

# 【神经符号 AI 讲座】复杂环境中的机器人规划与实践

**编者按：**《复杂环境中的机器人规划与实践》是任伟在上海交通大学神经符号人工智能暑期学校开学典礼暨社区研讨会上的“社区共享项目”研究进展报告。机器人规划中的不确定性、部分可观察性和目标可达性等问题，需要通过概率模型、马尔可夫决策过程和蒙特卡洛方法等工具来解决。基于概率的 PDDL 和马尔可夫决策过程模型为机器人提供了在复杂环境中进行高效规划的能力。通过结合传感器数据、滤波算法和概率规划，机器人能够在不确定条件下实现精确控制和任务执行。最后，以动力电池的拆解任务为例，对复杂动态环境中的机器人规划进行实践，研究了基于神经符号 AI 的机器人任务和运动规划系统，实现了在复杂多变的拆解工作环境中动力电池连接约束条件的智能拆解，验证了系统的可行性。

## 一、引言

### 1、机器人的控制

众所周知，机器人领域涵盖了各种形态和功能的机器人。图 1 中展示的是一本日本机器人控制入门书籍设计的案例，内容浅显易懂。

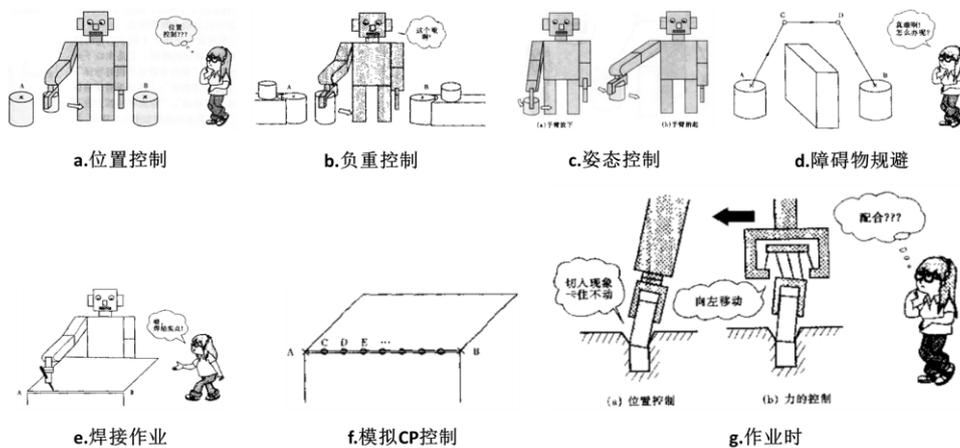


图 1. 机器人控制入门

- **位置控制**（图 1，a）：探讨机器人如何从 A 点移动到 B 点，具体路径和策略是什么。
- **负重控制**（图 1，b）：分析机器人在携带物体时，如何从 A 点移动到 B 点，考虑负重对运动的影响。

- **姿态控制**（图 1, c）：研究机器人在不同姿态下，如何调整力的控制以适应不同的任务需求。

- **障碍物规避**（图 1, d）：探讨机器人在遇到障碍物时，如何调整路径和控制策略以安全通过。

这些案例的讲解不仅有助于理解机器人控制的基本原理，而且在后续的电池拆解等复杂任务中，这些知识将发挥重要作用。通过这些实例，我们可以更深入地掌握机器人控制的关键技术。例如，在焊接作业中，从 A 点如何平滑运动到 B 点是一个关键问题（图 1, e）。目前，我们通常采用插值法来处理这一问题（图 1, f）。具体来说，就是将路径上的点密集地插入，例如 A、B、C、D、E 等点，逐步细化路径。插值的密度主要取决于机器人模型的特性，运动速度和任务控制精度等。在电机控制层面，当其接收到一个动作指令或波形时，电机会相应地动作，进而驱动机器人本体移动一定的距离。理想情况下，我们希望电机动作与机器人移动的距离达到 1:1 的比例。如果无法实现 1:1，则需要通过传动比来调整，例如电机转动三圈，机器人刚好完成差值距离的移动，这是一种较为理想的状态。这种方法实际上是一种模拟的连续点控制（CP 控制），即通过连续的点来控制机器人的运动。

机器人控制方法很多，如位置控制、力控制、非线性控制和自适应控制等等，其中位置控制和力控制（图 1, g）和我们的工作息息相关。位置控制关注的是机器人能否准确到达指定位置。例如，在拆卸螺丝时，螺丝旋具能否准确插入到螺栓中，以及每次插入深度是否一致。位置控制的目标是确保机器人在每次操作中都能准确到达相同的位置。而力控制是调节机器人末端执行器或系统的作用力，不仅仅是控制位置。它通常用于需要与环境交互的任务，比如抓取物体、打磨、装配或与人类协作的场景。在我们试验场景中，更加关注机器人末端执行器能否施加所需的力。例如，拧松螺丝可能需要 35 牛米的力，机器人能否准确施加这一力值，取决于力传感器的反馈以及末端执行器电机输出扭矩大小。电机控制通常采用三环控制，包括位置环、速度环和电流环，其中与力控制密切相关的是电流环。力控制是高层次的控制目标，通常通过调用电流环实现力的控制。而电流环是底层控制环，直接作用于伺服电机，通过调节电机电流实现精确的力矩输出。

在机器人系统设计和参数选型方面，通常需要重点考虑机器人的负载能力以及负载和性能之间的关系。

在机器人执行抓取任务时，例如完成从 A 点到 B 点的运动并抓取相应物品，其控制流程通常是：首先，系统会判断机械臂末端工作空间是否可达 B 点；若可达，则运动到 B 点后继续执行下一步操作，如抓取目标物体（电池）等。在抓取过程中需要确保物体不滑动，不仅需要考虑物体的形状、材质和重量，同时结合机器人力控算法、传感器反馈和夹爪设计进行优化。随后，机器人将物体转移到目标位置并完成释放，整个流程结束。这是机器人执行任务的标准流程。在施加力的补偿方面，我们通常基于已知负载重量和系统动力学模型进行计算，并结合机器人建模估算所需的作用力。常见的控制方法包括非线性控制和自适应控制，而力的补偿通常采用阻抗控制，通过调节虚拟阻抗参数实现力-位置关系的调节。未来有机会，我们可以进一步探讨这些技术细节以优化控制精度。

## 2、电池的机器人拆解任务

如图 2 所示，这是一个大型电池包，通过转运设施将其运送至拆解工作站。工业机械臂负责拆解电池包中的螺栓和其他关键连接件。机器人拆解电池通常包含三类主要任务：



图 2.电池的机器人拆解任务

- **紧固件和连接件的拆解**（图 2，a）：包括正常拆解和异常情况下的破坏性拆解。异常情况可能涉及电池腐蚀、生锈或材料损坏，这些在实际拆解场景中较为常见。

- **搬运**（图 2，b）：根据电池的状态，机器人通过规则抓取或无规则抓取，并将电池转运至指定位置。

- **转运**（图 2，c）：将电池从一个位置移动到另一个位置。

电池包转运、螺丝拆卸、连接件切割、零部件抓取等通常是预编程的短程任务，其特点是任务简单、环境可控、精度要求高、响应频率高，涉及主体（工业机器人）、客体（退役电池）、受体（目标位置的螺丝）和载体（执行任务的末端执行器），根据具体场景提前设计。比如，在路径规划方面，机器人（主体）需要从起点到终点找到一条最优路径，确保无碰撞且路径最短，而具体的拆解动作则取决于任务需求和退役电池（客体）的状态。在执行任务前，还需满足一定的前提条件。例如，拆解螺丝需要末端执行器的套筒（载体）与螺丝（受体）的精确配合，否则可能导致工具或工件损坏。最后，任务执行的结果，如拧松螺丝，是动作完成的直接体现。

另一类是机器人长程任务，通常指机器人在较大空间范围内完成的复杂操作，例如，基于神经符号具身智能框架设计的 BEAM-1（图 2，d），具有自主感知、自动规划、精确执行和持续学习的能力，只要给出可拆卸性连接件的大概位置，就能自主、连续完成大范围连接件的拆卸工作。这类任务的特点是任务范围广、环境动态复杂、依赖全局规划与控制，且执行时间较长。由于环境的不确定性和任务的复杂性，长程任务对机器人的感知、学习、决策、控制能力提出了更高要求。接下来我们将详细探讨规划部分的具体细节。

## 二、机器人规划

### 1、什么是规划？

规划的核心在于如何方便地表示问题，并高效地求解。为了更好地理解这一概念，我们可以从一个小游戏入手，比如俄罗斯方块或汉诺塔（图 3，a）。这些游戏的本质都是从初始状态到目标状态，通过一系列操作找到最优的解决方案。

举个例子，假设我要从上海去北京，我有多种选择：步行、骑行、坐飞机、乘大巴、自驾或坐火车。每种方式都有其优缺点，而最优的选择取决于当前的状态和约束条件。如果时间紧迫，比如需要在 4 小时内到达，那么飞机或高铁可能是最合理的选择；但如果时间充裕，慢悠悠的方式也未尝不可。这就是规划的核心——在给定的约束条件下，找到从初始状态到目标状态的最优路径。

规划的本质是通过一系列行为将初始状态转变为目标状态（图 3，b）。我们可以将这一过程进一步细分。以去北京为例，假设我选择高铁，途中可能会经过

南京、徐州或济南，我可以在这些地方停留处理一些事务。无论路径如何细分，只要在约束范围内找到从起点到终点的合适路径，这就是规划。

通常情况下，规划是正向思考的，即从当前状态逐步向目标状态推进。这种思考方式会形成一个“行为树”或“状态树”（图 3，c），从初始状态到目标状态，通过多个阶段逐步细分，直到无法再分为止。这种方法虽然传统，但在任务量较小且可控的情况下，仍然是最优的选择，尤其是在成本有限的情况下。

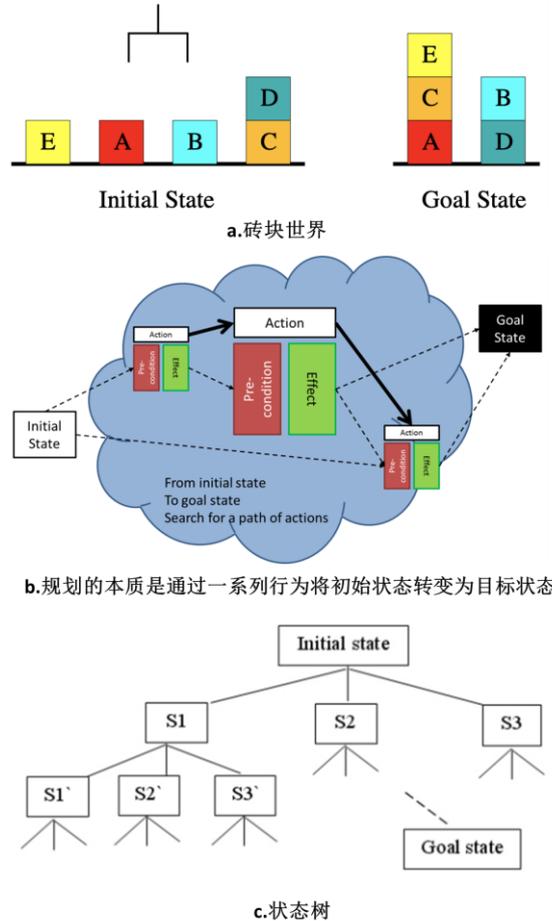


图 3.什么是规划

### 规划的代表与求解

规划主要包括两个部分：**表示**和**求解**。表示是指根据当前状态采取何种行动，而求解则是找到从初始状态到目标状态的操作顺序。在求解过程中，我们通常会根据状态进行搜索。例如，**状态空间搜索**，通过遍历所有可能的状态，找到从初始状态到目标状态的最优路径；**偏序规划**是一种选择性搜索方法，如果我认为某个路径更优，我会优先搜索该路径；**规划图**，类似于使用百度地图进行路线规划，也是一种从初始状态到目标状态的操作顺序搜索，通过图形化的方式表示状态和

行为的转换关系。当然，规划的方法远不止这些，选择最适合当前任务的方法才是关键。

### 状态与行为的表达

接下来，我们需要讨论如何表示状态和行为。**状态表示**是规划的基础，它描述了当前的环境和条件；**行为**则是从当前状态到下一状态的转换操作。规划就是将这些行为和状态组合起来，形成一个完整的解决方案。

我们可以将状态定义为一个**有限的命题符号集**，即一组状态的集合。每个任务的状态可以被视为这个符号集的一个子集。通过行为，我们可以改变当前的状态，而状态的改变会带来相应的结果。每一步行为都改变了当前的状态，这就是状态转移的首要意义——**当前的状态是由行为驱动的**。规划问题的核心在于从**初始状态到目标状态**，通过一系列合理的子任务逐步实现目标。换句话说，规划就是找到一系列动作，将初始状态转变为目标状态，同时完成所有必要的子任务。

这是一种传统的规划方式，通过清晰地表示状态和行为，我们可以更高效地进行规划求解，从而在复杂的任务中找到最优的路径和操作顺序。

## 2、规划语言的两种主流形式

在规划领域，有两种主流的表达语言：**STRIPS** 和 **PDDL**。

(1) **STRIPS**（斯坦福研究所问题解决机）：是较早流行的规划语言，其核心包括**状态、初始状态、目标状态和行为**。**后继状态**是指尚未执行的状态。

**代价函数**用于衡量行为所需的资源消耗。例如，在机器人领域，移动可能消耗电力、时间或机械臂末端工具的寿命，这些都是代价的一部分。

(2) **PDDL**（规划领域描述语言）：将规划问题分为两部分——**域文件**和**问题文件**。

**域文件**定义了操作算子和前提条件，用于精确描述任务规则。例如，在“砖块世界”中，域文件可能包含“移动”和“堆叠”等操作及其前提条件。

**问题文件**则指定了初始状态和目标状态，明确了需要处理的具体情况。例如，初始状态可能是“砖块 A 在位置 1，砖块 B 在位置 2”，而目标状态可能是“砖块 A 堆叠在砖块 B 上”。

**PDDL** 通过**谓词**（前置条件）和**行动**（带参数的操作）来描述状态和行为。例如，机器人移动到某个位置时，需要明确目标位置的参数。

通过将域文件和问题文件结合，我们可以清晰地定义任务的初始状态、目标状态以及实现目标所需的行为和约束。

### 3、机器人规划的核心：如何找到从初始状态到目标状态的路径？

规划的核心问题在于如何从初始状态找到目标状态。通常情况下，规划方法可以分为三大类：

- **向前规划 (Forward Planning)**：从初始状态出发，逐步向目标状态推进。这种方法类似于“从起点到终点”的路径搜索。
- **向后搜索 (Backward Search)**：从目标状态出发，逆向推导到初始状态。这种方法与向前规划完全相反，适用于目标明确但初始状态复杂的情况。
- **启发式搜索 (Heuristic Search)**：利用启发式函数指导搜索方向，既可以向前也可以向后搜索，以提高效率。

#### 搜索策略

在搜索过程中，我们可以采用不同的策略，如广度优先搜索和深度优先搜索，如图 4 所示。

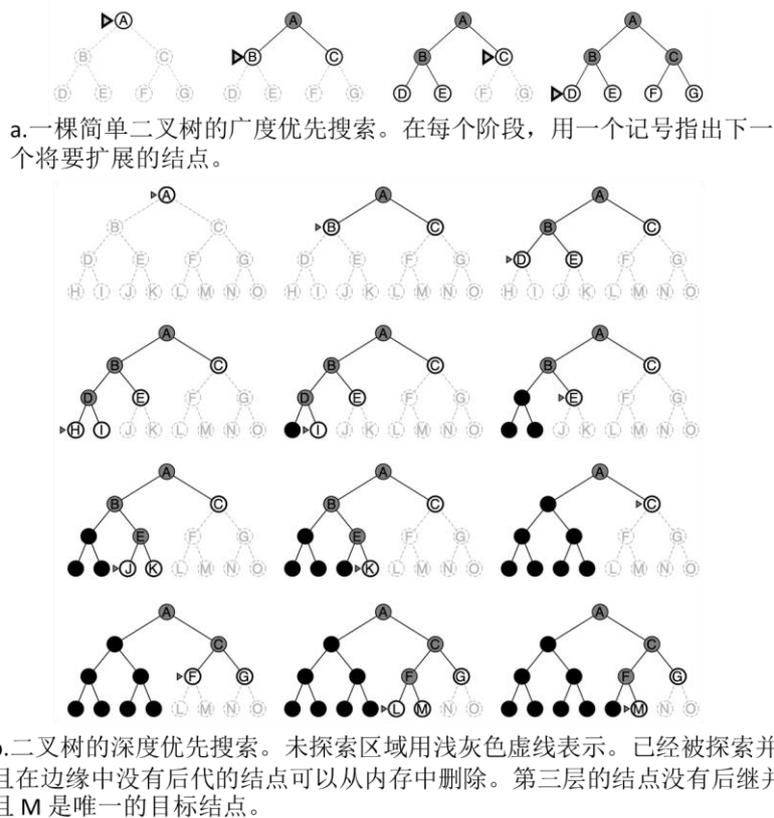


图 4. 广度优先搜索和深度优先搜索

- **广度优先搜索 (BFS)**：从初始状态开始，逐层扩展搜索范围。例如，先搜索 A，然后搜索 B 和 C，依此类推。优点是能够找到最短路径（如果存在），缺点是空间复杂度较高，尤其是在搜索树较宽时。

- **深度优先搜索 (DFS)**：从初始状态开始，沿着一条路径深入搜索，直到无法继续为止，然后回溯并尝试其他路径。优点是空间复杂度较低，缺点是可能会陷入局部最优，且不一定能找到最短路径。

这些搜索策略与数据结构中的树搜索（如二叉树或多叉树）原理一致。如果熟悉数据结构与算法，理解这些概念会更加容易。

### 搜索算法的评价指标

评价搜索算法的优劣通常基于以下几个指标：

- **完整性**：算法是否能在有限时间内找到解（如果解存在）。
- **最优性**：算法是否能找到最优解（如最短路径或最小代价）。
- **时间复杂性**：算法执行所需的时间。
- **空间复杂性**：算法执行所需的内存空间。

图 5 所示是广度优先搜索、深度优先搜索、迭代加深搜索 (ID)、爬山算法 (HC)、A\* 算法、迭代加深 A\* 算法 (IDA\*) 的比较。这些指标与数据结构与算法中的评价标准完全一致。搜索算法的时间复杂性和空间复杂性是衡量其效率的关键因素。

	DFS	BFS	ID	A*	HC	IDA*
<b>完整性</b>	不能	可以	可以	可以	不能	可以
<b>最优性</b>	不能	可以*	可以	可以	不能	可以
<b>时间复杂度</b>	$\infty$	$b^d$	$b^d$	$b^d$	$\infty$	$b^d$
<b>空间复杂度</b>	$b \cdot d$	$b^d$	$b \cdot d$	$b^d$	$b$	$b \cdot d$

- 参数： $d$  是解所在的深度； $b$  是分支因子
- 宽度优先搜索 (BFS) 满足最优性的前提是代价一致。
- A\* / IDA\* 满足最优性的前提是  $h$  具备 **可采纳性 (admissible)**： $h < h^*$

图 5. 几种搜索算法的比较

### 机器人分层任务规划——解决搜索空间的爆炸问题

传统的搜索方法（如一次性搜索）可能会导致**搜索空间爆炸**问题，尤其是在任务复杂度较高时。例如，当任务路径较长或状态空间较大时，计算量会急剧增加，导致搜索效率低下。为了解决搜索空间爆炸问题，可以采用**分层任务规划**

(Hierarchical Task Planning)。这种方法通过将复杂任务分解为多个子任务，逐步细化和优化，从而降低搜索空间的复杂度。

以烹饪机器人为例，如图 6 所示，机器人通过将复杂的烹饪任务分解为多个子任务（如准备食材、加热、搅拌等），并协调多个执行单元（如机械臂、传感器等）来完成整个任务。通过任务分解和分层规划，机器人能够有效减少搜索空间的复杂度，避免计算量过大。



图 6. 以烹饪机器人为例。分层任务网规划和经典规划类似，但增加一个方法集合，让系统将任务分解为更小的子任务（可能有偏序约束），规划过程就是递归的将那些非原子任务分解到原子任务。

通过分层任务规划，我们可以将复杂的任务分解为可管理的子任务，从而有效解决搜索空间爆炸问题，提高任务执行的效率和可行性。

#### 4、规划中的不确定性问题

在规划问题中，除了常见的确定性任务外，还存在不确定性和部分可观察性等问题。这些问题使得规划变得更加复杂，需要采用不同的方法来解决。

- **确定性问题：**在确定性问题中，状态和行为的结果是明确的。例如，拧螺丝时，螺丝的位置和状态是确定的，规划路径和目标状态也是清晰的。

- **不确定性问题：**在不确定性问题中，状态和行为的结果存在概率性。例如，天气预报无法精确预测某一天的具体降雨时间，这种不确定性需要通过概率模型来处理。

- **部分可观察性问题：**在某些情况下，机器人无法完全观察到环境的所有信息。例如，机器人无法直接观察到自身的背部或腹部状态，这种部分可观察性需要通过传感器和推理来解决。

- **目标可达性问题：**目标可达性是指通过一系列行为是否能够达到目标状态。这涉及到行为的选择、传感器的测量以及规划的有效性。

针对不确定性问题，机器人规划通常采用**概率模型**、**马尔可夫决策过程**、**蒙特卡洛法**等方法进行规划。

- **概率模型**：机器人通过传感器采集环境数据，并基于概率模型进行状态估计。例如，使用边缘检测或规则识别算法，机器人可以计算出观察到某个状态的概率，并根据概率进行二次规划。

- **马尔可夫决策过程 (MDP)**：MDP 是一种用于处理不确定性问题的经典方法。它通过计算状态转移的概率，找到从初始状态到目标状态的最优路径。例如，机器人可以通过 MDP 模型评估每个行为的成功概率，并选择最优的行为序列。

- **蒙特卡洛方法**：蒙特卡洛方法通过随机采样来估计状态转移的概率。例如，在机器人控制中，蒙特卡洛方法可以用于采样传感器数据，并通过滤波算法（如卡尔曼滤波）提高状态估计的准确性。

## 5、基于概率的 PDDL 规划

针对不确定性问题，PDDL（规划域定义语言）也提出了基于概率的扩展版本，其核心思想如下：

- **状态转移的概率**：在基于概率的 PDDL 中，每个行为都有一定的成功概率。例如，机器人尝试套接螺丝时，有 95% 的概率成功套接，5% 的概率失败。通过计算这些概率，机器人可以优化规划路径。

- **传感器测量与状态估计**：机器人通过传感器测量当前状态，并根据概率模型判断是否达到目标状态。例如，如果传感器测量结果显示螺丝套接深度达到预期值的 95%，机器人可以认为任务成功，并继续进行下一步规划。

在机器人控制中，概率规划广泛应用于位置控制、力控制等领域，例如：

- **PID 控制与概率采样**：在 PID 控制中，机器人通过传感器采样位置、电压和电流数据，并使用滤波算法（如卡尔曼滤波）提高数据的准确性。然后，基于采样结果，机器人通过 PID 控制器调整行为，实现精确控制。

- **蒙特卡洛定位与导航**：在机器人导航中，蒙特卡洛方法用于定位和路径规划。通过随机采样环境数据，机器人可以估计自身位置，并规划出最优路径。

- **马尔可夫决策过程 (MDP) 的应用:** MDP 模型通过计算状态转移概率, 帮助机器人在不确定环境中做出最优决策。例如, 在机器人抓取任务中, MDP 可以评估每个抓取行为的成功概率, 并选择最优的行为序列。

### **三、基于神经符号的动力电池拆解任务与运动规划**

机器人智能化自主拆解面临的主要困难是非结构化的拆解环境和多种不确定的条件。例如, 汽车制造商为了迎合消费者个性化动力的需求, 动力电池包不仅外观造型各异而且性能配置亦不同, 即使是同一品牌同一型号的新能源汽车动力电池包的配置也往往不同, 导致结构不同、连接约束件不同。此外, 在非结构化的使用环境中动力电池包存在变形、污浊、腐蚀等。而且, 由于消费者使用的频次、时长和习惯不同, 导致动力电池的健康状态和剩余寿命不同, 废旧动力电池包拆解厂商无法批量获得特定批次的动力电池进行拆解。因此, 拆解厂商无法像制造厂商一样, 针对同一批次的电池包借助固定结构化的预编程控制机器人进行大规模的拆解回收。

#### **主要创新与贡献**

为了解决这些问题, 我们的主要创新性贡献是, 把动力电池的智能拆解任务规范为基于神经符号的任务与运动规划问题, 并基于此形成一个感知和控制紧密结合的闭环控制系统。与经典的规划方法类似, 系统用 PDDL 描述每个拆解动作原语, 包括每个动作执行的前提条件, 以及该动作执行完成后对环境的影响。最佳方案搜索模块根据当前的状态和目标状态, 使用逻辑推理的方法自动地设计(搜索)出达到目标的动作规划(即, 动作原语序列)。整个过程具有可解释性, 系统清楚地知道为什么要做某步骤, 以及该步骤的前提条件。与经典规划方法不同的是, 我们引入了神经谓词, 利用神经网络帮助符号系统完成符号接地(symbol grounding)的问题(即, 将连续输入状态, 如机器人位姿或图像等, 映射到规划器生成所需的符号状态), 使该规划方法不局限于符号空间, 无需人工完成复杂的抽象工作, 根据实际物理空间中任务类型自主完成任务规划, 进而控制机器人执行任务。由于神经谓词的输出存在误差, 如果简单设定某个阈值完成映射, 很容易产生无效的状态。所以, 我们将搜索模块进行了相应的扩展, 使用基于概率逻辑的搜索策略。进一步, 我们利用逻辑系统的可解释性对系统进行扩展, 在执行

过程中，机器人根据当前的执行状态，判断出执行的结果，从而收集拆解现场的实时数据，并利用这些数据训练神经谓词，逐渐提高系统的感知能力。

### 实验中的可解释性探讨

我们设计了多组实验，验证系统在不同场景下的适应性和有效性。电池拆解任务对系统的可解释性要求很高。人们希望清楚地了解机器人目前正在做什么，以及为什么执行它以便支持安全的人机协同。我们提出了基于神经符号  $A_i$  的任务与运动规划方法，可以清楚地判断机器人的执行状态和原因。图 7 所示是实验过程中两个任意时刻，图的左边显示机器人的执行状态和传感器数据，右边是系统规划过程中的内部状态。这些状态基本上是符号化的，可以清晰地描述出：**当前状态**，**原计划**，是否需要**更新计划**，以及将要做什么操作（**执行原语**）。

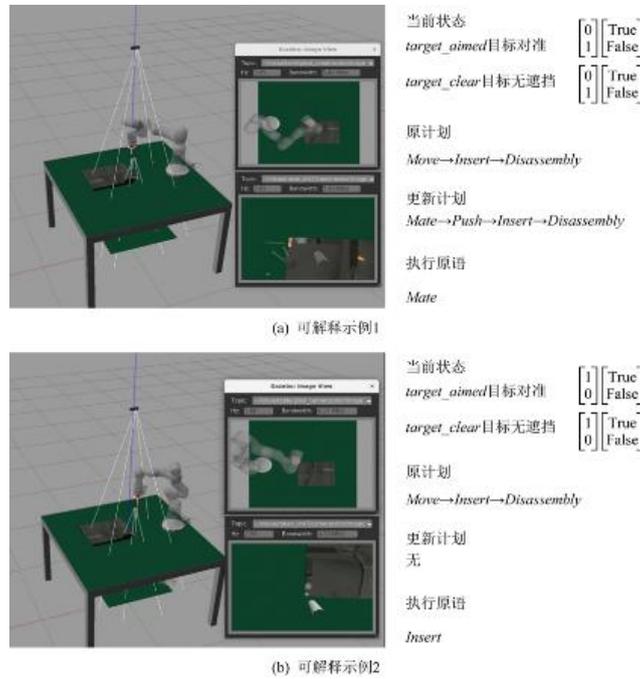


图 7.实验过程中两个任意时刻的可解释性示例

图 7(a)所示，系统完成 *Move* 操作后，发现当前状态与乐观估计值不一致，不但没有对准目标，而且目标附近还有障碍物。于是，基于当前状态重新规划，从 *Move* → *Insert* → *Disassembly* 更新为 *Mate* → *Push* → *Insert* → *Disassembly*，因此下一步系统将执行 *Mate* 操作。图 7(b)所示，系统完成 *Move* 操作后，发现当前状态与乐观估计值一致，无需重新规划，于是按照原有规划，继续执行 *Insert* 操作。通过这两个典型例子不难看出，人们可以通过符号化的表示轻松的理解机器人的行为，这为人机协作完成复杂的拆解任务打下良好的基础。

## 系统优势

基于神经符号 AI 的任务与运动规划系统具有以下四大优势：

- **自主性**。系统能够智能感知环境差别，自主选择并执行运动原语，实现动力电池连接约束件的自主解除；
- **可扩展性**。系统通过添加新的动作原语完成更复杂的任务；
- **可学习性**。系统通过引入神经谓词，实现从连续空间状态到规划器所需的符号状态的映射，是基于数据进行学习获得的，这种方法避免了像传统符号逻辑系统一样要求专业人员提前定义和抽象符号，而且，使得整个系统可以在工作中通过现场学习不断提高精度；
- **可解释性**。任务规划采取符号逻辑驱动，它能够掌握拆解动作执行的原因和结果，因此，人类可以轻松地干预和指导。考虑到系统在可扩展性和可学习性方面的优势，人类的干预和指导可以进一步促进系统的完善和性能的提高。

## 四、结束语

在复杂的非结构化的拆解工作环境下，由于种种不确定性导致解除紧固动力电池零部件的连接关系时拆解任务难以规划，我们提出了基于神经符号人工智能的任务与运动规划方法，让机器人在动态环境下也能够自主规划，完成任务，成功率达到 100%。该方法在自主性、可扩展性、可解释性、可学习性等方面与传统方法相比有明显的优越性。

在未来的工作中，我们将把现有的工作扩展到动力电池的抓取、搬运、分类整理等方面；同时进一步深入研究目前系统中有待提高的学习环节，充分利用神经符号 AI 所带来的可解释，可学习等优势，在拆解现场基于云-边-端系统不断地通过现场学习提升拆解效率和具身技能，最终真正实现的拆解智能化。

**神经符号AI，赋能绿色制造的人工智能引擎**  
**<https://www.nsaihome.org.cn>**



**NSAiHome**  
**神经符号人工智能社区**

(责任编辑 曹晓舟，审核 刘永光)