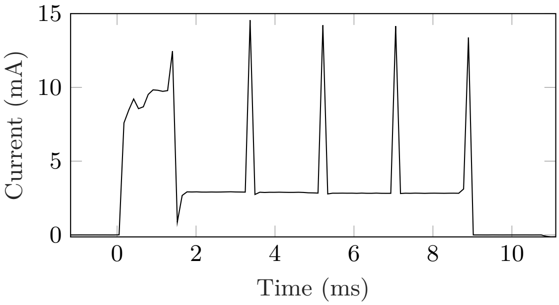


Figure 7.6: Current consumption of the flash device. After initialization for the first 1.7 ms, four blocks of data are transferred and written. Data writing is clearly visible as peaks in the power consumption.

meters and power analyzers ranges from 2.9 up to 5.1 µA. That corresponds to a power of 8.7 to 15.3 µW and is within the expected range.

7.3.2 Active Power Consumption

The active time can be split into multiple parts. After wakeup, the processor initializes the GPS front end and starts sampling the received data through the parallel interface. Then, the GPS front end is disabled and the processor formats the data for storing it. The final step is to power up the flash memory, transfer and write the data to the flash memory. The power profile of the flash chip during this procedure is shown in Figure 7.6. The full process takes around 13.5 milliseconds to complete. To estimate the power consumption, the maximum values from the data sheets are used. At full speed and with the required peripherals activated, the microcontroller consumes approximately 60 mW. This does not account for low-power modes or lower clock rates that could be used to reduce power consumption. The GPS front end and crystal oscillator consume at most 75 mW. During write, the flash memory will consume at most 75 mW. Depending on the efficiency of the power converters and other peripherals, these values can vary.

Our measurements (Figure 7.7) reveal that the power consumption of the flash and microcontroller are in the expected range, while the power consumption of the GPS front end is higher than expected, leading to an initial peak power of 60 mA. This probably originates from the initial charging of the stabilizing capacitors and setup of the GPS front end.

The average power consumption for the 13.5 ms active time is 54.9 mW. Our initial example of storing one GPS signal snapshot per hour corresponds

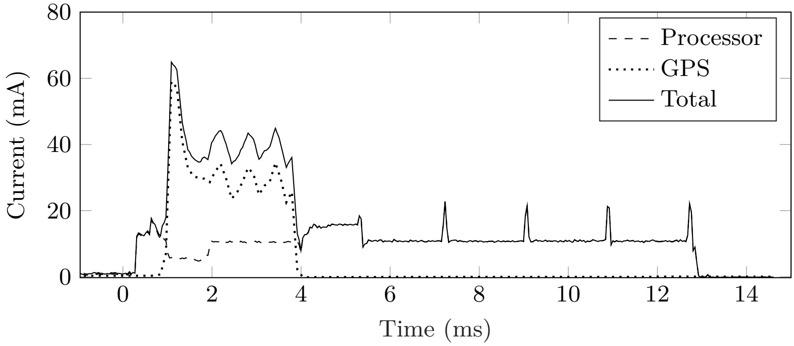


Figure 7.7: Active power consumption measured before the power converters by a power analyzer. The processing phases are visible. After the processor has run initial setup, the step-up converter is enabled and powers the GPS front end. After a short settling time the processor reads raw GPS data for one millisecond. Starting at 4 ms the processor prepares the data for transfer to the flash. At 5.5 ms to 13 ms the data is transferred in 4 blocks to the flash memory. The peaks in power consumption indicate when the data is written. After the last write has finished, the processor disables all peripherals and returns to sleep mode.

to a duty cycle of 13.5 ms/3600 s = 3.75e−6. The contribution of the power during the active time to the average power consumption is $54.9 * 3.75e{-6} =$ 205.9 nW. In case of snapshots every 15 minutes, the contribution increases to 823.5 nW. Both values are negligible compared to the standby power.

7.3.3 GPS Data Analysis

In a field test, our receiver is located on top of a university building. Recorded snapshots with one millisecond of data are evaluated with a branch-and-bound collective detection implementation [14]. *All* calculated locations are within 25 meters to the true receiver location. Although we did not do an extensive evaluation of the localization accuracy, this is an encouraging preliminary finding. In comparison, Liu et al. evaluated the snapshot localization accuracy with a GPS sampling dongle instead of their presented snapshot receiver and used 10 milliseconds of data instead of one. Still, they achieved less than 25 m error only in about 80 % of all cases and observed a maximum localization error of 725 m [74]. Our improved localization accuracy conveys the impression that collective detection more than compensates the expected accuracy degradation from the lower signal energy contained in the shorter snapshots. This result suggests that collective detection indeed improves localization robustness [14]. However, a larger set of experiments is needed to assess this hypothesis.

7.4 Conclusion

Our hardware design, implementation and evaluation show that low-power GPS receivers, which offload the location computation to the cloud, are a viable concept. Our receiver periodically takes snapshots of GPS signals and stores them locally. In some applications, our receiver may be retrieved after the data collection, in order to download the sampled data, from which the locations can be computed. Alternatively, the receiver can be connected to some communication module which transfers the data to the cloud regularly or even in real time. In the latter case, the flash chip could be omitted. The mean power consumption our receiver is about 10.4 µW with a 15 min duty cycle, allowing the receiver to run for two years on a single coin cell. This enables new applications that were impossible to date, for instance long-term tracking of small animals such as migratory birds. Our design exhibits a thousandfold improvement in standby power consumption and a hundredfold improvement in active power consumption over previous work.

A possible future research question is whether the coin cell could be replaced with some energy-harvesting device. For instance, a bird's movement or body temperature could be leveraged to produce the necessary energy for GPS tracking. As for bird tracking, the energy generation rate will be low, a low-decay supercapacitor might be used to accumulate energy until enough energy is available for a position snapshot. Using a capacitor which can store enough energy for a complete position snapshot would actually also be useful when powering our device with a coin cell. The power draw of our device exhibits some spikes, which are detrimental to coin cells, reducing their lifetime. Therefore, one could charge a capacitor slowly from the coin cell before taking a position snapshot powered by the charged capacitor.

8

Indoor Localization with Aircraft Signals

> *"I have not the smallest molecule of faith in aerial navigation other than ballooning, or of the expectation of good results from any of the trials we heard of. So you will understand that I would not care to be a member of the Aeronautical Society."*
>
> — Lord Kelvin

The British Science Association has called GPS technology the #1 invention "that changed the world".[1] Originally developed by the US military, GPS is now used in a majority of mobile devices. GPS has enabled a multitude of applications, which 50 years ago must have sounded like magic. However, GPS also has a major drawback — its satellite signals can hardly be received indoors.[2]

[1] Richard Gray. Top 10 'inventions' that changed the world. The Daily Telegraph. March 13th 2009. `http://www.telegraph.co.uk/news/science/4981964/Top-10-inventions-that-changed-the-world.html`

[2] Also, multipath effects are detrimental for GPS signal reception and localization accuracy. The official GPS website by the US government has a list of causes for inaccurate location estimates, which includes the "urban canyon" effect in inner cities and also less frequent causes: `http://www.gps.gov/systems/gps/performance/accuracy/#problems`

The problem is intrinsic. GPS satellites have an effective signal transmission power of 283-458 W [127, Section 2.2]. With an altitude of about 20,200 km, this relatively weak signal barely makes it to earth. Already the free-space path loss is about 182 dB. If only GPS satellites flew lower and had more power!

In contrast, airplanes and other aircraft are flying at an altitude below 13.7 km. The ADS-B standard prescribes a minimum transmission power including antenna gain between 116 and 331 W, depending on the type of aircraft [108]. With a free-space path loss of about 139 dB for an aircraft at a distance of 200 km, the aircraft signals are therefore received on the Earth surface with approximately 5,300-15,000 times higher power than GPS signals. More details concerning the GPS and ADS-B received signal powers are given in Section 8.2.2.

For safety reasons, airplanes and helicopters repeatedly transmit their location, pretty much like GPS satellites. These so-called ADS-B signals are strong enough to be received indoors, even with cheap hardware. But are these air traffic control signals *precise* enough to not only locate the aircraft but *any* mobile device?

As air traffic control signals have not been designed for indoor localization, we have to deal with three challenges:

1. Aircraft do not fly on an orbit; aircraft do not have accurate predetermined flight paths and unexpected changes to their route are always possible, for instance due to a holding pattern when approaching a crowded airport.

2. Aircraft are not uniformly distributed in the sky. This is in stark contrast to GPS satellites, which cover the sky in a regular pattern in order to maximize user localization performance.

3. Aircraft location signals are not precise: an aircraft has an unpredictable delay between learning its location from the GPS satellites and retransmitting this location;[3] unlike GPS satellites with their atomic clocks, aircraft transmissions do not include time information; some aircraft in fact do not even include location information.

But there are good features as well. Although aircraft do not fly on an orbit (1), passengers and crew certainly do not appreciate abrupt flight path changes. Also, even though the aircraft locations are not optimized for the localization of users on ground (2), but rather for air traffic safety, at least

[3] Uncompensated latency of up to 0.6 s: https://www.law.cornell.edu/cfr/text/14/91.227#e

in urban areas there are more aircraft available than satellites.[4] A large number of received signals hopefully reduces the statistical uncertainty of the location estimation from noisy measurements (3).

So, do these good news compensate for the bad news above? It turns out, we cannot simply use the aircraft signals. Instead, we propose to install a few receivers in known locations. Not many receivers are needed, in principle already a single receiver is enough to serve a large metropolitan area or even a small country. These receivers will calibrate the received aircraft messages, so that any mobile receiver can deduce its location without any additional infrastructure. More formally, our localization system consists of

- a network of receivers with known locations, which we call *ground stations*
- a receiver whose location should be determined – we refer to this receiver as the *handset*
- a *server*, which connects the ground stations and the handset

The ground stations receive aircraft signals, decode messages and precisely determine their arrival times. The messages and corresponding timestamps are aggregated to batches in a JSON file. Every three seconds, the latest file is sent to the server over the Internet using an HTTPS POST request. Given enough records for the same message, the server can then determine the message's transmission location and time. In practice, most aircraft in Europe and a growing number of aircraft in other countries already broadcast their location [20, 36]. So, the only unknown is the transmission time, in which case one ground station is sufficient.

To localize a handset, its received messages are sent to the server like the messages received at the ground stations. The server then matches the handset messages with the corresponding messages from the ground stations for which the transmission location and time are known. Using the signal arrival times measured by the handset — whose time is biased compared to the system time — the localization solution, consisting of the handset's location and time, is computed.

We use receivers which consist of a 7 cm long antenna, a USB software-defined radio receiver dongle and a Raspberry Pi 3 board. The total cost of

[4] At night, the frequency of received aircraft signals is substantially lower than during daytime. Still, some long-haul flights may be passing even at night and some airports operate 24 hours per day, ensuring regular traffic around them. So, enough aircraft for localization can be available at night, but the localization accuracy may be degraded. While most people are sleeping and do not need any localization service, for instance emergency situations also arise at night. Even inaccurate localization helps emergency services finding patients quickly.

such a receiver is less than $100. In the future, such a receiver design could easily be integrated into a smartphone. A 7 cm antenna is small enough to fit inside the casing, and current smartphones have more processing power than the Raspberry Pi 3.[5] Our signal reception and decoding software is a modified version of the open source project *dump1090*,[6] which we enhanced with more accurate time resolution. With these cheap receivers, messages sent from aircraft as far as 250 km away can be received.

The GPS Performance Standard [38] by the US government currently lists a worst-case horizontal GPS accuracy better than 17 meters in 95 % of all cases. Depending on the quality of the receiver and whether different advanced correction methods are available, the horizontal accuracy can be substantially better, in the order of 3-7 meters. Usually, indoor localization methods attempt to be even *more* accurate, for instance to minimize the path length of a vacuum cleaning robot. And optical motion capture systems achieve localization accuracies better than one millimeter [10].

In terms of accuracy, our method cannot compete with these indoor localization techniques, and does not even achieve the typical GPS outdoor accuracy. Our prototype implementation has a median error of about 25 meters. On the plus side, it works both indoors *and* outdoors!

Even though 25 meter accuracy is not exciting, 25 meters is still better than *nothing at all*, and various applications do not need a precise location. For instance, our method may tell you in which building you took some photograph. Also, our method can help you catch the next bus from your current location, since no precise location is necessary to determine the closest bus stop and look up the corresponding timetable. Moreover, one can automatically log the working time of employees by just knowing an approximate indoor location. Due to the higher received signal power of the aircraft signals, our system also works in urban canyons, where GPS receivers may not be able to detect the signals from the GPS satellites, because urban canyons behave similar to indoor environments. Generally, our system would be of value for applications which cover both indoor and outdoor locations, or when dedicated indoor localization infrastructure is too costly to deploy. Finally, and probably most importantly, our method may offer a more accurate initial guess for Assisted GPS (A-GPS) than

[5] OpenBenchmarking.org, which is one of the standard benchmark sites for Linux systems, shows benchmark results comparing the Raspberry Pi 3 to the ODROID-C2 development board, which features an ARM Cortex-A53 SoC, also found on current low-cost phones: `https://openbenchmarking.org/result/1603051-GA-ODROIDPI362`. Even this low-cost SoC, which is the core of the Qualcomm Snapdragon 410, 610 and 615 [115,116] beats the Raspberry Pi's performance in all tested workloads.

[6] `https://github.com/mutability/dump1090`

currently provided by cellular networks.[7] Having a better estimate of the receiver location and time can speed up the initial GPS fix, especially when a maximum likelihood method like collective detection [11, 14] is applied.

Another upside of our method is that, apart from sparsely distributed ground stations and a server, it is pretty much independent from additional infrastructure. For real-time tracking, only an Internet connection is necessary to synchronize information with the ground stations. It is also possible to post-process data, in which case no infrastructure is needed at all. The latter approach has for instance been successfully applied to GPS localization [75].

Since our method and implementation presented in this chapter are a proof of concept, future improvements might yield more accurate localization using the same aircraft signals and could thus allow such a localization system to be used for even more indoor applications.

Our method is one of the first basically infrastructure-free indoor localization methods. As discussed in Chapter 8.1, the accuracy of our method is comparable to LTE localization, which is a competitor to infrastructure-free indoor localization due to the wide availability of cellular networks. These networks are designed for communication purposes and therefore cover an area with at least one antenna, but often not more, in order to save costs. Moreover, cellular networks use transmission antennas located close to the ground. For localization, it is therefore often not possible to receive signals from a sufficient number of antennas. In contrast, aircraft are high up in the sky and can thus provide a better area coverage for instance in cities, forests and mountains.

First, in Section 8.2 we give an introduction to air traffic surveillance systems and the localization technique. Then, we dive into the details of our method in Section 8.3. We present our implementation in Section 8.4, followed by results in Section 8.5. To the best of our knowledge, there is no closely related prior work to our approach. We discuss various other existing indoor localization methods and air traffic surveillance systems in Section 8.1.

8.1 Related Work

Our method employs aircraft messages for the purpose of indoor localization. To the best of our knowledge, only Faragher et al. have used aircraft signals for self-localization before [42]. Unlike our method, which is based on time-of-flight (ToF) measurements, Faragher et al. employ angle-of-arrival

[7] Still, our system might use cellular networks for mobile receivers to communicate with the server.

(AoA) measurements. Because reflected signals, for instance at walls, result in wrong AoA measurements, their method is not well-suited for indoor and urban environments, but rather for situations with line of sight between aircraft and receiver. Their receiver needs two antennas which should be separated by a considerable distance to achieve useful localization accuracy. With antenna distances of 1.6 m and 14 cm, they get bearing measurements with standard deviations of around 1°–2°and 7°–15°, respectively. Using messages from aircraft closer than 100 km, collected over 100 s for each position fix, they get a localization accuracy of 1,200 m and 200 m with the 1.6 m and 14 cm antenna baselines, respectively. The long message collection duration might be necessary because of the restricted aircraft distance. However, using messages from more distant aircraft will deteriorate the localization accuracy, because the bearing error amplifies with the distance. In comparison, our receivers use a single, roughly 7 cm long, monopole antenna, which makes them suitable for handheld devices. Depending on the receiver location, messages from aircraft at distances up 250 km can be received. We aggregate 30 s of data for each position fix and the median localization accuracy is approximately 25 m. Further, Faragher et al.'s method is sensitive to multiple parameters which have to be calibrated. Those parameters are the antenna baseline, the antenna orientation and the phase between the antenna channels. The sensitivity is extreme, as for instance an antenna baseline error of only 1 cm results in a localization error of many kilometers. Since the antenna orientation may change in a mobile receiver, the use cases of their localization method may be limited. Also, they state that the phase between the channels changes for each recording, which may make duty cycling of the receiver impractical, because the sensitive, three-dimensional parameter calibration routine has to be rerun. And the authors propose that the calibration may use the help of GPS localization, which – again in the case of duty cycling – would make the ensuing localization through aircraft signals superfluous. Concerning the receiver hardware, Faragher et al. use custom-made antennas and a software-defined radio (SDR) which costs over $ 1,000, while our receiver design consists only of commercial off-the-shelf components and costs less than $ 100 in total.

In the following, we discuss other existing work in the two independent fields of indoor localization (Section 8.1.1) and air traffic surveillance and control (Section 8.1.2).

8.1.1 Indoor Localization

Much of the research on indoor localization focuses on providing accurate localization, for instance room-level or even sub-meter accuracy. The cost factors to get so accurate are

(I) the installation of dedicated **I**nfrastructure, for instance one beacon in each building up to several in each room;

(T) a **T**raining or initialization phase to gather data which is necessary for the subsequent localization;

(E) the usage of **E**xpensive user equipment.

Most methods do not have all three of these drawbacks, but at least one. In contrast, our method is almost infrastructure-free, does not need training and receivers are cheap. Simply adding a small antenna can turn a cheap smartphone into a user handset. Our method requires only one or a few ground stations for a region, which can be hundreds of kilometers in diameter. Therefore, our method is suitable to be used on a global scale, wherever aircraft are present. In contrast to GPS, which is also a global localization system, our method also works indoors. To get rid of all the cost factors listed above, we give up some accuracy. As outlined in the introduction, various applications exist for which localization precision is not essential. In other words, our method fills the void between cheap, global and easy to use outdoor localization such as GPS and precise, but local or expensive, indoor localization.

A plethora of indoor localization methods exist and different classifications are possible. For instance, Seco et al. [111] classify the methods into four categories: geometry-based, minimization of a cost function, fingerprinting and Bayesian. Another classification can be made based on the type of employed signals. For indoor localization, ultrasound and signals in basically the whole range of the electromagnetic spectrum up to light have been used [68, 70, 85]. A third possibility is to distinguish the methods by their kind of fundamental measurements, which include received signal power (also called *received signal strength (RSS)*), time of arrival (ToA), time difference of arrival (TDoA) or angle of arrival (AoA). An overview of these localization techniques is given by Liu et al. [73]. Our method uses a ToF technique, which uses ToAs measured by a handset with a time bias compared to the system time.

Here, we will discuss different indoor localization systems based on the employed signals' type. For each category, we indicate in parentheses which drawbacks the method has, using the letters from the list above.

WiFi (T) WiFi signals are popular for indoor localization, because WiFi hotspots are already widely in use. Therefore, no specific infrastructure, like beacons, is necessary for WiFi localization methods. In a survey by Liu et al. [73], many types of wireless indoor localization methods are compared and WiFi-based approaches are shown to generally have an accuracy of a few meters. This finding is confirmed by more recent results from the annual *Microsoft Indoor Localization Competition.*[8] WiFi localization methods require a training phase in which either the locations of WiFi access points are determined or fingerprints at different locations are gathered. Furthermore, infrastructure changes have to be detected and the database needs to be updated accordingly.

Ultrasound (I) In contrast to WiFi-based localization, which is infrastructure-free, ultrasound-based methods require dedicated hardware. However, ultrasound systems are relatively inexpensive and have proven to be accurate compared to many other indoor localization methods. For instance the SmartLOCUS [18] and SpiderBat [95] systems achieve centimeter-level accuracy. Still, ultrasound systems need high signal power to traverse distances more than a few meters and are prone to ambient noise, like for instance jingling keys. Also, ultrasound systems raise concerns about animal health compatibility – mostly pets like dogs and cats [70, Section 6.3].

Light (T,E) The most accurate results in the Microsoft Indoor Localization Competition are achieved by laser- and camera-based methods [81]. The best system achieves an accuracy of 5 cm using two lasers and multiple high-end cameras [81]. However, this system costs a quarter million dollars. Still, even cheap cameras today have a high number of pixels, resulting in a fine spatial resolution. For instance, smartphones featuring Google's *Project Tango* hardware[9] can achieve a localization accuracy as low as 2–5 % of the absolute distance to the object, which is less than 5 cm for objects closer than one meter [51]. However, the relative error of several percent is rather large. Also, because the maximum range of those phones' localization systems is limited to 4 m,[10] and because the compute requirements are high, mapping of large rooms or even a whole building takes a long time. After mapping a building, users can be localized by matching locally detected image features with the pre-constructed building model. Thereby, the local coordinate frame of the phone is linked to the coordinate system used when mapping the building. Like this, users can for instance learn in which room of a building they are.

The widespread use of light emitting diodes (LED) and the miniaturization of processors has opened a new field of visible light communication

[8] `https://www.microsoft.com/en-us/research/event/microsoft-indoor-localization-competition-ipsn-2014/#official-results`

[9] Currently, the *Lenovo Phab 2 Pro* and the *Asus ZenFone AR* are available.

[10] Google's Tango developer guide lists a range of 0.5 to 4 m: `https://developers.google.com/tango/overview/depth-perception#usability_tips`

and localization techniques. Pathak et al. [97] give an extensive overview of current methods. Among the advantages of visible light indoor localization, they list the large installation base of LEDs in buildings, which helps averaging out measurement noise. For visible light localization, approaches based on received signal power [72] and signal angle of arrival (AOA) [68] have been used. Signal power based approaches with light can achieve sub-meter accuracy [72] and therefore are one order of magnitude more accurate compared to WiFi signal power techniques [27]. Analyzing a camera image to perform AOA localization is even more accurate and can yield a localization error of 10 cm [68]. Sub-centimeter visible light localization has also been demonstrated, by using multiple receivers [134]. Similar to methods leveraging WiFi base stations, also LED transmitters' locations need to be learned in a training phase. In contrast to WiFi signals, light does not penetrate walls, which can be a benefit or a downside. On the one hand, interfering multipath signals from neighboring rooms, which introduce errors, are eliminated, but on the other hand, multiple LEDs have to be installed in every room.

Bluetooth (T,I) Another type of signal used for indoor localization is Bluetooth. Bluetooth is similar to WiFi in that both systems share the 2.4 GHz frequency band. Compared to WiFi, which can take tens of seconds for identifying base stations, faster response times can be achieved with Bluetooth [24, 83]. This is important, because at walking speed, the set of visible beacons can change quickly. Not being able to use signals from intermittently visible transmitters is detrimental for the localization accuracy.

Due to the protocol specifications, Bluetooth devices have to be paired before user data can be exchanged. However, it has been shown that only using publicly announced device names and received signal power of Bluetooth beacons is enough to achieve a localization error of less than 3 meters [62].

While most Bluetooth signal power based localization approaches achieve an accuracy of multiple meters, more elaborate approaches, for instance ones using neural networks can achieve sub-meter accuracy [3]. Accuracy using ToF measurements can be even better with an error less than 0.5 m [102].

RFID (I) There exists also work on localization with RFID tags. RFID tags come in two flavors: Active tags have an internal battery and passive tags do not. The latter therefore have limited capabilities. Since even active tags only have limited energy, RFID tags can only communicate over short distances and are mostly useful to identify the proximity of objects. Still, various more elaborate RFID localization schemes exists, measuring received signal power or signal arrival times. Bouet and Dos Santos [16] provide an overview of the work on RFID localization. RFID tags are cheap, but due to the short range, many tags have to be deployed for a localization system serving a whole building.

Sensor Fusion (not stand-alone) Sensor assisted localization methods are particularly favored in smartphone applications, because basically all of these devices feature an inertial measurement unit (IMU) comprising an accelerometer, a gyroscope and a compass. For instance, the accelerometer can be used to estimate the movement speed and the compass can provide the orientation of the device. Based on this principle and by counting steps of a person, pedestrian localization systems have been developed [61]. Although using the sensor data alone over long time periods is not accurate due to accumulating errors, the sensors can bridge gaps in the operation of another localization system. This technique is called *dead reckoning*. For example, cars driving into a tunnel will lose signals from GPS satellites, but based on measurements of the current driving speed, their location can be estimated until the tunnel ends [114]. Although IMUs are cheap and require no infrastructure, they are not suitable as stand-alone localization and navigation systems due to the drift of the estimated location. Therefore, continuous recalibration using a second localization system is necessary. Nevertheless, by relaxing the requirements from absolute three-dimensional localization to only detecting floor level changes, Ye et al. demonstrate that IMUs alone *can* provide the basis for sufficiently accurate results over multiple hours [135].

Cellular (T) Cellular networks, supporting protocols such as GSM or LTE, can also be leveraged for localization purposes. Promisingly, signals sent by cellular network antennas can be received over distances up to 35 km [25]. Therefore, in theory, few antennas can provide localization for large areas. However, because the available frequency spectrum is limited, the number of simultaneous users per cell is bounded. Thus, in order to serve more users, the signal power of practically all cellular antennas is intentionally set lower than the maximum. With this measure, also cells at distances closer than 35 km can reuse the same frequency spectrum. Normally, cells have a diameter of only a few kilometers [103]. Aircraft signals can be received over distances of up to 400 km, limited by the curvature of the Earth, which is an increase of two orders of magnitude compared to those practical cell sizes. And the covered area increases quadratically with the diameter. This means that our method needs far less ground stations than cellular network based localization.

The accuracy of GSM localization is 50 to 150 m outdoors [65], which is a factor of 2 to 6 less accurate than our method. Methods leveraging the wider band LTE should be able to achieve better accuracy than GSM-based methods. Cells sizes are designed for only small overlaps between neighboring cells, in order to allow handing off moving receivers between cells while allowing frequency reuse for a maximal number of cells. However, this can be a problem for localization, as signals from less than the required four antennas may be available. In such cases, no location can be determined.

An advantage of cellular localization methods is that they do not need additional infrastructure, as cellular antennas are already widely available all around the globe. Dense deployments exist in urban areas, which is beneficial for indoor localization, because indoor localization is most required in such areas. However, the cellular antenna coordinates are often not publicly available with sufficient accuracy, limiting the usefulness of cellular localization.

8.1.2 Air Traffic Surveillance and Guidance

Aircraft can be localized and tracked using time-of-flight (ToF) measurements of received messages, such as ADS-B messages. This technique is employed at airports around the world as a cheaper alternative to radar [8]. Using a network of ground stations, which are time-synchronized using GPS receivers, a median aircraft localization error of 128 m has been achieved [112]. Another test series has shown a horizontal localization error of 127 m 95 % of the time [35]. In simulation, it has been shown that the horizontal aircraft localization error could be as low as 11 meters if the geographic distribution of the ground stations is good [64].

Our system also employs a network of ground stations which determine the time and optionally the location of message transmissions by aircraft, but our ground stations do not need to be time-synchronized. However, we do not stop here, but use the determined message transmission locations and times to perform localization of a *handset*. To the best of our knowledge, only Faragher et al. have used aircraft messages for self-localization before (cf. Section 8.1) [42].

Aircraft Signal Receiver Networks

Aircraft tracking systems only need a relatively sparse network of ground stations Nevertheless, setting up a global ground station infrastructure is a practical challenge. At least two companies have taken part in this endeavor by leveraging the participation of hobbyists. However, those companies do not localize users.

FlightAware[11] is a large system of aircraft signal receivers organized by a company with the same name. The network mostly consists of ground stations installed in private homes, sending their data to a FlightAware server. The company provides online instructions, software and hardware to set up a ground station.[12] The ground stations decode aircraft messages and send them together with a timestamp to the company's servers. The aircraft locations are shown on an online map[13] or can be accessed through an API by registered users who contribute data. For tracking aircraft which do not send their location, the servers apparently localize the aircraft based on the timestamps associated with the received messages. However, the exact method is not disclosed to the best of our knowledge and therefore the accuracy is unknown. Based on the fact that the timestamps sent by the ground stations are less accurate than those in our method, we are relatively certain that the accuracy is not good. As we show in Section 8.5.9, even with our accurate timestamps from upsampled signals, determining aircraft locations accurately is difficult. For the purpose of displaying aircraft locations on a map, the locations do not need to be determined accurately. Therefore, for FlightAware's business case, this data is good enough. However, the more stringent accuracy requirements of user localization ask for more accurate aircraft locations. The advantage of FlightAware is their large user base, which gives their network good coverage of many regions. But compared to our method, FlightAware does not localize users.

Flightradar24[14] is another provider of a website displaying current aircraft locations on a map. This website integrates a bunch of additional information about aircraft, such as the aircraft model and technical data, and for commercial aircraft additionally the flight number, origin, destination and the current delay. Flightradar24 also collects data from ground stations installed by hobbyists, but the company also has an active program, in the course of which receivers are sent out to those interested persons, which increase their regional coverage the most.[15] The receivers, which are distributed by the company are expensive and feature a GPS receiver for accurate timing of received messages. Therefore, those receivers have more accurate timestamps than our proposed hardware, at the trade-off of much higher costs. Although we could not find data on the accuracy of the aircraft localization using these receivers, due to the synchronized clocks, the results

[11] https://flightaware.com/

[12] USB SDR ADS-B receiver dongles: http://flightaware.com/adsb/prostick/. PiAware ADS-B ground station software setup guide: http://flightaware.com/adsb/piaware/build

[13] Live map of aircraft currently in the sky: https://flightaware.com/live/

[14] https://www.flightradar24.com/

[15] Flightradar24.com – Apply for receiver: https://www.flightradar24.com/apply-for-receiver

should be good. However, not all of the ground stations in the Flightradar24 network are such expensive receivers. Also, Flightradar24 does not localize users.

Although we are not able to use ground station networks as dense and large as those of FlightAware and Flightradar24, we can still test our method well with a low number of receivers, because aircraft signals can be received over several hundred kilometers. Also, as mentioned in Section 8.2, many aircraft send their location and one ground station receiving a message is enough, because the message transmission time is the only variable.

While FlightAware and Flightradar24 do not provide historical data, OpenSky [109] is an effort to do so, mainly for research purposes. This project also relies on volunteers deploying ground stations and sharing the gathered data. One problem of OpenSky is that the necessary hardware costs about € 700, which is a significant entry barrier. In contrast, our ground stations cost less than $ 100. The project website claims that the network comprises "hundreds of receivers".[16]

8.2 Background

8.2.1 Air Traffic Surveillance

Primary Radar

Air traffic surveillance has emerged as a fundamental requirement for air traffic control. The classic method for localizing aircraft is radar technology. Radar systems send out powerful pulses of electromagnetic signals. When those pulses are reflected at an object, the back-scattered energy can be detected. Given the angle of arrival and the round trip time, the location of that object can be determined. Because the energy of the reflected signal is much weaker than the pulse transmitted from the radar installation, the energy of the radar pulse needs to be high in order to be able to detect the reflected signal. Due to the resulting high energy consumption, radar antennas are tied to the ground as fixed facilities. Today, this classic radar technique is termed the primary surveillance system.

Secondary Surveillance

The messages we employ for our method are part of the secondary surveillance system for air traffic control. Unlike the primary radar, aircraft actively send messages in the secondary surveillance system. Two types of secondary surveillance techniques exists: Aircraft can be "interrogated" by

[16] The OpenSky Network – Services: `https://opensky-network.org/services`

ground stations and respond appropriately, or they simply transmit messages periodically. The former is denominated *Secondary Surveillance Radar* and the latter is called *ADS-B*. Messages may or may not include various types of information, for instance the current location or velocity of the aircraft.

In contrast to the primary radar, the received power of the secondary surveillance signals is higher, because the signals are actively sent by the aircraft. Therefore, the secondary surveillance system can operate at an extended range with the same transmission power. Another advantage of the secondary surveillance system is that not only ground control stations are aware of aircraft and their location in the sky, but all aircraft in the sky can receive the messages sent by other aircraft. This latter functionality is supported by having two separate transmitters on the bottom and the top of the aircraft hull [54, Paragraph 3.1.2.10.4].

Secondary Surveillance Radar The *secondary surveillance radar (SSR)* is an established secondary surveillance system, which detects and identifies aircraft and receives barometric pressure measurement data, from which the flight level of aircraft can be determined. The radio signal uplink from the ground to the aircraft uses a carrier frequency of 1030 MHz and the downlink is at 1090 MHz. The SSR has multiple *interrogation modes* with different message formats. The most important modes are A, C and S. To Mode A interrogations, aircraft reply with their identity. Mode C is used to gather aircraft altitude calculated from the barometric pressure and Mode S supports different message formats, including ADS-B messages [126].

ADS-B A modern secondary surveillance technology is the *Automatic Dependent Surveillance – Broadcast (ADS-B)*. In contrast to the secondary surveillance radar, aircraft with ADS-B transponders send out messages periodically, that is without interrogation. ADS-B is a *dependent* surveillance system because ground stations depend on the aircraft sending out data. ADS-B messages can contain a variety of information. For our system, mainly the location and the velocity of the aircraft are of interest. The location data is always derived from a GPS receiver in the aircraft. Also information used for collision avoidance is transmitted over ADS-B.

ADS-B signals can be sent over different physical links, but the most commonly used is the 1090 MHz channel, because many aircraft already have a transponder for Mode C installed, which operates at this frequency. Mode S comprises a so-called *Extended Squitter* message format, which allows including ADS-B messages in Mode S packets. At the moment, most airliners in Europe and a growing number in North America are equipped with an ADS-B transponder [20, 36].

The data is transmitted at 1 Mbit/s with pulse position modulation. Each packet starts with a preamble of 8 bits followed by 56 or 112 bits of data. The data contains the message type, aircraft address, the actual information (depending on the type) and parity bits for error correction [121]. ADS-B messages are sent in response to an *interrogation* by a ground surveillance station and are additionally also transmitted approximately every second, by choosing a random time interval between 0.8 and 1.2 seconds between two transmissions [54, Paragraph 3.1.2.8.5.2].

8.2.2 Localization

Localization based on ToF measurements is an established technique and used in radio navigation systems such as GPS and LORAN-C. At a handset, signal arrival times from multiple broadcasting transmitters at known locations are determined. Often, the handset is not synchronized with the transmitters which results in a common time bias of all measurements. With signals sent at known times from at least four transmitters, the handset's location (in 3D) and time can be calculated. The system of equations is normally solved using a least-squares approach. But other methods can be used. For instance collective detection (CD) can tolerate more noise than the least-squares method and has been successfully applied to the GPS localization problem [14].

Signal Power Compared to GPS, our method employs signals transmitted from aircraft instead of satellites. Since aircraft travel at an altitude of around 10 km instead of over 20,000 km of the GPS satellites, the transmitters of aircraft are much closer to the user on the ground and result in received signals with higher power. Therefore, our method is more suitable for indoor localization than GPS.

Let us compute the energy advantage of ADS-B signals versus GPS signals. Assuming receivers with equal antenna gain and internal losses, the received signal energy depends on the signal transmission power, the transmission loss and the symbol duration, during which the signal's energy is accumulated in the receiver. With line of sight between transmitter and receiver, the transmission loss is the free-space path loss l_{FS} given by the formula

$$l_{FS} = \left(\frac{4\pi d}{\lambda}\right)^2$$

where λ is the signal wavelength and d is the distance between the transmitter and receiver antennas.

GPS satellites have an effective signal transmission power of 283-458 W [127, Section 2.2]. With an altitude of about 20,200 km and a signal frequency of 1575.42 MHz, the free-space path loss from a GPS satellite to the Earth surface is about 182 dB, according to the formula above. This amounts to a received GPS signal power on the Earth surface between -127 and -125 dBmW.

In contrast, airplanes and other aircraft are flying at an altitude below 13.7 km. The ADS-B standard prescribes a minimum transmission power between 70 and 200 W, depending on the type of aircraft [108, Tables 2-3 and 2-4]. Further, an antenna gain at least equivalent to a quarter-wave resonant antenna minus 3 dB is required [108, Sections 2.2.13.1 and 3.3.1], which we assume means at least 2.19 dB. This results in an effective transmission power of 116—331 W. Due to the curvature of the Earth, aircraft can be in line of sight up to distances of about 400 km, but the ADS-B standard requires only receiving ranges up to 120 nautical miles, which is 222 km [108, Table 3-2]. The ADS-B transmission frequency is 1090 MHz. The free-space path loss is therefore 140 dB at 222 km and 145 dB at 400 km. So, the received ADS-B signal power on the Earth surface is at least -94 dBmW, even if the transmitting aircraft is 400 km away and if it transmits at the lowest specified power of any aircraft type. As an intermediate finding, we can therefore conclude that on the Earth surface, ADS-B signals have a received power advantage over GPS signals of at least 31 dB, which corresponds to a factor of 1,259! At a distance of 100 km with a transmission power of 331 W, the received ADS-B signal power increases to -78 dBmW. The corresponding power advantage over GPS is 47 dB or a factor of 50,119!

GPS is designed with relatively long symbol durations to mitigate this drawback. Effectively, on the Earth surface, each GPS symbol with one millisecond duration has an energy of about -126 dBmW$\cdot$1 ms $= -156$ dBmWs, while each ADS-B pulse lasting half a microsecond has an energy between -94 dBmW$\cdot$0.5 µs $= -157$ dBmWs and -78 dBmW$\cdot$0.5 µs $= -141$ dBmWs. So, each ADS-B symbol has at least comparable energy to a GPS symbol and can be for instance 15 dB or 32 times stronger at an aircraft distance of 100 km. Given that the attenuation of a one foot thick concrete wall is about 14 dB at the ADS-B frequency [122, Table 4], signals from aircraft closer than 100 km may therefore be received in many buildings.

But we can even go a step further: While GPS correlates 1,023 code bits per symbol, ADS-B pulses are usually decoded individually. However, given an ADS-B message decoded by a ground station with good signal reception, another ADS-B receiver could also correlate with all 120 pulses of a message, similar to GPS receivers. Accordingly, that would increase the ADS-B energy by the factor 120 or 21 dB, allowing the penetration of a second concrete wall! While we do not correlate with whole ADS-B messages in this work, that technique should be a straightforward enhancement of our system.

Timing For the ToF method, the transmission time of the messages has to be known. Normally, for instance in GPS, this is achieved with time-synchronized transmitters which include their current time in the messages. None of the SSR or ADS-B messages contain a timestamp. Since the messages travel at the speed of light, even if an aircraft is 300 km away, the propagation time will only introduce a delay of one millisecond. Assuming the delay in the aircraft from the generation of the location to the location transmission is small or compensated, the location of the aircraft will have changed by only a few meters by the time the signal arrives at the ground. For Mode C messages, the height will probably still be the same to within the measurement error. Therefore, for air traffic control, send timestamps are not necessary. But this timing is not good enough for localizing a user, since every millisecond of time error also alters the measured distance to the aircraft by 300 kilometers. Therefore, we use a ground station with known location to determine the transmission time of messages. Multiple ground stations can be used to increase the range of the system. The details of our method are explained in the next section.

8.3 Method

In this section, we show how the challenges explained in the beginning can be addressed. The main idea of our method is to replace GPS satellites with aircraft in order to receive stronger signals, which are more suited for indoor reception than GPS signals. As aircraft messages do not include a timestamp, we also solve a time synchronization problem using a small number of ground stations with known locations. A significant fraction of aircraft transmit their location – which might or might not be accurate. For aircraft with legacy systems (SSR modes A and C), which do not provide their location, our infrastructure also determines the transmission locations of the aircraft messages. In the end, the system we present provides users with a system similar to GPS, but with aircraft instead of satellites. The user localization is done using time-of-flight (ToF) measurements like in GPS.

Figure 8.1 shows the concept of our system. The ground stations and the handset send the recorded ADS-B messages with the corresponding timestamps to a server. The server collects the messages and matches the messages from the handset to those from the ground stations and computes send times of the messages and the location of the handset by solving least-squares

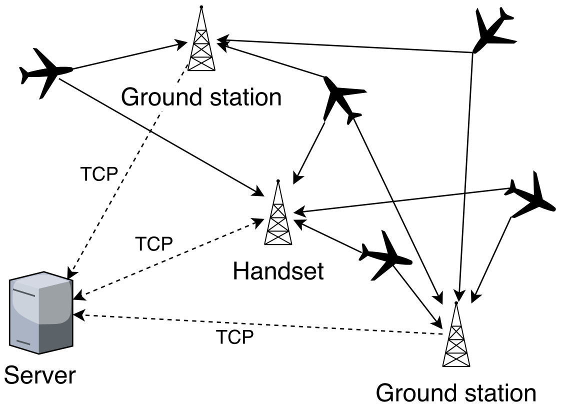


Figure 8.1: System consisting of a server, ground stations at known locations, a handset at unknown location and aircraft. The ground stations and handsets determine the receive time of messages from aircraft to then calculate the send time of the messages and the location of the handset.

problems using the relative timing of these messages and their transmission location. The localization method can be partitioned into two steps:

- Calculation of message send times and aircraft locations (if not given in messages): Matching messages received at multiple receivers are used to calculate the clock offset and drift of the receivers, the message send times and optionally the aircraft locations.

- User localization: The ToFs of the messages are computed from the arrival times and the (now) known send times. The ToFs yield a system of equations which allows finding the handset location.

These individual steps are explained in detail in the next sections.

As explained in the introduction, our localization system consists of three hardware components: unsynchronized ground stations that receive ADS-B or SSR messages from aircraft, a user handset and a server that collects all the received messages.

With this proposed system, the messages from the aircraft can be used for localization without large infrastructure costs. The ground stations and handsets do not need synchronized clocks and work with inexpensive software-defined radios (SDRs). Each receiver only needs little processing power to decode the messages and forward them to the server.

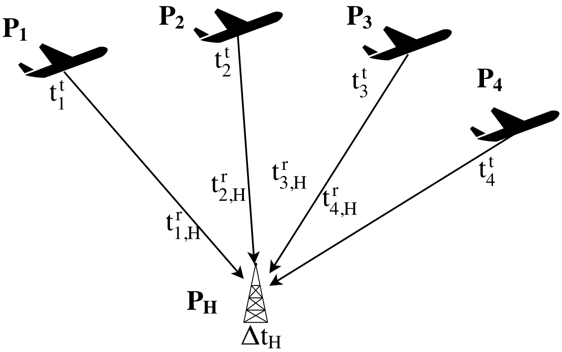


Figure 8.2: Handset localization: Messages received by the handset from multiple aircraft are used for the localization.

8.3.1 User Localization

Let us first discuss the handset localization assuming that the transmission position and time of the received messages are known.

The calculation of the handset location is similar to the localization in global navigation satellite systems, for instance GPS. Instead of satellites with known signal transmission location and time, we use aircraft transmitting ADS-B messages. Figure 8.2 shows the concept. For each received message, we can create a pseudorange equation

$$\|\mathbf{P_H} - \mathbf{P_j}\|_2 + c\Delta t_H \overset{!}{=} c(t^r_{j,H} - w_{j,H} - t^t_j)$$

where $\mathbf{P_j}$ is the location from where the message was sent, t^t_j the send time, $\mathbf{P_H}$ the handset location, Δt_H the clock offset of the handset to the ground stations and $t^r_{j,H}$ the receive time of the message at the handset with the noise $w_{j,H}$. The handset location and clock offset can be found by solving a least-squares problem.

$$\begin{aligned} e_j(\mathbf{P_H}, \Delta t_H) &= \frac{1}{c}\,\|\mathbf{P_H} - \mathbf{P_j}\|_2 + \Delta t_H + t^t_j - t^r_{j,H} + w_{j,H} \\ (\widehat{\mathbf{P_H}}, \widehat{\Delta t_H}) &= \underset{(\mathbf{P_H}, \Delta t_H)}{\arg\min} \sum_j e_j(\mathbf{P_H}, \Delta t_H)^2 \end{aligned}$$

This cost function is minimized using a non-linear least-squares solver. At least four points $\mathbf{P_j}$ are needed to estimate the handset location and time offset. We used the *Levenberg–Marquardt algorithm* which combines the *Gauss-Newton method* with *gradient descent* [46].

Dilution of Precision (DOP) is a concept originating from GPS, but it can also be applied to our problem. When doing localization using ToF measurements, the geometric distribution of the used points (for instance GPS satellites or aircraft and the handset) influences the quality of the location estimation. The location estimation error is the product of the signal arrival time measurement error and the DOP value. A rule of thumb is that the larger the volume spanned by the aircraft, the better the localization precision [69]. And the receiver should be close to the aircraft. We can leverage other advantages of aircraft signals, like higher received signal power or a higher number of available aircraft to reduce the measurement errors and get good localization accuracy.

Due to the aircraft geometry, the vertical localization uncertainty is often larger than the horizontal uncertainty, because most aircraft are at a low elevation angle from the perspective of a receiver on earth. Therefore, a vertical receiver location change results in a smaller change in the measured distance to the aircraft than a horizontal movement. Thus, it is useful to determine the handset height using a different method, for instance with a barometric pressure sensor. This can increase the localization accuracy significantly, as shown in Section 8.5.6. Indoor pressure usually does not differ from outdoor pressure by more than 40 Pa [71,119]. This corresponds to a vertical localization error of about 8 m according to the barometric formula near the earth surface.

8.3.2 Transmission Time and Location

To calculate the location of the user, the transmission time t_j^t of the message j at the aircraft has to be known. By using a ground station with a fixed location, the time of flight from the aircraft to the ground station can be calculated to then compute the transmission time.

To reduce the error of the timestamps and increase the covered area, multiple ground stations can be used, as depicted in Figure 8.3. But the message timestamps at the different ground stations are not synchronized. Moreover, due to small deviations of the receiver oscillators, the sampling rate is varying and not equal at different ground stations. In order to use the time-stamped messages from the different ground stations for the user localization, we have to compensate the clock offsets and drifts.

The receive timestamp t_{j,B_i}^r consists of the message send time at the aircraft (in global time) t_j^t plus the clock offset Δt_{B_i} and clock drift D_{B_i} of

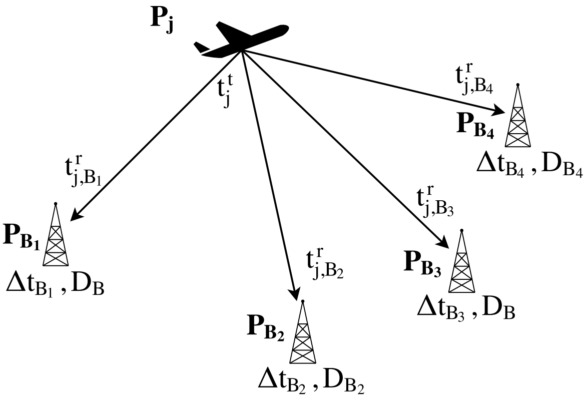


Figure 8.3: Calculation of message send time and aircraft location: Multiple ground stations receive the messages from the aircraft and can calculate the send time of the messages and optionally the location of the aircraft.

the ground station relative to station 1 plus the time of flight of the signal from location $\mathbf{P_j}$ to $\mathbf{P_{B_i}}$.

$$t^r_{j,B_i} - w_{j,B_i} - \frac{1}{c}\left\|\mathbf{P_{B_i}} - \mathbf{P_j}\right\|_2 = t^t_j + \Delta t_{B_i} + D_{B_i}(t^t_j - t^t_1) \tag{8.1}$$

This system of equations can be solved using a linear least-squares solver to compute the transmission times t^t_j of the messages in the synchronized time.

As the clock drift rate and offset may change over time, the synchronization has to be repeated regularly. When a few messages have been received at multiple ground stations, Equation (8.1) is solved. Additionally new clock offsets and drifts for the ground stations are calculated. Now, only the handset has a clock offset which is the same to all ground stations.

Not all messages sent by the aircraft contain the location. To be able to also use those messages in the localization of the handset, we can determine the aircraft location $\mathbf{P_j}$ at the time when the message was sent. In this case, Equation (8.1) is non-linear due to the distance term. To calculate $\mathbf{P_j}$, we therefore use a non-linear least-squares solver.

Ground Station Requirement Before going on, let us discuss the ground station requirement. By the end of the year 2019, all aircraft will be required to support ADS-B [20] and therefore send their location in their messages, and many aircraft already do it today, as mentioned in Section 8.2.1. Since signals can be received several hundred kilometers from an aircraft, in principle one ground station is enough to cover a large urban area or a small country. Using multiple ground stations has several benefits:

- Measurement errors can be reduced.
- In mountainous regions, a ground station should either be deployed on the highest mountain in order to receive aircraft signals from all directions or several receivers can be used, which all see a part of the sky. Note that the different parts need to have some overlap in order to be able to synchronize the ground stations' times.
- Fewer messages are lost and therefore a larger number of the messages of a handset can be matched by the server with a message from a ground station. This increases the number of measurements per time and thus allows tolerating faster moving receivers because a sufficient number of messages for localization can be collected at the handset during a shorter time.

The number of ground stations for a full deployment of our system is orders of magnitude less than for other localization techniques, like for instance cellular network or WiFi-based localization. One ADS-B receiver can cover an area with a radius of several hundred kilometers while cellular network antennas usually transmit over distances up to several kilometers only and WiFi signals can only be received up to a few ten meters from a base station. By leveraging already existing ADS-B receiver network communities (cf. Section 8.1.2), deploying a receiver network for our proposed method might be possible by simply providing updated software for these receivers.

Lastly, note that handsets do not need to be in line of sight of any ground station since the time synchronization of aircraft messages and the location computation of a handset are decoupled.

8.4 Implementation

8.4.1 Receiver

To receive the messages from the aircraft, we use software-defined radios (SDRs). For this project, USB dongles based on the RTL2832U chipset are used, which were originally designed to be DVB-T TV tuners. Therefore, the SDRs are inexpensive. In a second iteration we used FlightAware Pro

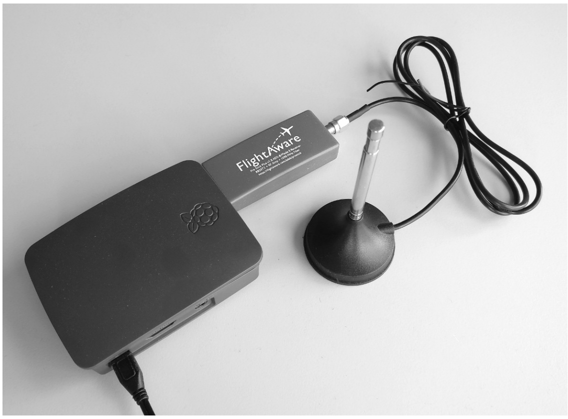


Figure 8.4: Hardware of ground station: Raspberry Pi 3 with FlightAware Pro Stick Plus and antenna. The Raspberry Pi decodes the messages and sends them to the server via LAN or WLAN.

Stick Plus, which is a USB dongle based on the same chipset and has been optimized for the reception of ADS-B. It contains a band-pass filter and an amplifier. These SDRs provide a stream of complex samples (I/Q samples) at a rate of 2.4 MHz. The power consumption of the SDR is at most 1.5 W.[17]

For the processing of the data, each SDR is connected via USB to a Raspberry Pi 3 model B, a single-board computer. The operating system is Raspbian, which is based on Debian. The Raspberry Pi consumes a maximum of 6.7 W under stress.[18]

The detection of the messages is done with dump1090-mutability.[19] It is able to run on the Raspberry Pi 3 without dropping samples. This software generates timestamps with a resolution of 83.3 ns. In order to improve the timestamp precision further, we add 25-fold local upsampling of the messages. The `GPU_FFT`[20] library is used to speed up the message correlation using the GPU.

After detecting the messages, the program sends the detected messages with the corresponding timestamps to the server where they get collected and the localization is performed.

[17] `http://de.flightaware.com/adsb/prostick/`

[18] `https://www.raspberrypi.org/help/faqs/#powerReqs`

[19] `https://github.com/mutability/dump1090`

[20] texttGPU_FFT is an FFT implementation for the BCM2835 SoC GPU found in Raspberry Pis. URL: `http://www.aholme.co.uk/GPU_FFT/Main.htm`

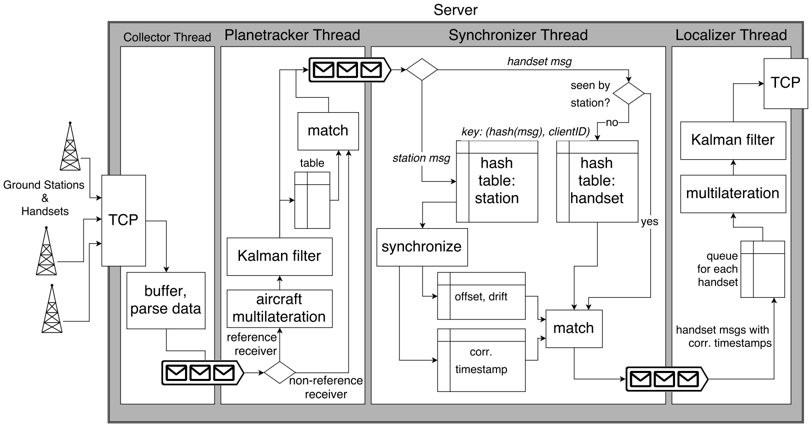


Figure 8.5: Server application: The server receives the messages from the ground stations and handsets, matches the messages from the different receivers. Then the clock offset and drift between the ground stations is computed with messages from different receivers. In the last stage, the localization of the handsets is performed.

Currently the ground stations and the handset are based on identical hardware configurations. A picture of the hardware is shown in Figure 8.4.

Barometric Pressure Sensor

Since the ground stations and the handsets have a similar altitude and the aircraft are close to the horizon, the vertical dilution of precision is much higher than the horizontal dilution of precision. A solution is to equip the handset with a barometric pressure sensor. The altitude value provided by the sensor can be used to perform two-dimensional localization which reduces the number of needed aircraft by one. The sensor has to be calibrated with the barometric pressure on sea level (QFF) which is weather-dependent. This value can be obtained from a weather API or can be calculated at a point with known altitude.

8.4.2 Server

Figure 8.5 shows the overview of the server application. The server consists of four different threads that are connected via message queues.

The *collector* thread accepts TCP-connections from the receivers and parses the received messages.

The *planetracker* thread determines the transmission location of messages that do not contain the location of the aircraft as described in Section 8.3.2. Furthermore, all messages are Kalman filtered using a constant velocity model. The model is updated with the received velocity and location messages and the locations calculated for all other messages. The hash of the messages and the message timestamps are then used to match the messages received at different ground stations and at the handset. The messages then get forwarded to the *synchronizer* thread.

The *synchronizer* thread performs the time synchronization of the ground stations. It looks for corresponding messages that have been received at multiple ground stations and uses them to calculate the time offset and clock drift of the ground stations and the transmission timestamps of the messages. When the transmission time of a message, that has also been received at the handset, has been calculated, it gets forwarded to the *localizer* thread.

The *localizer* thread calculates the handset location using the messages with the calculated transmission timestamp. For each handset, the messages are accumulated in a queue. As soon as it contains enough messages (at least four different aircraft), the localization of the handset is performed. Handsets with known altitude can be located in two dimensions and therefore only need three received messages.

8.5 Results

To evaluate our method, we deployed six ground stations like that shown in Figure 8.4 in a region approximately 110 km in diameter. The locations of the ground stations and the handset can be seen in Figure 8.6. Placing ground stations outdoors, for instance on the roofs of buildings or on hills, would be beneficial in order to maximize received signal energy, number of received messages and observed unique aircraft, as well as to reduce multipath errors. However, building weatherproof cases was outside of the scope of this work. Therefore, the ground stations for our evaluation are placed inside buildings. Note that our preliminary test setup uses ground stations with known, but inaccurate locations, which have been estimated manually by locating the receivers on a map, therefore introducing several meters of error.

In our setup, the received signal is sampled at 2.4 MHz and upsampled by a factor of 25 at the ground stations and the handset. The handset's height is determined using a barometric pressure sensor, so only the latitude and

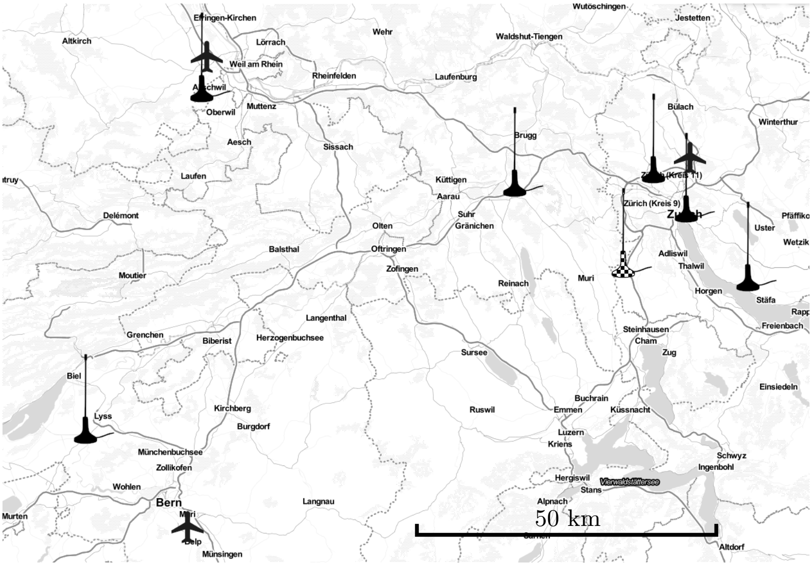


Figure 8.6: Locations of the ground stations (black) and a handset (checkered) for the evaluation. The ground stations span over a region approximately 100 km in diameter. The three international airports in the region are marked with an aircraft icon.

longitude of the handset are computed based on the data decoded from the aircraft signals.

Our evaluation should be considered as a mostly qualitative analysis, since we could not yet test our system performance extensively. Doing so would require measurements spanning a long time, maybe even a year, due to the variability of air traffic. Not only is there a daily cycle of flight patterns and air traffic density, but for instance different wind directions influence the routing of aircraft on different days and there are even seasonal differences, as during holidays, the number of passenger flights increases significantly. Also, aircraft geometry depends on the flight routes assigned in different regions of the world and is not mostly uniform, like for instance the GPS satellites with their nicely distributed orbits. Furthermore, the landscape around the receiver also influences the reception of aircraft signals, for instance by blocking the line of sight between a handset and some aircraft. Further variable parameters, like the geographic distribution of the ground stations, also influence the performance of our system. Finally, note

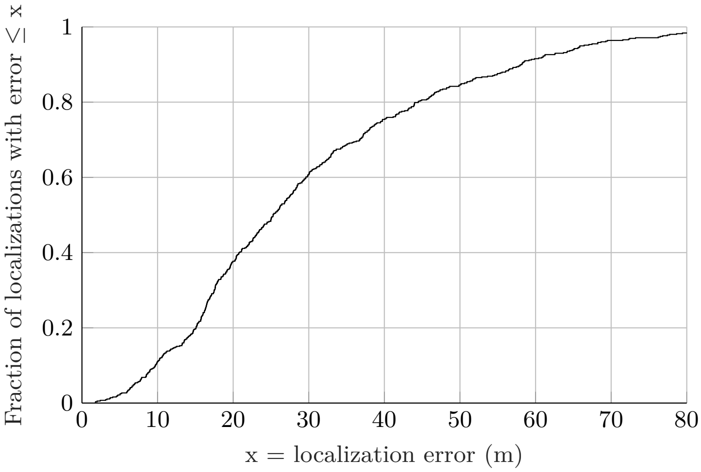


Figure 8.7: Cumulative distribution function of the localization errors of an outdoor handset using data from six ground stations. The handset altitude is known and the horizontal coordinates are determined using the aircraft signal measurements from the ground stations.

that all these parameters are not independent, which makes it challenging to evaluate our system thoroughly.

8.5.1 Reception Quality

The maximum range within which aircraft signals can be received, depends on the antenna characteristics and placement. Our ground stations, which are all located indoors, receive messages from up to approximately 190 km away. When placing the antenna on a roof, the range increases to about 250 km. Comparing the indoor and outdoor cases, we observe that indoors the number of unique aircraft from which signals are received decreases by a quarter. Note that our ground stations have cheap passive antennas. With more expensive active antennas, ideally mounted on a roof, the received signal energy should be higher, resulting in increased signal reception range and more received messages.

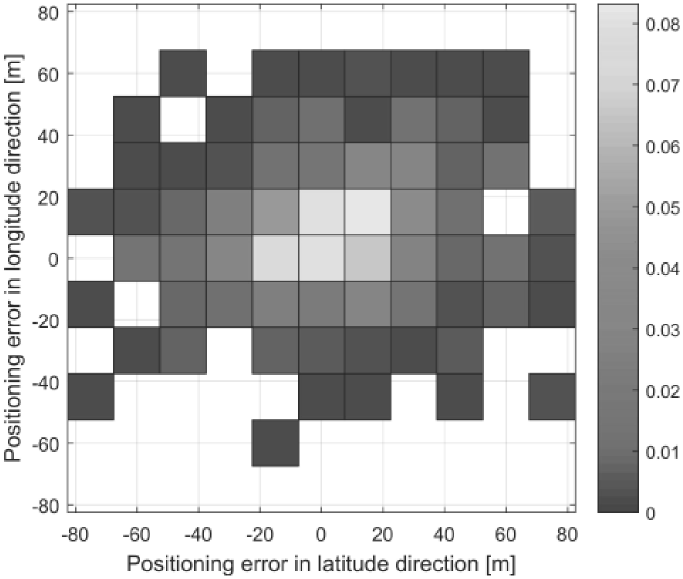


Figure 8.8: Distribution of localization errors in latitude and longitude direction. The handset altitude is known. The results are obtained using six ground stations and a handset outdoors. As expected, for instance with normally distributed errors, the plot is roughly circularly symmetric around a point close to the true location and the density of the location estimates is higher in the middle and fading outwards.

8.5.2 Localization Accuracy

Figure 8.7 shows the localization accuracy of our method using our six deployed ground stations and a handset outdoors. The median error between the computed locations and the ground truth is 25.3 meters and the maximum error is 118.6 meters. The ground truth was estimated using Google Maps,[21] the error of the ground truth should be less than 3 meters.

The results of our measurements are approximately normally distributed. An example distribution of the computed locations around the ground truth can be seen in Figure 8.8.

[21] https://maps.google.com

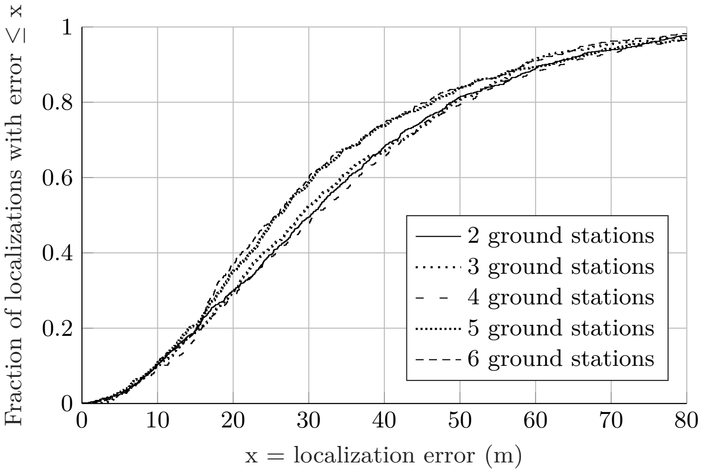


Figure 8.9: Absolute localization error outdoors for different numbers of ground stations. As expected we do not see large differences of the localization error since one ground station is sufficient to calculate the send time of an aircraft message.

8.5.3 Indoor vs. Outdoor Accuracy

We conducted experiments to evaluate the accuracy of our method indoors. Surprisingly, the accuracy indoors is close to the accuracy outdoors. The median error is only 5.6 % larger indoors, and the standard deviation increases by 14.7 % compared to the outdoor case. The cumulative distribution function looks almost identical to the outdoor case. This implies that localization errors are *not* dominated by noise, but rather other error sources such as inaccurate signal transmission locations, multipath environments or inexact signal ToF estimates.

8.5.4 Number of Ground Stations

To test the influence of different parameters, we conducted additional experiments. First, we tested the accuracy using different numbers of ground stations. The results are shown in Figure 8.9. We do not see a large increase in the localization accuracy when more ground stations are used. This is what we expected, since one ground station is sufficient to calculate the send time of a message from an aircraft. One reason for the slightly better results

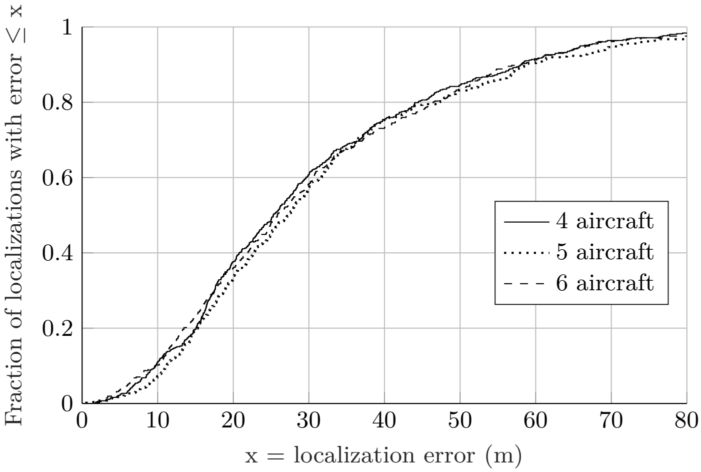


Figure 8.10: Absolute localization error for different numbers of used aircraft. The localization error does not decrease with more aircraft used. We assume that the synchronization error increases with more used aircraft because messages from longer time periods have to be combined for that.

with 5 and 6 ground stations might be that errors in the calculated clock offsets and drifts of the ground stations are averaged out.

8.5.5 Number of Observed Aircraft

Further, we tested the influence of the number of unique aircraft used for a localization solution. Figure 8.10 shows the results from this experiment. When using different numbers of aircraft, the localization error does not change. As in GPS with more satellites, we expected the localization error to decrease with more aircraft. However, when not many aircraft are within range of the handset and the receivers, messages from a longer time period might have to be combined. This can lead to a larger synchronization error between the ground stations.

An alternative hypothesis is that there could be systematic errors in the transmitted aircraft locations. Irrespective of the number of aircraft, this might result in imprecise localization due to considerable residuals in the system of equations solved for the localization. However, since aircraft are usually traveling in different directions, such systematic errors would have to be not relative to an aircraft, but fixed with respect to the Earth, that

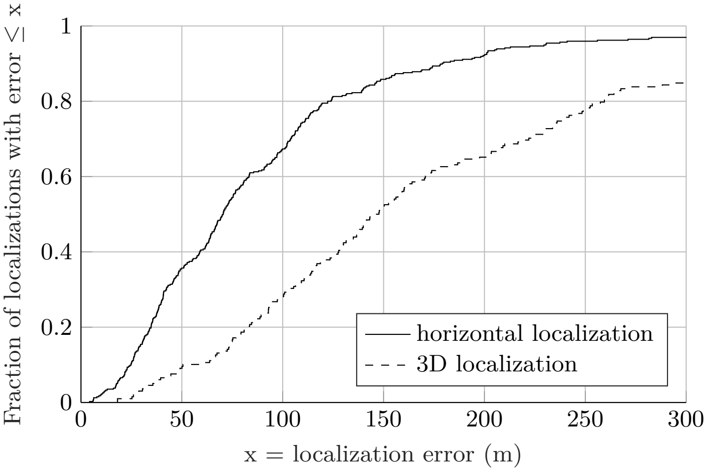


Figure 8.11: Localization error with known handset altitude (horizontal localization) compared to unknown handset altitude (3D localization). These results are from a preliminary experiment with a version of our implementation that did not include all the improvements present in the final version.

is, independent of the aircraft orientation. Otherwise, the errors should approximately cancel out when solving the localization problem. So, for instance slight delays in the aircraft transmitting their locations cannot be the reason for the localization error not decreasing with the number of aircraft.

8.5.6 Known vs. Unknown Altitude

Another interesting experiment evaluates the benefit of using a barometer for the receiver's height estimation. Figure 8.11 shows that the 3D localization error is much higher than the horizontal one with known receiver height. However, an increased error is expected due to an added degree of freedom of the solution. Also, most aircraft are at a low elevation angle from the receiver location. This results in a badly conditioned problem for the height estimation. The problem is that a height change of the handset does not influence the arrival time measurements as much as a horizontal movement. Since we conducted this test early in the development of our method, some improvements present in the final system described in this chapter were still missing. Therefore, the horizontal localization accuracy is also worse than the results shown before. The conclusion of this experiment is that a barometer is a useful feature of a handset localization method based on aircraft signals. Adding the barometer reduces the median error by 52 %.

8.5.7 Upsampling

As mentioned in Section 8.4.1, we upsample the received signal by a factor of 25. Figure 8.12 shows the achieved localization accuracy compared to that which results when using the timestamps of the standard dump1090 software, which determines the phase of the messages by correlating with five fixed patterns. As we understand from analyzing the source code, those patterns used in dump1090 correspond to the expected sample values when shifting the signal by multiples of a fifth of a sample duration. Therefore, the accuracy should be the same as for a five times upsampled signal. However, we observe that the five possible phases do not appear equally likely, which we interpret as an implementation error. Therefore, the results using the dump1090 timestamps might be somewhat worse than theoretically achievable using an optimal phase estimation technique. Note that like in the previous experiment, also the localization results of this test are less accurate than possible with our current implementation, because the results were derived using a preliminary implementation. Figure 8.12 demonstrates that the localization error decreases when the received signal is upsampled. However, the necessary computation power increases substantially. Due to performance limitations of our prototype ground stations featuring a Raspberry Pi 3, we were not able to continuously use 50-fold upsampling. In fact, even when upsampling by a factor of 25, on average it happens every few hours that the processor is overloaded. This manifests itself in dropped samples, because buffers are emptied slower than samples are recorded. The reason why this happens infrequently and unpredictably is that the number of received messages depends on the number of aircraft in the range of the receiver and therefore varies considerably. When the processor has a lot of load, it overheats and at 80°C, the clock rate of the CPU is automatically reduced. The result is that the processor is even less able to cope with all the incoming messages. This problem could probably be resolved by installing a heat sink on the processor. However, if regions are covered by multiple ground stations, short outages of one ground station for a few seconds or minutes can easily be tolerated. If this is not the case and only one ground station is available, its upsampling factor can also be reduced a bit, without resulting in a large increase in localization error, as Figure 8.12 shows.

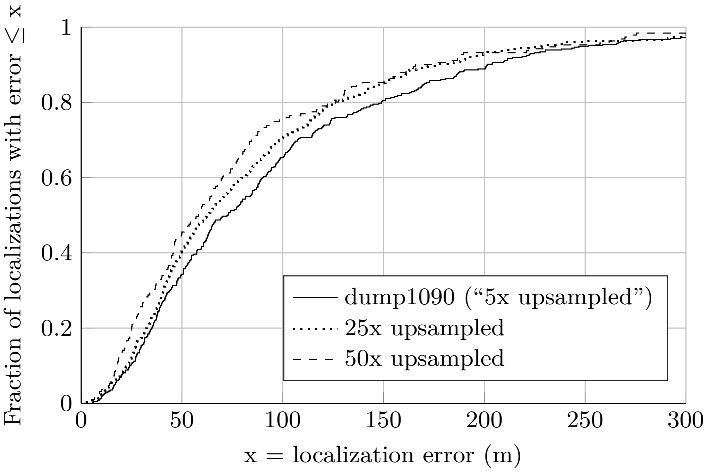


Figure 8.12: Localization error based on 25-fold and 50-fold upsampled timestamps, as well as the default timestamps output by *dump1090.* These results were obtained using two ground stations and also an old version of our implementation, like the results from the barometer experiment.

8.5.8 Error Sources

Multiple possible error sources of the localization exist. An overview is given in Table 8.13. The multipath effect caused by signal reflections from buildings and ground has a big impact, since the ground stations are placed in offices and residential buildings and do not have direct line of sight in all directions. Next, accurate timestamps are essential for the localization. As shown in Section 8.5.7, the timestamps can be improved using upsampling. Since the ground stations do not have synchronized clocks, the estimation of the clock offset and drift has to be repeated regularly. If there are only few aircraft in the sky, to get enough messages for a localization, messages have to be collected over a longer period of time during which the synchronization error accumulates and the timestamps become less accurate. Note that the opposite case, when many aircraft send messages, which then collide and therefore cannot be decoded, is not a problem. In that case, due to the large number of aircraft, also many messages can successfully be decoded. The extreme case in which basically all messages are lost should not occur in practice, since this would also be a threat to air traffic surveillance and guidance. In practice, less than 60 % of all messages are lost on average at

Table 8.13: Localization error sources

- multipath effect
- timestamps
- ground station synchronization
- ground station locations
- geometry (dilution of precision)
- message transmission delays (outdated locations)
- uncompensated distance between GPS and ADS-B antennas on aircraft

any time of the day [120, Figure 3b]. The timestamps are more precise if the signal-to-noise ratio (SNR) is high, in which case the correlation contains an easily distinguishable peak indicating the message arrival time. But this correlation can also deteriorate because of the frequency shift caused by the Doppler effect. Small errors are also introduced by imprecise ground station locations. Depending on the location of the aircraft and the handset, a bad dilution of precision can occur which increases the localization error. An additional error is introduced by the uncompensated latency in the aircraft between the reception of the GPS location and the transmission of the ADS-B message. Also, depending on the aircraft model, the GPS receiver and the ADS-B transponder have a location offset, which may be uncompensated.

8.5.9 Using Messages from Trilaterated Aircraft

As described in Section 8.3.2, the system can also compute the location of aircraft that do not transmit their location. Especially smaller aircraft are not equipped with ADS-B transponders. According to the OpenSky Report 2016 [109], about 70 % of transponders support ADS-B messages. ADS-B will be mandatory by 2020. In that report, it is also mentioned that 26 % of received Mode S messages are ADS-B messages. The most frequent types are altitude replies, which account for 35 % of all messages. As a result, only 2D localization has to be performed, since the altitude is already known.

Figure 8.14 shows the accuracy of the aircraft localization performed with six ground stations. For the evaluation of the location accuracy, the localization is performed on messages from aircraft that also transmit their

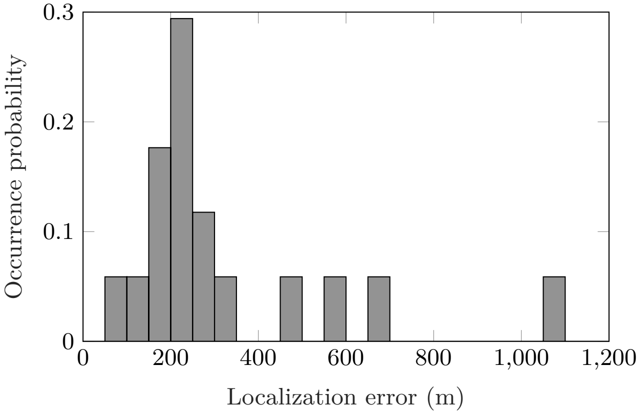


Figure 8.14: Distribution of aircraft localization errors using six ground stations. Errors are computed as the difference between the determined locations and the locations reported by the aircraft themselves in the transmitted messages.

location in other messages. The localization accuracy is evaluated against those reported locations. Therefore, part of the localization error is due to unknown errors in the transmitted aircraft locations. The calculated locations have a median error of approximately 300 m. With a median *handset* localization error of 25 m when using the aircraft with known locations, the calculated aircraft locations can not be used to get more accurate handset locations. But in cases with too few ADS-B equipped aircraft in range, the aircraft localization could still be used to get a rough handset location estimate.

We did not reach a conclusion yet, why these errors are larger than the handset localization errors. Different additional error sources are possible. Among them are:

- Large dilution of precision due to bad geographical distribution of the ground stations.

- Inaccurate ground truth of the location received from the aircraft: unpredictable delay from receiving location to transmitting it and unknown location offset between the GPS antenna and the aircraft transponder.

- Accumulation of synchronization errors between the ground stations.

8.6 Conclusion

We have shown that using aircraft signals to localize users is a viable approach, even when the receiver is indoors. Our method fills a gap between globally available outdoor localization and accurate but expensive or cumbersome indoor localization. A few ground stations are enough to serve a region several hundred kilometers in diameter, which makes our method basically infrastructure free.

To better understand the possibilities and limitations of localization using aircraft signals, a thorough evaluation of the influence of various parameters on the performance, as outlined in Section 8.5, is necessary. For instance, the time of day influences the density of air traffic, and it would be interesting to determine the effects on the area coverage, since even a low number of aircraft can be sufficient for localization.

Plenty of accuracy improvements to our prototype system are possible:

- Using more advanced signal processing to more precisely detect signal arrival times and detect more signals per time.

- Improvements to the RF chain, such as employing antennas designed for ADS-B.

- Applying enhancements to the location estimation, such as selecting only "good" measurements for the least-squares computation, computing a weighted least-squares solution, applying multipath mitigation techniques or using a different localization algorithm such as a maximum likelihood approach.

- Precisely localizing ground station locations.

- Choosing an optimal placement of the ground stations, to reduce error sources such as multipath and to maximize received signal energy, number of received messages and observed unique aircraft.

In the future, a larger number of aircraft will be equipped with ADS-B transponders due to regulatory requirements and growing air traffic, which will increase the availability of our proposed localization method and also improve its accuracy due to more possibilities for error correction.

The presented handset design shows that our method could be integrated in a smartphone. The only additional hardware required in a smartphone is a small antenna, which easily fits into such a form factor, and a few components for the RF front end. Given the usual level of system-on-chip integration, this should be an inexpensive addition.

9

Conclusion

> *"The vain presumption of understanding everything can have no other basis than never having understood anything. For anyone who had ever experienced just once the perfect understanding of one single thing, and had truly tasted how knowledge is accomplished, would recognize that of the infinity of other truths he understands nothing."*
>
> — Galileo Galilei, "Father of Science"

Due to its wide availability and meter-level accuracy, GPS is the current standard for global localization. Other GNSS exist or are in development, increasing the number of available satellites and extending the area coverage for GNSS localization. Also, more visible satellites improve the total received signal energy. Still, GNSS receivers draw relatively much power and work poorly indoors or between obstacles such as houses, mountains or even trees.

The *collective detection (CD)* method presented in this book mitigates both the energy consumption and the signal multipath problem. Multipath effects from reflected, and therefore delayed, signals are reduced by combining the signal power from all satellites. Only signals with equal delay in addition to the expected direct path delay align in the maximum likelihood formulation. This makes the GNSS localization robust, since in most environments, the direct signals outnumber similarly delayed reflections. On the energy side, CD enables computing a receiver's location from as little as one millisecond of recorded signals, compared to at least several seconds for classical receivers. Since those short signals generate only a few kilobytes of data, such signal snapshots can be stored for later processing. Saving the location computation hardware, which is the most energy-consuming part of a classical GPS receiver, greatly reduces the energy consumption of a snapshot receiver. Combined, the effects of shorter signals and offloading the location computation to an external device, such as a server in the cloud, reduces the GNSS receiver's energy consumption by three orders of magnitude.

Using a branch-and-bound algorithm to compute the localization solution makes CD viable for large search areas and high numbers of localization requests. The location probability distribution underlying CD receivers does not only help in the presence of multipath signals. The location likelihood function also reveals malicious signals from attackers trying to spoof the receiver. Over time, simultaneously present legitimate and fake signals can be discerned using external information such as accelerometer readings or a map. With more and more GNSS spoofing attacks becoming public, such as in Russia [37], GNSS spoofing mitigation techniques are rising in importance. The spoofing problem is only aggravated by the world's infrastructure increasingly depending on GNSS, as we have seen in Chapter 1.

GNSS snapshot receiver hardware can be built small, even if not designed as an ASIC (application-specific integrated circuit). Our hand-soldered, PCB-based snapshot GPS receiver has an area of 23 mm×14 mm and weighs only 1.3 g without the coin cell battery. The flash chip with 2 Gb storage space allows capturing 65600 snapshots with a length of one millisecond each. With quarter-hourly localization, this allows for a lifetime of 683 days. During these almost two years, the tracker will consume approximately a quarter of a CR2032 coin cell battery's capacity, leaving plenty of margin for self-discharge.

Although the robustness improvements of CD over classical least-squares localization are impressive, indoor localization with GNSS is still inadequate. Depending on the required accuracy for a certain application, it may be sufficient to use the last location before the receiver entered a building, for instance. If better accuracy is needed, one approach is to equip buildings with extra localization infrastructure, such as Bluetooth beacons. However, a common approach to save installation and maintenance costs is to use so-called *signals of opportunity*, whose main purpose is not localization, but which are already present. For instance, smartphones rely on matching observed Wi-Fi signals to known Wi-Fi hotspot locations to localize themselves.

An alternative localization method using ADS-B signals sent from most aircraft opens up the possibility for better area coverage than using only short-range signals such as Wi-Fi signals. A small number of ground stations are needed for each region with a diameter of a few hundred kilometers, in order to determine the transmisstion timestamps of the aircraft messages. With this information, receivers can be localized using the same principle employed by GNSS. Our preliminary experiments show localization accuracies of some tens to hundreds of meters. Although this suffices for some applications, further system improvements should increase the number of application scenarios substantially. For instance, the time resolution of the signal sampling can be improved for better message discovery probability. Also, a maximum likelihood method, such as CD, would help rejecting multipath signals. Apart from the algorithmic improvements, more ground stations can be deployed for large scale testing and a thorough indoor evaluation should be done to characterize the advantages of this localization system.

Apart from a commercial adoption of the presented localization systems, a vision for the future is a combination of multiple systems in a single receiver. In the industry, signals from all GNSS are already being joined for more accurate and robust localization. Replacing the least-squares localization algorithms with CD could further enhance the solution quality.

Moreover, the general form of the location probability distribution in CD allows for an easy integration of different localization systems and sensor data. Combining several systems to compute a single maximum likelihood solution should ensure smooth transitions between indoor and outdoor localization and a fail-safe handling of system outages, such as a loss of GNSS signals between buildings, apart from improved accuracy and robustness. GNSS observations could be combined with measurements of signals of opportunity, including Wi-Fi, Bluetooth, ADS-B, cellular, digital TV (DVB-T) and digital radio (DAB) signals, and with sensors like barometers and inertial measurement units (IMUs), consisting of accelerometers, gyroscopes and magnetometers. Working towards this goal, each signal type poses individual challenges. For instance, the exact locations of cellular ground stations are usually unknown and have to be determined first. Possibly, this could be achieved through crowdsourcing time-of-flight measurements from GPS-localized handsets. And DVB-T antennas all send exactly the same signal, which makes it difficult to assign received signals to their originating antenna. Using some prior knowledge about the coarse location of the receiver can help in this case. Generally, a fusion of localization systems should be suited for a variety of situations.

From a broad perspective, there is currently no single best localization system. Rather, the choice depends on the application. The Holy Grail of localization would be a system which is cheap, produces accurate, robust, fast and secure localization everywhere on Earth and whose receivers consume little energy. One could try to build a localization system with many, but cheap, transmitters. However, unless the deployment and maintenance of these transmitters can be automated, such a system will be costly. It seems more realizable to use a system with long-range signals, in order to cover large areas with little infrastructure, such that the cost can be amortized over many users. However, due to the inverse-square law, there is a trade-off between large area coverage and indoor penetration of the signals. While one GNSS satellite covers a large part of the Earth surface, its signals can barely be received indoors. Meanwhile, cellular signals only cover distances from a few hundred meters to some tens of kilometers, but are usually not blocked by buildings. While ADS-B signals from aircraft have a slightly longer range of multiple hundred kilometers, ADS-B signals still seem to reach indoor environments. Therefore, aircraft signals might be the sweet spot in the coverage versus indoor reception trade-off.

Let us recapitulate the state of the art in global localization, considering the goals above. GNSS is cheap and accurate. Snapshot receivers can determine their location fast with only a few milliseconds of signal – therefore being low-power – and CD makes the localization robust and relatively secure, especially in combination with other sensors. By offloading the location computation, snapshot receivers also enable low-power GNSS localization. The missing piece is the ubiquitous availability, with indoor and other obstructed environments being insufficiently covered by GNSS. ADS-B localization seems to be a fitting complement: The most and largest buildings are situated in urban areas, which usually have airports close-by. In those regions, the availability of aircraft for ADS-B localization should therefore be good. While our prototype ADS-B localization system uses commercial-off-the-shelf (COTS) hardware and elementary software, an optimized implementation should also bring us closer to the other goals. The receiver price is low and accuracy and robustness improvements are outlined above. With thousands of messages being received per second at our deployed ground stations, even a snapshot receiver approach, enabling localization from less than a second of recorded signals, should be feasible. That may require using signals which do not contain their transmission location. With at least four ground stations receiving the same signal, the transmission location can however be reconstructed. The same localization technique also helps identifying and eliminating spoofed signals, contributing to the security aspect of the localization. Hence, there is hope that a close-to-ideal localization system may emerge in the future, using a combination of GNSS and ADS-B signals, or some other signals of opportunity.

Bibliography

[1] D. Akos. *A Software Radio Approach to Global Navigation Satellite System Receiver Design.* PhD thesis, Ohio University, Athens, OH, 1997.

[2] Z. Altamimi, P. Rebischung, L. Métivier, and X. Collilieux. Itrf2014: A new release of the international terrestrial reference frame modeling nonlinear station motions. *Journal of Geophysical Research: Solid Earth*, 121(8):6109–6131, 2016.

[3] M. Altini, D. Brunelli, E. Farella, and L. Benini. Bluetooth Indoor Localization with Multiple Neural Networks. In *5th IEEE International Symposium on Wireless Pervasive Computing*, ISWPC, pages 295–300. IEEE, 2010.

[4] W. J. Andrewes. A Chronicle Of Timekeeping, February 2006. `https://www.scientificamerican.com/article/a-chronicle-of-timekeeping-2006-02/`, visited 22 April 2019.

[5] Anonymous. The History of Timekeeping, March 2011. `http://www.beaglesoft.com/maintimehistory.htm`, visited 22 April 2019.

[6] Anonymous. Timing and Synchronization for LTE-TDD and LTE-Advanced Mobile Networks. Technical report, Microsemi, 2014. `https://www.microsemi.com/document-portal/doc_download/133615-timing-sync-for-lte-tdd-lte-a-mobile-networks`.

[7] E. F. Arias. Bureau International des Poids et Mesures (BIPM) – Time Department. In *Report of the International Association of Geodesy 2011-2013 – Travaux de l'Association Internationale de Géodésie*

2011-2013, volume 38, München, Germany, 2013. International Association of Geodesy. `https://iag.dgfi.tum.de/fileadmin/IAG-docs/Travaux2013/08_BIPM.pdf`, visited 30 September 2019.

[8] E. a.s. Multilateration: Executive Reference Guide. online, March.

[9] I. S. Association et al. 1588-2008—ieee standard for a precision clock synchronization protocol for networked measurement and control systems. *International Standard*, 2008.

[10] A. M. Aurand, J. S. Dufour, and W. S. Marras. Accuracy map of an optical motion capture system with 42 or 21 cameras in a large measurement volume. *Journal of Biomechanics*, 58:237–240, 2017.

[11] P. Axelrad, B. K. Bradley, J. Donna, M. Mitchell, and S. Mohiuddin. Collective Detection and Direct Positioning Using Multiple GNSS Satellites. *Navigation*, 58(4):305–321, 2011.

[12] J. Beutel, S. Gruber, S. Gubler, A. Hasler, M. Keller, R. Lim, I. Talzi, L. Thiele, C. Tschudin, and M. Yücel. The PermaSense Remote Monitoring Infrastructure. In *Proceedings of the International Snow Science Workshop Davos (ISSW 09 Europe)*, volume 9, pages 187–191. Swiss Federal Institute for Forest, Snow and Landscape Research WSL, Birmensdorf, Switzerland, September 2009.

[13] P. Bissig. *Mobile Sensing: GPS Localization, WiFi Mapping, Applications, and Risks.* PhD thesis, ETH Zurich, 2017.

[14] P. Bissig, M. Eichelberger, and R. Wattenhofer. Fast and Robust GPS Fix Using One Millisecond of Data. In *Proceedings of The 16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), Pittsburgh, Pennsylvania, USA*, IPSN 2017, pages 223–234. IEEE, April 2017.

[15] K. Borre, D. M. Akos, N. Bertelsen, P. Rinder, and S. H. Jensen. A Software-Defined GPS and Galileo Receiver. *Fourier Analysis and Convexity*, pages 83–96, 2004.

[16] M. Bouet and A. L. Dos Santos. RFID Tags: Positioning Principles and Localization Techniques. In *1st IFIP Wireless Days*, WD 2008, pages 1–5. IEEE, November 2008.

[17] E. S. Bridge, J. F. Kelly, A. Contina, R. M. Gabrielson, R. B. MacCurdy, and D. W. Winkler. Advances in tracking small migratory birds: a technical review of light-level geolocation. *Journal of Field Ornithology*, 84(2):121–137, 2013.

[18] C. Brignone, T. Connors, G. Lyon, and S. Pradhan. Smartlocus: An autonomous, self-assembling sensor network for indoor asset and systems management. Technical report, Mobile Media Syst. Lab., HP Laboratories, Palo Alto, CA, June 2005.

[19] A. Broumandan, A. Jafarnia-Jahromi, and G. Lachapelle. Spoofing detection, classification and cancelation (sdcc) receiver architecture for a moving gnss receiver. *GPS Solutions*, 19(3):475–487, 2015.

[20] P. Bunce. The Clock is Ticking on ADS-B Equipage, April 2016. `http://www.aviationpros.com/article/12180962/the-clock-is-ticking-on-ads-b-equipage`, visited 11 April 2017.

[21] C. Bureau. Polar motion. Technical report, Federal Agency for Cartography and Geodesy, 2013. `https://www.iers.org/IERS/EN/Science/EarthRotation/PolarMotion.html`, visited 30 September 2019.

[22] M. Burgess. When a tanker vanishes, all the evidence points to Russia, 2017.

[23] A. Cavaleri, B. Motella, M. Pini, and M. Fantino. Detection of spoofed gps signals at code and carrier tracking level. In *Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC), 2010 5th ESA Workshop on*, pages 1–6. IEEE, 2010.

[24] S. S. Chawathe. Low-Latency Indoor Localization Using Bluetooth Beacons. In *12th International IEEE Conference on Intelligent Transportation Systems*, ITSC 2009, pages 1–7. IEEE, 2009.

[25] M. Y. Chen, T. Sohn, D. Chmelev, D. Haehnel, J. Hightower, J. Hughes, A. LaMarca, F. Potter, I. Smith, and A. Varshavsky. Practical Metropolitan-Scale Positioning for GSM Phones. In *International Conference on Ubiquitous Computing*, UbiComp 2006, pages 225–242. Springer, 2006.

[26] J. W. Cheong, J. Wu, A. G. Dempster, and C. Rizos. Efficient Implementation of Collective Detection. In *IGNSS Symposium*, pages 15–17. International Global Navigation Satellite Systems Society, 2011.

[27] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan. Indoor Localization Without the Pain. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, MobiCom 2010, pages 173–184. ACM, 2010.

[28] P. Closas, C. Fernández-Prades, and J. A. Fernández-Rubio. Maximum likelihood estimation of position in gnss. *IEEE Signal Processing Letters*, 14(5):359–362, 2007.

[29] P. Closas and A. Gusi-Amigo. Direct position estimation of gnss receivers: Analyzing main results, architectures, enhancements, and challenges. *IEEE Signal Processing Magazine*, 34(5):72–84, 2017.

[30] E. Commission. Commission Implementing Regulation (EU) No 1207/2011. *Official Journal of the European Union*, 305:35–52, 2011. `https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2011:305:0035:0052:EN:PDF`.

[31] E. Commission. Commission Delegated Regulation (EU) 2017/574 of 7 June 2016 supplementing Directive 2014/65/EU of the European Parliament and of the Council with regard to regulatory technical standards for the level of accuracy of business clocks. *Official Journal of the European Union*, L87:148–151, 2017.

[32] D. W. . Committee. Department of Defense World Geodetic System 1984, Its Definition and Relationships with Local Geodetic Systems. Technical report, The Defense Mapping Agency, Fairfax, VA, USA, 1991. DMA TR 8350.2, `https://apps.dtic.mil/dtic/tr/fulltext/u2/a280358.pdf`, visited 30 September 2019.

[33] J. C. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, J. J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild, et al. Spanner: Google's globally distributed database. *ACM Transactions on Computer Systems (TOCS)*, 31(3):8, 2013.

[34] S. Corporation. Application Note: Time Synchronization's Role in Real-Time Trading, May 2012. `https://www.orolia.com/sites/default/files/document-files/Real-Time-Trading_AN02-101C.pdf`, visited 22 April 2019.

[35] A. Daskalakis and P. Martone. A Technical Assessment of ADS-B and Multilateration Technology in the Gulf of Mexico. In *Proceedings of the 2003 IEEE Radar Conference*, pages 370–378. IEEE, 2003.

[36] J. Davidson. ADS-B Requirements Coming Into Effect, September 2013. `http://www.universalweather.com/blog/2013/09/ads-b-requirements-coming-into-effect/`, visited 11 April 2017.

[37] D. De Luce. Russia 'spoofing' GPS on vast scale to stop drones from approaching Putin, report says, March 2019. `https://www.nbcnews.com/news/vladimir-putin/russia-spoofing-gps-vast-scale-stop-drones-approaching-putin-report-n987376`, visited 22 April 2019.

[38] Department of Defense, United States of America. Global Positioning System Standard Positioning Service Performance Standard (GPS SPS PS), 4th Edition. Technical report, September 2008.

[39] M. Eichelberger, K. Luchsinger, S. Tanner, and R. Wattenhofer. Indoor Localization with Aircraft Signals. In *15th ACM Conference on Embedded Networked Sensor Systems (SenSys), Delft, The Netherlands*, November 2017.

[40] M. Eichelberger, F. von Hagen, and R. Wattenhofer. Multi-year gps tracking using a coin cell. In *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications (HotMobile), Santa Cruz, California, USA*, pages 141–146. ACM, February 2019.

[41] Energizer. Product Datasheet: Energizer CR2032, 2018. `http://data.energizer.com/pdfs/cr2032.pdf`, visited 19 October 2018.

[42] R. Faragher, P. F. MacDoran, and M. B. Mathews. Spoofing mitigation, robust collision avoidance, and opportunistic receiver localisation using a new signal processing scheme for ads-b or ais. In *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014). Tempa, Florida, USA*, pages 858–868, 2014.

[43] M. Fontaine, editor. *Comptes rendus des séances de la quinzième conférence générale des poids et mesures*, Paris, France, 1975. Bureau International des Poids et Mesures.

[44] N. T. S. T. Force. Time Synchronization in the Electric Power System. Technical report, North American Synchrophasor Initiative, March 2017.

[45] M. Foucras, B. Ekambi, F. Bacard, O. Julien, C. Macabiau Optimal, and C. Macabiau. Optimal GNSS Acquisition Parameters when Considering Bit Transitions. 2014.

[46] B. S. Garbow, K. E. Hillstrom, and J. J. More. Documentation for MINPACK Subroutine LMDIF – Double precision version, March 1980. `https://www.math.utah.edu/software/minpack/minpack/lmdif.html`, visited 10 April 2017.

[47] B. C. Geiger and C. Vogel. Influence of Doppler Bin Width on GPS Acquisition Probabilities. 2013.

[48] M. Goldstein, J. Kirschbaum, S. Moino, G. Davis, E. Albagli, M. Bodeau, K. Davis, R. Hung, B. Japikse, S. Moessbauer, J. Ormond, N. Padilla, and D. Rodriguez. GPS DISRUPTIONS – Efforts to Assess Risks to Critical Infrastructure and Coordinate Agency Actions Should Be Enhanced. Technical report, United States Government Accountability Office, November 2013.

[49] U. F. Government. Title 14 – Aeronautics and Space. *Code of Federal Regulations*, 2018. `https://www.ecfr.gov/cgi-bin/text-idx?SID=8137158693744ba666e318c1f474d81b&node=se14.2.91_1225&rgn=div8`.

[50] P.-P. Grassé and A. Couder, editors. *Comptes rendus des séances de la treizième conférence générale des poids et mesures*, Paris, France, 1967. Bureau International des Poids et Mesures.

[51] E. Gülch. Investigations on Google Tango Development Kit for Personal Indoor Mapping. In *The 19th AGILE International Conference on Geographic Information Science*, AGILE 2016, page 3, June 2016. Poster Abstract.

[52] T. E. Humphreys, J. a. Bhatti, D. P. Shepard, and K. D. Wesson. The Texas Spoofing Test Battery : Toward a Standard for Evaluating GPS Signal Authentication Techniques. *Proceedings of the 25th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2012) September 17 - 21, 2012 Nashville Convention Center, Nashville, Tennessee Nashville, TN*, (1):3569 – 3583, 2012.

[53] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, B. W. O'Hanlon, and P. M. Kintner. Assessing the spoofing threat: Development of a portable gps civilian spoofer. In *Radionavigation Laboratory Conference Proceedings*, 2008.

[54] I. C. A. O. (ICAO). Aeronautical Telecommunications - Surveillance and Collision Avoidance Systems. Annex 10 to the Convention on International Civil Aviation. Volume IV. July 2014.

[55] N. N. Imagery and M. Agency). Department of defense world geodetic system 1984: Its definition and relationships with local geodetic systems. 1997.

[56] S. E. . Integration. Navstar GPS Space Segment/Navigation User Interfaces. *Global Positioning Systems Directorate*, page 213, 2013.

[57] R. Ishida. Localization vs. Internationalization. W3C Internationalization (I18n) Activity. `https://www.w3.org/International/questions/qa-i18n`, visited 10 April 2019.

[58] W. Jackson. The serious side of GPS, where timing is everything, Nov 2013. `https://gcn.com/articles/2013/11/12/gps-timing-position.aspx`.

[59] A. Jafarnia-Jahromi, T. Lin, A. Broumandan, J. Nielsen, and G. Lachapelle. Detection and mitigation of spoofing attacks on a vector-based tracking gps receiver. *ION ITM*, 2012.

[60] Z. Jia. A type of collective detection scheme with improved pigeon-inspired optimization. *International Journal of Intelligent Computing and Cybernetics*, 9(1):105–123, 2016.

[61] A. R. Jimenez, F. Seco, C. Prieto, and J. Guevara. A Comparison of Pedestrian Dead-Reckoning Algorithms using a Low-Cost MEMS IMU. In *IEEE International Symposium on Intelligent Signal Processing*, WISP 2009, pages 37–42. IEEE, 2009.

[62] T. A. Johnson and P. Seeling. Localization Using Bluetooth Device Names. In *Proceedings of the thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc 2012, pages 247–248. ACM, 2012.

[63] T. Jones. *Splitting the Second: The Story of Atomic Time.* Institute of Physics, 2000.

[64] R. Kaune, C. Steffes, S. Rau, W. Konle, and J. Pagel. Wide Area Multilateration using ADS-B Transponder Signals. In *15th International Conference on Information Fusion*, FUSION, pages 727–734. IEEE, 2012.

[65] T. Kos, M. Grgic, and G. Sisul. Mobile User Positioning in GSM/UMTS Cellular Networks. In *48th International Symposium ELMAR-2006 focused on Multimedia Signal Processing and Communications*, pages 185–188. IEEE, June 2006.

[66] B. Krach, M. Lentmaier, and P. Robertson. Joint bayesian positioning and multipath mitigation in gnss. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 3437–3440. IEEE, 2008.

[67] M. Kumar. The “Real” Definition of “ITRF”. *Coordinates*, June 2007. `https://mycoordinates.org/the-real-definition-of-itrf/`, visited 30 September 2009.

[68] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta. Luxapose: Indoor Positioning with Mobile Phones and Visible Light. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, MobiCom 2014, pages 447–458. ACM, September 2014.

[69] R. B. Langley. Dilution of Precision. *GPS World*, 10(5):52–59, May 1999.

[70] P. Lazik and A. Rowe. Indoor Pseudo-ranging of Mobile Devices using Ultrasonic Chirps. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 99–112. ACM, 2012.

[71] V. Leivo, M. Kiviste, A. Aaltonen, M. Turunen, and U. Haverinen-Shaughnessy. Air Pressure Difference between Indoor and Outdoor or Staircase in Multi-family Buildings with Exhaust Ventilation System in Finland. *Energy Procedia*, 78:1218–1223, 2015.

[72] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao. Epsilon: A Visible Light Based Positioning System. In *Proceedings of the 11th USENIX Symposium on Networked Systems Design and Implementation*, NSDI 2014, pages 331–343, April 2014.

[73] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of Wireless Indoor Positioning Techniques and Systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1067–1080, 2007.

[74] J. Liu, B. Priyantha, T. Hart, Y. Jin, W. Lee, V. Raghunathan, H. S. Ramos, and Q. Wang. CO-GPS: Energy Efficient GPS Sensing with Cloud Offloading. *IEEE Transactions on Mobile Computing*, 15(6):1348–1361, 2016.

[75] J. Liu, B. Priyantha, T. Hart, H. Ramos, A. A. Loureiro, and Q. Wang. Energy Efficient GPS Sensing with Cloud Offloading. In *Proceedings of the 10th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pages 85–98. ACM, ACM, November 2012.

[76] S. Lo, D. De Lorenzo, P. Enge, D. Akos, and P. Bradley. Signal Authentication: A Secure Civil GNSS for Today. *Inside GNSS*, 4(5):30–39, 2009.

[77] M. Lombardi. Delivering NIST Time to Financial Markets Via Common-View GPS Measurements, 2015.

[78] M. Lombardi. Time and Frequency Traceability in Emerging Technologies: Synchronizing Financial Markets, October 2017.

[79] M. Lombardi. Time and Frequency Traceability in Emerging Technologies: Synchronizing Financial Markets, October 2017. `https://tf.nist.gov/sim/2017_Seminar/SIM_2017_Time_in_Financial_Markets.pptx`.

[80] G. López-Risueño and G. Seco-Granados. Cn/sub 0/estimation and near-far mitigation for gnss indoor receivers. In *2005 IEEE 61st Vehicular Technology Conference*, volume 4, pages 2624–2628. IEEE, 2005.

[81] D. Lymberopoulos. Microsoft Indoor Localization Competition. online, April 2016. `https://www.microsoft.com/en-us/research/event/microsoft-indoor-localization-competition-ipsn-2016/#official-results`, visited 09 August 2019.

[82] P. H. Madhani, P. Axelrad, K. Krumvieda, and J. Thomas. Application of successive interference cancellation to the gps pseudolite near-far problem. *IEEE Transactions on Aerospace and Electronic Systems*, 39(2):481–488, 2003.

[83] N. Mair and Q. H. Mahmoud. A collaborative bluetooth-based approach to localization of mobile devices. In *8th International Conference on Collaborative Computing: Networking, Applications and Worksharing*, CollaborateCom, pages 363–371. IEEE, 2012.

[84] S. Malys, J. H. Seago, N. K. Pavlis, P. K. Seidelmann, and G. H. Kaplan. Why the greenwich meridian moved. *Journal of Geodesy*, 89(12):1263–1272, 2015.

[85] R. Mautz. Overview of current indoor positioning systems. *Geodezija ir kartografija*, 35(1):18–22, 2009.

[86] D. D. McCarthy and G. Petit. Iers conventions (2003). Technical report, International Earth Rotation And Reference Systems Service (IERS)(Germany), 2004.

[87] D. L. Mills. Network Time Protocol (NTP). RFC 958, RFC Editor, September 1985. `https://www.rfc-editor.org/rfc/rfc958.txt`, visited 30 September 2019.

[88] G. Milner. Death by GPS, 2016.

[89] M. Morgenthaler. «Meine Frau nannte mich an guten Tagen einen Träumer». *Tages-Anzeiger*, February 2019.

[90] I. national de l'information géographique et forestiére (IGN). Science background – General concepts, January 2016. `http://itrf.ensg.ign.fr/general.php`, visited 22 April 2019.

[91] G. S. Navstar. Standard Positioning Service. page 46, 1995.

[92] A. Neri, F. Rispoli, and P. Salvatori. An analytical assessment of a gnss-based train integrity solution in typical ertms level 3 scenarios. In *Proc. Eur. Navigat. Conf.(ENC)*, 2015.

[93] Y. Ng and G. X. Gao. Mitigating jamming and meaconing attacks using direct gps positioning. In *Position, Location and Navigation Symposium (PLANS), 2016 IEEE/ION*, pages 1021–1026. IEEE, 2016.

[94] C. E. Noll. The Crustal Dynamics Data Information System: A resource to support scientific analysis using space geodesy. *Advances in Space Research*, 45(12):1421–1440, 2010.

[95] G. Oberholzer, P. Sommer, and R. Wattenhofer. SpiderBat: Augmenting Wireless Sensor Networks with Distance and Angle Information. In *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*, IPSN 2011, pages 211–222. IEEE, April 2011.

[96] B. W. Parkinson. Origins, Evolution, and Future of Satellite Navigation. *Journal of Guidance, Control, and Dynamics*, 20(1):11–25, 1997.

[97] P. H. Pathak, X. Feng, P. Hu, and P. Mohapatra. Visible Light Communication, Networking, and Sensing: A Survey, Potential and Challenges. *IEEE Communications Surveys & Tutorials*, 17(4):2047–2077, 2015.

[98] J. Paul. History of the Prime Meridian – Past and Present. November 1999. `http://gpsinformation.net/main/greenwich.htm`, visited 30 September 2019.

[99] N. Pavlis. EGM2008 - WGS 84 Version. Technical report, National Geospatial-Intelligence Agency (NGA), USA, 2012. `https://earth-info.nga.mil/GandG/wgs84/gravitymod/egm2008/egm08_wgs84.html`, visited 30 September 2019.

[100] A. Pinker and C. Smith. Vulnerability of the gps signal to jamming. *GPS Solutions*, 3(2):19–27, 1999.

[101] M. L. Psiaki and T. E. Humphreys. GNSS Spoofing and Detection. *Proceedings of the IEEE*, 104(6):1258–1270, 2016.

[102] A. N. Raghavan, H. Ananthapadmanaban, M. S. Sivamurugan, and B. Ravindran. Accurate Mobile Robot Localization in indoor environments using Bluetooth. In *IEEE International Conference on Robotics and Automation*, ICRA, pages 4391–4396. IEEE, 2010.

[103] M. Rahnema. Overview Of The GSM System and Protocol Architecture. *IEEE Communications Magazine*, 31(4):92–100, April 1993.

[104] A. Ranganathan, H. Ólafsdóttir, and S. Capkun. SPREE: A Spoofing Resistant GPS Receiver. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 348–360. ACM, 2016.

[105] B. Ritholtz. Could GPS Spoofing Cause Another Flash Crash?, August 2012. `https://ritholtz.com/2012/08/could-gps-spoofing-cause-another-flash-crash/`, visited 22 April 2019.

[106] C. R. P. Rodgers, R. Strachey, J. Janssen, and L. Cruls, editors. *International Conference Held at Washington for the Purpose of Fixing a Prime Meridian and a Universal Day. October, 1884.* Gibson Bros., Printers and Bookbinders, 1985. Archive copy at `http://www.gutenberg.org/files/17759/17759-h/17759-h.htm`, visited 30 September 2019.

[107] T. Roscoe, R. Wattenhofer, et al. Computer Systems. ETH Zurich, 2018. `https://www.systems.ethz.ch/courses/fall2018/computersystems`, visited 20 April 2019.

[108] RTCA DO-260B. Minimum Operational Perfomance Standards for 1090 MHz Extended Squitter Automatic Dependent Survelllance - Broadcast (ADS-B) and Traffic Inforamtion Services - Broadcast (TIS-B). Standard, RTCA Inc., Washington, DC, USA.

[109] M. Schäfer, M. Strohmeier, M. Smith, M. Fuchs, R. Pinheiro, V. Lenders, and I. Martinovic. OpenSky Report 2016: Facts and Figures on SSR Mode S and ADS-B Usage. In *35th Digital Avionics Systems Conference*, DASC, pages 1–9. IEEE/AIAA, Sept 2016.

[110] D. Schmidt, K. Radke, S. Camtepe, E. Foo, and M. Ren. A Survey and Analysis of the GNSS Spoofing Threat and Countermeasures. *ACM Computing Surveys (CSUR)*, 48(4), 2016.

[111] F. Seco, A. R. Jiménez, C. Prieto, J. Roa, and K. Koutsou. A survey of Mathematical Methods for Indoor Localization. In *IEEE International Symposium on Intelligent Signal Processing*, WISP 2009, pages 9–14. IEEE, August 2009.

[112] R. Seller and Á. Szüllő. Wide Area Multilateration Demonstration System. In *21st International Conference on Applied Electromagnetics and Communications*, ICECom, pages 1–5. IEEE, 2013.

[113] S.-H. Seo, B.-H. Lee, S.-H. Im, and G.-I. Jee. Effect of Spoofing on Unmanned Aerial Vehicle using Counterfeited GPS Signal. *Journal of Positioning, Navigation, and Timing*, 4(2):57–65, 2015.

[114] S. A. Shaukat, K. Munawar, M. Arif, A. I. Bhatti, U. I. Bhatti, and U. M. Al-Saggaf. Robust vehicle localization with gps dropouts. In *6th International Conference on Intelligent and Advanced Systems*, ICIAS, pages 1–6. IEEE, August 2016.

[115] A. L. Shimpi. Qualcomm Announces Snapdragon 410 Based on 64-bit ARM Cortex A53. AnandTech, December 2013. `http://www.anandtech.com/show/7573/qualcomm-announces-snapdragon-410-based-on-64bit-arm-cortex-a53-and-adreno-306-gpu`, visited 22 April 2019.

[116] A. L. Shimpi. Snapdragon 610 & 615: Qualcomm Continues Down its 64-bit Warpath with 4/8-core COrtex A53 Designs. AnandTech, February 2014. `http://www.anandtech.com/show/7784/snapdragon-610-615-qualcomm-continues-down-its-64bit-warpath-with-48core-cortex-a53-designs`, visited 22 April 2019.

[117] ShoaibTheExplorer. The Lone Traveller, Blog Entry, February 2019.

[118] J. J. Spilker. Fundamentals of Signal Tracking Theory. *Progress in Astronautics and Aeronautics*, 163:245–328, 1996. Chapter 7 in *Global Positioning System Theory and Applications*, Volume 1, Parkinson, B. et al.

[119] D. Stanke and B. Bradley. managing the ins and outs of commercial building pressurization. *Trane Engineers Newsletter*, 31, 2002.

[120] M. Strohmeier, M. Schäfer, V. Lenders, and I. Martinovic. Realities and Challenges of NextGen Air Traffic Management: The Case of ADS-B. *IEEE Communications Magazine*, 52(5):111–118, May 2014.

[121] J. Sun, J. Hoekstra, and J. EllerBroek. ADS-B Decoding Guide, 2017. `https://adsb-decode-guide.readthedocs.io/en/latest/content/introduction.html`, visited 10 April 2017.

[122] C. D. Taylor, S. J. Gutierrez, S. L. Langdon, K. L. Murphy, and W. A. Walton. Measurement of rf propagation into concrete structures over the frequency range 100 mhz to 3 ghz. In *Wireless personal communications*, pages 131–144. Springer, 1997.

[123] Texas Instruments. White Paper SWRA349, 2010. `https://e2echina.ti.com/cfs-file/__key/communityserver-discussions-components-files/104/7510.swra349-Coin-cells-and-peak-current-draw.pdf`, visited 19 October 2018.

[124] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun. On the requirements for successful gps spoofing attacks. In *Proceedings of the 18th ACM conference on Computer and communications security (CCS)*, pages 75–86. ACM, 2011.

[125] J. B.-Y. Tsui. *Fundamentals of Global Positioning System Receivers - A Software Approach.* John Wiley and Sons, Inc., New York, NY, 2000.

[126] P. Vabre. Air Traffic Services Surveillance Systems, Including An Explanation of Primary and Secondary Radar, 2010. `http://www.airwaysmuseum.com/Surveillance.htm`, visited 11 April 2017.

[127] F. S. T. Van Diggelen. *A-GPS: Assisted GPS, GNSS, and SBAS.* Artech House, 2009.

[128] F. Wang, H. Li, and M. Lu. Arpso-mle based gnss anti-spoofing method. In *Signal Processing, Communications and Computing (ICSPCC), 2015 IEEE International Conference on*, pages 1–5. IEEE, 2015.

[129] R. Wattenhofer. *Blockchain Science: Distributed Ledger Technology.* CreateSpace Independent Publishing Platform, 2019. third edition.

[130] K. D. Wesson. *Secure Navigation and Timing Without Local Storage of Secret Keys.* PhD thesis, May 2014.

[131] D. R. Williams. Earth Fact Sheet. Technical report, NASA Goddard Space Flight Center, Greenbelt, MD, USA, 2019. `https://nssdc.gsfc.nasa.gov/planetary/factsheet/earthfact.html`, visited 30 September 2019.

[132] . . . . . World Radiocommunication Conferences 1995, 1997, editor. *Radio Regulations.* International Telecommunication Union.

[133] G. Yan and L. Qing. Closely spaced multipath mitigation in gnss receiver based on maximum likelihood estimation. In *WCSP*, pages 1–5, 2013.

[134] S.-H. Yang, H.-S. Kim, Y.-H. Son, and S.-K. Han. Three-Dimensional Visible Light Indoor Localization Using AOA and RSS With Multiple Optical Receivers. *Journal of Lightwave Technology*, 32(14):2480–2485, 2014.

[135] H. Ye, T. Gu, X. Zhu, J. Xu, X. Tao, J. Lu, and N. Jin. FTrack: Infrastructure-free Floor Localization via Mobile Phone Sensing. In *IEEE International Conference on Pervasive Computing and Communications*, PerCom, pages 2–10. IEEE, 2012.

[136] D. Yuan, H. Li, and M. Lu. A method for gnss spoofing detection based on sequential probability ratio test. In *Position, Location and Navigation Symposium-PLANS 2014, 2014 IEEE/ION*, pages 351–358. IEEE, 2014.

[137] J.-M. Zogg. GPS – Compendium. Technical report, u-blox AG, Switzerland, 2001.

Index