Computer Science and Application 计算机科学与应用, 2022, 12(2), 486-497
Published Online February 2022 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2022.122049

Hans汉斯

基于经典进化策略的移动小车模型预测控制

陈肇江[1]，程良伦[2]

[1]广东工业大学自动化学院，广东 广州
[2]广东工业大学计算机学院，广东 广州

收稿日期：2022年1月22日；录用日期：2022年2月18日；发布日期：2022年2月25日

摘 要

模型预测控制(model predictive control, MPC)已经被证明了在无人驾驶、移动机器人导航、机械臂路径规划、过程控制等诸多领域都拥有着出色的控制性能，然而传统的MPC控制不具有进化能力，在复杂的环境下系统往往很难得到最优的控制效果。因此，提出了一种结合经典进化策略(canonical evolution strategies, Canonical ES)的模型预测控制算法，并用于双轮差速移动小车的避障运动规划。在移动小车目标逼近运动规划的基础上，通过训练策略网络可以使得处于未知环境的移动小车快速有效地学习避障策略，并且对于局部最优陷阱拥有更优异的处理方法。仿真实验结果证明了该方法的可行性与高效性。

关键词

MPC，Canonical ES，移动小车，避障规划，运动规划

Model Predictive Control for Mobile Cart Based on Canonical Evolution Strategies

Zhaojiang Chen[1], Lianglun Cheng[2]

[1]School of Automation, Guangdong University of Technology, Guangzhou Guangdong
[2]School of Computer Science and Technology, Guangdong University of Technology, Guangzhou Guangdong

Received: Jan. 22nd, 2022; accepted: Feb. 18th, 2022; published: Feb. 25th, 2022

Abstract

Model predictive control (MPC) has been identified to have excellent control performance in many fields such as unmanned driving, mobile robots navigation, path planning of manipulators and process control, etc. However, traditional MPC does have no ability to evolve, for which is often difficult for the system to get the best effects in complex environments. Therefore, the MPC algorithm combining canonical evolution strategies is proposed. The algorithm contributes to obstacle

文章引用: 陈肇江, 程良伦. 基于经典进化策略的移动小车模型预测控制[J]. 计算机科学与应用, 2022, 12(2): 486-497.
DOI: 10.12677/csa.2022.122049

avoidance motion planning of a two-wheel differential mobile cart. Based on the goal-approximation motion planning of the mobile cart, the mobile cart is able to learn the obstacle avoidance strategies effectively in unknown environment by training the policy network. Moreover, mobile cart will move better when facing local optimal traps. The feasibility and efficiency of the method are verified by experimental results.

Keywords

MPC, Canonical ES, Mobile Cart, Obstacle Avoidance Planning, Motion Planning

1. 引言

模型预测控制(MPC)的强鲁棒性已为大量的系统仿真和工业实践过程所验证。相比于 PID 等经典控制算法，MPC 可以轻松应对多输入多输出(Multi-Input and Multi-Output, MIMO)系统的挑战。然而随着工业生产的动态过程日益复杂化，对于性能非常依赖预测模型精度的 MPC 而言已无法满足日常工业生产需求。搭建更加复杂的物理模型，或者使用非线性的优化求解器是提高控制性能的常用做法，而这样的做法往往会面临过高的计算负荷和存储需求的问题，从而使得研究只能止步于计算机仿真，而无法大规模地投入生产实践。

为了改进 MPC 的性能，2012 年，基于学习的模型预测控制(Learning-Based MPC, LBMPC)理论被提出，RL 与 MPC 两种算法的结合成为可能。Zhong 等人利用传统 MPC 运行所生成的大量数据全局逼近最优价值函数，并通过求解线性可解马尔科夫决策过程(LMDP)获得 MPC 控制策略[1]，然而该方法比较依赖函数逼近的精确度，在大部分时间所表现的性能不佳；Sun 等人提出一种基于策略梯度(PG)的模型预测控制算法用于智能车辆的运动控制[2]，该方法是确定性策略梯度法(DDPG)的一个改进版，Rokonuzzaman 等人则是提出一种 LBMPC 方法[3]，该方法从人类的驾驶经验中学习并逼近终端代价函数，从而优化无人驾驶车辆的行为，然而这些方法都需要一个参考轨迹，并不适用于本文所描述的移动小车对于任意目标点的逼近规划任务，而同样的问题也存在于文献[4]与[5]所介绍的迭代学习控制(Iterative Learning Control, ILC)；Rosolia 等人提出了一种不需要任何参考轨迹的 ILC 方法[6]，该方法给 MPC 的代价函数添加了一个价值函数，同样是 LBMPC 的一个变种，然而该方法需要一个控制器用于将系统驱动到任意过去已访问过的状态中，这对于本文的移动小车避障任务而言显然是不现实的。

应当指出，当今主流的 LBMPC 算法所使用的 RL 方法大多为基于 PG 的方法，而基于 PG 的方法可能会存在一些问题，例如采样效率低下、ICS [7] (Internal Covariate Shift)、梯度爆炸、梯度消失以及低精度梯度计算问题等等。针对上述问题，Salimans 等人[8]和 Chrabaszcz 等人[9]分别使用进化策略(ES)代替 RL 算法，实验结果显示 ES 算法在大部分场景中(如 Atari 游戏、MuJoCo 人形机器人行走等)表现出比基于 PG 的 RL 算法更为优异的性能；而 Chrabaszcz 等人还指出，即使是最为经典(Canonical)的 ES 算法也可以达到媲美 Chrabaszcz 等人所达到的效果，这证明了 ES 算法在解决 RL 问题时相较于传统 RL 算法所表现的优势。

基于 Chrabaszcz 等人所使用的 ES 算法，本文提出一种基于经典 ES 的 MPC 控制算法，并解决了双轮差速移动小车的避碰规划问题。一方面，将 ES 的进化性融入 MPC，提高了移动小车对于未知环境的

规划能力；另一方面将 MPC 的预测性融入 ES，规避 RL 算法因缺少先验知识而导致的采样效率底下的问题。在实验的过程中，我们发现本文所提出的算法与文献[10]所提出的事后规划 MPC 算法(HI-MPC)在一般场景下所表现的性能不分伯仲，但在易失败的极端障碍场景下，本文的算法则表现得更为出色。

2. 基于 MPC 的小车运动控制问题建模

2.1. 小车运动模型建立

双轮差速移动小车的运动状态完全取决于双轮的转速，因此，可以建立如图 1 所示的小车运动模型。在图 1 中，p_x 与 p_y 分别为小车在坐标轴 x 和坐标轴 y 方向上的位置量，θ 为小车的航向角，l 为小车的轮间距，r 为车轮半径，w_l 与 w_r 分别为小车左右轮的转速。由图 1 可知，在某 t 时刻，当小车的航向角为 $\theta(t)$，双轮转速分别为 $w_l(t)$ 与 $w_r(t)$ 时，小车在坐标轴方向的速度分量与分别为

$$\dot{p}_x(t)=\frac{r\cos\theta(t)}{2}\cdot w_l(t)+\frac{r\cos\theta(t)}{2}\cdot w_r(t)$$

$$\dot{p}_y(t)=\frac{r\sin\theta(t)}{2}\cdot w_l(t)+\frac{r\sin\theta(t)}{2}\cdot w_r(t)$$

航向角的变化率为

$$\dot{\theta}(t)=-\frac{r}{l}\cdot w_l(t)+\frac{r}{l}\cdot w_r(t)$$

此时，令状态量 $\boldsymbol{x}=\left[p_x,p_y,\theta\right]$，控制量 $\boldsymbol{u}=\left[w_l,w_r\right]$，并假设双轮与地面的接触运动均为纯滚动而无滑动，则可建立如下所示的小车运动控制模型：

$$\begin{bmatrix}\dot{p}_x\\ \dot{p}_y\\ \dot{\theta}\end{bmatrix}=\begin{bmatrix}\frac{r\cos\theta}{2} & \frac{r\cos\theta}{2}\\ \frac{r\sin\theta}{2} & \frac{r\sin\theta}{2}\\ -\frac{r}{l} & \frac{r}{l}\end{bmatrix}\cdot\begin{bmatrix}w_l\\ w_r\end{bmatrix} \tag{1}$$

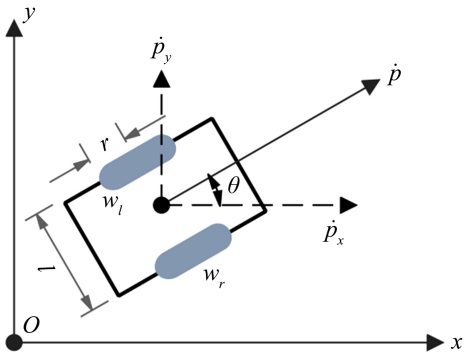

Figure 1. Kinematics model of vehicle
图 1. 小车运动模型

2.2. 简化为线性模型

由式(1)可知，该系统为非线性动态系统，无法使用状态矩阵与输入矩阵的形式进行表达，因而需要使用非线性控制方法进行求解，而非线性控制往往会面临过高的计算量，不利于实时控制。现有的对于

车辆的研究通常基于小航向角假设，由此可以将车辆的运动模型转化为线性模型进行求解[11]，然而在本文中小车的航向角是未知量，无法假设为小航向角，该方法自然不成立；另一种处理非线性系统的方法是利用泰勒展开将系统线性化，然而这种方法会导致模型精度降低，不利于控制。观察到在式(1)中，当 $w_l=-w_r$ 时，有

$$\dot{p}_x\big|_{w_l=-w_r}=\dot{p}_y\big|_{w_l=-w_r}=0$$

因此，航向角 θ 是一个独立可控的变量。基于这一点，将状态量 $\boldsymbol{x}$ 与控制量 $\boldsymbol{u}$ 重构为

$$\boldsymbol{x}=\left[p_x,p_y,\dot{p}_x,\dot{p}_y\right],\ \boldsymbol{u}=\left[\ddot{p}_x,\ddot{p}_y\right]$$

则新的小车运动控制模型可以构建为

$$\begin{bmatrix}\dot{p}_x\\ \dot{p}_y\\ \ddot{p}_x\\ \ddot{p}_y\end{bmatrix}=\begin{bmatrix}0&0&1&0\\0&0&0&1\\0&0&0&0\\0&0&0&0\end{bmatrix}\cdot\begin{bmatrix}p_x\\p_y\\\dot{p}_x\\\dot{p}_y\end{bmatrix}+\begin{bmatrix}0&0\\0&0\\1&0\\0&1\end{bmatrix}\cdot\begin{bmatrix}\ddot{p}_x\\\ddot{p}_y\end{bmatrix}\tag{2}$$

式(2)所示的状态方程具有可用状态矩阵与输入矩阵进行表达标准形式，为线性动态系统。此外，将系统重构为式(2)的形式的另一个优点是使用小车在坐标轴方向上的加速度作为控制量可以使得小车的运动更为平滑而不易发生抖动。

对于式(2)所表示的具有标准形式的时不变动态系统，可以使用前向欧拉法进行离散化，设采样时间为 T_s，则其离散状态方程为

$$\boldsymbol{x}(k+1)=\boldsymbol{A}\cdot\boldsymbol{x}(k)+\boldsymbol{B}\cdot\boldsymbol{u}(k)\tag{3}$$

其中

$$\boldsymbol{A}=\begin{bmatrix}1&0&T_s&0\\0&1&0&T_s\\0&0&1&0\\0&0&0&1\end{bmatrix},\ \boldsymbol{B}=\begin{bmatrix}0&0\\0&0\\T_s&0\\0&T_s\end{bmatrix}$$

2.3. 基于线性模型的 MPC 控制建模

对于任意第 k 个控制时刻，令预测步长与控制步长分别为 N_p 与 N_c，可以对离散状态方程(3)递推步长 N_p 可得预测时域内的离散状态方程组

$$\begin{aligned}&\boldsymbol{x}(k+1)=\boldsymbol{A}\cdot\boldsymbol{x}(k)+\boldsymbol{B}\cdot\boldsymbol{u}(k)\\&\boldsymbol{x}(k+2)=\boldsymbol{A}\cdot\boldsymbol{x}(k+1)+\boldsymbol{B}\cdot\boldsymbol{u}(k+1)=\boldsymbol{A}^2\cdot\boldsymbol{x}(k)+\boldsymbol{A}\cdot\boldsymbol{B}\cdot\boldsymbol{u}(k)+\boldsymbol{B}\cdot\boldsymbol{u}(k+1)\\&\qquad\vdots\\&\boldsymbol{x}(k+N_p)=\boldsymbol{A}^{N_p}\cdot\boldsymbol{x}(k)+\boldsymbol{A}^{N_p-1}\cdot\boldsymbol{B}\cdot\boldsymbol{u}(k)+\cdots+\boldsymbol{B}\cdot\boldsymbol{u}(k+N_c)\end{aligned}$$

上式中，$N_c=N_p-1$，以保证最高精度的求解效果。将上述预测方程组结合起来，并以矩阵的形式表达可得如下预测模型：

$$\boldsymbol{X}_p(k)=\boldsymbol{A}_p\cdot\boldsymbol{x}(k)+\boldsymbol{B}_p\cdot\boldsymbol{U}_p(k)\tag{4}$$

其中

$$\boldsymbol{X}_p(k)=\begin{bmatrix}\boldsymbol{x}(k+1)\\ \boldsymbol{x}(k+2)\\ \vdots\\ \boldsymbol{x}(k+N_p)\end{bmatrix},\ \boldsymbol{U}_p(k)=\begin{bmatrix}u(k)\\ u(k+1)\\ \vdots\\ u(k+N_c)\end{bmatrix}$$

$$\boldsymbol{A}_p=\begin{bmatrix}\boldsymbol{A}\\ \boldsymbol{A}^2\\ \vdots\\ \boldsymbol{A}^{N_p}\end{bmatrix},\ \boldsymbol{B}_p=\begin{bmatrix}\boldsymbol{B} & & & \\ \boldsymbol{AB} & \boldsymbol{B} & & \\ \vdots & \ddots & \ddots & \\ \boldsymbol{A}^{N_c}\boldsymbol{B} & \boldsymbol{A}^{N_c-1}\boldsymbol{B} & \cdots & \boldsymbol{B}\end{bmatrix}$$

对于预测模型(4)，为了优化预测状态量 $\boldsymbol{X}_p(k)$ 并获得最优控制序列 $\boldsymbol{U}_p(k)$，对于任意第 k 个控制时刻的状态量 $\boldsymbol{x}(k)$ 可以建立如下代价函数：

$$J\left(\boldsymbol{x}(k),\boldsymbol{U}_p(k)\right)=\frac{1}{2}\left\|\boldsymbol{X}_p(k)-\boldsymbol{X}_g\right\|_{\boldsymbol{Q}_g}^2+\frac{1}{2}\left\|\boldsymbol{U}_p(k)\right\|_{\boldsymbol{Q}_u}^2 \tag{5}$$

其中 $\boldsymbol{X}_g$ 为全局目标状态量，$\boldsymbol{Q}_g$、$\boldsymbol{Q}_u$ 分别为相应的权重矩阵。优化目标即是以最小的控制代价使当前状态渐进收敛到目标状态，并将最优控制序列 $\boldsymbol{U}_p^*(k)$ 的首个时刻的控制量作用于系统。

3. 进化策略算法

3.1. 进化策略与强化学习

人工智能(AI)的目标是开发具有在复杂的不确定环境中完成富含挑战性的特定任务能力的智能体(Agents)。近年来，解决这类问题较为流行的方法是基于马尔科夫决策过程(MDP-Based)的强化学习(RL)算法，例如 Q-Learning、策略梯度法(PG)等等。这类方法在游玩 Atari 游戏[12]、机械臂规划[13]、乒乓球机器人[14]等领域都取得了巨大的成功。

解决这类问题的另一种方法是采用以进化策略(ES)为代表的黑箱优化算法(Black-Box Optimization)。在 RL 中，ES 常常被用于直接策略搜索。但一直以来，研究者们认为与 Q-Learning、PG 等算法相比，ES 无法解决硬 RL 问题，或者在解决硬 RL 问题时效率较低。Salimans 等人[8]和 Chrabaszcz 等人[9]的结果证明了 ES 可以可靠地训练神经网络，并具备与 RL 抗衡甚至取代 RL 的优异性能。与 RL 相比，ES 具有以下优点：

1) ES 对于奖励函数的分布并不敏感，因为 ES 通常会采用一整个 episode 的奖励总和，同时 ES 也无需折扣因子；

2) ES 无需进行梯度反向传播，从而使得每一个 episode 的计算量大幅减少；

3) ES 在参数空间中进行扰动，致使搜索空间增大，在训练过程中更容易收敛到可行解。

3.2. 经典进化策略

本文所使用的经典进化策略算法隶属于(μ,λ)-ES 算法。(μ,λ)-ES 算法是 ES 算法大家族中最为杰出的一类，它会强迫上一代种群中的所有个体全部离开种群，只留下少数优异的个体存活至下一代，从而规避某些适应度较好但并非最好的个体因变异率选取不当而进化失败，但却强行存活到下一代甚至几代从而浪费种群位置的问题[15]。经典进化策略算法的步骤为：

Step 1：初始化种群数目 λ，优异种群数目 μ，变异率 σ，初代个体 θ_0，适应度函数 $F(\cdot)$；

Step 2：利用标准正态分布函数产生扰动 ε_i，令子代个体 $\theta_i=\theta_0+\sigma\cdot\varepsilon_i$ (其中 $i=1,2,\cdots,\lambda$)；

Step 3：计算适应度函数值 $f_i=F(\theta_i)$，并对所有的 f_i 进行排序，根据排序结果选取对应的最优的 μ

个个体 θ^j (其中 $j=1,2,\cdots,\mu$)；

Step 4：将 μ 个个体 θ^j 进行加权求和得到新个体 θ_n，并将 θ_n 作为新一代的 θ_0；

Step 5：重复上述步骤 Step 2~Step 4 直到满足终止条件。

4. 基于经典进化策略的 MPC 控制

对于本文的小车避障运动控制问题，由于系统(3)为线性系统，故代价函数(5)是一个标准的二次规划问题。就理论上而言，在满足既定约束条件的前提下，可以快速实时地求解出最优控制序列 $\boldsymbol{U}_p^*(k)$。然而随着障碍场景复杂化，约束条件的设置难度也随之增大，尤其是对于存在局部最优陷阱的障碍场景而言，$\boldsymbol{U}_p^*(k)$ 甚至无法求解。因此，本文在传统 MPC 的基础上引入经典进化策略算法，构建并离线训练策略网络，将策略网络的输出量与 MPC 的控制量进行叠加形成新的控制量 $\boldsymbol{U}(k)$，使得小车在不同的环境下都可以快速成功地避开障碍物并导航至目标位置。

4.1. 网络结构

本文所构建的策略网络如图 2 所示，该网络以 MPC 的状态量 $\boldsymbol{x}(k)$ 与小车航向角 θ 为作为输入，以 MPC 的控制量的调节量 $\boldsymbol{U}_p(k)$ 为输出，输入层与输出层之间包含两个隐含层。层与层之间通过激活函数进行连接，其中隐含层之间的激活函数为 ReLu 函数，隐含层与输出层之间的激活函数为输出在−1 到 1 之间的 tanh 函数，以使得策略网络的输出可以对 MPC 的控制量进行全覆盖的调节。当网络的输出为 0 时，控制器将输出原始 MPC 的最优控制量。

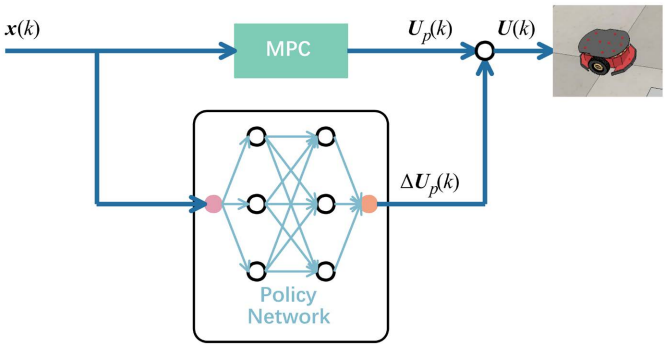

Figure 2. The policy network
图 2. 策略网络

4.2. 网络训练

在每一个 episode 中，以初始网络参数 θ_0 为基础通过变异产生 λ 个子网络并分别与原始 MPC 控制器结合成 λ 个 ES-MPC 控制器；在同一系统初始状态同时运行这 λ 个 ES-MPC 控制器；运行结束时每个控制器都对应产生一条数据轨迹(即小车的运动轨迹)。通过评估每一条轨迹的好坏程度就可以选出性能最为优异的 μ 个网络并产生下一个 episode 的初始网络参数 θ_0，由此不断更新网络从而完成网络的训练。

4.3. 代价评估

在策略网络的训练过程中，代价评估是最为关键的环节之一(这个环节相当于 RL 里的奖励函数设置)，代价函数的设置体现了对小车期望动作的要求，因此将直接影响到网络的训练效率以及算法的性能。为了准确评估 episode 中每一个策略网络的优异程度，本节设置了如下的代价函数用于评估策略网络输出的好坏：

1) 碰撞代价 C_c

避免小车在导航的过程中与环境中的障碍物发生碰撞，是 ES-MPC 算法设计中必须要遵循的重要原则，因此我们引入碰撞代价对 ES-MPC 运行过程中发生了小车碰撞的轨迹进行惩罚。ES-MPC 中碰撞代价的设计也代替了原始 MPC 算法中障碍约束的设计。

引入状态布尔量 $ColliState\left(\boldsymbol{u}_n\left(k\right)\right)$，它的值代表第 k 个采样时刻小车执行控制序列 $\boldsymbol{U}\left(k\right)$ 的前 n 个时刻的控制量 $\boldsymbol{u}_n\left(k\right)$ 后小车的碰撞状态(其中 $n=0,1,\cdots,N_c$)，0 代表未碰撞，1 代表碰撞，$C_c\left(\boldsymbol{u}_n\left(k\right)\right)$ 代表相应的代价值。当 $ColliState\left(\boldsymbol{u}_n\left(k\right)\right)$ 为 1 时，轨迹的全局碰撞 $ColliState$ 也设为 1。状态 $C_c\left(\boldsymbol{u}_n\left(k\right)\right)$ 的具体值有如下定义：

$$C_c\left(\boldsymbol{u}_n\left(k\right)\right)=\begin{cases}0, & ColliState\left(\boldsymbol{u}_n\left(k\right)\right)=0\\ \varepsilon^n a_c, & ColliState\left(\boldsymbol{u}_n\left(k\right)\right)=1\end{cases}$$

其中 a_c 为基础代价值，ε 为介于 0 和 1 之间的衰减因子，则第 k 时刻 $\boldsymbol{U}\left(k\right)$ 所对应的碰撞代价为

$$C_c\left(\boldsymbol{U}\left(k\right)\right)=\sum_{n=0}^{N_c}C_c\left(\boldsymbol{u}_n\left(k\right)\right)$$

2) 路径代价 C_p

路径代价 C_p 反映了小车对目标点导航效果的好坏，其意义与代价函数(3)相似，但是路径代价 C_p 反映了小车对目标点的全局导航效果，我们自然希望小车以最短路径逼近目标点。$\boldsymbol{p}_n\left(k\right)$ 代表第 k 个采样时刻小车执行控制序列 $\boldsymbol{U}\left(k\right)$ 的前 n 个时刻的控制量 $\boldsymbol{u}_n\left(k\right)$ 后的位置，并定义 $\boldsymbol{p}_{-1}\left(k\right)=\boldsymbol{p}\left(k\right)$ 为小车当前时刻的位置，路径代价 $C_p\left(\boldsymbol{u}_n\left(k\right)\right)$ 的具体值有如下定义：

$$C_p\left(\boldsymbol{u}_n\left(k\right)\right)=k_p\cdot\left\|\boldsymbol{p}_n\left(k\right)-\boldsymbol{p}_{n-1}\left(k\right)\right\|^2+k_g\cdot\left\|\boldsymbol{p}_n\left(k\right)-\boldsymbol{p}_g\right\|^2$$

其中 k_p 与 k_g 为惩罚系数，$\boldsymbol{p}_g$ 为目标位置，则第 k 时刻 $\boldsymbol{U}\left(k\right)$ 所对应的路径代价为

$$C_p\left(\boldsymbol{U}\left(k\right)\right)=\sum_{n=0}^{N_c}C_p\left(\boldsymbol{u}_n\left(k\right)\right)$$

3) 搜索代价 C_e

以上两个代价函数——C_c 与 C_p 的设置体现了小车运动规划的两个目标：避障，以及快速导航到目标位置；然而在策略网络训练的初期，C_c 与 C_p 很有可能会导致矛盾。例如当种群内的所有策略网络所输出的策略都让小车发生碰撞，而路径代价 C_p 的存在又限制了种群进化的出路，导致优化陷入死胡同。因此，设置搜索代价 C_e 从而鼓励种群内个体进行广泛搜索。$C_e\left(\boldsymbol{u}_n\left(k\right)\right)$ 的具体值有如下定义：

$$C_e\left(\boldsymbol{u}_n\left(k\right)\right)=-k_e\cdot\left\|\boldsymbol{p}_n\left(k\right)-\boldsymbol{p}\left(0\right)\right\|^2$$

其中 k_e 为搜索因子，$\boldsymbol{p}\left(0\right)$ 为小车初始时刻位置。由于该代价量为增益量，因此取负号。则第 k 时刻 $\boldsymbol{U}\left(k\right)$ 所对应的搜索代价为

$$C_e\left(\boldsymbol{U}\left(k\right)\right)=\sum_{n=0}^{N_c}C_e\left(\boldsymbol{u}_n\left(k\right)\right)$$

则长度为 N 的轨迹的总代价值为

$$C=\begin{cases}\sum_{k=1}^{N}C_p\left(\boldsymbol{U}\left(k\right)\right), & ColliState=0\\ \sum_{k=1}^{N}C_c\left(\boldsymbol{U}\left(k\right)\right)+C_e\left(\boldsymbol{U}\left(k\right)\right), & ColliState=1\end{cases}$$

5. 实验结果与分析

在本节中，我们在仿真实验中评估了 ES-MPC 算法的可行性。此外，我们还在不同的仿真场景中进行了与其他算法的对比实验，实验结果表明了 ES-MPC 的优越性。

5.1. 场景搭建

CoppeliaSim (原 V-rep)是一款支持各类机器人运动学及动力学仿真的软件，提供有面向 MATLAB 的远程 API 函数接口，通过 API 函数接口可以从 MATLAB 脚本中调用相关函数获得机器人信息或控制机器人运动。因此我们在 CoppeliaSim 平台搭建了复杂障碍环境与小车物理模型并进行小车运动可视化，在 MATLAB 内搭建了相同的场景与小车动态模型并训练策略网络。图 3 展示了 CoppeliaSim 内的场景以及 MATLAB 内相应的场景，由于小车在仿真训练中被简化为一个质点，因此对障碍作了膨胀化处理。

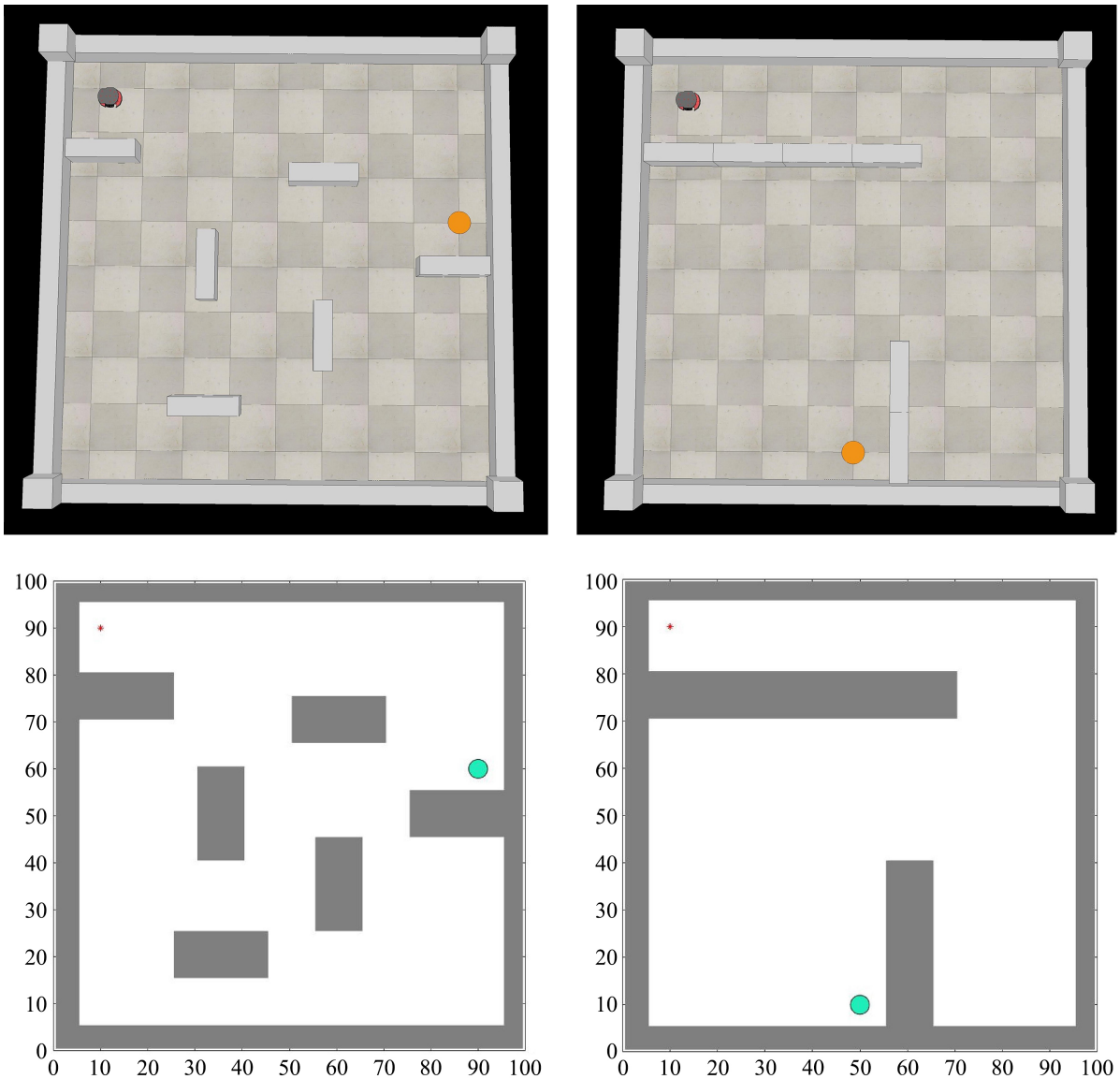

Figure 3. Scenes for simulation experiments
图 3. 仿真实验场景

5.2. 对比实验结果

在 ES-MPC 策略网络的训练过程中，全局目标 $\boldsymbol{X}_g$ 总是不变的，初始状态 $\boldsymbol{x}_0$ 则是随 episode 的更替而随机生成，以实现在绝大部分初始状态下小车都能成功导航到目标位置。策略网络训练过程中的参数设置情况如表 1 所示。

Table 1. Parameter setting
表 1. 参数设置

小车参数		ES 参数			
轮胎半径 r/m	0.0975	最大 episode	5000	基础代价 a_c	1000
轮间距 l/m	0.331	种群数目 λ	100	惩罚系数 k_p	6.6
MPC 参数		精英数目 μ	20	惩罚系数 k_g	0.18
预测步长 N_p	5	变异率 σ	0.032	衰减因子 ε	0.8
控制步长 N_c	4			搜索系数 k_e	0.16
运行步长 N	1000	隐含层神经元个数		128	
采样时间 T_s/s	0.05				

图 4 展示了在两个不同场景下，几类 MPC 算法所对应的小车运动轨迹。由图 4 可知在该场景 1 中，在不添加障碍约束的情况下，原始 MPC、HI-MPC 的事后预测步长 N'为 25 时无法完成导航任务，而 N'为 50 的 HI-MPC 与本文的 ES-MPC 都能提前预知障碍并导航至目标位置；而在场景 2 中由于存在局部最优陷阱，只有经过策略网络训练的 ES-MPC 能够寻找到一个好的避障策略并完成导航任务。

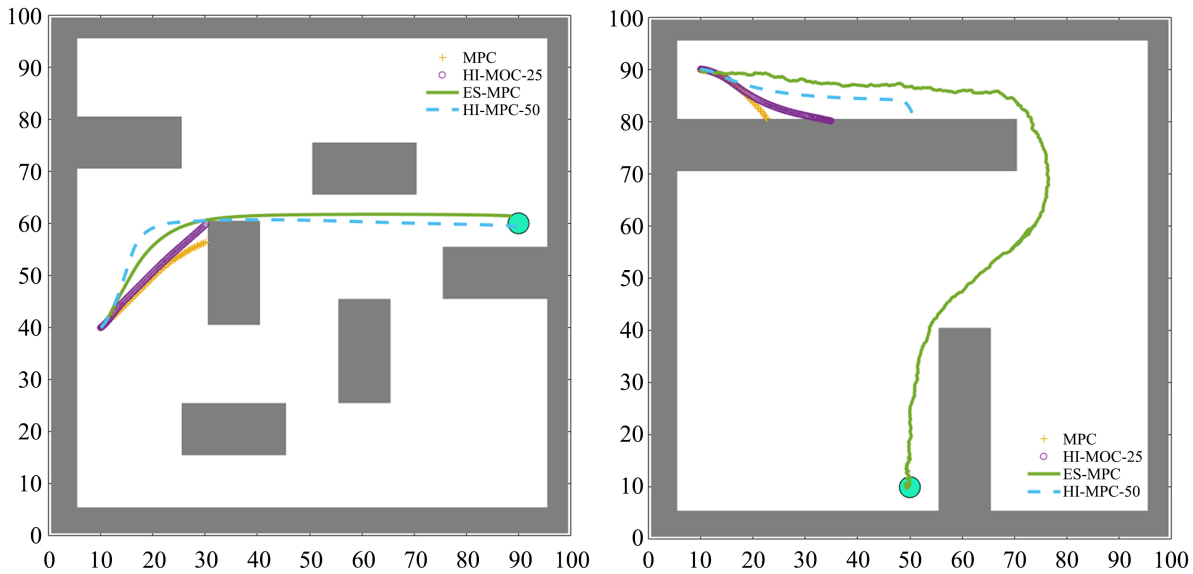

Figure 4. Figure of the comparison experiment
图 4. 对比实验图

此外，我们还在不同的初始状态或目标状态下进行了多次实验，实验结果如表 2 所示。根据不同的状态，我们将场景分为“开放”与“封闭”两种类型。由表 2 可知，在开放场景中，由于任务难度不大，因此 3 种 MPC 算法都可以完成导航；然而在封闭场景中，N'为 50 的 HI-MPC 可以完成一部分任务，而本文的 ES-MPC 则可以完成全部任务，这充分体现了 ES-MPC 的可行性与优越性。

Table 2. Comparative experimental results
表 2. 对比实验结果

场景 1					
起始点	航向角	场景类型	MPC	HI-MPC (N' = 50)	ES-MPC
(10, 90)	0˚	开放	√	√	√
(20, 40)	30˚	封闭	×	√	√
(50, 20)	60˚	封闭	×	√	√
(90, 10)	90˚	封闭	×	×	√
场景 2					
起始点	航向角	场景类型	MPC	HI-MPC (N' = 50)	ES-MPC
(90, 90)	−90˚	开放	√	√	√
(10, 90)	0˚	封闭	×	×	√
(90, 30)	90˚	封闭	×	×	√

5.3. 小车运动仿真结果

在上一节中我们成功训练了策略网络，在这一节中我们将使用策略网络控制小车在 CoppeliaSim 场景内运动。对模型(1)直接进行采样离散化可得

$$\begin{bmatrix} \dot{p}_x(k) \\ \dot{p}_y(k) \\ \dot{\theta}(k) \end{bmatrix} = \begin{bmatrix} \frac{r\cos\theta(k)}{2} & \frac{r\cos\theta(k)}{2} \\ \frac{r\sin\theta(k)}{2} & \frac{r\sin\theta(k)}{2} \\ -\frac{r}{l} & \frac{r}{l} \end{bmatrix} \cdot \begin{bmatrix} w_l(k) \\ w_r(k) \end{bmatrix} \tag{6}$$

由此可得小车双轮转速与模型(2)中小车坐标轴方向上的移动速度以及航向角变化率之间的关系。航向角变化率并未显含于模型(2)中，但由几何关系，有

$$\theta(k+1)+\theta(k) = 2\times\arctan\left(\frac{\dot{p}_y(k)}{\dot{p}_x(k)}\right)$$

因此有

$$\dot{\theta}(k) = \frac{\theta(k+1)-\theta(k)}{T_s} = \frac{2}{T_s}\times\left(\arctan\left(\frac{\dot{p}_y(k)}{\dot{p}_x(k)}\right)-\theta(k)\right)$$

由此可以决定小车在 k 时刻的双轮转速。

图 5 展示了 CoppeliaSim 内小车的运动轨迹，对应于图 4 中 ES-MPC 的轨迹，由图 5 可知基于模型(2)所训练的策略网络可以顺利地控制基于物理模型(1)的小车进行运动，由此验证了本文所提 ES-MPC 方法的可行性。理论上只要有足够的训练代数，所有可以显式建模的系统都可以使用 ES-MPC 方法。

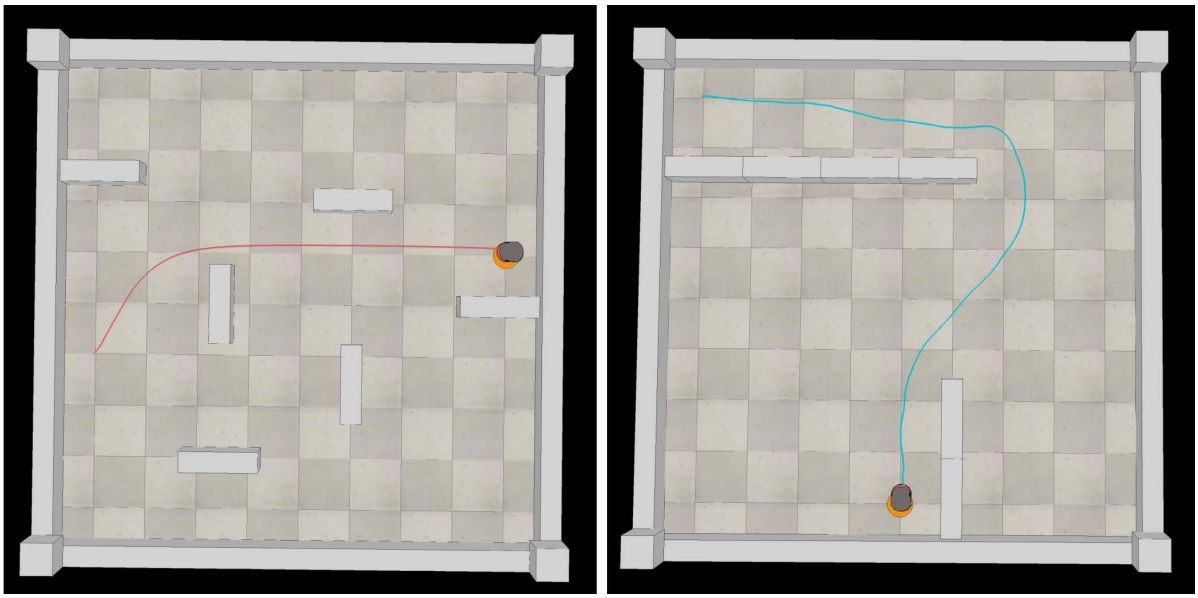

Figure 5. Cart trajectory in CoppeliaSim
图 5. CoppeliaSim 小车运动轨迹

6. 结论与展望

本文提出一种基于经典进化策略的模型预测控制方法(ES-MPC)，并用于移动小车的无碰路径规划任务。ES-MPC 通过种群进化训练策略网络，从而让小车在障碍环境中搜寻到避障策略。仿真实验结果显示，ES-MPC 方法可以更好地训练小车在局部最优陷阱中的避障决策能力。

目前，本文的工作仅仅针对静态场景，而没有考虑包含动态障碍的更加复杂的场景，对于动态场景的策略网络训练将更为复杂，这是未来的工作。另外，我们还将探索本文方法对于高维机械臂规划的可行性。

参考文献

[1] Zhong, M., Johnson, M., Tassa, Y., Erez, T. and Todorov, E. (2013) Value Function Approximation and Model Predictive Control. 2013 *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning* (*ADPRL*), Singapore, 16-19 April 2013, 100-107. https://doi.org/10.1109/ADPRL.2013.6614995

[2] 孙浩, 陈泽宇, 吴思凡, 潘峰. 基于策略梯度的智能车辆模型预测运动控制算法[J]. 汽车技术, 2021(12): 10-15. https://doi.org/10.19620/j.cnki.1000-3703.20210494

[3] Rokonuzzaman, M., Mohajer, N., Nahavandi, S. and Mohamed, S. (2020) Learning-Based Model Predictive Control for Path Tracking Control of Autonomous Vehicle. 2020 *IEEE International Conference on Systems, Man, and Cybernetics* (*SMC*), Toronto, 11-14 October 2020, 2913-2918. https://doi.org/10.1109/SMC42975.2020.9283293

[4] Bristow, D.A., Tharayil, M. and Alleyne, A.G. (2006) A Survey of Iterative Learning Control. *IEEE Control Systems*, **26**, 96-114. https://doi.org/10.1109/MCS.2006.1636313

[5] Wang, Y., Gao, F. and Doyle, F.J. (2009) Survey on Iterative Learning Control, Repetitive Control, and Run-to-Run Control. *Journal of Process Control*, **19**, 1589-1600. https://doi.org/10.1016/j.jprocont.2009.09.006

[6] Rosolia, U. and Borrelli, F. (2018) Learning Model Predictive Control for Iterative Tasks. A Data-Driven Control Framework. *IEEE Transactions on Automatic Control*, **63**, 1883-1896. https://doi.org/10.1109/TAC.2017.2753460 .

[7] Ioffe, S. and Szegedy, C. (2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *International Conference on International Conference on Machine Learning*, Lille, 6-11 July 2015, 448-456.

[8] Salimans, T., Ho, J., Chen, X., Sidor, S. and Sutskever, I. (2017) Evolution Strategies as a Scalable Alternative to Reinforcement Learning. https://arxiv.org/abs/1703.03864

[9] Chrabaszcz, P., Loshchilov, I. and Hutter, F. (2018) Back to Basics: Benchmarking Canonical Evolution Strategies for Playing Atari. *Proceedings of the* 27*th International Joint Conference on Artificial Intelligence*, Stockholm, 13-19 July 2018, 1419-1426. https://doi.org/10.24963/ijcai.2018/197

[10] Tamar, A., Thomas, G., Zhang, T., Levine, S. and Abbeel, P. (2017) Learning from the Hindsight Plan—Episodic MPC Improvement. 2017 *IEEE International Conference on Robotics and Automation* (*ICRA*), Singapore, 29 May-3 June 2017, 336-343. https://doi.org/10.1109/ICRA.2017.7989043

[11] 邹凯, 蔡英凤, 陈龙, 等. 基于增量线性模型预测控制的无人车轨迹跟踪方法[J]. 汽车技术, 2019(10): 1-7.

[12] Mnih, V., Kavukcuoglu, K., Silver, D., *et al*. (2013) Playing Atari with Deep Reinforcement Learning. https://arxiv.org/abs/1312.5602

[13] Nguyen, H. and La, H. (2019) Review of Deep Reinforcement Learning for Robot Manipulation. 2019 3*rd IEEE International Conference on Robotic Computing* (*IRC*), Naples, 25-27 February 2019, 590-595. https://doi.org/10.1109/IRC.2019.00120

[14] Mülling, K., Kober, J., Kroemer, O. and Peters, J. (2013) Learning to Select and Generalize Striking Movements in Robot Table Tennis. *The* 80 *International Journal of Robotics Research*, **32**, 263-279. https://doi.org/10.1177/0278364912472380

[15] Dan, S. (2013) Evolutionary Optimization Algorithms: Biologically Inspired and Population-Based Approaches to Computer Intelligence. Wiley & Sons, Inc., Hoboken.