

System-On-Chip based GRAPPA Reconstruction



Final Year Project Report

Presented

By

Tayaba Gul

CIIT/FA19-EEE-009/ISB

Nimra Naeem

CIIT/FA19-BEE-167/ISB

In Partial Fulfillment

of the Requirement for the Degree of

Bachelors of Science in Electrical Engineering

**DEPARTMENT OF ELECTRICAL AND COMPUTER
ENGINEERING**

COMSATS UNIVERSITY ISLAMABAD


May 2023

Declaration

We hereby declare that this project neither as a whole nor as a part of has been copied from any source. It is further declared that we have developed this project and the accompanying report entirely based on our personal efforts made under the sincere guidance of our supervisor. No portion of the work presented in this report has been submitted in support of any other degree or qualification of this or any other University or Institute of learning, if found we shall stand responsible.

Signature: 

Name: Tayaba Gul

Signature: 

Name: Nimra Naeem

COMSATS UNIVERSITY ISLAMABAD

July 2023

System-On-Chip based GRAPPA Reconstruction

An Undergraduate Final Year Project Report submitted to the

Department of

ELECTRICAL AND COMPUTER ENGINEERING

As a Partial Fulfillment for the award of Degree

Bachelors of Science in Electrical Engineering

by

Name	Registration Number
Tayaba Gul	CIIT/FA19-EEE-009/ISB
Nimra Naeem	CIIT/FA19-BEE-167/ISB

Supervised by

Dr. Hammad Omer

Tenured Associate Professor,

Department of Electrical and Computer Engineering

CU Islamabad

COMSATS UNIVERSITY ISLAMABAD

May 2023

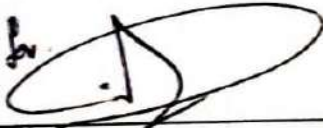
Final Approval
This Project Titled
System-On-Chip based GRAPPA Reconstruction
Submitted for the Degree of
Bachelor of Science in Electrical Engineering

by

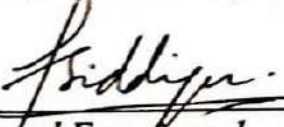
Name	Registration Number
Tayaba Gul	CIIT/FA19-EEE-009/ISB
Nimra Naeem	CIIT/FA19-BEE-167/ISB

has been approved for

COMSATS UNIVERSITY ISLAMABAD



Supervisor
Dr. Hammad Omer
Tenured Associate Professor



Internal Examiner-1
Dr. M. Faisal Siddiqui
Associate Professor



Internal Examiner-2
Dr. Syed Saud Naqvi
Associate Professor



External Examiner
Dr. Salman Abdul Ghafoor
Professor
SEECs, NUST Islamabad.

Dedication

We humbly dedicate our work to Allah, our Creator, The Almighty, and the source of strength and knowledge. We recognize that without His blessings, guidance, and mercy, this project would not have been possible.

We extend our sincerest gratitude to our beloved parents and respectful teachers for their unwavering support throughout the development of this project. Their love, encouragement, and guidance have been a source of inspiration and motivation. Without them, the challenges we faced would have been much more difficult to overcome.

We acknowledge that this project is not only a reflection of our own efforts, but also the contributions of those who have supported and guided us along the way. We hope that this work may serve as a testament to the importance of faith, family, and education in our lives.

Acknowledgements

Thoroughly thankful to Almighty Allah, who gave us the strength and patience to complete our Project. We acknowledge the invaluable contribution of Dr. Hammad Omer and Dr. Omair Inam, as our advisors, generously offering their time and expertise. Without their guidance and encouragement, this Final Year Project would not have been possible.

We are also thankful to Mr. Abdul Basit, currently doing his PHD at Comsats University Islamabad, who helped us understand the implementation methods and toolchains. His help was crucial in shaping our project and making it a success.

We wish to extend our heartfelt appreciation to our parents, family, and teachers for their unwavering support, prayers, and encouragement throughout this endeavor. Their blessings and good wishes have been a constant source of inspiration and motivation.

Finally, we are grateful to the academic staff of the ECE for their kindness and support. Their dedication and commitment to our education have been instrumental in shaping our academic and personal growth.

Tayaba Gul

Nimra Naeem

Table of Contents

List of Acronyms..... viii

List of Figure.....x

List of Table.....xi

Abstract..... 1

1. Introduction 2

1.1 Purpose and Overview.....2

1.1.1 Introduction to Parallel Magnetic Resonance Imaging (pMRI).....2

1.1.2 Challenges in pMRI Reconstruction.....3

1.2 Problem Statement.....3

1.3 Proposed Solution..... 4

1.4 Key Objectives.....5

1.5 Benefits of the Project5

1.6 Motivation.....6

1.7 Broader Impact (UN SDGs)..... 7

1.7.1 Targeted SDGs.....7

1.7.2 Potential Mapping (checklist)..... 8

2. Literature Review 10

2.1 Magnetic Resonance Imaging (MRI).....10

2.1.1 Background.....10

2.1.1 Technology advancement in MRI.....11

2.2 Parallel Magnetic Resonance Imaging(pMRI)...11

2.3 GRAPPA.....14

3. Proposed Methodologies for fast GRAPPA Reconstruction 19

3.1	Overview and Background.....	19
3.2	Channel compression in pMRI.....	19
3.2.1	Principal Component Analysis.....	20
3.2.2	Singular Value Decomposition.....	21
3.3	QR decomposition-based Channel Compression.....	22
3.3.1	QR Decomposition and Modified Gram-Schmidt Method.....	22
3.4	Random Projection in GRAPPA.....	24
3.4.1	Overview of Random Projection Method.....	24
3.4.2	Integration of Random Projection in GRAPPA Calibration Phase.....	24
3.4.3	Proposed Sparse Randomly Projected Matrix.....	26
3.5	System-on-Chip for GRAPPA reconstruction.....	27
3.5.1	FPGA based hardware accelerator.....	28
3.5.2	Datasets.....	28

4. SOC based GRAPPA Reconstruction 30

4.1	Channel Compression Accelerator.....	30
4.1.1	Random Projection Accelerator.....	30
4.1.2	Integration	32
4.2	Discussion.....	34

5. Results and Discussions **30**

4.1 Results..... 30

 4.1.1 Results of Channel Compression..... 30

 4.1.2 Random Projection Results..... 32

4.2 Discussion.....34

6. Conclusion and Future Work **36**

5.1 Conclusion.....36

5.2 Future Work.....37

 5.2.1 Enhanced Dimension Reduction Techniques.....37

 5.2.2 Integration with Deep Learning Approaches.....37

 5.2.3 Real-Time Implementation Considerations.....37

Bibliography **39**

List of Acronyms

MRI.....	Magnetic Resonance Imaging
pMRI.....	Parallel Magnetic Resonance Imaging
GRAPPA.....	Generalized Autocalibrating Partially Parallel Acquisitions
RP.....	Random Projection
PCA.....	Principal Component Analysis
FPGA.....	Field Programmable Gate Arrays
SoC.....	System-On-Chip

List of Figures

2.1: Excitation and Relaxation of Protons in MRI when an RF Pulse is applied	10
2.2 Parallel MRI acquisition and reconstruction illustration	12
2.3 The image series compares R = 2 accelerated acquisitions of the human heart using different pMRI reconstruction algorithms	14
2.4 The image depicts the steps involved in parallel magnetic. resonance imaging (pMRI)GRAPPA reconstruction.....	15
2.5 Schematic description of GRAPPA Reconstruction method using multiple lines from all coils to fit one line in a specific coil.....	17
3.1 Principal Components and Eigen Value Normalization	21
3.2 Channel Compression Algorithm Workflow	23
3.3 Pseudo code of QR decomposition using modified. Gram Schmidt Method.....	23
3.4 Block Diagram of the proposed Random Projection method with GRAPPA Reconstruction	27
3.5: Block Diagram of SoC design.....	28
4.1 This shows the comparative analysis of no of non-zero elements in proposed highly sparse random projected matrix with conventional Random projected matrix.....	33

List of Tables

3.1	Features Dataset used.....	29
4.1	Comparison of channel compression results for 30 coils cardiac data with acceleration factor (AF) = 2, kernel size 11 x 4 and 64 ACS	31
4.2	Comparison of the reconstruction results after coil compression for 30 coils cardiac data set.....	32
4.3	Comparative analysis of SNR and RMSE of a 12 coil cardiac data set, In comparison of proposed Random Projection Method with conventional Random Projection Method for various configuration settings.....	33

Abstract

pMRI (Parallel Magnetic Resonance Imaging) techniques have attracted the attention of researchers in MRI as it allows faster image acquisition without any increased gradient and additional RF (Radio frequency) power. Several methods have been proposed in literature to reconstruct images using pMRI methods including GRAPPA (Generalize Autocalibrating Partial Parallel Acquisition). GRAPPA enables under-sampled data acquisition in MRI and reconstructs artifact free images while using multiple receiver coils to reduce MRI scan time. However, this scan time reduction causes long image reconstruction time because as the number of receiver coils increases, the time to reconstruct images from each coil increases in GRAPPA.

The reconstruction time in GRAPPA can be reduced using Random Projection Method and Channel Compression. This project is aimed to design an accelerator based on System-On-Chip system using VIVADO High-Level-Synthesis (HLS) tool. The proposed design consists of two hardware accelerators for Random Projection and Channel Compression. The accelerators have been implemented on Zynq System-On-Chip board. The accelerators are coupled with GRAPPA algorithm. The proposed design accelerates the reconstruction process of undersampled parallel MRI data, as well as providing reduced power consumption, low cost, and higher reliability than the contemporary system. The proposed SoC is more efficient than the multi-chip systems, as fewer packages in the system reduces assembly cost and reduces the power consumption.

Chapter-01

Introduction

1.1 Purpose and Overview:

Magnetic Resonance Imaging (MRI) [1] is a noninvasive test used to produce detailed medical images in order to examine the internal structure, anatomy, and organs of the human body at the tissue level. The medical images produced by MRI are used to diagnose and treat different medical conditions including tumors, cancers, soft tissue injuries such as damaged ligaments, joint injuries, spinal injuries, internal organs injury including brain, heart, lungs, kidney, digestive organs, etc.

Unlike other clinical diagnoses, MRI doesn't produce any harmful radiation. However, longer scan time and image reconstruction time are major drawbacks of MRI which cause discomfort to the patients and poor image quality due to patient's mobility. To address these challenges, researchers have been working on various technical modifications to MRI processors. These modifications aim to reduce scan time by implementing real-time mathematical acceleration techniques. By improving the efficiency of data acquisition and reconstruction algorithms, these modifications can help minimize patient discomfort and enhance overall image quality.

1.1.2 Introduction to Parallel Magnetic Resonance Imaging (pMRI):

Within the past few years, parallel imaging techniques have been widely used in literature as well as commercially to accelerate MRI scan. Parallel Magnetic Resonance Imaging (pMRI) [2] techniques are used to enhance MRI methods virtually, allowing faster data acquisition without any increased gradient or additional radio frequency. parallel MRI techniques such as SENSE (Sensitivity Encoding) [3] and GRAPPA (Generalize Auto calibrating Partial Parallel Acquisition) [4] have been widely used in clinical and research settings with an aim to reduce scan times and improve patient comfort, particularly for patients who have difficulty holding still or are unable to tolerate longer scan times. pMRI techniques are also used to improve image quality in areas where conventional MRI may have limitations, such as imaging the beating heart for diagnosing cardiovascular diseases and planning interventions, functional MRI (fMRI) [5] to

map brain activity by measuring changes in blood flow, musculoskeletal imaging to diagnose conditions such as osteoarthritis and muscle injuries, interventional MRI to guide the placement of instruments and monitor the progress of the procedures such as biopsy, tumor ablation.

1.1.3 Challenges in pMRI Image Reconstruction:

pMRI offers the advantage of faster image acquisition using undersampled data, but it introduces challenges in the reconstruction of high-quality images from the acquired undersampled data. The acquired data is undersampled, meaning that not all spatial frequencies are fully captured. The under-sampling leads to artifacts and image distortions in the reconstructed images.

To overcome these challenges, various image reconstruction methods have been proposed in the literature, and one such method is the Generalized Autocalibrating Partial Parallel Acquisition (GRAPPA) technique. GRAPPA reconstructs MR images from the acquired under-sampled raw data (known as k-space data in MRI), by using information from the k-space data acquired during a calibration scan. The reconstruction process typically involves a series of mathematical operations to interpolate the missing k-space data in each receiver coil and reconstructs the final image. These operations can be computationally intensive, and the amount of computation required can increase with the number of receiver coils used.

1.2 Problem Statement:

The problem addressed in this research project is the long image reconstruction time associated with GRAPPA. GRAPPA enables faster image acquisition using multiple receiver coils, but the reconstruction process increases as the number of coils increases. To overcome this, efficient software-based solutions have been developed, including channel compression and dimension reduction. Channel compression aims to reduce the number of required channels for MRI reconstruction, while dimension reduction reduces the higher-dimensional calibration data of GRAPPA to a lower-dimensional space without compromising image quality.

1.3 Proposed Solution:

Real-time MRI reconstruction requires quick processing and analysis of large amounts of data, which can be a significant computational challenge. MRI reconstruction is a computationally intensive task that involves processing large amounts of data in real-time, making it an ideal candidate for hardware acceleration. In this work, two FPGA based accelerators embedded on chip are proposed for GRAPPA Reconstruction. The proposed accelerator for channel compression employs PCA based channel compression to reduce the computation complexity of the GRAPPA algorithm and the proposed accelerator for dimension reduction using random projection [6] incorporated in GRAPPA to reduce dimensions of the calibration phase. The proposed solution of dimension reduction aims to empirically select a random projection matrix to ensure preservation of the relevant information in the data, enabling an evaluation of the impact of dimension reduction on image quality and reconstruction accuracy. The proposed solution of channel compression is aimed to reduce the number of receiver coils by compressing the acquired data to fewer channels. The compression process is done using PCA [6] (Principal Component Analysis) which extracts essential components of data and removes the redundant information. The FPGA based accelerator for channel compression reduces the data that needs to be stored and processed by the GRAPPA algorithm, which reduces imaging time and storage resources.

The proposed accelerators can be an excellent choice for real-time MRI reconstruction, as they can offer high performance and low memory as well as processing requirements. proposed designs intend to utilize the customized FPGA based accelerators; to perform PCA based coil compression, and random projection, making GRAPPA highly efficient for real-time MRI reconstruction. The integration of FPGA-based accelerators into the GRAPPA reconstruction workflow holds significant potential for speeding up the reconstruction process and enhancing overall efficiency in real-time MRI. These accelerators can minimize the time required for image reconstruction, enabling faster diagnosis, and reducing patient discomfort during scanning. Additionally, the efficient utilization of computational resources contributes to improved resource management and cost-effectiveness in medical imaging applications.

1.4 Key Objectives:

The objective of this research-based project is to design a SoC system using VIVADO High Level Synthesis (HLS) tool. This SoC based system allows fast MRI reconstruction for real time clinical MRI applications. Following are the key objectives of this project:

1. To reduce the reconstruction time of GRAPPA without compromising the image quality. This process can enable faster diagnosis to allow patient comfortability and reduce the motion artifacts in the images.
2. To reduce the computational complexity of GRAPPA reconstruction process, thus reducing cost and power requirements of MRI systems.
3. This system can enable the development of smaller, more efficient, and portable MRI systems that can be used in far-off areas.
4. Another key objective of this project is to improve the overall efficiency of MRI system, improving patients' comfortability and fast reconstruction of high-quality MR imaging.

1.5 Benefits of the project:

The proposed SoC will significantly reduce the computation time and complexity in GRAPPA reconstruction. The development of the proposed hardware accelerators will reconstruct the MR images with higher frame rates, making the proposed design a valuable tool in various clinical applications. This project will offer several benefits:

Improved temporal resolution: Frame rate is a critical factor in defining the temporal resolution of real-time MRI. More detailed images of dynamic physiological processes with better temporal resolution are provided by higher frame rates. Thus, it allows better understanding to clinicians, better diagnosis, and better treatment plans.

Reduced motion artifacts: When the patient or organ under observation is in motion, it creates motion artifacts in the MR images. The motion artifacts can be reduced by higher frame rates which helps in better diagnosis and improved results.

Improved precision in image-guided interventions: Real-time MRI is used to guide procedures such as surgeries and biopsies. High frame rates provide accurate and precise visualizations and reduce risks of complications during the procedure.

Better measurement of physiological parameters: The accuracy of measuring physiological parameters such as blood flow, heart function etc. can also be improved with high frame rates. Thus, it provides better understanding of pathophysiology of diseases and can aid in treatment and monitoring.

Faster scan times: High frame rates provide shorter real-time MRI scans. This can increase patient comfort and reduce anxiety during the scan.

Dynamic imaging: Our proposed solution is a real-time application thus it will provide visualization of physiological processes in motion. It can also help in diagnosis, progression, and treatment of diseases better.

1.6 Motivation:

The key motivation of this research project is to address the computational complexity of the GRAPPA reconstruction algorithm. pMRI techniques have gained significant attention from researchers and technologists in the field of MRI, as it enables faster image acquisition without any additional RF power and increased gradient. This capability of pMRI has a potential to revolutionize the MRI systems due to its increased throughput, patient comfortability, and enhanced diagnostic capabilities.

GRAPPA is one of the main widely used methods for image reconstruction in pMRI. It reduces scan time by allowing acquisition of under-sampled data using multiple receiver coils. However, as the number of receiver coils increases, the image reconstruction time for each coil also increases. This limitation compromises the potential of pMRI techniques.

This project aims to design two accelerators (Channel Compression and Random Projection) based on the SoC system using the VIVADO High Level Synthesis (HLS) to overcome the limitations of GRAPPA. The novelty of this research includes the design of two accelerators for efficient GRAPPA reconstruction i.e. (i) an accelerator for Channel Compression to reduce the

number of receiver coils and (ii) an accelerator for Random Projection to reduce the complexity of calibration phase of GRAPPA.

The proposed SoC design offers many advantages over the conventional multi-chip systems such as less cost and less power consumption.

Successful implementation of this project can provide excellent benefits for pMRI techniques e.g., GRAPPA including shorter scan time, patient comfort, accurate diagnosis, less power consumption, and cost efficiency, etc.

1.7 Broader Impact (UN SDGs):

1.7.1 Targeted SDGs:

This project has a broader impact and aligns with several United Nations Sustainable Development Goals (SDGs). It contributes to following SDGs:

Goal 3: Good Health and Well-being: By improving the temporal resolution of real-time MRI, the project enhances the ability to diagnose and monitor diseases, leading to better healthcare outcomes and improved well-being for patients.

Goal 9: Industry, Innovation, and Infrastructure: The development of an FPGA-based hardware accelerator for MRI reconstruction represents a technological innovation in the field of medical imaging. It contributes to infrastructure development by advancing imaging technologies and improving their accessibility.

Goal 11: Sustainable Cities and Communities: By reducing scan times and improving efficiency in MRI, the project helps optimize healthcare resources in urban areas, leading to more sustainable healthcare systems and better access to medical services.

Goal 17: Partnerships for the Goals: Our project encourages interdisciplinary collaboration and knowledge sharing among researchers, engineers and healthcare professionals which is critical in the development of medical imaging technologies.

1.7.2 Potential Mapping:

This project has the potential to bring about several positive outcomes:

Improved Diagnostic Accuracy: This project helps doctors see physiological processes more clearly and accurately by reducing motion artifacts which leads to improved diagnostic accuracy and better treatment planning.

Enhanced Patient Experience: Reduced scan times during MRI lead to less discomfort in the patient experience, which ultimately improves patient satisfaction and promotes better adherence to medical procedures.

Technological Advancement: The development of FPGA-based hardware accelerators for GRAPPA reconstruction illustrates the potential for technological advancements in MRI reconstruction, hence opening novel opportunities for real-time imaging and dynamic process visualization.

Sustainable Healthcare Practices: The optimization of MRI resources through reduced scan times contributes to sustainable healthcare practices, ensuring an efficient use of medical equipment, energy, and resources.

Accessible Healthcare: By increasing the efficiency and speed of MRI scans, it will increase access to high-quality healthcare services, particularly in places with limited resources or underserved populations.

Research and Innovation: The collaboration between researchers, engineers and healthcare professionals will promote a culture of research and innovation between academia and industry, leading to continuous advancements in medical imaging technologies.

The project's distinctive impact and potential mapping can be expanded based on the project's scope and local healthcare contexts.

Chapter-02

Literature Review

2.1 Magnetic Resonance Imaging (MRI):

2.1.1 Background

Magnetic Resonance Imaging (MRI) involves a combination of advanced science and engineering, including the use of superconductivity, cryogenics, quantum physics, digital and computer technology, and all within the radiology department of a hospital [8]. There are other imaging techniques such as X-rays and computed tomography (CT scans) which are being used in medical imaging to diagnose and study anatomy of the human body, but MRI is an efficient and harmless medical imaging technique.

In the 1970s, MRI scanners were the domain of engineers and scientists. In the early days, the medical scanning technique named Nuclear Magnetic Resonance (NMR) imaging [9] was introduced; later its name was modified to MRI. Figure 2.1 demonstrates a visual representation of NMR working.

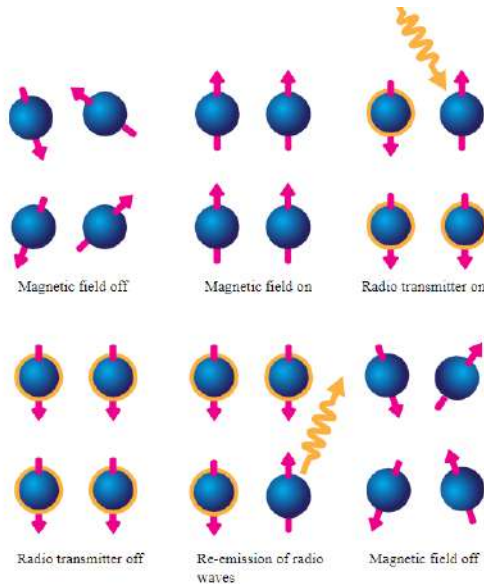


Figure 2.1: Excitation and Relaxation of Protons in MRI when an RF Pulse is applied [10]

It works on the basic principle of nuclear magnetic resonance, where the nuclei of certain atoms align in the presence of strong magnetic fields, such as hydrogen protons in water molecules. When radio frequency (RF) pulses are applied, the aligned protons emit high energy signals which are detected and processed to create images.

2.1.2 Technology Advancement in MRI:

Magnetic Resonance Imaging (MRI) has evolved significantly since its inception in the 1970s. Some improvements and advancements of MRI technology [9] are as follow:

Image Quality: MRI systems have shown remarkable improvements in image quality over the years. With the help of researchers and technologists, MRI hardware has improved such as stronger magnetic fields and high-performance radio frequency to enhance image clarity; thus, ensuring better diagnosis and treatment of diseases.

Magnet Field Strength: The strength of MRI scanner has enhanced day by day. In the 1970s, the normal magnetic field strength of MRI magnets ranged from 0.1 to 0.3 Tesla (T). Today, MRI scanners with magnetic field strengths of 1.5 T and 3 T are commonly used for clinical purposes. A stronger magnetic field provides a high signal-to-noise ratio and improved image quality.

Faster Imaging: MRI scan speed has improved by the development of techniques such as parallel imaging, thus enabling patient comfort, quick examination, and less power consumption.

2.2 Parallel Magnetic Resonance Imaging (pMRI):

In clinical magnetic resonance imaging (MRI), imaging speed is crucial alongside image contrast. However, current MRI scanners are already operating at their maximum potential speed due to technical and physiological challenges associated with rapidly switching magnetic field gradients. Parallel MRI (pMRI) has emerged as a solution to decrease acquisition time without requiring further enhancements in gradient performance [10].

The pMRI utilizes the spatial sensitivity information from an array of multiple receiver surface-coils to partially replace the time-consuming process of spatial encoding. This means that fewer encoding steps are required for image acquisition during the process, resulting in faster image

acquisition while maintaining full spatial resolution and contrast. Essentially, pMRI improves both the speed and the quality of MRI scans. It allows for quicker image acquisition at a given level of detail and enables higher spatial resolution within a specific imaging time. Furthermore, pMRI can help reduce susceptibility artifacts which appear as distortions in the images by minimizing the length of the echo train used in single and multi-shot imaging. Figure 2.2 demonstrates the Parallel MRI acquisition and reconstruction illustration.

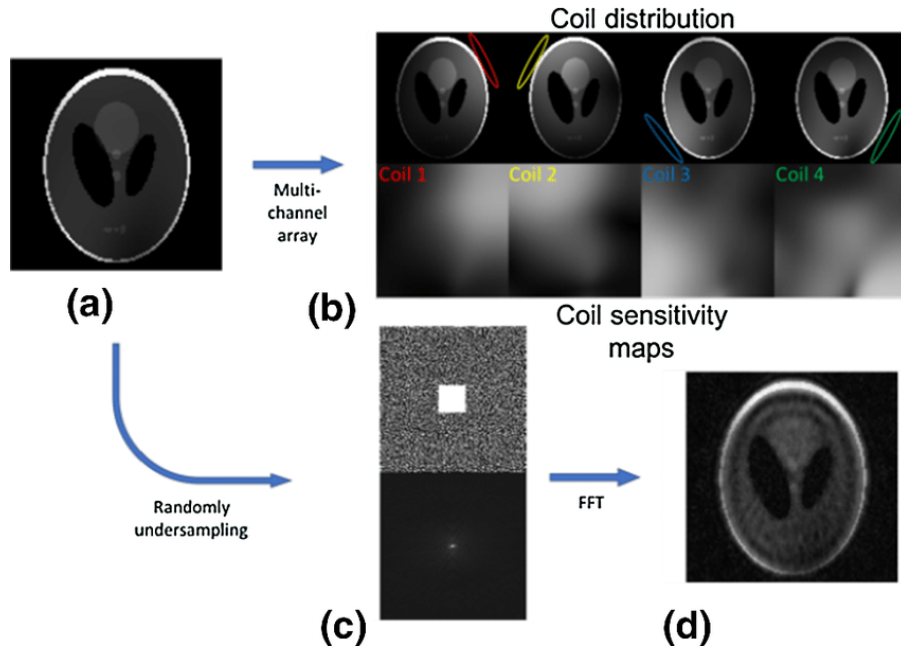


Figure 2.2: Parallel MRI acquisition and reconstruction illustration. The imaged object is shown in (a), and images acquired with different receiver coil elements are illustrated in (b). A commonly used random under-sampling pattern with fully sampled k -space center and its point spread function in spatial domain are shown in (c). The direct reconstruction with under-sampling pattern in (c) is the convolution of the point spread function of the sampling pattern and the original image, as plotted in (d); image referred from [13]

Modern MRI scanners using pMRI techniques can have up to 32 channels, allowing them to acquire images much faster than older systems. In clinical practice, they can speed up scans by 2 to 4 times, while in research settings, the acceleration can reach up to 9 -12 times. This represents a significant improvement in imaging speed compared to older MRI methods developed in the 1980s. [11]

Over the past decade, significant progress has been made in the development of pMRI methods, leading to various parallel imaging reconstruction techniques and strategies. Some well-known techniques include SMASH, SENSE and GRAPPA, while others like AUTO-SMASH, VD-AUTO-SMASH, GENERALIZED SMASH, MSENSE, PILS and SPACE RIP have also been proposed. All these methods require additional information about receiver coil sensitivity to compensate for the under-sampled k-space data. This sensitivity information can be obtained during patient setup or by acquiring a few additional k-space lines for each subsequent pMRI experiment (autocalibration), or a combination of both. The pMRI reconstruction methods can be broadly categorized into two groups: those that perform reconstruction in image space (e.g., SENSE, PILS) through an unfolding or inverse procedure, and the other category carries out the reconstruction in k-space (e.g., SMASH, GRAPPA) by calculating the missing k-space data. [12]

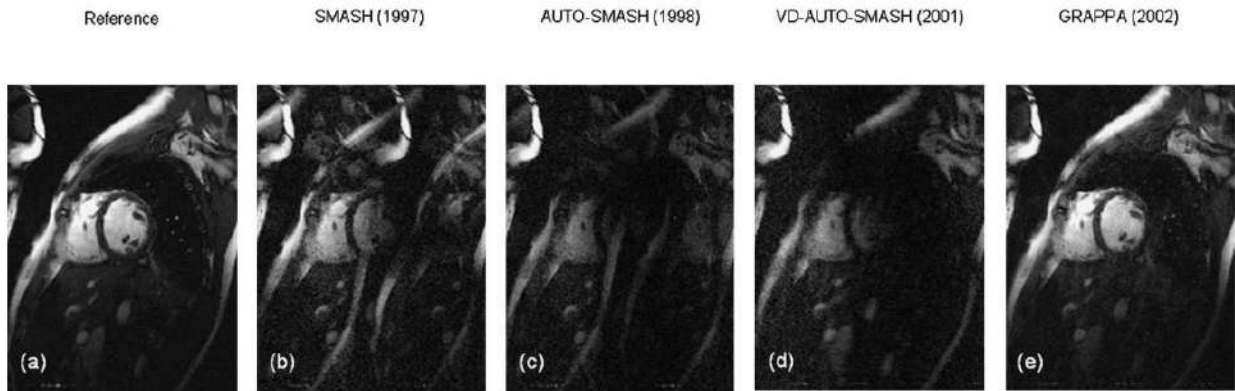


Figure 2.3: Comparison of $R = 2$ accelerated acquisitions of the human heart using (b) pure SMASH, (c) AUTO-SMASH, (d) VD-AUTO-SMASH, and (e) GRAPPA for image reconstruction. The images demonstrate the progression of k-space related pMRI reconstruction methods over time, with a continuous improvement in image quality. It is evident that GRAPPA achieves excellent image quality without phase cancellation problems or residual artifacts. Additionally, it optimizes the signal-to-noise ratio (SNR) by allowing a sum of squares reconstruction of the uncombined component coil images. For comparison, Figure (2.3a) displays a full field-of-view (FOV) acquisition. The reconstruction result shown is referred from [13]

Figure 2.3 demonstrates temporal evolution of k-space-related pMRI reconstruction algorithms, showcasing the improvements in image quality achieved with each technique. Pure SMASH (Figure 2.3b) suffers from signal losses and residual artifacts. AUTO-SMASH (Figure 2.3c)

introduces a self-calibrated reconstruction method but is still affected by limitations. VD-AUTO-SMASH (Figure 2.3d) achieves better artifact suppression with more accurate coil weight estimation. Finally, GRAPPA (Figure 2.3e) further improves the accuracy of coil weights and overcomes the drawbacks of signal cancellations and low SNR.

pMRI has shown significant benefits in various applications, including head, thoracic, and cardiac imaging, by significantly improving image acquisition capabilities. Most of the time, the primary focus is on two commercially available techniques i.e., SENSE and GRAPPA. However, having technical overview of the existing pMRI methods, discussing differences, similarities, advantages, and disadvantages can help clinicians choose the best pMRI method as per their requirement.

2.3 GeneRalized Autocalibrating Partially Parallel Acquisitions (GRAPPA):

GRAPPA (Generalized Autocalibrating Partially Parallel Acquisitions) method is a pMRI technique. It fits component coil signals to a single component coil ACS (Autocalibrating Signal) signal, which allows the derivation of linear weights to reconstruct missing k-space lines for each component coil.

This procedure is repeated for each component coil, considering that the coil sensitivities change along the read direction. GRAPPA algorithm determines the weights for reconstruction at multiple positions along the read direction.

After Fourier transformation, uncombined images for each single coil in the receiver array are obtained. The use of multiple k-space lines from all the receiver coils to fit one single coil results in increased accuracy and better artifact suppression. In GRAPPA algorithm, the uncombined single coil images can be combined using a magnitude reconstruction procedure (such as sum-of-squares), results in enhanced signal-to-noise ratio (SNR) performance, especially at low reduction factors. Furthermore, using a magnitude reconstruction procedure helps eliminate signal losses due to phase cancellations. The GRAPPA reconstruction process is represented schematically in Figure 2.4.

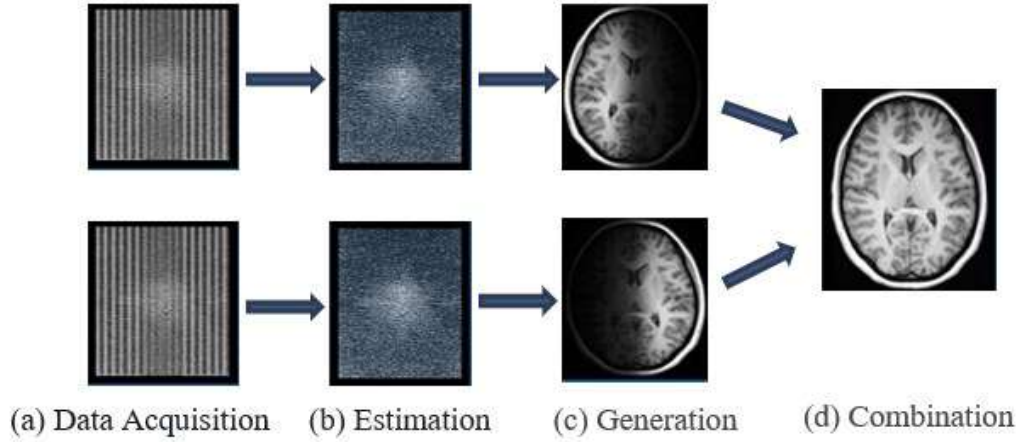


Figure 2.4: The image depicts the steps involved in parallel magnetic resonance imaging (pMRI) GRAPPA reconstruction. (a) *Signal Fitting:* the signal from each component coil is matched to the auto-calibration signal (ACS). (b) *Weight Determination:* Reconstruction is performed in the read direction, where the weights are calculated for each data point based on the information obtained from the ACS. (c) *Uncombined Coil Images Magnitude Reconstruction:* The reconstructed magnitude images of the uncombined coil data are generated by combining the information from different coils. This combination enhances the signal and creates a final image.

GRAPPA algorithm can be divided into two processing steps: (i) calibration phase (ii) synthesis phase.

During the calibration phase, calibration data collected using a coil array is represented by two matrices: the source matrix ($\mathbf{S}_{m \times n}$) and the target matrix ($\mathbf{T}_{m \times l}$). The source matrix contains information about the coil sensitivities, while the target matrix holds the fully sampled calibration image, during kernel repetitions over ACS lines, and forms a GRAPPA calibration equation as:

$$\mathbf{T}_{m \times l} = \mathbf{S}_{m \times n} \mathbf{W}_{n \times l} \text{ where } m \gg n, \quad (2.1)$$

The unknown coefficients (reconstruction coefficients or GRAPPA weight sets), represented by $\mathbf{W}_{n \times l}$, are estimated through a linear combination of Source and Target Matrices. In calibration

phase, conventional GRAPPA seeks least square fits to estimate the reconstruction coefficients (W):

$$w = \min \|Sw - t\|^2, \quad (2.2)$$

The problem in (2.1) is well overdetermined; therefore, pseudoinverse method can be used to estimate the best fit for (2.2):

$$W = (S_H S)^{-1} (S_H T), \quad (2.3)$$

where H denotes the conjugate transpose.

During the synthesis phase in GRAPPA, the estimated weight sets (W) in (2.3) are used to calculate the missing data points in the under-sampled region of the k-space for each channel in the array coil. Figure 2.5 represents a schematic description of the GRAPPA reconstruction method. The GRAPPA method, despite its advantages, has limitations. The use of massive array coils with numerous channels in imaging has resulted in longer computational times for GRAPPA reconstruction. This challenge has prompted the exploration of hardware and software approaches to reduce the number of channels effectively.

Hardware-based solutions involve using a specialized signal combiner, while software-based methods, like principal component analysis (PCA), compress the channels before reconstruction. These software approaches significantly reduce computation time by combining multiple channels coil data into a single coil.

In Literature, random projection is used to reduce computation complexity during GRAPPA calibration phase [28]. Random projection is a technique used in machine learning to reduce dimensions while preserving useful information. By applying random projection to GRAPPA, the computational time for calibration is reduced without sacrificing accuracy. The proposed random projection method achieves comparable reconstruction quality to traditional GRAPPA while significantly reducing computation time. This approach can be combined with PCA based channel reduction techniques for even greater efficiency. These limitations highlight the requirement for careful calibration and validation procedures when employing the GRAPPA method in clinical applications.

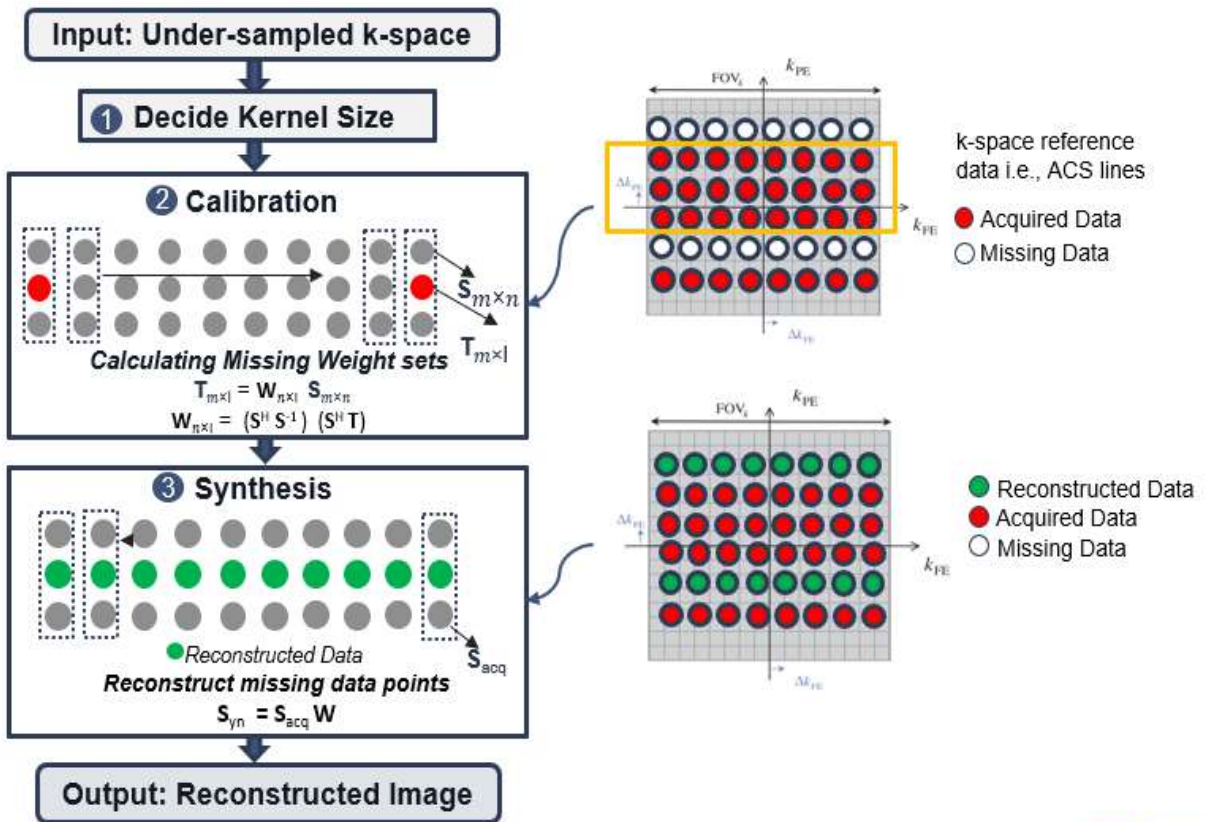


Figure 2.5: Schematic description of GRAPPA Reconstruction method using multiple lines from all the receiver coils to fit one line in a specific coil. This is repeated for each coil, resulting in uncombined coil images that can be combined using sum-of-squares.

Chapter-03

Proposed Methodologies for fast GRAPPA Reconstruction

3.1 Overview and background:

GRAPPA is a widely used pMRI technique for accelerating MRI scan time. GRAPPA uses multiple receiver coils to reconstruct artifact free images. However, as the number of receiver coils increases, time to reconstruct images from each coil increases as well. The calibration phase is the first, most important and computational extensive phase of GRAPPA. It consists of complex mathematical operations of complex value matrices. Therefore, its computational complexity increases the overall reconstruction time of GRAPPA. To address this issue, two simultaneous solutions are proposed: (i) Channel Compression. (ii) Random Projection.

3.2 Channel Compression in pMRI:

Parallel MRI enables faster image acquisition using multiple receiver coils for data acquisition. However, processing the data of a large number of receiver coils requires more memory and increases computational complexity. Channel compression techniques can be used to address this issue by compressing the receiver coil data into a smaller set of data. Therefore, reducing computational complexity and computer memory requirements for GRAPPA reconstruction. Channel Compression techniques use Eigen Value Decomposition (EVD) [15] methods such as Principal Components Analysis (PCA) [16] and Singular Value Decomposition (SVD) [21] to extract the essential information of a data set and remove redundant information. These techniques have applications in many fields of MRI including real time MRI, fast imaging etc.

3.2.1 Principal Component Analysis (PCA):

PCA is a dimensional reduction technique which maps a large dimensional data to lower dimensional data, by extracting the components which have the most variations. It consists of the following steps:

1. Standardize the range of continuous initial value variables. Mathematically, this is achieved by subtracting the mean and dividing by standard deviation for each variable value as shown in equation (3.1):

$$Z = \frac{\text{var-mean of the data}}{\text{standard deviation}} \quad (3.1)$$

- II. Covariance matrix computation is used to identify correlations in the data. Mathematically, this is done by taking sum of correlations between all the possible pairs of variables as shown in equation (3.2):

$$(x, y) = \frac{1}{n-1} \sum_{i=1}^n (x_i - E(x))(y_i - E(y)) \quad (3.2)$$

- III. Eigen value and Eigen vector computation of correlation matrix. Mathematically, this is done by subtracting the eigen values from the variable values and multiplying it with the eigen vector as shown in equation:

$$(A - \lambda I)x = 0 \quad (3.3)$$

where I is the identity matrix, λ is eigen value and x is the eigen vector.

- IV. Extract the principal components and recast the data along principal component axes. The eigen vectors and eigen values are used to compute the principal components of the data. These principal components are arranged in such a way that the most initial components contain most essential information of the data as shown in Figure 3.1. These principal components are used for data compression [22].

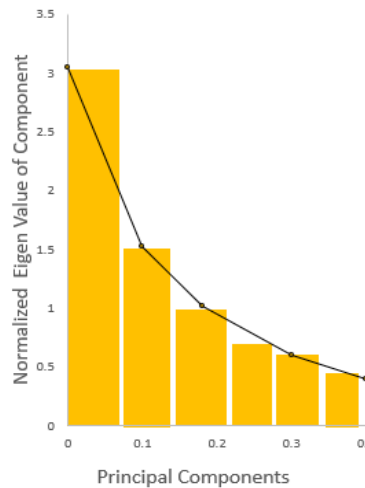


Figure 3.1: Principal Components and Eigen Value Normalization

Limitations of PCA for Channel Compression:

The computational complexity of calculating covariance of the data increases with the increase data sets. In this research, in order to transform eigen values of the matrix to principal components, Singular Value Decomposition (SVD) is used.

3.2.2 Singular Value Decomposition (SVD):

SVD is a linear algebra technique which transforms a matrix into three main matrices:

$$A = U\Sigma V^* \quad (3.4)$$

where U is the orthonormal row elements of matrix A , V^* is the conjugate of orthonormal column elements of matrix A containing the information of eigen vectors and Σ is the singular diagonal matrix containing eigen values [23]. With the help of the eigen vectors and eigen values, principal components of the data can be transformed.

Limitations of SVD:

The time complexity of computing SVD of an $m \times n$ matrix is $O(nm \min(m, n))$; and for high rank data sets, SVD becomes computationally expensive. Moreover, SVD does not have independent operations which could be exploited using hardware or software solutions to speed the process of compression.

To overcome the computational complexity of orthonormal column and rows matrices for extracting principal components, this research uses QR Decomposition [24] method for accelerating channel compression.

3.3 QR decomposition-based Channel Compression:

QR decomposition is one of the methods of matrix decomposition used for solving eigen value problems, linear systems, and least square approximations. QR decomposition decomposes a matrix into two matrices; an orthogonal Q matrix [24] which contains the information of eigen vectors, and an upper triangular R matrix [24] which contains the information of coefficients of eigen vectors.

3.3.1 QR Decomposition and Modified Gram-Schmidt Method

In literature, it is observed that for QR decomposition-based channel compression, the modified Gram-Schmidt [27] method is employed. In the modified Gram-Schmidt method, all the column vectors of matrix (A) are modified to be orthogonal to the first column vector of matrix (A) for faster computation as it orthogonalizes the column vectors simultaneously as shown in Figure-4. The proposed accelerator exploits the inherent parallelism in QR decomposition based on modified Gram – Schmidt method in comparison to the classical Gram – Schmidt method. Figure 3.2 shows the reduction of number of receiver channels for GRAPPA reconstruction.

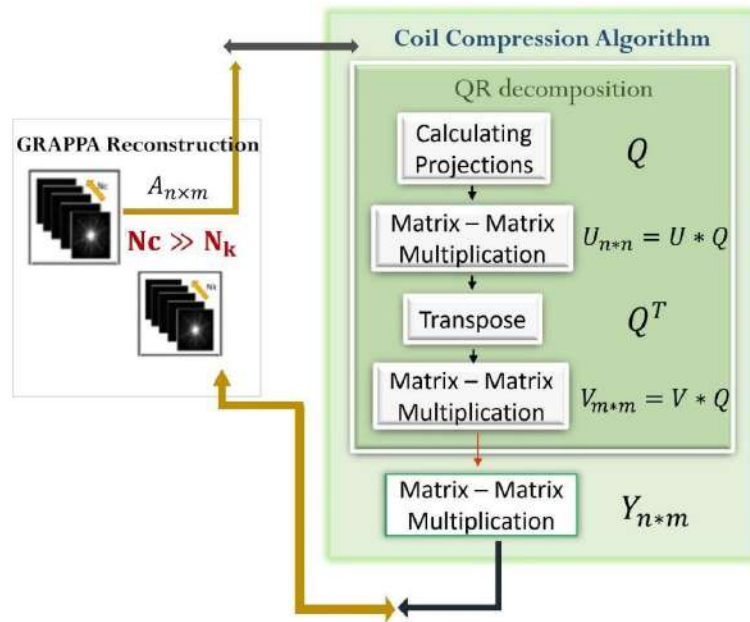


Figure 3.2: Channel Compression Algorithm Workflow

```

for j = 1 : n
    v_j = x_j
endfor
for j = 1 : n
    r_jj = ||v_j||_2
    q_j = v_j / r_jj
    for k = j + 1 : n
        r_jk = q_j^T v_k
        v_k = v_k - r_jk q_j
    endfor
endfor

```

3.4 Figure 3.3: Pseudo code of QR decomposition using modified Gram Schmidt Method [28]

3.4.1 Overview of Random Projection Method

Random projection is a powerful technique for reducing the dimensionality of datasets while preserving data relationships. By multiplying a random projection matrix with the original data matrix, a lower-dimensional representation is obtained.

Different random projection matrices, such as Achlioptas' matrix [29] and Li matrix [16], have been proposed in the literature, offering improved computational efficiency and accuracy. The choice of the projection matrix depends on the specific needs of the application, considering the trade-off between speed and convergence time based on the dataset's dimensionality.

Traditionally, random dense matrices [31] were used for random projection, but sparse projection matrices [32] have gained attention for their computational efficiency. These matrices have fewer non-zero entries, resulting in faster computations without significant loss of accuracy. However, when using a very sparse projection matrix, there is a risk of destroying non-zero entries, replacing them with zeros.

To ensure the preservation of pairwise distances, the projection matrix distribution, such as the standard normal distribution is commonly used, following the Johnson-Lindenstrauss theorem. Additionally, the sparsity of the random matrix further improves efficiency by setting lower bounds on sparsity.

3.4.2 Integration of Random Projection in GRAPPA Calibration Phase:

GeneRalized Autocalibrating Partially Parallel Acquisitions (GRAPPA) reconstruction is one of the most advanced techniques in magnetic resonance imaging (MRI) that consists of two phases i.e., calibration and synthesis. The calibration phase involves collecting training data from the source and target matrices, which is used to calculate the missing weight sets using a calibration equation. This phase is computationally expensive due to the complex-valued multiplication of an overdetermined set of linear equations.

To address the computational complexity of the calibration phase, a recent approach known as Random Projection on GRAPPA (RP-GRAPPA) has been proposed [33]. RP-GRAPPA utilizes random projection (RP) as a dimension reduction technique to reduce the dimensions of the problem. The main idea is to multiply the original matrices by a randomly generated projection matrix R , resulting in reduced-order source (S_{red}) and target (T_{red}) matrices.

The reduced set of equations during the calibration phase of GRAPPA can be approximated as:

$$\mathbf{S}_{m \times n} \mathbf{W}_{n \times l} - \mathbf{T}_{m \times l} \approx \mathbf{S}_{\text{red} m \times n} \mathbf{W}_{n \times l} - \mathbf{T}_{\text{red} m \times l}, \quad (3.5)$$

where λ represents a factor by which the order of the source and target matrices is reduced. A smaller value of λ implies a larger reduction in the number of linear equations. In RP-GRAPPA, the reconstruction coefficient (\mathbf{W}) is estimated using the pseudoinverse method by solving the reduced set of linear equations:

$$\mathbf{W}_{n \times l} = ((\mathbf{S}_{\text{red}})^H (\mathbf{S}_{\text{red}})^{-1}) ((\mathbf{S}_{\text{red}})^H (\mathbf{T}_{\text{red}})) \quad (3.6)$$

where \mathbf{S}_{red} and \mathbf{T}_{red} are the reduced-order matrices obtained through random projection.

The advantage of RP-GRAPPA is that it significantly reduces the total reconstruction time of GRAPPA. However, it should be noted that RP-GRAPPA may introduce large reconstruction errors if the value of λ is not appropriately chosen. Previous studies have recommended setting λ to be greater than 2.2 to avoid significant reconstruction errors [34]. Furthermore, it has been observed that the optimal value for λ to balance between reconstruction errors and reconstruction time is around 3. By incorporating random projection into the calibration phase of GRAPPA, RP-GRAPPA offers a promising approach to reduce computational complexity while maintaining reconstruction accuracy in MRI applications [19].

3.4.3 Proposed Sparse Randomly Projected Matrix:

The proposed method addresses the challenge of dimensionality reduction by utilizing a highly sparse randomly projected matrix, denoted as \mathbf{R} . This matrix is generated using entries $\{-1, 0, 1\}$ and probabilities $\{1/s, 1-1/s, 1/2s\}$, where s is computed as the square root of the product of the higher dimensional data (\mathbf{D}) and a scaling factor (λ). It is represented as,

$$(3.7) \quad \mathbf{R} = r_{ij} = \begin{cases} +1, & \text{with prob } 1/2s, \\ \sqrt{s} & 0, & \text{with prob } 1 - 1/s, \\ -1, & \text{with prob } 1/2s \end{cases}$$

Where $s = \sqrt{m * \lambda}$ and by adjusting the scaling factor λ , the number of non-zero elements in the matrix are significantly reduced, resulting in improved computational efficiency during the GRAPPA Calibration phase.

The impact of varying the scaling factor on the reduction of non-zero elements in the matrix. This reduction is significant as it effectively decreases the time complexity associated with complex-valued multiplication. Figure 3.4 presents the block diagram of the proposed method, demonstrating the implementation of the random projection in GRAPPA using the proposed sparser projection matrix with the scaling factor λ .

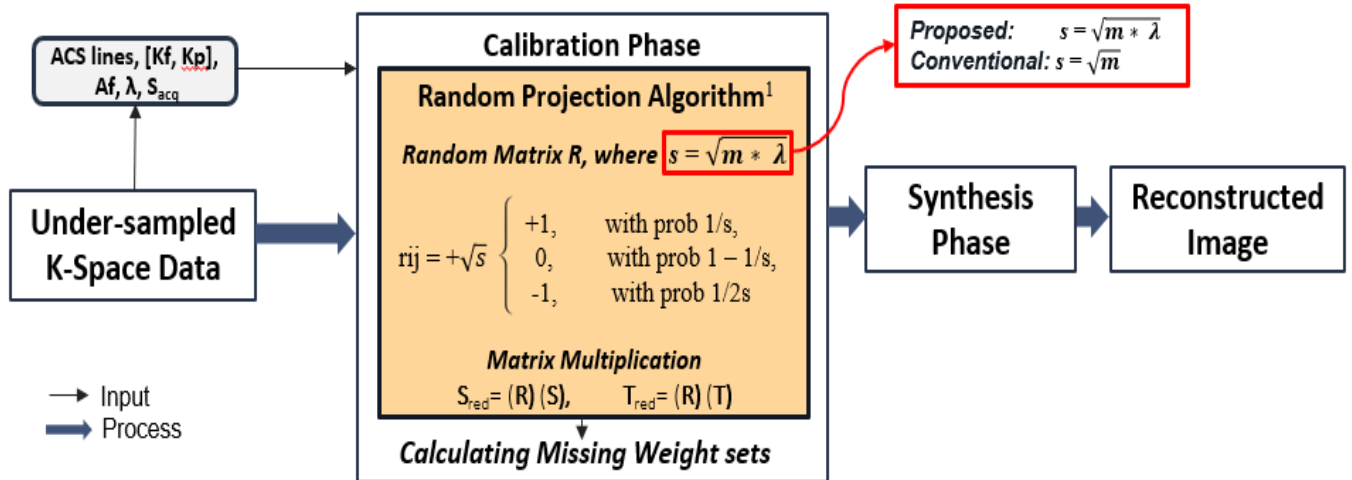


Figure 3.4: Block Diagram of the proposed Random Projection method with GRAPPA Reconstruction

3.5 Integration of Channel Compression and Random Projection with GRAPPA:

The proposed algorithm of Channel Compression and Random Projection aims to accelerate the GRAPPA reconstruction time. However, as shown in figure 3.5, by combining these algorithms with GRAPPA, acceleration can be enhanced more.

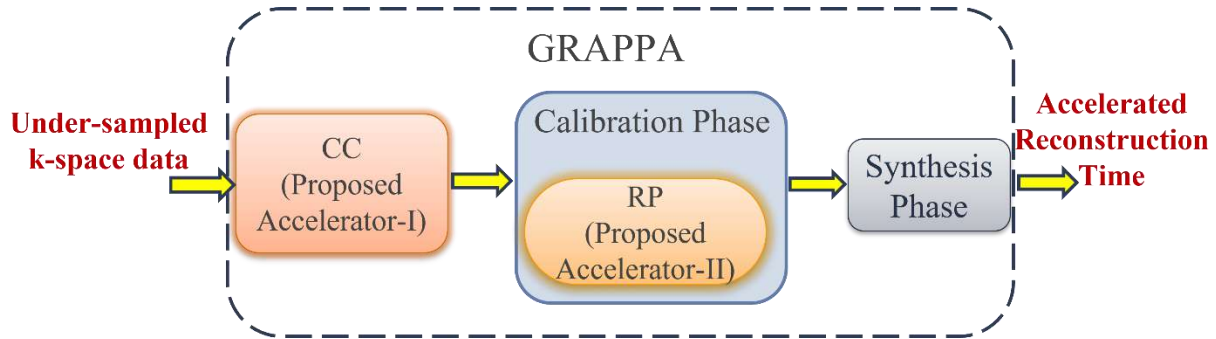


Figure 3.5: Data Flow of proposed methodologies

The integration of random projection and channel compression techniques in GRAPPA offers several advantages. Random projection reduces the number of calibration lines required, enabling faster acquisition, and reducing scan time. Channel compression, on the other hand, reduces the number of receiver channels used during reconstruction, thereby decreasing the computational complexity and memory requirements.

The integration is achieved by performing channel compression on the receiver channel data and reducing the data that needs to be stored and processed by GRAPPA. Subsequently, random projection is employed on the acquired data before applying the GRAPPA reconstruction algorithm. By projecting the data onto a lower-dimensional subspace, the number of calibration lines needed for accurate reconstruction is significantly reduced.

3.5.1 Benefits:

Enhanced efficiency:

The integration of random projection and channel compression in GRAPPA leads to improved efficiency in terms of acquisition time and computational complexity.

Reduced memory requirements:

By reducing the number of receiver channels, integration lowers the memory requirements during reconstruction, allowing for more efficient data processing.

Increased applicability:

The integration expands the feasibility of GRAPPA in clinical settings by enabling faster acquisitions and reducing the computational burden, making it more accessible for routine use.

3.5.2 FPGA based hardware accelerators:

Field Programmable Gate Arrays (FPGA) FPGAs are an excellent choice for efficient computation of real time data. Therefore, the hardware of our proposed accelerator is implemented on FPGA to deal with large data sets that will be used in the process of GRAPPA reconstruction. Recently, High Level Synthesis (HLS) tool provided by VIVADO tool chain can be used to generate RTL while reducing the design efforts. HLS uses C/C++ languages for design. In this project both the Hardware Descriptive Language (HDL) and HLS are used to generate RTL of the accelerator to be embedded on chip to accelerate GRAPPA reconstruction.

Chapter-04

SoC based GRAPPA Reconstruction

4.1 Proposed System-On-Chip Design for GRAPPA reconstruction:

In this project, we developed a SoC design for the GRAPPA reconstruction algorithm using: (i) Random Projection Method (ii) Channel Compression Method. The under-sampled data received from receiver coils will be processed in DDR4(4th Generation Double Data Rate) RAM [36] and will be compressed by PCA accelerator through system bus and will be acquired by GRAPPA which will be processed in ARM [37] processor as shown in figure 4.1.

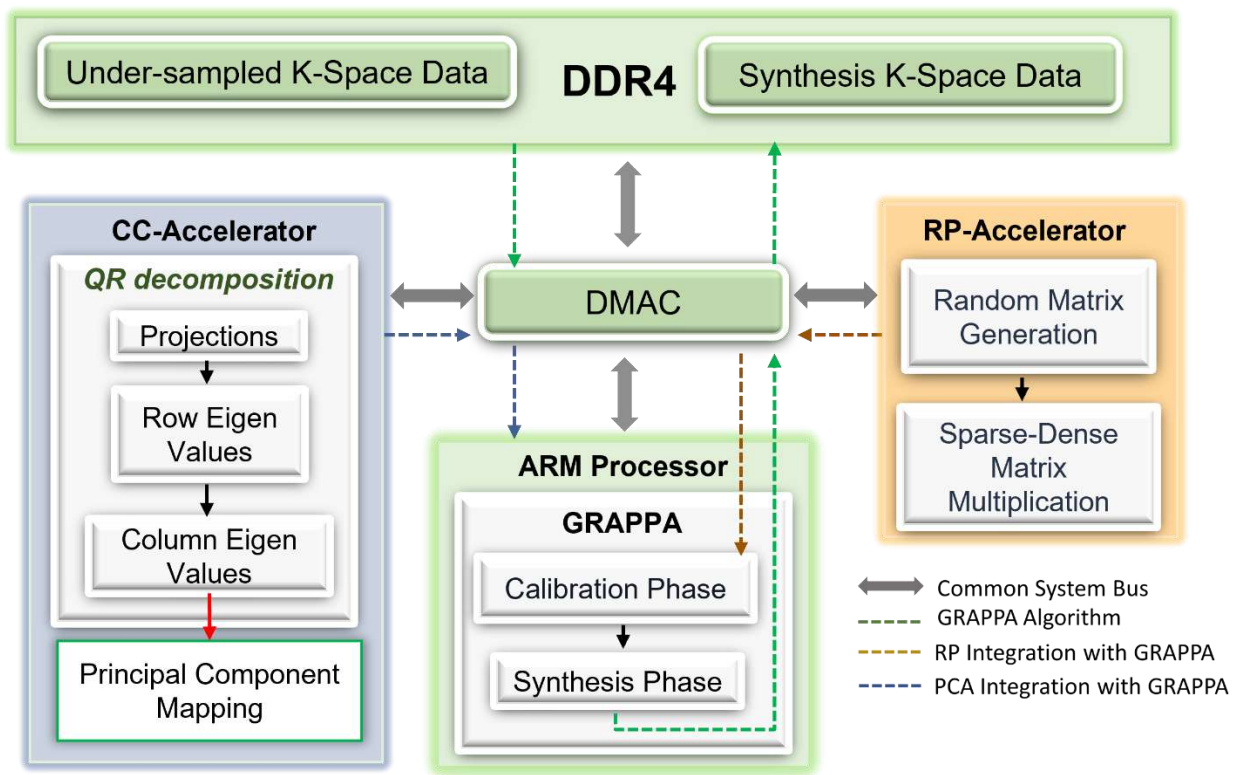


Figure 4.1: Block Diagram of the proposed SoC design with proposed accelerators using Channel Compression and Random Projection Method

Meanwhile in the calibration step of GRAPPA, dimensions of the acquired data will be reduced with Random projection accelerator. Both the accelerators will be integrated through DMAC [38] so that the accelerated data can be transferred to a multicore processor from its local memory in fast processing time, allowing computation and data transfer to process in parallel.

This system will provide synthesized k-space data which can be accessed through DDR4 with fewer computational expenses and lesser reconstruction time.

DDR4 is the fourth generation of **DDR (Double Data Rate) RAM**, which is a type of memory commonly used in computer systems. Due to good memory density and memory bank of 64GB. It has faster data transfer rates and larger capacities, DDR4 can operate at a lower voltage (1.2V) which makes it more power efficient. Faster speeds and increased memory bandwidth allow DDR4 SDRAM to keep up with the modern processors, including multi-core CPUs. Our system will be running on DDR4 for better computational time and faster speed. **Direct Access Control (DMAC)** is an IC that controls the data transfers between a device and the memory without needing a CPU. ARM [39] processors can execute instructions much faster than the Intel processors. By removing redundant instructions and pathway optimizations, ARM processor provides efficient performance while consuming less resources than CISC-based processor. As less resources are used and less power consumption, therefore ARM CPUs generate very less heat.

4.2 Proposed Architecture for Channel Compression:

The Channel Compression Algorithm, implemented using the Vivado HLS Design tool and tested on the Zynq UltraScale CU102 board, is divided into two main modules as demonstrated in Figure 4.2, which is controlled by a control unit as demonstrated in Figure 4.3:

4.2.1 QR Decomposition Module:

- **Input:**

The input to the QR decomposition module is a data matrix, denoted as A , of size $m \times n$, where m represents the number of rows and n represents the number of columns. Each element of the matrix A is denoted as a_{ij} , where i represents the row index and j represents the column index.

- **Output:**

The output of the QR decomposition module includes two main components:

Row Eigen Information: This provides information about the eigenvalues and eigenvectors of the matrix $A^T A$ (A transpose multiplied by A). The eigenvalues represent the amount of variance captured by each eigenvector, and the eigenvectors represent the principal directions in the data.

Column Eigen Information: This provides information about the eigenvalues and eigenvectors of the matrix AA^T (A multiplied by A transpose). The eigenvalues and eigenvectors in this case represent the singular values and right singular vectors, respectively.

- **Procedure:**

The QR decomposition module performs the following steps to obtain row and column eigen information:

QR Decomposition Computing: The module uses modified Gram-Schmidt algorithm to calculate the QR decomposition of the input matrix A . This yields two matrices: an orthogonal matrix Q and an upper triangular matrix R .

- Computing Row Eigen Information:

a. $A^T A$: Multiplication of the transpose of matrix A (A^T) with A itself to obtain the matrix $A^T A$.

b. Eigenvalues and Eigenvectors: Eigenvalues and eigenvectors calculation of $A^T A$. These eigenvalues and eigenvectors represent the row eigen information.

- Computing Column Eigen Information:

a. AA^T : Multiplication matrix A with its transpose (A^T) to obtain the matrix AA^T .

b. Eigenvalues and Eigenvectors: Eigenvalues and eigenvectors calculation of AA^T . These eigenvalues and eigenvectors represent the column eigen information.

- Output:

a. Row Eigen Information: The module provides the eigenvalues and eigenvectors obtained from the computation of $A^T A$ as the row eigen information. These eigenvalues indicate the amount of variance captured by each corresponding eigenvector.

b. Column Eigen Information: The module outputs the eigenvalues and eigenvectors obtained from the computation of AA^T as the column eigen information. These eigenvalues are the singular values, and the eigenvectors are the right singular vectors.

4.2.2 Principal Component Mapping Module:

- **Input:**

The input to the PCM module is two identity matrices represented by a matrix X of size $n \times n$, and Z $m \times m$, where m and n denotes the number of rows and columns elements of matrix A of size $m \times n$.

- **Output:**

The output of the PCM module includes two main components:

Compressed Data: This is the transformed dataset, where the original high-dimensional data is projected onto a lower-dimensional subspace defined by the selected principal components.

Procedure:

The PCM module performs the following steps to produce compressed data:

- **Principal Components:**
 - a. Determination of the number of principal components, k , based on a desired level of explained variance or a specific dimensionality reduction goal.
 - b. Selection of the top k eigenvectors corresponding to the highest eigenvalues as the principal components. These eigenvectors form the projection matrix P .
- **Compressed Data:**
 - a. Projection of the preprocessed data A onto the selected principal components by calculating the dot product of A and P . The resulting compressed data matrix Y is given by $Y = A * P$.
- **Output:**

Compressed Data: The module outputs the transformed dataset Y , which represents the original high-dimensional data mapped onto a lower-dimensional subspace defined by the selected principal components. The number of columns in Y corresponds to the number of selected principal components.

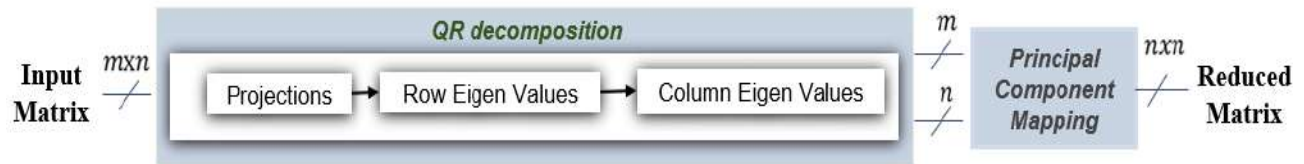


Figure 4.2: Channel Compression Accelerator

4.2.3 Finite State Machine Module:

The FSM module controls the execution of the QR module and PCA module, coordinating their operations based on three states: Idle, Running, and Done.

Idle State:

In the Idle state, the FSM module is waiting for a trigger signal to start the process. Upon receiving the trigger signal, the FSM module transitions to the Running state.

Running State:

In the Running state, the FSM module coordinates the execution of the QR module and PCA module. It starts by activating the QR module, which performs the QR decomposition on the input data. Once the QR decomposition is completed, the FSM module triggers the PCA module to process the decomposed data. The PCA module mapped the principal components to obtain the compressed data. While the PCA module is running, the FSM module remains in the Running state, waiting for the PCA module to finish its computations.

Done State:

In the Done state, the FSM module indicates the completion of the entire process. It may output the compressed data obtained from the PCA module and any other relevant results. The FSM module then returns to the Idle state, ready to receive the next trigger signal for a new process.

- **Transitions and Actions:**

Idle -> Running: Transition triggered by a start signal.

When the FSM module receives a start signal, it transitions from the Idle state to the Running state. The action associated with this transition activates the QR module and triggers its execution.

Running -> Done: Transition triggered when the PCA module completes its computations.

Once the PCA module finishes its computations and provides the compressed data, the FSM module transitions from the Running state to the Done state. The action associated with this transition involves outputting the compressed data and any other relevant results.

Done -> Idle: Transition triggered by a reset signal or completion of the process.

When the FSM module receives a reset signal or completes the process, it transitions from the Done state back to the Idle state. The action associated with this transition involves resetting internal variables, preparing for a new process, or waiting for the next trigger signal.

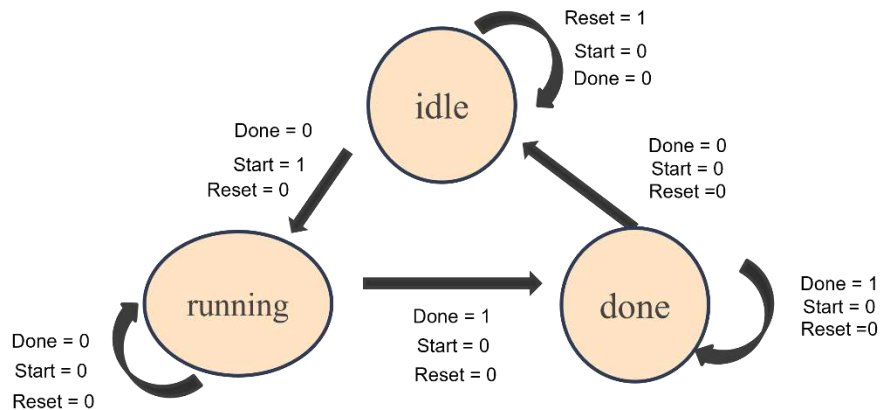


Figure 4.3: FSM Module for Proposed CC architecture

The FSM module described above controls the execution of the QR module and PCA module, coordinating their operations in a three-state system: Idle, Running, and Done. It starts with the Idle state, waiting for a trigger signal to initiate the process. Upon receiving the trigger signal, it transitions to the Running state, activating the QR module and triggering the PCA module. While the PCA module is running, the FSM module remains in the Running state. Once the PCA module completes its computations, the FSM module transitions to the Done state, where it may output results. Upon receiving a reset signal or completing the process, it transitions back to the Idle state, ready for the next trigger signal. To enhance computation efficiency, the QR and PCM

module utilizes HLS pragmas like loop unrolling and parallelism. These pragmas leverage parallel processing capabilities, distributing the workload across multiple units and optimizing resource utilization. This accelerates computation, resulting in faster generation of the final reduced matrix.

4.3 Proposed Architecture for Random Projection:

4.3.1 Accelerator Designing:

The Random Projection Algorithm, implemented using the Vivado HLS Design tool and tested on the Zynq UltraScale CU102 board, is divided into four main modules as demonstrated in Figure 4.4:

Module 1: Random Number Generator

The Random Number Generator module serves as an initial step in the Random Projection Algorithm, providing the necessary random numbers for subsequent operations. These random numbers will be further utilized in the following modules to determine the algorithm's behavior and enable the generation of the final output matrices. To generate random numbers, the module employs a pseudo-random number generator (PRNG) algorithm. This algorithm ensures that the generated random numbers exhibit a uniform distribution within the range of 0 to 1. By leveraging the PRNG algorithm, the module generates a variable called R_temp , which holds the random numbers.

The elements in R_temp represent independent random variables drawn from the uniform distribution. These random numbers are stored as single-precision values, utilizing a fixed number of bits to accommodate both the fractional and integer parts of the number. This precision allows for accurate representation and subsequent processing of the random numbers within the algorithm.

Module 2: Random Matrix

The Random Matrix module is responsible for creating a matrix based on the R_temp variable generated by the Random Number Generator module. This matrix has dimensions of $[m \times (n \cdot \lambda)]$. It is important to note that the rows and column indices are separated and treated differently.

During the execution of this module, the R_temp variable from Module 1 is utilized to populate the elements of the matrix. Each element of the matrix corresponds to a specific row and column index. The value of m determines the row index, while the product of n and λ determines the column index.

By separating the row and column indices, the Random Matrix module ensures that each element in the matrix corresponds to a unique position in the overall structure. This separation allows for more efficient storage and manipulation of the matrix, facilitating subsequent operations in the Random Projection Algorithm.

The Random Matrix module serves as an intermediary step in the algorithm, providing the necessary matrix representation based on the generated random numbers. The resulting matrix, denoted as R_t , acts as a fundamental component in subsequent computations, contributing to the overall behavior and performance of the Random Projection Algorithm.

Module 3: Comparator (Sparse Matrix Generation)

The Comparator module plays a crucial role in generating a highly sparse random projection matrix based on the values in Random matrix R_t . This module compares each element of R_t to the following conditions:

1. If the value of an element in R_t is greater than or equal to $1/s$, it is set to 0.
2. If the value of an element in R_t is less than $1/2s$, it is set to $-\sqrt{s}$.
3. If the value of an element in R_t falls between $1/s$ and $1/2s$, it is set to \sqrt{s} .

The value of s is calculated as the square root of the product of m and λ . It represents a scaling factor for the comparisons in the Comparator module.

Furthermore, during this comparison process, each element's indices that meet the specified conditions are stored alongside the corresponding values in the resulting sparse matrix R . This ensures that the sparsity pattern is preserved, allowing for efficient representation and computation in subsequent stages of the Random Projection Algorithm.

The Comparator module utilizes HLS pragmas, including pipelining, loop unrolling, and parallelism, for faster computation. These HLS pragmas optimize the execution of conditional comparisons and sparse matrix generation, leveraging the capabilities of the

Zynq UltraScale CU102 hardware. The use of these HLS programs ensures efficient and accelerated processing within the Comparator module, contributing to the overall speed and effectiveness of the Random Projection Algorithm.

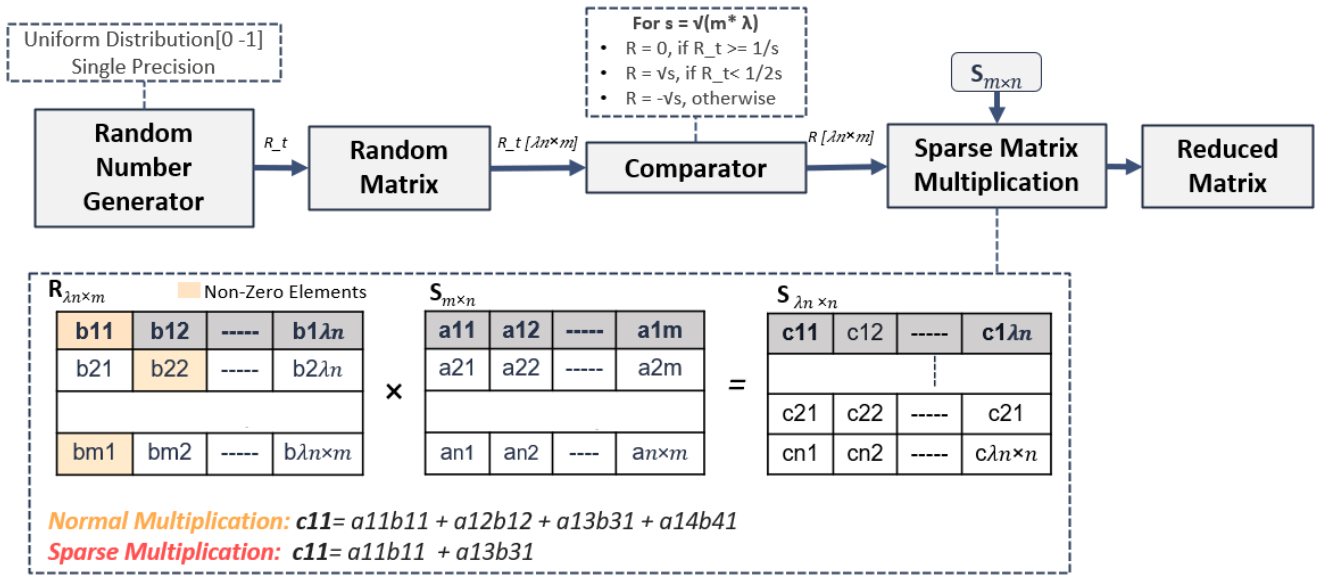


Figure 4.4: Block Diagram of Random Projection Algorithm on Vivado HLS Tool

Module 4: Sparse and Dense Matrix Multiplication:

The Sparse and Dense Matrix Multiplication module is a key component of the Random Projection Algorithm. Its purpose is to perform an efficient multiplication operation between the sparse matrix R and the dense matrix Src. To achieve this, the module utilizes smart multiplication techniques and leverages HLS programs, such as loop unrolling and parallelism.

In a traditional matrix multiplication, each element of the resulting matrix is calculated by multiplying corresponding elements from the dense and sparse matrices and summing them. However, the smart multiplication approach used in this module takes advantage of the sparsity pattern in the sparse matrix. Only the non-zero elements are considered for multiplication, while the indices of both the dense and sparse entries that would result in zero multiplication are ignored. This intelligent approach significantly reduces the number of multiplications and computations required, leading to faster processing.

To enhance computation efficiency, the Sparse and Dense Matrix Multiplication module utilizes HLS pragmas like loop unrolling and parallelism. These pragmas leverage parallel processing capabilities, distributing the workload across multiple units and optimizing resource utilization. This accelerates computation, resulting in faster generation of the final reduced matrix.

Chapter-05

Results and Discussion

5.1 Datasets:

The data used in this research is 30 receiver coil cardiac images provided by Case Western Reserve University, USA and Ohio State University, USA. The detailed features of the dataset used are shown in table 4.1.

Features	Dataset	Parameters	Configurations
Scanner	3.0T Siemen Scanner	Acceleration Factor	2
Image Size	512x252	No. of ACS line	32
Number of frames	11	Kernel Size	3x2
		Scaling Factor (λ)	1.1 - 3

Table 4.1: Feature of data set used in this research.

5.2 Results:

The results of the proposed methodology are presented here.

5.2.1 Channel Compression Results:

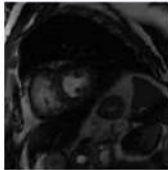
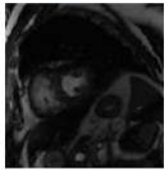
Parameters	CPU based CC-GRAPPA Reconstruction	
Number of Receiver Coils(N_c)	10 Virtual Coil data set	30 Coil original data set
Acceleration Factor = 2		
Reconstruction Time (sec)	3	6

Table 5.2 (a): This table demonstrates the Time Comparison of Channel Compression (CC-GRAPPA) against Conventional GRAPPA Reconstruction

Parameters	Conventional GRAPPA	CC- GRAPPA
Number of Receiver Coils(N_c)	30 Coil original data set	10 Virtual Coil data set (Reduced by PCA)
Signal to Noise Ratio (SNR) (dB)	22.93	21.74
Root Mean Square Error (RMSE)	0.100584	0.100614
Artifact Power	0.077097	0.078097

Table 5.2(b): Comparison of channel compression results for 30 coils cardiac data with acceleration factor (AF) = 2, kernel size 11×4 and 64 ACS.

The results in table 5.1 show that the proposed methodology speeds up the GRAPPA reconstruction times with acceptable SNR. The results show that channel compression using the proposed method offers speed up in the reconstruction time for GRAPPA without degrading the image quality. A complete examination of the performance of both the approaches was performed using MATLAB. When compared to the Conventional GRAPPA, the results clearly show improvements in overall reconstruction time, signal-to-noise ratio (SNR), and root mean square error (RMSE).

Parameters	FPGA based CC Accelerator	Single-core CPU		Multi-core CPU	
		10 Virtual Coil data set	30 Coil original data set	10 Virtual Coil data set	30 Coil original data set
Number of Receiver Coils(Nc)	10 Virtual Coil data set	10 Virtual Coil data set	30 Coil original data set	10 Virtual Coil data set	30 Coil original data set
Reconstruction Time(sec)	0.2	2.5	7	1.19	5.1

Table 5.3: Time Comparison of FPGA based Channel Compression accelerator against single-core and multi-core CPU.

By comparing the proposed method with other methods in literature, the proposed method highlights its effectiveness which demonstrates a significant contribution to the field of channel compression and improved computational efficiency for CC-GRAPPA as shown in table 5.3.

The Proposed Channel Compression Algorithm was tested using automated testing and verification features of the Vivado HLS tool. Furthermore, for performance evaluation the reconstruction time using the proposed method is compared with single-threaded and multi-threaded CPU-based counterparts. CPU-based counterparts are implemented on a Linux environment using NRTBEANS 13.0 integrated development environment (IDE). The synthesis results were reported for the target device i.e., Zynq UltraScale ZCU102, which offers a combination of programmable logic and processing capabilities. The synthesis (performance) results for resource utilization showed varying usage, with moderate utilization for the 10 Channel Coil Data demonstrated in Table 5.4. These results highlight the algorithm's adaptability and efficient resource utilization.

Parameters	FF	DSP	BRAM
------------	----	-----	------

Total Available Resources	554800	2020	1510
10-Channel Coil Data	105,412 ~19%	585.8~29%	317.1 ~21%

Table 5.4: Resource utilization of the Channel Compression Accelerator on ZynqCU102 FPGA Development Board

5.2.2 Random Projection Results:

The implementation of Random Projection on GRAPPA using the proposed sparse Random Projection Matrix shows noticeable reduction in the number of non-zero elements in a Random matrix, as illustrated in Figure 5.1. This reduction in non-zero elements directly translate to fewer multiplications during the calculation of missing weight sets using the pseudo inverse method in the GRAPPA Calibration Phase. Figure 5.1 provides a detailed comparison of the reduction in non-zero elements for different values of Lambda (λ) in both the Conventional Random Projection and the Proposed Random Projection. The introduction of a scaling factor λ results in subsequential reduction in non-zero elements which greatly enhance computational efficiency.

When compared to the Conventional Random Projection Method, the results clearly show improvements in overall reconstruction time, signal-to-noise ratio (SNR), and root mean square error (RMSE) as shown in Table 5.5 and Figure 5.2

By comparing the proposed method with other methods in literature, the proposed method highlights its effectiveness which demonstrates a significant contribution to the field of dimensionality reduction and improved computational efficiency for RP-GRAPPA.

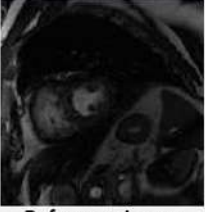
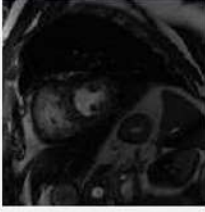
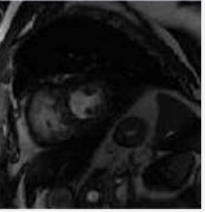

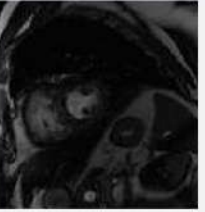
Parameters	Conventional RP-GRAPPA		Proposed RP-GRAPPA		
Scaling Factor (λ)	1.5	2.5	1.5	2.5	
SNR (dB)	23.698	23.182	22.446	23.204	
RMSE	0.08	0.1	0.08	0.08	
					
	<i>Reference Image</i>				

Table 5.5: This demonstrates a comparative analysis of the signal-to-noise ratio (SNR) and root mean square error (RMSE) for a 12-coil cardiac dataset between proposed RP-GRAPPA vs the conventional RP-GRAPPA under various configuration settings.

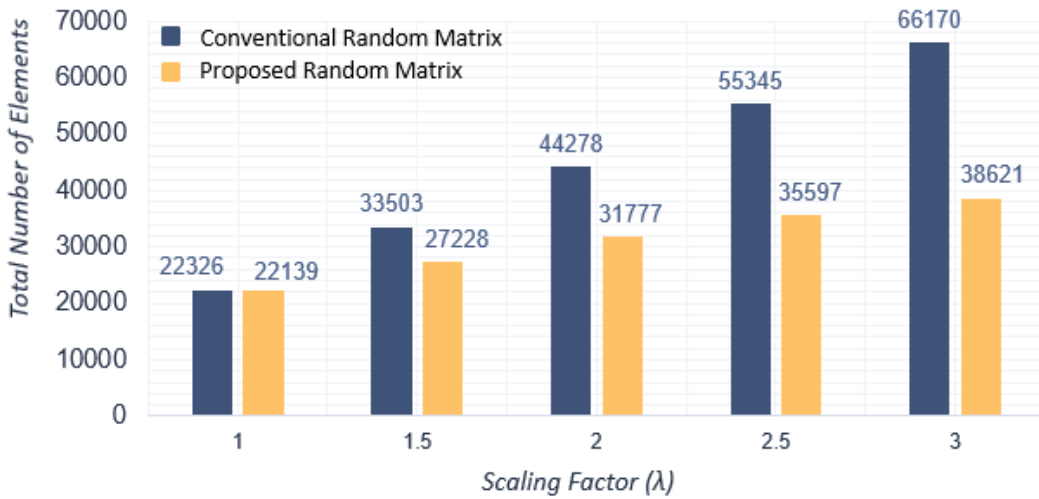


Figure 5.1: This shows the comparative analysis of no of non-zero elements in the proposed highly sparse random projected matrix with conventional Random projected matrix.

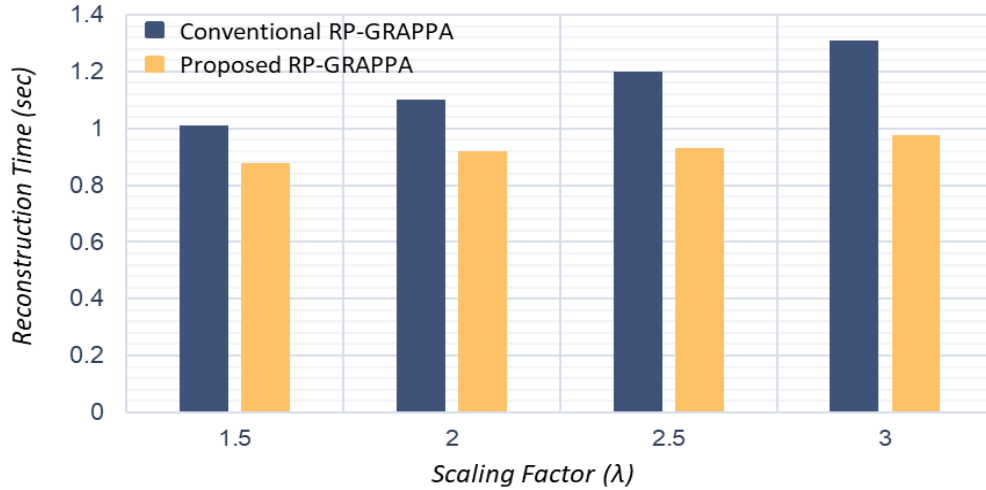


Figure 5.2: This graph shows the comparative analysis of reconstruction time of proposed highly sparse random projection method vs the conventional Random projection method.

The Proposed Random Projection Algorithm was tested using automated testing and verification features of the Vivado HLS tool. Furthermore, for performance evaluation the reconstruction time using the proposed method is compared with single-threaded and multi-threaded CPU-based counterparts. CPU-based counterparts are implemented on a Linux environment using NRTBEANS 13.0 integrated development environment (IDE). The synthesis results were reported for the target device i.e., Zynq UltraScale ZCU102, which offers a combination of programmable logic and processing capabilities. The synthesis (performance) results for resource utilization showed varying usage, with moderate utilization for the 10 Channel Coil Data and higher utilization for the 30 Channel Coil Data exhibit in Table 5.6 (a). The time comparison of FPGA based Random Projection accelerator against single-core and multi-core CPU demonstrated in Table 5.6 (b). These results highlight the algorithm's adaptability and efficient resource utilization.

Parameters	FF	DSP	BRAM
Total Available Resources	554800	2020	1510
10 Channel Coil Data	127,744 ~23%	505~32%	483 ~25%
30 Channel Coil Data	305,140 ~73%	1,131~55%	1101~56%

Table 5.6 (a): Resource utilization of the Random Projection Accelerator on ZynqCU102 FPGA Development Board

Parameters	FPGA based RP-Accelerator	Single-core CPU		Multi-core CPU	
Number of Receiver Coils(Nc)	10 Virtual Coil data set	10 Virtual Coil data set	30 Original Coil data set	10 Virtual Coil data set	30 Coil original data set
Reconstruction Time(sec)	1.8	6	7	3.7	5.1

Table 5.6 (b): Time Comparison of FPGA based Random Projection accelerator against single-core and multi-core CPU.

5.2.3 Integration of Channel Compression and Random Projection with GRAPPA:

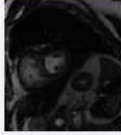
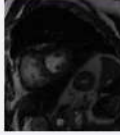


Parameters	Conventional GRAPPA Reconstruction	CC-GRAPPA Reconstruction	RP-GRAPPA Reconstruction	CC-RP GRAPPA Reconstruction
Acceleration Factor = 2				
Number of Receiver Coils (Nc)	30	10	30	10
Signal to Noise Ratio (SNR) (dB)	22.93	21.74	22.76	22.92
Reconstruction Time(sec)	6	3	3	2

Table 5.7: This comparison illustrates the performance of two proposed methods, CC-GRAPPA (Proposed Method I) and RP-GRAPPA (Proposed Method II), when integrated with GRAPPA. The results are compared with Conventional GRAPPA Reconstruction, focusing on key performance parameters such as Signal-to-Noise Ratio (SNR) and Total Reconstruction Time.

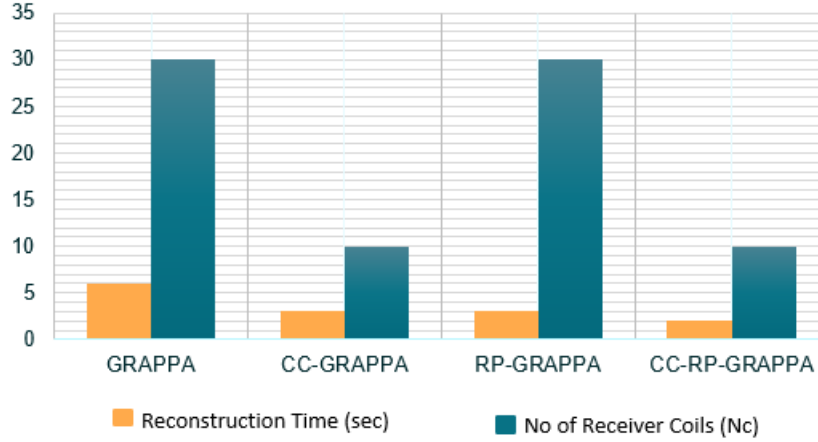


Figure 5.3: Comparative analysis of reconstruction time of proposed accelerators with GRAPPA.

The results demonstrated in table 5.7 and figure 5.3, that the integration of proposed methodologies speeds up the GRAPPA reconstruction times up to 3x with acceptable SNR. The results show that integration of Channel Compression (proposed method-I) and Random Projection (proposed method-II) with GRAPPA speeds up the reconstruction time without compromising the image quality.

5.3 Discussion:

The main contribution of this research is to accelerate the overall GRAPPA reconstruction in MRI. In this work, two simultaneous solutions (Channel Compression and Random Projection) are proposed to accelerate the GRAPPA reconstruction. The evaluation is conducted using the following settings:

- **GRAPPA Parameters:** Kernel Size ($k_f \times k_p$) = [3 x 2], Calibration Lines = 32, and Acceleration Factor (af) = 2.
- **Performance Measures:** Total number of non-zero elements, total number of zero elements, sparsity percentage of the Sparse Matrix, calibration time, synthesis time, and total time.
- **Evaluation Parameters:** Signal-to-Noise Ratio (SNR), Root Mean Square Error (RMSE), and Artifact Power (AP),

Following are the observations when comparing the results obtained from the Conventional and the Proposed Method:

1. **Reduction of receiver coil data:** The proposed channel compression method reduces the number of receiver coils data. This data is later stored and processed by GRAPPA reconstruction algorithm. This reduction significantly improves computational efficiency on FPGA by reducing the number of required multiplications during the GRAPPA Calibration Phase.
2. **Reduction of Non-zero Elements:** The Proposed Sparse Randomly Projected Matrix achieves a reduction of non-zero elements. This reduction significantly improves computational efficiency on FPGA by reducing the number of required multiplications during the GRAPPA Calibration Phase.
3. **Sparsity and Computational Performance:** The Proposed randomly generated Matrix demonstrates a higher sparsity percentage, as compared to the traditional RP-GRAPPA. This higher sparsity leads to improved computational performance as fewer calculations are needed.
4. **Total Time:** The proposed methods demonstrate faster processing and improved overall efficiency compared to the traditional GRAPPA method. This improvement signifies faster processing and overall efficiency gains.
5. **Performance Metrics:** The evaluation parameters including SNR, RMSE and AP, indicate the superiority of the proposed Sparse Randomly Projected Matrix. On average, it achieves 1.09% higher SNR, 0.29% lower RMSE, and 33.93% reduced artifact power compared to the conventional GRAPPA.

These findings demonstrate significant advantages of the proposed two accelerators (Random Projection and Channel Compression) in terms of computational efficiency and performance measures. The methods show substantial speedup, improved sparsity, and enhanced signal quality, making it a highly effective approach for GRAPPA reconstruction.

Chapter-06

Conclusion and Future work

6.1 Conclusion

In conclusion, this project focused on addressing the computational complexity and image reconstruction time of GRAPPA technique in Parallel Magnetic Resonance Imaging (pMRI). The aim of this project is to design a System-on-Chip (SoC) system using the VIVADO Design tool that enables fast MRI reconstruction for real-time clinical applications.

This project focused on two key aspects i.e., channel compression and dimension reduction. The channel compression accelerator utilized Principal Component Analysis (PCA) to reduce the number of receiver coils and compress the data acquired for image reconstruction. This reduced the imaging time and storage required, while still preserving the essential components of the data. On the other hand, the dimension reduction accelerator employing random projection is used to reduce the higher dimensional calibration phase of GRAPPA reconstruction to a lower dimensional space feature.

The proposed accelerators were designed using FPGA and implemented using the VIVADO Design tool. The experiments conducted on a 30-channel cardiac dataset demonstrated the effectiveness of the accelerators. The results showed that the accelerators significantly reduced the reconstruction times for GRAPPA method, achieving 15x speed-ups compared to single-core and multi-core CPUs while maintaining acceptable Signal-to-Noise Ratio (SNR) values. Furthermore, the proposed accelerators utilized fewer resources and reduced memory requirements, making them highly efficient for real-time MRI reconstruction.

The development of FPGA-based accelerators for channel compression and dimension reduction in GRAPPA reconstruction brings significant contributions to improve the efficiency and speed of real-time MRI. These accelerators offer several advantages e.g., they enhance patient comfort, reduce motion artifacts, improve precision in image-guided interventions, enable faster scan times, and provide better measurement of physiological parameters.

The real-time implementation of this project is a significant step forward in addressing the computational challenges of GRAPPA reconstruction in real-time MRI. It has the potential to revolutionize MRI systems by making the image reconstruction faster, more efficient, and accessible.

6.2 Future Work:

6.2.1 Enhanced Dimension Reduction Techniques:

Dimension reduction techniques have the potential to be further explored and improved in the years to come. While PCA and random projection have demonstrated promising results in lowering GRAPPA computational complexity, it would be interesting to examine advanced methods such as sparse representation [40], low-rank approximation [41] or sparse principal component analysis [42]. These techniques could provide additional opportunities to optimize the reconstruction time and improve the quality of the reconstructed images in parallel magnetic resonance imaging.

6.2.2 Integration with Deep Learning Approaches:

Deep learning integration with GRAPPA reconstruction shows promising potential for future research. Deep neural networks have demonstrated exceptional performance in a variety of image reconstruction tasks [43] and their application in parallel magnetic resonance imaging holds considerable potential. It may be able to learn complicated image reconstruction patterns and produce even faster and more precise reconstruction outcomes by training neural networks on large-scale datasets, which could lead to significant advancements to the MRI technique.

6.2.3 Real-Time Implementation Considerations:

The effectiveness and speed of GRAPPA reconstruction process are crucial in real-world applications [44]. The proposed SoC design should therefore be modified for real-time implementation. This involves performance improvements, algorithmic enhancements, and hardware optimizations, while considering protocols, scalability, and flexibility across MRI systems. The objective is to provide a valuable tool for seamless integration into clinical MRI processes, enabling faster and more precise image reconstruction.

Bibliography

- [1] Real-time MRI: Recent advances using radial FLASH. (n.d.). Open Access Journals in Scientific, Medicine and Technology. <https://www.openaccessjournals.com/articles/realtime-mri-recent-advances-using-radial-flash-9385.html>
- [2] Parallel Imaging (PI). (n.d.). Questions and Answers in MRI. Retrieved January 9, 2023, from <http://mriquestions.com/how-is-pi-different.html>.
- [3] Blaimer, M., Breuer, F., Mueller, M., Heidemann, R. M., Griswold, M. A., & Jakob, P. M. (2004). SMASH, SENSE, PILS, GRAPPA: how to choose the optimal method. Topics in magnetic resonance imaging: TMRI, 15(4), 223–236. <https://doi.org/10.1097/01.rmr.0000136558.09801.dd>
- [4] O. Inam, M. Qureshi, Z. Laraib, H. Akram, and H. Omer, “GPU accelerated Cartesian GRAPPA reconstruction using CUDA,” Journal of Magnetic Resonance, vol. 337, p. 107175, Apr. 2022, doi: <https://doi.org/10.1016/j.jmr.2022.107175>.
- [5] Radiological Society of North America (RSNA) and American College of Radiology (ACR). (n.d.). Functional MRI (fMRI). Radiologyinfo.org. <https://www.radiologyinfo.org/en/info/fmribrain>
- [6] GRAPPA/ARC? (n.d.). Questions and Answers in MRI. Retrieved January 9, 2023, from <http://mriquestions.com/grappaarc.html>.
- [7] FPGA-based hardware accelerator for SENSE (a parallel MR image reconstruction method). (2020, January 3). - ScienceDirect. <https://doi.org/10.1016/j.combiomed.2019.103598>
- [8] D. C. Preston, “MRI Basics,” Case.edu, 2020. <https://case.edu/med/neurology/NR/MRI%20Basics>.

- [9] W. T. Sobol, "Recent advances in MRI technology: Implications for image quality and patient safety," *Saudi Journal of Ophthalmology*, vol. 26, no. 4, pp. 393–399, Oct. 2012, doi: <https://doi.org/10.1016/j.sjopt.2012.07.005>.
- [10] A. Deshmane, V. Gulani, M. A. Griswold, and N. Seiberlich, "Parallel MR Imaging," *Journal of magnetic resonance imaging: JMRI*, vol. 36, no. 1, pp. 55–72, Jul. 2012, doi: <https://doi.org/10.1002/jmri.23639>.
- [11] R. M. Heidemann et al., "A brief review of parallel magnetic resonance imaging," *European Radiology*, vol. 13, no. 10, pp. 2323–2337, Oct. 2003, doi: <https://doi.org/10.1007/s00330-003-1992-7>.
- [12] M. A. Griswold et al., "Generalized autocalibrating partially parallel acquisitions (GRAPPA)," *Magnetic Resonance in Medicine*, vol. 47, no. 6, pp. 1202–1210, Jun. 2002, doi: <https://doi.org/10.1002/mrm.10171>.
- [13] Wang, Fang & Fang, Lei & Peng, Xuehua & Wu, Min & Wang, Wenzhi & Zhang, Wenhan & Zhu, Baiqu & Wan, Miao & Hu, Xin & Shao, Jianbo. (2021). Bias field correction for improved compressed sensing reconstruction in parallel magnetic resonance imaging. *Signal, Image, and Video Processing*. 15. 10.1007/s11760-020-01721-4.
- [14] Arshad, M., Qureshi, M., Inam, O., & Omer, H. (2021, March 27). Transfer learning in deep neural network-based receiver coil sensitivity map estimation - *Magnetic Resonance Materials in Physics, Biology and Medicine*. SpringerLink. <https://doi.org/10.1007/s10334-021-00919-y>
- [15] O. Inam, M. Qureshi, S. A. Malik, and H. Omer, "Iterative Schemes to Solve Low-Dimensional Calibration Equations in Parallel MR Image Reconstruction with GRAPPA," *BioMed Research International*, vol. 2017, p. e3872783, Sep. 2017, doi: <https://doi.org/10.1155/2017/3872783>.
- [16] J. Lyu, Y. Chang, and L. Ying, "Fast GRAPPA reconstruction with random projection," *Magnetic Resonance in Medicine*, vol. 74, no. 1, pp. 71–80, 2015.

- [17] ACM Digital Library. (n.d.). ACM Digital Library. <https://dl.acm.org/doi/abs/10.1145/3314545.3314560>
- [18] A Step-by-Step Explanation of Principal Component Analysis (PCA). (2023, March 29). Built In. <https://builtin.com/data-science/step-step-explanation-principal-component-analysis>.
- [19] Singular value decomposition - Wikipedia. (2023, January 19). Singular Value Decomposition - Wikipedia. https://en.wikipedia.org/wiki/Singular_value_decomposition
- [20] QR-decomposition based SENSE reconstruction using parallel architecture. (2018, February 2). QR-decomposition Based SENSE Reconstruction Using Parallel Architecture - ScienceDirect. <https://doi.org/10.1016/j.compbio.2018.01.013>
- [21] Classical Gram–Schmidt vs Modified Gram–Schmidt. (n.d.). from <https://arnold.hosted.uark.edu/NLA/Pages/CGSMGS.pdf>
- [22] Irturk, Ali & Benson, Bridget & Mirzaei, Shahnaz & Kastner, Ryan. (2010). GUSTO: An Automatic Generation and Optimization Tool for Matrix Inversion Architectures. *ACM Trans. Embedded Compute. Syst.* 9. 10.1145/1721695.1721698.
- [23] P. Kanerva, J. Kristoferson, and A. Holst. Random indexing of text samples for latent semantic analysis. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 22, 2000.
- [24] E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 245–250, 2001.
- [25] N. Ailon and B. Chazelle. The fast Johnson–Lindenstrauss transforms and approximate nearest neighbors. *SIAM Journal on Computing*, 39(1):302–322, 2009.
- [26] D. Achlioptas. Database-friendly random projections: Johnson–Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003.

- [27] A. Dasgupta, R. Kumar, and T. Sarlós. A sparse Johnson–Lindenstrauss transform. In Proceedings of the 42nd ACM Symposium on Theory of Computing, pages 341–350, 2010.
- [28] W. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into Hilbert space. Contemporary Mathematics, 26(189-206):1, 1984.
- [29] A. Andoni and P. Indyk. Near-optimal hashing algorithms for near neighbor problems in high dimension. Communications of the ACM, 51(1):117–122, 2008.
- [30] P. Li, T. Hastie, and K. Church. Very sparse random projections. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 287–296, 2006.
- [31] D. Kane and J. Nelson. Sparser Johnson–Lindenstrauss transforms. Journal of the ACM, 61(1):1–23, 2014.
- [32] Nir Ailon and Bernard Chazelle. 2006. Approximate nearest neighbors and the fast Johnson- Lindenstrauss transform. To appear in STOC '06.
- [33] Dimitris Achlioptas. 2001. Database-friendly random projections. In PODS '01: Proceedings of the Twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pages 274–281, New York, NY, USA. ACM Press.
- [34] Ping Li, Trevor J. Hastie, and Kenneth W. Church. 2006. Very sparse random projections. In KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 287–296, New York, NY, USA. ACM Press.
- [35] J. Lyu, Y. Chang, and L. Ying, “Fast GRAPPA reconstruction with random projection,” Magnetic Resonance in Medicine, vol. 74, no. 1, pp. 71–80, 2015.
- [36] DDR4 (Double Data Rate 4) Definition. (2022, January 17). DDR4 (Double Data Rate 4) Definition. <https://techterms.com/definition/ddr4>

- [37] What is an Arm processor? (2022, May 1). WhatIs.com.
<https://www.techtarget.com/whatis/definition/ARM-processor>
- [38] Direct memory access - Wikipedia. (2023, May 1). Direct Memory Access - Wikipedia.
https://en.wikipedia.org/wiki/Direct_memory_access
- [39] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, "A Survey of Sparse Representation: Algorithms and Applications," in *IEEE Access*, vol. 3, pp. 490-530, 2015, doi: 10.1109/ACCESS.2015.2430359.
- [40] Markovsky, Low Rank Approximation. 2012. doi: <https://doi.org/10.1007/978-1-4471-2227-2>.
- [41] Hui Zou, Trevor Hastie & Robert Tibshirani (2006) Sparse Principal Component Analysis, *Journal of Computational and Graphical Statistics*, 15:2, 265-286, doi: 10.1198/106186006X113430.
- [42] M. I. Sharif, J. P. Li, M. A. Khan, and M. A. Saleem, "Active deep neural network features selection for segmentation and recognition of brain tumors using MRI images," *Pattern Recognition Letters*, vol. 129, pp. 181–189, Jan. 2020, doi: <https://doi.org/10.1016/j.patrec.2019.11.019>.
- [43] A. B. Kerr et al., "Real-time interactive MRI on a conventional scanner," *Magnetic Resonance in Medicine*, vol. 38, no. 3, pp. 355–367, Sep. 1997, doi: <https://doi.org/10.1002/mrm.1910380303>.

