

東南大學

# 毕业设计(论文)报告

题目:

基于机器学习的矩阵优化算法在

大规模 MIMO 系统的应用研究

学号:

04017323

姓名:

赵文远

学院:

信息科学与工程学院

专业:

信息工程

指导教师:

汪磊

起止日期:

2021.01.01 - 2021.06.05

## 东南大学毕业（设计）论文独创性声明

本人声明所呈交的毕业（设计）论文是我个人在导师指导下进行的研究工作及取得的成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得东南大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

论文作者签名： 赵文远 日期： 2021 年 5 月 20 日

## 东南大学毕业（设计）论文使用授权声明

东南大学有权保留本人所送交毕业（设计）论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括刊登）论文的全部或部分内容。论文的公布（包括刊登）授权东南大学教务处办理。

论文作者签名： 赵文远 导师签名： 汪磊  
日期： 2021 年 5 月 20 日 日期： 2021 年 5 月 20 日

## 摘 要

大规模 MIMO 技术是 5G 和 6G 移动通信研究的热点内容，在大规模 MIMO 系统的下行信道，发射端通过利用预编码技术来减轻干扰并提高频谱效率。本文研究了大规模 MIMO 和预编码技术中涉及的矩阵求逆算法，提出了复值梯度神经网络模型（Complex-valued Gradient Neural Network, CVGNN）求解矩阵的广义逆矩阵，并对模型进行了仿真分析。

首先，论文对大规模 MIMO 技术的研究背景进行了总结，指出了高维矩阵求逆的计算开销问题。论文学习了大规模 MIMO 技术的系统模型，并梳理了现有的矩阵求逆算法。

接着，论文对大规模 MIMO 系统中预编码算法的矩阵运算进行了系统的讨论。论文总结了通信信道下行链路基本的预编码算法，这些算法中都包含一个高维矩阵的求逆，指出了研究矩阵优化算法的必要性。

然后，论文重点研究了求解矩阵广义逆矩阵的优化算法。从数值方法和机器学习方法两个层面进行分析，建立了求解广义逆矩阵的数学模型，并提出了 CVGNN 算法求解矩阵的广义逆。与传统的方法（仅用于实值矩阵求逆）不同，该方法直接对复数域的矩阵进行求解。同时，论文从理论层面对 CVGNN 的收敛性进行了证明，该模型能够以指数速率实现收敛。

最后，论文对 CVGNN 算法进行了仿真验证，从不同规模的随机矩阵，到服从瑞利分布的无线通信信道矩阵，提出的 CVGNN 算法能够有效求解矩阵的广义逆矩阵。CVGNN 在瑞利信道模型下的求逆效果远好于随机复值矩阵，总体来说，CVGNN 算法在大规模 MIMO 系统中有良好的应用前景。

关键词：大规模 MIMO，预编码，广义逆矩阵，CVGNN

## ABSTRACT

Massive MIMO is a hot topic in 5G and 6G mobile communication research. In the downlink of massive MIMO system, the transmitter uses precoding technology to reduce interference and improve spectrum efficiency. This paper mainly studies matrix inversion algorithms involved in massive MIMO system and precoding technology, proposed a complex-valued gradient neural network (CVGNN) to solve the Moore-Penrose inversion of the matrix, and analyzed the model by simulation.

First, the thesis summarizes the background of massive MIMO technology, and learned the system model of massive MIMO technology and matrix inversion algorithms to prepare for the subsequent proposal of new algorithms.

Then, the thesis systematically discusses the matrix operation of the precoding algorithm in the massive MIMO system. The thesis analyzes several basic precoding algorithms for the communication channel downlink, including the maximum ratio transmission algorithm, the zero-forcing algorithm, and the minimum mean square error algorithm.

Then, the thesis focuses on the optimization algorithm for solving the Moore-Penrose inversion of the matrix. Numerical methods and deep learning methods are analyzed. The thesis proposed CVGNN model to solve the Moore-Penrose inversion of the matrix. Unlike the traditional method (only used for real-valued matrix inversion), this method directly solves the matrix in the complex domain. At the same time, the paper proves the convergence of the CVGNN model from the theoretical level.

Finally, the paper conducted simulation on the CVGNN model. The convergence and computational overhead of CVGNN were analyzed and discussed. CVGNN model is effective to solve the matrix inversion, especially the inverse matrix under the Rayleigh channel model. The thesis also analyzes the limitations of the model and the direction for future study. Generally speaking, this model has good application prospects in massive MIMO systems.

Keywords: Massive MIMO, Precoding, Moore-Penrose Inversion, CVGNN

# 目 录

摘 要 .....	I
ABSTRACT .....	II
目 录 .....	III
第一章 绪论 .....	1
1.1 课题背景和意义 .....	1
1.2 研究现状 .....	1
1.2.1 大规模 MIMO 技术 .....	1
1.2.2 矩阵求逆算法 .....	2
1.3 本文研究内容及结构 .....	4
第二章 大规模 MIMO 系统中的矩阵运算 .....	5
2.1 多用户大规模 MIMO 系统 .....	5
2.2 混合预编码 .....	6
2.2.1 最大比发送 (MRT) 算法 .....	8
2.2.2 迫零 (ZF) 算法 .....	8
2.2.3 最小均方误差 (MMSE) 算法 .....	9
2.3 预编码中矩阵运算复杂度分析 .....	9
2.4 本章小结 .....	10
第三章 基于梯度神经网络的矩阵求逆算法 .....	11
3.1 矩阵广义逆 (Moore-Penrose Inversion) .....	11
3.2 矩阵的 QR 分解算法和 SVD 算法 .....	12
3.2.1 矩阵的 QR 分解算法 .....	12
3.2.2 矩阵的 SVD 算法 .....	14
3.3 复值梯度神经网络 (CVGNN) .....	14
3.3.1 损失函数的梯度 .....	16
3.3.2 CVGNN 算法及收敛性证明 .....	16
3.4 本章小结 .....	18
第四章 基于 MATLAB 的矩阵求逆算法仿真 .....	19
4.1 给定矩阵的求逆仿真 .....	19
4.2 随机矩阵的求逆仿真 .....	21

4.3 CVGNN 在大规模 MIMO 系统的矩阵求逆 .....	26
4.4 本章小结 .....	31
第五章 总结与展望 .....	32
5.1 工作总结 .....	32
5.2 工作展望 .....	33
参考文献 .....	34
致    谢 .....	37

# 第一章 绪论

## 1.1 课题背景和意义

近年来，无线通信和传输技术迅猛发展，通信技术愈来愈趋向智能化，物联网、人工智能辅助驾驶、VR、远程医疗等新兴技术正在萌芽和普及。但是，新技术也会带来新的问题和技术挑战，移动用户数量成几何增长，终端服务智能化，导致通信链路的吞吐量和复杂度高速增长<sup>[1]</sup>。为应对移动通信更加趋于高质量和高速通信、高密度用户和移动性场景、低时延和低能耗的需求，5G 移动通信目前已投入商用，6G 的研究工作也在 2019 年正式启动。

大规模 MIMO（Massive MIMO）技术是 5G 和 6G 的研究热点，受到了持续广泛的关注。大规模天线阵列可以提供更高的增益，使基站与用户终端的通信质量大幅提升。但是，在 6G 系统中，大规模 MIMO 的计算复杂度指数提高，系统进行高维矩阵求逆的计算开销问题亟需解决。利用传统数值方法已难以满足大规模 MIMO 系统进行矩阵求逆的需求。近年来，随着对神经网络的进一步研究，基于神经网络的方法也逐渐用于矩阵求逆研究。与传统的数值方法相比，基于神经网络的方法具有一些显著的优势，例如并行处理、分布式存储、硬件实现便捷等。并且，基于神经网络的方法不仅适用于实数矩阵运算，而且可以解决复数矩阵运算的问题。因此，研究一种可以简化矩阵运算复杂度的优化算法，用以解决 6G 移动通信系统中，大规模 MIMO 系统面临计算复杂度指数提高的问题，降低矩阵运算的复杂度，解决硬件成本、能耗问题。

## 1.2 研究现状

### 1.2.1 大规模 MIMO 技术

多输入输出（Multi-input Multi-output, MIMO）技术指在发射机和接收机都使用多根天线，形成有多个信道的通信系统。自 3G 以来，MIMO 技术已被成熟地应用于多个无线通信标准，包括，LTE、WiFi 等<sup>[2]</sup>。通常，发射机和接收机的天线数量越多，可达信道路径就越多，通信容量和频谱效率就越高。但是，更多的天线也导致硬件复杂度和能耗急剧增加。

近年来，随着 MIMO 系统的天线不断增加，天线规模从初期的 4 个或 8 个增加到数百、数千个天线，在此基础上提出了大规模 MIMO（Massive MIMO）。大规模 MIMO 打破了传统 MIMO 的方法，大规模的天线阵列可以自适应运行。大规模天线阵列通过将信号的能量

集中到更小的空间区域发送和接收来提高吞吐量和能量效率，尤其是多用户（例如几十或几百）同时使用时。图 1-1 是大规模 MIMO 系统的模型框图。

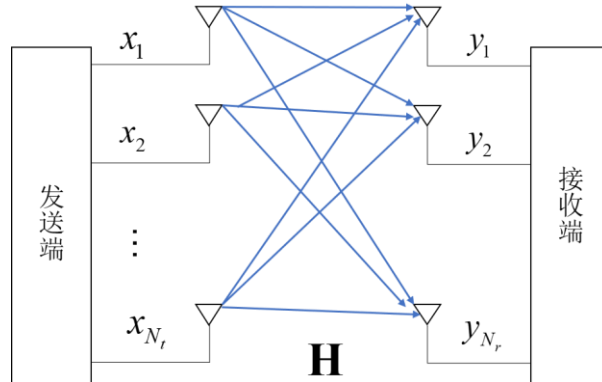


图 1-1 大规模 MIMO 系统框图

大规模 MIMO 系统的信号处理方法通过增加基站端天线来避免衰落、噪声、小区内干扰，只需要简单的线性设计即可实现较好的系统性能<sup>[3]</sup>。大规模 MIMO 中一般采用线性预编码算法，比如最大比发送算法、迫零算法、最小均方误差算法等。根据隆德大学的实验，大规模 MIMO 中低复杂度的线性预编码可以达到非线性预编码 98% 的性能<sup>[4]</sup>。

大规模 MIMO 引出了亟需关注的新问题和挑战，例如：导频污染，信道估计，硬件效率，数据检测，预编码等。在一定的资源中，正交导频信号的数量是有限的，使用相同的一组正交导频可能导致出现小区间干扰，导致导频污染。在频分双工系统中，下行链路传输导频信号所需的时间与 BS 天线的数量成正比，复杂度会随着天线数量增加。大规模 MIMO 接收器的高计算复杂度限制了在实际应用中可获得的增益。同时，大量天线导致较高的计算复杂度也降低了信号检测范围内可实现的吞吐量。

### 1.2.2 矩阵求逆算法

随着无线通信技术的日益发展和移动智能设备数量的急剧增长，大规模 MIMO 系统中的矩阵运算面临计算复杂度高、动态环境下运算成本过大的问题。在矩阵运算中，求解矩阵的逆是信号处理、优化算法中的核心问题，自 1980 年代中期以来，人们一直致力于快速矩阵求逆的计算，并提出了许多算法。

1988 年，Yeung 等提出了一种基于矩阵序列的多维离散傅里叶变换的反演新方法，解决了包含多个变量的矩阵求逆的问题，并给出了其在电子放大器中的应用<sup>[5]</sup>。但是，对于许多数值方法，运算通常与矩阵维度的立方成正比，因此，在计算机上执行的此类算法在



实时应用中可能不够高效。因此，学者们研究了用于矩阵求逆的并行计算方案。

动态方法是解决矩阵求逆问题的重要方法之一。最近，由于对深度学习和神经网络的深入研究，人们已经研究并开发了许多基于递归神经网络（Recurrent Neural Network, RNN）的动态求解器。Steriti 等提出了一种基于 Hopfield 神经网络的方法，其实现的结果与使用奇异值分解（SVD）获得的逆相比更高效、更具有鲁棒性<sup>[6]</sup>。Tank 等提出了一种简单的神经网络来解决实时线性规划问题，并在模拟电路上实现<sup>[7]</sup>。由于并行分布的特性以及硬件实现的便利性，基于神经网络的方法已经被视为在线计算的强大的替代方法。

此外，在线求逆的矩阵有时是随时间变化的，因而对时变矩阵的求逆算法也被学者研究。Yunong Zhang 等人基于一阶时间导数，提出了具有隐式动力学的通用 RNN 模型：Zeroing Neural Network（ZNN），以解决时变矩阵求逆的问题<sup>[8]</sup>。该方法通过定义矩阵值的误差函数而不是通常的标量值作为代价函数，可以优雅地引入神经网络，从而可以使计算误差在整体上基本减小到零。ZNN 模型的构建基于两点：一是选择适当的矩阵值误差函数（或称为张函数） $E(t)$ ；二是建立一种演化规则使  $E(t)$  收敛到 0。为了实现 ZNN 在现场可编程门阵列（FPGA）和专用集成电路（ASIC）中 ZNN 的应用，作者基于 MATLAB Simulink 实现了 ZNN 模型的建模和仿真验证<sup>[9]</sup>。作为 RNN 的典型代表，ZNN 已被证明是解决时变矩阵问题的强大并行处理神经求解器。同时，基于 ZNN 模型的衍生模型也被提出并应用在一些科学与工程学中。Xiao 提出了一种基于非线性演化公式建立的有限时间收敛神经网络，以解决随时间变化的矩阵求逆，它可以加速 ZNN 的收敛并精确地达到理论解<sup>[10]</sup>。Tan 等分析并研究了基于组合激活函数的 ZNN 模型在有噪声的情况下用于求时变矩阵广义逆解的鲁棒性，表明在受恒定常数或时变噪声污染的条件下，通过选择一些设计参数，可以使模型产生的解误差任意小，并成功完成了机器人应用实例，说明了该模型在冗余机器人操纵器运动学控制中的可行性和有效性<sup>[11]</sup>。

神经网络方法不仅可以用来解决实数域中的矩阵运算问题，而且可以通过变换的方法来解决复数域中的问题。复值神经网络可以看作是实值神经网络的扩展，最流行的方法是将复值神经网络分解为两个实值系统，与传统的实值神经网络相比，复值神经网络的最重要优势之一是它们与波动现象有良好兼容性。Zhang 等提出了三种不同的 ZNN 模型来寻找复杂领域中时变矩阵的逆，并且仿真结果验证了这些模型的有效性<sup>[12]</sup>。Wang 等提出了具有两个不同激活函数的复值 ZNN 模型来寻找时变的 Drazin 逆，并且这两个模型都可以在有限的时间内收敛<sup>[13]</sup>。Zhang 等提出了基于复值 Sylvester 方程的 ZNN 模型<sup>[14]</sup>。Liu 等构建

了基于非线性特殊结构的复值非线性神经网络，与先前的工作相比，该网络能在更短的时间内收敛<sup>[15]</sup>。

另一方面，上述提到的基于深度学习的方法均是基于在理想无扰动环境中求解可逆矩阵的逆。但是在大规模 MIMO 无线通信系统中，硬件实现过程不可避免地会出现一些外部噪声，这些噪声可以是恒定噪声、随机噪声、混合噪声等。尽管可以预先处理噪声（例如使用低通滤波器来处理误差补偿），整个过程仍然会浪费宝贵的时间。为了更好的硬件实现，有学者提出了一种 Complex-valued Noise-tolerant Zeroing Neural Network (CVNTZNN) 模型，并对其全局指数收敛性和噪声容忍度进行了理论分析，证明了在各种噪声伴随下，CVNTZNN 模型具有鲁棒性<sup>[16]</sup>。

### 1.3 本文研究内容及结构

本文研究内容及论文结构安排如下：

第一章：介绍了课题研究的背景和意义，调研了大规模 MIMO 技术的技术特点，梳理了矩阵求逆算法的研究现状。

第二章：对大规模 MIMO 中的矩阵运算，尤其是预编码中的矩阵运算进行系统的讨论。首先，阐述了大规模 MIMO 的优势和应用场景，构建了大规模 MIMO 的下行信道模型。然后，分析了三种常见的预编码算法中的矩阵运算。

第三章：系统的讨论了预编码算法中的矩阵运算。首先，给出了广义逆矩阵的定义和性质，建立了矩阵求逆的数学模型。然后，阐述了基于数值的矩阵求逆方法：QR 分解法和 SVD 算法。接着，提出了复值梯度神经网络 (CVGNN) 算法求解矩阵的广义逆，并从理论上证明了该网络的收敛性，并给出了指数收敛速度。

第四章：在 MATLAB 环境下对 CVGNN 算法求解广义逆矩阵的性能进行了仿真实验。首先，通过给定的复值矩阵分析了网络的各元素值、收敛速度和参数性能，初步证明了 CVGNN 算法的有效性。然后，对随机生成的不同规模的矩阵进行了实验，进一步分析了算法的效率，并指出了在随机矩阵实验下的局限性。然后，分析了大规模 MIMO 系统中瑞利信道的特征，并在瑞利信道条件下进一步实验，验证了 CVGNN 算法在大规模 MIMO 系统中应用的可行性，有一定的应用前景。

第五章：总结本文工作，提出后续改进方向。

## 第二章 大规模 MIMO 系统中的矩阵运算

### 2.1 多用户大规模 MIMO 系统

大规模 MIMO 是 MIMO 系统自 3G 以来引入的一种扩展。大规模 MIMO 系统的基站端配有成百上千根天线，同时服务于很多用户终端<sup>[17]</sup>。图 2-1 描述了一个大规模 MIMO 的应用场景，在特定小区创建定向波束服务于一个或少量用户，这带来了很多好处：

- **光谱效率：**大规模 MIMO 系统通过利用大型天线阵列来产生更多的复用增益以实现更高的频谱效率。因此，每个用户设备都有一个单独的下行光束，它提供的光谱效率是传统 MIMO 技术的十倍。
- **能源效率：**大规模 MIMO 系统通过将天线阵列的波束指向一个小区域，可以将传输信号的增益增加到指定用户的位置。因此，大规模 MIMO 系统辐射掉的功率更少，而且更节能。
- **用户跟踪：**通过将窄波束信号指向用户，用户跟踪变得更加可靠、准确。
- **成本：**大规模 MIMO 系统采用廉价的超低功率放大器，这降低了对昂贵的电子设备的需求。此外，它消除了连接基站的庞大的同轴电缆。

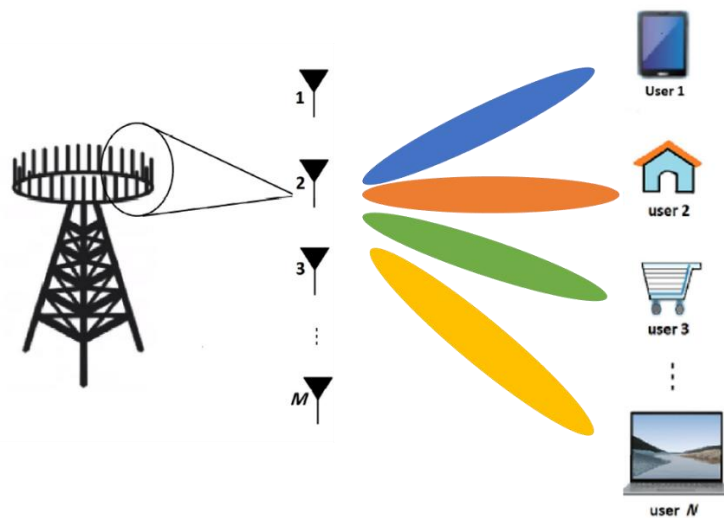


图 2-1 大规模 MIMO 应用场景

大规模 MIMO 系统的下行信道模型如下：一个配备有  $M$  个发射天线和  $N$  个接收天线的下行链路多用户大规模 MIMO 系统如图 2-2 所示。每个基站（base station, BS）同时服务于  $K$  个用户终端（user end, UE）。在不失一般性的前提下，我们假设每个用户的接收天

线数量是相同的。考虑平坦的信道衰落，进一步假设信道状态信息在发射机处是完全已知的，并且基站与用户同步。

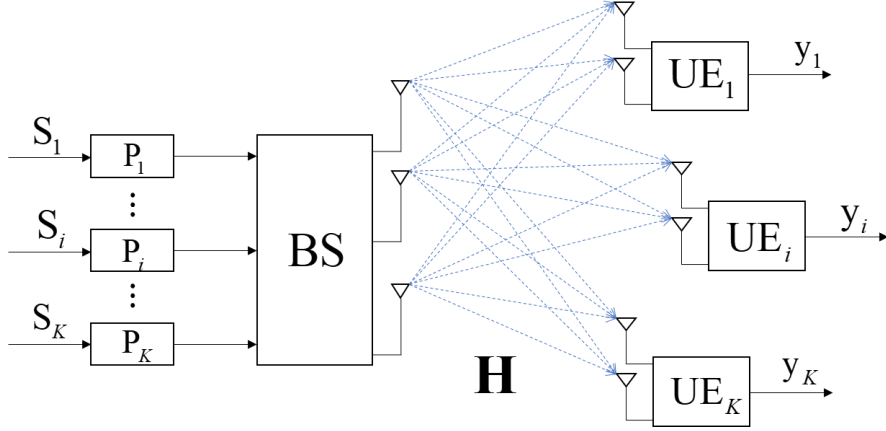


图 2-2 多用户大规模 MIMO 系统模型  $\mathbf{S}_k$  表示传输的符号向量，通

过预编码  $\mathbf{P}_k$  耦合到基站端，通过发射机发送到 UE， $\mathbf{H}_k$  为信道矩阵

将基站端耦合到第  $k$  个用户，平坦瑞利衰落下 MIMO 信道模型为：

$$\mathbf{H}_k = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1M} \\ h_{21} & h_{22} & \cdots & h_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ h_{l_k 1} & h_{l_k 2} & \cdots & h_{l_k M} \end{bmatrix} \quad (2.1)$$

这里  $\mathbf{H}_k \in \mathbb{C}^{l_k \times M}$  表示用户  $k$  的 MIMO 信道矩阵， $\mathbf{H}_k$  服从复高斯分布，元素  $h_{u,v}$  表示将第  $k$  个发射天线耦合到接收天线的信道脉冲响应。系统信道矩阵如下：

$$\mathbf{H} = [\mathbf{H}_1^T \quad \mathbf{H}_2^T \quad \cdots \quad \mathbf{H}_K^T]^T \quad (2.2)$$

设用户  $k$  的预编码矩阵为  $\mathbf{P}_k$ ，用户  $k$  的传输符号向量为  $\mathbf{s}_k$ ，且用户  $k$  存在加性白高斯噪声  $n_k$ ，满足零均值、方差  $\sigma^2$ ，则用户  $k$  的接收信号表达式为：

$$y_k = \mathbf{H}_k \mathbf{P}_k \mathbf{s}_k + \mathbf{H}_k \sum_{i=1, i \neq k}^K \mathbf{P}_i \mathbf{s}_i + n_k \quad (2.3)$$

## 2.2 混合预编码

大规模 MIMO 系统的一个主要概念是预编码技术，通过在发射机上使用强大的信号处理技术，将复杂的系统从接收终端转换到了基站侧。预编码是波束赋形的一个概念，它可

以减少干扰和路径损耗的影响，并增加吞吐量。在大规模 MIMO 系统中，基站可以通过从接收终端发送的上行导频信号来估计信道状态信息（Channel State Information, CSI）。但是由于无线信道上存在的环境障碍，在实际的无线通信环境中很难获得可靠的 CSI。尽管如此，基站端在下行链路的性能仍然很大程度上取决于估计的 CSI。大规模 MIMO 系统利用预编码技术和估计的 CSI 来减轻干扰并提高频谱效率。但是，这些好处带来了很高的计算复杂度，计算复杂度与天线数量成正比。因此，在大规模 MIMO 系统中必须使用低复杂度的预编码器。

大规模 MIMO 预编码通过控制波束的方向并将其指向特定的接收终端位置，预编码算法主要分为线性、非线性、PAPR 预编码和基于机器学习的预编码算法<sup>[18]</sup>。图 2-3 展示了通信系统中预编码技术的通用框图，图中  $\mathbf{P}$  是线性预编码的前馈矩阵， $\mathbf{B}$  是线性预编码的反馈矩阵，Modulo 算法用于调整平均功率。

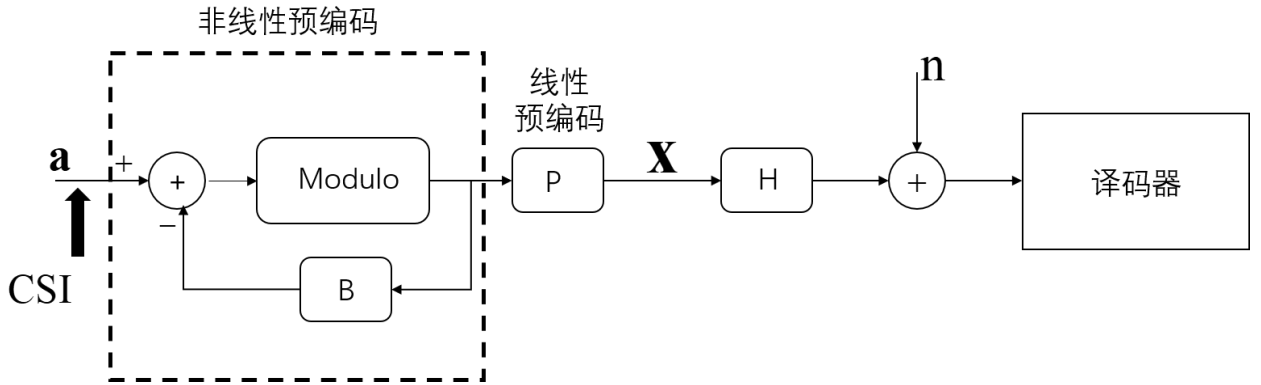


图 2-3 预编码通用框图

根据图 2-3 所示，拥有  $N$  个用户的下行链路传输可表示为：

$$\mathbf{x} = \sqrt{\rho} \mathbf{P} \mathbf{a}, \quad \mathbf{x} \in \mathbb{C}^{M \times 1} \quad (2.4)$$

$\mathbf{P}$  是  $M \times N (M > N)$  的线性预编码矩阵， $\sqrt{\rho}$  是平均传输功率。预编码矩阵  $\mathbf{P}$  与信道矩阵  $\mathbf{H}$  相关，在 TDD 模式下，下行信道是  $\mathbf{H}$  的转置，所以接收端的信号为：

$$\begin{aligned} \mathbf{y} &= \mathbf{H}^T \mathbf{x} + \mathbf{n} \\ &= \sqrt{\rho} \mathbf{H}^T \mathbf{P} \mathbf{a} + \mathbf{n}, \quad \mathbf{y} \in \mathbb{C}^{N \times 1} \end{aligned} \quad (2.5)$$

通常，预编码矩阵  $\mathbf{P}$  包含一个高维矩阵求逆运算，这会导致很高的计算复杂度。根据矩阵求逆的过程，线性预编码技术可分为基本的线性预编码、基于矩阵反演近似的线性预

编码、基于定点迭代的线性预编码器和基于矩阵分解的线性预编码。常见的线性预编码算法有最大比发送（Maximum Ratio Transmission, MRT）算法、迫零（Zero-Forcing, ZF）算法、最小均方误差（Minimum Mean Square Error, MMSE）算法等。

### 2.2.1 最大比发送（MRT）算法

MRT 的目标是使进入特点接收终端的信号增益最大化，对应于匹配滤波（Matched Filtering, MF）和共轭波束赋形（Conjugate Beamforming, CB）。MRT 的矩阵方程为：

$$\mathbf{P}_{\text{MRT}} = \sqrt{\beta} (\mathbf{H}^*) \quad (2.6)$$

其中  $\beta$  表示尺度功率因数， $\mathbf{H}^*$  是  $\mathbf{H}$  的复共轭。因此，接收端信号为：

$$\mathbf{y}_{\text{MRT}} = \sqrt{\beta} \sqrt{\rho} \mathbf{H}^T \mathbf{H}^* \mathbf{a} + \mathbf{n}, \quad \mathbf{y} \in \mathbb{C}^{N \times 1} \quad (2.7)$$

当  $M$  远大于  $N$  且  $M$  趋于无穷大时，MRT 算法可达到大规模 MIMO 系统的总容量。通常当用户间干扰（Inter-user Interference, IUI）与噪声的比值较小时，MRT 算法的性能接近最优。在 MRT 算法中，当  $M$  和  $N$  相当时，系统 IUI 较大。因此，每个用户的吞吐率变得很低，不利于大规模 MIMO 系统。MRT 算法的另一个特点是，基站中的每个天线都可以在本地进行信号处理，这使得大规模天线阵列可以分散配置，使大规模 MIMO 系统更加灵活<sup>[19]</sup>。

### 2.2.2 迫零（ZF）算法

ZF 算法是预编码技术的一种基本算法，对应于信道反演。ZF 算法将信号波束指向目标用户，同时使波束在其他用户所在的方向上无效，来减轻其他用户造成的干扰。ZF 将用户数据与预编码矩阵相乘，使使无关方向上的波束清零：

$$\mathbf{P}_{\text{ZF}} = \sqrt{\beta} \mathbf{H}^* (\mathbf{G}^{-1}), \quad \mathbf{G} = \mathbf{H}^T \mathbf{H}^* \quad (2.8)$$

其中  $\mathbf{G}$  是一个 *Gram* 矩阵，其对角元素表示功率不平衡抛出信道，非对角元素表示信道的互相关。当大规模 MIMO 系统中的发射天线的数量趋于无穷大时， $\mathbf{G}$  趋于一个单位阵。ZF 算法接收端的信号可表示为：

$$\mathbf{y}_{\text{ZF}} = \sqrt{\beta} \sqrt{\rho} \mathbf{H}^T \mathbf{H}^* (\mathbf{G}^{-1}) \mathbf{a} + \mathbf{n}, \quad \mathbf{y} \in \mathbb{C}^{N \times 1} \quad (2.9)$$

大规模 MIMO 信道模型忽略 AWGN 时，ZF 算法更加实用，而大规模 MIMO 的预编码也更容易实现。但是，实际情况下噪声不可被忽略，因而 ZF 算法很难给出大规模 MIMO 系统的最优解，只能在高信噪比（SNR）的条件下取得较准确的结果<sup>[20]</sup>。

### 2.2.3 最小均方误差（MMSE）算法

MMSE 算法兼顾了 MRT 和 ZF 的优点，因此它在正常噪声和干扰的系统中表现较优。MMSE 算法使用最小均方误差的方法使基站和接收终端间的误比特率最小。MMSE 的预编码矩阵为：

$$\mathbf{P}_{\text{MMSE}} = \sqrt{\beta} \mathbf{H}^* (\mathbf{G} + \mathbf{V} + \lambda \mathbf{I}_N)^{-1} \quad (2.10)$$

其中  $\lambda$  是大于 0 的正则化因子， $\mathbf{V}$  是  $N \times N$  的厄米特（Hermitian）非负正定矩阵，接收端的信号为：

$$\mathbf{y}_{\text{MMSE}} = \sqrt{\beta} \sqrt{\rho} \mathbf{H}^T \mathbf{H}^* (\mathbf{G} + \mathbf{V} + \lambda \mathbf{I}_N)^{-1} \mathbf{a} + \mathbf{n} \quad (2.11)$$

但是，上述预编码矩阵的计算中都包括一个高维矩阵的求逆，尤其是当  $M$  和  $N$  的值较大时。因而有必要提出一种降低基本预编码算法复杂度的方法。

## 2.3 预编码中矩阵运算复杂度分析

在移动通信系统中，通常选择基于码本的预编码矩阵操作，接收机根据估计的 CSI 进行量化，然后发送匹配的码本信息，以协助通信基站选择方案。如果立即反馈未量化的 CSI 或基于 TDD 估计 CSI，则生成的预编码矩阵将更加准确，但同时也会遇到很多运算复杂度问题。

在实际系统中，若大规模 MIMO 基站端天线数  $M = 128$ ，选用  $N = 1024$  个子载波，用户数  $K = 40$ ，则总信道参数是  $N \times M \times K = 5242880$ ，约 524 万个。此外，实际情况下用户大多处于移动状态，在 3.5GHz 频段步行时，Takes 衰落下环境信道相关性下降到 90% 时才更新 CSI，也需要约每秒 100 次更新。因此，如果不考虑复杂度问题，大规模 MIMO 大约需要每秒计算 5.2 亿个信道参数。其中存在大量的由于 ZF、MMSE 所带来的矩阵求逆和乘法运算。

对此，通常在算法侧、计算侧和硬件配置侧都有简化的计算方法。例如使用遗传算法获得快速但非全局的最优解，或者使用深度学习算法训练深层神经网络，直接输出 CSI 等。目前，华为使用的算法是随机森林。

在[21]中，考虑了子载波的可用性，以最大限度地提高传输速率和能源效率。在[22]中，提出了一种新的节能方法来降低计算复杂度。此外，ZF 和 MMSE 预编码矩阵的计算包含一个高维矩阵的反演，特别是当  $M$  和  $N$  的值较大时<sup>[23]</sup>。当  $M$  远大于  $N$  时，矩阵  $\mathbf{G}$

的对角元素将占主导地位，非对角分量趋于零，对角分量接近  $\mathbf{M}$ 。当前大量的研究通过近似或避免矩阵反演来降低运算复杂度，而不是直接计算它。除了矩阵反演的高复杂度外，矩阵反演中的另一个缺陷是近奇异和病态矩阵的反演。为了克服噪声，需要近似/避免直接计算矩阵反演的方法，例如：

- a) **Truncated Polynomial Expansion (TPE) 算法：**TPE 算法利用了已知预编码矩阵的近似值，而不是 MMSE 算法中的矩阵求逆，通过不同的截断顺序来平衡计算复杂度和可达数据速率<sup>[24]</sup>。
- b) **Neumann Series Approximation (NSA) 算法：**NSA 算法可以通过诺伊曼级数来解决高计算复杂度的问题。在 NSA 算法中，传统 MMSE 算法中的求逆过程被一系列的矩阵向量乘法（幂/和）所取代，数据流简单，而且可以高度并行<sup>[25]</sup>。
- c) **Newton Iteration (NI) 算法：**NI 算法是一种近似矩阵求逆的方法。NI 算法只需要一个简单的计算即可加速预编码过程。虽然 NI 在每次迭代中需要一个额外的矩阵乘法，但它的收敛速度比 NSA 算法要快。
- d) **Chebyshev Iteration (CI) 算法：**CI 算法通过迭代逼近线性预编码中的矩阵逆。CI 算法的收敛速度受到迭代初始值的影响，张等对 CI 算法的初始值进行优化<sup>[26]</sup>。仿真结果表明，优化的 CI 算法与 RZF 算法有相同的平均可达率。优化后的 CI 算法在经过一次迭代后就与 CI 算法有相同的性能。

## 2.4 本章小结

本章首先对大规模 MIMO 的应用场景和系统模型进行了介绍，大规模 MIMO 在光谱效率、能源效率、用户跟踪、成本、可靠性方面有很大的优势。接着，给出了大规模 MIMO 的下行信道模型，并对其中的预编码算法进行了研究，大规模 MIMO 利用预编码技术和估计的 CSI 来减轻干扰并提高频谱效率。然后，分析了几种常见的预编码算法及运算复杂度，包括 MRT、ZF 和 MMSE，并指出常用的预编码算法中均涉及一个高维矩阵的求逆，尤其当天线数和用户数量较大时，因而有必要提出一种降低基本预编码算法中矩阵求逆复杂度的方法。



### 第三章 基于梯度神经网络的矩阵求逆算法

在第二章中，基本预编码算法中当天线数量  $M$  和用户数量  $N$  较大时， $\mathbf{G}$  的求逆非常复杂。除了高维矩阵求逆的高复杂度，还需要面临近奇异和病态矩阵。因而，需要研究一种快捷、低复杂度的矩阵求逆算法。本章介绍了矩阵广义逆、基于数值方法的矩阵求逆，并基于梯度神经网络的矩阵求逆算法。

#### 3.1 矩阵广义逆（Moore-Penrose Inversion）

设  $\mathbf{A}$  是  $N \times N$  的复数矩阵，即  $\mathbf{A} \in \mathbb{C}^{N \times N}$ ， $\mathbf{A}$  的逆矩阵  $\mathbf{X}$  满足方程：

$$\mathbf{A}\mathbf{X}(t) = \mathbf{I} \in \mathbb{C}^{N \times N}, \text{ 或 } \mathbf{X}(t)\mathbf{A} = \mathbf{I} \in \mathbb{C}^{N \times N} \quad (3.1)$$

在过去的大多数研究中，通常将  $\mathbf{A} \in \mathbb{C}^{N \times N}$  分解为实部和虚部分别进行处理，用下标  $re$  表示矩阵的实数部分，下标  $im$  表示矩阵的虚数部分，例如  $\mathbf{A} = \mathbf{A}_{re} + j\mathbf{A}_{im}$ ，令  $j = \sqrt{-1}$ ，则 (3.1) 等价于：

$$[\mathbf{A}_{re} + j\mathbf{A}_{im}][\mathbf{X}_{re}(t) + j\mathbf{X}_{im}(t)] = \mathbf{I}_{re} + j\mathbf{I}_{im} \quad (3.2)$$

将(3.2)展开后可得到实数域的等价方程组：

$$\begin{cases} \mathbf{A}_{re}\mathbf{X}_{re}(t) - \mathbf{A}_{im}\mathbf{X}_{im}(t) = \mathbf{I}_{re} \in \mathbb{R}^{N \times N} \\ \mathbf{A}_{re}\mathbf{X}_{im}(t) + \mathbf{A}_{im}\mathbf{X}_{re}(t) = \mathbf{I}_{im} \in \mathbb{R}^{N \times N} \end{cases} \quad (3.3)$$

即： 
$$\begin{bmatrix} \mathbf{A}_{re} & -\mathbf{A}_{im} \\ \mathbf{A}_{im} & \mathbf{A}_{re} \end{bmatrix} \begin{bmatrix} \mathbf{X}_{re}(t) \\ \mathbf{X}_{im}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{re} \\ \mathbf{I}_{im} \end{bmatrix} \in \mathbb{R}^{2N \times N}$$

用  $\mathbf{B} \in \mathbb{R}^{2N \times N}$ ， $\mathbf{Y}(t) \in \mathbb{R}^{2N \times N}$  和  $\mathbf{D} \in \mathbb{R}^{2N \times N}$  替代(3.3)中的矩阵，即：

$$\mathbf{B} = \begin{bmatrix} \mathbf{A}_{re} & -\mathbf{A}_{im} \\ \mathbf{A}_{im} & \mathbf{A}_{re} \end{bmatrix}, \mathbf{Y}(t) = \begin{bmatrix} \mathbf{X}_{re}(t) \\ \mathbf{X}_{im}(t) \end{bmatrix}, \mathbf{D} = \begin{bmatrix} \mathbf{I}_{re} \\ \mathbf{I}_{im} \end{bmatrix}$$

那么原复数方程则可转化为实数域下的矩阵方程问题求解：

$$\mathbf{B}\mathbf{Y}(t) = \mathbf{D} \in \mathbb{R}^{2N \times N} \quad (3.4)$$

但是，随着大规模 MIMO 中天线数量和用户数量的增加，实际应用中天线数  $M$  和用户终端数  $N$  不一定相同，因而信道矩阵可能不是方阵。所以，求解矩阵的广义逆矩阵（Moore-Penrose Inversion）更加有必要。

复数矩阵  $\mathbf{Z} \in \mathbb{C}^{M \times N}$  的广义逆矩阵可以表示为： $\mathbf{Z}^\dagger \in \mathbb{C}^{N \times M}$ ，根据广义逆矩阵的定义<sup>[27]</sup>：

$$(ZZ^\dagger)^H = ZZ^\dagger, (Z^\dagger Z)^H = Z^\dagger Z, ZZ^\dagger Z = Z, Z^\dagger ZZ^\dagger = Z^\dagger \quad (3.5)$$

这里 $(\bullet)^H$ 表示矩阵的厄米特算子（Hermitian），即矩阵的共轭转置。

设 $\mathbf{A}$ 是 $M \times N$ 的复数矩阵，即 $\mathbf{A} \in \mathbb{C}^{M \times N}$ ， $\mathbf{X}$ 是 $\mathbf{A}$ 的广义逆矩阵 $\mathbf{X} = \mathbf{A}^\dagger \in \mathbb{C}^{N \times M}$ ，根据(3.5)有：

$$\begin{aligned} \mathbf{A}\mathbf{X}\mathbf{A} &= \mathbf{A}, \\ \mathbf{X}\mathbf{A}\mathbf{X} &= \mathbf{X}, \\ (\mathbf{A}\mathbf{X})^H &= \mathbf{A}\mathbf{X}, \\ (\mathbf{X}\mathbf{A})^H &= \mathbf{X}\mathbf{A} \end{aligned} \quad (3.6)$$

下面将介绍基于数值和基于神经网络方法的矩阵求逆算法。

## 3.2 矩阵的 QR 分解算法和 SVD 算法

### 3.2.1 矩阵的 QR 分解算法

QR 分解算法是硬件系统中矩阵求逆的一种常用算法，其核心思想是将待求矩阵 $\mathbf{A}$ 分解为一个正交矩阵 $\mathbf{Q}$ 和一个上三角矩阵 $\mathbf{R}'$ 的积，设 $\mathbf{A} \in \mathbb{R}^{M \times N} (M > N)$ ， $\mathbf{Q} \in \mathbb{R}^{M \times M}$ ， $\mathbf{R} \in \mathbb{R}^{M \times N}$ ，使：

$$\begin{aligned} \mathbf{A} &= \mathbf{Q}\mathbf{R}, \\ \mathbf{R} &= \begin{bmatrix} \mathbf{R}' \\ 0 \end{bmatrix} \end{aligned} \quad (3.7)$$

由于正交矩阵和上三角矩阵的逆较为容易实现，正交矩阵的逆等于其转置矩阵，根据矩阵运算的基本性质，只需先求出 $\mathbf{Q}^{-1}$ 和 $\mathbf{R}^{-1}$ ，那么 $\mathbf{A}$ 的逆矩阵即为 $\mathbf{R}^{-1}\mathbf{Q}^{-1}$ 。QR 分解算法最难实现的部分在于如何将矩阵 $\mathbf{A}$ 分解为符合(3.7)的形式，常用的算法有基于吉文斯旋转（Givens Rotation）的分解算法。

#### Givens Rotation

Givens Rotation 是一种正交变换，经过多次吉文斯变换可以将矩阵转换成上三角形式。平面 $\mathbb{R}^2$ 中旋转对应的变换矩阵为：

$$\mathbf{A} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad (3.8)$$

在  $n$  维欧氏空间中取一组标准正交基，沿平面旋转后的矩阵为：

$$\mathbf{G}_{ij} = \begin{bmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & \cos \theta & 0 & \cdots & 0 & \sin \theta & \\ & & & 0 & 1 & \cdots & 0 & 0 & \\ & & & \vdots & \vdots & & \vdots & \vdots & \\ & & & 0 & 0 & \cdots & 1 & 0 & \\ & & & -\sin \theta & 0 & \cdots & 0 & \cos \theta & \\ & & & & & & & & 1 & \ddots & \\ & & & & & & & & & & 1 \end{bmatrix} \quad (3.9)$$

对应非零的  $n$  维列向量  $\mathbf{x}$  如果以  $\mathbf{G}_{ij}$  左乘，则  $\mathbf{G}_{ij}\mathbf{x}$  将只改变  $\mathbf{A}$  的第  $i$  和第  $j$  个分量，不会改变其他元素。要使第  $j$  个分量简化为 0，只需令：

$$\sin \theta = \frac{x_j}{\sqrt{x_i^2 + x_j^2}}, \cos \theta = \frac{x_i}{\sqrt{x_i^2 + x_j^2}} \quad (3.10)$$

对非零  $n$  维列向量  $\mathbf{x}$  连续左乘以  $\mathbf{G}_{r,r+1}$ ,  $\mathbf{G}_{r,r+2}$ ,  $\cdots$ ,  $\mathbf{G}_{r,n}$ , 可使  $\mathbf{x}$  从第  $r+1$  到第  $n$  个分量全部转化为 0，即：

$$\mathbf{G}_{r,n} \cdots \mathbf{G}_{r,r+2} \mathbf{G}_{r,r+1} \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{r-1} \\ y_r \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (3.11)$$

如此，令第一个 Givens 矩阵使  $\mathbf{A}$  的第一列只有第一行不为 0，第二个 Givens 矩阵使  $\mathbf{A}$  的第二列只有前两行不为 0，依此类推：

$$\mathbf{R} = \mathbf{G}_n \mathbf{G}_{n-1} \cdots \mathbf{G}_1 \mathbf{A} \quad (3.12)$$

由于 Givens 矩阵本身即为正交矩阵，所以  $\mathbf{G}_n \mathbf{G}_{n-1} \cdots \mathbf{G}_1$  也为正交矩阵，所以：

$$\mathbf{Q} = \mathbf{G}_n \mathbf{G}_{n-1} \cdots \mathbf{G}_1 \quad (3.13)$$

### 3.2.2 矩阵的 SVD 算法

SVD(Singular Value Decomposition), 即奇异值分解, 是一种线性代数算法, 用来对矩阵进行拆分。拆分后的矩阵可以提取关键信息, 降低规模, 被广泛应用在不同领域。

设矩阵  $\mathbf{A} \in \mathbb{R}^{M \times N}$ ,  $\mathbf{\Sigma}$  是一个  $M \times N$  的对角矩阵, 该矩阵对角元素为该矩阵的奇异值, 其余元素均为 0。矩阵  $\mathbf{U}$  和  $\mathbf{V}$  都是  $N \times N$  的正交矩阵, 即满足  $\mathbf{U}\mathbf{U}^T = \mathbf{I}$ ,  $\mathbf{V}\mathbf{V}^T = \mathbf{I}$ 。根据 SVD 算法, 我们可以将矩阵  $\mathbf{A}$  分解为:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (3.14)$$

对  $\mathbf{A}^T \mathbf{A}$  进行特征分解, 假设得到的特征值是  $\lambda_i$ , 特征向量是  $\mathbf{v}_i$ , 代入特征值的性质可以得到:

$$(\mathbf{A}^T \mathbf{A}) \mathbf{v}_i = \lambda_i \mathbf{v}_i \quad (3.15)$$

这样的特征值和特征向量共有  $n$  个, 将其所有的特征向量组合在一起, 可以得到一个  $N \times N$  的矩阵  $\mathbf{V}$ , 即 SVD 分解之后的矩阵  $\mathbf{V}$ 。同理, 对  $\mathbf{A}\mathbf{A}^T$  进行特征分解可以得到一个  $M \times M$  的矩阵  $\mathbf{U}$ , 即 SVD 分解之后的  $\mathbf{U}$ 。根据特征矩阵的性质, 特征值矩阵等于奇异值矩阵的平方, 可以得到奇异值, 进而得到  $\mathbf{\Sigma}$ :

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_0 & & & \\ & \ddots & & \\ & & \sigma_i & \\ & & & \ddots \end{bmatrix} \quad (3.16)$$

### 3.3 复值梯度神经网络 (CVGNN)

过去, 人们已经为实现快速矩阵求逆做出了很多的努力, 并且开发了各种基于数值方法的算法来求解逆矩阵。但是, 由于这些算法的计算操作与矩阵的大小成比例, 因此它们对于实时应用程序和大规模基准计算可能不够有效。为了克服传统数值方法的计算瓶颈, 循环神经网络 (Recurrent Neural Network, RNN) 被应用到矩阵求逆问题。与传统的数值方法相比, RNN 具有高容错性、自适应性、分布式存储和并行处理。但是, 目前大多数研究

仍然集中于实数域矩阵的求逆，而通信系统中输入信号包含相位和幅度信息，需要对复数域矩阵进行求逆。如 3.1 节所述，将复数域的问题通过实部和虚部的分解转化到实数域解决，这会增加建模和计算的复杂度。本节将提出复值梯度神经网络(Complex-valued Gradient Neural Network, CVGNN)的矩阵求逆算法求复值矩阵的广义逆。

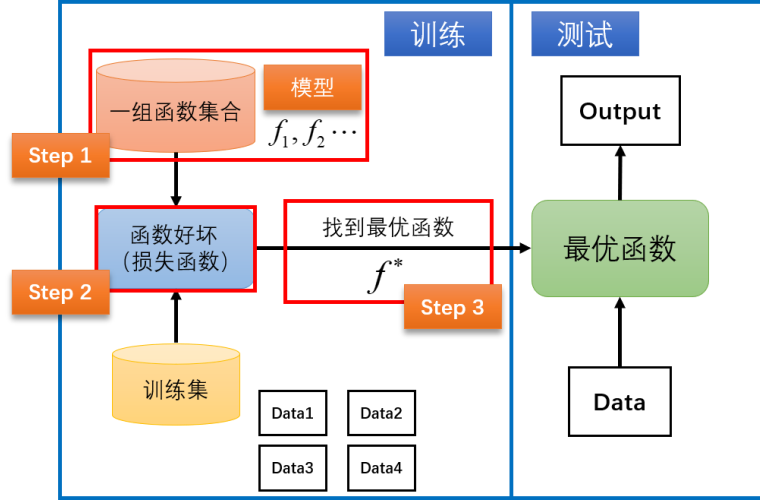


图 3-1 机器学习步骤

图 3-1 显示了机器学习的基本步骤：

STEP 1：根据问题建立模型。

STEP 2：确定损失函数，建立最优化问题模型，回归问题通常采用欧式距离，分类问题通常采用交叉熵代价函数。

STEP 3：找到最优解，常用的方法有梯度下降算法，最小二乘法等。

与常规设计处理不同，本文提出的 CVGNN 算法是完全基于复值矩阵求逆方程构造的。首先，定义损失函数为误差函数的 Frobenius 范数：令  $\Sigma = \mathbf{X}\mathbf{A}\mathbf{A}^H - \mathbf{A}^H$ ， $\text{tr}$  表示矩阵的迹，则

$$\varepsilon(\mathbf{X}) = \frac{1}{2} \|\Sigma\|^2 = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^m |\varepsilon_{ij}|^2 = \frac{1}{2} \text{tr}(\Sigma \Sigma^H) = \frac{1}{2} \Sigma^* : \Sigma \quad (3.17)$$

接着，根据梯度神经动力学方法，我们可以设计一个复值梯度算法，使函数沿下降速率最快的方向（即梯度方向）发展，直至达到该损失函数的最小值为止，即求解最优化问题  $\forall \mathbf{X} \in \mathbb{C}^{N \times M}, \arg \min \varepsilon(\mathbf{X})$ 。

### 3.3.1 损失函数的梯度

$\varepsilon(\mathbf{X})$  对  $\mathbf{X} \in \mathbb{C}^{N \times M}$  的偏导数可表示为:

$$\frac{\partial}{\partial \mathbf{X}} \varepsilon = \begin{bmatrix} \frac{\partial}{\partial x_{0,0}} \varepsilon & \cdots & \frac{\partial}{\partial x_{0,M-1}} \varepsilon \\ \vdots & & \vdots \\ \frac{\partial}{\partial x_{N-1,0}} \varepsilon & \cdots & \frac{\partial}{\partial x_{N-1,M-1}} \varepsilon \end{bmatrix}, \frac{\partial}{\partial \mathbf{X}^*} \varepsilon = \begin{bmatrix} \frac{\partial}{\partial x_{0,0}^*} \varepsilon & \cdots & \frac{\partial}{\partial x_{0,M-1}^*} \varepsilon \\ \vdots & & \vdots \\ \frac{\partial}{\partial x_{N-1,0}^*} \varepsilon & \cdots & \frac{\partial}{\partial x_{N-1,M-1}^*} \varepsilon \end{bmatrix} \quad (3.18)$$

$\frac{\partial}{\partial \mathbf{X}} \varepsilon$  和  $\frac{\partial}{\partial \mathbf{X}^*} \varepsilon$  分别被称为  $\varepsilon$  关于  $\mathbf{X}$  和  $\mathbf{X}^*$  的梯度<sup>[28]</sup>,  $\mathbf{X}^*$  表示  $\mathbf{X}$  的复共轭,  $\mathbf{X}^T$  表示  $\mathbf{X}$  的转置,  $\frac{\partial}{\partial \mathbf{X}^*} \varepsilon$  可用符号表示为:

$$\nabla_{\mathbf{X}} \varepsilon \triangleq \frac{\partial}{\partial \mathbf{X}^*} \varepsilon \quad (3.19)$$

表 3.1 函数  $f(\mathbf{Z}, \mathbf{Z}^*)$  的导数

$f(\mathbf{Z}, \mathbf{Z}^*)$	Differential $df$	$\frac{\partial}{\partial \mathbf{Z}} f$	$\frac{\partial}{\partial \mathbf{Z}^*} f$
$\text{Tr}\{\mathbf{Z}\}$	$\text{Tr}\{\mathbf{I}_N d\mathbf{Z}\}$	$\mathbf{I}_N$	$\mathbf{0}_{N \times N}$
$\text{Tr}\{\mathbf{Z}^*\}$	$\text{Tr}\{\mathbf{I}_N d\mathbf{Z}^*\}$	$\mathbf{0}_{N \times N}$	$\mathbf{I}_N$
$\text{Tr}\{\mathbf{A}\mathbf{Z}\}$	$\text{Tr}\{\mathbf{A} d\mathbf{Z}\}$	$\mathbf{A}^T$	$\mathbf{0}_{N \times M}$
$\text{Tr}\{\mathbf{Z}^H \mathbf{A}\}$	$\text{Tr}\{\mathbf{A}^T d\mathbf{Z}^*\}$	$\mathbf{0}_{N \times M}$	$\mathbf{A}$
$\text{Tr}\{\mathbf{Z} \mathbf{A}_0 \mathbf{Z}^H \mathbf{A}_1\}$	$\text{Tr}\{\mathbf{A}_0 \mathbf{Z}^H \mathbf{A}_1 d\mathbf{Z} + \mathbf{A}_0^T \mathbf{Z}^T \mathbf{A}_1^T d\mathbf{Z}^*\}$	$\mathbf{A}_1^T \mathbf{Z}^* \mathbf{A}_0^T$	$\mathbf{A}_1 \mathbf{Z} \mathbf{A}_0$

根据表 3.1 的求导公式和链式法则, 损失函数的梯度计算为:

$$\begin{aligned} \nabla \varepsilon(\mathbf{X}) &= \frac{\partial \varepsilon(\mathbf{X})}{\partial \mathbf{X}^*} = \frac{\partial \varepsilon(\mathbf{X})}{\partial \Sigma^*} \cdot \frac{\partial \Sigma^*}{\partial \mathbf{X}^*} \\ &= \frac{1}{2} \Sigma \frac{\partial (\mathbf{X} \mathbf{A} \mathbf{A}^H - \mathbf{A}^H)^*}{\partial \mathbf{X}^*} \\ &= \frac{1}{2} (\mathbf{X} \mathbf{A} \mathbf{A}^H - \mathbf{A}^H) \mathbf{A} \mathbf{A}^H \end{aligned} \quad (3.20)$$

### 3.3.2 CVGNN 算法及收敛性证明

用  $\Upsilon(\bullet)$  表示神经网络的矩阵映射, 我们可以得到梯度神经网络的动力学方程:

$$\dot{\mathbf{X}}(t) = -\mu \Upsilon(\nabla \varepsilon(\mathbf{X})) \quad (3.21)$$

其中,  $\mu > 0$  是需要人为设定的超参数, 表示步长, 可以调整神经网络的收敛速度,

在硬件允许范围内可设置为尽可能的大值。 $\Upsilon(\bullet)$ 中的激活函数可为神经网络中通常的单调增奇函数:如线性函数  $f(u)=u$ ; 双极性 sigmoid 函数  $f(u)=(1-\exp(-\xi u))/(1+\exp(-\xi u))$ ,  $\xi \geq 1$ ; 功率函数  $f(u)=u^p, p \geq 3$ 。本文为方便起见, 使用线性函数  $f(u)=u$ 。

根据 3.3.1 节求得的梯度信息(3.20)和(3.21), 我们可以构建如下的 CVGNN 模型:

$$\dot{\mathbf{X}}(t) = -\frac{1}{2}\mu(\mathbf{X}\mathbf{A}\mathbf{A}^H - \mathbf{A}^H)\mathbf{A}\mathbf{A}^H \quad (3.22)$$

众所周知, 收敛速度对于成功应用神经网络模型至关重要, 所以我们应该研究该模型的收敛性。

**引理 1:** 考虑(3.22)的 CVGNN 模型,  $\mathbf{X}(t)$ 从随机生成的  $\mathbf{X}(0) \in \mathbb{C}^{N \times M}$  开始, 可以指数收敛于理论逆  $\mathbf{X}^{opt}$ , 并且收敛速度是  $\mu/2$  和  $\mathbf{A}\mathbf{A}^H\mathbf{A}\mathbf{A}^H$  的最小特征值  $\alpha$  的乘积。

**证明:** 为方便起见, 令  $\mathbf{G} = \mathbf{A}\mathbf{A}^H$ , 定义  $\tilde{\mathbf{X}}(t) = \mathbf{X}(t) - \mathbf{X}^{opt}$ ,  $\mathbf{X}^{opt}$  是  $\mathbf{A}$  理论的广义逆,  $\mathbf{X}(t)$  由梯度神经网络产生。由此可得:

$$\dot{\tilde{\mathbf{X}}}(t) = \dot{\mathbf{X}}(t), \mathbf{X}(t) = \tilde{\mathbf{X}}(t) + \mathbf{X}^{opt} \quad (3.23)$$

因为  $\mathbf{X}^{opt}\mathbf{A}\mathbf{A}^H - \mathbf{A}^H = 0$ , 将(3.23)代入(3.22)可得:

$$\dot{\tilde{\mathbf{X}}}(t) = -\frac{1}{2}\mu\tilde{\mathbf{X}}(t)\mathbf{G}\mathbf{G}^H \quad (3.24)$$

我们可以选择一个 Lyapunov 函数  $v(t)$  如下:

$$v(t) = \frac{1}{2}\|\tilde{\mathbf{X}}(t)\mathbf{G}\|_{\text{F}}^2 = \frac{1}{2}\text{tr}\left((\tilde{\mathbf{X}}(t)\mathbf{G})(\tilde{\mathbf{X}}(t)\mathbf{G})^H\right) \geq 0 \quad (3.25)$$

所以, Lyapunov 函数是正定的, 当且仅当  $\tilde{\mathbf{X}}(t)=0$  时,  $v(t)=0$ 。另一方面, 可以进一步求出  $v(t)$  的导数:

$$\begin{aligned} \dot{v}(t) &= \frac{dv}{dt} = \text{tr}\left(\left(\dot{\tilde{\mathbf{X}}}(t)\mathbf{G}\right)(\tilde{\mathbf{X}}(t)\mathbf{G})^H\right) \\ &= -\frac{1}{2}\mu\text{tr}\left(\tilde{\mathbf{X}}(t)\mathbf{G}\mathbf{G}^H\mathbf{G}\mathbf{G}^H(\tilde{\mathbf{X}}(t))^H\right) \\ &= -\frac{1}{2}\mu\|\tilde{\mathbf{X}}(t)\mathbf{G}\mathbf{G}^H\|_{\text{F}}^2 \leq 0 \end{aligned} \quad (3.26)$$

所以,  $\dot{v}(t)$  是负定的, 当且仅当  $\tilde{\mathbf{X}}(t)=0$  时,  $\dot{v}(t)=0$ 。根据 Lyapunov 函数理论可知,

$\tilde{\mathbf{X}}(t)$  全局收敛于 0，即  $\mathbf{X}(t)$  全局收敛于理论解  $\mathbf{X}^{opt}$ 。

下面考虑指数收敛速度，根据(3.26)可得：

$$\begin{aligned}
 \dot{v}(t) &= -\frac{1}{2}\mu \|\tilde{\mathbf{X}}(t)\mathbf{G}\mathbf{G}^H\|_{\text{F}}^2 \\
 &= -\frac{1}{2}\mu \text{tr}\left(\tilde{\mathbf{X}}(t)\mathbf{G}\mathbf{G}^H\mathbf{G}\mathbf{G}^H\left(\tilde{\mathbf{X}}(t)\right)^H\right) \\
 &= -\frac{1}{2}\mu \text{tr}\left(\tilde{\mathbf{X}}(t)\mathbf{G}\mathbf{G}^H\mathbf{G}\left(\tilde{\mathbf{X}}(t)\mathbf{G}\right)^H\right) \\
 &\leq -\frac{1}{2}\alpha\mu \text{tr}\left(\tilde{\mathbf{X}}(t)\mathbf{G}\left(\tilde{\mathbf{X}}(t)\mathbf{G}\right)^H\right) \\
 &= -\frac{1}{2}\alpha\mu \|\tilde{\mathbf{X}}(t)\mathbf{G}\|_{\text{F}}^2
 \end{aligned} \tag{3.27}$$

$\alpha > 0$  是  $\mathbf{G}^H\mathbf{G} = \mathbf{A}\mathbf{A}^H\mathbf{A}\mathbf{A}^H$  的最小特征值， $v(t)$  可以由以下得出：

$$v(t) \leq v(0)\exp(-\alpha\mu t) \tag{3.28}$$

再者， $v(t) = \frac{1}{2}\text{tr}\left(\left(\tilde{\mathbf{X}}(t)\mathbf{G}\right)\left(\tilde{\mathbf{X}}(t)\mathbf{G}\right)^H\right) \geq \frac{1}{2}\alpha \text{tr}\left(\tilde{\mathbf{X}}(t)^H\tilde{\mathbf{X}}(t)\right) = \frac{1}{2}\alpha \|\tilde{\mathbf{X}}(t)\mathbf{G}\|_{\text{F}}^2$ ，且

$$\begin{aligned}
 v(0) &= \frac{1}{2}\|\tilde{\mathbf{X}}(0)\mathbf{G}\|_{\text{F}}^2 \leq \frac{1}{2}\|\tilde{\mathbf{X}}(0)\|_{\text{F}}^2\|\mathbf{G}\|_{\text{F}}^2, \\
 \frac{1}{2}\alpha \|\tilde{\mathbf{X}}(t)\|_{\text{F}}^2 &\leq v(t) \leq v(0)\exp(-\alpha\mu t)
 \end{aligned} \tag{3.29}$$

所以，

$$\|\mathbf{X}(t) - \mathbf{X}^{opt}\|_{\text{F}} \leq \frac{\|\tilde{\mathbf{X}}(0)\|_{\text{F}}\|\mathbf{G}\|_{\text{F}}}{\sqrt{\alpha/2}}\exp(-\alpha\mu t) \tag{3.30}$$

所以，当  $t \rightarrow \infty$  时， $\mathbf{X}(t)$  以指数速率  $\alpha\mu$  趋近于  $\mathbf{X}^{opt}$ ，即该 CVGNN 模型可使  $\mathbf{X}(t)$  指数收敛于理论解。证明完毕。

### 3.4 本章小结

本章从矩阵的广义逆矩阵出发，首先给出了矩阵的广义逆矩阵的定义及性质。然后，分析了几种常见的基于数值求解矩阵广义逆的方法，包括 QR 分解、SVD 算法。接着，提出了 CVGNN 算法求解复值矩阵的广义逆，并证明了该网络的收敛性，给出了收敛速度。理论证明可得 CVGNN 算法可以以指数速率收敛于理论解。



## 第四章 基于 MATLAB 的矩阵求逆算法仿真

第三章提出了 CVGNN 矩阵求逆算法，用于在线计算复值矩阵的广义逆矩阵，并给出了收敛性的理论证明。本章将通过基于 MATLAB 的仿真实验进一步说明 CVGNN 矩阵求逆算法的效率。

### 4.1 给定矩阵的求逆仿真

我们选择的待求逆的复值矩阵如下：

$$\mathbf{A} = \begin{bmatrix} \exp(j10) & -\exp(-j10) \\ -\exp(j10) & 0 \end{bmatrix}。$$

该矩阵求逆的理论解为：

$$\mathbf{X}^{opt} = \begin{bmatrix} 0 & 0.8391 - j0.5440 \\ 0.8391 + j0.5440 & 0.8390 + j0.5440 \end{bmatrix}。$$

首先，从任意初始状态  $\mathbf{X}(0) \in \mathbb{C}^{2 \times 2}$  开始，步长  $\mu = 0.01$ ，使用 CVGNN 模型计算上述复数矩阵的广义逆，图 4-1 显示了 CVGNN 模型矩阵求逆时各元素的瞬时值变化趋势，real axis 表示矩阵对应元素的实部，imaginary axis 表示矩阵对应元素的虚部，z 轴表示时间。

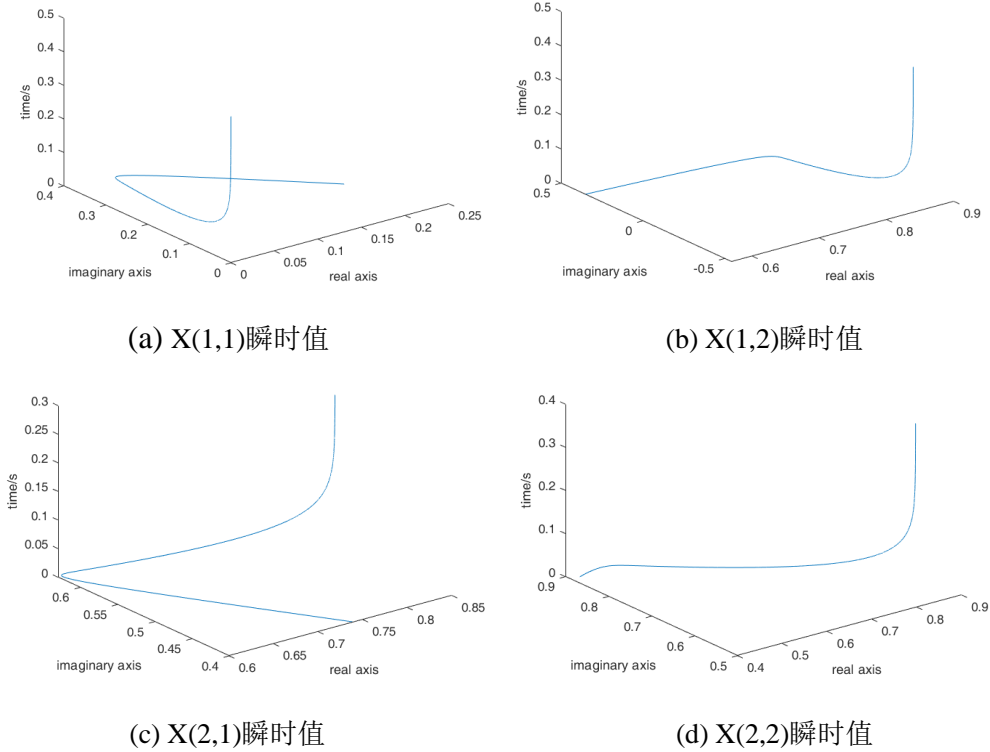


图 4-1  $\mu = 0.01$  时  $\mathbf{X}(t)$  各元素的瞬时值

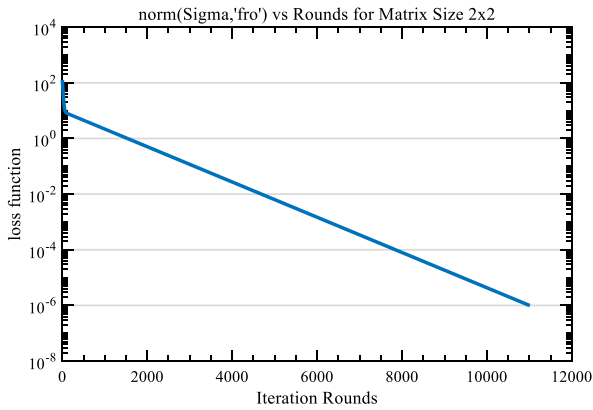
通过 MATLAB 函数求出的理论解为：

```
result_the =  
  
-0.0000 - 0.0000i    0.8391 - 0.5440i  
0.8391 + 0.5440i    0.8391 + 0.5440i
```

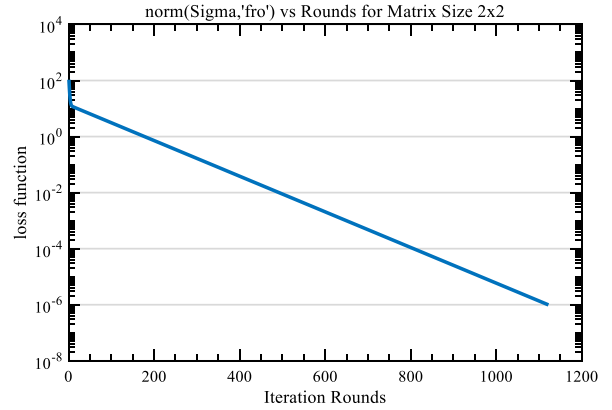
CVGNN 模型最终收敛的逆矩阵值为：

```
X =  
  
0.0000 + 0.0000i    0.8391 - 0.5440i  
0.8391 + 0.5440i    0.8391 + 0.5440i
```

由 CVGNN 生成的复值神经状态矩阵  $\mathbf{X}(t) \in \mathbb{C}^{2 \times 2}$  的各个元素在短时间内收敛为一条直线，并且与理论解  $\mathbf{X}^{opt}$  相比，这些值是矩阵  $\mathbf{A}$  求逆的理论解对应的各元素值。该结果表明，本文提出的 CVGNN 模型是有效的。



(a)  $\mu=0.01$ 时 CVGNN 的收敛性



(b)  $\mu=0.1$ 时 CVGNN 的收敛性

图 4-2 不同步长下 CVGNN 算法损失函数的收敛性 损失函数随更新次数以指数速率收敛，损失函数表示  $\mathbf{X}$  的当前值与理论解的误差，这里 x 轴为网络更新次数，y 轴为损失函数的对数坐标，在对数坐标系下图像呈直线，表示图像实际以指数速率下降。

为了更直观的说明 CVGNN 的效率，在相同条件下，我们得到了损失函数（剩余误差）的收敛性。图 4-2 显示了损失函数随网络更新次数的收敛性，为了更直观的表现指数收敛速度，x 轴表示网络更新次数，y 轴表示损失函数。CVGNN 模型的损失函数  $\varepsilon(\mathbf{X}) = \frac{1}{2} \|\mathbf{X}\mathbf{A}\mathbf{A}^H - \mathbf{A}^H\|^2$  随网络的更新呈指数速率下降，这与理论证明的指数收敛速度相吻合。另外，当  $\mu$  的值变大时，CVGNN 模型的收敛速度更快。当  $\mu$  从 0.01 增加到 0.1 时，

CVGNN 的收敛速度也提高了约 10 倍，所以适当的增加 CVGNN 模型的步长可以加快网络的收敛，更快的求解矩阵的广义逆矩阵。

## 4.2 随机矩阵的求逆仿真

4.1 节给出的是在给定复值矩阵下 CVGNN 的仿真，从实验层面证明了 CVGNN 算法求解复值矩阵的广义逆矩阵的有效性。但是，在实际通信系统中，信道矩阵不一定是方阵（例如，天线数  $M$  与用户数  $N$  不等时），因而本节将对不同大小的随机矩阵求逆进行仿真，分析 CVGNN 算法在一般性条件下求广义逆矩阵的性能。

在 MATLAB 中随机生成  $2 \times 3$  的复值矩阵，

```
A =
    0.8391 + 0.8264i    0.1319 + 0.0420i    0.5389 + 0.7690i
    0.1542 + 0.3797i    0.1628 + 0.7122i    0.5218 + 0.6336i
```

通过 MATLAB 函数求出的理论解为：

```
result_the =
    0.5827 - 0.4592i   -0.3500 + 0.2382i
   -0.1195 + 0.5360i    0.1430 - 1.0291i
   -0.0374 - 0.2470i    0.4530 - 0.2398i
```

通过 CVGNN 算法求出的广义逆矩阵为：

```
X =
    0.5827 - 0.4592i   -0.3500 + 0.2382i
   -0.1195 + 0.5360i    0.1430 - 1.0291i
   -0.0374 - 0.2470i    0.4530 - 0.2398i
```

比较可知，CVGNN 算法求解的广义逆矩阵与理论解一致。对于更大规模的矩阵求解的具体值，为了观察方便，我们用颜色分布表示矩阵实部和虚部值的大小。图 4-3 到 4-5 显示了 CVGNN 求解不同规模的随机复值矩阵各元素值的收敛情况，第一行红色的子图表示矩阵各元素值的实部，第二行蓝色的子图表示矩阵各元素值的虚部，最右侧的两个子图是待求矩阵广义逆的理论解，最后一行的子图表示 CVGNN 求解的  $X$  瞬时值与理论解的误差大小。实验过程中，误差每降低 10 的平方个量级输出一个状态矩阵以观察整个网络的收敛情况。

由图可知，初始矩阵实部和虚部的值逐渐接近图像最右侧显示的理论值，并逐渐将误差降低至 $10^{-6}$ 以下，实际上最后几列的误差已经非常小，最终输出的逆矩阵各元素值与理论解一致。该结果表明，本文提出的 CVGNN 算法能够求解随机复值矩阵的广义逆。

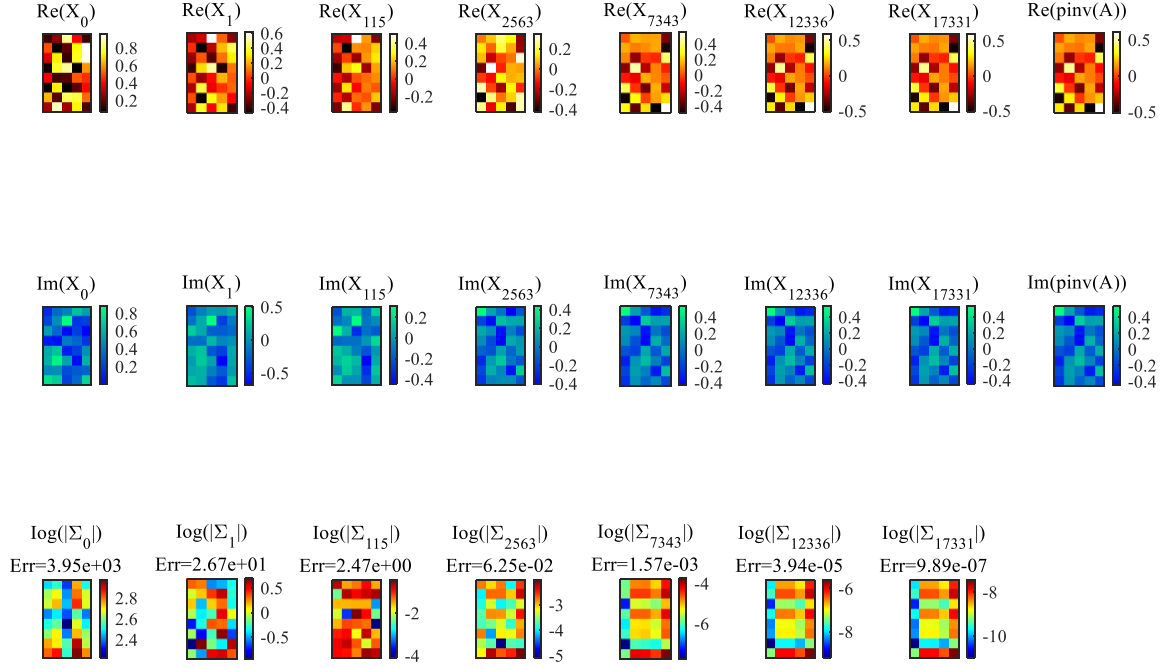


图 4-3 5\*8 的广义逆矩阵  $X$  随 CVGNN 更新时各元素的收敛情况。子图分别表示了待求广义逆矩阵  $X$  的状态矩阵随网络不断更新时各元素值的分布。其中第一行表示  $X$  实部的分布，第二行表示  $X$  虚部的分布，第三行表示误差矩阵  $\Sigma$  的分布。其中 1~7 列分别是  $X$  的初始随机值到求解精度不断提高的状态记录，精度每提高两个数量级输出一次。这里  $\text{Re}(X_i)$  表示第  $i$  次更新后状态矩阵  $X_i$  各元素值的实部， $\text{Im}(X_i)$  表示第  $i$  次更新后状态矩阵  $X_i$  各元素值的虚部， $\text{Err}$  表示误差，用损失函数的对数值表示；最后一列  $\text{pinv}(A)$  为对应的理论解。注意，虽然每一种颜色都对应了一个实值，但是不同子图 colorbar 对应的实值并不一定相同。

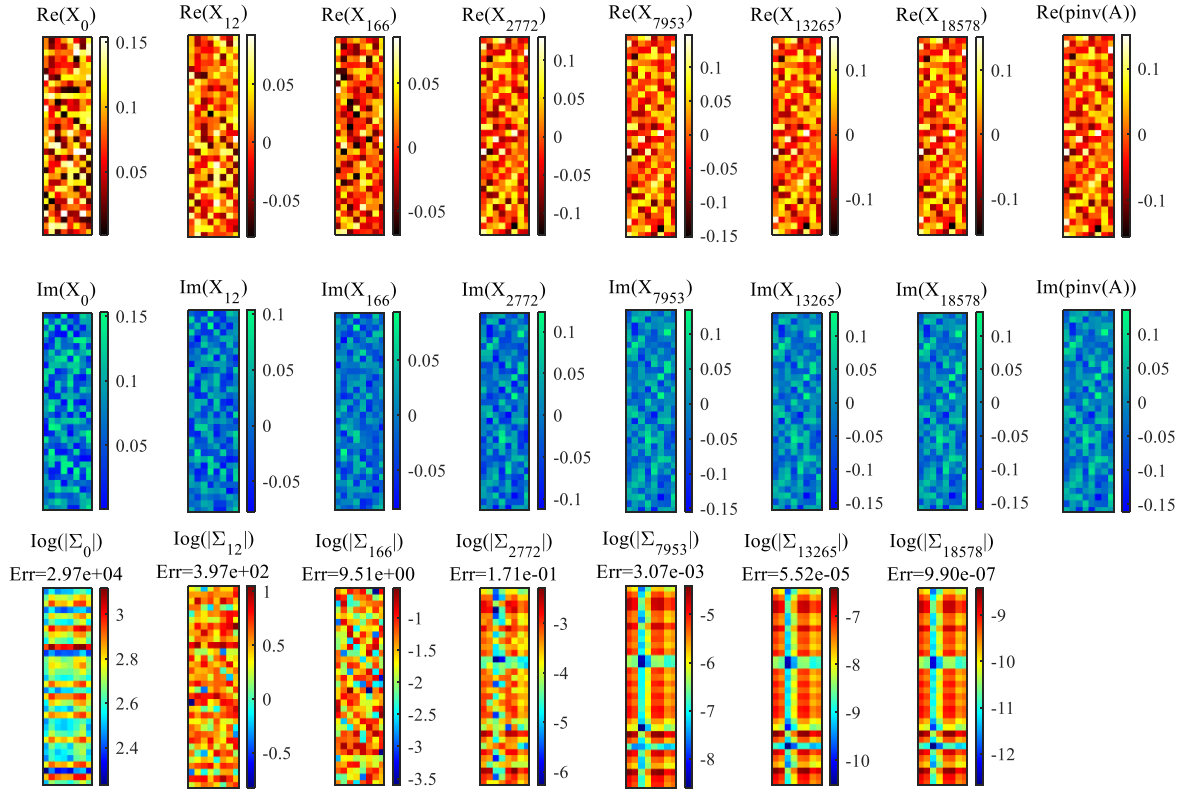


图 4-4 8\*32 的广义逆矩阵 X 随 CVGNN 更新时各元素的收敛情况

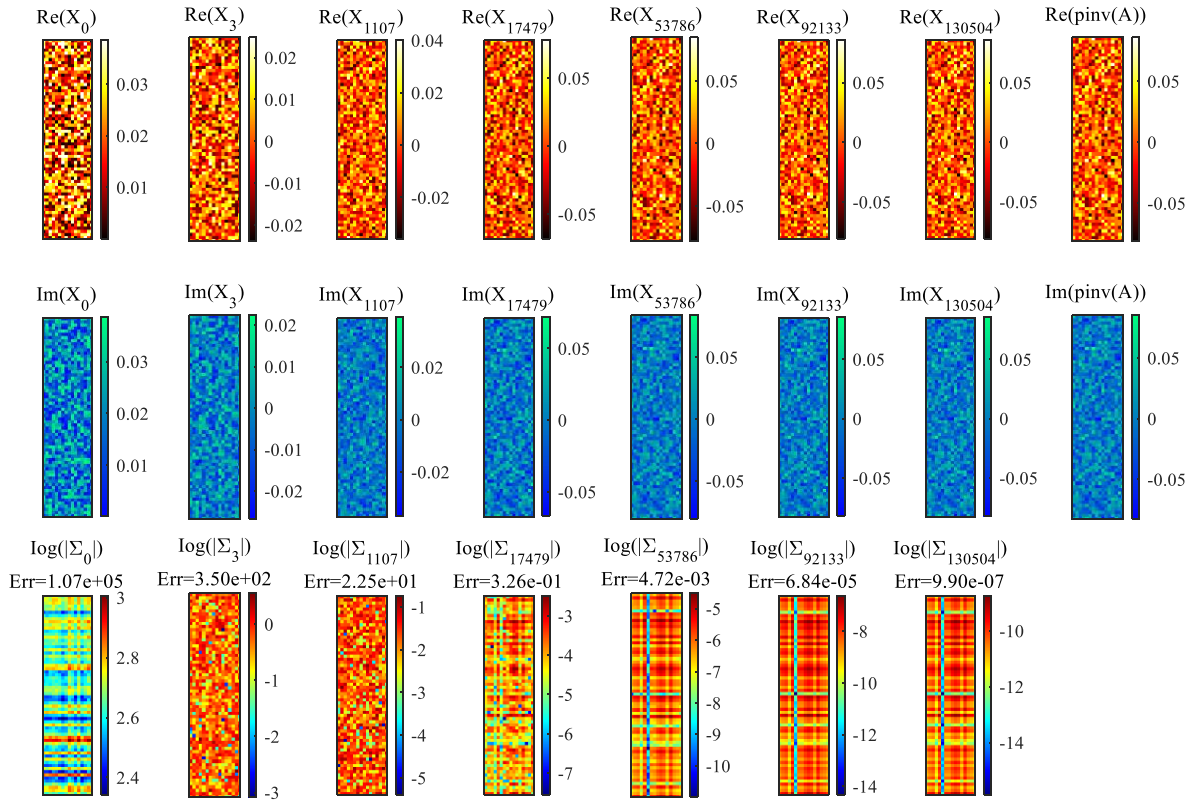


图 4-5 16\*64 的广义逆矩阵 X 随 CVGNN 更新时各元素的收敛情况

同样，我们需要分析 CVGNN 在求解随机复值矩阵时损失函数的收敛性和收敛速度。图 4-6 显示了 CVGNN 求解不同规模的随机复值矩阵时损失函数（剩余误差）的收敛性。随着矩阵规模的增加，CVGNN 的收敛时间增加，收敛速度减小，尤其当矩阵大小超过  $24 \times 256$  时，需要较长时间才能收敛。

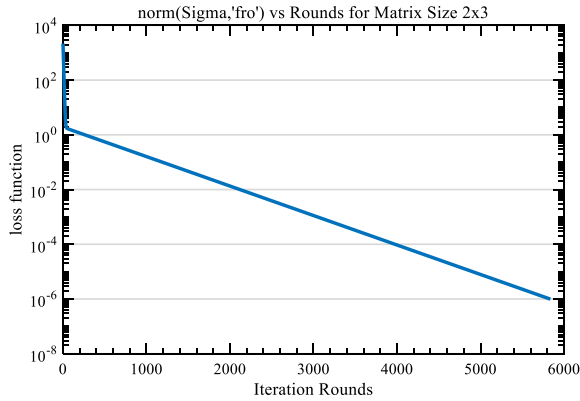
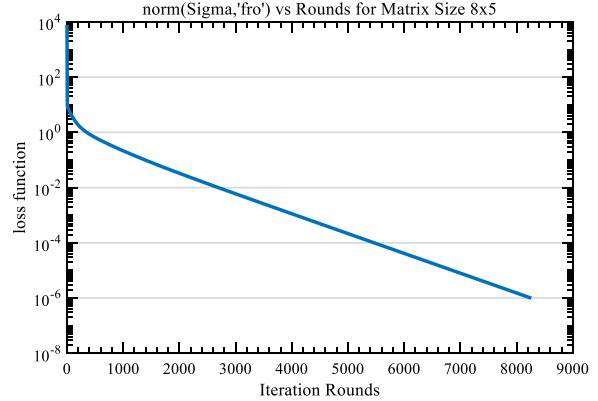
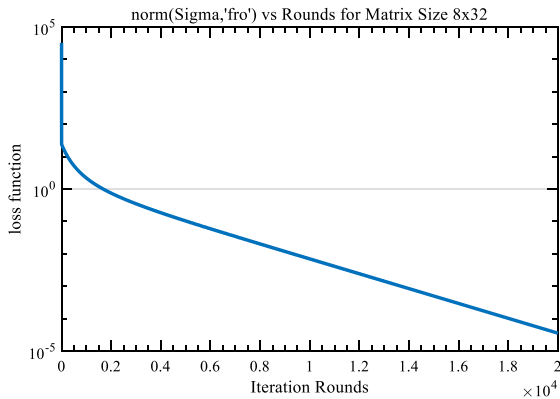
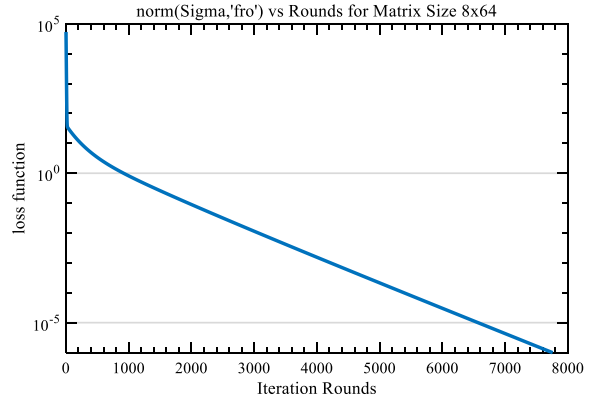
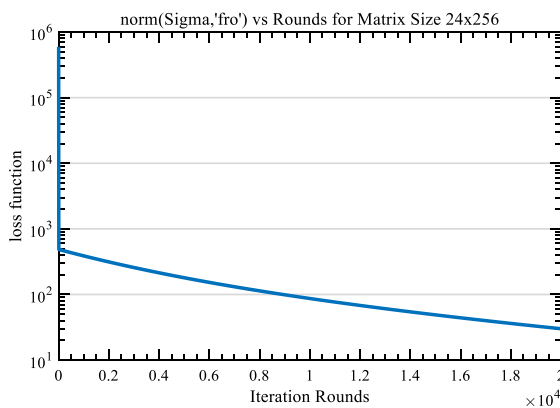
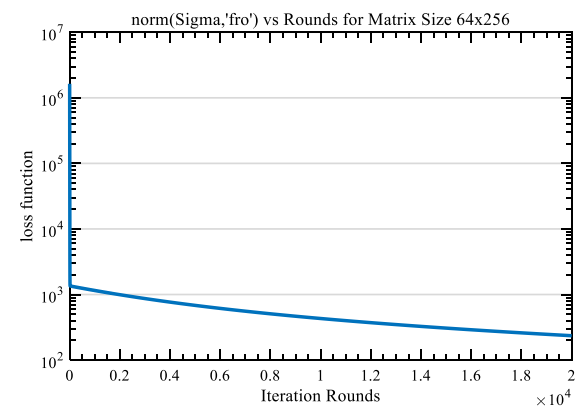
(a)  $2 \times 3$  矩阵(b)  $8 \times 5$  矩阵(c)  $8 \times 32$  矩阵(d)  $8 \times 64$  矩阵(e)  $24 \times 256$  矩阵(f)  $64 \times 256$  矩阵

图 4-6 随机生成矩阵下 CVGNN 算法损失函数的收敛性

因而，对于随机生成的小规模复值矩阵，CVGNN 仍然能较快求解理论解，但是对于完全随机的大规模复值矩阵，CVGNN 的收敛性不佳。根据 3.2 节的理论推导，CVGNN 模型从随机产生的初始矩阵开始，可以以指数速度收敛于理论逆，理论收敛速度是  $\mu/2$  和  $\mathbf{AA}^H\mathbf{AA}^H$  的最小特征值  $\alpha$  的乘积。造成 CVGNN 算法在大规模随机矩阵实验时性能不佳的原因可能是大规模随机矩阵  $\mathbf{AA}^H\mathbf{AA}^H$  的最小特征值  $\alpha$  有时较小，导致 CVGNN 收敛速度较慢。

根据 4.1 节可知，适当增加 CVGNN 的步长可以加快网络的收敛，但是，当步长过大时，可能导致错过最优解。我们对不同规模的随机矩阵进行仿真，得到了不同规模矩阵根据实验选取的较优参数  $\mu$ ，如表 4.1 所示，需要通过经验将  $\mu$  设定为一个合适的值。在后续研究中，更好的方法是利用机器学习算法对  $\mu$  进行训练，通过大量 CVGNN 模型求解矩阵广义逆的样本，训练出不同规模的矩阵下最优的  $\mu$  值，这样在实际应用的过程中，系统可以自适应的调节  $\mu$  值以达到最优性能。

表 4.1 不同规模的随机矩阵参数  $\mu$  的选择

矩阵维度	步长 $\mu$
2*3	0.1
5*8	0.01
8*32	0.0002
8*64	0.0001
24*256	0.0000001
64*256	0.00000004

另一方面，系统的计算开销和计算能力也是性能的一个重要指标。本文的实验均在安装了 Windows 10 的笔记本上运行，表 4.2 显示了进行仿真实验的硬件平台参数。表 4.3 显示了不同规模的矩阵变量在运行中占用的字节数。随着矩阵规模的增加，CVGNN 模型中

矩阵变量所占用的字节数也相应增大。

表 4.2 仿真实验硬件参数

硬件参数名	参数值
机型	惠普 G66pro G1
系统	Windows 10
处理器	Intel Corei7-8550U @ 1.80GHz 1.99 GHz
内存	16GB
GPU 内存	2GB
仿真平台版本	MATLAB 2021a

表 4.3 不同规模的矩阵变量占用的字节数

矩阵大小 \ 变量名	A/Byte	X/Byte	其他变量和/Byte	总计/KB
2*3	96	96	16072	15.88
5*8	640	640	40072	40.38
8*32	4096	4096	48072	54.95
8*64	8192	8192	56072	70.77
24*256	98304	98304	160072	348.32
64*256	163840	163840	240072	554.45

### 4.3 CVGNN 在大规模 MIMO 系统的矩阵求逆

4.2 节中分析了在随机生成待求矩阵下 CVGNN 的性能，虽然随机条件具有较好的一般性，但是随机生成的矩阵也存在一些隐患，如：近奇异矩阵、病态矩阵等，会导致模型在一定情况下收敛时间过长。由于本文主要探究的是该模型能否应用于大规模 MIMO 系统的矩阵求逆，所以下面将对无线通信信道下的矩阵应用 CVGNN 算法进行求逆仿真。

在无线通信系统中，信道矩阵的元素分布通常存在规律性，通常无线信道进行系统仿真时会采用瑞利衰落信道。瑞利衰落信道模型是无线传输信道的一种统计模型，信号经过瑞利信道后幅度是随机的，并且幅度包络服从瑞利分布<sup>[31]</sup>。瑞利分布指随机复数的实部和



虚部独立同分布于 0 均值，同方差的正态分布（即高斯分布）时，该复数的模（即信号的幅度）服从瑞利分布<sup>[34]</sup>。

如果无线信道存在相当大量的散射，接收机端的冲激响应表现为大量统计独立的随机变量的叠加，根据中心极限定理<sup>[33]</sup>，这一无线信道的冲激响应为高斯过程。如果不存在直射信号，那么该高斯过程均值为 0，且相位在  $0 \sim 2\pi$  内均匀分布。

瑞利衰落模型适用于描述建筑物密集的城市信道，在曼哈顿的实验证明，当地的无线信道环境确实接近瑞利衰落<sup>[30]</sup>，大多数论文中的实验也是在瑞利信道条件下完成的。因此，在无线通信信道环境中，电磁波经过反射折射散射等多条路径传播到达接收机后，信号的强度服从瑞利分布，即实部和虚部服从均值为 0，方差为  $\sigma^2$  的独立高斯分布<sup>[29]</sup>。根据以上描述，瑞利衰落信道本身可以通过根据独立正态高斯变量产生一个复数的实部和虚部进行建模，图 4-7 显示了不同方差  $\sigma^2$  下瑞利分布的概率密度函数<sup>[34]</sup>。

MATLAB 中可用  $\text{randn}(m,n)$  可生成高斯分布的随机矩阵，或用  $\text{wgn}(m,n,p)$  函数可生成指定功率谱密度的高斯白噪声（即符合一定高斯分布的随机序列），其中  $m,n$  是矩阵大小， $p$  是功率谱密度。两种方法是等价的。

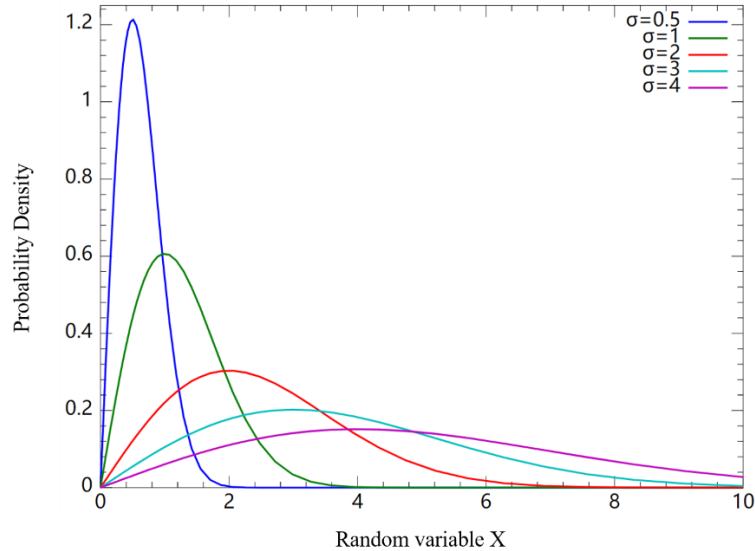


图 4-7 瑞利分布概率密度函数

由于瑞利信道矩阵服从均值为 0，方差为  $\sigma^2$  的独立高斯分布，所以取  $p=0$ ，在 MATLAB

中利用  $wgn(m, n, p)$  函数生成瑞利信道矩阵  $\mathbf{H}$  的公式如下：

$$\mathbf{H} = [wgn(m, n, 0) + j * wgn(m, n, 0)] / \text{sqrt}(2) \quad (4.1)$$

在瑞利信道的条件下对 CVGNN 求逆的性能进行实验,如图 4-8 所示,在大规模 MIMO 信道矩阵规模较大的情况下,该网络依然能够实现收敛。

事实上,瑞利信道下  $8 \times 32$  矩阵、 $16 \times 64$  矩阵、 $24 \times 256$  矩阵和  $64 \times 256$  矩阵收敛至理论解的平均时间与随机矩阵的实验相比有很大的提升。同时,根据仿真实验结果,瑞利信道下基站天线数  $M$  与用户数  $N$  的比值  $M/N$  越大,模型的收敛速度快,这与随机矩阵的实验结果有所不同。而实际应用中,大规模 MIMO 基站天线数  $M$  远大于其服务的用户数量  $N$ ,这也说明,CVGNN 模型在大规模 MIMO 系统中拥有比求解一般的随机复值矩阵更好的性能。

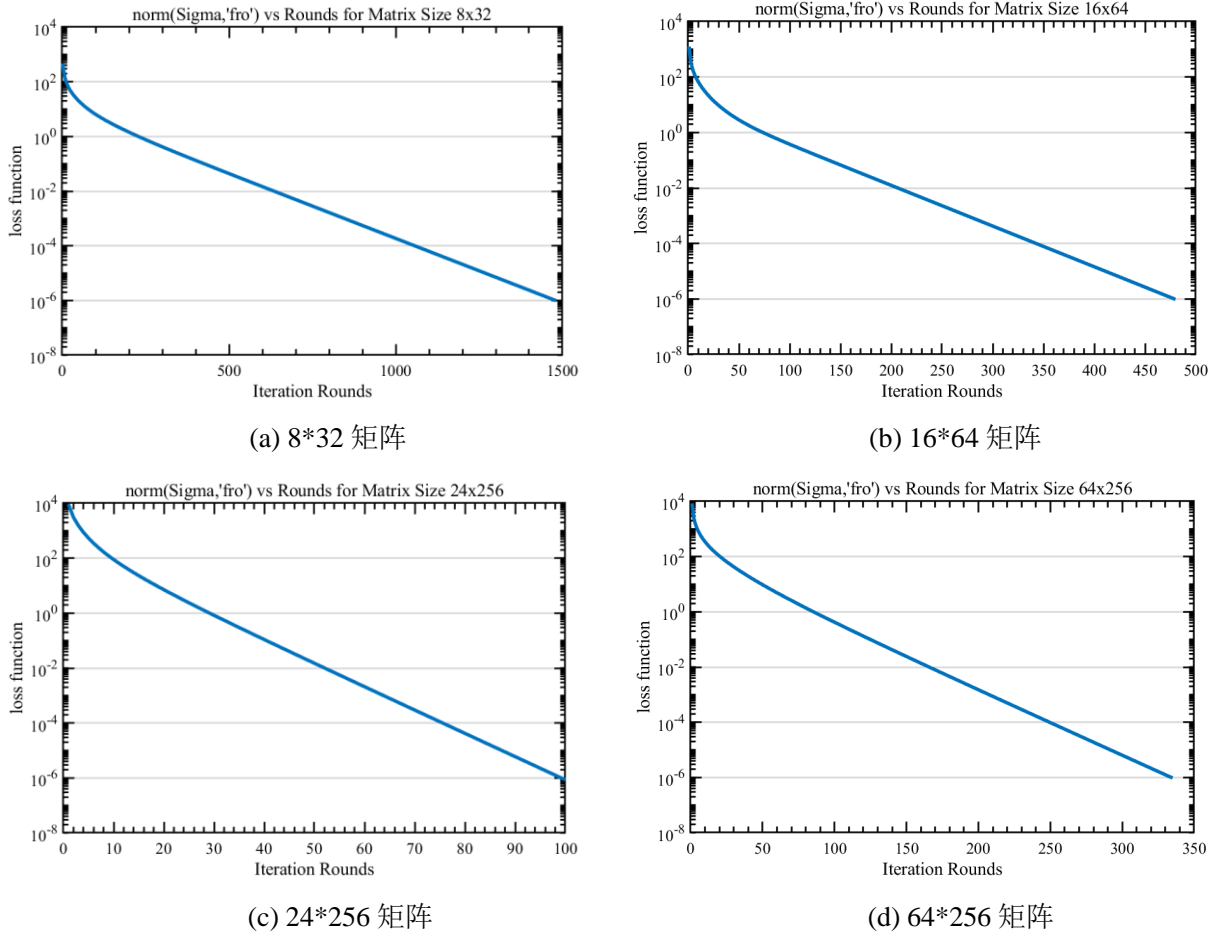


图 4-8 瑞利信道矩阵下 CVGNN 算法损失函数的收敛性

将 CVGNN 求解的结果与理论解将比较,用颜色分布表示矩阵实部和虚部值的大小。

图 4-9 到图 4-13 显示了 CVGNN 求解不同规模的瑞利信道时各元素值的收敛情况，第一行红色的图示表示矩阵各元素值实部的分布，第二行蓝色的图示表示矩阵各元素值虚部的分布，最右侧的两个图示是待求矩阵广义逆的理论解，最下方的图示表示 CVGNN 的瞬时值与理论解的误差大小。同样，为了更直观的观察矩阵整个网络的收敛情况，误差每降低 10 的平方个量级输出一个状态矩阵，直至收敛至理论解。

从图中可以看出，初始矩阵的实部和虚部的值逐渐接近图像最右侧显示的理论值，误差降低至 $10^{-6}$ 以下，对应的各元素值与理论解一致。而且，与 4.3 节随机复值矩阵的实验结果相比，不同规模的瑞利信道矩阵的各元素值都能更快的收敛至理论解。同时，瑞利信道矩阵  $M/N$  的比值越大，各元素的收敛速度越快，这与图 4-8 的实验结果相符。

所以，CVGNN 模型在大规模 MIMO 系统中有一定的应用前景，能够求解大规模 MIMO 系统预编码算法中矩阵的广义逆。

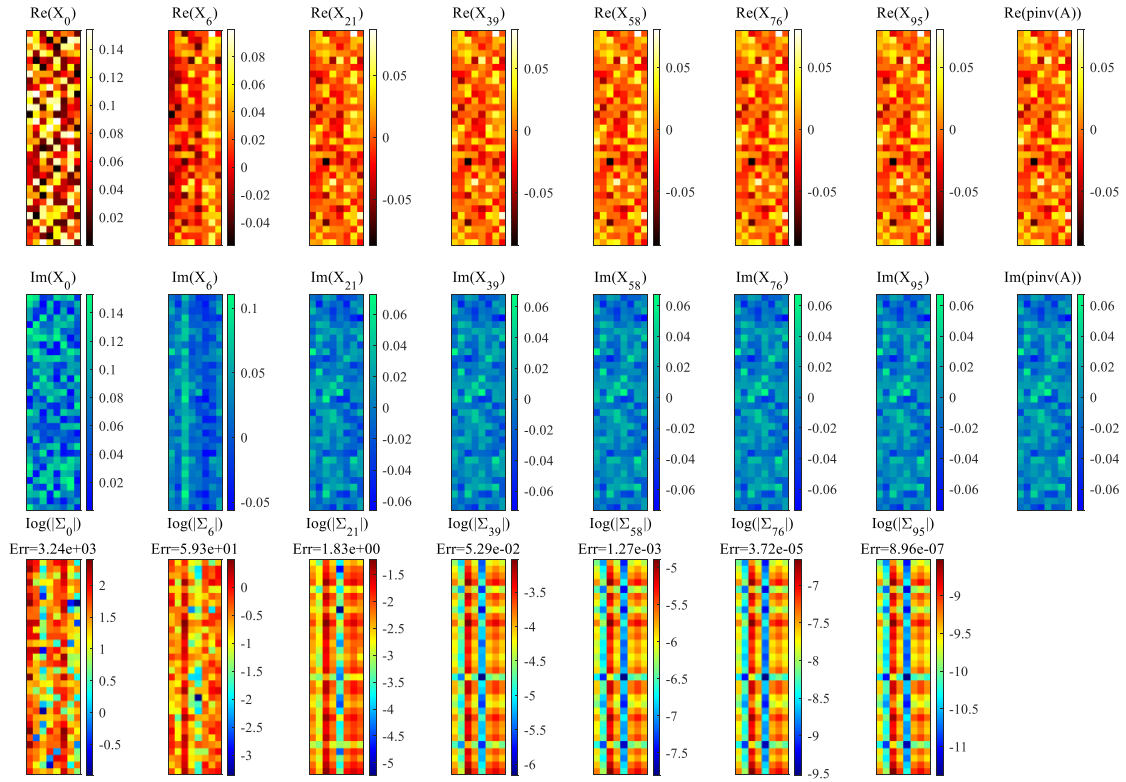


图 4-9 8\*32 的瑞利信道矩阵 X 随 CVGNN 更新时各元素的收敛情况

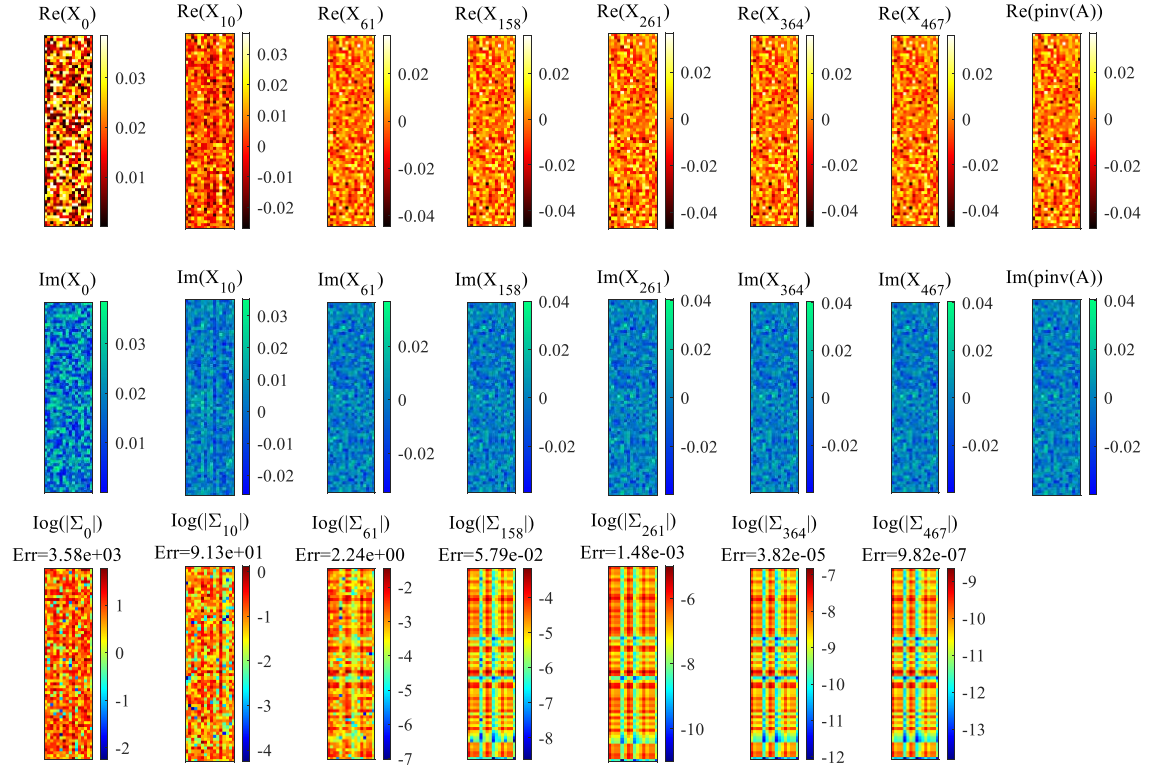


图 4-10 16\*64 的瑞利信道矩阵 X 随 CVGNN 更新时各元素的收敛情况

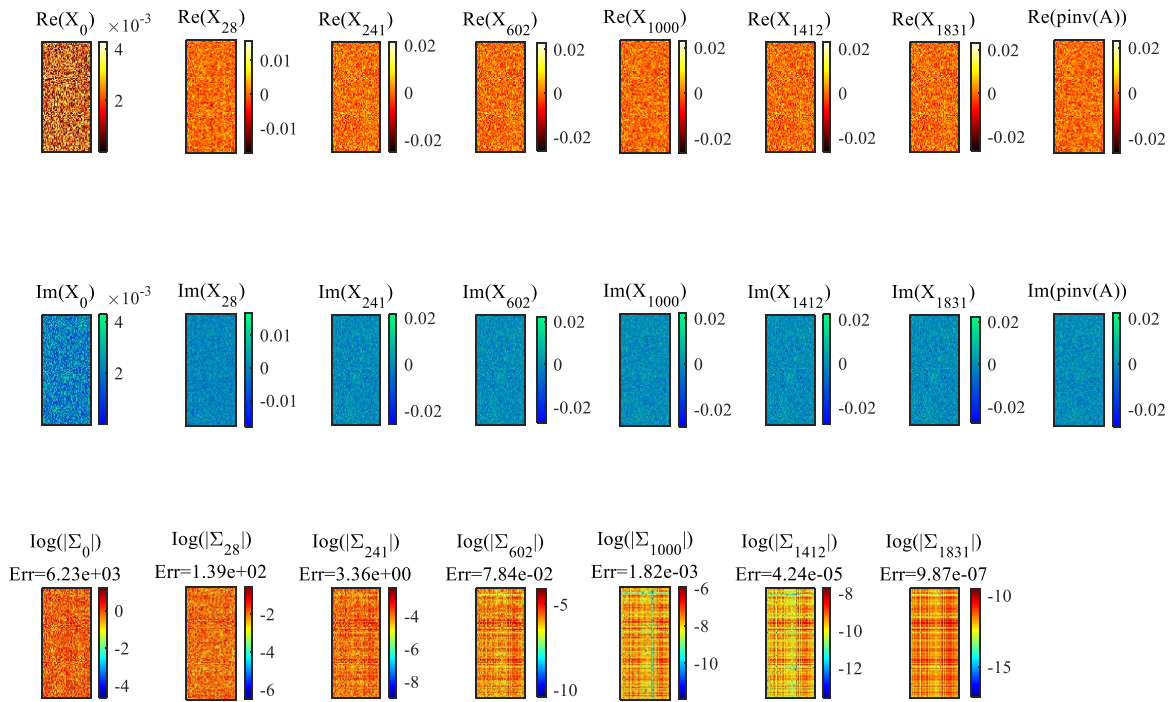


图 4-11 64\*144 的瑞利信道矩阵 X 随 CVGNN 更新时各元素的收敛情况

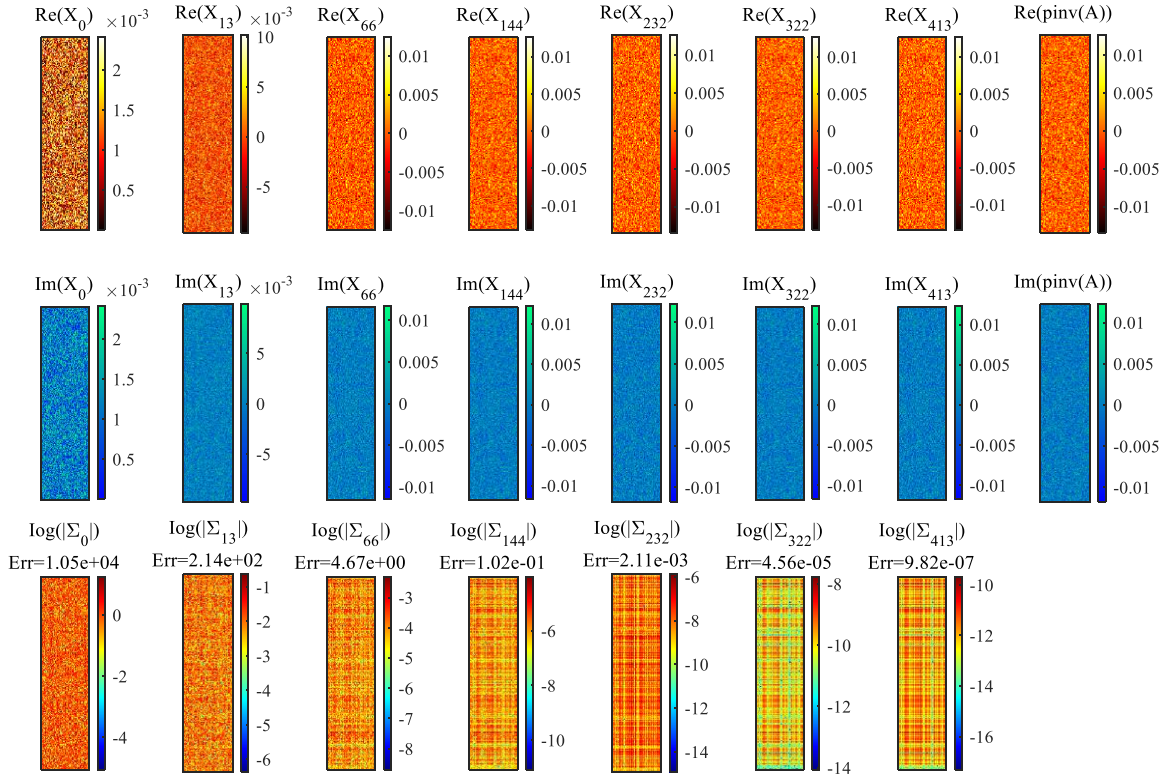


图 4-12 64\*256 的瑞利信道矩阵 X 随 CVGNN 更新时各元素的收敛情况

#### 4.4 本章小结

本章通过基于 MATLAB 的仿真实验分析了 CVGNN 矩阵求逆算法的性能。首先，从给定的复值矩阵出发，对矩阵的各元素瞬时值、收敛性、参数性能进行了分析，验证了 CVGNN 求解广义逆矩阵的有效性。然后，对不同规模的随机复值矩阵进行了仿真实验，分析了网络的收敛性和占用内存，进一步分析了 CVGNN 求解广义逆矩阵的性能，并指出了一些情况下可能出现收敛速度较慢、收敛时间较长的问题。最后，研究了大规模 MIMO 系统下，CVGNN 模型在求解瑞利信道矩阵广义逆的性能，与随机复值矩阵的实验结果相比，不同规模的瑞利信道矩阵的各元素值都能更快的收敛至理论解，体现了 CVGNN 模型在大规模 MIMO 系统的应用前景。

## 第五章 总结与展望

### 5.1 工作总结

本文对大规模 MIMO 系统中基于机器学习的矩阵优化算法进行了研究，提出了复值梯度神经网络矩阵求逆算法（CVGNN），以在复数域中找到矩阵的广义逆。本文所提出的方法不同于以往的研究，没有将复值矩阵转化为实值矩阵求逆，而是直接在复数域求解广义逆矩阵。理论分析和基于 MATLAB 的仿真表明，CVGNN 在求解复值矩阵广义逆，尤其应用于求解瑞利信道矩阵时具有较好的性能。本文具体工作如下：

1、学习并研究了矩阵运算的数学理论，对以往学者在矩阵求逆算法方面的研究进行了总结和梳理，比较了不同种方法的优劣。

2、学习并研究了大规模 MIMO 信道模型和预编码技术。讨论了大规模 MIMO 中的矩阵运算，并建立了大规模 MIMO 的下行信道模型，分析了几种常见的预编码算法及其计算复杂度。基本的预编码算法都面临高维矩阵求逆的高复杂度问题，表明了研究快速、低复杂度的矩阵求逆算法的必要性。

3、从数学层面给出了广义逆矩阵的定义和矩阵求逆的模型。分析了几种常见的基于数值的求逆方法，包括 QR 分解、SVD 算法。提出了复值梯度神经网络矩阵求逆算法（CVGNN），并从理论层面证明了该网络的收敛性：CVGNN 可以任意的初始矩阵出发，以指数速率收敛至广义逆矩阵的理论解。

4、研究了 CVGNN 的仿真性能，通过基于 MATLAB 的仿真实验说明了该算法的效率和在大规模 MIMO 系统中的应用前景。由 CVGNN 生成的复值神经状态矩阵可以以指数速率收敛，并可以通过调整网络的超参数对收敛速度进行控制。对不同规模的随机复值矩阵进行了仿真，分析了损失函数（剩余误差）的收敛性。在完全随机一般性条件下，分析了 CVGNN 的效率。

5、研究了 CVGNN 在求解大规模 MIMO 瑞利信道矩阵时的性能，展示了在瑞利分布下 CVGNN 对信道矩阵求广义逆的效率，体现了 CVGNN 模型在大规模 MIMO 的应用前景。

## 5.2 工作展望

本文主要研究了基于深度学习的矩阵优化算法在 6G 大规模 MIMO 系统中的应用，提出了复值梯度神经网络模型，在大规模 MIMO 系统中有应用前景。但是，该模型在一定条件下仍然存在局限性，为了成熟的实现矩阵优化算法在 6G 通信系统的应用，后续的工作仍然需要改进。

本文提出的 CVGNN 矩阵求逆算法虽然能求解瑞利信道矩阵的广义逆，但是算法过度阶段较长，精度受非线性激活函数设置和初始条件的影响较大。在后续的研究中，可考虑通过机器学习算法训练超参数步长的最优取值，通过大量 CVGNN 模型求解矩阵广义逆的样本，训练出不同规模的矩阵下最优的  $\mu$  值，这样在实际应用的过程中，系统可以自适应的调节  $\mu$  值以达到最优性能。另外 CVGNN 涉及输出的反馈，但是目前的通信系统前馈实现较为简单，而反馈实现的成本较为昂贵。所以，未来需要寻求构建更加适应矩阵运算需求的前馈神经网络模型（如：卷积神经网络，CNN）。CNN 已广泛应用于图像处理等问题，考虑到空域、时域或频域中的相关性，大规模 MIMO 系统的 CSI 矩阵通常可以被视为二维图像。因此，CNN 在大规模 MIMO 系统应用方面具有巨大的潜力。但是，基于卷积神经网络的时变矩阵求逆方法目前尚未存在成熟的研究，训练出实际可用的卷积神经网络还需时日。



## 参考文献

- [1] 卢安安,高西奇.大规模 MIMO 传输技术研究与展望[J].中国科学基金, 2020, 34(02):186-192.
- [2] NOTE A. Agilent MIMO Wireless LAN PHY Layer [RF] Operation & Measurement.
- [3] MARZETTA T L. Non-cooperative cellular wireless with unlimited numbers of base station antennas[J]. IEEE Transactions on Wireless Communication, 2010, 9(11):3590-3600.
- [4] GAO X, EDFORS O, RUSEK F, TUFVESSON F. Linear Pre-Coding Performance in Measured Very-Large MIMO Channels[C]. 2011 IEEE Vehicular Technology Conference, VTC, 2011, 1-5.
- [5] YEUNG K S, KUMBI F. Symbolic matrix inversion with application to electronic circuits[J]. IEEE Transactions on Circuits and Systems, 1998, 35(2):235-238.
- [6] STERITI R J, FIDDY M A. Regularized image reconstruction using SVD and a neural network method for matrix inversion[J]. IEEE Transactions on Signal Processing, 1993, 41(10):3074-3077.
- [7] TANK D, HOPFIELD J. Simple 'neural' optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit[J]. IEEE Transactions on Circuits and Systems, 1986, 33(5):533-541.
- [8] ZHANG Y, GE S S. Design and analysis of a general recurrent neural network model for time-varying matrix inversion[J]. IEEE Transactions on Neural Networks, 2005.16(6):1477-1490.
- [9] ZHANG Y, GUO X, MA W, et al. MATLAB Simulink Modeling and Simulation of Zhang Neural Network for Online Time-Varying Matrix Inversion[C]// IEEE International Conference on Networking. IEEE, 2016.
- [10] XIAO L. A new design formula exploited for accelerating Zhang neural network and its application to time-varying matrix inversion[J]. Theoretical Computer Science, 2016, 50-58.
- [11] TAN Z, HU Y, XIAO L, et al. Robustness Analysis and Robotic Application of Combined Function Activated RNN for Time-Varying Matrix Pseudo Inversion[J]. IEEE Access, 2019, 7:33434-33440.
- [12] ZHANG Y, GUO D. Time-varying Complex Matrix Inverse[J]. Springer, Berlin Heidelberg, 2015, 163-172.
- [13] WANG X, WEI Y, STANIMIROVIC P S. Complex neural network models for time-varying Drazin inverse[J]. Neural Computing, 2016, 28 (12):2790-2824.
- [14] ZHANG Z, ZHENG L. A Complex Varying-Parameter Convergent-Differential Neural-Network for Solving Online Time-Varying Complex Sylvester Equation[J]. IEEE Transactions on Cybernetics, 2018, (99):1-13.
- [15] XIAO L, ZHANG Y, LI K, et al. A novel recurrent neural network and its finite-time solution to



- time-varying complex matrix inversion[J]. *Neurocomputing*, 2019, 331: 483-492.
- [16] XIAO L, ZHANG Y, ZUO Q, et al. A noise-tolerant zeroing neural network for time-dependent complex matrix inversion under various kinds of noises[J]. *IEEE Transactions on Industrial Informatics*, 2019, 16 (6): 3757-3766.
- [17] BJRNSEN E, SANGUINETTI L, H WYMEERSCH, et al. Massive MIMO is a reality--What is next?: Five promising research directions for antenna arrays[J]. *Digital Signal Processing*, 2019, 94:3-20.
- [18] QIANG H, ZHANG M, GAO R, et al. Key Technologies in Massive MIMO[J]. *Itm Web of Conferences*, 2018, 17:01017.
- [19] LEE B. Simplified antenna group determination of RS overhead reduced massive MIMO for wireless sensor networks[J]. *Sensors*, 2017, 18(2):84.
- [20] FATEMA N, HUA G, XIANG Y, et al. Massive MIMO Linear Precoding: A Survey[J]. *IEEE Systems Journal*, 2017.12(4):39203931.
- [21] HASAN Z, BANSAL G, HOSSAIN E, et al. Energy-efficient power allocation in OFDM-based cognitive radio systems: A risk-return model[J]. *IEEE Transactions on Wireless Communication*, 2019, 8(12):6078-6088.
- [22] MIAO G, HIMAYAT N, LI G, et al. Low-complexity energy efficient scheduling for uplink OFDMA[J]. *IEEE Transactions on Communication*, 2012, 60(1):112-120.
- [23] SPEIDEL J. *Introduction to Digital Communications*[C]. Cham, Switzerland: Springer, Sep. 2018.
- [24] LEE K K, YANG Y, LI J. A low-complexity AEPDF assisted precoding scheme for massive MIMO systems with transmit antenna correlation[J]. *Signal Processing*, 2020, 92:1-11.
- [25] PRABHU H, EDFORS O, RODRIGUES J, et al. Hardware efficient approximative matrix inversion for linear pre-coding in massive MIMO[C]. *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2014, 1700-1703.
- [26] ZHANG C, LI Z, SHEN L, et al. A low complexity massive MIMO precoding algorithm based on Chebyshev iteration[J]. *IEEE Access*, 2017, 5:22545-22551.
- [27] GESBERT D, HJORUNGNES A. Complex-Valued Matrix Differentiation: Techniques and Key Results[J]. *IEEE Transactions on Signal Processing*, 2007, 55(6):2740-2746.
- [28] HAYKIN S, *Adaptive Filter Theory* 4th ed[M]. Englewood Cliffs, New Jersey, USA: Prentice Hall, 2002.
- [29] 潘武, 周世东, 姚彦. 瑞利衰落信道下差分跳频通信系统性能分析[J]. *无线通信技术*, 2003, 12(2):34-38.
- [30] CHIZHIK D, LING J, WOLNIANSKY P, et al. Multiple input multiple output measurements and modeling in Manhattan[J]. *IEEE Journal on Selected Areas in Communications*. 2003, 21(3):321–331.
- [31] JOHN G, PROAKIS. *Digital Communications* 3rd edition[M]. Singapore: McGraw-Hill Book Co. 1995:

767–768.

[32] BERNARD SKLAR. Rayleigh Fading Channels in Mobile Digital Communication Systems Part I: Characterization. IEEE Communications Magazine. July 1997, 35 (7): 90–100.

[33] 李贤平.概率论基础（第二版）[M]. 北京:高等教育出版社, 1997.

[34] Papoulis A, Pillai S. Probability, Random Variables and Stochastic Processes[M], McGraw-Hill College. 2001.

## 致 谢

本科生涯即将结束，我在东南大学学习生活了四年。东南大学环境优美，师生情谊深厚，“止于至善”的校训时时刻刻激励着我。带着东大的烙印，未来我无论在哪里、在做什么，都要践行“止于至善”的校训。

首先我要感谢我的毕设导师汪磊老师，本次毕设是在汪老师悉心的指导下完成的，汪老师给了我非常多的帮助。从选题到开题，再到中期，最后到毕设的完成，汪老师无时不耐心指导、热心帮助。在毕设的过程中遇到困惑或是迷失方向时，汪老师耐心细致的指点帮助我找到前进的方向。汪老师有丰富的理论实践经验和严谨的科研工作态度，是我今后努力的榜样。感谢汪老师科研给我完成这项毕设的机会，通过这次学习，我收获的不仅仅是这个课题的知识，更是整个科研工作的综合素养的提高。

感谢我的班主任刘升恒老师。作为班主任，刘老师非常关心同学们的学习和生活，经常与我们谈心。尤其在学习规划和心理建设上，刘老师给予了我非常大的帮助，为我指点迷津。我对今后的深造目标和职业规划有了更加清醒的认识。俗语云，听君一席话，胜读十年书，刘老师帮助我树立了一个东大人“止于至善”的信仰，未来无论在哪里、做什么，都会时刻践行。

感谢在大三教过我并给予我科研启蒙的张铖老师，张老师在课堂上就注重培养我们的科研探索和展示能力，让我们明白，科研不仅仅是埋头于实验室，更要有能力将自己的成果科学合理的表述出来。张老师是我的科研启蒙导师，帮助我迈入了科研的大门，并且坚定了我未来从事科学事业的决心。同时也非常感谢张老师愿意为我写美国大学研究生的推荐信，帮助我收获了 offer。

感谢我的室友们、同窗们、朋友们，生活中大家互帮互助、学习上大家合作交流，让我的大学四年无比充实、受益匪浅。祝愿我的朋友们未来一切顺利、事业蒸蒸日上。

最后，感谢东南大学对我的培养；感谢信息科学与工程学院为我提供的良好的学习氛围；感谢教过我的每一位老师、生活中帮助过我的每一个人！