KTH ROYAL INSTITUTE OF TECHNOLOGY

Degree Project in Technology

Second cycle, 30 credits

Digital Over-the-Air Computation with Software-defined Radio

A Practical System Design

QI XIONG

Stockholm, Sweden, 2024

Digital Over-the-Air Computation with Software-defined Radio

A Practical System Design

QI XIONG

Master's Programme, Information and Network Engineering, 120 credits
Date: May 16, 2024

Supervisor: Henrik Hellström
Examiner: Carlo Fischione
School of Electrical Engineering and Computer Science
Swedish title: Digital over-the-air beräkning med mjukvarudefinierad radio
Swedish subtitle: En praktisk systemdesign

Abstract

Traditional network scheme treats communication and computation are regarded as independent tasks, which brings the benefits of minimalist system complexity, high freedom of network architecture, and easier troubleshooting and maintaining. However, with the rapid growth of artificial intelligence and machine learning, computation-oriented tasks are more and more popular with giant internet companies, which call for a rising demand for function transmission. Over-the-air Computation (AirComp) becomes new research trend in Wireless Sensor Network (WSN) and distributed learning networks. The goal of AirComp is to compute functions of messages by exploiting electromagnetic interference. The vast majority of the work is based on theoretical methods and numerical simulation. There is a call for implementation of these methods over real wireless channels, using transceiver chains implemented in hardware.

In this thesis, an AirComp scheme to calculate the Majority Voting (MV) is proposed for Federated Edge Learning (FEEL), and an practical implementation of AirComp with the Software-defined Radio (SDR) device ADALM-PLUTO is demonstrated. This thesis addresses the theoretical study of digital AirComp by building an experimental platform and verifying the theory with the SDR device ADALM-PLUTO. Through simulations and experiment, it is proved that proposed AirComp scheme can provide a good MV performance even when the time-synchronization and the power control are not ideal under heterogeneous data distribution scenarios.

This thesis also addresses and analyzes the challenges encountered during the construction of the hardware experimental platform, proposing alternative solutions. Additionally, it discusses the platform's limitations and suggests directions for future work for AirComp.

Keywords

Over-the-air Computation, Software-defined Radio, Federated Edge Learning, Frequency Shift Keying, Majority Vote, Synchronization

Sammanfattning

Inside the following scontents environment, you cannot use a \includefilename as it will not end up in the for diva information. Additionally, you should not use a straight double quote character in the abstracts or keywords, use two single quote characters instead.

Enter your Swedish abstract or summary here!

Alla avhandlingar vid KTH **måste ha** ett abstrakt på både *engelska* och *svenska*.
Om du skriver din avhandling på svenska ska detta göras först (och placera det som det första abstraktet) - och du bör revidera det vid behov.

If you are writing your thesis in English, you can leave this until the draft version that goes to your opponent for the written opposition. In this way, you can provide the English and Swedish abstract/summary information that can be used in the announcement for your oral presentation.
If you are writing your thesis in English, then this section can be a summary targeted at a more general reader. However, if you are writing your thesis in Swedish, then the reverse is true – your abstract should be for your target audience, while an English summary can be written targeted at a more general audience.
This means that the English abstract and Swedish sammnfattning or Swedish abstract and English summary need not be literal translations of each other.

Do not use the \glspl{} command in an abstract that is not in English, as my programs do not know how to generate plurals in other languages. Instead, you will need to spell these terms out or give the proper plural form. In fact, it is a good idea not to use the glossary commands at all in an abstract/summary in a language other than the language used in the `acronyms.tex` file - since the glossary package does **not** support use of more than one language.

The abstract in the language used for the thesis should be the first abstract, while the Summary/Sammanfattning in the other language can follow

Nyckelord

Luftberäkning, Programvarudefinierad Radio, Federated Edge Learning, Frekvensskiftnyckel, Majoritetsomröstning, Synkronisering

If you are an exchange student, use the relevant language or languages for abstracts for your home university, as this will often avoid the need for writing another thesis for your home university.
If you are fluent in other languages, feel free to add the abstracts in one or more of them.

Note that you may need to augment the set of languages used in `polyglossia` or `babel` (see the file `kththesis.cls`). The following languages include those languages that were used in theses at KTH in 2018-2019, except for one in Chinese.
Remove those versions of abstracts that you do not need.
If you add a new language, when specifying the language for the abstract, use the three-letter ISO 639-2 Code – specifically the "B" (bibliographic) variant of these codes (note that this is the same language code used in DiVA).

Résumé

Résumé en français.

Mots-clés

Calcul en direct, Radio définie par logiciel, Apprentissage fédéré, Modulation par décalage de fréquence, Vote majoritaire, Synchronisation

Resumen

Résumé en espagnol.

Palabras claves

Computación inalámbrica, radio definida por software, aprendizaje perimetral federado, codificación por cambio de frecuencia, voto mayoritario, sincronización

Sommario

Sommario in italiano.

parole chiave

Calcolo over-the-air, Radio definita dal software, Federated Edge Learning, Frequency Shift Keying, Voto a maggioranza, Sincronizzazione

Sammendrag

Sammendrag på norsk.

Nøkkelord

Over-the-air-beregning, Programvaredefinert radio, Føderert kantlæring, Frekvensskiftnøkling, Flertallsavstemning, Synkronisering

Zusammenfassung

Zusammenfassung in Deutsch.

Schlüsselwörter

Over-the-Air-Berechnung, Softwaredefiniertes Radio, Federated Edge Learning, Frequenzumtastung, Mehrheitsabstimmung, Synchronisation

Resumé

Abstrakt på dansk.

Søgeord

Over-the-air-beregning, Softwaredefineret radio, Fødereret edge learning, Frequency shift keying, Flertalsafstemning, Synkronisering

Samenvatting

Samenvatting in het Nederlands.

Trefwoorden

Over-the-air Computation, Softwareradio, Randleren, Frequency Shift Keying, Meerderheid van stemmen, Synchronisatie

Kokkuvõte

Eesti keeles kokkuvõte.

Märksõnad

Üle-eetriline arvutamine, Tarkvaral põhinev raadio, Föderaalne servaõpe, Sageduse nihkepõhine võtmine, Enamushääletus, Sünkroniseerimine.

Acknowledgments

I would like to express my deepest gratitude to Professor Carlo Fischione for giving me this opportunity to discover this interesting research topic. His invaluable guidance throughout the course of this thesis project and rich expertise have been instrumental in the completion of this thesis.

I am also profoundly grateful to my supervisor, Henrik Hellström, for his constant support, patience, and constructive feedback. His unwavering belief in my abilities and thoughtful mentorship have significantly shaped this work. I wish him great success in his future endeavors and career progression.

Many thanks to my family and friends for their encouragement and understanding during this journey. Finally, I would like thank myself for starting this journey two years ago, this courage and confidence will continue through my later career and study life. Soon I will be 30 year old, and here is a quote for me and all my friends: the best time to plant a tree was 20 years ago, the second best time is now.

Stockholm, May 2024
Qi Xiong

Contents

List of Figures

List of Tables

List of acronyms and abbreviations

AI	Artificial Intelligence
AirComp	Over-the-air Computation
AM	Amplitude Modulation
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
CER	Computation Error Rate
CFO	Carrier Frequency Offset
COTS	commercial-off-the-shelf
CP	Cyclic Prefix
CSI	Channel State Information
DAC	Digital-to-analog converter
ED	Edge Device
ES	Edge Server
FEE	Function Estimation Error
FEEL	Federated Edge Learning
FL	Federated Learning
FM	Frequency Modulation
Frequency Xlating FIR Filter	Frequency Translating Finite Impulse Response Filter
GPS	Global Positioning System
IoT	Internet of ThingS
LDPC	Low-Density Parity-Check
LoS	Line of Sight
LSTM	Long Short-term Memory Network
LZW	Lempel-Ziv-Welch
MAC	Multiple Access Channel

MCU	Micro Controller Unit
ML	Machine Learning
MSE	Mean Squared Error
MSFE	Mean Squared Function Error
MV	Majority Voting
NMSE	Normalized Mean Square Error
NN	Neural Network
OFDM	Orthogonal Frequency Division Multiplexing
OOK	On-Off Keying
PM	phase Modulation
PO	Phase Offset
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RF	Radio Frequency
RTTY	Radio TeleTYpe
SDR	Software-defined Radio
SNR	Signal-to-noise Ratio
TO	Time Offest
USRP	Universal Software Radio Peripheral
VCO	Voltage controlled oscillator
Wi-Fi	Wireless Fidelity
WSN	Wireless Sensor Network

The list of acronyms and abbreviations should be in alphabetical order based on the spelling of the acronym or abbreviation.

Chapter 1

Introduction

In traditional network schemes, communication and computation are regarded as independent tasks, which brings the benefits of minimalist system complexity, high freedom of network design, and easier troubleshooting and maintaining. However, with the rapid growth of artificial intelligence and machine learning, computation-oriented tasks are more and more popular with giant internet companies, which call for a rising demand for function transmission. Over-the-air Computation (AirComp) comes up with the idea of computing functions by exploiting electromagnetic interference over transmission channels [1].

Ideally, with a perfect matching, function computation over a multiple access channel can increase the computation rate proportional to the number of users [2]. It is a great advantage especially for Federated Learning (FL), because of the huge amount of end devices involved to collect data for model training. What's more, OAC can also reduce the per-round communication latency, e.g., in Federated Edge Learning (FEEL) wireless networks [3], which is linearly increased with the growing number of Edge Devices (EDs). In [4], an experiment of practical federated learning is demonstrated, using relatively few rounds of communication to train high-quality models like Neural Network (NN) and Long Short-term Memory Network (LSTM). Therefore, it is a beneficial and innovative way to let communication and computation process together compared with treating them as independent tasks traditionally.

In this thesis, feasibility of various OAC applications via real wireless channels is studied through theoretical analysis and system simulation, and a practical communication system of digital OAC aiming at Majority Voting (MV) is simulated and implemented with Software-defined Radio (SDR).

1.1 Background

The concept of function computation over a multiple access channel was thoroughly explored in Bobak's seminal work [2], where the theoretical boundaries for computation over such channels were examined for a specific many-to-one function. Goldenbaum, in [5], was the first to link nomographic functions to OAC. Later, studies in [6] demonstrated that OAC could achieve a much higher computation rate compared to traditional methods that separate communication and computation. These compelling information-theoretical findings have sparked growing interest in OAC within the academic community. Initially, the focus was on communication challenges within interference channels, such as physical layer network coding [7], compute-and-forward relaying [8], and Wireless Sensor Network (WSN) addressing issues like accelerating gossip networks [9]. As the demand for intensive computation grows, OAC has expanded into multi-disciplinary domains, including Machine Learning (ML) over wireless networks [10], wireless control systems [11], and innovative computing frameworks such as wireless data centers [12].

1.2 Problem

The vast majority of the work is based on theoretical methods and numerical simulation. What are the AirComp performance differences between different digital communication schemes? Can we verify existing theoretical results for Over-the-Air Computation in a practical system?

1.2.1 Original problem and definition

There are many parameters that influence the quality of communication, and functions to be computed are also of various types. When AirComp combines communication and computation together, it is worth a careful consideration to determine system setup and metrics to compare the performance.

Computation rate and Mean Squared Error (MSE) for the computation are the common key AirComp-related metrics, but how these metrics are affected under different channel asynchronization offsets *i.e.*, Time Offest (TO), Phase Offset (PO), and Carrier Frequency Offset (CFO) is not well-assessed in the literature.

Computation rate $\mathcal{R}$ can be defined as the number of functions computed per channel use, and experssed as [1]

$$\mathcal{R} = \frac{N_f}{D}[functions/dimension] \tag{1.1}$$

where $D = 2N_{access}N_r$ is the number of real dimensions.

Similar to computation rate $\mathcal{R}$, computation throughput R is also a metric of computation capability, and can be defined as the number of functions computed per second, experssed as

$$R = \frac{N_f}{T}[functions/second] \tag{1.2}$$

where T is the time interval used for computing N_f functions.

Error definitions in AirComp are more complicated, since it's the errors with respect to functions instead of bit in communication systems. Let $\hat{f}(s[n])$ be an estimate of $f(s[n])$ for $s[n] \triangleq [s_1[n], ..., s_K[n]]^T$, for$n \in \{1, 2, ..., N_f\}$. The Function Estimation Error (FEE) can be expressed as

$$e_{FEE}(\hat{f}(s[n])) \triangleq \hat{f}(s[n]) - f(s[n]) \tag{1.3}$$

The classical MSE and Normalized Mean Square Error (NMSE) [13] can be expressed as

$$MSE(\hat{f}(s[n])) \triangleq \mathrm{E}\{||e_{FEE}(\hat{f}(s[n]))||_2^2\} \tag{1.4}$$

and

$$NMSE(\hat{f}(s[n])) \triangleq \frac{\mathrm{E}\{||e_{FEE}(\hat{f}(s[n]))||_2^2\}}{||f(s[n])||_2^2} \tag{1.5}$$

respectively. The Mean Squared Function Error (MSFE) is defined [14] by

$$MSFE \triangleq \frac{\mathrm{E}\{||e_{FEE}(\hat{f}(s[n]))||_2^2\}}{\mathrm{E}\{||f(s[n])||_2^2\}} \tag{1.6}$$

where the expectation is over both $s[n]$, $\forall n$, and the channel.

The Computation Error Rate (CER) is analogous to the Bit Error Rate (BER) in communication systems. It can be used to evaluate the computation when the arguments of the functions are discrete values, especially suitable for digital AirComp. Let $s_k[n]$ be a discrete random variable, $\forall k, n$, the CER can be defined as

$$P_{CER} \triangleq \mathrm{Pr}\left(\hat{f}(s[n]) \neq f(s[n])\right) \tag{1.7}$$

1.2.2 Scientific and engineering issues

As for scientific and engineering concern, there are several issues that affect the quality, reliability, and validity of this thesis.

First of all, the definition of the problem and the statement of the hypotheses must be clear and precise enough to allow for robust theoretical analysis and comprehensive experimental verification. This clarity ensures that researchers can develop efficient methods to investigate questions and construct experiments that test hypotheses under a variety of conditions. A well-defined question guides research focus, while a testable hypothesis allows for empirical review and data-driven conclusions. This combination fosters a systematic research approach that provides a solid foundation for testing or refuting the proposed hypotheses through empirical evidence and theoretical rigor.

Then choosing the appropriate method for data collection and analysis is crucial to ensure reliability and validity of results In scientific research, especially when dealing with complex data. It's essential to design a methodology that minimizes system errors, which are consistent inaccuracies due to flawed instruments or procedures, and random errors, which are unpredictable variations arising from random fluctuations. Reducing these errors enhances the accuracy of simulations and the confidence in the results. To improve simulation accuracy, researchers must implement robust calibration techniques, apply rigorous statistical analyses, and adopt validated models. Furthermore, cross-checking data with real-world observations and conducting sensitivity analyses can help identify and mitigate potential sources of error. By systematically addressing these issues, researchers can achieve more precise and reliable outcomes, ultimately contributing to scientific progress and engineering excellence.

At last, technical challenges in scientific research encompass several key aspects, such as access to the right equipment and resources, as well as the need for proper calibration and validation. Equipment issues can arise from limited access to advanced technology or inadequate maintenance, affecting the quality of data collected. Calibration ensures that instruments are accurate, while validation confirms that your results align with expected standards. Ensuring that others can replicate your research is equally important, which requires detailed documentation of methods, data, and analysis techniques. Addressing these technical challenges enhances the reliability and credibility of your research outcomes.

1.3 Purpose

According to theoretical studies, AirComp can increase the computation rate proportional to the number of users over Multiple Access Channels (MACs). It can also bring the benefit of latency reduction of communication per-round, which used to be linearly increased with the growing number of EDs. The purpose of this thesis is to prove those advantage of AirComp with convincing experiment results, and in particular, to design a practical MV AirComp system with SDR. Since SDR is open source with hardware, the experimental results can be easily reproduced by engineers who are interested in AirComp, and can also provide some practical basis for the subsequent theoretical research.

1.4 Goals

The goal of this project is to implement various theoretical AirComp methods and verify the practical performance of those theoretical studies. This has been divided into the following three sub-goals:

1. Subgoal 1: Study the prior art of Aircomp, choose the theoretical methods to be implemented on the suitable common wireless protocol.
2. Subgoal 2: Create the transceiver chain for a digital AirComp system using software-defined radio.
3. Subgoal 3: Obtain experimental results on the performance of digital MV AirComp over real wireless channels.

Ideally, the system design of GNU Radio and transceiver design based on PlutoSDR will be the deliverables of the project. With the software and hardware of PlutoSDR, one can easily replicate this project.

1.5 Research Methodology

In previous studies, it's known that AirComp theoretically brings many benefits such as higher computation rate, more efficient channel usage and reduced latency. To test these hypotheses, simulation experiment are chosen as the main method in this thesis. A simulation experiment often means programming a computer to perform a certain sequence of steps pr modules, which is used to represent something that is hard to manipulate directly in reality. For example, the channel fading and noise level may change variously under real communication conditions. Simulation experiment make it easier to control Signal-to-noise Ratio (SNR) by adjust the variable representing the signal energy and noise power assuming Additive White Gaussian Noise (AWGN) channel to test AirComp performance under different wireless channel conditions, which is a key result to support the advantage of AirComp in theoretical studies.

Other methods like mathematical proofs and empirical studies used in previous studies are lack of direct prove apart from rigorous mathematical reasoning, thus making it less convincing compared with experiment results. The important thing here is that the control and the intervention is done on a representation of the real world, rather than the real world itself. Those parameters in software are manipulated to get the certain experimental observation that can represent the performance of AirComp. We are not manipulating the target itself, that distinguishes simulations from field or laboratory experiments. Simulations are a critical tool for verifying theoretical analysis. Using software like MATLAB or Python-based frameworks, researchers create virtual environments to test how AirComp behaves under different conditions. Simulations can incorporate various parameters, such as noise, signal strength, and environmental factors, allowing researchers to test theoretical concepts in a controlled setting.

1.6 Delimitations

Among many theoretical methods of AirComp realization, this thesis focus on the implementation of a practical digital AirComp via wireless networks, and hardware platform used in the experiment is PlutoSDR. Due to the limited cost of equipments, the maximum number of EDs in our AirComp implementation is 3, other cases with more EDs is only analyzed theoretically and not included in the experiment.

Although wireless channel can be quite complicated and many models are available for different target distance and area, a Rayleigh fading wireless MAC with AWGN is considered in analysis and simulation. And our experiment is carried in the laboratory, which mean the AirComp is implemented under meters range and Line of Sight (LoS) is the main wireless path. Other complex condition like far range and reflection of walls and obstacles are not encountered in the experiment.

1.7 Structure of the thesis

First of all, the background of OAC is studied in chapter 2, answering basic questions of what is OAC, what functions can be computed via OAC, and current research trends. In chapter 3, fundamental knowledge of digital communication and SDR are reviewed to introduce further implementation of OAC via wireless networks, as well as discussions of method choice. Chapter 4 presents some essential problems of OAC in practice such as asynchronization, power dis-alignment, and channel estimation, as well as relevant simulation and experiments of SDR system using ADI Pluto. The simulation and experiment results and performance comparison between different OAC schemes are presented in chapter 5. At last, in chapter 6, summary of this thesis and some conclusions are drawn and potential future works are discussed.

Chapter 2

Background

This chapter provides the background information of AirComp, addressing foundational questions such as ”What is AirComp?”, ”What functions can be computed via AirComp?”, and discussing current research trends in the field. Additionally, this chapter reviews necessary concepts in digital communication and SDR related to later AirComp implementation on wireless networks.

2.1 Over-the-air Computation

Suppose we have a function $f(s_1, s_2, ..., s_k)$ to be evaluated at an ES, where s_k represents the symbol at the kth ED. These k EDs transmit their function simultaneously, and then the relevant computation of functions is done during transmission over the multiple access wireless channels. In the end, signals received by Edge Server (ES) are calculation results, the whole process is shown in Fig. 2.1. In many cases, the proposed computation coding outperforms the separation-based computation [15], especially when the number of sources becomes large. However, communication channels always suffer from noise and distortion, as well as synchronization errors, power management, and hardware limitations in wireless networks. AirComp relies on precise channel gain information and strict synchronization between edge devices, which is challenging in practical applications. Those practice problems of AirComp are investigated from different perspectives in recent research, and some results of them are discussed later in this chapter.

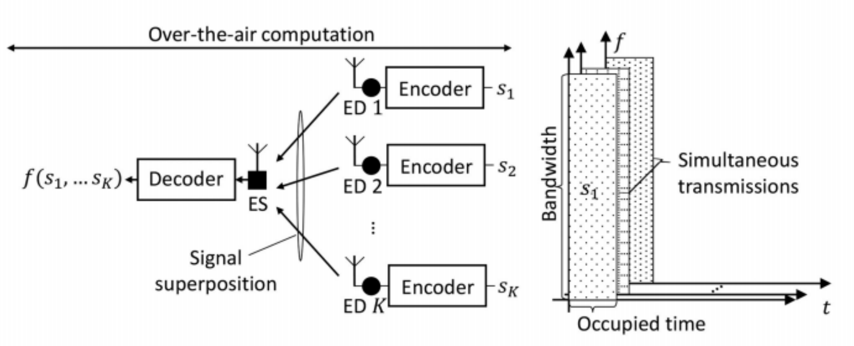

Figure 2.1: Computing functions via signal superposition while transmitting over multiple access channels

2.1.1 Nomographic Functions

AirComp aims to compute a multivariate function by relying on its representation that can structurally match with the underlying operation that multiple access channel naturally performs. Depending on different coding methods, AirComp can compute multiple weighted sums, arithmetic mean, polynomial functions, majority vote, production, and so on. They are called nomographic functions [5], or a set of nomographic functions over multiple wireless resources:

$$f(s_1, s_2, \ldots, s_k) = \varphi\left(\sum_{k=1}^{K} \psi_k(s_k)\right) \tag{2.1}$$

The name "nomographic functions" indicates that they are in line with the nomographs that solve certain equations through some graphs. Some examples are shown in Table 2.1. It is also mentioned that the universality of nomographs functions [16] provides robustness against uncertainty in

There are two interesting research directions from recent papers, one is to calculate an approximate nomographic function with a continuous and monotone post-processing function and continuous pre-processing functions [17], in which the analog solution of AirComp is mainly introduced for computing continuous function. In [18], the target function is interpreted with a bijective function such that the result function can be approximated well with a first-order analysis of variance decomposition. This could be of high practical value for edge computation in WSN applications. Another interesting direction is to compute the result function by expressing it as a solution to an

Table 2.1: Examples of functions can be computed via AirComp

Description	$f(s_1, s_2, ..., s_K)$	$\psi_k(x)$
Arithmetic mean	$\frac{1}{K}\sum_{k=1}^{K} s_k$	x
Weighted sum	$\sum_{k=1}^{K} w_k s_k, w_k \in \mathbb{R}$	$w_k x$
Polynomial function	$\sum_{k=1}^{K} c_k s_k^{k-1}, c_k \in \mathbb{R}$	$c_{k-1} x^{k-1} x$
Majority vote	$\text{sign}\left(\sum_{k-1}^{K} \text{sign}(s_k)\right)$	$\text{sign}(x)$
Counting number of EDs with the class $\mathcal{C}$	$\sum_{k=1}^{K} \mathbb{I}[s_k \in \mathcal{C}]$	x
p-norm	$\left(\sum_{k=1}^{K} \lvert s_k\rvert^p\right)^{1/p}$	$\lvert x\rvert^p$
Modulo-2 sum	$s_1 \oplus s_2 ... \oplus s_K, s_k \in \mathbb{Z}_2$	x
Approximation of the product	$\prod_k s_k, s_k \geq 0$	$\ln\left(x + \frac{1}{p_0(\epsilon)}\right)$
Approximation of the geometric mean	$\left(\prod_k s_k\right)^{1/K}, s_k \geq 0$	$\ln\left(x + \frac{1}{p_0(\epsilon)}\right)$

optimization problem, and the problem is solved through iterations that can be expressed with some elementary nomographic functions. For instance, the geometric median can be calculated through iterations over-the-air by using the Weiszfeld algorithm [19], which is a robust solution against Byzantine attacks. By using the binary representations, type-threshold functions [14] are achieved through AirComp, *e.g.*, maximum or minimum. Different digital communication schemes and modulation methods are usually investigated in this research direction to obtain better AirComp metrics targeting different result functions under complex practical channel conditions.

2.1.2 Synchronization

One of the critical issues in implementing AirComp is synchronization impairments, like CFO, PO, and TO. The CFO and PO between the kth ED and the ES can be present $\Delta f_k = f_{ED,k} - f_{ES}$ and $\Delta\theta_k = \theta_{ED,k} - \theta_{ES}$, respectively. Assume that t_0 denote the ideal synchronization point at the ES, and the time-of-arrival instant of the kth ED signal at the ES location is $\Delta t_{ED,k}$ seconds, and the synchronization point at the ES deviates by Δt_{ES} seconds, then the over TO can be expressed as $\Delta t_k = \Delta t_{ED,k} - \Delta t_{ES} - t_0$, as shown in Fig. 2.2.

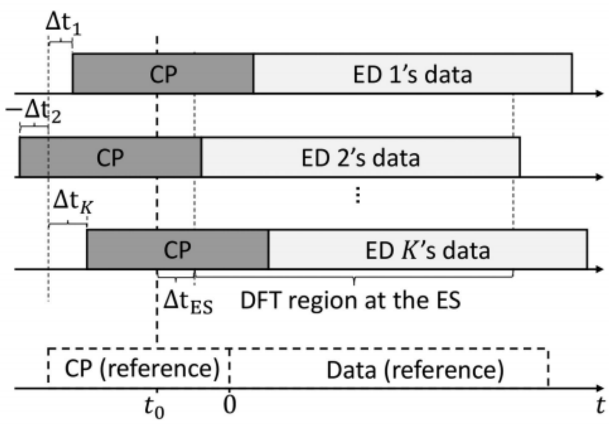

Figure 2.2: Time misalignment of AirComp at ES receiver relative to the ideal sync point t_0

There are many papers discussing the AirComp implemented on imperfect channel conditions. Taking the above three synchronization impairments into account, the passband signal received at ES can be expressed as:

$$s(t) = \Re\{e^{j2\pi f_{ES}t}\mathfrak{s}_k(t)\} \tag{2.2}$$

where $\mathfrak{s}_k(t)$ is the received baseband signal from kth ED given by:

$$\mathfrak{s}_k(t) = \sum_{p=1}^{P} a_{k,p}^{b}\mathfrak{s}_k(t-\tau_k)e^{j2\pi f_k t} \tag{2.3}$$

where P is the number of paths.

Based on 2.3, the timing errors at the EDs or ES not only translate the signal in time but also cause an additional phase rotation $2\pi\Delta f_k\Delta t_k$. Secondly, the CFO also results in an additional phase rotation, depending on the time offset and path delays [20]. At last, the CFO causes phase error accumulation that grows over time, which gives an additional compensation for phase synchronization.

In general, the existence of CFO causes inter-carrier interference while TO due to the imperfect time-of-arrivals or the synchronization errors at the ES results in phase rotations scaled with the subcarrier index [1]. The residual PO leads to a distortion independent from the subcarrier index [21]. To address these challenges of asynchronous AirComp, a Bayesian approach [22] can be implemented by letting each edge device transmit two pieces of statistical information to the fusion center such that Bayesian estimator can be devised to tackle the misalignments. In [22], a sum-product maximum a posteriori (SP-MAP) estimators is on an equal footing in terms of the MSE performance, and significantly better than the traditional ML estimator. Moreover, the SP-MAP estimator is computationally efficient, the complexity of which grows linearly with the packet length. Designing a suit symbol frame is also a good way to mitigate the interference between the adjacent symbols due to the random TOs. For example, in [23], a guard time between adjacent PPM or CSK symbols is set to compensate the TO. Based on the channel condition, a suitable guard time between two symbols can make the random TOs ignorable for an acceptable computation rate.

The joint impact of PO, TO, and CFO on received signals affects key AirComp metrics, such as computation rate or MSE. However, how these metrics are influenced by such offsets is not thoroughly examined in the literature. Additionally, it's important to highlight that the sensitivity of computation to residual offsets varies depending on the scheme and the function to be computed. For example, in an AirComp scheme that relies on phase synchronization among the EDs, accurate compensation for PO, TO, and CFO is crucial, especially in the context of analog aggregation. This approach requires precise sample-level synchronization, and any synchronization errors could significantly impact the MSE. Conversely, if the target function is a MV computation and the AirComp scheme uses keying approaches with non-coherent detection, it's shown that synchronization within the Cyclic Prefix (CP) range is adequate, and MSE remains stable even with synchronization errors. Thus, this latter scheme offers more tolerance to offset-induced errors [3]. Therefore, there are two directions to solving the problem of asynchronization. For those demands that don't require strict sync, one should choose a particular scheme parameter setting, e.g., longer symbol duration, guard time, and a larger energy calculation window, which makes the system insensitive to the synchronization, achieving acceptable computation rate of AirComp with tolerance to the asynchronizaiton caused by channel fading and devices. Another way is to improve the performance of the channel and devices so that the channel state is good enough and the defects of devices such as oscillator jitter and antenna are slight enough to be ignored. For example, to mitigate the error accumulation, extra signaling or better high-precision clocks can be utilized to achieve better performance of AirComp.

2.1.3 Power management

Power management is another issue that requires in-depth analysis. On the ES receiver side, achieving perfect amplitude alignment is crucial for fairness and accurate computation. In some situations, the weakest link in terms of channel quality can disproportionately affect computation performance. In other cases, a carefully designed metric is needed, and the power control strategy should be tailored to the specific application. For instance, in [24], it is shown that the power alignment for gradient aggregation is not always necessary for the convergence of FEEL. For static fading channels, channel inversion that minimizes MSE for multiple transmissions is studied. It is shown that the optimal channel inversion coefficient is a function of the number of re-transmission [25]. In [26], directly optimized power control for FL in implemented by minimizing an upper bound of the FL loss function. When a large number of devices are involved in the computation, the dynamic range of the superposed signal at the ES exceeds the dynamic range of the receiver [27]. Lowering the transmit power of the EDs or implementing adaptive gain control for AirComp are two possible approaches to tackle this issue. A sum-power constraint is introduced to conserve energy in [28].

2.1.4 Channel estimation

To achieve an accurate aggregation over fading channels, accurate and fresh Channel State Information (CSI) may need to be available at the EDs and/or the ES, depending on the AirComp method. While an inaccurate channel estimation can cause an incoherent aggregation, the aging of the CSI estimate due to the residual CFO or mobility can result in a larger overhead and limit the number of functions to be computed in one single packet. For AirComp methods that use precoding techniques, each ED requires a unique UL CSI, which can be obtained through a DL signal in a time-domain duplexing system or sent back to the ED based on the ES's channel estimates. The former approach involves a calibration process, while the latter introduces overhead that grows with the number of EDs, leading to increased latency. The inaccuracies resulting from either approach can be modeled as errors relative to the true CSI. The strategy proposed here accounts for imperfect CSI at the EDs to determine the optimal number of local update steps [29].

Some AirComp methods use sum-channel estimates, rather than the UL CSI for each link. For instance, in [31], it is proposed to use a procedure that optimizes the beamforming vectors at the EDs and ES iteratively by exploiting the sum-channel CSI acquisition. In this method, the ES first transmits a set of reference symbols and broadcasts its current beamforming vector. After each ED estimates the DL channel and designs its beamforming vector, all the EDs transmit a set of common pilot symbols concurrently so that the ES can estimate the sum channel. The key observation is that the ES can update its beamforming vector based on the sum channel [30].

2.2 Digital Communication

Consider digital communication with one ES and K EDs via multiple access channels, information is transmitted as a sequence of discrete digital signals over wireless networks. Suppose the fading coefficient of each wireless channel is randomly independent, the channel output of K EDs (received at ES) can be present as:

$$y = \sum_{i=1}^{K} h_i x_i + z \tag{2.4}$$

As for the implementation of AirComp of digital communication, the problem mainly lies in asynchrony and channel-gain misalignments. They can be symbolized as the imperfect transmission matrix H and noise distribution z.

Digital communication involves source encoding/decoding, channel encoding/decoding, modulation, and demodulation [31], and a brief introduction of each module is presented below.

2.2.1 Source Coding

Generally, source encoding is a fundamental process in the field of information theory and digital communications, which aims to represent information using fewer bits than the original data. This reduction in data size is beneficial for more efficient information transmission and storage via different systems, allowing for more efficient use of communication bandwidth. There are two primary types of source encoding: lossless and lossy. Lossless compression ensures that the original data can be perfectly reconstructed from the compressed form, making it ideal for applications where data integrity is critical, such as text files and executable programs. Common lossless algorithms include Huffman coding, Lempel-Ziv-Welch (LZW), and arithmetic coding. These methods work by identifying patterns, repetitions, or frequently occurring elements within the data, allowing for a more compact representation. On the other hand, lossy compression sacrifices some information to achieve higher compression rates. It is typically used in multimedia applications like audio, image, and video files, where a certain level of data loss is acceptable in exchange for significant compression. For example, JPEG for images and MP3 for audio are lossy compression, which use perceptual coding to remove information less noticeable to the human eye or ear. Compressed data is restored to its original form through decoding, which requires knowledge of the encoding algorithm that used at the transmitter. As the inverse process of encoding, proper source decoding is essential for retrieving compressed information accurately.

2.2.2 Channel Coding

Channel encoding is another critical process in communication systems, which aims to protect transmitted data errors caused by noise, interference, or other imperfect channel issues. It plays a fundamental role in ensuring the reliability and integrity of data as it moves through various communication channels, such as wireless networks, fiber-optic networks, or satellite links. The main purpose of channel encoding is to add redundancy to the original data in such a way that errors can be detected and corrected during transmission. This redundancy helps to safeguard the data against various types of errors, improving the chances of accurate data retrieval at the receiver. Error-correcting code is commonly used in various communication systems, such as Hamming codes, Reed-Solomon codes, Low-Density Parity-Check (LDPC) codes, and convolutional codes. These codes work by encoding the original data into a longer sequence, with additional error detection and correction bits. For example, in Hamming codes, redundancy is achieved through parity bits that allow the receiver to identify and correct single-bit errors. Reed-Solomon codes are widely used in applications like CDs, DVDs, and QR codes, where they provide strong error correction for burst errors. Convolutional encoding is often used in combination with Viterbi decoding, which provids resilience against errors by examining the data in context with previous bits. This technique is based on a convolutional structure, and is commonly applied in wireless communication systems and satellite transmissions. Channel decoding is the complementary process to encoding, involving the recovery of the original data from the encoded form. It requires an understanding of the specific encoding scheme used and often involves complex algorithms for error correction. Proper channel decoding ensures that the data arrives accurately and intact, minimizing the impact of noise and other transmission errors.

2.2.3 Modulation

Modulation is the process of varying one or more properties of a carrier signal, typically a sinusoidal wave, to represent digital or analog baseband information. Since the math expression of sinusoidal waves can be written as $Asin(ft + \varphi)$, three intuitive ways of modulation are Amplitude Modulation (AM), Frequency Modulation (FM), and phase Modulation (PM). AM changes the amplitude of the carrier signal in proportion, different energy represents signal bits of data being transmitted. In FM, the frequency of the carrier signal varies according to the input signal, and it is widely used in radio broadcasting, offering greater resistance to noise compared to AM. PM involves altering the phase of the carrier signal based on the data signal, which always requires accurate phase sync in digital communication systems.

Modern communication systems use more complicated advanced modulation methods like Quadrature Phase Shift Keying (QPSK), Quadrature Amplitude Modulation (QAM), and Orthogonal Frequency Division Multiplexing (OFDM). QPSK and QAM are digital modulation techniques used to transmit data by manipulating the phase and amplitude of a carrier signal. QPSK can be regarded as a special 4QAM(although the actual phase angle is different). QPSK encodes two bits per symbol by shifting the phase of the carrier to one of four predefined angles, providing a balance between data rate and noise resilience. It is widely used in satellite communications, digital television, and cellular networks. QAM extends this concept by varying both the amplitude and phase, allowing for higher data rates. Common in cable modems, Wireless Fidelity (Wi-Fi), and advanced cellular networks, QAM can encode multiple bits per symbol, offering greater bandwidth efficiency. OFDM is a sophisticated digital modulation technique used in modern communication systems to efficiently transmit data across a wide frequency band while combating multi-path distortion and interference. It is particularly effective in environments where signal reflections and fading occur, such as wireless and broadband communications. In OFDM, the data stream is divided into multiple smaller substreams, each transmitted over a distinct carrier frequency. These frequencies, or subcarriers, are closely spaced but designed to be orthogonal, minimizing interference between them. This orthogonality allows multiple data streams to be transmitted simultaneously without crosstalk, thereby maximizing spectral efficiency. Due to the resilience to multipath fading and high efficiency of bandwidth usage, OFDM is a popular choice in various applications, such as Wi-Fi, 4G and 5G cellular networks, digital television, and broadband internet. The technology's robustness and flexibility enable it to handle high data rates and diverse communication conditions, contributing to its widespread adoption in contemporary communications. In [32], the implementation of AirComp via the asynchronous multi-user OFDM systems shows the practical value of computation over modern communication.

2.2.4 SDR

SDR is a type of radio communication system where the components typically implemented in hardware are instead realized with software on a general-purpose computer or reconfigurable hardware [33]. In an SDR, functions like modulation, demodulation, filtering, signal processing, and other key radio operations are performed using software algorithms, allowing greater flexibility and adaptability compared to traditional hardware-based radios. There are various SDR platforms and relevant learning materials [34].

GNU radio is one software that can make simulations and experiments of customized SDR systems with by beautiful GUI and a plant of useful modules [35]. As shown in Fig 2, graphic modules are easily understandable by humans, but not the computer. Therefore GNU Radio has converted the program file into Python code, and finally into C++ code. Understanding the underlying running logic of GNU radio code will help one better master this software and make customized modules on one's own.

2.3 Related Work area

A reliable superposition in a wireless channel is one of the major challenges for AirComp. To address this issue, some solutions in the literature are described below.

2.3.1 Major related work 1

Carrier clouds have been suggested as a way to reduce the delay between the users and the cloud server that is providing them with content. However, there is a question of how to find the available resources in such a carrier cloud. One approach has been to disseminate resource information using an extension to OSPF-TE, see Roozbeh, Sefidcon, and Maguire [36].

broadband analog aggregation (BAA) over OFDM is investigated. It is proposed to modulate the OFDM subcarriers with the model parameters at the EDs. To achieve a coherent superposition at the ES, the symbols on the OFDM subcarriers are multiplied with the inverse of the channel coefficients and the subcarriers that fade are excluded from the transmissions, which is known as truncated-channel inversion (TCI) in the literature [37]

2.3.2 Major related work 2

one-bit broadband digital aggregation (OBDA) [38] is proposed to facilitate the implementation of FEEL. In this method, by considering distributed training by MV with the sign stochastic gradient descent (signSGD) [16], the EDs transmit quadrature phase-shift keying (QPSK) symbols over OFDM subcarriers along with TCI, where the signs of the stochastic gradients, *i.e.*, votes, are mapped to the real and imaginary parts of the QPSK symbols. At the ES, the signs of the real and imaginary parts of the superposed received symbols are calculated to obtain the MV.

2.3.3 Major related work 3

An AirComp scheme to compute the MV for FEEL is proposed in [39]. The proposed approach uses orthogonal time-frequency resources, *i.e.*, subcarriers for FSK-MV and wide-band pulses for PPM-MV, to indicate the sign of the local stochastic gradients. Thus, it allows the ES to detect the MV with a non-coherent detector and eliminates the need for CSI at the EDs at the expense of a larger number of time and frequency resources.

2.3.4 Minor related work 1

AirShare [40] is another way to address the asynchronization problem, by transmitting a shared clock on the air and feeding it to the wireless nodes as a reference clock, hence eliminating the root cause for incoherent transmissions. Designing and delivering such a shared clock could be quite challenging, but it can avoid the different jitter of oscillators on different devices, which makes tight phase coherence achievable for some AirComp applications requiring strict phase synchronization.

2.3.5 Minor related work 2

In a practical network, time synchronization can be maintained via an external timing reference such as the Global Positioning System (GPS) [41]. RFClock-leader allows follower clocks to synchronize with mean offset under 0.107Hz, and then corrects the time/phase alignment to be within a 5ns deviation. RFClock is designed to operate in generalized environments: as standalone unit, it generates a 10MHz/1PPS signal reference suitable for most commercial-off-the-shelf (COTS) SDRs today; it does not require custom protocol-specific headers or messaging; and it is robust to interference through a frequency-agile operation.

2.4 Summary

In this chapter, background and recent research trend of AirComp are reviewed, especially for the challenges in AirComp implementation. Necessary knowledge of digital communication and SDR are also revised for later AirComp implementation on wireless networks. To the best of our knowledge, the state-of-the-art OAC schemes for FEEL do not address the case where CSI is unavailable to both EDs and ES. Therefore in our experiment, suppose CSI is estimated through manual test and compensated with calibration. Although CSI IS not used at the EDs, the sum of the superposed channel needs to be available at the ES.

Chapter 3

Method or Methods

The purpose of this chapter is to provide an overview of the research method used in this thesis. Section 3.1 describes the research process. Section 3.2 details the research paradigm. Section 3.3 focuses on the data collection techniques used for this research. Section 3.4 describes the experimental design and section 3.5 describes the method used for the data analysis.

3.1 Research Process

Brief steps in the process are list below:

Step 1 study prior art,

Step 2 select suitable system design from theoretical analysis,

Step 3 simulate the chosen system,

Step 4 plan experiment,

Step 5 conduct experiment,

Step 6 analyze data from the experiment, and

Step 7 discuss the results of the analysis.

3.2 Research Paradigm

In this thesis, a constructivist qualitative case study with a deductive (predetermined) chosen approach is adopted. And an inductive approach is also adopted when drawing conclusion from gathered experimental data and observation. The goal of this thesis is not only on identifying ”what” and

”how” AirComp works but also on understanding the underlying meanings, interpretations, and interesting contexts associated with its implementation. The qualitative case study approach allows for an in-depth exploration of the complexities and nuances inherent in implementing Aircomp with SDR. Through in-depth interviews, observations, and document analysis, rich and detailed data can be collected to illuminate the complexities of practical system of AirComp.

This thesis involves a deductive chosen approach, where initial research questions and hypotheses are guided by existing theories and literature. However, inductive reasoning is also involved when drawing general conclusions of performance comparison from specific observations or data during the implementation process. This iterative process of data collection and analysis enables the exploration of unexpected phenomena and the refinement of theoretical frameworks to better capture the intricacies of practical Aircomp system.

3.3 Data Collection

Monte Carlo simulation is used to evaluate the behavior of AirComp under uncertainty by random sampling. Massage transmitted in the experiment is randomly generated, and metrics of each design are evaluated through several repeated trials to avoid random error.

Since both simulations and related experiments are conducted in the laboratory with our hardware devices, the source of the data does not have any ethical or social security issues.

3.4 Experimental design and Planned Measurements

In this section, relevant design of equipment of experiment are presented, as well as expected measurement and observation are planned.

3.4.1 Test environment/test bed/model

The experiment was carried out in the laboratory. As shown in Figure 4.5, one device was used as ES, the other two or three devices were used as ED. regrading to different system implementation, IIO software was used for oscilloscope. A computer with at least three USB port is necessary, and of course more computers are better. Python environment is necessary for RNU Radio, the software to design and implement custom radio system.

3.4.2 Hardware/Software to be used

GNU Radio is the software we used for SDR system implementation. Show in Fig. 3.1, functions like modulation, demodulation, filtering, signal processing, and other key radio operations are performed using software algorithms, allowing greater flexibility and adaptability compared to traditional hardware-based radios. And ADALM-PLUTO is the hardware used in the experiment, simplified block diagram is shown in Fig. 3.2, and detailed features are listed in Table 3.1.

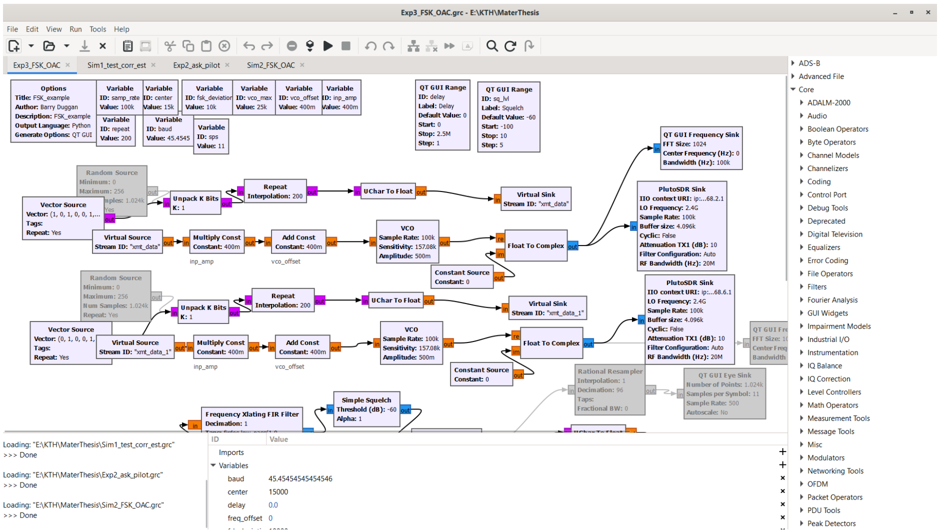

Figure 3.1: GNU Radio Software UI of an example project file

3.4.3 Validity of method

Several key steps and considerations are involved to ensure the validity of experimental results. Designing the experiment carefully to ensure that it accurately represents the intended scenario and conditions. This includes defining clear objectives, selecting appropriate parameters, and minimizing potential sources of bias or confounding variables. On the other hand, it's also necessary to calibrate all equipment, including antennas, transceivers, and processing units, to ensure accurate measurements and reliable performance.

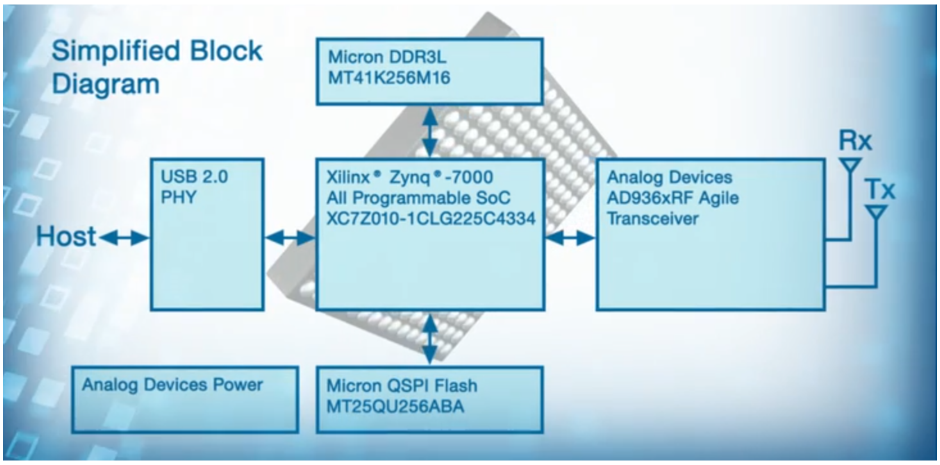

Figure 3.2: Simplified Block Diagram of ADALM-PLUTO

Validation involves verifying that the experimental setup produces consistent and expected results under known conditions. At last, conduct experiments in a controlled environment to minimize external interference and environmental factors that could affect the results.

3.4.4 Reliability of method

3.4.5 Data validity

Reproducibility is a good way to valodity the data. By conducting the experiment multiple times under the same conditions, it can be ensured that the results are consistent and reproducible. Consistency across repeated trials indicates the reliability of the findings.

We can also Validate the results using alternative methods or approaches to confirm their accuracy. Cross-validation involves comparing the outcomes obtained from different experimental setups or analysis techniques to ensure their agreement.

3.5 Planned Data Analysis

MSE analysis is planned to compare the performance of AirComp

Table 3.1: Features and Benefits of ADALM-PLUTO

Feature
Portable self-contained RF learning module
Cost-effective experimentation platform
Based on Analog Devices AD9363–Highly Integrated RF Agile Transceiver and Xilinx® Zynq Z-7010 FPGA
RF coverage from 325 MHz to 3.8 GHz
Up to 20 MHz of instantaneous bandwidth
Flexible rate, 12-bit ADC and DAC
One transmitter and one receiver, half or full duplex
MATLAB®, Simulink® support
GNU Radio sink and source blocks
libiio, a C, C++, C#, and Python API
USB 2.0 Powered Interface with Micro-USB 2.0 connector
High-quality plastic enclosure

3.5.1 Data Analysis Technique

Descriptive Statistics *i.e.,* mean, median, standard deviation, distribution, histogram can be used for data analysis. Descriptive statistics provide a summary of the central tendency, dispersion, and distribution of the MSE data, allowing you to gain insights into the variability and overall performance.

3.5.2 Software Tools

RNU Radio for simulation and SDR system design, and also necessary python code for random data generation and results analysis.

3.6 Evaluation framework

Firstly, simulation Studies are carried out to assess AirComp performance under controlled conditions and varying parameters. Simulations allow for systematic exploration of different scenarios and configurations, enabling a thorough evaluation of AirComp's capabilities and limitations.

Secondly, experimental observations can Validate AirComp performance through real-world experiments conducted in relevant environments. Experimental validation provides empirical evidence of AirComp's effectiveness in practical settings and helps confirm the applicability of simulation results to real-world scenarios.

Finally, in addition to MSE, consider incorporating other performance metrics relevant to the specific application or context of AirComp. These metrics could include computational time, energy consumption, communication overhead, or accuracy of computation results. A comprehensive set of performance metrics provides a more holistic evaluation of AirComp performance.

3.7 System documentation

Systemdokumentation
Med vilka dokument och hur skall en konstruerad prototyp dokumenteras? Detta blir ofta bilagor till rapporten och det som problemägaren till det ursprungliga problemet (industrin) ofta vill ha. Bland dessa bilagor återfinns ofta, och enligt någon angiven standard, kravdokument, arkitekturdokument, designdokumnet, implementationsdokument, driftsdokument, testprotokoll mm.

If this is going to be a complete document consider putting it in as an appendix, then just put the highlights here.

Please refer to appendix for project file and more detail.

Chapter 4

System Design and Implementation

In this chapter, the communication system scheme design of digital AirComp are demonstrated, and hardware and software used in the experiment are described. Before experiment, simulation and calibration are necessary steps for a successful implementation. At last, realization of AirComp by both time-domain and frequency-domain are implemented, and relevant experimental results and observation are presented in chapter 5.

4.1 Digital communication scheme

App to app system scheme with three layer network are shown in Figure 4.1.

In this framework diagram, the data (such as histogram or model weight matrix) is generated from FEEL apps in the orange blocks, with necessary pre-computation of functions involved. The data length in one CP duration is limited due to imperfect synchronization, so in green blocks, segmentation and heading are processed and turn the data into bit stream. Channel information are also extract in this layer to guide the communication parameters like symbol duration and guard time in lower layer. With digital communication process, the bit stream is encoded and modulated on a carrier wave, emitted to wireless channels via Digital-to-analog converter (DAC) and Radio Frequency (RF) circuit. In this experiment, the hardware used for the TX and RX parts is ADALM-PLUTO. Other radio platforms like Universal Software Radio Peripheral (USRP), or custom RF circuit implementations in the WSN are also used in practice.

Time alignment, or symbol window alignment is very important in the

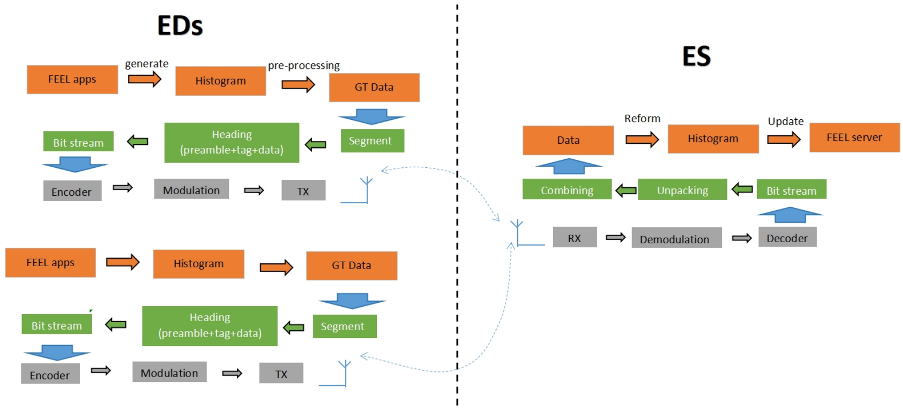

Figure 4.1: System diagram of FEEL over AirComp

communication process, the process of communication is demonstrated in Figure 4.2

First of all, ES and ED should have their own TX and RX. At the beginning of each communication (computation) cycle, the ES initially sends a broadcast pilot, and then the RX of ES is responsible for monitoring the pilot in the channel through the correlation detection of the pilot sequence. The pilot sequence has to be determined by server and known to every nodes for a stable network. Once EDs received the pilot, all EDs prepare their data and transmit them to ES. Usually, a delay compensation parameter inside each ED is necessary to control the arriving time variance at ES (how to calibrate this parameter is discussed later), in order to compensate for both the difference in the processing time of the respective Micro Controller Unit (MCU) of devices and the DIFFERENT transmission distance. All the signals emitted by ED are superimposed in the wireless channel, and the superimposed signals are received by the ED RX.

4.2 Simulation

In order to obtain the delay information of some channels and compensate for the time difference in the previous section, ES needs to monitor the time difference of all symbols, and has made compensation for system parameters, such as increasing symbol duration and guard time to counter the greater difference in response time caused by too many devices. The effect of pilot sequence is simulated here. The simulation uses two ED to send a designed pilot+ random data with a length of frame to an ES. In the simulation,

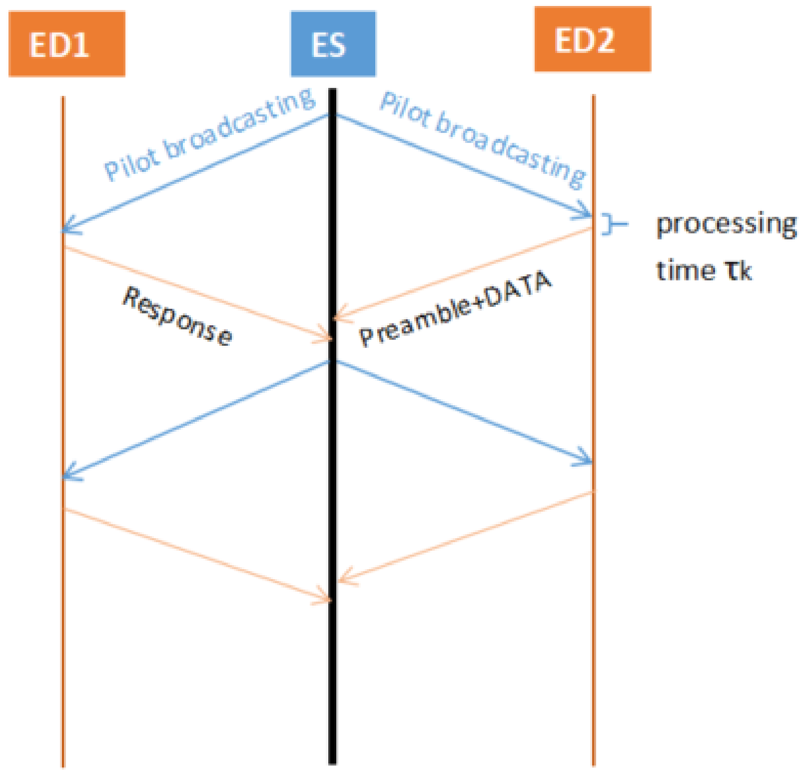

Figure 4.2: Example of AirComp communication scheme

delay variables can be adjusted by software setting parameters to simulate the different processing plus transmission time for each ED in the actual system. The simulation of the module setup in the GNU radio software, and the following Figure 4.3 and Figure 4.4 show the effect when the delay is 0 and 60ms respectively. When the delay is 0, the corr peaks coincide. When the delay is 60ms, the peaks of the two corr signals represent that the arrival time difference of two EDs is tested as 60 ms. As can be seen, an appropriate pilot sequence can be used to measure the overall time variance of all EDs counting from the pilot signal broadcast at ES to the data transmission back from EDs. The processing time τ_k may be quite different depending on the state of MCU and response time of interrupt, the transmission time due to different distance is alleviative due to the experiment environment in the laboratory. In this experiment, by setting a suitable the symbol duration and guard time in communication scheme, this arrival time variance can be ignored during a CP cycle.

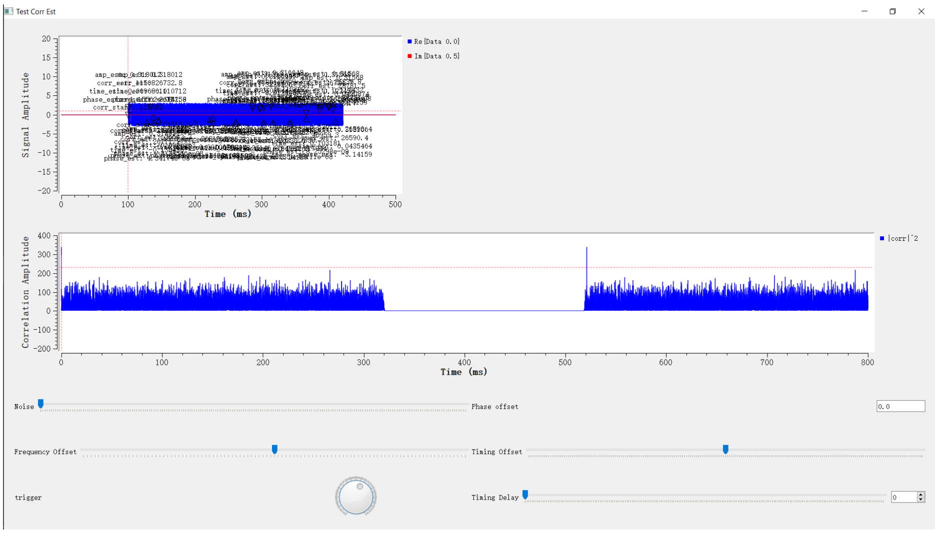

Figure 4.3: Correlation curve at ES when simultaneous transmission of EDs

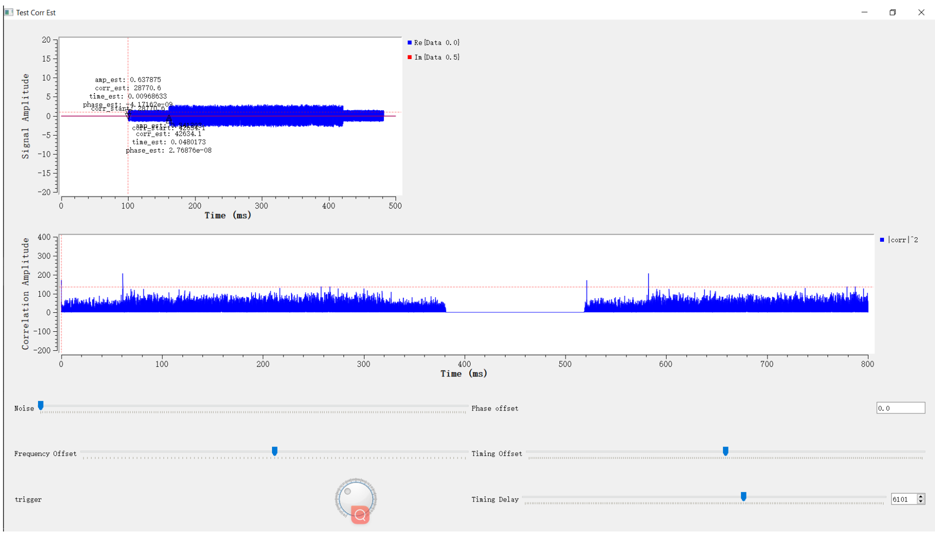

Figure 4.4: Correlation curve at ES when one ED delayed 60 ms

4.3 Calibration

All test devices and observation devices need to be calibrated before the experiment, such as frequency offset after demodulation. The table shows the deviation of center frequency of three devices transmitting the same FSK signal to different devices at the same time. After obtaining the frequency offset parameters, parameter design can be carried out for the FSK system, so that the frequency difference between 0 and 1 is higher than their respective frequency offset. At the same time, an ES combination is selected to obtain the minimum current frequency offset. Finally, Pluto with IP address 192.168.4.1 is selected as ES, while other Pluto is selected as ED. In practice, repeated testing can reduce random errors, so the frequency shift data are test thrice for every pair of devices.

Table 4.1: Frequency shift on different device pair

Device pair	Δf_{center}**(kHz) 1st Trial**	Δf_{center}**(kHz) 2nd Trial**	Δf_{center}**(kHz) 3rd Trial**
SN#4 to SN#2	-8.03	-7.87	-7.81
SN#4 to SN#6	-9.65	-9.46	-9.54
SN#2 to SN#4	+7.88	+7.81	+7.78
SN#2 to SN#6	-1.57	-1.44	-1.51
SN#6 to SN#2	+1.61	+1.44	+1.48
SN#6 to SN#4	+9.05	+9.39	+9.18

4.4 Implementation

The digital AirComp hardware platform is implemented with 4 PlutoSDR devices, as shown in Figure 4.5. IP address of them are set to 192.163.2.1 (SN#2), 192.163.4.1 (SN#4), 192.163.5.1 (SN#5) and 192.163.6.1 (SN#6). As discussed in section 4.3, SN#4 Pluto is selected as ES, and all the other Pluto are used as EDs. Time-domain or frequency-domain realization of wireless communication systems has been a long-standing debate. When it concerns misalignments, the time-domain realization is sensitive to time offsets among edge devices, while the frequency-domain realization is sensitive to the CFO among edge devices [21]. So two kinds of realization are both implemented in experiment. The superposition of time domain signal is more direct to be observed, and it can achieve function computation through energy level estimation in each symbol window. In FSK, ”0” and ”1” are represented as signals of different frequency, then majority vote can be achieved through , while the frequency offset is converted into the error of the decision level, easier to control.

Figure 4.5: The device ADALM-PLUTOs with one as ES and others as EDs

4.4.1 Superposition (Time-domain realization)

The system is constructed as shown in Figure 4.6. Two vector sources repeatedly generate bit stream for two EDs, respectively, and multiply the bit signal with a cosine wave with a frequency of 5k, so as to obtain an On-Off Keying (OOK) modulated carrier signal. Finally, this cosine signal is connected to the device through PlutoSDR Sink module for emission. At the receiver side, the PlutoSDR Source module transmits the signal received by the RX into GNU Radio. The demodulation of the signal is also very simple, directly multiplied by the carrier signal of the same frequency, then with a low-pass filter, the baseband signal is demodulated. In chapter 5, the results of the signal superposition are demonstrated.

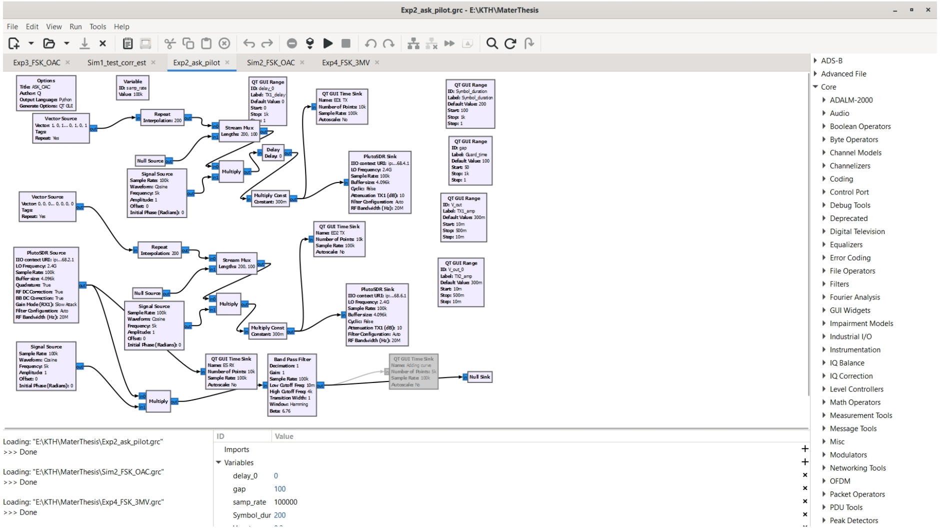

Figure 4.6: System design of ASK AirComp on GNU Radio

4.4.2 Majority vote (Frequency-domain realization)

The system is constructed as shown in Figure 4.7. At the transmitter, Voltage controlled oscillator (VCO) module produces a sinusoid of frequency based on the amplitude of the input. For example, the standard Radio TeleTYpe (RTTY) tones of 2295 (mark) and 2125 (space) are generated. The calculations for this follow [42]:

1. Choosing a full-scale frequency of 2500Hz with an input of +1.0:

$$\text{VCO Sensitivity} = \frac{2\pi \times 2500}{1.0} = 15708 \tag{4.1}$$

2. At the output of the Low Pass Filter, a Mark has a value of +1.0 and a Space has a value of 0.0.

3. When the output of the Low Pass Filter is +1.0, the input of the VCO is calculated as follows:

$$\text{Input of VCO} = (1.0 \times 0.068) + 0.850 = 0.918 \tag{4.2}$$

4. This generates a frequency of:

$$0.918 \times 2500 = 2295 \text{ Hz} \tag{4.3}$$

The parameter of VCO module in our experiment can be obtained through the similar way, using multipuly const and add const modules to convert "0" and "1" bit stream into the expected input value of VCO. The center frequency and margin between FSK symbols should be chosed based on the calibration test results in section 4.3, to leave enough margin for a good decision level and noise tolerance. The frequency spectrum of ED(upper) and ES(bottom) are shown in Figure 4.8 upper and bottom graph, respectively.

At the receiver side, The Frequency Translating Finite Impulse Response Filter (Frequency Xlating FIR Filter) block performs a frequency translation on the signal and simultaneously downsamples the signal via a decimating FIR filter. The main use of this block is an effective channelizer, to pull out a narrowband portion of a wideband signal, without that narrowband portion having to be centered in frequency [43]. Through this block, the FSK symbols are shifted by f_{center} , then pass the Quadrature Demod block (with necessary suitable level of noise removing) and can be determined as "0" or "1" based on the positive or negative value of the block output. In chapter 5, the results of majority vote are demonstrated.

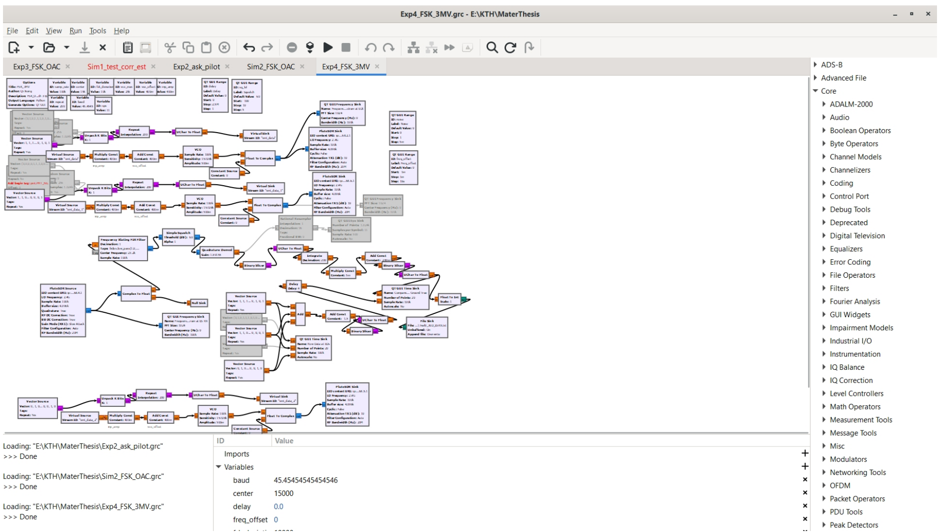

Figure 4.7: System design of MV AirComp on GNU Radio

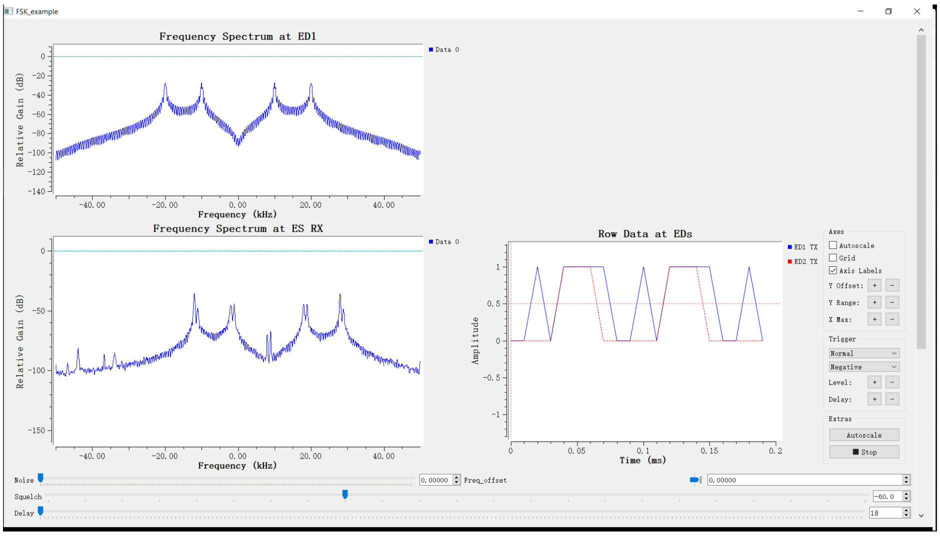

Figure 4.8: Frequency spectrum of FSK AirComp for MV on GNU Radio

Chapter 5

Results and Analysis

In this chapter, the results of superposition and majority vote are presented.

In Figure 5.1, one ED sent ”0101” and another ED sent ”1000”. With symbol duration of 2 ms and guard time of 1 ms, the superposition results are quite nice in time domain, as shown in the bottom graph. With suitable algorithm of energy level estimation in each symbol window, function computation *i.e.,* sum and average can be achieved. but power management is very important as discussed in section 2.1.3, since it’s easy to reach the power limit at receiver side with a large number of EDs.

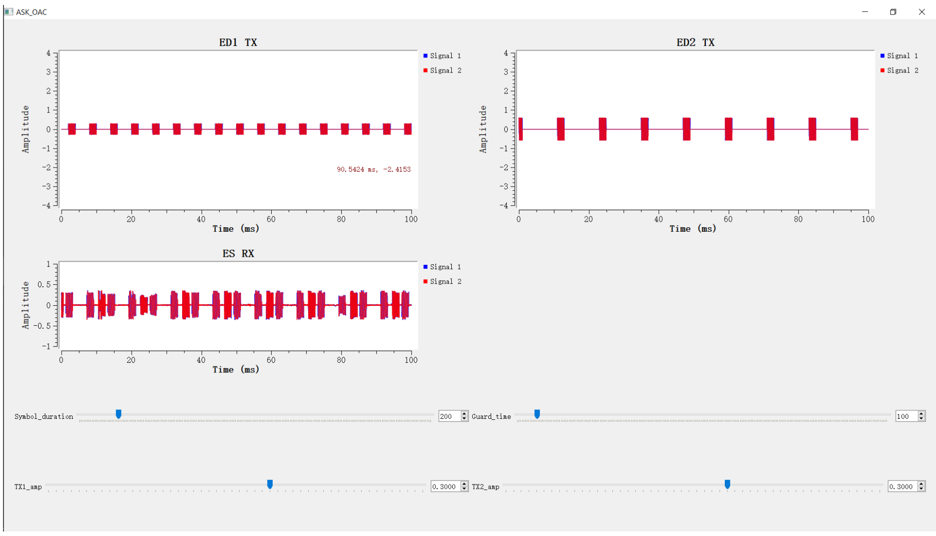

Figure 5.1: Signal superposition example

In the meantime, time offset will have a greater impact on error. If a large time dislocation occurs, then in the detection window of ES, less energy of the symbol will lead to misjudgment, thus increasing the MSE of function computations.

To implement majority vote, we send a binary sequence using 2 EDs and 3 EDs respectively. In the case of 2 EDs, we agree that if the number of 0 and 1 are equal, the result outputof voting will be 1. The truth value table of the experiment is shown in Table 5.1. And the exact results of received symbol ”11100101” in Figure 5.2 and ”11000001” in Figure 5.3.

According to the experimental results, I think the majority vote prototype of the short sequence has been achieved. It is worth noting that the channel condition is very good in the experiment, since the nodes are not far away from each other (within several meters) and there is no obstacles in LoS and negligible effect of multipath interference.

Next obvious things to do is testing the MSE of MV AirComp with long random generated sequence. With file source and file sink blocks from the GNU Radio, it should be easy to achieve bit stream to bit stream majority vote and then get the MSE by comparing the received results with ground truth. However, the processing time of the file source is not controlled by python code, thus making the superposition misaligned. More discussion is provided in section 6.3.1.

Table 5.1: Majority vote sequence result

Configuration	**2EDs to 1ES MV**	**3EDs to 1ES MV**
ED1 sent	11100101	11100101
ED2 sent	11000001	11000001
ED3 sent		01010010
ES received	11100101	11000001

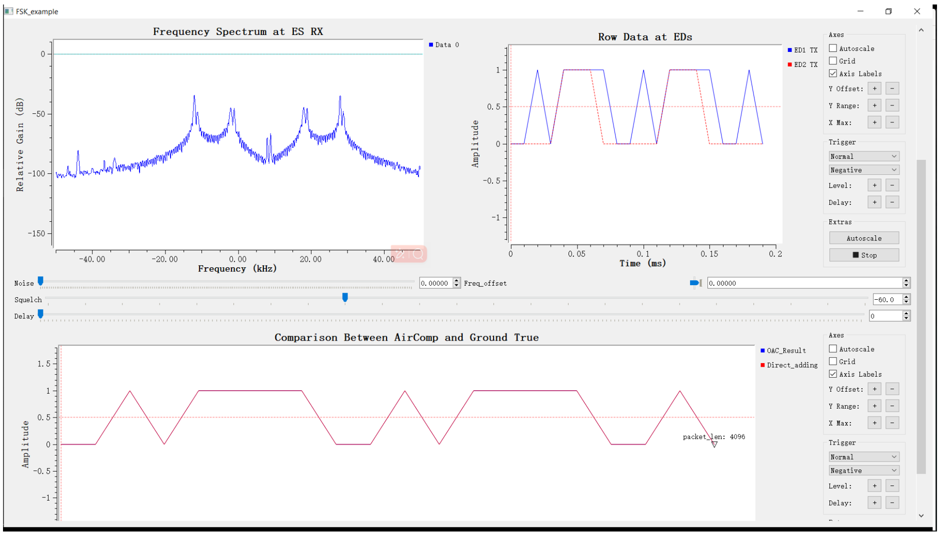

Figure 5.2: 2EDs to 1ES Majority Vote

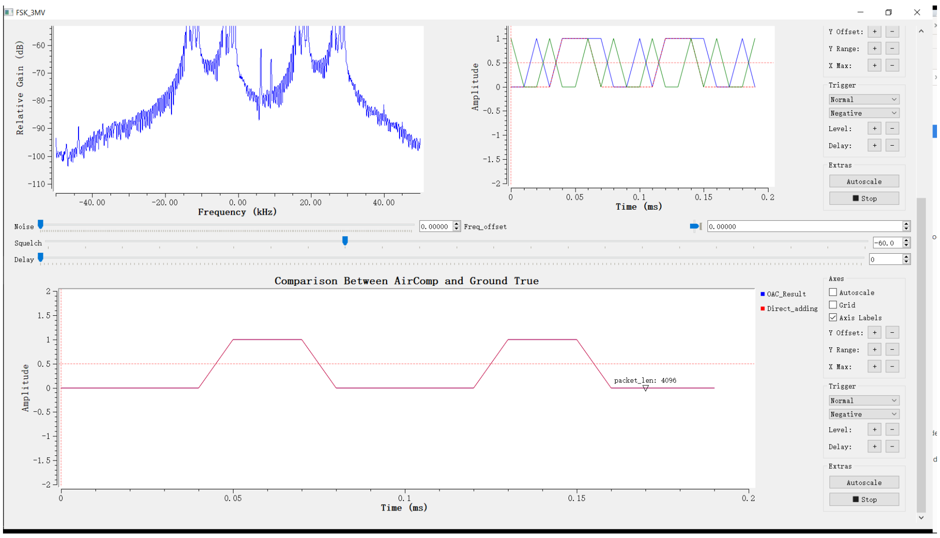

Figure 5.3: 3EDs to 1ES Majority Vote

Chapter 6

Conclusions and Future work

6.1 Conclusions

The future network demands *i.e.*, Internet of ThingS (IoT) and Artificial Intelligence (AI) are expected to operate with a vast number of nodes and substantial amounts of data. Often, WSNs and wireless distributed learning networks are not focused on individual data points, but rather on the outcomes of functions derived from this data. The AirComp is an innovative and effective way to complete those tasks by combining computation and transmission together via signal superposition in wireless channels.

This thesis describes how to build a simulator to validate the platform at thesample level in GNURadio, and compares the effect of the number of nodes and the length of symbols sequence in the network on the results through two numerical experiments. It is concluded that more nodes or symbols per data can neutralize the effect of random phases in the channel by increasing the number of manually encoded random phases, making the calculation more accurate.

This thesis outlines the process of developing a simulator in GNU Radio to validate the platform at the sample level. It includes a comparison of the effects of varying the number of nodes and the length of symbol sequences in the networks. What's more, both time domain and frequency domain implementation of AirComp are demonstrated to show the practical value of theoretical studies in this field. An AirComp prototype of majority vote was built with 4 ADALM-PLUTO devices and GNU Radio. The observations indicate that superposition of signal from different EDs are well-aligned for majority vote with suitable symbol duration and laboratory wireless channel conditions.

6.2 Limitations

Majority vote of short sequence is implemented via AirComp, but the MSE test data for validation for random longer sequence are left undone. The design of AirComp communication scheme need more time and programming work to be implemented on the hardware platform, resulting in not getting MSE analysis if random sequence test data to verify the application as planned.

In addition, due to limited time and author's limited experience on SDR, custom GNU Radio block and relevant programming on pluto FPGA are also left undone, which is suppose to form a complete set of digital OAC solution from the reference layer to the physical layer. If one can build a custom AirComp protocol similar to the Bluetooth protocol that can be implemented in WSN networks, it will be a huge boost for this field.

6.3 Future work

In the state-of-the-art, AirComp has been considered for a wide range of applications such as localization, wireless control systems, WSN, FEEL, and so on. Among these applications, distributed optimization is currently the leading research field of AirComp due to the advances in machine learning and artificial intelligence and the desire to use these techniques over wireless networks. In this thesis,

One biggest problem that AirComp has to address before mature implementation is improving the computation rate, and three main research directions are aiming at this challenge. The first direction is to design an AirComp scheme that takes the practical limitations into account. The next direction is the algorithm can be designed to facilitate AirComp, and the last one is to create protocols for AirComp with standards. In fact, AirComp has recently been discussed in AI/ML Technical Interest Group for IEEE 802.11 for distributed learning.

6.3.1 What has been left undone?

The prototype evades more stringent synchronization requirements, *i.e.,* strict time synchronization, and moreover phase synchronization. This means at least below work is left undone.

6.3.1.1 Random longer sequence majority vote

There are file source and file sink blocks for user to use custom data for transmission and obtain the received data for further analysis. However, the processing time of the file reader block are uncontrolled, thus it needs additional parameter in the system to compensate the time variance for different signal emission at the TX of EDs. the plan for whole AirComp system can be some data generated from FEEL apps stored as *e.g.,* TXT files in the memory in the EDs, then through transmission and SDR software, the final majority vote results are also stored as TXT files in the server, and error analysis can be implemented based on the files.

6.3.1.2 FGPA on PlutoSDR

The hardware of PlutoSDR provide FPGA for users who write custom software or HDL that runs directly on the PlutoSDR. However, connecting to Pluto over JTAG requires a standard JTAG programmer from Xilinx and additional connector assembly on PCBA.

6.3.2 Next obvious things to be done

customized GNU block can be built to realize simultaneous transmission of signals above different EDs, so that symbol Windows of longer random sequences were aligned, and MSE test of majority vote can be continued to complete the validation of this implementation.

In particular, the author of this thesis wishes to point out the reliability of AirComp remains a problem to be solved. Since the computation is processed during transmission, reliable AirComp means not only robust computation via various methods but also flow control and error control, like other communication protocols. In the context of AirComp, flow control means reaching the highest computation rate ASAP under certain channel condition with acceptable MSE. And error control means to re-transmit(re-calculate) the function when the MSE of AirComp is unacceptably high, or to detect the worst case when the channel is too noisy to be suitable for computation and can only achieve data transmission.

In the references, let Zotero or other tool fill this in for you. I suggest an extended version of the IEEE style, to include URLs, DOIs, ISBNs, etc., to make it easier for your reader to find them. This will make life easier for your opponents and examiner.
IEEE Editorial Style Manual: https://www.ieee.org/content/dam/ieee-org/ieee/web/org/conferences/style_references_manual.pdf

Låt Zotero eller annat verktyg fylla i det här för dig. Jag föreslår en utökad version av IEEE stil - att inkludera webbadresser, DOI, ISBN osv. - för att göra det lättare för läsaren att hitta dem. Detta kommer att göra livet lättare för dina opponenter och examinator.

References

[1] A. Şahin and R. Yang, "A survey on over-the-air computation," *IEEE Communications Surveys & Tutorials*, 2023. [Pages 1, 3, and 13.]

[2] B. Nazer and M. Gastpar, "Computation over multiple-access channels," *IEEE Transactions on information theory*, vol. 53, no. 10, pp. 3498–3516, 2007. [Pages 1 and 2.]

[3] A. Şahin, "A demonstration of over-the-air computation for federated edge learning," in *2022 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2022, pp. 1821–1827. [Pages 1 and 14.]

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282. [Page 1.]

[5] M. Goldenbaum, H. Boche, and S. Stańczak, "Harnessing interference for analog function computation in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 61, no. 20, pp. 4893–4906, 2013. [Pages 2 and 10.]

[6] ——, "Nomographic functions: Efficient computation in clustered gaussian sensor networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 4, pp. 2093–2105, 2014. [Page 2.]

[7] S. Katti, S. Gollakota, and D. Katabi, "Embracing wireless interference: Analog network coding," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 397–408, 2007. [Page 2.]

[8] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6463–6486, 2011. [Page 2.]

[9] B. Nazer, A. G. Dimakis, and M. Gastpar, “Local interference can accelerate gossip algorithms,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 876–887, 2011. [Page 2.]

[10] G. Zhu, J. Xu, K. Huang, and S. Cui, “Over-the-air computing for wireless data aggregation in massive iot,” *IEEE Wireless Communications*, vol. 28, no. 4, pp. 57–65, 2021. [Page 2.]

[11] S. Cai and V. K. Lau, “Modulation-free m2m communications for mission-critical applications,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 2, pp. 248–263, 2017. [Page 2.]

[12] X. Wu, S. Zhang, and A. Özgür, “Stac: Simultaneous transmitting and air computing in wireless data center networks,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 4024–4034, 2016. [Page 2.]

[13] W. Liu, X. Zang, Y. Li, and B. Vucetic, “Over-the-air computation systems: Optimization, analysis and scaling laws,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 8, pp. 5488–5502, 2020. [Page 3.]

[14] L. Chen, N. Zhao, Y. Chen, F. R. Yu, and G. Wei, “Over-the-air computation for iot networks: Computing multiple functions with antenna arrays,” *IEEE Internet of Things journal*, vol. 5, no. 6, pp. 5296–5306, 2018. [Pages 3 and 12.]

[15] S.-W. Jeon, C.-Y. Wang, and M. Gastpar, “Computation over gaussian networks with orthogonal components,” *IEEE Transactions on Information Theory*, vol. 60, no. 12, pp. 7841–7861, 2014. [Page 9.]

[16] M. Goldenbaum, H. Boche, and S. Stańczak, “Analog computation via wireless multiple-access channels: Universality and robustness,” in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 2921–2924. [Page 10.]

[17] M. Goldenbaum and S. Stanczak, “Robust analog function computation via wireless multiple-access channels,” *IEEE Transactions on Communications*, vol. 61, no. 9, pp. 3863–3877, 2013. [Page 10.]

[18] S. Limmer, J. Mohammadi, and S. Stańczak, “A simple algorithm for approximation by nomographic functions,” in *2015 53rd Annual Allerton*

Conference on Communication, Control, and Computing (Allerton). IEEE, 2015, pp. 453–458. [Page 10.]

[19] S. Huang, Y. Zhou, T. Wang, and Y. Shi, "Byzantine-resilient federated machine learning via over-the-air computation," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2021, pp. 1–6. [Page 12.]

[20] H. Guo, Y. Zhu, H. Ma, V. K. Lau, K. Huang, X. Li, H. Nong, and M. Zhou, "Over-the-air aggregation for federated learning: Waveform superposition and prototype validation," *Journal of Communications and Information Networks*, vol. 6, no. 4, pp. 429–442, 2021. [Page 13.]

[21] Y. Shao, D. Gündüz, and S. C. Liew, "Federated edge learning with misaligned over-the-air computation," *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 3951–3964, 2021. [Pages 13 and 31.]

[22] ——, "Bayesian over-the-air computation," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 3, pp. 589–606, 2022. [Page 13.]

[23] A. Şahin, B. Everette, and S. S. M. Hoque, "Over-the-air computation with dft-spread ofdm for federated edge learning," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2022, pp. 1886–1891. [Page 13.]

[24] W.-T. Chang, M. S. E. Mohamed, and R. Tandon, "On the necessity of aligning gradients for wireless federated learning," in *2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2021, pp. 226–230. [Page 14.]

[25] H. Hellström, V. Fodor, and C. Fischione, "Over-the-air federated learning with retransmissions," in *2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2021, pp. 291–295. [Page 14.]

[26] X. Cao, G. Zhu, J. Xu, Z. Wang, and S. Cui, "Optimized power control design for over-the-air federated edge learning," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 342–358, 2021. [Page 14.]

[27] A. Gadre, F. Yi, A. Rowe, B. Iannucci, and S. Kumar, "Quick (and dirty) aggregate queries on low-power wans," in *2020 19th ACM/IEEE*

International Conference on Information Processing in Sensor Networks (IPSN). IEEE, 2020, pp. 277–288. [Page 14.]

[28] W. Zhang, J. Xu, W. Xu, X. You, and W. Fu, "Worst-case design for ris-aided over-the-air computation with imperfect csi," *IEEE Communications Letters*, vol. 26, no. 9, pp. 2136–2140, 2022. [Page 14.]

[29] J. Mao, H. Yang, P. Qiu, J. Liu, and A. Yener, "Charles: Channel-quality-adaptive over-the-air federated learning over wireless networks," in *2022 IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC)*. IEEE, 2022, pp. 1–5. [Page 15.]

[30] F. Ang, L. Chen, N. Zhao, Y. Chen, and F. R. Yu, "Robust design for massive csi acquisition in analog function computation networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2361–2373, 2019. [Page 15.]

[31] U. Madhow, *Fundamentals of digital communication*. Cambridge university press, 2008. [Page 16.]

[32] X. Zhao, L. You, R. Cao, Y. Shao, and L. Fu, "Broadband digital over-the-air computation for asynchronous federated edge learning," in *ICC 2022-IEEE International Conference on Communications*. IEEE, 2022, pp. 5359–5364. [Page 18.]

[33] T. S. Rappaport, "Wireless communications–principles and practice, (the book end)." *Microwave Journal*, vol. 45, no. 12, pp. 128–129, 2002. [Page 18.]

[34] K. Bonne, "Some interesting links, book and other info sdr," https://github.com/on1arf/sdr/blob/master/links.md, accessed: March 2024. [Page 18.]

[35] EscapeTHU, "Gnuradio-mimo-ofdm," https://github.com/EscapeTHU/gnuradio-MIMO-OFDM/tree/main, accessed: March 2024. [Page 18.]

[36] A. Roozbeh, A. Sefidcon, and G. Q. Maguire, "Resource Monitoring in a Network Embedded Cloud: An Extension to OSPF-TE," in *2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing*. Dresden, Germany: IEEE, Dec. 2013. doi: 10.1109/UCC.2013.36. ISBN 978-0-7695-5152-4 pp. 139–146. [Online]. Available: http://ieeexplore.ieee.org/document/6809350/ [Page 19.]

[37] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 491–506, 2019. [Page 19.]

[38] G. Zhu, Y. Du, D. Gündüz, and K. Huang, "One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 2120–2135, 2020. [Page 19.]

[39] A. Şahin, "Distributed learning over a wireless network with non-coherent majority vote computation," *IEEE Transactions on Wireless Communications*, 2023. [Page 19.]

[40] O. Abari, H. Rahul, D. Katabi, and M. Pant, "Airshare: Distributed coherent transmission made seamless," in *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015, pp. 1742–1750. [Page 20.]

[41] K. Alemdar, D. Varshney, S. Mohanti, U. Muncuk, and K. Chowdhury, "Rfclock: Timing, phase and frequency synchronization for distributed wireless networks," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 15–27. [Page 20.]

[42] GNURadio, "Voltage controlled oscillator," https://wiki.gnuradio.org/index.php/VCO, accessed: March 2024. [Page 33.]

[43] ——, "Frequency xlating fir filter," https://wiki.gnuradio.org/index.php?title=Frequency_Xlating_FIR_Filter, accessed: March 2024. [Page 34.]

If you do not have an appendix, do not include the \cleardoublepage command below; otherwise, the last page number in the metadata will be one too large.

Appendix A

Supporting materials

Here is a place to add supporting material that can help others build upon your work. You can include files as attachments to the PDF file or indirectly via URLs. Alternatively, consider adding supporting material uploaded as separate files in DiVA.

The BibTeX references used in this thesis are attached.

Some source code relevant to this project can be found at `https://github.com/gqmaguirejr/E-learning` and `https://github.com/gqmaguirejr/Canvas-tools`.

Your reader can access the attached (embedded) files using a PDF tool such as Adobe Acrobat Reader using the paperclip icon in the left menu, as shown in Figure A.1 or by right-clicking on the push-pin icon in the PDF file and then using the menu to save the embedded file as shown in Figure A.2.

An argument for including supporting material in the PDF file is that it will be available to anyone who has a copy of the PDF file. As a result, they do not have to look elsewhere for this material. This comes at the cost of a larger PDF file. However, the embedded files are encoded into a compressed stream within the PDF file; thus, reducing the number of additional bytes. For example, the references.bib file that was used in this example is 10 617 B in size but only occupies 4 261 B in the PDF file.

DiVA is limited to ≈1 GB for each supporting file. If you have very large amounts of supporting material, you will probably want to use one of the data repositories. For additional help about this, contact KTH Library via researchdata@kth.se.
As of Spring 2024, there are plans to migrate this supporting data to

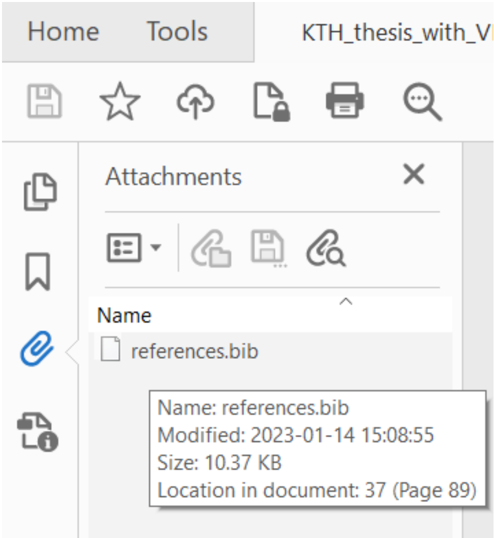

Figure A.1: Adobe Acrobat Reader using the paperclip icon for the attached references.bib file

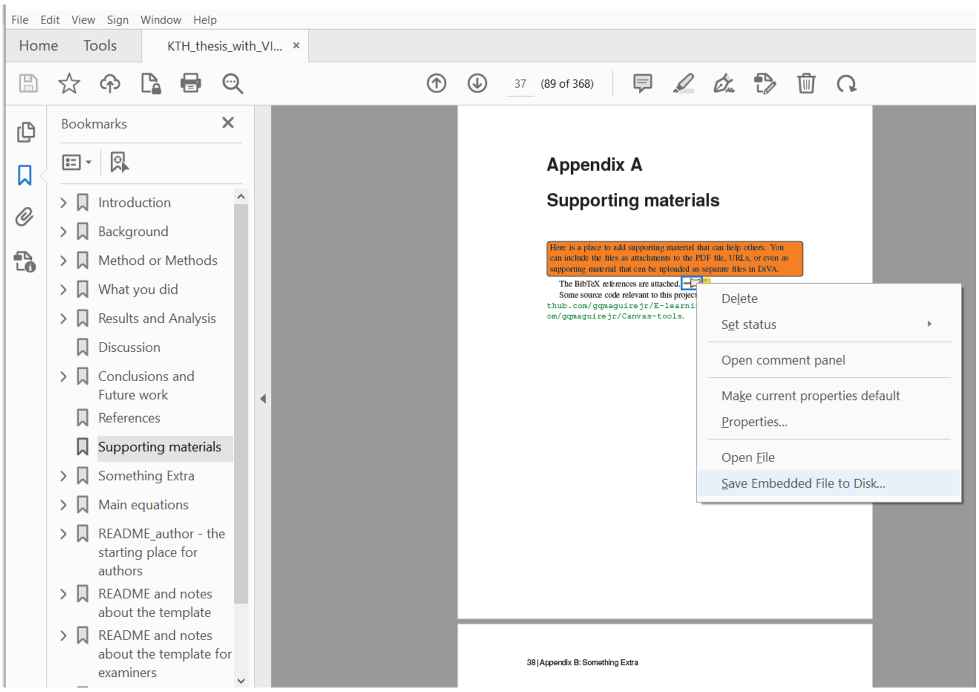

Figure A.2: Adobe Acrobat Reader after right-clicking on the push-pin icon for the attached references.bib file

TRITA-EECS-EX- 2023:0000

€€€€ For DIVA €€€€

{
"Author1": { "Last name": "Xiong",
"First name": "Qi",
"Local User Id": "u100001",
"E-mail": "qxiong@kth.se",
"organisation": {"L1": "School of Electrical Engineering and Computer Science",
}
},
"Cycle": "2",
"Course code": "EA260X",
"Credits": "30.0",
"Degree1": {"Educational program": "Master's Programme, Information and Network Engineering, 120 credits"
,"programcode": "TINNM"
,"Degree": "Master's Programme, Computer Science, 120 credits"
,"subjectArea": "Technology"
},
"Title": {
"Main title": "Digital Over-the-Air Computation with Software-defined Radio",
"Subtitle": "A Practical System Design",
"Language": "eng" },
"Alternative title": {
"Main title": "Digital over-the-air beräkning med mjukvarudefinierad radio",
"Subtitle": "En praktisk systemdesign",
"Language": "swe"
},
"Supervisor1": { "Last name": "Hellström",
"First name": "Henrik",
"Local User Id": "u100003",
"E-mail": "hhells@kth.se",
"organisation": {"L1": "School of Electrical Engineering and Computer Science",
"L2": "Computer Science" }
},
"Examiner1": { "Last name": "Fischione",
"First name": "Carlo",
"Local User Id": "u1d13i2c",
"E-mail": "carlofi@kth.se",
"organisation": {"L1": "School of Electrical Engineering and Computer Science",
"L2": "Computer Science" }
},
"National Subject Categories": "10201, 10206",
"Other information": {"Year": "2024", "Number of pages": "1,52"},
"Copyrightleft": "copyright",
"Series": { "Title of series": "TRITA-EECS-EX" , "No. in series": "2023:0000" },
"Opponents": { "Name": "Pengzhan Jiang"},
"Presentation": { "Date": "2022-05-30 13:00"
,"Language":"eng"
,"Room": "via Zoom https://kth-se.zoom.us/j/ddddddddddd"
,"Address": "Isafjordsgatan 22 (Kistagången 16)"
,"City": "Stockholm" },
"Number of lang instances": "10",
"Abstract[eng]": €€€€


```
%\generalExpl{Enter your abstract here!}
%Write an abstract that is about 250 and 350 words (1/2 A4-page)  with the following components:



Traditional network scheme treats communication and computation are regarded as independent tasks,
which brings the benefits of minimalist system complexity, high freedom of network architecture, and
easier troubleshooting and maintaining. However, with the rapid growth of artificial intelligence and
machine learning, computation-oriented tasks are more and more popular with giant internet companies,
which call for a rising demand for function transmission. \gls{OAC} becomes new research trend in
\gls{WSN} and distributed learning networks. The goal of AirComp is to compute functions of messages
by exploiting electromagnetic interference. The vast majority of the work is based on theoretical
methods and numerical simulation. There is a call for implementation of these methods over real
wireless channels, using transceiver chains implemented in hardware.

In this thesis, an \gls{OAC} scheme to calculate the \gls{MV} is proposed for \gls{FEEL}, and an
practical implementation of \gls{OAC}  with the \gls{SDR} device ADALM-PLUTO is demonstrated. This
thesis addresses the theoretical study of digital AirComp by building an experimental platform and
verifying the theory with the \gls{SDR} device ADALM-PLUTO. Through simulations and experiment, it is
proved that proposed AirComp scheme can provide a good MV performance even when the
time-synchronization and the power control are not ideal under heterogeneous data distribution
scenarios.

This thesis also addresses and analyzes the challenges encountered during the construction of the
hardware experimental platform, proposing alternative solutions. Additionally, it discusses the
```

```
platform's limitations and suggests directions for future work for AirComp.

\begin{comment}


% key parts of the abstract
\begin{itemize}
  \item What is the topic area? (optional) Introduces the subject area for the project.
  \item Short problem statement
  \item Why was this problem worth a Bachelor's/'Masters thesis project? (\ie, why is the problem
  both significant and of a suitable degree of difficulty for a Bachelor's/'Masters thesis project?
  Why has no one else solved it yet?)
  \item How did you solve the problem? What was your method/insight?
  \item Results/Conclusions/Consequences/Impact: What are your key results/\linebreak[4]conclusions?
  What will others do based on your results? What can be done now that you have finished - that could
  not be done before your thesis project was completed?
\end{itemize}
\end{comment}
```

€€€€,
"Keywords[eng]": €€€€
Over-the-air Computation, Software-defined Radio, Federated Edge Learning, Frequency Shift Keying, Majority Vote, Synchronization €€€€,
"Abstract[swe]": €€€€

```
\generalExpl{Enter your Swedish abstract or summary here!}
\sweExpl{Alla avhandlingar vid KTH \textbf{måste ha} ett abstrakt på både \textit{engelska} och
\textit{svenska}.\\
Om du skriver din avhandling på svenska ska detta göras först (och placera det som det första
abstraktet) - och du bör revidera det vid behov.}

\engExpl{If you are writing your thesis in English, you can leave this until the draft version that
goes to your opponent for the written opposition. In this way, you can provide the English and
Swedish abstract/summary information that can be used in the announcement for your oral
presentation.\\If you are writing your thesis in English, then this section can be a summary targeted
at a more general reader. However, if you are writing your thesis in Swedish, then the reverse is
true - your abstract should be for your target audience, while an English summary can be written
targeted at a more general audience.\\This means that the English abstract and Swedish sammnfattning
or Swedish abstract and English summary need not be literal translations of each other.}

\warningExpl{Do not use the \textbackslash glspl\{\} command in an abstract that is not in English,
as my programs do not know how to generate plurals in other languages. Instead, you will need to
spell these terms out or give the proper plural form. In fact, it is a good idea not to use the
glossary commands at all in an abstract/summary in a language other than the language used in the
\texttt{acronyms.tex file} - since the glossary package does \textbf{not} support use of more than
one language.}

\engExpl{The abstract in the language used for the thesis should be the first abstract, while the
Summary/Sammanfattning in the other language can follow}
```

€€€€,
"Keywords[swe]": €€€€
Luftberäkning, Programvarudefinierad Radio, Federated Edge Learning, Frekvensskiftnyckel, Majoritetsomröstning, Synkronisering €€€€,
"Abstract[fre]": €€€€

```
Résumé en français.
```

€€€€,
"Keywords[fre]": €€€€
Calcul en direct, Radio définie par logiciel, Apprentissage fédéré, Modulation par décalage de fréquence, Vote majoritaire, Synchronisation
€€€€,
"Abstract[spa]": €€€€

```
Résumé en espagnol.
```

€€€€,
"Keywords[spa]": €€€€
Computación inalámbrica, radio definida por software, aprendizaje perimetral federado, codificación por cambio de frecuencia, voto mayoritario, sincronización €€€€,
"Abstract[ita]": €€€€

```
Sommario in italiano.
```

€€€€,
"Keywords[ita]": €€€€
Calcolo over-the-air, Radio definita dal software, Federated Edge Learning, Frequency Shift Keying, Voto a maggioranza, Sincronizzazione
€€€€,
"Abstract[nor]": €€€€

```
Sammendrag på norsk.
```

€€€€,
"Keywords[nor]": €€€€
Over-the-air-beregning, Programvaredefinert radio, Føderert kantlæring, Frekvensskiftnøkling, Flertallsavstemning, Synkronisering €€€€,
"Abstract[ger]": €€€€

Zusammenfassung in Deutsch.

€€€€,
"Keywords[ger]": €€€€
Over-the-Air-Berechnung, Softwaredefiniertes Radio, Federated Edge Learning, Frequenzumtastung, Mehrheitsabstimmung, Synchronisation €€€€,
"Abstract[dan]": €€€€

Abstrakt på dansk.

€€€€,
"Keywords[dan]": €€€€
Over-the-air-beregning, Softwaredefineret radio, Fødereret edge learning, Frequency shift keying, Flertalsafstemning, Synkronisering €€€€,
"Abstract[dut]": €€€€

Samenvatting in het Nederlands.

€€€€,
"Keywords[dut]": €€€€
Over-the-air Computation, Softwareradio, Randleren, Frequency Shift Keying, Meerderheid van stemmen, Synchronisatie €€€€,
"Abstract[est]": €€€€

Eesti keeles kokkuvõte.

€€€€,
"Keywords[est]": €€€€
Üle-eetriline arvutamine, Tarkvaral põhinev raadio, Föderaalne servaõpe, Sageduse nihkepõhine võtmine, Enamushääletus, Sünkroniseerimine. €€€€,
}

acronyms.tex

```
%%% Local Variables:
%%% mode: latex
%%% TeX-master: t
%%% End:
% The following command is used with glossaries-extra
\setabbreviationstyle[acronym]{long-short}
% The form of the entries in this file is \newacronym{label}{acronym}{phrase}
%                                      or \newacronym[options]{label}{acronym}{phrase}
% see "User Manual for glossaries.sty" for the  details about the options, one example is shown below
% note the specification of the long form plural in the line below
\newacronym[longplural={Debugging Information Entities}]{DIE}{DIE}{Debugging Information Entity}
%
% The following example also uses options
\newacronym[shortplural={OSes}, firstplural={operating systems (OSes)}]{OS}{OS}{operating system}

% note the use of a non-breaking dash in long text for the following acronym
\newacronym{IQL}{IQL}{Independent -QLearning}

% example of putting in a trademark on first expansion
\newacronym[first={NVIDIA OpenSHMEM Library (NVSHMEM\texttrademark)}]{NVSHMEM}{NVSHMEM}{NVIDIA OpenSHMEM Library}

\newacronym{KTH}{KTH}{KTH Royal Institute of Technology}

\newacronym{LAN}{LAN}{Local Area Network}
\newacronym{VM}{VM}{virtual machine}
% note the use of a non-breaking dash in the following acronym
\newacronym{WiFi}{-WiFi}{Wireless Fidelity}

\newacronym{WLAN}{WLAN}{Wireless Local Area Network}
\newacronym{UN}{UN}{United Nations}
\newacronym{SDG}{SDG}{Sustainable Development Goal}

% Acronyms used in my thesis added here by Qi
\newacronym{OAC}{AirComp}{Over-the-air Computation}
\newacronym{FL}{FL}{Federated Learning}
\newacronym{FEEL}{FEEL}{Federated Edge Learning}
\newacronym{ED}{ED}{Edge Device}
\newacronym{ES}{ES}{Edge Server}
\newacronym{NN}{NN}{Neural Network}
\newacronym{LSTM}{LSTM}{Long Short-term Memory Network}
\newacronym{WSN}{WSN}{Wireless Sensor Network}
\newacronym{AI}{AI}{Artificial Intelligence}
\newacronym{ML}{ML}{Machine Learning}
\newacronym{LDPC}{LDPC}{Low-Density Parity-Check}
\newacronym{QPSK}{QPSK}{Quadrature Phase Shift Keying}
\newacronym{QAM}{QAM}{Quadrature Amplitude Modulation}
\newacronym{OFDM}{OFDM}{Orthogonal Frequency Division Multiplexing}
\newacronym{SDR}{SDR}{Software-defined Radio}
\newacronym{CFO}{CFO}{Carrier Frequency Offset}
\newacronym{TO}{TO}{Time Offest}
\newacronym{PO}{PO}{Phase Offset}
\newacronym{MSE}{MSE}{Mean Squared Error}
\newacronym{FSK}{FSK}{Frequency Shift Keying}
\newacronym{GPS}{GPS}{Global Positioning System}
\newacronym{FPGA}{FPGA}{Field Programmable Gate Array}
\newacronym{MV}{MV}{Majority Voting}
\newacronym{CP}{CP}{Cyclic Prefix}
\newacronym{CSI}{CSI}{Channel State Information}
\newacronym{MAC}{MAC}{Multiple Access Channel}
\newacronym{CER}{CER}{Computation Error Rate}
\newacronym{FEE}{FEE}{Function Estimation Error}
\newacronym{NMSE}{NMSE}{Normalized Mean Square Error}
\newacronym{MSFE}{MSFE}{Mean Squared Function Error}
\newacronym{BER}{BER}{Bit Error Rate}
\newacronym{SNR}{SNR}{Signal-to-noise Ratio}
\newacronym{AWGN}{AWGN}{Additive White Gaussian Noise}
\newacronym{LOS}{LoS}{Line of Sight}
\newacronym{LZW}{LZW}{Lempel-Ziv-Welch}
\newacronym{AM}{AM}{Amplitude Modulation}
\newacronym{FM}{FM}{Frequency Modulation}
\newacronym{PM}{PM}{phase Modulation}
\newacronym{COTS}{COTS}{commercial-off-the-shelf}
\newacronym{DAC}{DAC}{Digital-to-analog converter}
\newacronym{RF}{RF}{Radio Frequency}
\newacronym{USRP}{USRP}{Universal Software Radio Peripheral}
\newacronym{MCU}{MCU}{Micro Controller Unit}
```

```
\newacronym{OOK}{OOK}{On-Off Keying}
\newacronym{VCO}{VCO}{Voltage controlled oscillator}
\newacronym{RTTY}{RTTY}{Radio TeleTYpe}
\newacronym{FXFIR}{Frequency Xlating FIR Filter}{Frequency Translating Finite Impulse Response Filter}
\newacronym{IoT}{IoT}{Internet of ThingS}
```