

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：宋伟璞副研究员

2017 年 5 月 22 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<sub>E</sub>X; L<sup>A</sup>T<sub>E</sub>X; CJK; template; thesis

# 目 录

目 录 .....	III
第 1 章 引言：混合量子系统简介 .....	1
1.1 混合量子系统的重要性 .....	1
1.2 超导量子系统与常见固态自旋系统 .....	1
1.3 超导量子系统与自旋系综的耦合 .....	3
1.4 超导量子比特的制备与测量 .....	5
第 2 章 自旋与谐振腔耦合强度仿真 .....	6
2.1 新型 3D 谐振腔与自旋系综的耦合 .....	6
2.2 2D 平面波导谐振腔与自旋系综的耦合 .....	8
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 .....	10
2.3.1 参数估计 .....	10
2.3.2 设计与优化 .....	11
第 3 章 PPMS 测量系统 .....	15
3.1 测量系统概述 .....	15
3.2 测量系统的程序编写 .....	15
3.3 样品托的设计与优化 .....	17
第 4 章 2.5 维谐振腔的制备与测量 .....	18
4.1 器件测量与制备工艺概述 .....	18
4.2 光刻板的设计 .....	18
4.3 谐振腔响应信号的拟合 .....	20
插图索引 .....	21
表格索引 .....	23
公式索引 .....	24
参考文献 .....	26
致 谢 .....	28

声 明 .....	29
附录 A 超导量子比特的原理 .....	30
附录 B 微纳加工工艺 .....	31
附录 C 测量系统 MATLAB 代码 .....	32
C.1 VNA 控制代码 .....	32
C.2 PPMS 控制代码 .....	47
C.3 扫描温度与磁场实验的代码 .....	53
C.3.1 扫描温度 .....	53
C.3.2 扫描磁场 .....	56
附录 D 外文资料原文 .....	60
附录 E 外文资料的调研阅读报告或书面翻译 .....	61
附录 F 其它附录 .....	62
在学期间参加课题的研究成果 .....	63

## 主要符号对照表

dll	动态链接库
NV	氮-空位色心
PPMS	物理性质测量系统
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：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<sup>[1]</sup>，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。<sup>[2]</sup> 自从量子加密通信的 BB84 协议<sup>[3]</sup>，量子搜索算法<sup>[4]</sup> 与量子质因数分解算法<sup>[5]</sup> 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

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<sup>[6]</sup>。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<sup>[7]</sup>，离子阱<sup>[8]</sup>，固态自旋系统<sup>[9]</sup>，超导量子比特<sup>[10]</sup> 以及线性光学<sup>[11]</sup> 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。

### 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：

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



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<sup>[12]</sup>。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。

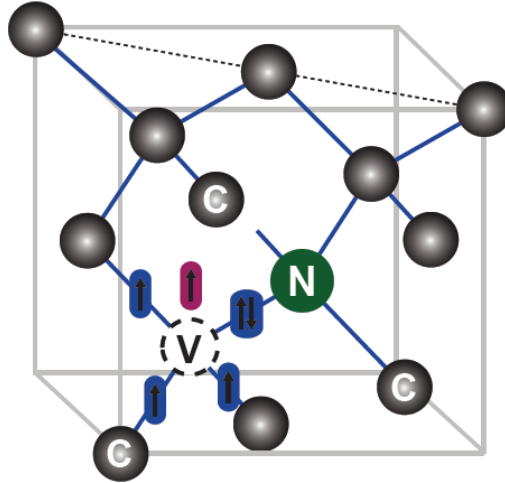


图 1.1 金刚石色心结构示意图，其中碳空位由 V 表示，氮掺杂由 N 表示<sup>[13]</sup>。

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

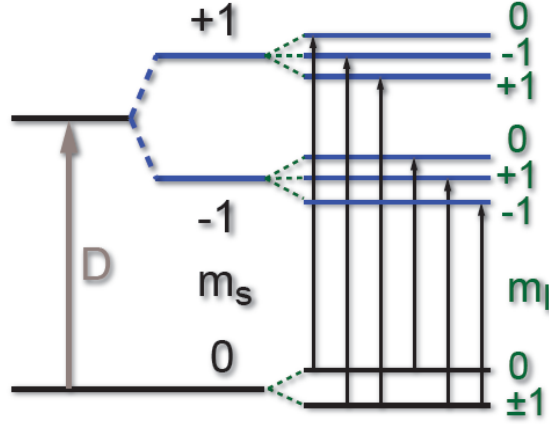


图 1.2 金刚石色心在考虑零场劈裂与外加磁场后的能级示意图。其中  $D$  为零场劈裂导致的能级分裂， $m_s = \pm 1$  的两个态之间的能量差来源于应力与局域电场。不同核自旋量子数  $m_I$  的态之间的能级分裂来源于超精细相互作用<sup>[13]</sup>。

简并能级发生分裂，相应的哈密顿量为<sup>[13]</sup>

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

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统的耦合

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，因此常采用自旋系统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<sup>[13]</sup>。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<sup>[13]</sup>

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

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

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

其中自旋-谐振腔耦合系数

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

$\delta B$  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  $\kappa$  为谐振腔的衰减率,  $\gamma$  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,  $g \sim 2\pi \cdot 10\text{Hz}$ <sup>[13]</sup>, 远远小于  $\kappa, \gamma$  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)<sup>[14]</sup>

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

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。

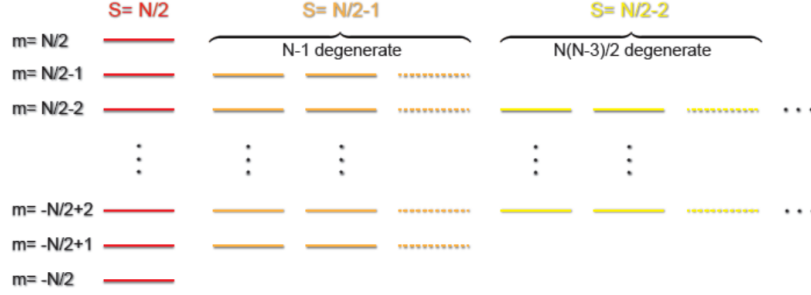


图 1.3  $N$  个金刚石自旋构成的自旋系综的能级示意图<sup>[13]</sup>。可以看见  $S \neq N/2$  的态均为高度简并态。

当自旋系综从谐振腔吸收一个光子时，相应的态为  $|B\rangle = |N/2, -N/2 + 1\rangle \equiv S_+ |G\rangle / |S_+ |G\rangle| = \sum_k |E_k\rangle / \sqrt{N}$ 。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<sup>[12,15]</sup>，制备方法<sup>[16,17]</sup> 以及测量方法<sup>[18]</sup> 进行调研、总结与实验。

## 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极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<sup>[19]</sup>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<sup>[20]</sup>。这种三维谐振腔中电流与磁场分布示意图如下图所示

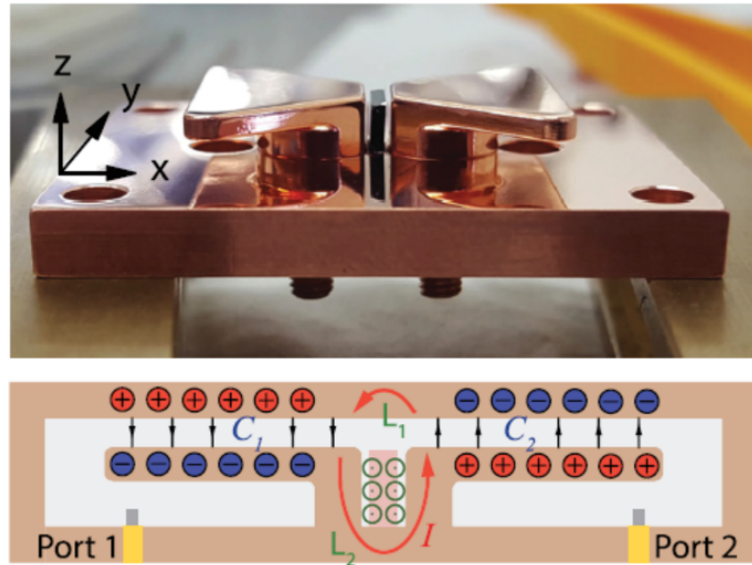


图 2.1 新型三维谐振腔的几何结构与电流，磁场分布示意图<sup>[20]</sup>

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。

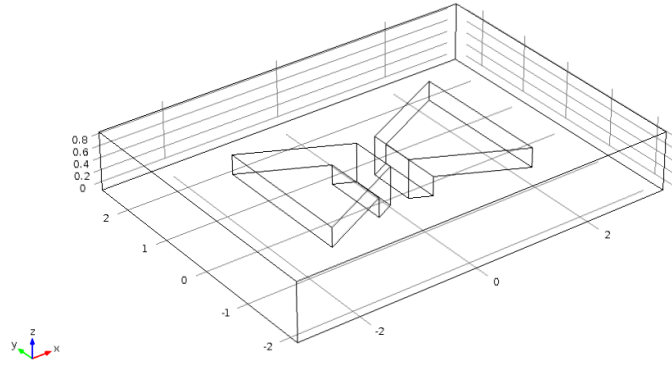


图 2.2 新型三维谐振腔仿真的几何设计 (透视图)

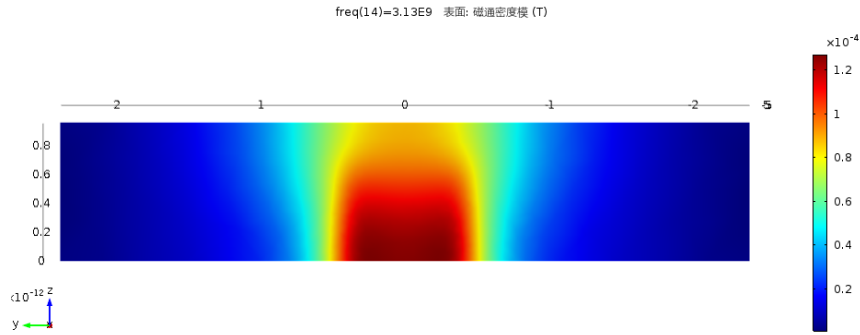


图 2.3 新型三维谐振腔磁场在自旋系综所处空间横截面上的大小分布

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<sup>[13]</sup>

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

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  $\kappa, \kappa_L$ ，进而求出对应任意功率下的谐振腔中的光子数<sup>[13]</sup>

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

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<sup>[20]</sup>

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

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<sup>[21,22]</sup>。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<sup>[23]</sup>。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

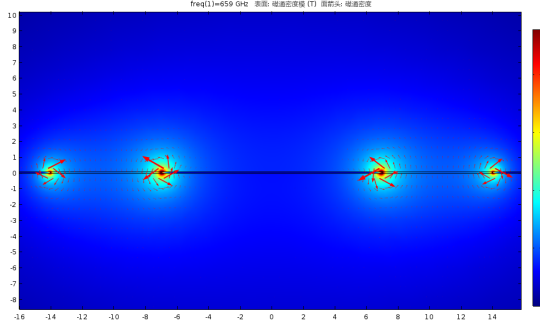


图 2.4 仿真所得磁场空间分布截面图

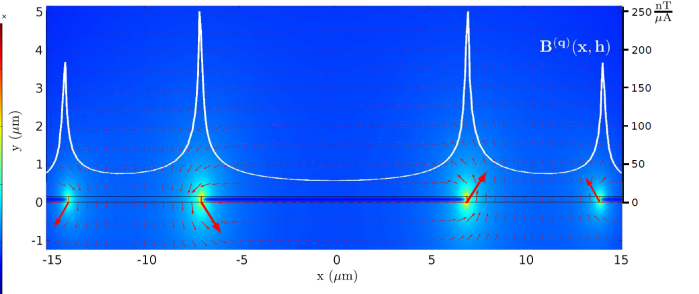


图 2.5 文献<sup>[23]</sup> 中所示磁场空间分布截面图

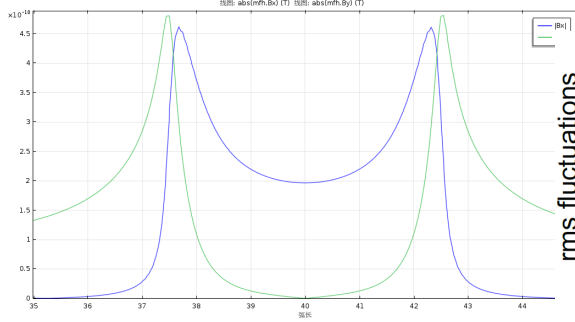


图 2.6 仿真所得磁场在位于距金属上表面  $0.1\mu\text{m}$  处水平截线上的分布

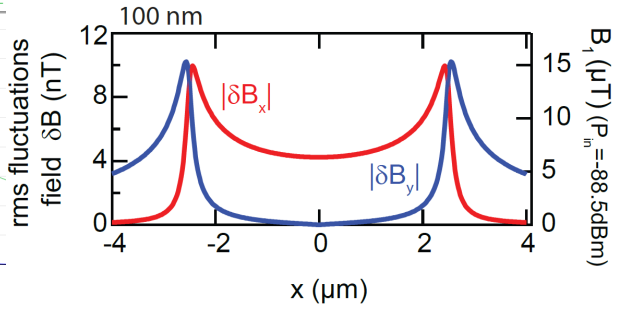


图 2.7 文献<sup>[26]</sup> 中所示磁场在位于距金属上表面  $0.1\mu\text{m}$  处水平截线上的分布

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<sup>[23-25]</sup>

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

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

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

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。

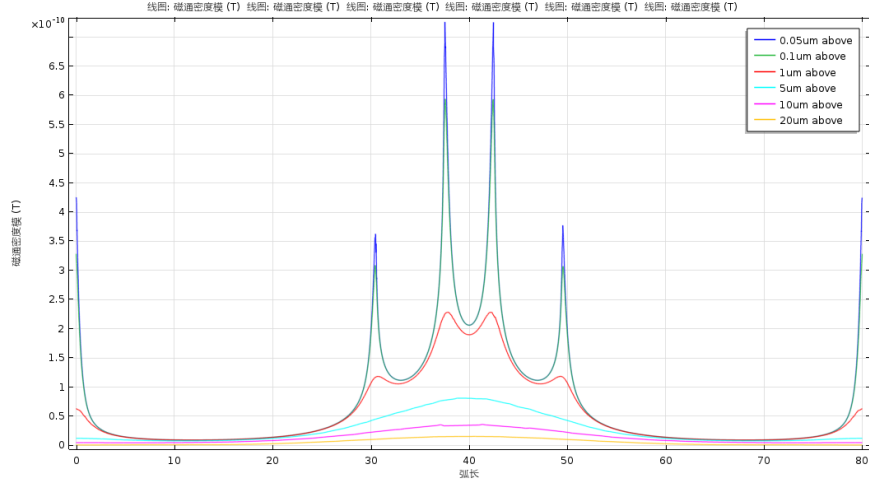


图 2.8 基于我们制备的二维平面波导的常见几何尺寸进行仿真所得的磁感应强度分布图。

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  $3\text{GHz}$  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  $\text{nF}$  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<sup>[19]</sup>，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  $10\text{nm}$  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

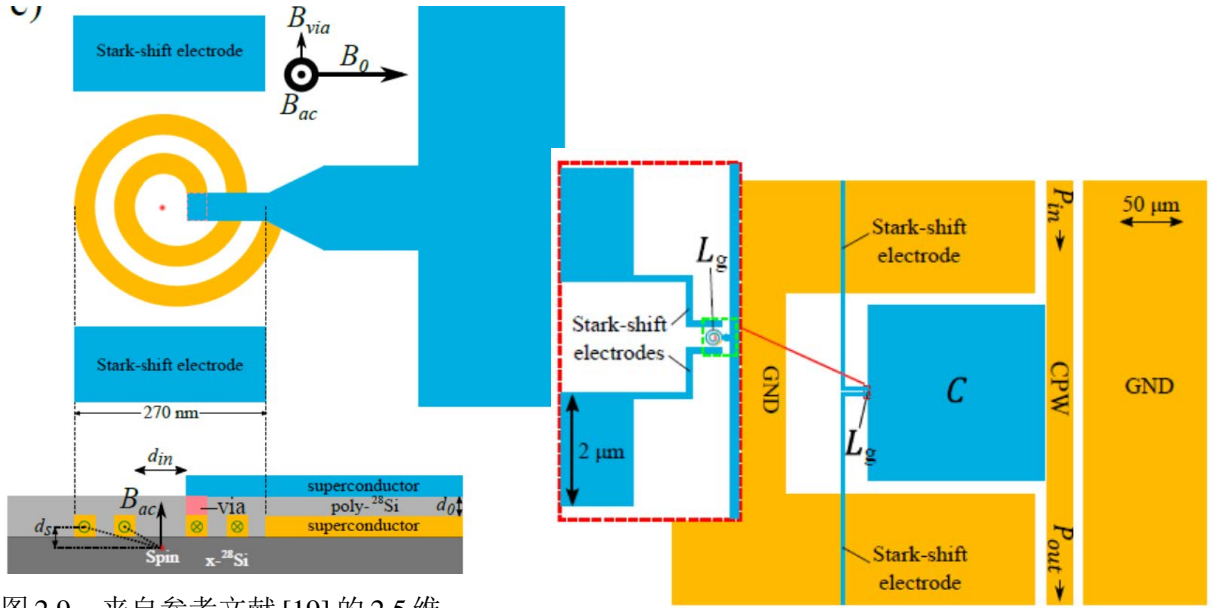


图 2.9 来自参考文献 [19] 的 2.5 维谐振腔的设计（右图中红色虚线部分的局部放大图）。

图 2.10 来自参考文献 [19] 的 2.5 维谐振腔的设计（整体）。

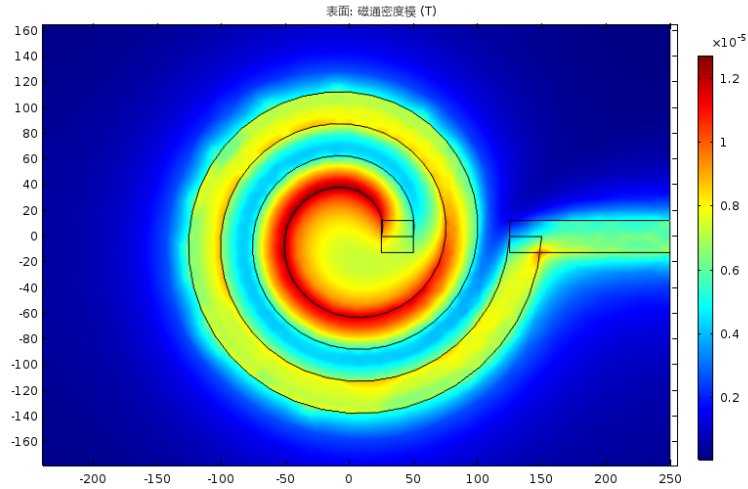


图 2.11 2.5 维谐振腔的螺旋状电感处水平截面上的磁感应强度分布。

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会自然增大导线的自感。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%。

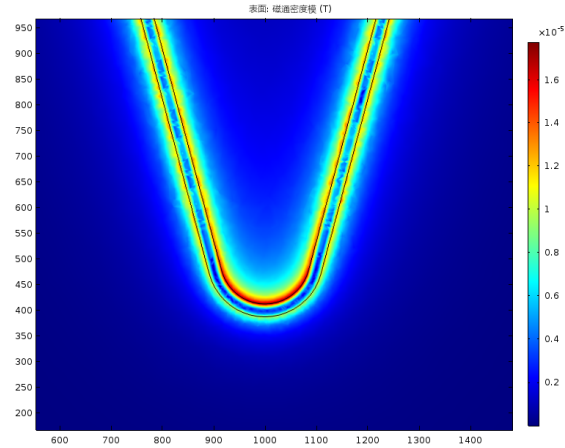


图 2.12 第一次改进后的 2.5 维谐振腔的电感处水平截面上的磁感应强度分布。

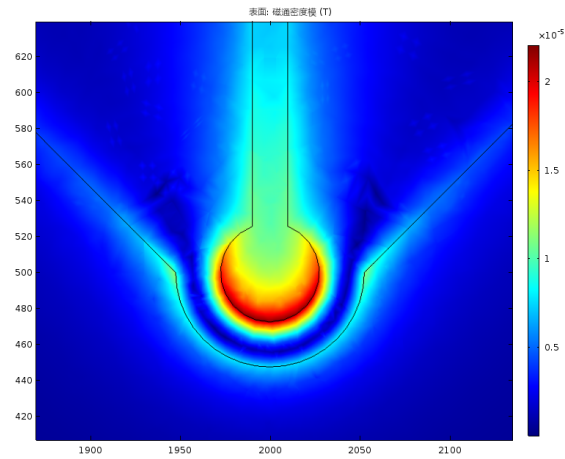


图 2.13 第二次改进后的 2.5 维谐振腔的电感处水平截面上的磁感应强度分布，电感部分的电感大小得到了降低。

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

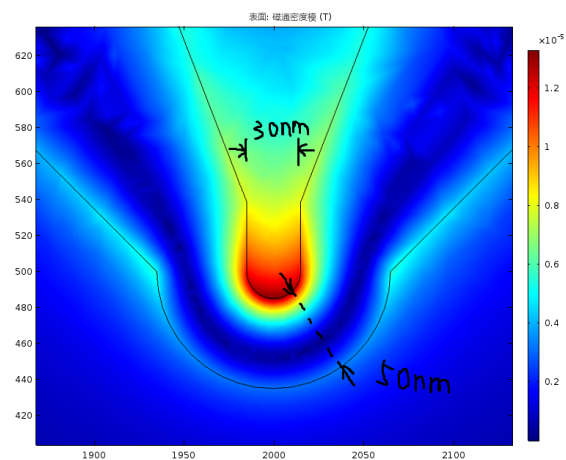


图 2.14 第三次改进后的 2.5 维谐振腔的电感处水平截面上的磁感应强度分布，通过加厚材料并加粗宽度提升了临界电流大小，并不影响磁场较强区域的强度的基础上改进了设计以提高制备的成功率。

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。

实验中所用的 PPMS 为 QuantumDesign 公司的 DynaCool, 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?? 所示,

### 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C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你和结果。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] = ...
    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 C.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
```

### 3.3 样品托的设计与优化



## 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. Photolithography to make transmission line, bottom layer of the 3D capacitor
2. ALD SiO<sub>2</sub> or Al<sub>2</sub>O<sub>3</sub>, currently Al<sub>2</sub>O<sub>3</sub>
3. Photolithography and ICP (or wet) etching of Al<sub>2</sub>O<sub>3</sub> (alignment: 10  $\mu$ m)
  - (a) remaining SiO<sub>2</sub> only covers the lower capacitor layer
4. Photolithography to make the top Nblayer (alignment 1-2  $\mu$ m)
5. EBL to make the other half of the 2D capacitor and the inductor
  - (a) 2D capacitor connecting to the top layer of the 3D capacitor
  - (b) alignment 100-1000 nm

### 4.2 光刻板的设计

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。第三步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所示

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，

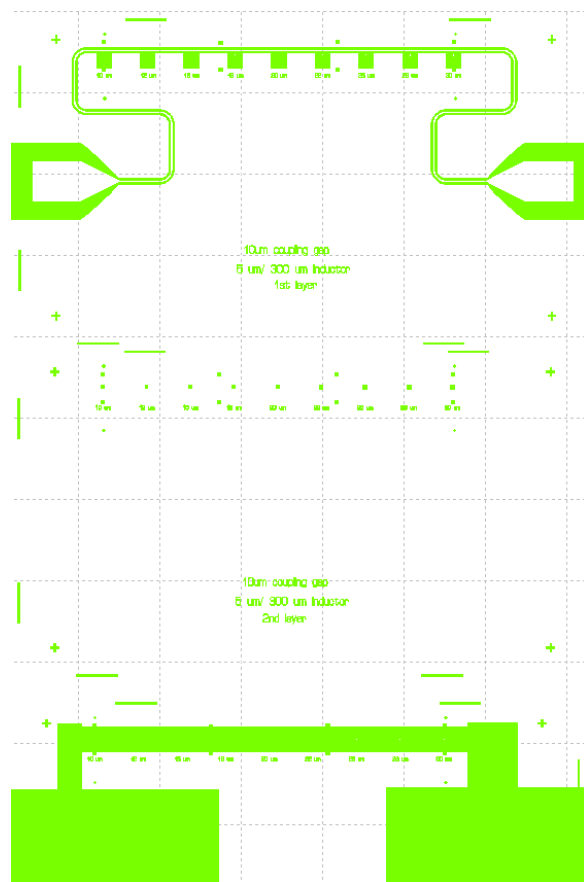


图 4.1 制备一个器件所需的三个光刻步骤对应的一组三个模板图案

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。

目前器件制备已经进行到了4.1制备工艺概述中的第三步，我们需要对刻蚀介电层的速率进行测试与标定。刻蚀介电层成功后即可进行第三步光刻，从而获得第一批待测器件。

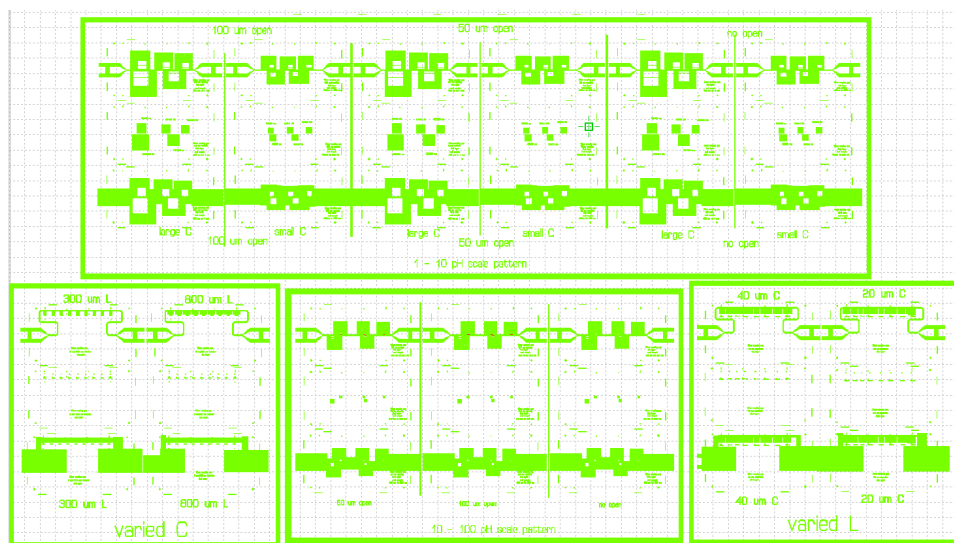


图 4.2 完整的光刻板图案

### 4.3 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 <sup>[13]</sup> 。2
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 <sup>[13]</sup> 。 .....3
1.3	N 个金刚石自旋构成的自旋系综的能级示意图 <sup>[13]</sup> 。可以看见 $S \neq N/2$ 的态均为高度简并态。 .....5
2.1	新型三维谐振腔的几何结构与电流, 磁场分布示意图 <sup>[20]</sup> .....6
2.2	新型三维谐振腔仿真的几何设计 (透视图).....7
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.....7
2.4	仿真所得磁场空间分布截面图 .....9
2.5	文献 <sup>[23]</sup> 中所示磁场空间分布截面图.....9
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 .....9
2.7	文献 <sup>[26]</sup> 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 9
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。 ..... 10
2.9	来自参考文献 [19] 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)。 ..... 12
2.10	来自参考文献 [19] 的 2.5 维谐振腔的设计 (整体)。 ..... 12
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。 ..... 12
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。 ..... 13

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。.....	13
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。	14
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 .....	19
4.2	完整的光刻板图案 .....	20

## 表格索引

## 公式索引

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

公式 2-10 .....	11
公式 2-11 .....	11



## 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] 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
- [13] 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
- [14] Tavis M, Cummings F W. Exact solution for an n-molecule-radiation-field hamiltonian. Physical Review, 1968, 170(2):379
- [15] Schuster D I. Circuit quantum electrodynamics. 2007
- [16] Krantz P. Investigation of transmon qubit designs: a study of plasma frequency predictability. Chalmers University of Technology, 2010
- [17] Kelly J S. Fault-tolerant superconducting qubits. 2015

- [18] 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
- [19] 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.
- [20] 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
- [21] 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
- [22] 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
- [23] 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
- [24] 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
- [25] Tosi G, Mohiyaddin F A, Huebl H, et al. Circuit-quantum electrodynamics with direct magnetic coupling to single-atom spin qubits in isotopically enriched  $^{28}\text{Si}$ . *AIP Advances*, 2014, 4(8):087122
- [26] 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
- [27] 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

## 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

## 附录 C 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。

### C.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{:});
69             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 end
88
89 function delete(obj)
90     delete(obj.visa);
91 end
92
93 %% frequency set & get
94 function value = get.freqStart(obj)
95     value = obj.read(':sens:freq:star?','%f');
96 end
97 function set.freqStart(obj,val)
98     obj.write(':sens:freq:star %f',val);
99 end
100 function value = get.freqStop(obj)
101     value = obj.read(':sens:freq:stop?','%f');
102 end
103 function set.freqStop(obj,val)
104     obj.write(':sens:freq:stop %f',val);
105 end
106 function value = get.freqCenter(obj)
107     value = obj.read(':sens:freq:cent?','%f');
108 end
109 function set.freqCenter(obj,val)
110     obj.write(':sens:freq:cent %f',val);
111 end
112 function value = get.freqSpan(obj)
113     value = obj.read(':sens:freq:span?','%f');
114 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 = ...
            obj.read(':SOURce:POWer:LEVel:IMMediate:AMPLitude?', ...
                '%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         %   External Trigger
178         %   Generates a trigger when the trigger signal is ...
179         %   inputted externally via the Ext Trig connector or the ...
180         %   handler interface.
181         %   Manual Trigger
182         %   Generates a trigger when the key operation of ...
183         %   Trigger > Trigger is executed from the front panel.
184         %   Bus Trigger
185         %   Generates a trigger when the SCPI.IEEE4882.TRG ...
186         %   object is executed.
187         obj.write(':TRIG:SEQ:SOUR %s', val);
188     end
189     %% set & get configurations
190     function setConfig(obj, config)
191         % setConfig apply parameters to E5071C
192         % config should have same or less fields as E5071C properties
193         % with set & get methods
194         flds = fieldnames(config);
195         for ii = 1:length(flds)
196             fld = flds{ii};
197             obj.(fld) = config.(fld);
198         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 + 1i*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     % See the inputParser below for more options
313     %
314
315     p = inputParser;
316     p.addParameter('start', 1e9, @isnumeric); % start frequency
317     p.addParameter('stop', 8e9, @isnumeric);  % stop frequency
318     p.addParameter('res', 1e6, @isnumeric);    % frequency ...
319         resolution
320     p.addParameter('avg', 1, @isnumeric);      % number of ...
321         average
322     p.addParameter('ifbw', 100, @isnumeric);   % ifbw of vna
323     p.addParameter('points', obj.MAX_POINTS, @isnumeric); % ...
324         number of points of vna
325     p.addParameter('pow', -100, @isnumeric); % power of vna, ...
326         default below
327
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 p.parse(varargin{:});
340 expandStructure(p.Results);
341
342 freqSectionSpan = res * points;
343 freqSectionSpan = 1e6 * round(freqSectionSpan/1e6);
344 if span ~= 0 && center ~= 0
345     start = center - span/2;
346     stop = center + span/2;
347 end
348 numOfSections = ceil((stop - start)/freqSectionSpan);
349 stop = start + numOfSections * freqSectionSpan;
350 totalPoints = points * numOfSections;
351
352 % initialize
353 freqs = NaN(1, totalPoints);
354 trace.X = freqs;
355 trace.Y = freqs;
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_%.2fBm_AVG_%.2f', ...
370     start/1e9, stop/1e9, res/1e6, obj.power, avg);
371 title(str, 'interpreter', 'none');
372 fprintf('\tsweep from %.2fGHz to %.2fGHz, %d sections, ...
373     %ds per section\n\ttotal points: %d, total time: ...
374     %ds.\n', ...
375     start/1e9, stop/1e9, ...
376     numSections, waitTime, totalPoints, waitTime * ...
377     numSections);
378 for i = 1:numSections
379     obj.freqCenter = start + freqSectionSpan/2 + ...
380     freqSectionSpan * (i-1);
381     fprintf('sweeping %.2fGHz to ...
382     %.2fGHz...\n', obj.freqStart/1e9, obj.freqStop/1e9);
383     pause(waitTime);
384     tmptrace = obj.trace;
385     tmpfreqs = obj.freqs;
386     freqs((1 + (i-1)*points):(i*points)) = tmpfreqs;
387     trace.X((1 + (i-1)*points):(i*points)) = tmptrace.X';
388     trace.Y((1 + (i-1)*points):(i*points)) = tmptrace.Y';
389     figure(hfig);
390     plot(freqs/1e9, 20*log10(abs(trace.X + 1i*trace.Y)));
391 end
392 fprintf('Sweep finished!\n');
393
394 figure(hfig);
395 xlabel freq/GHz
396 ylabel SParameter/dB
397 str = ...
398     sprintf('start_%.2fGHz_stop_%.2fGHz_res_%.2fMHz_pow_%.2fBm_AVG_%.2f', ...
399     start/1e9, stop/1e9, res/1e6, pow, avg, notes);
400 title(str, 'interpreter', 'none');
401
402 if issavedata
403     config = obj.getConfig;
404     save([str '.mat'], 'freqs', 'trace', 'str', 'config');
405 end
406 obj.h_fig = hfig;
407
408 end

```



```

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
427         % the lowest ...
428         power of ...
429         E5071C,
430         % hence this
431         % parameter only
432         % takes ...
433         effect if it
434         % is given a ...
435         valid
436         % value
437     p.addParameter('center', 0, @isnumeric); % frequency ...
438         sweep can also be defined
439
440         % by center and ...
441         span,
442         % if they are ...
443         given a
444         % valid value
445
446     p.addParameter('span',0, @isnumeric);
447
448     % useless, but required for input parser to be identical ...
449     with input parser for manualSweep;
450     p.addParameter('issavedata',true,@islogical);
451     p.addParameter('hfig',233,@isnumeric);    % figure handle
452     p.addParameter('notes','',@ischar);       % notes to ...
453         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 + ...
            freqSectionSpan* (i-1);
471
472         tmpfreqs = obj.freqs;
473         freqs((1 + (i-1)*points):(i*points)) = tmpfreqs;
474     end
475
476 end
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 ...
        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 ] = ...
      vna.fit('fitall',true,'issavefig',false,...
487 %
      ...
      'xdata',freqs,'ydata',20*log10(abs(SParams)),...
488 %
      ...
      'titleNotes','_pow_-10dBm' );
489 %
490 % See the inputParser below for more options
491 %
492
493 p = inputParser;
494 p.addParameter('issavefig',true,@islogical);% if true, ...
      save fig to png file
495 p.addParameter('fitall', false,@islogical); % if true, ...
      fit all
496
497                                     % plotted data,
498                                     % else ask two
499                                     % input for ...
500                                     % range
      p.addParameter('xdata',[],@isnumeric); % xdata in ...
      GHz frequency
501 p.addParameter('ydata',[],@isnumeric); % ydata given ...
      in dB
502 p.addParameter('titleNotes','',@ischar); % notes to ...
      add to the figure title
503 p.addParameter('QGuess',1e5,@isnumeric);
504 p.parse(varargin{:});
505 expandStructure(p.Results);
506
507 dataObj = get(gca,'children');
508 if isempty(xdata)
509     xdata = get(dataObj,'xdata');
510 end
511 if isempty(ydata)
512     ydata = get(dataObj,'ydata');
513 end
514
515 if gcf == obj.h_fig
516     figure(obj.h_fig);
517 else
518     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
527     end
528
529     if fitall
530         leftInd = 1;
531         rightInd = length(xdata);
532     else
533         fprintf('Select the X range for fitting:');
534         tmpPoints = ginput(2);
535         leftX = min(tmpPoints(:,1));
536         rightX = max(tmpPoints(:,1));
537         leftInd = find(leftX < xdata, 1);
538         rightInd = find(rightX < xdata, 1);
539     end
540
541     t = xdata(:);
542     y = ydata(:);
543     % assume xdata given in GHz
544     t = t(leftInd:rightInd)*1e9;
545     % assume ydata given in dB, convert to linear
546     y = 10.^(y(leftInd:rightInd)./20);
547
548     % guess initial parameters
549     peakInd = find(abs(ydata) >= max(abs(ydata)), 1);
550     freq0 = xdata(peakInd); % in GHz
551     x1 = [freq0, QGuess/1e4, QGuess/1e4, 0, 0, 0];
552
553     % fit with complex S21 deduced theoretically
554     % 8 parameter, linear base
555     F = ...
556         @(x, xdata) (20.*log10(abs(x(6).*(1+x(5).*(xdata-x(1).*1e9)./(x(1).*1e9)).*(1+...
557             + ...
558             x(3).^2.*1e4./cos(x(4)))./(x(3).^2.*1e4).*(cos(x(4))+1i.*sin(x(4)))./(1+2.*...
559             + ...
560             x(3).^2.*1e4./cos(x(4))).*(xdata-x(1).*1e9)./(x(1).*1e9))))+x(7).*xdata.*1...
561
562     % 7 parameter, constant base
563     F = ...
564         @(x, xdata) (abs(x(6).*(1+x(5).*(xdata-x(1).*1e9)./(x(1).*1e9)).*(1-(x(2).^2...
565             + ...
566             x(3).^2.*1e4./cos(x(4)))./(x(3).^2.*1e4).*(cos(x(4))+1i.*sin(x(4)))./(1+2.*...
567             + ...

```

```

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) = ...
      (x(2).*x(3)./cos(x(4)))./(x(2) + x(3)./cos(x(4))), in 1e4
557 %x(3): |Qe|, parameter Q, 1/Qc = Re (1/Qe) = ...
      cos(theta)/Qe, in 1e4
558 %x(4): theta, phase of parameter Q
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

```

## C.2 PPMS 控制代码

```

1 classdef PPMS < handle
2     % PPMS describe and control QDInstrument DynaCool at IIIS via ...
3     % QDInstrument.dll,
4     % which is much faster than using dll created from LabVIEW.
5     %
6     % 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 prirvate 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
69     FIELD_MODE = containers.Map({'Persistent', 'Driven'}, [0,1]);
70
71 end
72
73 properties
74     dllfilepath
75     address
76     isremote
77     instrType
78
79     QDInstr
80     vi
81
82     temp
83     field
84     tempStatus
85     fieldStatus
86
87     tempApproach
88     tempRate
89     fieldMode
90     fieldRate
91     fieldApproach
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',@ischar);
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        p.parse(varargin{:})
103        expandStructure(p.Results);
104
105        obj.address = address;
106        obj.dllfilepath = dllfilepath;
107        obj.isremote = isremote;
108        obj.instrType = instrType;
109        obj.tempApproach = 'Unknown';
110        obj.fieldMode = 'Unknown';
111        obj.fieldApproach = 'Unknown';
112        obj.tempRate = NaN;
113        obj.fieldRate = NaN;
114
115        obj.QDInstr = NET.addAssembly(dllfilepath);
116        pause(1);
117        obj.tempStatus = ...
118            QuantumDesign.QDInstrument.TemperatureStatus.TemperatureUnknown;
119        obj.fieldStatus = ...
120            QuantumDesign.QDInstrument.FieldStatus.MagnetUnknown;
121
122        % initialize .NET object which is the vi for the ppms
123        obj.vi = ...
124            QuantumDesign.QDInstrument.QDInstrumentFactory.GetQDInstrument(...
125                QuantumDesign.QDInstrument.QDInstrumentType.(instrType),...
126                isremote,address,uint16(11000) );
127        fprintf('PPMS %s at %s connected.\n',obj.instrType, obj.address);
128
129    end
130
131    %% set & get temperature
132    function value = get.temp(obj)
133        [~, value, obj.tempStatus] = GetTemperature(obj.vi, ...
134            double(0), obj.tempStatus);

```

```

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

```

## C.3 扫描温度与磁场实验的代码

### C.3.1 扫描温度

```
1 %%
2 function [tempStableTmr, setTempTmr] = TempSweep(ppms,vna,temps,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 %
25 % ATTENTION: the waiting process (for the temperature 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 % In development (May 13, 2017)
31 %
32 % TODO:
33 %   Done. add input parser
34 %   Done. choose vna sweep mode. Change 'plotTrace' to 'fetchTrace'
35 %   Done. choose temperature wait mode ('Near' or 'Stable')
36 %   Done. fetch actual temperatures and add timeStamps;
37 %
38 % Wentao, May 2017
39 %
```

```

40 p = inputParser;
41 p.addParameter('vnaMode','plotTrace',@ischar);      % fetchTrace will ...
    directly fetch vna trace.
42
    % Else, the ...
    E5071C/manualSweep ...
    method is called
43 p.addParameter('checkTempPeriod',10,@isnumeric);
44 p.addParameter('tempWaitMode','Stable',@ischar);    % another option ...
    is 'Near'
45 p.addParameter('manualSweepConfig',struct([]),@isstruct); % ...
    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 tempStableTmr = timer;
57 setTempTmr = timer;
58 set(tempStableTmr,'ExecutionMode','fixedRate');
59 set(tempStableTmr,'period',checkTempPeriod);        % check temperature ...
    stable period
60 set(tempStableTmr,'TimerFcn',@timerCalled);
61 set(tempStableTmr,'userdata',struct('ppms',ppms,'numOfTemps',length(temps),...
    'manualSweepConfig',manualSweepConfig,...
62     'tempWaitMode',tempWaitMode,...
63     'vnaMode',vnaMode,...
64     'freqs',freqs,...
65     'timeStamps',NaN(1,length(temps)),...
66     'actualTemps',NaN(1,length(temps)),...
67     'vna',vna,'t',setTempTmr,'waitTime',waitTime,...
68     'cnt',1,'SParams',NaN(length(temps),length(freqs)) ...
69         ));
70
71 set(setTempTmr,'userdata',struct('ppms',ppms,'t',tempStableTmr,'temps',temps,'cnt',1) ...
    );
72 set(setTempTmr,'TimerFcn',@setTempTimerCalled);
73
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(:)' + 1i*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 %s. 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

```

### C.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 = ...
9 %           0.1 mT)
10 %   varargin: optional input arguments, see inputParser below for detail
11 %
12 % Returns:
13 %   Two timer object tempStableTmr and setTempTmr.
14 %

```

```

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 ...
    remained
18 % in the memory:
19 %     delete(fldStableTmr);
20 %     delete(setFldTmr);
21 %     clear fldStableTmr;
22 %     clear setFldTmr;
23 %
24 % ATTENTION: the waiting process (for the field 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 16, 2017)
30 %
31 % Wentao, May 2017
32 %
33
34 p = inputParser;
35 p.addParameter('vnaMode','plotTrace',@ischar);      % fetchTrace will ...
    directly fetch vna trace.
36
    % Else, the ...
    E5071C/manualSweep ...
    method is called
37 p.addParameter('checkFldPeriod',5,@isnumeric);
38 p.addParameter('fldWaitMode','StableDriven',@ischar); % another ...
    option is 'Near'
39 p.addParameter('manualSweepConfig',struct([]),@isstruct); % ...
    parameters for the E5071C/manualSweep method as a structure
40
41 p.parse(varargin{:});
42 expandStructure(p.Results);
43
44 if strcmpi(vnaMode, 'plotTrace')
45     freqs = vna.freqs;
46     waitTime = vna.sweepTime;
47 else
48     [freqs,waitTime] = vna.manualSweepFreqs(manualSweepConfig);
49 end
50 fldStableTmr = timer;
51 setFldTmr = timer;
52 set(fldStableTmr,'ExecutionMode','fixedRate');
53 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
66 set(setFldTmr, 'userdata', struct('ppms', ppms, 't', fldStableTmr, ...
67                                   'BFields', BFields, 'cnt', 1) );
68
69 set(setFldTmr, 'TimerFcn', @setFldTimerCalled);
70
71 start(setFldTmr);
72 end
73
74 %%
75 function timerCalled(thisObj, event)
76 ud = thisObj.UserData;
77 expandStructure(ud);
78 stat = char(ToString(ppms.fieldStatus));
79 if strcmpi(stat, fldWaitMode)
80     fprintf('Field %s, start vna averaging (%.2fs)...\n', ...
81            fldWaitMode, waitTime+0.1)
82
83     % vna sweep
84     if strcmpi(vnaMode, 'plotTrace')
85         vna.clearAvg;
86         pause(waitTime+0.1);
87         trace = vna.trace;
88     else
89         [~, trace] = vna.manualSweep(manualSweepConfig);
90     end
91     fprintf('vna trace fetched.\n');
92
93     % save SParams back into timer
94     ud.actualFields(cnt) = ppms.field;
95     ud.timeStamps(cnt) = now;
96     ud.SParams(cnt,:) = trace.X(:)' + 1i*trace.Y(:)';
97     ud.cnt = cnt + 1;
98     thisObj.UserData = ud;
99     stop(thisObj);
100    fprintf('Field timer stopped at %s.\n', datestr(now, 30))

```



## 附录 D 外文资料原文

## 附录 E 外文资料的调研阅读报告或书面翻译

英文资料的中文标题

**摘要：**本章为外文资料翻译内容。如果有摘要可以直接写上来，这部分好像没有明确的规定。

## 附录 F 其它附录

前面两个附录主要是给本科生做例子。其它附录的内容可以放到这里，当然如果你愿意，可以把这部分也放到独立的文件中，然后将其 `\input` 到主文件中。

## 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)