浙江工业大学

本科毕业设计（论文、创作）

题目：基于马尔可夫决策过程的人机共享自治方法研究

作者姓名 赵丹波

指导教师 赵云波教授

专业班级 自动化 1602

学　　院 信息工程学院

提交日期 2020 年 6 月 13 日

Dissertation Submitted to Zhejiang University of Technology
for the Degree of Bachelor

Research on Autonomy of Human-Machine Sharing Based on Markov Decision Process

Student: Zhao Danbo

Advisor: Professor Zhao Yunbo

College of Information Engineering

Zhejiang University of Technology

June 2020

浙江工业大学

本科生毕业设计(论文、创作)诚信承诺书

本人慎重承诺和声明：

1. 本人在毕业设计（论文、创作）撰写过程中，严格遵守学校有关规定，恪守学术规范，所呈交的毕业设计（论文、创作）是在指导教师指导下独立完成的；

2. 毕业设计（论文、创作）中无抄袭、剽窃或不正当引用他人学术观点、思想和学术成果，无虚构、篡改试验结果、统计资料、伪造数据和运算程序等情况；

3. 若有违反学术纪律的行为，本人愿意承担一切责任，并接受学校按有关规定给予的处理。

学生（签名）：

年　　月　　日

浙 江 工 业 大 学

本科生毕业设计（论文、创作）任务书

专 业 自动化 班 级 自动化 1602 学生姓名/学号 赵丹波/201603080530

一、 设计（论文、创作）题目：

基于马尔可夫决策过程的人机共享自治方法研究

二、主要任务与目标：

1. 阅读相关文献，了解本领域研究现状；

2. 详细了解 MDP 相关的算法及共享控制的相关研究内容；

3. 选择或改造合适的 MDP 方法用于实现共享自治，并在合适的场景下进行验证；

4. 撰写毕业论文。

三、主要内容与基本要求：

本课题旨在研究基于马尔可夫决策过程的人机共享自治的方法来更好地实现特定目标。资料：将提供若干相关的参考文献及相关书籍，同时培养学生查询相关文献的能力。本课题将主要基于文献阅读整理、算法研究和仿真或真实场景下对方方法的验证等。学生须具备的技能：要求具有一定的算法基础和计算机编程能力；具有一定的英文阅读水平，能够较为独立地研读英文科技文献；具有一定的总结提炼能力。

四、计划进度：

- 2020 开学前 收集相关资料文献，学习相关知识，完成外文翻译、文献综述；熟悉课题，做好开题准备 - 第 1-3 周 完成开题报告，参加开题交流 - 第 4-8 周 完成基于 MDP 的共享自治方法研究，接受中期检查 - 第 9-14 周 在合适的场景下对方法进行验证，撰写毕业论文 - 第 15 周 修改毕业论文，参加毕业答辩，提交相关文档资料

五、主要参考文献：

[1] Fu J, Topcu U. Synthesis of shared autonomy policies with temporal logic specifications[J]. IEEE Transactions on Automation Science and Engineering, 2015, 13(1): 7-17. [2] Zhou S, Mu T, Goel K, et al. Shared autonomy for an interactive AI system[C]//The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings. ACM, 2018: 20-22. [3] Reddy S, Dragan A D, Levine S. Shared autonomy via deep reinforcement learning[J]. arXiv preprint arXiv:1802.01744, 2018.

任务书下发日期 2019 年 12 月 31 日

设计（论文、创作）工作自 2020 年 01 月 01 日 至 2020 年 6 月 10 日

设计（论文、创作）指导教师 赵云波

系主任（专业负责人）

主管院长

基于马尔可夫决策过程的人机共享自治方法研究

摘　要

在许多领域中，交互式系统要么为我们自主地做出决策，要么为我们提供决策权并发挥支持作用。但是，许多设置例如在教育或工作场所中的设置,受益于在用户和系统之间共享这种自主权，并因此受益于随着时间的推移而适应他们的系统。针对钢琴教学，传统的视频教学虽然节省了人力更为方便，但是重复的视频容易使用户失去专注，无法保证学习的效率；将控制权交予用户又会使得操作过于繁琐。所以需要设计交互式共享自主的钢琴教学系统。

在本文中,主要研究的是基于马尔可夫决策过程的人机共享自治的方法来更好地实现特定目标，我们设计了一种用于钢琴教学的交互式共享自主系统，可播放乐曲的不同片段，供学生模仿和练习。运用部分可观察的马尔可夫决策过程，以人的性能和注意力作为可观察的状态变量，使用二进制值来测量状态变量，在动态贝叶斯网络中对这些变量之间的过渡进行建模形成信念，推测用户的期望，从而来进行自治权的授予。论文的主要工作如下：

1. 叙述论文的研究背景以及研究意义，介绍人机交互系统以及人机关系中人的作用。
2. 运用 POMDP 决策方法可以在不确定性环境和条件下做决策的特性，基于 POMDP 对人机交互式系统进行建模研究，构造出一种从人机系统的人的可观测因素中去推断系统中人的未知的内部状态形成信念的方法，并运用这种方法对人机关系的自治权归属进行判别。
3. 改造合适的 MDP 模型建立于钢琴教学的交互式共享自主系统，通过对系统自治权是控制实现系统功能的最优化。并通过仿真实验对钢琴教学系统的有效性进行验证。

关键词：交互式系统，部分可观马尔可夫决策过程，人机关系

Research on Autonomy of Human-Machine Sharing Based on Markov Decision Process

ABSTRACT

In many fields, interactive systems either make decisions for us autonomously, or provide us with decision-making power and play a supporting role. However, many settings, such as those in education or the workplace, benefit from sharing this autonomy between users and systems, and therefore benefit from systems that adapt to them over time. For piano teaching, although traditional video teaching saves manpower and is more convenient, repeated videos can easily cause users to lose focus and can not guarantee the efficiency of learning; giving control to users will make the operation too cumbersome. Therefore, it is necessary to design an interactive sharing and independent piano teaching system.

In this article, the main research is based on the Markov decision process of human-computer sharing autonomy to better achieve specific goals. We designed an interactive shared autonomous system for piano teaching that can play different music. Snippets for students to imitate and practice. Using the partially observable Markov decision process, using human performance and attention as observable state variables, using binary values to measure state variables, and modeling the transition between these variables in a dynamic Bayesian network Form beliefs, speculate on user expectations, and grant autonomy. The main work of the paper is as follows:

1. Describe the research background and significance of the paper, and introduce the role of humans in human-computer interaction systems and human-computer relationships.
2. Using the POMDP decision-making method to make decisions under uncertain environments and conditions, based on POMDP modeling research on human-computer interactive systems, constructing a kind of inference from the human observable factors of the human-machine system The method

of forming beliefs in the unknown internal state of people in the system, and using this method to judge the ownership of the autonomy of human-machine relations.

3. Transform the appropriate MDP model to build an interactive shared autonomous system for piano teaching, and realize the optimization of system functions through control of system autonomy. And through the simulation experiment to verify the effectiveness of the piano teaching system.

Keywords: Interactive system, partly observable Markov decision process, human-machine relationship.

目　　录

第1章 绪 论

1.1 研究背景及意义

1.1.1 人机系统的定义

从定义上来讲，机器被定义为执行机械运动，并利用传递或者转变各类信息如能量，进而实现特定目标的一类设备的总称[1]。例如，可以集成各种计算设备以执行控制算法操作的控制器、测量控制系统中受控对象的状态或输出信息的传感器以及根据既定的控制策略等执行特定任务的执行器等。

人机系统本质上是由人和机器组成的系统，可以根据人和机器之间的相互作用来完成某些功能[1]，它是一个由两个子系统组成的整个系统，即人与机器相互作用并依赖于机器实现其目的系统。不仅是工程心理学研究的主要内容，而且是现代生产管理和工程技术设计的主要内容[2]。如何设计合理的人机系统，使该系统能否发挥可靠和有效的作用一直是一个基本问题。

在人机系统的人机关系中，人的作用是不可替代的。从本质上说，控制系统的功能目标是由人来决定的，是人这一主体的需求为机器设定了目标、赋予机器价值[3]。毫无疑问，人是系统目标的来源，因为无论是多么复杂的控制系统它设计的初衷都是为了满足人的需求，是为了实现某一特定的目标才出现的。从功能的角度来说，人的一些特殊因素也会影响机器功能的实现，这种影响或好或坏。所以，机器想实现更好的功能绝对不能忽视人的影响。

如今，人工智能技术的崛起，使得可以实现人机交互自治的系统逐步普及到日常生活中，功能上也从单方面由机器辅助人完成任务的模型，逐步转型为人与机器之间的协同控制以及人机交互自治实现目标的模型。通过人与机器合作更好地完成任务在这个时代已经成为一种常见的提升任务完成的质量即人机系统的性能的方法。当然，在本课题的研究中，我们简化对人的行为的检测部分的研究，将重点放在通过对人的内部期望和系统自主权的选择的研究来实现系统的最优效率。

1.1.2 交互式系统的定义

工具在人类生活中的作用的发展往往带动着人机系统的发展。人机交互系统是由人和机器组成的可以进行协同操作来完成任务的系统。它是机器设备为了完成设定的自主任务，例如测量或者判定时发生的可观察过程与测量过程之间的关系[4]。传统意义下的控制系统主要是满足人类的需求，在这类系统中，人是系统的主体，而机器则起辅助作用。但是，随着人工智能技术的不断发展成熟，在人机系统中，机器已经不单单只起到辅助作用。同理，人也不仅仅只处于主体地位。从作为辅助工具的机器到人机交互的人机交互。在人机关系领域中，人机交互正变得越来越普遍。

在许多领域中，交互式系统要么为我们自主地做出决策，要么为我们提供决策权并发挥支持作用。许多设置尤其是教育工作方面的设置，受益于在用户和系统之间共享这种自主权，并因此受益随着时间的推移而适应他们的系统[5]。针对钢琴教学，传统的视频教学虽然节省了人力更为方便，但是重复的视频容易使用户失去专注，无法保证学习的效率；将控制权交予用户又会使得操作过于繁琐。所以需设计交互式共享自主的钢琴教学系统[6]。

1.1.3 人机系统中的认知偏差的定义

认知偏差是一种在特定情况下特定于环境的思维和行为倾向，即偏离判断标准或理性的系统模式[2]，最终将导致理性的系统偏见，即个人成员使用投入感来形成自己的“主观社会现实”。本文的目的是分析人类的认知特征，并从原因，心理机制，研究领域，认知偏差的纠正和应用等方面来识别认知偏差。

认知偏见已在各个领域得到了广泛研究，包括其成因，影响和应对策略。认知偏见在人机交互领域并不普遍，关于人类认知偏见的认识还很少[7]。认知偏差被认为很难检测和掩盖。我们已经知道了常见的认知偏差及其成因，旨在使机器能够在人机交互过程中识别人的认知偏差并实现必要或更好的人机切换策略。

结合人机系统的特殊性，我们可以给出人机系统中认知偏差的定义：它是指会影响人机系统性能的异常内部状态[8]。鉴于人机系统中人机认知偏差的特殊性，由人机系统中人为偏差引起的决策错误将反映在人机系统的输出数据中，实际上，在优化机器在人机系统的性能方面，我们不在乎人们会有什么样的认知偏差，因

为它们的本质是引起人的内部状态异常，从而影响人机系统的性能。我们可以忽略认知偏差本身的某些属性，而将注意力集中在它在人机系统中的表现，这将系统输出作为黑匣子因素而受到影响[5]。

1.1.4 课题意义

因为人们往往需要在未知的场景中进行决策，同时在未知环境中人的不确定因素会显著影响系统的结果也就是性能。而在人机系统中，机器总是充当着系统的手和眼，也就是传感器，控制器和执行器的角色[9]。如果我们可以根据系统对人确定的可观测的状态和行为进行监测，并推断人的内部不确定状态，通过系统状态的反馈得出人的内部状态对系统性能的影响，从而可以使人机系统具有更好的性能。

本课题旨在研究基于马尔可夫决策过程的人机共享自治的方法来更好地实现特定目标。课题目标为基于马尔可夫决策过程建立用于钢琴教学的交互式共享自主系统。运用需要部分可观察的马尔可夫决策过程，以人的性能和注意力作为可观察的状态变量来进行自治权的切换。建立人机系统的模型，综合考虑人的内部状态、人的行为、机器的状态等，更好地实现人机共享自治,鉴于这个系统的部分状态可观测性和状态的时刻性以及结合前人的工作经验，初步的想法是运用部分可观的马尔可夫决策过程实现钢琴教学系统的共享自治，并在合适的场景下对模型进行仿真验证。

1.2 国内外研究现状综述

传统的自动控制系统只对机器的状态进行监测，以防止人为错误，提高安全性，很少再对人进行监测和喂养. 以电动汽车的驾驶为例,人和车是互动的两部分。当人们想要控制汽车时，汽车会给人们反馈。例如，如果发现一些异常情况，机器会提醒人们。该系统增加了对人的监控和对人机的反馈，可以早期发现异常情况，在一定程度上降低风险系数[10]。

由于人机系统缺乏与人的可靠和充分的意图沟通，需要从人的行为甚至从行为引起的环境变化中推断意图。决定什么时候反馈人与机器的状态，进行人机共享自治主体的转换，我们需要同时进行人的生理状态以及意图的估计和机器状态

的检测[11]。但是相对于机器的状态而言，对于人的意图或者状态大多数情况下并不能直接观测到或者是预测到，实际测量人类的心理或者说大脑状态在现有技术下也不可行，因此我们需要从另外的角度去思考，从人的行为去推断人类的意图。

既然无法直接观测人类的心理活动，那么可以将人类的行为精准描述为一组由马尔可夫链（Markov dynamic model, MDM）排列在一起的一组动态 MDP 模型，其中他们将多个动态模型定义为内部状态，他们使用这些马尔可夫模型来识别观察数据中的人类行为[12]。

在不确定和动态环境中的运动计划中，部分可观察的马尔可夫决策过程（POMDP）为不确定性下的规划提供了一个原则性的数学框架，这是机器在不确定性和动态环境中运行的基本能力[13]。然而，在人机交互技术中通常避免使用 POMDP，因为精确地解决 POMDP 在计算上是棘手的。就算了现在最好的算法也需要花费数小时来计算。这对现实的人机交互实现是完全不现实的。所以近年来，基于点的 POMDP 算法[6]通过计算良好的近似解而取得了令人瞩目的进步：具有数百个状态的 POMDP 在几秒钟之内就得到了解决。这些算法有可能使 POMDP 适用于机器人技术及其以外的许多应用。基于点的 POMDP 算法的关键思想是从置信空间中采样一组有代表性的点，并将其用作空间的近似表示。为了提高效率，最新算法从$R(b_0)$采样，在任意动作序列下，从给定点$b_0 \in B$可到达的点集。理论分析表明，当$R(b_0)$的覆盖范围较小时，可以有效地计算近似的 POMDPs 解[14]。

在关于通过最佳逼近可信任空间近似基于点的 POMDP 规划论文中，Hanna 和他的同事开发了一种新的基于点的 POMDP 算法，该算法利用最佳可达置信空间的概念来提高计算效率。在仿真中，他们成功地将算法应用于一组常见的机器人任务，包括海岸导航，抓取，移动机器人探索和目标跟踪，所有这些均建模为具有大量状态的 POMDP。

在使用连续状态 POMDP 的无人飞机防撞实验中，David 和他的搭档将飞机防撞建模为部分可观察的马尔可夫决策过程，并通过求解 POMDP 模型为防撞系统自动生成威胁解决逻辑。但是，现有的离散状态 POMDP 算法无法应对碰撞避免 POMDP 中的高维状态空间。使用新开发的称为蒙特卡洛值迭代的算法，他们构建了多个连续状态 POMDP 模型并直接求解，而不会离散状态空间。仿真结果表明，与早期的 2-D 离散状态 POMDP 模型相比，他们的 3-D 连续状态模型可将

碰撞风险降低多达 70 倍。这一成功证明了用于避免碰撞系统的连续状态 POMDP 模型的优点，以及解决这些复杂模型的最新算法进展[15]。

POMDP 可以用于表示不确定的环境中的未知状态，同时可以将这种未知的隐藏状态用于后续的任务中[16]。POMDP 策略是状态上的概率分布也就是信念映射到系统的操作上。这种信念分布由代理人在任务执行的过程中反馈的观察数据而进行更新迭代。而 POMDP 模型包括隐藏状态与代理商端所观察到的各种状态之间形成的概率关系。在 POMDP 中用于监控这种信念分布的方法中最常用是标准贝叶斯跟踪。部分可观的马尔可夫决策过程的目的是通过对信念的监控和更新，运用策略使预期回报最大化[17]。

贝叶斯推断方法与其他概率推理方法相比具有深厚的优势，因为贝叶斯能在获取信息有限的情况下可以做出尽可能好的预测。因为对于人这一个体而言，他们永远无法对所处的环境有完整而准确的认识，所以人通常会在不确定的情况下采取行动[17]。一般来说贝叶斯推理都是基于贝叶斯公式：

$$P(h|e) = P(h)\frac{P(e|h)}{(e)}$$

e是已知信息，h 是要求解的问题。贝叶斯定理即在已知先验概率$P(h)$和可能性函 数$P(e|h)/P(e)$的情况下，可以求得后验概率$P(h|e)$。通常情况下，先验概率$P(h)$是根据已有的经验预估出的h事件的概率，可能性函数$P(e|h)/P(e)$是一个调整因子，即新信息 B 带来的调整，作用是使得先验概率更接近真实概率。利用贝叶斯的原理，即通过专家知识或经验给出先验概率，使用观察到的信息来增强或减弱先验概率，得到后验概率[18]。

但是，需要解决方案的问题和已知信息往往并不是直接简单相关，因此需要包含一些相互关联的变量和状态以及变量的条件独立性的表示形式。基于这些问题需求，贝叶斯网络无疑是最优解。因为 BN 是有向无环图，用于编码条件独立性假设。在 BN 图中用节点来表示系统随机变量，用弧线表示变量之间的关系。考虑动态交互式人机系统的时变性，运用动态贝叶斯网络（DBN）来描述交互式人机系统的动态演变是最佳的解决方案。

1.3 论文组织结构

本课题论文的研究内容主要是实现马尔可夫决策过程下的钢琴教学交互式

共享自治系统。章节概述如下：

第 1 章 叙述论文的研究背景以及研究意义，介绍人机交互系统以及人机关系中人的作用。同时叙述了人的认知偏差对人机系统性能的影响，并对国内外的研究成果进行总结。最后概括了本文的文章组织结构。

第 2 章 运用 POMDP 决策方法可以在不确定性环境和条件下做决策的特性，基于 POMDP 对人机交互式系统进行建模研究，构造出一种从人机系统的人的可观测因素中去推断系统中人的未知的内部状态形成信念的方法，并运用这种方法对人机关系的自治权归属进行判别。

第 3 章 基于钢琴教学的共享自治方法仿真验证。为了验证本文提出的方法的有效性，本章基于钢琴教学在 Julia 平台上做了代码方面的仿真实验，结果显示本文提出的方法能够有效地根据人的行为来判断人的内在状态，并且能对其做出良好的反馈。

第 4 章 总结与展望。总结整个课题研究的内容与成果，提出课题研究过程中出现的问题和存在的不足。同时，展望课题后续可以开展的研究工作和未来的发展方向。

1.4 本章小结

本章作为全文的起始绪论部分，主要讲述本课题的研究背景和研究意义，然后对国内外 POMDP 模型的一些应用现状进行综述，最后对本文主要研究内容安排进行介绍。

第 2 章　基于 POMDP 的人机共享系统模型

因为我们设想运用系统对人类状态的反馈使系统的预期回报达到或者接近最大化，但是在人机系统的人机关系中，人的内部状态的未知的、不确定的，所以这就要求系统能在不确定的环境下做出决策策略。基于此，本课题将采用 POMDP 决策过程实现这一设想。

2.1　POMDP 模型基本方法介绍

2.1.1　MDP 概述

马尔可夫模型是一种统计模型，一般被广泛用于语音识别，语音标记的自动部分，语音到单词的转换，概率语法和其他自然语言处理应用程序。经过长期的发展，特别是在语音识别的成功应用中，它已成为一种通用的统计工具[19]。

马尔可夫决策过程是顺序决策的数学模型，用于模拟随机性策略和收益，该行为可以由具有系统状态的马尔可夫性质的环境中的代理实现。MDP 的名称来自俄罗斯数学家安德烈·马尔科夫（Andre Markov），以纪念他对马尔可夫链的研究。MDP 建立在一组交互对象上，即代理和环境。它的要素包括状态，行动，策略和奖励。在 MDP 的仿真中，智能体验感知当前的系统状态，根据策略对环境进行操作，从而改变环境状态并获得奖励。随着时间的推移，累积的奖励称为奖励。

MDP 的理论基础是马尔可夫链，因此也被认为是考虑了作用的马尔可夫模型。以离散时间建立的 MDP 称为“离散时间 MDP”，否则称为“连续时间 MDP”。此外，MDP 有一些变体，包括一些可观察到的马尔可夫决策过程，约束马尔可夫决策过程和模糊马尔可夫决策过程。

MDP 立足于 agent 与环境的直接交互，只考虑离散时间，假设 agent 与环境的交互过程可分解为一系列“阶段”，每个阶段由“感知——决策——行动”构成。MDP 模型是一个四元组 (S, A, T, R)，其中

- S是一个有限集，其中每一个元素$s \in S$代表一个状态；

• A是一个有限集，其中每一个元素$a \in A$代表一个行动；

• T 是状态转移函数，表示的是状态间的一组转移概率，$T(s,a,s') \coloneqq Pr\ (s_{t+1} = s'|s_t, a_t = a)$表示在状态 s 执行动作$a$达到状态$s'$的概率；

• R为回报（reword）函数，回报函数 $R(s,a)$ 表示在s上执行行动a所得到的即时回报；

MDPs 研究的主题是，给定一个 MDP 模型，如何求最优策略，即期望回报最大的策略。

2.1.2 POMDP 模型框架

POMDP 是环境状态部分可知动态不确定环境下序贯决策的理想模型，其核心点在于，agent 无法知道自己所处的环境状态，需要借助于额外的传感器，或者与其他的 agent 进行交互等方式才能获知自己的 state，能够客观、准确地描述真实世界，是随机决策过程研究的重要分支。

MDP 刻画了行动的不确定性：事先不知道，事后知道；观察是确定的，知道行动的效果。但是一般情况下，行动和观察都是不确定的。POMDP 是使用 MDP 在具有不完全可观察状态的系统模型中做出最佳决策的过程。POMDP 框架涵盖了人的隐藏的内部状态和具有不确定性的动作效果，作为通用的框架，适用于对实际场景下的序列决策问题进行建模，例如，智能机器人的视觉检测问题，教学交互系统以及某些场景下的一些具有不确定性的计划[20]。离散时间的 POMDP 可以由以下七个元组 $(S, A, T, R, \Omega, O, \gamma)$表示，其中：

• S是一个有限集，其中每一个元素$s \in S$代表一个状态；

• A是一个有限集，其中每一个元素$a \in A$代表一个行动；

• T是状态转移函数，表示状态之间的一组转移概率，$T(s,a,s') \coloneqq P_r\ (s_{t+1} = s'|s_t, a_t = a)$表示在状态 s 执行动作 a 达到状态s'的概率；

• R为回报（reword）函数，回报函数$R(s,a)$表示在s上执行行动a所得到的即时回报；

• Ω是一个有限集，其中元素称为“观察”；

• O称为观察函数，有时记为$O(s',a,o)$，a代表执行的行动，s'是a达到的结果状态，o在上述条件下出现的观察，$O(s',a,o)$是执行a达到 s'观察到 o 的

概率$P_r\ (o|s',a)$；

• $\gamma \in [0,1]$是折扣因子；

在 POMDP 问题中，“环境”被看成一个“黑箱”，一个状态是黑 箱某个时期的内部情况，观察是这个黑箱的“输出”。POMDP 的目标是在每个时间步长中找到最佳行动，以使累积回报最大化。POMDP 可以被认为是具有信念状态的马尔可夫决策过程（MDP）。MDP 算法可以分为值迭代和策略迭代。因此，还可以扩展值迭代算法以解决具有有限时间步长的 POMDP 的最优策略：

• 初始化：$t=0$；对于所有的$b \in B$，都有$V_0(b)\ =\ 0$；

• 若$|V_{t+1}(b)-V_t(b)|>\varepsilon$，对于所有的$b \in B$按下式计算 $V_{t+1}(b)$ ，t更新为$t+1$；

$$V_{t+1}(b)=\underset{a\in A}{max}\left[R^b(b,a)+\sum_{b'\in B}T^b(b,a,b')V_{t-1}^*(b')\right]$$

在实际应用中，在计算中难以解决部分 POMDP 问题，因此开发了一些离线或在线的近似规划算法。由于离线计划算法仍然是解决大规模问题的繁重工作，因此在线计划算法有许多扩展。在线规划算法的主要思想包括分支定界切割法，蒙特卡洛采样法和启发式搜索法[11]。

2.2 POMDP 模型基本方法介绍

通过用变量抽象来表示人机系统的每个部分，根据每个变量之间的关系，可以获得下图所示的人机系统框图，图中变量定义如下：

• 作为人内部状态的集合，可以是人的意图或生理状态，$s_h \in S_h$；

• 作为人的外部状态或行为的集合，$es_h \in ES_h$；

• 作为人的外部状态的一组观察，$\widetilde{es_h} \in O_{es_h}$；

• 人对机器施加的控制输入信息，$i_m \in I_m$；

• 作为人对机器施加的控制输入信息的观察，$i_m \in O_{i_m}$；

• 作为机器状态的集合，也可视为人机系统的输出$s_m \in S_m$；

• 作为机器状态/系统输出的一组观察，$\widetilde{s_m} \in O_{s_m}$；

• 对人的控制反馈集合（比如警告信号或者报错提醒），$a_h \in A_h$；

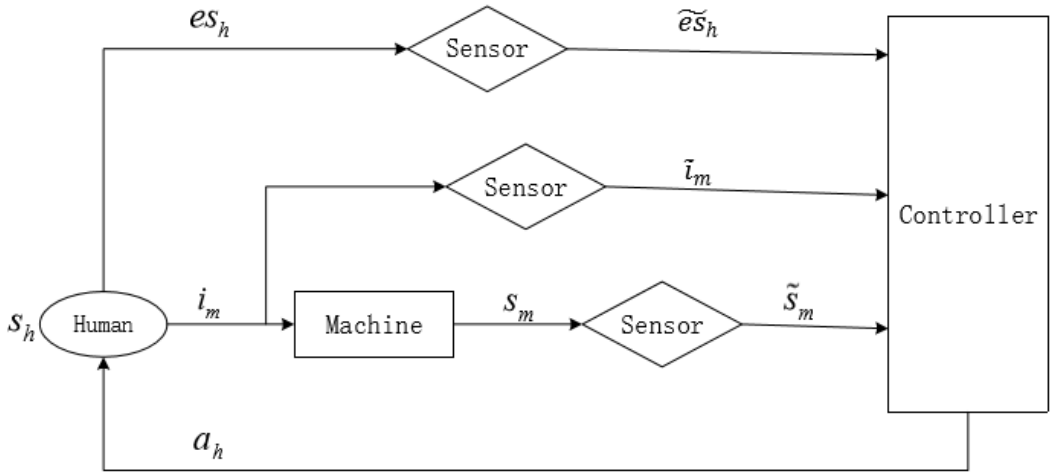

图 2-1　人机系统框图

如果假设上述所有集合都是有限的，那么根据上图可以得到，人类具有一种内部状态s_h，这可能是人类的意图或目标，或者是诸如期待、失望、烦躁等生理状态。对于人类，它将具有外部状态性能或行为es_h，并且还将根据其感知的机器状态或系统输出将控制的输入信息强加到机器上，以次达成人类的意图。例如钢琴教学系统中，如果学习者希望能够得到系统的控制权来进行教学材料的切换，那么他的外在表现则为学习能力下降，注意力不集中，甚至有可能停止演奏。

系统的控制输入是人的行为动作，所以系统的输出状态s_m就具有了时变的特性。而人的可观察的外部状态的性能或行为则通过各类传感器就那些测量。我们把人可观测的外部状态或行为的度量表示为人对机器施加的控制输入的度量es_h，作为对机器状态或系统输出的度量i_m，并将系统状态或系统输出的信息作为s_m。

在本文的课题实验重，系统的信念状态表示的是人的内部状态即隐藏状态的概率分布，即人对自主权的需求概率。由上述对 POMDP 框架的概述内容，我们可知需要首先取得状态转移概率以及观察函数，才能进行信念状态的更新迭代。所以接下来本文将对如何获得状态转移概率以及观察模型进行分别论述。

在人机系统中，由控制器先将传感器的观测结果以及各个变量之间的关系作为输入信号，然后，取决于是否添加了对人类的反馈，根据两个值的相对比较结果来预估隐藏状态s的概率分布，同时对人是否有偏差或异常内部状态进行判断预读。

因为该进程是同时针对所有时间步进行迭代计算，所以我们可以将全过程视为具有一组隐藏状态的马尔可夫决策过程。在这组 MDP 中即有行动集合以及控制集合 $S_h * I_m * S_m$，隐藏状态的集合A_h，也存在观测集 $O_{esh} * O_{im} * O_{sm}$ 和转移概率：

$$P(s'_h, i'_m, s'_m | s_h, i_m, s_m, a_h) = P(s'_h | s_h, i_m, s_m, a_h) \times P(i'_m | s_h, i_m, s_m, a_h) \times P(s'_m | s'_h, i'_m, s_h, i_m, s_m, a_h)$$

该步骤符合并基于概率链式规则。转移概率本身的计算过于复杂，但是通过做出一些合理的条件独立假设，我们可以对转移概率进行简化，从而更好的使用它我们假设是机器的状态只和当前时刻的机器状态以及人对机器的下一时间的控制输入信号，这可以称为机器动态模型，即：

$$P(s'_h | s_h, i_m, s_m, a_h) = P(s'_h | s_h, a_h)$$

第二个假设是人对机器的控制输入只取决于其下一时刻的内部状态以及机器当前时刻的状态，即：

$$P(i'_m | s'_h, s_h, i_m, s_m, a_h) = P(i'_m | s'_h, s_m)$$

同理最后一个假设是机器的状态只取决于当前时刻的机器状态和人下一时刻对机器的控制输入，我们将其称为机器动态模型，即：

$$P(s'_m | s'_h, i'_m, s_h, i_m, s_m, a_h) = P(s'_m | i'_m, s_m)$$

由上述可知，我们得出在 POMDP 框架下的人机系统的转移概率可以简化为：

$$P(\widetilde{es}_h, \tilde{s}_m, \tilde{s}_m | s_h, es_h, i_m, s_m, a_h) = P(\tilde{s}'_h | s_h, a_h) P(i'_m | s'_h, a_h) P(s'_m | i'_m, s_m)$$

在取得状态转移概率以及观察模型后，我们便可以对信念状态进行更新迭代。

2.3 本章总结

本章内容介绍了运用 POMDP 决策方法可以在不确定性环境和条件下做决策的特性，基于 POMDP 对人机交互式系统进行建模研究，构造出一种从人机系统的人的可观测因素中去推断系统中人的未知的内部状态形成信念的方法，并运用这种方法对人机关系的自治权归属进行判别。

第 3 章　基于马尔可夫决策过程的钢琴教学系统仿真实验

3.1　流程编译及优化

针对钢琴教学，传统的视频教学虽然节省了人力更为方便，但是重复的视频容易使用户失去专注，无法保证学习的效率；将控制权交予用户又会使得操作过于繁琐。所以需要设计交互式共享自主的钢琴教学系统。运用部分可观察的马尔可夫决策过程。以人的性能和注意力作为可观察的状态变量，使用二进制值来测量状态变量，在动态贝叶斯网络中对这些变量之间的过渡进行建模形成信念，推测用户的期望，从而来进行自治权的授予。

在学习的每个阶段，单独的 AI 系统都会为学生提供默认的音乐片段，供学生接下来练习。SharedKeys 在此系统上构建以支持用户自治：SharedKeys 具有分层策略，首先选择授予系统或用户自治权；然后如果系统决定采用默认课程，或由用户决定，则给用户细分选择。为了做出此决定，SharedKeys 既要保持对用户所需的自主权的估计，又要保持最有效的帮助，使用户能够快速地学习。该系统可以随着时间的推移与每个用户进行不同的交互，从而随着时间的推移根据个人的表现和注意力给予或不给予自主权。

更准确地说，我们使用 POMDP 形式化共享自治，其潜在状态空间是学习者期望的自治（对或错），可观察的状态变量（性能和注意力）和动作空间（给予自治但不给予自治）。在系统的第一次迭代中，我们使用二进制值来测量状态变量。我们在动态贝叶斯网络中对这些变量之间的过渡进行建模，并设置近似的过渡值。 我们通过 Julia 运行该模型，以生成决策树，该树实时确定给定观察到的性能和关注度后系统应采取的行动。SharedKeys 策略经过优化，可以最大程 度地提高我们对注意力的观察，我们认为这将使学生更快地沿着课程学习。

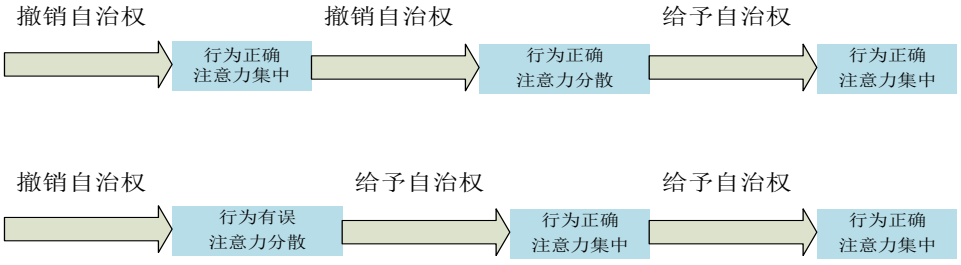

图 3-1　系统示例序列图

上图为根据不同学生的表现和专心程度，对不同学生的系统动作（给予自主权或不给予自主权）的示例序列。

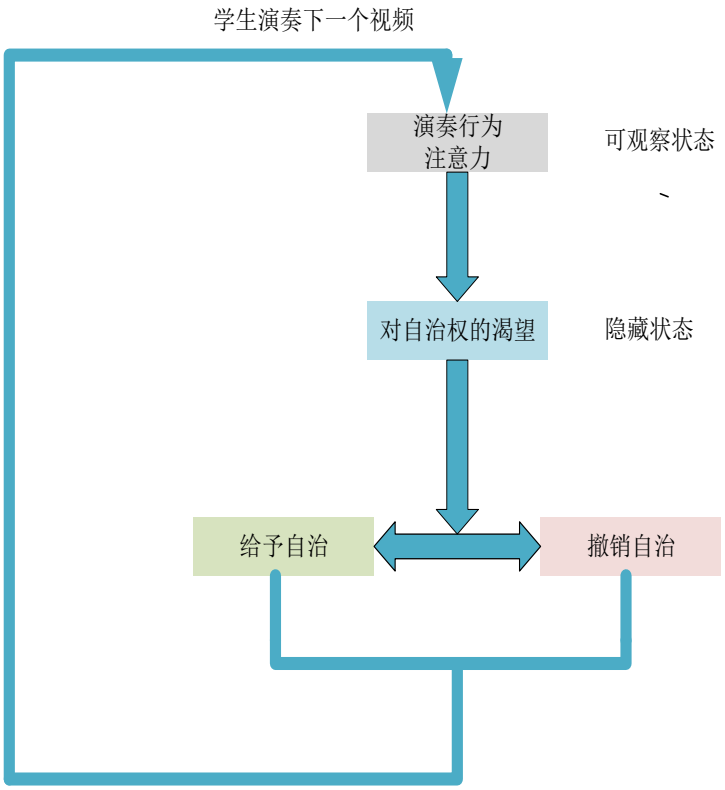

图 3-2　系统流程图

由上图可知，系统在第一个视频片段上观察用户的表现和注意力， 从而形成对自己期望的自主权的信念。使用这种信念，它选择给予或不给予自治。用户播放由系统或用户自己选择的下一个片段。这个过程一直持续到用户完成作品为止。

3.2　模型建立

教学系统是现在应用最广的的人机交互系统，且钢琴的教学系统对人行为的检测并不复杂，因此在本课题中我们基于钢琴教学来做实验，以验证本文提出的 POMDP 在识别人机系统中人的内部状态的预估和对最终成果的影响。针对仿真实验我们做如下假设：用户的目标是完整弹奏视频给出的教学音符；影响系统自治权授予的是用户的可观测外部状态；决定系统自治权的是用户内部对自治权的渴望；最终的成果是用户的学习效率。仿真实验的具体人机系统模型如下：用户的内部状态有：

$$S_h = \{\text{渴望自治}, \text{拒绝自治}\}$$

如果用户的内部状态时渴望自治，则系统会在下一个视频片段授予用户自治

权。反之，则不授予自治权。由这两个内部状态可以推断，用户可观测的外部状态或行为表现为两个方面：行为和注意力。所以：

$$ES_{h1}=\{\text{行为良好，行为有误}\}$$

$$ES_{h2}=\{\text{注意力集中，注意力分散}\}$$

当用户在演奏过程中对自治权的归属与系统产生分歧时，会通过外部的可观察因素表现出来，而被系统检测到。而系统内部则根据已有的外部观察以及动作后续的学习效率产生对系统自治权授予的信念，进而对用户进行系统动作：

$$A_h=\{\text{给予，撤销}\}$$

因为转移概率取决于人体内部状态模型、人类行为模型和机器动态模型。在实际问题的应用中，最好是能够从数据中学习得到相关的模型，但是为了方便起见，该仿真实验中我们根据人工对用户学习过程的观察得到外部状态，以此选择仿真中的概率，得到人体状态和系统信念的模型树图。

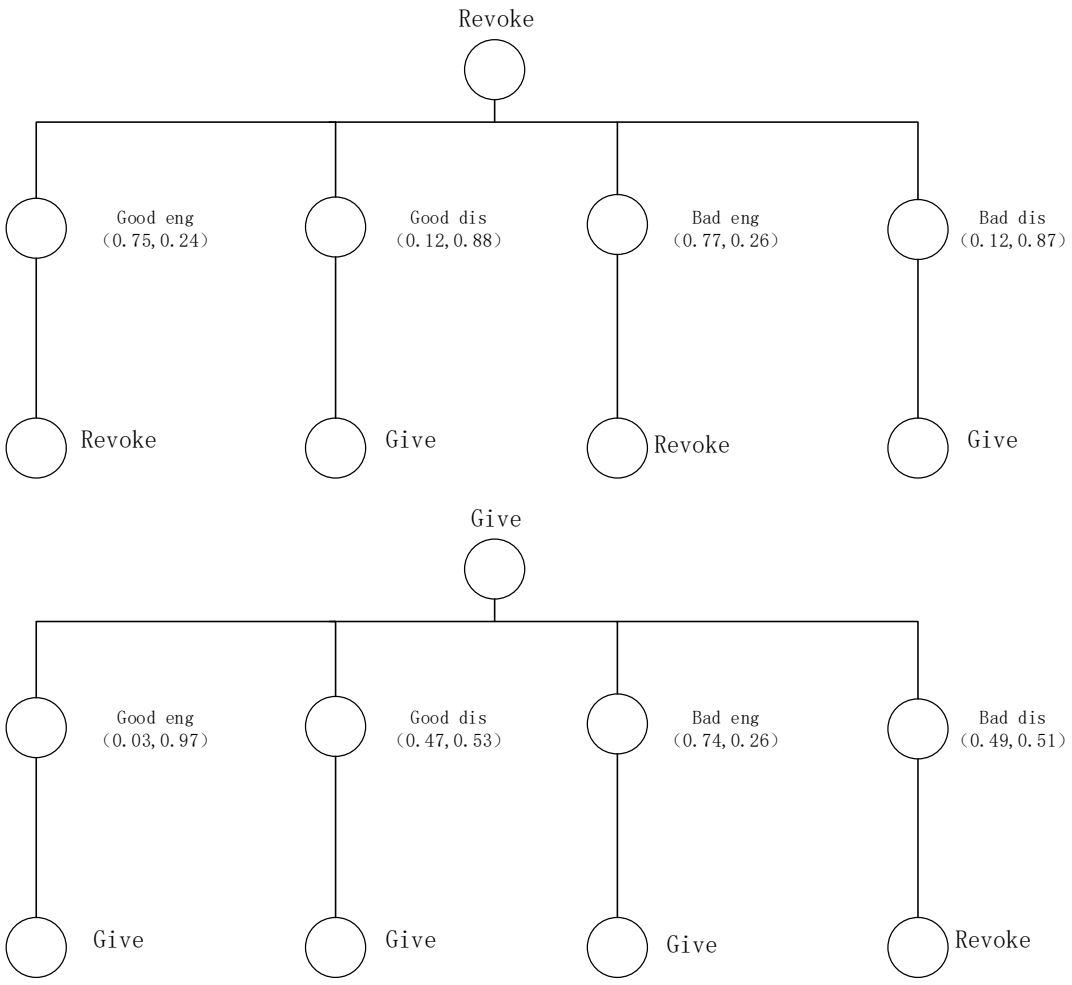

图3-3　人的状态和系统信念树图

3.3 仿真测试过程

3.3.1 仿真界面

实验部分验证上述方案的有效性，因为条件限制，我们采取虚拟钢琴来进行实验。运用 MATLAB 编写了一个简易的钢琴界面。

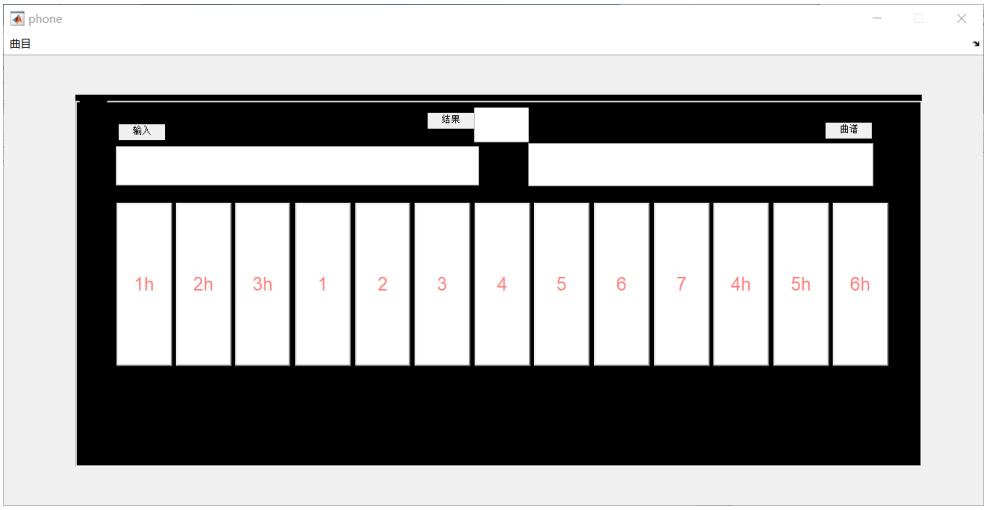

图3-4 虚拟钢琴演示界面

在上图的界面中，用户可以通过对应按键执行相应功能，当系统开始执行时，右边“曲谱”文本栏会显示默认曲谱，用户可以通过键盘进行学习弹奏，同时在左边“输入”文本栏显示用户的输入。当弹奏完毕后，系统会对照曲谱进行判断是否有误，并显示在“结果”文本栏中。

3.3.2 对照测试

寻找两名条件相似的同学做进行实验验证：一名同学完全由系统决定学习曲目，另一名同学则使用共享自治模型的仿真结果来进行学习。双方一共进行 5 段曲目学习（为方便展示每段曲目长度为 20）。

1. 第一段曲目

因系统初始状态为不给予自治权，所以双方时间基本一致，正确率也较低：

	所用时间/s	正确率/%
A 同学	66.37	75
B 同学（给予自治）	67.03	80

同时观察 B 同学这状态：

	演奏行为	注意力
B 外部状态	True	True

2. 第二段曲目

将 B 同学的行为输入模型，得到的最佳结果为“给予自治权”，所以切换 B 同学学习曲目为自动选择，A 同学继续保持默认。得到结果：

	所用时间/s	正确率/%
A 同学	60.13	85
B 同学（给予自治）	56.23	85

同时观察 B 同学这状态：

	演奏行为	注意力
B 外部状态	True	False

3. 第三段曲目

将 B 同学的行为输入模型，得到的最佳结果为“撤销自治权”，所以切换 B 同学学习曲目为自动选择，A 同学保持默认。得到结果：

	所用时间/s	正确率/%
A 同学	52.30	85
B 同学（给予自治）	54.63	90

同时观察 B 同学这状态：

	演奏行为	注意力
B 外部状态	False	False

后续以此类推，完成五段曲目的演奏。

4. 测试结果

完成五段学习曲目后，设定三个不同的曲谱，但长度都为 50 个字符，给予两位同学相同的曲谱，记录两位同学所用时间和正确率，同时计算三次测试的平均时间和正确率。

1. 第一次测试

	所用时间/s	正确率/%
A 同学	127.56	84
B 同学（给予自治）	120.33	86

2. 第二次测试

	所用时间/s	正确率/%
A 同学	124.63	84
B 同学（给予自治）	125.82	82

3. 第三次测试

	所用时间/s	正确率/%
A 同学	122.44	82
B 同学（给予自治）	118.32	88

4. 综合平均

	所用时间/s	正确率/%
A 同学	124.88	82.67
B 同学（给予自治）	121.53	85.33

3.3.3 对照测试（2）

在第一次的对照实验中，虽然在最后的测试结果中通过共享自治的钢琴系统进行学习的 B 同学的正确率和所用时间都优于 A 同学，但是学习样本和测试样本都较少两位同学的差异不明显，且无法排除偶然性的因素。所以在后续实验中在五段学习曲目的基础上继续就进行了十段曲目、十五段曲目、二十段曲目的学习测试，得到结果如下：

表3-1 AB 同学测试平均时间

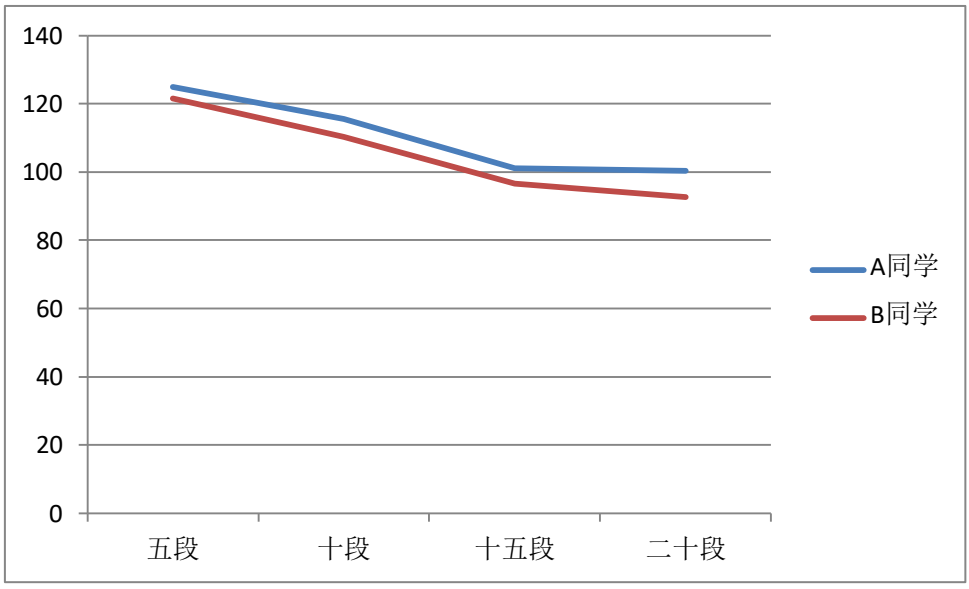

表3-2 AB 同学测试平均正确率

3.4 实验结果

通过上述实验，我们可以得知，在两名同学同样进行相同数量曲谱的学习的前提下，A 同学通过传统方法由系统默认曲谱进行学习，而 B 同学则通过共享自治的钢琴系统进行学习，在演奏学习过程中 B 同学进步比 A 同学更加迅速；在最后的测试演奏中不论是所用时间还是正确率，B 同学都显著优于 A 同学。

由此，我们可以得出结论，在使用共享自治方法之后，用户的学习效率相较于传统的机器视频教学的学习效率其效果是显著增加的。

第 4 章　总结与展望

4.1　毕设工作总结

由于人机具有各自的优势和薄弱环节，因此人机协作完成任务往往比单独完成人或机更有效，所以产生了人机交互式系统。在许多领域中，交互式系统要么为我们自主地做出决策，要么为我们提供决策权并发挥支持作用。但是，许多设置例如在教育或工作场所中的设置,受益于在用户和系统之间共享这种自主权，并因此受益于随着时间的推移而适应他们的系统。我们介绍了一种用于钢琴教学的交互式共享自主系统，可播放乐曲的不同片段，供学生模仿和练习。仿真研究表明，与系统共享自主权的学生学习得更快，并且认为系统更加智能。

本课题将认知偏见从狭义的心理学定义扩展到人的内部状态，包括生理和心理部分。因为人机系统人的内部状态会影响人的外部行为，进而对系统的性能也就是最终的目标造成影响，所以通过对人外部可观测状态的监控，推测人内部状态的信念分布，进而达到系统目标最优性能。本课题的研究内容是通过识别钢琴教学中人的行为和注意力，推测人内部对系统自治权获取的信念状态，从而达到钢琴学习效率的最优化。本文的主要工作如下：

1)　介绍本课题的研究背景以及研究意义，介绍人机交互系统以及人机关系中人的作用。

2)　运用 POMDP 决策方法可以在不确定性环境和条件下做决策的特性，基于 POMDP 对人机交互式系统进行建模研究，构造出一种从人机系统的人的可观测因素中去推断系统中人的未知的内部状态形成信念的方法，并运用这种方法对人机关系的自治权归属进行判别。

3)　改造合适的 MDP 模型建立于钢琴教学的交互式共享自主系统，通过对系统自治权是控制实现系统功能的最优化。

4.2　未来展望

通过文献回顾与分析，我了解到运用马尔可夫决策过程来进行共享自治的方

法不仅可以延伸到其他的教学领域，对现在热门的智能驾驶技术也是有极大帮助的。智能驾驶不仅仅是对道路情况的检测，同时也需要对驾驶员的情况进行综合考量。

如果你的目标是让车辆保持在单车道的平稳匀速行驶，而驾驶车辆的主体也就是人的自身生理状态可能是困倦的也可能是清醒的，这就对机器也就是汽车本身的控制器提出来不同要求。当驾驶员陷入昏昏欲睡的状态时，人机系统应该给予检测并反馈给控制器，由控制器再给予主体报警信号。如果警告无法奏效，则控制器就会接管电动汽车的方向盘，予以汽车减速直至制动，让车停在车道中间。这就是人机共享自治的初步实现，进一步可以实现在紧急情况下，自动控制器可以接管人的控制，进而尽可能减少车祸的发生或者将损失降到最低。

甚至，当对人和道路进行综合考量之后，可以设计出驾驶权在人和系统之间快速有效切换的共享自治的自动驾驶汽车，实现复杂路况的自动驾驶。

参 考 文 献

[1] 李玲，解洪成，陈圻．复杂人机系统人机协作模型的探讨[J]．人类工效学，2007, 13(4): 36-38.

[2] Haselton M G, Nettle D & Andrews P W. The evolution of cognitive bias. In D. M. Buss (Ed.), The Handbook of Evolutionary Psychology[J]. John Wiley & Sons Inc. 2005, 724-746.

[3] Gigerenzer G, Goldstein D G. Reasoning the fast and frugal way: Models of bounded rationality[J]. Psychological Review. 1996, 103(4): 650-669.

[4] Tversky A & Kahneman D. Judgement under uncertainty: Heuristics and biases[J]. Science. 1974, 185(4157): 24-31.

[5] Cialdini R B, Martin S J, & Goldstein N J. Small behavioral science-informed changes can produce large policyrelevant effects[J]. Behavioral Science & Policy, 2015, 21-27.

[6] 周鹏生．认知偏差的产生及其与认知闭合需要的关系[J]．心理学研究，2017, 10(5): 11-18.

[7] Tahboub K A. Intelligent Human-Machine Interaction Based on Dynamic Bayesian Networks Probabilistic Intention Recognition[J]. Journal of Intelligent and Robotic Systems, 2006, 45(1): 31-52.

[8] Pentland A and Liu A. Modeling and prediction of human behavior[J]. Neural Computation, 1999, 11(1): 229-242.

[9] Takano W, Matsushita A, Iwao K, et al. Recognition of human driving behaviors based on stochastic symbolization of time series signal[C]. IEEE/RSJ International Conference on Intelligent Robots & Systems, 2008, 22-26.

[10] Chipalkatty R, Daepp H, Egerstedt M, et al. Human-in-the-loop MPC for shared control of a quadruped rescue robot[C]. IEEE/RSJ International Conference on Intelligent Robots & Systems, 2011, 25-30.

[11] Munir S, Stankovic J A, Liang C J M, et al. Cyber Physical System Challenges for Human-in-the-Loop Control[C]. Presented as part of the 8th International Workshop on Feedback Computing, 2013.

[12] Anderson S J, Peters S C, Pilutti T E, et al. An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios[J]. International Journal of Vehicle Autonomous Systems, 2010, 8(4):190.

[13] Sutton RS, Barto AG. Reinforcement learning: an introduction[J]. IEEE Trans Neural Netw. 1998, 9(5):1054.

[14] Colman AM. Cooperation, psychological game theory, and limitations of rationality in social interaction[J]. Behav Brain Sci. 2003, 26(2):139-153.

[15] Broz F, Nourbakhsh I, Simmons R. Planning for Human–Robot Interaction in Socially Situated Tasks[J]. International Journal of Social Robotics, 2013, 5(2):193-214.

[16] Kaelbling LP, Littman ML, Cassandra AR. Planning for Human – Robot Interaction in partially observable stochastic domains[J]. Artif Intell. 1998, 101(2):99-134.

[17] Bosch Lt, Oostdijk N, Ruiter JPd. Durational aspects of turn-taking in spontaneous face-to-face and telephone dialogues[C]. Proc of text, speech, and dialogue, 7th international conference (TSD). 2004, 563-570.

[18] 王丽. 浅析贝叶斯公式及其在概率推理中的应用[A]. 科技创新导报[C]. 北京: 北京师范大学出版社, 2010, 136-138.

[19] 章宗长,陈小平. 杂合启发式在线 POMDP 规划[J]. 软件学报, 2013, 24(7): 1589-1600.

[20] Paul Dagum, Adam Galper, Eric Horvitz. Dynamic Network Models for Forecasting[C]. Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence. 1992, 41-48.

附 录

```
using POMDPs, POMDPModelTools, POMDPPolicies, BeliefUpdaters,
POMDPSimulators, Random
using QMDP
using ParticleFilters
using BasicPOMCP

struct State
    # Latent variables
    desired_autonomy::Bool

    # Observable variables
    performance::Bool
    given_autonomy::Bool

    # last engagement, to be used for reward
    engagement::Bool
end

# struct Act
#     give_autonomy::Bool
# end

struct Obs
    performance::Bool
    given_autonomy::Bool

    # Using duration (1 = engaged/'just right', 0 = too long / too short on task)
    #   as a proxy for engagement
```

```
    duration::Bool      # engagement
end

struct MOMDP <: POMDP{State, Symbol, Obs} #TODO mutable struct - ideally make p_ability change over time
    # CPT: P(u' | u, p, gu)
    # 上一时刻的状态（期望自治程度、表现、是否给予自主权） 影响 当前时刻的期望自治程度，即影响用户的内部状态
    p_autonomy_when_desired_good_given::Float64
    p_autonomy_when_desired_good_not_given::Float64
    p_autonomy_when_desired_bad_given::Float64
    p_autonomy_when_desired_bad_not_given::Float64
    p_autonomy_when_not_desired_good_given::Float64
    p_autonomy_when_not_desired_good_not_given::Float64
    p_autonomy_when_not_desired_bad_given::Float64
    p_autonomy_when_not_desired_bad_not_given::Float64

    # CPT: P(i' | u', gu')
    # 当前时刻的期望自治程度 + 当前时刻是否给予自主权 影响 下一时刻的注意力
    p_engaged_when_desired_given::Float64
    p_engaged_when_desired_not_given::Float64
    p_engaged_when_not_desired_given::Float64
    p_engaged_when_not_desired_not_given::Float64

    # For now, ability is a probabilistic constant for a student that determines performance independent of attempt
    # 把能力定义为一个概率常数，不随时间变化
    p_ability::Float64
```

```
    # Reward for being engaged ("just right", vs. took too long or too short ;
    #     using duration as a proxy for engagement)
    # 根据注意力给出的 reward
    r_engagement::Float64

    discount::Float64   #折扣因子
end

# Transition values from CPTs for default constructor
MOMDP() = MOMDP(0.9, 0.9, 0.3, 0.8, 0.8, 0.1, 0.01, 0.2,
                0.9, 0.3, 0.2, 0.9,
                0.5, # p_ability      TODO: draw from distribution (first pass: tune manually to see diffs)
                1.0, # r_engagement
                0.95 # discount
                )

POMDPs.discount(m::MOMDP) = m.discount

const num_states = 2*2*2*2
const num_actions = 2
const num_observations = 2*2*2
POMDPs.n_states(::MOMDP) = num_states
POMDPs.n_actions(::MOMDP) = num_actions
POMDPs.n_observations(::MOMDP) = num_observations

POMDPs.actions(m::MOMDP) = [:give_autonomy, :revoke_autonomy]

function POMDPs.actionindex(m::MOMDP, a::Symbol)
    if a == :give_autonomy
```

```
        return 1
    elseif a == :revoke_autonomy
        return 2
    end
    error("invalid MOMDP action: $a")
end

const all_observations = [Obs(performance, given_autonomy, duration) for performance = 0:1, given_autonomy = 0:1, duration = 0:1]
POMDPs.observations(m::MOMDP) = all_observations

function POMDPs.transition(m::MOMDP, s::State, a::Symbol, rng::AbstractRNG=MersenneTwister(1))
    sp_desired_autonomy = true
    sp_engagement = true
    sp_performance = rand(rng) < m.p_ability ? true : false

    # Next latent state of desired autonomy P(u' | u, p, gu)
    # If user wants autonomy
    if s.desired_autonomy
        # Does well
        if s.performance
            # And we give them autonomy
            if a == :give_autonomy
                # Then the prob for next desired_autonomy, and the given autonomy, updated in the state
                p_sp_desired_autonomy = m.p_autonomy_when_desired_good_given
                sp_given_autonomy = true
            else
```

```
                    p_sp_desired_autonomy =
m.p_autonomy_when_desired_good_not_given
                    sp_given_autonomy = false
                end
            else
                if a == :give_autonomy
                    p_sp_desired_autonomy =
m.p_autonomy_when_desired_bad_given
                    sp_given_autonomy = true
                else
                    p_sp_desired_autonomy =
m.p_autonomy_when_desired_bad_not_given
                    sp_given_autonomy = false
                end
            end
        else    # user doesnot wants autonomy
            if s.performance    # does well
                if a == :give_autonomy      # give autonomy
                    p_sp_desired_autonomy =
m.p_autonomy_when_not_desired_good_given
                    sp_given_autonomy = true
                else
                    p_sp_desired_autonomy =
m.p_autonomy_when_not_desired_good_not_given
                    sp_given_autonomy = false
                end
            else
                if a == :give_autonomy
                    p_sp_desired_autonomy =
m.p_autonomy_when_not_desired_bad_given
```

```
                sp_given_autonomy = true
            else
                p_sp_desired_autonomy = m.p_autonomy_when_not_desired_bad_not_given
                sp_given_autonomy = false
            end
        end
    end

    # Next engagement level P(i' | u', gu)
    if sp_given_autonomy
        p_sp_engagement_desired = m.p_engaged_when_desired_given
        p_sp_engagement_not_desired = m.p_engaged_when_not_desired_given
    else
        p_sp_engagement_desired = m.p_engaged_when_desired_not_given
        p_sp_engagement_not_desired = m.p_engaged_when_not_desired_not_given
    end

    # Let's say performance is a general ability that's constant throughout the curriculum for now
    # 假设 performance 是一种普遍的能力，在整个课程中是不变的
    p_sp_performance = m.p_ability

    return SparseCat(sps, probs)  # SparseCat(values, probabilities)创建一个稀疏的分类分布
end

# Rewarded for being engaged  改善注意力有助于提高学习效率
function POMDPs.reward(m::MOMDP, s::State, a::Symbol)
```

```
    return s.engagement ? m.r_engagement : 0.0 #TODO: try -1.0 here
end

# initial_state_distribution(m::MOMDP) = SparseCat(states(m), ones(num_states) / num_states)
p_initially_motivated = 0.5 # 0.5 is uniform prior
#
State{desired_autonomy,performance,given_autonomy,engagement}
init_state_dist = SparseCat([State(true, false, false, false), State(false, false, false, false)], [p_initially_motivated, 1.0-p_initially_motivated])
POMDPs.initial_state_distribution(m::MOMDP) = init_state_dist


# Solver
momdp = MOMDP()

# QMDP
# solver = QMDPSolver(max_iterations=20, belres=10.0, verbose=true)
solver = QMDPSolver(max_iterations=100, belres=1e-3, verbose=false)
# solve 函数在 DiscreteValueIteration 包中的 vanilla.jl 文件中
# 可以在这个文件中的 solve 函数里加一些 println 将想要看得更清楚的步骤显示出来
# 先在 TigerPOMDP 问题里试验一下
# 目前对钢琴问题里面啊α -向量求解过程还不是很清楚 ————————————————————————————2020/1/6————————————————————————————2020/1/6
policy = solve(solver, momdp) # solve()——>ValueIterationPolicy()
# print(policy)
```

```
# 构造一个顺序重要性重采样粒子滤波器
# 调用一个基础粒子滤波，实现 POMDPs.jl 的更新器接口
# 即 filter 的类型是 Updater 类型
filter = SIRParticleFilter(momdp, 10000) # 不知道这个滤波器是怎么起作用的，即如何实现更新器接口
**********************************************************************

init_dist = initial_state_distribution(momdp)
init_belief = initialize_belief(filter, init_dist)

# input 函数输出 true / false
# 可传入的参数为 performance、engagement
function input(ask::String="performance")::Bool
    if ask == "performance"
        prompt = "performed well? (y/n) "
    else
        prompt = "engaged? (y/n) "
    end
    print(prompt)
    user_input = chomp(readline()) # chomp(s)从字符串中删除一个尾随换行符
    if user_input == "n"
        return false
    else
        return true
    end
end

function unroll_particles(particles::ParticleFilters.ParticleCollection{State})
    d = [0.0 for i = 1:num_states]
```

```
    d = []      # 为什么对 d 这个数组有两个不同的定义？ **********
    for i = 1:num_states
        push!(d, pdf(particles, all_states[i]))
    end
    return d
end

function generate_next_action(particle_belief::Any=init_belief, iteration::Int64=1)
    println("Step: ", iteration)
    belief = unroll_particles(particle_belief)
    println("Belief: ", belief)
    # println(policy.alphas)
    Alpha = policy.alphas
    value1 = 0
    value2 = 0
    i = 1

    while i <= 16
        value1 += belief[i]*Alpha[1][i]
        value2 += belief[i]*Alpha[2][i]
        i = i + 1
    end
    println("value of give_autonomy : ",value1)
    println("value of revoke_autonomy : ",value2)

    if value1 > value2        #   size(belief)[1] = 1
    # if action_idx == 1
        action = :give_autonomy
        action_val = true
    else
```

```
            action = :revoke_autonomy
            action_val = false
        end
        println("Next action is: ", action)
        # Input user's performance and engagement
        inputed_performance = input("performance")
        inputed_engagement = input("engagement")

        o = Obs(inputed_performance, action_val, inputed_engagement)
        next_belief = update(filter, particle_belief, action, o)
        return generate_next_action(next_belief, iteration + 1)
end

generate_next_action()
```

致　谢

大学四年果然比想象中更快地溜走了，作为人生中不可或缺的四年，一路走来，喜怒哀乐。有失望的泪水、奋斗的汗水，也有收获的喜悦。有志同道合的同学伙伴，更有循循善诱的师长。在这个接近四年尾声的时刻，对曾经帮助过我的老师同学和默默付出的父母表达深深的感谢。

首先，应当感谢的是母校浙江工业大学。是她给我提供了一个优越的学习环境，使我可以安心地在知识的海洋里畅游，追寻自己的梦想。

然后，非常需要感谢的是我的指导教师赵云波教授。赵老师严谨的治学态度深深吸引着我。从最初的选题、开题答辩，到后面的中期答辩、毕业论文的撰写，赵老师给予了我非常大的帮助。每次遇到瓶颈觉得无从下手的时候，与赵老师的交流总能让我豁然开朗，帮助我明确了前进的方面。正是有了赵老师的指导，我在进行毕业设计的过程中才少走了许多弯路。另外也非常感谢指导我毕设研究的吴芳学姐，在整个毕业设计的过程中无论我在什么时候提出问题，学姐都能及时详细地为我解答，学姐的督促和指导是我毕业设计能顺利完成的又一大助力。

最后，我想特别感谢我的父母，是你们给我提供了良好且来之不易的学习条件，使我不用担心除学习以外的事情。每当我遭受挫折或者面临抉择时，与你们的交谈总能让我安心乐观，是你们给了我走下去的勇气和动力。我们怀着同样的心、以不同的方式朝着同样的目标前进——为了更好的自己、为了更好的家，再次感谢我的父母。

大学四年，我收获的不仅仅是知识和技能，更多的是面对未知挑战的信心，勇于挑战自己的决心以及面对失败的勇气。尽管我可能还并不成熟，还有着这样那样的不足，尽管在未来的道路上还有很多的挑战在等待着我，但是，我一定会坚持下去，找到属于自己的道路！