

# 一种循迹控制参数调节器及其训练集构建方法\*

赵克刚, 区伟麟, 张政, 梁志豪

(华南理工大学机械与汽车工程学院, 广州 510641)

**[摘要]** 为提升智能车循迹控制器在变工况应用时的控制精度, 控制器一般采用基于工况特征的多维控制参数表。工程师在对多维控制参数表进行人工整定时, 工作量较大且整定效果不尽如人意。为了能使循迹控制器获得参数动态调整能力, 本文基于径向基(radial basis function, RBF)神经网络提出了车速与曲率自适应参数调节器。针对构建调节器训练集过程中遇到的实车测试交互次数过多、整定工作量过大的问题, 本文提出了一种基于蒙特卡罗学习控制概率推理(Monte-Carlo probabilistic inference for learning control, MC-PILCO)算法的训练集构建方法, 根据车速对训练集构建过程中涉及到的典型工况进行分组, 每个车速工况分组内所有不同曲率工况均使用该车速下跟踪直线场景采集到的数据训练出来的动力学模型进行参数整定, 通过共享模型的方式实现了实车交互次数的减少。实车实验表明, 在中低速工况下, 本文提出的参数自适应循迹控制器相比参数固定的控制器有更好的横向轨迹跟踪效果。

**关键词:** 轨迹跟踪控制; 径向基神经网络; 多维控制参数; 训练集构建; MC-PILCO

## A Tuner of Trajectory Control Parameters and the Construction Method of its Training Set

Zhao Kegang, Ou Weilin, Zhang Zheng & Liang Zhihao

School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510641

**[Abstract]** To improve the control accuracy of intelligent vehicle tracking controllers in variable operating conditions, controllers generally use multidimensional control parameter tables based on operating condition characteristics. When engineers manually adjust multidimensional control parameter tables, the workload is large and the tuning effect is not satisfactory. In order to enable the tracking controller of dynamic parameter adjustment capability, in this paper a vehicle speed and curvature adaptive parameter tuner is proposed based on radial basis function (RBF) neural network. Besides, a training set construction method based on Monte Carlo Probabilistic Inference for Learning Control (MC-PILCO) algorithm is proposed to address the problems of excessive real vehicle testing interactions and heavy tuning workload encountered during the training of tuner. By grouping typical operating conditions based on vehicle speed in the construction process of the training set, all different curvature working conditions within each vehicle speed working condition group are trained using the dynamic model trained on the data collected from tracking the straight-line scene at that vehicle speed for parameter tuning. By sharing the model, the number of real vehicle interactions is reduced. Real vehicle experiments show that the parameter adaptive tracking controller proposed in this paper has better lateral trajectory-tracking performance compared to controllers with fixed parameters under medium and low speed conditions.

**Keywords:** trajectory tracking control; RBF neural network; multidimensional control parameters; training set construction; MC-PILCO

\* 人工智能与数字经济广东省实验室(广州)(PZGDL2023001)资助。

原稿收到日期为2024年07月24日, 修改稿收到日期为2024年09月02日。

通信作者: 梁志豪, 博士, E-mail: meliangzh2021@mail.scut.edu.cn。

## 前言

汽车控制器的控制精度除了受控制算法本身控制质量高低影响以外,还与控制器自身控制参数的整定息息相关。车辆的控制器均包含数量较多的控制参数,当前在汽车产品开发过程中,控制器参数的整定通常依赖工程师结合理论知识及经验人工完成<sup>[1]</sup>。

面对实际工程应用中存在的多工况,一套参数往往无法满足要求。国内外有一些学者已经研究过车辆领域参数自适应调节的可行性。Kothari等<sup>[2]</sup>早在1997年就基于RBF神经网络设计了自适应电力系统稳定器,在大范围的负载条件及大扰动下提供了比传统的自适应电力系统稳定器更好的动态性能。杨建伟等<sup>[3]</sup>提出了由1个RBF神经网络辨识器和4个神经网络控制器组成的半主动悬挂自适应控制系统,该系统较好地提高了道路友好性和平顺性。陈刚等<sup>[4]</sup>针对一般PID(proportional-integral-derivative)控制器控制速度时存在的波动大、参数难以调节的问题,提出了自适应的模糊PID车速控制器,可以自适应调整速度误差及车速变化率。从以上现有成果可以做出合理假设:使控制器具备自适应参数调节功能具备可行性且对循迹控制精度的提升有帮助。

参数调节功能的实现可以通过拟合出模型——参数调节器来完成,调节器的训练需要使用人工的方式整定出典型工况下的参数,费时费力。针对控制器参数人工整定较为繁琐的问题,学术界已经有学者进行过参数自动整定方法的探索。Zhang等<sup>[5]</sup>通过规划路径和实际路径的误差实现了前视距离参数的自动调整。Mooren等<sup>[6]</sup>针对前馈控制器提出了一种参数迭代整定方法。Amadio等<sup>[7]</sup>提出的MC-PILCO作为一种当前最为先进且灵活的有模型强化学习算法,在模拟倒立摆场景的数值比较中表现出较好的数据效率和整定性能。可以看出,强化学习运用于参数的自动整定具备潜力,MC-PILCO作为一种数据效率较高的有模型强化学习算法,同样具备潜力作为替代人工整定以获得调节器的训练集的核心算法。

为此,本文做了如下工作:基于径向基神经网络提出了一种应用在智能车循迹控制器上的车速与曲率自适应参数调节器;提出了一种新的基于MC-

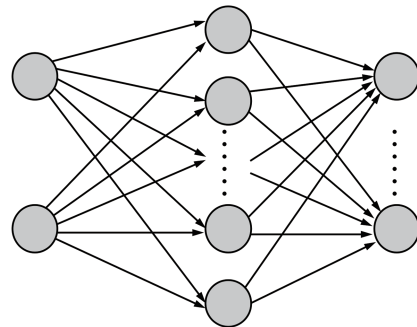
PILCO算法的调节器训练集构建方式,以解决调节器训练集构建时存在的与实车交互工作量较大的问题。

## 1 自适应参数调节器

### 1.1 基于RBF神经网络的自适应参数调节器

RBF神经网络是一种使用径向基函数作为激活函数的3层前向网络,包括输入层、中间隐藏层和输出层<sup>[8]</sup>。其具备与函数逼近理论相吻合、结构简单<sup>[9]</sup>、学习收敛速度快的优点,能有效避免局部极小值。

影响控制器的泛化性的因素有很多,包括车速、曲率、路面附着系数、路面平整度、路面坡度等,考虑到车速和曲率直接影响车辆的行驶稳定性和运动轨迹,且车速可以通过传感器进行测量,曲率可以通过规划轨迹计算得到,相对而言容易得到,因此将它们作为影响控制器泛化性的主要因素,设计了车速与曲率自适应参数调节器。其结构图如图1所示。



输入:  $[\hat{v}_p, k_r]^T$  多二次函数激活函数 控制参数:  $\psi$

图1 车速与曲率参数自适应调节器结构图

从图1中可以看出,车速与曲率自适应调节器的输入为二维的车速和道路曲率向量,输出为  $n$  维的车辆横纵向控制参数,  $n$  为控制器控制参数个数。中间隐藏层的个数为

$$\sum_{i=1}^{n_{center}} numsi + n_{center} \quad (1)$$

式中:  $numsi$  为车速  $v_i$  下对应的曲率工况个数;  $n_{center}$  为隐藏层内的径向基函数个数,个数越多,参数调节器的拟合越精确。对于本参数调节器而言,输出层中计算公式中只有权重系数  $\omega$  以及径向基函数参数  $\gamma$  两个未知量,因此可以通过训练数据对神经网络中的权重系数矩阵  $\mathbf{W}$  和径向基函数参数向量  $\boldsymbol{\gamma}$  进行优化。

### 1.2 调节器训练需求分析

调节器需要建立训练集进行训练,训练完毕后通过插值的方式根据轨迹跟踪过程中的实时工况(车速与曲率)自适应调整控制器的参数,控制参数是连续变化的。调节器训练集构建的过程就是获取典型工况下的控制参数的过程。对于车速与曲率自适应参数调节器而言,一个车速工况往往包含多个不同的道路曲率工况,这导致典型工况数量较多。如果采用人工整定或者传统强化学习的方式进行参数的获取,需要逐一对各个工况进行真实车辆系统与控制器的交互,整定工作量较大。因此,需要找到一种算法能够减少实车交互量。

若使用基于模型的强化学习算法进行参数的自动整定,对模型有以下两点要求:若所要搭建的模型对小样本泛化能力不强,进行模型搭建前的数据采集工作往往也需要耗费大量的人力物力以采集足够多的数据,因此模型最好具备对小样本足够的泛化能力,以保证算法学习效果;参数整定过程中涉及到多种不同曲率的工况,使用确定性模型进行轨迹预测时轨迹本身存在的不确定性会导致模型输出的结果不可靠,且难以对不确定性进行进一步的分析,因此模型最好是不确定性模型。

## 2 跟踪控制器的自动整定框架

本章中将介绍基于MC-PILCO算法的控制参数自动整定框架。内容包括MC-PILCO算法在处理参数自动整定问题时的优势、建模和状态预测流程,以及自动整定框架的组成。

### 2.1 MC-PILCO 算法

学习控制概率推理(probabilistic inference for learning control, PILCO)是一种具有较高数据效率的有模型强化学习算法<sup>[10]</sup>,使用高斯过程对环境进行建模,并通过梯度下降的方式进行参数优化调整。PILCO的算法框架主要分为概率动力学建模(learning a probabilistic dynamics model)、策略评估(policy evaluation, PEV)、策略优化(policy improvement, PIM)3部分,具体可以参考文献[11]。

Amadio等<sup>[7]</sup>提出的MC-PILCO是一种改进的PILCO算法,其主要步骤如图2所示。与传统PILCO算法使用矩匹配进行车辆状态的长期预测不同的是,MC-PILCO使用了如图3所示<sup>[7]</sup>的基于粒子的方法,有效避免了矩匹配只能对单峰分布进行建模导

致对于系统的长期动力学假设过于粗糙的问题,同时能一定程度保证对未见过的数据的泛化性。此外,MC-PILCO方法还在策略优化期间使用dropout以防止过拟合<sup>[12]</sup>。

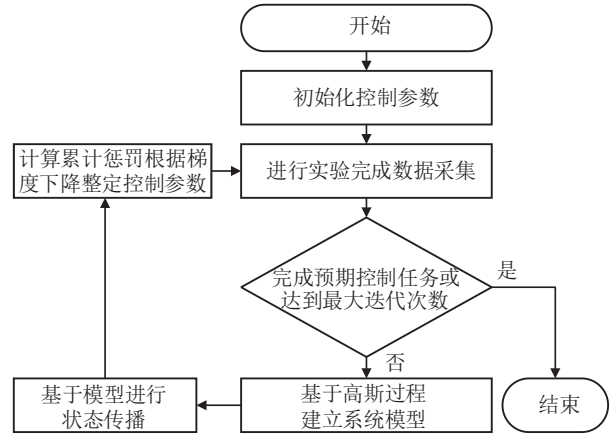


图2 MC-PILCO主要步骤

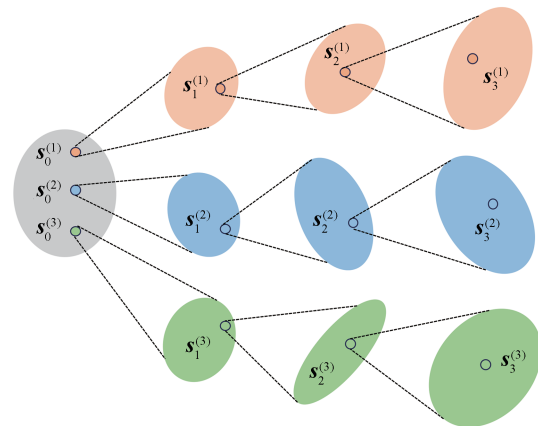


图3 MC-PILCO粒子传播车辆状态示例图

对于参数自动整定工作而言,MC-PILCO更有其独特的优势。其采用高斯过程(Gaussian process, GP)进行建模。高斯过程是一种基于贝叶斯理论和统计学习理论发展起来的机械学习算法,是纯数据驱动的随机过程,适用于高维、小样本、非线性等回归问题的求解<sup>[13]</sup>。因其在样本量较少时仍能具有较强的泛化能力,MC-PILCO算法得以在数据量较少的场景中得到发挥;高斯过程建立的不确定性模型能输出具有概率意义的预测结果,方便对不确定性做出更全面的评估。因此本文选择了其作为参数自动整定框架的核心算法。

### 2.2 跟踪控制器控制指令和控制参数

本文使用当前应用较为广泛的线性二次型

(linear quadratic regulator, LQR)控制器和比例积分 (proportional-integral, PI)控制器组合而成的 LQR-PI 控制器作为研究对象。搭建如图 4 红色实线框所

示的智能车横纵向控制系统, LQR-PI 横纵向控制器中 LQR 控制器进行横向控制, 纵向控制器则分为双 PI 纵向上层控制器和纵向下层控制器。

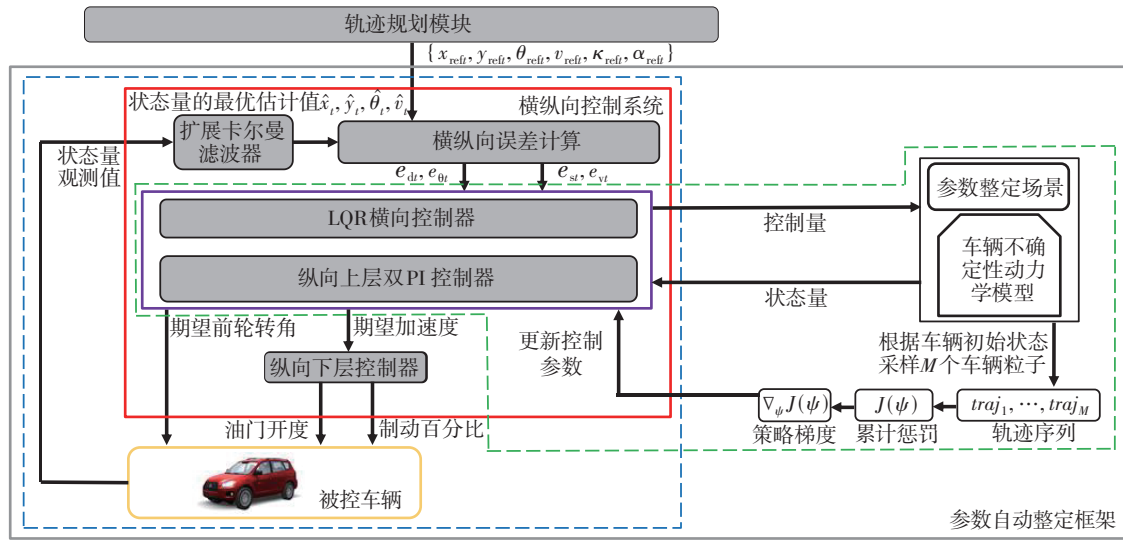


图 4 参数自动整定框架

从图 4 中可以看出,在  $t$  时刻车载传感器获取到车辆状态观测值后,使用扩展卡尔曼滤波消除噪声的干扰<sup>[14]</sup>得到当前时刻的最优估计值,结合由上层规划模块给出的目标轨迹点信息可以计算得到横向距离误差  $e_{dt}$ 、航向误差  $e_{\theta t}$ 、纵向距离误差  $e_{st}$  和纵向速度误差  $e_{vt}$ 。

为了建立 LQR 反馈调节控制,基于图 5 所示的考虑轮胎侧向力的 2 自由度动力学模型推导出车辆轨迹跟踪误差状态方程。 $e_{dt}$  和  $e_{\theta t}$  输入到 LQR 横向控制器之后,设置目标函数:

$$J = \sum_{k=0}^{\infty} (e_k^T Q e_k + u_{tSteerExp}^T R u_{tSteerExp}) \quad (2)$$

并通过迭代求解 LQR 的黎卡提方程得到反馈矩阵进而计算出该时刻期望的前轮转角  $u_{tSteer}$ 。式中:  $e_t = [e_{dt}, \dot{e}_{dt}, e_{\theta t}, \dot{e}_{\theta t}]^T$  为状态变量;  $u_{tSteerExp}$  为当前时刻的车辆前轮转角;  $Q$  和  $R$  分别为决定损失函数的矩阵。期望的前轮转角  $u_{tSteer}$  结合 Rajamani 等<sup>[15]</sup> 给出的前馈控制  $u_f$ , 通过式 (3) 可以得出约束后的期望转角  $u_{tSteerExpRst}$ 。

$$u_{tSteerExpRst} = u_{SteerMax} \cdot \tanh\left(\frac{u_{tSteer} + u_f}{u_{SteerMax}}\right) \quad (3)$$

式中  $u_{SteerMax}$  为前轮转角最大约束值。

对应误差状态变量  $e_t = [e_{dt}, \dot{e}_{dt}, e_{\theta t}, \dot{e}_{\theta t}]^T$ , 设置  $Q$  矩阵, 包含  $Q(1,1)$ ,  $Q(2,2)$ ,  $Q(3,3)$ ,  $Q(4,4)$  4 个参数, 分别代表横向距离误差、横向距离误差变化

率、航向误差、航向误差变化率的惩罚, 数值越大跟踪误差越小。 $R$  矩阵则包含  $R(1,1)$  一个参数, 代表了控制器对控制量的约束大小, 数值越小则允许的控制量越大。LQR 横向控制器的参数由  $Q$  和  $R$  两个矩阵的参数组成。

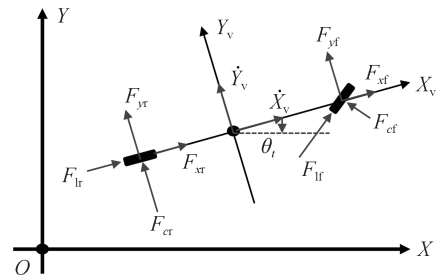


图 5 考虑轮胎侧向力的 2 自由度车辆动力学模型

为了使优化策略过程中梯度下降后两个矩阵仍然保持正定性, 对矩阵  $Q$  和  $R$  作了如下的参数化处理:

$$Q = \begin{bmatrix} Q(1,1) & 0 & 0 & 0 \\ 0 & Q(2,2) & 0 & 0 \\ 0 & 0 & Q(3,3) & 0 \\ 0 & 0 & 0 & Q(4,4) \end{bmatrix} = \begin{bmatrix} e^{\tilde{q}(1,1)} & 0 & 0 & 0 \\ 0 & e^{\tilde{q}(2,2)} & 0 & 0 \\ 0 & 0 & e^{\tilde{q}(3,3)} & 0 \\ 0 & 0 & 0 & e^{\tilde{q}(4,4)} \end{bmatrix} \quad (4)$$

$$R = [R(1, 1)] = [e^{\tau(1,1)}] \quad (5)$$

纵向上层控制器是位置速度双PI控制器,通过  $e_{st}$  和  $e_{vt}$  使用式(6)计算得到期望加速度  $u_{tAccel}$ 。

$$u_{tAccel} = a_{tRef} + K_{pv}(e_{vt} + K_{ps}e_{st}) + K_{iv} \int (e_{vt} + K_{ps}e_{st}) dt \quad (6)$$

式中:  $a_{tRef}$  为参考加速度;  $K_{ps}$ 、 $K_{pv}$  和  $K_{iv}$  分别为纵向距离的比例系数、纵向速度的积分系数和比例系数,这3个系数将作为双PI上层控制器的3个控制参数。然后同样通过与式(3)类似的方式得到经过约束后的期望加速度  $u_{tAccelExpRst}$ 。做实车实验分别采集驱动力矩和制动百分比与车辆加速度的关系,使用线性插值拟合出车辆加速动力学模型和减速动力学模型,共同组成纵向下层控制器。使用类似“查表”的方式,由经过约束后的期望加速度就能进一步获得驱动控制指令  $u_{tThrottle}$  和制动控制指令  $u_{tBreak}$ ,进而下发执行器用于实际控车。驱动控制指令为车辆驱动力矩,制动控制指令为制动百分比。

进行参数初始化时,一般选择控制较为保守的参数。对于本文而言,横向控制参数初始化值分别选择了  $e^0, e^{-2}, e^0, e^{-2}, e^6$ , 其中  $R(1, 1)$  设置为  $e^6$  是为了尽可能地保证控制稳定性,避免振荡。纵向参数分别初始化为 1.0, -1.0, 1.0, 绝对值均为 1。

### 2.3 车辆不确定性动力学模型与状态预测

本文选择高斯过程进行车辆不确定性动力学模型的搭建。高斯过程通过训练集建立先验分布,然后在测试集上得到后验分布,用概率的形式给出估计值,具体推导过程见文献[16]。

核函数的选择是决定GP模型平滑性和周期性的关键。考虑到本文以不确定性作为重要数据来源,因此需要核函数具备较强的数据抗干扰能力,选择平方指数核函数作为GP核函数,形式如下:

$$k_{SE}(x_i, x_j) = \lambda^2 \exp\left(-\|x_i - x_j\|_{\Lambda^{-1}}^2\right) \quad (7)$$

式中:  $\lambda$  为水平因子;  $\Lambda$  为特征长度  $l$  的对角矩阵,  $\Lambda = \text{diag}([l_1^2, \dots, l_n^2])$ 。核函数的超参数包括高斯白噪声标准差  $\sigma_n$ 、水平因子  $\lambda$ 、特征长度  $l$ 。令  $\theta = [\sigma_n, \lambda, l]$ , 超参数可以通过最大化边缘似然函数  $\log p(y|X, \theta)$  训练获得。

实际建模时,使用待整定控制器进行实车实验采集2.2节中提及到的控制量数据与状态量数据,考虑包括数据传输时延和机械响应时延在内的控制时延<sup>[17]</sup>后得到前轮转角响应值  $u_{tSteerExp}$  和加速度响应值  $u_{tAccelExp}$ , 具体公式如下:

$$u_{tSteerExp} = \frac{\tau_{u_{steer}} - \Delta t}{\tau_{u_{steer}}} u_{t-1SteerExp} + \frac{\Delta t}{\tau_{u_{steer}}} u_{tSteerExpRst} - len_{steer} \quad (8)$$

$$u_{tAccelExp} = \frac{\tau_{u_{throttle}} - \Delta t}{\tau_{u_{throttle}}} u_{t-1AccelExp} + \frac{\Delta t}{\tau_{u_{throttle}}} u_{tAccelExpRst} - len_{throttle} \quad (9)$$

式中参数  $\tau_{u_{steer}}$ 、 $\tau_{u_{throttle}}$ 、 $len_{steer}$ 、 $len_{throttle}$  须通过阶跃响应测试确定。之后根据图6搭建车辆不确定性动力学模型,图中的  $\xi$  为车辆转向盘转角与前轮转角之间的比例关系。4个独立的高斯过程分别输出车辆位置变化量  $\Delta x'_i$ 、 $\Delta y'_i$ 、航向变化量  $\Delta \theta'_i$ 、速度变化量  $\Delta v_i$  的均值和协方差。其中各变化量的计算根据假设的  $t+1$  时刻的最优状态估计值  $\hat{x}_{t+1}$ 、 $\hat{y}_{t+1}$ 、 $\hat{\theta}_{t+1}$ 、 $\hat{v}_{t+1}$  和图7所示的几何关系得到。

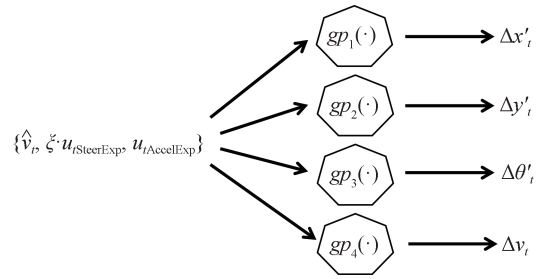


图6 基于高斯过程的车辆不确定性动力学模型

车辆状态预测分为单步预测和迭代预测。在进行车辆状态单步预测时,对高斯过程模型预测出来的变化量进行随机采样得到车辆局部坐标系下的车辆位置和航向变化量:

$$\begin{cases} \Delta \tilde{x}'_i \sim p(\Delta x'_i | \mu_{\Delta x'_i}, \Sigma_{\Delta x'_i}) \\ \Delta \tilde{y}'_i \sim p(\Delta y'_i | \mu_{\Delta y'_i}, \Sigma_{\Delta y'_i}) \\ \Delta \tilde{\theta}'_i \sim p(\Delta \theta'_i | \mu_{\Delta \theta'_i}, \Sigma_{\Delta \theta'_i}) \\ \Delta \tilde{v}_i \sim p(\Delta v_i | \mu_{\Delta v_i}, \Sigma_{\Delta v_i}) \end{cases} \quad (10)$$

转换为全局坐标系可得:

$$\begin{cases} \Delta x_i = \Delta \tilde{x}'_i \cos(\hat{\theta}_i) - \Delta \tilde{y}'_i \sin(\hat{\theta}_i) \\ \Delta y_i = \Delta \tilde{x}'_i \sin(\hat{\theta}_i) + \Delta \tilde{y}'_i \cos(\hat{\theta}_i) \\ \Delta \theta_i = \Delta \tilde{\theta}_i \end{cases} \quad (11)$$

结合  $t$  时刻的状态量最优估计和噪声,得到包含高斯噪声的状态量预测:

$$s_{t+1}^c = [x_{t+1}^c, y_{t+1}^c, \theta_{t+1}^c, v_{t+1}^c]^T \quad (12)$$

经过卡尔曼滤波可预测  $t+1$  时刻的车辆状态:

$$\hat{s}_{t+1} = [\hat{x}_{t+1}, \hat{y}_{t+1}, \hat{\theta}_{t+1}, \hat{v}_{t+1}]^T \quad (13)$$

在进行车辆状态的长期迭代预测时,采用图3所示基于粒子的方法从初始车辆状态分布中随机采样出多个车辆状态,每个状态对应一个车辆粒子。每个粒子基于相同的车辆不确定性模型在一定的控制周期内进行多次单步预测,以此得到车辆的状态和约束控制指令组成的轨迹序列。

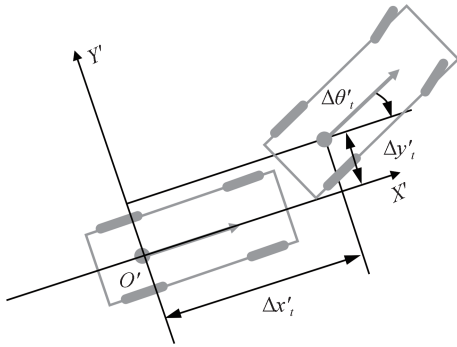


图7 局部坐标系下的车辆状态变化量

## 2.4 跟踪控制器参数自动整定框架

参数自动整定框架如图4所示。选择如图8所示的场景进行跟踪控制以采集实车状态量数据,完成MC-PILCO算法在进行控制参数迭代优化前须先搭建车辆不确定性动力学模型的工作。为了评估横纵向控制器的跟踪能力,选择具有初始横向偏差 $e_{d0}$ 的直线加速场景。该轨迹还存在加速和匀速阶段,可用于评估纵向控制器的跟踪性能。

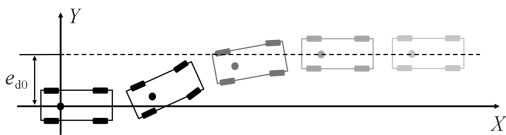


图8 跟踪控制器参数整定场景

结合2.3节迭代预测得到的 $M$ 个车辆粒子在 $N$ 时刻的轨迹序列以及2.2节得到的横纵向跟踪误差,设计累计折扣惩罚函数:

$$J(\psi) = \sum_{t=0}^N \left( \frac{1}{M} \sum_{m=1}^M c(e_t^{(m)}, \mathbf{u}_{t\text{Cmd}}^{(m)}) \right) \quad (14)$$

式中 $c(e_t^{(m)}, \mathbf{u}_{t\text{Cmd}}^{(m)}) = 1 - \exp(-\text{dist})$ 为第 $m$ 个车辆粒子在 $t$ 时刻下的惩罚值, $e_t^{(m)}$ 为跟踪误差, $\mathbf{u}_{t\text{Cmd}}^{(m)}$ 为约束控制指令, $\text{dist}$ 表示 $e_t^{(m)}$ 和 $\mathbf{u}_{t\text{Cmd}}^{(m)}$ 的惩罚和,可用式(15)计算得到。

$$\text{dist} = (C_t e_t^{(m)} - e_t^{\text{des}})^T L_{e_t} (C_t e_t^{(m)} - e_t^{\text{des}}) + (\mathbf{u}_{t\text{Cmd}}^{(m)} - \mathbf{u}_{t\text{Cmd}}^{\text{des}})^T L_{\mathbf{u}_{t\text{Cmd}}} (\mathbf{u}_{t\text{Cmd}}^{(m)} - \mathbf{u}_{t\text{Cmd}}^{\text{des}}) \quad (15)$$

式中: $C_t$ 为折扣因子; $e_t^{\text{des}}$ 为期望误差量; $\mathbf{u}_{t\text{Cmd}}^{\text{des}}$ 为期望

控制量; $L_{e_t} = \text{diag}([l_{e_d}, l_{e_\theta}, l_{e_x}, l_{e_y}])^2$ 为累计惩罚对 $e_{dt}$ 、 $e_{\theta t}$ 、 $e_{xt}$ 、 $e_{yt}$ 的惩罚系数对角矩阵; $L_{\mathbf{u}_{t\text{Cmd}}} = \text{diag}([l_{u_{\text{steer}}}, l_{u_{\text{brake}}}]^2)$ 为约束后的期望前轮转角和约束后的期望加速度的惩罚系数对角矩阵。

参数自动整定框架包含内外两层循环,其中内层循环如图4绿色虚线框所示,负责进行模型与控制器的交互,而图4蓝色虚线框所示的外层循环则是控制器与真实车辆系统进行的实车交互,用于对模型进行更新。进行每一次整定时,通过外层循环进行一次实车交互得到目标工况下的行驶数据,使用2.3节所示的方法搭建真实车辆系统的不确定性动力学模型。之后基于得到的模型进行多次内层循环。每一次循环流程如下:基于图8所示的训练场景,根据车辆初始状态分布采样出多个车辆粒子,由每个车辆粒子基于同一个动力学模型进行长期状态预测得到的轨迹计算累计惩罚 $J(\psi)$ ,然后反向传播计算累计惩罚对横纵向控制参数的梯度 $\nabla_{\psi} J(\psi)$ ,最后基于设置好的学习率进行梯度下降并更新控制参数。整定得到的参数可以通过一次外层循环予以验证。每一次整定结束后视情况而定对模型进行更新。

## 3 训练集的构建和调节器的训练

这一章节将介绍基于自动整定框架的训练集构建方法并给出参数调节器训练结果。

### 3.1 训练集构建方法

由1.2节调节器训练需求分析可知,由于典型工况较多,若对于每个工况都先进行一次实车交互和建模然后通过内层循环进行参数的多次优化,势必导致实车交互次数过多。为了减少实车交互次数,提高训练集生成效率,本文在2.4节自动整定框架的基础上提出了一种改进的训练集构建方法,具体流程如图9所示。

由2.1节可知,由于MC-PILCO使用高斯过程进行模型学习,高斯过程建模对小样本适应能力强,模型在预测未见过的数据时预测结果仍具有参考价值,因此每个车速工况分组内所有不同曲率工况均可基于该车速下跟踪直线场景采集到的数据训练出来的动力学模型进行参数整定。

实际整定时,将训练集中的所有典型工况按车速工况进行分组,每个分组对0曲率的工况进行一次实车交互和不确定性动力学模型搭建,每个车速

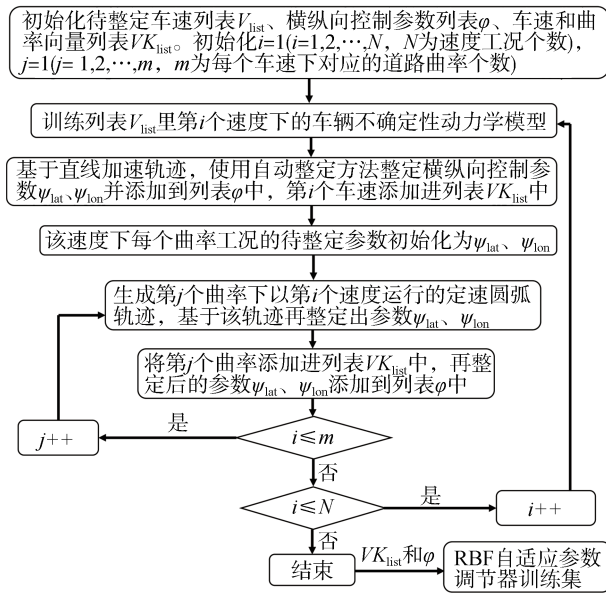


图9 车速与曲率自适应调节器训练集构建流程图

工况对应的模型训练集数据量为200组,单个模型输入维度为3。运用自动整定框架内层循环进行多次模型交互整定出参数。这些初次整定得到的参数

将作为该分组中剩余曲率工况待整定参数的初始值。根据剩余曲率工况的需要更新目标轨迹并基于0曲率工况时建立的模型进行分别再次整定,即可得到该分组所有工况的控制参数。也就是说如果训练集包含N种典型车速工况,每种车速工况下又包含m种曲率工况,使用本文提出的方法构建调节器训练集时的实车交互复杂度从O(N×m)降低到了O(N)。

### 3.2 调节器的训练结果

在典型工况的选择上,由公路路线设计规范<sup>[18]</sup>对高速公路圆曲线曲率的限定可知,实际应用中车辆在高速工况下极少进行大曲率的运动,只有在低速工况下控制精度才会对曲率变化较为敏感,因此选择10、20、30 km/h作为典型速度工况并设置不同的曲率工况。将3.1节所述方法编写成Python脚本后,使用运算平台运行程序得到调节器的训练集如表1所示。运算平台处理器为Intel Core i5-13490F,显卡为Nvidia GeForce RTX 4070, 32 GB RAM。训练集构建完毕之后即可基于RBF神经网络进行调节器的训练。训练过程中epochs设置为20 000,学习率设置为0.001。

表1 面向不同车速与曲率的自适应参数调节器训练集

速度工况	曲率工况	LQR横向控制器参数					双PI控制器参数		
		Q(1,1)	Q(2,2)	Q(3,3)	Q(4,4)	R(1,1)	K <sub>ps</sub>	K <sub>pv</sub>	K <sub>iv</sub>
10 km·h <sup>-1</sup>	0	e <sup>0.72</sup>	e <sup>-0.61</sup>	e <sup>3.31</sup>	e <sup>0.95</sup>	e <sup>2.30</sup>	1.70	-4.50	-0.03
	1/25	e <sup>1.85</sup>	e <sup>-1.86</sup>	e <sup>3.77</sup>	e <sup>-0.08</sup>	e <sup>0.83</sup>	1.70	-4.50	-0.03
	1/15	e <sup>2.07</sup>	e <sup>-2.15</sup>	e <sup>3.97</sup>	e <sup>-0.21</sup>	e <sup>0.61</sup>	1.70	-4.50	-0.03
20 km·h <sup>-1</sup>	0	e <sup>0.89</sup>	e <sup>-0.32</sup>	e <sup>2.72</sup>	e <sup>-3.90</sup>	e <sup>3.67</sup>	0.37	-4.31	-7.53
	1/65	e <sup>2.08</sup>	e <sup>-0.08</sup>	e <sup>3.77</sup>	e <sup>-5.16</sup>	e <sup>2.09</sup>	0.37	-4.31	-7.53
	1/55	e <sup>1.84</sup>	e <sup>-0.35</sup>	e <sup>3.26</sup>	e <sup>-4.77</sup>	e <sup>2.27</sup>	0.37	-4.31	-7.53
30 km·h <sup>-1</sup>	0	e <sup>1.22</sup>	e <sup>-0.50</sup>	e <sup>2.45</sup>	e <sup>-4.39</sup>	e <sup>4.49</sup>	1.74	-4.37	-1.35
	1/65	e <sup>2.11</sup>	e <sup>-0.16</sup>	e <sup>3.10</sup>	e <sup>-2.24</sup>	e <sup>3.14</sup>	1.74	-4.37	-1.35
	1/55	e <sup>2.11</sup>	e <sup>-0.41</sup>	e <sup>2.88</sup>	e <sup>-1.99</sup>	e <sup>3.19</sup>	1.74	-4.37	-1.35

为了实验完整性的需要,补充了车速自适应参数调节器的训练,表1中每个车速工况下的0曲率工况参数即为其训练集。图10是两种调节器训练过

程中损失函数变化情况。图11(a)和图11(b)分别是车速自适应调节器的训练结果。车速与曲率自适应调节器训练结果将在下文结合实车实验场景给出。

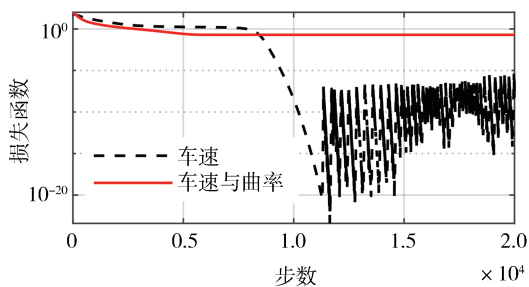


图10 自适应参数调节器训练过程中的损失函数变化

## 4 实车实验验证

### 4.1 实车实验平台

本次实车实验采用的是经过线控改装的广汽传祺GE3自动驾驶车辆,如图12所示。各车载设备的具体型号如表2所示。

### 4.2 验证场景设计

为了检验使用本文提出的训练集构建方法得到

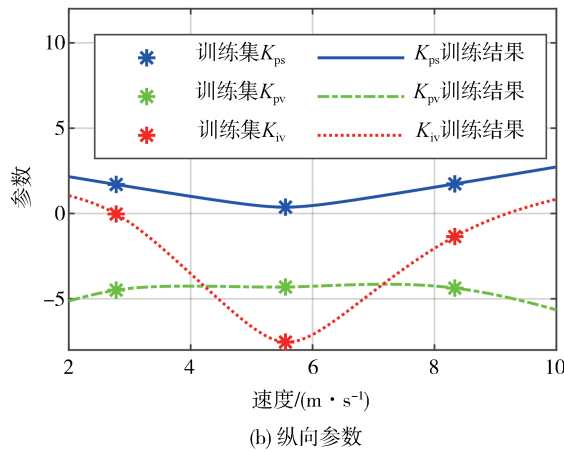
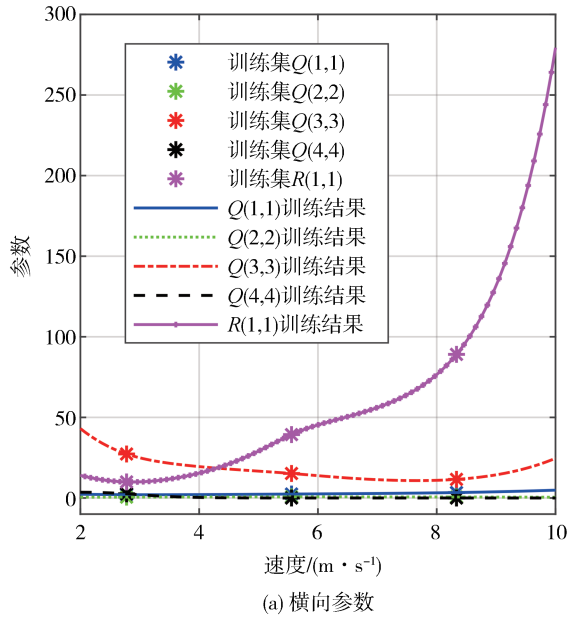


图 11 车速自适应参数调节器训练结果



图 12 实车实验车辆

的参数自适应循迹控制器的跟踪效果,于广东省广州市天河区内一环湖道路开展实车实验,该环湖场景的卫星影像如图 13 所示。设置固定参数 LQR-PI 控制器作为对照组,固定的参数具体见表 3。采用人工驾驶的方式录得目标跟踪轨迹如图 14 所示。图 15 展示了目标跟踪轨迹的车速与曲率变化,黑色

表 2 实车实验平台车辆参数与车载设备信息

车辆参数/车载设备	参数大小/设备型号
车长	4 337 mm
车宽	1 825 mm
车高	1 658 mm
车轴距	2 560 mm
车载激光雷达	RoboSense RS-LIDAR-128
车载定位设备	广州导远 INS570D
车载工控机	Ubuntu 16.04
	Intel Core i9-10900F
	32 GB RAM
	Nvidia GeForce RTX 2060

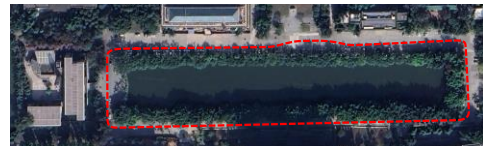


图 13 用于自适应控制的环湖道路俯视图

表 3 固定参数 LQR-PI 控制器的控制参数

LQR 横向控制器参数					双 PI 控制器参数		
$Q(1, 1)$	$Q(2, 2)$	$Q(3, 3)$	$Q(4, 4)$	$R(1, 1)$	$K_{ps}$	$K_{pv}$	$K_{iv}$
$e^{1.24}$	$e^{-0.48}$	$e^{2.53}$	$e^{-4.88}$	$e^{4.49}$	1.45	-5.40	-0.08

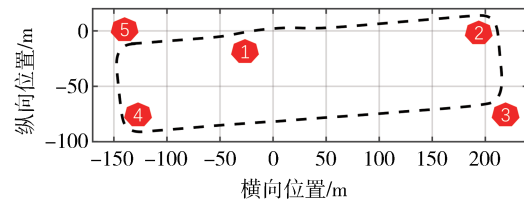


图 14 人工录制的目标跟踪轨迹

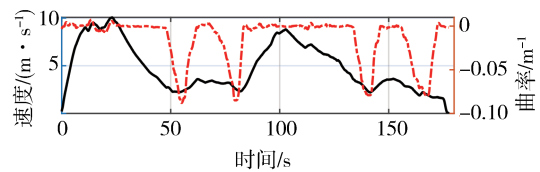


图 15 目标跟踪轨迹的车速与曲率变化

实线为车速变化曲线,红色虚线为曲率变化曲线。

从图 14 可以看出,目标跟踪轨迹并非标准的矩形,在位置 1 处设置有两次速度在 35 km/h 附近的连续变道,用于测试换道轨迹下自适应参数调节器的表现。位置 2、3、4、5 为 90°转向轨迹,用于测试控制器的低速大曲率跟踪性能。

训练后调节器在实车场景不同工况下控制参数随时间的变化如图 16(a)和图 16(b)所示。

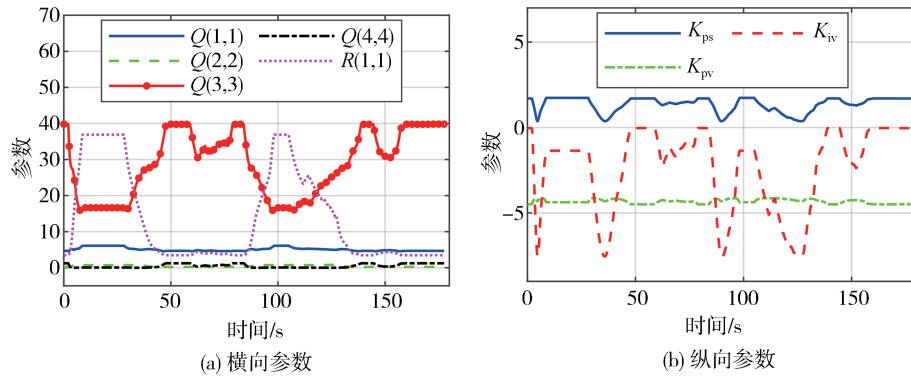


图16 车速与曲率调节器训练结果

### 4.3 跟踪控制结果与对比分析

对比固定参数的LQR-PI控制器和自适应参数调节LQR-PI横纵向控制器对同一条目标轨迹跟踪

结果,不同控制器的跟踪误差统计如表4所示,数据均通过四舍五入保留两位小数。图17为控制器的跟踪误差对比图,图18为跟踪过程中控制量的变化

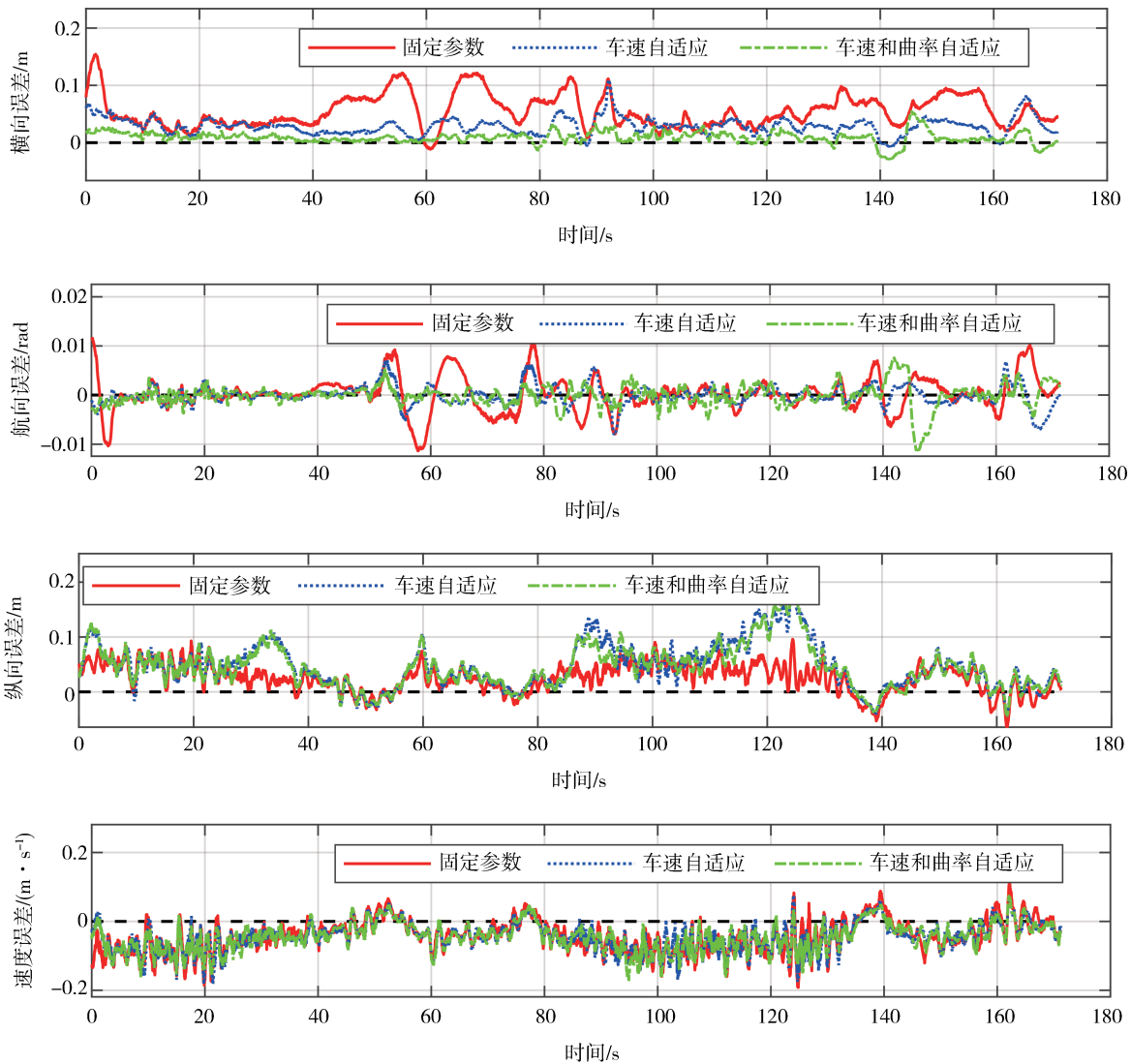


图17 固定参数LQR-PI控制器与LQR-PI横纵向参数自适应控制器跟踪误差对比

情况。由表4可以看出,加入车速与曲率自适应功能后,循迹控制器的最大绝对横向距离误差由0.15下降至0.05 m,绝对横向距离误差均值由0.06下降至0.01 m,横向跟踪精度得到一定程度的提升。绝对纵向距离误差则有所增大。

结合图17和表4可知,在总共约170 s的环湖轨迹跟踪实验中,包含车速自适应、车速与曲率自适应参数调节器的LQR-PI横纵向控制器的横向跟踪精

度、航向跟踪精度上均比固定参数控制器更好,而纵向距离误差上略差于固定参数的控制器。在纵向速度误差上3个控制器则表现相近。实车实验表明,基于本文提出的训练集构建方法,参数自适应调节器可以在较小的工作量和数据量下完成训练。训练后的参数自适应调节器根据不同工况实现控制器参数的动态调整,相比较参数固定的控制器而言有更好的横向轨迹跟踪效果。

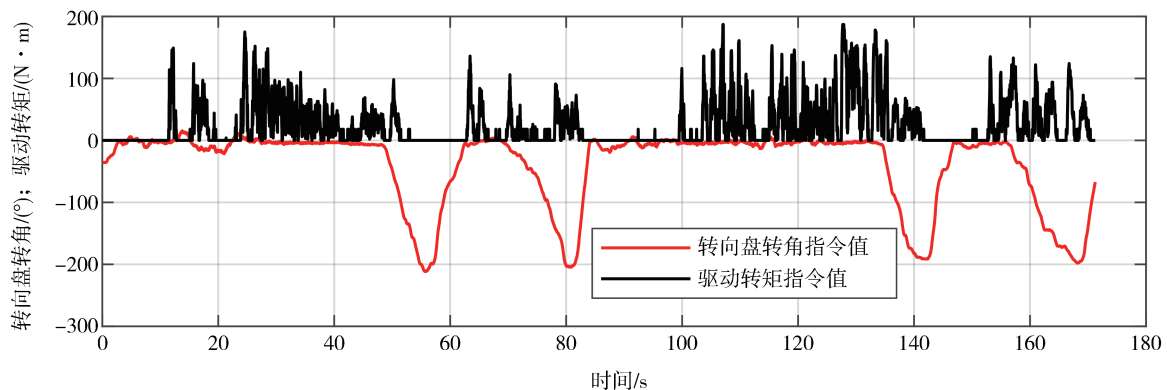


图18 轨迹跟踪过程中的转向盘转角指令值与驱动转矩指令值

表4 不同控制器的跟踪误差统计表

误差名称	单位	固定参数	车速自适应调节	车速与曲率自适应调节
$ e_d _{\max}$	m	0.15	0.11	0.05
$ e_\theta _{\max}$	rad	0.01	0.01	0.01
$ e_s _{\max}$	m	0.10	0.22	0.19
$ e_v _{\max}$	$\text{m}\cdot\text{s}^{-1}$	0.19	0.18	0.17
$\text{mean}( e_d )$	m	0.06	0.03	0.01
$\text{mean}( e_\theta )$	rad	0.00	0.00	0.00
$\text{mean}( e_s )$	m	0.03	0.05	0.05
$\text{mean}( e_v )$	$\text{m}\cdot\text{s}^{-1}$	0.05	0.05	0.05

## 5 结论

为了应对控制器变工况实际应用时控制参数调整的需要,基于RBF神经网络提出了应用于循迹跟踪控制器中的车速与曲率自适应参数调节器。基于MC-PILCO算法,提出对训练集中包含的所有典型工况根据速度进行分组并只对每个分组中的0曲率工况进行实车交互与建模。每个分组中剩余工况的参数基于该分组中0曲率工况对应的模型进行整定,大大减少因曲率工况的加入导致的车速与曲率

组合工况较多,从而使得实车交互次数过多的问题,减少了参数整定的工作量。

设计实车环湖跟踪控制实验。实验结果表明加入车速与曲率自适应功能后,循迹控制器的最大绝对横向距离误差由0.15下降至0.05 m,绝对横向距离误差均值由0.06下降至0.01 m。验证了控制器在加入参数自适应调节器后得到了更好的横向控制精度。

车辆在行驶过程中,受载质量和轮胎气压等因素变化的影响,整定后控制参数的跟踪控制精度有所下降。在未来可考虑对车辆不确定动力学模型进行实时更新以实现控制参数的在线自整定与调整。

## 参考文献

- [1] 高建平,余佳衡,孟焱,等. 电动车辆控制参数自动优化标定系统的研究及验证[J]. 中国机械工程, 2022, 33(1): 118-125.  
GAO J P, YU J H, MENG Y, et al. Research and verification of control parameter automatic optimization calibration simulation system for electric vehicles [J]. China Mechanical Engineering, 2022, 33(1): 118-125.
- [2] KOTHARI M L, MADNANI S, SEGAL R. Orthogonal least squares learning algorithm based radial basis function (RBF) network adaptive power system stabilizer [J]. IEEE, 1997: 542-547.
- [3] 杨建伟,刘海波,孙守光. 基于RBF网络自适应控制的车辆道

- 路友好性与平顺性研究[J]. 汽车工程, 2010, 32(12): 1077-1082.
- YANG J W, LIU H B, SUN S G. A study on the road-friendliness and ride comfort of vehicle based on RBF network adaptive control[J]. *Automotive Engineering*, 2010, 32(12): 1077-1082.
- [4] 陈刚, 张为公. 基于模糊自适应PID的汽车驾驶机器人的车速控制[J]. 汽车工程, 2012, 34(6): 511-516.
- CHEN G, ZHANG W G. Speed control of vehicle robot driver based on adaptive fuzzy PID control[J]. *Automotive Engineering*, 2012, 34(6): 511-516.
- [5] ZHANG M W, TIAN F J, HE Y Q, et al. Adaptive path tracking for unmanned ground vehicle[J]. *IEEE International Conference on Unmanned Systems (ICUS)*, 2017: 139-145.
- [6] MOOREN N, WITVOET G, OOMEN T. Feedforward motion control: from batch-to-batch learning to online parameter estimation[J]. *American Automatic Control Council*, 2019: 947-952.
- [7] AMADIO F, DALLA-LIBERA A, ANTONELLO R, et al. Model-based policy search using monte carlo gradient estimation with real systems application[J]. *IEEE Transactions on Robotics*, 2022.
- [8] 梁军, 朱方博, 蔡英凤, 等. 面向复杂曲率变化的智能车路径跟踪控制[J]. 汽车工程, 2021, 43(12): 1771-1779.
- LIANG J, ZHU F B, CAI Y F, et al. Intelligent vehicle path tracking control based on complex curvature variation[J]. *Automotive Engineering*, 2021, 43(12): 1771-1779.
- [9] 赵健, 杜金朋, 朱冰, 等. 基于自适应动态滑模控制的智能汽车纵向巡航控制[J]. 汽车工程, 2022, 44(1): 8-16.
- ZHAO J, DU J P, ZHU B, et al. Longitudinal cruise control of intelligent vehicles based on adaptive dynamic silding mode control[J]. *Automotive Engineering*, 2022, 44(1): 8-16.
- [10] DEISENROTH M, RASMUSSEN C E. PILCO: a model-based and data-efficient approach to policy search[C]. *Proceedings of the 28th International Conference on Machine Learning*, 2011: 465-472.
- [11] 梁俊朗, 高健. 基于PILCO架构的自适应内模控制器优化[J]. 组合机床与自动化加工技术, 2023(9): 59-62.
- LIANG J L, GAO J. An adaptive internal model optimization based on PILCO architecture[J]. *Modular Machine Tool & Automatic Manufacturing Technique*, 2023(9): 59-62.
- [12] SRIVASTAVA N, HINTON G, KRIZHEVSKY A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. *Journal of Machine Learning Research*, 2014, 15(1): 1929-1958.
- [13] 王萍, 彭香园, 程泽. 基于DTV-IGPR模型的锂离子电池SOH估计方法[J]. 汽车工程, 2021, 43(11): 1710-1719.
- WANG P, PENG X Y, CHENG Z. SOH estimation method for lithium-ion batteries based on DTV-IGPR model[J]. *Automotive Engineering*, 2021, 43(11): 1710-1719.
- [14] 王艳辉. 复杂非线性动态系统的滤波与平滑方法研究[D]. 成都: 电子科技大学, 2019.
- WANG Y H. Research on filtering and smoothing methods for complex nonlinear dynamic systems[D]. Chengdou: University of Electronic Science and Technology of China, 2019.
- [15] RAJAMANI R. Vehicle dynamics and control[M]. Springer Science & Business Media, 2011.
- [16] 王耀健, 顾洁, 温洪林, 等. 基于在线高斯过程回归的短期风电功率概率预测[J]. 电力系统自动化, 2024.
- WANG Y J, GU J, WEN H L, et al. Short-term wind power probability prediction based on online gaussian process regression[J]. *Automation of Electric Power Systems*, 2024.
- [17] XU S B, PENG H, TANG Y F. Preview path tracking control with delay compensation for autonomous vehicles[J]. *IEEE transactions on Intelligent Transportation Systems*, 2021, 22(5): 2979-2989.
- [18] 公路路线设计规范: JTG D20—2017[S].2017
- Design specification for highway alignment: JTG D20—2017[S].2017