

基于过零碰撞检测和自然极角的“板凳龙”运动研究

摘要

“板凳龙”，是一项盛行于我国南方的传统习俗活动。每逢元宵，当地人会将上百条板凳首尾相连，在龙头的带领下组成壮观的圆盘状长龙。本文主要研究板凳龙在一等距螺线轨道上行进时的运动状态，建立了基于过零碰撞检测和自然极角坐标的运动状态检测模型。

针对问题一，从第一节龙身的前把手开始，各把手可以构成一个**等距螺线上的等直线距离点列**。本文基于等距螺线在极坐标系下的方程以及余弦定理，可推出“板凳龙”每对相邻把手的极角满足的方程，再**根据龙头把手位置即可迭代确定其余所有把手的位置**。接着本文利用弧微分和微分方程，得到时间关于龙头把手极角的关系。最后通过链式法则串联起各相邻把手极角方程的**隐函数导数**，计算出各把手的速度，发现把手速度随时间逐渐变慢，且越靠后的把手运动越慢。

针对问题二，本文根据凸多边形性质反证法证明：对于凳角而言，**碰撞只会发生在龙头或第一个龙身的左侧凳角上**。然后利用正弦定理可推出各板凳四个凳角的坐标形式。本文以原点和待检验的点是否位于待检验的边缘异侧作为是否发生碰撞的判据，由此将四个可能发生碰撞的点关于外圈边沿所在向量的一**系列行列式的最大值作为碰撞检验函数**，当发生过零时，即发生碰撞。从而得出不发生碰撞的**最大终止时间为** 412.473838s。

针对问题三，本文仍使用问题二的思路，只将可变量由时间变为螺距，求得**最小螺距为** 0.417138m。在求解过程中，本文发现检验量与螺距呈近似线性关系。这引导我们发现一些这一本为复杂的几何动力系统的性质，例如迭代三角形中下一极径关于螺距近似恒定。

针对问题四，本文先基于几何证明得出：在给定的条件下，**调头路径唯一确定，且当两段圆弧的半径比可以控制时，调头路径的长度恒定**，并且排除了碰撞和后退，将调头路径分为三段。考虑到路径的极角与极径不能表示为一一对应关系，我们构建了在全过程连续变化，且与点的真实极角有简单关系的**自然极角**。最后结合关于真实极角的分段极坐标方程和关于自然极角的各变化量，求出各把手的位置和速度。

针对问题五，本文首先证明，在所有时刻下，**各板凳运动速度均与龙头把手速度成正比**（但不同时刻比值不同）。利用这一点，实际上我们只需要计算 $v_h = 1\text{m/s}$ 时关于 θ_0 （自然坐标）的最大值，再等比转化即可。分析可得最大值必落在调头区域内，通过**单纯形法**求得最大 $v_h = 1.244612\text{m/s}$ 。

关键词：等距螺线上的等距点列 链式法则 基于边沿同异侧的碰撞检验 自然极角 单纯形法

一、问题背景与重述

1.1 问题背景

“板凳龙”这一传统文化活动普遍存在于我国南方地区。每逢元宵，当地人会将上百条板凳由把手相连，在龙头的带领下组成蜿蜒的圆盘状长龙。现有某板凳龙包含 223 节板凳。龙头的板凳长 341 cm，其余板长均为 220 cm，所有板凳的宽度均为 30 cm。每节板凳上均有两个孔径为 5.5 cm 的圆孔，孔的中心距板头 27.5 cm。

1.2 问题重述

问题一：已知板凳龙盘入方式，行进速度，初始位置等信息。建立数学模型，给出从初始到 300s 为止，每秒舞龙队每个板凳的位置和速度。并给出其他不同时间节点，龙头及特定节板凳把手的位置与速度。

问题二：板凳龙沿问题 1 给出的螺线盘入，要求确定舞龙队在板凳发生碰撞时的位置与速度。同时给出刺是特定板凳前后把手的位置与速度。

问题三：设定一个以螺线中心为圆心，直径为 9m 的掉头空间。请确定最小螺距，使板凳龙可以沿路盘入此空间边界。

问题四：设定盘入螺线和盘出螺线的情况，让板凳龙在问题 3 给出的空间内调转龙头。要求掉头路径是与两圆弧相切连接而成的 S 形曲线，请求出最短掉头曲线。tn 此外，需要除调头开始时间前后，

问题五：板凳龙沿问题 4 给出的轨迹行进，确定龙头最大均匀行进速度，使整条板凳龙各把手速度在全过程不超过 2m/s。

二、问题分析

2.1 问题一的分析

问题一要求在等距螺线螺距、龙头前把手行进速度已知的情况下，求出从出发到 300s 为止，“板凳龙”的实时位置以及速度。已知从第一节龙身的前把手开始，各把手构成了一个等距螺线上的等直线距离点列。首先可通过等距螺线在极坐标系下的方程和余弦定理推出“板凳龙”相邻把手的极角满足的方程，分析可知它们的极角是可以相互确定的，因此根据龙头把手位置就可以迭代确定所有把手的位置。接着借助弧微分和微分方程等工具，可以算出时间关于龙头把手极角的解析关系，再利用数值反解得到龙头把手位置与时间的关系。最后通过链式法则串联起各相邻把手极角方程的隐函数导数，进而给出各把手的速度。

2.2 问题二的分析

问题二要求在问题一盘入方式的基础上，求出板凳发生碰撞的时间点以及此时“板凳龙”各把手的位置以及移速。我们首先根据凸多边形性质反证法证明论断：对于凳角而言，碰撞只会发生在龙头或第一个龙身的左侧凳角上，然后通过正弦定理推出各板凳四个凳角的坐标形式。我们以原点和待检验的点是否位于待检验的边缘异侧作为是否发生碰撞的判据，由此将四个可能发生碰撞的点关于外圈边沿所在向量的一系列行列式的最大值作为碰撞检验函数，当发生过零时，即为发生碰撞。

2.3 问题三的分析

问题三要求最小螺距，满足龙头前把手能沿螺线盘入调头空间。仍然可使用问题二的思路，将可变量由时间变为螺距即可。

2.4 问题四的分析

问题四要求在盘入螺线螺距 1.7m 的基础上，满足相切等各个要求，做出调头路径。并且求出在调头前后各把手的位置与移速。首先可证明：在调头始末点确定的情况下，调头路径唯一确定，所以无法调整圆弧使调头曲线变短。此外，几何分析也可以排除碰撞和后退的情况。接下来，为了沿用问题一的方法，但考虑到路径的极角与极径不能表示为一一对应关系，本文构建在全过程连续变化，且与点的真实极角有简单关系的自然极角。最后结合关于真实极角的分段极坐标方程和关于自然极角的各变化量，求出各把手的位置和速度。

2.5 问题五的分析

问题五要求在板凳龙诸把手速度都小于 2m/s 的要求下，求出龙头的最大速度。可以证明，在所有时刻下，各板凳运动速度均与龙头把手速度成正比。进而计算 $v_h = 1m/s$ 时关于 θ_0 （自然坐标）的最大值，再等比转化即可。分析可得最大值必落在调头区域内，通过**单纯形法**求解。

三、模型假设

1. 情境中等距螺线（即阿基米德螺线）的中心位于原点，极轴对应直角坐标系中的 x 轴正半轴，即方程满足 $r = k\theta$ 。
2. 现实中，“板凳龙”内相邻的两条板凳，是由前一节板凳后部盖在后一节前部上并通过把手相连的。虽然板凳实际存在厚度，但人们抬板凳的高度相差无几，可知板凳平面与地面存在一个夹角。又因为实际板凳的长度远大于厚度，故此夹角影响较小，可忽略。因此本文构建一个理想化模型，假设各板凳厚度近似于无穷小。虽然板凳间存在重叠，在此假设下就可将“板凳龙”近似看作在同一平面内

运动。进而我们定义：**板凳间发生碰撞，是指非相邻的两个板凳边界相交。**（相邻板凳重叠部分不算碰撞）

3. 为了整个舞龙队的美观，我们应当认为，当给定前一节板凳前把手中心的位置时，如果后一节板凳的前把手中心的位置有多种可能的情况，选择前把手中心极角增量最小的情况。

四、符号说明

符号	说明	值	单位
d	等距螺线螺距	/	m
l_h	龙头两把手间距	2.86	m
l	龙身/尾两把手间距	1.65	m
l_b	把手到板凳短边的距离	0.275	m
θ_b	板凳角和最近把手连线与板凳中轴的夹角	$\arctan \frac{6}{11}$	rad
w	板凳宽度	0.30	m
θ_b	板凳角和最近把手连线与板凳中轴的夹角	$\arctan \frac{6}{11}$	rad
l_c	板凳角到最近把手的距离	0.313249	m
v_h	龙头前把手的速度	/	m/s
v_i	龙身龙尾各把手的速度	/	m/s
θ_0	龙头前把手的极角	/	rad
θ_i	第 i 节龙身前把手的极角	/	rad
r_i	第 i 节龙身前把手的极径	/	m
A_i	第 i 节龙身左前凳角	/	/
B_i	第 i 节龙身左后凳角	/	/
C_i	第 i 节龙身右后凳角	/	/
D_i	第 i 节龙身右前凳角	/	/
R	掉头空间的半径	4.5	m
θ_{in}	螺线切入圆弧点的极角	16.6320	rad
θ_{out}	圆弧切出螺线点的极角	13.4904	rad
R_1	第一段圆弧的半径	3.0054	m
R_2	第二段圆弧的半径	1.5027	m
θ_{cin}	第一段圆弧圆心的极角	/	rad
θ_{cout}	第二段圆弧圆心的极角	/	rad
θ_{nature}	自然极角	/	rad
$\theta_{reverse}$	回转点的极角	12.9961	rad

五、模型的建立与求解

5.1 工作流程示意图

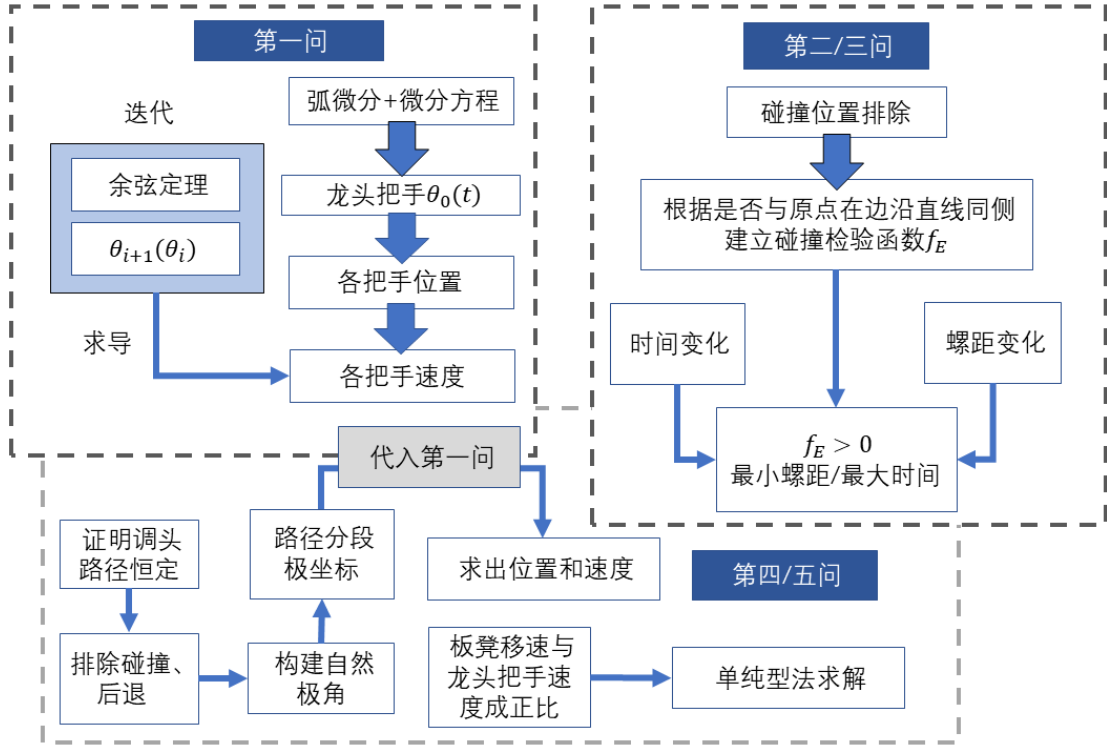


图 1 工作流程图

5.2 问题一：给定盘入轨迹、终止时刻和龙头速度的舞龙队运动状态确定

5.2.1 各把手的位置关系

根据题中所给信息，各把手的中心（后文中提到“把手位置”即指其中心位置）在等距螺线上运动。该等距螺线在极坐标系下的方程为：

$$r(\theta) = \frac{d\theta}{2\pi} \quad (1)$$

其中 d 是螺距。

由于等距螺线带来的约束，我们可以通过把手的极角的唯一的确定把手的位置。为方便起见，我们记第 i 节龙身前把手的极径为 r_i （龙头前把手为 r_0 ，龙尾后把手为 r_{223} ）。

第一节龙身的前把手到龙尾的后把手共 223 个点构成了一个等距螺线上等直线距离的点列。由于同一条板凳上的前后把手之间的距离固定，因此根据余弦定理，对第 i 节龙身的前后把手有

$$r_i^2 + r_{i+1}^2 - 2r_i r_{i+1} \cos(\theta_i - \theta_{i+1}) = l^2 \quad (2)$$

即

$$\theta_i^2 + \theta_{i+1}^2 - 2\theta_i\theta_{i+1}\cos(\theta_i - \theta_{i+1}) = \frac{4\pi^2 l^2}{d^2} (i = 1, 2, \dots, 224) \quad (3)$$

同样地，对龙头的前后把手有

$$\theta_0^2 + \theta_1^2 - 2\theta_0\theta_1\cos(\theta_0 - \theta_1) = \frac{4\pi^2 l_h^2}{d^2} \quad (4)$$

式 (3) 和式 (4) 具有相同的形式，以式 (3) 为例，给定 θ_i 时， θ_{i+1} 可能是多解的。根据假设，为了使整个舞龙队更加“舒展”，应当认为 θ_{i+1} 选取的是满足 $\theta_{i+1} > \theta_i$ 的最小的满足式 (3) 的值。

等距螺线上任一点处的曲率比过该点的以极点为圆心的圆的曲率要小。而在圆周上，给定小于直径的弦长和一个顶点的位置，在该顶点沿圆周向某一方向另一顶点的位置就确定，所以从直觉上来看，当 θ_i 的极径足够大时，我们应当在 $\theta_{i+1} \in (\theta_i, \theta_i + \pi]$ 范围内搜索解。事实上，由方程图象可知当 $\theta_i > 15$ 时，在 $(\theta_i, \theta_i + \pi]$ 范围内有唯一的 θ_{i+1} 。模型求解结果也显示，在当前参数下完全不会涉及到 $\theta_i \leq 15$ 的情形。

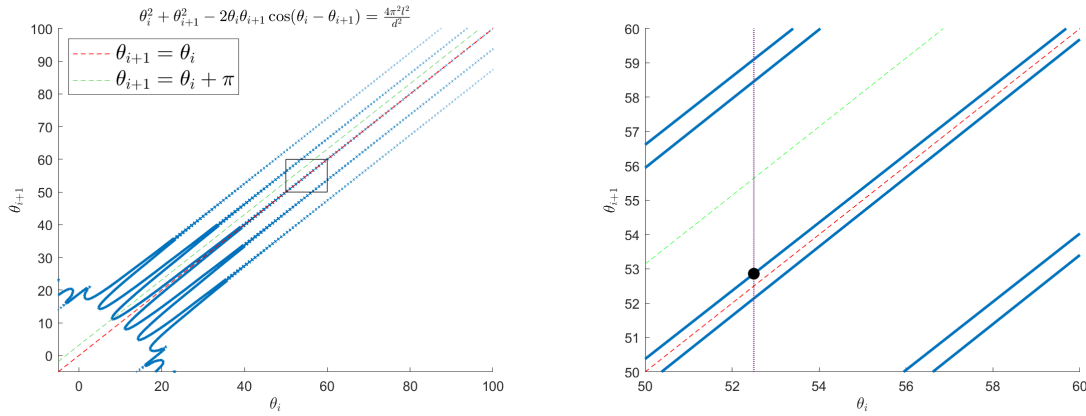


图 2 θ_{i+1} 和 θ_i 的关系示意图

通过迭代求解上述两个方程，我们可以由龙头前把手的极角 θ_0 依次得到龙身、龙尾各把手的极角 $\theta_i (1 \leq i \leq 223)$ 。

5.2.2 龙头前把手的极角与运动时间的关系

为了得到各把手在不同时间的位置，我们首先要求解龙头前把手的极角 θ_0 与时间 t 的关系，然后通过上述的把手间位置关系得到其他把手的极角。注意到，龙头前把手的速度为：

$$v_h = -\frac{dL_0}{dt} = -\frac{dL_0}{d\theta_0} \frac{d\theta_0}{dt} \quad (5)$$

其中 L_0 为龙头前把手到中心沿螺线的弧线距离。再根据弧微分公式有：

$$\frac{dL_0}{d\theta_0} = \frac{\sqrt{dr^2 + r^2 d\theta_0^2}}{d\theta_0} = \frac{d\sqrt{\theta_0^2 + 1}}{2\pi} \quad (6)$$

龙头前把手的速度为常量 $v_h = 1\text{m}$ ，我们得到 θ_0 关于 t 的微分方程：

$$\frac{d\theta_0}{dt} = -\frac{2\pi v_h}{d\sqrt{\theta_0^2 + 1}} \quad (7)$$

对于该微分方程，分离变量后两边积分得

$$\int_{\theta_0(0)}^{\theta_0} \sqrt{\theta_0^2 + 1} d\theta_0 = -\frac{2\pi v_h}{d} t \quad (8)$$

作双曲代换 $\theta_0 = \sinh(u)$ ，并利用三角恒等式

$$\cosh^2 u = \frac{1 + \cosh(2u)}{2} \quad (9)$$

我们可以得到 t 关于 θ_0 的解析结果：

$$t(\theta_0) = -\frac{d}{4v_h\pi} \left[\sinh^{-1}(\theta) + \theta\sqrt{\theta^2 + 1} \right] \Big|_{\theta_0(0)}^{\theta_0} \quad (10)$$

其中 $\theta_0(0) = 32\pi$ 为 0 时刻龙头把手的极角。理论上，通过数值方法可反解得到 $\theta_0(t)$ 。这里我们考虑直接对式 (7) 所示微分方程求解和对 $t(\theta_0)$ 进行反解，本质上是从两个不同的方向迭代，而且考虑到 $\theta_0(0) = 32\pi$ 虽是解析结果，但使用了反三角函数，多此一步反而增加了误差，所以当直接求 $\theta_0(t)$ 时，仍然直接对微分方程使用龙格-库塔法求解。

5.2.3 各把手的速度

将前两部分的结果复合，现在我们已经得到了各把手与运动时间的关系。龙头前把手的速度已知，接下来通过导数的链式法则求解其后各把手的速度：

$$\begin{aligned} v_i &= \frac{dL_i}{dt} \\ &= \frac{dL_i}{d\theta_i} \frac{d\theta_i}{dt} = \frac{dL_i}{d\theta_i} \frac{d\theta_i}{d\theta_{i-1}} \frac{d\theta_{i-1}}{dt} \\ &= \dots \\ &= \frac{dL_i}{d\theta_i} \left(\prod_{k=0}^{i-1} \frac{d\theta_{k+1}}{d\theta_k} \right) \frac{d\theta_0}{dt} \end{aligned} \quad (11)$$

与式 (6) 同样地，其中

$$\frac{dL_i}{d\theta_i} = \frac{d\sqrt{\theta_i^2 + 1}}{2\pi} \quad (12)$$

根据式 (3) 所示的隐函数，可得：

$$\frac{d\theta_{i+1}}{d\theta_i} = \frac{\theta_i - \theta_{i+1} \cos(\theta_i - \theta_{i+1}) + \theta_i \theta_{i+1} \sin(\theta_i - \theta_{i+1})}{\theta_i \cos(\theta_i - \theta_{i+1}) - \theta_{i+1} + \theta_i \theta_{i+1} \sin(\theta_i - \theta_{i+1})} \quad (13)$$

综合式 (11)(12)(13)，即可求解各把手的速度。

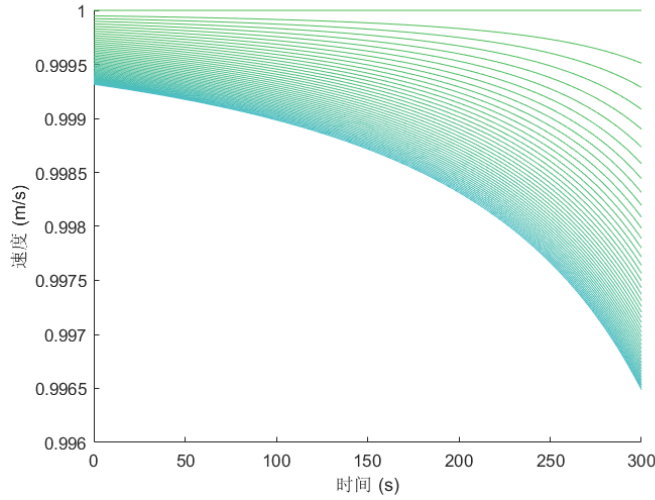


图 3 在 300s 内一些把手速度与时间的关系（从上至下 i 增大）

5.2.4 误差控制

误差在迭代的过程中会产生累积，因此我们主要控制 $\theta_{i+1}(\theta_i)$ 的误差，使结果的精度达到 10^{-6} 。

首先检查一下 $\theta_0(t)$ 的误差，计算累积误差得

$$\sqrt{\sum (t(\theta_0(t)) - t)^2} < 10^{-9} \quad (14)$$

结合式 (7)，可知 θ_0 的误差在 10^{-11} 以内。

考虑式 (13)。由于 $|\theta_i - \theta_{i+1}|$ 相对 θ_i 的值很小，近似认为 $\sin(\theta_i - \theta_{i+1}) = \theta_i - \theta_{i+1}$, $\cos(\theta_i - \theta_{i+1}) = 1 - (\theta_i - \theta_{i+1})^2$ ，于是 $\frac{d\theta_{i+1}}{d\theta_i} \approx 1$ 。所以迭代运算的方法误差是线性累积的。事实上，通过敏感性分析我们可以发现，由 θ_0 小范围波动引起的 θ_{224} 的相对误差甚至可能小于 θ_0 的相对误差。总共迭代了 224 次，所以我们控制每次迭代求解的计算精度达到 10^{-9} 即可。

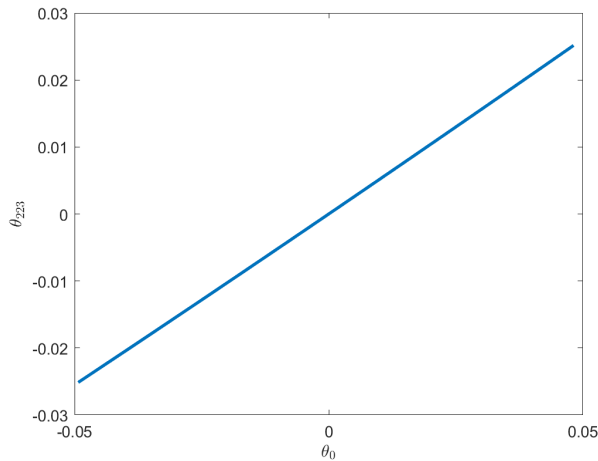


图 4 θ_{224} 在 θ_0 不太小时发生 5% 波动的相对误差

5.2.5 结果呈现和模型验证

下面本文给出 0 s、60 s、120 s、180 s、240 s、300 s 时，龙头前把手、龙头后面第 1、51、101、151、201 节龙身前把手和龙尾后把手的位置和速度。

表 1 问题一部分时间点部分把手的速度

	0s	60s	120s	180s	240s	300s
龙头 (m/s)	1	1	1	1	1	1
第 1 节龙身 (m/s)	0.999971	0.999961	0.999945	0.999917	0.999859	0.999709
第 51 节龙身 (m/s)	0.999575	0.999662	0.999538	0.999331	0.998941	0.998065
第 101 节龙身 (m/s)	0.999575	0.999453	0.999269	0.998971	0.998435	0.997302
第 151 节龙身 (m/s)	0.999448	0.999299	0.999078	0.998727	0.998115	0.996861
第 201 节龙身 (m/s)	0.999348	0.999180	0.998935	0.998551	0.997894	0.996574
龙尾 (后) (m/s)	0.999311	0.999136	0.998883	0.998489	0.997816	0.996478

表 2 问题一部分时间点部分把手的位置 (单位: m)

	0s	60s	120s	180s	240s	300s
龙头 x	8.800000	5.799209	-4.084887	-2.963609	2.594494	4.420274
龙头 y	0.000000	-5.771092	-6.304479	6.094780	-5.356743	2.320429
第 1 节龙身 x	8.363824	7.456758	-1.445473	-5.237118	4.821221	2.459489
第 1 节龙身 y	2.826544	-3.440399	-7.405883	4.359627	-3.561949	4.402476
第 51 节龙身 x	-9.518732	-8.686317	-5.543150	2.890455	5.980011	-6.301346
第 51 节龙身 y	1.341137	2.540108	6.377946	7.249289	-3.827758	0.465829
第 101 节龙身 x	2.913983	5.687116	5.361939	1.898794	-4.917371	-6.237722
第 101 节龙身 y	-9.918311	-8.001384	-7.557638	-8.471614	-6.379874	3.936008
第 151 节龙身 x	10.861726	6.682311	2.388757	1.005154	2.965378	7.040740
第 151 节龙身 y	1.828753	8.134544	9.727411	9.424751	8.399721	4.393013
第 201 节龙身 x	4.555102	-6.619664	-10.627211	-9.287720	-7.457151	-7.458662
第 201 节龙身 y	10.725118	9.025570	1.359847	-4.246673	-6.180726	-5.263384
龙尾 (后) x	-5.305444	7.364557	10.974348	7.383896	3.241051	1.785033
龙尾 (后) y	-10.676584	-8.797992	0.843473	7.492370	9.469336	9.301164

图 3 显示各把手速度与时间的关系。容易发现，除龙头把手以外，所有的把手速度随时间逐渐变慢，且越靠后的把手运动越慢。这与模型中 $\frac{dL_i}{d\theta_i} > 0, \frac{d\theta_0}{dt} < 0$ 是一致的。

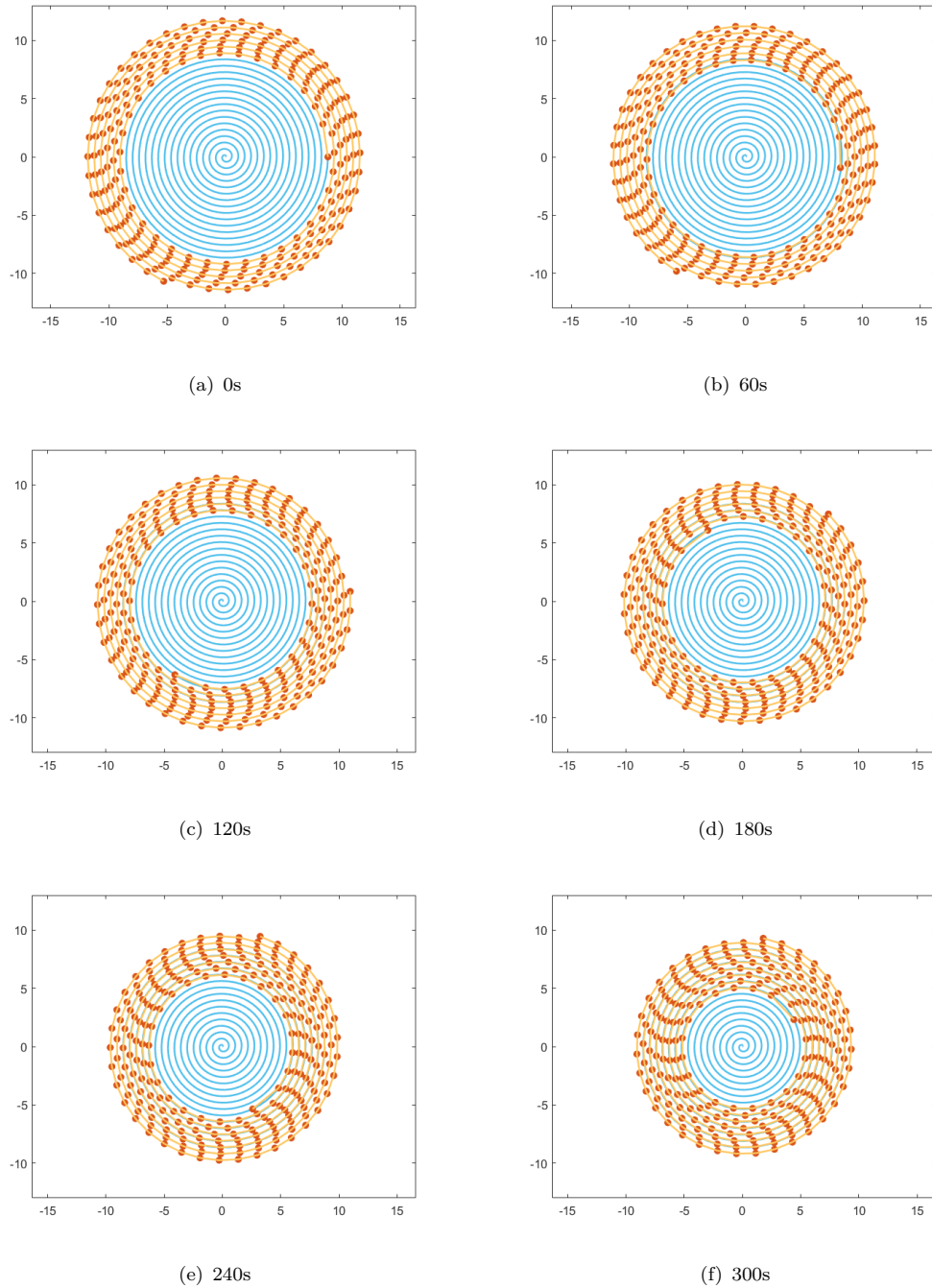


图 5 不同时间下“板凳龙”位置示意图

5.3 问题二：给定盘入轨迹、龙头速度的碰撞前最长运动时间确定

5.3.1 碰撞位置推理

碰撞发生在内圈板凳外边沿和外圈板凳内边沿之间，只有当龙头非常靠近螺线中心时才会存在内圈板凳内侧边沿与外圈板凳相撞的情况。本文先针对第一种情况进行讨论，可证明该情况下的碰撞，只能发生在龙头的左侧前、后凳角和第一节龙身的左侧前、后凳角与外圈板凳之间，即 A_0, A_1, B_0, B_1 四个位置中的一个。

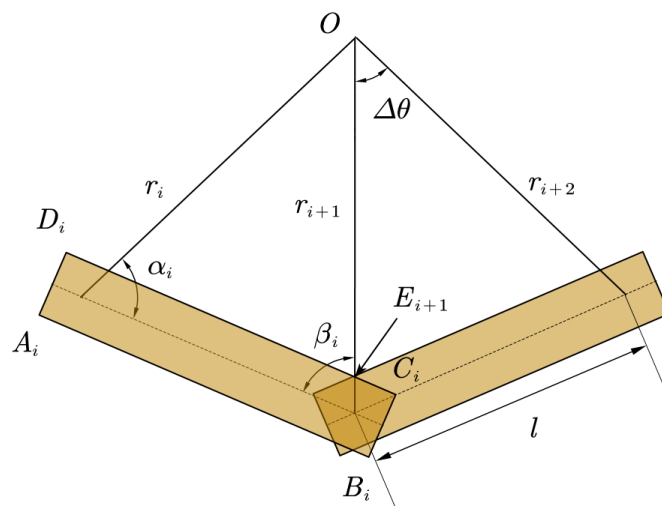


图 6 板凳中轴与两把与螺线中心连线的关系示意图

证明. 首先证明碰撞的内圈位置是板凳的左侧顶点。第一，内圈左侧可看作一个凸多边形外部的一部分，外圈可看作一个凸多边形内部的一部分。假设内圈非顶点的位置发生了碰撞，那么外圈发生碰撞处在顶点，产生矛盾。第二，根据阿基米德螺线沿 θ 减小方向的渐进性，龙头的 $A_0 D_0$ 边上的极径必然是由 A_0 向 D_0 减小的。故 D_0 点不会发生碰撞。

再证明可能发生碰撞的内圈顶点只有可能是龙头和第一个龙身上 A_0, A_1, B_0, B_1 四个凳角。已知舞龙队的运动在第一次碰撞后立即停止，且由问题一可知，若任意把手的极角确定，整个舞龙队各节的位置和方向即完全确定。因此不妨设第一次碰撞发生在其他板凳的左侧顶点上，假设在 t 时刻发生在内圈的 $A_i (i > 1)$ 和外圈的第 j 节龙身之间 ($j > i$)，此时对应的第 i 节龙身前把手的极角为 θ_i 。由于各把手极角变化的连续性，存在 Δt ，在 $t - \Delta t$ 时第一节龙身的极角也为 θ_i ，此时第 $j - i + 1$ 节龙身的位置和方向与 t 时刻第 j 节龙身完全相同。也就是说，碰撞在 Δt 之前已经发生了，产生矛盾。对龙身板凳的左后凳角同理。 \square

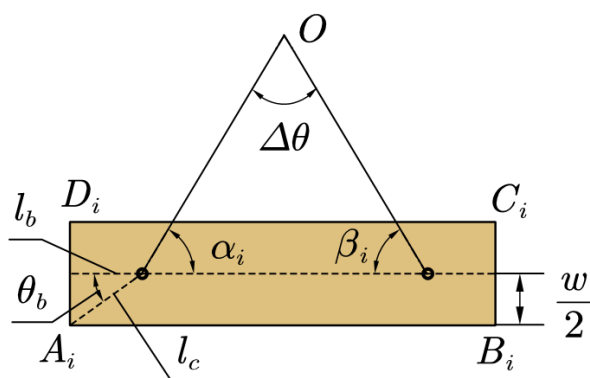


图 7 板凳把手和四个凳角的位置关系示意图

5.3.2 凳角位置和碰撞检测原理

在已知各 θ_i 的情况下, 根据图 7 所示的几何关系, 我们可以给出每张板凳四凳角的坐标。由正弦定理,

$$\begin{cases} \frac{r_0}{\sin \beta_0} = \frac{r_1}{\sin \alpha_0} = \frac{l_h}{\sin(\theta_1 - \theta_0)} \\ \frac{r_i}{\sin \beta_i} = \frac{r_{i+1}}{\sin \alpha_i} = \frac{l}{\sin(\theta_{i+1} - \theta_i)}, i \geq 1 \end{cases} \quad (15)$$

据此可以求得板凳中轴与两个把手与螺线中心连线夹角 α_i, β_i , 从而

$$A_i : \begin{cases} x = r_i \cos \theta_i + l_c \cos(\theta_i - \alpha_i + \theta_b) \\ y = r_i \sin \theta_i + l_c \sin(\theta_i - \alpha_i + \theta_b) \end{cases} \quad (16)$$

$$B_i : \begin{cases} x = r_{i+1} \cos \theta_{i+1} + l_c \cos(\theta_{i+1} + \beta_i - \theta_b) \\ y = r_{i+1} \sin \theta_{i+1} + l_c \sin(\theta_{i+1} + \beta_i - \theta_b) \end{cases} \quad (17)$$

$$C_i : \begin{cases} x = r_{i+1} \cos \theta_{i+1} + l_c \cos(\theta_{i+1} + \beta_i + \theta_b) \\ y = r_{i+1} \sin \theta_{i+1} + l_c \sin(\theta_{i+1} + \beta_i + \theta_b) \end{cases} \quad (18)$$

$$D_i : \begin{cases} x = r_i \cos \theta_i + l_c \cos(\theta_i - \alpha_i - \theta_b) \\ y = r_i \sin \theta_i + l_c \sin(\theta_i - \alpha_i - \theta_b) \end{cases} \quad (19)$$

在不碰撞的情况下, A_0, A_1, B_0, B_1 四点应当始终与原点在一列直线 $D_{i_0}C_{i_0}, D_{i_0+1}C_{i_0+1}, \dots$ 的同侧, 我们假设当 $i < i_0$ 时, 可以保证四点不与对应的内边沿发生碰撞。由于原点总是在射线 D_iC_i 的左侧, $\det(\overrightarrow{D_iO}, \overrightarrow{D_iC_i}) < 0$ 。因此我们要保证 $\det(\overrightarrow{D_iX}, \overrightarrow{D_iC_i}) < 0, X = A_0, A_1, B_0, B_1$, 如图 8 所示。

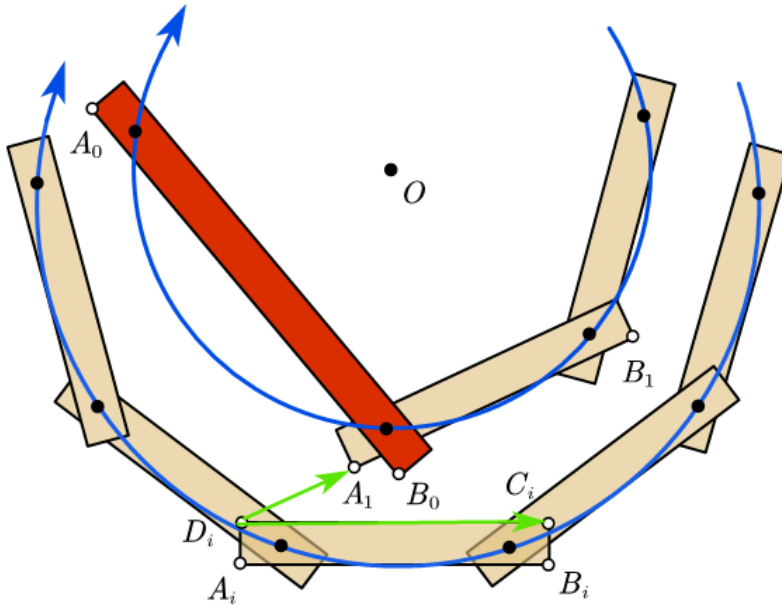


图 8 碰撞检测原理示意图

各 θ_i 均为关于 θ_0 的函数，而

$$\det(\overrightarrow{D_i X}, \overrightarrow{D_i C_i}) = (x_X - x_{D_i})(y_{C_i} - y_{D_i}) - (x_{C_i} - x_{D_i})(y_X - y_{D_i}) \quad (20)$$

是关于 $\theta_0, \theta_i, \theta_{i+1}$ 的函数。因此 $\det(\overrightarrow{D_i X}, \overrightarrow{D_i C_i})$ 是一系列关于 θ_0 的函数。我们要求发生碰撞时 θ_0 的值，即求这若干个函数中 θ_0 最大的零点 θ_0^* 。

反过来，我们记检验函数

$$f_E(\theta_0) = \max\{\det(\overrightarrow{D_i X}, \overrightarrow{D_i C_i}) \mid X \in \{A_0, A_1, B_0, B_1\}, i \in I\} \quad (21)$$

其中 I 是一个合适的指标集。考虑到碰撞发生在内外圈之间，相关板凳把手的极角相差约 2π ，我们取关于 A_0 的 $I = \{i \mid \theta_i \in (\theta_0 + \pi, \theta_0 + 3\pi)\}$ ，关于 A_1, B_0 和 B_1 的 I 中 θ_0 相应地分别换成 θ_1, θ_2 。于是有

$$\theta_0^* = \max\{\theta_0 \mid f_E(\theta_0) = 0\} \quad (22)$$

即所求 θ_0^* 是 θ_0 从大到小变化的第一个过零点。

根据问题一中求出的 t 与 θ_0 的关系，我们可以算得 $t^* = t(\theta_0^*)$ ，即为终止时刻。

5.3.3 模型求解

通过割线搜索法得到 $\theta_0^* = 26.1465$ ，相应的 $t^* = 412.4738\text{s}$ 。

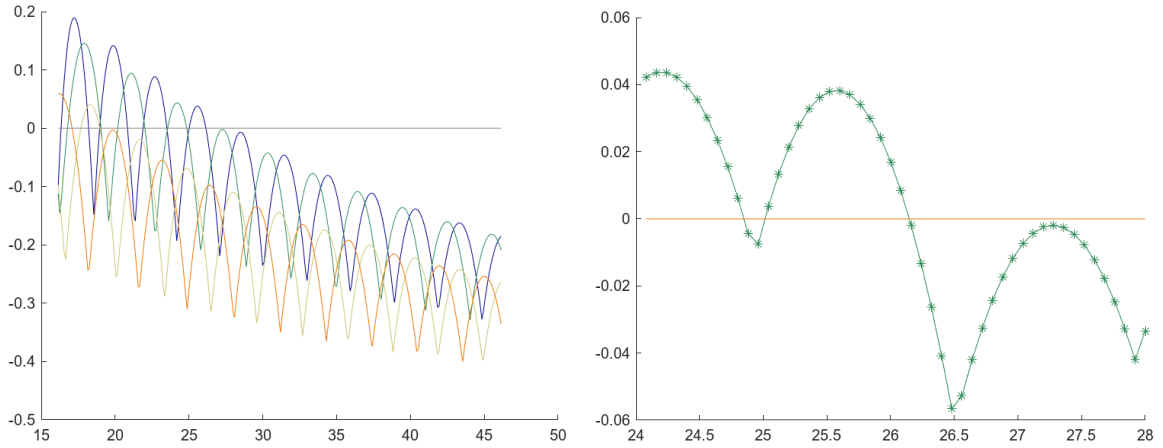


图 9 t 的检验函数在不同尺度下的图像

特别说明：我们求解的是理论值，即碰撞距离阈值为 0。在实际情况中，可能为了安全，需要选取一个大于零的阈值。例如在本情境中，最大零点的前一个峰值的绝对值不大于 0.01，其实已经是算是“碰撞”到了。

表 3 问题二中最大终止时间时刻部分把手的位置和速度

	横坐标 x (m)	纵坐标 y(m)	速度 (m/s)
龙头	1.209931	1.942784	1.000000
第 1 节龙身	-1.643792	1.753399	0.944301
第 51 节龙身	1.281201	4.326588	0.969868
第 101 节龙身	-0.536246	-5.880138	0.970493
第 151 节龙身	0.968840	-6.957479	0.970748
第 201 节龙身	-7.893161	-1.230764	0.970888
龙尾 (后)	0.956217	8.322736	0.970931

5.4 问题三：保证龙头把手能够在发生碰撞前进入掉头空间的最小螺距确定

5.4.1 模型重述

问题三和问题二具有类似的解题思路，只是可变量从时间变成了螺距。在问题二中，我们已经论证了碰撞只能发生在龙头的左侧前、后凳角和第一节龙身的左侧前、后凳角与外圈板凳之间，并且求出了各板凳四凳角坐标的形式。

在问题二中，我们巧妙地利用待测点和原点在一列直线同侧等价于不碰撞这一特性，通过 $\det(\overrightarrow{D_i X}, \overrightarrow{D_i C_i})$ 来判断是否与对应直线碰撞，并求所有这样的式子的最大值来作为检验函数，检验函数在不碰撞时为零，过零则说明发生碰撞。

我们继续使用这一策略，根据问题一中的等距螺线方程（式 1），当极径已知时，有

$$\theta(d) = \frac{2\pi r}{d} \quad (23)$$

将其与问题二中的检验函数 $f_E(\theta_0)$ ，我们可以得到当龙头前把手的极径已知时，关于螺距 d 的检验函数 $f_{Ed}(d) = f_E(\theta(d))$ 。

5.4.2 模型求解

使用割线搜索法求得最小螺距为 0.417138m。

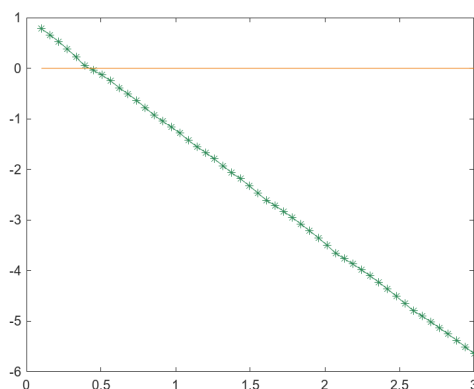


图 10 f_{Ed} 和 d 的曲线图

5.4.3 附加分析

在求解的过程中，我们注意到碰撞检验量关于螺距 d ，呈现线性关系。这启示我们发现系统的一些特性。这对于系统上一些优化问题的分析有巨大帮助。

当给定不太小的 r_i 时，我们根据式 (2) 绘制 r_{i+1} 关于 d 的方程，发现当 d 较大时， r 几乎不发生变化。相反，在 d 很小时，系统出现了“混沌锥”。此时迭代三角形的另两边都确定，而由于 r_{i+1} 变化很小，整个三角形的形状近似确定。以此类推，检验量（其几何意义除碰撞判定外，还有两向量之间的平行四边形面积）受角度影响较小，主要受螺距影响。在分析模型时，我们可以将迭代极径或极角的变化视为螺距变化的小尺度量。

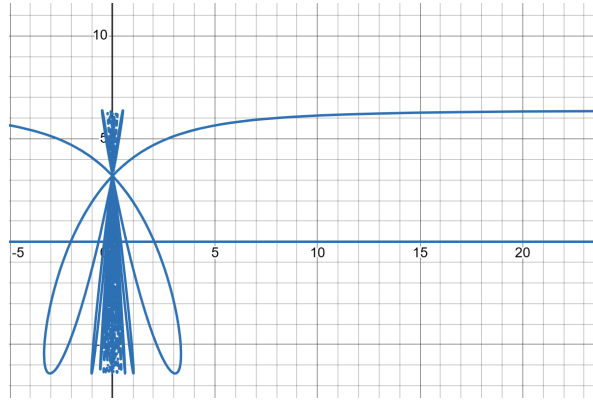


图 11 r_{i+1} 关于 d 的曲线图

5.5 问题四：考虑调头返回的运动状态确定

5.5.1 调头路径的确定及关于碰撞和后退的讨论

在本问中，舞龙队需在调头空间内以由两段圆弧组成的 S 形曲线路径完成调头。由题意，调头路径应满足以下四个条件：

- (i) 使盘出螺线与盘入螺线关于螺线中心呈中心对称；
- (ii) 调头路径与盘入、盘出螺线均相切，调头路径的两段圆弧相切；
- (iii) 在调头空间的边界由螺线切入圆弧及由圆弧切出螺线；
- (iv) 前一段圆弧的半径是后一段的 2 倍。

下面我们将证明，在给定的条件下，调头路径唯一确定，且当去掉第四个条件，即两段圆弧的半径比可以控制时，调头路径的长度仍然一定。

证明. 由

$$\frac{d\theta}{2\pi} = R^2 \quad (24)$$

其中 $R = 4.5\text{m}$ 是调头空间的半径，解得由螺线切入圆弧的切点极角为 $\theta_{\text{in}} = 16.6320$ 。由盘入、盘出螺线关于螺线中心呈中心对称，由圆弧切出螺线的切点极角为 $\theta_{\text{in}} = 16.6320 - \pi = 13.4904$ 。

如图 13 所示，盘入、盘出螺线与调头路径的交点 E_1 和 E_2 确定，调头曲线在交点的切线是 l_1 和 l_2 。过 E_1 作直线垂直于 l_2 ，垂足为 F_1 ，同理作出 F_2 。因为等距螺线形状大小固定，所以 $\overline{E_1E_2}$ 与 $\overline{E_1F_1}$ 的夹角 γ 也确定。为满足三个相切的要求，最后做出的调头路径曲线大致为黑线，设黑线是满足四个条件的掉头曲线，蓝线是两段圆弧半径比异于黑线两段圆弧半径比的另一条调头曲线，其满足两段圆弧相切连接并与盘入、盘出螺线均相切。设黑色路线长 S_0 ，蓝色路线长 S'_0 。则有：

$$S_0 = (r_1 + r_2) * (\pi - 2 * \gamma) \quad (25)$$

蓝色路径的长度：

$$S'_0 = (r'_1 + r'_2) * (\pi - 2 * \gamma) \quad (26)$$

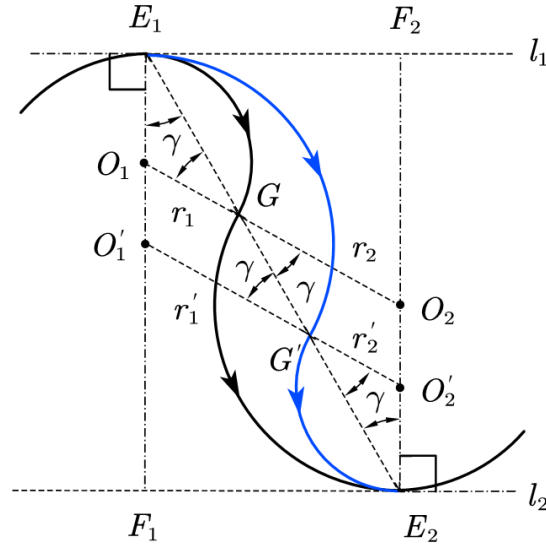


图 12 掉头路径恒定的证明示意图

由于 $\overline{O_1O_2}$ 平行于 $\overline{O'_1O'_2}$ ，且 $\overline{E_1F_1}$ 平行于 $\overline{E_2F_2}$ ，因此 $\overline{O_1O_2} = \overline{O'_1O'_2}$ ，即 $r'_1 + r'_2 = r_1 + r_2$ ， $S_0 = S'_0$ 。说明 E_1 和 E_2 确定时，内部调头路径长度与形状无关。 \square

根据盘入/出点与螺线的切点坐标极角，可知切线法线斜率

$$k_{\perp} = \frac{\theta_{\text{in}} \cos \theta_{\text{in}} + \sin \theta_{\text{in}}}{-\theta_{\text{in}} \sin \theta_{\text{in}} + \cos \theta_{\text{in}}} \quad (27)$$

从而可求出

$$\gamma = \theta_{\text{in}} - \tan^{-1}(k_{\perp}) - \pi/2 \quad (28)$$

根据半径比为 $2:1$ ，可知两圆弧切点坐标 $(\theta_{\text{in}} - \pi, r_{\text{in}}/3)$ 。联立切线与法线，即得两圆弧圆心坐标（如图所示），半径分别为 3.0054m 和 1.5027m 。

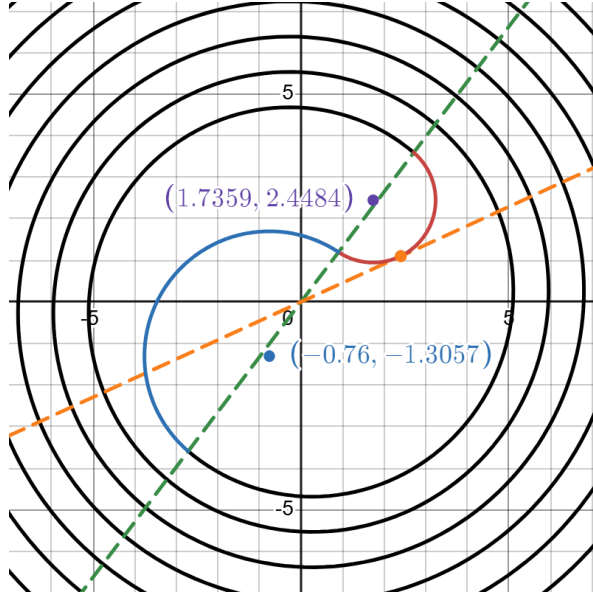


图 13 调头路径示意图

因为需要调头，考虑螺距的一半，即 0.85m。该值大于 $2l_c$ 。同时，任意板凳不可能跨过某段弧，且两弧均劣，故不存在某板凳前把手前进而后把手后退的情况。综合两天，不会产生碰撞。

5.5.2 自然极角的构造

在本问中，由于涉及掉头，舞龙队的运动路径不能同一个极坐标方程来表示。特别是在极坐标系中，把手经过过极点的第二段圆弧的切点（我们称为“回转点”，并记其极角为 θ_{reverse} ，其值为 12.9961）前后，其极角先增大后减小，不能直接地将极角与极径一一对应起来。因此，我们希望设置一个与整条路径上的点一一对应的变量 θ_{nature} （称为“自然极角”），这个变量在全过程连续变化，并且尽可能地方便用来转换到点的真实极角。

我们设整个运动路径关于这个变量的方程为

$$r = r(\theta_{\text{nature}}) \quad (29)$$

下面我们来构建这个自然极角 θ_{nature} 和它与对应点的实际位置 $(\theta_{\text{real}}, r)$ 的关系。

注意到在回转点之前，路径方程可以以连续递减极角作为自变量来表示 θ ，而在回转点后极角又转为递增，所以我们的目标有二：一是建立回转点之前的圆弧方程时，使极角可以沿路径连续地递减，此时回转点之前（包括螺线部分）的自然极角与真实极角 θ_{real} 是一致的；二是在回转点之后继续连续地减小自然极角，并且保证真实极角是对应自然极角的分段线性函数。这样，整条路径上各点的极径就完全地表示为一个关于自然极角 θ_{nature} 的连续分段函数。在行进的过程中，自然极角是连续严格单调递减的。

首先, 为圆的极坐标方程选取合适的自变量范围。设某圆的圆心极坐标为 (θ_c, r_c) , 半径为 R , 根据余弦定理, 其隐式的极坐标方程为

$$r^2 - 2rr_c \cos(\theta - \theta_c) + r_c^2 = R^2 \quad (30)$$

转化为显示形式即为

$$r = r_c \cos(\theta - \theta_c) \pm \sqrt{r_c^2 \cos^2(\theta - \theta_c) - r_c^2 + R^2} \quad (31)$$

其中取正或负号均可以作出完整图形, 可以依据具体情况选定。

我们将掉头曲线以两段圆弧的切点和回转点为界分为三段, 分别记为 C_1, C_2, C_3 , 圆心的位置及半径已求出。为了自变量范围选取的方便, 它们的极坐标方程中正负号分别取正、负、正。

为了降低计算难度, 我们希望 C_1, C_2, C_3 的极坐标方程的自变量范围分别为 $[\theta_{\text{in}}, \theta_{\text{con}}], [\theta_{\text{con}}, \theta_{\text{reverse}}], [\theta_{\text{reverse}}, \theta_{\text{out}}]$ 。其中 θ_{con} 是大小圆弧的切点极角, 根据前文的论证, 它和 θ_{out} 是相同的。我们选取的具体 θ_{reverse} 应当满足 $\theta_{\text{reverse}} \in [\theta_{\text{in}}, \theta_{\text{in}} - 2\pi]$ 的值, 即 $\theta_{\text{reverse}} = 12.9960$ 。

接下来我们建立自然极角和真实极角的关系。在回转点处, 真实极角达到最小值。根据我们的需求, 当 $\theta_{\text{nature}} \geq \theta_{\text{reverse}}$ 时, $\theta_{\text{nature}} = \theta_{\text{real}}$; 当 $\theta_{\text{nature}} < \theta_{\text{reverse}}$ 时, $\theta_{\text{nature}} = \theta_{\text{reverse}} - (\theta_{\text{real}} - \theta_{\text{reverse}})$ 。即

$$\begin{aligned} \theta_{\text{real}} &= \begin{cases} \theta_{\text{nature}}, & \theta_{\text{nature}} \geq \theta_{\text{reverse}} \\ 2\theta_{\text{reverse}} - \theta_{\text{nature}}, & \theta_{\text{nature}} < \theta_{\text{reverse}} \end{cases} \\ &= \theta_{\text{reverse}} + |\theta_{\text{nature}} - \theta_{\text{reverse}}| \end{aligned} \quad (32)$$

将各段曲线上的真实极角转化为自然极角 (在每一段上, 真实极角和自然极角是一一对应的) 带入相应的极坐标方程, 即可得到整条路径曲线与自然极角 θ_{nature} 的关系 (为了阅读方便, 下式中用 θ 代替 θ_{nature}):

$$r(\theta) = \begin{cases} \frac{d\theta}{2\pi}, & \theta \geq \theta_{\text{in}} \\ r_{\text{cin}} \cos(\theta - \theta_{\text{cin}}) + \sqrt{r_{\text{cin}}^2 \cos^2(\theta - \theta_{\text{cin}}) - r_{\text{cin}}^2 + R_1^2}, & \theta_{\text{cin}} > \theta \geq \theta_{\text{con}} \\ r_{\text{cin}} \cos(\theta - \theta_{\text{cin}}) - \sqrt{r_{\text{cin}}^2 \cos^2(\theta - \theta_{\text{cin}}) - r_{\text{cin}}^2 + R_1^2}, & \theta_{\text{con}} > \theta \geq \theta_{\text{reverse}} \\ \frac{r_{\text{cout}} \cos(\theta_{\text{reverse}} + |\theta - \theta_{\text{reverse}}| - \theta_{\text{cout}}) + \sqrt{r_{\text{cout}}^2 \cos^2(\theta_{\text{reverse}} + |\theta - \theta_{\text{reverse}}| - \theta_{\text{cout}}) - r_{\text{cout}}^2 + R_2^2}}{d(\theta_{\text{reverse}} + |\theta - \theta_{\text{reverse}}| + \pi)}, & \theta_{\text{reverse}} > \theta \geq \theta_{\text{out}} \\ \frac{d(\theta_{\text{reverse}} + |\theta - \theta_{\text{reverse}}| + \pi)}{2\pi}, & \theta < \theta_{\text{out}} \end{cases} \quad (33)$$

5.5.3 运动状态的求解

根据问题一中给出的式 (11), 我们需要仿照问题一中的方法, 在更复杂的情况下求出 $\frac{dL_i}{d\theta_i}$, $\frac{d\theta_{i+1}}{d\theta_i}$ 和 $\frac{d\theta_0}{dt}$ 。

与问题一类似地，我们需要找到 θ_i, θ_{i+1} 的关系，然后通过迭代，就能在给定龙头把手位置 θ_0 的条件下求出所有把手的位置。只是这里各 θ_i 指的都是自然极角。与问题一同样地我们有

$$\theta_{i+1} = \min \left\{ \theta > \theta_i \mid \text{dis}(\theta_i, \theta) = \begin{cases} l_h, & i = 0 \\ l, & i \geq 1 \end{cases} \right\} \quad (34)$$

其中 $\text{dis}(\theta_i, \theta_j)$ 表示自然极角 θ_i, θ_j 所对应的路径上的点的距离：

$$\begin{aligned} \text{dis}(\theta_i, \theta_j) &= d[(\theta_{\text{real}}(\theta_i), r(\theta_i)), (\theta_{\text{real}}(\theta_j), r(\theta_j))] \\ &= \sqrt{[(r(\theta_i) \cos \theta_{\text{real}}(\theta_i) - r(\theta_j) \cos \theta_{\text{real}}(\theta_j))^2 + [(r(\theta_i) \sin \theta_{\text{real}}(\theta_i) - r(\theta_j) \sin \theta_{\text{real}}(\theta_j))^2]} \end{aligned} \quad (35)$$

在问题一中，式 (13) 已经给出了盘入螺旋线段的 $\frac{dL_i}{d\theta_i}$ ，我们还要求出其他几段的 $\frac{dL_i}{d\theta_i}$ （各自变量都是真实极角 θ_{real} ）。其中第一段圆弧为：

$$\frac{dL_i}{d\theta_i} = \sqrt{r(\theta_i)^2 + \left(-r_{\text{cin}} \sin(\theta_i - \theta_{\text{cin}}) - \frac{r_{\text{cin}}^2 \cos(\theta_i - \theta_{\text{cin}}) \sin(\theta_i - \theta_{\text{cin}})}{\sqrt{r_{\text{cin}}^2 \cos^2(\theta_i - \theta_{\text{cin}}) - r_{\text{cin}}^2 + R_1^2}} \right)^2} \quad (36)$$

第二段圆弧的第一部分为：

$$\frac{dL_i}{d\theta_i} = \sqrt{r(\theta_i)^2 + \left(-r_{\text{cin}} \sin(\theta_i - \theta_{\text{cin}}) + \frac{r_{\text{cin}}^2 \cos(\theta_i - \theta_{\text{cin}}) \sin(\theta_i - \theta_{\text{cin}})}{\sqrt{r_{\text{cin}}^2 \cos^2(\theta_i - \theta_{\text{cin}}) - r_{\text{cin}}^2 + R_1^2}} \right)^2} \quad (37)$$

第二段圆弧的第二部分为：

$$\frac{dL_i}{d\theta_i} = \sqrt{r(\theta_i)^2 + \left(-r_{\text{cout}} \sin(\theta_i - \theta_{\text{cout}}) - \frac{r_{\text{cout}}^2 \cos(\theta_i - \theta_{\text{cout}}) \sin(\theta_i - \theta_{\text{cout}})}{\sqrt{r_{\text{cout}}^2 \cos^2(\theta_i - \theta_{\text{cout}}) - r_{\text{cout}}^2 + R_2^2}} \right)^2} \quad (38)$$

盘出螺旋线段为：

$$\frac{dL_i}{d\theta_i} = \frac{d\sqrt{(\theta_i + 1)^2 + 1}}{2\pi} \quad (39)$$

对于对应的 θ_{nature} 区域，将 $\theta_{\text{real}}(\theta_{\text{nature}})$ 代入，即得关于 θ_{nature} 的弧微分分段函数。

接着求 $\frac{d\theta_{i+1}}{d\theta_i}$ 。除了一张板凳可能跨越第二段圆弧的第一部分和第二段圆弧的第二部分外，不存在一张板凳同时处于两段路径上的情况。所以 θ_{i+1} 是关于 θ_i 的十一段分段函数。对每一段，我们分别对相应的隐函数求导，即得 $\frac{d\theta_{i+1}}{d\theta_i}$ 。结果较为复杂，可通过符号计算求出，在此略去。

最后是求 $\frac{d\theta_0}{dt}$ 。联立式 (5)，以及式 (6) 和式 (37)~(39) 共同组成的分段函数，即可得到形式为

$$\frac{d\theta_0}{dt} = f(\theta_0) \quad (40)$$

的微分方程，其中 $f(\theta_0)$ 是分段函数。

特别地，注意到 $f(\theta_0)$ 含有瑕点，这是由于在回转点附近， θ_{nature} 关于弧长的变化率趋近于零。所以我们考虑以回转点为界将微分方程分段处理。

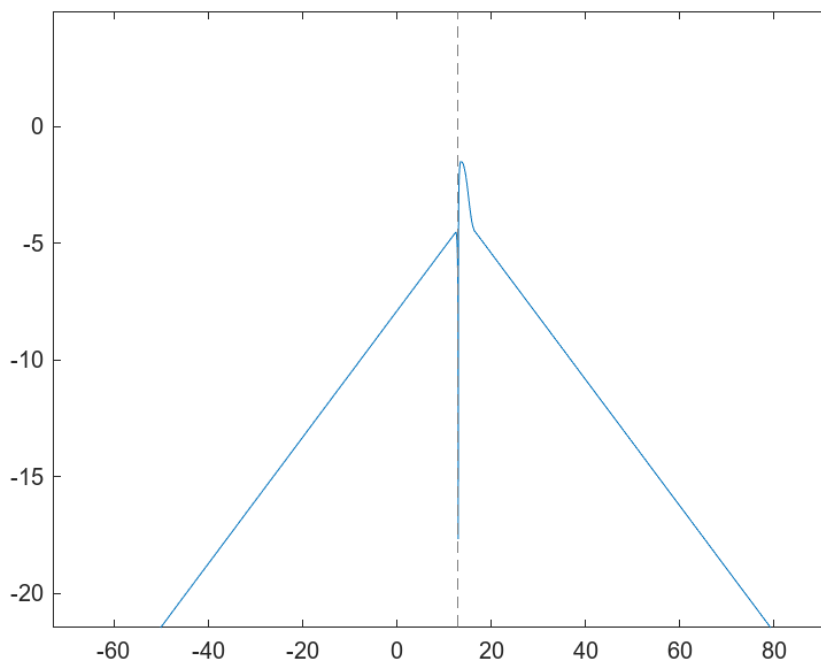


图 14 $\frac{dL}{d\theta_0}$ 和 θ_{nature} 的关系

综上，我们就可以求得各把手速度 v_i 与时间 t 的关系了。

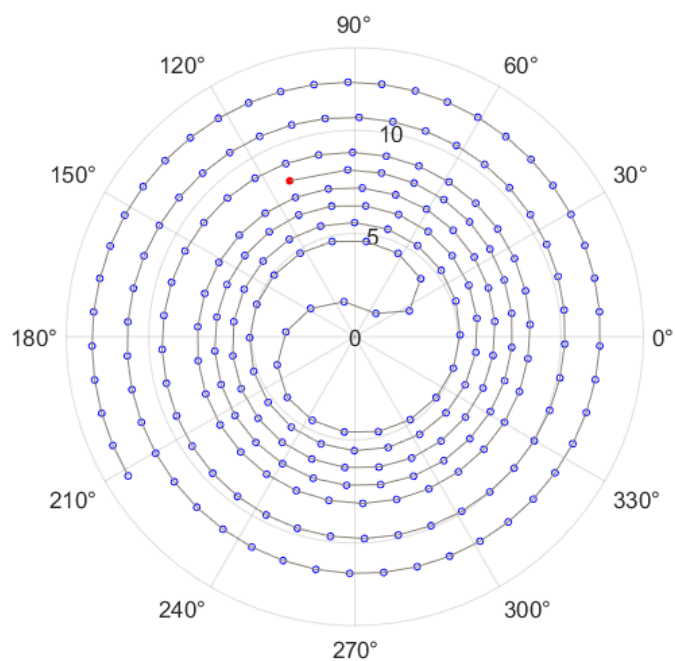


图 15 包含调头运动的示意图（动画见支撑材料）

表 4 问题四部分时间点部分把手的位置 (单位: m)

	-100s	-50s	0s	50s	100s
龙头 x	2.516423	2.137980	-0.877365	0.444609	-0.043320
龙头 y	1.202612	0.614339	-1.161819	-0.242262	0.086738
第 1 节龙身 x	2.008882	1.736354	-0.020555	0.546276	-0.048119
第 1 节龙身 y	1.976286	1.447925	-1.511170	-0.020784	0.045222
第 51 节龙身 x	-3.432012	-1.174394	0.795870	-0.541847	1.050620
第 51 节龙身 y	0.916071	-2.900053	-2.516459	-1.965640	-0.248697
第 101 节龙身 x	-3.857364	3.275990	0.973336	-2.455495	-2.355007
第 101 节龙身 y	-1.553711	-1.932197	3.270410	1.675360	0.677361
第 151 节龙身 x	-4.642981	4.197724	-2.265608	-1.490954	3.057587
第 151 节龙身 y	-0.640910	-1.232763	3.344480	-3.360002	-1.154785
第 201 节龙身 x	-3.867128	3.404341	-2.223566	0.107856	2.764905
第 201 节龙身 y	3.418735	-3.496520	4.006138	-4.263350	2.777308
龙尾 (后) x	-0.327107	0.061409	-0.625585	1.896630	-3.557521
龙尾 (后) y	-5.347156	5.086072	-4.760130	4.080134	-2.181568

表 5 问题四要求时间点把手的速度

	-100s	-50s	0s	50s	100s
龙头 (m/s)	1.000000	1.000000	1.000000	1.000000	1.000000
第 1 节龙身 (m/s)	0.999904	0.999762	0.998687	1.000363	1.000124
第 51 节龙身 (m/s)	0.999346	0.998642	0.995134	0.950360	1.003966
第 101 节龙身 (m/s)	0.999091	0.998248	0.994448	0.948906	1.100500
第 151 节龙身 (m/s)	0.998944	0.998047	0.994156	0.948462	1.099539
第 201 节龙身 (m/s)	0.998849	0.997925	0.993994	0.948247	1.099165
龙尾 (后) (m/s)	0.998817	0.997885	0.993944	0.948184	1.099064

5.6 问题五：龙头把手速度最大值求解

5.6.1 速度关系与约束条件

首先确定各把手速度与龙头行进速度的关系。由式 (11), 及 $v_h = \frac{dL_h}{d\theta_0} \frac{d\theta_0}{dt}$ 可得:

$$v_i = \frac{dL_i}{d\theta_i} \frac{d\theta_i}{d\theta_{i-1}} \cdots \frac{d\theta_1}{d\theta_0} \frac{v_h}{\frac{dL_0}{d\theta_0}} \quad (41)$$

其中 $\frac{dL_i}{d\theta_i}$ 是第四问中的关于自然坐标 θ_{nature} 的分段函数, 各 θ_i 都是自然坐标。因为等式右边的各项除 v_h 以外均仅为 $\theta_i, i = 0, 1, 2, \dots, 224$ 的函数, 因此在所有时刻下, $v_i, i = 1, 2, \dots, 224$ 均与 v_h 成正比 (但不同时刻比值不同)。可见, v_h 的变化不会改变某一时刻各把手速度的相对大小。又由于 $\theta_i, i = 1, 2, \dots, 224$ 均可由 θ_0 求解得到,

因此 v_i 与 v_h 之间的比值关系是 θ_0 的函数, 有 $v_i = f_{v_i}(\theta_0)v_h$, 其中 f_{v_i} 为对应把手速度与 v_h 的比值。

其次确定使得舞龙队各把手的速度均不超过 2 m/s 这一条件对 v_h 的限制。这一条件可以表达为:

$$\max_{\theta} \left(\max_i (v_i(\theta_0)) \right) \leq 2 \quad (42)$$

也就是:

$$\max_{\theta} \left(\max_i (f_{v_i}(\theta_0)) \right) \leq \frac{2}{v_h} \quad (43)$$

其中 $i = 0, 1, 2, \dots, 224$ 。因此, v_h 的最大值为:

$$v_h^* = \frac{2}{\max_{\theta} \left(\max_i (f_v(\theta_0)) \right)} \quad (44)$$

5.6.2 极值搜索

然后, 为了求解该极值, 我们需要确定 θ_0 的范围。通过前几问的求解和探索, 我们发现, 对于“板凳龙”的把手这样间距固定的点列, 在等距螺线上时, 总是靠外侧的点运动的较慢, 且越靠外点间速度的差值越小。

由此可知, 当“板凳龙”盘入调头区域时(龙头把手未进入掉头区域), 龙身、龙尾的把手速度总是小于龙头的行进速度; 当“板凳龙”盘离调头区域时(龙尾后把手已离开掉头区域), 龙身、龙尾的速度均大于龙头行进速度, 龙尾后把手速度总是最快, 且在即将离开掉头区域时龙尾后把手速度最快。

综上, 我们可以将龙头前把手进入调头区域之前和龙尾后把手离开调头区域的临界状态之后的 θ_0 排除到求解范围之外。也就是说, 所有把手上的最大速度出现在自龙头前把手即将盘入掉头区域到龙尾后把手即将盘出调头区域这段过程中。最后, 通过单纯形法在相应的 θ_0 取值范围中搜索得到, 在 $\theta_0 = 12.316397$ 时, 有 $v_h^* = 1.244612\text{m/s}$ 。

六、模型的评价、改进与推广

6.1 模型优点

1. 分析精确, 充分利用点间距固定的性质提高了运动状态解算的准确性。
2. 判断严谨高效, 利用原点和碰撞判断点的关系作为碰撞条件, 在保持严格的判定条件的同时兼顾了计算效率。
3. 创造性地在求解过程中引入自然极角, 统一了盘入、盘出两种运动地表达方式。

6.2 模型缺点及改进

1. 迭代部分运行时间较长, 可以考虑利用类似问题 3 的方法, 发现一些隐含的近似不变关系, 从而发现模型中小尺度量, 对模型进行近似。

七、参考文献

- [1]. 韦华丽. 北泉板凳龙文化的现状与传承 [D]. 西南大学,2012.
- [2]. 刘崇军. 等距螺旋的原理与计算 [J]. 数学的实践与认识, 2018, 48 (11): 165-174.
- [3]. 敖特根. 单纯形法的产生与发展探析 [J]. 西北大学学报 (自然科学版),2012,42(05):861-864.
- [4]. 刘少辉. 浦江板凳龙考察研究 [J]. 民族艺术,2009,(04):106-108+115

附录

附录 1：支撑材料的文件列表

- [1]. 结果附件- (result1.xlsx, result2.xlsx, result4.xlsx)
- [2]. 第四题动画.mp4
- [3]. 第一题动画.mp4
- [4]. 流程图.pptx
- [5]. 代码- (hualuoxian.m, huitu.m, suanjijing.m, Combinedd.m, CombinedFunction.m, crossfunctions.m, func.m... 等 14 个文件)
- [6]. 2024Atex.zip

附录 2-1：预备代码

```
1      syms r theta_0 d theta_temp v t_q1 theta
2
3  % 计算弧长距离对极角的函数和极角对时间的导数
4  r = d/(2*pi)*theta_temp;
5  f = sqrt(r^2 + diff(r,1,theta_temp));
6  % L = int(f,theta_temp,0,theta);
7  L = d/(2*pi)*(0.5*theta*sqrt(1+theta^2)+0.5*log(theta+sqrt(1+theta^2)));
8  dLdtheta = diff(L,1,theta);
9  dthetadt = - v/ dLdtheta;
10
11 % 获得到中心弧线长度对极角的函数的函数对象从到中心弧线长度对极角的导数对极角的
    函数的函数对象
12
13 L_fun = matlabFunction(L, 'Vars', [theta,d]);
14 dLdtheta_func = matlabFunction(dLdtheta, 'Vars', [theta,d]);
15 % 不同点上间的距离
16 function dis = getdis(theta_i,theta_j,d)
17     r_i = d/(2*pi)*theta_i;
18     r_j = d/(2*pi)*theta_j;
19     dis = sqrt(r_i^2+r_j^2-2*r_i*r_j*cos(theta_i - theta_j));
20 end
21
22 % 螺线上点从前一个点的极角推后一个点的极角
23 function theta_iplus1 = solveTheta_onSprail(theta_i, l, d)
24     % 定义符号变量
25     syms theta_next
26
27     % 定义方程
28     eqn = theta_i^2 + theta_next^2 - 2*theta_i*theta_next*cos(theta_i -
        theta_next) == (4*pi^2*l^2)/d^2;
29
```



```

30 % 设置求解范围
31 range = [theta_i, theta_i+pi]; % 解的范围
32
33 % 使用 vpasolve 在指定范围内求解方程
34 theta_ip1_sol = vpasolve(eqn, theta_next, range);
35
36 % 检查是否有解
37 if isempty(theta_ip1_sol)
38     error('No solution found within the specified range. ');
39 end
40
41
42 theta_ip1_sol = double(theta_ip1_sol);
43 end
44
45
46 % 增加一层封装,
47 % 接受一个矩阵 (纵向为不同情况下的点的极角,
48 % 横向为不同的点, 下标依次增大), 返回值为在矩指定的最后一列后再写入一列
49 function theta_new=getNextPoint(theta, l, d_t, indexOflastPoint)
50     theta_last = theta(:, indexOflastPoint);
51     theta_new_last = zeros(size(theta_last));
52 if isscalar(d_t)
53     d = ones(size(theta_last)) * d_t;
54 elseif ismatrix(d_t)
55     if size(d_t) == size(theta_last)
56         d=d_t;
57     else
58         disp("d与要迭代的情况个数不一致");
59     end
60 else
61     % 错误提示
62     disp('不是标量也不是矩阵');
63 end
64 %parfor i = 1:length(theta_last)
65 %     theta_new_last(i) = solveTheta(theta_last(i), l, d);
66 %end
67 %theta_new =[theta, theta_new_last];
68 %theta_new_last = zeros(size(theta_last));
69 %parfor i = 1:length(theta_last)
70 %     theta_new_last(i) = solveTheta(theta_last(i), l, d);
71 %end
72 theta_new_ = theta;
73 n = length(theta_new_last);
74 parfor i = 1:n
75     theta_new_last(i) = solveTheta_onSprail(theta_last(i), l, d(i));
76 end
77 theta_new(:, indexOflastPoint+1) = theta_new_last;
78 theta_new = theta_new_;

```

```

79 end
80 %theta_11 = arrayfun(@(theta) solveTheta(theta, 2.86, 0.55), theta_0_val_q1);
81 %dis =arrayfun(@(theta1,theta2) getdis(theta1,theta2,0.55),theta_0_val_q1,
    theta_11);
82
83 % 再增加一层封装,
84 % 接受一个列向量, (纵向为不同时间下的龙头焦点的极角,
85 % 返回一个矩阵, 第一列为输入的列向量, 其后为龙头后指定个数个点的极角
86 function thetaIncludingAfter=solveAfterPoint(theta_0,d,howManyPointsAfterHead,
    l,l_h)
87     [n,~] = size(theta_0);
88     thetas = zeros(n ,1+howManyPointsAfterHead);
89     %disp(thetas)
90     thetas(:,1)=theta_0;
91     thetas = getNextPoint(thetas,l_h,d,1);
92     for i=2:howManyPointsAfterHead
93         thetas = getNextPoint(thetas,l,d, i);
94
95     end
96     thetaIncludingAfter = thetas;
97 end
98
99 % 不从龙头开始计算的版本
100 function thetaIncludingAfter=solveFollowingPoint(theta_0,d,
    howManyPointsAfterHead,l)
101     [n,~] = size(theta_0);
102     thetas = zeros(n ,1+howManyPointsAfterHead);
103     %disp(thetas)
104     thetas(:,1)=theta_0;
105     for i=1:howManyPointsAfterHead
106         thetas = getNextPoint(thetas,l,d, i);
107
108     end
109     thetaIncludingAfter = thetas;
110 end
111
112 % 在不同曲线上点从前一个点的极角推后一个点的极角
113 function theta_iplus1 = solveTheta_arbitrary(theta_i,l,curve,range)
114     % curve 是两个点所在的曲线方程, range是预期的解的范围
115     % curve 都是函数对象, 传入极角返回极径
116     % 定义符号变量
117     ri = curve(theta_i);
118     % 定义方程
119     f = @(theta_next) ri^2 + curve(theta_next)^2 -2*ri*curve(theta_next)*cos(
        theta_i-theta_next)-l^2;
120
121     % 设置求解范围 待处理 (可能弃用)
122     % range = [theta_i,theta_i+pi]; % 解的范围
123

```

```

124 % 使用 vpasolve 在指定范围内求解方程
125 theta_ip1_sol = fzero(f, range);
126
127 % 检查是否有解
128 if isempty(theta_ip1_sol)
129     error('No solution found within the specified range. ');
130 end
131
132 theta_ip1_sol = double(theta_ip1_sol);
133 end
134
135 % 增加一层封装,
136 % 接受一个矩阵 (纵向为不同情况下的点的极角,
137 % 横向为不同的点, 下标依次增大), 返回值为在矩指定的最后一列后再写入一列, 也就
    是不同时间下下一个点
138 % 支持输入曲线方程
139 function theta_new=getNextPoint_arbitrary(theta,l,indexOflastPoint,curve)
140     theta_last = theta(:,indexOflastPoint);
141     theta_new_last = zeros(size(theta_last));
142     % 仅留档
143     %parfor i = 1:length(theta_last)
144     %     theta_new_last(i) = solveTheta(theta_last(i), l, d);
145     %end
146     %theta_new =[theta, theta_new_last];
147     %theta_new_last = zeros(size(theta_last));
148     %parfor i = 1:length(theta_last)
149     %     theta_new_last(i) = solveTheta(theta_last(i), l, d);
150     %end
151     theta_new_ = theta;
152     n = length(theta_new_last);
153
154     for i = 1:n
155         range = [theta_last(i),theta_last(i)+pi];
156         theta_new_last(i) = solveTheta_arbitrary(theta_last(i), l,curve,range)
            ;
157     end
158     theta_new_(:,indexOflastPoint+1) = theta_new_last;
159     theta_new_ =theta_new_;
160 end
161 %theta_11 = arrayfun(@(theta) solveTheta(theta, 2.86, 0.55), theta_0_val_q1);
162 %dis =arrayfun(@(theta1,theta2) getdis(theta1,theta2,0.55),theta_0_val_q1,
    theta_11);
163
164 % 再增加一层封装,
165 % 接受一个列向量, (纵向为不同时间下的龙头焦点的极角,
166 % 返回一个矩阵, 第一列为输入的列向量, 其后为龙头后指定个数个点的极角
167 % 支持输入曲线方程
168 function thetaIncludingAfter=solveAfterPoint_arbitrary(theta_0,
    howManyPointsAfterHead,l,l_h,curve)

```

```

169     [n,~] = size(theta_0);
170     thetas = zeros(n ,1+howManyPointsAfterHead);
171     %disp(thetas)
172     thetas(:,1)=theta_0;
173     thetas = getNextPoint_arbitrary(thetas,l_h,1, curve);
174     for i=2:howManyPointsAfterHead
175         thetas = getNextPoint_arbitrary(thetas,l, i, curve);
176
177     end
178     thetaIncludingAfter = thetas;
179 end
180
181 % solveAfterPoint(theta_0_val_q1,0.55,223,1.65,2.86)
182 % solveAfterPoint(theta_0_val_q1(1:3),0.55,20,1.65,2.86)
183 % solveAfterPoint(thetas_sries,0.55,223,l,l_h)
184
185 % 相邻点的极角，后者对前者的导数
186 syms theta_i theta_ip1 l d
187 F = theta_i^2 + theta_ip1^2 - 2*theta_i*theta_ip1*cos(theta_i - theta_ip1) -
    (4*pi^2*l^2)/d^2;
188 dFdtheta_i = diff(F,theta_i);
189 dFdtheta_ip1 = diff(F,theta_ip1);
190 dtheta_ip1dtheta_i = - dFdtheta_i/dFdtheta_ip1;
191 dtheta_ip1dtheta_i_func=matlabFunction(dtheta_ip1dtheta_i, 'Vars', [theta_i,
    theta_ip1]);

```

附录 2-2: 第一问

```

1     m = 128; n = 256;
2     % 128x256矩阵，每个元素服从Gauss随机分布
3     A = randn(m, n);
4     % 精确解mu只有10%元素非零，每一个非零元素也服从高斯分布
5     % 可保证mu是方程组唯一的非零元素最少的解
6     u = sprandn(n, 1, 0.1);
7     b = A * u;

```

附录 2-3: 第二问

```

1     l_h = 2.86;
2     l = 1.65;
3     % 第一问
4     % 积分得时间对龙头极角的函数符号对象
5     syms theta_0_start
6     d_q1=0.55;
7     v_q1=1.00;
8     dthetadt_sprial = subs(dthetadt,theta,theta_0);
9     t_theta_0_q1_ =int(1/dthetadt_sprial,theta_0);
10    t_theta_0_q1 = t_theta_0_q1_ - subs(t_theta_0_q1_,theta_0,theta_0_start);
11
12

```

```

13
14
15 % 求不同时间下龙头极角的值 (数值解)
16 % 定义微分方程
17 dthetadt_val_q1 = subs(dthetadt_sprial,[d,v],[d_q1,v_q1]);
18 dthetadt_func_q1 = matlabFunction(dthetadt_val_q1, 'Vars', theta_0);
19
20 % 设置初始条件和时间范围
21 theta0_q1 = 32*pi; % 初始条件
22 tspan_q1 = 0:1:300; % 时间范围从 0 到 300
23
24 % 调用 ode45 求解
25 opts = odeset('RelTol',1e-20,'AbsTol',1e-20);
26 [t_q1, theta_0_val_q1] = ode45(@(t, theta_temp) dthetadt_func_q1(theta_temp),
    tspan_q1, theta0_q1, opts);
27 % [t, theta_0_val] = ode45(@(t, theta) dthetadt_func(theta), tspan, theta0);
28 % 绘制结果
29 plot(t_q1, theta_0_val_q1);
30 xlabel('时间_t');
31 ylabel('解_\theta(t)');
32
33
34 % 将时间对龙头极角的函数的数值对象带入第一问的参数并转化为函数对象
35 t_theta_0_val_q1 = subs(t_theta_0_q1,[d,v],[d_q1,v_q1]);
36 t_theta_0_func_q1 = subs(t_theta_0_val_q1,theta_0_start,32*pi);
37 t_theta_0_func_q1 = matlabFunction(t_theta_0_func_q1, 'Vars', theta_0);
38
39 % 累计误差
40 error_q1=sqrt(sum((t_q1-t_theta_0_func_q1(theta_0_val_q1)).^2)/301)
41
42 % thetas_val_q1 = theta_0_val_q1
43 %thetas_val_q1 = addNextPoint(thetas_val_q1,l_h,d_q1)
44 % 计算不同时刻下龙头后各点的极角
45 [n,~]= size(theta_0_val_q1);
46 thetas_val_q1_all = zeros(n,224);
47 thetas_val_q1_all(:,1)=theta_0_val_q1;
48 thetas_val_q1_all = getNextPoint(thetas_val_q1_all,l_h,d_q1,1);
49 for i=2:224
50     thetas_val_q1_all = getNextPoint(thetas_val_q1_all,l,d_q1,i);
51 end
52 thetas_val_q1 = thetas_val_q1_all(:,2:end);
53
54 %%thetas_val_q1_all_test = solveAfterPoint(theta_0_val_q1(1:3),d_q1,
    20,1.65,2.86)
55
56 % 点的距离检查
57 dis=zeros(n,223);
58 parfor i=1:223
59     for j=1:n

```

```

60         dis(j,i)= getdis_a(thetas_val_q1_all(j,i),thetas_val_q1_all(j,i+1),
61                               d_q1);
62     end
63 end
64 % 计算速度
65 dtheta_idtheta_ip1_val = zeros(n,223);
66 parfor i = 1:223
67     for j = 1:n
68         dtheta_idtheta_ip1_val(j,i) = dtheta_ip1dtheta_i_func(
69             thetas_val_q1_all(j,i),thetas_val_q1_all(j,i+1));
70     end
71 end
72 dtheta_idtheta_ip1_val_multiplicative = dtheta_idtheta_ip1_val;
73 for i = 2:223
74     for j = 1:n
75         dtheta_idtheta_ip1_val_multiplicative(j,i) =
76             dtheta_idtheta_ip1_val_multiplicative(j,i)*
77             dtheta_idtheta_ip1_val_multiplicative(j,i-1);
78     end
79 end
80 % 计算速度
81 vs = zeros(size(thetas_val_q1));
82 parfor i = 1:223
83     for j = 1:n
84         vs(j,i)=dLdtheta_func(thetas_val_q1(j,i), d_q1)*
85             dtheta_idtheta_ip1_val_multiplicative(j,i)*dthetadt_func_q1(
86                 theta_0_val_q1(j));
87     end
88 end
89 vshow = -vs
90 rho = d_q1/(2*pi)*thetas_val_q1_all
91 % 创建一个新的图形窗口
92 figure;
93 % 遍历每一行并绘制极坐标散点图
94 c =ones(1,224)*3
95 c(1)=2
96 for i = 1:size(thetas_val_q1_all, 1)
97     polarscatter(0, max(rho),5);
98     polarscatter(thetas_val_q1_all(i, :), rho(i, :),25,c,'filled');
99     pause(0.05);
100 end
101
102

```

```

103 % 动图
104 drow(thetas_val_q1_all,@(t_n) d_q1/2/pi*t_n,@(t_n) t_n);

```

附录 2-4: 第三问

```

1 % 第三问
2 % 求临界极角
3 syms d
4 theta0_edge = 9*pi/d
5 theta0_edge_func = matlabFunction(theta0_edge,"Vars",d);
6
7 % 绘制碰撞条件
8 low = 0.1;
9 up = 3;
10 NN3 =50;
11
12 d_sries = zeros(NN3+1,1);
13 thetas_sries3 = zeros(NN3+1,1);
14 d_sries(1) = low;
15 thetas_sries3(1) = theta0_edge_func(low);
16 for i =1:NN3
17     d_t = low + (up-low)*i/NN3;
18     d_sries(i+1) = d_t;
19     thetas_sries3(i+1) = theta0_edge_func(d_t);
20
21     %maxs(i) = getmax(theta_t,l_h,l,0.55,222);
22 end
23
24 all_theta3=solveAfterPoint(thetas_sries3,d_sries,223,l,l_h);
25
26
27 max_err3=NaN(NN3+1,1);
28 for i = 1:(NN3+1)
29     check=crossfunctions(all_theta3(i,1), all_theta3(i,2:end),...
30         d_sries(i) * all_theta3(i,1)/2/pi, ...
31         d_sries(i) * all_theta3(i, 2:end)/2/pi, 222, l_h, l);
32     max_err3(i) = max(check,[],"all");
33 end
34
35 plot(thetas_sries3, max_err3, '-*b', ...
36     thetas_sries3, zeros(size(thetas_sries3)), '-r')
37
38 plot(d_sries, max_err3, 'Marker','*', 'Color',[46 139 87]/255)
39 hold on
40 plot(d_sries, zeros(size(thetas_sries3)), 'Color',[238 118 0]/255)
41 hold on
42
43 % 割线法搜索
44 x0 = 0.4; % 初始猜测1
45 x1 = 0.45; % 初始猜测2

```

```

46 tol = 1e-7; % 容差
47 max_iter = 300; % 最大迭代次数
48 for i = 1:max_iter
49     disp(i)
50     theta_x0 = theta0_edge_func(x0);
51     theta_x1 = theta0_edge_func(x1);
52     f_x0 = getmax_single(theta_x0,l_h,l,x0,223);
53     f_x1 = getmax_single(theta_x1,l_h,l,x1,223);
54     x2 = x1 - f_x1 * (x1 - x0) / (f_x1 - f_x0);
55     if abs(x2 - x1) < tol
56         break;
57     end
58     x0 = x1;
59     x1 = x2;
60 end
61 d_at_edge= x2;
62 theta_at_edge3 = theta0_edge_func(d_at_edge)
63
64 getmax_single(theta_at_edge3,l_h,l,d_at_edge,223)
65
66 q3_all_thetas = solveAfterPoint(theta_at_edge3,d_at_edge,223,l,l_h);
67
68 % 计算速度
69 dthetadt_val_q3 = subs(dthetadt_sprial,[d,v],[d_at_edge,v_q1])
70 dthetadt_func_q3 = matlabFunction(dthetadt_val_q3, 'Vars', theta_0);
71
72 dtheta_idtheta_ip1_val_q3 = zeros(1,223);
73 parfor i = 1:223
74     dtheta_idtheta_ip1_val_q3(1,i) =...
75         dtheta_ip1dtheta_i_func(q3_all_thetas(1,i), ...
76         q3_all_thetas(1,i+1));
77
78 end
79 dtheta_idtheta_ip1_val_multiplicative_q3 = dtheta_idtheta_ip1_val_q3;
80 for i = 2:223
81     dtheta_idtheta_ip1_val_multiplicative_q3(1,i) =...
82     dtheta_idtheta_ip1_val_multiplicative_q3(1,i) ...
83     *dtheta_idtheta_ip1_val_multiplicative_q3(1,i-1);
84 end
85
86 % 计算速度
87 vs3 = zeros(1,223);
88 parfor i = 1:223
89     vs3(1,i)=...
90     dLdtheta_func(q3_all_thetas(1,i), d_at_edge) ...
91     *dtheta_idtheta_ip1_val_multiplicative_q3(1,i) ...
92     *dthetadt_func_q3(q3_all_thetas(1));
93 end
94 vs3=abs(vs3)

```


附录 2-5: 第四问

```

1 % 第四问
2
3 % 确定盘入、盘出掉头区域的的具体参数
4 R = 4.5; % 掉头空间的半径
5 d_q4 = 1.7; % 螺距
6 theta_sprial_in = R/d_q4*2*pi; % 掉头空间边界与盘入螺线交点的极角
7 theta_change_arc = theta_sprial_in - pi;
8
9 h_fun = @(x) (x*cos(x) + sin(x))/(-x*sin(x)+cos(x));
10 h = h_fun(theta_sprial_in); % 螺线与圆弧的切线的斜率
11 atan_h = atan(h); % 螺线与圆弧的切线的倾斜角(-pi/2, pi/2)
12 gamma = mod(theta_sprial_in - atan_h - pi/2, pi); % 螺线与圆弧的切线的法线与
    进出点连线的夹角
13 x_sprial_in = R*cos(theta_sprial_in);
14 y_sprial_in = R*sin(theta_sprial_in);
15 x_con = -1/3*x_sprial_in;
16 y_con = -1/3*y_sprial_in;
17 x_sprial_out = -x_sprial_in;
18 y_sprial_out = -y_sprial_in;
19 f11 = @(x) -1/h*(x - x_sprial_in) + y_sprial_in;
20 f12 = @(x) -1/h*(x - x_sprial_out) + y_sprial_out;
21 f2 = @(x) tan(atan_h + pi/2 + 2*gamma)*(x - x_con) + y_con;
22
23 x_arc_c_in = fzero(@(x) f11(x)-f2(x), 0); % 第一段弧的圆心
24 y_arc_c_in = f2(x_arc_c_in);
25 R_arc_in = norm([x_arc_c_in y_arc_c_in] - [x_sprial_in y_sprial_in]);
26 r_arc_c_in = norm([x_arc_c_in, y_arc_c_in]);
27 theta_arc_c_in = atan(y_arc_c_in/x_arc_c_in) + pi;
28
29 x_arc_c_out = fzero(@(x) f12(x)-f2(x), 0); % 第二段弧的圆心
30 y_arc_c_out = f2(x_arc_c_out);
31 R_arc_out = norm([x_arc_c_out y_arc_c_out] - [x_sprial_out y_sprial_out]);
32 r_arc_c_out = norm([x_arc_c_out, y_arc_c_out]);
33 theta_arc_c_out = atan(y_arc_c_out/x_arc_c_out);
34
35 % 求极角反向点
36 theta_reverse = atan(y_arc_c_out/x_arc_c_out) - asin(R_arc_out/norm([
    x_arc_c_out, y_arc_c_out]));
37 k_reverse = tan(theta_reverse);
38
39 while theta_reverse + 2*pi < theta_sprial_in
40     theta_reverse = theta_reverse + 2*pi;
41 end
42
43 theta_sprial_out = 2*theta_reverse - theta_sprial_in+pi;
44
45
46 % 求解不同自然极角下的极径

```

```

47 % 确定各段曲线方程
48 v_h = 1;
49 syms theta_real
50
51 r_sprial_in = d_q4 /2/pi*theta_real
52 r_sprial_out = d_q4/2/pi*(theta_real+pi)
53
54 r_sprial_in_func = matlabFunction(r_sprial_in, 'Vars',theta_real);
55 r_sprial_out_func = matlabFunction(r_sprial_out, 'Vars',theta_real);
56
57 r_arc_in_plus = r_arc_c_in*cos(theta_real-theta_arc_c_in) ...
58 + sqrt(r_arc_c_in^2*cos(theta_real-theta_arc_c_in)^2- r_arc_c_in^2+R_arc_in^2)
59 ;
60 r_arc_in_minus = r_arc_c_in*cos(theta_real-theta_arc_c_in) ...
61 - sqrt(r_arc_c_in^2*cos(theta_real-theta_arc_c_in)^2- r_arc_c_in^2+R_arc_in^2)
62 ;
63 r_arc_out_plus = r_arc_c_out*cos(theta_real-theta_arc_c_out) ...
64 + sqrt(r_arc_c_out^2*cos(theta_real-theta_arc_c_out)^2- r_arc_c_out^2+
65 R_arc_out^2);
66 r_arc_out_minus = r_arc_c_out*cos(theta_real-theta_arc_c_out) ...
67 - sqrt(r_arc_c_out^2*cos(theta_real-theta_arc_c_out)^2- r_arc_c_out^2+
68 R_arc_out^2);
69
70
71 % 打包成从自然极角得到极径的函数
72 r_from_theta_natural = CombinedFunction();
73 r_from_theta_natural=r_from_theta_natural.setFunction(r_sprial_in_func,
74 r_sprial_out_func, ...
75 r_arc_in_plus_func, r_arc_out_minus_func, r_arc_out_plus_func);
76 r_from_theta_natural=r_from_theta_natural.setThreshold(theta_sprial_in,
77 theta_sprial_out, theta_reverse,theta_change_arc);
78
79
80 figure
81 % 定义极角 theta
82 %theta_plot_n = linspace(-4*pi, 4*pi, 1000);
83 theta_plot_n = linspace(theta_sprial_out-4*pi, theta_sprial_in+pi, 5000);
84 %theta_plot_n = linspace(theta_reverse, theta_sprial_in+pi, 5000);
85 %theta_plot_n = linspace(theta_sprial_out+pi, theta_reverse-0.001, 5000);
86
87 % 定义极径 r, 假设 getr 是一个函数句柄
88 r_plot = arrayfun(getr,theta_plot_n);
89 theta_plot_r = r_from_theta_natural.getRealTheta(theta_plot_n);

```

```

90 % 绘制极坐标图
91 polarplot(theta_plot_r, r_plot);
92 title('极坐标图');
93
94
95
96 % 求解不同自然极角下的弧线长度对真实极角的导数
97 % 求解不同自然极角下弧线长度对真实极角的导数
98 syms theta_real
99
100 dLdtheta_r_sprial_in = - simplify(sqrt((r_sprial_in)^2+diff(r_sprial_in,
    theta_real)^2));
101 dLdtheta_r_sprial_out = simplify(sqrt((r_sprial_out)^2+diff(r_sprial_out,
    theta_real)^2));
102 dLdtheta_r_arc_in_plus = - simplify(sqrt((r_arc_in_plus)^2+diff(r_arc_in_plus,
    theta_real)^2));
103 dLdtheta_r_arc_out_minus = - simplify(sqrt((r_arc_out_minus)^2+diff(
    r_arc_out_minus, theta_real)^2));
104 dLdtheta_r_arc_out_plus = simplify(sqrt((r_arc_out_plus)^2+diff(r_arc_out_plus
    , theta_real)^2));
105 % 求解不同自然极角下弧线长度对自然极角的导数
106 dLdtheta_n_sprial_in = dLdtheta_r_sprial_in;
107 % dLdtheta_n_sprial_out = - subs(dLdtheta_r_sprial_out, theta_real, 2*
    theta_reverse - theta_real);
108 dLdtheta_n_sprial_out = - dLdtheta_r_sprial_out;
109 dLdtheta_n_arc_in_plus = dLdtheta_r_arc_in_plus;
110 dLdtheta_n_arc_out_minus = dLdtheta_r_arc_out_minus;
111 % dLdtheta_n_arc_out_plus = - subs(dLdtheta_r_arc_out_plus, theta_real, 2*
    theta_reverse - theta_real);
112 dLdtheta_n_arc_out_plus = - dLdtheta_r_arc_out_plus;
113 % 转化为函数对象
114 dLdtheta_n_sprial_in_func = matlabFunction(dLdtheta_n_sprial_in, 'Vars',
    theta_real);
115 dLdtheta_n_sprial_out_func = matlabFunction(dLdtheta_n_sprial_out, 'Vars',
    theta_real);
116 dLdtheta_n_arc_in_plus_func = matlabFunction(dLdtheta_n_arc_in_plus, 'Vars',
    theta_real);
117 dLdtheta_n_arc_out_minus_func = matlabFunction(dLdtheta_n_arc_out_minus, 'Vars
    ', theta_real);
118 dLdtheta_n_arc_out_plus_func = matlabFunction(dLdtheta_n_arc_out_plus, 'Vars',
    theta_real);
119 % 打包成从自然极角得到弧线长度对自然极角的导数的函数
120 dLdtheta_n_from_theta_natural = CombinedFunction();
121 dLdtheta_n_from_theta_natural=dLdtheta_n_from_theta_natural.setFunction(
    dLdtheta_n_sprial_in_func, dLdtheta_n_sprial_out_func, ...
122     dLdtheta_n_arc_in_plus_func, dLdtheta_n_arc_out_minus_func,
        dLdtheta_n_arc_out_plus_func);
123 dLdtheta_n_from_theta_natural=dLdtheta_n_from_theta_natural.setThreshold(
    theta_sprial_in, theta_sprial_out, theta_reverse, theta_change_arc);

```

```

124
125 getdLdtheta = @(theta_n) dLdtheta_n_from_theta_natural.evaluate(theta_n);
126
127 fplot(getdLdtheta, [-100, 100]);
128
129 % 对龙头的, 不同自然极角下自然极角对时间的导数
130 dthetandt_sprial_in = simplify(v_h/dLdtheta_n_sprial_in);
131 dthetandt_sprial_out = simplify(v_h/dLdtheta_n_sprial_out);
132 dthetandt_arc_in_plus = simplify(v_h/dLdtheta_n_arc_in_plus);
133 dthetandt_arc_out_minus = simplify(v_h/dLdtheta_n_arc_out_minus);
134 dthetandt_arc_out_plus = simplify(v_h/dLdtheta_n_arc_out_plus);
135
136 dthetandt_sprial_in_func= matlabFunction(dthetandt_sprial_in, 'Vars',
    theta_real);
137 dthetandt_sprial_out_func= matlabFunction(dthetandt_sprial_out, 'Vars',
    theta_real);
138 dthetandt_arc_in_plus_func= matlabFunction(dthetandt_arc_in_plus, 'Vars',
    theta_real);
139 dthetandt_arc_out_minus_func= matlabFunction(dthetandt_arc_out_minus, 'Vars',
    theta_real);
140 dthetandt_arc_out_plus_func= matlabFunction(dthetandt_arc_out_plus, 'Vars',
    theta_real);
141 % 打包成对龙头的从自然极角得到弧线长度对时间的导数的函数
142 dthetandt_from_theta_natural = CombinedFunction();
143
144 dthetandt_from_theta_natural=dthetandt_from_theta_natural.setFunction(
    dthetandt_sprial_in_func, dthetandt_sprial_out_func, ...
145 dthetandt_arc_in_plus_func, dthetandt_arc_out_minus_func,
    dthetandt_arc_out_plus_func);
146
147 dthetandt_from_theta_natural=dthetandt_from_theta_natural.setThreshold(
    theta_sprial_in, theta_sprial_out, theta_reverse, theta_change_arc);
148
149 getdtheta0ndt = @(theta_n) dthetandt_from_theta_natural.evaluate(theta_n);
150
151 fplot(getdtheta0ndt, [10, 20]);
152
153
154
155 % 求不同时间下龙头极角的值 (数值解)
156 % 定义微分方程
157 % dthetadt 就是 getdthetardt
158 % 设置初始条件和时间范围
159 theta0_q4 = theta_sprial_in; % 初始条件
160 tspan_q4_pos = 0:1:100; % 时间范围从 0 到 300
161
162 % 调用 ode45 求解
163 opts = odeset('RelTol', 1e-10, 'AbsTol', 1e-10);
164 [t_q4_pos, theta_0_val_q4_pos] = ode45(@(t, theta_temp) getdtheta0ndt(

```

```

        theta_temp), tspan_q4_pos, theta0_q4,opts);
165 % [t, theta_0_val] = ode45(@(t, theta) dthetadt_func(theta), tspan, theta0);
166
167 tspan_q4_neg = -100:1:0; % 时间范围从 0 到 300
168
169 % 调用 ode45 求解
170 opts = odeset('RelTol',1e-10,'AbsTol',1e-10);
171 [t_q4_neg, theta_0_val_q4_neg] = ode45(@(t, theta_temp) getdtheta0ndt(
        theta_temp), fliplr(tspan_q4_neg), theta0_q4,opts);
172 % [t, theta_0_val] = ode45(@(t, theta) dthetadt_func(theta), tspan, theta0);
173 % 绘制结果
174 theta_0_val_q4_neg = fliplr(theta_0_val_q4_neg);
175 t_q4 = [t_q4_neg, t_q4_pos];
176 theta_0_val_q4_show = [theta_0_val_q4_neg, theta_0_val_q4_pos];
177 plot(t_q4, theta_0_val_q4_show);
178 xlabel('时间 t');
179 ylabel('解 \theta(t)');
180 theta_0_val_q4 = [flipud(theta_0_val_q4_neg); theta_0_val_q4_pos(2:end)];
181
182
183 % thetas_val_q1 = theta_0_val_q1
184 % thetas_val_q1 = addNextPoint(thetas_val_q1, l_h, d_q1)
185 % 计算不同时刻下龙头后各点的极角 theta_0, howManyPointsAfterHead, l, l_h, curve
186 thetas_val_q4_all = solveAfterPoint_arbitrary(theta_0_val_q4, 223, l, l_h, getr);
187 thetas_val_q4 = thetas_val_q4_all(:, 2:end);
188
189 % 计算速度
190 % 首先分别讨论 dtheta_i+1/dtheta_i 在不同情况下的函数 符号对象
191 syms theta_ip1 theta_i
192
193 getdd = @(r_ip1_, r_i_, theta_ip1_, theta_i_) - diff(r_ip1_^2 + r_i_^2 - 2*r_ip1_*r_i_
        *cos(theta_ip1_ - theta_i_), theta_i_) ...
194 / diff(r_ip1_^2 + r_i_^2 - 2*r_ip1_*r_i_*cos(theta_ip1_ - theta_i_), theta_ip1_);
195 dd_s_in_s_in = simplify(getdd(subs(r_sprial_in, theta_real, theta_ip1), subs(
        r_sprial_in, theta_real, theta_i), theta_ip1, theta_i));
196 dd_s_in_a_in = simplify(getdd(subs(r_sprial_in, theta_real, theta_ip1), subs(
        r_arc_in_plus, theta_real, theta_i), theta_ip1, theta_i));
197 dd_a_in_a_in = simplify(getdd(subs(r_arc_in_plus, theta_real, theta_ip1), subs(
        r_arc_in_plus, theta_real, theta_i), theta_ip1, theta_i));
198 dd_a_in_a_o1 = simplify(getdd(subs(r_arc_in_plus, theta_real, theta_ip1), subs(
        r_arc_out_minus, theta_real, theta_i), theta_ip1, theta_i));
199 dd_a_in_a_o2 = simplify(getdd(subs(r_arc_in_plus, theta_real, theta_ip1), subs(
        r_arc_out_plus, theta_real, theta_i), theta_ip1, theta_i));
200 dd_a_o1_a_o1 = simplify(getdd(subs(r_arc_out_minus, theta_real, theta_ip1), subs(
        r_arc_out_minus, theta_real, theta_i), theta_ip1, theta_i));
201 dd_a_o1_a_o2 = simplify(getdd(subs(r_arc_out_minus, theta_real, theta_ip1), subs(
        r_arc_out_plus, theta_real, theta_i), theta_ip1, theta_i));
202 dd_a_o1_s_out = simplify(getdd(subs(r_arc_out_minus, theta_real, theta_ip1), subs(
        r_sprial_out, theta_real, theta_i), theta_ip1, theta_i));

```

```

203 dd_a_o2_a_o2 = simplify(getdd(subs(r_arc_out_plus,theta_real,theta_ip1),subs(
    r_arc_out_plus,theta_real,theta_i),theta_ip1,theta_i));
204 dd_a_o2_s_ou = simplify(getdd(subs(r_arc_out_plus,theta_real,theta_ip1),subs(
    r_sprial_out,theta_real,theta_i),theta_ip1,theta_i));
205 dd_s_ou_s_ou = simplify(getdd(subs(r_sprial_out,theta_real,theta_ip1),subs(
    r_sprial_out,theta_real,theta_i),theta_ip1,theta_i));
206
207
208 % 转为函数对象
209 dd_s_in_s_in_func = matlabFunction(dd_s_in_s_in,'Vars',[theta_ip1,theta_i]);
210 dd_s_in_a_in_func = matlabFunction(dd_s_in_a_in,'Vars',[theta_ip1,theta_i]);
211 dd_a_in_a_in_func = matlabFunction(dd_a_in_a_in,'Vars',[theta_ip1,theta_i]);
212 dd_a_in_a_o1_func = matlabFunction(dd_a_in_a_o1,'Vars',[theta_ip1,theta_i]);
213 dd_a_in_a_o2_func = matlabFunction(dd_a_in_a_o2,'Vars',[theta_ip1,theta_i]);
214 dd_a_o1_a_o1_func = matlabFunction(dd_a_o1_a_o1,'Vars',[theta_ip1,theta_i]);
215 dd_a_o1_a_o2_func = matlabFunction(dd_a_o1_a_o2,'Vars',[theta_ip1,theta_i]);
216 dd_a_o1_s_ou_func = matlabFunction(dd_a_o1_s_ou,'Vars',[theta_ip1,theta_i]);
217 dd_a_o2_a_o2_func = matlabFunction(dd_a_o2_a_o2,'Vars',[theta_ip1,theta_i]);
218 dd_a_o2_s_ou_func = matlabFunction(dd_a_o2_s_ou,'Vars',[theta_ip1,theta_i]);
219 dd_s_ou_s_ou_func = matlabFunction(dd_s_ou_s_ou,'Vars',[theta_ip1,theta_i]);
220
221 % 打包成类对象
222 dd_ip1_i = Combinedd();
223 dd_ip1_i = dd_ip1_i.setFunction( ...
224     dd_s_in_s_in_func,...
225     dd_s_in_a_in_func,...
226     dd_a_in_a_in_func,...
227     dd_a_in_a_o1_func,...
228     dd_a_in_a_o2_func,...
229     dd_a_o1_a_o1_func,...
230     dd_a_o1_a_o2_func,...
231     dd_a_o1_s_ou_func,...
232     dd_a_o2_a_o2_func,...
233     dd_a_o2_s_ou_func,...
234     dd_s_ou_s_ou_func);
235 dd_ip1_i = dd_ip1_i.setThreshold(theta_sprial_in, theta_sprial_out,
    theta_reverse,theta_change_arc);
236 get_dtheta_ip1_dtheta_i = @(t_ip1,t_i) dd_ip1_i.evaluate(t_ip1,t_i);
237
238
239 % 计算速度
240 function vs=getvs_q4(thetas_val_all, dthetaip1_dthetai,dtheta0ndt,dLdtheta)
241
242 [n,nn] =size(thetas_val_all);
243 nn = nn-1;
244 dtheta_idtheta_ip1_valq4 = zeros(n,nn);
245 for i = 1:nn
246     for j = 1:n
247         dtheta_idtheta_ip1_valq4(j,i) = dthetaip1_dthetai(thetas_val_all(j

```

```

        , i+1), thetas_val_all(j, i));
248     end
249 end
250 dtheta_idtheta_ip1_val_multiplicative4 = dtheta_idtheta_ip1_valq4;
251 for i = 2:nn
252     for j = 1:n
253         dtheta_idtheta_ip1_val_multiplicative4(j, i) = ...
254             dtheta_idtheta_ip1_val_multiplicative4(j, i)*
                dtheta_idtheta_ip1_val_multiplicative4(j, i-1);
255     end
256 end
257
258 vs_ = zeros(n, nn);
259 for i = 1:nn
260     for j = 1:n
261         vs_(j, i) = dtheta0ndt(thetas_val_all(j, 1))*
                dtheta_idtheta_ip1_val_multiplicative4(j, i)*dLdtheta(
                thetas_val_all(j, i+1));
262     end
263 end
264 vs = abs(vs_);
265 end
266 % 增加一层封装，只返回最大值速度和其索引
267 function [vmax, index] = getvsmax_q5(thetas_val_all, dthetaip1_dthetai,
    dtheta0ndt, dLdtheta)
268 % 计算 vs 矩阵
269 vs = getvs_q4(thetas_val_all, dthetaip1_dthetai, dtheta0ndt, dLdtheta);
270
271 % 找到 vs 矩阵中的最大值及其线性索引
272 [vmax, index] = max(vs, [], 2);
273
274 % 将线性索引转换为行列索引
275 % [row, col] = ind2sub(size(vs), linearIndex);
276 end
277
278 vs_q4 = getvs_q4(thetas_val_q4_all, get_dtheta_ip1_dtheta_i, getdtheta0ndt,
    getdLdtheta);
279
280 close all;
281
282 function no = draw(thetas, rfromthetas, real_theta_from_theta)
283     rho = arrayfun(rfromthetas, thetas);
284     thetas_val_all_plot = real_theta_from_theta(thetas);
285     max_r = max(rho, [], 'all');
286
287 % 遍历每一行并绘制极坐标散点图
288 for i = 1:size(thetas_val_all_plot, 1)
289     % 绘制第一个点
290     % rlim([0 max_r]);

```

```

291     polarplot(thetas_val_all_plot(i,1), rho(i,1), 'ro','MarkerSize',3, '
        MarkerFaceColor', 'r'); % 红色实心点
292     hold on;
293
294     % 绘制其余的点
295     polarplot(thetas_val_all_plot(i,2:end), rho(i,2:end), 'bo','MarkerSize
        ',3); % 蓝色点
296
297     % 绘制连线
298     polarplot(thetas_val_all_plot(i,:) ', rho(i,:) ', '-','Color',[139 131
        120]/255); % 绿色线
299     %polarscatter(thetas_val_all_plot(i, :), rho(i, :),25,c,'filled ');
300     hold off
301     pause(0.05);
302
303     end
304     no=1;
305 end
306
307 figure
308 drow(thetas_val_q4_all,getr,@(t_n) r_from_theta_natural.getRealTheta(t_n))
309
310 [maxv,maxindex]=max(vs_q4,[],2)

```

附录 2-6: 第五问

```

1 % 第五问
2 % 利用第四问的结果，在龙头把手速度取1时计算各节点在不同位置时的速度，取其最大
   值并调整龙头速度令其等于2m/s
3 % 龙头把手位置搜索范围的确定
4 theta0_searching_range = linspace(theta_sprial_in + pi, -27, 1000)';
5 thetas_all_searching = solveAfterPoint_arbitrary(theta0_searching_range,223,l,
   l_h,getr);
6 v_max_infor = zeros(length(theta0_searching_range),2);
7 axv_and_theta_and_theta0_and_index=zeros(length(theta0_searching_range),4);
8
9
10 drow(thetas_all_searching ,getr,@(t_n) r_from_theta_natural.getRealTheta(t_n))
11
12 goal = get_goal_func;goal = goal.setFunction(@(x) solveAfterPoint_arbitrary(x
   ,223,l,l_h,getr), ...
13     @(x) getvs_q4(x,get_dtheta_ip1_dtheta_i,getdtheta0ndt,getdLdtheta));
14 maxtheta = @(theta) -goal.evaluate(theta);
15 x0 = 4*pi;
16
17 % 调用搜索
18 [x, fval] = fminsearch(maxtheta, x0);
19
20 % 显示结果
21 disp(['最优解:␣', num2str(x)]);

```



```

22 disp(['目标函数值:␣', num2str(fval)]);
23
24 2/fval
25
26 vs_q5 = getvs_q4(theras_all_searching, get_dtheta_ip1_dtheta_i, getdtheta0ndt,
    getdLdtheta);
27
28
29 [maxVsee, maxIndexsee] = max(vs_q5, [], 2);
30 plot(theta0_searching_range, maxVsee)
31
32 goal = get_goal_func; goal = goal.setFunction(@(x) solveAfterPoint_arbitrary(x
    ,223,l,l_h,getr), ...
33     @(x) getvs_q4(x, get_dtheta_ip1_dtheta_i, getdtheta0ndt, getdLdtheta));
34 maxtheta = @(theta) goal.evaluate(theta);
35 x0 = 0;
36
37 % 调用模拟退火算法
38 [x, fval] = simulannealbnd(fun, x0);
39
40 % 显示结果
41 disp(['最优解:␣', num2str(x)]);
42 disp(['目标函数值:␣', num2str(fval)]);

```