

“板凳龙”行进过程的优化设计

摘 要

“板凳龙”活动是浙闽地区的民俗文化活动。本文根据等距螺线性质，对板凳龙行进过程进行分析，建立舞龙队运动模型，通过龙头位置递推龙身、龙尾位置，并通过龙头前把手速度递推各把手速度。最后对舞龙队行进路线、速度进行优化，使用二分法求解，提供兼具观赏性和安全性的优化方案。

针对问题一：构建了舞龙队运动模型。首先，选定极坐标系设立等距螺线方程，转换为自然坐标系遍历龙头前把手位置。将第 i 个把手所在螺线近似为曲率圆，按把手间距离截取曲率圆得到第 $i+1$ 个把手位置。其次，通过速度传递模型将龙头速度向后递推，得到各把手行进速度。最后，通过递推求解得到所有把手的位置和速度，并在 result1.xlsx 中给出。计算得到 300 s 时龙头坐标为 (4.420274m, 2.320429m)。观察发现，龙身、龙尾把手速度在龙头速度附近上下波动。

针对问题二：构建了板凳碰撞检验模型，判断是否存在某板凳一角越入另一非相邻板凳内，得知舞龙队能否继续盘入。遍历各板凳，通过坐标变换得到板凳的四个顶点位置，计算顶点到另一板凳轴线和垂线距离，判断是否小于板凳宽和长的一半。沿用问题一的运动模型通过遍历算法得出板凳龙恰好未相撞的时刻为 426.78 s，以及该时刻所有把手的位置和速度，并在 result2.xlsx 中给出。

针对问题三：构建了最小螺距优化模型。以等距螺线的螺距为决策变量，螺距最小为目标函数，板凳间不发生碰撞为约束条件。板凳龙运动轨迹遵从问题一的模型，通过初步分析将搜索范围初值缩小至 [30cm, 50cm]，使用二分法求解出龙头恰能进入掉头空间的最小螺距为 42.06598 cm。

针对问题四：构建了舞龙队全局运动模型。掉头区域半径确定时，掉头曲线唯一确定。在掉头区域外沿用问题一模型，建立自然坐标系与直角坐标系转换关系，对龙头在掉头曲线上的情况进行分析。根据不同情况由几何关系确定龙身、龙尾递推关系，基于问题一的算法，递推求解出舞龙队各把手在 ± 100 s 内每秒的位置和速度，并在 result4.xlsx 中给出。求解发现 +100 s 时存在某把手速度是龙头速度的七倍。

针对问题五：构建了龙头最大速度优化模型。以龙头行进速度为决策变量，龙头行进速度最大为目标函数，各把手最大速度不超过 2 m/s 为约束条件。分别对时间步长取 50、10、5、1、0.5、0.1、0.01 s 运用二分法进行搜索，发现最大允许速度随时间步长的缩短收敛于 0.1875 m/s。

本文针对问题一进行了模型检验，发现板凳龙身长的最大误差仅为 0.01 mm。

关键词： 板凳龙；等距螺线；二分法；坐标变换；优化模型

一、问题重述

“板凳龙”活动是浙闽地区的民俗文化活动。某“板凳龙”由 223 节板凳组成，除去龙头、龙尾外的 221 节板凳组成龙身。各板凳参数与连接方式确定，建立数学模型以解决以下问题。

问题一：舞龙队沿等距螺线顺时针盘入，各把手中心均位于螺线上。螺线螺距为 55 cm，龙把手前进速度保持 1 m/s。初始时刻，龙头位于(880 cm,0)位置。给出 300 s 内整个舞龙队的位置与速度。

问题二：在问题一的基础上，确定舞龙队的终止时刻，使得板凳之间不发生碰撞。给出此时的舞龙队位置和速度。

问题三：螺线中心直径 9 m 区域为掉头空间，用于转换舞龙队前进方向使其能够逆时针盘出。确定最小螺距，使龙头前把手能够沿该螺线盘入掉头区域边界。

问题四：在问题三给定的掉头空间的基础上，给定盘入螺线螺距为 1.7 m，盘出螺线关于盘入螺线中心对称。掉头路径是两段圆弧相切而成的 S 曲线，且前一圆弧半径为后圆弧的两倍。调整圆弧，满足路线各部分相切，使掉头曲线最短。同时给出掉头开始时刻 ± 100 s 的舞龙队位置和速度。

问题五：在问题四的基础上，确定龙头最大行进速度，使舞龙队各把手的速度均不超过 2 m/s。

二、问题分析

2.1 问题一的分析

问题一给定舞龙队盘入路线、初速度以及初始位置，并限制把手位置，要求计算出舞龙队特定把手在特定时刻的位置以及速度。

为计算上述结果，需要建立龙头、龙身位置与速度模型。建立极坐标系与自然坐标系。通过极坐标系与自然坐标对应关系，建立龙头位置模型，确定龙头每一时刻位置。递推计算龙身位置时，视龙身与下一把手位于曲率圆内。联立两圆方程并筛选结果，得到下一把手位置。

通过相邻把手的位置关系，根据相邻两把手速度沿板凳方向的分速度相等关系，解出下一把手的绝对速度。

2.2 问题二的分析

问题二要求计算出舞龙队恰好不发生碰撞的时刻，并给出此时各把手的位置和速度。

为计算上述结果，需要在问题一舞龙队运动模型的基础上，建立碰撞检验模型。选定某一段板凳矩形端点，通过判断其与非相邻龙身矩形板轴线的距离以及与中垂线的距离是否小于该板凳的宽与长的一半，判断两板是否相撞。递推得出相撞时刻。

2.3 问题三的分析

问题三给定舞龙队掉头空间，要求调整等距螺线各项参数，使得龙头可以沿着掉头区域的边界盘入。

为此，设定等距螺线的螺距为决策变量，结合问题一与问题二中的模型建立最小螺距优化模型，设螺线与掉头空间边界相切为终止条件，限定龙头在到达边

界前各部位不发生碰撞，优化得出最小螺距。

2.4 问题四的分析

问题四在问题三原有掉头空间不变的基础上，增大螺距，给予限定条件，要求设计掉头空间内的掉头曲线。保持速度不变，以掉头开始时刻为零时刻，要求计算±100 s内每秒舞龙队特定把手的位置和速度。

为解决上述问题，设定掉头曲线，改进问题一、问题二模型，对掉头区域内舞龙队各把手的运动情况分类讨论，建立新的龙头、龙身位置模型。通过递推得出各把手速度与位置并判断是否相撞。

2.5 问题五的分析

问题五在问题四的基础上保持掉头曲线不变。需要建立速度优化模型，优化龙头的行进速度，使各把手速度不超过要求值，最后计算出龙头最大行进速度。

三、模型假设

1. 假设等距螺线为自中心从 x 轴正半轴开始，逆时针方向向外发散的曲线；
2. 假设初始状态下舞龙队仍沿等距螺线按要求排列；
3. 递推龙身位置时，假设前一把手与后一把手间曲率半径的变化可被忽略；
4. 判断板凳之间碰撞时，忽略相邻板凳区域重合产生的碰撞。

四、符号说明

符号	符号意义	单位
θ	极坐标系下的极角	rad
r	极坐标系下的极径	cm
s	自然坐标系下的路程	cm
ρ	曲率半径	cm
v	行进速度	m/s
d	把手间距	cm

五、模型的建立与求解

5.1 问题一模型的建立与求解

5.1.1 模型建立

1. 极坐标系与自然坐标系的转化

以中心点为极点， x 轴为极轴建立极坐标系。极角 θ 取不小于零的任意值。舞龙队把手沿等距螺线前进。以极坐标形式表示该等距螺线方程，为

$$r = b \frac{\theta}{2\pi}, \quad [1] \quad (1)$$

式中参数 b 为螺距。龙头位置可通过(1)式由参数 θ 完全确定。

等距螺线(1)在 0 到 θ 段的弧长为

$$s(\theta) = \frac{b}{4\pi} \left[\theta \sqrt{1+\theta^2} + \ln(\theta + \sqrt{1+\theta^2}) \right]. \quad [2] \quad (2)$$

视 s 为以原点为起始点，自然坐标系下的坐标，如图 1 所示。

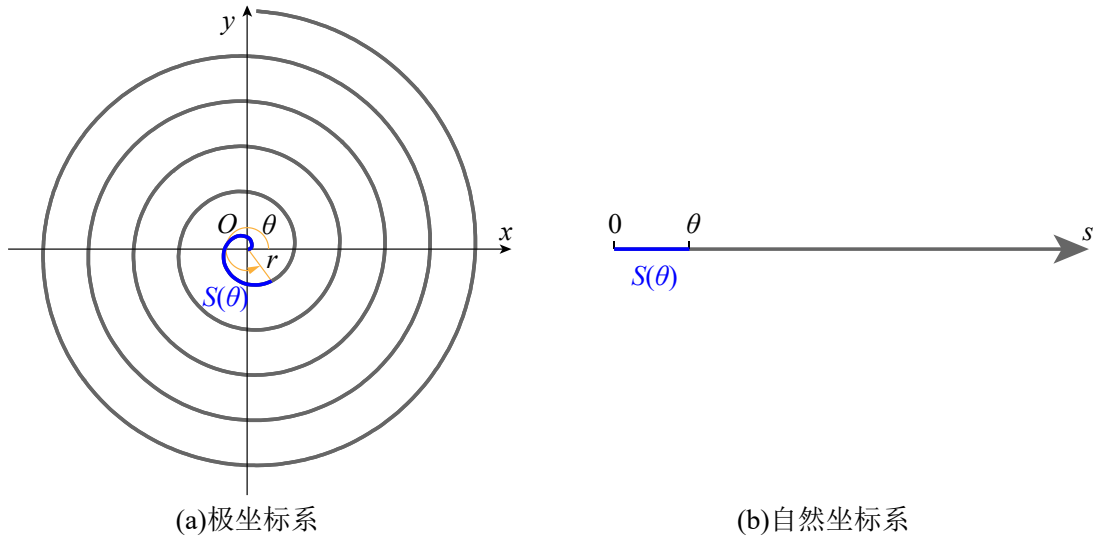


图 1 “板凳龙”路径极坐标系与自然坐标系转化图

2. 龙头位置模型

设(2)式有反函数

$$\theta = \theta(s), \quad (3)$$

可通过与原点路程 s 确定极角 θ 。

初始状态下，龙头处于第 16 圈与 x 轴交点位置，此时有

$$s_0 = s(16 \times 2\pi),$$

龙头匀速前进，即 t 时刻处于

$$s_t = s_0 - vt \quad (4)$$

位置，式中 v 为龙头行进速度。

将 s_t 代入(3)式，进而确定龙头位置。

3. 龙身、龙尾位置模型

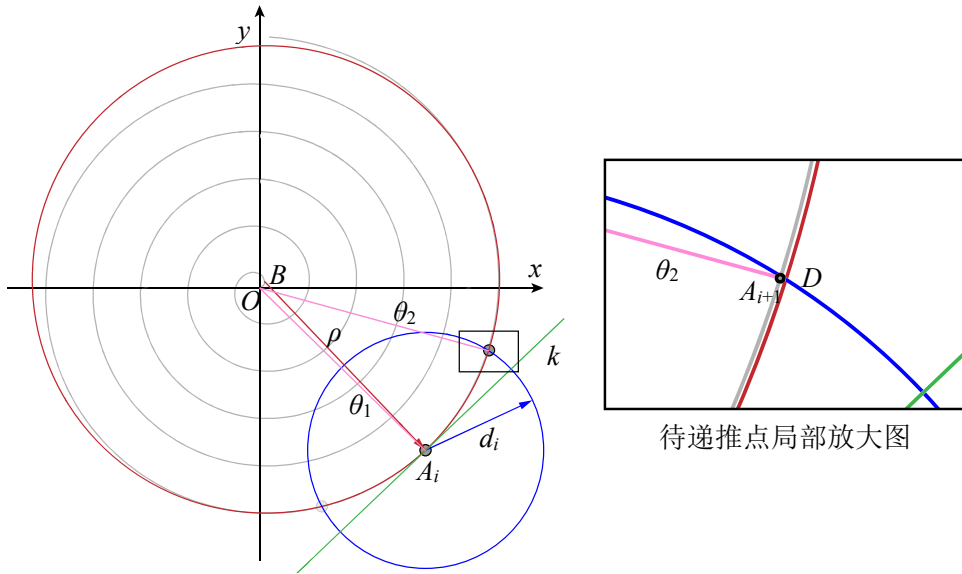


图 2 龙身、龙尾把手坐标递推示意图

现根据第 i 个龙身把手 A_i 递推第 $i+1$ 个龙身（或龙尾）把手 A_{i+1} 。近似考虑 A_i 与 A_{i+1} 同时在等距螺线 A_i 所处曲率圆上，如图 2 圆 B 所示， B 点为曲率圆中

心, ρ 为曲率半径。

(1) 曲率圆的确定

螺线(1)在 θ 处的曲率半径为

$$\rho = \frac{b}{2\pi} \frac{(\theta^2 + 1)^{3/2}}{\theta^2 + 2}, \quad (5)$$

切线斜率为

$$k = \frac{dy}{dx} = \frac{dy/d\theta}{dx/d\theta} = \frac{\sin \theta + \theta \cos \theta}{\cos \theta - \theta \sin \theta}. \quad (6)$$

以 ρ 为曲率半径, 切线斜率为 k 的圆共有两个, 圆心为

$$(x_0 \pm \rho \sin \gamma, y_0 \mp \rho \cos \gamma),$$

式中 x_0, y_0 为 A_i 坐标, $\gamma = \arctan k$ 。取靠近原点的圆心作为曲率圆圆心 B 点坐标 (x_B, y_B) 。

因此, 曲率圆 B 方程为

$$(x - x_B)^2 + (y - y_B)^2 = \rho^2. \quad (7)$$

(2) 把手近似坐标的确定

以 A_i 为圆心, d_i 为半径的圆方程为

$$(x - x_0)^2 + (y - y_0)^2 = d^2, \quad (8)$$

式中, d_i 为板凳 i 两孔间距, 有

$$d_i = \begin{cases} 286 \text{ cm}, & i = 1, \\ 165 \text{ cm}, & \text{其它}. \end{cases}$$

联立(7)(8)两式, 得两个交点 D_1, D_2 。满足关于原点逆时针旋转的向量 \overrightarrow{AD} 的 D 即为下一把手的近似坐标, 此时的极角

$$\theta_2 = \begin{cases} \arctan \frac{y_D}{x_D}, & x > 0 \text{ 且 } y > 0, \\ \arctan \frac{y_D}{x_D} + \pi, & x < 0 \text{ 且 } y > 0, \\ \arctan \frac{y_D}{x_D} + 2\pi, & x < 0 \text{ 且 } y < 0, \\ \arctan \frac{y_D}{x_D} + 2\pi, & x > 0 \text{ 且 } y < 0. \end{cases} \quad (9)$$

(3) 把手坐标的修正

如图 2 中局部放大图所示, 点 D 是位于曲率圆上的点, 不严格处于螺线上。取点 D 在极坐标系下极角 θ_2 , 由方程(1)映射至螺线上

$$\theta_{i+1} = \theta_2, \quad (10)$$

得到严格处于螺线上、与 A_i 距离近似为 d_i 的把手 A_{i+1} 。

4. 龙身、龙尾速度模型的建立

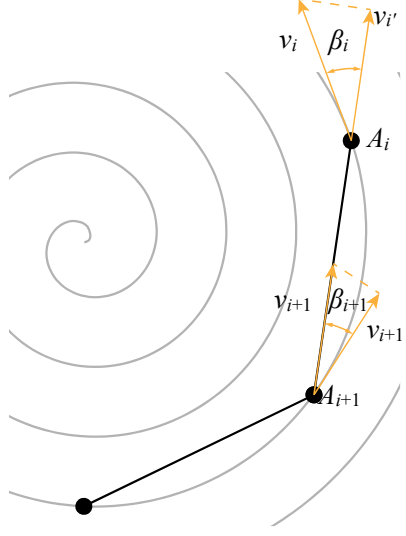


图 3 龙身速度递推过程示意

如图 3 所示，考虑已知 A_i 把手的绝对速度 v_i ，递推 A_{i+1} 把手绝对速度 v_{i+1} 。将 v_i 沿板凳方向投影，得

$$v_i' = v_i \cos \beta_i。$$

这一速度沿板凳方向传递时保持不变，即

$$v_{i+1}' = v_i'，$$

可得 A_{i+1} 的绝对速度为

$$v_{i+1} = \frac{v_{i+1}'}{\cos \beta_{i+1}} = v_i \frac{\cos \beta_i}{\cos \beta_{i+1}}。 \quad (11)$$

龙头始终保持 1 m/s 匀速率运动，可由此不断递推至龙尾。

5. 舞龙队运动模型给出

综上所述，建立舞龙队运动模型：

$$\left\{ \begin{array}{l} \text{龙头} \left\{ \begin{array}{l} \theta = \theta(s_0 - vt) \\ v = 1 \text{ m/s} \end{array} \right. \\ \text{龙身与龙尾} \left\{ \begin{array}{l} \theta = \begin{cases} \arctan \frac{y_D}{x_D}, & x > 0 \text{ 且 } y > 0, \\ \arctan \frac{y_D}{x_D} + \pi, & x < 0 \text{ 且 } y > 0, \\ \arctan \frac{y_D}{x_D} + 2\pi, & x < 0 \text{ 且 } y < 0, \\ \arctan \frac{y_D}{x_D} + 2\pi, & x > 0 \text{ 且 } y < 0. \end{cases} \\ v_{i+1} = v_i \frac{\cos \beta_i}{\cos \beta_{i+1}} \end{array} \right. \end{array} \right. \quad (12)$$

5.1.2 模型求解

1. 龙头位置的求解

超越方程(2)的反函数较难求得, 但考虑(2)式严格单调递增, 因此考虑采用二分法获得给定 s 所对应的极角 θ , 算法如下。

二分算法

Step1. 考虑龙头位置始终不会在第 16 圈之外, 因此设定初始搜索范围为 $\theta \in [0, 16 \times 2\pi]$ 。分别记左右界为 $\theta_{\text{left}}, \theta_{\text{right}}$ 。

Step2. 计算 θ 范围中点 $\theta_{\text{middle}} = \frac{\theta_{\text{left}} + \theta_{\text{right}}}{2}$ 对应路程 $s = s(\theta_{\text{middle}})$ 。若 s 小于目标

值, 则极角位于区间 $(\theta_{\text{middle}}, \theta_{\text{right}})$ 内, 故以 θ_{middle} 代替 θ_{left} ; 否则极角位于区间

$(\theta_{\text{left}}, \theta_{\text{middle}})$ 内, 以故以 θ_{middle} 代替 θ_{right} 。

Step3. 重复 step2, 直至 θ 范围小于 10^{-8}rad , 以保证最终结果的 6 位小数精度。

Step4. 将此时 θ 范围中点输出, 该值即为(3)式中 s 对应的 θ 。

2. 龙身、龙尾位置与速度的递推

初始时刻的龙头位置为 $\theta_0 = 32\pi$, 由(9)、(10)递推初始时刻龙身及龙尾各把手位置。龙头移动后位置变更为 $\theta_t = \theta(s_0 - vt)$, 重新递推龙身及龙尾各把手位置。

得到某时刻各把手位置后, 由于龙头速度恒为 1 m/s , 由(11)式推出此时刻龙身和龙尾各把手速度。

5.1.3 结果分析

1. 舞龙队位置的结果及其分析

求解得到龙头、龙身、龙尾各把手位置与速度, 已写入支撑材料 result1.xlsx 中。其中部分把手、部分时刻的位置如表 1 所示。

表 1 龙头、龙身、龙尾部分把手在对应时刻的位置

	0 s	60 s	120 s	180 s	240 s	300 s
龙头 $x(\text{m})$	8.800000	5.799209	-4.084887	-2.963608	2.594494	4.420274
龙头 $y(\text{m})$	0.000000	-5.771092	-6.304478	6.094780	-5.356742	2.320429
第 1 节龙身 $x(\text{m})$	8.363796	7.456808	-1.445310	-5.237280	4.821480	2.458731
第 1 节龙身 $y(\text{m})$	2.826627	-3.440292	-7.405916	4.359435	-3.561609	4.402916
第 51 节龙身 $x(\text{m})$	-9.518807	-8.686509	-5.543841	2.889233	5.981164	-6.301684
第 51 节龙身 $y(\text{m})$	1.340636	2.539473	6.377357	7.249791	-3.826003	0.461970
第 101 节龙身 $x(\text{m})$	2.914762	5.687972	5.363087	1.900728	-4.914997	-6.240583
第 101 节龙身 $y(\text{m})$	-9.918089	-8.000786	-7.556839	-8.471200	-6.381745	3.931588
第 151 节龙身 $x(\text{m})$	10.861561	6.681293	2.387062	1.002749	2.961989	7.037580
第 151 节龙身 $y(\text{m})$	1.829782	8.135393	9.727842	9.425030	8.400954	4.398190
第 201 节龙身 $x(\text{m})$	4.553979	-6.620914	-10.627481	-9.286601	-7.454619	-7.454911
第 201 节龙身 $y(\text{m})$	10.725605	9.024667	1.357856	-4.249176	-6.183836	-5.268804
龙尾 (后) $x(\text{m})$	-5.304290	7.365808	10.974203	7.381877	3.237135	1.778398
龙尾 (后) $y(\text{m})$	-10.677168	-8.796959	0.845572	7.494392	9.470714	9.302498

绘制每分钟“板凳龙”各把手位置，如图 4 所示。由图可见，“板凳龙”沿顺时针向内盘入，各把手均位于螺线上。

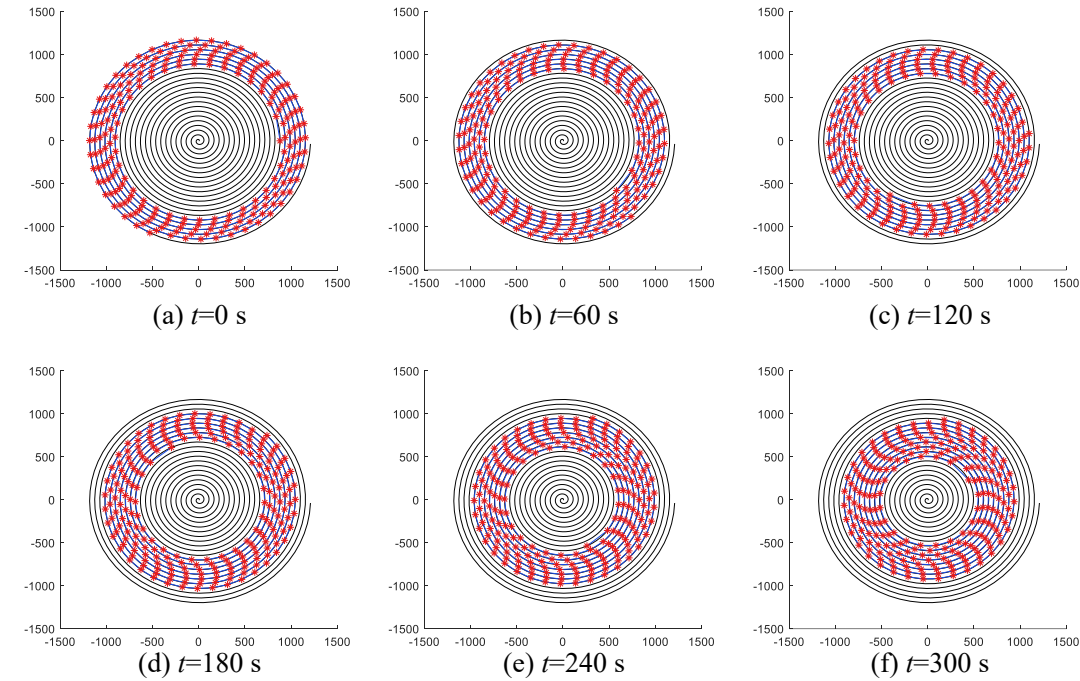


图 4 “板凳龙”每分钟运动情况

2. 舞龙队速度的结果及其分析

表 2 龙头、龙身、龙尾部分把手每分钟的速度

	0 s	60 s	120 s	180 s	240 s	300 s
龙头(m/s)	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
第 1 节龙身(m/s)	1.003046	0.999961	0.999945	0.999917	0.999859	0.999709
第 51 节龙身(m/s)	0.995223	1.014532	1.000221	1.016792	1.026401	0.976750
第 101 节龙身(m/s)	0.996448	1.026450	0.994626	0.998896	1.031478	0.974576
第 151 节龙身(m/s)	1.002137	1.022767	0.997579	0.996819	1.040970	0.969392
第 201 节龙身(m/s)	1.001042	1.019727	0.990896	0.990757	1.039508	0.976355
龙尾（后）(m/s)	1.000519	1.023604	0.990183	0.993794	1.041326	0.983021

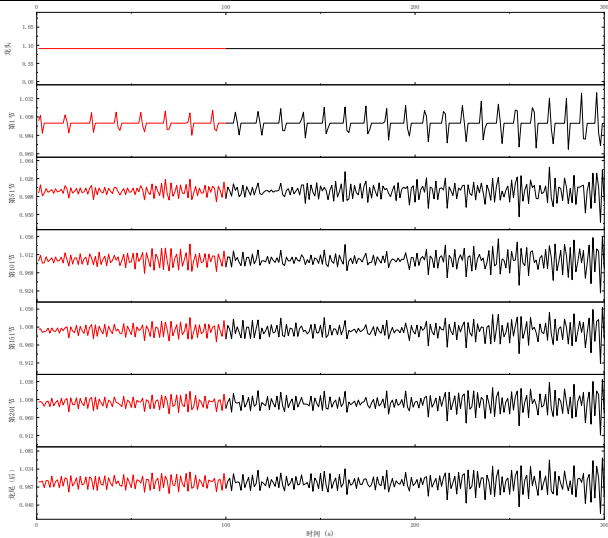


图 5 部分把手绝对速度变化曲线

绘制龙头、龙身及龙尾 1、51、101、151、201 节的速度随时间变化曲线，如图 5 所示。观察发现龙身各把手速度在 1 m/s 附近波动，且随时间增加，波动幅度增加。这是因为舞龙队盘入过程中，曲率半径的变化增快，导致速度变化增快。

5.1.4 模型检验

龙身、龙尾位置递推模型中近似下一把手位于上一把手的曲率圆上，这会导致“板凳龙”中各节板凳长度与实际情况存在误差。计算每分钟龙身位置结果中，相邻把手之间的直线距离，绘得箱线图，图 6 所示。

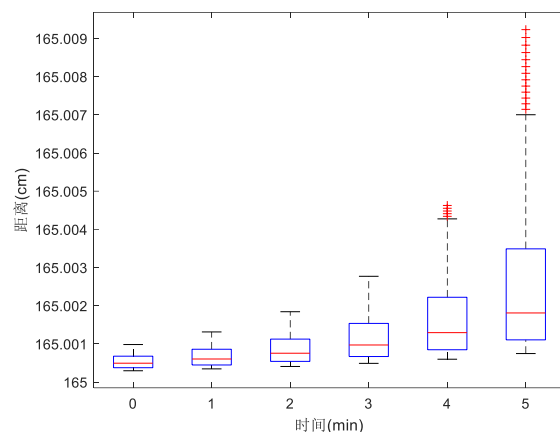


图 6 每分钟龙身把手直线距离箱线图

由图可见，各把手距离与实际距离（165 cm）误差在 $6 \times 10^{-3}\%$ 内，最大偏差仅为 0.01 mm，近似合理。

5.2 问题二模型的建立与求解

5.2.1 模型建立

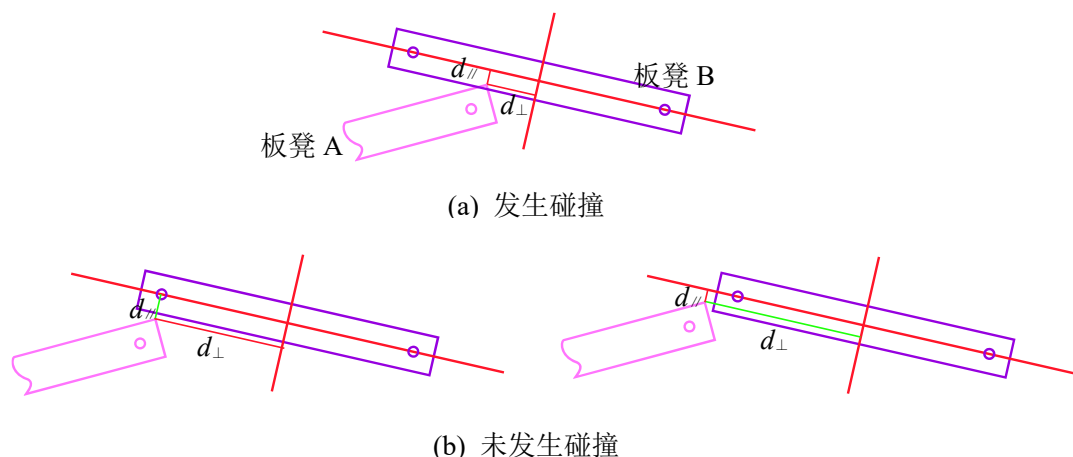


图 7 碰撞或未碰撞的三种代表情形

碰撞初次发生时，必定存在一板凳（板凳 A）某角进入另一板凳（板凳 B）长方形区域的情形，如图 7 所示。这一过程可描述为，存在板凳 A 四角中的某一角，使得该点与板凳 B 轴线的距离小于板凳宽度的一半，且该点与板凳 B 中垂线的距离小于板凳长度的一半。其余形式的碰撞发生前，必将经历图 7(a)的情形。因此，这两个条件中任何一点未满足，认为两板凳均不发生碰撞。

1. 板凳 A 顶点坐标

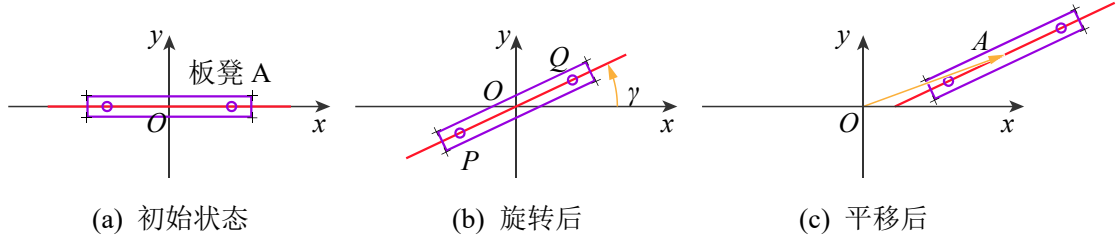


图 8 板凳 A 坐标变换过程示意图

现对板凳 A 四个顶点进行坐标变换。

(a) 初始状态下，板凳 A 横置，中心位于原点，如图 8(a)所示。此时板凳 A 四角坐标分别为

$$\begin{pmatrix} m \\ n \end{pmatrix}, \begin{pmatrix} m \\ -n \end{pmatrix}, \begin{pmatrix} -m \\ n \end{pmatrix}, \begin{pmatrix} -m \\ -n \end{pmatrix}, \quad (13)$$

其中 m 与 n 分别为板凳长与宽的一半。

(b) 对顶点坐标进行**旋转变换**。旋转角度由实际状态下板凳两把手斜率决定。设两把手坐标分别为 $(x_{AP}, y_{AP}), (x_{AQ}, y_{AQ})$ ，则逆时针旋转角度为

$$\gamma = \arctan \frac{y_{AP} - y_{AQ}}{x_{AP} - x_{AQ}}。$$

旋转矩阵

$$\mathbf{T} = \begin{pmatrix} \cos \gamma & -\sin \gamma \\ \sin \gamma & \cos \gamma \end{pmatrix}$$

描述逆时针旋转 γ 角过程。旋转后四点坐标 (x', y') 为

$$\begin{pmatrix} x'_1 & x'_2 & x'_3 & x'_4 \\ y'_1 & y'_2 & y'_3 & y'_4 \end{pmatrix} = \mathbf{T} \begin{pmatrix} m & m & -m & -m \\ n & -n & n & -n \end{pmatrix}。$$

(c) 对旋转后顶点坐标进行**平移**。取板凳 A 中点坐标为 (x_A, y_A) ，则平移后得到实际状态下板凳顶点坐标分别为

$$\begin{pmatrix} x''_i \\ y''_i \end{pmatrix} = \begin{pmatrix} x'_i \\ y'_i \end{pmatrix} + \begin{pmatrix} x_A \\ y_A \end{pmatrix}, \quad i = 1, 2, 3, 4。$$

综上所述，板凳顶点的实际坐标为

$$\begin{pmatrix} x''_i \\ y''_i \end{pmatrix} = \begin{pmatrix} \cos \gamma & -\sin \gamma \\ \sin \gamma & \cos \gamma \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} x_A \\ y_A \end{pmatrix}。 \quad (14)$$

2. 板凳 B 轴线与中垂线方程

设板凳 B 两把手坐标为 $(x_{BP}, y_{BP}), (x_{BQ}, y_{BQ})$ ，则板凳 B 轴线可由两点式方程描述：

$$B_{//} : \frac{y - y_{BP}}{y_{BQ} - y_{BP}} = \frac{x - x_{BP}}{x_{BQ} - x_{BP}}。$$

计算板凳 B 轴线斜率，并转化为与之垂直的中垂线斜率，再利用板凳 B 中点，可得板凳 B 中垂线方程：

$$B_{\perp} : x + \frac{y_{BP} - y_{BQ}}{x_{BP} - x_{BQ}} y - x_B - \frac{y_{BP} - y_{BQ}}{x_{BP} - x_{BQ}} y_B = 0 ,$$

其中 (x_B, y_B) 为板凳 B 中心坐标。

3. 判断是否碰撞

由点到直线距离公式

$$d = \frac{|Ax_i'' + By_i'' + C|}{\sqrt{A^2 + B^2}}, \quad i = 1, 2, 3, 4$$

可分别计算四点到轴线、垂线的距离 $d_{//}$ 、 d_{\perp} 。判断是否发生碰撞：

$$\begin{cases} d_{//} \leq n \text{ 且 } d_{\perp} \leq m, & \text{发生碰撞,} \\ \text{否则,} & \text{未发生碰撞.} \end{cases}$$

4. 板凳碰撞模型给出

综上所述，当同时满足

$$\begin{cases} d_{//} = \frac{|A_{//}x_i'' + B_{//}x_i'' + C_{//}|}{\sqrt{A_{//}^2 + B_{//}^2}} \leq n, \\ d_{\perp} = \frac{|A_{\perp}x_i'' + B_{\perp}x_i'' + C_{\perp}|}{\sqrt{A_{\perp}^2 + B_{\perp}^2}} \leq m, \end{cases} \quad (15)$$

时，视为存在两板凳碰撞的情况。式中

$$A_{//} = y_{BQ} - y_{BP}, \quad B_{//} = -(x_{BQ} - x_{BP}), \quad C_{//} = (x_{BQ} - x_{BP})y_{BP} - (y_{BQ} - y_{BP})x_{BP},$$

$$A_{\perp} = 1, \quad B_{\perp} = \frac{y_{BP} - y_{BQ}}{x_{BP} - x_{BQ}}, \quad C_{\perp} = -x_B - \frac{y_{BP} - y_{BQ}}{x_{BP} - x_{BQ}} y_B.$$

“板凳龙”运动约束同(12)式。

5.2.2 模型求解

考虑到时间步长过小可能导致发生短时间的碰撞而未被检测，因此取时间步长为 $\Delta t = 0.01 \text{ s}$ 提高计算准确性。

由于螺线内部曲率较大，更容易发生碰撞。因此，针对每一时刻，视龙头及其后部 3 个龙身作为板凳 A，将板凳 A 后续 2~100 个板凳视为板凳 B 进行搜索，具体过程如下。

搜索算法

Step1. 赋初值 $t = 0 \text{ s}$ ，给定螺距为 $b = 55 \text{ cm}$ 。

Step2. 使用(12)式确定 t 时刻龙头位置，并完成龙身及龙尾位置递推

Step3. 基于(15)式判断是否发生碰撞。若未发生碰撞，则时间 t 自增 0.01 s ，否则返回上一时刻各把手位置，并使用(12)式计算绝对速度。

5.2.3 结果分析

求解得到恰好未碰撞时刻为 $t=426.78\text{ s}$ ，此时龙头距中心 1.65 m，此时舞龙队的位置与速度已写入 result2.xlsx 中。

1. 舞龙队碰撞前位置及其分析

表 3 龙头、龙身、龙尾部分把手碰撞前位置

	$x(\text{m})$	$y(\text{m})$
龙头	1.652292	0.078572
第 1 节龙身	-0.864211	1.613696
第 51 节龙身	-0.988294	-4.125183
第 101 节龙身	-3.791322	4.257858
第 151 节龙身	-6.672359	1.570789
第 201 节龙身	-0.196422	7.837232
龙尾（后）	8.126038	-1.339494

与问题一对照，龙头在 426.78 s 时更接近中心，与实际情况相符。
观察碰撞发生过程，如图 9 所示。龙头后第 1 节龙身与第 8 节龙身发生碰撞，具体过程如图 9(b)所示。由图可见，板凳 A 上方与板凳 B 发生碰撞。

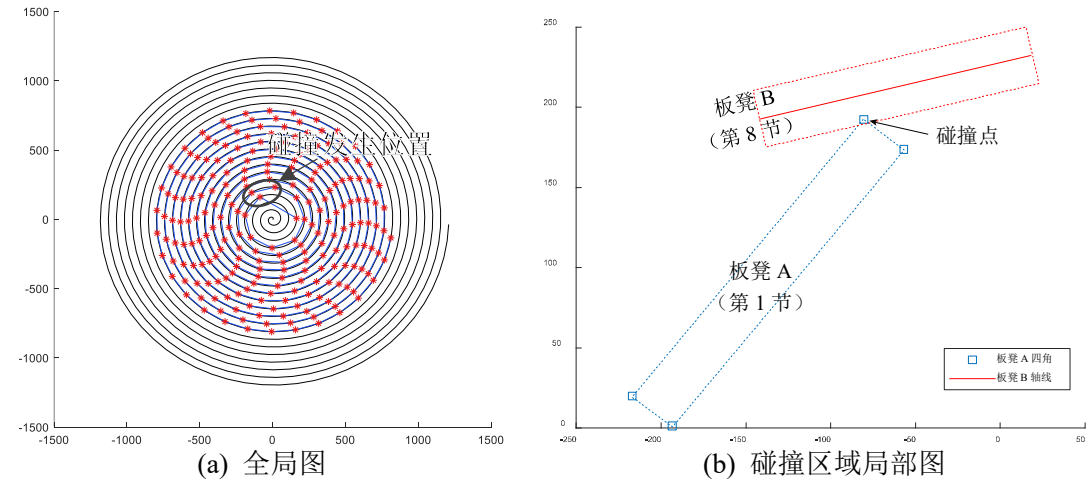


图 9 碰撞发生时“板凳龙”位置情况

2. 舞龙队碰撞前速度及其分析

表 4 龙头、龙身、龙尾部分把手碰撞前速度

	$v(\text{m/s})$
龙头	1.000000
第 1 节龙身	0.528180
第 51 节龙身	0.572559
第 101 节龙身	0.573862
第 151 节龙身	0.578043
第 201 节龙身	0.586320
龙尾（后）	0.589241

“板凳龙”碰撞前速度如表 4 所示。与问题一对照，发现龙头更靠近中心时，自龙头向龙身和龙尾传递的速度变小。这是因为，更靠近中心时，龙头与第 1 节龙身轴线夹角增大，(11)式中速度传递系数减小。

5.3 问题三模型的建立与求解

5.3.1 模型建立

1. 决策变量

在其余条件不变的情况下，“板凳龙”轨迹由螺距完全确定，因此等距螺线的螺距 b 是问题三模型的决策变量。

2. 目标函数

问题需要确定满足条件的最小螺距，因此最小螺距 b 是模型的目标函数：

$$\min b。$$

3. 约束条件

(1) 不发生碰撞

在盘入半径为 4.5 m 的掉头空间前，“板凳龙”不能发生碰撞。沿用问题二模型，即当 $x_0''^2 + y_0''^2 \geq (4.5 \text{ m})^2$ 时，有

$$\begin{cases} d_{//} = \frac{|A_{//}x_i'' + B_{//}x_i'' + C_{//}|}{\sqrt{A_{//}^2 + B_{//}^2}} \leq n, \\ d_{\perp} = \frac{|A_{\perp}x_i'' + B_{\perp}x_i'' + C_{\perp}|}{\sqrt{A_{\perp}^2 + B_{\perp}^2}} \leq m. \end{cases} \quad (16)$$

(2) 龙头、龙身运动的系统约束

龙头、龙身的具体运动轨迹满足问题一的模型，由此构成系统约束。

4. 最小螺距优化模型给出

综上所述，建立非线性优化模型：

$$\begin{aligned} & \min b \\ & \text{s.t.} \begin{cases} x_0''^2 + y_0''^2 \geq (4.5 \text{ m})^2, \\ d_{//} = \frac{|A_{//}x_i'' + B_{//}x_i'' + C_{//}|}{\sqrt{A_{//}^2 + B_{//}^2}} \leq n, \\ d_{\perp} = \frac{|A_{\perp}x_i'' + B_{\perp}x_i'' + C_{\perp}|}{\sqrt{A_{\perp}^2 + B_{\perp}^2}} \leq m. \end{cases} \end{aligned}$$

“板凳龙”运动约束同(12)式。

5.3.2 模型求解

当螺距缩小到一定程度时，会因龙身碰撞而不再满足约束条件，即目标函数具有单调性。因此，使用二分法进行求解，具体算法如下。

二分算法

Step1. 经问题二计算发现, $b=55\text{ cm}$ 时能够进入掉头区域, 同时 $b<30\text{cm}$ 时必定发生碰撞。因此, 设定初始搜索范围为 $b \in [30\text{ cm}, 55\text{ cm}]$ 。分别记左右界为

$b_{\text{left}}, b_{\text{right}}$ 。

Step2. 判断 b 范围中点 $b_{\text{middle}} = \frac{b_{\text{left}} + b_{\text{right}}}{2}$ 是否满足(16)式约束条件。若不满足约

束条件, 因螺距过小而发生了碰撞, 故以 b_{middle} 代替 b_{left} ; 否则以 b_{middle} 代替 b_{right} 。

Step3. 重复 step2, 直至 b 搜索范围小于 10^{-4} cm 。

Step4. 将此时 b 范围中点输出, 该值即为满足约束条件的最小螺距。

5.3.3 结果分析

1. 求解结果

求解得到的**最小螺距为 42.06598 cm**, 此时龙头距螺线中心 4.499942 m。舞龙队具体位置图 10, 虚线区域示掉头空间。

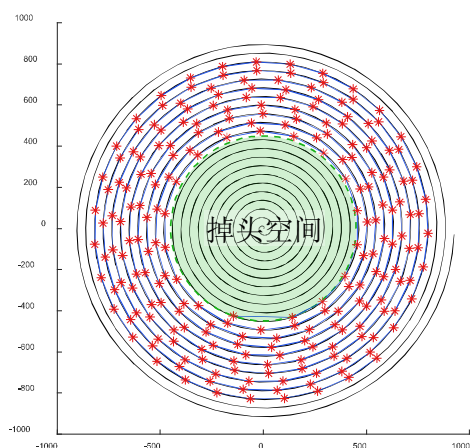


图 10 恰进入掉头空间时“板凳龙”位置

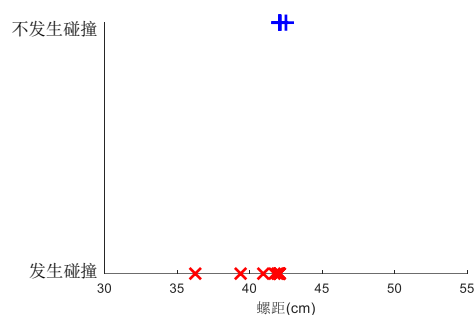


图 11 二分搜索历程

2. 结果分析

螺距的减小易导致板凳间的碰撞, 且这一碰撞更容易发生在靠近螺线中心位置。因此, 适当增加掉头空间, 或增加螺距可以使板凳间不发生碰撞。

5.4 问题四模型的建立与求解

5.4.1 模型建立

1. 掉头路径唯一性说明

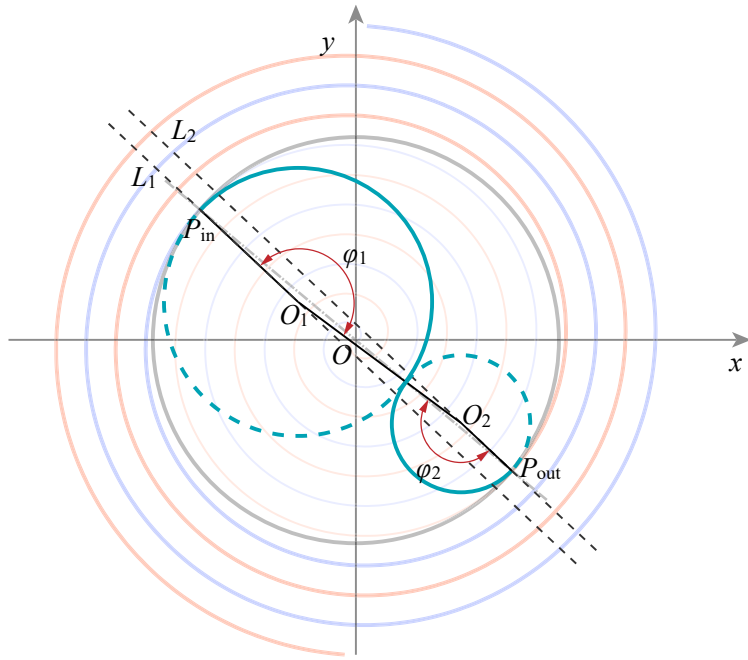


图 12 符合要求的掉头曲线示意图

掉头曲线要求与盘入、盘出螺线相切，如图 12 所示。由于等距螺线任一点处法线均不经过原点，因此两段圆弧圆心角 φ_1 与 φ_2 均小于 180° ，圆心与入点（出点）连成的直线 L_1 与 L_2 平行。为满足 $R_1/R_2 = 2$ ，即要求图中满足关系

$$P_{in}O_1 : O_1O_2 : O_2P_{out} = 2 : 3 : 1。 \quad (17)$$

再次调整两圆半径时，(17)式都不再成立。

综上所述，当掉头空间半径确定时，掉头曲线唯一确定，无法变短。

2. 掉头路径龙头位置模型

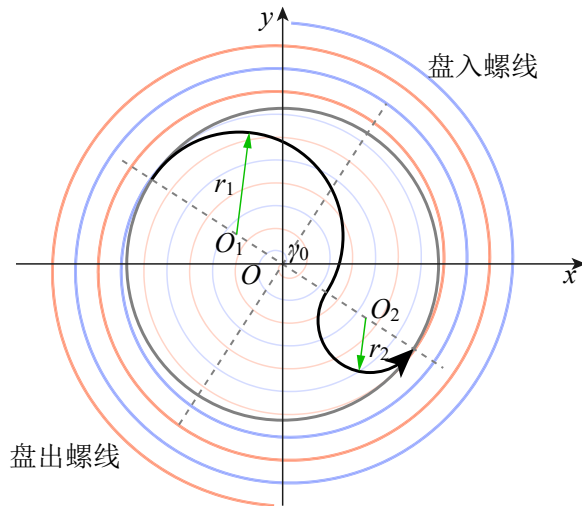


图 13 龙头自然坐标系示意图

当龙头进入掉头曲线时，取掉头曲线起点为原点，建立自然坐标系。龙头位置可通过路程 s 唯一确定：

$$\begin{cases} s < \pi r_1, & \text{龙头在大圆弧内,} \\ s \geq \pi r_1, & \text{龙头在小圆弧内.} \end{cases}$$

- 龙头在大圆弧的情形

龙头位置与大圆弧的相对极角

$$\theta_1 = \frac{s}{\pi r_1} \pi, \quad (18)$$

可知龙头在直角坐标系下的坐标为

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos \gamma_0 & -\sin \gamma_0 \\ \sin \gamma_0 & \cos \gamma_0 \end{pmatrix} \begin{pmatrix} r \cos \theta_1 \\ r \sin \theta_1 + R/3 \end{pmatrix}, \quad (19)$$

式中, γ_0 为 S 曲线逆时针旋转角度, R 为掉头空间半径。

- 龙头在小圆弧的情形

与(19)式类似, 龙头在小圆弧时的直角坐标为

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos \gamma_0 & -\sin \gamma_0 \\ \sin \gamma_0 & \cos \gamma_0 \end{pmatrix} \begin{pmatrix} r \cos \theta_2 \\ r \sin \theta_2 - 2R/3 \end{pmatrix}, \quad (20)$$

其中

$$\theta_2 = \frac{(s - \pi r_1)}{\pi r_2} \pi.$$

3. 掉头路径龙身、龙尾位置模型

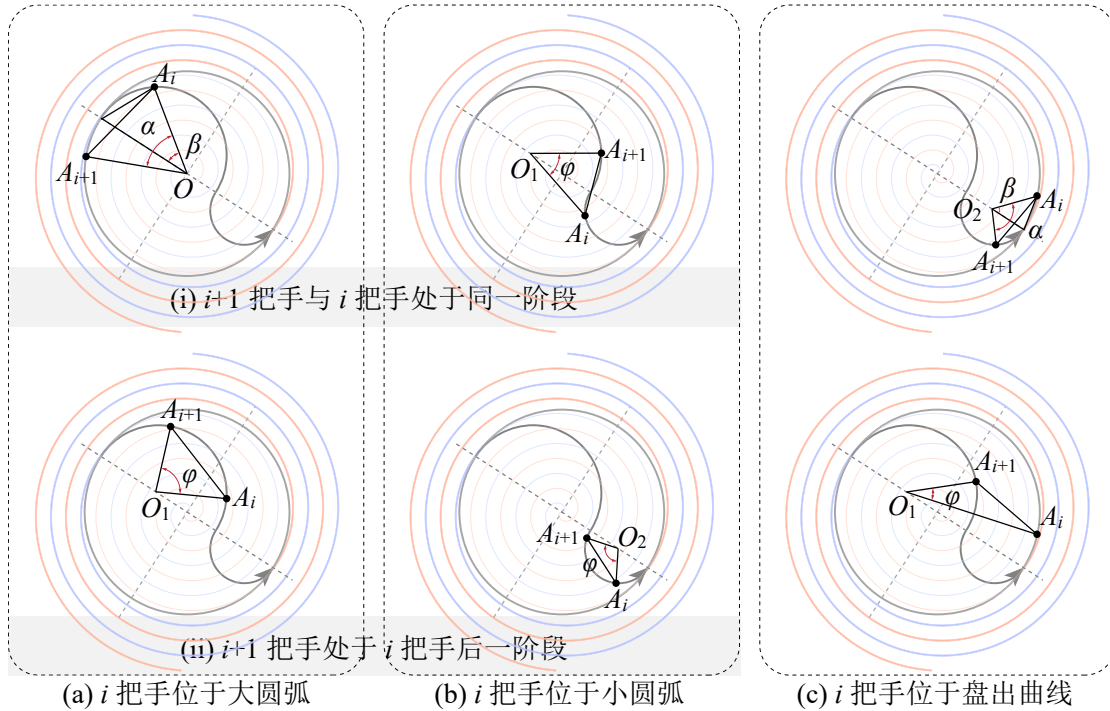


图 14 掉头路径内把手递推的六种情形

(a) i 把手位于大圆弧内

- 当 $i+1$ 把手与 i 把手处于同一阶段时, 如图 14(a i)所示。有关系

$$\alpha = \arccos \frac{R^2 + x_i^2 + y_i^2 - d^2}{2R\sqrt{x_i^2 + y_i^2}},$$

$$\beta = \arccos \frac{R^2 + x_i^2 + y_i^2 - [(x_i - x_{in})^2 + (y_i - y_{in})^2]}{2R\sqrt{x_i^2 + y_i^2}}$$

成立。 $i+1$ 把手近似坐标由入点坐标逆时针旋转 $(\alpha - \beta)$ 得到:

$$\begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} \cos(\alpha - \beta) & -\sin(\alpha - \beta) \\ \sin(\alpha - \beta) & \cos(\alpha - \beta) \end{pmatrix} \begin{pmatrix} x_{in} \\ y_{in} \end{pmatrix}. \quad (21)$$

$i+1$ 把手精确坐标由近似坐标极角向盘入螺线映射后得到。

● 当 $i+1$ 把手处于 i 把手后一阶段时, 如图 14(a ii)所示。此时,

$$\varphi = 2 \arcsin \frac{d}{2r_1}, \quad (22)$$

对 $\overrightarrow{OA_i}$ 逆时针旋转 φ , 平移后得到 $i+1$ 把手精确坐标:

$$\begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix} \begin{pmatrix} x_i - x_{o1} \\ y_i - y_{o1} \end{pmatrix} + \begin{pmatrix} x_{o1} \\ y_{o1} \end{pmatrix}. \quad (23)$$

(b) i 把手位于小圆弧内

针对情况(i), 转角为

$$\varphi = \arccos \frac{r_1^2 + (x_i - x_{o1})^2 + (y_i - y_{o1})^2 - d^2}{2r_1\sqrt{(x_i - x_{o1})^2 + (y_i - y_{o1})^2}}, \quad (24)$$

将 $\overrightarrow{OA_i}$ 逆时针旋转 φ 后取单位向量, 并乘以 r_1 , 得 $i+1$ 把手坐标

$$\overrightarrow{r_{O_i A_{i+1}}} = \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix} \begin{pmatrix} x_i - x_{o1} \\ y_i - y_{o1} \end{pmatrix}, \quad (25)$$

$$\begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = r_1 \frac{\overrightarrow{r_{O_i A_{i+1}}}}{|\overrightarrow{r_{O_i A_{i+1}}}|}. \quad (26)$$

针对情况(ii), 与(22)式类似:

$$\varphi = \varphi = 2 \arcsin \frac{d}{2r_2},$$

$$\begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix} \begin{pmatrix} x_i - x_{o2} \\ y_i - y_{o2} \end{pmatrix} + \begin{pmatrix} x_{o2} \\ y_{o2} \end{pmatrix}. \quad (27)$$

(c) i 把手位于盘出曲线内

$i+1$ 把手在小圆弧内时, 与情况(a i)类似, 有

$$\alpha = \arccos \frac{r_1^2 + R^2 - d^2}{2r_1 R},$$

$$\beta = \arccos \frac{r_2 + R^2 - [(x_i - x_{\text{out}})^2 + (y_i - y_{\text{out}})^2]}{2r_2 R},$$

$$\begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} \cos(\alpha - \beta) & -\sin(\alpha - \beta) \\ \sin(\alpha - \beta) & \cos(\alpha - \beta) \end{pmatrix} \begin{pmatrix} x_{\text{out}} \\ y_{\text{out}} \end{pmatrix}. \quad (28)$$

$i+1$ 把手位于大圆弧内时, 与情况(b i)完全一致, 算式同(24)、(25)、(26)。

4. 盘出曲线“板凳龙”位置模型

盘出过程是盘入过程的逆过程, 求解(9)式时保留顺时针方向点。同时, 取各点横纵坐标相反数作为把手坐标。

5. 掉头路径龙身、龙尾速度模型

在问题一模型(11)的基础上考虑掉头路径中圆弧的切线斜率

$$k = -\frac{x_i - x_o}{y_i - y_o},$$

可得掉头路径龙身速度模型

$$v_{i+1} = \frac{v_{i+1}'}{\cos \beta_{i+1}} = v_i \frac{\cos \beta_i}{\cos \beta_{i+1}}.$$

6. 舞龙队运动模型给出

i 与 $i+1$ 把手同时在盘入、盘出曲线时的模型与问题一模型(12)一致。在掉头区域内时, 舞龙队运动模型为

$$\left\{ \begin{array}{l} \text{龙头} \left\{ \begin{array}{l} \begin{cases} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos \gamma_0 & -\sin \gamma_0 \\ \sin \gamma_0 & \cos \gamma_0 \end{pmatrix} \begin{pmatrix} r \cos \theta_1 \\ r \sin \theta_1 + R/3 \end{pmatrix}, & s < \pi r_1, \\ \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos \gamma_0 & -\sin \gamma_0 \\ \sin \gamma_0 & \cos \gamma_0 \end{pmatrix} \begin{pmatrix} r \cos \theta_2 \\ r \sin \theta_2 - 2R/3 \end{pmatrix}, & s \geq \pi r_1. \end{cases} \\ v \equiv 1 \text{ m/s} \end{array} \right. \\ \text{龙身与龙尾} \left\{ \begin{array}{l} \begin{cases} \begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} \cos(\alpha - \beta) & -\sin(\alpha - \beta) \\ \sin(\alpha - \beta) & \cos(\alpha - \beta) \end{pmatrix} \begin{pmatrix} x_{\text{in}} \\ y_{\text{in}} \end{pmatrix}, & \text{(a i)} \\ \begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix} \begin{pmatrix} x_i - x_{o1} \\ y_i - y_{o1} \end{pmatrix} + \begin{pmatrix} x_{o1} \\ y_{o1} \end{pmatrix}, & \text{(a ii)} \end{cases} \\ \begin{cases} \begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = r_1 \frac{\overrightarrow{r_{O_1 A_{i+1}}}}{|r_{O_1 A_{i+1}}|}, & \text{(b i)} \\ \begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix} \begin{pmatrix} x_i - x_{o2} \\ y_i - y_{o2} \end{pmatrix} + \begin{pmatrix} x_{o2} \\ y_{o2} \end{pmatrix}, & \text{(b ii)} \end{cases} \\ \begin{cases} \begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} \cos(\alpha - \beta) & -\sin(\alpha - \beta) \\ \sin(\alpha - \beta) & \cos(\alpha - \beta) \end{pmatrix} \begin{pmatrix} x_{\text{out}} \\ y_{\text{out}} \end{pmatrix}, & \text{(c)} \\ v_{i+1} = v_i \frac{\cos \beta_i}{\cos \beta_{i+1}} \end{cases} \end{array} \right. \quad (29)$$

5.4.2 模型求解

修改螺距为 1.7 m，按时间递推出龙头位置，再由龙头位置递推龙身、龙尾位置，具体算法如下。

求解算法
Step1. 判断龙头是否首次进入掉头曲线，若是，记录当前时间；若不是，且龙头已经过掉头曲线，则判断当前时间是否经过 100 s，若是，则算法结束。
Step2. 按(4)、(19)、(20)式求出龙头位置。依据不同情况按(21)、(23)、(26)、(27)、(28)式逐步递推至龙尾位置。
Step3. 返回 Step1。

5.4.3 结果分析

求解得到进入掉头曲线前后 100 s 内每秒“板凳龙”把手位置与速度，已写入 result4.xlsx 中。

1. “板凳龙”位置及其分析

表 5 “板凳龙”进入掉头曲线前后 0、50、100 s 部分把手位置

	-100 s	-50 s	0 s	50 s	100 s
龙头 $x(m)$	7.954427	6.704027	-3.106462	1.316229	-3.173126
龙头 $y(m)$	3.284750	1.441806	-3.233923	6.179599	7.542452
第 1 节龙身 $x(m)$	6.516737	5.638824	-0.543685	3.848793	-0.364190
第 1 节龙身 $y(m)$	5.757501	4.096900	-4.611306	4.852540	8.078984
第 51 节龙身 $x(m)$	-3.106462	1.316229	-3.173126	1.316229	-3.173126
第 51 节龙身 $y(m)$	-3.233923	6.179599	7.542452	6.179599	7.542452
第 101 节龙身 $x(m)$	-0.543685	3.848793	-0.364190	3.848793	-0.364190
第 101 节龙身 $y(m)$	-4.611306	4.852540	8.078984	4.852540	8.078984
第 151 节龙身 $x(m)$	1.999693	-2.155714	2.076457	2.495565	2.076457
第 151 节龙身 $y(m)$	-7.892757	-5.898628	4.044824	4.782984	4.044824
第 201 节龙身 $x(m)$	3.454366	-7.279693	-6.669503	8.099852	6.800684
第 201 节龙身 $y(m)$	9.952152	5.581260	3.474073	2.866222	0.017783
龙尾（后） $x(m)$	-6.601826	-5.061940	8.778157	-8.944531	-7.357491
龙尾（后） $y(m)$	10.585993	-10.158619	-4.917769	6.204708	-6.067428

经检验，各把手均位于行进路线上，进入掉头曲线后 0、50、100 s 舞龙队位置如图 15 所示。

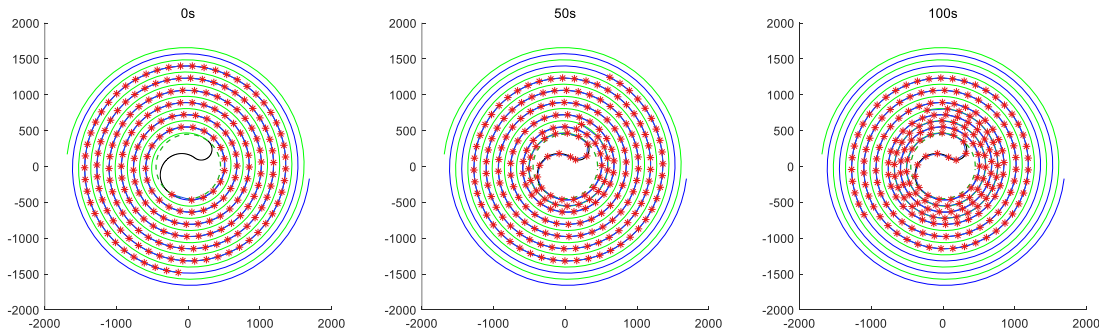


图 15 0 s、50 s、100 s 舞龙队位置图

2. “板凳龙”速度及其分析

表6 “板凳龙”进入掉头曲线前后 0、50、100 s 部分把手速度

	-100 s	-50 s	0 s	50 s	100 s
龙头(m/s)	1.000000	1.000000	1.000000	1.000000	1.000000
第 1 节龙身(m/s)	0.995408	1.000362	1.000124	1.000362	1.000124
第 51 节龙身(m/s)	1.003322	1.329024	0.991262	4.381630	0.991262
第 101 节龙身(m/s)	0.997369	1.319104	2.144635	4.419380	7.646170
第 151 节龙身(m/s)	1.000729	1.314188	2.118134	4.402172	7.697211
第 201 节龙身(m/s)	1.002221	1.315979	2.139530	4.405507	7.658535
龙尾(后)(m/s)	1.000695	1.318616	2.131418	4.397156	7.654938

观察发现,进入掉头区域前,“板凳龙”把手速度在龙头速度上下波动,且波动幅度较小。进入调头曲线后,由于掉头曲线曲率增加,导致速度传递系数增大,100 s 时存在把手速度超出龙身速度 7 倍。

5.5 问题五模型的建立与求解

5.5.1 模型建立

1. 决策变量

舞龙队路径确定,龙头行进速度 v 是此模型的决策变量。

2. 目标函数

问题需要解出满足条件的最大龙头速度,因此龙头最大行进速度 v 是模型的目标函数:

$$\max v。$$

3. 约束条件

(1) 各把手行进速度均不超过 2m/s, 即

$$v_i(t) \leq 2, i = 1, 2 \dots 224, t \geq 0。$$

(2) 龙头、龙身、龙尾运动满足问题四模型,组成此模型的系统约束条件。

4. 最大龙头速度模型给出

由此可得最大龙头速度优化模型:

$$\max v \quad (30)$$

$$\text{s.t. } v_i(t) \leq 2, i = 1, 2 \dots 224, t \geq 0$$

“板凳龙”运动约束同(29)式。

5.5.2 模型求解

与问题三求解类似,采用二分法对模型(30)进行求解。

考虑到龙头速度超过 2 m/s 时,由于螺线内速度波动,必定存在速度超过 2 m/s 的龙身或龙尾把手。因此,设置龙头速度搜索范围为 $v \in (0, 2 \text{ m/s})$ 。

时间步长的给定值过大易导致部分把手速度超过 2 m/s 的情形未被发现,给定值过小会导致运算量增大。本文对时间步长取 50、10、5、1、0.5、0.1、0.01 s 时分别求解。

5.5.3 结果分析

表 7 不同时间步长下搜索得到的龙头最大允许速度

时间步长	50 s	10 s	5 s	1 s	0.5 s	0.1 s	0.01 s
最大允许速度(m/s)	0.3249	0.1937	0.1875	0.1875	0.1875	0.1875	0.1875

求解得到不同时间步长下龙头最大允许速度如表 7 所示。观察发现，最大允许速度随时间步长的减小而收敛于 **0.1875 m/s**。

龙头速度明显小于 2 m/s，这是因为掉头曲线内部分点曲率比值过大，出现速度传递系数较大的情况，如图 16 所示。这与问题四中 100 s 时出现的情况类似。

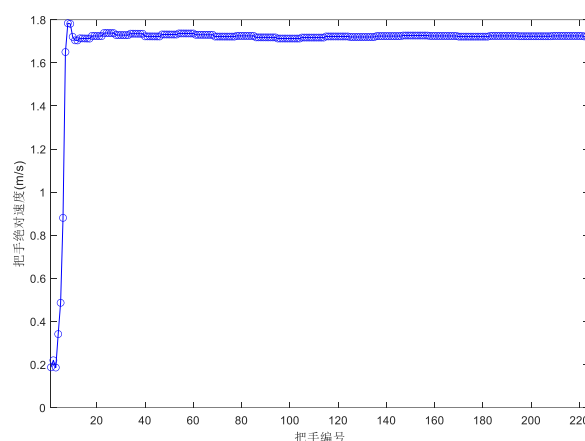


图 16 最大速度时刻各把手绝对速度图

六、模型的评价

6.1 模型的优点

1. 递推龙身、龙尾位置时，采用曲率圆近似螺线，避免了繁复的运算；
2. 问题四龙身、龙尾位置模型考虑到了所有可能出现的情形，模型通用性好；
3. 优化模型考虑实际情况，采用二分法进行求解，加快了求解速度。

6.2 模型的不足

龙身、龙尾递推过程中采取的近似，易使靠近内侧的位置误差稍大。

6.3 模型的推广

模型可推广至其它龙身参数、其它螺距、其它掉头空间半径中。

参考文献

- [1] 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.
- [2] Weisstein, Eric W. Archimedes' Spiral[EB/OL].<https://mathworld.wolfram.com/ArchimedesSpiral.html>.

附录说明	
附录 1	支撑材料清单
附录 2	源程序代码

附录 1 支撑材料清单

支撑材料清单		
问题一	result1.xlsx	求解结果
	Que1.m	求解程序
问题二	result2.xlsx	求解结果
	Que2.m	求解程序
问题三	Que3.m	求解程序
问题四	result4.xlsx	求解结果
	Que4.m	求解程序
问题五	Que5.m	求解程序

注. 拓展名为.m 的文件为 matlab 程序文件, 本文的运行环境是 MATLAB R2022b;
拓展名为.xlsx 的问题为 excel 文件, 存放所求结果。

附录 2 源程序代码

Que1.m
问题一求解程序
<pre>% 问题一求解程序 %% 初始化 clear clc close all %% 参数赋值 v=100; % 龙头速度(cm/s) b=55; % 螺距(cm) s=get_s(b,16*2*pi); % 自然坐标系龙头位置初值(cm) data=zeros((223+1)*2,300+1); % 存放最终结果 vdata=ones(223+1,300+1); % 速度(m/s) %% 时间递推 for t=0:300 % t-时间(s) theta=zeros(1,223+1); r=zeros(1,233+1); %% 求解龙头位置 if t~=0 s=s-v; % 龙头位置更新 end theta(1)=inv_s(b,s); % 龙头极角(rad) r(1)=get_curve(b,theta(1)); % 龙头极径(cm) [data(1,t+1),data(2,t+1)]=get_xy(r(1),theta(1));</pre>

```

%% 递推龙身位置
for idx=1:223
    if idx==1
        R=341-27.5*2;
    else
        R=220-27.5*2;
    end
    theta(idx+1)=get_body_theta(b,theta(idx),R);

[data(idx*2+1,t+1),data(idx*2+2,t+1)]=get_xy(get_curve(b,theta(idx+1)),theta(idx+1));

%% 求速度
x0=data(idx*2-1,t+1);
y0=data(idx*2,t+1);
x1=data(idx*2+1,t+1);
y1=data(idx*2+2,t+1);
k=(y1-y0)/(x1-x0); % 割线斜率
k0=(sin(theta(idx))+theta(idx)*cos(theta(idx)))/(cos(theta(idx))-theta(idx)*sin(theta(idx))); % 切线0斜率
k1=(sin(theta(idx+1))+theta(idx+1)*cos(theta(idx+1)))/(cos(theta(idx+1))-theta(idx+1)*sin(theta(idx+1))); % 切线1斜率
the0=abs(abs(atan(k0))-abs(atan(k)));
the1=abs(abs(atan(k))-abs(atan(k1)));
vdata(idx+1,t+1)=vdata(idx,t+1)*cos(the0)/cos(the1);
end
end

%% 检验
t_idx=[0,60,120,180,240,300]+1;
num=3:2:445;
figure
hold on
[x_std,y_std]=get_xy(get_curve(b,0:0.1:44*pi),0:0.1:44*pi);
for i=1:length(t_idx)
    idx=t_idx(i);
    dist(:,i)=((data(num,idx)-data(num+2,idx)).^2+(data(num+1,idx)-data(num+3,idx)).^2).^0.5;
    subplot(2,3,i)
    hold on
    plot(x_std,y_std,"-","Color",[0.01,0.01,0.01])
    plot(data(1:2:end-1,idx),data(2:2:end,idx),"-","Color",[0.1,0.3,0.9])
    plot(data(1:2:end-1,idx),data(2:2:end,idx),"*","Color",[0.9,0.1,0.1])
end
figure
boxplot(dist,'Labels',{0,1,2,3,4,5})
xlabel("时间(min)")
ylabel("距离(cm)")
data=round(data/100,6);
vdata=round(vdata,6);

%% 螺线方程
function r=get_curve(b,theta)
    % in:b-螺距(cm);theta-极角(rad)
    % out:r-极径(cm)
    r=b.*theta./(2.*pi);
end

```

```

%% 螺线方程转平面直角
function [x,y]=get_xy(r,theta)
    x=r.*cos(theta);
    y=r.*sin(theta);
end

%% 螺线弧长方程
function s=get_s(b,theta)
    % in:b-螺距(cm);theta-极角(rad)
    % out:s-弧长(cm)
    s=b./(4*pi).*(theta.*(1+theta.^2).^0.5+log(theta+(1+theta.^2).^0.5));
end

%% 反-螺线弧长方程,获取龙头极角
function theta=inv_s(b,s)
    % in:b-螺距(cm);s-弧长(cm)
    % out:theta-极角(rad)
    %% 二分求解
    left=0;
    right=17*2*pi;
    err=right-left;
    while err>1e-8
        mid=(right+left)/2;
        mid_s=get_s(b,mid);
        if mid_s>s
            right=mid;
        else
            left=mid;
        end
        err=right-left;
    end
    theta=(right+left)/2;
end

%% 递推龙身
function t=get_body_theta(b,theta_bef,R)
    [x0,y0]=get_xy(get_curve(b,theta_bef),theta_bef);
    p0=[x0,y0];
    ktemp=(sin(theta_bef)+theta_bef*cos(theta_bef))/(cos(theta_bef)-
theta_bef*sin(theta_bef)); % 切线斜率
    k=-1/ktemp;
    gamma=atan(ktemp);
    rho=get_rho(b,theta_bef);
    point_E1=[x0+rho*sin(gamma),y0-rho*cos(gamma)];
    point_E2=[x0-rho*sin(gamma),y0+rho*cos(gamma)];
    if norm(point_E1)<norm(point_E2)
        point_E=point_E1;
    else
        point_E=point_E2;
    end
    t=get_cross_point(theta_bef,point_E,rho,p0,R,R,x0,y0);
end

%% 求曲率半径
function rho=get_rho(b,theta)
    rho=b/(2*pi)*(theta^2+1)^(3/2)/(theta^2+2);
end

```



```

%% 求两圆交点
function theta=get_cross_point(theta_bef,p1,r1,p2,r2,R,x0,y0)
    % p1,p2-输入点横向量;r1,r2-圆半径
    d=norm(p1-p2);
    a=(r1^2-r2^2+d^2)/(2*d);
    h=sqrt(r1^2-a^2);
    p=p1+a*(p2-p1)/d;
    x1=p(1)+h*(p2(2)-p1(2))/d;
    y1=p(2)-h*(p2(1)-p1(1))/d;
    x2=p(1)-h*(p2(2)-p1(2))/d;
    y2=p(2)+h*(p2(1)-p1(1))/d;
    if y1*y2<0
        if x0>0
            if y1>0
                y=y1;
                x=x1;
            else
                x=x2;
                y=y2;
            end
        else
            if y1<0
                y=y1;
                x=x1;
            else
                x=x2;
                y=y2;
            end
        end
    elseif x1*x2<0
        if y0>0
            if x1<0
                x=x1;
                y=y1;
            else
                x=x2;
                y=y2;
            end
        else
            if x1>0
                x=x1;
                y=y1;
            else
                x=x2;
                y=y2;
            end
        end
    else
        if y0<0 || (y0==0 && x0<0)
            x=max(x1,x2);
        elseif y0>0
            x=min(x1,x2);
        end
        if x0>0 || (x0==0 && y0>0)
            y=max(y1,y2);
        else
            y=min(y1,y2);
        end
    end
end

```

```

        end
    end
    R1=norm(p2);
    R2=norm([x,y]);
    dtheta=acos((R1^2+R2^2-R^2)/(2*R1*R2));
    theta=(dtheta)+theta_bef;
end

```

Que2.m

问题二求解程序

```

% 问题二求解程序
%% 初始化
clear
clc
close all

%% 参数赋值
v=100; % 龙头速度(cm/s)
b=55; % 螺距(cm)
s=get_s(b,16*2*pi); % 自然坐标系龙头位置初值(cm)

xdata=zeros(223+1,100000); % x
ydata=zeros(223+1,100000); % y
vdata=ones(223+1,100000); % 速度(m/s)
%% 时间递推
t=-1; % t-时间(s)
while true
    t=t+1;
    theta=zeros(1,223+1);
    r=zeros(1,223+1);
    %% 求解龙头位置
    if t~=0
        s=s-0.01*v; % 龙头位置更新
    end
    theta(1)=inv_s(b,s); % 龙头极角(rad)
    r(1)=get_curve(b,theta(1)); % 龙头极径(cm)
    [xdata(1,t+1),ydata(1,t+1)]=get_xy(r(1),theta(1));
    %% 递推龙身位置
    for idx=1:223
        if idx==1
            R=341-27.5*2;
        else
            R=220-27.5*2;
        end
        theta(idx+1)=get_body_theta(b,theta(idx),R);

        [xdata(idx+1,t+1),ydata(idx+1,t+1)]=get_xy(get_curve(b,theta(idx+1)),theta(idx+1));
        %% 求速度
        x0=xdata(idx,t+1);
        y0=ydata(idx,t+1);
        x1=xdata(idx+1,t+1);
        y1=ydata(idx+1,t+1);
        k=(y1-y0)/(x1-x0); % 割线斜率
        k0=(sin(theta(idx))+theta(idx)*cos(theta(idx)))/(cos(theta(idx))-theta(idx)*sin(theta(idx))); % 切线θ斜率
    end
end

```

```

k1=(sin(theta(idx+1))+theta(idx+1)*cos(theta(idx+1)))/(cos(theta(idx+1))-
theta(idx+1)*sin(theta(idx+1))); % 切线 1 斜率
the0=abs(abs(atan(k0))-abs(atan(k)));
the1=abs(abs(atan(k))-abs(atan(k1)));
vdata(idx+1,t+1)=vdata(idx,t+1)*cos(the0)/cos(the1);
end
%% 判断是否相撞
for point_wood_idx=1:4
    if point_wood_idx==1
        chang=314/2;
    else
        chang=220/2;
    end
    kuan=30/2;
    point=[chang, chang, -chang, -chang;
           kuan, -kuan, kuan, -kuan];
    zhuan=-atan((ydata(point_wood_idx+1,t+1)-
ydata(point_wood_idx,t+1))/(xdata(point_wood_idx+1,t+1)-
xdata(point_wood_idx,t+1)));
    T=[cos(zhuan), sin(zhuan);
       -sin(zhuan), cos(zhuan)];
    point_aft_T=T*point;
    move=[(xdata(point_wood_idx+1,t+1)+xdata(point_wood_idx,t+1))/2;
          (ydata(point_wood_idx+1,t+1)+ydata(point_wood_idx,t+1))/2];
    point_aft_move=point_aft_T+move; % 2x4 array
    for line_wood_idx=point_wood_idx+2:100
        x1=xdata(line_wood_idx,t+1);
        x2=xdata(line_wood_idx+1,t+1);
        y1=ydata(line_wood_idx,t+1);
        y2=ydata(line_wood_idx+1,t+1);
        % 平行线系数 ABCp
        Ap=(y1-y2);
        Bp=-(x1-x2);
        Cp=(x1-x2)*y2-(y1-y2)*x2;

        dis_p=abs(Ap*point_aft_move(1,:)+Bp*point_aft_move(2,:)+Cp)/(Ap^2+Bp^2).^
0.5; % 到平行线距离
        if any(dis_p<15)

            line_mid=[(xdata(line_wood_idx+1,t+1)+xdata(line_wood_idx,t+1))/2;
                      (ydata(line_wood_idx+1,t+1)+ydata(line_wood_idx,t+1))/2];
            k=(ydata(line_wood_idx+1,t+1)-
ydata(line_wood_idx,t+1))/(xdata(line_wood_idx+1,t+1)-
xdata(line_wood_idx,t+1));
            % 中垂线系数 ABCc
            Ac=1;
            Bc=k;
            Cc=-line_mid(1)-k*line_mid(2);

            dis_c=abs(Ac*point_aft_move(1,:)+Bc*point_aft_move(2,:)+Cc)/(Ac^2+Bc^2).^
0.5; % 到中垂线距离
            if any(dis_c<110)
                figure
                hold on

                plot(point_aft_move(1,:),point_aft_move(2,:), "s", [x1,x2], [y1,y2], "r")
            end
        end
    end
end

```

```

        legend("板凳 A 四角","板凳 B 轴线")
        drawnow
        %% 画图

[x_std,y_std]=get_xy(get_curve(b,0:0.1:44*pi),0:0.1:44*pi);
        figure
        hold on
        plot(x_std,y_std,"-","Color",[0.01,0.01,0.01])
        plot(xdata(:,t),ydata(:,t),"-","Color",[0.1,0.3,0.9])
        plot(xdata(:,t),ydata(:,t),"*","Color",[0.9,0.1,0.1])
        return
    end
end
end
end
end

%% 检验
t_idx=[0,60,120,180,240,300]+1;
num=3:2:445;
figure
hold on
[x_std,y_std]=get_xy(get_curve(b,0:0.1:44*pi),0:0.1:44*pi);
for i=1:length(t_idx)
    idx=t_idx(i);
    dist(:,i)=((data(num,idx)-data(num+2,idx)).^2+(data(num+1,idx)-
data(num+3,idx)).^2).^0.5;
    subplot(2,3,i)
    hold on
    plot(x_std,y_std,"-","Color",[0.01,0.01,0.01])
    plot(data(1:2:end-1,idx),data(2:2:end,idx),"-","Color",[0.1,0.3,0.9])
    plot(data(1:2:end-1,idx),data(2:2:end,idx),"*","Color",[0.9,0.1,0.1])
end
figure
boxplot(dist,'Labels',{0,1,2,3,4,5})
xlabel("时间(min)")
ylabel("距离(cm)")
data=round(data,6);
vdata=round(vdata,6);

%% 螺线方程
function r=get_curve(b,theta)
    % in:b-螺距(cm);theta-极角(rad)
    % out:r-极径(cm)
    r=b.*theta./(2.*pi);
end

%% 螺线方程转平面直角
function [x,y]=get_xy(r,theta)
    x=r.*cos(theta);
    y=r.*sin(theta);
end

%% 螺线弧长方程
function s=get_s(b,theta)
    % in:b-螺距(cm);theta-极角(rad)
    % out:s-弧长(cm)
    s=b./(4*pi).*(theta.*(1+theta.^2).^0.5+log(theta+(1+theta.^2).^0.5));

```

```

end

%% 反-螺旋线弧长方程, 获取龙头极角
function theta=inv_s(b,s)
    % in:b-螺距(cm);s-弧长(cm)
    % out:theta-极角(rad)
    %% 二分求解
    left=0;
    right=17*2*pi;
    err=right-left;
    while err>1e-8
        mid=(right+left)/2;
        mid_s=get_s(b,mid);
        if mid_s>s
            right=mid;
        else
            left=mid;
        end
        err=right-left;
    end
    theta=(right+left)/2;
end

%% 递推龙身
function t=get_body_theta(b,theta_bef,R)
    [x0,y0]=get_xy(get_curve(b,theta_bef),theta_bef);
    p0=[x0,y0];
    ktemp=(sin(theta_bef)+theta_bef*cos(theta_bef))/(cos(theta_bef)-
theta_bef*sin(theta_bef)); % 切线斜率
    k=-1/ktemp;
    gamma=atan(ktemp);
    rho=get_rho(b,theta_bef);
    point_E1=[x0+rho*sin(gamma),y0-rho*cos(gamma)];
    point_E2=[x0-rho*sin(gamma),y0+rho*cos(gamma)];
    if norm(point_E1)<norm(point_E2)
        point_E=point_E1;
    else
        point_E=point_E2;
    end
    t=get_cross_point(theta_bef,point_E,rho,p0,R,R,x0,y0);
end

%% 求曲率半径
function rho=get_rho(b,theta)
    rho=b/(2*pi)*(theta^2+1)^(3/2)/(theta^2+2);
end

%% 求两圆交点
function theta=get_cross_point(theta_bef,p1,r1,p2,r2,R,x0,y0)
    % p1,p2-输入点横向量;r1,r2-圆半径
    d=norm(p1-p2);
    a=(r1^2-r2^2+d^2)/(2*d);
    h=sqrt(r1^2-a^2);
    p=p1+a*(p2-p1)/d;
    x1=p(1)+h*(p2(2)-p1(2))/d;
    y1=p(2)-h*(p2(1)-p1(1))/d;
    x2=p(1)-h*(p2(2)-p1(2))/d;
    y2=p(2)+h*(p2(1)-p1(1))/d;

```

```

if y1*y2<0
    if x0>0
        if y1>0
            y=y1;
            x=x1;
        else
            x=x2;
            y=y2;
        end
    else
        if y1<0
            y=y1;
            x=x1;
        else
            x=x2;
            y=y2;
        end
    end
elseif x1*x2<0
    if y0>0
        if x1<0
            x=x1;
            y=y1;
        else
            x=x2;
            y=y2;
        end
    else
        if x1>0
            x=x1;
            y=y1;
        else
            x=x2;
            y=y2;
        end
    end
else
    if y0<0
        x=max(x1,x2);
    elseif y0>0
        x=min(x1,x2);
    end
    if x0>0
        y=max(y1,y2);
    else
        y=min(y1,y2);
    end
end
R1=norm(p2);
R2=norm([x,y]);
dtheta=acos((R1^2+R2^2-R^2)/(2*R1*R2));
theta=(dtheta)+theta_bef;
end

```

Que3.m

问题三求解程序

% 问题三求解程序

```

%% 初始化
clear
clc
close all

%% 参数赋值
v=100; % 龙头速度(cm/s)
return_radius=4.5*100; % 掉头区间半径(cm)
%% 二分查找
b_left=30;
b_right=55;
figure
hold on
while b_right-b_left>1e-4
%% 进行尝试
    b=(b_left+b_right)/2 % 螺距(cm)
    s=get_s(b,16*2*pi); % 自然坐标系龙头位置初值(cm)
    %% 时间递推
    t=1; % t-时间(s)
    bk=0; % 跳出 flag
    while true
        t=t+1;
        xdata=zeros(223+1,1); % x
        ydata=zeros(223+1,1); % y
        vdata=ones(223+1,1); % 速度(m/s)
        theta=zeros(1,223+1);
        r=zeros(1,233+1);
        %% 求解龙头位置
        if t~=0
            s=s-0.01*v; % 龙头位置更新
        end
        theta(1)=inv_s(b,s); % 龙头极角(rad)
        r(1)=get_curve(b,theta(1)); % 龙头极径(cm)
        [xdata(1),ydata(1)]=get_xy(r(1),theta(1));
        %% 递推龙身位置
        for idx=1:223
            if idx==1
                R=341-27.5*2;
            else
                R=220-27.5*2;
            end
            theta(idx+1)=get_body_theta(b,theta(idx),R);

            [xdata(idx+1),ydata(idx+1)]=get_xy(get_curve(b,theta(idx+1)),theta(idx+1));

            %% 求速度
            x0=xdata(idx);
            y0=ydata(idx);
            x1=xdata(idx+1);
            y1=ydata(idx+1);
            k=(y1-y0)/(x1-x0); % 割线斜率

            k0=(sin(theta(idx))+theta(idx)*cos(theta(idx)))/(cos(theta(idx))-theta(idx)*sin(theta(idx))); % 切线 0 斜率

            k1=(sin(theta(idx+1))+theta(idx+1)*cos(theta(idx+1)))/(cos(theta(idx+1))-theta(idx+1)*sin(theta(idx+1))); % 切线 1 斜率
            the0=abs(abs(atan(k0))-abs(atan(k)));

```

```

        the1=abs(abs(atan(k))-abs(atan(k1)));
        vdata(idx+1)=vdata(idx)*cos(the0)/cos(the1);
    end
    %% 判断是否相撞
    for point_wood_idx=1:4
        if point_wood_idx==1
            chang=314/2;
        else
            chang=220/2;
        end
        kuan=30/2;
        point=[chang, chang, -chang, -chang;
            kuan, -kuan, kuan, -kuan];
        zhuan=-atan((ydata(point_wood_idx+1)-
ydata(point_wood_idx))/(xdata(point_wood_idx+1)-xdata(point_wood_idx)));
        T=[cos(zhuan), sin(zhuan);
            -sin(zhuan), cos(zhuan)];
        point_aft_T=T*point;
        move=[(xdata(point_wood_idx+1)+xdata(point_wood_idx))/2;
            (ydata(point_wood_idx+1)+ydata(point_wood_idx))/2];
        point_aft_move=point_aft_T+move; % 2x4 array
        for line_wood_idx=point_wood_idx+2:100
            x1=xdata(line_wood_idx);
            x2=xdata(line_wood_idx+1);
            y1=ydata(line_wood_idx);
            y2=ydata(line_wood_idx+1);
            % 平行线系数 ABCp
            Ap=(y1-y2);
            Bp=-(x1-x2);
            Cp=(x1-x2)*y2-(y1-y2)*x2;

            dis_p=abs(Ap*point_aft_move(1,:)+Bp*point_aft_move(2,:)+Cp)/(Ap^2+Bp^2).^
0.5; % 到平行线距离
            if any(dis_p<15)

                line_mid=[(xdata(line_wood_idx+1)+xdata(line_wood_idx))/2;

(ydata(line_wood_idx+1)+ydata(line_wood_idx))/2];
                k=(ydata(line_wood_idx+1)-
ydata(line_wood_idx))/(xdata(line_wood_idx+1)-xdata(line_wood_idx));
                % 中垂线系数 ABCc
                Ac=1;
                Bc=k;
                Cc=-line_mid(1)-k*line_mid(2);

                dis_c=abs(Ac*point_aft_move(1,:)+Bc*point_aft_move(2,:)+Cc)/(Ac^2+Bc^2).^
0.5; % 到中垂线距离
                if any(dis_c<110)
                    can=0;
                    bk=1;
                    break
                end
            end
        end
        if bk==1
            break
        end
    end
end

```



```

        if bk==1
            break
        end
        %% 判断是否进入调头区域
        if (xdata(1)^2+ydata(1)^2)^0.5<=return_radius
            can=1;
            break
        end
    end
    %% 二分更新
    if can==1
        plot(b,1,"b+", 'MarkerSize',12, 'LineWidth',2);
        % 可掉头,b 太大
        b_right=b;
    else
        plot(b,0,"rx", 'MarkerSize',12, 'LineWidth',2);
        % 不可掉头,b 太小
        b_left=b;
    end
end

%% 画图
% [x_std,y_std]=get_xy(get_curve(b,0:0.1:44*pi),0:0.1:44*pi);
% figure
% hold on
% plot(x_std,y_std,"-", 'Color',[0.01,0.01,0.01])
% plot(xdata(:),ydata(:),"-", 'Color',[0.1,0.3,0.9])
% plot(xdata(:),ydata(:),"*", 'Color',[0.9,0.1,0.1])

%% 螺线方程
function r=get_curve(b,theta)
    % in:b-螺距(cm);theta-极角(rad)
    % out:r-极径(cm)
    r=b.*theta./(2.*pi);
end

%% 螺线方程转平面直角
function [x,y]=get_xy(r,theta)
    x=r.*cos(theta);
    y=r.*sin(theta);
end

%% 螺线弧长方程
function s=get_s(b,theta)
    % in:b-螺距(cm);theta-极角(rad)
    % out:s-弧长(cm)
    s=b./(4*pi).*(theta.*(1+theta.^2).^0.5+log(theta+(1+theta.^2).^0.5));
end

%% 反-螺线弧长方程,获取龙头极角
function theta=inv_s(b,s)
    % in:b-螺距(cm);s-弧长(cm)
    % out:theta-极角(rad)
    %% 二分求解
    left=0;
    right=17*2*pi;
    err=right-left;
    while err>1e-8

```

```

        mid=(right+left)/2;
        mid_s=get_s(b,mid);
        if mid_s>s
            right=mid;
        else
            left=mid;
        end
        err=right-left;
    end
    theta=(right+left)/2;
end

%% 递推龙身
function t=get_body_theta(b,theta_bef,R)
    [x0,y0]=get_xy(get_curve(b,theta_bef),theta_bef);
    p0=[x0,y0];
    ktemp=(sin(theta_bef)+theta_bef*cos(theta_bef))/(cos(theta_bef)-
theta_bef*sin(theta_bef)); % 切线斜率
    k=-1/ktemp;
    gamma=atan(ktemp);
    rho=get_rho(b,theta_bef);
    point_E1=[x0+rho*sin(gamma),y0-rho*cos(gamma)];
    point_E2=[x0-rho*sin(gamma),y0+rho*cos(gamma)];
    if norm(point_E1)<norm(point_E2)
        point_E=point_E1;
    else
        point_E=point_E2;
    end
    t=get_cross_point(theta_bef,point_E,rho,p0,R,R,x0,y0);
end

%% 求曲率半径
function rho=get_rho(b,theta)
    rho=b/(2*pi)*(theta^2+1)^(3/2)/(theta^2+2);
end

%% 求两圆交点
function theta=get_cross_point(theta_bef,p1,r1,p2,r2,R,x0,y0)
    % p1,p2-输入点横向量;r1,r2-圆半径
    d=norm(p1-p2);
    a=(r1^2-r2^2+d^2)/(2*d);
    h=sqrt(r1^2-a^2);
    p=p1+a*(p2-p1)/d;
    x1=p(1)+h*(p2(2)-p1(2))/d;
    y1=p(2)-h*(p2(1)-p1(1))/d;
    x2=p(1)-h*(p2(2)-p1(2))/d;
    y2=p(2)+h*(p2(1)-p1(1))/d;
    if y1*y2<0
        if x0>0
            if y1>0
                y=y1;
                x=x1;
            else
                x=x2;
                y=y2;
            end
        else
            if y1<0

```

```

        y=y1;
        x=x1;
    else
        x=x2;
        y=y2;
    end
end
elseif x1*x2<0
    if y0>0
        if x1<0
            x=x1;
            y=y1;
        else
            x=x2;
            y=y2;
        end
    else
        if x1>0
            x=x1;
            y=y1;
        else
            x=x2;
            y=y2;
        end
    end
end
else
    if y0<0
        x=max(x1,x2);
    elseif y0>0
        x=min(x1,x2);
    end
    if x0>0
        y=max(y1,y2);
    else
        y=min(y1,y2);
    end
end
R1=norm(p2);
R2=norm([x,y]);
dtheta=acos((R1^2+R2^2-R^2)/(2*R1*R2));
theta=(dtheta)+theta_bef;
end

```

Que4.m

问题四求解程序

```

% 问题三求解程序
%% 初始化
clear
clc
close all

%% 参数赋值
v=100; % 龙头速度(cm/s)
b=170; % 螺距(cm)
s=get_s(b,8*2*pi); % 自然坐标系龙头位置初值(cm)
head_s=-1; % 龙头在掉头曲线自然坐标系下坐标(cm)
dt=1; % 采样时间(s)

```

```

%% 进行尝试
return_radius=450; % 掉头空间半径(cm)
boundary_theta=2*pi*return_radius/b; % 进入掉头空间的临界极角(rad)
rotate=pi/2-mod(boundary_theta,2*pi);
t_in=-1; % 进入掉头区域时间索引
%% 时间递推
t=-1; % t-时间(s)
bk=0; % 跳出 flag
rec_x=zeros(223+1,3000); % 每秒结果保存(m)
rec_y=zeros(223+1,3000); % 每秒结果保存(m)
rec_v=zeros(223+1,3000); % 每秒结果保存(m/s)
res_pot=zeros((223+1)*2,3000); % 贴入 excel 的位置数据
while true
    t=t+1;
    if t_in~-1 && t>t_in+100/dt
        break
    end
    xdata=zeros(223+1,1); % x(cm)
    ydata=zeros(223+1,1); % y(cm)
    vdata=ones(223+1,1)*v/100; % 速度(m/s)
    theta=zeros(1,223+1); % 极角
    r=zeros(1,233+1);
    state=zeros(233+1,1); % 节点状态:0-盘入,1-掉头圆 1,2-掉头圆 2,3-盘出
    %% 求解龙头位置
    if t~=0 && (head_s== -1 || head_s== -2)
        s=s-dt*v; % 龙头位置更新
    end
    theta(1)=inv_s(b,s); % 龙头极角(rad)
    if theta(1)>boundary_theta && head_s == -1
        % 掉头前
        r(1)=get_curve(b,theta(1)); % 龙头极径(cm)
        [xdata(1),ydata(1)]=get_xy(r(1),theta(1));
    elseif (abs(theta(1))<boundary_theta || head_s > 0) &&
head_s<(calculate_return_length(return_radius)-dt*v) && head_s ~= -2
        % 掉头时
        state(1)=1;
        if head_s == -1
            % 初次进入
            head_s=(s+dt*v)-get_s(b,boundary_theta); % 进入路程(cm)
        else
            head_s=head_s+dt*v;
        end
        if head_s>pi*return_radius*2/3
            state(1)=2;
        end
        if t_in== -1
            t_in=t;
        end
        [xdata(1),ydata(1)]=from_s_get_return_xy(head_s,return_radius,rotate);
    else
        % 掉头后
        state(1)=3;
        if s>0 % 初次离开
            s=-get_s(b,boundary_theta)-(head_s-
calculate_return_length(return_radius)+dt*v);
            head_s=-2;
        end
    end
end

```

```

end
theta(1)=inv_s(b,abs(s));
r(1)=abs(get_curve(b,theta(1))); % 龙头极径(cm)
[xdata(1),ydata(1)]=get_xy(r(1),theta(1));
xdata(1)=-xdata(1);
ydata(1)=-ydata(1);
end
%      plot(xdata(1),ydata(1),"*r")
%% 递推龙身位置
for idx=1:223
    if idx==1
        L=341-27.5*2;
    else
        L=220-27.5*2;
    end
    T=[cos(rotate),sin(rotate);
        -sin(rotate),cos(rotate)];
    res=T*[0,
        0;
        return_radius/3, -return_radius*2/3];
    xo1=res(1,1);
    yo1=res(2,1);
    xo2=res(1,2);
    yo2=res(2,2);
    ro1=return_radius*2/3;
    ro2=return_radius/3;
    [xin,yin]=get_xy(return_radius,boundary_theta);

    if state(idx)==0
        % 前一节点在盘入
        theta(idx+1)=get_body_theta(b,theta(idx),L);

[xdata(idx+1),ydata(idx+1)]=get_xy(get_curve(b,theta(idx+1)),theta(idx+1)
);
    elseif state(idx)==1
        % 前一节点在掉头圆 1
        temp1=acos((xdata(idx)^2+ydata(idx)^2+return_radius^2-
L^2)/(2*return_radius*(xdata(idx)^2+ydata(idx)^2)^0.5));
        temp2=acos((return_radius^2+xdata(idx)^2+ydata(idx)^2-
(xdata(idx)-xin)^2-(ydata(idx)-
yin)^2)/(2*return_radius*(xdata(idx)^2+ydata(idx)^2)^0.5));
        if temp1>temp2 && isreal(temp1)
            % 此节点未进入掉头圆 1
            state(idx+1)=0;
            theta(idx+1)=boundary_theta+temp1-temp2;

[xdata(idx+1),ydata(idx+1)]=get_xy(get_curve(b,theta(idx+1)),theta(idx+1)
);
        else
            % 此节点已进入掉头圆 1
            state(idx+1)=1;
            temp3=2*asin(L/(2*ro1));
            Trans=[cos(temp3),-sin(temp3);
                sin(temp3),cos(temp3)]; % 逆时针旋转
            tem=Trans*[xdata(idx)-xo1;ydata(idx)-yo1];
            xdata(idx+1)=tem(1)+xo1;
            ydata(idx+1)=tem(2)+yo1;
        end
    elseif state(idx)==2

```

```

% 前一节点在掉头圆 2
T=[cos(rotate),-sin(rotate);
   sin(rotate),cos(rotate)];
res=T*[xdata(idk);ydata(idk)];
y_bef=res(2);
if y_bef>=-L^2/(2*ro2)-return_radius/3
    % 此节点未进入掉头圆 2
    state(idk+1)=1;
    temp=acos(((xdata(idk)-xo1)^2+(ydata(idk)-
yo1)^2+return_radius^2-L^2)/(2*return_radius*((xdata(idk)-
xo1)^2+(ydata(idk)-yo1)^2)^0.5));
    Trans=[cos(temp),-sin(temp);
           sin(temp),cos(temp)]; % 逆时针旋转
    res=Trans*[xdata(idk)-xo1;ydata(idk)-yo1];% +[xo1;yo1];
    det_r=res;%-[xo1;yo1];
    det_r=det_r/norm(det_r)*ro1;
    xdata(idk+1)=det_r(1)+xo1;
    ydata(idk+1)=det_r(2)+yo1;
else
    % 此节点已进入掉头圆 2
    state(idk+1)=2;
    temp3=2*asin(L/(2*ro2));
    Trans=[cos(temp3),sin(temp3);
           -sin(temp3),cos(temp3)]; % 顺时针旋转
    tem=Trans*[xdata(idk)-xo2;ydata(idk)-yo2];
    xdata(idk+1)=tem(1)+xo2;
    ydata(idk+1)=tem(2)+yo2;
end
else
    % 前一节点在盘出
    temp1=acos(((xdata(idk)-xo2)^2+(ydata(idk)-yo2)^2+ro2^2-
L^2)/(2*ro2*((xdata(idk)-xo2)^2+(ydata(idk)-yo2)^2)^0.5));
    temp2=acos(((xdata(idk)-xo2)^2+(ydata(idk)-yo2)^2+ro2^2-
(xdata(idk)+xin)^2-(ydata(idk)+yin)^2)/(2*ro2*((xdata(idk)-
xo2)^2+(ydata(idk)-yo2)^2)^0.5));
    if temp1>temp2 && theta(idk)<boundary_theta+pi
        % 此节点未进入盘出曲线
        state(idk+1)=2;
        xout=-xin;
        yout=-yin;
        temp3=temp1-temp2;
        Trans=[cos(temp3),sin(temp3);
               -sin(temp3),cos(temp3)]; % 顺时针旋转
        tem=Trans*[xout-xo2;yout-yo2];
        xdata(idk+1)=tem(1)+xo2;
        ydata(idk+1)=tem(2)+yo2;
    else
        % 进入盘出曲线
        state(idk+1)=3;
        theta(idk+1)=get_body_theta_out(b,theta(idk),L);
        [xtemp,ytemp]=get_xy(get_curve(b,theta(idk+1)),theta(idk+1));
        xdata(idk+1)=-xtemp;
        ydata(idk+1)=-ytemp;
    end
end
%% 求速度
x0=xdata(idk);

```

```

        y0=ydata(idk);
        x1=xdata(idk+1);
        y1=ydata(idk+1);
        k=(y1-y0)/(x1-x0); % 割线斜率
        if state(idk)==0

k0=(sin(theta(idk))+theta(idk)*cos(theta(idk)))/(cos(theta(idk))-
theta(idk)*sin(theta(idk))); % 切线0斜率
        elseif state(idk)==3
            k0=-
(sin(theta(idk))+theta(idk)*cos(theta(idk)))/(cos(theta(idk))-
theta(idk)*sin(theta(idk))); % 切线0斜率
        elseif state(idk)==1
            k0=-1/((y01-y0)/(x01-x0));
        else
            k0=-1/((y02-y0)/(x02-x0));
        end
        if state(idk+1)==0

k1=(sin(theta(idk+1))+theta(idk+1)*cos(theta(idk+1)))/(cos(theta(idk+1))-
theta(idk+1)*sin(theta(idk+1))); % 切线1斜率
        elseif state(idk+1)==3
            k1=-
(sin(theta(idk+1))+theta(idk+1)*cos(theta(idk+1)))/(cos(theta(idk+1))-
theta(idk+1)*sin(theta(idk+1))); % 切线1斜率
        elseif state(idk+1)==1
            k0=-1/((y01-y1)/(x01-x1));
        else
            k0=-1/((y02-y1)/(x02-x1));
        end
        the0=abs(abs(atan(k0))-abs(atan(k)));
        the1=abs(abs(atan(k))-abs(atan(k1)));
        vdata(idk+1)=vdata(idk)*cos(the0)/cos(the1);

    end
    %% 绘图
    if t_in~-1 && (t-t_in==0 || t-t_in==50 || t-t_in==100)
        figure

[x_std,y_std]=get_xy(get_curve(b,boundary_theta:0.1:44*pi),boundary_theta
:0.1:44*pi);
        [x_circle,y_circle]=get_xy(return_radius,0:0.1:2*pi);
        S_x=[];
        S_y=[];
        for sss=0:0.1:calculate_return_length(return_radius)

[S_x(end+1),S_y(end+1)]=from_s_get_return_xy(sss,return_radius,rotate);
            end
            hold on
            plot(x_std,y_std,"-","Color","b")
            plot(-x_std,-y_std,"-","Color","g")
            plot(S_x,S_y,"-","Color","k")
            plot(x_circle,y_circle,"--","Color",[0.1,0.7,0.1],'LineWidth',1)
            plot(xdata(:),ydata(:),"-","Color",[0.1,0.3,0.9])
            plot(xdata(:),ydata(:),"*","Color",[0.9,0.1,0.1])
            axis square
        end
        if abs(mod(t*dt,1))<1e-10

```

```

        rec_x(:,round(t*dt)+1)=round(xdata(:,1)/100,6);
        rec_y(:,round(t*dt)+1)=round(ydata(:,1)/100,6);
        rec_v(:,round(t*dt)+1)=round(vdata(:,1),6);
        res_pot(1:2:end,round(t*dt)+1)=rec_x(:,round(t*dt)+1);
        res_pot(2:2:end,round(t*dt)+1)=rec_y(:,round(t*dt)+1);
    end
end
fprintf("结果存于变量 res_pot、rec_v 中")
fprintf("截取两变量最后 101 列写于 excel 中")

%% 画图
figure
t_ls=[t_in,t_in+50,t_in+100];
for i=1:3
    xdata=rec_x(:,t_ls(i))*100;
    ydata=rec_y(:,t_ls(i))*100;
    subplot(1,3,i)

    [x_std,y_std]=get_xy(get_curve(b,boundary_theta:0.1:20*pi),boundary_theta:0.1:20*pi);
    [x_circle,y_circle]=get_xy(return_radius,0:0.1:2*pi);
    S_x=[];
    S_y=[];
    for sss=0:0.1:calculate_return_length(return_radius)

        [S_x(end+1),S_y(end+1)]=from_s_get_return_xy(sss,return_radius,rotate);
    end
    hold on
    plot(x_std,y_std,"-","Color","b")
    plot(-x_std,-y_std,"-","Color","g")
    plot(S_x,S_y,"-","Color","k")
    plot(x_circle,y_circle,"--","Color",[0.1,0.7,0.1],'LineWidth',1)
    plot(xdata(:),ydata(:),"-","Color",[0.1,0.3,0.9])
    plot(xdata(:),ydata(:),"*","Color",[0.9,0.1,0.1])
    axis square
    title(strcat(num2str(t_ls(i)-t_in),"s"))
end

%% 螺线方程
function r=get_curve(b,theta)
    % in:b-螺距(cm);theta-极角(rad)
    % out:r-极径(cm)
    r=b.*theta./(2.*pi);
end

%% 螺线方程转平面直角
function [x,y]=get_xy(r,theta)
    x=r.*cos(theta);
    y=r.*sin(theta);
end

%% 螺线弧长方程
function s=get_s(b,theta)
    % in:b-螺距(cm);theta-极角(rad)
    % out:s-弧长(cm)
    s=b./(4*pi).*(theta.*(1+theta.^2).^0.5+log(theta+(1+theta.^2).^0.5));
end

```



```

%% 反-螺旋弧长方程,获取龙头极角
function theta=inv_s(b,s)
    % in:b-螺距(cm);s-弧长(cm)
    % out:theta-极角(rad)
    %% 二分求解
    left=0;
    right=17*2*pi;
    err=right-left;
    while err>1e-8
        mid=(right+left)/2;
        mid_s=get_s(b,mid);
        if mid_s>s
            right=mid;
        else
            left=mid;
        end
        err=right-left;
    end
    theta=(right+left)/2;
end

%% 盘入递推龙身
function t=get_body_theta(b,theta_bef,R)
    % in:b-螺距(cm);theta_bef-前节点极角(rad),R-龙身节点间长度
    (cm);rerurn_radius-掉头空间半径(cm)
    % out:t-下一节点极角(rad)
    [x0,y0]=get_xy(get_curve(b,theta_bef),theta_bef);
    p0=[x0,y0];
    ktemp=(sin(theta_bef)+theta_bef*cos(theta_bef))/(cos(theta_bef)-
theta_bef*sin(theta_bef)); % 切线斜率
    gamma=atan(ktemp);
    rho=get_rho(b,theta_bef);
    point_E1=[x0+rho*sin(gamma),y0-rho*cos(gamma)];
    point_E2=[x0-rho*sin(gamma),y0+rho*cos(gamma)];
    if norm(point_E1)<norm(point_E2)
        point_E=point_E1;
    else
        point_E=point_E2;
    end
    t=get_cross_point(theta_bef,point_E,rho,p0,R,R,x0,y0);
end

%% 盘出递推龙身
function t=get_body_theta_out(b,theta_bef,R)
    % in:b-螺距(cm);theta_bef-前节点极角(rad),R-龙身节点间长度
    (cm);rerurn_radius-掉头空间半径(cm)
    % out:t-下一节点极角(rad)
    [x0,y0]=get_xy(get_curve(b,theta_bef),theta_bef);
    p0=[x0,y0];
    ktemp=(sin(theta_bef)+theta_bef*cos(theta_bef))/(cos(theta_bef)-
theta_bef*sin(theta_bef)); % 切线斜率
    gamma=atan(ktemp);
    rho=get_rho(b,theta_bef);
    point_E1=[x0+rho*sin(gamma),y0-rho*cos(gamma)];
    point_E2=[x0-rho*sin(gamma),y0+rho*cos(gamma)];
    if norm(point_E1)<norm(point_E2)
        point_E=point_E1;

```

```

else
    point_E=point_E2;
end
t=get_cross_point_out(theta_bef,point_E,rho,p0,R,R,x0,y0);
end

%% 求曲率半径
function rho=get_rho(b,theta)
    rho=b/(2*pi)*(theta^2+1)^(3/2)/(theta^2+2);
end

%% 求两圆交点
function theta=get_cross_point(theta_bef,p1,r1,p2,r2,R,x0,y0)
    % p1,p2-输入点横向量;r1,r2-圆半径
    d=norm(p1-p2);
    a=(r1^2-r2^2+d^2)/(2*d);
    h=sqrt(r1^2-a^2);
    p=p1+a*(p2-p1)/d;
    x1=p(1)+h*(p2(2)-p1(2))/d;
    y1=p(2)-h*(p2(1)-p1(1))/d;
    x2=p(1)-h*(p2(2)-p1(2))/d;
    y2=p(2)+h*(p2(1)-p1(1))/d;
    if y1*y2<0
        if x0>0
            if y1>0
                y=y1;
                x=x1;
            else
                x=x2;
                y=y2;
            end
        else
            if y1<0
                y=y1;
                x=x1;
            else
                x=x2;
                y=y2;
            end
        end
    elseif x1*x2<0
        if y0>0
            if x1<0
                x=x1;
                y=y1;
            else
                x=x2;
                y=y2;
            end
        else
            if x1>0
                x=x1;
                y=y1;
            else
                x=x2;
                y=y2;
            end
        end
    end
end
end

```

```

else
    if y0<0
        x=max(x1,x2);
    else
        x=min(x1,x2);
    end
    if x0>0
        y=max(y1,y2);
    else
        y=min(y1,y2);
    end
end
R1=norm(p2);
R2=norm([x,y]);
dtheta=acos((R1^2+R2^2-R^2)/(2*R1*R2));
theta=(dtheta)+theta_bef;
end

%% 求盘出两圆交点
function theta=get_cross_point_out(theta_bef,p1,r1,p2,r2,R,x0,y0)
% p1,p2-输入点横向量;r1,r2-圆半径
d=norm(p1-p2);
a=(r1^2-r2^2+d^2)/(2*d);
h=sqrt(r1^2-a^2);
p=p1+a*(p2-p1)/d;
x1=p(1)+h*(p2(2)-p1(2))/d;
y1=p(2)-h*(p2(1)-p1(1))/d;
x2=p(1)-h*(p2(2)-p1(2))/d;
y2=p(2)+h*(p2(1)-p1(1))/d;
if norm([x1,y1])<norm([x2,y2])
    x=x1;
    y=y1;
else
    x=x2;
    y=y2;
end
% if y1*y2<0
%     if x0>0
%         if y1>0
%             y=y1;
%             x=x1;
%         else
%             x=x2;
%             y=y2;
%         end
%     else
%         if y1<0
%             y=y1;
%             x=x1;
%         else
%             x=x2;
%             y=y2;
%         end
%     end
% elseif x1*x2<0
%     if y0>0
%         if x1<0
%             x=x1;

```

```

%         y=y1;
%     else
%         x=x2;
%         y=y2;
%     end
% else
%     if x1>0
%         x=x1;
%         y=y1;
%     else
%         x=x2;
%         y=y2;
%     end
% end
% else
%     if y0<0
%         x=max(x1,x2);
%     else
%         x=min(x1,x2);
%     end
%     if x0>0
%         y=max(y1,y2);
%     else
%         y=min(y1,y2);
%     end
% end
R1=norm(p2);
R2=norm([x,y]);
dtheta=acos((R1^2+R2^2-R^2)/(2*R1*R2));
theta=-(dtheta)+theta_bef;
end

%% 求掉头曲线 s 对应的两相对 theta
function [theta1,theta2]=get_return_theta(s,return_radius)
    r1=return_radius*2/3;
    r2=return_radius/3;
    theta1=-1;
    theta2=-1;
    s1=pi*r1;
    s2=pi*r2;
    if s<s1 % 在前圆内
        theta1=pi*s/s1;
    else % 在后圆内
        theta2=pi*(s-s1)/s2;
    end
end

%% 掉头曲线 s 自然坐标转直角坐标
function [x,y]=from_s_get_return_xy(s,return_radius,rotate)
    % in:s-自然坐标(cm),return_radius-掉头空间半径(cm),rotate-旋转角度(rad)
    % out:x,y-坐标(cm)
    [theta1,theta2]=get_return_theta(s,return_radius);
    r1=return_radius*2/3;
    r2=return_radius/3;
    if theta1 >= 0 % 在前圆内
        x1=r1*sin(theta1);
        y1=r1*cos(theta1)+return_radius/3;
    else % 在后圆内

```

```

        x1=-r2*sin(theta2);
        y1=r2*cos(theta2)-return_radius*2/3;
    end
    %% 旋转
    T=[cos(rotate),sin(rotate);
        -sin(rotate),cos(rotate)];
    res=T*[x1;y1];
    x=res(1);
    y=res(2);
end

function s=calculate_return_length(r)
    % in:r-掉头空间半径(cm)
    % out:s-掉头曲线长度(cm)
    s=pi*r;
end

```

Que5.m

问题五求解程序

```

% 问题三求解程序
%% 初始化
clear
clc
close all

%% 参数赋值
% v=100; % 龙头速度(cm/s)
b=170; % 螺距(cm)
s=get_s(b,8*2*pi); % 自然坐标系龙头位置初值(cm)
head_s=-1; % 龙头在掉头曲线自然坐标系下坐标(cm)
dt=0.1; % 采样时间(s)
%% 二分查找
v_left=15;
v_right=20;
while v_right-v_left>1e-4
    %% 进行尝试
    % hold on
    return_radius=450; % 掉头空间半径(cm)
    v=(v_left+v_right)/2
    % v=18.74
    boundary_theta=2*pi*return_radius/b; % 进入掉头空间的临界极角(rad)
    rotate=pi/2-mod(boundary_theta,2*pi);
    t_in=-1; % 进入掉头区域时间索引
    %% 时间递推
    t=-1; % t-时间(s)
    bk=0; % 跳出 flag
    rec_x=zeros(223+1,3000); % 每秒结果保存(m)
    rec_y=zeros(223+1,3000); % 每秒结果保存(m)
    rec_v=zeros(223+1,30000); % 每秒结果保存(m/s)
    state=zeros(233+1,1); % 节点状态:0-盘入,1-掉头圆 1,2-掉头圆 2,3-盘出
    while true
        t=t+1;
        if t_in~-1 && t>t_in+100/dt
            break
        end
        xdata=zeros(223+1,1); % x(cm)
        ydata=zeros(223+1,1); % y(cm)
    end
end

```

```

vdata=ones(223+1,1)*v/100; % 速度(m/s)
theta=zeros(1,223+1); % 极角
r=zeros(1,233+1);
%% 求解龙头位置
if t~=0 && (head_s==-1 || head_s==-2)
    s=s-dt*v; % 龙头位置更新
end
theta(1)=inv_s(b,s); % 龙头极角(rad)
if theta(1)>boundary_theta && head_s == -1
    % 掉头前
    r(1)=get_curve(b,theta(1)); % 龙头极径(cm)
    [xdata(1),ydata(1)]=get_xy(r(1),theta(1));
elseif (abs(theta(1))<boundary_theta || head_s > 0) &&
head_s<(calculate_return_length(return_radius)-dt*v) && head_s ~= -2
    % 掉头时
    state(1)=1;
    if head_s == -1
        % 初次进入
        head_s=(s+dt*v)-get_s(b,boundary_theta); % 进入路程(cm)
    else
        head_s=head_s+dt*v;
    end
    if head_s>pi*return_radius*2/3
        state(1)=2;
    end
    if t_in==-1
        t_in=t;
    end
    [xdata(1),ydata(1)]=from_s_get_return_xy(head_s,return_radius,rotate);
else
    % 掉头后
    if t_in==-1
        t_in=t;
    end
    state(1)=3;
    if s>0 % 初次离开
        s=-get_s(b,boundary_theta)-(head_s-
calculate_return_length(return_radius)+dt*v);
        head_s=-2;
    end
    theta(1)=inv_s(b,abs(s));
    r(1)=abs(get_curve(b,theta(1))); % 龙头极径(cm)
    [xdata(1),ydata(1)]=get_xy(r(1),theta(1));
    xdata(1)=-xdata(1);
    ydata(1)=-ydata(1);
end
%
plot(xdata(1),ydata(1),"*r")
%% 递推龙身位置
for idx=1:223
    if idx==1
        L=341-27.5*2;
    else
        L=220-27.5*2;
    end
    T=[cos(rotate),sin(rotate);
        -sin(rotate),cos(rotate)];
    res=T*[0,
        0;

```

```

        return_radius/3, -return_radius*2/3];
    xo1=res(1,1);
    yo1=res(2,1);
    xo2=res(1,2);
    yo2=res(2,2);
    ro1=return_radius*2/3;
    ro2=return_radius/3;
    [xin,yin]=get_xy(return_radius,boundary_theta);
    if state(idx)==0
        % 前一节点在盘入
        theta(idx+1)=get_body_theta(b,theta(idx),L);

[xdata(idx+1),ydata(idx+1)]=get_xy(get_curve(b,theta(idx+1)),theta(idx+1)
);
        elseif state(idx)==1
            % 前一节点在掉头圆 1
            temp1=acos((xdata(idx)^2+ydata(idx)^2+return_radius^2-
L^2)/(2*return_radius*(xdata(idx)^2+ydata(idx)^2)^0.5));
            temp2=acos((return_radius^2+xdata(idx)^2+ydata(idx)^2-
(xdata(idx)-xin)^2-(ydata(idx)-
yin)^2)/(2*return_radius*(xdata(idx)^2+ydata(idx)^2)^0.5));
            if temp1>temp2
                % 此节点未进入掉头圆 1
                state(idx+1)=0;
                theta(idx+1)=boundary_theta+temp1-temp2;

[xdata(idx+1),ydata(idx+1)]=get_xy(get_curve(b,theta(idx+1)),theta(idx+1)
);
            else
                % 此节点已进入掉头圆 1
                state(idx+1)=1;
                temp3=2*asin(L/(2*ro1));
                Trans=[cos(temp3),-sin(temp3);
                    sin(temp3),cos(temp3)]; % 逆时针旋转
                tem=Trans*[xdata(idx)-xo1;ydata(idx)-yo1];
                xdata(idx+1)=tem(1)+xo1;
                ydata(idx+1)=tem(2)+yo1;
            end
        elseif state(idx)==2
            % 前一节点在掉头圆 2
            T=[cos(rotate),-sin(rotate);
                sin(rotate),cos(rotate)];
            res=T*[xdata(idx);ydata(idx)];
            y_bef=res(2);
            if y_bef>=-L^2/(2*ro2)-return_radius/3
                % 此节点未进入掉头圆 2
                state(idx+1)=1;
                temp=acos(((xdata(idx)-xo1)^2+(ydata(idx)-
yo1)^2+return_radius^2-L^2)/(2*return_radius*((xdata(idx)-
xo1)^2+(ydata(idx)-yo1)^2)^0.5));
                Trans=[cos(temp),-sin(temp);
                    sin(temp),cos(temp)]; % 逆时针旋转
                res=Trans*[xdata(idx)-xo1;ydata(idx)-yo1];% +[xo1;yo1];
                det_r=res;%-[xo1;yo1];
                det_r=det_r/norm(det_r)*ro1;
                xdata(idx+1)=det_r(1)+xo1;
                ydata(idx+1)=det_r(2)+yo1;
            else

```

```

        % 此节点已进入掉头圆 2
        state(idx+1)=2;
        temp3=2*asin(L/(2*ro2));
        Trans=[cos(temp3),sin(temp3);
               -sin(temp3),cos(temp3)]; % 顺时针旋转
        tem=Trans*[xdata(idx)-xo2;ydata(idx)-yo2];
        xdata(idx+1)=tem(1)+xo2;
        ydata(idx+1)=tem(2)+yo2;
    end
else
    % 前一节点在盘出
    temp1=acos(((xdata(idx)-xo2)^2+(ydata(idx)-yo2)^2+ro2^2-
L^2)/(2*ro2*((xdata(idx)-xo2)^2+(ydata(idx)-yo2)^2)^0.5));
    temp2=acos(((xdata(idx)-xo2)^2+(ydata(idx)-yo2)^2+ro2^2-
(xdata(idx)+xin)^2-(ydata(idx)+yin)^2)/(2*ro2*((xdata(idx)-
xo2)^2+(ydata(idx)-yo2)^2)^0.5));
    if temp1>temp2
        % 此节点未进入盘出曲线
        state(idx+1)=2;
        xout=-xin;
        yout=-yin;
        temp3=temp1-temp2;
        Trans=[cos(temp3),sin(temp3);
               -sin(temp3),cos(temp3)]; % 顺时针旋转
        tem=Trans*[xout-xo2;yout-yo2];
        xdata(idx+1)=tem(1)+xo2;
        ydata(idx+1)=tem(2)+yo2;
    else
        % 进入盘出曲线
        state(idx+1)=3;
        theta(idx+1)=get_body_theta_out(b,theta(idx),L);

[xtemp,ytemp]=get_xy(get_curve(b,theta(idx+1)),theta(idx+1));
        xdata(idx+1)=-xtemp;
        ydata(idx+1)=-ytemp;
    end

end

%% 求速度
x0=xdata(idx);
y0=ydata(idx);
x1=xdata(idx+1);
y1=ydata(idx+1);
k=(y1-y0)/(x1-x0); % 割线斜率
if state(idx)==0

k0=(sin(theta(idx))+theta(idx)*cos(theta(idx)))/(cos(theta(idx))-
theta(idx)*sin(theta(idx))); % 切线 0 斜率
elseif state(idx)==3
    k0=-
(sin(theta(idx))+theta(idx)*cos(theta(idx)))/(cos(theta(idx))-
theta(idx)*sin(theta(idx))); % 切线 0 斜率
elseif state(idx)==1
    k0=-1/((y01-y0)/(x01-x0));
else
    k0=-1/((y02-y0)/(x02-x0));
end
if state(idx+1)==0

```



```

k1=(sin(theta(idx+1))+theta(idx+1)*cos(theta(idx+1)))/(cos(theta(idx+1))-
theta(idx+1)*sin(theta(idx+1))); % 切线 1 斜率
elseif state(idx+1)==3
    k1=-
(sin(theta(idx+1))+theta(idx+1)*cos(theta(idx+1)))/(cos(theta(idx+1))-
theta(idx+1)*sin(theta(idx+1))); % 切线 1 斜率
elseif state(idx+1)==1
    k0=-1/((yo1-y1)/(xo1-x1));
else
    k0=-1/((yo2-y1)/(xo2-x1));
end
the0=abs(abs(atan(k0))-abs(atan(k)));
the1=abs(abs(atan(k))-abs(atan(k1)));
vdata(idx+1)=vdata(idx)*cos(the0)/cos(the1);
%% 绘图
if 0
    figure

[x_std,y_std]=get_xy(get_curve(b,boundary_theta:0.1:44*pi),boundary_theta
:0.1:44*pi);
    [x_circle,y_circle]=get_xy(return_radius,0:0.1:2*pi);
    S_x=[];
    S_y=[];
    for sss=0:0.1:calculate_return_length(return_radius)

[S_x(end+1),S_y(end+1)]=from_s_get_return_xy(sss,return_radius,rotate);
        end
        hold on
        plot(x_std,y_std,"-","Color","b")
        plot(-x_std,-y_std,"-","Color","g")
        plot(S_x,S_y,"-","Color","k")
        plot(x_circle,y_circle,"--
", 'Color',[0.1,0.7,0.1], 'LineWidth',1)
        plot(xdata(:),ydata(:),"-","Color",[0.1,0.3,0.9])
        plot(xdata(:),ydata(:),"*","Color",[0.9,0.1,0.1])
        axis square
    end
end

%     if abs(mod(t*dt,1))<1e-10
%         rec_x(:,round(t*dt)+1)=xdata(:,1)/100;
%         rec_y(:,round(t*dt)+1)=ydata(:,1)/100;
%         rec_v(:,t+1)=vdata(:,1);
%     end
%     if any(vdata>2)
%         bk=1;
%         break;
%     end
end
%% 二分更新
if bk==1
    % 可掉头,r 太大
    v_right=v;
else
    % 不可掉头,r 太小
    v_left=v;
end
end

```

```

end

%% 画图
figure
[x_std,y_std]=get_xy(get_curve(b,boundary_theta:0.1:44*pi),boundary_theta:0.1:44*pi);
[x_circle,y_circle]=get_xy(return_radius,0:0.1:2*pi);
S_x=[];
S_y=[];
for sss=0:0.1:calculate_return_length(return_radius)

[S_x(end+1),S_y(end+1)]=from_s_get_return_xy(sss,return_radius,rotate);
end
hold on
plot(x_std,y_std,"-","Color","b")
plot(-x_std,-y_std,"-","Color","g")
plot(S_x,S_y,"-","Color","k")
plot(x_circle,y_circle,"--","Color",[0.1,0.7,0.1],'LineWidth',1)
plot(xdata(:),ydata(:),"-","Color",[0.1,0.3,0.9])
plot(xdata(:),ydata(:),"*","Color",[0.9,0.1,0.1])
axis square

%% 螺线方程
function r=get_curve(b,theta)
    % in:b-螺距(cm);theta-极角(rad)
    % out:r-极径(cm)
    r=b.*theta./(2.*pi);
end

%% 螺线方程转平面直角
function [x,y]=get_xy(r,theta)
    x=r.*cos(theta);
    y=r.*sin(theta);
end

%% 螺线弧长方程
function s=get_s(b,theta)
    % in:b-螺距(cm);theta-极角(rad)
    % out:s-弧长(cm)
    s=b./(4*pi).*(theta.*(1+theta.^2).^0.5+log(theta+(1+theta.^2).^0.5));
end

%% 反-螺线弧长方程,获取龙头极角
function theta=inv_s(b,s)
    % in:b-螺距(cm);s-弧长(cm)
    % out:theta-极角(rad)
    %% 二分求解
    left=0;
    right=17*2*pi;
    err=right-left;
    while err>1e-8
        mid=(right+left)/2;
        mid_s=get_s(b,mid);
        if mid_s>s
            right=mid;
        else
            left=mid;
        end
    end
end

```

```

        err=right-left;
    end
    theta=(right+left)/2;
end

%% 盘入递推龙身
function t=get_body_theta(b,theta_bef,R)
    % in:b-螺距(cm);theta_bef-前节点极角(rad),R-龙身节点间长度
    (cm);rerurn_radius-掉头空间半径(cm)
    % out:t-下一节点极角(rad)
    [x0,y0]=get_xy(get_curve(b,theta_bef),theta_bef);
    p0=[x0,y0];
    ktemp=(sin(theta_bef)+theta_bef*cos(theta_bef))/(cos(theta_bef)-
theta_bef*sin(theta_bef)); % 切线斜率
    gamma=atan(ktemp);
    rho=get_rho(b,theta_bef);
    point_E1=[x0+rho*sin(gamma),y0-rho*cos(gamma)];
    point_E2=[x0-rho*sin(gamma),y0+rho*cos(gamma)];
    if norm(point_E1)<norm(point_E2)
        point_E=point_E1;
    else
        point_E=point_E2;
    end
    t=get_cross_point(theta_bef,point_E,rho,p0,R,R,x0,y0);
end

%% 盘出递推龙身
function t=get_body_theta_out(b,theta_bef,R)
    % in:b-螺距(cm);theta_bef-前节点极角(rad),R-龙身节点间长度
    (cm);rerurn_radius-掉头空间半径(cm)
    % out:t-下一节点极角(rad)
    [x0,y0]=get_xy(get_curve(b,theta_bef),theta_bef);
    p0=[x0,y0];
    ktemp=(sin(theta_bef)+theta_bef*cos(theta_bef))/(cos(theta_bef)-
theta_bef*sin(theta_bef)); % 切线斜率
    gamma=atan(ktemp);
    rho=get_rho(b,theta_bef);
    point_E1=[x0+rho*sin(gamma),y0-rho*cos(gamma)];
    point_E2=[x0-rho*sin(gamma),y0+rho*cos(gamma)];
    if norm(point_E1)<norm(point_E2)
        point_E=point_E1;
    else
        point_E=point_E2;
    end
    t=get_cross_point_out(theta_bef,point_E,rho,p0,R,R,x0,y0);
end

%% 求曲率半径
function rho=get_rho(b,theta)
    rho=b/(2*pi)*(theta^2+1)^(3/2)/(theta^2+2);
end

%% 求两圆交点
function theta=get_cross_point(theta_bef,p1,r1,p2,r2,R,x0,y0)
    % p1,p2-输入点横向量;r1,r2-圆半径
    d=norm(p1-p2);
    a=(r1^2-r2^2+d^2)/(2*d);
    h=sqrt(r1^2-a^2);

```

```

p=p1+a*(p2-p1)/d;
x1=p(1)+h*(p2(2)-p1(2))/d;
y1=p(2)-h*(p2(1)-p1(1))/d;
x2=p(1)-h*(p2(2)-p1(2))/d;
y2=p(2)+h*(p2(1)-p1(1))/d;
if y1*y2<0
    if x0>0
        if y1>0
            y=y1;
            x=x1;
        else
            x=x2;
            y=y2;
        end
    else
        if y1<0
            y=y1;
            x=x1;
        else
            x=x2;
            y=y2;
        end
    end
elseif x1*x2<0
    if y0>0
        if x1<0
            x=x1;
            y=y1;
        else
            x=x2;
            y=y2;
        end
    else
        if x1>0
            x=x1;
            y=y1;
        else
            x=x2;
            y=y2;
        end
    end
else
    if y0<0
        x=max(x1,x2);
    else
        x=min(x1,x2);
    end
    if x0>0
        y=max(y1,y2);
    else
        y=min(y1,y2);
    end
end
R1=norm(p2);
R2=norm([x,y]);
dtheta=acos((R1^2+R2^2-R^2)/(2*R1*R2));
theta=(dtheta)+theta_bef;
end

```

```

%% 求盘出两圆交点
function theta=get_cross_point_out(theta_bef,p1,r1,p2,r2,R,x0,y0)
    % p1,p2-输入点横向量;r1,r2-圆半径
    d=norm(p1-p2);
    a=(r1^2-r2^2+d^2)/(2*d);
    h=sqrt(r1^2-a^2);
    p=p1+a*(p2-p1)/d;
    x1=p(1)+h*(p2(2)-p1(2))/d;
    y1=p(2)-h*(p2(1)-p1(1))/d;
    x2=p(1)-h*(p2(2)-p1(2))/d;
    y2=p(2)+h*(p2(1)-p1(1))/d;
    if norm([x1,y1])<norm([x2,y2])
        x=x1;
        y=y1;
    else
        x=x2;
        y=y2;
    end
    R1=norm(p2);
    R2=norm([x,y]);
    dtheta=acos((R1^2+R2^2-R^2)/(2*R1*R2));
    theta=-(dtheta)+theta_bef;
end

%% 求掉头曲线 s 对应的两相对 theta
function [theta1,theta2]=get_return_theta(s,return_radius)
    r1=return_radius*2/3;
    r2=return_radius/3;
    theta1=-1;
    theta2=-1;
    s1=pi*r1;
    s2=pi*r2;
    if s<s1 % 在前圆内
        theta1=pi*s/s1;
    else % 在后圆内
        theta2=pi*(s-s1)/s2;
    end
end

%% 掉头曲线 s 自然坐标转直角坐标
function [x,y]=from_s_get_return_xy(s,return_radius,rotate)
    % in:s-自然坐标(cm),return_radius-掉头空间半径(cm),rotate-旋转角度(rad)
    % out:x,y-坐标(cm)
    [theta1,theta2]=get_return_theta(s,return_radius);
    r1=return_radius*2/3;
    r2=return_radius/3;
    if theta1 >= 0 % 在前圆内
        x1=r1*sin(theta1);
        y1=r1*cos(theta1)+return_radius/3;
    else % 在后圆内
        x1=-r2*sin(theta2);
        y1=r2*cos(theta2)-return_radius*2/3;
    end
    %% 旋转
    T=[cos(rotate),sin(rotate);
        -sin(rotate),cos(rotate)];
    res=T*[x1;y1];
    x=res(1);

```

```
    y=res(2);  
end  
function s=calculate_return_length(r)  
    % in:r-掉头空间半径(cm)  
    % out:s-掉头曲线长度(cm)  
    s=pi*r;  
end
```