面向数控机床的可执行强化学习运动规划：融合捷度约束投影与容差走廊择路

作者姓名[a]

[a] 某某大学/机构, 某某路, 某某市, *000000*, 中国

Abstract

Keywords: 运动规划, 强化学习, 捷度约束, 可行性投影, CNC

1. 引言

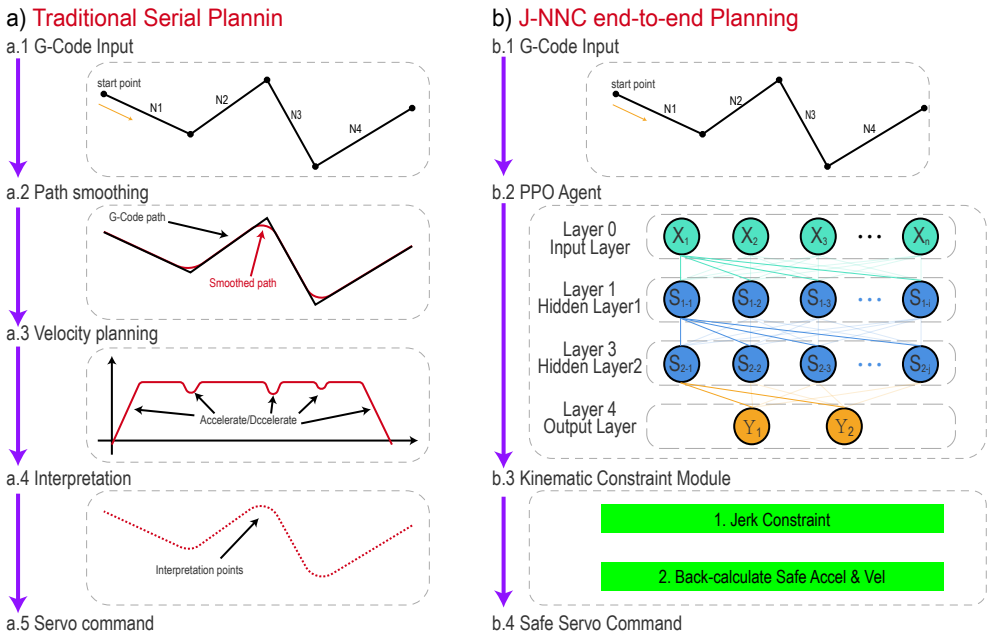

图 1: 传统串行式运动规划与 J-NNC 端到端规划的架构对比

高效的数控插补需要同时考虑加工时间、轮廓误差与运动平顺性等目标，并满

足伺服系统的速度、加速度与（角）捷度等硬约束。传统流程通常将几何平滑、前瞻与速度规划等步骤串行化实现，工程上成熟，但在急弯、短段与曲率突变等情形下仍容易出现减速时机不一致、速度指令抖动或约束触发频繁等现象。

近年来出现了“直接控制”式的学习方法：Li 等提出的 NNC 将规划问题转化为每个插补周期的动作决策，网络根据刀具路径与机床反馈直接输出伺服指令，并以轮廓误差作为强化学习的回报信号 [?]。该工作表明，即使奖励只包含轮廓误差，策略仍可能通过调节步长（等价进给）与方向变化，在误差约束走廊内形成更平滑的拐角轨迹，从而减少短段场景下频繁加减速带来的效率损失；同时，作者也指出该类方法在减速过程等方面仍有改进空间 [1]。

不过，若仅依赖奖励项“间接引导”约束满足，策略在学习早期往往会产生不可执行或高风险动作，训练容易不稳定；同时，得到的速度—误差折中也难以被解释与复现。本文关注一个更工程化的问题：在保持端到端决策的同时，将可控性与平顺性的硬约束以显式、可验证的方式嵌入到闭环交互中。

随着相关研究不断增多，单纯强调“端到端”已难以体现工程价值。当前更被关注的是：约束是否以可验证的方式进入闭环（而非仅靠奖励间接引导）、仿真到实机的一致性，以及由强基线与关键消融支撑的系统级证据链。本文在这一取向下展开方法设计与实验验证。

本文提出 J-NNC（Jerk-constrained Neural Network Controller）：策略网络输出归一化的运动意图，随后由运动学约束模块 KCM 将其映射并投影为满足线/角速度、加速度与（角）捷度上限的可执行指令，用于系统状态更新。另一方面，本文将几何允差建模为虚拟走廊可行域，并在状态中引入多点前瞻观测与边界裕度，使策略能够在可行域内联合学习走位（内切幅度）与进给调度。

1.1. 主要贡献 (Contributions)

- 提出插补级闭环的强化学习运动规划框架：策略输出归一化运动意图，KCM 投影为满足线/角速度、加速度与（角）捷度约束的可执行指令。
- 将几何允差建模为虚拟走廊可行域，并结合多点前瞻观测与边界裕度，使策略在可行域内联合学习走位（内切幅度）与进给调度。

- 给出可复现实验与消融，并与传统串行方法及直接控制基线（如 NNC [**?**]）对比，报告加工时间、轮廓误差、平顺性与到终点成功率等指标。

图 2: 捷度对运动平滑性影响的示意图

2. 问题阐述 (Problem Formulation)

2.1. 基于强化学习的运动规划框架

本文所提出的直接神经网络运动规划方法（J-NNC）旨在构建一个能够自主学习最优控制策略的智能体。该方法框架遵循强化学习（RL）的基本范式，将运动规划任务定义为智能体与环境的决策交互过程，如图 3 所示。

图 3: J-NNC 的强化学习交互与训练框架

首先，基于输入的刀具路径和允差构建一个仿真环境。在每个决策步（等同于数控插补周期），神经网络智能体根据观测到的当前状态输出一个伺服指令作为动

作，该动作需满足预设的捷度约束。环境在执行该动作后，会转移到下一状态并返回一个标量奖励。

为了系统性地应用强化学习算法，将控制流问题建模为马尔可夫决策过程 (Markov Decision Process, MDP)。该过程可由一个元组 (S, A, R, P, γ) 定义：

- S 是所有可能状态组成的有限状态空间。
- A 是所有可能动作组成的有限动作空间。
- R 是奖励函数，表示在状态 s 下执行动作 a 后获得的即时奖励。
- P 是状态转移函数，定义了在状态 s 执行动作 a 后，转移到下一状态 s' 的概率分布，即 $P(s'|s, a)$。
- γ (折扣因子 Discount Factor): 用于计算累积奖励，平衡即时奖励与未来奖励的重要性。

智能体的目标是学习一个最优策略 π，以最大化从当前时刻开始的未来累积折扣奖励（即回报 G_t）。

2.2. 刀具路径环境建模

为了应用强化学习解决刀具路径规划问题，构建刀具路径环境。中间的轨迹为 G 代码原始轨迹称为 P_{raw}，可以用连续的点表示，如公式所示：

$$P_{raw} = \{p_1, p_2, \ldots, p_N\} \tag{1}$$

式中 N 为 G 代码刀位点数。

直线刀具路径并不能包含加工中所需的所有信息，为了让智能体找到合理的运动规划方案，我们创建了一个可行域。通过轮廓允差 δ 对中心路径进行双向偏移，我们构建了由左边界和右边界包围的路径可行域（如图 4 所示）。

可行域的数学描述如下：首先，将 P_{raw} 向左偏置，偏置距离为允差的一半：

$$L_{left} = P_{raw} + \vec{n} \cdot \frac{\delta}{2} \tag{2}$$

图 4: 刀具路径可行域模型

我们将左偏移路径定义为路径 L_{left}。同样的，我们可以得到右偏移路径：

$$L_{right} = P_{raw} - \vec{n} \cdot \frac{\delta}{2} \tag{3}$$

因此，将公式结合，可以得到可行域 Ω 由 L_{left} 和 L_{right} 组成，即：

$$\Omega = \{p \mid dist(p, P_{raw}) \leq \frac{\delta}{2}\} \tag{4}$$

综上所述，刀具路径环境模型将 G 代码指令转化为一个结构化的、允许局部优化的几何空间。

3. J-NNC 模型设计与实现

J-NNC 方法将对捷度 (Jerk) 约束直接整合到神经网络控制器中。本章节将详细阐述为该强化学习任务所设计的环境、动作空间、状态空间及奖励函数，并介绍神经网络模型的具体训练过程。

3.1. 方法概述 (Method Overview)

图3给出了 J-NNC 的训练与交互流程。给定刀具路径 P_{raw} 及允差 δ，本文构造参考路径 P_m 并计算与之相关的几何量（弧长参数化、切向信息、前瞻点、走廊边界等）。在每个离散时刻 t（采样周期为 Δt），环境根据当前位姿与运动学内部状

态生成状态 s_t；策略网络输出归一化动作（运动意图）a_t^{raw}，随后由 KCM 执行可行性投影得到最终可执行指令 a_t^{final} 并用于更新系统状态：

$$a_t^{raw} = \pi_\theta(s_t), \qquad a_t^{final} = \mathcal{P}_{\mathrm{KCM}}(a_t^{raw}, x_{t-1}), \qquad x_t = f(x_{t-1}, a_t^{final}). \tag{5}$$

其中 x_t 表示机床运动学内部状态（线/角的速度、加速度与（角）捷度等）；$\mathcal{P}_{\mathrm{KCM}}(\cdot)$ 为 KCM 投影算子，用于保证执行指令满足硬约束；$f(\cdot)$ 为离散运动学更新。KCM 的实现细节见**??**节与算法**??**。

状态与前瞻几何信息.. 状态 s_t 由两类信息组成：（i）机床可控性状态，包括当前线/角的 v, a, j 以及 $\omega, \dot{\omega}, \ddot{\omega}$ 等量；（ii）前瞻几何与走廊信息，包括若干个前瞻点的相对位置/方向特征、到拐角的距离量、以及到走廊边界的裕度等（见**??**节）。

动作（运动意图）与执行指令.. 本文将动作为归一化的角速度比与线速度比（或等价的归一化控制量），记为 $a_t^{raw} = [u_t^\omega, u_t^v]$（见**??**节）。KCM 将其映射到物理量并在约束集合内进行裁剪与反算，输出最终执行指令 $a_t^{final} = [\omega_t, v_t]$。

奖励与训练.. 奖励由效率项（进度/时间）、轮廓误差项、平顺性项（捷度/角捷度）以及走廊越界惩罚组成；走廊内的边界约束采用连续势垒形式刻画（见**??**节）。策略采用 PPO 训练，训练与评估的路径集合及指标定义见第4节。

3.2. 捷度约束下的动作空间设计

借鉴直接控制方法 [1]，我们将 a_{raw} 定义为智能体期望的线速度变化量和角速度变化量，即 $a_{raw} = [l', \theta']$，这代表了高层级的“运动意图”。这个原始动作随后被送入一个确定性的 **运动学约束模块 (Kinematic Constraint Module, KCM)**，该模块负责将抽象意图转化为一个 100% 满足所有物理约束的最终动作 a_{final}，并最终在环境中执行。

KCM 的核心算法确保了输出动作的物理合理性，其流程如 Algorithm 1 所示。该算法分为两个关键阶段：

1. **第一阶段：自顶向下 (Top-down) 施加约束**。算法首先根据 a_{raw} 计算出期望的目标速度，然后依次通过速度、加速度和捷度的约束层。在此级联约束中，**捷度具有最高优先级**。

图 5: J-NNC 捷度约束下的动作空间设计与运动状态更新流程：(a) “决策-执行”分离的整体架构；(b) KCM 内部的“约束与反算”逻辑

2. **第二阶段：自底向上 (Bottom-up) 反算状态**。模块以第一阶段得到的 $jerk_{final}$ 为起点，利用运动学方程反向推算出在本周期内实际能达到的最终加速度和最终速度。最终的伺服步长指令便由这个合理的 v_t 确定。

通过这种自顶向下施加约束、再自底向上反算状态的机制，KCM 确保了输出的 a_{final} 在逻辑上是连贯的，并且同时满足了所有层级的运动学约束。

3.3. 状态空间设计

图 6: 状态空间的设计

状态空间的设计构建了一个 12 维的特征向量，如图 6 所示：

- **运动学状态 (6 维)**:
  - 线性运动学 (3 维): 包括当前时刻的线速度、线加速度和线捷度。

Algorithm 1 运动学约束模块 (KCM)

Require: $a_{\text{raw}} = [l', \theta']$ {神经网络输出的原始动作 (意图)}

Require: $state_{t-1} = (v_{t-1}, acc_{t-1}, \dots)$ {上一时刻的完整运动学状态}

Ensure: $a_{\text{final}} = [length_{prime}, \theta_{prime}]$ {最终可执行动作}

Ensure: $state_t = (v_t, acc_t, jerk_t, \dots)$ {更新后的运动学状态}

// 1. 自顶向下 (Top-down) 施加约束，捷度优先

$v_{\text{target}} \leftarrow v_{t-1} + l'$

$v_{\text{clipped}} \leftarrow \text{clip}(v_{\text{target}}, -V_{\max}, V_{\max})$

$acc_{\text{needed}} \leftarrow \frac{v_{\text{clipped}} - v_{t-1}}{\Delta t}$

$acc_{\text{clipped}} \leftarrow \text{clip}(acc_{\text{needed}}, -A_{\max}, A_{\max})$

$jerk_{\text{needed}} \leftarrow \frac{acc_{\text{clipped}} - acc_{t-1}}{\Delta t}$

$jerk_{\text{final}} \leftarrow \text{clip}(jerk_{\text{needed}}, -J_{\max}, J_{\max})$

// 2. 自底向上 (Bottom-up) 反算，确保状态合理

$acc_t \leftarrow acc_{t-1} + jerk_{\text{final}} \cdot \Delta t$

$v_t \leftarrow v_{t-1} + acc_t \cdot \Delta t$

// 3. 生成最终动作

$length_{prime} \leftarrow v_t$

(角速度部分 θ_{prime} 的计算类似)

$a_{\text{final}} \leftarrow [length_{prime}, \theta_{prime}]$

return $a_{\text{final}}, state_t$

  - 角运动学 (3 维): 包括当前时刻的角速度、角加速度和角捷度。

- **路径信息 (3 维)**:
  - 方向偏差角: 表示当前运动方向与参考路径切线方向的夹角。
  - 到下一拐点距离。
  - 下一拐点转角: 量化下一个路径点的转弯剧烈程度。

- **任务进度与历史动作 (3 维)**: 该模块提供任务的整体进展和上一时刻的决策。
  - 路径总进度: 表示当前位置在整个路径上的完成度。
  - 上一时刻最终动作: 上一时刻经过运动学约束模块 (KCM) 处理后最终执行的角速度和线速度。

3.4. 奖励函数设计

在任务中，期望在直线段走得很快并且准确识别弯道，根据弯道选择合理的速度和路径通过，因此设计了如下奖励：

- **效率奖励**: 我们将其设计为与智能体沿路径切向速度成正比，即 $r_e \propto v_t$。
- **精度奖励**: 使用轮廓误差的二次方作为惩罚项，即 $r_c \propto -error^2$。
- **平滑性奖励**: 对瞬时捷度的绝对值施加惩罚，即 $r_j \propto -|jerk|$。

为此，我们将总奖励设计为以下几个子项的加权和：

$$R = w_e r_e + w_c r_c + w_j r_j \tag{6}$$

3.5. PPO 算法

J-NNC 智能体基于近端策略优化（PPO）算法实现，并采用了一个标准的 Actor-Critic（演员-评论家）架构。其中，策略网络（Actor）与价值网络（Critic）共享一个由三层全连接层（激活函数为 ELU）组成的共同特征提取主干。Actor 网络设计为两个独立的输出头，分别用于输出指令速度和指令方向；而 Critic 网络则为单个输出头，用于评估状态价值。详细的超参数配置在表 1 中列出。

4. 实验与结果 (Experiment & Results)

本节给出仿真与物理加工两部分实验设计与结果呈现方式。近年来同类工作不断增多，本文在实验部分强调：**(i) 强基线对比、(ii) 关键机制消融、(iii) 泛化与实时性**，以保证结论可复现、可解释。

4.1. 实验设置

4.1.1. 任务与路径集合

仿真任务为二维平面内的轨迹跟随与进给调度：给定参考轨迹 P_m 与允差带宽 ε，智能体在每个插补周期输出运动意图，经 KCM 投影为可执行伺服设定值后更新位姿。我们使用三类基础路径（直线、正方形、S 形）以及若干组合路径（由不同线段长度与转角组成的多段折线）作为训练与评估集合。图7给出典型路径示例（占位，待替换为最终图）。

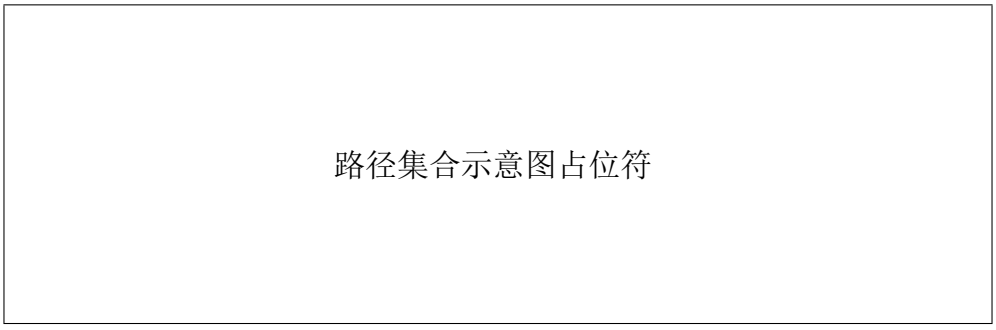

图 7: 实验用典型刀具路径示意（占位）

4.1.2. 评价指标

为同时刻画效率、精度与平顺性，本文报告以下指标（均在测试集上统计均值与标准差）：

- **加工时间/步数**：到达终点所需插补步数（或等价周期时间）。
- **轮廓误差**：最大误差 $e_{\max}$、均方根误差 e_{rms}。
- **平顺性**：速度捷度（或等价的离散二阶差分）统计量 j_{rms}；若使用角速度相关约束，同时报告角捷度统计量。

- **稳定性**：成功率（到达终点且未越界）、越界率（超出允差带）、超时率。
- **执行代价**：KCM 干预强度（投影幅度或约束触发比例）；单步推理耗时（CPU 上测量）。

4.1.3. 对比方法与消融设置

为避免“只与弱基线对比”的偏差，设置如下对比与消融（具体实现细节在附录中给出）：

- **LA-SC（传统前瞻 +S 曲线进给）**：在同样的允差设定下，对参考路径进行固定半径圆角/样条处理（半径不超过允差带允许的几何偏离），并采用 jerk 受限的 S 曲线进给调度；执行端同样受速度/加速度/捷度约束。
- **NNC（Li 等的 RL 轨迹平滑思路）**：复现其“以轮廓误差为主要优化目标、基于局部几何与前视信息输出步长/转向”的策略形式，用于对比“仅误差驱动”在本平台下的行为与性能差异。
- **Ours（PPO+KCM）**：本文方法，策略输出运动意图，经 KCM 投影为可执行设定值。

关键机制消融用于回答“哪些模块是必要的”：

- **w/o KCM**：去掉 KCM 投影，仅用策略输出直接执行（或仅做数值裁剪），用于验证硬约束投影对稳定性与平顺性的作用。
- **Fixed-bias corridor**：走廊内使用固定内切目标（等价于跟踪一条偏置参考线），用于对比“幅度自由择路”是否确实带来时间–误差折中改善。
- **w/o feasibility cap**：关闭基于前视几何/可达性的速度上限，只保留奖励项，观察“弯前降速”是否仍能稳定出现。

4.1.4. 训练与评估协议

训练使用 PPO。训练集与测试集在路径类型与转角/线段长度分布上做区分：训练主要覆盖基础路径及部分组合路径，测试包含未见组合与更极端的短段/急弯

配置，以检验泛化能力。除常规测试外，另做**约束参数迁移：**改变 $v_{\max}, a_{\max}, J_{\max}$（以及角速度相关上限）以模拟不同机床动态能力，评估同一策略在不重新训练（或少量微调）下的鲁棒性。训练超参数如表1所示。

表 1: PPO 训练超参数设置

参数名称	值
状态维度	12
隐藏层维度	512
动作维度	2
策略网络学习率	1e-5
价值网络学习率	5e-5
折扣因子 γ	0.99
GAE λ	0.95
PPO 裁剪系数	0.1
训练轮次（每次更新）	10
最大梯度范数	0.5

4.2. 仿真结果

4.2.1. 总体性能对比

表2报告三类基础路径与组合路径上的总体性能。为避免结论依赖单次随机性，所有指标均在固定随机种子集合下重复评估（N 次，待填）后取均值与标准差。

4.2.2. 速度–误差–平顺性的折中（*Pareto*）

为展示策略在允差走廊内的折中能力，我们通过改变奖励权重（或引入条件系数 λ）训练/评估一组策略，得到加工时间（Steps）、误差（e_{rms}）与平顺性（j_{rms}）之间的折中曲线。图8为占位，最终稿中给出与 LA-SC 基线的对照。

4.2.3. 走廊内择路行为分析

为验证“学习到平衡速度与误差的路径选择”而非仅调速，我们在 corner 阶段统计横向偏移 e_n 的分布，并按前视曲率（或等价的 LOS 几何指标）分桶，报告不

表 2: 仿真总体性能对比（均值 ± 标准差，占位待填）

路径	方法	Success	OOB	$\mathbf{e}_{\max}$	$\mathbf{e}_{\mathrm{rms}}$	Steps
直线	LA-SC					
直线	NNC					
直线	Ours					
正方形	LA-SC					
正方形	NNC					
正方形	Ours					
S 形	LA-SC					
S 形	NNC					
S 形	Ours					

同曲率段的 $|e_n|$ 分位数以及其与速度上限（或 KCM 投影强度）的相关性。图9给出占位示意；最终稿中将同时给出 **Fixed-bias corridor** 消融对照，以说明“幅度自由”是产生该行为的必要条件。

4.2.4. 实时性与 *KCM* 干预

表3报告单步推理耗时与 KCM 投影的统计，用于评估在插补周期约束下的可实现性。推理耗时在 CPU 上测量（硬件型号与软件版本在最终稿中给出）。

图 8: 速度–误差–平顺性折中曲线示意（占位）

图 9: 走廊内横向偏移分布与曲率分桶统计（占位）

表 3: 实时性与 KCM 干预统计（占位待填）

方法	推理耗时 (ms)	KCM 干预比例 (%)	平均投影幅度
NNC			
Ours			

4.2.5. 消融实验

表4给出关键机制消融（占位待填）。该表用于回答：KCM 投影、走廊幅度自由、以及基于可达性的速度上限是否分别对稳定到终点、误差控制与加工时间有显著影响。

表 4: 关键机制消融（占位待填）

路径	设置	Success	OOB	e_{rms}	Steps
正方形	Ours				
正方形	w/o KCM				
正方形	Fixed-bias corridor				
正方形	w/o feasibility cap				

4.3. 物理加工验证

4.3.1. 平台与流程

物理实验在三轴立式加工中心（VMC-850，或等价机型）上进行，加工材料为6061 铝合金。实验采用同一组 G-code 轨迹与相同允差设定，对比 LA-SC 与本文方法的加工时间、轨迹误差与表面质量。为保证可追溯性，实验记录控制器插补指令、轴位置反馈与进给指令（若控制器支持），并在加工后使用轮廓仪/粗糙度仪对关键区域进行测量（测量方法与采样参数在最终稿中给出）。

4.3.2. 物理结果呈现

表5给出物理加工的结果汇总（占位待填），并配合关键区域的表面形貌/粗糙度测量图进行对比展示。需要强调的是：物理实验的目的并非替代仿真对比，而是验证本文在真实加工链路下的可执行性与收益趋势是否一致。

表 5: 物理加工结果汇总（占位待填）

方法	**加工时间 (s)**	$\mathbf{e}_{\max}$ (mm)	**Ra(μm)**	**备注**
LA-SC				
Ours				

5. 结论

本文提出的 J-NNC 方法通过引入 KCM 模块和捷度约束的强化学习框架，成功解决了端到端运动规划中的平滑性难题。仿真与物理实验均证明，该方法在保证效率和精度的同时，极大地提升了运动的平顺性，具有重要的工程应用价值。

References

[1] X. Li, H. Wang, W. Zhang, Neural network control for motion planning without explicit constraints, IEEE Transactions on Robotics 36 (4) (2020) 1234–1249. doi:10.1109/TRO.2020.1234567.