

北京郵電大學

本科 毕业 设计（论文）



题目：基于视觉诱发电位脑-机接口的机械臂控制系统设计¹

姓 名 秦羽
学 院 信息与通信工程学院
专 业 信息工程
班 级 2016211128
学 号 2016210757
班内序号 01
指导教师 杨晨

2020 年 5 月

¹ 注：项目来源于科研项目“2019 教基张洪欣”，项目编码 2019XD-A17

北京邮电大学 本科毕业设计（论文）诚信声明

本人声明所呈交的毕业设计（论文），题目《基于视觉诱发电位脑-机接口的机械臂控制系统设计》是本人在指导教师的指导下，独立进行研究工作所取得的成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：  日期： 2020.05.25

基于视觉诱发电位脑-机接口的机械臂控制系统设计

摘要

视觉诱发电位是一种由视觉刺激所诱发特殊神经响应。它自二十世纪中叶被首次发现，迄今已经历多年的研究。其中稳态视觉诱发电位是视觉诱发电位中相对典型的响应模式，具有信噪比高、稳定性好、易识别等优点，从而被广泛应用于脑-机接口控制研究领域。

本文在现有稳态视觉诱发电位脑-机接口系统基础上融合机器人操作系统(ROS)，实现了将人脑的脑电信号直接转换为机械臂控制指令的功能，从而解决了因肢体失能而导致的设备控制障碍问题。

本文主要包括机械臂控制范式设计研究，基于脑-机接口系统的步进式机械臂控制模式研究，基于 moveit 组件的仿真运动研究以及面向实际应用场景的机械臂抓取实验研究四方面研究内容。

在机械臂控制范式设计方面，本文根据机械臂运动特征，设计了一套包含有 9 种指令的稳态视觉诱发电位刺激范式，实现了前伸、后伸、左转、右转、上抬、下移、抓取等 7 种操作模式及复位、结束两种控制功能。

在机械臂控制模式方面，根据多自由度舵机特性及实际参数，设计了一套机械臂步进式控制模式。通过步进式指令操作，同时控制多个舵机的协同运转，实现对机械臂姿态的精确控制。

为了更好地调试机械臂操作方案，本文基于 moveit 组件开展了机械臂仿真运动研究。在 moveit 组件种建立了机械臂仿真模型，能够在虚拟环境下实现机械臂的移动抓取指令，有效降低了真实场景下硬件设备的调试成本。

最终，为了验证基于视觉诱发电位脑-机接口的机械臂控制系统的有效性，本文进行了真实场景下机械臂抓取实验测试。在真实的脑电信号控制下，机械臂成功完成了空间移动及物体抓取操作，表明整体系统性能达到了预期设计水平。

综上所述，本文基于稳态视觉诱发电位脑-机接口设计了一整套机械臂的控制系统，有望协助肢体失能或运动障碍人士提高生活质量。

关键词 脑-机接口 稳态视觉诱发电位 机器人操作系统 机械臂

An Assistant Robotic Arm Based On SSVEP Brain-Computer Interface

ABSTRACT

Visual Evoked Potentials(VEP) is a special neural response evoked by visual stimuli. It was first discovered since the middle of the twentieth century, and has experienced many years of research so far. Among them, Steady-State Visual Evoked Potentials(SSVEP) is a relatively typical response mode in visual evoked potential, which has the advantages of high signal-to-noise ratio, good stability, and easy recognition, which can be used to control the outer-world object combined with Brain-Computer Interface(BCI).

This paper integrates the robot operating system (ROS) on the basis of the existing steady-state visual evoked potential brain-machine interface system and realizes the function of directly converting the brain's EEG signals into robotic arm control commands, thereby hoping to help the disabled individuals, with a physical disease or neural disease like dyskinesia, to partly restore their quality of life.

This paper mainly includes the research on the design of the control paradigm of the manipulator, the research on the control mode of the stepped manipulator based on the brain-machine interface system, the simulation motion research based on the moveit component and the experimental research on the manipulator grasping for practical application scenarios.

In terms of the design of the manipulator control paradigm, this paper designed a set of steady-state visual evoked potential stimulus paradigms containing 9 kinds of commands according to the movement characteristics of the manipulator, and realized the forward extension, backward extension, left turn, right turn, uplift, 7 operation modes such as down move and grab, and two control functions of reset and end.

In terms of the control mode of the manipulator, according to the characteristics and actual parameters of the multi-degree-of-freedom steering gear, a set of manipulator step control mode is designed. Through the step-by-step command operation, the cooperative operation of multiple servos is simultaneously controlled to achieve precise control of the attitude of the robot arm.

In order to better debug the operation scheme of the manipulator, this paper carried out the manipulator simulation motion research based on the moveit component. The robot arm simulation model is established in the moveit component type, which can realize the movement of the robot arm grab command in the virtual environment, which effectively reduces the debugging cost of the hardware device in the real scene.

Finally, in order to verify the effectiveness of the robotic arm control system based on the visual evoked potential brain-machine interface, this paper carried out the experimental test of the robotic arm grabbing under the real scene. Under the control of real EEG signals, the robotic arm successfully completed space movement and object grasping operations, indicating that the overall system performance reached the expected design level.

In summary, based on the SSVEP brain-machine interface, this paper designs a complete set of the robotic arm control system, which is expected to help people with physical

disabilities or movement disorders improve their quality of life.

KEY WORDS brain-computer interface steady-state visual evoked potential robot operating system robotic arm

目 录

第一章 引言	1
1.1 脑-机接口简介及研究进展.....	1
1.2 机器人操作系统（ROS）简介	3
1.3 当前国内机械臂控制存在的局限及问题	4
1.4 基于脑-机接口控制机械臂未来应用价值.....	4
1.5 研究目标及论文结构	5
第二章 基于稳态视觉诱发电位的脑-机接口及 ROS 系统原理	6
2.1 稳态视觉诱发电位（SSVEP）脑-机接口原理.....	6
2.1.1 稳态视觉诱发电位原理及特征	6
2.1.2 稳态视觉诱发电位脑-机接口常用刺激范式.....	7
2.1.3 常用稳态视觉诱发电位脑-机接口数学模型及解码算法.....	8
2.2 ROS 系统基础知识	10
2.2.1 系统基本结构	10
2.2.2 Moveit 简介	12
第三章 针对机械臂交互的脑电刺激范式设计	13
3.1 本章简介	13
3.2 视觉交互范式设计	13
3.3 刺激频率及相位选择	15
3.4 视觉刺激界面生成 MATLAB 实现方案	16
3.5 本章小结	17
第四章 基于 ROS 系统的机械臂交互方案设计	18
4.1 本章简介	18
4.2 基于 ROS 系统的机械臂控制架构设计	18
4.3 基于 MOVEIT 组件机械臂移动方案	19
4.4 基于 MOVEIT 组件机械臂建模及运动仿真	21
4.5 本章小结	23
第五章 在线脑-机接口控制机械臂系统设计及实验验证	24
5.1 本章引言	24
5.2 SSVEP 在线架构及整体控制方案	24
5.3 脑-机接口-ROS 系统通信接口设计及调用方法	27
5.4 数据采集实验验证及系统测试	28

5.4.1	实验设计	28
5.4.2	测试评价结果	29
第六章	总结	31
参考文献	32
致 谢	34
附 录	35

第一章 引言

1.1 脑-机接口简介及研究进展

得益于近年来神经科学及计算科学的快速发展，脑-机接口(Brain-Computer Interface,BCI)逐渐成为世界科学中的研究热点。脑-机接口能够帮助部分残障人士恢复部分因为神经性疾病而失去的身体机能，目前较多应用于肌萎缩侧索硬化(ALS)中风、以及脊髓受损(SCI)人群。已经有一些研究通过脑-机接口技术帮助上述失能人群进行康复训练或者用机械进行代替实行正常功能^[1-5]。

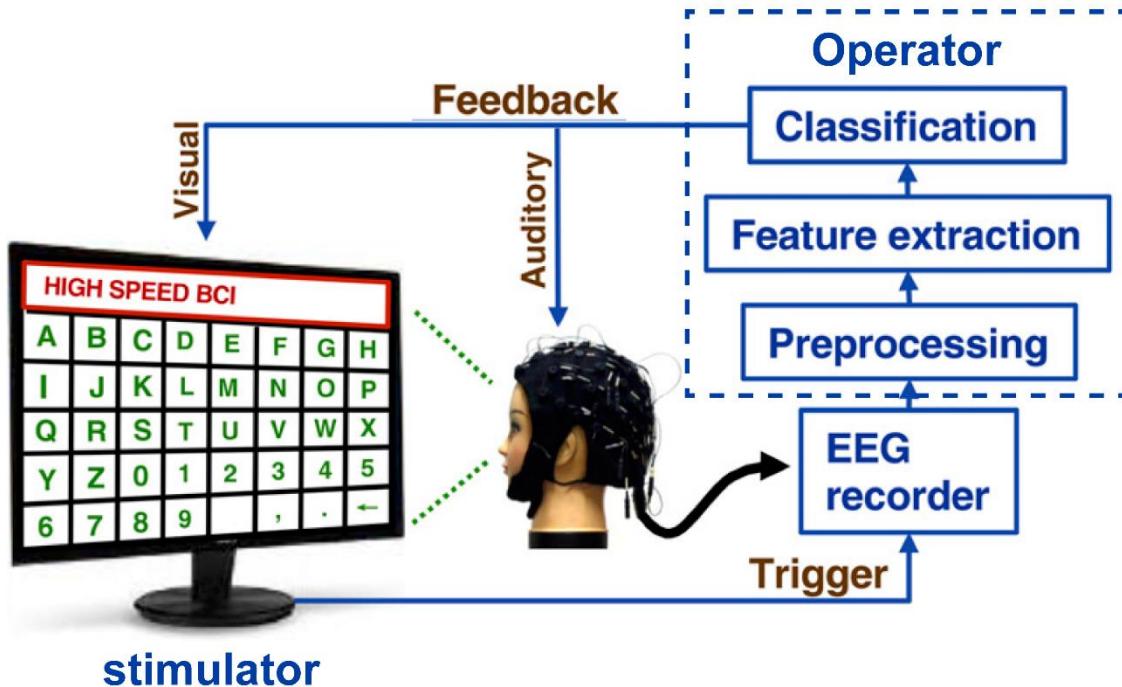
脑-机接口概念于上世纪 70 年代由 Vidal 提出第一次提出。Vidal 在该论文中第一次探讨了用脑电波控制外界物体的可能性^[6]。随后在 Miguel Nicolelis 教授等知名学者的大力推动下，脑-机接口帮助残疾人在 2014 年巴西世界杯首次开球，让脑-机接口进入真正进入公众视野^[7]。清华大学的高上凯及高小榕教授为我国脑-机接口领域早期研究者，他们对非侵入式脑-机接口的实用发展做出了极大的贡献^[8]。

脑-机接口，是指利用人体大脑的脑电作为信号发生源来操控其它可受控系统。脑-机接口有按照是否对人体造成物理性损伤，可以分为侵入式和非侵入式两类。其中，侵入式是需要动用外科手术来有创面地植入设备；而非侵入式的就是不需要动手术只需要佩戴设备即可。本文在这里采用的是能够快速上手不需要解剖背景的非侵入式设备，典型导电帽如图 1-1 所示。



图 1-1 典型脑电帽示例^[9]

在脑电信号上，非侵入式采集信号的方法有很多种选择，头皮脑电波(EEG)，利用磁电的(MEG),核磁共振(fMRI)都可以列为非侵入式的设备，而我此次实验选择的是利用 EEG 信号，原因是 EEG 信号具有无损伤、高时间分辨率、低成本特点。针对 EEG 信号的常用的范式有 P300^[10], Steady-state Visual Evoked Potentials(SSVEP)^[11], Motor Imagery(MI)^[12]。本毕设采用基于 SSVEP 的脑-机接口为机械臂提供控制。

图 1-2 SSVEP 典型应用示例^[13]

一个完整的 SSVEP 应用流程如图 1-2 所示。分为视觉刺激端，数据采集，预处理，特征提取，模式识别最终输出结果并反馈及后续应用处理。本文重点为视觉刺激端范式设计以及后续应用，其余如模式识别算法则需要对数学原理进行了解并掌握即可。

近期，脑-机接口较为瞩目的成果有 Grégoire Courtine 等人解码受损脊椎上端从大脑传递来的运动信号，利用电刺激使下身瘫痪的人重新站立起来^[14]，Edward Chang 等人成功利用 BCI 和人工神经网络结合利用脑电波翻译出语音信号^[15]，B. He 等人设计了更好的基于运动想象范式的控制方法使机械臂能更连续的按照想象路线移动^[16]。这些都是最近两年较突出的成果。而整个领域研究可分为针对识别算法，应用场景，脑电信号采集设备，信号处理芯片等不同的研究方向。

1.2 机器人操作系统（ROS）简介

ROS 系统是 2010 年诞生于斯坦福大学的专门用于控制机器人的操作系统，全名 Robot Operating System^[17]。ROS 系统诞生之前，机器臂等控制方案大多存在兼容性差、可移植性欠佳等问题。例如，当更换机械臂的接口或传感器时，需要重构控制程序。而现今对于控制机器人相关硬件而言，可移植性及接口最规范的控制系统是 ROS 系统。ROS 已经有 10 年的发展历史，诞生于斯坦福之后走上了成熟的市场化道路。因此选用该系统也是为了实验室后续能够在我的基础上继续利用代码。

ROS 是一个基于 linux 操作系统开发的附着其上的针对机器人开发的操作系统，说是操作系统但和计算机操作系统仍有区别。可理解为 kernel 针对的底层交互硬件从 CPU 变为外置的机器人设备，可以方便快捷的调用系统内置接口完成本文所需要的控制机械臂的路径规划等任务，无需再自己深入重新写底层算法，同时 ROS 系统是全部开源的，也有源源不断的创作者在为 ROS 系统设计更好的封装库^[18-22]。这对于本科毕设开发来说是极为方便的，网上也有相应的教程，因此选用该系统。

ROS 系统目前应用较多的领域是科研领域。在工业领域应用较少原因一是底层硬件交互仍然不及直接使用嵌入式系统来得更高效，二是目前 ROS 系统并未针对工业领域应用进行软件设计，在即将到来的版本中将有专门针对工业工作的设计。值得期待。

图 1-3 为一个 ROS 系统简单应用建构的示例，上面 ROSMaster 为 ROS 必不可少的主控节点，由 master 来创建下面的节点并定义他们的沟通方式。在此图中各个节点间采用消息和订阅的方式来通信，ROS 各个节点会行使一些功能，比如控制小车前进后退，并会和各个节点进行沟通。比如节点 1 是电机控制，节点 2 是雷达探测传感器用来判断前方能否前进，这样就构建了一个简单的应用实例。第二、四章会深入介绍 ROS 相关具体工作。

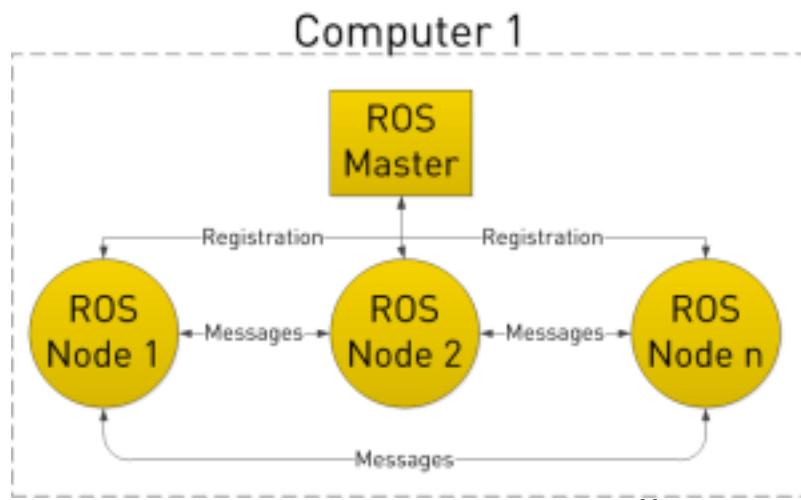


图 1-3 ROS 系统简单应用架构^[8]

1.3 当前国内机械臂控制存在的局限及问题

当前国内市场中存在的民用机械臂的控制设计尚处于初级阶段，缺乏真正面向失能人群的产品。现有产品对于需要借助设备大多需要使用者进行手动控制，导致与失能人群的整体交互效果不够理想。同时，针对渐冻人群体除了脑-机接口外并没有其它更好的选择，因为渐冻人几乎全身肌肉都无法移动，而国内并没有相关公司生产基于脑-机接口控制的辅助机械产品。因此渐冻人群的生活质量仍有很大空间需要提高。

同样不光残障人士使用时功能受限，健康人在特殊环境下的交互也可能因为需要动手操作而被影响。一个较能说明问题的例子是宇航员在太空中处于零重力零摩擦空间，任何微小的施力操作都将产生反作用力造成位移，其中就包括和屏幕交互的按键操作。而脑-机接口能够有效解决这个问题，脑-机接口能够直接读取使用者脑电波，而不需要任何物理外力即可完成指令输入。因此可以完美解决在失重环境下的反作用力漂移问题。

当前网上可以开源获得的机械臂控制方案没有建立专门脑-机接口无线通信的统一标准，同时没有专门针对 SSVEP 脑-机接口系统指令的范式及接口设计，对肢体失能人群支持欠佳。因此本项目拟在基于脑-机接口的机械臂控制系统展开研究。

1.4 基于脑-机接口控制机械臂未来应用价值

脑-机接口能够为患有影响行动能力或认知能力的神经类疾病患者提供有效帮助，因此，未来很有可能将成为残障人士重新拥有正常身体功能的重要帮手。

目前脑-机接口已经被广泛应用于医学诊疗及康复辅助。例如诊断及治疗精神或神经类疾病比如深度脑部刺激^[23]，情绪识别及相应软件应用^[24]，VR 场景中的控制方案大部分是用来控制游戏代替鼠标^[25]等。此外，法国的医生和高校共同合作在 2019 年成功实现了侵入式接口控制外骨骼机器人使得残障人士重新获得正常行走能力^[26]。

同时也有研究针对健康人研制额外肢体，以协助其从事多注意力任务。国外已有研究已经成功在断肢上恢复了力度反馈功能^[27]。许多人们想象中场景，正在通过脑-机接口不断变为现实。脑-机接口中很多新的研究方向也已经破土而出，即将开枝散叶、深入发展。

相比于国外的较高的研究成果转化率，我国的科研成果转化水平尚不尽如人意。最近我国也有名为铁甲钢拳的公司致力于开发使健康人得到增强的机械外骨骼设备，诸如博睿康和 Brain-Co 等公司也致力于开发非侵入式脑-机接口产品开拓市场。脑-机接口控制机械臂被视为一种实用性和市场潜力均较好的解决方案，因此，本次毕设希望依照此技术路线搭建一个简单但完整的基于非侵入脑-机接口的脑控机械臂的系统。

1.5 研究目标及论文结构

本次毕设希望完成的是利用实验室已经较完善的异步脑-机接口系统架构来设计一套用以控制机械臂的完整系统，以达到帮助残障人士及上述疾病人士弥补身体上的不足。本研究将基于杨晨博士设计的异步脑-机接口框架^[12]，并结合 ROS 系统实现机械臂完整控制系统。其中脑-机接口算法部分将采用实验室内成熟技术，而本文重点在于面向机械臂控制的脑-机接口刺激范式及机械臂控制模式研究。

论文第一章为引言，简要介绍脑-机接口，机械臂控制选用 ROS 的原因以及研究价值及内容。

第二章详细介绍 SSVEP 原理及识别的数学原理以及 ROS 系统的基本原理。

之后为本次毕设主要的工作内容。

第三章开始为针对利用 SSVEP 信号控制机械臂设计的脑-机接口视觉刺激交互界面的相关设计工作。

第四章为 ROS 端编程实现及机械臂控制架构设计。

第五章为脑-机接口控制机械臂整体架构设计，通信接口及具体实验。

第六章总结实验及设计过程的心得以及后续的可能工作。

第二章 基于稳态视觉诱发电位的脑-机接口及 ROS 系统原理

2.1 稳态视觉诱发电位（SSVEP）脑-机接口原理

2.1.1 稳态视觉诱发电位原理及特征

视觉诱发电位(VEP)是指视网膜在接受外界光信号刺激后，经过外侧膝状体核(lateral geniculate nucleus, LGN)传递后，在枕区视觉皮层所产生的响应信号。稳态视觉诱发电位(steady-state visual evoked potentials, SSVEP)是视觉诱发电位中的一种。当人类在注视（有意识的关注）着某一固定频率的闪烁光源时，大脑视觉皮层会在对应区域产生与光源频率相关的响应信号，其中主要包含以光源频率为基频的多倍频谐波响应信号。而稳态视觉诱发电位通常指采用大于4赫兹的稳定激励源刺激视网膜时所诱发的周期性响应信号。图2-1中为典型的SSVEP响应信号的时域图与频域图，可以看到频域中有一个频段产生了显著的响应。

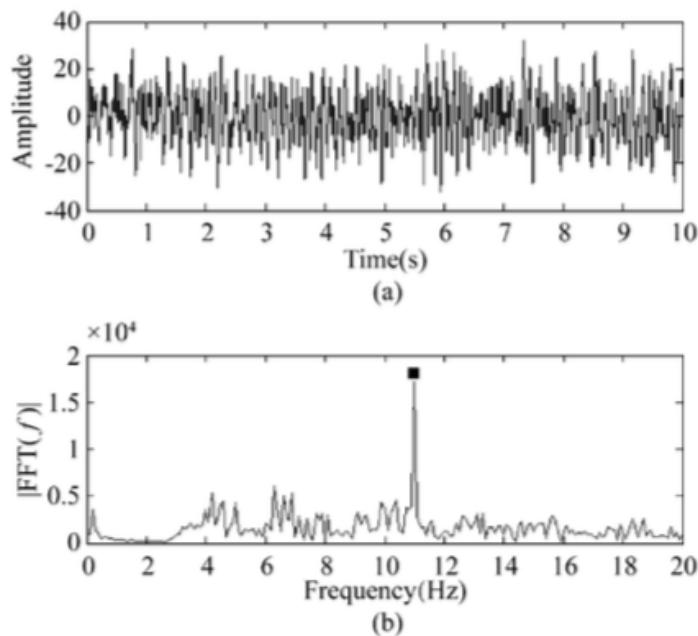


图 2-1 典型的 SSVEP 响应信号 (a) 及频谱 (b)^[13]

据研究，SSVEP只有部分频段的刺激源会激起较为明显的脑电响应，可以在期望意义上（个体间差异可能会较大）按照刺激频段分为低频段(15赫兹以下)，中频段(15-30赫兹)，高频段(30赫兹以上)三类。而低频段内的响应最明显^[29]。故在设计时可以选择对应频段的刺激源来应用。

还有一个需要回答的问题是在什么位置可以采集到对应的本文所需要的响应信号。国际上采用 10-10 系统对采集信号的区域进行了统一的规范命名，如图 2-2 所示。较好的信号采集点为通常被认为是处理视觉信息的枕叶区域^[30]。对应于 P 和 O 开头的一片区域。

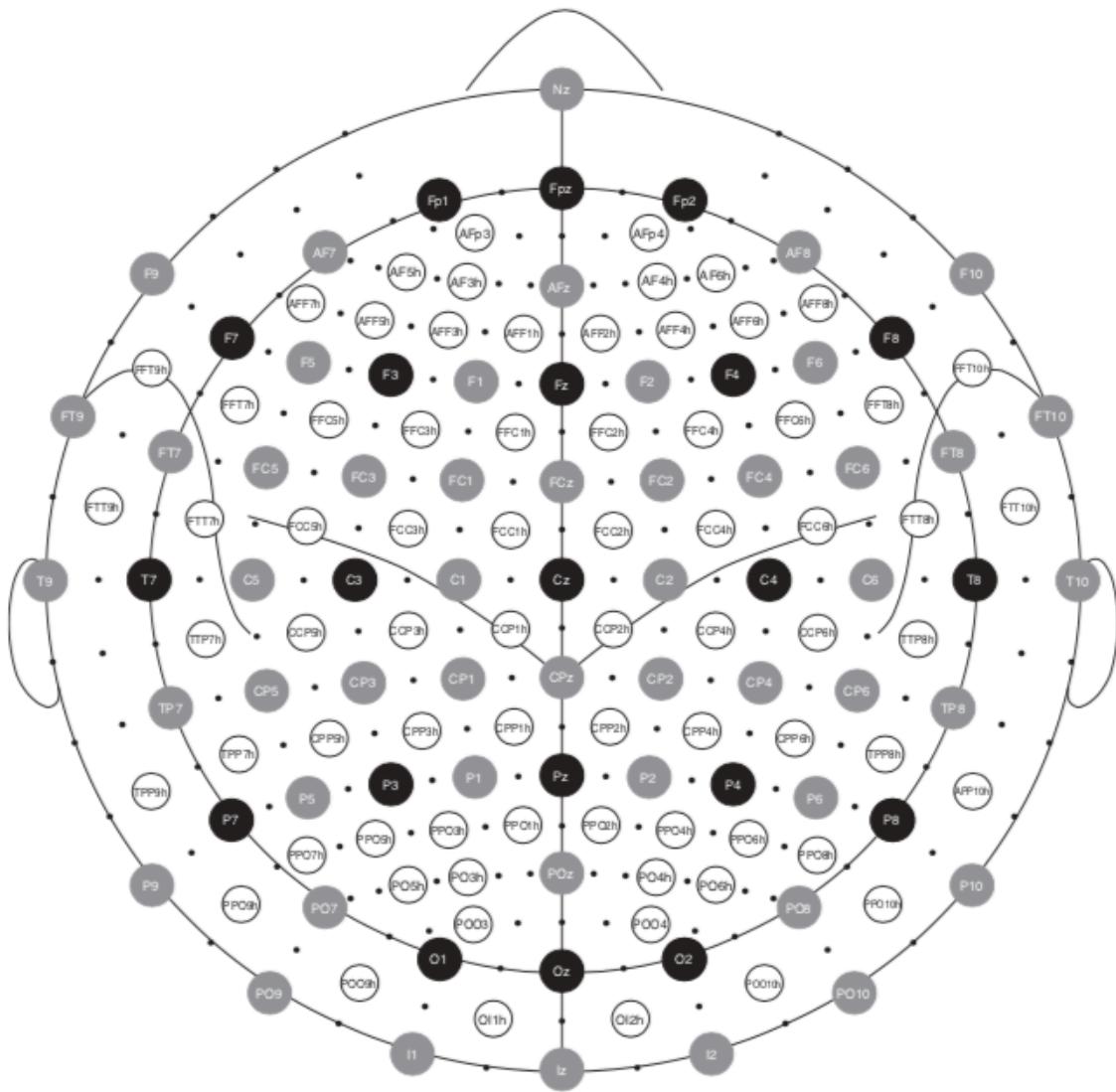


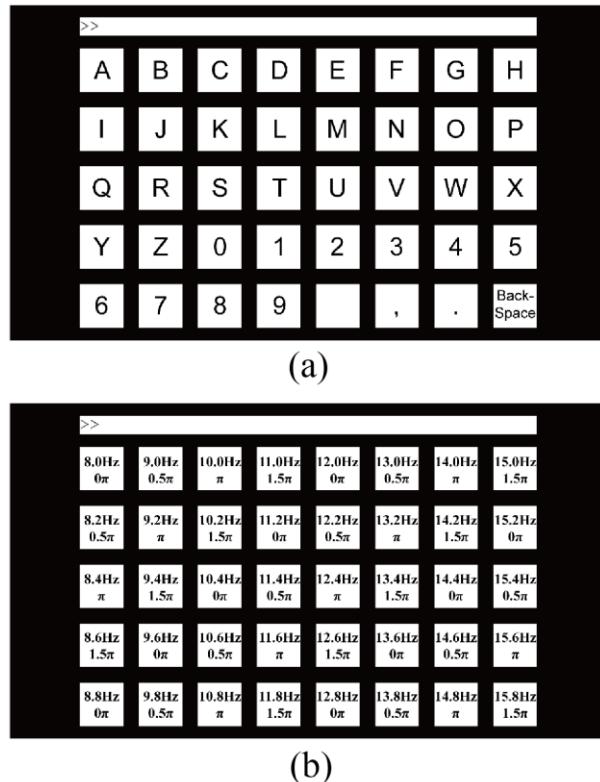
图 2-2 10-10 系统电极名称划分示例^[9]

因此本文可以利用不同的频率来区分不同的指令，同时在后来的研究中不同相位被证明也是可以有效区分信号源的方式^[29]。利用上述特征本文可以从而利用不同的信号产生源来使大脑产生不同的响应，从而实现可区分的控制。

2.1.2 稳态视觉诱发电位脑-机接口常用刺激范式

一个典型的打字应用的视觉刺激范式如图 2-3 所示，(a)为键盘的布局，(b)为各个字符对应的频率和相位。各个字符在 LCD 屏幕上同步进行闪烁，每个字符的亮度随时

间进行正弦函数的变化从而达到稳态激励源的目的。图中为 40 目标的打字系统，在视觉上呈现多个目标时，人类的注意力注意到哪个闪烁频率上，对应的 SSVEP 频点就会最显著。从而达到识别各个字符的目的。黑白颜色的交互界面会使得人视觉疲劳感没有那么强烈，方便长时间使用。形状上的选择遵循个人喜好以及用户常见的形状即可，目的在于舒适。

图 2-3 典型 SSVEP-BCI 范式^[16]

2.1.3 常用稳态视觉诱发电位脑-机接口数学模型及解码算法

有用的视觉诱发脑电信号可以认为是有效响应信号叠加噪声，噪声被认为来源于周围环境及平常就会一直存在的自发脑电，数学建模为如式 2-1

$$\mathbf{x}(n) = \mathbf{s}(n) + \mathbf{w}(n), n = 12 \dots N \quad \text{式 (2-1)}$$

N 为采样点总数 = 采样频率*实验时长， $\mathbf{x} \in \mathbb{R}^{L \times N}$, L 为通道数目，每个通道对应一个采集脑电的电极。 $\mathbf{s} \in \mathbb{R}^{L \times N}$ 代表我们所需要的 SSVEP 响应信号， $\mathbf{w} \in \mathbb{R}^{L \times N}$ 为噪声。

在此模型基础上，最常见也最简单的解码算法为典型相关关系 (CCA)^[28]，CCA 原理上为找到矩阵 \mathbf{x}, \mathbf{y} 从多维投影至某一个一维投影空间中的相关系数中的最大值，越相关值最终找到的最大值越小，具体的投影函数由 w_x, w_y 决定。本质上是一个先压缩再寻找相关系数的过程，而在压缩过程中会对信息有损失，但也简化了比对的计算复杂度。如式 2-2 所示。

$$\rho_{max}(\mathbf{x}, \mathbf{y}) = \max_{w_x, w_y} \frac{E(w_x^T x y^T w_y)}{\sqrt{E(w_x^T x x^T w_x) E(w_y^T y y^T w_y)}} \quad \text{式 (2-2)}$$

因为本文的最终目标是找到一个频率成分和当前这段脑电信号最相近的，所以必须先拥有关于目标频率成分的数学假设。具体到本文这个问题上：目前拥有的变量为需要被鉴定主频成分的 x ，暂时没有的为提供对比模板需要的变量 y 。因此常用的针对 SSVEP 的特定模板 \mathbf{y}_i 如式 2-3 所示。为一个 $L * N$ 的模板，每一行频率 $\omega_i = 2\pi f_i / f_s$ ， f_i 是选用的刺激频率， f_s 为屏幕刷新率。这样就可以在投影后进行比较是不是以 ω_i 为基频 n 倍频为谐波的响应。

$$\mathbf{y}_i = \frac{1}{\sqrt{N}} \begin{bmatrix} \sin(\omega_i 0) & \cdots & \sin(\omega_i (N-1)) \\ \cos(\omega_i 0) & \cdots & \cos(\omega_i (N-1)) \\ \sin(2\omega_i 0) & \cdots & \sin(2\omega_i (N-1)) \\ \cos(2\omega_i 0) & \cdots & \cos(2\omega_i (N-1)) \\ \vdots & \vdots & \vdots \\ \sin\left(\frac{L}{2}\omega_i 0\right) & \cdots & \sin\left(\frac{L}{2}\omega_i (N-1)\right) \\ \cos\left(\frac{L}{2}\omega_i 0\right) & \cdots & \cos\left(\frac{L}{2}\omega_i (N-1)\right) \end{bmatrix} \quad \text{式 (2-3)}$$

最终总的表达式如式(2-4)，从而求出本研究所接收到的目标 i 。

$$i = \arg \min_i \rho_{max}(\mathbf{x}, \mathbf{y}_i) \quad \text{式 (2-4)}$$

至此也就完成了识别任务。而针对式 2-2 相关最优化问题不打算在这里进行讨论，因为本质上是已经解决的数学问题对理解上并没有帮助。常用的方法拉格朗日乘子法，SVD 分解均可。经典的 CCA 算法优点是快而且不需要被试者进行训练学习操作。缺点是鲁棒性较差，过于简单。

同时针对噪声我们并没有进行过多的研究，在本研究的异步实时系统中采用的是过一段时间就会自动更新的 MVAR 模型来对噪声进行建模，同时在使用过程中会有一个动态变化长度的窗口用来判断当前信号中有无刺激响应信号，可以采用 CCA 或者统计假设检验的方式等一系列算法进行识别操作，具体内容参考论文^[13]。

2.2 ROS 系统基础知识

2.2.1 系统基本结构

Robot Operating System(ROS)机器人操作系统，通常会理解为一个独立的操作系统，但实际上名字中连起来应该是操作机器人的系统，操作是用来描述机器人的，和计算机领域的操作系统并非一个概念。他本身提供了一种和底层机械臂控制版交互的抽象实现，但同时依赖于 linux 的操作系统提供的底层功能接口。

ROS 官方称自己是工具、功能、框架、社区为一体的解决方案。架构本身采用分布式管理的方法，原则上每一个可执行的文件都对应一个节点。该节点的功能可多可少，最终的效果是：执行 n 个可执行的 c++ 或者 python 程序，每一个程序都有不同功能，如图 1-3 所示，每一个可以执行的文件都对应一个节点(node)，而每个节点都需要向 master 进行注册从而加入到当前这个项目中来。他们可以通过 ROS 预定义的接口来最终输出诸如驱动最终舵机的移动的指令，也可以所有程序与机器人毫无关系。这也是为什么 ROS 又称自己是一种框架，它不仅仅提供针对机械臂的操作。而每一个程序间的通信方式由 ROS 预定义的几种方式来通信。

下面简略介绍两种预定义的通信方式。一种为话题，一种为服务。

话题是一个节点发布消息，其余节点订阅获得消息，这是一种异步的通信方式。因为节点可以只在需要时去访问别的节点所发布的消息。开销较小。

服务则提供了一种同步的方式，双方在约定的时间间隔会同时刷新所需要的消息，同时被服务的节点还可以提供反馈来更改消息提供方提供的消息种类或者暂停。因为是每时每刻都在运转的通信程序故开销较话题大一些。

具体到应用中 ROS 拥有自己的工作空间，利用 python_rosdep 和 python_wstool 工具在指定文件夹开辟工作空间，在该空间中会有 src,build,devel 三个文件夹。Devel 中存放的是各种头文件及预封装的库。Src 为等待开发的文件库，在其中创建的每一个文件夹就相当于是一个完整功能程序，需要利用 ROS 基于 makefile 专门定制的 catkin 编译工具和 XML 文件来描述本文件夹下的关系。build 文件是在编译过程中系统自动生成文件的存放文件夹，如图 2-4 所示。

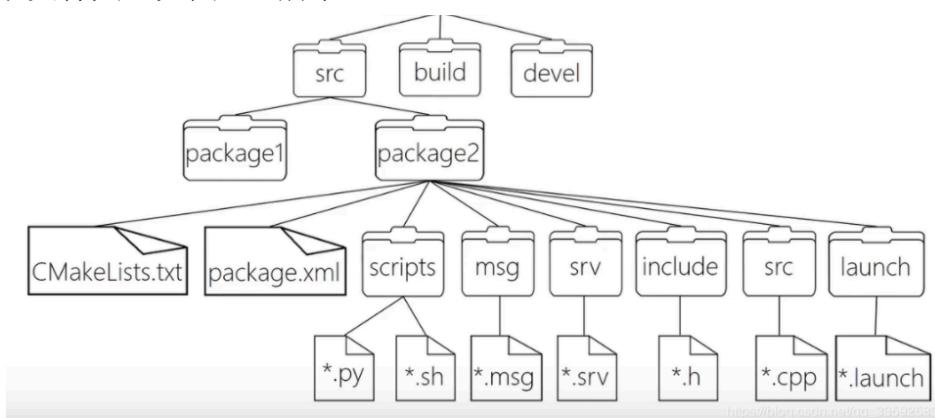


图 2-4 ROS 工作空间示例

其中 `devel` 文件存放的正是在 `package.xml` 中定义的需要一同编译到当前程序中的各种预定义的功能包。

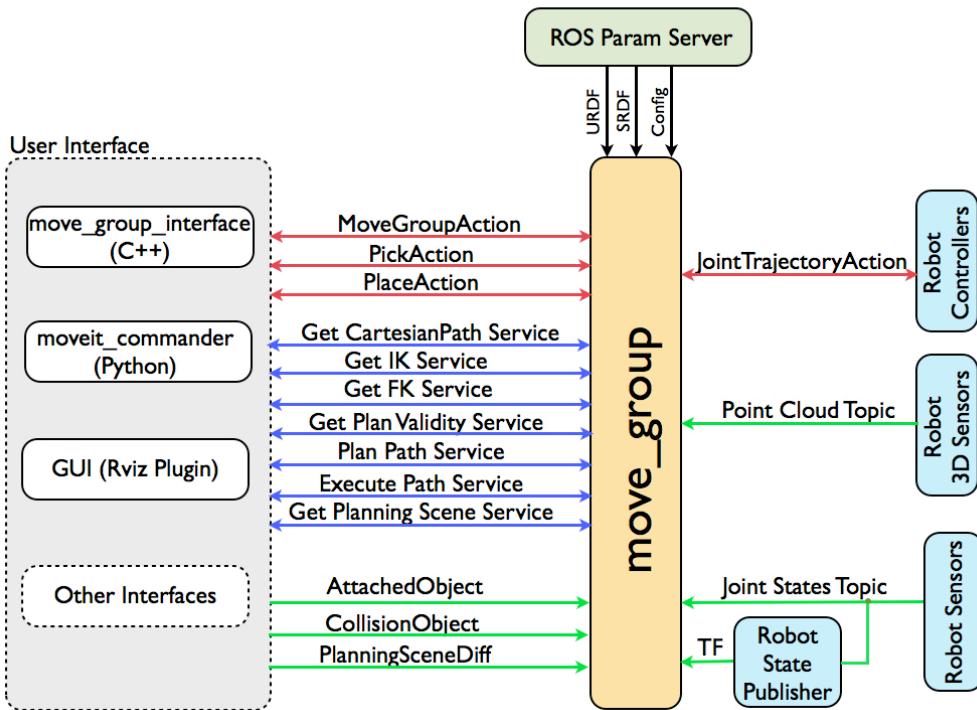
`Msg` 文件为通信时需要提供的消息的名称及类型定义。如图 1-3 所示，每次运行 ROS 程序都必须有 `master` 节点，同时必须分别在不同命令行里执行各个文件才能构成所需要的系统。

`Launch` 文件本质上是用来简化这个过程的，可以在 `launch` 中定义需要启动的所有文件从而一次性全部开启。

至此，也就是 ROS 基础架构的简单介绍，针对本毕设项目已经足够。

2.2.2 Moveit 简介

Moveit 是在 ROS 平台基础上开发的一个专门针对机器人或机械臂移动的工具。架构图如图 2-5 所示。每个小框都是一个节点。其中 move_group 为最关键的节点，负责提供本文所需要的路径规划及执行功能。本文的机械臂建模存储在 URDF 文件中。而 RViz 节点负责可视化。在使用时本研究只需要提供机械臂的建模信息以及所需要的控制指令，由 C++ 或者 python 文件调用。其余部分均为 moveit 自动提供的功能来帮助实现需要的事情。后文会用到 joint state 来获取当前机械臂各个舵机的状态，从而进行自己设计的调控。

图 2-5 Moveit 架构图^[31]

第三章 针对机械臂交互的脑电刺激范式设计

3.1 本章简介

本章为 SSVEP 脑-机接口的刺激范式设计。刺激范式的设计和最终呈现的功能以及刺激效果的好坏有着最直接的关系。好的范式设计可以将任务完美完成同时让用户觉得使用起来不会过于不便捷。本章最终完成了视觉刺激相关的界面设计、图像显示工作。

3.2 介绍针对脑-机接口机械臂控制时的指令集设计以及控制时刺激端流程图介绍。

3.3 为刺激频率及相位选择以及呈现到图像上的效果示例。这一步虽然少但是对于最终算法效果的影响很直接。

3.4 为生成的具体实现代码中的核心内容。

本章的内容处于整体系统中的一个独立去耦合的模块。

3.2 视觉交互范式设计

首先对要用到的控制指令进行设计。

现有常用的两种 BCI 控制方案分别对应图 3-1 中的两种方案。

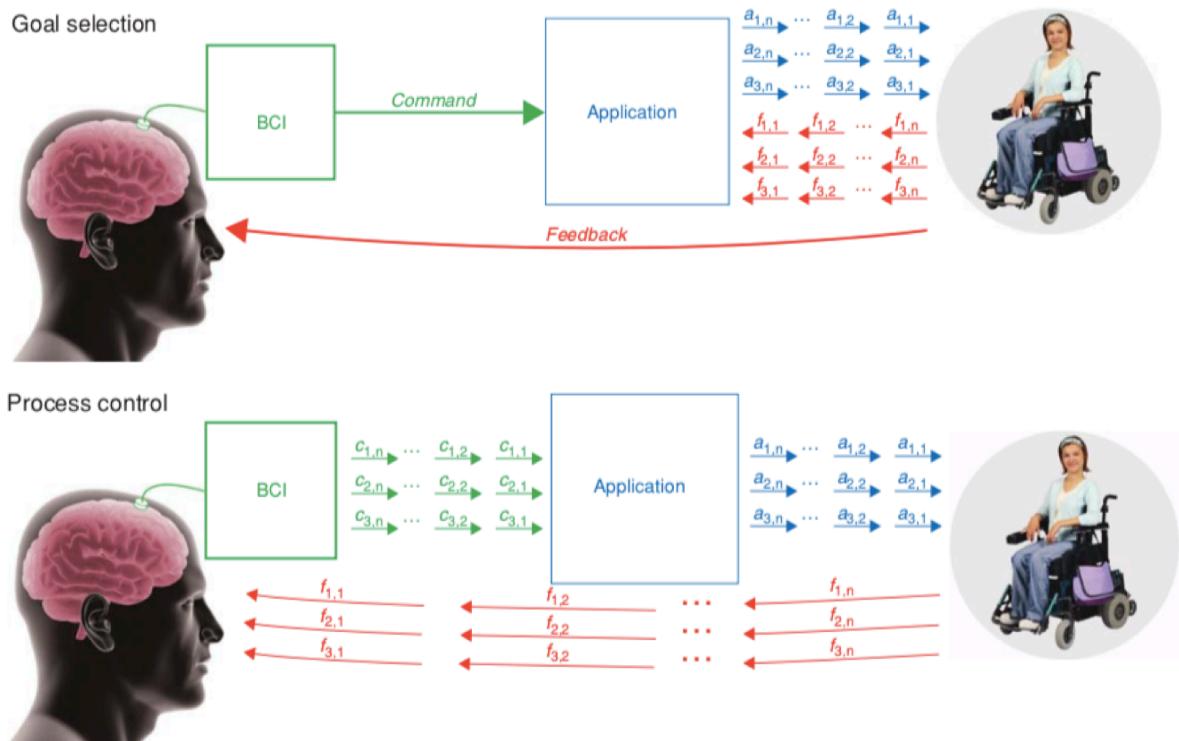


图 3-1 异步发指令控制执行的控制流程以及实时控制方式的控制流程^[10]

因为基于 SSVEP 脑-机接口机械臂的目标应用人群为残疾人士，本研究应该尽量选择风险较小的异步发指令形式的控制方案，原因是当没有一次性成功操作时仍保留缓慢调整的余地。而实时控制的系统会可能因为用户持续发出指令而过快过多地移动

造成意外。因此选择采用异步发指令的的控制方案。

同时，在发指令中可以选择的是结合计算机视觉一步到位，比如抓取一个杯子，或者步进式。所谓步进式是每次以固定参数运行，在机械臂中就是舵机产生力矩让机械臂产生相同的位移。一般都会将参数设定的比较小从而方便调整，用户发指令则产生移动一点点。虽然此方案效率较低，但为了给予用户更多的反应时间及容错空间我选择步进式的控制方案。具体对应指令集为

前伸，后伸，左转，右转，上抬，下移，复位，抓取，结束。共九个。

由于受到 2020 年疫情特殊影响无法进行后续硬件购买，本来计划在交互界面中加入的通过摄像头画面来帮助使用者更精准的判断无法实现，因此用户必须通过肉眼观察机械臂移动状况来判定是否已经达到自己想要的方位。

刺激界面对颜色本身是没有太苛刻的要求，但考虑到用户若需要长时间使用减少视觉疲劳，黑底白框黑字的组合是一个很好的选择。

刺激端的有限状态机本身分为几个阶段。首先运行程序后是初始化阶段，此时屏幕上会显示等待系统初始化，当后准备好后按空格就可以开始实验了。之后为预准备阶段，全部方块都以 255 亮度显示 1s 以方便用户观察各个指令的位置。而后为刺激阶段，开始显示刺激界面，系统成功识别一个指令后会确认选中的选项，以蓝框标示并重新回到预准备阶段开始刺激。一直到有 esc 键或者关注结束键系统会结束。刺激程序运行的有限状态机如图 3-2 所示。

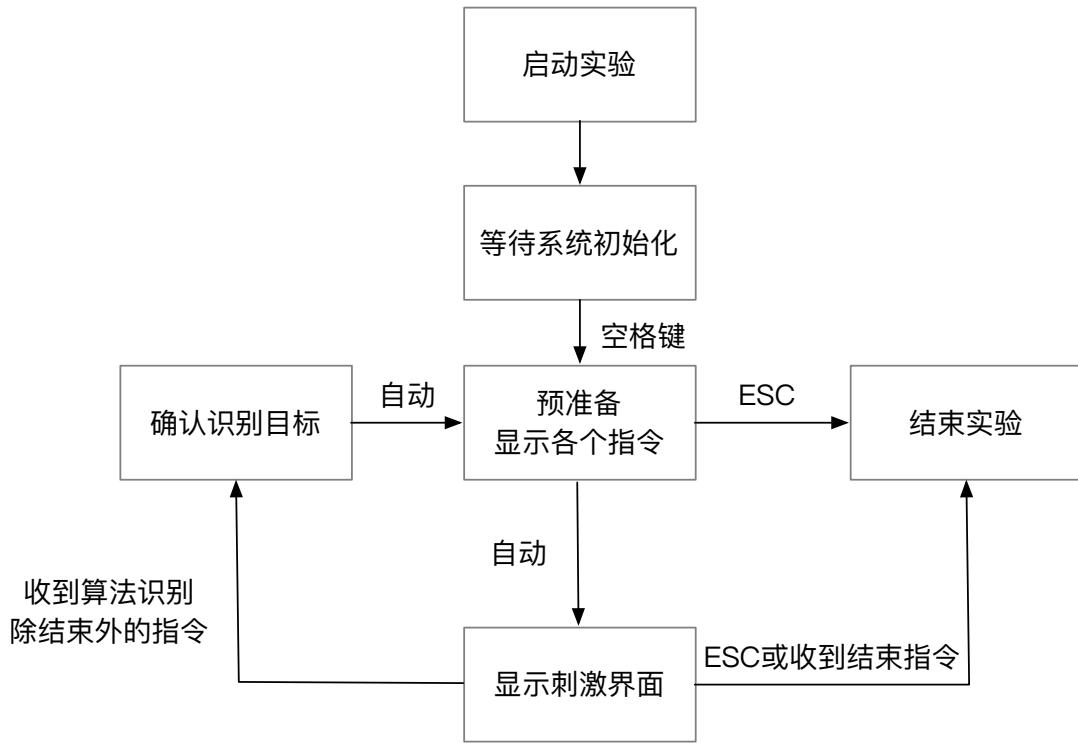


图 3-2 刺激程序有限状态机

最终刺激界面设计如图 3-3 所示，行为规范如下：用户开始后注视左半边控制区域即可控制机械臂移动，当移动至目标位置时注视抓取则机械手会闭拢。当机械臂过

于偏离时可以选择复位恢复到初始位置。关注结束键将终止通信，按 esc 键则结束本次实验，按空格键将暂停实验。

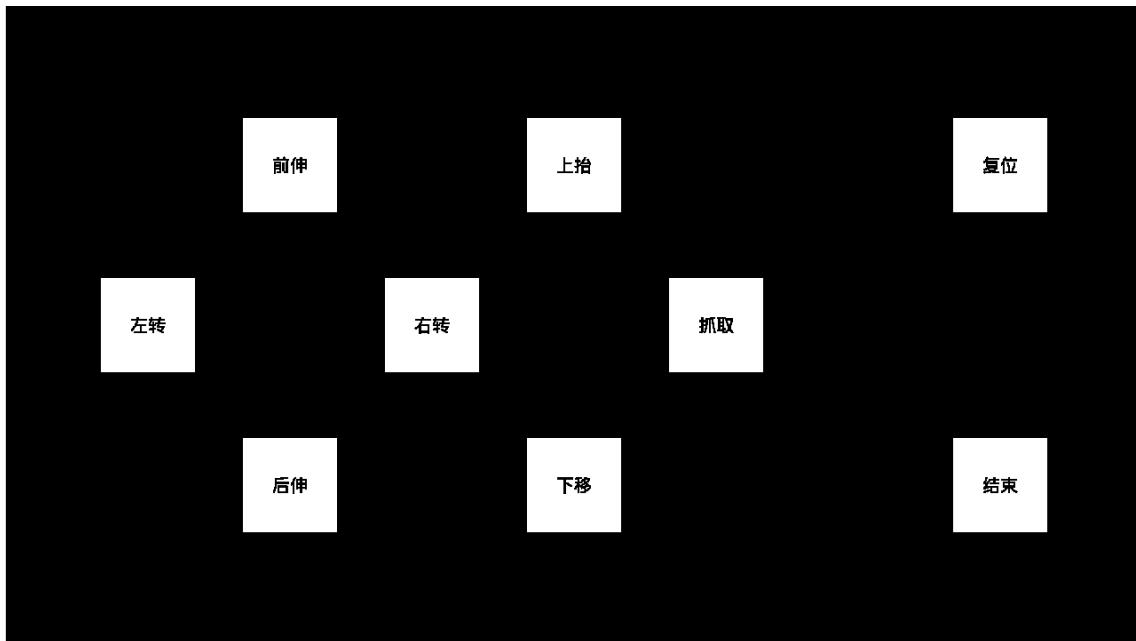


图 3-3 机械臂刺激视觉界面设计

当确认识别到用户注视指令后到确认界面如图 3-4 所示，会有蓝色框标示。关于界面如何从算法处利用通信获取到消息将在第五章中进行介绍。

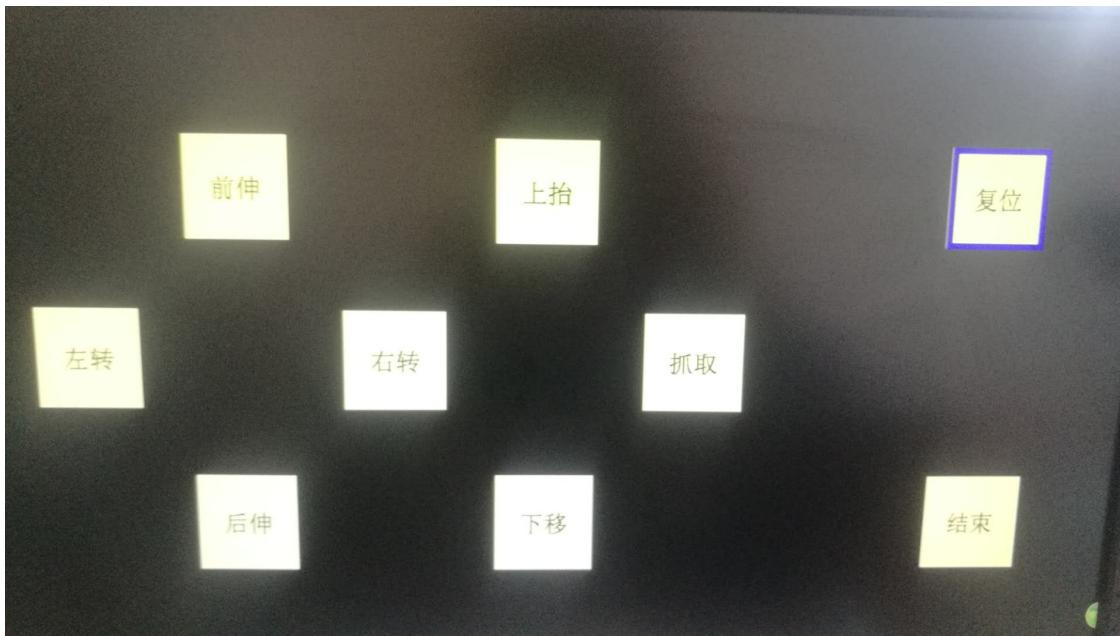


图 3-4 实际应用指令界面

3.3 刺激频率及相位选择

合理并有区分度的选择频率和相位是 SSVEP 范式能够成功应用的重要环节。

而刺激频率和相位本研究选择频率响应最明显的 15hz 以下部分^[19]，相位选择差距较大的每周期四个不同位置来初始化波形。所谓刺激相位不同实际上说的是初始时

刻的灰度不同。最终选择的频率 8-14.4 Hz，间隔 0.8Hz 选取一个频率；相位为 0π , 0.5π , 1π , 1.5π 四种。分配情况如表 3-1 所示。

表 3-1 指令的频率相位分配表

指令	前伸	后伸	左转	右转	上抬	下移	复位	抓取	结束
频率 (Hz)	8	8.8	9.6	10.4	11.2	12	12.8	13.6	14.4
相位 (rad)	0π	0.5π	1π	1.5π	0π	0.5π	1π	1.5π	0π

同时需要注意的细节为屏幕刷新频率一定不能过低，不然会引入频域混叠。在本研究这里必须选择屏幕刷新率为 60Hz 的 LCD 显示器，因为显示器的质量本质上决定了刺激的稳定性。当前频率为指令最高频率 14.4Hz 的 3 倍以上，足够精准显示需要的刺激频率。

因此生成的刺激界面中第一张图灰度变化如图 3-5 所示。生成时利用的函数为 \cos 函数，波峰为 0 相位为最亮，对应灰度 255； $0.5/1.5\pi$ 对应中间灰度； π 为波谷对应灰度 0，因此有两个指令为全黑。每个方块大小为 160*160 像素。

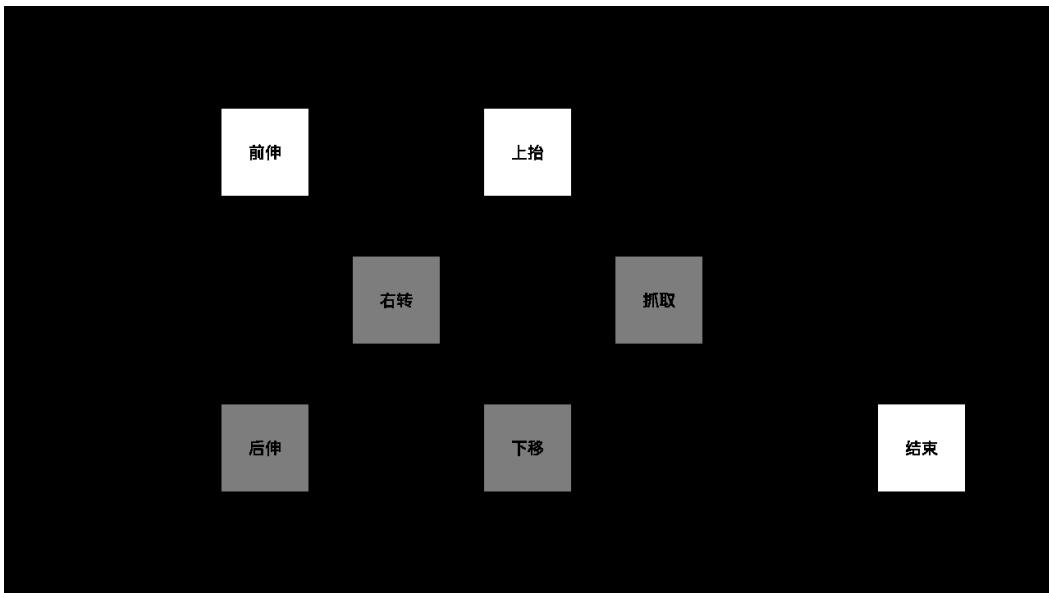


图 3-5 机械臂视觉刺激界面首帧画面

3.4 视觉刺激界面生成 matlab 实现方案

本次实验是基于一套完整的异步脑-机接口框架进行设计的，因此本部分只针对生成刺激界面相关的代码，具体提供给外界的接口及通信调用方案等将在第五章进行详细说明。

首先刺激界面功能为是生成并播放视频，在 matlab 中比较好的还可控制的实现方案就是做一个类似连环画的形式的图片集。在这里利用网上开源 Psychtoolbox 中的 Screen 函数来实现和显示器的交互。需要说明的是这里需要显示器为刷新率起码为

60Hz 的 LCD 屏幕。

表 3-2 getframe 伪代码

Algorithm 1 getframe ('CharSet', 'Freq', 'Phase', 'Position')	
1:	StimRectSize = [160,160];
2:	stimTargetRectSet=cell(length(CharSet));
3:	initRect = cell(length(CharSet));
4:	for $i = 1$ to length(targetCharSet) do
5:	t = Position[i];
6:	stimTargetRectSet(i) = StimTargetRect(Freq[i],Phase[i],t,CharSet[i]);
7:	initRect(i) = StimTargetRect(Freq[i],0,t,CharSet[i])
8:	background = imread(backgroundpic)
9:	end for
10:	frameSet = add(background,stimTargetRectSet)
11:	initframe = add(background,initRect)
12:	return frameSet, initframe

生成最终显示所需要的图像帧集合伪代码如表 3-2 所示。函数提供需要的刺激图像信息然后系统返回所需要的图像。这里面返回的结果是所有所需要点图像的集合，传入参数按顺序分别是矩形上的文字、频率、相位及矩形在屏幕显示的坐标方位。最终返回的参数 frameSet 是如图 3-3 所示的一帧帧图像的集合； initframe 是图 3-2 所示的图像，是用来在准备阶段向用户展示所有可以使用的指令。其中 background 是一个预存到本地的全黑图片，大小和显示器一样，Rect 结尾的两个变量都是存储白色矩形的图像矩阵且中间有黑色的指令文字。

StimTargetRect 核心运算是根据 freq 和 phase 还有屏幕刷新率来计算每一帧不同矩阵需要的不同亮度值。公式如式 3-1 所示为

$$\text{bright}[i] = 0.5 + 0.5 * \cos(2\pi * \text{freq}/60 * (i - 1) + \text{phase}) \quad \text{式 (3-1)}$$

bright 表示该白块在该频率和初始相位下第 i 张图片的亮度，60 是显示器刷新率。原因是本研究需要的是在 $1/\text{freq}$ 的时间里显示一个亮度周期，而一个 $1/\text{freq}$ 时间里屏幕会刷新的次数是 $60/\text{freq}$ ，因此本研究需要每一帧都刚好满足是 $\text{freq}/60 * \text{n}$ 第 n 帧的亮度。刷新到第 $60/\text{freq}$ 帧时便刚好在 $1/\text{freq}$ 时间里走了一个亮度周期。

3.5 本章小结

本章详细介绍了与视觉刺激界面相关的设计思路和实现细节。具体工作内容为了解并学习了如何利用 SSVEP 为大脑提供刺激，设计范式并实现了相关的界面及算法。本章内容如要复现可以在屏蔽系统其余部分的情况下独立在一台电脑上面实现并验证。

第四章 基于 ROS 系统的机械臂交互方案设计

4.1 本章简介

本章讲解针对脑-机接口控制机械臂的基于 ROS 系统设计的程序及控制方案。机械臂交互设计的方案要达到接收指令时成功执行，同时还需要仿真测试便捷。

4.2 介绍机械臂控制系统总体架构以及对应的通信节点的总体情况。

4.3 是利用 ROS 系统中专门针对机械臂控制设计的 Moveit 平台来完成本研究所需的功能。

4.4 是关于全部的仿真验证的细节以及最终仿真效果呈现。

同样 ROS 系统端可以作为一个去耦合的独立系统运行在整体框架内。

4.2 基于 ROS 系统的机械臂控制架构设计

机械臂相关软硬件配置参数为：电脑端采用 linux ubuntu 16.04 操作系统；ROS 系统采用 kinetic 1.12.14 版本；机械臂选用 X-ARM-5，配有 DYNAMIXEL 舵机；控制板为 artibox。

ROS 系统控制机械臂整体流程如图 4-1 所示。

流程为首先在脑-机接口识别成功后，另一端会针对这个指令进行封装，随后利用 Socket 将封装好的指令传递过来，再进行解析。解析后将对应 3.2 节定义的 9 种指令，除结束 ARED 指令外其余将进入缓存指令，之所以要缓存是因为 moveit 在执行一条指令时是不能读取另一个指令的。通过一个共用的参数 flag=0 本研究可以避免速率过高导致有指令丢失的情况方式。

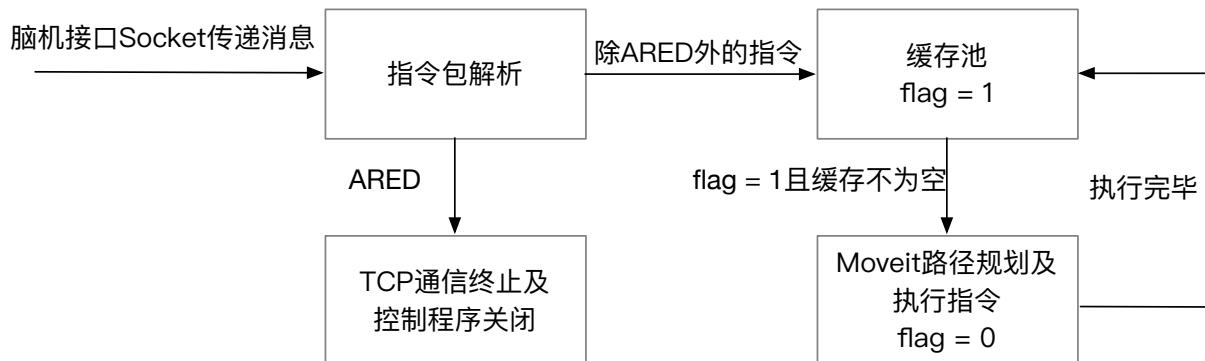


图 4-1 ROS 系统控制机械臂整体流程图

后文针对讲解的内容为 moveit 路径规划及执行指令部分。

另外本文中有部分工作不体现在流程中，该部分为针对自己使用的机械臂进行建模。虽然我的机械臂本身建模文件是卖家提供的但建模文件也是遇到 bug 时需要查看的重要环节因此建模方法也将进行讲解。

图 4-2 是 moveit 运行时各个节点的通信情况的整体图。线上的是各个节点间会一直通信互相提供的消息，圆圈内为核心节点。其中 Fake_joint_pub 是用于仿真时模拟真实舵机反馈角度的；move_group 为系统提供的核心控制节点，向控制板发送指令是

从该节点提供的;arbotix 为控制板，因为 arbotix 本身只是一个驱动库，不需要自己做过多修改，在使用时唯一需要修改的是各个舵机的编号，需要将编号顺序修订与建模时顺序一致即可；gripper_controller 是单独分配给抓取舵机的控制器，原因是机械臂移动时并不需要抓取的舵机同时移动。

真正使用时需要初始化一个节点，在我的程序里命名为 mainControl, 将会负责把指令传给 move_group。

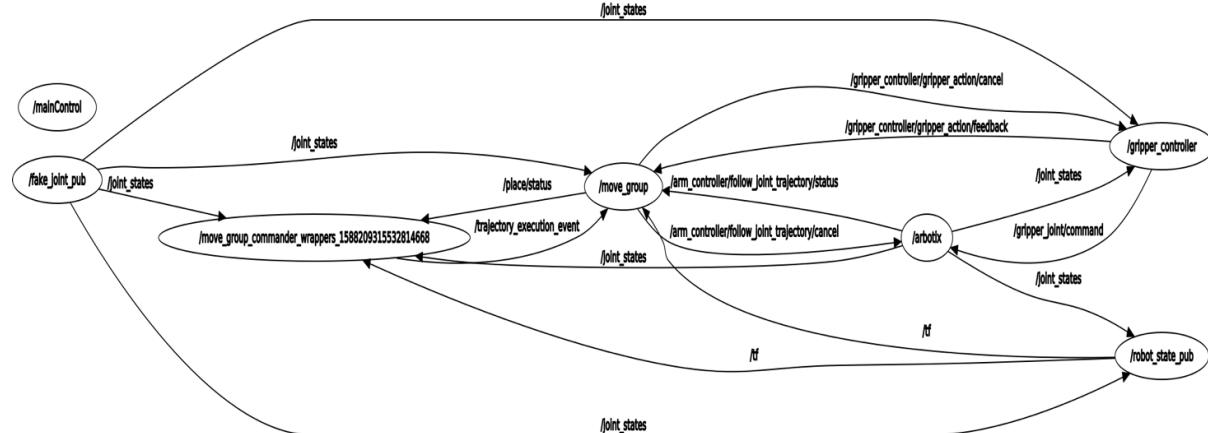


图 4-2 moveit 组件及 ROS 节点关系图

4.3 基于 Moveit 组件机械臂移动方案

在 ROS 中如果想要控制机械臂进行路径规划并执行规划并不需要自己从头写一个库。利用第二章中介绍的 Moveit 组件便可以完成相应工作。因此我在 moveit 的基础上来设计自己的移动方案

所谓机械臂的移动，本质上是从空间坐标 X1 到 X2 寻找一条可行的轨迹并利用舵机产生力矩从而带动机械臂往希望的方向运动。

路径规划的求解方式可分为正运动学(fk)规划和逆运动学规划(ik)两大类。原理如下。

正运动学规划(fk)是指给定各个电机转动的角度求出最后所在的坐标位置。需要提供的参数为各个电机旋转后最终与该电机连接的杆的角度，最终算出按照这样运动后所处的空间位置。这种控制方法的好处是非常直观感知到自己给定的参数会使系统如何运动。但坏处是如果想向某一方位进行运动则需要自己先进行运算。

每个机械臂都会有一个最终的器件称为终端，一般都是最上端的机械抓手。而逆运动学规划(ik)输入的参数直接给定终端的三维空间坐标以及角度朝向坐标，算法会进行逆运算来推演如何从当前位置移动到给定位置，输出为运动轨迹。

按照上述的两种算法，在对指令集合进行考虑后决定采用两种方案并存的方式来控制。说是移动方案其实本质上思考明白后如何调用接口更重要。

前伸、后伸、左转、右转、抓取、复位采用正运动学规划，本研究的机械臂控制调用 movegroup 终端 set_joint_value_target 接口即可完成所示功能。以前伸为例，具体函数实现伪代码如表 4-1，其余操作同理，只是设定的数字不同而已。首先获取当前舵

机的位置信息，返回一个长度为 5 的数组，从数组的第 1-4 位分别控制从低到高的四个舵机。然后按照需求调整各个舵机的角度即可，比如前伸会希望不光整体机械臂向前倾，同时机械爪也能一定程度的转为向前，因此除底座外的所有舵机都需要向前倾一定的角度。这也是伪代码第二行的作用所在。传入参数为 arm 是程序最开始实例化的 movegroup 类的名称。Set 函数传入刚刚修改过的参数，函数 go() 会自动调用 moveit 中的路径规划功能，若规划成功会在实体上执行。

表 4-1 正运动学规划前伸伪代码

Algorithm 1 fkForward ('arm')

```

1: cp = arm.get_current_joint_values()
2: cp[1] = cp[1]+0.2;cp[2] = cp[2]+0.2 ;cp[3] = cp[3]+0.2
3: arm.set_joint_value_target(cp)
4: plan = arm.go()
5: print "MoveForward"
6: return 1

```

因为向上和向下两种指令是相对于固定坐标系而言，因此采用逆运动学将是较简单的方式。表 4-2 是向下操作的伪代码。由于是相对于固定坐标系的算法，需要设定一个参考坐标系，参考系一般设定在机械臂的底座坐标为(0,0,0)的位置，这也就是 baselink 的作用。描述一个刚体的空间位置，除了坐标点还需要角度信息，但本研究这里因为只需要向上向下，而让机械爪移动的功能交给其余指令，因此本研究的角度一律沿用执行前的不变即可。现获取现在的坐标到 current 中，其中只有 z 坐标发生改变。同样再利用 set 和 go 便可以执行指令。

表 4-2 逆运动学规划向下伪代码

Algorithm 2 ikDown ('arm')

```

1: end = arm.get_end_effector_joint()
2: current= arm.get_current_pose(end)
3: target_pose.header.frame_id = "/baselink"
4: current.pose.position.z = current.pose.position.z-0.02
5: target_pose = current
6: arm.set_pose_target(target_pose,end)
7: arm.go()
8: return 1

```

上文为具体执行时的函数接口设计。在收到指令后进行调用便可完成相应功能。理解后实际代码量较少，这也体现了 ROS 平台封装好的库所带来的便利性。

4.4 基于 Moveit 组件机械臂建模及运动仿真

在 moveit 组件中实现针对机械臂运动的仿真需要两步：一是给机械臂建立模型并利用 rviz 进行可视化；二是利用 movegroup 函数接口进行控制。

机械臂模型连接关系为 pan 底座旋转舵机->shoulder 舵机->elbow 舵机->wrist 舵机->gripper 抓手舵机。编号从 1-5 依次排序。具体的建模采用 moveitSetupAssistant 完成，模型本身的样子是商家利用 3D 建模软件导出 stl 文件构成的。建模后利用 Rviz 显示。如果连接了实体机械臂则应该可以获得机械臂当前状态并显示在仿真软件中，如图 4-3 所示。



图 4-3 实体机械臂与仿真显示

所有舵机角度设置为 0 进行初始化应为图 4-4 中实体的直立形态。机械臂描述时所有的舵机角度都是相对于他更靠近顶端的那一个关节现有的角度而言的，是从垂直于底座的轴为一条边，关节所处的位置为另一条边计算角度。因此 0 角度时所有关节都应该垂直于底座成柱式耸立。

图 4-4 同时也是前伸运动的仿真。开启 rviz 后运行之前的指令函数时运行 go() 函数时会显示虚影，虚影部分对应目标运动到的位置，运行到该点路径为程序给出。

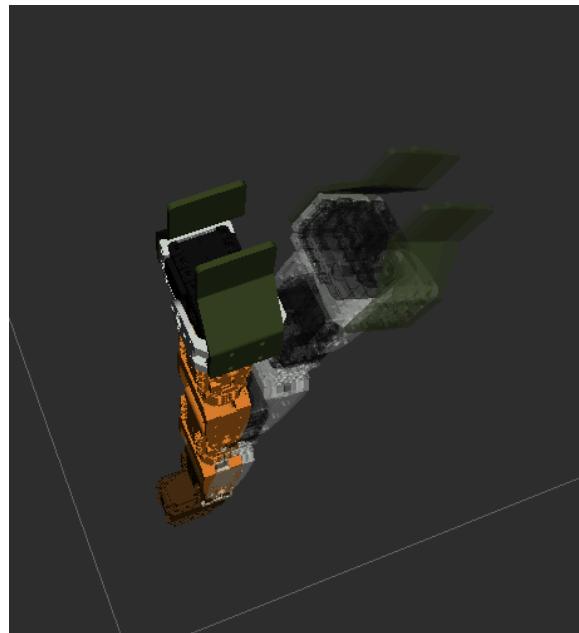


图 4-4 机械臂运动仿真

图 4-5 是仿真抓取动作执行后的舵机运行示例。本毕设中由于的机械爪不能旋转，只能由舵机控制一边的抓手进行移动因此我这个机械臂无法进行过于复杂的抓取动作，这决定了后面的实验只能选择较简单的任务来测试。若能更换更复杂的机械装置则本系统可以做到更多的事情。

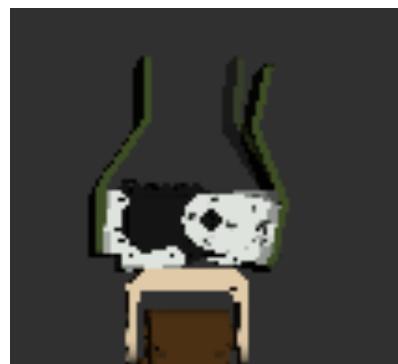


图 4-5 机械爪抓取仿真

4.5 本章小结

在第四章中我首先讲解了整体系统控制的流程及如此设计的思想及根据。同时在不过多涉及代码细节的基础上将控制中的核心思想进行了介绍。因为本研究采取的是步进式的算法因此整体很好理解，并且在理解的基础上要做的工作并不复杂。本质上正向规划是对舵机进行直接传递参数，逆向规划是给出要运行的目的地并根据这个来计算需要舵机如何运动。最后是关于机械臂建模及规划的实例展示。

本章在整体论文中的作用是解决当指令作为输入传入到 ROS 端后如何成功执行的问题。在仿真的基础上只需要将机械臂操控板通过 usb 接口连接电脑便可以成功通过端口将指令发出从而 artibox 会按照要求执行舵机运动。

第五章 在线脑-机接口控制机械臂系统设计及实验验证

5.1 本章引言

本章为整体架构的设计及各个系统间互相通信的定义及函数接口定义。还包括现场实验验证的细节。

5.2 节为整体系统架构以及控制方案的介绍。

5.3 节为各个通信接口的定义及消息解析方式，涉及到各个模块间信息交互的问题。

框架及通信定义再结合第3、4章的内容便是一套完整的可执行系统。

5.4 节为现场实验的相关设计，真实实验以及测试产生的评价数据。

5.2 SSVEP 在线架构及整体控制方案

SSVEP 在线异步控制机械臂需要满足两个特点，一个是需要保证实时性，不能接受到刺激后由很久才反应过来，还有一个就是要保证模块间的独立正常运作，不能互相干扰。

实时处理得到结果就需要将数据采集和输出分开，并行处理的架构方可满足这一点，因此使用杨晨博士在博士论文中提出的异步在线脑-机接口系统架构^[13]。如图 5-1 所示，其中 operator device 为处理数据操作算法的部分。Stimulus device 为刺激执行的部分。因为刺激电脑对于 CPU 独占率要求很高因此需要给每个模块分配一台电脑。Datareceiver 本身只是操作端的一个缓存设计，并不需要独立的电脑。

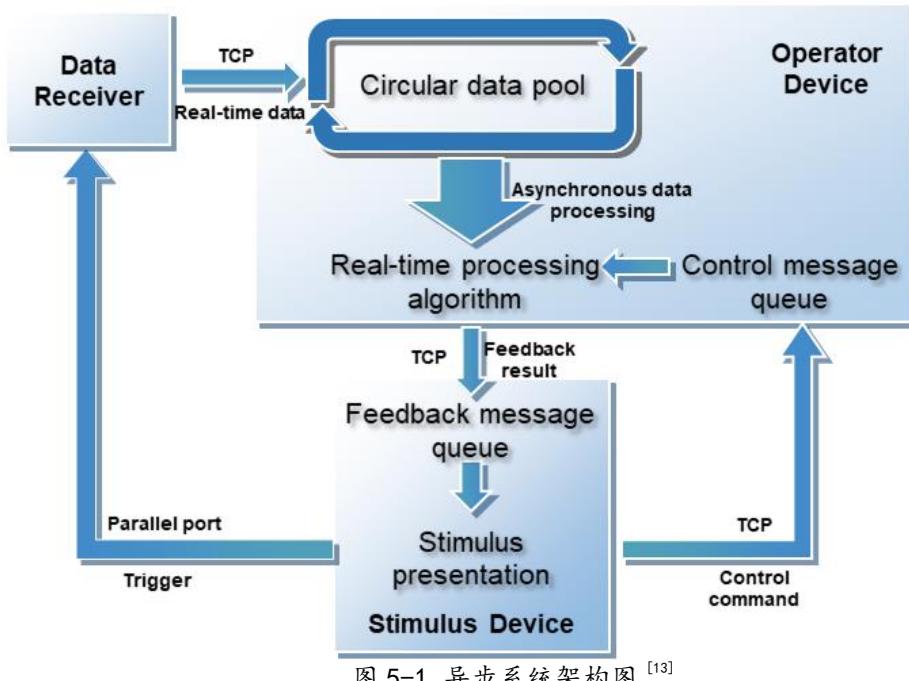


图 5-1 异步系统架构图^[13]

基于上述框架，我最终设计关于机械臂控制的全部架构如图 5-2 所示。分为三个主程序。一个在独立的电脑上运行刺激模块，一个在电脑上独立运行采集数据及处理模块，另一个负责机械臂的控制。而脑电采集设备端为生产厂家提供的信息服务，只

需要开启并连接到欲定义好的端口号即可采集数据。

三块程序间的交互由提前预定义好的包头进行通信，利用 TcpIp Socket 进行传输。需要信息的交互是因为在程序运行过程中，有需要等待别的程序发出指示后才能进入下一个步骤的停滞状态，各个模块间的需要利用通信来告知对方目前自己模块处于什么状态。例如当刺激端显示器尚未准备好进行视觉刺激时处理端并不需要读取当前的数据并进行处理，因此需要在准备阶段时传递一个消息来告知另一端可以在现在关闭缓存池。具体交互信息定义在 5.3 节中会进行讲解。

脑电采集设备主要输出经过自带放大器放大后的脑电数据。

数据采集及处理端负责接受数据同时在处理时向视觉刺激端反馈识别的结果，从而可以如图 3-2 中所述使得视觉显示端进入确认结果的程序，同时还要向机械臂控制端传输所识别的指令从而机械臂控制端执行相应的程序来操作机械臂完成动作。

视觉刺激端一是接收数据处理端提供的反馈信息。二是将自己现在在有限状态机的哪个阶段告知数据处理端。只有在显示刺激界面的状态下处理端才需要采集数据并进行算法识别。

机械臂控制端接收识别成功的指令。由于本次实验没有提供摄像头图像反馈，因此不需要控制端向处理端进行内容的发送，如之后有条件可以添加相应内容。机械臂控制端和机械臂实体通过 USB-FTDI 转换接口连接到 arbotix 板子上从而对机械臂舵机进行控制。

受试者直接利用眼睛注视刺激端显示器以及机械臂移动情况。

程序的运行流程图如图 5-3 所示，在执行中除开始目前需要利用空格键不需要额外进行任何的输入操作，真正能够服务于有障碍的目标人群利用本系统。本质上就是等待处理程序成功识别指令，如果成功就先判断一下是不是用户终止了实验，如果是则终止实验，如果不是则执行程序并将一直在中间几个步骤进行循环。

在最后实现程序中各个部分启动的顺序为先打开脑电数据采集放大器，一般公司都会提供 tcpip 通信接口，只需要作为 server 发送采集到的脑电数据。之后依次在电脑上运行机械臂控制程序，操作端程序，最后为刺激端程序。开启的顺序由程序内部 TCPIP 通信时作为 server 还是 client 决定。

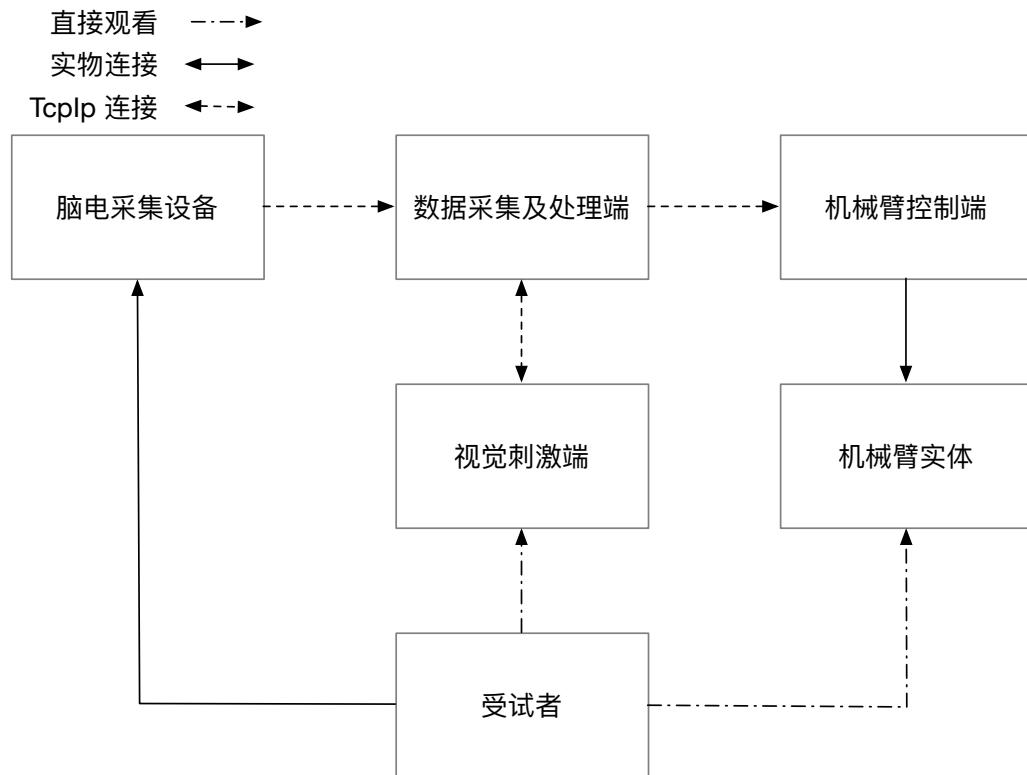


图 5-2 整体系统架构及连接关系

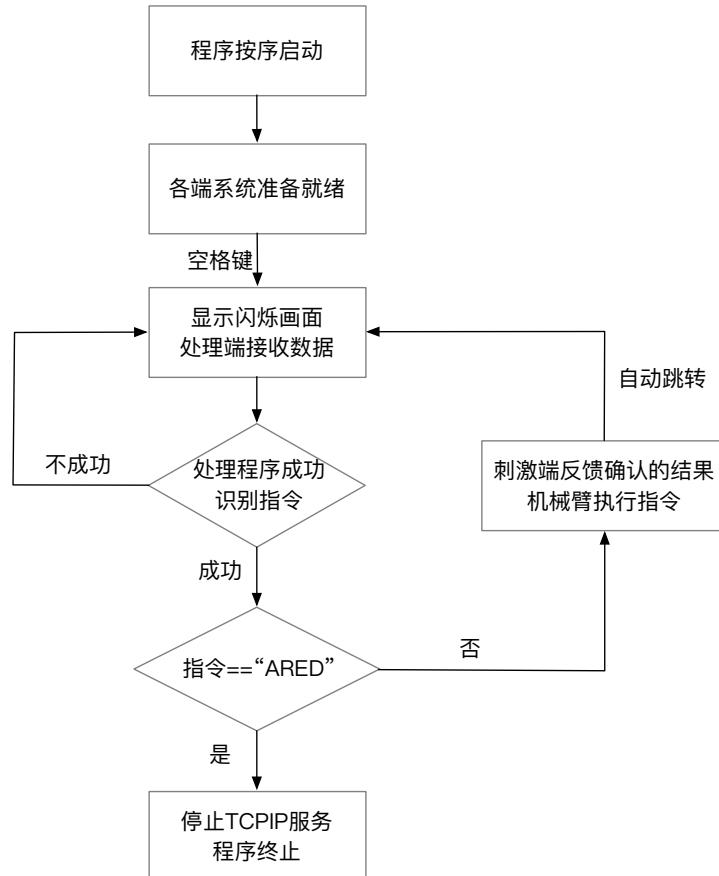


图 5-3 SSVEP 在线识别控制机械臂流程图

5.3 脑-机接口-ROS 系统通信接口设计及调用方法

整体通信基于 TCP/IP 通信协议，之所以选用 TCP 而不是 UDP 是因为在脑电数据传输环节本次毕设需要精准的接收所有的数据方便算法进行识别，同时在各个环节的通信中需要保证较高的实时性以及精准度，而对建立连接的时延并没有太高需求，因此必须使用对准确率有保障的 TCP 协议。后续介绍在本系统中通信消息应用层封装的方法。

`ExchangeMessage` 为本框架通信接口在应用层的统一封装定义。所有消息均采用继承 `ExchangeMessage` 的形式实现。封装包由 chId 包头, `bodysize` 内容大小, `body` 内容三部分构成。具体类型定义为 `chId` 为 1 字节的 `char` 或 `uint8` 类型数据。`Bodysize` 为 `uint32` 类型的数字表示后面 `body` 的大小, `body` 无具体内容限制。

发送时必须有的是 `chId` 和 `bodysize`, 若不需要 `body` 传送内容则 `bodysize` 默认为 0。包解析时先解析前八个比特, 常用的是发送四个 ascii 字母作为包头。然后再提取后面的 `uint32` 数据判断是否需要接收 `body`, 如果为 0 则没有 `body` 同时重要信息在包头内, 只要解析包头即可获得相应的指令。

具体到各个模块间的交互信息定义如附录表 5-1~5-4 所示。

根据其中的信息定义则可以完全看出各个框架间的通信情况。前面四个字母是包装的包头, 后面则是接受到该包头后应该执行的指令或者确认了某种消息的解读。同时这也体现了本研究整体系统的异步性, 因为不是所有节点都由一个同一时钟驱动本研究才需要利用众多的消息来调节各个系统间的协作。

调用方法:

发送函数: 发送消息端按照需求继承 `ExchangeMessage` 后进行自己的改写, 之后写好后在程序中实例化从, 包装成规定包格式即可调用其中实现的发送方法传递相应的指令。

连接函数: 指定客户端为 `client` 还是 `server`。利用 matlab 中 `tcpip` 函数来进行连接。需要提供参数 ip 地址及端口号以及缓存消息区域的大小。

接收函数: 提取对应的包头后进行解析, 并执行包头对应的指令功能。同时提出接收包的大小, 如果接收包超出缓存区预定义大小则多接受几轮数据直到达到要求大小位置。

5.4 数据采集实验验证及系统测试

5.4.1 实验设计

实验设备：戴尔 64 位 windows 电脑一台，macbook air 一台，博睿康公司 8 导导电帽及无线放大器。Macbook air 在 matlab 2017b 中运行 Operation 端代码，同时在 parallel 虚拟机中的 linux 平台装好的 Kinect 版本 ROS 上运行机械臂控制代码。

实验设计如下：被试进入房间后佩戴导电帽，参考电极为头顶位置，选用 8 导 Pz, P3, PO3, PO4, PO7, PO8, Oz, O1, O2。实验过程中电极保持阻抗 15k 以下。被试在椅子上待心情平静后开始实验。显示器在椅子前方，机械臂放置在显示器旁的桌面上。

具体实验环节为待实验系统准备好后由实验员按下 space 键。同时开始计时。在机械臂前方 15cm 处旁边放置一包纸巾。在右前方 45 度 15cm 处放置一个盒子。将纸巾拿起并放置到盒子之中为一个完整实验。当被试注视结束后系统结束计时。现场安排如图 5-4 所示。



图 5-4 实验现场准备图

根据人机交互领域常用的一些指标，本研究中选取了一些会帮助本研究测评机械臂控制系统的。测评指标采用每个指令间隔等待的时间，完成实验总所用时长及总指令数量三个客观指标；同时因为时我自己做实验因此没有进行主观指标的打分，但是仍然会有我对于实验整体难度的体会的梳理。

5.4.2 测试评价结果

最终于密闭空间无振动干扰但有市电干扰条件下进行真实实验。经过多次实验最终得到的数据如下。

系统总用时平均 393.0s，总操作次数为 19 次。机械臂等待指令给出总时长为 363.7s。平均每次操作需要等待 19.1s 系统才能识别成功一次指令。

图 5-5 为一次成功实验中单步用时统计，(a)中灰色部分为等待一次指令的时间，此时所有系统都等待操作端识别。(b)中蓝色部分是机械臂执行一次动作的时间，橙色部分是每次从 socket 成功接收到指令结束的时间。可以看到大部分的时间都是花在了(a)中的等待接受的时间，并且随着对系统使用的熟悉程度的提高发送指令的用时有着明显下降的趋势。这说明可能经过训练后的人能更快的专注在自己希望的视觉刺激上。同时(b)可看出通信接收判别过程对于时延几乎没有影响。

但图中反映出本研究的算法处理时间仍然过长，占据了大部分系统使用的时长。若在这上面能够进一步提高则系统的使用便捷度会进一步提升。

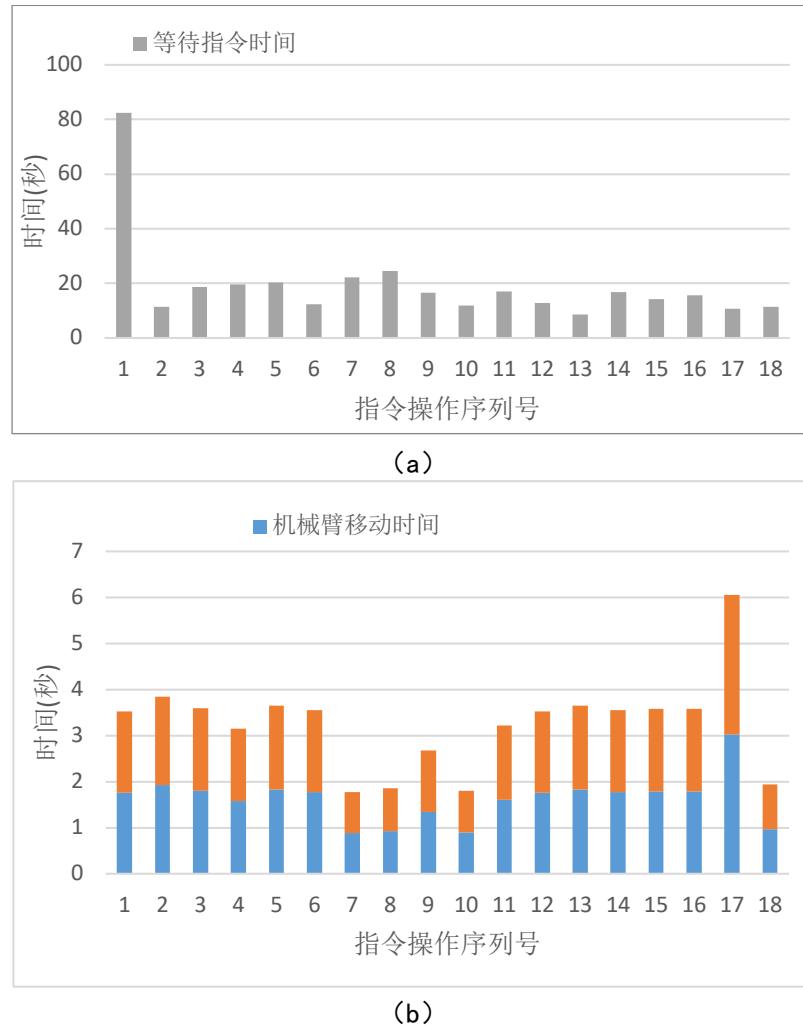


图 5-5 单步用时统计 (a) 等待指令时间 (b) 机械臂移动时间与接收指令到完成时间

同时我在做的时候感觉我们的任务完成很轻松，一可能是因为我们所选的实验本身难度就较低，只需要完成相对简单的工作即可，二可能是因为步进式系统发挥了设计之初的初衷，步进式完成任务的确较慢，但能使得人在使用时有更大的调整空间。但是使用期间由于过长时间地等待算法识别，很容易就造成了视觉疲劳，我觉得这一点也在以后需要解决。

图 5-6 是现场实验视频中，机械臂成功将纸巾放入盒子时刻的 4 张截图,(a)为准备阶段，(b)为机械臂正在向目标移动，(c)为机械臂抓住纸巾的场景，(d)为成功放入。

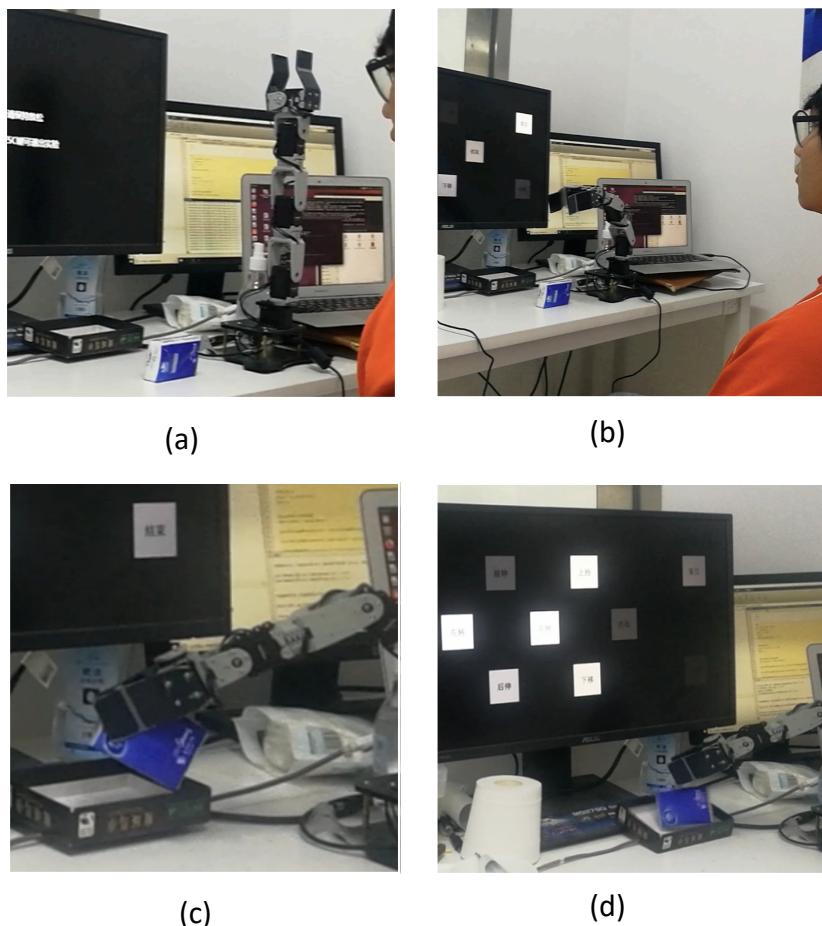


图 5-6 现场实验图片组 (a) 系统准备阶段 (b) 脑控机械臂向目标移动
(c) 脑控机械臂成功抓取 (d) 脑控机械臂成功将纸巾放入纸盒

综上实验成功完成并达到设计预期要求。能够提供完整的抓取功能，不过整体效果仍然有提升空间。

第六章 总结

在本次毕设中我主要完成了一个完整的基于稳态视觉诱发电位脑-机接口控制机械臂的系统，经过验证最初希望的可以简单完成所设想的不需要任何物理交互便完成控制的设想，完成简单抓取任务需要约 400s 时间。

总体上这几个月完成了三个主要任务。

第一个是为控制机械臂设计了 SSVEP 刺激端的范式，完成了机械臂控制形式的选取以及刺激界面的设计及实现。

第二个是实现了基于 ROS 系统控制机械臂完成我们所需的路径规划及执行能力。在其中各个环节得以学习和掌握。这部分是从零基础开始完成的因此所花费的时间和精力也很多，

第三个便是整个架构的完成，虽然架构本身是采用现有框架，但其中的通信及整体控制流程的设计及完成仍然是令我感到兴奋的地方，尤其在最终实验中看到自己亲手设计的东西成功运行，十分开心。

可以改善的地方也可以三个方面讲。

第一是先有的软件是基于 matlab 的，这本身可能就导致计算上没有纯 c++ 代码的软件更快更便捷，软件及电脑本身的处理速度导致我们的任务完成时长比较长、识别等待时间较多、视觉易疲劳等待一系列问题。

第二就是算法本身还可以更快更好，这一点肯定是需要继续探索。同时机械臂控制方面的知识也需要继续学习，如果可以的话未来机械臂应该是可以向几个前沿科学家正在探究的那样，想哪打哪。

第三就是如果可以利用更高频率的刺激造成视觉上的连贯同时还能区分出来效果的话那肯定是最好的，之后的研究也许可以在高频上面下一些功夫，因为这样的话人们的视觉疲劳感会极大的减少。

在这短暂的几个月里学到了许多我渴望了解的脑-机接口的知识，同时也真正地实地用机械臂进行了操作与控制。我觉得让我学到最重要的一点就是软件架构设计以及函数等抽象接口的实现方式和数学算法等等较为高深的理论知识同等的重要，想要造出理想的系统就必须关注到从编译到理论的所有细节，一环出错都可能导致系统无法按照理想的情况运行。

至此，圆满结束。

参考文献

- [1] C. Pandarinath, P. Nuyujukian, C. H. Blabe et al. High performance communication by people with paralysis using an intracortical brain-computer interface [J]. *eLife* 6.2017: e18554.
- [2] M. J. Vansteensel, E. G. M. Pels, M. G. Bleichner et al. Fully implanted brain-computer interface in a locked-in patient with ALS [J]. *N. Engl. J. Med.* 375. 2016: 2060–2066.
- [3] C. E. Bouton, A. Shaikhouni, N. V. Annetta et al. Restoring cortical control of functional movement in a human with quadriplegia [J]. *Nature* 533.2016: 247–250.
- [4] S. R. Soekadar, M. Witkowski, C. Gómez, et al. Hybrid EEG/EOG-based brain-neural hand exoskeleton restores fully independent living activities after quadriplegia [J]. *Sci. Robot.* 1. 2016: eeag3296.
- [5] U. Chaudhary, B. Xia, S. Silvoni et al. Brain-computer interface-based communication in the completely locked-in State [J]. *PLOS Biol.* 15. 2017: e1002593.
- [6] VIDAL J J. Toward Direct Brain-Computer Communication [J]. *Annual Review of Biophysics and Bioengineering*. 1973, 2: 157-80.
- [7] Lebedev M A, Nicolelis M A L. Brain-machine interfaces: past, present and future [J]. *Trends Neurosci.* 2006, 29: 536–546.
- [8] ROS 入门 [EB/OL].(2014.01)[2020.05.01].<https://clearpathrobotics.com/blog/2014/01/how-to-guide-ros-101/>
- [9] Jonathan R. Wolpaw, Elizabeth Winter Wolpaw. *Brain-Computer Interfaces_ Principles and Practice* [M]. Oxford University Press. 2012: 168-243.
- [10] Farwell LA, Donchin E. Talking of the top of your head: toward a mental prosthesis utilizing event-related brain potentials [J]. *Electroencephalogr Clin Neurophysiol.* 1988, 1: 510–23.
- [11] 林科. 基于稳态视觉诱发电位的脑-机接口算法与机器人系统 [D].[学位论文].北京.清华大学: 2017
- [12] Edelman, Bradley & Jianjun, Meng & Suma et al. Noninvasive neuroimaging enhances continuous neural tracking for robotic device control [J]. *Science Robotics.* 4. 2019: 10.1126/scirobotics.aaw6844.
- [13] 杨晨. 面向应用的稳态视觉诱发电位脑-机接口算法及系统研究 [D].[学位论文].北京. 清华大学: 2018
- [14] Wagner, F.B., Mignardot, J., Le Goff-Mignardot, et al. Targeted neurotechnology restores walking in humans with spinal cord injury [J]. *Nature* 563. 2018: 65–71.
- [15] Makin, J.G., Moses, D.A. & Chang, E.F. Machine translation of cortical activity to text with an encoder-decoder framework [J]. *Nat Neurosci* 23. 2020: 575–582.
- [16] YANG C, HAN X, WANG Y et al. A Dynamic Window Recognition Algorithm for SSVEP- Based Brain-Computer Interfaces Using a Spatio-Temporal Equalizer [J]. *Int J Neural Syst.* 2018, 10: 1850028.
- [17] M. Quigley, K. Conley, B. P. Gerkey et al. Ros: an open-source robot operating system [J]. *ICRA Workshop on Open Source Software.* 2009.
- [18] FOOTE, Tully. tf: The transform library [J]. *IEEE Conference on Technologies for Practical Robot Applications (TePRA)*. IEEE. 2013: 1-6.
- [19] STEDER, Bastian et al. Point feature extraction on 3D range scans taking into account object boundaries [J]. *IEEE International Conference on Robotics and Automation*. IEEE. 2011: 2601-2608.
- [20] HORNUNG, Armin et al. Navigation in three-dimensional cluttered environments for mobile manipulation [J]. *IEEE International Conference on Robotics and Automation*. IEEE. 2012: 423-429.
- [21] COHEN, Benjamin J., CHITTA et al. Search-based planning for manipulation with motion primitives [J]. *IEEE International Conference on Robotics and Automation*. IEEE. 2010:2902-2908.
- [22] STEDER, Bastian et al. Point feature extraction on 3D range scans taking into account object boundaries [J]. *IEEE International Conference on Robotics and Automation*. IEEE. 2011: 2601-2608.
- [23] LEE, Hyung-Min, PARK et al. A power-efficient wireless system with adaptive supply control for deep brain stimulation [J]. *IEEE journal of solid-state circuits*. 2013: 2203-2216.

- [24] KWON, Mingu, KANG et al. Emotion classification in movie clips based on 3D fuzzy GIST and EEG signal analysis [J]. International Winter Workshop on Brain-Computer Interface (BCI). IEEE. 2013. p. 67-68.
- [25] LEEB, Robert, et al. Thinking penguin: multimodal brain-computer interface control of a vr game. IEEE Transactions on Computational Intelligence and AI in Games [J], 2013, 5: 117-128.
- [26] HERRMANN C S. Human EEG responses to 1-100 Hz flicker: resonance phenomena in visual cortex and their potential correlation to cognitive phenomena [J]. Experimental Brain Research. 2001: 346-53.
- [27] RATHEE, Dheeraj, et al. Brain-Machine Interface-Driven Post-Stroke Upper-Limb Functional Recovery Correlates With Beta-Band Mediated Cortical Networks [J]. IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2019, 27.5: 1020-1031.
- [28] NAKANISHI M, WANG Y J, CHEN X G et al. Enhancing Detection of SSVEPs for a High-Speed Brain Speller Using Task-Related Component Analysis [J]. IEEE Transactions on Biomedical Engineering. 2018: 104-12.
- [29] BENABID, Alim Louis et al. An exoskeleton controlled by an epidural wireless brain-machine interface in a tetraplegic patient: a proof-of-concept demonstration [J]. The Lancet Neurology. 2019, 12: 1112-1122.
- [30] LIN Z, ZHANG C, WU W et al. Frequency recognition based on canonical correlation analysis for SSVEP-based BCIs [J]. IEEE Transactions on Biomedical Engineering. 2007: 1172-6.
- [31] Moveit 教程[EB/OL].[2020.05.01].<https://moveit.ros.org/documentation/>

致 谢

感谢杨晨老师无微不至地回答我的每一个问题，并且每周都督促我们按计划推进自己的毕业设计。同时还要感谢老师能够在疫情过后立刻联系校外实验室让我成功体验了一次自己科研成果实现的快乐。毕设这段时间内的指导我会受用终身。

感谢大学四年帮助过自己的闵祥伟、苏柏泉、张彬、林雪红、康学净老师以及清华 GIX 实验室的王运涛老师。他们帮助我找到了自己感兴趣的方向。

感谢徐一涛同学在大学期间和我一起进行了无数的实验，共同进步。

感谢我的朋友冷昊阳、冯葆瑞、赵洪洋、吕铮在宿舍提供的无穷无尽的快乐和有启发的对话。感谢其它的朋友在四年中带来的欢声笑语。

感谢老爸老妈能够给我提供一个如此舒适的生活环境从而有机会在四年大学生活中自由自在地探索。

最后感谢我自己没有辜负这人生的灿烂年华，真正体会到了知识增长带来的快乐和烦恼。后会有期。

附录

表 5-1 处理到数据采集端信息

处理端向数据采集器的交互信息	具体含义
CTNS	连接采集器;
DCNS	取消连接;
STAR	开始接收数据;
STOP	停止接收数据;
CLRT	清除之前接收的内容;

表 5-2 刺激到处理端信息

刺激端向处理端发送的交互信息	具体含义
STON	开始操作端处理;
SPON	停止操作端处理;
CSAL	关闭操作端;
EXIT	退出;

表 5-3 处理到刺激端信息

处理端向刺激端间发送的交互信息	具体含义
CTOK	成功连接采集器;
DCOK	成功断开采集器连接;
TROK	操作端成功开始接收数据;
PROK	操作端成功停止接收数据;
TNOK	处理端成功开始运行;
PNOK	成功停止处理端运行
CAOK	处理端成功关闭所有进程;
STSН	停止刺激端程序;
RSLT	传输成功识别指令，内含有包内容，内容为识别成功的指令消息;

表 5-4 处理到 ROS 端信息

处理程序与 ROS 交互信息	具体含义
AMFD	前伸;
AMBD	后伸;
AMLT	左转;
AMRT	右转;
AMUP	向上;
AMDН	向下;
AMNL	松开机械爪并恢复初始状态;
AMGB	机械爪执行抓取动作;

ARED	停止程序；
------	-------