

Caging in Time: A Framework for Robust Object Manipulation under Uncertainties and Limited Robot Perception

Gaotian Wang^{*1}, Kejia Ren^{*1}, Andrew S. Morgan², and Kaiyu Hang¹

¹Department of Computer Science, Rice University, Houston, TX 77005, USA

²The AI Institute, Cambridge, MA 02142, USA

^{*}Equal Contribution

Abstract

^{*}警告：该PDF由GPT-Academic开源项目调用大语言模型+Latex翻译插件一键生成，版权归原文作者所有。翻译内容可靠性无保障，请仔细鉴别并以原文为准。项目Github地址 https://github.com/binary-husky/gpt_academic/。当前大语言模型: gpt-4o-mini，当前语言模型温度设定: 0。为了防止大语言模型的意外谬误产生扩散影响，禁止移除或修改此警告。

真实世界中的物体操作常常受到物理不确定性和感知限制的挑战。作为一种有效的策略，尽管基于笼罩配置的操作框架已经成功提供了稳健的解决方案，但由于它们对多个机器人的可用性、广泛分布的接触点或机器人与物体的特定几何形状的严格要求，这些框架并不广泛适用。为此，本文提出了一种新概念，称为时间中的笼罩（Caging in Time），**以允许即使只有一个机器人参与任务时也能形成笼罩配置**。这个新概念可以通过一个观察来解释，即使需要笼罩配置来限制物体的运动，笼罩的一小部分在任何时候都是主动操控的。因此，我们可以策略性地切换机器人的配置，这样通过在时间中（in time）收缩其配置，我们就能看到一个笼子形成，并且在需要时其必要部分处于活动状态。我们在具有挑战性的**准静态和动态**操控任务上实例化我们的时间中的笼罩理论，证明时间中的笼罩可以在包括基于几何和基于能量的状态空间中普遍实现。通过广泛的实验，我们展示了在开放回路方式下的稳健和准确操作，而无需详细了解物体的几何形状或物理特性，也不需要操作状态进行实时准确反馈。除了作为一种有效且稳健的开放回路操控解决方案外，所提出的理论还可以作为其他受不确定或有限机器人感知影响的操控系统的补充策略。

1 Introduction

物体操控是机器人参与涉及物理交互任务的基本能力 (Billard and Kragic 2019)。尽管这一领域的研究在数据驱动框架和增强感知系统方面取得了显著进展 (Kaelbling 2020; Yuan et al. 2017; Lee et al. 2019)，但我们仍然看不到很多机器人系统在我们周围稳健地工作。通常观察到两个主要原因导致这一挑战：1) 现实世界的物理不确定性比任何实验室环境复杂得多，常常使某些建模假设失效 (Rodriguez 2021)，以及 2) 感知或传感在现实世界的机器人应用中几乎总是不足够好。建模假设与现实之间的差距、领域变异以及随机遮挡往往恶化感知性能，甚至导致感知系统完全失去对目标物体的追踪 (Bohg et al. 2017)。因此，完全依赖于闭环控制进行机器人操控的方法在实际应用中面临着持续挑战。

与专注于接触的方法不同，基于笼式结构的操控提供了一种新的范式，可以显著降低感知需求。更重要的是，尽管它并不旨在实现精确控制，笼式结构可以通过完全忽视物理不确定性的影响而稳健地工作，如在抓取、手内操控和多机器人协调任务中所见 (Rodriguez et al. 2012; Song et al. 2021; Bircher et al. 2021)。具体而言，笼式结构旨在完全限制目标物体在已知区域（笼子）内所有可能的配置。当笼子移动或变形时，目标

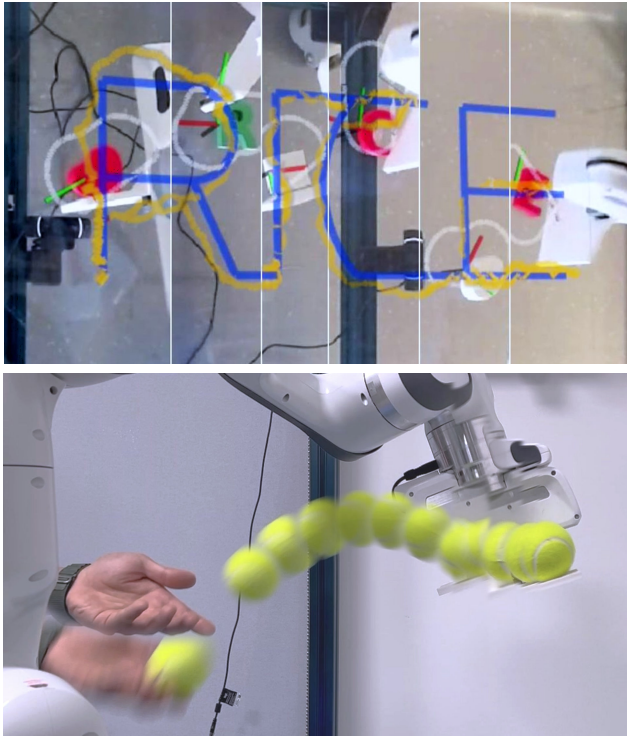


图 1. 示例对象操作任务通过 时间笼裹 (Caging in Time)。上: 物体平面推压以描绘“RICE”。在没有感知反馈的情况下, 未知物体在操作过程中被随机替换。录音在不同时间进行, 并进行拼接以显示所有物体。下: 在没有任何感知反馈的情况下, 平端执行器上的球捕捉。

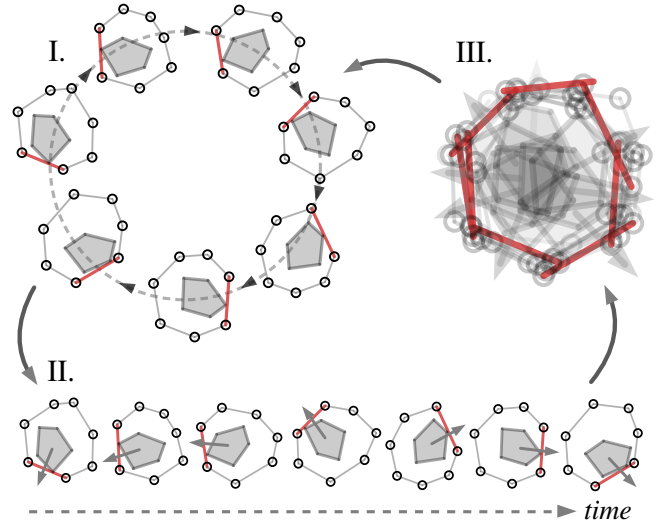


图 2. 时间中的笼子理论通过一个平面推动任务的例子进行可视化。I: 一个由线形条 (灰色) 形成的虚拟笼子, 将一个物体稳健地推送通过一个圆形轨迹。II: 在时间维度上, 我们可以看到一次只有一个条 (红色) 有效接触, 而其他条似乎是多余的。III: 如果我们将有效条 (红色) 的所有配置在时间上进行折叠, 便形成了一个笼子, 可以沿时间展开, 以便只用一个红色条完成I中的任务。

物体被操控, 同时目标物体的配置得以保证跟随笼子以完成任务。然而, 将这一想法应用于一般的操控任务一直是一项挑战, 因为形成笼式结构对硬件有非常严格的要求, 例如需要多个机器人进行协调、广泛分布的接触, 或机器人或物体的特定几何形状以允许此类结构的存在 (Makita and Wan 2018)。

这项工作提出了一种新概念, 称为时间中的围护, 旨在将基于围护配置的操作扩展到更一般的问题, 而不依赖于特定硬件假设。图 1 展示了示例应用。这个新理论的高级思想可以解释如下。在一个极端情况下, 假设我们有一个物体进行操作, 而一个机器人可以在物体的任何地方进行无限次接触。这样, 物体就完全被围护, 任意的操作都可以通过同时移动所有接触点来实现。在另一个极端情况下, 假设一个机器人一次只能与物体接触一次, 但它能够无限快地切换接触到其他位置, 以便实际上可以在物体的各个地方都存在接触。与前一种情况等效, 使用这个虚拟围护也可以实现任意操作。我们对时间中的围护的想法利用了这两种极端之间的可能性: 我们假设机器人一次可以做一个或几个接触, 并且可以根据需要快速切换到其他接触, 从而在时间上形成一个围护。这个想法通过图 2 以一个平面推动任务可视化, 其中一个物体被一个具有多个杆的虚拟围护 (灰色) 通过圆形轨迹推动。请注意, 在物理和时间上, 完成任务只需要一个杆 (红色)。此外, 通过在时间上折叠有效杆的配置, 可以形成一个围护, 并能在时间上展开以完成任务, 仿佛一个完整的围护一直存在。

我们在具有真实机器人的准静态和动态操作任务上实例化了时间中的围护理论。值得注意的是, 示例实例化表明, 时间中的围护可以应用于一般的状态空间, 包括基于几何和基于能量的空间。在没有任何感知反馈的情况下, 时间中的围护在所有实验任务上都显示出保障的任务成功, 即使当操作受到任务扰动和未知物体形状变化的物理影响时。与基线闭环控制方法相比, 我们展示了我们的框架同样准确, 同时在感知不确定性方面显著更具鲁棒性, 因为零依赖于精确的感知反馈。贡献: 提出的时间中的围护概念做出了三项理论贡献: 1) 提供了一种稳健机器人操作的规划范式, 能够完全消除感知不确定性的影响; 2) 将传统上范围较窄的基于围护配置的操作扩展到更一般的操作框架, 利用战略性连续的机器人运动; 3) 首次提供无需感知反

馈即可精确控制机器人操作的可能性,以支持在感知不可靠的场景下进行稳健操作,例如,现实世界中的物体跟踪通常是嘈杂的,并且由于遮挡会时不时地丢失跟踪。局限性:本文报告的工作是朝着时间中的围护概念的一般应用迈出的第一步。由于强调理论基础的推导,这项工作并未开发针对一般操作问题的算法,通过时间中的围护技能进行处理。具体来说,我们在提出理论的细节之前,识别出本工作当前范围的以下主要局限性:1)作为所提理论的示例实例化,报告的操作算法仅设计用于在特定任务上实现时间中的围护;2)为了明确验证理论,操作物理或动力学模型是被分析推导的,尽管仿真或基于学习的模型可能更高效。尽管如此,我们强调所提出的时间中的围护框架对于一般操作问题是一个完整的理论,如第4节所讨论的。

2 Related Works

机器人操控:接触特性、物体的几何和物理属性,以及完美的物体感知,通常被假设为足够提供用于建模机器人操控系统 (Suomalainen et al. 2022; Bütepage et al. 2019)。作为替代,基于学习的方法假设用于训练操控模型的数据能够充分覆盖相关任务领域中的所有潜在变化,并且任务中使用的感知系统与训练集中的感知系统相同或相似,例如,相似的相机姿态 (Lee et al. 2019; Kaelbling 2020; Andrychowicz et al. 2020; Kroemer et al. 2021)。在其他方面,交互感知显著降低了操控系统对某些先前知识和直接感知的要求 (Bohg et al. 2017; Hang et al. 2021),同时仍然假设通过某些通道可为机器人提供可靠的感知,能够反复估计操控系统的状态。尽管许多操控方法表现优异,但在现实世界中总存在不可靠感知的挑战,例如物体跟踪中的噪声或物体遮挡,这将显著影响大多数操控系统的鲁棒性。与大多数现有工作不同,本研究旨在使用时间维度上的围栏配置来消除对不可靠感知的依赖,因此消除其负面影响。请注意,本工作的目标并不是用时间中的围栏替代其他操控规划或控制方法。相反,我们旨在提供一种替代或补充方法,与其他方法一起,提高在不可靠感知下的机器人操控的鲁棒性。

基于围栏配置的操控:围栏配置最初被提议作为一个工具,供SE(2)中的多机器人团队来约束目标物体的运动,以便机器人团队的运动将决定被围物体的运动 (Pereira et al. 2004; Sudsang and Ponce 2000)。这一概念后来扩展到2D中的机器人抓取问题,其目标是将目标物体固定在一个小区域内 (Rodriguez et al. 2012)。通过进一步建模机器人与物体之间的拓扑关系,基于拓扑空间的围栏操作得以发展 (Stork et al. 2013)。最近,一种特定类型的平面灵巧操控通过手内围栏配置得以实现,使得物体通过在时间上变形的控制能量状态来进行操控 (Bircher et al. 2021)。在一个移动机器人积极尝试逃离其他机器人团队的情况下,围栏配置也使得基于队形的运动规划成为可能,以将这样的机器人聚集到目标区域 (Song et al. 2021)。本质上,所有现有框架首先旨在构建一个围栏以围住目标物体,然后在移动或变形围栏的同时保持该围栏,以实现操控任务。最近,基于围栏配置的操控已扩展到结合机器人运动,以放宽机器人或物体的特定几何属性的要求 (Dong and Pokorny 2024; Dong et al. 2024)。这一研究方向专注于设计度量来预测操控的鲁棒性,而自主围栏规划和控制则留作待解的问题。

传统围栏的挑战:传统围栏的最根本限制是需要多个机器人 (ZhiDong and Kumar 2002)、广泛分布的接触点 (Rodriguez et al. 2012),或机器人的特定几何形状 (Varava et al. 2016),这使得围栏作为一种概念不适用于更一般的任务设置。在本研究中,借助提出的时间中的围栏框架,我们展示了即使只有一个机器人,我们也能在时间上虚拟构建围栏以满足上述要求。此外,传统方法需要非常复杂的算法来验证围栏配置 (Varava et al. 2021),而对于部分围栏,这可能更为困难 (Makita and Nagata 2015)。正如将在时间中的围栏中看到的,由于理论上我们可以随需虚拟拥有我们的机器人或接触点,因此围栏验证通常是一个更简单的问题,因为我们只需确保物体不穿透预定义的围栏边界。

3 Preliminaries

在本节中,我们首先在 Sec. 3.1 中引入传统栅栏定义中的符号,然后在 Sec. 3.2 中将这些符号扩展到更一般的任务空间,以便推导我们提出的 时间中的栅栏 (Caging in Time) 框架。

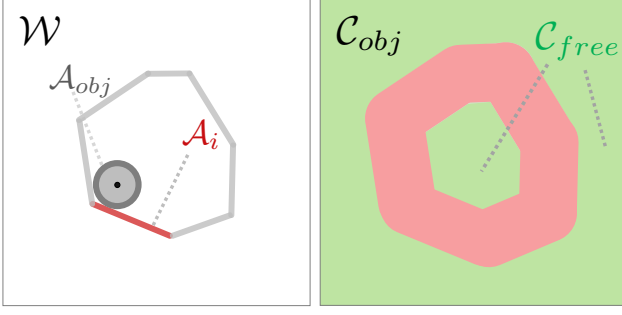


图 3. 一个由七个线性机器人围绕的圆形物体的工作空间 $\mathcal{W}$ (左) 和 C-空间 $\mathcal{C}_{obj}$ (右) 的示例。左: 红线是第 i 个机器人 $\mathcal{A}_i$ 的几何形状, 实心灰色圆圈是物体 $\mathcal{A}_{obj}$ 的几何形状。右: 红色区域是物体与机器人碰撞的所有配置, 绿色区域是物体的自由 C-空间 $\mathcal{C}_{free} \subset \mathcal{C}_{obj}$ 。

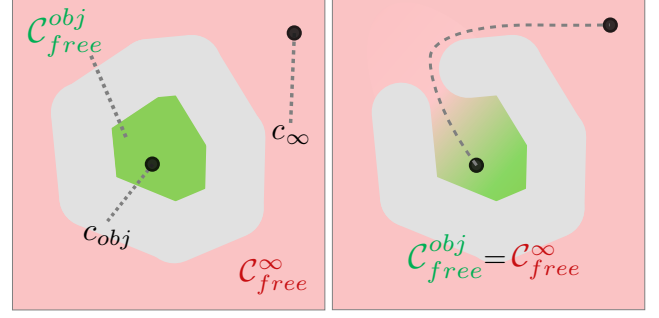


图 4. 对传统笼罩条件的说明。左: 绿色区域和红色区域分别代表 $\mathcal{C}_{free}^{obj}$ 和 $\mathcal{C}_{free}^{\infty}$ 。在这种情况下, 物体被笼罩, 因为 $\mathcal{C}_{free}^{obj}$ 和 $\mathcal{C}_{free}^{\infty}$ 不连通。右: 非笼罩场景, 其中 $\mathcal{C}_{free}^{obj}$ 和 $\mathcal{C}_{free}^{\infty}$ 变得连通。物体可以沿着虚线所示的轨迹逃脱。

3.1 Traditional Caging as Object Closure

在多个机器人围住物体的场景中, 我们将物体的配置空间 (C-space) 表示为 $\mathcal{C}_{obj}$, 物体的配置 (例如, 位置和方向) 表示为 $c_{obj} \in \mathcal{C}_{obj}$ 。第 i 个机器人的 C-space 和配置分别用 $\mathcal{C}_i$ 和 $c_i \in \mathcal{C}_i$ 表示。物体和所有机器人共享一个公共工作空间 $\mathcal{W}$, 其中 $\mathcal{A}_{obj}, \mathcal{A}_i \subset \mathcal{W}$ 是物体和第 i 个机器人在工作空间中的几何形状。因此, 物体的自由 C-space $\mathcal{C}_{free} \subset \mathcal{C}_{obj}$ 定义为物体与任何机器人不发生碰撞的所有可能状态的集合:

$$\mathcal{C}_{free} := \{c_{obj} \in \mathcal{C}_{obj} \mid \forall i : \mathcal{A}_{obj}(c_{obj}) \cap \mathcal{A}_i(c_i) = \emptyset\} \quad (1)$$

上述符号的说明见图 3。

传统上对囚禁的定义, 最初称为对象闭合 (object closure), 由 ZhiDong and Kumar (2002) 首次提出, 作为一个物体被机器人困住的条件。换句话说, 当物体无法从其当前配置移动到一个无限远处的配置时, 满足囚禁条件。

如图 4 所示, 给定一个与当前的 c_{obj} 在 C-空间中无限远的点 $c_{\infty} \in \mathcal{C}_{obj}$, 定义两个不同的集合 $\mathcal{C}_{free}^{obj}, \mathcal{C}_{free}^{\infty} \subset \mathcal{C}_{free}$ 如下。 $\mathcal{C}_{free}^{obj}$ 是 $\mathcal{C}_{free}$ 的最大子集, 在该子集中, 每个配置 c 都可以通过无碰撞路径从当前配置 c_{obj} 到达; 类似地, $\mathcal{C}_{free}^{\infty}$ 是 $\mathcal{C}_{free}$ 的最大子集, 在该子集中, 每个 c 都可以到达 c_{∞} :

$$\begin{aligned} \mathcal{C}_{free}^{obj} &= \{c \in \mathcal{C}_{free} \mid \text{connected}(c, c_{obj})\}, \\ \mathcal{C}_{free}^{\infty} &= \{c \in \mathcal{C}_{free} \mid \text{connected}(c, c_{\infty})\} \end{aligned} \quad (2)$$

满足囚禁条件当且仅当:

$$\begin{cases} \mathcal{C}_{free}^{obj} \neq \emptyset \\ \mathcal{C}_{free}^{obj} \cap \mathcal{C}_{free}^{\infty} = \emptyset \end{cases} \quad (3)$$

在这里, 第一个标准确保了笼式配置的存在, 而第二个条件保证没有可行路径连接 $\mathcal{C}_{free}^{obj}$ 和 $\mathcal{C}_{free}^{\infty}$ 。这种传统的笼式概念作为我们提出的 时间笼式 框架的基础。

3.2 Generalized Representations

为了能够推导出时间笼罩理论, 我们现在将从配置空间到状态空间的一般化表征 (representation) 和符号。这允许对操作任务进行更全面的描述, 并适应各种动态场景和不确定性, 而不是传统的基于准静态配置的笼罩定义。

我们用 $\mathcal{S}_{obj}$ 表示一个对象的状态空间, 它包含比 $\mathcal{C}_{obj}$ 更广泛的属性。对象在时间 t 的状态用 $\mathbf{q}_t \in \mathcal{S}_{obj}$ 表示。根据具体的操作任务, $\mathbf{q}_t$ 可能不仅包括配置 (configuration, 例如位置和方向), 还可能包括速度、加速度或其他相关属性。

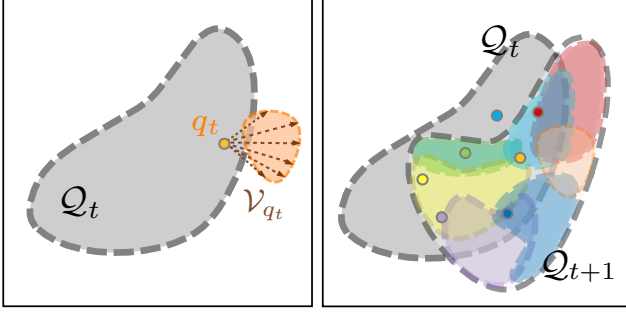


图 5. 一个 PSS 传播的示意图。左: 对于单个状态 $q_t \in Q_t \subset \mathcal{S}_{obj}$ 的传播。通过一组可能的动作 $\mathcal{V}_{q_t}$, q_t 可以传播到一组不同的状态 (橙色区域)。右: 从整个 PSS Q_t 传播到 Q_{t+1} , 通过传播 Q_t 中的所有点。 Q_{t+1} 中不同颜色的区域说明了来自多个不同的 $q_t \in Q_t$ 的传播。

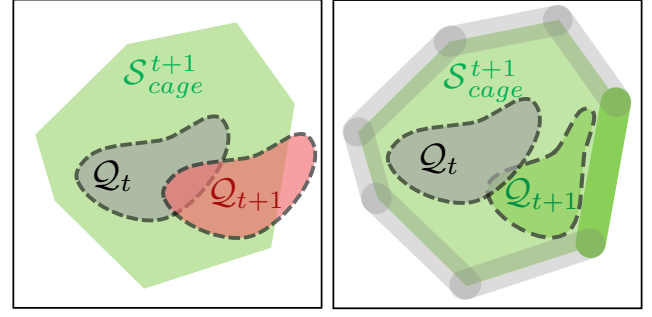


图 6. 一个二维推力示例, 展示了机器人动作如何影响 PSS 传播。左: 在没有机器人干预的情况下, Q_t 传播到 Q_{t+1} 的场景。 Q_{t+1} 将超过笼区域 $\mathcal{S}_{cage}^{t+1}$ 。右: 通过从绿色条所示姿势出发, 朝着 $\mathcal{S}_{cage}^{t+1}$ 中心的精心选择的机器人推送, Q_{t+1} 将变形为不同的形状, 并保持 $\mathcal{S}_{cage}^{t+1}$ 内部。在这个例子中, 工作空间和物体的状态空间 (即物体的二维位置) 共享同一空间 $\mathbb{R}^2$, 因此我们将机器人推力 (灰色条) 叠加到物体的状态空间中以便于可视化。

此外, 现实系统中, 时间 t 时对象的状态通常由于感知不确定性或建模简化而不是完全确定的。为了考虑这些不确定性, 我们定义对象的潜在状态集 (PSS) 为在时间 t 时所有可能状态的集合, 用 $Q_t \subset \mathcal{S}_{obj}$ 表示。

4 Caging in Time

在本节中, 我们将正式介绍我们提出的时间围捕理论的定义。通过仅与对象互动的一台机器人, 时间围捕验证对象的状态在操作过程中保持被围捕状态, 从而实现开环操作并保证被操作对象的期望运动。作为我们提出的理论的一个重要组成部分, 我们需要预测对象的受限运动, 这涉及到随时间传播 PSS, 将在第 4.1 节中介绍。接着, 我们将在第 4.2 节中正式定义时间围捕。

为了对提出的概念有更直观的说明, 本节中的图形是在二维空间中绘制的。然而, 重要的是强调 q_t 可以根据具体操作任务的要求采取任何形式和维度。选择二维可视化是为了视觉上的简单性, 并不限制状态空间概念的普遍性。

4.1 Propagation of PSS

在时刻 t , 对于 PSS 中对象的每一个特定状态 (state) $q_t \in Q_t$, 我们用 $\mathcal{V}_{q_t}$ 表示所有允许的运动的集合, 这些运动可以转移对象的状态。我们定义记号 $T_x \mathcal{M}$ 来表示流形 $\mathcal{M}$ 在点 $x \in \mathcal{M}$ 处的切空间。注意到 $\mathcal{V}_{q_t} \subset T_{q_t} \mathcal{S}_{obj}$, 即 $\mathcal{V}_{q_t}$ 位于对象的状态空间 (state space) 在状态 q_t 处的切空间。此外, 基于切丛理论 (Jost 2008), 我们定义时刻 t 的运动丛为公式 (4) 中的集合, 该集合由对象的所有可能的状态 (state)-运动对组成:

$$\mathcal{QV}_t := \{(q_t, v_t) \mid q_t \in Q_t, v_t \in \mathcal{V}_{q_t}\} \quad (4)$$

我们需要定义一个传播函数 $\pi: \mathcal{QV}_t \mapsto \mathcal{S}_{obj}$ 以预测给定可能的状态 (state) q_t 和物体的运动 v_t 时, 物体的状态 (state) q_{t+1} 在下一个时间步骤的值, 即 $q_{t+1} = \pi(q_t, v_t)$ 。

然而, 上述传播函数仅预测下一时间步的单一状态 (state)。在我们的时间中的笼养 (Caging in Time) 框架中, 我们需要知道从先前步骤传播的所有可能状态 (states), 即 PSS Q_{t+1} 。为此, 我们将传播函数 π 扩展为另一函数 $\Pi: 2^{\mathcal{QV}_t} \mapsto 2^{\mathcal{S}_{obj}}$, 其定义域为运动束 $\mathcal{QV}_t$ 的幂集。如图 5 所示, 它用于通过传播每个可能的 q_t 来获得时间 $t+1$ 时所有可能状态 (states) 的整个集合:

$$\begin{aligned}\mathcal{Q}_{t+1} &= \Pi(\mathcal{QV}_t) \\ &= \{\mathbf{q}_{t+1} = \pi(\mathbf{q}_t, \mathbf{v}_t) \mid (\mathbf{q}_t, \mathbf{v}_t) \in \mathcal{QV}_t\}\end{aligned}\quad (5)$$

4.2 Caging in Time

与在节 3.1 中介绍的传统围栏不同，我们的时间围栏框架只需要一个机器人来操纵物体。这个围栏是通过将单个机器人切换到不同的配置，并在每个时间步与物体以不同方式互动而逐渐形成的。图 6 展示了一个示例，其中机器人推动器可以从相对于物体的多种可能初始位置（在右侧图中由灰色条标记）推动物体。在每个时间步，通过预测物体的可能运动（使用在节 4.1 中定义的传播函数 Π ），机器人需要确定选择哪个候选推动（例如，图 6 右侧图中的绿色条），以防止物体逃逸。因此，随着机器人推动器在不同位置之间切换并以不同方向推动物体，物体始终可以被限制在一个有界区域内。

具体来说，对于每个时间步 t ，我们希望将物体的状态（state）始终限制在其状态空间中的区域 $\mathcal{S}_{cage}^t \subset \mathcal{S}_{obj}$ 内，我们将该区域称为“围栏区域”。“围栏区域” $\mathcal{S}_{cage}^t$ 不必是静态的；它可以根据所涉及的操纵任务而变形和位移。此外，需要根据操纵任务的上下文预先定义一组候选机器人动作 $\mathcal{U}$ 。例如，在图 6 中示出的平面推物操纵场景中， $\mathcal{U}$ 可以由机器人推动器的不同起始位置和推动方向定义的一组机器人推动动作。这样的机器人动作会影响物体的可能运动。例如，当机器人通过接触与物体互动时，物体将无法向穿透机器人的方向移动。换句话说，对于特定的物体状态（state） $\mathbf{q}_t \in \mathcal{S}_{obj}$ ，允许的物体运动 $\mathcal{V}_{\mathbf{q}_t}$ 将受到机器人动作 $u \in \mathcal{U}$ 的影响。我们使用一个函数 U 来表示这种影响通过

$$\mathcal{V}_{\mathbf{q}_t} = U(\mathbf{q}_t, u) \subset \mathbf{T}_{\mathbf{q}_t} \mathcal{S}_{obj} \quad (6)$$

应用机器人动作 u 可能导致物体在下一个时间步骤向笼区域 $\mathcal{S}_{cage}^{t+1}$ 外部的方向移动。为了保证物体的状态（state）始终位于 $\mathcal{S}_{cage}^t$ 内，对于每个时间步骤 $t = 1, \dots, T$ ，必须存在某个机器人动作 $u_t \in \mathcal{U}$ ，使得所有可能的物体状态（states）在执行该机器人动作后包含在 $\mathcal{S}_{cage}^{t+1}$ 内。因此，时间中的笼控（Caging in Time）的定义正式如下。

Definition 4.1. For a time-varying cage $\mathcal{S}_{cage}^t$ in an object's state space where $t = 1, \dots, T$, Caging in Time is achieved if and only if: $\forall t = 0, 1, \dots, T-1, \exists u_t \in \mathcal{U}$ such that

$$\begin{aligned}\mathcal{QV}_t &= \{(\mathbf{q}_t, \mathbf{v}_t) \mid \mathbf{q}_t \in \mathcal{Q}_t, \mathbf{v}_t \in U(\mathbf{q}_t, u_t)\}, \\ \mathcal{Q}_{t+1} &= \Pi(\mathcal{QV}_t) \subset \mathcal{S}_{cage}^{t+1}\end{aligned}\quad (7)$$

值得注意的是，时间笼困并不能保证存在这样的机器人动作 $\{u_t\}_{t=0}^{T-1}$ 来笼住对象。然而，如果一系列机器人动作 $\{u_t\}_{t=0}^{T-1}$ 被验证满足时间笼困条件，则可以保证对象的状态始终位于随时间变化的笼子 $\mathcal{S}_{cage}^t$ 内。

判断是否满足时间笼困的过程在 Alg. 1 中有详细说明。当时间笼困条件满足时，确保通过某一序列的机器人动作以开环方式将对象困在一个笼子区域 $\mathcal{C}_{cage}^t$ 内。如果我们应用这一系列的机器人动作，尽管没有任何感知反馈，对象也将永远不会逃逸。如前所述，笼子区域 $\mathcal{S}_{cage}^t$ 不必是静态的，它可以随时间变化。例如，如果 $\mathcal{S}_{cage}^t$ 随时间移动并跟随对象的状态空间中的某一轨迹，则时间笼困的对象将能够跟随 $\mathcal{S}_{cage}^t$ 的相同轨迹。

动作可行性验证：在实际操作中，由于机器人的速度限制或可达性，机器人动作 u_t 可能无法实现，这在不同的机器人平台上会有所不同。例如，在连续动作 u_{t-1} 与 u_t 之间的执行过程中，如果没有某些假设（如准静态运动），对象可能会继续移动如果没有。这要求机器人快速切换到下一动作执行，这可能超过硬件的速度限制。为此，在 Alg. 1 的第 3 行，为了在现实条件下实现时间笼困，必须考虑实际硬件的能力来验证动作 u_t 的可行性。

Algorithm 1: The Verification of *Caging in Time***Input:** The time-varying cage region $\mathcal{S}_{cage}^t \subset \mathcal{S}_{obj}$, a sequence of robot actions $\{u_t \in \mathcal{U}\}$ **Output:** The object is caged in time or not

```

1  $\mathcal{Q}_0 \leftarrow$  initial PSS of the object
2 for  $t = 0, 1, \dots, T - 1$  do
3   if not FEASIBLE( $u_t$ ) then
4     return false
5   end
6    $\mathcal{QV}_t \leftarrow \{(\mathbf{q}_t, \mathbf{v}_t) \mid \mathbf{q}_t \in \mathcal{Q}_t, \mathbf{v}_t \in U(\mathbf{q}_t, u_t)\}$ 
7   ▷ Construct the Motion Bundle
8    $\mathcal{Q}_{t+1} \leftarrow \Pi(\mathcal{QV}_t)$ 
9   ▷ Propagation of PSS via  $\Pi$ 
10  if  $\mathcal{Q}_{t+1} \not\subset \mathcal{S}_{cage}^{t+1}$  then
11    return false
12  end
13 return true

```

5 Quasi-static Tasks

在本节中，我们在一个准静态平面推力问题上实例化了时间笼(Caging in Time)理论。为了便于实例化，我们还开发了相关工具，以便在二维空间中传播物体的状态(状态)(state)并生成开放环路的机器人推力动作来对物体进行笼式控制。通过将时间笼框架应用于平面推力问题，我们可以使单个机器人以开放环路的方式推动一个形状未知的物体沿特定轨迹移动，而无需感知反馈、精确的几何信息或物体的任何其他物理属性。这是在基于几何的状态(状态)(state)空间中实例化的时间笼。

5.1 Problem Statement

机器人被任务推动一个二维平面上的物体。为了在不同形状的物体上进行概括而不需要精确的几何建模，我们用半径为 r 的简化边界圆来表示物体的几何形状，即 $\mathcal{A}_{obj}(\mathbf{q}) = \{\mathbf{p} \in \mathbb{R}^2 \mid \|\mathbf{p} - \mathbf{q}\| \leq r\}$ ，其中 $\mathbf{q}$ 表示物体的位置。半径 r 的选择基于物体的大小，确保边界圆覆盖物体的实际形状。机器人配备了一个固定长度的线性推动器，在二维空间中由线段 $\mathcal{L}$ 表示。当 $\mathcal{L}$ 与物体的边界圆相交时，物体被认为与线性推动器发生了碰撞，即从 $\mathbf{q}$ 到线段 $\mathcal{L}$ 的距离不大于半径 r 。

虽然一个准静态二维物体的配置空间通常是 $SE(2)$ ，要求物体的位置和方向，但我们使用边界圆使物体的状态 (state) 空间简化为仅其位置： $\mathcal{S}_{obj} \subset \mathbb{R}^2$ 。因此，物体的状态变为 $\mathbf{q}_t = (x_t, y_t) \in \mathcal{S}_{obj}$ 。因此，PSS $\mathcal{Q}_t$ 和捕获区 $\mathcal{S}_{cage}^t$ 都可以被概念化为这个二维平面内的区域。在这个特定的平面推动任务中，我们进一步定义潜在物体区域 (Potential Object Area, POA) $\mathcal{P}_t \subset \mathbb{R}^2$ 为由工作空间中可能被物体的几何形状占据的所有点组成的集合：

$$\mathcal{P}_t := \bigcup_{\mathbf{q}_t \in \mathcal{Q}_t} \mathcal{A}_{obj}(\mathbf{q}_t) \quad (8)$$

“笼区域” $\mathcal{S}_{cage}^t \subset \mathcal{S}_{obj}$ 为了围住对象的几何覆盖 $\mathcal{P}_t$ 而给出。具体而言，在每个时刻 t ，我们要求对象的状态 (state) 始终被围住，使得对象的几何形状 (即边界圆) 在该状态下，位于一个手动设置的圆 $\mathcal{S}_A^t$ 内。圆 $\mathcal{S}_A^t$ 的中心位于 (x_t^c, y_t^c) ，半径为常数 R 。因此，对象状态空间中的“笼区域” $\mathcal{S}_{cage}^t$ 也是一个与 $\mathcal{S}_A^t$ 共享同一中心的圆，且半径为 $R - r$ ，即， $\mathcal{S}_{cage}^t = \{(x, y) \in \mathbb{R}^2 \mid (x - x_t^c)^2 + (y - y_t^c)^2 \leq (R - r)^2\}$ 。 $\mathcal{S}_{cage}^t$ 的半径 $R - r$ 被称为笼的大小。

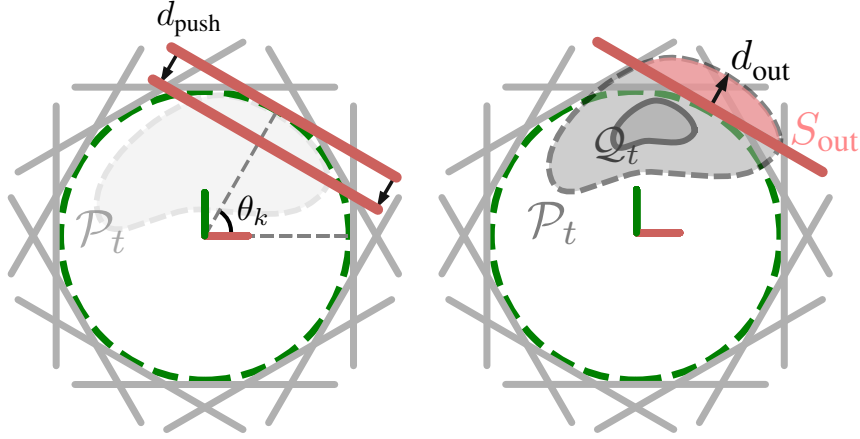


图 7. 机器人动作的表征 (representation) 和平面推送的启发式评估。左: 绿色虚线圆圈是 S_A^t 。每个 $K = 12$ 候选动作 (grey bars) 由一个角度 θ_k 表征 (characterized)。机器人将线性推杆 $\mathcal{L}$ (红色) 放置于 θ_k 并朝圆心推动, 距离为 d_{push} 。右: 启发式评估, 其中 S_{out} (粉色区域) 是在推杆后面的 P_t (POA) 的一部分, d_{out} (黑色箭头) 是到 P_t 边界的最大距离。

Algorithm 2: Pushing by Caging in Time

Input: a trajectory as a sequence of waypoints $\mathcal{T} = \{(x_1^d, y_1^d), \dots, (x_t^d, y_t^d), \dots, (x_T^d, y_T^d)\}$

Output: The task finished with success or not

```

1  $\mathcal{Q}_0 \leftarrow \{q_0\}$  ▷ Observe the Object's Position
2 for  $t = 0, \dots, T - 1$  do
3    $(x_{t+1}^c, y_{t+1}^c) \leftarrow (x_{t+1}^d, y_{t+1}^d)$  ▷ Center of Cage
4    $\mathcal{S}_{cage}^{t+1} = \{(x, y) \mid (x - x_{t+1}^c)^2 + (y - y_{t+1}^c)^2 \leq (R - r)^2\}$ 
5   ▷ Construct the Circular Cage
6    $u_t \leftarrow \text{FINDPUSH}(\mathcal{Q}_t, \mathcal{U}, \mathcal{S}_{cage}^{t+1})$  ▷ Alg. 4
7    $\mathcal{Q}_{t+1} \leftarrow \text{PROPAGATE}(\mathcal{Q}_t, u_t)$  ▷ Alg. 3
8   if  $\mathcal{Q}_{t+1} \not\subset \mathcal{S}_{cage}^{t+1}$  then
9     return false
10  end
11 end
12 return true

```

机器人动作由一个标量角度 θ 来表征, 如图 7 左侧所示。机器人首先将线性推动器的中心放置在由 θ 确定的起始位置。该起始位置始终位于圆 S_A^t 的边界上, 可以通过 $(x_t^c + R \cos \theta, y_t^c + R \sin \theta)$ 计算得到。线性推动器将被定位为切线于 S_A^t 的边界圆。然后, 机器人将推动器朝向 S_A^t 的中心移动固定距离 d_{push} 以推动物体。我们假设有 K 个候选机器人动作, 即 $\mathcal{U} = \{\theta_k\}_{k=1}^K$ 。每个动作的 value 通过 $\theta_k = 2k\pi/K$ 获得, 使得这 K 个候选动作的起始位置均匀围绕 S_A^t 。

如第 4 节所定义, 我们要求物体的状态 (state) 始终被限制在圆形笼区 $\mathcal{S}_{cage}^t$ 之内, 即物体的 PSS (PSS) 为 $\mathcal{Q}_t \subset \mathcal{S}_{cage}^t$ 。然后, 为了使物体沿着某个轨迹 $\mathcal{T} = \{(x_1^d, y_1^d), \dots, (x_t^d, y_t^d), \dots, (x_T^d, y_T^d)\}$ 运动, 该轨迹由物体的状态空间 (state space) 中的一系列轨迹点 (即所需位置) 表示, 我们只需沿相同轨迹移动 $\mathcal{S}_{cage}^t$ 的中心, 即 (x_t^c, y_t^c) 。由于物体始终被限制在移动的 $\mathcal{S}_{cage}^t$ 内, 对于每个时间步 $t = 1, \dots, T$, 物体的位置 (position) 被保证跟随轨迹 $\mathcal{T}$, 位置误差不大于笼子大小 $R - r$ 。

接下来, 我们基于提出的时间笼子 (Caging in Time) 理论开发了一种平面推动问题的算法。在算法 2 中, 详细描述了所开发算法的高级描述。在每个时间步 t , 将找到一个开放循环机器人动作 $u_t \in \mathcal{U}$ 来在时间中笼住对象, 具体将在 Sec. 5.3 中详细讨论; 然后, 对象的状态将根据机器人动作 u_t 进行传播, 相关内容将

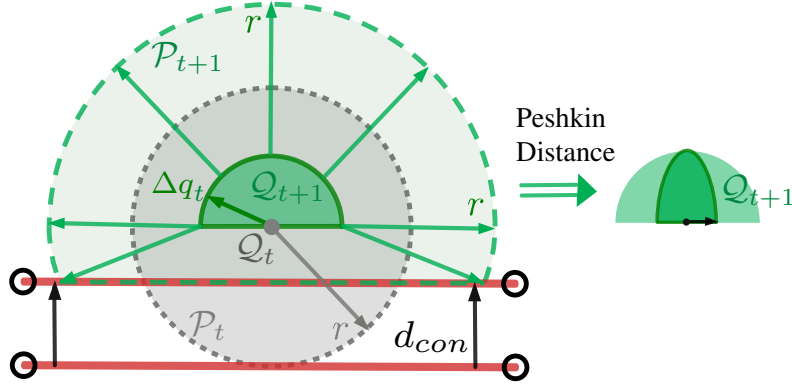


图 8. 对象位移界限的说明。对象的最大位移等于 d_{con} 。因此，包含唯一中央灰色点的 Q_t 可以传播到一个半圆形 Q_{t+1} （深绿色区域的半圆形），其半径等于 d_{con} （即， $\|\Delta q_t\| = d_{con}$ ）。相应地，半径为 r 的 POA P_t （灰色边界圆）将演变为更大的半透明绿色区域 P_{t+1} 。基于佩什金距离（Peshkin Distance）的分析将进一步限制对象的可能位移，使得 Q_{t+1} 变为一个半椭圆，如右侧所示。

在 Sec. 5.2 中讨论。如果传播的 PSS 始终包含在笼子区域 S_{cage}^t 中，则被推动的对象将沿着运动笼子的期望轨迹移动；否则，object 可能位于笼子外部，表示任务失败。

5.2 Unknown Object Shape and Its Motion

对于平面推动，我们通过对象在相邻时间步骤之间的位置的位移来表示对象的运动，记作 $\mathbf{v}_t = \Delta \mathbf{q}_t = (\Delta x_t, \Delta y_t)$ 。然后，在第 4.1 节中最初定义的传播函数 π ，以单个对象的配置和运动作为输入，变为 $\pi(\mathbf{q}_t, \mathbf{v}_t) = \pi(\mathbf{q}_t, \Delta \mathbf{q}_t) = \mathbf{q}_t + \Delta \mathbf{q}_t$ ，用于评估在下一步骤传播的对象的配置 q_{t+1} 。

对于特定对象的配置 $\mathbf{q}_t$ ，所有可能的动作 $\mathcal{V}_{\mathbf{q}_t}$ 通过对象与机器人的线性推动器 $\mathcal{L}$ 之间的交互进行建模，当动作 u_t 被执行时，表示为公式 U 中定义的函数。线性推动器在执行 u_t 的过程中以直线距离 d_{push} 接近对象。如果对象的配置与线性推动器之间的距离 $\mathbf{d}(\mathbf{q}_t, \mathcal{L})$ 超过 $r + d_{push}$ ，则对象的几何形状（即，包围圆）将不会与线性推动器碰撞。对象将不会移动，因此 $\mathcal{V}_{\mathbf{q}_t} = \{(0, 0)\}$ 。然而，如果对象的包围圆与线性推动器碰撞，这表明对象可能会被线性推动器推移，我们记 $d_{con} \leq d_{push}$ 为线性推动器在接触对象的包围圆后移动的距离。如图 8 所示，我们可以使用摩擦理论和 Peshkin 距离 Peshkin and Sanderson (1988) 推导出该位移的界限如下：

$$\begin{aligned} \mathcal{V}_{\mathbf{q}_t} &= U(\mathbf{q}_t, u_t) = \\ &\text{if } \mathbf{d}(\mathbf{q}_t, \mathcal{L}) > r + d_{push} \text{ or } u_t \text{ is None:} \\ &\quad \{(0, 0)\} \\ &\text{otherwise:} \\ &\quad \left\{ \mathbf{v}_t = R(u_t) \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \mid \frac{(\Delta x)^2}{d_{con}^2} + \frac{(\Delta y)^2}{\left(\frac{d_{con}}{2}\right)^2} \leq 1, \Delta x \leq 0 \right\} \end{aligned} \quad (9)$$

在这里， $R(u_t) \in SO(2)$ 是角度 u_t 的旋转矩阵。推导的详细信息见附录。8.1

PSS Q_t 和潜在动作集 $\mathcal{V}_{\mathbf{q}_t}$ 都可以在连续空间中包含无穷多个点，从而使得 PSS 的传播变得不可处理。为此，我们以离散化的方式实现 PSS 的传播，类似于处理二值图像。具体而言，我们在每个时间步 t 构造一个二值图像 $\mathcal{I}_t \in \mathbb{R}^{H \times W}$ ，其高度为 H ，宽度为 W ，中心位于 (x_t^c, y_t^c) ，即笼子的中心。我们用 $\mathcal{I}_t(x, y)$ 表示对应于点 (x, y) 的像素强度值。强度值为 1 表示点 (x, y) 存在于物体的 PSS 中，而 0 表示不存在，即 $\mathcal{I}_t(x, y) = 1 \iff (x, y) \in Q_t$ 和 $\mathcal{I}_t(x, y) = 0 \iff (x, y) \notin Q_t$ 。图 9 显示了增强了其他特征的 P_t 的几个示例图像，包括机器人的动作、笼子、物体的 POA 等的可视化。在图中，白色像素（被绿色像素包围，绿色像素是物体的 POA）对应于物体的 PSS，其他颜色的像素表示 PSS 的缺失。

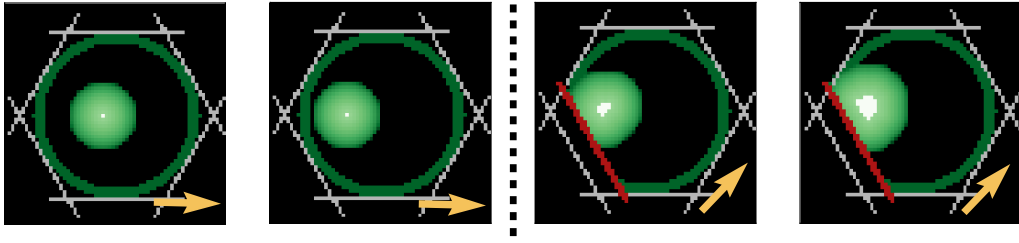


图 9. 一个示例显示了 PSS（白色区域）和 POA（绿色区域）相对于移动笼子（黄色箭头表示）在离散空间中的演变。左：在没有机器人干预的情况下，PSS 和 POA 保持不变。然而，由于笼子向右移动，它们将向左平移。右：如果预测 POA 将移动到 S_A^t （深绿色圆圈）之外，选择的线性推杆（红色）将从左下方与 POA 相互作用，以将其变形为不同的形状。

Algorithm 3: Propagate($\cdot$)

Input: The PSS $\mathcal{Q}_t$, a robot action $u_t \in \mathcal{U}$

Output: The PSS at the next time step $\mathcal{Q}_{t+1}$

```

1  $(x_{t+1}^c, y_{t+1}^c) \leftarrow$  the center of the cage  $\mathcal{S}_{cage}^{t+1}$ 
2  $\mathcal{Q}_{t+1} \leftarrow \{\}$ 
3  $\mathcal{I}_{t+1} \leftarrow$  initialized with zeros
4 for  $\mathbf{q}_t \in \mathcal{Q}_t$  do
5    $\mathcal{V}_{\mathbf{q}_t} \leftarrow U(\mathbf{q}_t, u_t)$   $\triangleright$  Eq. (9)
6   for  $\mathbf{v}_t = (\Delta x, \Delta y) \in \text{DISCRETIZED}(\mathcal{V}_{\mathbf{q}_t})$  do
7      $\mathcal{I}_{t+1}(x_t + \Delta x_t - x_{t+1}^c, y_t + \Delta y_t - y_{t+1}^c) \leftarrow 1$ 
8      $\mathcal{Q}_{t+1} \leftarrow \mathcal{Q}_{t+1} \cup \{(x_t + \Delta x_t, y_t + \Delta y_t)\}$ 
9   end
10 end
11 return  $\mathcal{Q}_{t+1}$ 

```

Algorithm 4: FindPush($\cdot$)

Input: The PSS $\mathcal{Q}_t$, the set of all candidate

robot actions $\mathcal{U}$, the cage region $\mathcal{S}_{cage}^{t+1}$

Output: The selected robot action $u_t \in \mathcal{U}$ to be taken

```

1 if  $\mathcal{Q}_t \subset \mathcal{S}_{cage}^{t+1}$  then
2   return None
3 end
4 for  $k = 1, \dots, K$  do
5    $h^k \leftarrow \lambda_1 S_{out}^k + \lambda_2 (d_{out}^k)^2$   $\triangleright$  Evaluate Heuristics
6 end
7  $\mathcal{U}' \leftarrow \{\text{top five } \theta_k \text{ with greatest } h^k\}$ 
8  $u_t \leftarrow \arg \min_{\theta \in \mathcal{U}'} |\theta - u_{t-1}|$   $\triangleright$  Closest to  $u_{t-1}$ 
9 return  $u_t$ 

```

使用这个离散化的表征（representation），物体的 PSS 由 $\mathcal{I}_t$ 上的有限像素集实现。我们还对可能的运动集 $\mathcal{V}_{\mathbf{q}_t}$ 进行了类似的离散化。因此，为了在离散化空间中传播 PSS，我们只需要逐一传播 PSS 中的每个像素与离散化有限集 $\mathcal{V}_{\mathbf{q}_t}$ 中的每个运动。如前所述， $\mathcal{I}_t$ 始终以移动的笼子为中心。因此，每当我们在 $\mathcal{I}_t$ 上的像素位置与其在原始连续空间中的实际坐标之间转换时，都需要处理一个偏移量 (x_t^c, y_t^c) 。Alg. 3 概述了 PSS 传播的离散化实现。

5.3 Cage the Push in Time

在每个时间步 t ，我们需要从所有候选的推动动作 $\mathcal{U}$ 中选择一个机器人推动动作 u_t 。一个 naive（naive）的方法是模拟 $\mathcal{U}$ 中的每一个动作，传播每一个动作的 PSS，然后选择一个将 PSS 保持在笼子 $\mathcal{S}_{cage}^{t+1}$ 内的动作。然而，传播 PSS 可能会计算昂贵，特别是当候选动作数量 K 很大时。因此，我们提出了一种基于启发式的策略，无需对所考虑的每个动作进行 PSS 传播，具体细节见 Alg. 4。值得一提的是，基于启发式的策略旨在使动作选择更高效，从而减轻对所有可能动作的暴力搜索的负担。这并不与时间中的笼养的定义的完整性相冲突。如果启发式选择的推动动作被验证符合时间中的笼养条件，则对象的状态（state）仍然保证被始终笼养。然而，这项工作并不专注于开发最佳的动作选择策略，我们承认未来可能会开发出更好的策略。

随着笼子从上一个时间步的 $\mathcal{S}_{cage}^t$ 移动到跟随期望轨迹的 $\mathcal{S}_{cage}^{t+1}$ ，如果 PSS 仍然在移动笼子内，即 $\mathcal{Q}_t \subset \mathcal{S}_{cage}^{t+1}$ ，则不需要机器人动作，Alg. 4 将返回 None。然而，如果 $\mathcal{Q}_t \not\subset \mathcal{S}_{cage}^{t+1}$ ，这表明如果不采取机器人

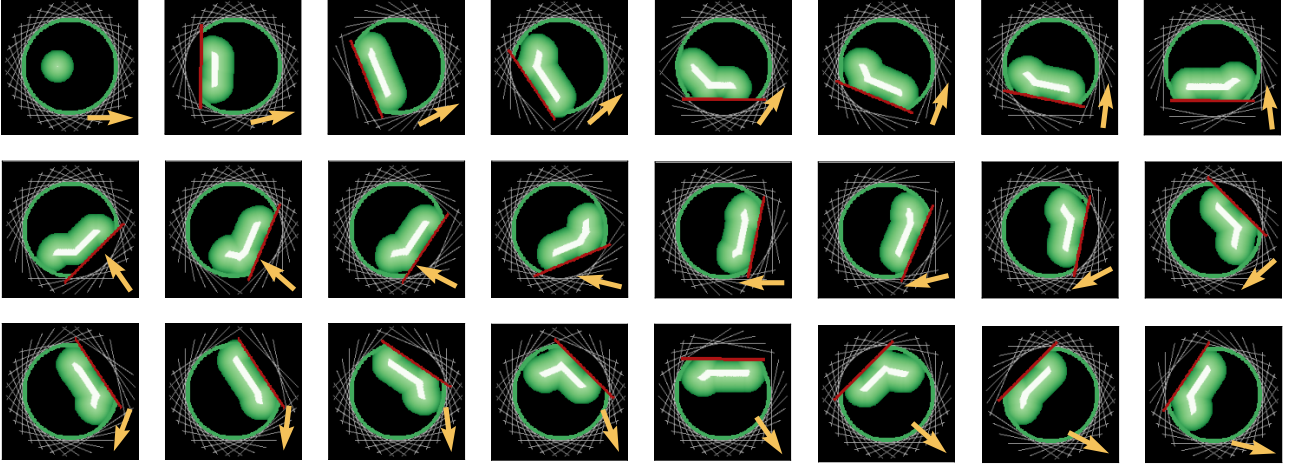


图 10. PSS (白色区域) 和 POA (绿色区域) 在笼子沿圆形路径移动时的传播。每个图像都以笼子的中心为中心。每个图像中的黄色箭头表示笼子的动作方向 (motion direction)。在这个例子中, 使用了 $K = 32$ 个候选动作 (candidate actions), 由灰色条表示。通过为每个步骤选择不同的推动动作 (pushing action) (红色条), 物体在时间上始终被困在笼中, POA 始终保持在 $\mathcal{C}_A^t$ (深绿色圆圈) 内。

动作, 物体的 PSS 将会越过笼子。在这种情况下, 我们必须选择一个机器人动作来移动和变形 PSS, 以保持对象被笼养。

回顾在 Sec. 5.1 中, $\mathcal{U}$ 中的每个候选动作 (action) 由一个角度 θ_k 表示, 该角度决定了线性推送器的起始姿态。对于 $\mathcal{U}$ 中索引为 $k = 1, \dots, K$ 的每个候选动作 (action), 我们将通过 $h^k = \lambda_1 S_{out}^k + \lambda_2 (d_{out}^k)^2$ 评估一个启发式评分, 其中 λ_1 和 λ_2 是两个权重因子。如 Fig. 7 右侧所示, S_{out}^k 是当我们将推送器放置在由角度 θ_k 确定的起始姿态时, 线性推送器后面的 POA 区域, d_{out}^k 是从线性推送器后面到 POA 边界的最远距离。值得注意的是, 计算每个动作 (action) 的启发式评分 h^k 不需要传播 PSS, 即 PSS 保持为 $\mathcal{Q}_t$ 。直观上, 较大的 h^k 表明如果不采取第 k 个动作 (action), 则 PSS 的很大一部分将会超出笼子。

具有最高启发式评分的前五个候选动作 (action), 组成一个集合 $\mathcal{U}'$, 将进一步考虑以选择最佳动作 (action)。在这五个候选动作 (action) 中, 将选择与先前动作 (action) u_{t-1} (即, 上一次推送的角度) 最接近的一个, 来将 PSS 传播至下一步。这一机制减少了在相邻时间步的线性推送器重新定位, 从而在真实机器人操纵器上实施时可以大大提高运动效率。在准静态设置下, 我们知道当物体不与机器人推送器相互作用时, 它不会移动。这一假设确保物理机器人能够及时切换到后续的推送动作 (action), 而不需要足够快的操作速度。因此, 动作 (action) 可行性的验证 (Alg. 1 第 3 行) 始终得到满足。Fig. 10 展示了在笼子移动的过程中 PSS 和 POA 的整体演变过程, 通过策略性选择的推送动作 (action) 确保物体在整个任务中保持被笼住状态。

6 Dynamic Tasks

在这一节中, 我们将我们的 *Caging in Time* 理论从准静态任务扩展到动态场景, 在这里我们在一个具有挑战性的动态球平衡问题上实例化相同的框架。我们开发了工具来传播球的位置和速度的 PSS (概率状态空间), 以及生成开环机器人动作, 以保持球的平衡, 同时机器人末端执行器 follows 不同的轨迹, 而不需要任何传感反馈。这是 *Caging in Time* 在基于能量的状态 (state) 空间中的一个实例化。

6.1 Problem Statement

机器人被赋予在倾斜的板 (末端执行器) 上平衡一个滚动物体的任务, 同时该板在工作空间中沿着不同的轨迹移动。该板是一个 n 维平面, 位于维度为 $n+1$ 的工作空间中, 其中 $n = 1$ 或 2 。板的每个维度的大小在 $-l$ 到 l 之间。物体的质量为 m , 半径为 r_b , 估计的滚动摩擦系数为 μ_r , 以及估计的转动惯量为 I_b 。我们假设物体仅在板上滚动而不会滑动。图 11 展示了 $n = 1$ 和 2 的示例设置。板的位置记作 $\mathbf{x}_p \in \mathbb{R}^{n+1}$, 其平移加速度

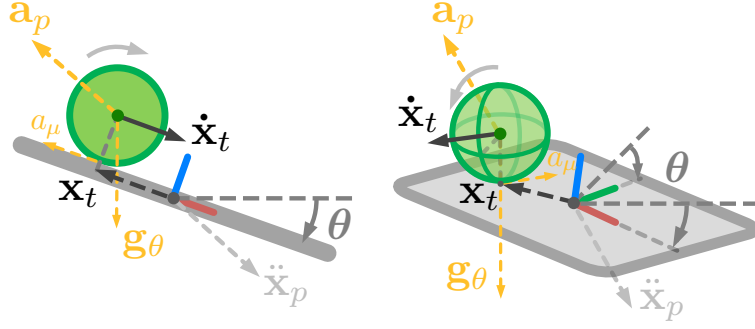


图 11. 动态球平衡系统在板尺寸 $n = 1$ 和 $n = 2$ 的示意图。左: 当 $n = 1$ 时的物理设置。 $\mathbf{x}_t$ 和 $\dot{\mathbf{x}}_t$ 分别表示球的位置和速度。板的倾斜角度用 θ 表示。作用于球的加速度包括重力 $\mathbf{g}_\theta$ 、由板的加速度 $\ddot{\mathbf{x}}_p$ 产生的加速度 $\mathbf{a}_p$, 以及滚动摩擦 a_μ 。右: 当 $n = 2$ 时的物理设置, 其中符号与 $n = 1$ 的情况相同。注意这里的 θ 是包含两个角度的倾斜角度向量。

为 $\ddot{\mathbf{x}}_p \in \mathbb{R}^{n+1}$ 。其方向由一个倾斜角向量 $\boldsymbol{\theta} \in \mathbb{R}^n$ 来表征。虽然只有 $\boldsymbol{\theta}$ 和 $\ddot{\mathbf{x}}_p$ 最初是在世界坐标系（惯性）中定义的，但所有其他定义和模型都是在板坐标系（非惯性）中进行的。在这种情况下，我们引入球体的非惯性加速度 $\mathbf{a}_p \in \mathbb{R}^{n+1}$ 和由于 $\ddot{\mathbf{x}}_p$ 和 $\boldsymbol{\theta}$ 引起的重力 $\mathbf{g}_\theta \in \mathbb{R}^{n+1}$ ，以便所有计算可以在板坐标系中进行，如图 11 所示。

物体在时间 t 的状态定义为 $\mathbf{q}_t = (\mathbf{x}_t, \dot{\mathbf{x}}_t) \in \mathbb{R}^{2n}$ ，其中 $\mathbf{x}_t \in \mathbb{R}^n$ 是相对于板中心的物体位置向量， $\dot{\mathbf{x}}_t \in \mathbb{R}^n$ 是其速度向量。在这个动态系统中，状态空间记作 $\mathcal{S}_{obj} \subset \mathbb{R}^{2n}$ ，它包含了物体在板坐标系中可能具有的所有状态。

为了以连续的方式表示状态的分布，我们将物体处于状态 $\mathbf{q}_t$ 的概率密度定义为 $p(\mathbf{q}_t) = p(\mathbf{x}_t, \dot{\mathbf{x}}_t)$ 。PSS $\mathcal{Q}_t$ 被表示为所有具有非零概率密度的可能状态的集合：

$$\mathcal{Q}_t = \{\mathbf{q}_t \in \mathcal{S}_{obj} | p(\mathbf{q}_t) > 0\} \quad (10)$$

在定义 PSS 之后，需要一个“笼区域”来约束这些状态（state）。与几何定义笼子的准静态任务不同（见 Sec. 5.1），动态系统需要不同的方法，因为它们的状态空间同时涵盖配置和速度。对于这样的系统，能量提供了一个强大且直观的框架，用于表示和约束包含运动的状态。

$$\mathcal{S}_{cage}^t = \{\mathbf{q}_t \in \mathbb{R}^{2n} | E(\mathbf{q}_t, \mathbf{g}_\theta, \mathbf{a}_p) < E_{max}(\mathbf{g}_\theta, \mathbf{a}_p)\} \quad (11)$$

系统的总能量 $E(\mathbf{q}_t, \mathbf{g}_\theta, \mathbf{a}_p)$ ，对于给定状态 $\mathbf{q}_t = (\mathbf{x}_t, \dot{\mathbf{x}}_t)$ 表达为：

$$E(\mathbf{q}_t, \mathbf{g}_\theta, \mathbf{a}_p) = E_k(\dot{\mathbf{x}}_t) + E_{ve}(\mathbf{x}_t) + E_p(\mathbf{x}_t, \mathbf{g}_\theta, \mathbf{a}_p) \quad (12)$$

在这里， E_k 、 E_{ve} 和 E_p 分别代表板框架中的动能、虚拟弹性能（virtual elastic energy）和势能。

为了建立一个上限，确保物体无法从板中“逃脱”的允许能量，我们考虑一个极端场景，即物体静态地位于板的边缘，从板的框架来看。这个最大允许能量 E_{max} 被定义为：

$$E_{max}(\mathbf{g}_\theta, \mathbf{a}_p) = E_{ve}(l) + E_p(l, \mathbf{g}_\theta, \mathbf{a}_p) \quad (13)$$

如果物体的总能量在任何时间步超过 E_{max} ，即使它当前的位置在板内，它也将有足够的能量超出板的边界。通过保持物体的能量在条件 $E(\mathbf{q}_t, \mathbf{g}_\theta, \mathbf{a}_p) < E_{max}(\mathbf{g}_\theta, \mathbf{a}_p)$ 之下，我们确保它在整个任务过程中保持在板内，从而有效地创建一个基于能量的“笼子”。

这种基于能量的笼区定义可以捕捉位置和速度约束，并建立边界，以防止物体离开板，从而有效地将其围困在状态空间内，并实现稳健的动态物体操作。

机器人的控制动作通过调整板的倾斜角度向量 $\boldsymbol{\theta}$ 来定义，该角度由一个连续变量 $\mathbf{u}_t = d\boldsymbol{\theta}(t) \in \mathbb{R}^n$ 表示，代表倾斜角度的变化率。板的平移运动由预定轨迹 $\mathcal{T} = \{\mathbf{x}_p(t)\}_{t=0}^T$ 在世界坐标系中引导，其中 $\mathbf{x}_p(t) \in \mathbb{R}^{n+1}$

表示板中心在工作空间中的位置。然后通过使用速度的数值微分计算板的加速度轨迹 $\mathcal{T}_a = \{\ddot{\mathbf{x}}_p(t)\}_{t=0}^T$, 速度又是从位置轨迹 $\mathcal{T}$ 导出的。根据第 4 节中定义的理论, 我们的目标是在板按照其预定轨迹 $\mathcal{T}$ 运动时, 保持物体的状态在能量约束笼 $\mathcal{S}_{cage}^t$ 内, 确保物体的 PSS 始终满足 $\mathcal{Q}_t \subset \mathcal{S}_{cage}^t$ 。

接下来, 我们基于所提出的时间笼困理论开发一个解决动态球平衡问题的算法。在每个时间步 t , 我们计算最大可允许的能量, 计算当前 PSS 的加权平均能量, 并解决一个优化问题, 以找到保持球体在时间内被囚禁的最佳控制输入 $\mathbf{u}_t = d\boldsymbol{\theta}(t)$ 。然后, 我们根据系统动态传播 PSS, 并检查能量约束和位置约束是否满足。如果在任何时间步中违反了任何约束, 任务将被视为失败。

6.2 Ball Dynamics and PSS Propagation

如前所述, 系统在时间 t 的状态 $\mathbf{q}_t = (\mathbf{x}_t, \dot{\mathbf{x}}_t)$ 及其相应的概率密度 $p(\mathbf{q}_t)$ 。我们将系统在相邻时间步之间的运动表示为 $\mathbf{v}_t = \Delta\mathbf{q}_t = (\dot{\mathbf{x}}_t\Delta t, \ddot{\mathbf{x}}_t\Delta t)$, 其相关概率为 $p(\mathbf{v}_t)$ 。根据 Eq. (6) 中的定义, 某一 $\mathbf{q}_t$ 的潜在运动集合 $\mathcal{V}_{\mathbf{q}_t}$ 可以表示为:

$$\mathcal{V}_{\mathbf{q}_t} = U(\mathbf{q}_t, u_t) = \{\mathbf{v}_t \mid \mathbf{v}_t = (\dot{\mathbf{x}}_t\Delta t, \ddot{\mathbf{x}}_t\Delta t)\} \quad (14)$$

状态运动对 $(\mathbf{q}_t, \mathbf{v}_t)$ 的传播函数 π 从 $\mathbf{q}_t$ 传播到 $\mathbf{q}_{t+1}$ 由联合概率 $p(\mathbf{q}_t, \mathbf{v}_t)$ 给出:

$$\pi(\mathbf{q}_t, \mathbf{v}_t) = (\mathbf{x}_t + \dot{\mathbf{x}}_t\Delta t, \dot{\mathbf{x}}_t + \ddot{\mathbf{x}}_t\Delta t) \quad (15)$$

注意, π 是一个用于状态 (state) 转移的确定性函数。

$$\begin{aligned} p(\mathbf{q}_{t+1}) &= \iint_{\mathbf{q}_t \in \mathcal{Q}_t, \mathbf{v}_t \in \mathcal{V}_{\mathbf{q}_t}} p(\mathbf{q}_t, \mathbf{v}_t) \mathbb{1}(\pi(\mathbf{q}_t, \mathbf{v}_t) = \mathbf{q}_{t+1}) d\mathbf{v}_t d\mathbf{q}_t \\ &= \int_{\mathbf{q}_t \in \mathcal{Q}_t} p(\mathbf{q}_t) \int_{\mathbf{v}_t \in \mathcal{V}_{\mathbf{q}_t}} p(\mathbf{v}_t | \mathbf{q}_t) \mathbb{1}(\pi(\mathbf{q}_t, \mathbf{v}_t) = \mathbf{q}_{t+1}) d\mathbf{v}_t d\mathbf{q}_t \end{aligned} \quad (16)$$

其中 $p(\mathbf{v}_t | \mathbf{q}_t)$ 的分解可以表示为:

$$\begin{aligned} p(\mathbf{v}_t | \mathbf{q}_t) &= p(\dot{\mathbf{x}}_t | \mathbf{q}_t) \cdot p(\ddot{\mathbf{x}}_t | \mathbf{q}_t) \\ &= p(\ddot{\mathbf{x}}_t | \mathbf{q}_t) \\ &= p(\ddot{\mathbf{x}}_t) \end{aligned} \quad (17)$$

其中 $p(\dot{\mathbf{x}}_t | \mathbf{q}_t) = 1$ 因为 $\dot{\mathbf{x}}_t$ 完全由 $\mathbf{q}_t$ 决定。为了计算 $p(\ddot{\mathbf{x}}_t)$, 我们需要考虑系统动态。根据第 6.1 节中提到的球体的物理特性, 我们可以为给定状态 $\mathbf{q}_t$ 和控制输入 u_t 定义潜在加速度的集合 $\mathcal{A}_{\mathbf{q}_t}$:

$$\begin{aligned} \mathcal{A}_{\mathbf{q}_t} &= \left\{ \ddot{\mathbf{x}}_t \mid \ddot{\mathbf{x}}_t = f(\mathbf{q}_t, u_t, \eta_m, \boldsymbol{\eta}_p, \eta_\mu), \right. \\ &\quad \eta_m \sim \mathcal{N}(0, \sigma_m^2), \\ &\quad \boldsymbol{\eta}_p \sim \mathcal{N}(\mathbf{0}, \Sigma_p), \\ &\quad \left. \eta_\mu \sim \mathcal{N}(0, \sigma_\mu^2) \right\} \end{aligned} \quad (18)$$

其中 $f(\cdot)$ 表示系统动态方程, 详细内容见附录 8.2.1。术语 η_m 、 $\boldsymbol{\eta}_p$ 和 η_μ 分别表示质量、平板加速度和摩擦系数的不确定性。

利用高斯分布在线性运算下的闭合性, 我们可以得出加速度 $\ddot{\mathbf{x}}_t$ 是高斯分布不确定性的线性组合, 因此也服从高斯分布:

$$\ddot{\mathbf{x}}_t \sim \mathcal{N}(\boldsymbol{\mu}_{\ddot{\mathbf{x}}}, \boldsymbol{\Sigma}_{\ddot{\mathbf{x}}}) \quad (19)$$

其中 $\boldsymbol{\mu}_{\ddot{\mathbf{x}}}$ 和 $\boldsymbol{\Sigma}_{\ddot{\mathbf{x}}}$ 是从系统动力学方程和不确定性项的分布中推导出来的。

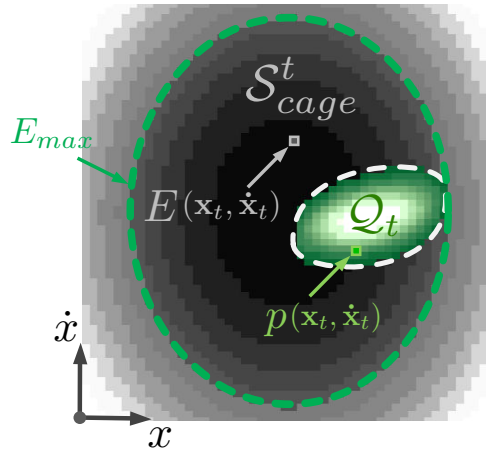


图 12. 离散化的状态空间 $\mathcal{S}_{obj}$ 与能量映射。离散化映射的每个像素表征 (representation) 一个状态 $(\mathbf{x}_t, \dot{\mathbf{x}}_t)$ 的球。笼子区域 $\mathcal{S}_{cage}^t$ 被最大能量 E_{max} 限定。梯度灰色区域显示不同状态下的能量价值 (value) $E(\mathbf{x}_t, \dot{\mathbf{x}}_t)$ 。而梯度绿色区域显示 PSS $\mathcal{Q}_t$ 与球的概率密度 $p(\mathbf{x}_t, \dot{\mathbf{x}}_t)$ 在不同状态下。

Algorithm 5: DynamicPropagate($\cdot$)

Input: Current PSS $\mathcal{Q}_t$, a robot action $u_t \in \mathcal{U}$, plate acceleration $\ddot{\mathbf{x}}_p$, time step Δt

Output: The PSS at next time step $\mathcal{Q}_{t+1}$

```

1  $\mathcal{Q}_{t+1} \leftarrow \{\}$ 
2  $\mathcal{I}_{t+1} \leftarrow$  initialized with zeros
3 for  $q_t \in \mathcal{Q}_t$  do
4    $\mathcal{V}_{q_t} \leftarrow U(\mathbf{q}_t, u_t)$  ▷ Eq. (14)
5   for  $\mathbf{v}_t = (\dot{\mathbf{x}}_t \Delta t, \ddot{\mathbf{x}}_t \Delta t) \in \text{DISCRETIZED}(\mathcal{V}_{q_t})$  do
6      $\mathbf{q}_{t+1} \leftarrow \text{DISCRETIZED}(\pi(\mathbf{q}_t, \mathbf{v}_t))$ 
7     if  $\mathbf{q}_{t+1} \in \mathcal{S}_{obj}$  then ▷ Eq. (17)
8        $P_{q,v} \leftarrow \text{CALCPROB}(\mathbf{q}_t, \mathbf{v}_t)$ 
9        $\mathcal{I}_{t+1}(\mathbf{q}_{t+1}) \leftarrow \mathcal{I}_{t+1}(\mathbf{q}_{t+1}) + P_{q,v}$ 
10    end
11  end
12 end
13  $\mathcal{I}_{t+1} \leftarrow \text{NORMALIZE}(\mathcal{I}_{t+1})$ 
14 return  $\mathcal{I}_{t+1}$ 

```

鉴于 $\ddot{\mathbf{x}}_t$ 的高斯分布，我们现在可以计算对于任何 $\ddot{\mathbf{x}}_t \in \mathcal{A}_{q_t}$ 的 $p(\ddot{\mathbf{x}}_t)$ ，这使得我们能够基于 Eq. (16) 和 Eq. (17) 进一步计算 $p(\mathbf{q}_{t+1})$ 。

由于 $p(\mathbf{q}_{t+1})$ 的概率是从 $\mathcal{Q}_t$ 中的所有状态传播而来的，因此我们可以以与 Eq. (10) 相同的方式获得下一时间步的 PSS。

$$\mathcal{Q}_{t+1} = \{\mathbf{q}_{t+1} \in \mathcal{S}_{obj} | p(\mathbf{q}_{t+1}) > 0\} \quad (20)$$

为了数值实现由 Eq. (20) 和 Eq. (16) 描述的 PSS 传播，我们采用了状态空间的离散化表征 (representation)，与准静态情况 (Sec. 5.2) 中使用的方法类似。然而，我们使用连续值而非二进制值来表示状态空间中 PSS 的概率分布。

我们在每个时间步 t 构造一个 $2n$ 维数组 $\mathcal{I}_t \in \mathbb{R}^{N^{2n}}$ ，其中 N 是状态每个维度的网格大小。数组 $\mathcal{I}_t$ 在状态空间中以原点为中心，满足 $\mathbf{x} \in [-\mathbf{x}_{max}, \mathbf{x}_{max}]$ 和 $\dot{\mathbf{x}} \in [-\dot{\mathbf{x}}_{max}, \dot{\mathbf{x}}_{max}]$ 。状态空间在每个维度上均匀离散化，位

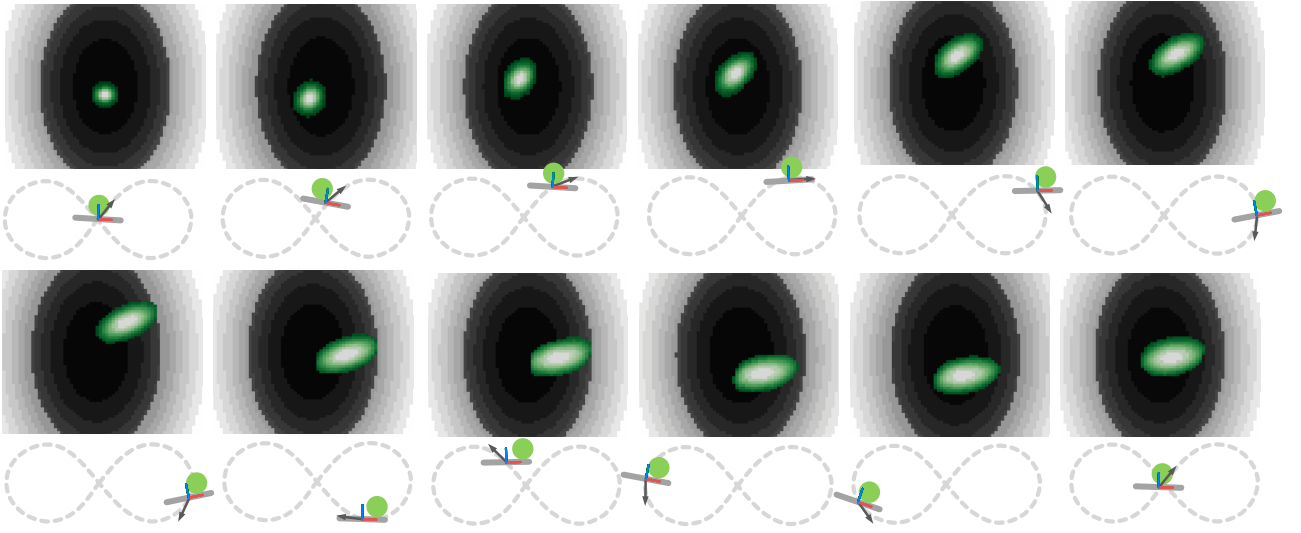


图 13. 球体 PSS 在动态平衡过程中的演化序列，其中板的尺寸 $n = 1$ 。顶部行：PSS Q_t (绿色) 在笼区 S_{cage}^t (深色背景) 内的状态空间表征 (状态空间 representation)，如图 12 中详细描述。底部行：系统在世界坐标系中的相应配置。深灰色箭头显示了板的平移运动方向。

置和速度分量具有适当的网格间距。每个元素 $\mathcal{I}_t(\mathbf{q}_t)$ 表示离散化版本的状态 $\mathbf{q}_t$ 发生的概率，这个概率是由 $p(\mathbf{q}_t)$ 推导而来的。图 Fig. 12 展示了在 $n = 1$ 时，上述数值表征下的可视化示例。

PSS 传播在 Alg. 5 中实现，该算法将 Eq. (16) 中的连续积分离散化。对于当前 PSS Q_t 中的每个 $\mathbf{q}_t$ ，我们使用 Eq. (14) 计算潜在运动的集合 $\mathcal{V}_{\mathbf{q}_t}$ 。然后，我们对这些运动的离散化版本进行迭代，根据 Eq. (17) 更新结果状态 $\mathbf{q}_{t+1}$ 的概率：

$$\mathcal{I}_{t+1}(\mathbf{q}_{t+1}) = \mathcal{I}_{t+1}(\mathbf{q}_{t+1}) + \mathcal{I}_t(\mathbf{q}_t) \cdot p(\mathbf{v}_t | \mathbf{q}_t) \quad (21)$$

其中 $p(\mathbf{v}_t | \mathbf{q}_t)$ 的计算如公式 (17) 所示。在更新所有概率后，我们对其进行归一化，以确保网格的值 (value) 总和为 1，作为有效分布。为了提高计算效率，我们对 $\mathcal{I}_{t+1}$ 应用阈值，将低于某一值 (value) 的概率 (例如， 10^{-3}) 设置为零。

这基于像素的方法提供了对状态空间中 PSS (部分状态空间) 的计算效率高的表征 (representation)，避免了与先前在长时间任务中使用的采样方法相关的计算成本的指数增长 (Welle et al. 2021; Makapunyo et al. 2012)。图 13 说明了在追踪一维板的 ∞ 形轨迹期间，PSS 在状态空间中的传播和囚禁。PSS 最初表现为一个紧凑的高斯分布，随后扩展，最后收敛到一个由能量边界囚禁的稳定大小。

6.3 Cage the Ball in Time

为了实现动态球平衡问题的时间笼养，我们采用了一种控制优化策略，该策略结合了控制障碍函数 (Ames et al. 2019) 和控制李雅普诺夫函数 (Blanchini 1995; Anand et al. 2021)，以保持球在板上并沿着期望轨迹移动。

在我们的案例中，控制障碍函数 (CBF) 用于在动态环境中实现“笼养”的概念。就像一个物理笼子限制物体在一定空间内一样，CBF 建立了一个基于能量的笼子，可以通过 PSS 中可达到的最大能量与预定义能量边界之间的差距来表征：

$$h(Q_t) = E_{max}(\mathbf{g}_\theta, \mathbf{a}_p) - \max_{\mathbf{q}_t \in Q_t} E(\mathbf{q}_t, \mathbf{g}_\theta, \mathbf{a}_p) \quad (22)$$

这创造了一个能量障碍，并作为我们笼子的数学表征 (representation)。通过确保 $h(Q_t) > 0$ 在所有时候，我们保证球的能量 $E(\mathbf{q}_t, \mathbf{g}_\theta, \mathbf{a}_p)$ 在状态 $\mathbf{q}_t$ 中，如 Eq. (12) 所计算的，永远不会超过在 Eq. (13) 中定义的最大允许能量 E_{max} 。

为了进一步确保我们的控制动作保持这个能量障碍,我们实施了一个 CBF 条件。这提供了一种前瞻性的约束,不仅考虑当前状态,还考虑系统的未来行为,而不仅仅是强制 $h(Q_t) > 0$ 。

$$L_f h(Q_t) + L_g h(Q_t) d\theta + \alpha(h(Q_t)) \geq 0 \quad (23)$$

在这种情况下, $L_f h(Q_t)$ 和 $L_g h(Q_t)$ 是 CBF $h(Q_t)$ 在向量场 f 和 g 下的李导数,其中 f 和 g 在系统动态方程 $\ddot{\mathbf{x}}_t = f + gu$ 中定义,详见附录 8.2.2。 $L_f h(Q_t)$ 表示由于系统的自然动态而导致的障碍函数的变化,而 $L_g h(Q_t) d\theta$ 表示控制输入导致的变化。项 $\alpha(\cdot)$ 是一个类 $\mathcal{K}$ 函数,塑造障碍函数的收敛行为,确保在系统接近安全集的边界时其增加得更快。

控制李雅普诺夫函数 (Control Lyapunov Functions, CLFs) 另一方面用于通过将系统驱动到期望的状态来优化控制性能。在我们的上下文中,我们使用 CLF 来鼓励球体保持在板子的中心附近,并维持一个良好分布的概率状态,同时考虑系统的能量和熵:

$$V(Q_t) = \sum_{\mathcal{I}_t(\mathbf{q}_t) > 0} \mathcal{I}_t(\mathbf{q}_t) E(\mathbf{q}_t, \mathbf{g}_\theta, \mathbf{a}_p) - k_S S(Q_t) \quad (24)$$

其中 $\mathcal{I}_t(\mathbf{q}_t)$ 是离散概率分布中状态 $\mathbf{q}_t$ 的概率值, $E(\mathbf{q}_t, \mathbf{g}_\theta, \mathbf{a}_p)$ 是状态 $\mathbf{q}_t$ 下的系统能量, k_S 是一个权重因子, $S(Q_t)$ 是系统熵。

这个 CLF (控制Lyapunov函数) 平衡了两个目标:最小化系统的期望能量,这趋向于将球保持在盘子的中心附近;最大化概率分布的熵,这维持了良好分布的概率状态,在面对不确定性时确保了系统的稳定性和鲁棒性。通过最小化这个 CLF,我们驱动系统朝向一个状态,在此状态下,球可能接近盘子的中心,但又保持一定的不确定性,以应对意外的干扰。

类似于 CBF (控制Barrier函数) 条件,我们强制执行 CLF 条件,以确保 CLF 随时间减少:

$$L_f V(Q_t) + L_g V(Q_t) d\theta + cV(Q_t) \leq \delta \quad (25)$$

其中 $L_f V(Q_t)$ 和 $L_g V(Q_t)$ 分别是系统动态和控制输入下的 CLF (控制李雅普诺夫函数) 的李导数, c 是收敛速度, δ 是松弛变量。这个条件确保我们的控制输入 $d\theta$ 以足够的速度将系统驱动到期望的行为,同时松弛变量 δ 在必要时允许暂时违反降低条件,以满足 CBF (控制屏障函数) 安全约束。

通过结合 CBF 和 CLF 条件,我们制定了一个二次规划 (QP),不仅可以保持球在盘子上,还可以尽量保持其居中和稳定。

$$\begin{aligned} \min_{d\theta, \delta} \quad & (d\theta)^2 + \lambda \delta^2 \\ \text{s.t.} \quad & L_f h(Q_t) + L_g h(Q_t) d\theta + \alpha(h(Q_t)) \geq 0 \\ & L_f V(Q_t) + L_g V(Q_t) d\theta + c(V(Q_t)) \leq \delta \\ & d\theta_{\min} \leq d\theta \leq d\theta_{\max} \end{aligned} \quad (26)$$

在这里, $\lambda > 0$ 是一个权重因子,用于平衡最小化控制努力 $(d\theta)^2$ 与 CLF (控制李雅普诺夫函数) 约束违背 δ^2 之间的权衡。 $d\theta_{\min}$ 和 $d\theta_{\max}$ 的值源自机器人手臂电机的硬件扭矩限制。我们利用 OSQP (算子分裂二次规划) 求解器在每个时间步高效地解决这个优化问题。

通过在每个时间步解决这个 QP,我们获得了最佳控制输入,使得球保持在基于能量的笼子内,同时遵循期望的轨迹。详细的实现呈现在附录 8.2.2。

这种方法使我们能够在倾斜板上的动态条件下及时地控制球,考虑到球的动态、不可预知性和能量约束。通过在每个时间步解决 QP,我们生成一系列动作 (action),使球保持在基于能量的笼子区域内,同时板子跟随期望轨迹。

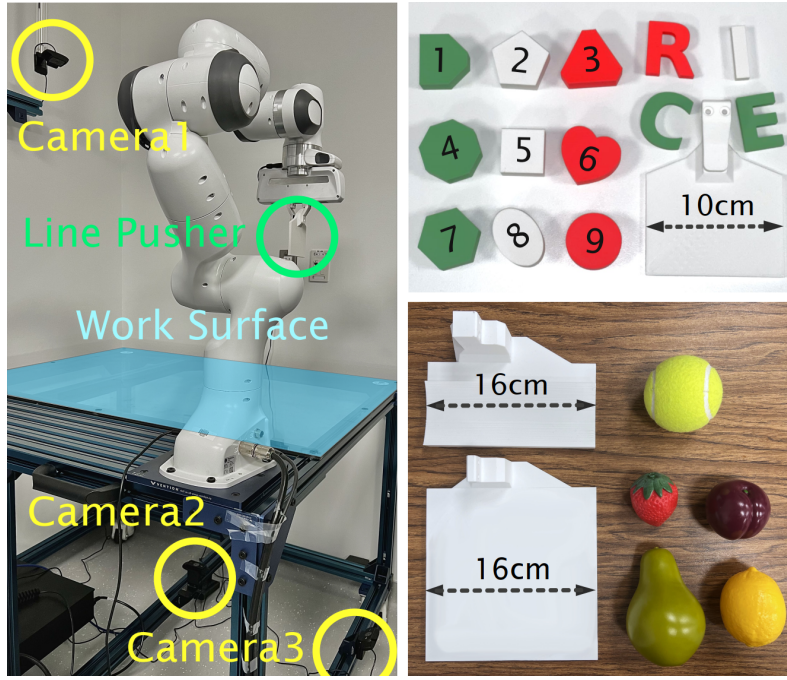


图 14. 实验设置。左侧：机器人和相机设置。机器人在一个透明表面（蓝色）上进行实验，预装了一个线推器（绿色）。我们在表面下方使用相机 2 和 3 记录（而不是跟踪）物体的轨迹，通过 April-Tag。相机 1 用于从上方视角记录实验。右上角：实验中使用的 3D 打印物体。1. 有两个修剪尖端的正方形，2. 五边形，3. 有修剪尖端的三角形，4. 八边形，5. 正方形，6. 心形，7. 六边形，8. 椭圆形，9. 圆形，以及字母形状的物体：R, I, C, E。右下角：用于动态任务的 3D 打印板、网球和塑料水果。正方形板用于当维度 $n = 2$ 时的球平衡，而较薄的板用于当维度 $n = 1$ 时的球平衡和捕捉，带有导轨以限制球的运动仅在所需维度上。塑料水果包括草莓（YCB #12）、梅子（YCB #18）、梨（YCB #16）和柠檬（YCB #14）。

7 Experiments

我们进行了广泛的实验,以评估我们的 *Caging in Time* 框架,使用 Franka Emika Panda 机器人手臂进行不同末端执行器如图 14 所示的准静态推送任务和动态平衡任务。所有算法均使用 Python 实现,并在 3.4 GHz 的 AMD Ryzen 9 5950X CPU 的单线程上执行。

我们的实验设置利用多台摄像机进行录制:如图 14 左侧所示的摄像机用于准静态任务,另有一台摄像机用于动态任务。在准静态实验中,物体跟踪使用了 AprilTags (Olson 2011),而在动态场景中,则使用 OpenCV 跟踪球的运动。需要注意的是,物体跟踪仅用于轨迹可视化和准确性评估,所有实验均在开环方式下进行。

为了量化操作精度,我们采用平均绝对误差 (Mean Absolute Error, MAE) 作为我们的主要指标。这是通过计算操作过程中实际轨迹与参考轨迹之间的绝对距离的平均值得出的。通过这些实验,我们旨在全面评估我们的 *Caging in Time* 框架在一系列准静态和动态操作任务中的性能、鲁棒性和多功能性。

7.1 Quasi-static Tasks

通过准静态实验,我们希望通过三个方面调查我们框架的性能:1) 在没有任何感知反馈(除了边界圆的大小和初始位置)的情况下,能否通过提议的 *Caging in Time* 实现稳健的平面操作? 2) 如果可以,能够达到什么精度,以及不同框架设置和外部干扰对稳健性有何影响? 3) 在位置噪声和网络延迟等不完美感知下, *Caging in Time* 能否超越闭环方法?

如图 14 所示,任务中的机器人使用了一个 100 毫米的线性推动器,并且各种形状的物体是以相似的尺寸 3D 打印的,适合进入半径为 $r = 25$ 毫米的边界圆。每个动作的推动距离设置为 $d_{push} = 20$ 毫米。

7.1.1 Evaluation of Cage Settings 在本实验中,实验任务是沿着预定义的圆形轨迹推动物体。该实验重点评估了通过改变两个关键参数来评价我们框架的精度: $R - r$, 笼子的大小,以及 K , 候选动作 (action) 的数量。我们为 $R - r$ 和 K 选择了来自集合 $R - r = \{10, 20, 30, 40\}$ mm 和 $K = \{16, 32, 64, 128\}$ 的值。

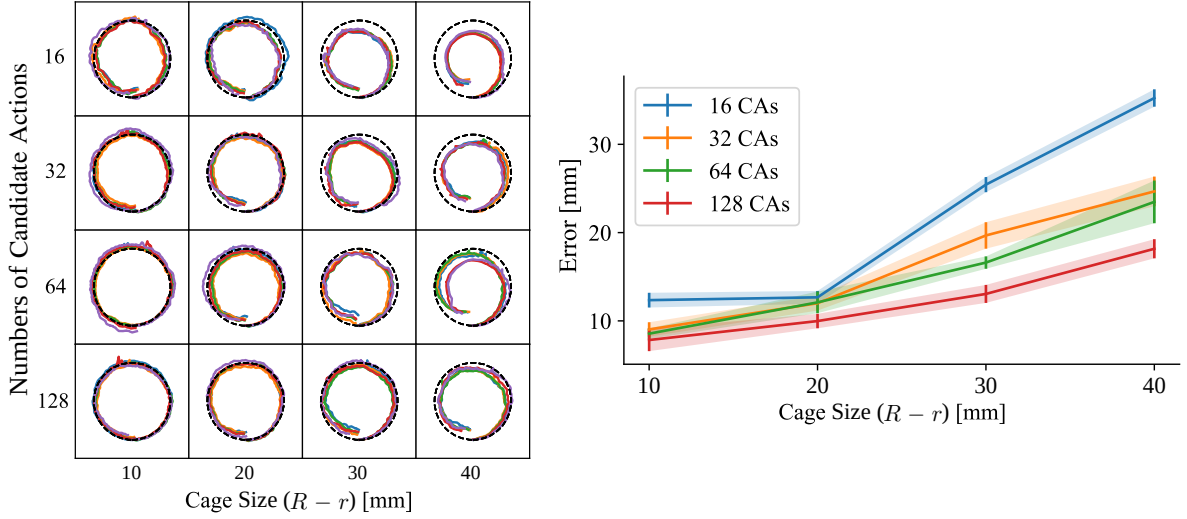


图 15. 评估结果涉及笼子大小 ($R - r$) 和候选动作 (candidate action) 的数量。顶部: 所有轨迹 (每个圆圈中叠加了九个不同物体的轨迹) 底部: 在不同笼子设置下显示轨迹的平均绝对误差 (Mean Absolute Error)。

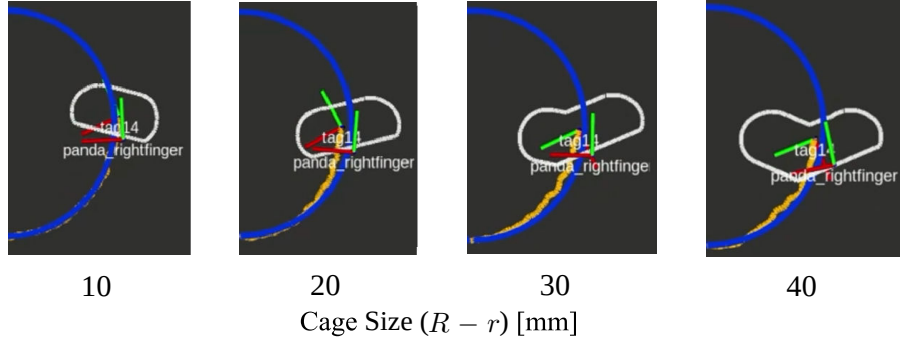


图 16. 在不同笼子尺寸下真实推动实验记录中POAs (白色边界) 的可视化。

对于每个 $R - r$ 和 K 的组合, 使用五种不同的物体 (物体1-5, 请参见图 14) 进行了五次试验。我们针对每个 $(R - r, K)$ 设置仅生成了一个机器人动作序列, 遵循算法2, 并以开环方式将其应用于不同的物体。所有试验的平均绝对误差 (MAE) 被计算并呈现在图 15中, 该图还包括不同设置和不同物体的实际记录轨迹。

从结果中, 我们可以看到性能遵循两个主要趋势。一个是较大的栅栏半径 (cage radius) 往往会增加平均误差。另一个趋势是候选动作 (candidate actions) 数量的增加通常会导致误差的减少。如图 16 所示, 随着栅栏尺寸的增大, POA 也会增大, 从而增加物体可能偏离参考轨迹的范围。此外, 候选动作数量的增加提高了在算法 4 中找到最优动作的可能性。

通过检查图 15 中的轨迹, 我们观察到所有路径都保持在各自的栅栏区域内。值得注意的是, 由相同动作序列生成的轨迹在不同物体之间显示出相似的模式。例如, 拥有 16 个候选动作和 40 毫米栅栏尺寸 ($R - r$) 的轨迹大多数移动到参考路径的左侧, 这表明动作序列在性能中相较于物体形状起着更主导的作用。

7.1.2 Why Caging in Time 虽然线性推动器可能被视为一种以开放循环方式推动物体的稳定方法, 但这种稳定性往往受到物体物理特性和形状不可预测性的影响, 这就需要实施 *Caging in Time* (时域围困)。

为了验证我们方法的必要性和稳健性, 一个物体被要求重复沿着 ∞ 形状的轨迹导航 10 次循环。我们与一种天真的推动策略进行了比较分析, 在这种策略中, 推动器简单地跟随目标轨迹, 并使其方向与轨迹方向平行。如图 17 所示, 与天真的方法相比, 后者在几步之后就失去了对物体的控制, 而我们的方法则在 10 轮后成功维持了对物体的控制, 平均误差为 10.09 mm。

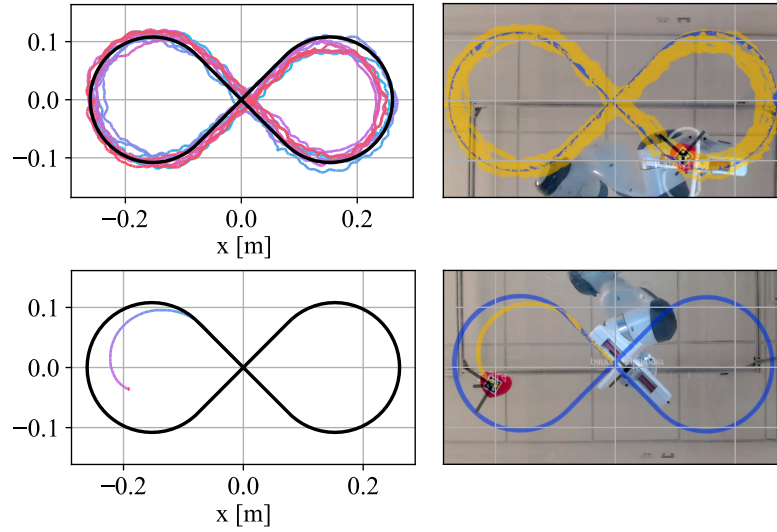


图 17. 使用 ∞ 形轨迹进行鲁棒性验证。该实验使用心形物体 (9) 进行, 由相机 2 记录。上: 由 *Caging in Time* 框架进行的十个循环。下: 一种幼稚推理策略的失败尝试。

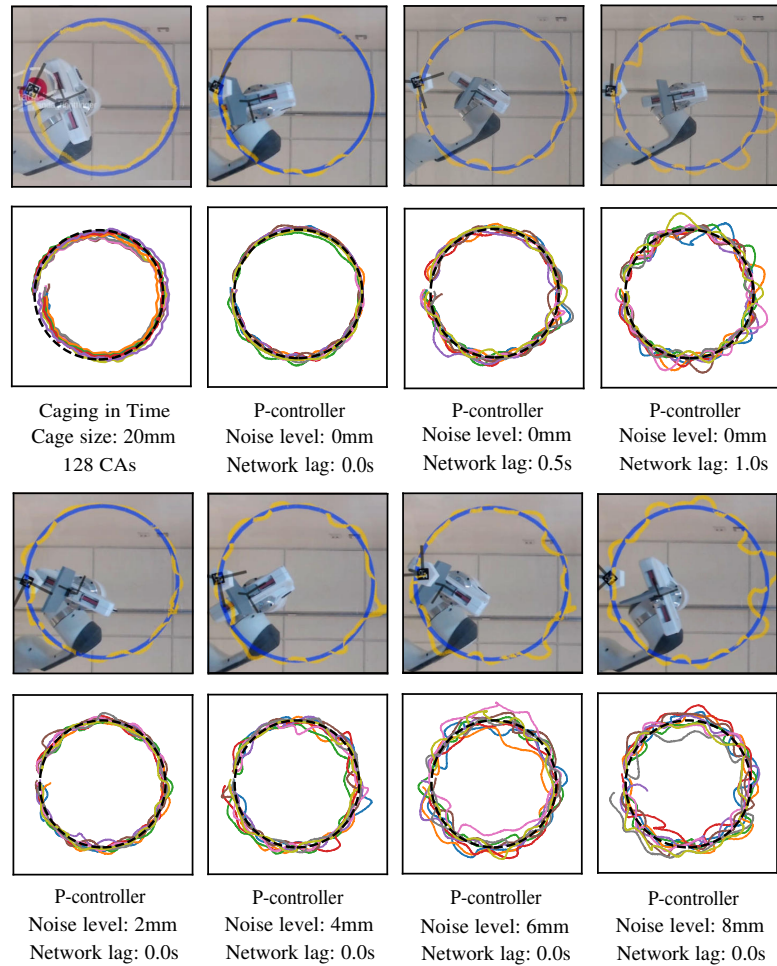


图 18. 与20mm笼子尺寸和128个候选动作的时间笼子 (*Caging in Time*) 在不同位置噪声水平和网络延迟下的P控制器的轨迹比较。奇数行: 单个物体的代表性记录。偶数行: 来自所有九个测试物体的整体轨迹。

7.1.3 Comparison with a Closed-loop Method 为了进一步验证我们框架的有效性,我们进行了与一种闭环推送方法的对比研究:在准静态假设下实现的比例控制器 (P-controller)。P-controller 通过从物体当前位置到参考轨迹上最近的路径点计算误差向量来工作。然后,该误差向量通过 P 增益 (设置为 0.5) 进行缩放,以确定推

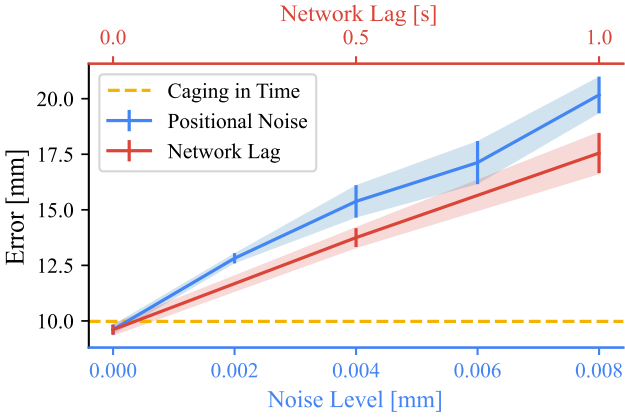


图 19. 在20mm笼大小和128个候选动作下, P控制器在噪声水平(蓝色)和网络延迟(红色)方面的评估结果与Caging in Time框架(黄色)相比。

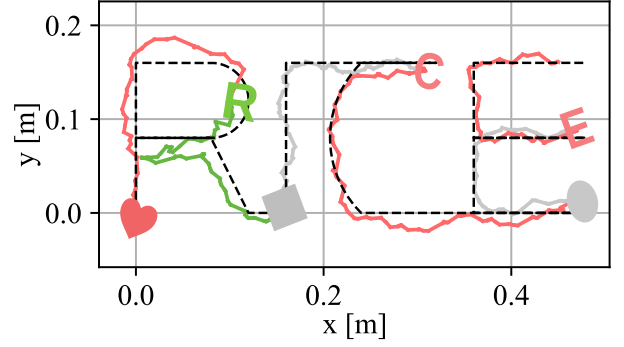


图 20. 随着任务干扰的轨迹跟踪, 使用六种不同的物体(4, 7, 9, R, C, E)来追踪“RICE”。在每个段落的起始点, 操控的物体在切换后显示, 物体的颜色与其轨迹相匹配。真实的记录展示在图 1 中。

送动作的方向和幅度。具体而言, 推送方向与缩放后的误差向量对齐, 而推送距离每次动作限制为 20 mm, 这与我们在 Caging in Time 实验中使用的 d_{push} 一致。

我们引入了两种类型的干扰, 以模拟具有不完美感知和通信延迟的实际场景。首先, 位置噪声作为高斯噪声添加, 噪声水平的标准偏差在 x 和 y 方向上均匀分布, 模拟来自低光照条件下 RGBD 相机的不完美感知效果。其次, 为了模拟数据传输过程中可能发生的数据包丢失或物体从相机视野突然遮挡的情况, 我们引入了网络延迟, 对于每次执行步骤, 控制器有 50% 的概率收到来自 0.5 秒或 1 秒前的物体位置。

实验结果如图 Fig. 18 和 Fig. 19 所示, 证明了我们的 Caging in Time 方法在准确性上与完美感知下的 P-controller 相媲美。此外, P-controller 在感知噪声或网络延迟的影响下, 其准确性显著下降。如图 Fig. 18 所示, 当高斯噪声较大时, 结果轨迹倾向于表现出随机运动模式, 而网络延迟则在轨迹中引发经典的正弦波波动。

这些发现突显了我们的 Caging in Time 框架的稳健性, 特别是在感知和通信往往不完美的实际场景中。我们方法的开放环路特性消除了对持续反馈的需求, 使其本质上对感知噪声和网络延迟具有更强的抵抗力, 后者可能会显著削弱闭环方法的性能。

7.1.4 In-task Perturbations 时间中的笼的鲁棒性 (robustness) 通过沿着字母“RICE”所形状的特别设计轨迹对物体进行操作进一步进行了测试, 如图 20 所示。值得注意的是, 在这项任务中, 物体被人类操作员间歇性地随机替换, 以测试系统在保持单一任务轨迹精度的同时, 适应新形状和位置扰动的能力。结果进一步显示了我们提出的 时间中的笼框架的鲁棒性: 只要新物体被放置在当前的 PSS Q_t 内, 机器人就能够稳定地对其进行笼控 (cage) 和沿参考轨迹进行操作, 证明了该方法的鲁棒性和适应性。

7.2 Dynamic Tasks

为了评估 Caging in Time 在处理各种动态任务中的能力和局限性, 我们进行了系列实验, 使用直径为 6.6cm 的网球和不同尺寸的末端效应器板, 如图 14 所示。

与准静态任务类似, 动态实验同样关注三个关键方面: 1) Caging in Time 能否在没有实时传感反馈的情况下, 实现沿不同轨迹在移动板上稳健地平衡球? 2) 该框架在更极端的动态场景下表现如何, 例如以开放环路方式抓住一个球? 3) Caging in Time 在处理较高维度状态空间的不确定性时有多稳健?

在以下实验中, 板沿着预定义的平移轨迹 $\mathcal{T} = \{\mathbf{x}_p(t)\}_{t=0}^T$ 进行运动。虽然板的轨迹是预设的, 但每个时间步长的板的倾斜角度由我们的框架计算。接着, 采用笛卡尔空间运动控制控制机器人末端执行器以纯开环的方式跟随该轨迹。对于所有动态实验, 我们还使用均值绝对误差 (MAE) 来量化性能。由于板能够准确跟踪目标轨迹, MAE 的计算为在所有时间步长和所有试验中, 球与板中心之间的平均距离。

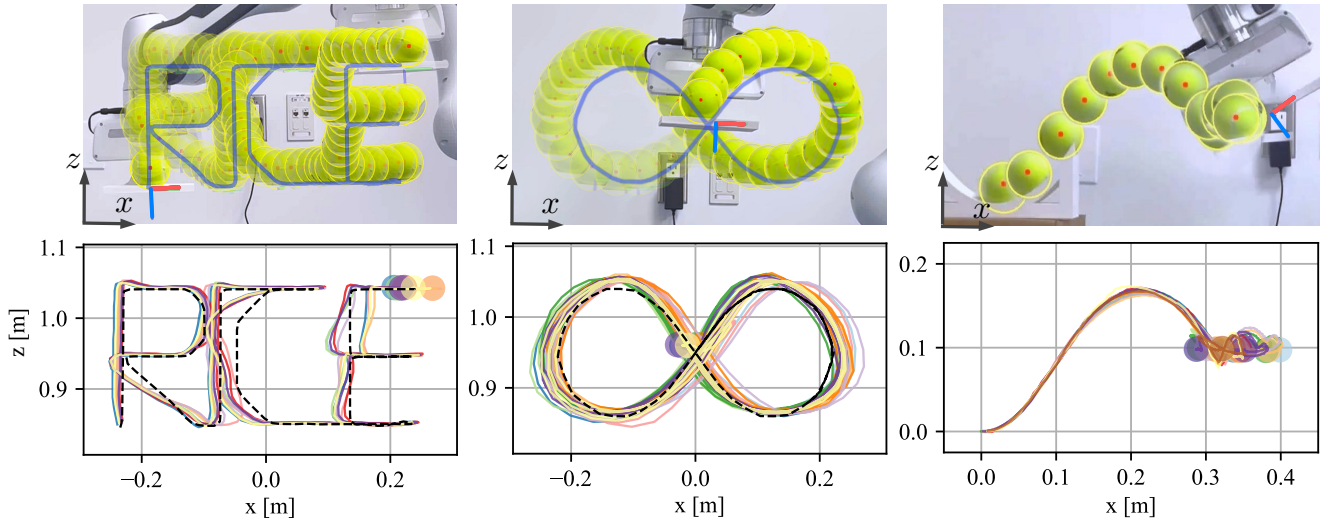


图 21. 实验验证 时间的笼罩 在动态操控任务中的应用, 使用网球和支撑面, 其中 维度 $n = 1$ (参见图 14)。上排: 来自视频录制的代表性帧, 重叠的透明球显示了不同时间点上球的位置。下排: 每个任务的10次重复试验的轨迹图。左: 在绘制“RICE”时的平衡球。中: 沿着 ∞ 形轨迹的平衡球。右: 在斜坡上从同一高度滚下的球接球, 以实现一致的初始投掷速度。

7.2.1 Ball Balancing 我们首先在两个不同的轨迹上测试了球的平衡 维度为 $n = 1$, 如图 21 所示。该任务中使用的平板长16厘米, 并配有导轨以限制球的滚动仅沿 x 轴进行, 如图 14 右下角所示。具有急转弯的“RICE”轨迹导致的平均误差为 20.12 ± 5.66 毫米。 ∞ 形轨迹重复四次以测试平滑的重复曲线, 得出的平均误差为 26.59 ± 7.54 毫米。两个实验均重复了20次, 球始终保持在平板上, 展示了 *Caging in Time* 在长时间动态任务中的鲁棒性和稳定性。

我们观察到, 平板轨迹中的急转弯导致 $\ddot{x}_p$ 的突然变化, 增加了控制难度, 并导致球轨迹的误差增大。此外, 在跟踪 ∞ 形轨迹时, $\ddot{x}_p$ 在大多数时间内为非零, 从而导致持续的非零跟踪误差。这导致 ∞ 形轨迹的平均误差高于“RICE”轨迹, 即使球轨迹的最大位置误差相对较小。

7.2.2 Ball Catching 为了进一步挑战 *Caging in Time* 在处理复杂动态任务中的能力, 我们使用带有1D轨道约束的相同平台实施了一个开放式循环保球实验其中维度 $n = 1$ 。在实际实施过程中, 为了确保启动动作序列的适当时机, 我们利用OpenCV检测球何时进入相机帧。检测到后, 开放式动作被触发, 伴随手动调节的固定时间偏移。在 *Caging in Time* 框架内, 我们认为任务在球与平台接触时开始, 忽略弹跳效果。此外, 我们还加入了硬编码的平移后退动作, 以最小化球在平台上的反弹。

我们进行了两组实验。在第一次实验中, 球从固定高度的倾斜坡道上释放, 确保起始位置一致, 且球离开坡道时的速度大致相同。在20次试验中, 系统在所有20次尝试中成功捕捉到球, 如图 21 (右) 所示。这证明了在明确定义的初始条件下, *Caging in Time* 的可靠性。

在第二个实验中, 我们通过让人手动将球扔到盘子上, 增加了更大的不确定性, 如图 22 和图 1 所示。由于手抛球的着陆位置控制难度固有, 我们只记录了成功落在盘子上的尝试。在20次成功的投掷中, 该系统在13次试验中能够捕捉并稳定球。这个部分成功率展示了该框架在一定范围内对速度不确定性的鲁棒性。捕捉人抛球的部分成功率显示了 *Caging in Time* 框架预设误差容忍度的局限性。这种容忍度受到限制, 以确保在机器人硬件扭矩限制内可行的动作, 而一些人抛的球显然引入了超出该容忍范围的速度变化。

7.2.3 Caging with Uncertainty 为了测试框架对形状不确定性的稳健性, 我们在图 14 所示的 YCB 数据集中平衡了各种水果, 沿着“RICE”轨迹 维度 $n = 1$, 如图 23 所示。尽管形状不确定性给系统动态带来了更高的不确定性, 但该框架仍然能够确保物体运动保持在被限制的 PSS 中。在 10 次实验中, 所有物体都保持在托盘上, 展示了 *Caging in Time* 对形状变化的适应性。

值得注意的是, 这些任务的成功部分归因于不规则形状的特性, 由于其非均匀几何形状所产生的固有能量陷阱, 使其不太可能自由滚动。这一特性实际上有助于维持稳定性, 因为物体倾向于沉降到局部能量最小值, 从而补充了我们框架的限制策略。

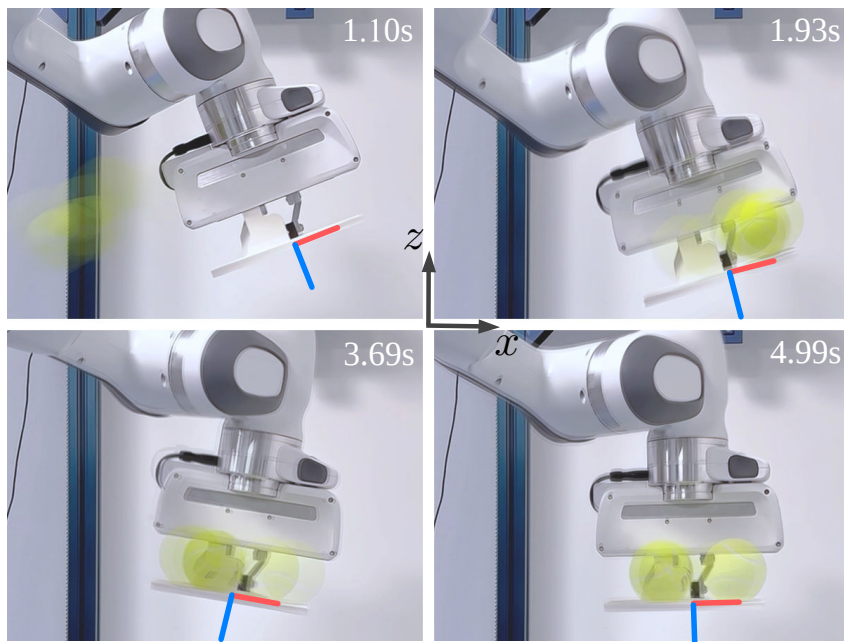


图 22. 使用带时间标签的9个叠加试验的代表性帧示教（demonstration）人类投掷的接球动作。结果与图 1 一起显示，*Caging in Time* 能够使用图 21 中相同的开放式动作（action）序列来接住人类不准确投掷的球。

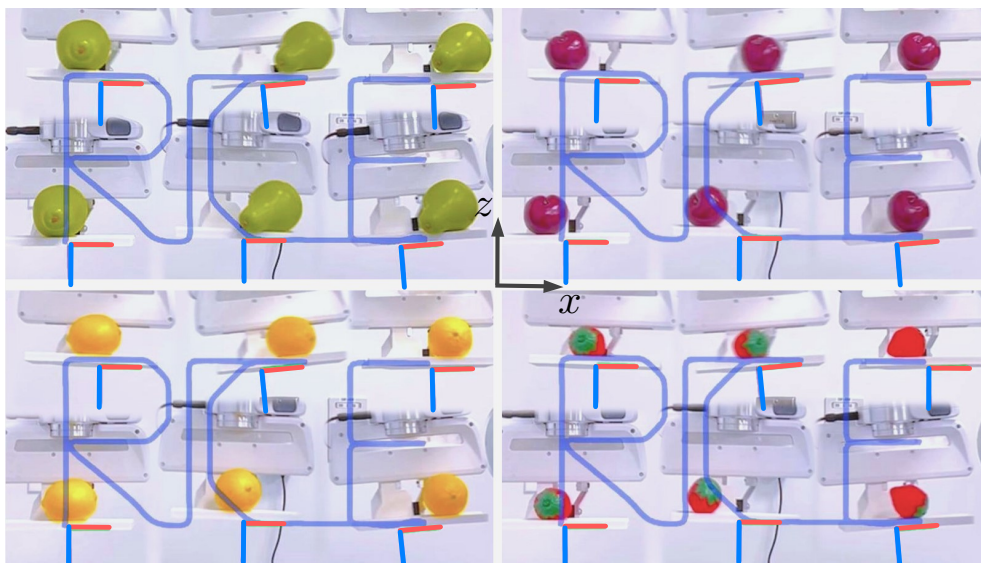


图 23. 在*Caging in Time*中的对象无关鲁棒性示教（demonstration）。使用相同的开环动作（action）序列平衡来自图14的多样形状YCB水果其中维度 $n = 1$ 。每个图形展示了同一种水果的多个实例,表示在平衡任务中不同时间点的其位置。

7.2.4 Caging in Higher Dimensions 最后，我们将实验扩展到高维空间，使用一个 $16\text{cm} \times 16\text{cm}$ 的平板，在没有导轨的情况下，允许球在X和Y方向上自由移动，这显著增加了平衡任务的复杂性。如图 24 所示，我们进行了球平衡实验 维度为 $n = 2$ ，其中球的状态在一个4D状态空间 $(x_t, y_t, \dot{x}_t, \dot{y}_t)$ 中表示，比之前所有任务的维度都高两个维度。

在10次试验中，球始终保持在平板上，展示了*Caging in Time*在高维场景中的适用性。图 24中显示的轨迹仅用于说明，由于相机的非正交放置，可能不代表精确的定量表现，我们可以明显看出这些轨迹相较于之前的平衡实验表现出更大的偏差。这种增加的误差归因于此设置中无限制的滚动方向，引入了更大的不确定性和控制挑战。

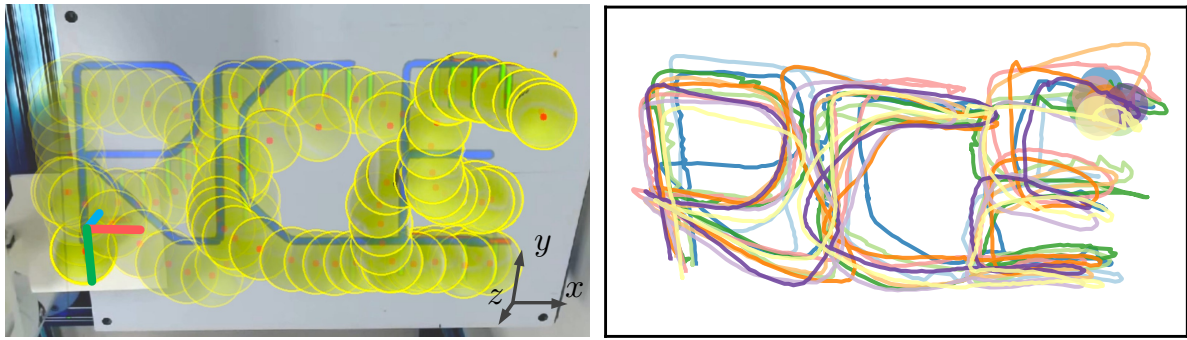


图 24. 使用时间中的围合进行球体平衡，维度为 $n = 2$ ，同时追踪“RICE”。左：视频记录中的代表性帧。叠加的透明球体显示了在任务执行过程中不同时间点的球体位置。右：每个任务 10 次重复试验的定性轨迹图。

8 Conclusion

在这项工作中，我们提出并评估了 *Caging in Time*（时间笼因素），这是一种用于稳健物体操作的新理论。我们的框架在准静态和动态任务中均表现出高精度，而无需详细的物体信息或实时反馈。严格的评估凸显了该框架对各种物体和动态场景的韧性和适应性。*Caging in Time* 方法在处理新物体、位置扰动和具有挑战性的动态任务方面证明了其有效性，展示了在不确定环境（environment）中可靠操作的潜力。

尽管当前的 *Caging in Time* 框架显示出了良好的结果，但重要的是要意识到其局限性。目前，该框架需要手动定义特定任务的参数和 PSS 传播（propagation）函数。这种对人类专业知识的依赖可能限制了其在更广泛操作任务中的通用性。此外，当前的方法可能无法完全捕捉某些现实场景中物体交互和环境因素极具不可预测性的复杂性。

展望未来，我们旨在解决这些局限性并拓展 *Caging in Time* 的视野。未来研究的一个关键方向是整合学习技术和大型语言模型（Large Language Models, LLMs），以从日常场景中自主发现和学习 *Caging in Time* 任务，这可能使框架能够自动推导出适当的传播函数和策略。我们还计划探索其在多物体场景和更复杂案例中的应用，潜在地利用机器学习来处理增加的任务复杂性和环境变异性。

随着 *Caging in Time* 理论的发展，形成越来越通用的算法来处理愈加复杂的任务动态，我们预计能够实现许多有趣而稳健的现实应用。这些应用可能包括利用柔性机器人手进行手中操作、通过对抗环境约束进行杂物中的机器人拾取，以及在没有固定抓取的情况下动态运输大型物体。该框架在处理准静态和动态任务方面所展示的能力为多种机器人操作场景中的稳健开环控制策略打开了新的可能性。

Acknowledgements

This work was supported by the National Science Foundation under grant FRR-2240040.

References

- Ames AD, Coogan S, Egerstedt M, Notomista G, Sreenath K and Tabuada P (2019) Control barrier functions: Theory and applications. In: *2019 18th European Control Conference (ECC)*.
- Anand A, Seel K, Gjerum V, Håkansson A, Robinson H and Saad A (2021) Safe learning for control using control lyapunov functions and control barrier functions: A review. *Procedia Computer Science* 192: 3987–3997.
- Andrychowicz OM, Baker B, Chociej M, Jozefowicz R, McGrew B, Pachocki J, Petron A, Plappert M, Powell G, Ray A et al. (2020) Learning dexterous in-hand manipulation. *The International Journal of Robotics Research* 39(1): 3–20.
- Billard A and Kragic D (2019) Trends and challenges in robot manipulation. *Science* 364(6446).
- Bircher WG, Morgan AS and Dollar AM (2021) Complex manipulation with a simple robotic hand through contact breaking and caging. *Science Robotics* 6(54): eabd2666.
- Blanchini F (1995) Nonquadratic lyapunov functions for robust control. *Automatica* 31(3): 451–461.
- Bohg J, Hausman K, Sankaran B, Brock O, Kragic D, Schaal S and Sukhatme GS (2017) Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics* 33(6): 1273–1291.

- Bütepage J, Cruciani S, Kokic M, Welle M and Kragic D (2019) From visual understanding to complex object manipulation. *Annual Review of Control, Robotics, and Autonomous Systems* 2: 161–179.
- Dong Y, Cheng X and Pokorný FT (2024) Characterizing manipulation robustness through energy margin and caging analysis. *IEEE Robotics and Automation Letters* 9(9): 7525–7532.
- Dong Y and Pokorný FT (2024) Quasi-static soft fixture analysis of rigid and deformable objects. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 6513–6520.
- Hang K, Bircher WG, Morgan AS and Dollar AM (2021) Manipulation for self-identification, and self-identification for better manipulation. *Science Robotics* 6(54): eabe1321.
- Jost J (2008) *Riemannian geometry and geometric analysis*, volume 42005. Springer.
- Kaelbling LP (2020) The foundation of efficient robot learning. *Science* 369(6506): 915–916.
- Kroemer O, Niekum S and Konidaris G (2021) A review of robot learning for manipulation: Challenges, representations, and algorithms. *Journal of machine learning research* 22(30): 1–82.
- Lee MA, Zhu Y, Srinivasan K, Shah P, Savarese S, Fei-Fei L, Garg A and Bohg J (2019) Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks. In: *IEEE International Conference on Robotics and Automation (ICRA)*. pp. 8943–8950.
- Makapuno T, Phoka T, Pipattanasomporn P, Niparnan N and Sudsang A (2012) Measurement framework of partial cage quality. In: *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. pp. 1812–1816.
- Makita S and Nagata K (2015) Evaluation of finger configuration for partial caging. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 4897–4903.
- Makita S and Wan W (2018) A survey of robotic caging and its applications. *Advanced Robotics* 31(19): 1071–1085.
- Olson E (2011) Apriltag: A robust and flexible visual fiducial system. In: *IEEE International Conference on Robotics and Automation (ICRA)*. pp. 3400–3407.
- Pereira GAS, Campos MFM and Kumar V (2004) Decentralized algorithms for multi-robot manipulation via caging. *The International Journal of Robotics Research* 23(7-8): 783–795.
- Peshkin M and Sanderson A (1988) The motion of a pushed, sliding workpiece. *IEEE Journal on Robotics and Automation* 4(6): 569–598.
- Rodriguez A (2021) The unstable queen: Uncertainty, mechanics, and tactile feedback. *Science Robotics* 6(54): 4667.
- Rodriguez A, Mason MT and Ferry S (2012) From caging to grasping. *The International Journal of Robotics Research* 31(7): 886–900.
- Song H, Varava A, Kravchenko O, Kragic D, Wang MY, Pokorný FT and Hang K (2021) Herding by caging: a formation-based motion planning framework for guiding mobile agents. *Autonomous Robots* 45(5): 613–631.
- Stork JA, Pokorný FT and Kragic D (2013) A topology-based object representation for clasping, latching and hooking. In: *IEEE International Conference on Humanoid Robots (HUMANOIDS)*. pp. 138–145.
- Sudsang A and Ponce J (2000) A new approach to motion planning for disc-shaped robots manipulating a polygonal object in the plane. In: *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2. pp. 1068–1075.
- Suomalainen M, Karayiannidis Y and Kyrki V (2022) A survey of robot manipulation in contact. *Robotics and Autonomous Systems* 156: 104224.
- Varava A, Carvalho JF, Kragic D and Pokorný FT (2021) Free space of rigid objects: caging, path non-existence, and narrow passage detection. *The International Journal of Robotics Research* 40(10): 1049–1067.
- Varava A, Kragic D and Pokorný FT (2016) Caging grasps of rigid and partially deformable 3-d objects with double fork and neck features. *IEEE Transactions on Robotics* 32(6): 1479–1497.
- Welle MC, Varava A, Mahler J, Goldberg K, Kragic D and Pokorný FT (2021) Partial caging: a clearance-based definition, datasets, and deep learning. *Autonomous Robots* 45(5): 647–664.
- Yuan W, Dong S and Adelson E (2017) Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors* 17(12): 2762.
- ZhiDong W and Kumar V (2002) Object closure and manipulation by multiple cooperating mobile robots. In: *IEEE International Conference on Robotics and Automation (ICRA)*. pp. 394–399.

Appendix

8.1 PSS Propagation for Quasi-static Pushing

如图 8 所示, 假设线性推杆足够长, 我们将 $d_{con} \leq d_{push}$ 表示为线性推杆与物体边界圆接触后的移动距离。假设物体的运动是准静态的, 我们可以应用能量守恒和力平衡的原理来界定物体的位移。力平衡条件表明摩擦力等于推力, 即 $F_{friction} = F_{push}$ 。推力所做的功 W_{push} 部分被平移摩擦能量 E_{loss} 吸收, 因此我们有 $W_{push} = F_{push} \cdot d_{con} \geq F_{friction} \cdot \|\Delta q_t\| = E_{loss}$ 。这为物体位移的大小提供了界限 $\|\Delta q_t\| \leq d_{con}$ 。

此外, 通过使用 Peshkin 距离 (Peshkin and Sanderson 1988) 的进一步分析, 我们可以进一步界定物体的侧位移 $\|\Delta q_t\|$, 该侧位移是物体相对于线性推杆的平行位移。

8.1.1 The Peshkin Distance 佩希金距离定义了为实现物体边缘与围栏对齐所需的最小平移位移。这个距离是物体旋转中心的一个函数, 对应于导致最小旋转速度的粘附条件。对于重新定向过程, 如图 25 所示, 在粘附-最慢旋转场景下, 佩希金距离 m 的计算方式为:

$$m = \frac{a^2 + c^2}{2c} \left(\ln \left| \frac{1 - \cos \beta_1}{1 + \cos \beta_1} \right| - \ln \left| \frac{1 - \cos \beta_0}{1 + \cos \beta_0} \right| \right), \quad (27)$$

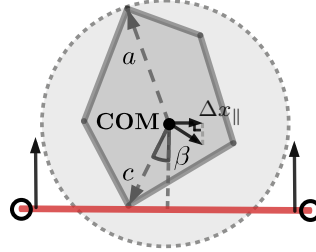


图 25. 与 Peshkin 距离计算相关的符号说明。Peshkin 距离将物体的侧向位移限制在 $m/2$, 其中 m 表示推动距离 (与 Sec. 5.2 中的 d_{con} 相同)。

其中 a 代表以质心为中心的物体外接圆的半径, c 是质心到接触点的距离, β 是运动方向与连接接触点和质心的直线之间的夹角。角度 β_0 和 β_1 对应于此旋转的初始和最终方向角。

为了逆向工程这一过程并计算在固定推力距离 m (即在第 5.2 节中的 d_{con}) 下 β 的最大变化 (记作 $\Delta\beta$), 使用以下推导:

$$e^{\frac{cm}{a^2+c^2}} = \frac{\tan \frac{\beta_1}{2}}{\tan \frac{\beta_0}{2}} = \frac{\tan \frac{\beta_0+\Delta\beta}{2}}{\tan \frac{\beta_0}{2}} \approx 1 + \csc \beta_0 \Delta\beta \quad (28)$$

假设推送距离 m 远小于 a 和 c , 我们进行近似:

$$e^{\frac{cm}{a^2+c^2}} \approx 1 + \frac{cm}{a^2+c^2} \quad (29)$$

因此, β 的最大变化可以近似为:

$$\Delta\beta = \frac{c \sin \beta_0}{a^2 + c^2} m \quad (30)$$

这些推导为预测平面推动交互作用对物体的旋转效应提供了数学基础, 假设接触点没有滑动。

我们进一步可以得到侧位移 $\Delta x_{\parallel}$ (对应于 $\|\Delta q_t\|$) 的界限:

$$\begin{aligned} \Delta x_{\parallel} &= c \Delta\beta_0 \cos \beta_0 \\ &= \frac{mc^2}{a^2 + c^2} \sin \beta_0 \cos \beta_0 \\ &\leq m \sin \beta_0 \cos \beta_0 \leq \frac{m}{2} \end{aligned} \quad (31)$$

因此,侧向位移 $\|\Delta q_t\|$ 被限制在 $\|\Delta q_t\| \leq d_{con}/2$ 。这个界限加上另一个界限 $\|\Delta q_t\| \leq d_{con}$ 约束了对象 $\mathcal{V}_{q_t}$ 所有可能的位移,使其位于图 8 右侧深绿色区域所示的半椭圆内部,数学表达式如方程 (32) 所示。

总之,将对象的包围圆与推动器发生碰撞的情况以及不发生碰撞的情况结合起来,特定配置 q_t 的所有可能运动集合由以下方程给出,即方程 (6) 中定义的抽象函数 U 的显式表达。

$$\begin{aligned} \mathcal{V}_{q_t} &= U(q_t, u_t) = \\ &\text{if } \mathbf{d}(q_t, \mathcal{L}) > r + d_{push} \text{ or } u_t \text{ is None:} \\ &\quad \{(0, 0)\} \\ &\text{otherwise:} \\ &\quad \left\{ v_t = R(u_t) \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \mid \frac{(\Delta x)^2}{d_{con}^2} + \frac{(\Delta y)^2}{\left(\frac{d_{con}}{2}\right)^2} \leq 1, \Delta x \leq 0 \right\} \end{aligned} \quad (32)$$

在方程中, $R(u_t) \in SO(2)$ 是 u_t 的旋转矩阵 (rotation matrix)。

8.2 Dynamic Control Implementation Details

在这里,我们提供有关动态控制策略 (dynamic control strategy) 实施的更多细节。

8.2.1 System Dynamics 倾斜板上球体的系统动力学可以通过以下方程描述:

$$\begin{aligned} \ddot{\mathbf{x}}_t &= f(\mathbf{q}_t, u_t, \eta_m, \boldsymbol{\eta}_p, \eta_\mu) \\ &= \frac{m(1 + \eta_m)}{m + \frac{I_b}{r_b^2}} \left((I_{n+1} - \mathbf{n}(\boldsymbol{\theta})\mathbf{n}(\boldsymbol{\theta})^T) \times (\mathbf{g}_\theta + \mathbf{a}_p + \boldsymbol{\eta}_p) \right) - (\mu_r + \eta_\mu)\dot{\mathbf{x}}_t \end{aligned} \quad (33)$$

其中 I_{n+1} 是单位矩阵, μ_r 是估计的滚动摩擦系数, $\mathbf{n}(\boldsymbol{\theta}) \in \mathbb{R}^{n+1}$ 是作为 $\boldsymbol{\theta}$ 的函数的板的法向量。参数 m 、 I_b 和 r_b 分别代表质量、转动惯量和球的半径。 $\mathbf{g}_\theta$ 是板框架中的重力加速度向量, $\mathbf{a}_p$ 是板的加速度。

术语 $\eta_m \in \mathbb{R}$ 、 $\boldsymbol{\eta}_p \in \mathbb{R}^{n+1}$ 和 $\eta_\mu \in \mathbb{R}$ 分别表示考虑质量、板加速度和摩擦系数不确定性的随机分布。这些不确定性遵循高斯分布:

$$\begin{aligned} \eta_i &\sim \mathcal{N}(0, \sigma_i^2), \quad i \in \{m, \mu\} \\ \boldsymbol{\eta}_p &\sim \mathcal{N}(\mathbf{0}, \Sigma_p) \end{aligned} \quad (34)$$

其中 $\Sigma_p \in \mathbb{R}^{(n+1) \times (n+1)}$ 是协方差矩阵。

为了推导球加速度的概率分布 $p(\ddot{\mathbf{x}}_t)$, 我们考虑系统动力学方程作为随机变量 η_m , $\boldsymbol{\eta}_p$ 和 η_μ 的函数。鉴于这些不确定性遵循高斯分布, $\ddot{\mathbf{x}}_t$ 也可以近似为高斯分布。我们可以将这个分布表示为:

$$\ddot{\mathbf{x}}_t \sim \mathcal{N}(\boldsymbol{\mu}_{\ddot{\mathbf{x}}}, \boldsymbol{\Sigma}_{\ddot{\mathbf{x}}}) \quad (35)$$

在这里, $\boldsymbol{\mu}_{\ddot{\mathbf{x}}}$ 是平均加速度向量, $\boldsymbol{\Sigma}_{\ddot{\mathbf{x}}}$ 是协方差矩阵。平均值 $\boldsymbol{\mu}_{\ddot{\mathbf{x}}}$ 通过在系统动态方程中将所有不确定性设定为其期望值 (零) 获得。

对于特殊情况 $n = 1$, 我们可以将系统动态重新制定为控制仿射形式, 以便利于控制屏障函数 (Control Barrier Functions, CBFs) 和控制李雅普诺夫函数 (Control Lyapunov Functions, CLFs) 的应用:

$$\ddot{\mathbf{x}}_t = \mathbf{f}_t + \mathbf{g}_t u \quad (36)$$

where:

$$\begin{aligned} \mathbf{f}_t &= \frac{m}{m + \frac{I_b}{r_b^2}} (1 + \eta_m) ((g + (\ddot{z}_p + \eta_z)) \sin \theta + (\ddot{x}_p + \eta_x) \cos \theta) - (\mu_r + \eta_\mu) \dot{\mathbf{x}}_t \\ \mathbf{g}_t &= \frac{m}{m + \frac{I_b}{r_b^2}} (1 + \eta_m) ((g + (\ddot{z}_p + \eta_z)) \cos \theta - (\ddot{x}_p + \eta_x) \sin \theta) \end{aligned} \quad (37)$$

Algorithm 6: DynamicControl($\cdot$)**Input:** Initial state Q_0 , desired trajectory $\mathcal{T} = \{\mathbf{x}_p(t)\}_{t=0}^T$, time step Δt **Output:** Success or failure of the task

```

1 for  $t = 0, \dots, T - 1$  do
2    $S(Q_t) \leftarrow \text{EstimateEntropy}(Q_t)$  ▷ Eq. (39)
3   Compute  $E_{max}(\mathbf{g}_{\theta_t}, \mathbf{a}_p)$  and  $E(\mathbf{q}_t, \mathbf{g}_{\theta_t}, \mathbf{a}_p)$  for all states in  $Q_t$  ▷ Eq. (11) (12)
4   Calculate  $L_f h, L_g h, L_f V, L_g V$  ▷ Eq. (40)
5    $d\theta \leftarrow \text{SolveQP}(L_f h, L_g h, L_f V, L_g V)$  ▷ Eq. (26)
6    $\theta_{t+1} \leftarrow \theta_t + d\theta$ 
7    $Q_{t+1} \leftarrow f(Q_t, \theta_{t+1})$  ▷ Eq. (16)
8   if  $\max_{\mathbf{q}_t} E(\mathbf{q}_t, \mathbf{g}_{\theta_{t+1}}, \mathbf{a}_p) > E_{max}(\mathbf{g}_{\theta_{t+1}}, \mathbf{a}_p)$  then
9     return false
10  end
11 end
12 return true

```

这里, $\eta = \{\eta_m, \eta_x, \eta_z, \eta_\mu\}$ 表示用于考虑系统不确定性的随机分布。这种表征将动力学分为漂移项 f_t (没有控制输入的系统行为) 和控制项 $g_t u$ 。

这种形式简化了 CBFs 和 CLFs 的 Lie 导数的计算。它允许通过 η 项对不同系统组件中的不确定性进行明确表征, 并解耦控制输入 $u = d\theta$, 便于分析控制变化如何影响系统动力学。

在实践中, 这些不确定性的方差是手动确定的。较大的方差表示对错误的容忍度更高, 但也增加了找到可行动作的难度。

8.2.2 CBF and CLF The CLF is defined as:

$$V(\mathbf{x}) = \sum_{\mathbf{q}_t} \mathcal{I}_t(\mathbf{q}_t) E(\mathbf{q}_t, \mathbf{g}_{\theta_t}, \mathbf{a}_p) - k_S S(\mathbf{x}) \quad (38)$$

其中 $\mathcal{I}_t(i, j)$ 是离散概率分布中网格点 (i, j) 的概率, $S(\mathbf{x})$ 是系统熵。CLF 选定为系统能量和负熵的加权和, 以平衡减少能量消耗和维持良好分布概率状态的目标, 确保在不确定性面前的稳定性和鲁棒性。

熵 $S(Q_t)$ 计算如下:

$$S(Q_t) = - \sum_{\mathbf{q}_t} \mathcal{I}_t(\mathbf{q}_t) \log \mathcal{I}_t(\mathbf{q}_t) \quad (39)$$

在以下情况下计算 CBF (控制屏障函数, Control Barrier Function) 和 CLF (控制律函数, Control Lyapunov Function) 的李导数 (Lie derivatives):

$$\begin{aligned} L_f h(\mathbf{x}) &= \frac{\partial h}{\partial \mathbf{x}} f(\mathbf{x}), L_g h(\mathbf{x}) = \frac{\partial h}{\partial \mathbf{x}} g(\mathbf{x}) \\ L_f V(\mathbf{x}) &= \frac{\partial V}{\partial \mathbf{x}} f(\mathbf{x}), L_g V(\mathbf{x}) = \frac{\partial V}{\partial \mathbf{x}} g(\mathbf{x}) \end{aligned} \quad (40)$$

其中 $\frac{\partial h}{\partial \mathbf{x}}$ 和 $\frac{\partial V}{\partial \mathbf{x}}$ 分别是关于状态 $\mathbf{x}$ 的 h 和 V 的梯度。

在实际应用中, 由于 h 和 V 的表达式的复杂性, 这些导数通常通过数值方法进行近似。

8.2.3 Numerical Implementation 控制策略的数值实现概述在算法 6 中。

该算法通过反复求解 QP (Quadratic Programming) 并更新系统状态 (state) 来实现控制策略。它确保小球在基于能量的笼子内保持, 同时遵循期望的轨迹。