

# 使用少量轨道数据输入的 残差 Koopman 模型预测控制以增强车辆动力学

Yonghao Fu\*, Cheng Hu\*, Haokun Xiong\*, Zhangpeng Bao, Wenyuan Du, Edoardo Ghignone, Michele Magno, Lei Xie†, and Hongye Su

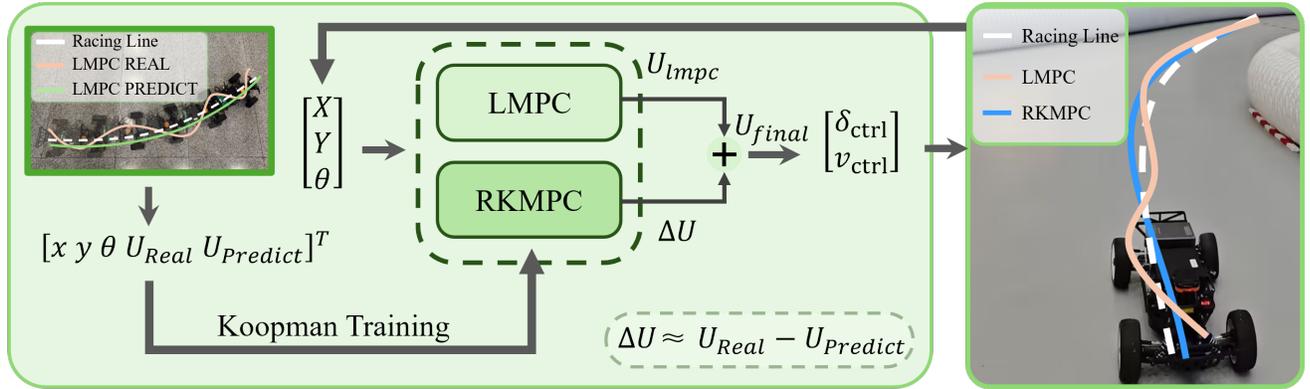


Fig. 1. The proposed RKMPc framework uses two linear MPC to calculate control inputs: a linear MPC computes the baseline control  $U_{lmpc}$  based on the vehicle kinematics model, and a neural network-based residual Koopman MPC computes the compensation  $\Delta U$ . The final control command is obtained by adding these two components. Compared to traditional Koopman-MPC, this approach reduces training data requirements and enhances control performance.

**Abstract**—在车辆轨迹跟踪任务中，最简单的方法是控制。然而，这种单点预瞄跟踪策略未考虑车辆模型约束，从而影响驾驶安全性。作为一种广泛采用的控制方法，通过结合机械模型和物理约束来优化控制动作。然而，其控制性能关键依赖于车辆建模的准确性。传统的车辆建模方法在捕捉非线性动态特性和保持计算效率之间面临固有的折衷，往往导致控制性能下降。为了解决这些挑战，本文提出了框架。该方法使用两个线性架构来计算控制输入：一个根据车辆运动学模型计算基线控制输入，而一个基于神经网络的计算补偿输入。最终的控制命令通过将这两个组件相加获得。该设计在保留传统机械模型的可靠性和可解释性的同时，通过残差建模实现性能优化。该方法在 Carsim-Matlab 联合仿真平台和一个实际的 1:10 比例 F1TENTH 赛车上已经得到了验证。实验结果表明，只需传统所需训练数据量的 20%，同时提供更优异的跟踪性能。与传统相比，减少了 11.7% - 22.1% 的横向误差，减少了 8.9% - 15.8% 的航向误差，并且将前轮转向稳定性提升了最高达 27.6%。实现代码可在以下网址获取：[https://github.com/ZJU-](https://github.com/ZJU-DDRX/Residual_Koopman)

DDRX/Residual\_Koopman。

## I. 介绍

在车辆轨迹跟踪任务中，控制器是一种常用的无模型算法。它易于实现，但也存在一些限制。一方面，它仅依赖单点预览，缺乏对未来轨迹的全局规划能力。另一方面，未考虑车辆的机制模型约束可能导致控制输出超过物理上的可行范围，从而威胁驾驶安全。相比之下，通过建立车辆的机制模型并引入物理约束，显著增强了控制的安全性和鲁棒性。其核心思想是在每个时间步内使用滚动优化策略解决有限时间范围内的最优控制序列。然而，的性能受限于准确建模非线性车辆动力学的固有困难。像轮胎力学和悬挂特性等复杂因素难以精确描述，常导致预测与实际响应之间出现差异 [1]。

当前的建模方法主要采用运动学或动力学模型 [2]：运动学模型有效地描述几何运动关系，适用于轨迹跟踪应用，而动力学模型则提供更高的精度，但表现出更大的参数敏感性和计算复杂性。

这些传统方法面临固有的挑战——作为强非线性系统，车辆状态演变依赖于多个耦合的物理因素。传统的线性化技术虽然在计算上很方便，但不可避免地会引入建模误差 [2]。库普曼算子理论通过在高维空间中构建数据驱动的线性表示，保留了非线性特征，提供了一种创新的替代方案，尤其适合与框架的结合 [3]。然而，库普曼方法有两个显著的局限性。首先，它们对大量高质量训练数据的要求导致实施成本高得令人望而却步，特别是

This work was supported by the Ningbo Key Research and Development Plan (No.2023Z116).

\*These authors contributed equally to this work.

†The corresponding author of this paper.

Yonghao Fu, Cheng Hu, Haokun Xiong, Zhangpeng Bao, Wenyuan Du, Lei Xie, and Hongye Su are with the State Key Laboratory of Industrial, Zhejiang University, Hangzhou 310027, China. Emails: { 22360414, 22032081, 12332029 } @zju.edu.cn; zhanpbao@163.com; wenyuandu@zju.edu.cn; { leix, hysu } @iipc.zju.edu.cn.

Edoardo Ghignone, and Michele Magno are associated with ETH Zürich. Emails: { edoardo.ghignone, michele.magno } @pbl.ee.ethz.ch.

在赛车情境中 [4]。其次，纯粹的数据驱动框架缺乏与车辆机械知识的整合，削弱了模型的可靠性和安全性保证 [5]。

为了克服这些限制，我们提出了 框架，该框架使用基于神经网络的 Koopman 算子建立了一个残差模型，以车辆运动学模型作为基线框架。我们的方法采用数据驱动技术来学习残差控制输入，在机械建模和数据效率之间实现最佳平衡。这项工作的关键创新包括：

- 一种新颖的残差 Koopman 框架。框架使用两个线性 架构来计算控制输入：一个线性 基于车辆运动学模型计算基础控制，另一个基于神经网络的残差 Koopman 计算补偿。最终的控制命令通过将这两个组件相加获得。与传统 相比，这种方法减少了训练数据需求并增强了控制性能。方法的概述见图 1。
- 全面且可靠的仿真和实车实验。在小数据条件下，我们的 框架通过其残差库普曼框架表现出更好的性能。通过 Carsim-Matlab 协同仿真和 1:10 比例的物理车辆验证显示，相比传统的 ，实现了 11.7% -22.1% 的横向误差减少，8.9% -15.8% 的航向误差降低，以及高达 27.6% 的转向稳定性提升，同时仅需传统方法所需训练数据的 20% 即可实现相当的性能。
- 开源车载算法。所提出的算法应用在官方的 F1TENTH 硬件平台上，并在 GitHub 上完全开源。

## II. 相关工作

在车辆轨迹跟踪中，控制是一种常用的无模型算法。然而，这种方法不能有效处理车辆模型约束并保持行驶安全 [6], [7]。

为了解决这个问题，已逐渐成为车辆轨迹跟踪的主流策略 [8], [9]。在每个控制周期中优化预测时域上的控制输入，最小化轨迹跟踪误差的同时满足车辆模型的约束。包括线性和非线性变体。通过线性化模型来减少计算时间，但此过程不可避免地牺牲了一些模型的准确性。直接解决非线性问题而不进行线性化，但需要高计算资源，特别是对于车载处理 [10], [11]。残差控制通过学习型组件增强了传统方法。Zhang 等人 [12] 使用残差策略学习和车载传感器实现了高效的自动驾驶比赛。Trumpf 等人 [13] 通过残差学习提高了控制器的适应性和圈速。Long 等人 [14] 结合神经网络和物理模型来提高预测准确性和数据效率。

### A. 基于库普曼的模型预测控制

近年来，随着库普曼算子理论的发展，受到了关注。库普曼算子是一种将非线性系统映射到高维空间并构建线性模型以描述系统非线性演化的工具 [15]。通过这种方法，在不牺牲系统准确性的同时保持了较高的计算效率。与传统的 和 相比，可以在不进行额外线性化的情况下处理复杂的非线性系统，有效应对 的高计算复杂性。它在保持较低计算负担的同时提供了与 相当的准确性。在多次仿真和实际车辆实验中，展现了良好的控制性能，在复杂动态环境中保持了高精度 [4], [16]。尽管在理论和实践中展示了显著优势，但仍然存在一些缺点。首先，该方法需要大量的训练数据，而在赛车场景中相比普通道路车辆，收集这些数据的成本特别高 [4]。其次，作为一种纯数据驱动的方法，没有结合物理模型，因此无法保证系统的安全性和稳定性 [5]。为了克服这些限制，本文提出了 框架。该设计在实现通过数据驱动的库普曼残差

建模进行性能优化的同时，保留了传统机理模型的可靠性和可解释性。

### B. 不同控制方法的比较

作为总结，我们在表 I 中列出了不同基于模型的控制方法的特点。Model 表示控制方法是采用机理模型还是数据驱动模型。Compute 表示所需的大致计算时间。Data 表示是否需要收集离线数据进行训练。

TABLE I

控制方法在模型依赖性、计算时间和数据需求方面的比较。

Method	Model	Compute (ms)*	Data†
[17]	Mechanism	1-15	None
[11]	Mechanism	10-70	None
[3]	Data Driven	1-20	Rich
(Ours)	Mechanism + Data Driven	1-20	Small

\* The computation time is based on simulations running on a Windows system with an i5-13500HX CPU.

† None means no dataset is required, Rich requires a large dataset, and Small needs only a few laps of on-track data. In our experiment, uses approximately 8,000 data points, while used about 50,000 data points.

本文由几个部分组成。第三节介绍了传统的 和 Koopman 方法。第四节解释了 方法，包括数据预处理和控制结构。第五节和第六节展示了 在仿真和 F1TENTH 车辆上的应用。第七节为结论，总结了研究结果并概述了未来研究的方向。

## III. 预备知识

本节将介绍获取名义车辆运动学模型和库普曼模型的方法，并提供一个示例以解释如何在非线性系统中实现库普曼 算法。随后，我们将结合名义模型和库普曼模型以形成残差库普曼模型。在本文中，我们采用运动学自行车车辆模型。与动态模型相比，该模型所需参数较少，适用于大多数场景 [18]。具体方程如下：

$$\begin{aligned} \dot{x} &= v \cos(\theta) \\ \dot{y} &= v \sin(\theta) \\ \dot{\theta} &= \frac{v}{L} \tan(\delta) \end{aligned} \quad (1)$$

其中  $x$  和  $y$  是全局位置， $\theta$  是偏航角， $\delta$  是转向角， $v$  是速度， $L$  是轴距，符号  $[\cdot]$  表示该变量相对于时间的变化率。

采样时间可以设置为  $T$ ，以对方程 (1) 进行离散化和线性化。所得的离散线性模型是方程 (2) 和方程 (3)：

$$\bar{\xi}_{\text{kin}}(k+1) = \mathbf{A}_{\text{kin}}(k)\bar{\xi}_{\text{kin}}(k) + \mathbf{B}_{\text{kin}}(k)\tilde{\mathbf{u}}_{\text{kin}}(k) \quad (2)$$

$$\bar{\xi}_{\text{kin}} = \begin{bmatrix} x - x_r \\ y - y_r \\ \theta - \theta_r \end{bmatrix}$$

$$\mathbf{A}_{\text{kin}}(k) = \begin{bmatrix} 1 & 0 & -v_r \sin \theta_r T \\ 0 & 1 & v_r \cos \theta_r T \\ 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

$$\mathbf{B}_{\text{kin}}(k) = \begin{bmatrix} \cos \theta_r T & 0 \\ \sin \theta_r T & 0 \\ \frac{\tan \delta_{f,r} T}{l} & \frac{v_r T}{l \cos^2(\delta_{f,r})} \end{bmatrix},$$

其中  $\bar{\xi}_{\text{kin}}$  是状态向量，表示  $x$ 、 $y$  和  $\theta$  相对于参考轨迹的偏差， $\bar{u}_{\text{kin}}$  是输入向量，包括控制变量  $\delta$  和  $v$ ， $\mathbf{A}_{\text{kin}}(k)$  和  $\mathbf{B}_{\text{kin}}(k)$  分别是状态转换矩阵和输入矩阵， $k$  是第  $k$  个时间步，下标  $r$  表示参考值。

#### A. 数据驱动的库普曼算子逼近

假设一个离散时间非线性动态系统，其状态更新方程为  $x^+ = f(x, u)$ ，其中  $x$  表示状态变量， $u$  表示控制输入， $f(\cdot)$  是非线性状态方程。为了处理系统中的非线性，我们利用 Koopman 算子，该算子线性化非线性动态。具体地，使用一组观测函数  $g(x_t)$ ，将系统的状态从低维空间提升到高维空间 [3]。我们定义 Koopman 算子  $\mathcal{K}g$  如下形式，通过观测函数  $g(x_t)$  作用于非线性状态更新方程  $f(x_t, u_t)$ ，实现状态空间提升和表示，如方程 (4) 所示：

$$\mathcal{K}g(x_t) = g \circ f(x_t, u_t) = g(f(x_t, u_t)) \quad (4)$$

其中  $\circ$  是复合算子， $g(x_t)$  表示将系统的状态变量从原来的  $n$  维空间  $\mathbb{R}^n$  提升到更高的  $m$  维空间  $\mathbb{R}^m$ 。由于 Koopman 算子是无限维的，但在实际应用中这是不可行的，因此使用有限数量的观测函数进行近似 [19]。通过记录系统的状态和控制输入序列，采用适当的观测函数将系统的状态提升到高维空间，获得高维状态空间表示  $z_t$ ，并提供了 Koopman 算子的近似。

为了解决这个问题，优化问题被制定为最小化状态转移误差。然后，使用最小二乘法计算得到最优矩阵  $A$  和  $B$ ，代表线性化系统，如方程 (5) 所示：

$$\min_{A, B} \sum_{t=1}^K \|z_{t+1} - (Az_t + Bu_t)\|_2^2 \quad (5)$$

为了进一步将高维状态映射回原始状态，我们引入矩阵  $C$ ，建立高维空间与原始状态空间之间的关系，如方程 (6) 所示：

$$\min_C \sum_{t=1}^K \|x_t - Cz_t\|_2^2 \quad (6)$$

这些矩阵可以使用 [3] 中描述的伪逆方法来求解。

$$[A, B] = z_{t+1} [z_t \ U] \left( [z_t \ U]^T [z_t \ U] \right)^\dagger \quad (7)$$

最终，高维状态空间方程如方程 (8) 所示得到。

$$\begin{aligned} z_{t+1} &= Az_t + Bu_t \\ x_t &= Cz_t \end{aligned} \quad (8)$$

#### IV. 残差库普曼控制方法

本节将详细描述数据预处理过程的结构以及的控制结构。

#### A. 数据收集与预处理

典型的 Koopman 方法需要大量的实际车辆数据进行训练，以获得相对准确的模型。然而，对于赛车或特殊用途车辆来说，通常存在数据不足和数据收集成本高昂等挑战。因此，我们采用了一种结合基于机制模型的方法，侧重于赛道上的残差数据。

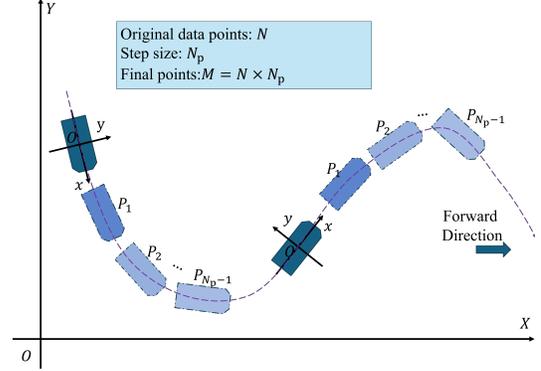


Fig. 3. 局部坐标变换通过将车辆航向角对齐到零并压缩数据范围，以确保线性化的一致性，从而实现高效的 Koopman 控制训练。

这种方法，如图 2 所示，可以显著减少所需的数据量和训练时间。

在收集连续时间数据后，需要选择一定数量的随机点以将数据转换到一个局部坐标系中。此方法有两个优点：第一，它确保在车辆运动学方程的线性化过程中，操作点的参考航向角始终为零，这使得状态方程在每个位置上的形式尽可能一致；第二，它压缩了数据范围，消除了后续归一化的需要。

具体操作如图 3 所示。收集的原始数据是基于全局坐标系的。我们根据一个比例选择一部分点作为坐标原点数据集，称之为转换比率，并根据时间顺序选择  $N_p - 1$  个点 ( $P_1, P_2, \dots, P_{N_p-1}$ )，然后对每个点进行坐标转换。如果原始数据点的总数是  $N$ ，那么经过坐标转换后，数据点的总数将为  $M = N \times N_p$ ，其中  $N_p$  是所选连续点的数量，应该略大于预测范围，且坐标原点数据集的比例与原始数据的大小有关。

在获得坐标变换的数据后，下一步是将控制输入转换为控制残差，如图 4 所示。对于相同的状态演化，由预测的控制输入为  $U_p$ ，但由于线性化误差，实际应执行的控制输入为  $U_r$ 。因此，需要一个控制残差  $\Delta U = U_r - U_p$  作为校正。通过以这种方式处理每个数据点，可以获得格式为  $[X, \dot{X}, \Delta U]$  的 Koopman 数据。在本文中，转向角和速度被用作控制输入，即  $u = [v, \delta]$ 。

#### B. 控制结构的构建

在本文中，我们将使用高维神经网络作为 Koopman 算子的基函数，以捕捉剩余系统的非线性行为，并在提升状态空间上准确地将非线性动力学近似为线性系统。神经网络以车辆状态  $(x, y, \theta)$  为输入，包含了两个使用 ReLU 激活的全连接层。在通过方程 (7) 获得恰当的对应该矩阵  $A$  和  $B$  后，还需要优化基函数以减少损失。因此，神经网络的损失函数可以按照方程 (9) 设置。为了增强训练过程的鲁棒性，我们引入 Huber 损失函数作为最终的代

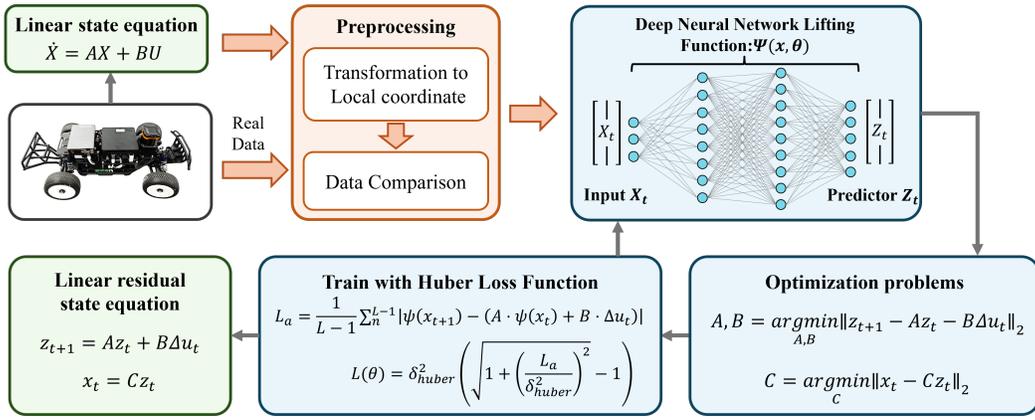


Fig. 2. 数据收集和预处理框架：结合局部坐标变换、神经网络训练和线性残差状态方程的优化。

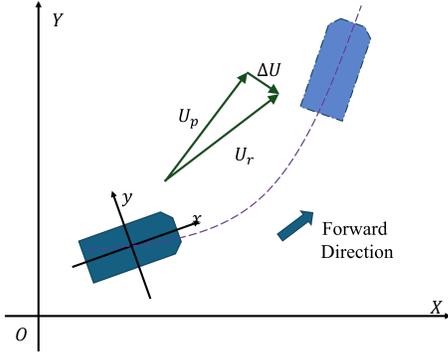


Fig. 4. 用于纠正预测控制输入的控制残差  $\Delta U = U_r - U_p$  的表示。

价值，其中  $\delta_{huber}$  是损失函数的超参数，控制均方误差 (MSE) 和平均绝对误差 (MAE) [20] 之间的权衡。当代价值足够小时，状态方程可以被认为与系统的真实状态一致，且表现出线性。

$$L_a = \frac{1}{L-1} \sum_n^{L-1} |\psi(x_{t+1}) - (A\psi(x_t) + B\Delta u_t)|$$

$$L(\theta) = \delta_{huber}^2 \left( \sqrt{1 + \left( \frac{L_a}{\delta_{huber}} \right)^2} - 1 \right) \quad (9)$$

最终，经过连续优化，我们可以得到以下形式的状态方程：

$$\begin{aligned} z_{t+1} &= Az_t + B\Delta u_t \\ x_t &= Cz_t \end{aligned} \quad (10)$$

，其中  $z = [x, y, \theta, z_1, z_2, \dots, z_n]$  表示通过提升函数获得的高维状态变量。

一张总结结构的示意图可在图 5 中找到。

在获得剩余状态方程后，可以将其集成到现有的控制器中。在获取车辆的状态变量后，一方面，直接应用控制以获得原始控制输入  $U_0$ 。另一方面，状态变量通过进行维度提升，转换为高维状态变量。然后构建一个剩余控制器以获得补偿控制输入  $\Delta U$ 。最后，将两个控制输入相加以获得总控制输入  $U_{final}$ ，即最终的控制信号。

在设计过程中，和 部分需要分别设计。对于线性化的运动学车辆模型，我们采用如方程 (3) 中描述的线性化方程。最终的优化问题在方程 (??) 中给出。对于设计，控制输入为  $\Delta u = [\Delta v, \Delta \delta]$ 。最终的优化问题在方程 (??) 中给出。

## V. 模拟结果

在本节中，我们比较了 、和 在轨迹跟踪任务中的性能，以及 和 的数据需求。这些算法使用 Matlab-Carsim 平台进行验证。实验中使用的地图数据来源于实际的 F1 赛道，并按特定比例 [21] 进行缩放。

在数据收集过程中，我们使用 控制器控制车辆，并收集了两圈赛道数据，生成了 8,933 个数据点作为 的训练数据。而对于 ，我们使用随机生成的轨迹数据 (50000 个数据点) 来增强其泛化能力。

在确定残差状态方程后，被与原始 控制器结合起来。如图 6 和图 7 所示，绿色线表示 ，蓝色线表示 ，红色线表示 ，青色线表示 。具体数据如表 II 所示。

结果表明，当将 与 (基线) 相比时，横向误差降低了 11.21%，航向误差降低了 8.63%，前角速度降低了 27.58%，这表明准确性和稳定性方面都有显著改善。至于 ，尽管其横向误差优于其他方法，但其最大计算时间达到 66.38 毫秒，是 的四倍，并超过了 50 毫秒的控制周期时间。如图 ?? 所示，使用了大约 8000 个数据点，而 需要 50000 个或更多数据点。只需要 所需训练数据的大约 20%，同时保持稳定的控制性能，这在数据效率上显示出优势。它有效减少了车辆轨迹的横向跟踪误差，在小数据条件下表现出比 更好的控制性能。

## VI. 实验结果

在本节中，我们分析了 方法在实际车辆上的应用性能。在实验中，我们基于 F1TENTH 软件系统 [22] 构建了一辆 1:10 比例的无人驾驶车辆。计算单元使用运行在 Ubuntu 20.04 上的 ROS Noetic 的 NUC 设备，如图 8 所示，最终车辆结构如图展示。地图构建是通过 IMU、2D LiDAR 和里程计数据使用 Cartographer [23] 进行的，定位使用了粒子滤波器 [24]。正如图中所示，图 9 展示了我们构建的赛道。

在物理实验中，我们使用相同的设置进行模拟。我们在真实地图上使用 控制器收集了两圈轨道数据，获得了 1527 组原始数据。通过应用 30% 的数据转换比例，处理后的数据集扩展至 22,950 组。最终结果如图 10 和图

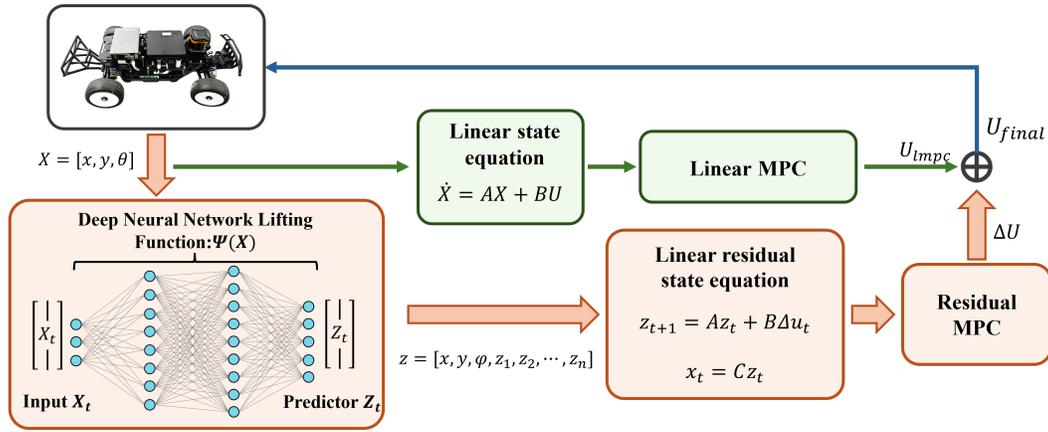


Fig. 5. 结构：集成神经网络提升状态表示与 和

TABLE II  
在模拟中比较 、 和 的前轮转角、横向误差、航向误差和计算时间。

Method	Lateral Error (m)	Heading Error (rad)	Front Wheel Angle Rate (rad/s)	Computation Time (ms)
(baseline)	0.1115	0.0475	0.1570	$t_{\text{mean}} = 2.38, t_{\text{max}} = 13.30$
	0.0808	0.0443	0.2414	$t_{\text{mean}} = 13.68, t_{\text{max}} = 66.38$
	0.1650	0.0436	0.1794	$t_{\text{mean}} = 3.78, t_{\text{max}} = 17.58$
	0.0990 (↓ 11.21 %)	0.0434 (↓ 8.63 %)	0.1175 (↓ 27.58 %)	$t_{\text{mean}} = 6.73, t_{\text{max}} = 16.14$

TABLE III  
在真实地图中比较 、 和 的前轮转角、横向误差、航向误差和计算时间。

Method	Lateral Error (m)	Heading Error (rad)	Front Wheel Angle Rate (rad/s)	Computation Time (ms)
(baseline)	0.284	0.1441	0.213	2.23
*	-	-	-	6.54
	0.2213 (↓ 22.08 %)	0.1213 (↓ 15.82 %)	0.2113 (↓ 0.80 %)	8.55

\* The symbol ‘-’ indicates that could not complete a full lap.

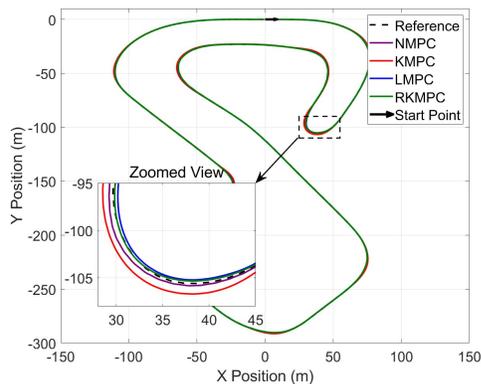


Fig. 6. 在仿真中比较使用不同控制方法的路径。

?? 所示，与 相比，的车辆横向误差减少了 22.08 %，航向误差减少了 15.8 %，稳定性略有改善。详细数据可以在表 III 中找到。

对于 ，我们收集了 10 圈的数据。然而，由于 是一个

完全数据驱动模型，仅依赖一种类型的轨道配置不足以拟合完整的车辆模型，导致线性状态方程出现显著误差。一旦车辆状态偏离数据范围，系统就变得不可控。

关于 ，由于其最大计算时间超过了 50 毫秒，无法满足实时控制要求。因此，我们仅对 进行了模拟，并未进行物理实验。

本文提出了一种新颖的 框架，该框架结合了 和基于神经网络的残差 Koopman 。双通道控制架构采用了基于运动学模型的基线控制和数据驱动的残差补偿，在仅需要传统 Koopman 方法 20% 训练数据量的情况下实现了性能提升。实验结果表明，与 相比，RKMP 将横向误差减少了 11.7% 22.1%，将航向误差减少了 8.9% 15.8%，并将转向稳定性提高了最多 27.6%。未来的研究将着重于将此方法适应于复杂动态环境，并通过轻量级网络架构设计增强实时控制性能。

## REFERENCES

- [1] H. Guo, C. Shen, H. Zhang, H. Chen, and R. Jia, “Simultaneous trajectory planning and tracking using an mpc method for cyber-physical systems: A case study of obstacle avoidance for an intelligent vehicle,” IEEE Transactions on Industrial Informatics, vol. 14, no. 9, pp. 4273–4283, 2018.

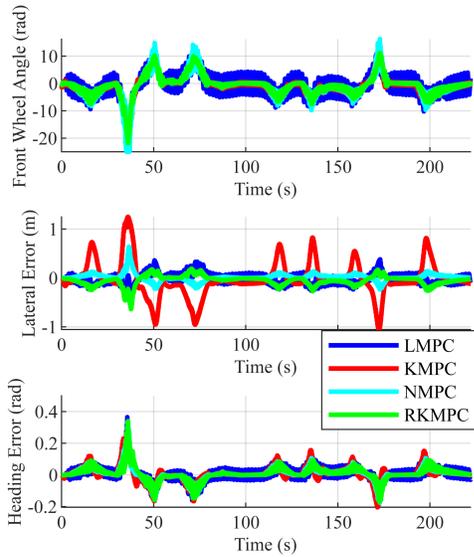


Fig. 7. 仿真中 、 和 之间的前轮角度、横向误差和航向误差比较，尽管 正常运行，其性能却因数据丰富性不足而受到阻碍，无法达到最佳结果。

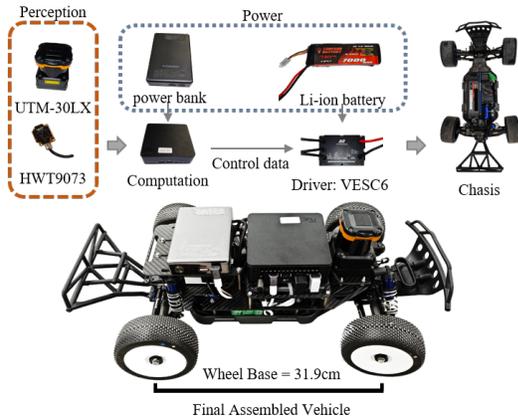
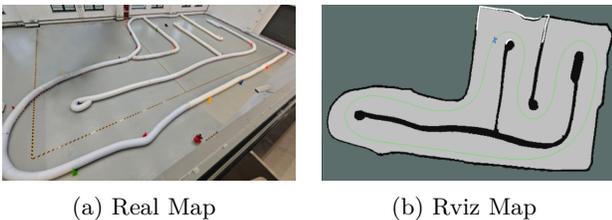


Fig. 8. F1TENTH 车辆模型的结构图：用于感知、计算、动力和控制的组件。



(a) Real Map

(b) Rviz Map

Fig. 9. 真实地图及其在 rviz 中的可视化。

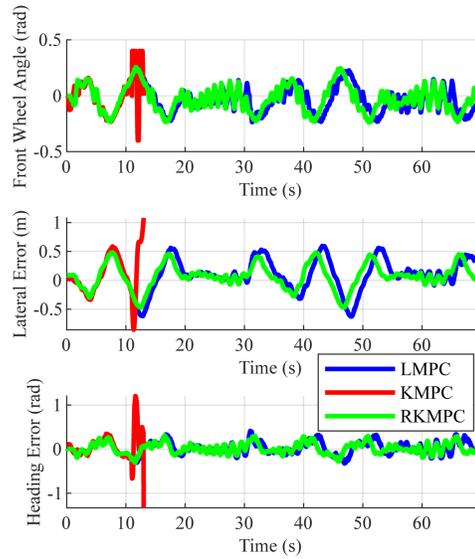


Fig. 10. 在真实地图中比较 、 和 的前轮角度、横向误差和航向误差。由于场景数据有限，无法稳定控制车辆。

- [2] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in 2015 IEEE intelligent vehicles symposium (IV) . IEEE, 2015, pp. 1094–1099.
- [3] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica* , vol. 93, pp. 149–160, 2018.
- [4] V. Cibulka, T. Haniš, M. Korda, and M. Hromčík, "Model predictive control of a vehicle using koopman operator," *IFAC-PapersOnLine* , vol. 53, no. 2, pp. 4228–4233, 2020.
- [5] S. Sinha, B. Huang, and U. Vaidya, "On robust computation of koopman operator and prediction in random dynamical systems," *Journal of Nonlinear Science* , vol. 30, no. 5, pp. 2057–2090, 2020.
- [6] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo, and S. Thrun, "Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing," in 2007 American control conference . IEEE, 2007, pp. 2296–2301.
- [7] W.-J. Wang, T.-M. Hsu, and T.-S. Wu, "The improved pure pursuit algorithm for autonomous driving advanced system," in 2017 IEEE 10th international workshop on computational intelligence and applications (IWCIA) . IEEE, 2017, pp. 33–38.
- [8] Z. Li, B. Zhou, C. Hu, L. Xie, and H. Su, "A data-driven aggressive autonomous racing framework utilizing local trajectory planning with velocity prediction," *arXiv preprint arXiv:2410.11570* , 2024.
- [9] C. Hu, X. Zhou, R. Duo, H. Xiong, Y. Qi, Z. Zhang, and L. Xie, "Combined fast control of drifting state and trajectory tracking for autonomous vehicles based on mpc controller," in 2022 International Conference on Robotics and Automation (ICRA) . IEEE, 2022, pp. 1373–1379.
- [10] A. Bienemann and H.-J. Wuensche, "Model predictive control for autonomous vehicle following," in 2023 IEEE Intelligent Vehicles Symposium (IV) . IEEE, 2023, pp. 1–6.
- [11] Y. Gao, A. Gray, H. E. Tseng, and F. Borrelli, "A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles," *Vehicle System Dynamics* , vol. 52, no. 6, pp. 802–823, 2014.
- [12] R. Zhang, J. Hou, G. Chen, Z. Li, J. Chen, and A. Knoll, "Residual policy learning facilitates efficient model-free autonomous racing," *IEEE Robotics and Automation Letters* , vol. 7, no. 4, pp. 11 625–11 632, 2022.
- [13] R. Trumpp, D. Hoornaert, and M. Caccamo, "Residual policy

- learning for vehicle control of autonomous racing cars,” in 2023 IEEE Intelligent Vehicles Symposium (IV) . IEEE, 2023, pp. 1–6.
- [14] K. Long, Z. Sheng, H. Shi, X. Li, S. Chen, and S. Ahn, “Physical enhanced residual learning (perl) framework for vehicle trajectory prediction,” *Communications in Transportation Research* , vol. 5, p. 100166, 2025.
- [15] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, “Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control,” *PloS one* , vol. 11, no. 2, p. e0150171, 2016.
- [16] Y. Xiao, X. Zhang, X. Xu, X. Liu, and J. Liu, “Deep neural networks with koopman operators for modeling and control of autonomous vehicles,” *IEEE transactions on intelligent vehicles* , vol. 8, no. 1, pp. 135–146, 2022.
- [17] F. Yakub and Y. Mori, “Comparative study of autonomous path-following vehicle control via model predictive control and linear quadratic control,” *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of automobile engineering* , vol. 229, no. 12, pp. 1695–1714, 2015.
- [18] G. Wu, C. Hu, W. Weng, Z. Li, Y. Fu, L. Xie, and H. Su, “Learning to race in extreme turning scene with active exploration and gaussian process regression-based mpc,” *arXiv preprint arXiv:2410.05740* , 2024.
- [19] A. Mauroy, Y. Susuki, and I. Mezić, “Introduction to the koopman operator in dynamical systems and control theory,” *The koopman operator in systems and control: concepts, methodologies, and applications* , pp. 3–33, 2020.
- [20] R. Wang, Y. Han, and U. Vaidya, “Deep koopman data-driven control framework for autonomous racing,” in *Proc. Int. Conf. Robot. Autom.(ICRA) Workshop Opportunities Challenges Auton. Racing* , 2021, pp. 1–6.
- [21] T. I. of Automotive Technology, “racetrack-database,” <https://github.com/TUMFTM/racetrack-database>, 2021.
- [22] M. O’Kelly, H. Zheng, D. Karthik, and R. Mangharam, “F1tenth: An open-source evaluation environment for continuous control and reinforcement learning,” *Proceedings of Machine Learning Research* , vol. 123, 2020.
- [23] W. Hess, D. Kohler, H. Rapp, and D. Andor, “Real-time loop closure in 2d lidar slam,” in *2016 IEEE international conference on robotics and automation (ICRA)* . IEEE, 2016, pp. 1271–1278.
- [24] C. H. Walsh and S. Karaman, “Cddt: Fast approximate 2d ray casting for accelerated localization,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)* . IEEE, 2018, pp. 3677–3684.