高等动力学

Paper No.5
移动小车的动力学建模与控制

姓名	学号	学院
朱明仁	3120200445	机械与车辆学院
周高立	3120200443	机械与车辆学院

2020 年 12 月 25 日

北京理工大學
BEIJING INSTITUTE OF TECHNOLOGY

Contents

摘　要

本报告为高等动力学课程结课大作业报告，主要内容为：（1）理解原论文 Dynamic Object Tracking Control for a Non-Holonomic Wheeled Autonomous Robot 的动力学建模过程；（2）理解原论文关于移动小车轨迹控制的技术方法并进行实验复现；（3）自由探索其他可供研究的内容。我们在对原论文进行研读的基础上，结合课堂所学理解了移动小车的动力学建模过程，在查阅资料后理解了原论文中所谈及的两种控制方法：李雅普诺夫方法与计算力矩法，并利用 Simulink 进行控制器的构建与仿真，调参后得到了与原论文相似的控制结果，完成了对原论文的复现。此外，我们还根据现代控制理论中的最优控制方法对移动小车的动力学控制技术进行了研究，并利用神经网络与深度强化学习技术对两种最优控制方法进行了实验探索，实验表明基于最小值原理所得的离散控制方法比一般变分法所得的连续控制方法更加容易得到稳定的控制策略。

关键词: 移动小车，李雅普诺夫，计算力矩法，最优控制，强化学习

1. 原论文回顾

1.1 移动小车的动力学建模

1.1.1 移动小车的结构示意图

原论文所研究的移动小车的结构示意图如图1.1（原论文中为 Figure.1）。

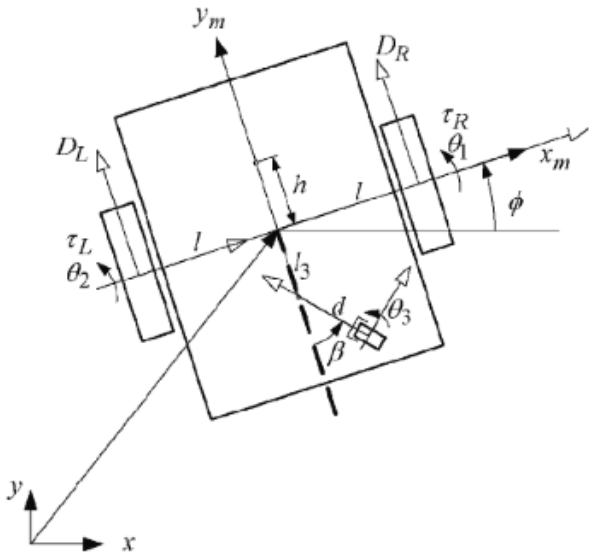

图 1.1: 移动小车结构示意图

由图1.1可见，原论文中一共为小车引入了 7 个状态变量：

(1) 小车几何中心的横坐标 x；

(2) 小车几何中心的纵坐标 y；

(3) 小车的偏向角 ϕ；

(4) 小车右轮的自转角 θ_1；

(5) 小车左轮的自转角 θ_2；

(6) 小车导向轮的自转角 θ_3；

(7) 小车导向轮的偏向角 β。

其中 x,y,ϕ 为移动小车的主要状态变量，$\theta_1,\theta_2,\theta_3,\beta$ 为内禀状态变量。之后会看到主要状态变量是研究和控制的对象，而内禀状态变量将会被用来自动满足约束条件而消去。

图1.1还引入了 4 个几何尺寸：

(1) 小车的半身宽度 l；

(2) 小车导向轮偏转轨道中心到小车几何中心的距离 l_3；

(3) 小车导向轮偏转轨道半径 d；

(4) 未知参数 h，原论文并未交代其几何意义，虽然在原论文 Figure.6 中出现，但根据上下文此处的 h 应该是 l_3。

另外还有两个几何尺寸在图1.1中未画出，左右轮的半径 r 和导向轮的半径 r_3。

图1.1中的 D_R、D_L 为两轮上的驱动力，τ_R、τ_L 为产生驱动力的驱动力矩，作为后续控制所用的输入量。

1.1.2 移动小车的非完整约束

接下来原论文考虑了小车所受到的运动约束，首先是小车三个轮子的纯滚动约束：

$$-\dot{x}\sin\phi+\dot{y}\cos\phi-r\dot{\theta}_1=0 \tag{1.1}$$

$$\dot{x}\sin\phi-\dot{y}\cos\phi+r\dot{\theta}_2=0 \tag{1.2}$$

$$-\dot{x}\sin\phi+\beta+\dot{y}\cos\phi+\beta-l_3\dot{\phi}\sin\beta+r_3\dot{\theta}_3=0 \tag{1.3}$$

其次是三个轮子的无侧向滑动约束：

$$\dot{x}\cos\phi+\dot{y}\sin\phi=0 \tag{1.4}$$

$$-\dot{x}\cos\phi-\dot{y}\sin\phi=0 \tag{1.5}$$

$$\dot{x}\cos\phi+\beta+\dot{y}\sin\phi+\beta+(d+l_3\cos\beta)\dot{\phi}+d\dot{\beta}=0 \tag{1.6}$$

由式 (1.1) 和式 (1.2) 得左右两轮的角速度可表示为三个主要状态变量的函数：

$$\begin{bmatrix}\omega_R\\ \omega_L\end{bmatrix}=\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix}=\frac{1}{r}\begin{bmatrix}-\sin\phi & \cos\phi & l\\ -\sin\phi & \cos\phi & -l\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\phi}\end{bmatrix} \tag{1.7}$$

式 (1.4) 和式 (1.5) 则代表了关于主要状态变量的同一个约束：

$$\begin{bmatrix}\cos\phi & \sin\phi & 0\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\phi}\end{bmatrix} = \mathbf{A}(\mathbf{q})^T\dot{\mathbf{q}} = 0 \tag{1.8}$$

而对于式 (1.3) 和式 (1.6) 而言，由于小车的导向轮是从动轮，无主动力驱动，且质量忽略不计，则可以通过自由调整 $\theta_3, \dot{\theta}_3, \beta, \dot{\beta}$ 来满足。这样，整个系统就可以使用主要状态变量 $\mathbf{q}$ 及其导数 $\dot{\mathbf{q}}$ 来完全表示。

1.1.3 移动小车的动力学建模

引入带非完整约束的拉格朗日方程：

$$\frac{d}{dt}\frac{\partial \mathbf{L}}{\partial \dot{\mathbf{q}}} - \frac{\partial \mathbf{L}}{\partial \mathbf{q}} = \mathbf{A}(\mathbf{q})\lambda + \mathbf{B}(\mathbf{q})\mathbf{u} \tag{1.9}$$

其中拉格朗日量 $\mathbf{L}(\mathbf{q},\dot{\mathbf{q}}) = \frac{1}{2}\dot{\mathbf{q}}^T\mathbf{M}(\mathbf{q})\dot{\mathbf{q}}$，$\mathbf{M}(\mathbf{q})$ 为位于移动小车几何中心的固定坐标系下的刚度矩阵：

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix}M & 0 & 0\\ 0 & M & 0\\ 0 & 0 & I_0\end{bmatrix} \tag{1.10}$$

M 是小车车身的质量，I_0 是小车车身的转动惯量。

将拉格朗日量带入式 (1.9) 可得：

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} = \mathbf{A}(\mathbf{q})\lambda + \mathbf{B}(\mathbf{q})\mathbf{u} \tag{1.11}$$

其中：

$$\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) = \frac{d\mathbf{M}(\mathbf{q})}{dt} - \frac{1}{2}\frac{\partial}{\partial \mathbf{q}}[\dot{\mathbf{q}}^T\mathbf{M}(\mathbf{q})] = \mathbf{0} \tag{1.12}$$

式 (1.9) 另一项广义力 $\mathbf{B}(\mathbf{q})u$ 是由小车左右轮上的驱动力生成的：

$$\mathbf{B}(\mathbf{q})\mathbf{u} = \begin{bmatrix}-\sin\phi & -\sin\phi\\ \cos\phi & \cos\phi\\ l & -l\end{bmatrix}\begin{bmatrix}D_R\\ D_L\end{bmatrix} \tag{1.13}$$

将以上分析整理为：

$$M\ddot{x} = \lambda \cos\phi - (D_R + D_L)\sin\phi \tag{1.14}$$

$$M\ddot{y} = \lambda \sin\phi + (D_R + D_L)\cos\phi \tag{1.15}$$

$$I_0\ddot{\phi} = l(D_R - D_L) \tag{1.16}$$

式 (1.14)-(1.15) 联合式 (1.8) 代表的非完整约束，这就是系统完整的动力学方程。

1.1.4 移动小车的状态控制方程

由式 (1.8) 可知广义速度 $\dot{\mathbf{q}}$ 必须存在于 $\mathbf{A}(\mathbf{q})$ 的零空间中。求解零空间的一组基构成的矩阵为：

$$\mathbf{S}(\mathbf{q}) = \begin{bmatrix} -\sin\phi & 0 \\ \cos\phi & 0 \\ 0 & 1 \end{bmatrix} \tag{1.17}$$

则广义速度可以表示为：

$$\dot{\mathbf{q}} = \mathbf{S}(\mathbf{q})\eta(\mathbf{q}, \dot{\mathbf{q}}) \tag{1.18}$$

求解式 (1.18) 可得：

$$\eta(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} -\dot{x}\sin\phi + \dot{y}\cos\phi \\ \dot{\phi} \end{bmatrix} \tag{1.19}$$

同时由于朗格朗日乘子 λ 是无法进行观测的，不利于控制系统的设计，因此需要想办法将其消除。以 $\mathbf{S}(\mathbf{q})^T$ 左乘式 (1.11) 并代入式 (1.18) 整理可得：

$$\dot{\eta} = \begin{bmatrix} \frac{1}{M} & \frac{1}{M} \\ \frac{l}{I_0} & -\frac{l}{I_0} \end{bmatrix} \begin{bmatrix} D_R \\ D_L \end{bmatrix} \tag{1.20}$$

这就变成了一组新的状态变量的状态方程。然而，在实际控制中主动轮的驱动力不是很方便作为控制量的输入，相较之下驱动力矩则能更好地得到调整，因此还需要进行控制量的变换。原论文假设主动轮的力系示意如图1.2所示（原论文中为 Figure.2）。

其中 D 为驱动力，τ 为驱动力矩，ω 为自转角速度，I_w 为自转转动惯量，c 为阻尼系数。不考虑转弯的情况（$\dot{\phi} \ll \omega$）则对于任意一个主动轮有：

$$I_w\dot{\omega} = \tau - rD - c\omega \tag{1.21}$$

同时结合式 (1.7) 和式 (1.18) 可得：

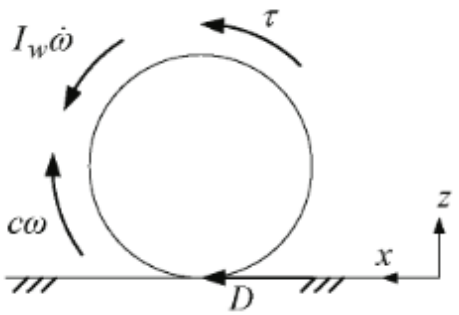

图 1.2: 主动轮上的作用力系

$$\begin{bmatrix}\omega_R\\ \omega_L\end{bmatrix} = \frac{1}{r}\begin{bmatrix}1 & l\\ 1 & -l\end{bmatrix}\begin{bmatrix}\eta_1\\ \eta_2\end{bmatrix} \tag{1.22}$$

将式 (1.21) 和式 (1.22) 代入式 (1.20) 可得:

$$\mathbf{M}'\dot{\eta} + \mathbf{C}'\eta = \mathbf{B}'\tau \tag{1.23}$$

其中:

$$\mathbf{M}' = \begin{bmatrix}\frac{Mr^2+2I_w}{Mr} & 0\\ 0 & \frac{I_0r^2+2I_wl^2}{I_0r}\end{bmatrix} \tag{1.24}$$

$$\mathbf{C}' = \begin{bmatrix}\frac{2c}{Mr} & 0\\ 0 & \frac{2cl^2}{I_0r}\end{bmatrix} \tag{1.25}$$

$$\mathbf{B}' = \begin{bmatrix}\frac{1}{M} & \frac{1}{M}\\ \frac{l}{I_0} & -\frac{l}{I_0}\end{bmatrix} \tag{1.26}$$

这就是原论文最终的状态控制方程，相应的系统方框图如图1.3所示（原论文的 Figure.3）。

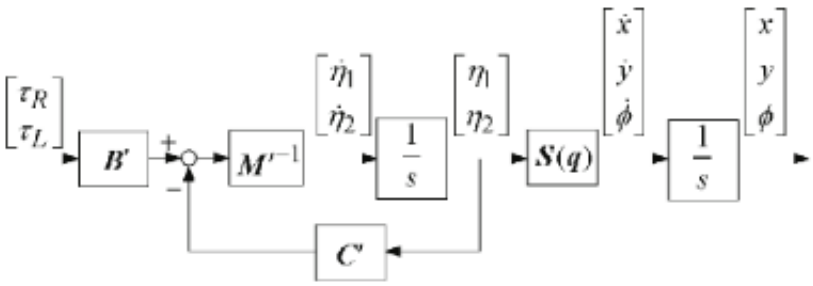

图 1.3: 系统方框图

1.2 移动小车的控制器设计

1.2.1 李雅普诺夫控制器

李雅普诺夫控制器是利用李雅普诺夫第二方法构造反馈控制，使得系统的目标状态成为某类稳定点。设李雅普诺夫函数如下：

$$V(\eta,\tilde{\mathbf{q}}) = \frac{1}{2}\eta^T \mathbf{M}'\eta + \frac{1}{2}\tilde{\mathbf{q}}^T \mathbf{K_P}\tilde{\mathbf{q}} \tag{1.27}$$

其中比例增益矩阵 $\mathbf{K_P}$ 是一个对称正定阵。$\tilde{\mathbf{q}} = \mathbf{q}_d - \mathbf{q}$ 为位置偏差，$\mathbf{q}_d$ 为期望位置。

对式 (1.27) 求关于时间的一阶导数并将式 (1.23) 代入可得：

$$\dot{V} = -\eta^T \mathbf{C}'\eta + \eta^T(\mathbf{B}'\tau - \mathbf{S}^T \mathbf{K_P}\tilde{\mathbf{q}}) \tag{1.28}$$

原论文中取：

$$\mathbf{B}'\tau = \mathbf{S}^T(\mathbf{K_P}\tilde{\mathbf{q}} - \mathbf{K_D}\dot{\mathbf{q}}) \tag{1.29}$$

其中微分增益矩阵 K_D 为对称正定阵。

那么式 (1.28) 变为：

$$\dot{V} = -\eta^T \mathbf{C}'\eta - \dot{\mathbf{q}}^T \mathbf{K_D}\dot{\mathbf{q}} \leq 0 \tag{1.30}$$

根据李雅普诺夫第二方法的稳定性判定法则可知，$\mathbf{q} = \mathbf{q}_d$ 为系统的一个李雅普诺夫意义下的稳定点。稳定点的必要条件是：

$$\eta = \mathbf{0} \tag{1.31}$$

$$\dot{\mathbf{q}} = \mathbf{0} \tag{1.32}$$

$$\mathbf{B}'\tau = \mathbf{0} \tag{1.33}$$

当 $\mathbf{K_P}$ 取为正定球形张量时可得稳定点 $y = y_d, \phi = \phi_d$，然而，x 可能并不会到达期望值。使用李雅普诺夫控制器的系统方框图如图1.4所示（原论文 Figure.4）。

1.2.2 计算力矩控制器

为了弥补李雅普诺夫控制器无法对横坐标进行良好控制的不足，原论文提出了第二种控制方法：计算力矩法。

引入新的状态变量：

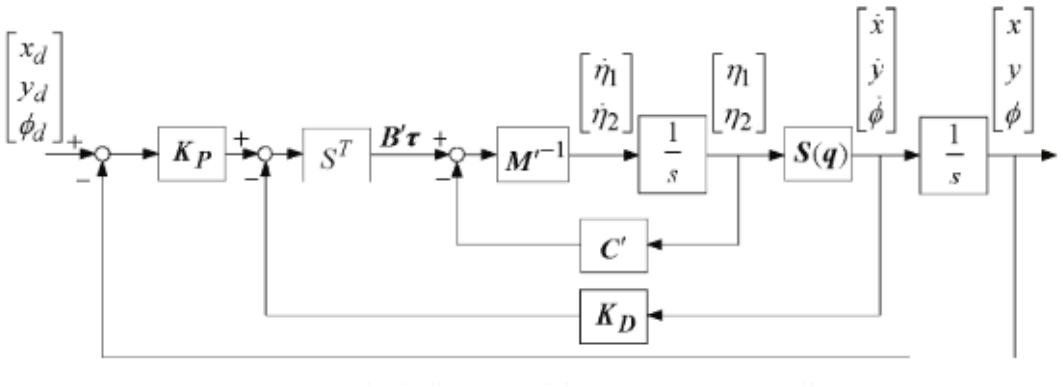

图 1.4: 李雅普诺夫控制器系统方框图

$$\overline{\mathbf{q}} = \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} = \begin{bmatrix} x + l_3 \sin\phi \\ y - l_3 \cos\phi \end{bmatrix} \tag{1.34}$$

对式 (1.34) 求关于时间的二阶导数并将式 (1.18) 代入可得:

$$\ddot{\overline{\mathbf{q}}} = \mathbf{C}''(\mathbf{q}, \dot{\mathbf{q}})\eta + \mathbf{B}''\mathbf{v} \tag{1.35}$$

其中:

$$\mathbf{C}''(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} -\dot{\phi}\cos\phi & -l_3\dot{\phi}\sin\phi \\ -\dot{\phi}\sin\phi & l_3\dot{\phi}\cos\phi \end{bmatrix} \tag{1.36}$$

$$\mathbf{B}'' = \begin{bmatrix} -\sin\phi & l_3\cos\phi \\ \cos\phi & l_3\sin\phi \end{bmatrix} \tag{1.37}$$

$$\dot{\eta} = \mathbf{v} \tag{1.38}$$

使用计算力矩法设计 $\mathbf{v}$ 控制率:

$$\mathbf{v} = \mathbf{B}''^{-1}(-\mathbf{C}''\eta + \ddot{\overline{\mathbf{q}}}_\mathbf{d} + \mathbf{K_P}\tilde{\mathbf{q}} + \mathbf{K_D}\dot{\tilde{\mathbf{q}}}) \tag{1.39}$$

其中 $\mathbf{K_P}, \mathbf{K_D}$ 为对称正定阵，$\tilde{\mathbf{q}} = \overline{\mathbf{q}}_\mathbf{d} - \overline{\mathbf{q}}$。注意此处与李雅普诺夫控制器中的三阶不同，均变为二阶的了。另外:

$$B''^{-1} = \begin{bmatrix} -\sin\phi & \cos\phi \\ \frac{1}{l_3}\cos\phi & \frac{1}{l_3}\sin\phi \end{bmatrix} \tag{1.40}$$

这里修正了原论文（式 (15)）中的 $\mathbf{B}''^{-1}$ 缺了一个负号的错误。

将式 (1.38) 代入式 (1.35) 可得:

$$\ddot{\tilde{\mathbf{q}}} + \mathbf{K_D}\dot{\tilde{\mathbf{q}}} + \mathbf{K_P}\tilde{\mathbf{q}} = \mathbf{0} \tag{1.41}$$

由此可知 $\bar{\mathbf{q}}$ 可以得到有效的控制。

结合式 (1.23)、式 (1.35) 与式 (1.38) 可得驱动力矩的控制率：

$$\tau = r\mathbf{H}'\mathbf{v} + \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \tag{1.42}$$

其中：

$$\mathbf{H}' = \begin{bmatrix} \frac{M}{2} & \frac{I_0}{2l} \\ \frac{M}{2} & -\frac{I_0}{2l} \end{bmatrix} \tag{1.43}$$

$$\mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = c\omega + I_w\dot{\omega} \tag{1.44}$$

ω 如式 (1.7)（原论文采用）或式 (1.22)。

使用计算力矩控制器的系统方框图如图1.5所示（原论文 Figure.6）。

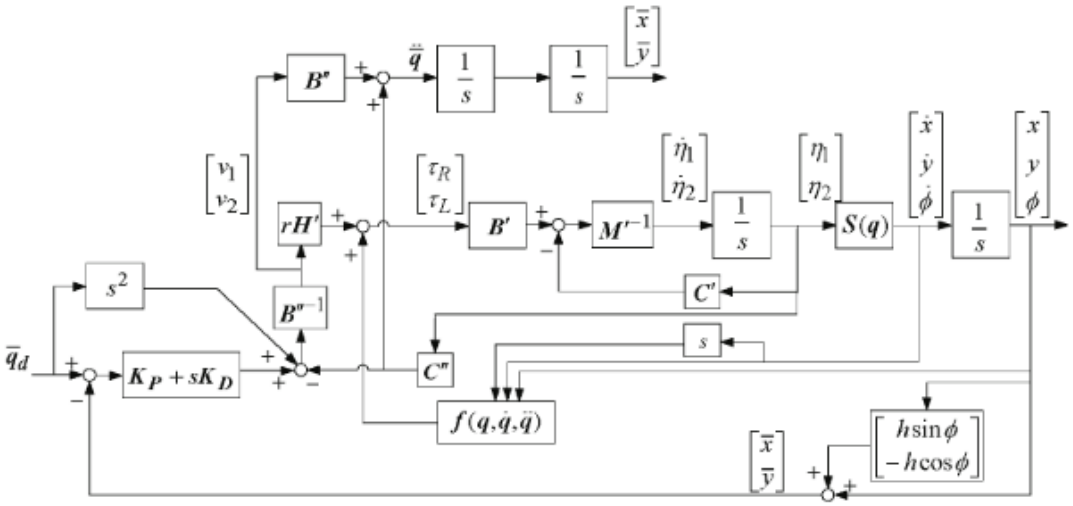

图 1.5: 计算力矩控制器系统方框图

如 1.1.1 小节所述，图1.5中的 h 应修改为 l_3。

2. 实验仿真复现

由于原论文中并没有给出移动小车和控制器的有关参数，所有的参数都必须由我们自己进行选择和调整，这为实验结果的复现带来了一定的难度。所幸我们根据一定的比例关系（不计量纲）得到了一组可行的移动小车参数，如表2.1所示。

表 2.1: 移动小车有关参数

小车质量	小车转动惯量	车身半宽	
$M = 10$	$I_0 = 10$	$l = 10$	
主动轮转动惯量	主动轮半径	主动轮阻尼系数	从动轮偏向中心距
$I_w = 1$	$r = 1$	$c = 1$	$l_3 = 0.5$

后续控制器设计中所遇到的常数矩阵均可通过表2.1中的参数值计算得出。实验复现的环境为 Win10 Matlab 2016b。

2.1 李雅普诺夫控制器实验

我们通过 Simulink 构建的系统方框图如图 2.1 所示。

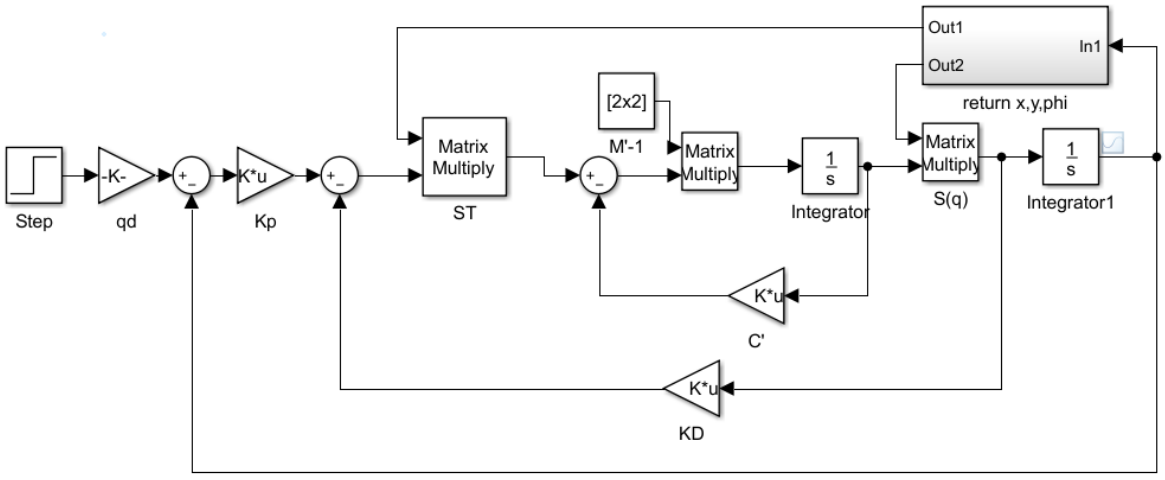

图 2.1: 李雅普诺夫控制器-Simulink

控制器中有两个三阶常参数矩阵 $\mathbf{K_P}, \mathbf{K_D}$，经过调参之后所得结果为:

$$\mathbf{K_P} = \mathbf{Diag}(1,2,1) \tag{2.1}$$

$$\mathbf{K_D} = \mathbf{Diag}(5,20,1) \tag{2.2}$$

和原论文一样，我们设置了不同初始位置下的移动小车控制实验，目标状态为系统原点 $\mathbf{q}_d^T = (0,0,0)$。不同的初始位置如表 2.2 所示。

表 2.2: 李雅普诺夫控制器实验初始位置

x	y	ϕ	x	y	ϕ
10	0	$-\pi$	5	0	$-\pi$
-10	0	$-\pi$	-5	0	$-\pi$
0	10	$-\pi$	0	5	$-\pi$
0	-10	$-\pi$	0	-5	$-\pi$
7	7	$-\pi$	3	3	$-\pi$
7	-7	$-\pi$	3	-3	$-\pi$
-7	7	$-\pi$	-3	3	$-\pi$
-7	-7	$-\pi$	-3	-3	$-\pi$

在这些初始状态的设置下通过李雅普诺夫控制器进行移动小车的目标控制结果如图 2.2 所示。原论文的结果如图 2.3 所示（原论文 Figure.8）。

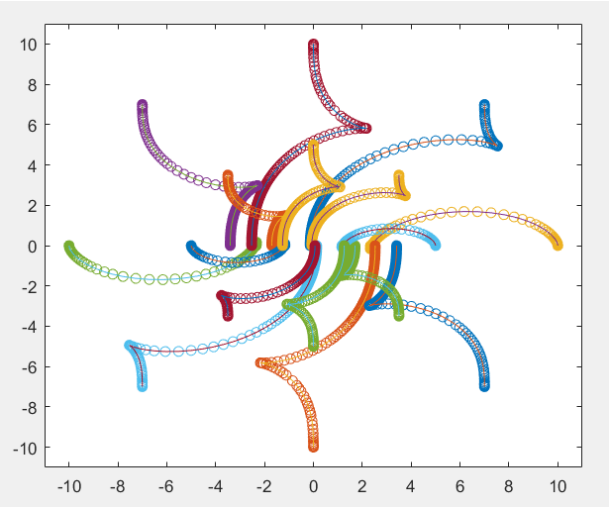

图 2.2: 实验复现结果

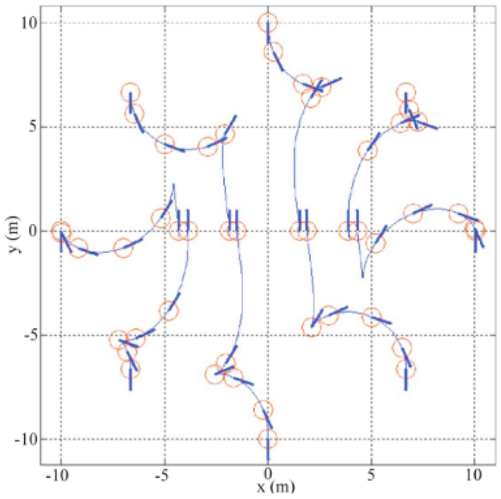

图 2.3: 原论文结果

可以看到实验复现结果与原论文结果十分接近，两者的纵坐标和角度均会回到原点状态，而横坐标则不一定，这也符合 1.2.1 小节中的理论分析。

2.2 计算力矩控制器实验

我们通过 Simulink 构建的系统方框图如图 2.4 所示。

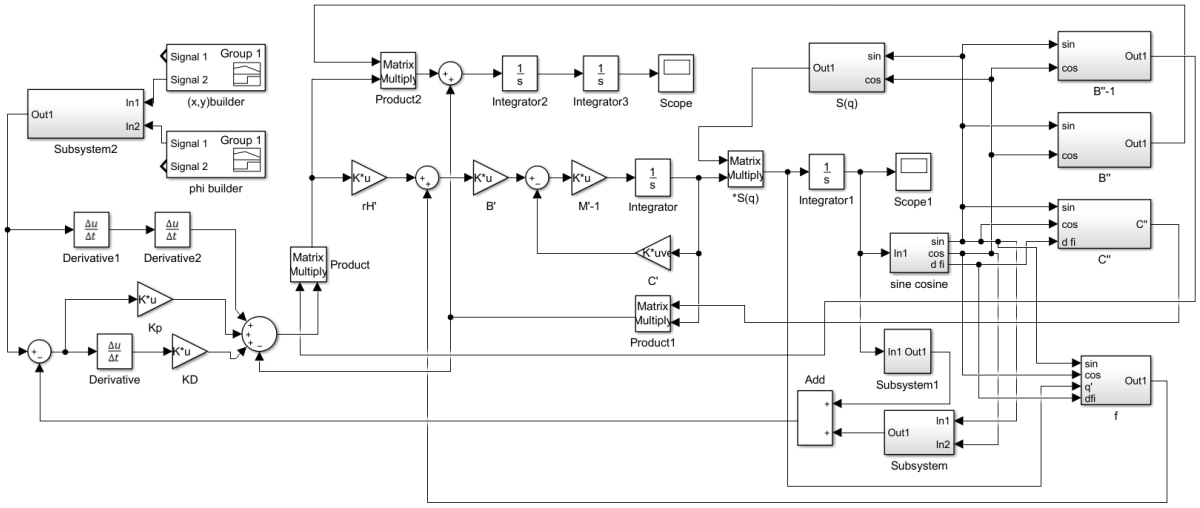

图 2.4: 计算力矩控制器-Simulink

控制器中有两个二阶常参数矩阵 $\mathbf{K_P}, \mathbf{K_D}$，经过调参之后所得结果为:

$$\mathbf{K_P} = \mathbf{Diag}(20, 20) \tag{2.3}$$

$$\mathbf{K_D} = \mathbf{Diag}(1, 1) \tag{2.4}$$

和原论文一样，我们设置了不同初始位置下的移动小车控制实验，目标状态为系统原点 $\mathbf{q}_d^T = (0, 0, 0)$。不同的初始位置如表 2.3 所示。

表 2.3: 计算力矩控制器实验初始位置

x	y	ϕ	x	y	ϕ
10	0	$-\pi$	10	0	$\pi/2$
-10	0	$-\pi$	-10	0	$-\pi/2$
0	10	$-\pi$	0	10	$-\pi$
0	-10	$-\pi$	0	-10	0
7	7	$-\pi$	7	7	$3\pi/4$
7	-7	$-\pi$	7	-7	$\pi/4$
-7	7	$-\pi$	-7	7	$-3\pi/4$
-7	-7	$-\pi$	-7	-7	$-\pi/4$

在表 2.3 左列初始状态的设置下通过计算力矩控制器进行移动小车的目标控制结果如图 2.5 所示。图 2.6 为原论文结果（原论文 Figure.9）。

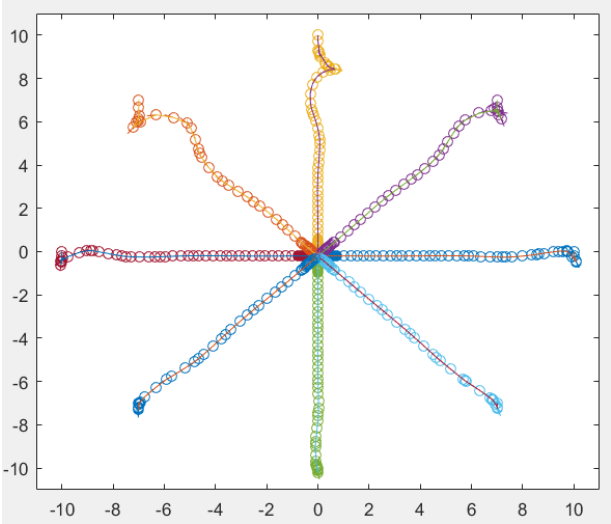

图 2.5: 实验复现结果 1

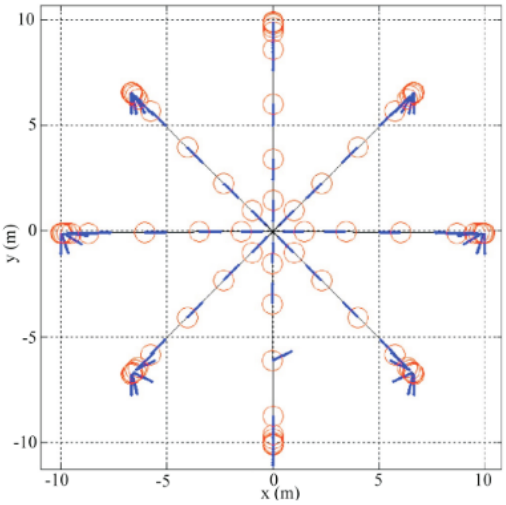

图 2.6: 原论文结果

可以看到的是，使用计算力矩控制器对移动小车进行轨迹控制时会首先调整小车的方向，然后小车将直奔目标位置，这与原论文所得的实验结果相似。

如果一开始设置角度正对位置原点，则小车将径直向目标移动并最终到达原点附近，如图 2.7 所示，其初始参数设置见表 2.3 右列。

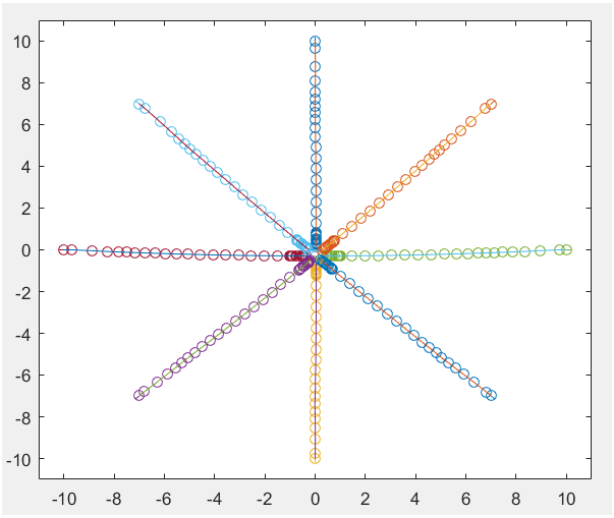

图 2.7: 实验复现结果 2（初始正对目标）

3. 最优控制分析

3.1 系统状态方程

移动小车的轨迹控制可以用现代控制理论中的最优控制方法进行分析，其一是一般形式的变分法，其二是控制输入量带有上下界约束的最小值原理。无论采用哪种方式进行分析，都必须先给出能完备描述系统状态变化的状态控制方程。

按照现代控制理论，系统原本的状态变量一共有 6 个，即：$x, y, \phi, \dot{x}, \dot{y}, \dot{\phi}$，但是由于约束式 (1.8) 的存在，系统的自由度少了一个，因此只需要 5 个状态变量即可完整描述系统的状态。这 5 个系统状态可以根据一定的规则自由选择，在此选为：

$$z_1 = x \tag{3.1}$$
$$z_2 = y \tag{3.2}$$
$$z_3 = \phi \tag{3.3}$$
$$z_4 = \eta_1 \tag{3.4}$$
$$z_5 = \eta_2 \tag{3.5}$$

再做变换：$u_1 = \tau_R + \tau_L, u_2 = \tau_R - \tau_L$，结合表 2.1 整理可得系统的状态方程如下：

$$\dot{z}_1 = -z_4 \sin z_3 \tag{3.6}$$
$$\dot{z}_2 = z_4 \cos z_3 \tag{3.7}$$
$$\dot{z}_3 = z_5 \tag{3.8}$$
$$\dot{z}_4 = -\frac{1}{6} z_4 + \frac{1}{12} u_1 \tag{3.9}$$
$$\dot{z}_5 = -\frac{20}{21} z_5 + \frac{1}{21} u_2 \tag{3.10}$$

采用矢量形式可以表达为：

$$\dot{\mathbf{z}} = \mathbf{f}(\mathbf{z}, \mathbf{u}) \tag{3.11}$$

之后的分析均基于此系统状态方程进行。

对于给定初始位置（初始速度为 0）和分段可积的控制输入量有积分形式的解：

$$z_1 = z_1(0) - \int_0^t z_4 \sin z_3 dt \tag{3.12}$$

$$z_2 = z_2(0) + \int_0^t z_4 \cos z_3 dt \tag{3.13}$$

$$z_3 = z_3(0) + \int_0^t z_5 dt \tag{3.14}$$

$$z_4 = \frac{1}{12} e^{-\frac{1}{6}t} \int_0^t e^{\frac{1}{6}\tau} u_1(\tau) d\tau \tag{3.15}$$

$$z_5 = \frac{1}{21} e^{-\frac{20}{21}t} \int_0^t e^{\frac{20}{21}\tau} u_2(\tau) d\tau \tag{3.16}$$

该解可由数值分析方法进行计算，然而若要想求得最优的控制输入量 $\mathbf{u}$ 使得某个性能指标达到最佳，则需要借助最优控制方法。

3.2 控制输入量无约束的情形

在控制输入量不受限制的情况下，上述问题可用一般变分法进行表达和求解。设性能泛函如下：

$$J = \int_0^{t_f} L(\mathbf{z}) dt, \quad L(\mathbf{z}) = z_1^2 + z_2^2 + z_3^2 \tag{3.17}$$

在控制输入量无约束（但要满足连续二阶可微）的条件下，最优控制问题可以用一般变分法表达为波尔扎问题：

$$min\ J = \int_0^{t_f} L(\mathbf{z}) dt \tag{3.18}$$

$$s.t.\ \dot{\mathbf{z}} = \mathbf{f}(\mathbf{z}, \mathbf{u}) \tag{3.19}$$

$$\mathbf{z}(0) = \mathbf{z_0} \tag{3.20}$$

$$\mathbf{z}(t_f) = \mathbf{0} \tag{3.21}$$

求解该变分问题可得关于最优状态轨线和最优控制输入的偏微分方程组：

$$\frac{\partial H}{\partial \mathbf{u}} = \mathbf{0} \tag{3.22}$$
$$\frac{\partial H}{\partial \mathbf{z}} = -\dot{\boldsymbol{\lambda}} \tag{3.23}$$
$$\frac{\partial H}{\partial \boldsymbol{\lambda}} = \dot{\mathbf{z}} \tag{3.24}$$

其中 H 为系统的哈密顿函数，$\boldsymbol{\lambda}$ 为拉格朗日乘子：

$$H = L + \boldsymbol{\lambda}^T \mathbf{f}(\mathbf{z}, \mathbf{u}) \tag{3.25}$$

由式 (3.22) 可得控制方程：

$$\frac{1}{12}\lambda_4 = 0 \tag{3.26}$$
$$\frac{1}{21}\lambda_5 = 0 \tag{3.27}$$

由式 (3.23) 可得协态方程：

$$\dot{\lambda}_1 = -2z_1 \tag{3.28}$$
$$\dot{\lambda}_2 = -2z_2 \tag{3.29}$$
$$\dot{\lambda}_3 = -2z_3 + \lambda_1 z_4 \cos z_3 + \lambda_2 z_4 \sin z_3 \tag{3.30}$$
$$\dot{\lambda}_4 = \lambda_1 \sin z_3 - \lambda_2 \cos z_3 + \frac{1}{6}\lambda_4 \tag{3.31}$$
$$\dot{\lambda}_5 = -\lambda_3 + \frac{20}{21}\lambda_5 \tag{3.32}$$

由式 (3.24) 可得状态方程式 (3.11)。
整理上述偏微分方程组可得：

$$\dot{\mathbf{z}} = \mathbf{f}(\mathbf{z}, \mathbf{u}) \tag{3.33}$$
$$\dot{z}_3 z_4 (\cos z_3 - z_3 \sin z_3) - \dot{z}_4 z_3 \cos z_3 = z_1 z_4^2 \tag{3.34}$$
$$\dot{z}_3 z_4 (\sin z_3 + z_3 \cos z_3) - \dot{z}_4 z_3 \sin z_3 = z_2 z_4 \tag{3.35}$$

以上常微分方程组的解在初始速度为 0 的条件下有且仅有一个：

$$\mathbf{z} = \mathbf{z}_0 \tag{3.36}$$

$$\mathbf{u} = \mathbf{0} \tag{3.37}$$

即状态将完全保持不变！这显然不符合控制的要求。即不存在连续二阶可微的控制输入量能使得任意速度为 0 的初始状态下的小车回到状态原点。

3.3 控制输入量有界的情形

当控制输入量有界时，上述最优控制问题可以使用最小值原理进行求解。设控制输入量的绝对值不超过 1，原问题可表达如下：

$$min\ J = \int_0^{t_f} L(z)dt \tag{3.38}$$

$$s.t.\ \dot{\mathbf{z}} = \mathbf{f}(\mathbf{z}, \mathbf{u}) \tag{3.39}$$

$$\mathbf{z}(0) = \mathbf{z}_0 \tag{3.40}$$

$$\mathbf{z}(t_f) = \mathbf{0} \tag{3.41}$$

$$|\mathbf{u}| \leq 1 \tag{3.42}$$

根据极小值原理有可得与 3.2 小节相同的状态方程、协态方程与边界条件，不过状态方程则变为了下面的极小值原理：

$$\mathbf{u} = \min_{|\mathbf{u}| \leq 1} H(\mathbf{z}, \mathbf{u}, \lambda) \tag{3.43}$$

由极小值原理可得：

$$u_1 = -sign(\lambda_4) \tag{3.44}$$

$$u_2 = -sign(\lambda_5) \tag{3.45}$$

其中 *sign* 为符号函数。

显然若 $\lambda_4 = \lambda_5 = 0$，则极小值原理所得结果与一般变分法相同，初始速度为 0 的情况下小车状态将恒定不变。也就是说极小值原理的解空间包含了一般变分法的情况，这将使得极小值原理能够拥有比一般变分法更大的可能性。这一点在数学上也有体现，虽然输入控制量受到上下界的制约，但是不必要再满足连续二阶可微的条件，只需要为分段常值函数即可。

4. 深度强化学习探索

强化学习与控制论结合十分紧密，在神经网络和深度学习兴起之后，以 DQN（Deep Q-learning Network）为代表的一系列深度强化学习算法在控制领域大放异彩，取得了不少令人惊叹的成就。本节主要是探索性的研究，使用两种深度强化学习算法来对移动小车进行轨迹控制，观察它们的效果。

4.1 马尔科夫决策过程

马尔科夫决策过程（Markov Decision Process，MDP）是动态系统的状态在系统进行某一动作下基于马尔科夫性而发生变化的随机过程，在状态变化的同时将产生相应的奖励回馈给系统。这里涉及到 MDP 的三个主要要素：状态、动作以及奖励。

状态（s）是指系统各个元素及其属性的值构成的集合，通过状态可以了解系统的一切。在 MDP 中，系统的状态变化满足马尔科夫性，即下一时刻的状态将仅由这一时刻的状态所决定而无关于在此之前时刻的历史状态，用概率形式表达如下：

$$P(s_{t+1}|s_t, s_{t-1}, \dots) = P(s_{t+1}|s_t) \tag{4.1}$$

动作（a）是指系统在处于某一状态时采取的行为，该行为参与对系统状态变化的影响。在 MDP 中，系统在特定状态下的行为是一个随机变量，即：

$$a_t \sim \pi(a|s_t) \tag{4.2}$$

奖励（r）是指系统在采取动作使得状态发生变化间所得的收益，在 MDP 中一般是动作以及状态变化的函数，即：

$$r_t = R(s_t, a_t, s_{t+1}) \tag{4.3}$$

由以上三要素所组成的序列 $(s_t, a_t, r_t)_{t=0}^{T}$ 即为 MDP 的一条采样序列 τ。

定义 MDP 状态值函数为当前状态及之后的累积奖励期望，即：

$$v(s_t) = E_\tau(\sum_{i=0} r_{t+i}) \tag{4.4}$$

为了保证以上序列能够收敛，同时考虑奖励的时间效应，可以引入折扣系数 $\gamma \in (0,1]$，上式变为：

$$v(s_t) = E_\tau(\sum_{i=0} \gamma^i r_{t+i}) \tag{4.5}$$

定义 MDP 状态-动作值函数为当前状态下执行当前动作之后的累积奖励期望，即：

$$q(s_t, a_t) = E_{\tau|a_t}(\sum_{i=0} \gamma^i r_{t+i}) \tag{4.6}$$

易知两者之间存在如下关系：

$$v(s_t) = \sum_{a_t} \pi(a_t|s_t) q(s_t, a_t) \tag{4.7}$$

$$q(s_t, a_t) = \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t)(r_t + \gamma v(s_{t+1})) \tag{4.8}$$

结合两式可得：

$$q(s_t, a_t) = \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t)(r_t + \gamma \sum_{a_{t+1}} \pi(a_{t+1}|s_{t+1})) \tag{4.9}$$

式 (4.9) 称为状态-动作值函数的贝尔曼方程，是众多强化学习算法的基础核心。

虽然以上关于 MDP 的描述均是基于随机过程的，但是实际上对于确定性系统而言也同样适用，只需要将状态转移概率修改为状态方程即可，这也是本节所采用的方法，状态方程见式 (3.11)。因此此处的状态即为 $\mathbf{z}$，动作即为 $\mathbf{u}$。对于用深度强化学习技术做系统控制而言，是以神经网络为反馈调节器，将系统的状态输入神经网络，神经网络输出相应的动作值再输入系统中，同时迭代调节神经网络的参数，使之能成为一个好的反馈调节器。

4.2 离散控制：DDQN

DDQN（Double DQN）是一类基于值近似的深度强化学习算法，它利用深度神经网络对马尔可夫决策过程中的状态-动作价值函数（被称为 Q 函数）进行逼近。在一个特定的状态下，一个动作的价值越大代表它越能够带来更大的长期收益，因此每次都应该选择价值最大的动作执行。

DDQN 的模块结构如图 4.1 所示。

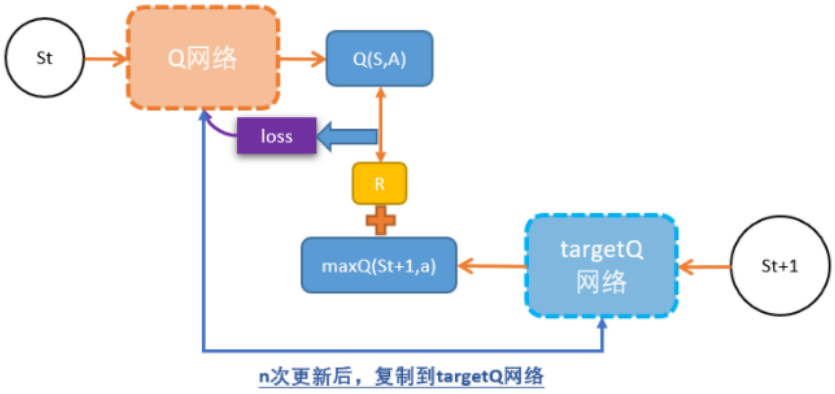

图 4.1: DDQN 模块结构图

从图 4.1 中可以清楚地看到这个算法被命名为 DDQN 的原因：不仅有一个 Q 网络，还有一个 targetQ 网络。通过这种双网络的构造，DDQN 相对于 DQN 可以提高训练过程的稳定性，使得神经网络更快收敛。

DDQN 的具体算法如图 4.2 所示。

Algorithm 1: Double DQN Algorithm.

input : $\mathcal{D}$ – empty replay buffer; θ – initial network parameters, θ^- – copy of θ
input : N_r – replay buffer maximum size; N_b – training batch size; N^- – target network replacement freq.
for *episode* $e \in \{1, 2, \ldots, M\}$ **do**
  Initialize frame sequence $\mathbf{x} \leftarrow ()$
  for $t \in \{0, 1, \ldots\}$ **do**
    Set state $s \leftarrow \mathbf{x}$, sample action $a \sim \pi_{\mathcal{B}}$
    Sample next frame x^t from environment $\mathcal{E}$ given (s, a) and receive reward r, and append x^t to $\mathbf{x}$
    if $|\mathbf{x}| > N_f$ **then** delete oldest frame $x_{t_{min}}$ from $\mathbf{x}$ **end**
    Set $s' \leftarrow \mathbf{x}$, and add transition tuple (s, a, r, s') to $\mathcal{D}$,
      replacing the oldest tuple if $|\mathcal{D}| \geq N_r$
    Sample a minibatch of N_b tuples $(s, a, r, s') \sim \text{Unif}(\mathcal{D})$
    Construct target values, one for each of the N_b tuples:
    Define $a^{\max}(s'; \theta) = \arg\max_{a'} Q(s', a'; \theta)$
    $y_j = \begin{cases} r & \text{if } s' \text{ is terminal} \\ r + \gamma Q(s', a^{\max}(s'; \theta); \theta^-), & \text{otherwise.} \end{cases}$
    Do a gradient descent step with loss $\|y_j - Q(s, a; \theta)\|^2$
    Replace target parameters $\theta^- \leftarrow \theta$ every N^- steps
  end
end

图 4.2: DDQN 算法

结合 3.3 小节的分析，设置动作的输出共有 9 种，即：

$$\begin{aligned}(u1, u2) \in \{&(1,1),(1,0),(1,-1),\\ &(0,1),(0,0),(0,-1),\\ &(-1,1),(-1,0),(-1,-1)\}\end{aligned} \tag{4.10}$$

实验环境为 Win10 Python3.7，深度学习框架为 Tensorflow2.3。

实验设置为从状态 $(10, 10, 0, 0, 0)$ 出发，控制回到系统原点。迭代训练 500 次，其中第 350 次迭代结果如图 4.3 所示。

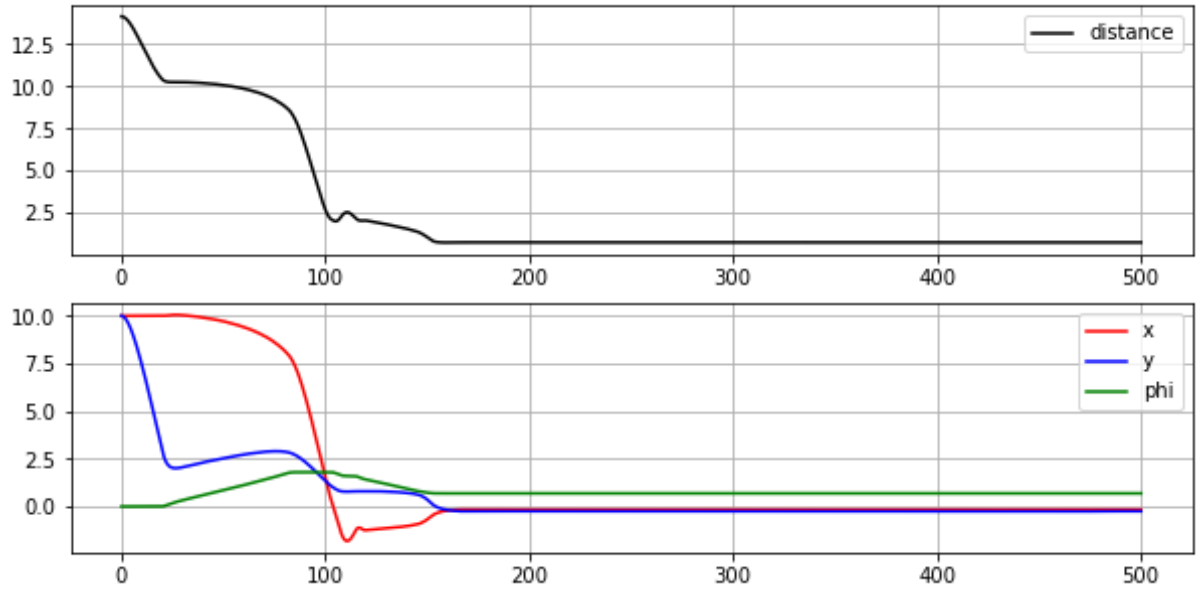

图 4.3: DDQN 实验结果图（No.350）

图 4.3 的上半部分代表小车的状态距离系统原点的欧氏距离随时间的变化，下半部分代表小车的位置和角度随时间的变化。可以看到小车在大约 150 个时间单位之后达到稳定，且距离系统原点十分接近。

在实验中与图 4.3 相似的模式大约占总迭代次数的近 50% 的比例，这说明采用离散控制的 DDQN 算法具有良好的稳定性。

4.3 连续控制：DDPG

由于 DQN 本身的特点使得其只能对离散动作空间（如式 (4.10)）进行建模，因此对连续控制需要换一种新的算法。DDPG（Deep Determine Policy Gradient）是一类基于策略梯度的深度强化学习算法，它采用 Actor-Critic 框架进行训练。Critic 部分实际上是一个 Q 网络，用来近似状态-动作值函数，Actor 部分则是接受状态作为输入而直接以动作值作为输出的，因此训练好之后只需要保留 Actor 部分即可。

DDPG 的模块结构如图 4.4 所示。

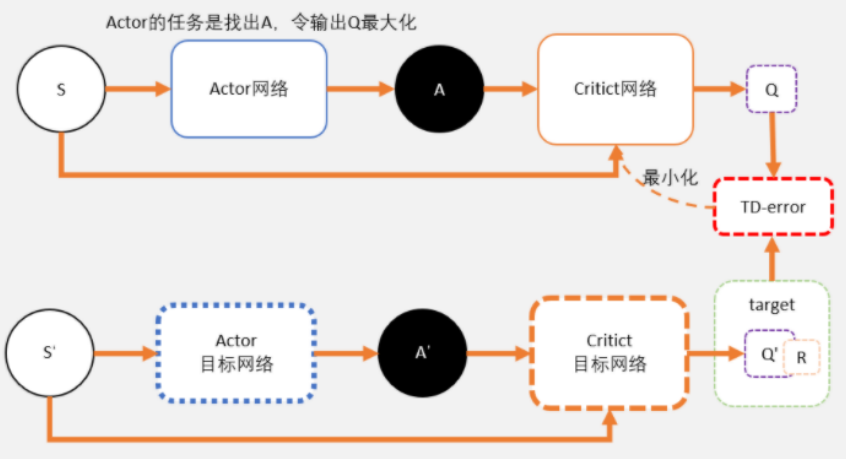

图 4.4: DDPG 模块结构图

从图 4.4 中可以看到 DDPG 算法一共用到了 4 个网络：两个 Actor 与两个 Critic，即借鉴了 DDQN 的双网络结构。

DDQN 的具体算法如图 4.5 所示。

Algorithm 1 DDPG algorithm

Randomly initialize critic network $Q(s,a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights θ^Q and θ^μ.
Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer R
for episode = 1, M **do**
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial observation state s_1
  for t = 1, T **do**
    Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration noise
    Execute action a_t and observe reward r_t and observe new state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in R
    Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from R
    Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
    Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
    Update the actor policy using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i \nabla_a Q(s,a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i}$$

    Update the target networks:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$
$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$$

  end for
end for

图 4.5: DDPG 算法

在连续动作控制下，不设置边界条件，也就是任何实数值都可取。

与 4.2 小节相同的实验环境和设置，迭代训练 500 次，其中第 450 次迭代结果如图 4.6 所示。

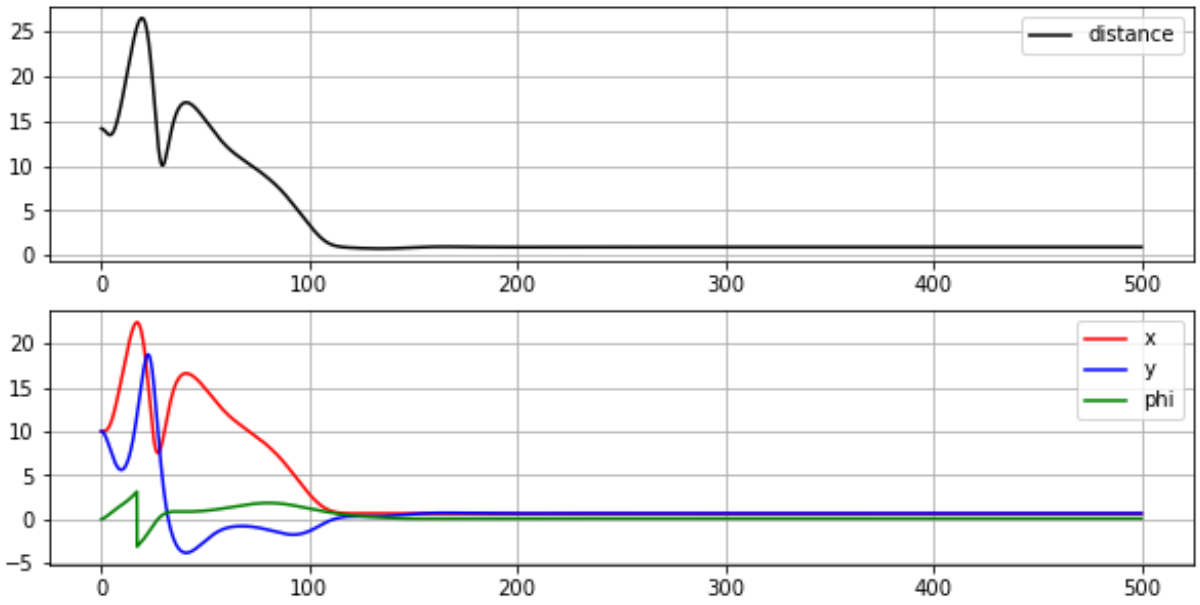

图 4.6: DDPG 实验结果图（No.450）

可以看到小车在大约 110 个时间单位后达到稳定，且距离系统原点十分接近，甚至超过了 DDQN 实验的结果。然而，在总共 500 次迭代中，类似于图 4.6 的模式仅仅出现约 2%，这个比例远小于 DDQN 的离散控制结果，这意味着 DDPG 的稳定性远不如 DDQN。事实上，采用 DDPG 算法对移动小车进行控制可以看作是 3.2 小节的近似，其如此不稳定的表现除了与 DDPG 算法本身有关外，还可能如 3.2 小节中的理论分析所言，连续的控制输入量是很难将小车回到系统原点的。

5. 总结与讨论

本报告作为高等动力学课程结课大作业报告，首先回顾了论文 Dynamic Object Tracking Control for a Non-Holonomic Wheeled Autonomous Robot 的动力学建模与控制器设计过程，然后利用 Simulink 仿真工具箱对原论文中的实验进行了复现构建，最后基于最优控制方法和深度强化学习对移动小车系统进行了分析探索。

本次报告实践中遇到的难点主要有以下三点：

(1) 在原论文的阅读理解过程中，因为我们之前并没有接触过计算力矩控制这一方法，因此也需要查阅资料将其理解清楚；

(2) 在原论文的实验复现过程中，因为原论文中并没有给出实验的相关参数，因此将系统模型在 Simulink 中构建好了之后又花费了大量时间用于调整参数，使复现结果尽可能与原论文一致；

(3) 在自由探索环节，深度强化学习算法的调参工作也占了很大一部分，原因是深度学习方法典型的黑箱性，为了得到好的控制结果，需要反复调整模型和算法的参数。

然而尽管如此，本次报告实践过程仍具有很好的启发性，我们通过对已发表论文的阅读、理解、复现和探索，对课堂上所学的高等动力学知识有了更深入的了解，感受到了动力学与控制论的紧密联系，自由探索部分也增加了我们自身的研究和实践能力。总之，收获满满。高等动力学课程最终结束，在此感谢老师一个学期的辛勤指导！