

清华 大学

综合 论文 训 练

题目：基于超导量子比特与固态自旋
的混合量子系统

系 别：物理系

专 业：物理学

姓 名：蒋文韬

指导教师：宋祎璞副研究员

2017 年 5 月 26 日

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：学校有权保留学位论文的复印件，允许该论文被查阅和借阅；学校可以公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存该论文。

(涉密的学位论文在解密后应遵守此规定)

签 名：_____ 导师签名：_____ 日 期：_____

中文摘要

论文的摘要是对论文研究内容和成果的高度概括。摘要应对论文所研究的问题及其研究目的进行描述，对研究方法和过程进行简单介绍，对研究成果和所得结论进行概括。摘要应具有独立性和自明性，其内容应包含与论文全文同等量的主要信息。使读者即使不阅读全文，通过摘要就能了解论文的总体内容和主要成果。

论文摘要的书写应力求精确、简明。切忌写成对论文书写内容进行提要的形式，尤其要避免“第 1 章……；第 2 章……；……”这种或类似的陈述方式。

本文介绍清华大学论文模板 THUTHESIS 的使用方法。本模板符合学校的本科、硕士、博士论文格式要求。

本文的创新点主要有：

- 用例子来解释模板的使用方法；
- 用废话来填充无关紧要的部分；
- 一边学习摸索一边编写新代码。

关键词是为了文献标引工作、用以表示全文主要内容信息的单词或术语。关键词不超过 5 个，每个关键词中间用分号分隔。（模板作者注：关键词分隔符不用考虑，模板会自动处理。英文关键词同理。）

关键词：`TeX`; `LATEX`; CJK; 模板; 论文

ABSTRACT

An abstract of a dissertation is a summary and extraction of research work and contributions. Included in an abstract should be description of research topic and research objective, brief introduction to methodology and research process, and summarization of conclusion and contributions of the research. An abstract should be characterized by independence and clarity and carry identical information with the dissertation. It should be such that the general idea and major contributions of the dissertation are conveyed without reading the dissertation.

An abstract should be concise and to the point. It is a misunderstanding to make an abstract an outline of the dissertation and words “the first chapter”, “the second chapter” and the like should be avoided in the abstract.

Key words are terms used in a dissertation for indexing, reflecting core information of the dissertation. An abstract may contain a maximum of 5 key words, with semi-colons used in between to separate one another.

Keywords: T_EX; L^AT_EX; CJK; template; thesis

目 录

目 录	III
第 1 章 引言：基于超导量子比特与固态自旋的混合量子系统简介	1
1.1 混合量子系统的重要性与现有研究状况	1
1.2 超导量子系统与常见固态自旋系统	2
1.3 超导量子系统与自旋系综的耦合	4
1.4 超导量子比特的制备与测量	6
第 2 章 自旋与谐振腔耦合强度仿真	7
2.1 新型 3D 谐振腔与自旋系综的耦合	7
2.2 2D 平面波导谐振腔与自旋系综的耦合	9
2.3 螺旋状电感谐振腔与 2.5 维谐振腔与单个自旋的耦合	11
2.3.1 参数估计	11
2.3.2 设计与优化	12
第 3 章 PPMS 测量系统	16
3.1 测量系统概述	16
3.2 测量系统的程序编写	19
3.3 测量系统测试	21
3.3.1 频率扫描	21
3.3.2 温度扫描	21
3.3.3 磁场扫描	21
3.4 样品托的设计与优化	21
第 4 章 2.5 维谐振腔的制备与测量	24
4.1 器件测量与制备工艺概述	24
4.2 光刻板的设计与改进	24
4.3 器件制备情况与改进	26
4.4 谐振腔响应信号的拟合	29
插图索引	30

表格索引	32
公式索引	33
参考文献	35
致 谢	38
声 明	39
附录 A 超导量子比特的原理	40
附录 B 微纳加工工艺	41
B.1 光刻	41
B.2 介电层生长与刻蚀	41
B.2.1 刻蚀介电层后去除光刻胶	41
B.3 光刻胶 Lift off	42
B.4 磁控溅射镀膜	42
B.5 Argon milling 去除氧化层	43
B.6 点焊	43
附录 C PPMS 系统的常用操作	44
C.1 调节温度与磁场	44
C.2 更换样品	44
附录 D 测量系统 MATLAB 代码	45
D.1 VNA 控制代码	45
D.2 PPMS 控制代码	60
D.3 扫描温度与磁场实验的代码	66
D.3.1 扫描温度	66
D.3.2 扫描磁场	69
附录 E 外文资料调研阅读报告	73
在学期间参加课题的研究成果	74

主要符号对照表

ALD	原子层沉积
CPW	平面传输线波导
dll	动态链接库
EBL	电子束曝光
ICP	电感耦合等离子体
IPA	异丙醇
MF-319	实验中使用的光刻胶显影液
NMP	实验中使用过的光刻胶去胶液
NV	氮-空位色心
PECVD	等离子体增强化学气相沉积
PG remover	电子束曝光胶去胶液，也可用于去除光刻胶
PPMS	综合物性测量系统
S1805	实验中所用的光刻胶型号
SCPI	可编程仪器标准命令
VI	虚拟仪器
VISA	Virtual Instrument Standard Architecture
VNA	网络分析仪

第1章 引言：基于超导量子比特与固态自旋的混合量子系统简介

自从量子理论于上世纪初被提出、建立与发展以来，对世界产生了众多深远的影响。而在 1970 到 1980 年间，一些学者开始以可设计的角度来看待与研究量子系统^[1]，这带来了一系列观念的变化，人们开始思考如何制备与设计量子系统以达到不同的目的，以及综合物理、计算机科学以及信息论来提出一些全新的问题。^[2] 自从量子加密通信的 BB84 协议^[3]，量子搜索算法^[4] 与量子质因数分解算法^[5] 被提出后，人们看见了基于量子力学原理的计算机与通信系统能够在一些问题上达到超越经典系统的性能，进而促进了量子信息实验的进展。

1.1 混合量子系统的重要性与现有研究状况

基于量子力学的计算机的最基本的组成元素为量子比特。一个量子比特是一个二能级量子系统的统称。为了满足组建量子计算机的目的，一个好的二能级系统需要可扩展，可初始化，退相干时间远大于单次操作时间，可构建任意的量子逻辑门，可被独立测量这五个条件^[6]。人们对许多不同的微观二能级系统进行了尝试，包括量子点^[7]，离子阱^[8]，固态自旋系统^[9]，超导量子比特^[10] 以及线性光学^[11] 等。

这些不同系统各自建立的量子比特有不同的特点，例如基于离子阱的量子比特有很高的操作与测量成功率，但其扩展性相对较差；超导量子比特有较好的扩展性，并且容易操作，但其退相干时间则相对较短；基于光子的量子比特则是量子通讯的最佳选择。因此，通过将不同量子系统耦合起来，分别利用他们各自的优点进行相应的操作，是量子计算与量子通信的发展趋势。本文主要关注由超导量子比特与固态自旋系统构成的混合量子系统。

利用自旋系综作为量子存储器的想法在近十年前便开始有人提出^[12-14]，并且进行了相关基础实验，如自旋系综与超导微波谐振腔的耦合^[15]，并且能够达到强耦合的程度^[16]，也即耦合强度超过了自旋系综以及谐振腔的衰减与退相干速率。这些实验充分说明利用自旋系综与超导谐振腔耦合这一课题的重要性与意义。另一方面，也有直接将自旋系综或是单个自旋与超导量子比特进行耦合

的相关理论计算^[17]与实验工作^[18,19]。这篇文章主要考虑单个自旋或自旋系综与谐振腔的耦合，进而通过谐振腔再与超导量子比特进行耦合，不考虑单个自旋或自旋系综与超导量子比特的直接耦合。通过自旋与谐振腔的耦合，还可以通过谐振腔达到量子极限的测量精度^[20]以及通过谐振腔的 Purcell 效应控制自旋系综的能量衰减速率^[21]，这些都是十分有意义的工作。

由于自旋通过谐振腔中的磁场与谐振腔形成耦合，磁场的不均匀性将导致自旋与谐振腔耦合强度的不均匀性。在提高谐振腔的磁场均匀性这个方面，有相关工作通过在传输线谐振腔的基础上增加中心传输线的条数达到改善谐振腔磁场均匀性的效果^[22,23]，也有相关工作从改良三维谐振腔的角度出发，在保持谐振腔中磁场的均匀性的效果前提下增加磁场强度进而增加耦合强度^[24]。在单个自旋与谐振腔耦合这方面，通过改进电感部分的设计，可以局域地增强磁场强度，进而增大谐振腔与单个自旋的耦合强度^[25,26]，从而可能通过单个自旋达到强耦合的程度^[27]。

本文将计算并重现对传输线谐振腔磁场的仿真计算并估计自旋与传输线谐振腔的耦合强度，也从改良三维谐振腔的角度进行了仿真。另一方面，本文仿真并改进了通过重新设计谐振腔电感部分以增强耦合强度的设计，并进一步开始尝试该设计的微纳加工实现，并整理了微纳加工流程。对制备出来的器件，通过 PPMS 测量系统进行了测量。本文的第2章中介绍了前文提到的多个仿真工作。对于测量系统的搭建与改进的相关内容在第3章中有详细介绍，并在第4章中展示了改良的谐振腔的微纳加工工艺并利用搭建的测量系统对其进行测量。在附录A中介绍了超导量子比特的基础知识，随后在附录B中包含了微纳加工各个步骤的详细流程与相关参数。测量系统搭建过程中编写的控制程序也附于附录D中供查阅。

1.2 超导量子系统与常见固态自旋系统

常见的超导量子比特由以下哈密顿量描述：

$$H_{sc} = 4E_C(\hat{n} - n_g)^2 - E_J \cos \phi \quad (1-1)$$

其中 E_C 与 E_J 分别为电容能量与约瑟夫森能量，两者比值的不同取值范围对应不同的超导量子比特种类，本文的工作将集中关注 Transmon 超导量子比特，这种量子比特对应 $E_J/E_C \sim 50$ 的量级^[28]。通过哈密顿量可以看出，超导量子比特对应非线性的谐振子，其能级非均匀分布，因而可以控制其量子态处于两个选定的本征态构成的态空间内，一般选择其基态与第一激发态，进而近似作为一个两能级系统构成量子比特。经过一系列化简，可将 Transmon 超导量子比特的哈密顿量写成两能级系统的标准形式

$$H_{trans} = \frac{1}{2}\hbar\omega_a\sigma_z \quad (1-2)$$

其中 $\omega_a = \sqrt{8E_J E_C}/\hbar$ 为基态与第一激发态的能量差对应的频率。通过将超导量子比特与平面谐振腔进行耦合，即可进行超导量子比特的操作与读取。

自旋为很多微观粒子具有的量子特性，如电子自旋与核自旋。因为自旋与环境作用相对较弱，因此具有较长的退相干时间，是理想的存储介质。常见的自旋系统如金刚石色心（NV centers），由金刚石中一个碳原子被氮原子替代，以及相邻的一个碳原子空缺共同组成，构成一个等效的自旋为一的量子系统，如图1.1所示。

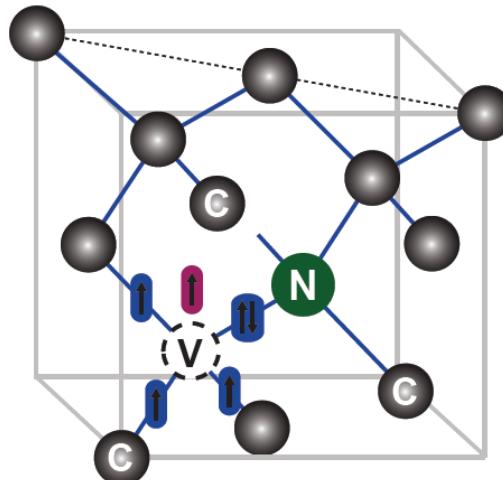


图 1.1 金刚石色心结构示意图，其中碳空位由 V 表示，氮掺杂由 N 表示^[29]。

考虑应力产生的零场劈裂以及外加静磁场后，一个金刚石色心中的自旋的

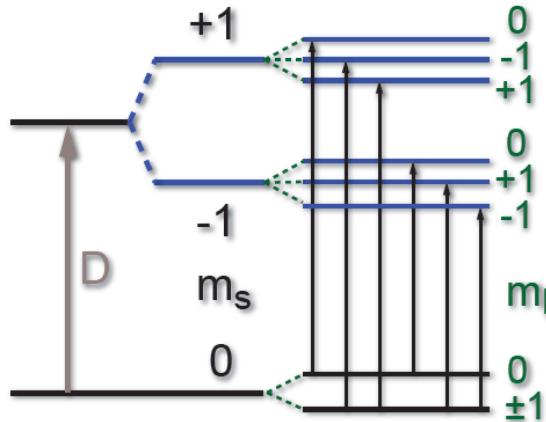


图 1.2 金刚石色心在考虑零场劈裂与外加磁场后的能级示意图。其中 D 为零场劈裂导致的能级分裂, $m_S = \pm 1$ 的两个态之间的能量差来源于应力与局域电场。不同核自旋量子数 m_I 的态之间的能级分裂来源于超精细相互作用^[29]。

简并能级发生分裂, 相应的哈密顿量为^[29]

$$H/\hbar = \mathbf{S} \cdot \bar{\mathbf{D}} \cdot \mathbf{S} - \gamma_e \mathbf{B}_{NV} \cdot \mathbf{S} + \mathbf{S} \cdot \bar{\mathbf{A}} \cdot \mathbf{I} + PI_Z^2 \quad (1-3)$$

其中 $\gamma_e = -g_e \mu_B / \hbar = -2\pi \times 2.8 \text{MHz/Gs}$ 为 NV 电子自旋的旋磁比。 $\bar{\mathbf{D}}$ 为零场劈裂张量, \mathbf{B}_{NV} 为金刚石色心所处位置的磁场, $\bar{\mathbf{A}}$ 为超精细相互作用张量, 最后一项为氮原子四极矩产生的能量项。知道了系统的哈密顿量后, 即可选取两个态构成量子比特, 这样近似下的哈密顿量与二能级系统的哈密顿量相同。通过将自旋与不同量子系统进行耦合, 即可达到存储与读取量子信息的目的。

1.3 超导量子系统与自旋系综的耦合

目前已有许多关于超导量子比特与固态自旋耦合的相关实验。由于自旋通过磁场与外界耦合, 强度很弱, 因此常采用自旋系综与平面波导谐振腔的磁场耦合, 谐振腔再与超导量子比特耦合的方法^[29]。这种方法能够实现多次的存储与读取, 本节将对这方面的理论工作与实验实现进行总结。

首先考虑单个 NV 自旋与谐振腔的耦合。单个 NV 自旋与谐振腔构成的混合量子系统, 可由以下哈密顿量描述

$$H = H_r + H_a + H_{int} \quad (1-4)$$

其中 $H_r = \hbar\omega_r a^\dagger a$ 为谐振腔的哈密顿量, H_a 即由1-3所描述的 NV 自旋自身的哈密顿量, 而 H_{int} 为两个系统相互作用的哈密顿量^[29]

$$H_{int} = -\gamma_e \mathbf{S} \cdot \mathbf{B} \quad (1-5)$$

$$= -\frac{\gamma_e}{\sqrt{2}} [\sigma_x \delta B_x(\mathbf{r}) + \sigma_y \delta B_y(\mathbf{r})] (a + a^\dagger) \quad (1-6)$$

$$= g^* a \sigma_+ + g a^\dagger \sigma_- \quad (1-7)$$

其中自旋-谐振腔耦合系数

$$g = -\frac{\gamma_e [\delta B_x(\mathbf{r}) + i \delta B_y(\mathbf{r})]}{\sqrt{2}} \quad (1-8)$$

δB 为谐振腔零场的磁场涨落。耦合系数 g 是自旋-谐振腔系统最关键的系数之一, 也是我们想要通过仿真进行估算以及通过改进器件设计与制备来提高其数值的物理量。为了达到 NV 自旋与谐振腔中的电磁模式的强耦合, 进而实现两者间量子信息的交换, 我们需要 $g \gg \kappa, \gamma$, 其中 κ 为谐振腔的衰减率, γ 为 NV 自旋的衰减率。对于单个自旋与二维平面波导传输线谐振腔间的耦合, $g \sim 2\pi \cdot 10\text{Hz}$ ^[29], 远远小于 κ, γ 的数量级, 因此我们需要改进用一个自旋系综与谐振腔耦合, 或者改进谐振腔的设计, 来提高耦合系数。

对于一个自旋系综与谐振腔耦合的系统, 其哈密顿量为 T-C 模型 (Tavis-Cummings model)^[30]

$$H_{TC}/\hbar = \omega_r (a^\dagger a + 1/2) + \frac{\omega_s}{2} \sum_{j=1}^N \sigma_z^{(j)} + g \sum_{j=1}^N (a \sigma_+^{(j)} + a^\dagger \sigma_-^{(j)}) \quad (1-9)$$

其中 $\sigma_{z,\pm}^{(j)}$ 为第 j 个自旋的泡利算符。所有自旋的态可以写为 $\prod_{j=1,\dots,N} |i\rangle_j$, 其中 $i = g, e$ 。为简化记号, 定义自旋系综的基态为 $|G\rangle \equiv |g_1 \dots g_N\rangle$, 以及第 j 个自旋被激发的激发态 $|E_j\rangle \equiv |g_1 \dots e_j \dots g_N\rangle$ 。通过引入系综自旋算符 $\mathcal{S}_{X,Y,Z} \equiv \sum_{j=1}^N \sigma_{x,y,z}^{(j)}/2$ 以及系综升降算符 $\mathcal{S}_\pm \equiv \sum_{j=1}^N \sigma_\pm^{(j)}$, 上式中的 T-C 模型哈密顿量可简化为

$$H_{TC}/\hbar = \omega_r (a^\dagger a + 1/2) + \omega_s \mathcal{S}_z + g(a \mathcal{S}_+ + a^\dagger \mathcal{S}_-) \quad (1-10)$$

基于系综的自旋算符, 可以发现总自旋算符 $\mathcal{S}^2 = \mathcal{S}_X^2 + \mathcal{S}_Y^2 + \mathcal{S}_Z^2$ 与整个 H_{TC} 交

换，即 $S(S+1)$ 为好量子数。因此我们通过 S^2 与 S_Z 的共同本征态来描述自旋系综系统。整个自旋系综系统的能级如下图所示。

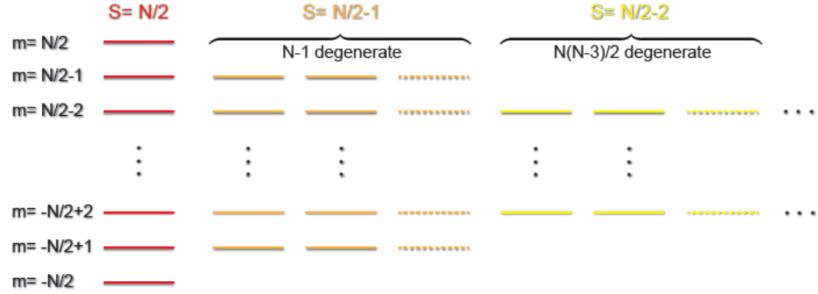


图 1.3 N 个金刚石自旋构成的自旋系综的能级示意图^[29]。可以看见 $S \neq N/2$ 的态均为高度简并态。

当自旋系综从谐振腔吸收一个光子时，相应的态为 $|B\rangle = |N/2, -N/2 + 1\rangle \equiv S_+ |G\rangle / |\mathcal{S}_+ |G\rangle| = \sum_k |E_k\rangle / \sqrt{N}$ 。其余 $N-1$ 个单激发的激发态可写为 $|D_j\rangle = \sum_{k=0}^{N-1} \exp(ijk2\pi/N) |E_k\rangle / \sqrt{N}$ ，其中 $j = 1, \dots, N-1$ ，且易验证 $\langle D_j | B \rangle = 0$ 。因此由能级图 1.3 易看出所有 $|D_j\rangle$ 对应的态均为 $S = N/2 - 1$ ，因此不可能通过 H_{TC} 与基态 $|G\rangle$ 耦合起来。综合上述讨论，

$$\langle E, 0 | H_{TC} | G, 1 \rangle = (1/\sqrt{N}) \sum_i g = g\sqrt{N} \quad (1-11)$$

$$\langle D_j, 0 | H_{TC} | G, 1 \rangle = 0 \quad (1-12)$$

通过上述计算可以看出，对于 N 个自旋构成的自旋系综，系综整体与谐振腔的耦合强度比单个自旋的耦合强度大了系数 \sqrt{N} 。

1.4 超导量子比特的制备与测量

建立超导量子比特与自旋的混合量子系统的基础之一是两者的成功耦合，以及超导量子比特的制备与调控。因此，本文也将对超导量子比特的理论基础^[28,31]，制备方法^[32,33] 以及测量方法^[34] 进行调研与总结。并以附录和文献综述的形式给出。

第2章 自旋与谐振腔耦合强度仿真

通过文献综述我们看到，单个自旋与常见谐振腔的耦合强度较弱，因此我们希望通过利用自旋系综与谐振腔进行耦合或者尝试新的谐振腔设计来解决这个问题。对自旋系综与谐振腔进行耦合的情况，由于自旋系综在空间有分布，而谐振腔所产生的磁场在空间也有所分布，因此探究谐振腔产生的磁场的空间分布即成为估计耦合系数的强度大小及其空间分布的重要步骤。

另一方面，与平面波导谐振腔相对应，三维谐振腔的电磁场空间分布更均匀，但强度相对更弱，因此耦合强度更小。在尝试更新的谐振腔设计时，我们也考虑改良三为谐振腔使之在保持自旋系综存在空间部分电磁场仍旧相对均匀的同时尽可能增加场强，而对于平面波导谐振腔，我们从进一步增大耦合强度并与单个自旋耦合的角度出发在现有提案^[27]的基础上进行了仿真，改进与优化。

2.1 新型3D谐振腔与自旋系综的耦合

我们首先发现并重复了对优化电磁场分布均匀性的3D谐振腔的仿真^[24]。这种三维谐振腔中电流与磁场分布示意图如下图所示

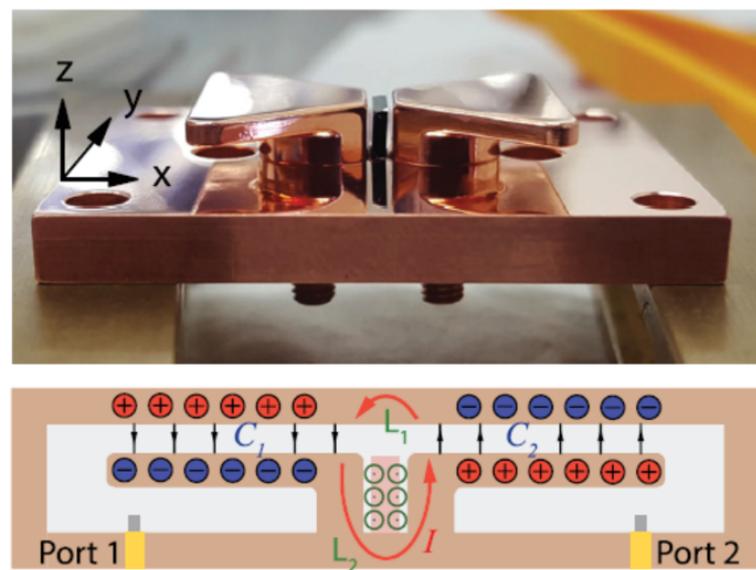


图 2.1 新型三维谐振腔的几何结构与电流，磁场分布示意图^[24]

其中这种三维谐振腔在普通密闭金属盒构成的三维谐振腔的基础上将电场与磁场局限于更小的体积当中，电场主要分布于两个扇形与盒顶之间，磁场主要分布于两扇形竖直支撑部分的两个平面之间，也为固定自旋系综的空间区域。因此这种三维谐振腔通过减小模式体积提高了零场涨落的大小，并同时仍保证自旋系综与磁场有较为均匀的耦合。

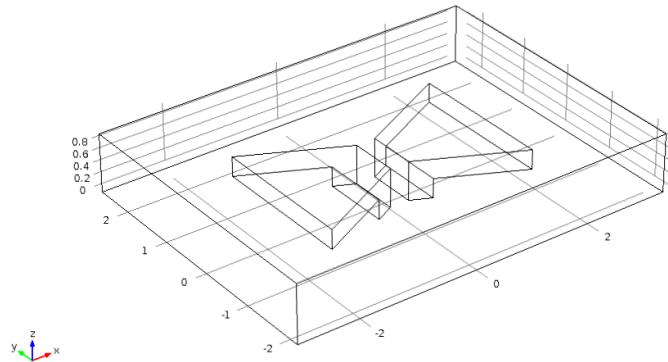


图 2.2 新型三维谐振腔仿真的几何设计 (透视图)

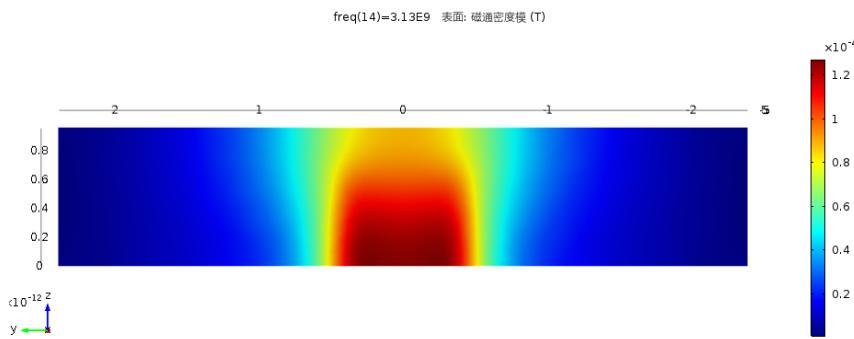


图 2.3 新型三维谐振腔磁场在自旋系综所处空间横截面上的大小分布

通过对 Angerer 等人使用的三维谐振腔的观察，我设计了如图2.2所示几何形状的三维谐振腔。利用 COMSOL 求解谐振腔中电磁场的分布，能够得到自旋系综所在的横截面上的磁场分布如图2.3所示。谐振腔的频率响应为频率和衰减率的函数^[29]

$$|S_{21}|^2 = \left| \frac{i\kappa/2}{2\pi(\delta f - \delta f_c) + i(\kappa + \kappa_L)/2} \right|^2 \quad (2-1)$$

通过对仿真所得的谐振腔的频率响应进行拟合，能够得到谐振腔的衰减率 κ, κ_L ，进而求出对应任意功率下的谐振腔中的光子数^[29]

$$n = \frac{2\kappa}{hf_c(\kappa + \kappa_L)^2} P \quad (2-2)$$

谐振腔中仿真的场与功率相关，知道给定功率下的谐振腔中的电磁场分布，以及相应腔内光子数后，即可简单计算得单位光子数对应的电磁场大小与分布。通过单位光子数的磁场大小，即可计算自旋与谐振腔中该模式的耦合强度^[24]

$$|g_0| = \sqrt{\frac{2}{3}} \frac{\mu_B g_e}{2\hbar} |\mathbf{B}_0| |\mathbf{S}| \sim 100 \text{ mHz} \quad (2-3)$$

通过仿真，拟合与计算得到的耦合强度，与 Angerer 等人所得到的耦合强度的数量级相符，验证了我们的仿真与计算的正确性。综合考虑后，我们认为这种方法对耦合强度的增加不明显，没有数量级的提升，并且这种三维谐振腔的制备较为复杂，我们没有继续进行这种新型三维谐振腔的制备。

2.2 2D 平面波导谐振腔与自旋系综的耦合

目前有很多工作通过将自旋系综与二维平面波导谐振腔进行耦合，达到了强耦合的效果^[15,16]。对于这类耦合，谐振腔的耦合强度的大小及其分布依赖于电磁场的空间分布。因此，我对二维平面波导的电磁场的空间分布进行了仿真，并与文献进行了比较。

对于二维平面波导的仿真，空间中场的分布由金属中电流密度的分布决定，因此仿真的关键为得到可靠的电流密度分布，从而得到空间中场的分布。超导效应对金属中场的分布体现在电流穿透金属表面的深度有限，即与高频电流导致的趋肤效应十分类似，因此可通过高频电流的趋肤效应对超导电流的分布进行模拟^[19]。趋肤效应的深度与电流频率相关：

$$\lambda_{skin} = \sqrt{\frac{2}{\sigma\omega\mu}} \quad (2-4)$$

通过使趋肤效应的深度等于超导电流的穿透深度，我们估计得仿真所需采用的

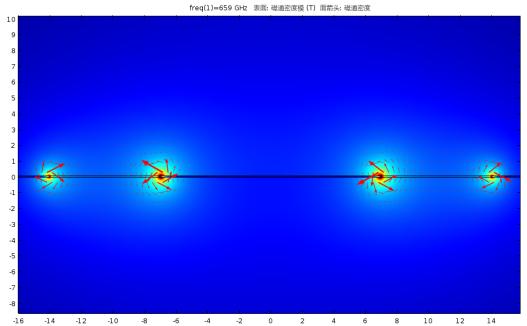


图 2.4 仿真所得磁场空间分布截面图

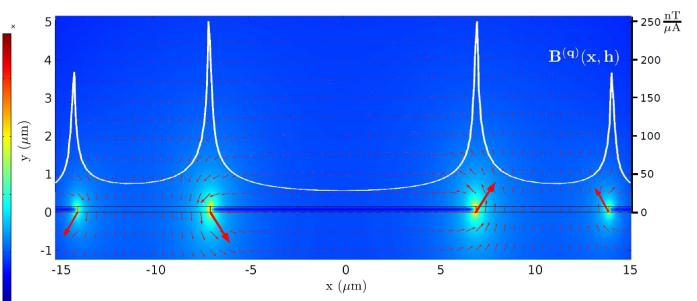


图 2.5 文献^[19]中所示磁场空间分布截面图

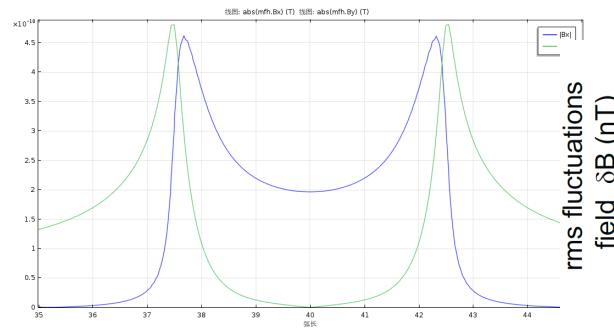


图 2.6 仿真所得磁场在位于距金属上表面 $0.1\mu\text{m}$ 处水平截线上的分布

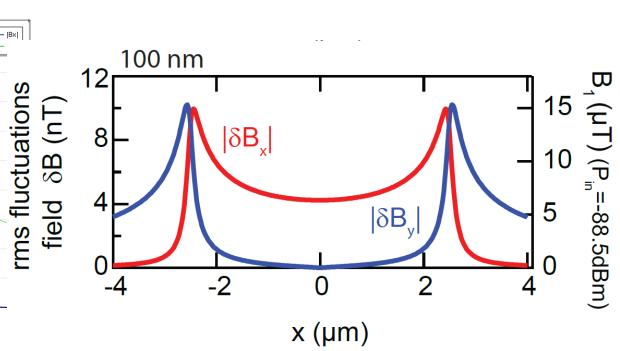


图 2.7 文献^[20]中所示磁场在位于距金属上表面 $0.1\mu\text{m}$ 处水平截线上的分布

电流频率为 $\omega \approx 660\text{GHz}$ 。这个频率并不对应实际物理系统中的任何频率，仅为仿真所采用的一个参数。零场涨落下的电磁场分布直接通过使总电流的大小为零场电流涨落的大小来得到。零场电流的大小通过计算可得^[19,35,36]

$$\frac{\hbar\omega}{2} = 2 \times \frac{1}{2} L (\delta i_{rms})^2 \quad (2-5)$$

$$L = \frac{2Z_0}{\pi\omega} \quad (2-6)$$

$$\delta i_{rms} = \omega \sqrt{\frac{\hbar\pi}{4Z_0}} \approx 50\text{nA} \quad (2-7)$$

通过上述方法，我采用了与两篇文章中相同的器件几何尺寸对磁场进行了仿真，如图所示。

通过利用上述方法对两篇独立的文章中的几何结构进行仿真并且与文章中的结果比较，可以看出结果相符。验证上述方法的可行性后，我对我们制备的二

维平面波导的常见几何尺寸进行了仿真，并绘制了距离波导金属表面不同高度处的水平截线上的磁感应强度分布，如图2.8所示。

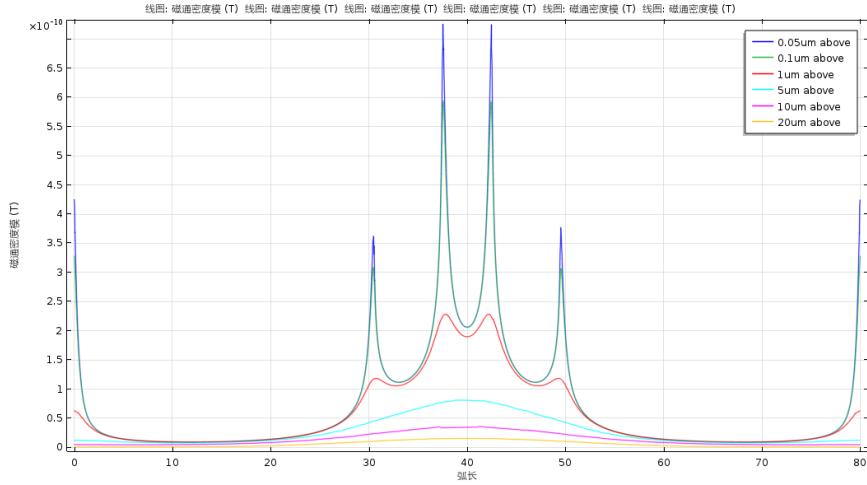


图 2.8 基于我们制备的二维平面波导的常见几何尺寸进行仿真所得的磁感应强度分布图。

通过仿真结果可以看出，磁场在金属上方 $10\mu\text{m}$ 左右的范围内开始分布均匀，并且强度在 $0.5 \times 10^{-10}\text{T}$ 的范围左右，对应的耦合强度 $g \sim 2\pi \cdot 10\text{Hz}$ ，与文献中对单个自旋与二维平面波导谐振腔的耦合常数的估计值相符很好。因此，通过这种仿真方法，我们能够很快得到给定任意几何形状的二维平面波导的空间磁场分布，进而得到空间内任意一点处的自旋与基于该二维平面波导构成的谐振腔的模式的耦合强度。

2.3 螺旋状电感谐振腔与 2.5 维谐振腔与单个自旋的耦合

前面讨论了利用与自旋系综耦合来增大耦合常数。通过观察耦合常数的表达式1-8，可以看见还可以通过提高零场涨落的大小来提高单个自旋与谐振腔模式的耦合常数。

2.3.1 参数估计

通过对零场电流大小进行估计的2-5式，可以看到零场电流涨落

$$\delta I = \sqrt{\frac{\hbar\omega}{2L}} \quad (2-8)$$

而对于选定的自旋种类以及基于选定自旋的能级系统定义出的二能级量子比特，其能级间能量差大致固定，因此谐振腔的谐振频率也大致固定在该能量所对应的频率。而谐振腔的频率由 $\omega = 1/\sqrt{LC}$ 确定。通过上述讨论可以看到，通过增大零场电流涨落可以增大磁场，进而增大单个自旋与谐振腔模式的耦合常数，增大零场电流涨落可由减小谐振腔的电感 L 实现，而对于固定频率的谐振腔，减小 L 意味着增大电容 C 。如果我们想要达到的耦合强度 $g/2\pi \sim 1\text{MHz}$ ，并且利用如基于金刚石色心的能量差对应频率在 3GHz 左右的自旋系统，可以估计出相应的零场电流涨落，对应的谐振腔电感与电容的数量级为

$$\delta I \sim 1\mu\text{A} \quad (2-9)$$

$$L \sim 1\text{pH} \quad (2-10)$$

$$C \sim 1\text{nF} \quad (2-11)$$

对于 nF 数量级的电容，无法通过如齿状二维电容等二维设计实现，而可使用三维平板电容实现。因此，我把这种电感为二维结构而电容为三维结构的谐振腔称为 2.5 维谐振腔。已有研究人员提出基于这种思路设计的 2.5 维谐振腔^[27]，我基于他们的谐振腔设计进行了仿真与优化。

2.3.2 设计与优化

Sarabi 等人提出的设计如图2.9所示

其中主要图示均为俯视图，左图的下方的小图为横截面示意图。右图中蓝色的部分即为层状电容的俯视图。通过根据相关几何参数进行仿真，我得到了这种谐振腔的电感部分的磁场分布图，如图2.11所示。通过分析我们发现，耦合系数能够得到数量级上的提升的根本原因是选取了很小的电感 L ，从而得到了较大的零场磁场涨落，并且自旋距离电感部分导线的距离较近，为 10nm 的数量级。而与之相对的，螺旋状电感的螺旋圈数则相对不那么重要，不会对耦合强度产生数量级上的影响，反而加大了微纳加工制备的难度。因此，我们进一步对螺旋状电感进行了分析，改良与仿真。

首先，我简化了螺旋状的电感结构，直接采用一根细导线作为电感，并通过磁场仿真来估算电感的数量级，仿真结果如图2.12所示。通过仿真估算得到的电感值为 $L \approx 2 \times 10^{-12}\text{H}$ ，比理想的电感值多出一倍左右。考虑到所需的强磁场区

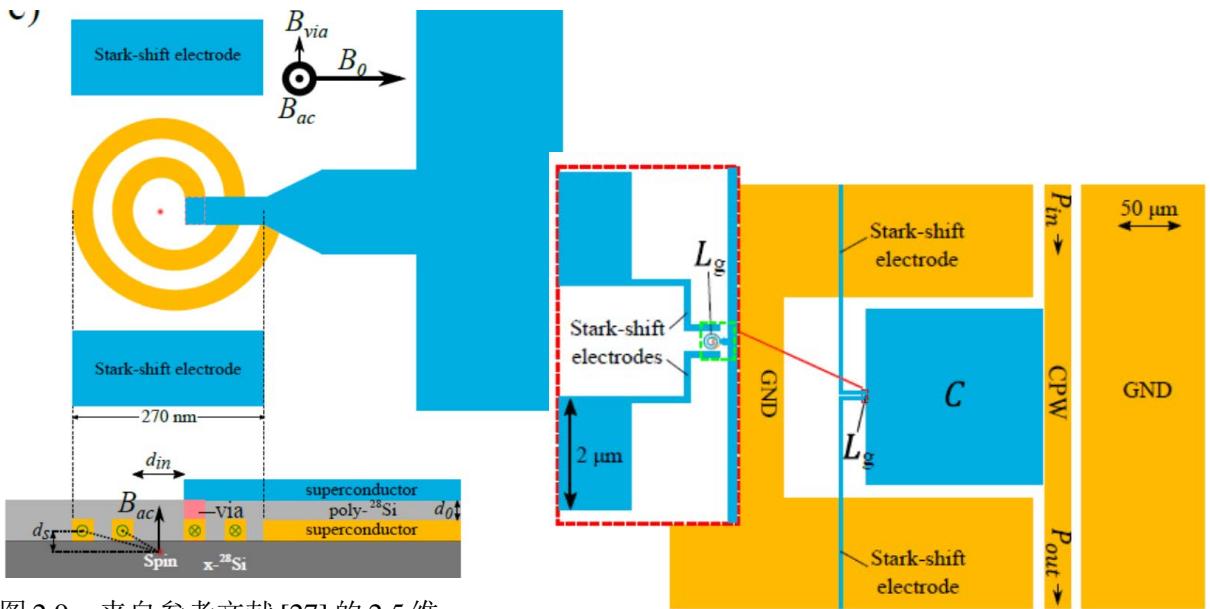


图 2.9 来自参考文献 [27] 的 2.5 维谐振腔的设计（右图中红色虚线部分的局部放大图）。

图 2.10 来自参考文献 [27] 的 2.5 维谐振腔的设计（整体）。

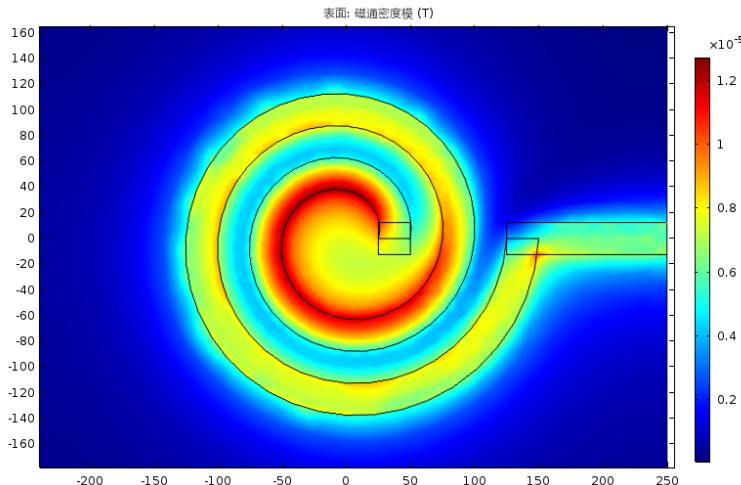


图 2.11 2.5 维谐振腔的螺旋状电感处水平截面上的磁感应强度分布。

域并不需要分布于整个电感导线，而只需要在圆弧附近即可，而强磁场存在的区域更大自然会增大导线的自感。基于这个想法，我进一步改进了电感的几何设计，仅保留电感中间的圆弧部位较细，这样电流密度增大，磁场相应增大，而对于电流流入与流出圆弧部位的部分，使导线变宽，如图2.13所示，即可减小大部分区域的磁场大小。通过仿真得出，电感大小的确减小到 $L \approx 7 \times 10^{-13} \text{H}$ ，减小超过 50%。

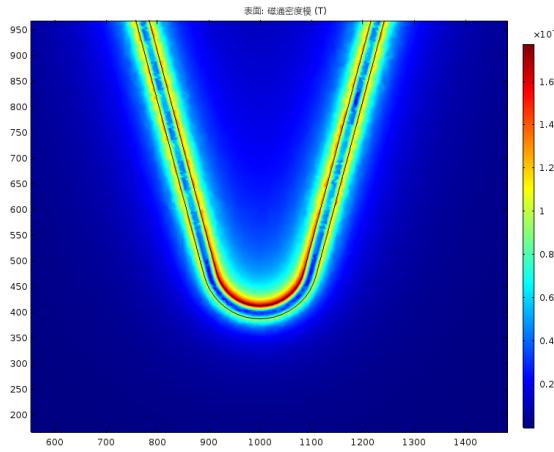


图 2.12 第一次改进后的 2.5 维谐振腔的电感处水平截面上的磁感应强度分布。

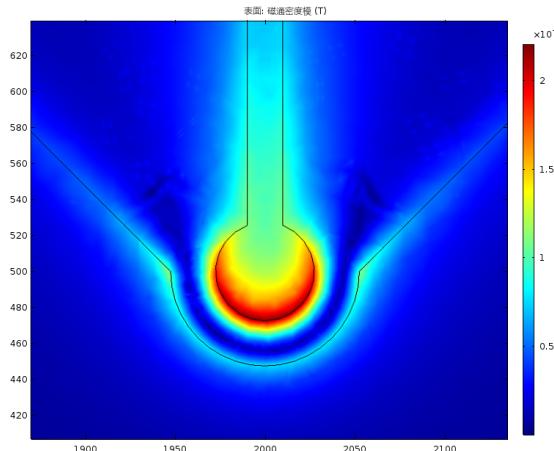


图 2.13 第二次改进后的 2.5 维谐振腔的电感处水平截面上的磁感应强度分布，电感部分的电感大小得到了降低。

通过进一步讨论并考虑到测量系统的温度下限对应的噪声大小，我们希望谐振腔中能有远多于 100 个光子的信号。通过简单估计我们发现对于图2.13以及其之前的结构，达到理想的光子数会使电流密度超过所用材料的超导临界电流密度，这样极有可能使器件损耗大大增加，并且因非超导态的电阻发热使电感部分的细导线烧断，导致器件损毁，因此需要进一步改进设计使同样临界电流密度的材料能够承载更多的电流。另一方面，图2.13的设计由于电感环状结构的前后为尽可能增大导线宽度使输入与输出导线中部距离十分靠近，为 $\sim 10\text{nm}$ 的数量级，在微纳加工过程中极易连接起来，在实际测试中也出现输入与输出两部分连接起来的现象。考虑上述两个因素后，我进一步改进了电感的设计，通过加厚材料并加宽导线圆弧部分，提高了可承载总电流的大小，并使导线的输入与

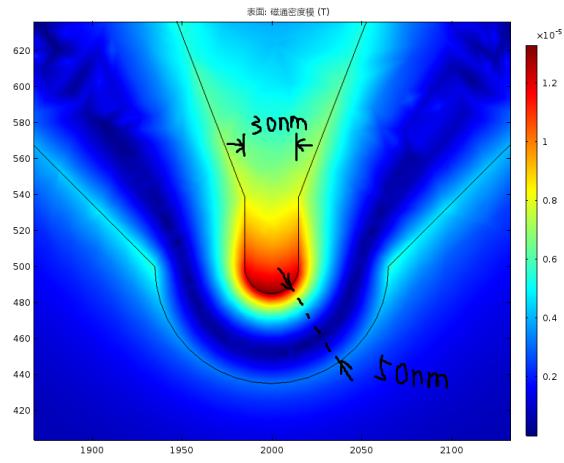


图 2.14 第三次改进后的 2.5 维谐振腔的电感处水平截面上的磁感应强度分布，通过加厚材料并加粗宽度提升了临界电流大小，并在不影响磁场较强区域的强度的基础上改进了设计以提高制备的成功率。

输出部分分离更远距离，提高了制备的成功率。最终的电感设计如图2.14所示。

第3章 PPMS 测量系统

器件的测量在 PPMS (Physical Property Measurement System) 中进行，为基础的二端微波测量。在 PPMS 中器件被冷却到 2K 左右，此时金属 Nb 进入超导态。通过网络分析仪直接测量二维平面波导的透射信号，即可在谐振腔的谐振频率处看到透射信号被吸收形成的凹陷。我利用现有的其他类型谐振腔对测量系统进行了测试，并通过拟合可得到谐振腔的相关参数。在测量系统搭建较为完善的基础上，进一步进行 2.5 维 LC 谐振腔的测量。

3.1 测量系统概述

实验中使用的网络分析仪为 Agilent Technologies E5071C ENA series network analyzer，输出频率范围为 300kHz 至 20GHz，下文中将简称其为 VNA。



图 3.1 PPMS DynaCool 测量设备实物图^[37]

实验中所用的 PPMS 为 QuantumDesign 公司的 DynaCool^[37]，本文中都将简称它为 PPMS。该 PPMS 可降温至 1.8 K，加磁场最大为 9T 或 14T，取决于磁体

的型号。PPMS 本体如图3.1所示，主要由控制电脑与仪器腔体两部分组成。仪器腔体的内部结构由图3.2所示。该图为仪器腔体在竖直平面内的横截面，可以看见样品室为被制冷环境包围住的一个立体圆柱形结构，直径不到一分米，有效的温度与磁场区域的高度大概在一分米左右。

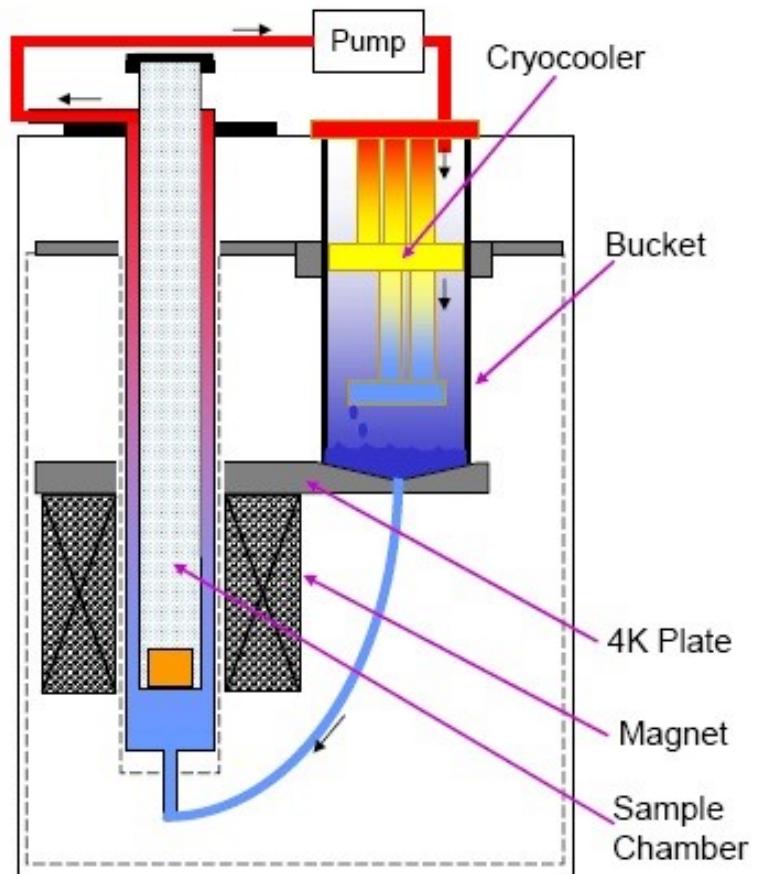


图 3.2 PPMS DynaCool 腔体内部结构^[37]

由于测量样品所在区域较小，因此无法进行需要很多端口的微波测量，但对于测量谐振腔性质所需的两个端口甚至一个端口来说较为合适。通过减小样品空间所取得的优点即在于，该 PPMS 系统的制冷系统长时间处于 4K 的低温，样品位于相对独立的样品室中，通过样品架与制冷系统的物理接触达到降温的效果，因此样品室可以较快地升温降温，而不需要对整个制冷系统进行升温降温。实际使用过程中，升温过程与降温过程所需时间均仅为 30 分钟左右，使得更换样品极为便捷。

在开始本文的测量相关项目之前，该 PPMS 系统多用于直流测量，没有微

波测量所需的设备。该部分设备的改造由交叉信息研究院孙麓岩研究组的郭星翰同学完成。改装后的样品通过样品托固定于样品杆上，进而通过样品杆插入 PPMS 的样品室中。由郭星翰设计的样品托如图3.3中蓝色矩形部分所示，由底座与盖子两部分组成。更换器件时，首先使 PPMS 样品室回到常温常压，取出样品杆，将样品托底座从盖子上拆下，再将旧样品从底座上取下，新样品安装上底座后将底座装回，插回样品杆即可。如图3.3所示，所测量的器件大小为 4mm x 7mm，放置于样品盒底座上的 PCB 板中央，通过点焊与 PCB 板相连。PCB 板上焊接有两个 SMP 接头，与样品杆上的微波线相连并接入 VNA 的输入与输出端口。

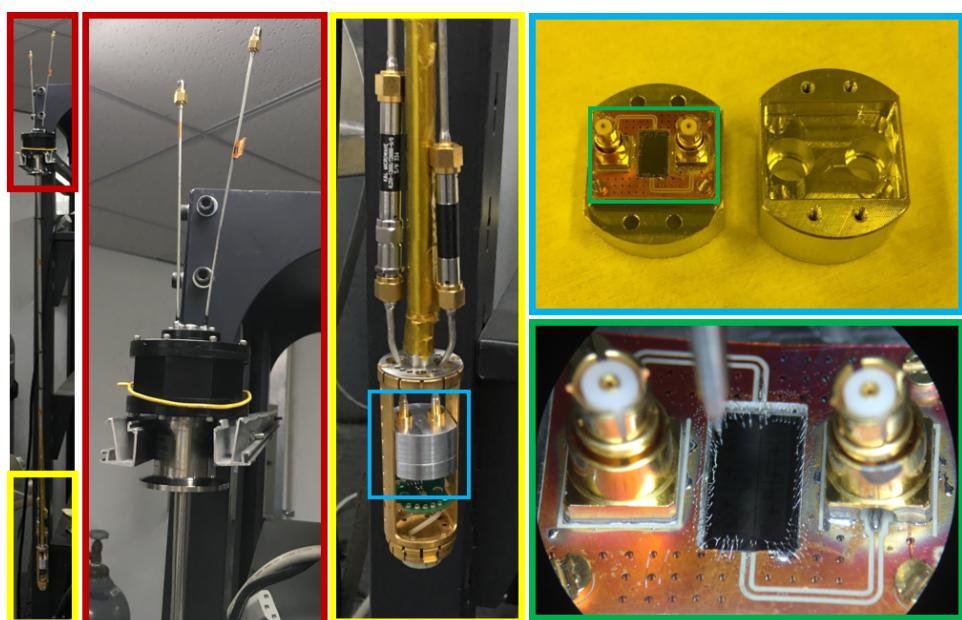


图 3.3 样品杆与样品盒。其中红色矩形部分为样品杆末端微波信号的输入与输出端，黄色矩形部分为样品盒所处位置，蓝色矩形为样品盒以及拆卸后的样品盒。绿色矩形内为样品盒底座上的 PCB 板，SMP 接头以及器件及其放大图

在实验进行过程中，我们考虑到将来进行自旋与谐振腔耦合的相关实验时需要加入竖直方向的磁场以改变自旋能级间距，而现有设计下磁场垂直与器件表面，对超导态下的器件的性质影响较大。因此，我在现有样品托的基础上改进了样品托底座与盖子以及 PCB 板的设计，具体在第3.4节中详细进行描述。

3.2 测量系统的程序编写

通过 VNA 测量谐振腔的频率响应时，需要调整 VNA 的输出功率，IFBW，扫描频率范围，平均次数等一系列参数。这些调整步骤既可通过仪器前面板的按钮进行，也可通过远程通讯来控制。考虑到测量的便捷性，我希望通过 GPIB 线与 VNA 进行通讯并取得数据。通过 Agilent 提供的 GPIB-USB 转接头建立硬件连接，并安装 Keysight Instrument Control Bundle 软件后，转接头上的工作指示灯正常。Keysight Instrument Control Bundle 软件提供了 Keysight IO Libraries Suite 软件的安装与 Command Expert 软件的安装。前者能够方便地查看仪器的连接状态，后者则提供仪器的 SCPI 指令集并可测试通过指令控制仪器。

通过电脑与 VNA 建立通讯后，我以 MATLAB 提供的 visa 类为基础，通过 Command Expert 对 VNA 的相关控制指令进行了测试后，编写了通过 MATLAB 控制 VNA 的代码，具体代码可在附录D中查看。对于能够简单进行更改与询问的参数，重载了 MATLAB 的 get 与 set 方法。对于大范围的精细的扫描，编写了 manualSweep 方法，可自动将扫描范围分段进行，并返回最终扫描结果。对于数据处理环节，我将拟合相关的代码也整合进入测量阶段，进而能够节省重新导入数据的过程直接快速得到拟合结果。使用 MATLAB 代码进行谐振腔的测量，在 1 个小时内即可完成大范围搜索谐振腔的共振频率位置，对不同频率的谐振腔进行细扫并拟合得到品质因子这一系列实验步骤。以下为一部分示例代码。大部分代码采用了 inputParser 处理输入参数，使程序规范且易于理解。

```
1 % Example of using class E5071C
2 vna = E5071C('address', 6); % initialize the instrument object with ...
   % GPIB address 6
3 vna.plotTrace; % fetch current trace on VNA and plot in ...
   % a figure
4 vna.freqCenter % query and display the center frequency
5 vna.freqSpan = 10e6; % set the frequency span to 10MHz
6 [freqs, trace] = ... % manual ...
   vna.manualSweep('start', 1e9, 'stop', 10e9, 'res', 1e5); % manual ...
   % sweep from 1GHz to 10GHz with 0.1MHz resolution
7 vna.freqCenter = 3.021e9; % set frequency center
8 vna.freqSpan = 1e6; % set frequency span
9 vna.plotTrace('issavedata', true, 'avg', 10); % wait for 10 averages ...
   % and save data while fetching the trace
10 vna.fit('fitall', true); % fit the data in the current figure
```

对于 PPMS 的控制，仪器商为这台仪器提供了配套的 LabVIEW 程序，可以通过.NET 网络协议远程控制 PPMS。仪器商所提供的 LabVIEW 程序基于一个动态链接库文件 QDInstrument.dll 实现控制功能。为了通过 MATLAB 控制 PPMS，我尝试将 LabVIEW 程序的 VI 封装成 dll 文件，再通过 MATLAB 加载与调用其中的函数。但由于 LabVIEW 的单个 VI 都会首先与 PPMS 建立连接，因此导致 MATLAB 中每调用一次 PPMS 的状态查看或是设置函数，就会重新建立一次连接，使程序运行缓慢，并且导致多个 client 同时与 PPMS 控制电脑上的 server 保持连接，可能导致潜在的问题。综合考虑后，我决定直接调用仪器商编写 LabVIEW 程序所调用的 dll 文件。

由于不清楚 QDInstrument.dll 文件中程序的构成与相关接口，而这些信息是调用其中的函数所必需的。通过查询，我使用了 ILSpy 对该 dll 文件进行了反编译，结合 LabVIEW 程序对该 dll 的使用方法，确定了在 MATLAB 中正确调用该 dll 文件的方法，并以它为基础编写了通过 MATLAB 控制 PPMS 程序的代码。需要注意的是，在 MATLAB 中的 PPMS 代码的构造函数中我一添加了加载该动态链接库的 MATLAB 指令，但每次重启 MATLAB 后初始化一个 PPMS 实例时总是会遇到 MATLAB 无法找到或识别 dll 中应有的命名空间的错误。目前较为确定的解决办法是每次重新启动 MATLAB 时，需在命令行（Command Line）中手动通过 NET.addAssembly 方法加载 dll 文件，随后初始化 PPMS 实例。这时 MATLAB 仍然会报错，但此时尝试手动在命令行中输入相关内容，通过使用 TAB 键能够发现 MATLAB 已经能够识别出 dll 中的内容。这时再初始化 PPMS 实例仍然会得到报错，必须在命令行中输入调用 dll 中的任意对象，比如输入 QuantumDesign.QDInstrument.QDInstrumentType.DynaCool 后回车，随后再初始化 PPMS 实例即可成功。以下为加载 dll 与控制 PPMS 的代码示例，完整的 PPMS 控制代码附在 D.2 中。

```
1 % Example of using class PPMS
2 NET.addAssembly('path\to\the\file.dll');      % load the dll
3 ppms = PPMS;                                % This will get error message. See the ...
   constructor for more parameters
4 QuantumDesign.QDInstrument.QDInstrumentType.DynaCool; % nothing ...
   happened, but required
5 ppms = PPMS;                                % This time it should work
6 ppms.temp                                     % query and print the temperature
```

```

7 ppms.field           % query and print the magnetic field
8 ppms.tempStatus      % query the temperature status. It will ...
    return a QuantumDesign.QDInstrument.TemperatureStatus object
9 ppms.tempStatusStr   % query the temperature status. It ...
    returns a string, such as 'Chasing', 'Stable', etc.
10 ppms.fieldStatus
11 ppms.fieldStatusStr % similar as for temperature status
12 ppms.setTemp(300,'tempRate',10,'tempApproach','FastSettle');      % ...
    set temperature
13 ppms.setField(1000,'fieldRate',100,'fieldApproach','Linear');      % ...
    set field

```

有了以上测量程序的编写，即可完全通过测量电脑询问与控制 PPMS 的温度与磁场，以及调节 VNA 的相关参数并取得 VNA 的扫描数据。更进一步的，通过 Windows 的远程桌面可以远程连接到测量电脑，从而使测量变得更为便捷。实际中我们只需要在取出和放入样品时到 PPMS 设备附近，其余时间均远程进行控制与测量。

3.3 测量系统测试

对于上述 PPMS 测量系统，我利用自己编写的测量程序利用现有常见的 CPW 谐振腔进行了一系列测量测试，说明测量系统工作良好。

3.3.1 频率扫描

3.3.2 温度扫描

3.3.3 磁场扫描

3.4 样品托的设计与优化

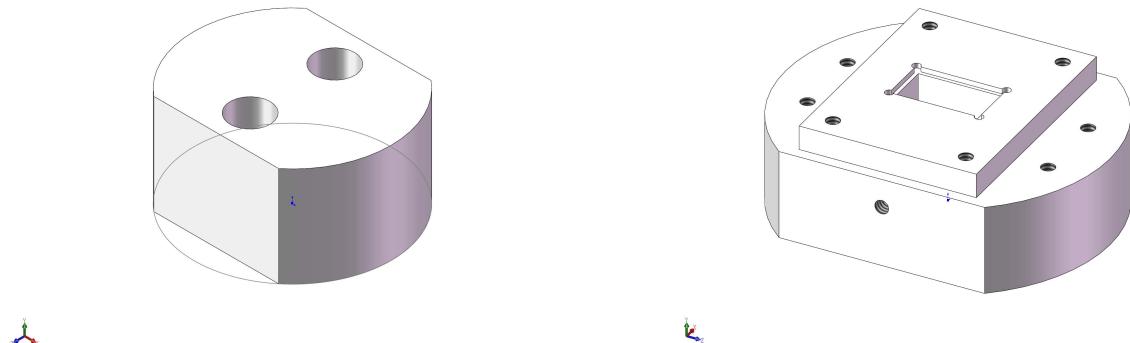


图 3.4 原有样品盒盖子与底座设计



图 3.5 改进后样品盒盖子与底座设计

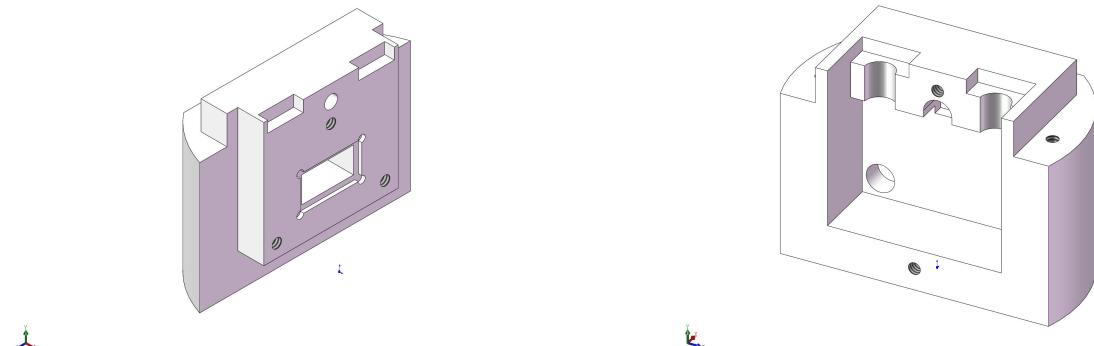


图 3.6 为竖直放样品所设计的样品盒底座拆分出的盖子与底座设计

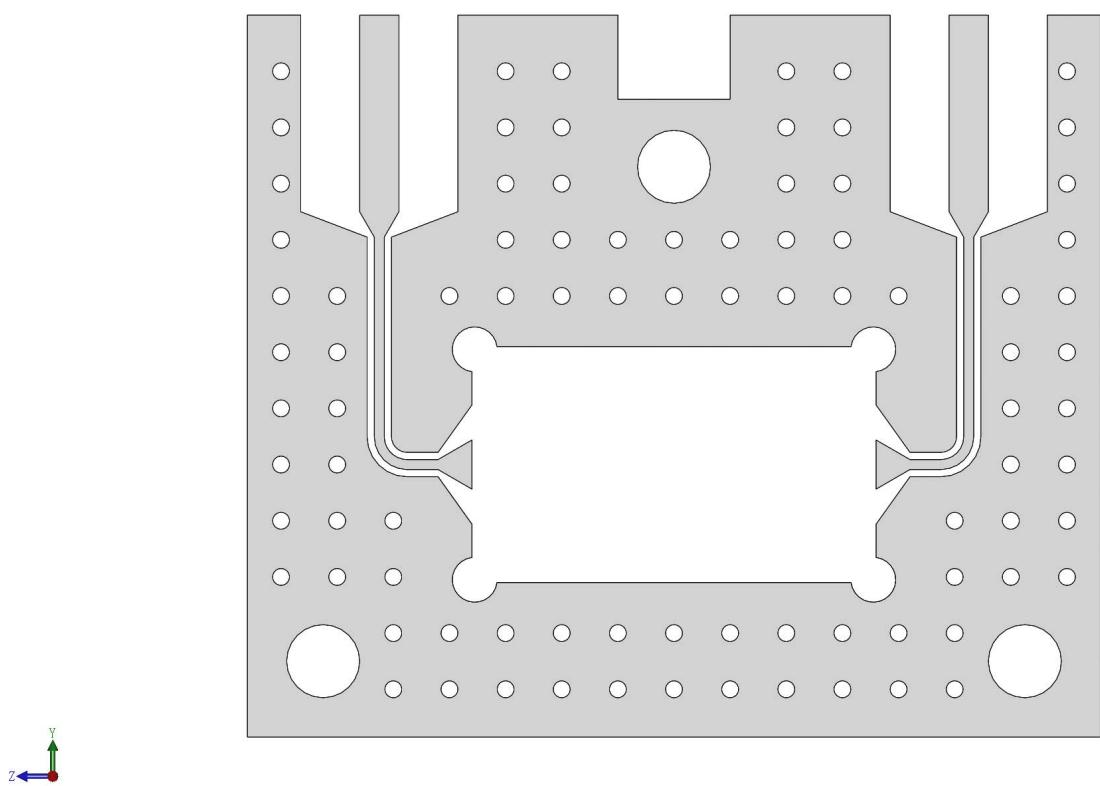


图 3.7 为新的竖直放置样品的样品盒所设计的 PCB 板

第4章 2.5维谐振腔的制备与测量

4.1 器件测量与制备工艺概述

由于 2.5 维谐振腔的电容部分由上下两层金属以及中间的介电层组成，工艺较为复杂，需要至少三步光刻来完成。而细小的电感部分则需要电子束曝光 (EBL) 来定义形状，再由蒸发镀膜完成。因此，总的制备工艺大致如下：

1. 通过光刻，磁控溅射蒸镀金属 Nb，Lift off 三步制作平面波导与三层电容的第一层
 2. 通过 ALD 生长 Al_2O_3 ，或通过 PECVD 生长 SiO_2 或 SiN_x 作为电容三层结构中的介电层
 3. 光刻定义掩膜后通过 ICP 刻蚀或湿法刻蚀介电层
 - (a) 显影后剩余光刻胶应完全盖住三层电容的第一层金属以及上方的介电层
 - (b) 刻蚀后剩余的介电层仅遮盖住三层电容的底层，其余电介质需被刻蚀以点焊与继续制作其他结构
 - (c) 去除介电层上方残存的光刻胶
 4. 利用光刻制作出三层电容的顶层图案并蒸镀金属，对准精度约 1 微米，Lift off
 5. 利用 EBL 制作微小电感图案
 - (a) 电感图案将与三层电容的上下级板相连
 - (b) 对准精度 100-1000 nm
 6. 旋涂光刻胶，将器件切为 4mm x 7mm 大小，清洁器件
 7. 通过点焊将器件与 PCB 板线路相连接
- 每一步制备工艺的具体步骤与相关参数在附录B中给出。

4.2 光刻板的设计与改进

由于制备工艺需要三步光刻，对于完整制备一个器件来说，需要的光刻板的图案为一组三个，分别对应4.1制备工艺概述中的前三步。第一步完成二维平

面波导以及接地平面的制作，以及三维电容的第一层金属。第二步在生长介电材料后通过光刻制作掩膜覆盖住需要保留的电容的第二层介电层部分，并刻蚀掉没有被覆盖掉的电介质。第三部制作出电容最上层的金属。最后一步通过电子束曝光与蒸镀制作微小的电感部分，并各自与三维电容的上、下两层连接起来。光刻的三个步骤所需的模板作图如图4.1所示

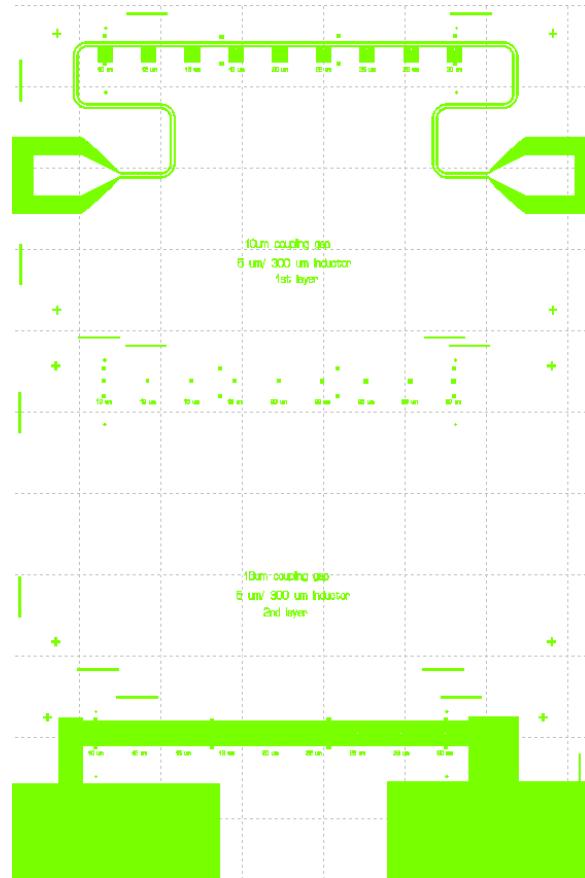


图 4.1 制备一个器件所需的三个光刻步骤对应的一组三个模板图案

由于没有这类谐振腔的制备经验，对于其频率的估算并没有太多把握，因此我们希望有尽可能多的谐振频率与设计相对应的数据，来辅助下一步对仿真结果的修正与器件制备的改进。由于对电容和电感估值的准确度均有待确认，我们需要至少两组不同的器件设计的组合，一组固定 L 变化 C 的大小，另一组固定 C 变化 L 的大小，这样可以通过拟合确定出每个器件的 L 与 C 的值。因此，一个二维平面波导可以与多个谐振腔耦合，可方便测量。另一方面，考虑到电子束曝光难度与耗时均较大，我们打算先尝试中等数值的电容与电感组合，使电

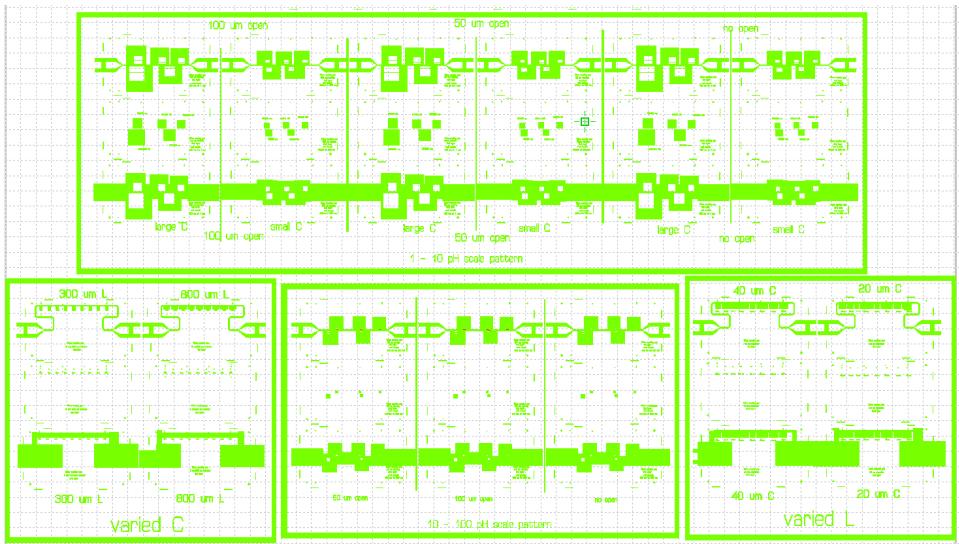


图 4.2 完整的光刻板图案

感的尺寸能够通过光刻制备，这样即可在三层电容制备的最后一步同时制作出电感，省去了电子束曝光制备电感的步骤，大大加快样品制备与测试速度。解决三层电容的制备后，再通过电子束曝光制作电感。

综合上述讨论，我们总共设计了若干种不同的模板几何形状，如图4.2所示，覆盖了较大范围的电容与电感值，为第一次摸索器件制备工艺以及尝试性测量提供较大的频率变化区间，尽可能保证能够测到谐振腔的共振频率。

4.3 器件制备情况与改进

按照上述计划，我进行了器件制备的工作，目前已完成了两轮完整的器件制备流程。

在第一轮器件制备过程中，发现了许多问题，并依次进行了分析与解决。第一轮器件制备的器件图像如图4.3所示。

由于 PPMS 测量系统仅可放下大小为 $4\text{mm} \times 7\text{mm}$ 的器件，我们最初即采用该大小的硅片作为衬底开始进行微纳加工。由于器件尺寸较小，在旋涂光刻胶的过程中导致胶厚分布不均，因此光刻第一层图案后效果明显不好，如图4.3(a)所示。图4.3(a) 中绿色部分为显影后剩余的光刻胶，也即不会被蒸镀上金属的部分。靠上的两个长条形的绿色部分即为平面波导的中心线与两侧接地部分间的间隙，而下面大块绿色部分包围着的即应该为三维电容结构的第一层。

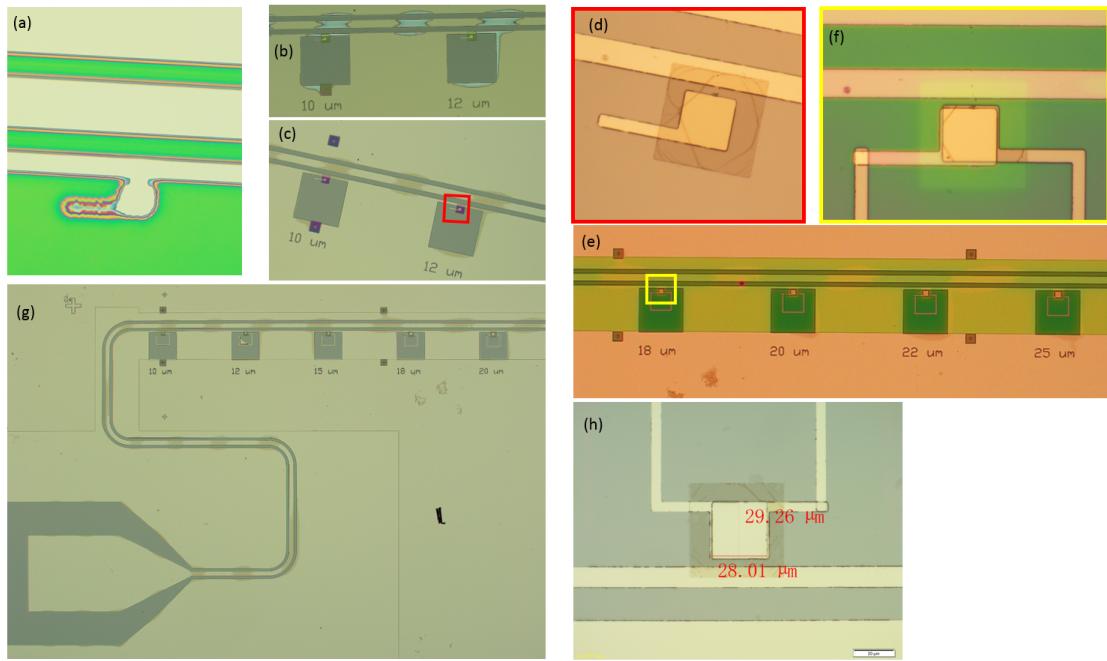


图 4.3 第一轮器件制备过程

由于光刻胶厚度不均匀，曝光过程中发生了衍射，导致图案变形。因此，我们改用 8mm x 8mm 左右大小的硅片进行微纳加工，再在最后点焊前将器件切割为 PPMS 所要求的大小。进行这步改进后，光刻效果很好，器件的第一层金属顺利完成，对应第4.1节中的步骤 1。

器件制备的第二步，也即生长介电层的步骤较为简单，利用超净间内现有的 PECVD 或 ALD 的相关程序即可自动完成。第一轮器件制备选用的介电层为 PECVD 生长的 30nm 的 SiO₂。在进行第三步，也即第二层光刻的显影阶段时，我观察到了器件上有淡蓝色的杂质，疑似为未清洗干净的被曝光的光刻胶，如图4.3(b) 所示，其中浅绿色的小方块即为理应存在剩余光刻胶的部分，成功观察到了完整盖住三维电容第一层金属以及上方介电层的光刻胶。分析后我们认为，残存的光刻胶可能导致该处剩余未被刻蚀的介电层，对器件不会产生本质影响，因此我们暂时忽略这一步的不理想，继续器件制备。进行 ICP 刻蚀后，器件如图4.3(c) 所示，可以看见电容处的电介质与残留的光刻胶整体呈现深紫色，而图4.3(b) 中的淡蓝色部分仍旧为灰色阴影，应为未刻蚀干净的介电层。图4.3(c) 中还能看到其他两块紫色区域，为第二层光刻对准第一层结构所用的标记符号，因为也没被曝光所以留下了介电层与光刻胶，从而呈紫色。在最后去除介电层上方残留光刻胶时，使用了常用的 S1805 光刻胶去胶液 NMP，但去胶后观察到

了剩余的介电层表面有不明灰色纹路，如图4.3(d) 所示。图4.3(d) 即图4.3(c) 中红色矩形部分去胶后的放大图。这些灰色纹路疑似为未被去净的光刻胶，因此在第二轮器件制备过程中，我们改用 PG remover 去胶，得到了更好的效果。

第二、三步完成后，进行最后一步的制作电容的顶层金属。首先进行顶层图案的光刻，光刻的效果如图4.3(e) 与 (f) 所示，其 (f) 为 (e) 中黄色矩形的放大图。可以看出，顶层与底层间有一定的对准误差。需要注意，在将电感部分与顶层金属和底层金属连接上时，需先通过 Argon milling 去除底层金属表面的氧化层，随后磁控溅射 Nb，才能达到良好的电接触。蒸镀完成后，即可去胶使设计的金属结构存留。此时因为与上下两层电容金属极板相连的电感在平面内围成闭合区域，去胶液难以进入。正常情况下可以尝试超声，但目前器件上有较薄的介电层，容易在超声过程中脱落，因此我们小心尝试了超声的功率与持续时间，最终在保持介电层完整的条件下达到了较好的去胶效果，如图4.3(g) 所示。可以看到，绝大部分应被去掉的金属均成功被去掉，除了第二个谐振腔中电感部分的左下角内仍有一小块金属。

至此，器件制备基本结束，制备出的 2.5 维谐振腔如图4.3(h) 所示，电容的下层与上层极板间的对准误差大概在 1um，与预期相符。灰色方形部分为上下极板间的介电层，细条形为电感。最后旋涂光刻胶，切片并清洗后，即可进行点焊，并放入 PPMS 进行测量。然而对于第一批制备出的器件并没有测到明显的谐振腔模式。我们分析后认为原因可能有以下两方面：一方面，由于介电材料相对于空气而言有损耗，谐振腔的 Q_i 比常见传输线谐振腔更小，而由于光刻板的设计使谐振腔与传输线中的场间有一段距离的接地平面，有一定屏蔽作用，导致谐振腔与传输线中的模式耦合较弱， Q_c 较大，综合这两点，预计谐振腔响应的峰形较浅较宽，可能被背景噪声覆盖；另一方面，器件制备过程中最上层与最下层的对齐误差导致电容偏小，谐振腔频率偏高，可能使谐振频率超出 PPMS 与 VNA 测量系统的正常测量范围 1 ~ 10GHz。因此，我们下一步改良了光刻板的设计，增大谐振腔与传输线的耦合强度从而增强谐振腔响应的信号。同时开始第二批器件的制备，采用谐振频率变化范围更广的器件组以使得有谐振腔的频率落在可测范围内的概率更大。

第二批器件的制备过程如图所示。

4.4 谐振腔响应信号的拟合

插图索引

1.1	金刚石色心结构示意图, 其中碳空位由 V 表示, 氮掺杂由 N 表示 ^[29] 。3
1.2	金刚石色心在考虑零场劈裂与外加磁场后的能级示意图。其中 D 为零场劈裂导致的能级分裂, $m_s = \pm 1$ 的两个态之间的能量差来源于应力与局域电场。不同核自旋量子数 m_I 的态之间的能级分裂来源于超精细相互作用 ^[29] 。 4
1.3	N 个金刚石自旋构成的自旋系综的能级示意图 ^[29] 。可以看见 $S \neq N/2$ 的态均为高度简并态。 6
2.1	新型三维谐振腔的几何结构与电流, 磁场分布示意图 ^[24] 7
2.2	新型三维谐振腔仿真的几何设计(透视图) 8
2.3	新型三维谐振腔磁场在自旋系综所处空间横截面上的大小分布 8
2.4	仿真所得磁场空间分布截面图 10
2.5	文献 ^[19] 中所示磁场空间分布截面图 10
2.6	仿真所得磁场在位于距金属上表面 $0.1\mu m$ 处水平截线上的分布 10
2.7	文献 ^[20] 中所示磁场在位于距金属上表面 $0.1\mu m$ 处水平截线上的分布 10
2.8	基于我们制备的二维平面波导的常见尺寸进行仿真所得的磁感应强度分布图。 11
2.9	来自参考文献 [27] 的 2.5 维谐振腔的设计(右图中红色虚线部分的局部放大图)。 13
2.10	来自参考文献 [27] 的 2.5 维谐振腔的设计(整体)。 13
2.11	2.5 维谐振腔的螺旋状电感处水平截面上的磁感应强度分布。 13
2.12	第一次改进后的 2.5 维谐振腔的电感处水平截面上的磁感应强度分布。 14

2.13	第二次改进后的 2.5 维谐振腔的电感处水平截面上的磁感应强度分布，电感部分的电感大小得到了降低。	14
2.14	第三次改进后的 2.5 维谐振腔的电感处水平截面上的磁感应强度分布，通过加厚材料并加粗宽度提升了临界电流大小，并在不影响磁场较强区域的强度的基础上改进了设计以提高制备的成功率。	15
3.1	PPMS DynaCool 测量设备实物图 ^[37]	16
3.2	PPMS DynaCool 腔体内部结构 ^[37]	17
3.3	样品杆与样品盒。其中红色矩形部分为样品杆末端微波信号的输入与输出端，黄色矩形部分为样品盒所处位置，蓝色矩形为样品盒以及拆卸后的样品盒。绿色矩形内为样品盒底座上的 PCB 板，SMP 接头以及器件及其放大图	18
3.4	原有样品盒盖子与底座设计	22
3.5	改进后样品盒盖子与底座设计	22
3.6	为竖直放样品所所设计的样品盒底座拆分出的盖子与底座设计	22
3.7	为新的竖直放置样品的样品盒所设计的 PCB 板	23
4.1	制备一个器件所需的三个光刻步骤对应的一组三个模板图案	25
4.2	完整的光刻板图案	26
4.3	第一轮器件制备过程	27

表格索引

公式索引

公式 1-1	2
公式 1-2	3
公式 1-3	4
公式 1-4	4
公式 1-5	5
公式 1-6	5
公式 1-7	5
公式 1-8	5
公式 1-9	5
公式 1-10	5
公式 1-11	6
公式 1-12	6
公式 2-1	8
公式 2-2	9
公式 2-3	9
公式 2-4	9
公式 2-5	10
公式 2-6	10
公式 2-7	10
公式 2-8	11
公式 2-9	12

公式 2-10	12
公式 2-11	12

参考文献

- [1] Feynman R P. Simulating physics with computers. *International journal of theoretical physics*, 1982, 21(6):467–488
- [2] Nielsen M A, Chuang I. *Quantum computation and quantum information*, 2002
- [3] Bennett C. Quantum cryptography: Public key distribution and coin tossing. *Proc. of IEEE Int. Conf. on Computers, Systems and Signal Processing*, 1984, 1984
- [4] Grover L K. A fast quantum mechanical algorithm for database search. *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. ACM, 1996. 212–219
- [5] Shor P W. Algorithms for quantum computation: Discrete logarithms and factoring. *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*. IEEE, 1994. 124–134
- [6] DiVincenzo D P, et al. The physical implementation of quantum computation. *arXiv preprint quant-ph/0002077*, 2000.
- [7] Loss D, DiVincenzo D P. Quantum computation with quantum dots. *Physical Review A*, 1998, 57(1):120
- [8] Häffner H, Roos C F, Blatt R. Quantum computing with trapped ions. *Physics reports*, 2008, 469(4):155–203
- [9] Gershenfeld N A, Chuang I L. Bulk spin-resonance quantum computation. *science*, 1997, 275(5298):350–356
- [10] Devoret M H, Schoelkopf R J. Superconducting circuits for quantum information: an outlook. *Science*, 2013, 339(6124):1169–1174
- [11] Kok P, Munro W J, Nemoto K, et al. Linear optical quantum computing with photonic qubits. *Reviews of Modern Physics*, 2007, 79(1):135
- [12] Dutt M V G, Childress L, Jiang L, et al. Quantum register based on individual electronic and nuclear spin qubits in diamond. *Science*, 2007, 316(5829):1312
- [13] Imamoğlu A. Cavity qed based on collective magnetic dipole coupling: Spin ensembles as hybrid two-level systems. *Physical Review Letters*, 2009, 102(8):083602. PRL
- [14] Wesenberg J H, Ardavan A, Briggs G A D, et al. Quantum computing with an electron spin ensemble. *Physical Review Letters*, 2009, 103(7):070502. PRL
- [15] Schuster D I, Sears A P, Ginossar E, et al. High-cooperativity coupling of electron-spin ensembles to superconducting cavities. *Physical Review Letters*, 2010, 105(14):140501. PRL
- [16] Kubo Y, Ong F R, Bertet P, et al. Strong coupling of a spin ensemble to a superconducting resonator. *Physical Review Letters*, 2010, 105(14):140502. PRL

- [17] Marcos D, Wubs M, Taylor J M, et al. Coupling nitrogen-vacancy centers in diamond to superconducting flux qubits. *Physical Review Letters*, 2010, 105(21):210501. PRL
- [18] Zhu X, Saito S, Kemp A, et al. Coherent coupling of a superconducting flux qubit to an electron spin ensemble in diamond. *Nature*, 2011, 478(7368):221–224
- [19] Mark J, Thomas H, María José M P, et al. Coupling single-molecule magnets to quantum circuits. *New Journal of Physics*, 2013, 15(9):095007
- [20] Bienfait A, Pla J J, Kubo Y, et al. Reaching the quantum limit of sensitivity in electron spin resonance. *Nat Nano*, 2016, 11(3):253–257
- [21] Bienfait A, Pla J J, Kubo Y, et al. Controlling spin relaxation with a cavity. *Nature*, 2016, 531(7592):74–77
- [22] Benningshof O W B, Mohebbi H R, Taminiau I A J, et al. Superconducting microstrip resonator for pulsed esr of thin films. *Journal of Magnetic Resonance*, 2013, 230:84–87
- [23] Mohebbi H R, Benningshof O W B, Taminiau I A J, et al. Composite arrays of superconducting microstrip line resonators. *Journal of Applied Physics*, 2014, 115(9):094502
- [24] Angerer A, Astner T, Wirtitsch D, et al. Collective strong coupling with homogeneous rabi frequencies using a 3d lumped element microwave resonator. *Applied Physics Letters*, 2016, 109(3):033508
- [25] Jenkins M D, Naether U, Ciria M, et al. Nanoscale constrictions in superconducting coplanar waveguide resonators. *Applied Physics Letters*, 2014, 105(16):162601
- [26] Eichler C, Sigillito A J, Lyon S A, et al. Electron spin resonance at the level of 10^4 spins using low impedance superconducting resonators. *Physical Review Letters*, 2017, 118(3):037701. PRL
- [27] Sarabi B, Huang P, Zimmerman N M. Prospective two orders of magnitude enhancement in direct magnetic coupling of a single-atom spin to a circuit resonator. *arXiv preprint arXiv:1702.02210*, 2017.
- [28] Koch J, Terri M Y, Gambetta J, et al. Charge-insensitive qubit design derived from the cooper pair box. *Physical Review A*, 2007, 76(4):042319
- [29] Grèzes C, Kubo Y, Julsgaard B, et al. Towards a spin-ensemble quantum memory for superconducting qubits. *Comptes Rendus Physique*, 2016, 17(7):693–704
- [30] Tavis M, Cummings F W. Exact solution for an n-molecule-radiation-field hamiltonian. *Physical Review*, 1968, 170(2):379
- [31] Schuster D I. Circuit quantum electrodynamics[D]. 2007
- [32] Krantz P. Investigation of transmon qubit designs: a study of plasma frequency predictability. Chalmers University of Technology, 2010
- [33] Kelly J S. Fault-tolerant superconducting qubits. 2015
- [34] Weber S J, Murch K W, Kimchi-Schwartz M E, et al. Quantum trajectories of superconducting qubits. *Comptes Rendus Physique*, 2016, 17(7):766–777

- [35] Haikka P, Kubo Y, Bienfait A, et al. Proposal for detecting a single electron spin in a microwave resonator. *Phys. Rev. A*, 2017, 95:022306
- [36] Tosi G, Mohiyaddin F A, Huebl H, et al. Circuit-quantum electrodynamics with direct magnetic coupling to single-atom spin qubits in isotopically enriched ^{28}Si . *AIP Advances*, 2014, 4(8):087122
- [37] Quantum Design, Inc. Ppms dynacool[EB/OL]. [2017-5-24]. <http://www.qd-china.com/products2.aspx?id=45>
- [38] Khalil M S, Stoutimore M J A, Wellstood F C, et al. An analysis method for asymmetric resonator transmission applied to superconducting devices. *Journal of Applied Physics*, 2012, 111(5):054510
- [39] Megrant A, Neill C, Barends R, et al. Planar superconducting resonators with internal quality factors above one million. *Applied Physics Letters*, 2012, 100(11):113510
- [40] Bruno A, Lange G d, Asaad S, et al. Reducing intrinsic loss in superconducting resonators by surface treatment and deep etching of silicon substrates. *Applied Physics Letters*, 2015, 106(18):182601
- [41] Astner T, Nevlacsil S, Peterschofsky N, et al. Coherent coupling of remote spin ensembles via a cavity bus. *Physical Review Letters*, 2017, 118(14):140502. PRL
- [42] Kubo Y, Diniz I, Dewes A, et al. Storage and retrieval of a microwave field in a spin ensemble. *Physical Review A*, 2012, 85(1):012333. PRA
- [43] Wu H, George R E, Wesenberg J H, et al. Storage of multiple coherent microwave excitations in an electron spin ensemble. *Physical Review Letters*, 2010, 105(14):140503. PRL

致 谢

衷心感谢导师 xxx 教授和物理系 xxx 副教授对本人的精心指导。他们的言传身教将使我终生受益。

在美国麻省理工学院化学系进行九个月的合作研究期间，承蒙 xxx 教授热心指导与帮助，不胜感激。感谢 xx 实验室主任 xx 教授，以及实验室全体老师和同学们的热情帮助和支持！本课题承蒙国家自然科学基金资助，特此致谢。

感谢 THUTHESIS，它的存在让我的论文写作轻松自在了许多，让我的论文格式规整漂亮了许多。

声 明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名: _____ 日 期: _____

附录 A 超导量子比特的原理

附录 B 微纳加工工艺

B.1 光刻

1. Spin coating S1805, 500 nm
 - (a) 500rpm, 10s, 100rpm/s
 - (b) 4000rpm, 45s, 2000rpm/s
 - (c) 0rpm, 5s, 2000rpm/s
2. Bake at 115 degree C for 1 min
3. Align mask
4. Expose with 405 nm UV light for 9.5s, power 350W
5. Develop in MF-319 for 45s
6. Clean in DI-water for 1min and blow dry

B.2 介电层生长与刻蚀

1. PECVD 生长 SiO_2 速率: 70nm/min
2. PECVD 生长 SiN_x 速率: 16nm/min
3. ALD 生长 Al_2O_3 速率:
4. MF-319 刻蚀 Al_2O_3 速率:
5. MF-319 刻蚀光刻胶速率:
6. ICP 刻蚀 SiO_2 速率: 43.3nm/min
7. ICP 刻蚀 SiN_x 速率: 50nm/min

B.2.1 刻蚀介电层后去除光刻胶

1. 置于 PG Remover 中加热至 85 摄氏度, 将烧杯口用铝箔纸盖住, 保持 85 度 30 分钟
2. 用丙酮, IPA 清洗, 氮气枪吹干

B.3 光刻胶 Lift off

1. 于 NMP 去胶液或丙酮中浸泡一晚，用铝箔纸盖住烧杯口
2. 加热至 85 摄氏度 15 至 30 分钟，用铝箔纸盖住烧杯口
3. 用镊子夹住片子后用胶头滴管于去胶液中吹洗
4. 对于难以去除的区域，可尝试超声
5. 对于生长了介电层后的器件，超声时应小心勿将介电层超声至脱落。对于 30nm 的 SiO_2 或 30nm 的 SiN_x ，可用 60% 功率，45kHz 超声 10 秒，重复三次，即可去除对 2.5 维谐振腔样品可能出现的 lift off 不完全的情况，并保证介电层完整
6. 目前尝试过用 100% 功率，45kHz 超声 15 秒，仍可保证介电层完整
7. 从去胶液中取出后迅速放入丙酮中清洗，随后再放入 IPA 中清洗，最后用氮气枪吹干

B.4 磁控溅射镀膜

1. Transfer to sputter chamber
2. chamber -> Process
3. open Ar valve, 100 sccm
4. set DC parameter, $I = 100\text{mA}$
5. DC -> on. If error, try with shutter open
6. set $I = 400\text{mA}$
7. set Ar to 6sccm
8. set chrono end action to DC & RF off and sputter shutters off
9. Holder motion: Rotate
10. open shutter and timing with chrono
11. 25nm/min for sputtering Nb, hence 4min total time for 100nm Nb
12. wait until end
13. set Ar to 0sccm, close Ar valve
14. Stop rotation, Chamber -> pump
15. Holder motion -> transfer, start
16. Transfer after pressure stable and chip cool

B.5 Argon milling 去除氧化层

1. transfer to oxid chamber
2. Process-> IonGun
3. Holder motion -> Target -> IonGun, start. Might error, start again
4. Open Ar valve, set to 6sccm
5. set Vbeam = 250V, Ibeam = 10mA, Vacc = 60V. Input parameters again even if they are already correct
6. Discharge on, wait stable
7. Beam on, wait 2min
8. set Vbeam = 400V, Ibeam = 20mA, Discharge 40V, Vacc = 60V, Iacc = 1.1mA, wait stable
9. ordinary status: pressure 1.2e-4Torr, cathode 7.1V, 6.22A, 40V/0.23A, 399V/20.1mA, 60V/1.1mA
10. Set chrono end action to IonGun off
11. open shutter and timing with chrono, 3min30s
12. wait until end
13. close shutter
14. Ar off, Ar valve off
15. oxid chamber -> pump
16. Holder motions -> Transfer, start
17. transfer after pressure stable

B.6 点焊

1. 一焊由于 PCB 上的 SMP 接头的空间位置原因需位于器件上，二焊位于 PCB 板上，焊接使用铝线
2. 一焊参数：功率 220，时间 30ms，力 19
3. 二焊参数：功率 330，时间 40ms，力 19
4. 点焊时应使焊线尽可能地短而密

附录 C PPMS 系统的常用操作

对于 PPMS 的详细介绍可参阅 PPMS 的用户手册，本附录在读者熟悉 PPMS 相关术语与参数（如 Chamber Pressure 等）的前提下，为使用者提供一个方便参考的操作流程。

在正常的测量状态下，Chamber Pressure 与样品室温度有关，样品温度为 2K 左右时一般为 300 至 400mTorr，chamber state 应该为 Purged，意为样品室已用 He 气清洗并抽真空。在非远程控制 PPMS 时，一般通过与 PPMS 相连的电脑上的 MultiVu 程序控制 PPMS。该程序主面板下方有若干小面板，点击这些小面板可调出设定温度，磁场与样品室状态的分面板。常用的分面板也即前文提到的 Temperature，Chamber，Field 面板。

C.1 调节温度与磁场

在 Temperature 面板中，status 显示当前样品温度。通过 Control 一栏即可设置目标温度（set point）以及调温的速率（Rate）。Mode 选项有 Fast settle 和 No overshoot 两种，也即快速和非振荡的平稳到达目标温度两种模式。

升温时最大速率不应超过 20K/min，降温时最大速率不应超过 10K/min。

调节磁场与调节温度十分类似。

C.2 更换样品

首先升温至室温，300K，从 2K 开始升温则需约 30 分钟。样品室升至常温后，样品室气压约为 20 40Torr。点击 Chamber 分面板的 Vent/Seal，此时样品室气压将快速升至常压，约 790Torr，此后即可取出样品杆。若取出样品杆时间较长，应插入空样品杆并点击 Chamber 分面板的 Purge/Seal 以使样品室处于较好的密封环境。

更换完样品并插入封好样品杆后，点击 Chamber 分面板的 Purge/Seal，此时 state 变为 Purging。样品室气压将在常压与 30Torr 间来回反复若干次，约 5 分钟后 state 变为 Purged，此时即可开始制冷。

附录 D 测量系统 MATLAB 代码

通过 MATLAB 控制仪器，能够十分方便地调整仪器的各项参数以及从仪器采集所需数据。对于封装较好的代码，能够可扩展地编写与控制更为复杂的实验。因此我通过 MATLAB 实现了对 VNA 的控制与数据采集，基于 PPMS 仪器商提供的动态链接库文件实现了对 PPMS 系统的状态读取与控制。在这二者的基础上，编写了扫描不同温度与磁场下的频率响应的实验的代码。

本附录中将给出 VNA 与 PPMS 的 MATLAB 控制程序代码，以及扫描温度与磁场的 MATLAB 代码。

D.1 VNA 控制代码

```
1 classdef E5071C < handle
2 % E5071C describe and control the agilent E5071C ENA
3 %
4 % EXAMPLES (assume instance named 'vna'):
5 % initialization:
6 %     vna = E5071C;
7 %     vna = E5071C('address',8,'InputBufferSize',100000);
8 % set & get parameters
9 %     % fetch & return the start frequency:
10 %         freq = vna.freqStart;
11 %         % fetch & return the IFBW:
12 %         bw = vna.ifbw;
13 %         % set the stop frequency to 5GHz:
14 %         vna.freqStop = 5e9;
15 %         % set the number of average and turn on averaging:
16 %         vna.avg = 999;
17 % fetch trace:
18 %     % fetch the trace data, return a structure
19 %     trace = vna.trace
20 %
21 % See E5071C/plotTrace, E5071C/fit and E5071C/manualSweep for ...
22 % detail usage
23 % Wentao, April 2017
```

```

24 %
25
26 properties (Constant)
27     MAX_POINTS = 1601;
28     MAX_AVG = 999;
29
30 end
31
32 properties
33     visa
34     InputBufferSize
35     TimeOut
36
37     address
38
39 % properties with set & get methods
40 freqStart
41 freqStop
42 freqSpan
43 freqCenter
44 avg
45 numOfPoints
46 ifbw
47 meas
48 outp
49 power
50 trigMode
51
52 trace
53
54 freqs
55 h_fig      % figure handle
56 end
57
58 methods
59
60 function obj = E5071C(varargin)
61     % Initialize E5071C object
62     %
63     p = inputParser;
64     p.addParameter('address',6, @isnumeric);           % GPIB ...
65             address
66     p.addParameter('InputBufferSize',30000, @isnumeric);
67     p.addParameter('TimeOut',20, @isnumeric);
68     p.parse(varargin{:});
       expandStructure(p.Results);

```

```

69
70     obj.address = address;
71     obj.TimeOut = TimeOut;
72     obj.InputBufferSize = InputBufferSize;
73
74     obj.visa = ...
75         visa('agilent',sprintf('GPIB::%d::INSTR',obj.address));
76         fprintf('%s\nConnected.\n',obj.read('*IDN?','%s'));
77         set(obj.visa,'InputBufferSize', obj.InputBufferSize);
78         set(obj.visa,'TimeOut', obj.TimeOut);
79         % Set byte order to swapped (little-endian) format
80         fprintf('Set byte order to little-endian...');
81         obj.write(':FORMAT:BORD SWAP');
82         fprintf('Done.\n')
83         % Set data type to real 64 bit binary block
84         fprintf('Set data type to real 64 bit binary block...');
85         obj.write(':FORMAT:DATA REAL');
86         fprintf('Done.\n');
87
88     function delete(obj)
89         delete(obj.visa);
90     end
91
92     %% frequency set & get
93     function value = get.freqStart(obj)
94         value = obj.read(':sens:freq:star?', '%f');
95     end
96     function set.freqStart(obj,val)
97         obj.write(':sens:freq:star %f',val);
98     end
99     function value = get.freqStop(obj)
100        value = obj.read(':sens:freq:stop?', '%f');
101    end
102    function set.freqStop(obj,val)
103        obj.write(':sens:freq:stop %f',val);
104    end
105    function value = get.freqCenter(obj)
106        value = obj.read(':sens:freq:cent?', '%f');
107    end
108    function set.freqCenter(obj,val)
109        obj.write(':sens:freq:cent %f',val);
110    end
111    function value = get.freqSpan(obj)
112        value = obj.read(':sens:freq:span?', '%f');
113    end

```

```

114     function set.freqSpan(obj, val)
115         obj.write(':sens:freq:span %f', val);
116     end
117
118     %% sweep setup: measurement parameter, points, average, ifbw
119     function value = get.meas(obj)
120         value = obj.read(':CALC:PAR:DEF?', '%s');
121     end
122     function set.meas(obj, val)
123         obj.write(':CALC:PAR:DEF %s', val);
124     end
125     function value = get.numOfPoints(obj)
126         value = obj.read(':sens:swe:poin?', '%f');
127     end
128     function set.numOfPoints(obj, val)
129         obj.write( ':sens:swe:poin %d', val);
130     end
131
132     function value = get.avg(obj)
133         value = obj.read( ':sens:aver:count?', '%f');
134     end
135     function set.avg(obj, val)
136         obj.write( ':sens:aver:count %d', val);
137         obj.write(':SENSe:AVERage:STATE 1');
138     end
139     function clearAvg(obj)
140         obj.write(':SENSe:AVERage:CLE');
141     end
142
143     function value = get.ifbw(obj)
144         value = obj.read(':sens:BWID:RES?', '%f');
145     end
146     function set.ifbw(obj, val)
147         obj.write( ':sens:BWID:RES %f', val);
148     end
149
150     function val = sweepTime(obj)
151         val = obj.read('SENS:SWE:TIME:DATA?', '%f');
152     end
153
154
155     %% output & trigger
156     function value = get.power(obj)
157         value = ...
158             obj.read(':SOURce:POWER:LEVel:IMMediate:AMPLitude?', ...
159             '%f');

```

```

158     end
159     function set.power(obj, val)
160         obj.write(':SOURce:POWer:LEVel:IMMediate:AMPLitude %d', val);
161     end
162     function value = get.outp(obj)
163         value = obj.read(':OUTP:STATE?', '%f');
164     end
165     function set.outp(obj, val)
166         obj.write(':OUTP:STATE %d', val);
167     end
168     function value = get.trigMode(obj)
169         value = obj.read(':TRIG:SEQ:SOUR?', '%s');
170     end
171     function set.trigMode(obj, val)
172         % set.trigMode sets trigger mode
173         % available options: 'INT', 'EXT', 'MAN', 'BUS'
174         % Internal Trigger
175         % Uses the internal trigger to generate continuous ...
176         % triggers automatically.
177         %
178         % External Trigger
179         % Generates a trigger when the trigger signal is ...
180         % inputted externally via the Ext Trig connector or the ...
181         % handler interface.
182         %
183         % Manual Trigger
184         % Generates a trigger when the key operation of ...
185         % Trigger > Trigger is executed from the front panel.
186         %
187         % Bus Trigger
188         % Generates a trigger when the SCPI.IEEE4882.TRG ...
189         % object is executed.
190
191         obj.write(':TRIG:SEQ:SOUR %s',val);
192     end
193
194     %% set & get configurations
195     function setConfig(obj, config)
196         % setConfig apply parameters to E5071C
197         % config should have same or less fields as E5071C properties
198         % with set & get methods
199         flds = fieldnames(config);
200         for ii = 1:length(flds)
201             fld = flds{ii};
202             obj.(fld) = config.(fld);
203         end

```

```

199     end
200
201     function params = getConfig(obj)
202         % getConfig returns E5071C object parameters for saving
203         % configuration
204         flds = {'freqStart',...
205                 'freqStop',...
206                 'freqSpan',...
207                 'freqCenter',...
208                 'avg',...
209                 'numOfPoints',...
210                 'ifbw',...
211                 'meas',...
212                 'outp',...
213                 'power',...
214                 'trigMode'};
215         for ii = 1:length(flds)
216             fld = flds{ii};
217             params.(fld) = obj.(fld);
218         end
219     end
220
221
222 %% get & plot trace
223 function autoScale(obj)
224     % autoScale auto-scales the y axis
225     % for viewing the image via web server
226     obj.write(':DISP:WIND:TRAC:Y:SCAL:AUTO');
227 end
228 function value = get.freqs(obj)
229     value = obj.freqStart:((obj.freqStop...
230             - obj.freqStart)/obj.numOfPoints):obj.freqStop;
231     value = value(1:end-1);
232 end
233 function value = get.trace(obj)
234     % adopted from https://community.keysight.com/thread/22342
235     fopen(obj.visa);
236     fprintf(obj.visa, 'CALC:DATA:SDAT?');
237     [data, count, msg] = binblockread(obj.visa, 'double');
238     fclose(obj.visa);
239     value.count = count;
240     value.msg = msg;
241     value.X = data(1:2:end);
242     value.Y = data(2:2:end);
243 end
244
```

```

245 function titleStr = plotTrace(obj, varargin)
246     % plotTrace fetch & plot trace data
247     % EXAMPLE (assume the object is named 'vna'):
248     %   vna.plotTrace;
249     %   vna.plotTrace('issavefig', true);
250     %   vna.plotTrace('issavefig', true, 'filename','test');
251     %
252     % See the inputParser below for more options
253     %
254
255     p = inputParser;
256     p.addParameter('issavedata',false,@islogical);
257     p.addParameter('issavefig',false,@islogical);
258     p.addParameter('avg',1,@isnumeric);
259     p.addParameter('filename','','@ischar');
260     p.addParameter('format','png',@ischar);
261     p.parse(varargin{:});
262     expandStructure(p.Results);
263
264     if avg > 1
265         pause(round(avg*obj.sweepTime + 1));
266     end
267
268     hfig = figure;
269     obj.h_fig = hfig;
270     trace = obj.trace;
271
272     plot(obj.freqs/1e9, ...
273          20*log10(abs(trace.X + li*trace.Y)));
274     xlabel freq/GHz
275     ylabel SParameter/dB
276     titleStr = ['start_' num2str(obj.freqStart/1e9) ...
277                 'GHz_stop_' num2str(obj.freqStop/1e9) ...
278                 'GHz_pow_' num2str(obj.power) ...
279                 'dBm_AVG_' num2str(avg)];
280     title(titleStr, 'interpreter','none');
281
282     if issavefig
283         if ~isempty(filename)
284             titleStr = filename;
285         end
286         saveas(hfig, [titleStr format]);
287     end
288
289     if issavedata
290         str = titleStr;

```

```

291         freqs = obj.freqs;
292         config = obj.getConfig;
293         save([str '.mat'],'freqs','trace','str','config');
294     end
295
296     end
297
298     function [freqs, trace] = manualSweep(obj, varargin)
299         % manualSweep defines and does a manual frequency sweep
300         % main purpose is for wide sweep with high resolution for
301         % finding modes
302         %
303         % EXAMPLE:
304         %   [freqs, trace] = ...
305         %   vna.manualSweep('start',1e9,'stop',9e9,'res',1e5);
306         %   [freqs, trace] = ...
307         %   vna.manualSweep('start',3.5e9,'stop',3.6e9,'res',1e4, ...
308         %   'avg',999);
309         %   [freqs, trace] = ...
310         %   vna.manualSweep('center',4.9655e9,'span',1e6,'res',0.001e6,'avg',2,'ifbw',1
311         %
312         %
313         % See the inputParser below for more options
314         %
315
316     p = inputParser;
317     p.addParameter('start',1e9,@isnumeric); % start frequency
318     p.addParameter('stop',8e9,@isnumeric); % stop frequency
319     p.addParameter('res',1e6,@isnumeric); % frequency ...
320         resolution
321     p.addParameter('avg', 1, @isnumeric); % number of ...
322         average
323     p.addParameter('ifbw', 100, @isnumeric); % ifbw of vna
324     p.addParameter('points', obj.MAX_POINTS, @isnumeric); % ...
325         number of points of vna
326     p.addParameter('pow', -100, @isnumeric); % power of vna, ...
327         default below
328             % the lowest ...
329             % power of ...
330             % E5071C,
331             % hence this
332             % parameter only
333             % takes ...
334             % effect if it
335             % is given a ...
336             % valid

```

```

325                                     % value
326 p.addParameter('center', 0, @isnumeric);    % frequency ...
327                                         sweep can also be defined
328                                         % by center and ...
329                                         span,
330                                         % if they are ...
331                                         given a
332                                         % valid value
333 p.addParameter('span',0, @isnumeric);
334 p.addParameter('issavedata',true,@islogical);
335 p.addParameter('hfig',233,@isnumeric);        % figure handle
336 p.addParameter('notes','','@ischar');          % notes to ...
337                                         add in file name
338
339
340 freqSectionSpan = res * points;
341 freqSectionSpan = 1e6 * round(freqSectionSpan/1e6);
342 if span != 0 && center != 0
343     start = center - span/2;
344     stop = center + span/2;
345 end
346 numOfSections = ceil((stop - start)/freqSectionSpan);
347 stop = start + numOfSections * freqSectionSpan;
348 totalPoints = points * numOfSections;
349
350
351 % initialize
352 freqs = NaN(1, totalPoints);
353 trace.X = freqs;
354 trace.Y = freqs;
355
356
357 % apply parameters
358 obj.ifbw = ifbw;
359 if pow > -85
360     obj.power = pow;
361 end
362 obj.numOfPoints = points;
363 obj.freqSpan = freqSectionSpan;
364 obj.avg = obj.MAX_AVG;
365
366 sweepTime = obj.sweepTime;
367 waitTime = ceil(avg * sweepTime) + 1;
368
369 figure(hfig);
370 xlabel freq/GHz

```

```

367     ylabel SParameter/dB
368     str = ...
369         sprintf('start_%.2fGHz_stop_%.2fGHz_res_%.2fMHz_pow_%ddBm_AVG_%d',...
370             start/1e9,stop/1e9,res/1e6,obj.power,avg);
371         title(str,'interpreter','none');
372
373         fprintf('\tsweep from %.2fGHz to %.2fGHz, %d sections, ...
374             %ds per section\n\ttotal points: %d, total time: ...
375             %ds.\n',...
376             start/1e9,stop/1e9, ...
377             numOfSections,waitTime,totalPoints, waitTime * ...
378             numOfSections);
379         for i = 1:numOfSections
380             obj.freqCenter = start + freqSectionSpan/2 + ...
381                 freqSectionSpan*(i-1);
382             fprintf('sweeping %.2fGHz to ...
383                 %.2fGHz...\n',obj.freqStart/1e9,obj.freqStop/1e9);
384             pause(waitTime);
385             tmptrace = obj.trace;
386             tmpfreqs = obj.freqs;
387             freqs((1 + (i-1)*points):(i*points)) = tmpfreqs;
388             trace.X((1 + (i-1)*points):(i*points)) = tmptrace.X';
389             trace.Y((1 + (i-1)*points):(i*points)) = tmptrace.Y';
390             figure(hfig);
391             plot(freqs/1e9, 20*log10(abs(trace.X + li*trace.Y)));
392         end
393         fprintf('Sweep finished!\n');
394
395         figure(hfig);
396         xlabel freq/GHz
397         ylabel SParameter/dB
398         str = ...
399         sprintf('start_%.2fGHz_stop_%.2fGHz_res_%.2fMHz_pow_%ddBm_AVG_%d%s',...
400             start/1e9,stop/1e9,res/1e6,pow,avg,notes);
401         title(str,'interpreter','none');
402
403         if issavedata
404             config = obj.getConfig;
405             save([str '.mat'],'freqs','trace','str','config');
406         end
407         obj.h_fig = hfig;
408
409     end
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
779

```

```

405 function [freqs,totalWaitTime] = manualSweepFreqs(obj, varargin)
406 % manualSweepFreqs quickly calculate frequencies of
407 % manualSweep, does not do the sweep
408 %
409 % ATTENTION: this method will modify the vna sweep frequency!
410 %
411 % See the inputParser below for more options
412 %
413
414 p = inputParser;
415 p.addParameter('start',1e9,@isnumeric); % start frequency
416 p.addParameter('stop',8e9,@isnumeric); % stop frequency
417 p.addParameter('res',1e6,@isnumeric); % frequency ...
418 % resolution
419 p.addParameter('avg', 1, @isnumeric); % number of ...
420 % average
421 p.addParameter('ifbw', 100, @isnumeric); % ifbw of vna
422 p.addParameter('points', obj.MAX_POINTS, @isnumeric); % ...
423 % number of points of vna
424 p.addParameter('pow', -100, @isnumeric); % power of vna, ...
425 % default below
426 % the lowest ...
427 % power of ...
428 % E5071C,
429 % hence this
430 % parameter only
431 % takes ...
432 % effect if it
433 % is given a ...
434 % valid
435 % value
436 p.addParameter('center', 0, @isnumeric); % frequency ...
437 % sweep can also be defined
438 % by center and ...
439 % span,
440 % if they are ...
441 % given a
442 % valid value
443 p.addParameter('span',0, @isnumeric);
444
445 % useless, but required for input parser to be identical ...
446 % with input parser for manualSweep;
447 p.addParameter('issavedata',true,@islogical);
448 p.addParameter('hfig',233,@isnumeric); % figure handle
449 p.addParameter('notes','','@ischar'); % notes to ...
450 % add in file name

```

```

438
439
440     p.parse(varargin{:});
441     expandStructure(p.Results);
442
443     freqSectionSpan = res * points;
444     freqSectionSpan = 1e6 * round(freqSectionSpan/1e6);
445     if span ~= 0 && center ~= 0
446         start = center - span/2;
447         stop = center + span/2;
448     end
449     numOfSections = ceil((stop - start)/freqSectionSpan);
450     stop = start + numOfSections * freqSectionSpan;
451     totalPoints = points * numOfSections;
452
453     % initialize
454     freqs = NaN(1, totalPoints);
455
456     % apply parameters
457     obj.ifbw = ifbw;
458     if pow > -85
459         obj.power = pow;
460     end
461     obj.numOfPoints = points;
462     obj.freqSpan = freqSectionSpan;
463     obj.avg = obj.MAX_AVG;
464
465     sweepTime = obj.sweepTime;
466     waitTime = ceil(avg * sweepTime) + 1;
467     totalWaitTime = waitTime * numOfSections;
468
469     for i = 1:numOfSections
470         obj.freqCenter = start + freqSectionSpan/2 + ...
471                         freqSectionSpan* (i-1);
472
473         tmpfreqs = obj.freqs;
474         freqs((1 + (i-1)*points):(i*points)) = tmpfreqs;
475     end
476
477
478     %% fit
479     function [ f_r,Q_i,Q_c,Q_l ] = fit(obj,varargin)
480         % select range and fit plot
481         % ATTENTION: use vna.plotTrace or vna.manualSweep first ...
482             and then use vna.fit!

```

```

482      % EXAMPLE:
483      %   vna.plotTrace;
484      %   vna.fit('fitall',true);
485      % you can also give data to this method:
486      %     [ f_r,Q_i,Q_c,Q_l ] = ...
487      %           vna.fit('fitall',true,'issavefig',false,...)
488      %             ...
489      %               'xdata',freqs,'ydata',20*log10(abs(SParams)),...
490      %               ...
491      %               'titleNotes','_pow_-10dBm' );
492      %
493      % See the inputParser below for more options
494      %
495
496      p = inputParser;
497      p.addParameter('issavefig',true,@islogical);% if true, ...
498      % save fig to png file
499      p.addParameter('fitall', false,@islogical); % if true, ...
500      % fit all
501      % plotted data,
502      % else ask two
503      % input for ...
504      % the fit
505      % range
506      % xdata in ...
507      % GHz frequency
508      p.addParameter('ydata',[],@isnumeric); % ydata given ...
509      % in dB
510      p.addParameter('titleNotes','','@ischar'); % notes to ...
511      % add to the figure title
512      p.addParameter('QGuess',1e5,@isnumeric);
513      p.parse(varargin{:});
514      expandStructure(p.Results);
515
516      dataObj = get(gca,'children');
517      if isempty(xdata)
518          xdata = get(dataObj,'xdata');
519      end
520      if isempty(ydata)
521          ydata = get(dataObj,'ydata');
522      end
523
524      if gcf == obj.h_fig
525          figure(obj.h_fig);
526      else
527          figure(obj.h_fig);

```

```

519 plot(xdata, ydata);
520 % assume GHz frequency
521 xlabel frequency/GHz;
522 ylabel S/dB;
523 title([sprintf('start_%.4fGHz_stop_%.4fGHz', ...
524 min(xdata)/1e9, max(xdata)/1e9) ...
525 titleNotes], 'interpreter', 'none');
526 end
527
528 if fitall
529     leftInd = 1;
530     rightInd = length(xdata);
531 else
532     fprintf('Select the X range for fitting:');
533     tmpPoints = ginput(2);
534     leftX = min(tmpPoints(:,1));
535     rightX = max(tmpPoints(:,1));
536     leftInd = find(leftX < xdata, 1);
537     rightInd = find(rightX < xdata, 1);
538 end
539 t = xdata(:);
540 y = ydata(:);
541 % assume xdata given in GHz
542 t = t(leftInd:rightInd)*1e9;
543 % assume ydata given in dB, convert to linear
544 y = 10.^((y(leftInd:rightInd))./20);
545
546 % guess initial parameters
547 peakInd = find(abs(ydata) ≥ max(abs(ydata))), 1);
548 freq0 = xdata(peakInd); % in GHz
549 x1 = [freq0, QGuess/1e4, QGuess/1e4, 0, 0, 0];
550
551 % fit with complex S21 deduced theoretically
552 % 8 parameter, linear base
553 % F = ...
554 % @ (x, xdata) (20.*log10(abs(x(6).*(1+x(5).*(xdata-x(1).*1e9)./(x(1).*1e9)).*(1+...
555 % + ...
556 % x(3).^2.*1e4./cos(x(4)))./(x(3).^2.*1e4).* (cos(x(4))+li.*sin(x(4)))./(1+2.*...
557 % + ...
558 % x(3).^2.*1e4./cos(x(4))).*(xdata-x(1).*1e9)./(x(1).*1e9)))))+x(7).*xdata.*1e9;
559 % 7 parameter, constant base
560 F = ...
561 % @ (x, xdata) (abs(x(6).*(1+x(5).*(xdata-x(1).*1e9)./(x(1).*1e9)).*(1-(x(2).^2.*...
562 % + ...
563 % x(3).^2.*1e4./cos(x(4)))./(x(3).^2.*1e4).* (cos(x(4))+li.*sin(x(4)))./(1+2.*...
564 % + ...

```

```

      x(3).^2.*1e4./cos(x(4))).*(xdata-x(1).*1e9)./(x(1).*1e9))););
555 %x(1): f, center frequency, in GHz
556 %x(2): Qi, intrinsic Q, Ql = Qi*Qc/(Qi + Qc) = ...
557 %x(3): |Qe|, parameter Q, 1/Qc = Re (1/Qe) = ...
558 %x(4): cos(theta)/Qe, in 1e4
559 %x(5): alpha
560 %x(6): amplitude A
561
562
563 opt=optimset('MaxIter',10000,'MaxFunEvals',10000,'tolx',1e-16,'tolf',1e-9);
564 for loop_fit=1:5
565     [x_fit1,resnorm,~,exitflag,output] = ...
566         lsqcurvefit(F,x1,t,y,[],[],opt);
567     x1=x_fit1;
568     if ((x1(4)>pi/2)|| (x1(4)<-pi/2))
569         tmp = floor(abs(x1(4))./(pi/2));
570         if x1(4)>0
571             x1(4)=x1(4)-tmp.*pi/2;
572         end
573         if x1(4)<0
574             x1(4)=x1(4)+tmp.*pi/2;
575         end
576     end
577
578     f_r = x1(1)*1e9; % center frequency, in Hz
579     Q_i = x1(2).^2.*1e4; % interal Q
580     Q_c = x1(3).^2./cos(x1(4)).*1e4; % coupled Q
581     Q_l = Q_i.*Q_c./(Q_i + Q_c); % loaded Q
582
583     figure(obj.h_fig);
584     hold on
585     plot(t/1E9,20*log10(y),'.',t/1E9,20*log10(F(x_fit1,t)), 'LineWidth',2);
586     f_text=['f_r = '];
587     f_text=[f_text num2str(f_r/1e9)];
588     f_text=[f_text 'GHz'];
589     Ql_text=['Q_l = ' num2str(round(Q_l))];
590     Qi_text=['Q_i = ' num2str(round(Q_i))];
591     Qc_text=['Q_c = ' num2str(round(Q_c))];
592     text_pos=[(max(20*log10(y))-min(20*log10(y)))/4+min(20*log10(y)),min(20*log10(y));
593     text(t(1)/1E9,text_pos(1),f_text,'FontSize',18);
594     text(t(1)/1E9,text_pos(2),Ql_text,'FontSize',18);
595     text(t(round(end/1.5))/1E9,text_pos(1),Qi_text,'FontSize',18);
596     text(t(round(end/1.5))/1E9,text_pos(2),Qc_text,'FontSize',18);
```

```

597         hold off
598
599     if issavefig
600         str = ['Fit_ get(get(gca,'Title'), 'String') '.png'];
601         saveas(obj.h_fig,str);
602         fprintf(['Image ' str ' saved.\n'])
603     end
604
605 end
606
607
608 end
609
610
611
612 %% private methods
613 methods (Access = private)
614     function val = read(obj, varargin)
615         % varargin{1:(end-1)} are commands to be sent as a formatted
616         % string
617         % varargin{end} is the read format
618         fopen(obj.visa);
619         fprintf(obj.visa, varargin{1:(end-1)});
620         val = fscanf(obj.visa, varargin{end});
621         fclose(obj.visa);
622     end
623     function write(obj, varargin)
624         fopen(obj.visa);
625         fprintf(obj.visa, varargin{:});
626         fclose(obj.visa);
627     end
628 end
629 end

```

D.2 PPMS 控制代码

```

1 classdef PPMS < handle
2 % PPMS describe and control QDInstrument DynaCool at IIIS via ...
3 % which is much faster than using dll created from LabVIEW.
4 %
5 % Calling dll created from LabVIEW is slow and generates new client

```

```

6    % at each function call, which is very bad and troublesome. This
7    % version of PPMS avoided the above two problems.
8    %
9    % EXAMPLES (assuming instance named 'ppms'):
10   %
11   % Initialization:
12   %   ppms = PPMS;
13   %   ppms = ...
14   %       PPMS('address','101.6.98.151','isremote',true,'dllfilepath','C:\Users\IIIS\Document'
15   % Get temperature value and status:
16   %   temp = ppms.temp;
17   %   stat = ppms.tempStatus;      % This returns a .NET object, use
18   %                               % char(stat.ToString()) to get the
19   %                               % string, or directly use ...
20   %       char(ToString(ppms.fieldStatus))
21   %   statStr = ppms.tempStatusStr; % Directly get the temperature
22   %                               % status string
23   % Get field value and status:
24   %   fld = ppms.field;
25   %   stat = ppms.fieldStatus;    % See notes for ppms.tempStatus above
26   %   statStr = ppms.fieldStatusStr; % Directly get the field
27   %                               % status string
28   %
29   % Set temperature:
30   %   ppms.setTemp(4);
31   %   ppms.setTemp(2,'tempRate',5,'tempApproach','NoOvershoot');
32   %   ppms.setTemp(300,'tempRate',20,'tempApproach','FastSettle');
33   %
34   % Set field, field strength in Gauss (Oe):
35   %   ppms.setField(0,'fieldRate',50,'fieldApproach','Linear');
36   %   ppms.setField(200,'fieldRate',100);
37   %   ppms.setField(500,'fieldMode','Persistent');
38   %
39   % See Constant properties for available options for 'tempApproach',
40   % 'fieldApproach', and 'fieldMode'
41   %
42   % ATTENTION:
43   %   1. Setting 'fieldMode' to 'Persistent' etc. is not working as
44   %      expected.
45   %   2. Sometimes you might need to manually load the dll file ...
46   %      using NET.addAssembly(dllfilepath)
47   %          when you first started MATLAB, try initializing ppms and ...
48   %          also try
49   %              calling QuantumDesign.QDInstrument.QDInstrumentType.DynaCool
50   %              etc. for multiple times until it works. It will work when
51   %              auto-completion (by using TAB button) works.

```

```

48      %
49      % In development (May 12, 2017)
50      % Testing (May 13, 2017)
51      % Add get status in string format (May 14, 2017)
52      % Wentao, May 2017
53      %
54
55 properties (Constant)
56     % the second argument (numeric arrays) of these Constants is of ...
57     % no use
58     % the containers.Map is used for utilizing the isKey method, see
59     % the private methods for parameter verification.
60     INSTR_TYPE = ...
61         containers.Map({'PPMS', 'VersaLab', 'DynaCool', 'SVSM'}, [0,1,2,3]);
62
63     TEMP_APPROACH = containers.Map({'FastSettle', 'NoOvershoot'}, [0,1]);
64
65     FIELD_APPROACH = ...
66         containers.Map({'Linear', 'NoOvershoot', 'Oscillate'}, ...
67                         [0,1,2]);
68     FIELD_MODE = containers.Map({'Persistent', 'Driven'}, [0,1]);
69
70 end
71
72 properties
73     dllfilepath
74     address
75     isremote
76     instrType
77
78     QDInstr
79     vi
80
81     temp
82     field
83     tempStatus
84     fieldStatus
85
86     tempApproach
87     tempRate
88     fieldMode
89     fieldRate
90     fieldApproach
91
92 end
93
94 methods

```

```

91 function obj = PPMS(varargin)
92     % initialize PPMS
93     p = inputParser;
94     p.addParameter('address','101.6.98.151',@ischar);           % ip ...
95     % address of PPMS computer
96     p.addParameter('dllfilepath','C:\Users\IIIS\Documents\MATLAB\PPMS\QDInstrument.dll');
97     p.addParameter('isremote',true,@islogical);                 % is ...
98     % remote (is MATLAB and MultiVu on different computer)
99     p.addParameter('instrType','DynaCool',@checkInstrType);   % ...
100    % instrument type
101
102
103    obj.address = address;
104    obj.dllfilepath = dllfilepath;
105    obj.isremote = isremote;
106    obj.instrType = instrType;
107    obj.tempApproach = 'Unknown';
108    obj.fieldMode = 'Unknown';
109    obj.fieldApproach = 'Unknown';
110    obj.tempRate = NaN;
111    obj.fieldRate = NaN;
112
113    obj.QDInstr = NET.addAssembly(dllfilepath);
114    pause(1);
115    obj.tempStatus = ...;
116    % QuantumDesign.QDInstrument.TemperatureStatus.TemperatureUnknown;
117    obj.fieldStatus = ...;
118    % QuantumDesign.QDInstrument.FieldStatus.MagnetUnknown;
119
120    % initialize .NET object which is the vi for the ppms
121    obj.vi = ...;
122    % QuantumDesign.QDInstrument.QDInstrumentFactory.GetQDInstrument(...);
123    % QuantumDesign.QDInstrument.QDInstrumentType.(instrType), ...
124    % isremote,address,uint16(11000) );
125    fprintf('PPMS %s at %s connected.\n',obj.instrType, obj.address);
126
127
128    %% set & get temperature
129    function value = get.temp(obj)
130        [~, value, obj.tempStatus] = GetTemperature(obj.vi, ...
131            double(0), obj.tempStatus);

```

```

130     end
131     function value = get.tempStatus(obj)
132         [~, obj.temp, value] = GetTemperature(obj.vi, double(0), ...
133                                         obj.tempStatus);
134     end
135     function str = tempStatusStr(obj)
136         str = char(ToString(obj.tempStatus));
137     end
138
139     function setTemp(obj,varargin)
140         p = inputParser;
141         p.addRequired('temperature');
142         p.addParameter('tempRate',10,@isnumeric);
143         p.addParameter('tempApproach','FastSettle',@obj.checkTempApproach);
144         p.parse(varargin{:});
145         expandStructure(p.Results);
146
147         obj.tempApproach = tempApproach;
148         obj.tempRate = tempRate;
149
150         SetTemperature(obj.vi, double(temperature),double(tempRate),...
151                         QuantumDesign.QDInstrument.TemperatureApproach.(tempApproach));
152
153     end
154
155
156     %% set & get field
157     function value = get.field(obj)
158         [~, value, obj.fieldStatus] = GetField(obj.vi, 0, ...
159                                         obj.fieldStatus);
160     end
161     function value = get.fieldStatus(obj)
162         [~, obj.field, value] = GetField(obj.vi, 0, obj.fieldStatus);
163     end
164     function str = fieldStatusStr(obj)
165         str = char(ToString(obj.fieldStatus));
166     end
167
168     function setField(obj,varargin)
169         p = inputParser;
170         p.addRequired('b_field');
171         p.addParameter('fieldRate',100,@isnumeric);
172         p.addParameter('fieldMode','Driven',@obj.checkFieldMode)
173         p.addParameter('fieldApproach','Linear',@obj.checkFieldApproach);
174         p.parse(varargin{:});

```

```

174     expandStructure(p.Results);
175
176     obj.fieldMode = fieldMode;
177     obj.fieldRate = fieldRate;
178     obj.fieldApproach = fieldApproach;
179
180     SetField(obj.vi, double(b_field),double(fieldRate),...
181                 QuantumDesign.QDInstrument.FieldApproach.(fieldApproach),...
182                 QuantumDesign.QDInstrument.FieldMode.(fieldMode));
183
184 end
185
186 %% TODO: add set & get chamber (not so necessary since we have ...
187 %% remote desktop)
188
189 %% quick methods
190 function warmup(obj)
191     obj.setField(0);
192     obj.setTemp(300, 'tempRate', 20);
193 end
194
195 end
196
197 methods(Access = private)
198     % methods for checking string input validity
199     function passed = checkInstrType(obj, instrType)
200         passed = isKey(obj.INSTR_TYPE, instrType);
201     end
202
203     function passed = checkTempApproach(obj, tempMode)
204         passed = isKey(obj.TEMP_APPROACH,tempMode);
205     end
206
207     function passed = checkFieldMode(obj, fieldMode)
208         passed = isKey(obj.FIELD_MODE,fieldMode);
209     end
210     function passed = checkFieldApproach(obj, fieldApproach)
211         passed = isKey(obj.FIELD_APPROACH,fieldApproach);
212     end
213 end
214 end

```

D.3 扫描温度与磁场实验的代码

D.3.1 扫描温度

```
1 %%  
2 function [tempStableTmr, setTempTmr] = TempSweep(ppms,vna,temp, varargin)  
3 % temperature sweep  
4 %  
5 % Parameters:  
6 % ppms: the PPMS object  
7 % vna: the E5071C object  
8 % temps: temperatures to sweep  
9 % varargin: optional input arguments, see inputParser below for detail  
10 %  
11 % Returns:  
12 % Two timer object tempStableTmr and setTempTmr.  
13 %  
14 % Sweep results will be automatically saved.  
15 % You can also find parameters and sweep results in  
16 % tempStableTmr.UserData and setTempTmr.UserData  
17 % It's better to delete and clear the timer object since they'll ...  
18 % remained  
19 % in the memory:  
20 %     delete(tempStableTmr);  
21 %     delete(setTempTmr);  
22 %     clear setTempTmr;  
23 %     clear tempStableTmr;  
24 % ATTENTION: the waiting process (for the temperature to be stable) is  
25 % going in the background, when you can do stuff in the command line at  
26 % the same time. HOWEVER, the vna sweep is not and the command line  
27 % won't react during the vna sweep.  
28 %  
29 % In development (May 13, 2017)  
30 %  
31 % TODO:  
32 % Done. add input parser  
33 % Done. choose vna sweep mode. Change 'plotTrace' to 'fetchTrace'  
34 % Done. choose temperature wait mode ('Near' or 'Stable')  
35 % Done. fetch actual temperatures and add timestamps;  
36 %  
37 % Wentao, May 2017  
38 %  
39
```

```

40 p = inputParser;
41 p.addParameter('vnaMode','plotTrace',@ischar);           % fetchTrace will ...
42                                         directly fetch vna trace.
43                                         % Else, the ...
44                                         E5071C/manualSweep ...
45                                         method is called
46
47 p.parse(varargin{:});
48 expandStructure(p.Results);
49
50 if strcmpi(vnaMode, 'plotTrace')
51     freqs = vna.freqs;
52     waitTime = vna.sweepTime;
53 else
54     [freqs,waitTime] = vna.manualSweepFreqs(manualSweepConfig);
55 end
56 tempStableTmr = timer;
57 setTempTmr = timer;
58 set(tempStableTmr,'ExecutionMode','fixedRate');
59 set(tempStableTmr,'period',checkTempPeriod);           % check temperature ...
60                                         stable period
61 set(tempStableTmr,'TimerFcn',@timerCalled);
62 set(tempStableTmr,'userdata',struct('ppms',ppms,'numOfTemps',length(temp),...
63                                         'manualSweepConfig',manualSweepConfig,...)
64                                         'tempWaitMode',tempWaitMode,...)
65                                         'vnaMode',vnaMode,...)
66                                         'freqs',freqs,...)
67                                         'timeStamps',NaN(1,length(temp)),...)
68                                         'actualTemps',NaN(1,length(temp)),...)
69                                         'vna',vna,'t',setTempTmr,'waitTime',waitTime,...)
70                                         'cnt',1,'SParams',NaN(length(temp),length(freqs)) ...
71                                         ) );
72 set(setTempTmr,'userdata',struct('ppms',ppms,'t',tempStableTmr,'temp',temp,'cnt',1) ... )
73 set(setTempTmr,'TimerFcn',@setTempTimerCalled);
74 start(setTempTmr);
75 end
76
77 %%

```

```

78 function timerCalled(thisObj, event)
79 ud = thisObj.UserData;
80 expandStructure(ud);
81 stat = char(ToString(ppms.tempStatus));
82 if strcmpi(stat, tempWaitMode)
83     fprintf('Temperature %s, start vna averaging (%.2fs)...\\n',...
84             tempWaitMode,waitTime+0.5)
85
86     % vna sweep
87     if strcmpi(vnaMode, 'plotTrace')
88         vna.clearAvg;
89         pause(waitTime+0.5);
90         trace = vna.trace;
91     else
92         [~,trace] = vna.manualSweep(manualSweepConfig);
93     end
94     fprintf('vna trace fetched.\\n');
95
96     % save SParams back into timer
97     ud.actualTemps(cnt) = ppms.temp;
98     ud.timeStamps(cnt) = now;
99     ud.SParams(cnt,:) = trace.X(:)' + li*trace.Y(:)';
100    ud.cnt = cnt + 1;
101    thisObj.UserData = ud;
102    stop(thisObj);
103    fprintf('Temperature timer stopped at %s.\\n', datestr(now,30))
104    if ud.cnt < numOfTemps
105        fprintf('\\n');
106        start(t);
107    else
108        fprintf('Temperature sweep finished at %s\\n',datestr(now,30));
109        % save data
110        expandStructure(ud);
111        ud2 = t.UserData;
112        setTemps = ud2.temps;
113        actualTemps = ud.actualTemps;
114        timeStamps = ud.timeStamps;
115        startF = min(ud.freqs);
116        stopF = max(ud.freqs);
117        pow = vna.power;
118        config = vna.getConfig;
119        minT = min(setTemps);
120        maxT = max(setTemps);
121        fname = ...
122            sprintf('TempSweep_start_%.4fGHz_stop_%.4fGHz_pow_%.1fdBm_minT_%.3fK_maxT_%.3fK',...
123                startF/1e9,stopF/1e9,pow,minT,maxT,length(setTemps),datestr(now,30));

```

```

123     save(fname,'SParams','waitTime','setTemps','actualTemps',...
124         'freqs','config','timeStamps','manualSweepConfig');
125     fprintf('%s saved.\n',fname);
126 end
127 else
128     fprintf('Temperature is %. Waiting for temperature to be ...
129             %s...\n',...
130             stat,tempWaitMode)
131 end
132 end
133
134 %%
135 function setTempTimerCalled(thisObj,event)
136 ud = thisObj.UserData;
137 expandStructure(ud);
138 temp = temps(cnt);
139 fprintf('Set ppms temperature to %.4f K, %s\n',temp,datestr(now,30));
140 ppms.setTemp(temp,'tempApproach','NoOvershoot','tempRate',1);
141 pause(0.5);
142 start(t);
143 ud.cnt = cnt+1;
144 thisObj.UserData = ud;
145
146 end

```

D.3.2 扫描磁场

```

1 %%
2 function [fldStableTmr, setFldTmr] = ...
3     FieldSweep(ppms,vna,BFields,varargin)
4 %
5 % Parameters:
6 %     ppms: the PPMS object
7 %     vna: the E5071C object
8 %     BFields: Magnetic fields to sweep, in ***GAUSS***!!! (1 gauss = ...
8 %              0.1 mT)
9 %     varargin: optional input arguments, see inputParser below for detail
10 %
11 % Returns:
12 %     Two timer object tempStableTmr and setTempTmr.
13 %

```

```

14 % Sweep results will be automatically saved.
15 % You can also find parameters and sweep results in
16 % fldStableTmr.UserData and setFldTmr.UserData
17 % It's better to delete and clear the timer object since they'll ...
18 % remained
19 % in the memory:
20 %     delete(fldStableTmr);
21 %     delete(setFldTmr);
22 %     clear fldStableTmr;
23 %     clear setFldTmr;
24 %
25 % ATTENTION: the waiting process (for the field to be stable) is
26 % going in the background, when you can do stuff in the command line at
27 % the same time. HOWEVER, the vna sweep is not and the command line
28 % won't react during the vna sweep.
29 %
30 %
31 % Wentao, May 2017
32 %
33
34 p = inputParser;
35 p.addParameter('vnaMode','plotTrace',@ischar);           % fetchTrace will ...
36 % directly fetch vna trace.
37 % Else, the ...
38 % E5071C/manualSweep ...
39 % method is called
40
41 p.addParameter('checkFldPeriod',5,@isnumeric);
42 p.addParameter('fldWaitMode','StableDriven',@ischar);    % another ...
43 % option is 'Near'
44 p.addParameter('manualSweepConfig',struct([]),@issstruct); % ...
45 % ... parameters for the E5071C/manualSweep method as a structure
46
47 p.parse(varargin{:});
48 expandStructure(p.Results);
49
50 if strcmpi(vnaMode, 'plotTrace')
51     freqs = vna.freqs;
52     waitTime = vna.sweepTime;
53 else
54     [freqs,waitTime] = vna.manualSweepFreqs(manualSweepConfig);
55 end
56 fldStableTmr = timer;
57 setFldTmr = timer;
58 set(fldStableTmr,'ExecutionMode','fixedRate');
59 set(fldStableTmr,'period',checkFldPeriod);      % check field stable period

```

```

54 set(fldStableTmr,'TimerFcn',@timerCalled);
55 set(fldStableTmr,'userdata',struct('ppms',ppms,'numOfFlds',length(BFields),...
56                                         'manualSweepConfig',manualSweepConfig,...)
57                                         'fldWaitMode',fldWaitMode,...)
58                                         'vnaMode',vnaMode,...)
59                                         'freqs',freqs,...)
60                                         'timeStamps',NaN(1,length(BFields)),...)
61                                         'actualFields',NaN(1,length(BFields)),...)
62                                         'vna',vna,'t',setFldTmr,'waitTime',waitTime,...)
63                                         'cnt',1,'SParams',NaN(length(BFields),length(freqs)) ...
64                                         ) );
65 set(setFldTmr,'userdata',struct('ppms',ppms,'t',fldStableTmr,...)
66                                         'BFields',BFields,'cnt',1) );
67 set(setFldTmr,'TimerFcn',@setFldTimerCalled);
68
69 start(setFldTmr);
70 end
71
72 %%
73 function timerCalled(thisObj,event)
74 ud = thisObj.UserData;
75 expandStructure(ud);
76 stat = char(ToString(ppms.fieldStatus));
77 if strcmpi(stat, fldWaitMode)
78     fprintf('Field %s, start vna averaging (%.2fs)... \n',...
79             fldWaitMode,waitTime+0.1)
80
81 % vna sweep
82 if strcmpi(vnaMode, 'plotTrace')
83     vna.clearAvg;
84     pause(waitTime+0.1);
85     trace = vna.trace;
86 else
87     [~,trace] = vna.manualSweep(manualSweepConfig);
88 end
89 fprintf('vna trace fetched.\n');
90
91 % save SParams back into timer
92 ud.actualFields(cnt) = ppms.field;
93 ud.timeStamps(cnt) = now;
94 ud.SParams(cnt,:) = trace.X(:)' + 1i*trace.Y(:)';
95 ud.cnt = cnt + 1;
96 thisObj.UserData = ud;
97 stop(thisObj);
98 fprintf('Field timer stopped at %s.\n', datestr(now,30))

```

```

99     if ud.cnt <= numOfFlds
100        fprintf('\n');
101        start(t);
102    else
103        fprintf('Field sweep finished at %s\n',datestr(now,30));
104        % save data
105        expandStructure(ud);
106        ud2 = t.UserData;
107        setFields = ud2.BFields;
108        actualFields = ud.actualFields;
109        timeStamps = ud.timeStamps;
110        startF = min(ud.freqs);
111        stopF = max(ud.freqs);
112        pow = vna.power;
113        config = vna.getConfig;
114        minB = min(setFields);
115        maxB = max(setFields);
116        fname = ...
117            sprintf('FieldSweep_start_%.4fGHz_stop_%.4fGHz_pow_%.1fdbm_minB_%.0fG_maxB_%.0fG');
118            startF/1e9,stopF/1e9,pow,minB,maxB,length(setFields),datestr(now,30));
119            save(fname,'SParams','waitTime','setFields','actualFields',...
120            'freqs','config','timeStamps','manualSweepConfig');
121            fprintf('%s saved.\n',fname);
122    end
123 else
124     fprintf('Field is %s, Waiting for field to be %s...\n',...
125         stat, fldWaitMode)
126 end
127 end
128
129 %%
130 function setFldTimerCalled(thisObj,event)
131 ud = thisObj.UserData;
132 expandStructure(ud);
133 fld = BFields(cnt);
134 fprintf('Set ppms field to %.4f Gauss, %s\n',fld,datestr(now,30));
135 ppms.setField(fld,'fieldApproach','Linear','fieldRate',50);
136 ud.cnt = cnt+1;
137 thisObj.UserData = ud;
138 pause(0.5);
139 start(t);
140 end

```

附录 E 外文资料调研阅读报告

在学期间参加课题的研究成果

个人简历

1995 年 4 月 26 日出生于四川省德阳市。

2013 年 9 月考入清华大学工程物理系，后转入物理系物理学专业。

发表的学术论文

- [1] Yipu Song, Haonan Xiong, Wentao Jiang, Hongyi Zhang, Xiao Xue, Cheng Ma, Yulin Ma, Luyan Sun, Haiyan Wang, and Luming Duan. Coulomb Oscillations in a Gate-Controlled Few-Layer Graphene Quantum Dot. *Nano Lett.*, 2016, 16 (10), pp 6245–6251 DOI: 10.1021/acs.nanolett.6b02522.
- [2] Haonan Xiong*, Wentao Jiang*, Yipu Song and Luming Duan. Bound state properties of ABC-stacked trilayer graphene quantum dots. 2017 *J. Phys.: Condens. Matter* 29 215002. DOI: 10.1088/1361-648X/aa6aac.arXiv: 1611.09208 (* 共同一作)