基于凸集图的时序逻辑运动规划与凸优化

Vince Kurtz*IEEE* 学生会员 和 Hai Lin *IEEE* 高级会员

arXiv:2301.07773v2 [cs.RO] 1 Jun 2023

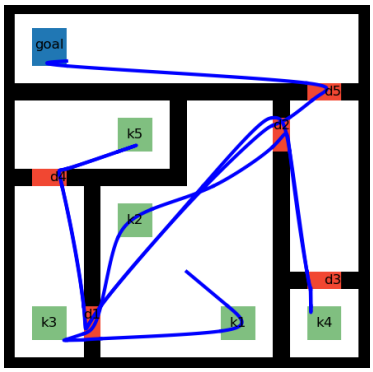
摘要——**时序逻辑是一种简洁描述复杂任务的方式。然而，要实现时序逻辑规范所要求的运动规划并不容易，现有方法难以扩展到复杂规范和具有高维系统动力学的系统。本文将线性时序逻辑（LTL）运动规划转化为凸集图（GCS）中的最短路径问题，并通过凸优化求解。该方法结合了现代基于优化的时序逻辑规划器与较早的自动机理论方法的优点，克服了各自的局限性：通过用连续贝塞尔曲线表示路径，避免了裁剪和直通；计算复杂度关于采样点数量呈多项式增长（而非指数增长）；可以认证全局最优性（但不保证达到全局最优）；在温和的假设条件下，可保证可靠性和概率完备性；最重要的是，该方法能够扩展到复杂规范和高维系统，包括一台30自由度人形机器人。开源代码见 https://github.com/vincekurtz/ltl_gcs。**

I. 引言

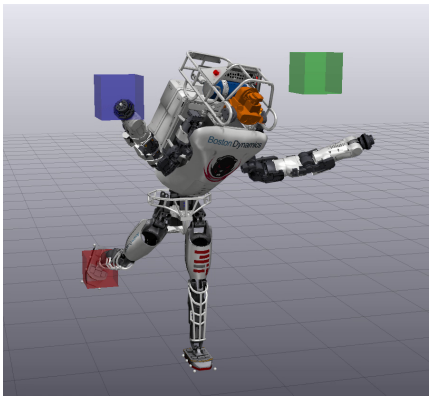
机器人和信息物理系统通常需要完成的不只是避开障碍物或在设定点附近调节。为此，线性时序逻辑（LTL）、度量时序逻辑（MTL）和信号时序逻辑（STL）等时序逻辑日益普及，成为表达复杂控制目标的简洁方式。除了兼具表达力和简洁性之外，这些逻辑也相对容易为人理解，因为它们结合了熟悉的布尔运算符（“与”“或”“非”）以及名称直观的时序运算符（“下一步”“始终”“最终”）。时序逻辑的实用性在图 1a 所示的情形中尤为明显：移动机器人在拾取相应的钥匙（绿色）之前不得通过某扇门（红色），并且最终必须到达目标（蓝色）。

时序逻辑运动规划是一个本质上很难（NP难）的问题，对于高自由度（DoF）系统尤其具有挑战性。目前最先进的方法是采用混合整数凸规划（MICP）[1]。MICP 方法是可靠且完备的（如果存在解），但它有一些显著的缺点。具体而言，标准的混合整数凸规划编码会为每个时间步上的每个子公式引入二元变量。由于混合整数凸规划的最坏情况复杂度随二元变量数量呈指数增长，因此无论是规范的复杂度还是用于表示路径 [1, 2] 的采样点数量增加，混合整数凸规划的扩展性都较差。

谨此感谢美国国家科学基金会的资助（资助编号：CNS-1830335、IIS-2007949）。Vince Kurtz 和 Hai Lin 均任职于美国印第安纳州圣母镇圣母大学电气工程系。

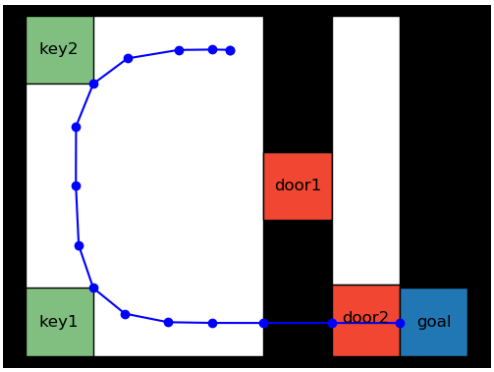


(a) 来自 [3]，的一项基准测试：移动机器人必须先拾取钥匙（$k1$，$k2$，.....），再穿过门（$d1$，$d2$，.....），并到达目标。已报告的最快求解时间为 **49.5 秒**，对应一个分段线性解 [4]。我们提出的方法在**5.8 秒**内找到了一个 $\mathcal{C}^2$ 全局最优解。

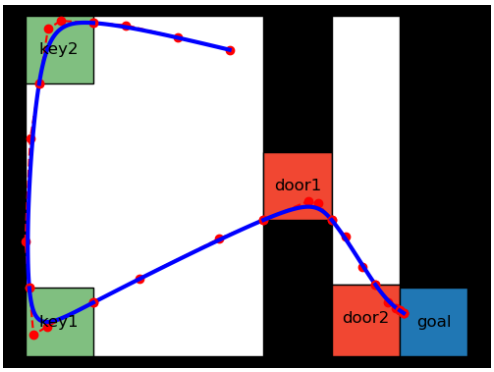


(b) 一台 30自由度人形机器人需要先用左手触碰绿色目标，再用右脚触碰红色目标，最后用右手触碰蓝色目标。凸优化可在不到 10 秒内求得一个平滑的构型空间解。

图 1：我们提出的方法可扩展至复杂任务规范 (a) 和高维系统 (b)。



(a) 标准混合整数凸规划



(b) 我们的方法

图 2：移动机器人必须先拾取两把钥匙，才能穿过对应的门并抵达目标区域，如公式 (10) 所示。标准混合整数凸规划 (2a) 使用固定的时间离散化，导致路径穿过障碍物。我们提出的方法 (2b) 考虑贝塞尔样条上的所有取值，从而避免裁剪和穿越。

标准混合整数凸规划编码所采用的时间离散化会加剧这一问题。较粗的离散化会导致“裁剪”现象，即路径可能在时间步之间与障碍物相交，从而违反规范。图 2a 展示了一个示例。采用更精细的离散化可以缓解裁剪问题，但会使计算成本呈指数级增长 [1]。

鉴于该问题是 NP难的，随着规范复杂度增加而产生指数级开销不可避免；但随着采样点数量增加而产生指数级开销是否也不可避免，则尚不明确。直观而言，使用更精细的离散化来规划并满足同一规范，似乎不会从根本上增加问题的难度。事实上，我们证明可以避免这种指数复杂度：我们提出的方法的计算规模随采样点数量呈多项式增长。

我们的核心思路是将线性时序逻辑运动规划重新表述为凸集图（GCS）中的最短路径问题[5]。由此得到的混合整数凸规划具有非常紧的凸松弛——事实上，它足够紧，通常可以通过凸优化和舍入求得全局最优解[6]。即使该混合整数凸规划的近似解并非最优（已有次优性界 [6]，而且在实践中，凸优化通常能找到全局最优解），任何整数可行解都保证满足该规范。

我们提出的方法能够很好地适用于复杂规范和高维构型空间，在众多基准问题上优于最先进水平（见第七节）。此外，我们使用贝塞尔样条对运动规划进行参数化，从而能够设计出不会发生裁剪或穿越的平滑路径。

我们的主要贡献总结如下：

- 我们证明，线性时序逻辑运动规划可以表述为凸集图中的最短路径问题，并可通过凸规划高效求解。
- 通过使用平滑的贝塞尔样条表示路径，我们避免了大多数离散时间时序逻辑公式所带来的裁剪和穿越问题。
- 与标准方法的指数复杂度不同，我们提出的方法的计算复杂度随用于表示平滑路径的控制点数量呈多项式增长。
- 我们给出了可靠性和概率完备性的证明。
- 我们通过若干仿真示例，展示了所提出方法在复杂规范和高维构型空间中的可扩展性，并提供了可复现这些结果的开源代码 [7]。

本文其余部分安排如下：第二节回顾时序逻辑运动规划的相关工作。第三节给出形式化的问题描述和相关定义。第四节介绍贝塞尔曲线和凸集图的背景知识。第五节阐述我们的主要成果——一种时序逻辑运动规划的凸规划求解方案。第六节证明所提方法的可靠性、完备性和计算复杂度。第七节给出示例，第八节讨论所提方法的局限性，最后在第九节总结全文。

II. 相关工作

早期基于时序逻辑的控制研究主要采用自动机理论方法 [3, 8]。这些方法通常假设系统可以建模为有限状态转移系统。将给定的线性时序逻辑公式转换为等价的自动机后，在转移系统与自动机的积上进行简单的图搜索，即可找到一条满足路径

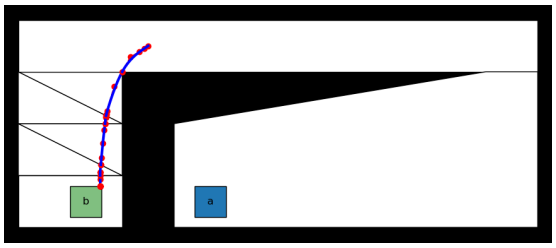


图 3：移动机器人需要最终到达区域 a（蓝色）或区域 b（绿色），如公式（9）所示。标准的基于抽象的方法可能会选择通往 a 的较长路径，因为这条路径经过的区域较少。我们提出的方法则能直接找到通往 b 的较短路径。

[8, 第 5] 章。对于有限状态转移系统，这类基于自动机的方法是可靠、完备且计算高效的。类似技术已在形式化验证和模型检测应用中得到广泛采用 [9]。

将基于自动机的方法应用于运动规划并非易事，因为我们必须在（非有限的）构型空间中进行规划。文献中的主流思路是采用某种有限转移系统抽象 [3, 10, 11],，通常与构型空间中的带标签区域相关联。一旦找到满足条件的抽象路径，低层运动规划器便会搜索一条经过相应区域的路径。尽管这种方法在计算上可能很高效，因为它将逻辑约束与物理动力学分离开来，但要考虑场景几何与抽象转移系统之间的差距并不容易。实际而言，这意味着抽象路径在转化为运动规划后，可能在动力学上不可行或效率低下：抽象中的最短路径不一定对应构型空间中的最短路径，如图 3 所示。

在先前的工作中，我们尝试通过一种受反例引导的归纳综合（CEGIS）[12] 启发的规划与重新规划循环，弥补这一空白，该循环同时用于高层基于自动机的规划器和低层运动规划器 [10, 11]。尽管这种方法能够提供完备性保证，但即使实现局部最优性也很困难，而且可能需要进行多轮重新规划。

许多基于自动机的方法的另一个缺点，源于它们依赖基于采样的运动规划 [10，13，14，15]。尽管这类方法通常具有良好的可扩展性，但在处理包含狭窄通道 [16] 的问题时往往表现不佳，而且难以保证动力学可行性。此外，由于基于采样的方法无法提供不可行性证书，因此往往难以判断某个规范是否不可行。

一类特别有趣的基于自动机的规划器不会搜索具体的满足路径，而是生成能够保证规范 [17，18] 闭环满足的反馈控制器。尽管这些方法提供了一种优雅的方式来弥合连续动力学与离散规范之间的差距，但它们的可扩展性受限于这样一个事实：它们需要在整个状态空间中搜索解。据我们所知，这类技术尚无法扩展到超过 4 维的状态空间。

尽管存在这些局限性，本文仍深受自动机理论方法的启发。具体而言，我们使用 GCS 框架，在转移系统与自动机的积上执行图搜索。与现有的基于抽象的方法不同，这种图搜索直接考虑场景的几何特征，无需分别设置低层和高层规划器。与基于采样的方法不同，我们的 GCS 方法能够轻松通过狭窄通道，并确保微分平坦系统的动力学可行性 [19]。与搜索反馈控制器的方法不同，我们的方法可扩展至高维系统，包括 30自由度人形机器人。

为了解决基于抽象的方法的局限性，研究者开发了采用混合整数规划的基于优化的方法 [1]。混合整数凸规划方法通常不需要任何抽象，而是直接考虑状态（或构型）空间。连续变量表示路径上离散时间步的状态。通过添加二元变量和额外约束来满足规范 [20, 1]。混合整数凸规划方法是可靠且完备的，并且（与基于自动机的方法不同）能够保证全局最优性。混合整数凸规划在 STL [20, 2]，中尤为流行，但也已应用于 LTL [21]，、MTL [2], 及其他多种时序逻辑。混合整数凸规划方法被广泛认为是时序逻辑运动规划领域的最先进水平 [1]。

尽管如此，混合整数凸规划方法仍有一些显著局限。每个时间步的每个子公式（或谓词）都会引入新的二元变量。这意味着计算复杂度不仅随公式规模呈指数增长，也随时间步数量呈指数增长。更糟的是，时间步过少可能导致裁剪和穿越问题，即路径在时间步之间与障碍物相交（图 2a）。

缓解这些局限性是当前一项活跃的研究领域。[22] 在时间步之间使用控制屏障函数，以避免裁剪和穿越。 [23] 和[24]缩小了混合整数凸规划的规模，但其复杂度对于时间步数量仍呈指数增长。 [4]基于分段线性路径构建混合整数凸规划，并考虑时间步之间约束的满足情况。[4] 为我们提供了重要的灵感来源，而我们使用贝塞尔样条表示平滑路径，则是对这一思路的推广。除了让我们能够对整条路径而不仅仅是

采样点施加约束外，贝塞尔样条还使我们能够对微分平坦系统强制施加动力学可行性约束，并优化路径长度、速度和加速度等指标。

作为提高 MICP 效率的另一种方式，我们提出了一种公式，它使用更多二元变量，但具有更紧的凸松弛 [2]，因此在实践中能提升分支定界求解器的性能。本文的结果对这一基本思路作出了实质性改进。我们提出的方法得到的 MICP 具有更紧的凸松弛：事实上，它紧到可以通过凸优化和舍入 [6]求解。有趣的是，只有借助与较早的自动机理论方法之间的联系，我们才能构造出这种可高效求解的 MICP。从这个意义上说，我们相信，所提出的方法兼具基于自动机的方法与基于混合整数凸规划的方法的优势。

最后，我们承认，近期有一种趋势，试图通过非凸优化 [25, 26, 27, 28], 、学习[29, 30], 或控制障碍函数 [31, 32]提供近似解，从而完全规避时序逻辑运动规划的 NP难性。尽管这类方法可能极为高效，并且适用于某些应用，但它们提供的完备性保证有限，甚至完全没有，而且很少能够扩展到图 1a 所示的那种极其复杂的规范。

III. 问题表述

A. 线性时序逻辑

本节介绍线性时序逻辑的基础知识。更多细节请参见 [8]和 [9]。

线性时序逻辑的语法或文法定义如下：

$$\varphi := \top \mid a \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \bigcirc\varphi \mid \varphi_1 \mathsf{U} \varphi_2, \quad (1)$$

其中， $\top$ 表示“真”， a 是集合 AP中的一个原子命题， $\neg$ （“非”）是否定算子， $\wedge$ （“与”）是合取算子， $\bigcirc$ 是“下一步”时序算子，而 U 是“直到”时序算子。这些算子可以组合形成新的算子，例如“或”（$\varphi_1 \vee \varphi_2 = \neg$（$\neg\varphi_1 \wedge \neg\varphi_2$）） 、“最终”（$\Diamond\varphi = \top \mathsf{U} \varphi$）和“总是”（$\Box\varphi = \neg\Diamond\neg\varphi$）。

线性时序逻辑的语义（即含义）定义在由原子命题序列构成的、称为词的对象上：

定义 1. 词 $\sigma = A_0, A_1, A_2, \ldots$ 是由原子命题组成的序列，其中 $A_i \in 2^{AP}$ 是原子命题的集合。我们将从索引 j 开始的后缀记为 $\sigma[j \ldots] = A_j, A_{j+1}, \ldots$。

鉴于此，线性时序逻辑的语义递归定义如下，其中我们用 $\sigma \vDash \varphi$ 表示词 σ 满足线性时序逻辑公式 φ：

- $\sigma[j \ldots] \vDash \top$.
- $\sigma[j \ldots] \vDash a$ if and only if $a \in A_j$.
- $\sigma[j \ldots] \vDash \varphi_1 \wedge \varphi_2$ 当且仅当 $\sigma[j \ldots] \vDash \varphi_1$ 且 $\sigma[j \ldots] \vDash \varphi_2$。
- $\sigma[j \ldots] \vDash \neg\varphi$ 当且仅当 $\sigma[j \ldots] \nvDash \varphi$。
- $\sigma[j \ldots] \vDash \bigcirc\varphi$ 当且仅当 $\sigma[j+1 \ldots] \vDash \varphi$。
- $\sigma[j \ldots] \vDash \varphi_1 \mathsf{U} \varphi_2$ 当且仅当存在 $\exists k \geq j$，使得 $\sigma[k \ldots] \vDash \varphi_2$，并且对所有 $j \leq i < k$， $\sigma[i \ldots] \vDash \varphi_1$。

如上所定义的完整线性时序逻辑是在无限长词 $\sigma \in (2^{AP})^\omega$ 上进行求值的。线性时序逻辑的一个重要子集（或称片段）是语法上协安全的公式集合：

$$\varphi := \top \mid a \mid \neg a \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \bigcirc\varphi \mid \varphi_1 \mathsf{U} \varphi_2. \quad (2)$$

此类余安全公式是否满足，可以由有限长词唯一确定。对于运动规划问题而言，这一片段尤其重要，因为大多数运动规划的长度都是有限的。不过，如果规划中包含环，无限词也可能与运动规划有关：例如， $\varphi = \Box\Diamond a$ 可能表示机器人必须无限次访问标记为“a”的充电站。本文介绍适用于余安全片段和完整线性时序逻辑的运动规划算法，但我们只能为余安全公式提供最强的完备性和最优性保证。

我们将在此强调的线性时序逻辑的最后一个方面，是它与自动机的关系。简而言之，任何线性时序逻辑公式都可以转换为确定性 Büchi 自动机（DBA），而任何余安全线性时序逻辑公式都可以转换为确定性有限自动机（DFA）。定义如下：

定义 2。 确定性自动机是一个元组 $\mathcal{A} = (Q, q_0, \Sigma, \delta, F)$，其中

- Q是一个状态集合，
- $q_0 \in Q$ 是一个初始状态，
- Σ 是一个输入字母表，
- δ： $Q \times \Sigma \to Q$ 是转移关系
- $F \subseteq Q$ 是一个接受状态集合

如果 $\mathcal{A}$ 是一个确定有限自动机，任何接受运行都必须以 F 中的某个状态结束。如果 $\mathcal{A}$ 是一个确定性 *Büchi* 自动机，任何接受运行都必须无限次访问 F。

图4展示了一个线性时序逻辑公式及其对应的自动机示例。只有满足该线性时序逻辑公式的词才能作为输入被自动机接受。初始状态为 (1)，接受状态 (2) 则用双圆圈标记。

LTL 与 DFA/DBA 之间转换过程的详细信息可见 [9, 8]。虽然该转换的最坏情况复杂度相对于线性时序逻辑公式的大小呈双指数级，但已有成熟的模型检测软件，能够较快地转换规模适中的公式 [8, 33, 34]。

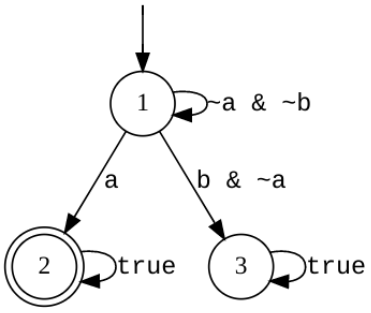


图 4：由 LTLf2DFA 工具 [33]生成的、与线性时序逻辑公式$\varphi = \neg b \mathrm{U} a$相对应的自动机。

B. 线性时序逻辑运动规划

为了将线性时序逻辑公式应用于运动规划，我们假设已知机器人构型空间中的一组带标签凸集。更正式地说，我们用 $\mathcal{X}$ 表示机器人构型空间中的一组凸区域，$\mathcal{X}_i \subseteq \mathbb{R}^n$。我们还定义 $L\colon \mathcal{X} \to 2^{AP}$，作为从这些区域到一组原子命题的映射。标签 $L(\mathcal{X}_i)$ 表示区域 $\mathcal{X}_i$ 中成立的原子命题。该标签适用于 $\mathcal{X}_i$ 中的每个构型。某个区域中可能没有任何原子命题成立，即 $L(\mathcal{X}_i) = \emptyset$，并且这些区域可能重叠，也可能互不重叠。此外，带标签区域不必覆盖整个构型空间。

简单来说，我们的目标是找到构型空间中的一条路径，使其满足给定的规范。更形式化地说，我们将路径满足线性时序逻辑定义如下：

定义 3. 路径 $p\colon \mathbb{R}^+ \to \mathbb{R}^n$ 是从时间 $t \in \mathbb{R}^+$ 到构型[1] $q \in \mathbb{R}^n$的映射，其中 $q_t = p(t)$ 表示系统在时间 t的构型。

定义 4. 路径 p 的轨迹是与p所经过的区域 $\mathcal{X}_i$ 相关联的标签序列 $L(\mathcal{X}_i)$，即 $\mathrm{trace}(p) = L(\mathcal{X}_0)\ L(\mathcal{X}_1) \ldots$

这样，我们可以说，路径 p 满足规范 φ，当且仅当 $\mathrm{trace}(p) \vDash \varphi$。

C. 问题

考虑到路径满足这一概念，我们将问题正式表述如下：

问题 1. 给定初始构型 q_0、凸区域 $\mathcal{X}$、标签 L 和 *LTL* 规范 φ，求一条满足该规范的最小成本路径，即，

$$\min_{p} \ J(p) \tag{3a}$$
$$\text{s.t. } \mathrm{trace}(p) \vDash \varphi, \tag{3b}$$
$$p \in \mathcal{C}^d, \quad d \geq 0, \tag{3c}$$
$$p(0) = q_0. \tag{3d}$$

代价 J 可以是路径 p 或其时间导数的任意凸函数。例如，我们可能希望最小化路径长度或加加速度（加速度的二阶导数） [37]。第五节 V-A 节讨论了参数化 p中的多种凸代价。约束 $p \in \mathcal{C}^d$ 确保 d 次连续可微性，从而能够为任何微分平坦系统 [19]强制满足动力学可行性。

IV. 背景

A. 贝塞尔曲线

要以数值方式求解优化问题 (3)，我们需要以某种方式用有限个点表示连续路径 p。时序逻辑文献中的标准做法是对路径进行离散化，仅考虑路径在等间距采样点 [1]处的值。但是，正如第 II 节所述，这种离散化会导致裁剪和穿越问题。

在本文中，我们用一系列贝塞尔曲线（即贝塞尔样条）对路径 p 进行参数化。贝塞尔曲线 $b\colon [0,1] \to \mathbb{R}^n$ 定义如下：

$$b(s) = \sum_{i=0}^{k} \binom{k}{i} (1-s)^{k-i} s^i \gamma_i, \tag{4}$$

其中，k 为阶数，$\gamma_i \in \mathbb{R}^n$ 为 $k+1$ 控制点。请注意，分段线性路径（例如 [4], 中讨论的路径）是其中 $k = 1$的特殊情况。

贝塞尔曲线具有许多优良性质，事实上，它们已广泛应用于机器人运动规划领域的文献中 [38],，尽管在时序逻辑领域并不常见：

- 次数为 k 的贝塞尔曲线可微 $k-1$ 次。
- 贝塞尔曲线的时间导数仍是一条贝塞尔曲线。
- 通过对控制点施加约束，可以将贝塞尔曲线首尾相接，形成一条路径（样条曲线）；也就是说，前一段的最后一个控制点必须与下一段的第一个控制点相等。还可以施加类似的约束，以确保达到所需的可微性。
- 贝塞尔曲线位于其控制点构成的凸包内。

图 2、3 和 7 展示了这样的贝塞尔样条曲线。控制点以红点表示，并用虚线连接；路径则以蓝色实线表示。

[1]如果用广义贝塞尔样条表示路径 [35, 36]，则也可以考虑构型空间非 $\mathbb{R}^n$ 的系统。不过，对于微分平坦系统，即使构型空间是非欧几里得的，平坦输出通常仍处于 $\mathbb{R}^n$ [19]。

B. 凸集图

我们所提出方法背后的计算核心是凸集图（GCS）。这一框架最初由 [5]提出，并应用于标准运动规划问题（到达目标并避开障碍物），见 [6]。本研究深受 [6],启发，并表明凸集图良好的计算特性可应用于比经典的到达-避障问题复杂得多的运动规划问题。具体而言，我们证明线性时序逻辑运动规划可转化为凸集图中的最短路径问题，并可利用 [5, 6] 开发的工具高效求解。

简而言之，GCS 框架旨在寻找图中的最短路径，其中图的每个顶点都与一个凸集相关联，而路径长度取决于所选的集合内点。 [5, 图 1] 展示了一个简单示例。更正式地说，GCS 定义如下：

定义 5. 凸集图 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ 是一个有向图，其中

- $\mathcal{V}$ 是一个顶点集合，
- 每个顶点 $v \in \mathcal{V}$ 都与一个凸集$\mathcal{X}_v$ 和一个点 $x_v \in \mathcal{X}_v$相关联，
- $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ 是一个边集合，
- 每条边 $e = (u, v) \in \mathcal{E}$ 都关联一个凸非负长度函数 $l_e(x_u, x_v)$，以及一个（可选的）凸约束 $(x_u, x_v) \in \mathcal{X}_e$。

给定一个凸集图，目标是找到从源顶点 $v_0 \in \mathcal{V}$ 到目标顶点 $v_T \in \mathcal{V}$的最小代价路径。将路径 ξ 定义为顶点序列，并将连接 v_0 和 v_T 的所有路径集合记为 Ξ，再将路径 ξ经过的边集记为 $\mathcal{E}_\xi$，则该问题可形式化表述如下：

$$\min \sum_{e=(u,v)\in\mathcal{E}_\xi} l_e(x_u, x_v) \tag{5a}$$

$$\text{s.t. } \xi \in \Xi \tag{5b}$$

$$x_v \in \mathcal{X}_v \quad \forall v \in p \tag{5c}$$

$$(x_u, x_v) \in \mathcal{X}_e \quad \forall e = (u, v) \in \mathcal{E}_\xi. \tag{5d}$$

事实证明，这个问题一般而言是 NP难的，不过，文献 [5]提出了一种高效的混合整数凸规划编码（利用透视函数并借助图论联系）。该混合整数凸规划具有非常紧的凸松弛，这意味着它与凸优化相当接近，通常可以由分支定界求解器高效求解。事实上，凸松弛非常紧密，因此往往可以结合凸优化与舍入找到较好的解 [6]。以这种方式使用凸优化时，即使不求解完整的混合整数凸规划，也有可能界定最优性间隙 [6,第 4 节。2], 而凸优化通常能够找到全局最优解——图 1a 中的示例便是如此。

尽管将问题 (5) 转写为凸规划时涉及许多重要细节，但有兴趣的读者可参阅 [5, 6]了解详情。本文使用 Drake [39] 提供的凸集图工具。基于此，我们的主要贡献是证明问题 1 可以改写为形式为 (5) 的凸集图问题；随后即可使用现有工具通过凸规划快速求解 (5)。

V. 将线性时序逻辑运动规划表述为凸集图

本节介绍将线性时序逻辑运动规划问题转化为 GCS 问题的主要结果。得到形式如式 (5) 的 GCS 问题后，我们可以使用混合整数凸规划精确求解式 (5)，也可以使用凸优化和舍入近似求解。

在 V-A 节中，我们考虑余安全 LTL 公式这一片段，它包含了与运动规划相关的大多数规范。在 V-B 节中，我们将这些结果推广到完整线性时序逻辑公式，并通过环考虑无限长度路径。

A. 余安全公式

算法 1 概述了基本思路：首先，我们构造一个有限状态转移系统抽象，并将给定的余安全 LTL 公式转换为确定有限自动机。然后，我们通过转移系统与确定有限自动机的积，构造一个凸集图。该积凸集图中的最短路径对应于一条满足 LTL 规范的最小成本路径。

算法 1 余安全线性时序逻辑运动规划

要求： 规范 φ、区域 $\mathcal{X}$、标签 L、初始状态 q_0
$TS = TransitionSystem(\mathcal{X},\ L)$ ▷定义 6
$\mathcal{A} = DFA(\varphi)$
$\mathcal{G} = TS \otimes \mathcal{A}$ ▷定义 7
$p^* =$ $\mathcal{G}$中的最短路径 ▷混合整数凸规划或凸优化。
返回 p^*

该过程与经典的模型检测方法非常相似，事实上也深受其启发：这些方法同样对 DFA 和转移系统取积，然后执行图搜索，以验证满足性[8]。本方法的关键区别在于，场景的物理布局以及对连续路径的约束都被纳入乘积图中，而该乘积图就是 GCS。现有的基于自动机的线性时序逻辑运动规划方法通过图搜索返回一个区域序列，还需要额外的低层规划器来寻找与该区域序列一致的连续路径[3, 10, 11]。相比之下，我们的方法将

逻辑约束的基于自动机的处理与连续路径的基于优化的考量，通过 GCS 框架一步结合起来。

首先，我们按如下方式从给定的带标签凸区域构造一个转移系统：

定义 6。转移系统抽象是一个元组$TS = (S, s_0, \to, \mathcal{L}, \mathcal{P})$，其中

- S 是与每个区域对应的一组状态，
- $\mathcal{P} : S \to \mathcal{X}$ 是从状态到构型空间中凸集的映射，
- s_0 是一个初始状态，使得 $q_0 \in \mathcal{P}(s_0)$，
- $\to$ 是一个转移关系，其中 $s \to s'$ 当且仅当 $\mathcal{P}(s) \cap \mathcal{P}(s') \neq \emptyset$，
- $\mathcal{L} : S \to 2^{AP}$ 是一个标记函数，$\mathcal{L}(s) = L(\mathcal{P}(s))$。

注意，只要区域之间存在非空交集，我们就定义一条转移。相交区域可能有显著重叠，也可能只是彼此相邻。

要将线性时序逻辑公式 φ 转换为确定有限自动机 $\mathcal{A}$，我们可以使用现有的多种软件工具中的任意一种。在第七节的示例中，我们使用 LTLf2DFA 工具 [33]。有关此转换过程的更多细节，请参阅 [8, 9]。

最后一步是将 TS 与 $\mathcal{A}$的积表示为一个凸集图：

定义 7。积 $\mathcal{G} = TS \otimes \mathcal{A}$ 是一个如下所示的凸集图：

- $\mathcal{V} = \{S \times Q, v_T\}$ 是顶点，
- $v_0 = (s_0, q_0)$，
- v_T 是一个额外的顶点，仅当 $q \in F$ 时才存在边 $(s, q) \to v_T$，
- 中其他边的定义方式如下：当且仅当 $s \to s'$ 是 TS 中的一次转移时，边 $(s, q) \to (s', q')$ 才存在；而 $\delta(q, \mathcal{L}(s)) = q'$ 是 $\mathcal{A}$ 中的一次转移，
- $\mathcal{X}_v = \mathcal{P}(s)^{k+1}$，其中 k 是所需的贝塞尔曲线次数，$\mathcal{Y}^k$ 表示凸集 $\mathcal{Y}$ 的笛卡尔幂，
- $x_v = [(\gamma_0^v)^T, (\gamma_1^v)^T, ..., (\gamma_k^v)^T]^T$ 是与顶点 v 相关联的贝塞尔曲线控制点，
- $\mathcal{X}_e$ 的定义使得 $\gamma_k^u = \gamma_0^v$，从而确保相邻贝塞尔曲线的连续性。对贝塞尔曲线导数的控制点也施加类似约束，以达到所需的光滑度。

基本思路是为 DFA 和 TS 中的每一对唯一状态定义一个顶点。仅当 DFA 和 TS 中都存在对应的边时，顶点之间才连接边：这确保同时满足场景的物理约束和线性时序逻辑公式的逻辑约束。目标状态 v_T 的定义方式是，任何通往 v_T 的路径都必须先经过 DFA 的一个接受状态 (F)，从而确保公式得到满足。每个顶点对应的凸集 $\mathcal{X}_v$ 表示一个凸区域。连续变量 $x_v \in \mathcal{X}_v$ 是贝塞尔曲线的控制点，该曲线被约束在对应区域内。边约束 $\mathcal{X}_e$ 确保相邻状态的控制点对齐，从而形成连续且平滑的路径。

按此方式定义，凸集图$\mathcal{G}$ 中从 v_0 到 v_T的任意路径都对应于构型空间中的一条光滑贝塞尔样条，并且保证满足给定的余安全线性时序逻辑规约。

最后一步是定义边长 $l_e(x_u, x_v)$，这对应于在式（3）中设定路径代价 $J(p)$。虽然可以使用任意凸函数，但采用贝塞尔曲线长度的近似值，例如

$$l_e(x_u, x_v) = \sum_{i=0}^{k} \|\gamma_i^u - \gamma_{i+1}^u\| \tag{6}$$

尤其值得采用。如果选择 $\|\cdot\|$ 为 L2 范数，那么 $J(p)$是实际路径长度的严格上界，且通常使用的控制点越多，近似就越精确。对于超大规模问题，选择 $\|\cdot\|$ 为 L1 范数可能更合适。尽管这会使路径长度的近似较为宽松，但由此得到的凸规划是线性规划（LP），而非二阶锥规划（SOCP） [5], ；线性规划的数值求解器更加成熟，且可提升可扩展性。由于贝塞尔曲线的导数仍是另一条贝塞尔曲线，因此可以采用类似的方法，为路径的时间导数添加相关代价项。

B. 完整线性时序逻辑

为了考虑完整线性时序逻辑规范，我们需要以某种方式，用有限数量的决策变量来描述无限长度路径。同样受模型检测文献的启发，我们选择用环来表示这类无限路径 [9],例如，

$$\sigma = A_0, A_1, \ldots, A_k, (B_0, B_1, \ldots, B_l)^\omega. \tag{7}$$

这种表示方式使我们能够通过搜索两条有限路径来描述无限行为：一条只执行一次的前缀，以及一条永远重复的后缀。

上文第五节 V-A 中建立的大部分机制都可以直接应用于完整线性时序逻辑情形。类似地，我们可以定义一个转移系统 TS，将线性时序逻辑公式 φ 转换为确定性 Büchi 自动机（而不是确定有限自动机） $\mathcal{A}_B$，并将其作为积 $\mathcal{G} = TS \otimes \mathcal{A}_B$计算得到一个凸集图。

关键区别在于确定性 Büchi 自动机的接受条件。对于确定性 Büchi 自动机，仅仅到达一个接受状态是不够的：我们既必须到达一个接受状态，也必须找到一个会无限次经过该接受状态的环

。这个环对应于第二条有限路径，其轨迹为 B_0，B_1，...，B_l。

我们通过将求解过程拆分为两次独立的凸集图求解来解决这个问题。算法 2 概述了这一流程。第一次求解与上文针对余安全公式所述的流程相同：我们只需找到一条从初始状态到接受状态的路径。这条路径的轨迹是第一个有限词A_0，A_1，.....，A_k。第二次凸集图求解旨在寻找一个环，其起点和终点均为第一次求解所得的同一贝塞尔曲线段。该环的轨迹对应于重复的有限词 B_0，B_1，.....，B_l。将这两条路径连接起来，即可得到一条包含环且满足规范的无限长度路径。

算法 2 完整线性时序逻辑运动规划

要求： 规范 φ，区域 $\mathcal{X}$，标签 L，初始状态 q_0
$TS = TransitionSystem(\mathcal{X},\ L)$
$\mathcal{A}_B = DBA(\varphi)$
$\mathcal{G} = TS \otimes \mathcal{A}_B$
当 $\mathcal{G}$ 包含接受状态 **时**
$p_1 =$ 在 $\mathcal{G}$ 中从 v_0 到 v_T的最短路径
$v_F = (s_F,\ q_F)$ $q_F \in F$是来自 p_1的接受状态
$p_2 =$ 在 $\mathcal{G}$ 中从 v_F 到 v_F的最短路径
如果 $p_2 \neq \emptyset$，**则**
返回 $p^* = p_1 \cdot p_2$
else
$\mathcal{G}.pop(v_F)$ 移除 v_F 并重试
结束 if
结束 while
返回未找到解

即使其他初始路径可能允许有效的环，也有可能第一次凸集图求解返回的路径不存在任何可行的环。在这种情况下，我们采用受反例引导的归纳综合启发的方法，舍弃第一次凸集图求解中使用的接受状态，并尝试重新求解问题。正如第六节将进一步详细探讨的那样，与用于余安全公式的过程相比，这一过程所能提供的形式化保证不那么令人满意。

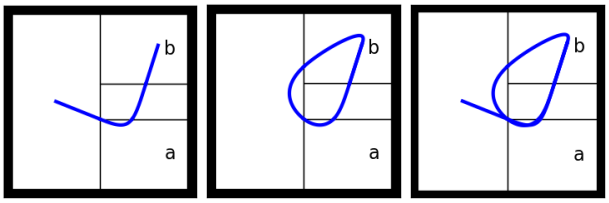
尽管如此，这个两步过程仍能为完整线性时序逻辑规范找到解，例如图 5 所示的规范。在该示例中，规范

$$\varphi = \square(\lozenge a \wedge \lozenge b) \tag{8}$$

要求移动机器人无限次访问两个区域。第一段 p_1 驱使机器人访问区域 a 和 b，到达自动机 $\mathcal{A}_B$ 的一个接受状态。第二段 p_2 构成一个环，并在区域 b 与 p_1 对齐。

VI. 理论分析

本节将对我们提出的方法的可靠性、完备性和计算复杂度进行形式化分析



(a) p_1 (b) p_2 (c) $p*$

图 5：LTL 规范 (8) 的一个解，该规范要求机器人无限次访问区域 a 和 b，且不属于余安全片段。

。简而言之，对于余安全 LTL 公式，我们的方法是可靠且完备的；对于一般的 LTL 公式，该方法是可靠的；其复杂度关于公式大小呈双指数级，关于控制点的数量和构型空间维数均呈多项式级。

在所有这些分析中，需要考虑四种情况，因为我们既可以使用余安全 LTL 公式，也可以使用一般 LTL 公式；并且可以通过混合整数凸规划或凸优化与舍入来求解 GCS 问题。

A. 可靠性

本节讨论可靠性。如果运动规划器返回的每个运动规划都满足给定规范，则该 LTL 运动规划器是可靠的。

可靠性相对容易保证；无论使用余安全还是完整线性时序逻辑公式，也无论通过混合整数凸规划还是凸优化来求解 GCS 问题，我们提出的方法都是可靠的。

定理 1（余安全 LTL 公式的可靠性）。 如果算法 *1* 针对余安全线性时序逻辑规约 φ 返回一条非空路径 p，那么轨迹(p) $\vDash \varphi$。

证明。 根据 GCS 的构造（定义 7），$\mathcal{G}$ 中任何到达目标顶点 v_T 的路径，都对应于确定有限自动机 $\mathcal{A}$ 中一条到达接受状态 $q_F \in F$ 的路径。根据线性时序逻辑公式 φ 与确定有限自动机 $\mathcal{A}$ 之间的关系，这意味着连续路径 p 经过一系列区域，其对应的标签序列满足 φ。因此，无论使用混合整数凸规划还是凸规划来寻找这样的路径，该定理均成立。 □

定理 2（完整线性时序逻辑公式的可靠性）。 如果算法 *2* 针对给定的 *LTL* 规范 φ 返回非空路径 p，那么 轨迹(p) $\vDash \varphi$。

证明。 根据凸集图 $\mathcal{G}$ 的构造，算法 2 的两步过程的任何解都对应于一条路径，该路径到达确定性 Büchi 自动机 $\mathcal{A}_B$ 中的接受状态 $q_F \in F$，并返回环 q_F。因此，该路径会访问

F 无穷多次，从而满足规范。因此，无论使用混合整数凸规划还是凸规划来寻找这样的路径，定理均成立。□

值得注意的是，这些可靠性属性适用于整个路径 p，而不仅仅是预先设定的采样点。这是相较于许多离散时间时序逻辑运动规划领域既有工作的显著改进 [1, 8],；在那些工作中，时间采样点之间的裁剪可能会导致规范失效（例如，导致与障碍物发生碰撞）。我们的方法保证每个贝塞尔曲线段都完全包含在单个区域内，从而消除了这一可靠性缺口。

B. 完备性

本节讨论算法 1 和 2 的完备性。如果某个解存在，LTL 运动规划器总能找到该解，则称其具有完备性。本节中的完备性保证是在给定的一组凸区域上提出的：如何将构型空间最优地分解为带标签凸区域，仍是未来研究的重要方向。

我们首先考虑使用混合整数凸规划求解凸集图子问题的余安全线性时序逻辑规范情形。在相对温和的假设下，这种情形能够提供最强的完备性保证：

定理 3（使用 MICP 时协安全线性时序逻辑的完备性）。假设存在一个满足条件的贝塞尔样条解，其次数和光滑度符合要求，且每个线段的控制点均包含在同一个区域内。如果使用 *MICP* 求解 *GCS* 子问题，那么算法 *1* 将返回一条满足给定协安全线性时序逻辑规约的路径。

证明。根据凸集图 $\mathcal{G}$ （定义 7）的构造，如果存在满足路径，那么就存在一条从 v_0到 v_T 、位于 $\mathcal{G}$中的路径。这是由规范 φ 与自动机 $\mathcal{A}$之间的关系得出的。根据 [5,定理 5，6], 用于求解（5）的 MICP 保证能够找到一条从 v_0 到 v_T的最优路径。□

该定理所需的主要假设值得进一步讨论。可能存在一条满足路径，但无法用符合要求的次数和光滑度的贝塞尔样条表示。也可能存在贝塞尔样条解，但该解所需的控制点位于某个区域之外。在这两种情况下，即使解确实存在，算法 1 也可能无法找到解。

幸运的是，这个问题有一个简单实用的解决办法：增加控制点的数量，和/或降低期望的光滑度。增加控制点的数量可以提高灵活性，通常还能使控制点更靠近路径。降低光滑度（例如， $\mathcal{C}^1$ 而不是 $\mathcal{C}^2$）可以消除对控制点的一些约束，也可能因此找到此前无法获得的解。

对于采用凸优化和舍入来求解 GCS 子问题的余安全线性时序逻辑规范，我们可以证明其具有概率完备性：

定理 4（采用凸优化的余安全线性时序逻辑完备性）。 假设存在一条满足路径，该路径由具有所需次数和连续性的贝塞尔曲线段构成，且每一段的控制点均包含在同一个区域内。此外，假设 *GCS* 子问题的凸松弛在最优路径上为二元变量赋予非零值。若使用 *[6]* 中所述的凸优化和舍入方案求解 *GCS* 子问题，则随着舍入迭代次数趋于无穷，算法 *1* 将以概率 *1* 返回一条满足给定余安全线性时序逻辑规范的路径 p 。

证明。根据凸集图的构造 $\mathcal{G}$ （定义 7），如果存在满足路径，则存在一条从 v_0到 v_T 的路径，该路径位于 $\mathcal{G}$中。这是由规范 φ 与自动机 $\mathcal{A}$之间的关系得出的。 [6,第 4 节中所述的舍入方案将凸松弛中二元变量的取值视为概率，并通过带回溯的随机深度优先搜索执行舍入。随着舍入迭代次数趋于无穷，该过程最终将以概率 1 遍历 $\mathcal{G}$ 中的所有可能路径。如果（5）的凸松弛不可行，则该混合整数凸规划不可行，因此不存在解（根据定理 3）。

□

注 1. 在实际应用中，要求最优路径上的二元变量非零，这一假设的限制性很小。事实上，分支定界混合整数凸规划求解器通常会根据凸松弛中的整数值二元变量剪枝 *[40]*。

在实际应用中（见第七节中的示例），凸优化与混合整数凸规划之间的差距相当小。对于本文考察的所有示例，凸优化只需经过少数几次（< 10）舍入迭代，就能找到满足规范的解，即便未能找到经认证的全局最优解。

对于完整线性时序逻辑规范，由于必须找到一个环，完备性保证会受到显著限制。尽管如此，在一组限制性更强的假设下，我们仍能保证完备性：

定理 5. 假设定理 *3* 的条件成立。此外，假设对于凸集图 $\mathcal{G}$ 中至少一个接受状态 $v = (s, q)$ $q \in F$，每条包含在 X_v 中的贝塞尔曲线 b 都存在一条经过 $\mathcal{G}$ 、以 开始并结束的环

。那么，算法 *2* 将返回一条满足路径。

证明。 根据定理 3，算法 2 最终会找到一条路径 p_1，到达特殊的接受状态 v。如果算法 2 找到一条路径 p_1 通往某个接受状态 v'，但其中的每条贝塞尔曲线$X_{v'}$都无法形成环，那么 v' 将从图中移除，并重新计算一个p_1 。找到合适的 p_1 后，根据定理 3 以及环存在这一假设，同样可以保证找到一个环 p_2。完整路径$p* = p_1 \cdot p_2$ 对应于无限次访问接受集 F，因此定理得证。 □

对于完整线性时序逻辑情形，使用凸优化而非混合整数凸规划，按照与定理 4 类似的推理，可以获得概率完备性保证。

还需要额外假设：至少一个接受状态中的每条贝塞尔曲线都能形成环，以确保算法 2 中受反例引导归纳综合启发的过程最终返回一个解。假设每条贝塞尔曲线都能形成环尤其严格——可能存在一个环（p_2），但它无法从终点为 p_1的起始贝塞尔曲线段出发。之所以需要这一限制，是因为凸集图框架不允许未通过边连接的连续变量x_v 之间存在约束。这意味着，要找到一个环，我们必须将初始和最终连续变量 x_0 和 x_T 约束为某些预先固定的值。

C. 复杂度

本节将正式刻画所提出方法的计算复杂度，重点关注按照 [6] 中所述，使用凸优化和舍入求解 GCS 子问题的情况。

乍看之下，鉴于线性时序逻辑运动规划是 NP难问题、凸优化可在多项式时间内求解，且我们使用凸优化求解线性时序逻辑运动规划，似乎我们已经证明了 $P = NP$。我们强调，事实并非如此，原因有二：首先，凸优化只能为 GCS 子问题提供近似解，而该子问题本身是 NP难问题 [5]。其次，也是更重要的一点，自动机 $\mathcal{A}$ 的规模相对于线性时序逻辑公式 φ 的规模呈双指数增长，因此，GCS $\mathcal{G}$ 的规模相对于线性时序逻辑公式的规模也呈双指数增长。

定理 6. 给定凸集图 $\mathcal{G}$，使用凸优化求解最短路径问题并找到近似解的时间复杂度为 $O(2^{2^{|\varphi|}})$，其中 $|\varphi|$ 是 φ 的规模。

证明。 将 φ 转换为自动机 $\mathcal{A}$ 后，得到一个包含 $O(2^{2^{|\varphi|}})$ 个状态 [8], 的自动机，因此凸集图 $\mathcal{G}$ 具有 $O(2^{2^{|\varphi|}})$ 个顶点和边。 [6] 的凸优化编码会产生一个 LP 或 SOCP，其中为 $\mathcal{G}$中的每个顶点和边设置决策变量。由于 LP 和 SOCP 的时间复杂度关于决策变量的数量为多项式级别 [41], ，因此总体时间复杂度为 $O(2^{2^{|\varphi|}})$。

□

由于该问题具有 NP 难性，公式大小带来的指数复杂度不可避免；不过，我们提出的方法对于实际关注的变量数量呈多项式扩展：

定理 7. 给定凸集图 $\mathcal{G}$，通过凸优化求解最短路径问题近似解的时间复杂度关于每个贝塞尔曲线段的次数 k 呈多项式关系。

证明。 这是因为决策变量的数量随 k 线性增长，而该凸优化问题是 LP 或 SOCP，可通过内点法在多项式时间内求解 [41]。

□

这意味着，复杂度关于用于表示曲线的控制点总数 γ_i 呈多项式关系。这与标准的基于混合整数凸规划的方法形成对比：这类方法会随着用于表示路径的采样点数量呈指数扩展。在实践中，我们发现增加控制点数量对求解时间几乎没有实际影响，见图6。

定理 8. 给定凸集图 $\mathcal{G}$，通过凸优化求解最短路径问题近似解的时间复杂度关于构型空间的维数 n 呈多项式关系。

证明。 这是因为决策变量的数量随 n 线性增长，而凸优化问题是线性规划或二阶锥规划，可通过内点法在多项式时间内求解 [41]。

□

这是使我们的方法能够扩展到高维配置空间的一个关键因素，我们将在第七节-C 中探讨这一点。

VII. 示例

在本节中，我们通过几个仿真实例展示所提出方法的可扩展性。可在 [7] 获取用于复现这些实例的代码。所有实验均在一台配备 Intel i7 CPU 和 32GB 内存的笔记本电脑上完成。底层凸优化求解器为 MOSEK [42],, 通过 Drake [39] Python 绑定调用。

所有示例的求解时间如表 I 所示。除了最终求解时间外，我们还报告了将 LTL 规范 φ 转换为确定有限自动机的耗时

$\mathcal{A}$ 以及构建凸集图 $\mathcal{G} = \mathcal{A} \otimes TS$所需的时间。由于此操作具有双指数复杂度，将 LTL 转换为确定有限自动机对于复杂规范而言可能代价尤为高昂 [8]。

在这些步骤中，只有最后的求解是在线操作；其他步骤都可以离线执行，并将结果用于不同的初始条件。在所有示例中，我们都使用凸优化而非 MICP 来求解 GCS 问题。

Name	Fig.	线性时序逻辑到确定有限自动机	构建凸集图	**求解**
最短路径	3	0.13	0.05	**0.34**
简单钥匙门	2	0.54	0.34	**0.29**
复杂钥匙门	1a	32 分钟	547	**5.82**
多目标	7	0.64	3.72	**7.28**
机械臂	9	0.10	0.08	**1.09**
人形机器人	1b	0.11	0.18	**7.72**

表 I：本节各示例的时间细分。除非另有说明，时间单位均为秒。除“求解”外，所有操作均可离线执行。

A. 平面运动规划

本节研究简单平面机器人的运动规划。由于我们考虑的是路径规划而非轨迹规划，因此不使用机器人的具体动力学。不过，我们可以设计具有所需光滑度的路径（见第四节-A），从而确保其适用于任意微分平坦系统 [19]。

图 3 所示的第一个示例说明了同时考虑逻辑约束和场景几何的优势（这正是大多数基于自动机的方法难以做到的）。一个移动机器人需要最终到达两个目标区域之一，分别标记为 a 和 b:

$$\varphi = \Diamond(a \vee b). \tag{9}$$

基于抽象的运动规划器（例如 [11] ）可能会选择通往区域 a的较长路径，因为这样在 $\mathcal{G}$中所需的转移更少。另一方面，我们的规划器直接找到通往目标 b的较短路径。

本例中的代价是对路径长度的简单惩罚（控制点之间距离的 L2 范数）。采用次数为 $k = 4$ 的贝塞尔曲线，并强制满足 $\mathcal{C}^2$ 连续性。图 3 中的红点表示贝塞尔控制点，最终路径则以蓝色显示。

在第二个平面运动示例中，如图 2 所示，机器人必须先拾取两把钥匙（即访问标记为 $key1$ 和 $key2$的绿色区域），才能穿过相应的门（红色，标记为 $door1$ 和 $door2$），最后到达目标（蓝色，标记为 $goal$）。更正式地说，该规范可以写成一个线性时序逻辑公式

$$\varphi = (\neg door1 \mathsf{U} key1) \wedge (\neg door2 \mathsf{U} key2) \wedge \Diamond goal. \tag{10}$$

图 1a 展示了该规范的一个更复杂版本，其中有五把钥匙和五扇门。这个基准测试示例最初由 [3], 提出，并通过一种基于抽象的方法求解，但未报告求解时间。据我们所知，该基准测试已报告的最快求解时间是 [4]的 49.5 秒。 [4] 使用混合整数凸规划求得一条分段线性路径（避免裁剪和穿越）。只要离线完成从线性时序逻辑到确定有限自动机的转换以及凸集图的构建，我们提出的方法就能快约一个数量级（表 I）。此外，我们的方法生成的路径具有二阶连续可微性，并且经证明是全局最优的。从线性时序逻辑转换为确定有限自动机所需的极长时间，表明该操作具有双指数复杂度 [8]。

我们使用更简单的钥匙与门示例（含两扇门），将其与标准的基于 MICP 的时序逻辑运动规划器进行比较。后者采用时间离散化，而不是用连续值曲线表示运动规划。具体而言，我们将其与 [20] 的混合整数凸规划方法进行比较，该方法由 *stlpy* [24]python 库[2] 实现。

标准 MICP 使用预先指定数量的时间步来表示连续路径。如果时间步太少，裁剪和穿越问题可能导致路径在时间步之间与障碍物相交，如图 2a 所示。另一方面，我们提出的方法考虑整条曲线，因此即使使用的控制点相对较少，也不会出现裁剪或穿越问题，如图 2b 所示。

通常，通过对路径进行更精细的采样、增加时间步来解决裁剪和穿越问题。然而，这会导致标准方法的时间复杂度呈指数级增长。另一方面，我们的方法的规模仅随控制点数量呈多项式增长（定理 7）。这一事实在实践中的影响如图6所示。

在图6中，我们比较了标准混合整数凸规划（ MICP）与所提出方法的求解时间：对于标准方法，改变时间步的数量；对于我们的方法，改变所使用控制点的总数（大致相当于时间步的数量）。正如预期，标准混合整数凸规划的求解时间呈指数增长：对于该规范，使用超过约30个时间步并不实际。另一方面，随着控制点总数的增加，我们所提出方法的求解时间基本保持不变。在对求解时间影响极小的情况下，我们最多可以使用90个控制点——在本例中，每个贝塞尔曲线段使用10个控制点。

我们的最后一个平面运动规划示例如图 7 所示，展示了一个更复杂的场景，同时

[2]虽然该库面向的是信号时序逻辑（STL）而非线性时序逻辑（LTL），但相关规范不包含有效的时间约束，因此STL规范与LTL规范是等价的。

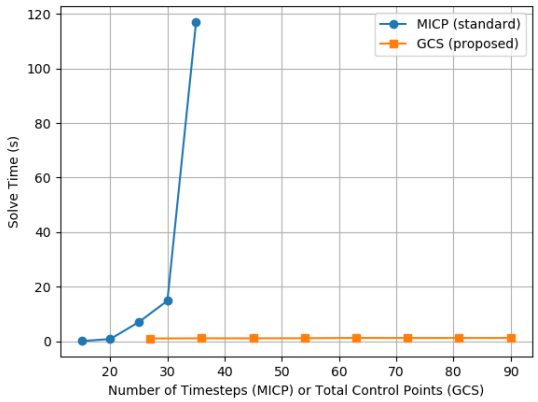


图6：随着用于表示路径的采样点数量增加（标准 MICP 中为时间步数量，我们的方法中为控制点数量）,求解时间的变化图。标准方法的计算规模随时间步数量呈指数增长，而我们提出的方法的求解时间几乎不随控制点数量的增加而变化。

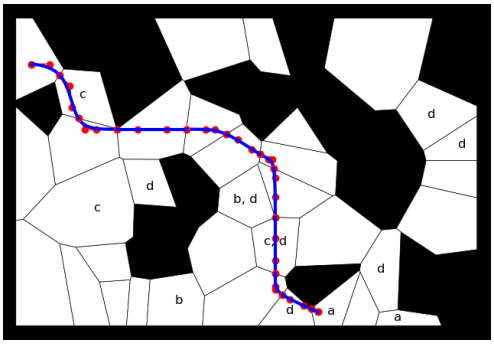


图7：移动机器人必须在一个包含多个带标签目标的环境中导航，最终到达 a、 c 和 d，同时始终避开 b。这由规范 (11) 形式化描述。

展示了一个不属于余安全片段的规范。机器人需要最终访问标记为 a、 c 和 d 的区域，同时始终避开 b:

$$\varphi = \Diamond a \wedge \Box\neg b \wedge \Diamond c \wedge \Diamond d. \tag{11}$$

全称算子 (□) 意味着该规范在语法上并非协安全 [8]。不过，用于表示无限长度行为的最终环是平凡的：机器人只需停留在最终配置处。

这个多目标示例的凸集图规模相对较大；如果使用 L2 范数近似来估计路径长度，那么最短路径子问题就是一个规模庞大且可能条件不佳的二阶锥规划。我们发现，MOSEK [42] 无法收敛到一个该二阶锥规划的解，并返回“求解器状态：未知”错误代码。为避免这种情况，我们改用路径长度的 L1范数近似（见 V-A节）。这样得到的是线性规划，而非二阶锥规划；对于线性规划，求解器更加成熟可靠。使用这种 L1/ 线性规划形式，MOSEK 得到了图 7 所示的解。

B. 与基于采样的运动规划的比较

对于简单的到达-避障问题（线性时序逻辑运动规划的一种特殊情况），已有研究表明，基于图形凸集的运动规划优于基于采样的方法 [6]。本节将探讨，在 LTL 规范下进行运动规划是否也具有类似优势。

我们将所提出的方法与 [14] 中介绍的最先进的基于采样的规划器进行比较。这种方法尤其有吸引力，因为它不需要计算转移系统与规范自动机的积，而是通过逐步构建树来探索积的状态空间，无需显式构造该积。

具体而言，我们考虑 [14，第 VII.A 节] 中的示例。该示例（[14, 图 6]）包含六个感兴趣区域（r_i）和两个障碍物（obs）。任务由以下线性时序逻辑公式指定：

$$\varphi = \Diamond(r_1 \wedge \Diamond r_3) \wedge (\neg r_1 \mathsf{U} r_2) \wedge \Diamond(r_5 \wedge \Diamond(r_6 \wedge \Diamond r_4)) \wedge (\neg r_4 \mathsf{U} r_5) \wedge \Box\neg obs. \tag{12}$$

该规范要求移动机器人完成以下任务：最终访问区域 r_1，然后访问 r_3；在访问 r_2 之前不得访问 r_1；最终依次访问 r_5、r_6 和 r_4；在访问 r_5 之前不得访问 r_4；并避开障碍物。

为了比较我们的方法与 [14]，我们随机生成了 100 个初始位置，并分别使用我们的方法和基于采样的规划方法求解运动规划问题。对于基于采样的规划器，我们最多使用 $n_{max} = 1000$ 次迭代，并采用 [14] 提供的实现中的默认参数。对于我们的方法，我们求解带有 L2 范数路径长度惩罚的一阶贝塞尔样条，并执行一次舍入迭代。求解时间直方图见图 8。对于两种方法，我们都不计构建自动机所需的时间，因为这一步可以离线完成。

与基于采样的方法相比，我们提出的方法平均求解时间更短（均值为 0.30 秒，而基于采样的方法为 0.69 秒），且更加稳定（标准差为 0.024 秒，而基于采样的方法为 0.52 秒）；不过，对于某些初始条件，基于采样的方法找到解的速度更快。

C. 高自由度系统

我们提出的方法能够很好地扩展到高维构型空间。为此，

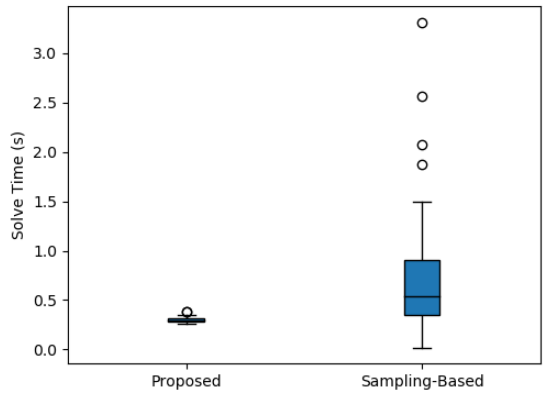


图 8：在 100 次采用不同初始条件的试验中，使用我们提出的方法和基于采样的方法求解规范 (12) 所需的时间。 [14]

我们考虑图 9 所示的 Franka Panda 机械臂 7 自由度模型，以及图 10 所示的 Atlas 人形机器人 30 自由度模型。这两个模型均来自 Drake[39]。

*1) 7*自由度机械臂：该机械臂需要到达一个目标构型，使其末端执行器穿过门口（红色方框），但在穿过门口之前必须按下一个按钮（蓝色半圆）。这一规范可写为

$$\varphi = \Diamond target \wedge \neg doorway \mathsf{U} button. \tag{13}$$

使用 Drake [39] 中实现的 C-IRIS [43], , 在构型空间中生成了带标签的无碰撞凸多胞体。

与时序逻辑运动规划的最先进方法相比，我们提出的方法在大型构型空间中带来了显著的性能提升。在先前的工作 [28], 中，我们展示了如何利用非凸轨迹优化来解决类似 7 自由度机械臂的时序逻辑运动规划问题，并在 160 秒内求解了一个相对简单的规范 ($\Diamond(a \vee b)$)。相比之下，我们提出的方法只需约一秒即可求解复杂得多的规范 (13)。

*2) 30*自由度人形机器人：Atlas 人形机器人的任务是触及三个目标框，如图 1b 中的绿色、蓝色和红色目标所示。我们使用逆运动学和 C-IRIS[43] 来定义构型空间中的带标签凸区域，分别对应于：(a) 左手触碰绿色目标；(b) 右脚触碰红色目标；以及 (c) 右手触碰蓝色目标。我们还定义了一个不带运动学约束的无标签凸区域 (d)。用于生成这些凸集的种子构型如图 10 所示。

规范随后定义为

$$\varphi = \Diamond(a \wedge \Diamond(b \wedge \Diamond c)), \tag{14}$$

这表示机器人应按顺序最终到达目标 a、 b和 c。

我们固定骨盆在世界坐标系中的位姿，不考虑与地面的接触交互。我们不考虑平衡、接触扳手锥或其他动力学可行性约束：我们的目标仅仅是找到一条满足规范的构型空间路径。不采用浮动基座时，这是一个 30 自由度系统。

我们使用次数为 $k = 3$ 的贝塞尔样条搜索满足路径，并施加 C^2 光滑度约束。除了对路径长度施加 L2 范数惩罚外，我们还对构型空间加速度施加较小的惩罚。如表 I 所示，我们在 7.72 秒内找到了局部最优路径。

VIII. 局限性

本节简要概述所提出方法的局限性，并提出在实践中缓解这些局限性的具体建议。

将线性时序逻辑公式转换为自动机的一个主要缺点是其双指数复杂度 [8]。表 I 最清楚地展示了这一局限性：在复杂钥匙门场景中，将公式转换为确定有限自动机耗时超过半小时。我们使用 LTLf2DFA Python 库 [33], 执行了这一转换，该库使用 MONA [34] 完成底层转换。尽管其他模型检测软件（参见 [8] 中的列表）可能性能更好，但任何此类转换软件都无法避免这一转换固有的双指数复杂度。

为缓解实际应用中的这一问题，我们注意到，只要事先已知带标签区域和规范，就可以离线完成向自动机的转换以及凸集图的构建。对于这个离线计算得到的凸集图，可以使用任意初始条件：只有初始顶点 $v_0 = (s_0, q_0)$ 会发生变化。给定一个凸集图后，凸优化的求解时间通常相对较短（约为几秒），即使规范转换为自动机的过程极其缓慢也是如此。

另一个相关的局限是需要带标签的凸区域。对于第七节-A 中这类简单的平面场景，这或许是合理的预期；但对于具有高维构型空间的复杂系统，例如第七节-C 中的机械臂，如何获得这类带标签区域并不总是一目了然。所幸，近年来在将自由空间分解为凸集方面取得了显著进展。IRIS [44], 和 C-IRIS [43], 等算法提供了扩张自由空间中凸区域的迭代过程。这类算法为构建高维系统的带标签凸集提供了便利途径

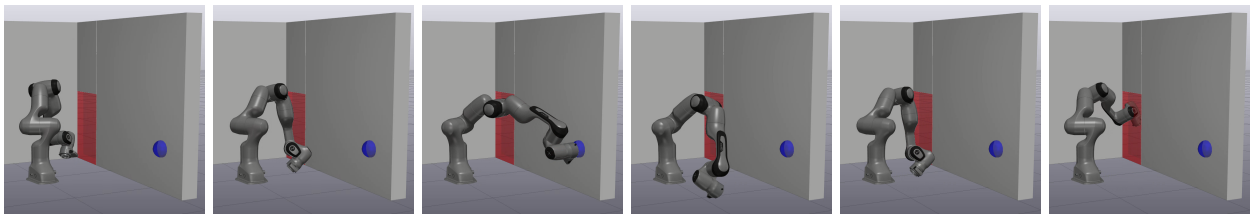
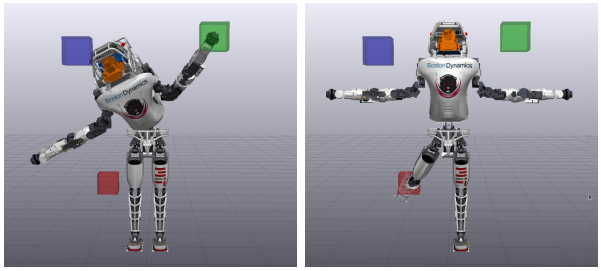


图 9：根据公式 (13)，7自由度机械臂需要先推动一个按钮（蓝色），然后穿过门口（红色）。使用 C-IRIS [43] 在线下生成了构型空间的带标签凸分解。



(a) 标签： a (b) 标签： b

(c) 标签： c (d) 标签： $\emptyset$

图 10：用于为 Atlas 人形机器人生成带标签凸区域的种子构型。对于标记为 a、b 和 c 的区域，逆运动学约束确保手或脚保持在目标框内。

。直接根据时序逻辑标签高效生成凸区域，仍是未来研究的重要方向。

该方法的另一个缺点是，我们考虑的是路径规划，而非轨迹规划。这意味着，满足规范的路径不考虑系统的自然动力学，因此可能不具备动力学可行性。然而，贝塞尔样条能够确保达到所需的光滑度，因此可以为任何微分平坦系统确保动力学可行性。微分平坦系统是一大类系统，其中包括四旋翼无人机 [37] 和差速驱动移动机器人 [19]。

从理论角度来看，凸优化方法提供的保证不如混合整数凸规划（MICP）充分，因为凸优化和舍入无法保证找到全局最优解。在实践中，这似乎并不是一个重要问题。截至撰写本文时，我们尚未发现任何混合整数凸规划（MICP）能够找到可满足解、而凸优化却无法找到的案例。

最后，我们强调，对于完整线性时序逻辑（非余安全）规范，可提供的完备性保证有限（定理 5）。这一局限归根结底源于在 GCS 框架中编码环路约束的困难。尽管如此，许多具有实际意义的线性时序逻辑运动规划规范，要么属于余安全片段（9, 10, 13），要么虽然在语法上不属于余安全片段，但存在环路平凡的解（11）。

IX. 结论与未来工作

我们提出了一种用于时序逻辑运动规划的凸优化解法。其核心思想是将线性时序逻辑运动规划问题转化为凸集图上的最短路径问题。这个 GCS 问题可以通过混合整数规划精确求解，也可以通过凸优化与舍入获得实用的近似解 [5, 6]。

我们提出的方法可扩展至复杂的规范（图1a）和高维构型空间（图9），并克服了标准时序逻辑运动规划方法的诸多局限性。我们通过使用贝塞尔样条表示路径，避免了离散时间时序逻辑公式中与之相关的裁剪和穿越问题 [45]。与标准的基于混合整数凸规划的方法 [20, 1, 4],不同，我们提出的方法不会随着用于表示路径的采样点数量增加而呈指数级扩展（图6）。与局部优化 [25, 26, 27], 、基于学习的方法[29, 30], 或基于 CBF 的方法 [31] 不同，在适度的假设条件下，我们提出的方法能够提供概率完备性保证。

未来工作的潜在方向包括拓展至多智能体系统和分布式系统，拓展至 STL 和 MTL 等定时序时序逻辑，以及将其应用于模型预测控制框架。

参考文献

[1] Calin Belta 和 Sadra Sadraddini。控制综合的形式化方法：一种优化视角。*AnnualReview of Control, Robotics, andAutonomous Systems*，2:115–140，2019年。[2] Vince Kurtz 和 Hai Lin。一种更具可扩展性的度量时序逻辑混合整数编码。*IEEEControl SystemsLetters*，6:1718–1723，2021。[3] William Vega-Brown 和 Nicholas Roy。用于近似最优任务与运动规划的容许抽象。*arXivpreprint arXiv:1806.00805*，2018。[4]Dawei Sun、Jingkai Chen、Sayan Mitra 和 Chuchu Fan。基于信号时序逻辑规范的多智能体运动规划。*IEEE Roboticsand AutomationLetters*，7(2):3451–3458，2022年。[5] Tobia Marcucci、Jack Umenberger、Pablo A Parrilo 和 Russ Tedrake。凸集图中的最短路径。*arXivpreprint arXiv:2101.11565*，2021。[6]Tobia Marcucci、Mark Petersen、David von Wrangel 和 Russ Tedrake。利用凸优化进行避障运动规划。*arXivpreprintarXiv:2205.04422*，2022年。[7] https://github.com/vincekurtz/ltl_gcs。[8]Calin Belta、Boyan Yordanov 和 Ebru Aydin Gol。离散时间动力系统的形式化方法，第15卷。Springer，2017年。[9] Christel Baier 和 Joost-Pieter Katoen。模型检测原理。MIT Press，2008年。[10] Rafael Rodrigues da Silva、Vince Kurtz 和 Hai Lin。基于时序逻辑规范的主动感知与控制。*IEEEControl Systems Letters*，3(4):1068–1073，2019年。[11] Rafael Rodrigues da Silva、Vince Kurtz 和 Hai Lin。实时定时逻辑的自动轨迹综合。*IEEETransactionsonAutomaticControl*，67(2):780–794，2021。[12]Edmund Clarke、Orna Grumberg、Somesh Jha、Yuan Lu 和 Helmut Veith。反例引导的抽象细化。载于*InternationalConference onComputer AidedVerification*，第154–169页。Springer，2000年。[13] Cristian Ioan Vasile 和 Calin Belta。基于采样的时序逻辑路径规划。载于*2013IEEE/RSJ InternationalConference on IntelligentRobots and Systems*，第4817–4822页。IEEE，2013年。[14] Xusheng Luo、Yiannis Kantaros 和 Michael M Zavlanos。一种用于多机器人时序逻辑最优控制综合的无抽象方法。*IEEE Transactionson Robotics*，37(5):1487–1507，2021。

[15] Yiannis Kantaros、Samarth Kalluraya、Qi Jin 和 George J Pappas。基于感知的时序逻辑规划：面向不确定语义地图。*IEEE Transactions on Robotics*，2022年。[16] Liangjun Zhang 和 Dinesh Manocha。一种高效的基于回缩的 RRT 规划器。载于*2008*年 *IEEE* 国际机器人与自动化会议，第 3743–3750 页。IEEE，2008年。[17] Marius Kloetzer 和 Calin Belta。基于时序逻辑规范的线性系统控制全自动化框架。*IEEETransactionsonAutomatic Control*，53(1)：287–297，2008年。[18]Hadas Kress-Gazit、Georgios E Fainekos 和 George J Pappas。基于时序逻辑的反应式任务与运动规划。*IEEE Transactions on Robotics*，25(6)：1370–1381，2009年。[19]Richard M Murray、Muruhan Rathinam 和 Willem Sluis。机械控制系统的微分平坦性：原型系统目录。载于*ASME*国际机械工程大会暨展览会。Citeseer，1995年。[20] Vasumathi Raman、Alexandre Donz´e、Mehdi Maa-soumy、Richard M Murray、Alberto Sangiovanni-Vincentelli 和 Sanjit A Seshia。带信号时序逻辑规范的模型预测控制。载于第*53*届*IEEE*决策与控制会议，第 81–87 页。IEEE，2014年。[21] Eric M Wolff、Ufuk Topcu 和 Richard M Murray。带线性时序逻辑规范的基于优化的轨迹生成。载于*2014*年 *IEEE* 国际机器人与自动化会议（*ICRA*），第 5319–5325 页。IEEE，2014年。[22]Guang Yang、Calin Belta 和 Roberto Tron。基于控制障碍函数的连续时间信号时序逻辑规划。载于*2020*年美国控制会议（*ACC*），第 4612–4618 页。IEEE，2020年。[23] Sadra Sadraddini 和 Calin Belta。正单调系统控制策略的形式化综合。*IEEETransactions onAutomatic Control*，64(2)：480–495，2018年。[24]Vincent Kurtz 和 Hai Lin。采用较少二元变量的信号时序逻辑混合整数规划。*IEEEControl Systems Letters*，6：2635–2640，2022年。[25]Yash Vardhan Pant、Houssam Abbas 和 Rahul Mangharam。平滑算子：利用时序逻辑的平滑鲁棒性进行控制。载于*2017*年 *IEEE* 控制技术与应用会议（*CCTA*），第 1235–1240 页。IEEE，2017年。[26]Noushin Mehdipour、Cristian-Ioan Vasile 和 Calin Belta。用于信号时序逻辑规范控制的算术-几何平均鲁棒性。载于*2019*年美国控制会议

（*ACC*），第 1690–1695 页。IEEE，2019年。[27] Yann Gilpin、Vince Kurtz 和 Hai Lin。一种用于符号控制的信号时序逻辑平滑鲁棒性度量。*IEEEControl SystemsLetters*，5(1)：241–246，2020。[28]Vince Kurtz 和 Hai Lin。面向高维非线性系统的轨迹优化：基于 STL 规范。*IEEE ControlSystems Letters*，5(4)：1429–1434，2020。[29] Mingyu Cai、Shaoping Xiao、Baoluo Li、Zhiliang Li 和 Zhen Kan。基于强化学习的时序逻辑控制及最大概率满足度。载于*2021*年 *IEEE* 国际机器人与自动化会议（*ICRA*），第 806–812 页。IEEE，2021年。[30]K. Leung 和 M. Pavone。针对信号时序逻辑规范的半监督轨迹反馈控制器综合。载于美国控制会议，2022年。[31] Lars Lindemann 和 Dimos V Dimarogonas。用于信号时序逻辑任务的控制障碍函数。*IEEE Control Systems Letters*，3(1)：96–101，2018。[32] Mohit Srinivasan 和 Samuel Coogan。在时序逻辑规范下利用障碍函数控制移动机器人。*IEEETransactionsonRobotics*，37(2)：363–374，2020。[33]Francesco Fuggitti。LTLf2DFA，2019年3月。https://doi.org/10.5281/zenodo.3888410。[34] Nils Klarlund 和 Anders Møller。*MONA 1.4* 版用户手册。BRICS，计算机科学系，奥胡斯大学，2001年1月。笔记系列 NS-01-1。可从 `http://www.brics.dk/mona/`获取。[35] Myoung-Jun Kim、Myung-Soo Kim 和 Sung Yong Shin。具有简单高阶导数的单位四元数曲线通用构造方案。载于第*22*届计算机图形学与交互技术年会论文集，第 369–376 页，1995年。[36] Zhongxuan Luo、Qian Wang、Xin Fan、Yaqi Gao 和 Panpan Shui。用于刚体运动设计的广义有理 B´ézier 曲线。*TheVisualComputer*，32(9)：1071–1084，2016年。[37] Daniel Mellinger 和 Vijay Kumar。四旋翼无人机的最小跃度轨迹生成与控制。载于*2011*年 *IEEE* 国际机器人与自动化会议，第 2520–2525 页。IEEE，2011年。[38] Boris Lau、Christoph Sprunk 和 Wolfram Burgard。使用样条曲线的移动机器人运动学-动力学运动规划。载于*2009*年 *IEEE/RSJ* 国际智能机器人与系统会议，第 2427–2433 页。IEEE，2009年。[39] Russ Tedrake 和 Drake 开发团队。

Drake：机器人学的基于模型的设计与验证，2019。[40]Michele Conforti、G´erard Cornu´ejols、Giacomo Zambelli 等。整数规划，第 271 卷。Springer，2014年。[41] Yurii Nesterov 和 Arkadii Nemirovskii。凸规划中的内点多项式算法。SIAM，1994年。[42] MOSEK ApS。*MOSEK* 优化工具箱。*9.0* 版。，2022年。[43]Alexandre Amice、Hongkai Dai、Peter Werner、Annan Zhang 和 Russ Tedrake。为机器人机械臂在构型空间中寻找并优化经过认证的无碰撞区域。载于国际机器人算法基础研讨会，第 328–348 页。Springer，2023年。[44] Robin Deits 和 Russ Tedrake。通过半定规划计算大型无障碍空间凸区域。载于机器人学算法基础 *XI*，第 109–124 页。Springer，2015年。[45] Hai Lin。任务圆满完成：移动机器人运动规划与控制中形式化方法简介。《无人系统》，2(02)：201–216，2014年。