

# 基于几何模型的板凳龙运动路径与碰撞问题

## 摘要

“板凳龙”是浙闽地区的一项元宵节习俗，在保证舞龙队能顺利的盘入和盘出的情况下，其所需面积越小，速度越快，则观赏效果越好。本文将盘龙的过程抽象为数学模型，在龙头、龙身和龙尾的尺寸数据已知的条件下研究不同螺距、不同龙头行进速度时，不同时刻下舞龙队各部位的位置和速度，为深入研究如何优化盘龙所需的面积和行进速度提供了参考。

对于问题一，为**建立舞龙队的运动状态模型**，本文利用**等距螺线极坐标方程、等距螺线弧长微分式**和龙头运动的物理计算式，微分得龙头极角的常微分方程，运用 **Runge-Kutta 算法**， $h = 0.01s$  求解得到其每个时刻的龙头前把手坐标。本文采用建立位置迭代公式的方法，由龙头前把手位置，通过**二分法（精度取  $10^{-6}$ ）**求出龙身每个前把手和龙尾前后把手的位置，求解得到龙身各把手位置。为求得各时刻各部位的前把手和龙尾后把手的速度，本文根据角速度递推公式和**等距螺线的线速度定义式**，建立相邻两把手线速度迭代公式。结合龙头前把手速度恒为  $1m/s$ ，求解出各个时刻舞龙队的线速度。部分结果见表 2 和表 3

对于问题二，我们建立了**龙头碰撞检测模型**。本文首先分析整个舞龙队盘入的过程，**反证法**证明碰撞一定发生在龙头左前顶点与龙身右侧长边之间。在问题一中本文确定了各个时刻把手的位置，使用坐标变换矩阵确定顶点的位置。根据几何关系，当龙头左前顶到外圈龙身板凳前后把手所在直线的距离小于  $0.15m$  时，判定发生碰撞。先用**二分法**确定龙头碰撞时极角，通过动画模拟，发现龙头与第 9 节龙身发生碰撞，进而通过两次 **Runge-Kutta 算法（第 2 次精度达到  $10^{-6}$ ）**计算得不能再到继续盘入的时间为 **412.4738s**。

对于问题三，本文建立了**等距螺线螺距优化模型**。为了方便计算，以板凳几何中心为基准重新完成四顶点坐标表示。在螺距已知的情况下，结合问题一、问题二的模型，引入**分离轴定理**作碰撞检测条件，本文能够得出在不同的螺距情况下舞龙队盘入的终止位置。因此，本文首先以螺距最小为目标函数建立**单目标优化模型**，以碰撞时间约束和几何约束为约束条件，使用**二分法**求得使终止时刻龙头前把手恰好位于调头空间边界的螺距为  $0.417137m$ 。

对于问题四，首先，我们利用调头曲线的三个**相切关系**对调头空间做几何分析，得出**S 形曲线长度优化模型**，发现其长度与半径比无关；然后根据关键位置的坐标（关键坐标见），**分类讨论**得到 10 种不同运动情况下的各把手位置和速度的迭代公式并用递推法求出数值结果。

**关键字：** 等距螺线极坐标方程   Runge-Kutta 算法   二分法   分离轴定理

## 一、问题重述

假设有一由 223 节板凳组成的板凳龙，其中第 1 节为龙头，接着 221 节龙身，最后 1 节为龙尾。龙头的板长为 341cm，龙身与龙尾的板长均为 220cm，板凳的板宽均为 30cm，相邻两条板凳钻孔并通过把手连接，孔径为 5.5cm，钻孔中心距最近的板头 27.5cm。

### 问题一重述

舞龙队使用上述板凳龙，沿螺距为 55cm 的等距螺线顺时针盘入，各把手中心均位于螺线上。龙头前把手的线速度保持 1 m/s 不变。0s 时，龙头位于螺线第 16 圈 A 点处。求出 0~300s 内每秒每个板凳前把手及龙尾后把手中心的位置和速度，结果保留 6 位小数。

### 问题二重述

舞龙队还沿螺距为 55cm 的等距螺线顺时针盘入，求解舞龙队盘入的终止时刻，即舞龙队使得板凳之间发生碰撞的时刻，并给出此时舞龙队每个板凳前把手及龙尾后把手中心的位置和速度。结果保留 6 位小数。

### 问题三重述

假定一个以  $d(i)$  为螺距的等距螺线，下称盘入曲线。舞龙队沿盘入曲线盘入后进入调头空间，再由顺时针盘入调头切换为逆时针盘出。已知调头空间是以螺线中心为圆心、直径为 9m 的圆形区域，求解最小螺距  $d(i)$ ，使得龙头前把手能够沿着盘入曲线盘入到调头空间的边界。

### 问题四重述

保持龙头前把手速度始终为 1m/s，将盘入螺线的螺距更改为 1.7m，盘出与盘入螺线中心对称。舞龙队在调头空间内调头，路径为两段相切圆弧所构成的 S 形曲线，前段圆弧半径为后段圆弧的两倍，且与螺线相切。问能否调整圆弧以缩短调头路径，同时保持各部分相切。还要求出以调头开始时间为零时刻，从 -100 s 开始到 100 s 为止，每秒舞龙队的位置和速度。

### 问题五重述

在问题四的基础上，在龙头速度在运动过程中保持不变的情况下，确定最大龙头线速度，使所有把手的线速度都不大于 2m/s。

## 二、模型假设

1. 理想化连接：将把手中心处连接视为理想铰链连接，忽略其摩擦和间隙
2. 刚体假设：板凳尺寸符合数据，无弹性变形，保证计算的一致性
3. 忽略板凳厚度：只考虑二维平面上的长和宽
4. 忽略外部干扰：不考虑外部干扰，如地面坡度、舞者操作误差、天气等带来的影响

### 三、符号说明

本模型所用主要变量符号及意义如表 1 所示。

表 1 主要变量符号说明表

符号	定义	单位
$\rho$	极半径	$m$
$\theta$	极角	$rad$
$t$	时间	$s$
$s$	运动弧长	$m$
$V$	把手线速度	$m/s$
$l_i$	第 $i$ 节（龙头为 1）板凳的长度	$m$
$\theta'$	把手角速度	$rad/s$
$\alpha$	板凳中轴线斜率角	$rad$
$k$	板凳中轴线斜率	$/$
$d(t)$	龙头左前顶点 $A_1$ 到龙身板凳中轴线的距离	$m$
$\vec{e}$	分离轴单位向量	$/$

### 四、问题分析

#### 4.1 问题一分析

由于龙头前把手沿盘入螺线的线速度恒定，结合已知螺线的极坐标方程，采用积分的方式去确定其某段时间内走过的弧长与走过的角度的关系，微分得龙头极角的常微分方程，**Runge-Kutta 算法**（ODE45， $h = 0.01s$ ）求解得到每一秒龙头的位置坐标。由此，本文能够利用极坐标方程与每节板凳的数据，建立起由前把手中心位置得到后把手中心位置的迭代公式，并用**二分法**求解所有把手各时刻位置。利用角速度递推公式和等距螺线的线速度定义式，建立相邻两把手线速度迭代公式。结合龙头前把手速度恒为  $1m/s$ ，求解出各个时刻舞龙队的线速度。

## 4.2 问题二分析

对于问题二，在舞龙队沿着螺线盘入的过程中，随着龙头前把手逐渐接近圆心，由于螺距的限制，舞龙队的板凳会在某时刻发生了碰撞。经过分析，因为龙身与龙尾所有板凳形状完全相同，而且在盘入的轨迹也相同。因此这些板凳是否发生碰撞只需要考虑龙头是否与其他板凳发生碰撞。因为如果沿盘入曲线盘入过程中，龙头未与其他板凳发生碰撞，后面的龙身就不会与其他板凳发生碰撞。故显然**只有龙头板凳左前顶点，可能会与板凳右侧边相撞。**

在问题一中，我们已经确定了各个时刻各把手中心位置，并且已知两块板凳把手中心与板凳的顶点的相对位置是固定的，本文由此计算出各时刻龙头左前顶点的位置坐标。同时，在龙头外一层螺线上分布着一圈板凳，板凳的两个把手中心连接形成的线段是板凳的一条中轴线，并且可以利用问题一中所得到的坐标求出中轴线的解析式。因此本文利用顶点到中轴线的距离来判断碰撞与否，当距离小于板凳的宽度的一半时说明已经发生了碰撞。建立以上碰撞检测模型后，用**二分法**计算碰撞时龙头极角，动画模拟找出碰撞的龙身，运用 **Runge-Kutta 算法** (ODE45) 求出碰撞的时间。最后用问题一模型求出碰撞时各把手位置和速度。

## 4.3 问题三分析

问题三实质上是使龙头前把手到达调头空间前不碰撞的最小螺距优化问题。结合问题一和问题二的模型并引入基于分离轴定理的碰撞检验模型，得出不同螺距下，盘入的碰撞位置。本文建立以螺距最短为目标建立单目标优化模型，约束条件包含等距螺线几何约束、碰撞时间约束和螺距上下界约束，由**二分法**求出精度为  $10^{-6}$  的最短螺距。

## 4.4 问题四分析

问题四在问题三的基础上，引入了调头曲线的概念，可转化为两个子问题：

1. 建立 S 形曲线长度优化模型。
2. 各把手位置和速度求解。

对于子问 1，首先，我们对调头区域做几何分析，接着建立曲线长度优化模型，发现其**长度与半径比无关**。对于子问 2，基于求得的调头曲线方程，对问题一中的舞龙队运动状态模型加以更改，分类讨论 10 种不同运动情况，得到不同的位置、速度递推公式，利用**递推法**求得了各把手的运动状态。

## 五、模型建立与求解

### 5.1 问题一舞龙队的运动状态模型的建立与求解

#### 5.1.1 舞龙队的运动状态模型的建立

##### 螺旋线方程

第  $i$  节板凳前把手坐标  $(u_i(t), v_i(t))$ ; 第  $i$  节板凳前把手极坐标  $(\rho_i(t), \theta_i(t))$

由螺距  $d=0.55\text{m}$  和 A 点极坐标:  $(8.8, 32\pi)$  可得螺旋线方程:

$$\rho(\theta) = \frac{11}{40\pi}\theta \quad (1)$$

转换为直角坐标为:

$$\begin{cases} u_i(t) = \rho \cos \theta \\ v_i(t) = \rho \sin \theta \end{cases} \quad (2)$$

##### 龙头运动轨迹

龙头运动弧长微分式:

$$ds = \frac{11}{40\pi} \sqrt{1 + \theta^2} d\theta \quad (3)$$

龙头运动弧长物理计算式 (弧长 = 线速度  $\times$  时间) :

$$s(t) = Vt \quad t \in [0, 300] \quad (4)$$

对式 (3) 做上限为  $32\pi$  (起点), 下限为  $\theta_1(t)$  ( $t$  时终点) 的积分:

$$s(t) = \frac{11}{40\pi} \int_{\theta_1(t)}^{32\pi} \sqrt{1 + \theta^2} d\theta = Vt \quad t \in [0, 300] \quad (5)$$

式 (5) 对  $t$  求导得一阶常微分方程

$$\theta_1'(t) = -\frac{40\pi}{11\sqrt{1 + \theta_1^2}} \quad (6)$$

##### 龙身位置递推公式

平面直角坐标系下前后两个板凳的前把手位置关系 (勾股定理):

$$(u_{i+1}(t) - u_i(t))^2 + (v_{i+1}(t) - v_i(t))^2 = l_i^2, i = 1, 2, \dots, 223 \quad (7)$$

将式 (2) 和式 (1) 代入上式 (7) 得极坐标系下前后两个板凳的前把手位置关系:

$$\left(\frac{0.55}{2\pi}\right)^2 (\theta_{i+1}^2(t) + \theta_i^2(t) - 2\theta_{i+1}(t)\theta_i(t)\cos(\theta_{i+1}(t) - \theta_i(t))) = l_i^2 \quad (8)$$

##### 龙身速度递推公式

式 (8) 对  $t$  求导得龙身各前把手角速度递推公式:

$$\theta'_{i+1}(t) = \frac{-\theta_i(t) + \theta_{i+1}(t) \cos(\theta_{i+1}(t) - \theta_i(t)) + \theta_{i+1}(t)\theta_i(t) \sin(\theta_{i+1}(t) - \theta_i(t))}{\theta_{i+1}(t) - \theta_i(t) \cos(\theta_{i+1}(t) - \theta_i(t)) + \theta_{i+1}(t)\theta_i(t) \sin(\theta_{i+1}(t) - \theta_i(t))} \theta'_i(t) \quad (9)$$

设角速度递推系数为  $g_{i+1}$

$$g_{i+1} = \frac{-\theta_i(t) + \theta_{i+1}(t) \cos(\theta_{i+1}(t) - \theta_i(t)) + \theta_{i+1}(t)\theta_i(t) \sin(\theta_{i+1}(t) - \theta_i(t))}{\theta_{i+1}(t) - \theta_i(t) \cos(\theta_{i+1}(t) - \theta_i(t)) + \theta_{i+1}(t)\theta_i(t) \sin(\theta_{i+1}(t) - \theta_i(t))} \quad (10)$$

迭代可得角速度通式:

$$\theta'_{i+1}(t) = \theta'_1(t) \prod_{j=1}^{i+1} g_j = -\frac{40\pi}{11\sqrt{1+\theta_1^2}} \prod_{j=1}^{i+1} g_j \quad (11)$$

根据等距螺线的线速度定义:

$$V = \sqrt{\left(\frac{d\rho}{dt}\right)^2 + (\rho \cdot \theta')^2} \quad (12)$$

将式 (1) 代入式 (12) 得等距螺线运动的线速度公式:

$$V = \frac{11}{40\pi} \theta' \sqrt{1 + \theta^2} \quad (13)$$

将式 (11) 代入式 (13) 得龙身前把手的线速度通式和递推公式 (因为不考虑龙身的线速度方向, 仅关注其大小, 故保持  $V$  值为正):

$$V_{i+1}(t) = \sqrt{\frac{1 + \theta_{i+1}^2}{1 + \theta_1^2}} \prod_{j=1}^{i+1} |g_j| \quad (14)$$

$$V_{i+1}(t) = \sqrt{\frac{1 + \theta_{i+1}^2}{1 + \theta_i^2}} |g_{i+1}| V_i(t) \quad (15)$$

综上, 舞龙队的运动状态模型为:

$$\text{位置: } \begin{cases} \theta'_1(t) = -\frac{40\pi}{11\sqrt{1+\theta_1^2}} \\ \left(\frac{0.55}{2\pi}\right)^2 (\theta_{i+1}^2(t) + \theta_i^2(t) - 2\theta_{i+1}(t)\theta_i(t)\cos(\theta_{i+1}(t) - \theta_i(t))) = l_i^2 \\ \theta_1(0) = 32\pi \\ l_1 = 2.86 \\ l_i = 1.65 \quad (i = 2, 3, \dots, 223) \end{cases} \quad (16)$$

$$\text{速度: } \begin{cases} V_{i+1}(t) = \sqrt{\frac{1+\theta_{i+1}^2}{1+\theta_1^2}} \prod_{j=1}^{i+1} |g_j| \\ V_1 = 1 \\ (\frac{0.55}{2\pi})^2 (\theta_{i+1}^2(t) + \theta_i^2(t) - 2\theta_{i+1}(t)\theta_i(t)\cos(\theta_{i+1}(t) - \theta_i(t))) = l_i^2 \\ \theta_1(0) = 32\pi \\ l_1 = 2.86 \\ l_i = 1.65 \quad (i = 2, 3, \dots, 223) \\ g_{i+1} = \frac{-\theta_i(t) + \theta_{i+1}(t) \cos(\theta_{i+1}(t) - \theta_i(t)) + \theta_{i+1}(t)\theta_i(t) \sin(\theta_{i+1}(t) - \theta_i(t))}{\theta_{i+1}(t) - \theta_i(t) \cos(\theta_{i+1}(t) - \theta_i(t)) + \theta_{i+1}(t)\theta_i(t) \sin(\theta_{i+1}(t) - \theta_i(t))} \end{cases} \quad (17)$$

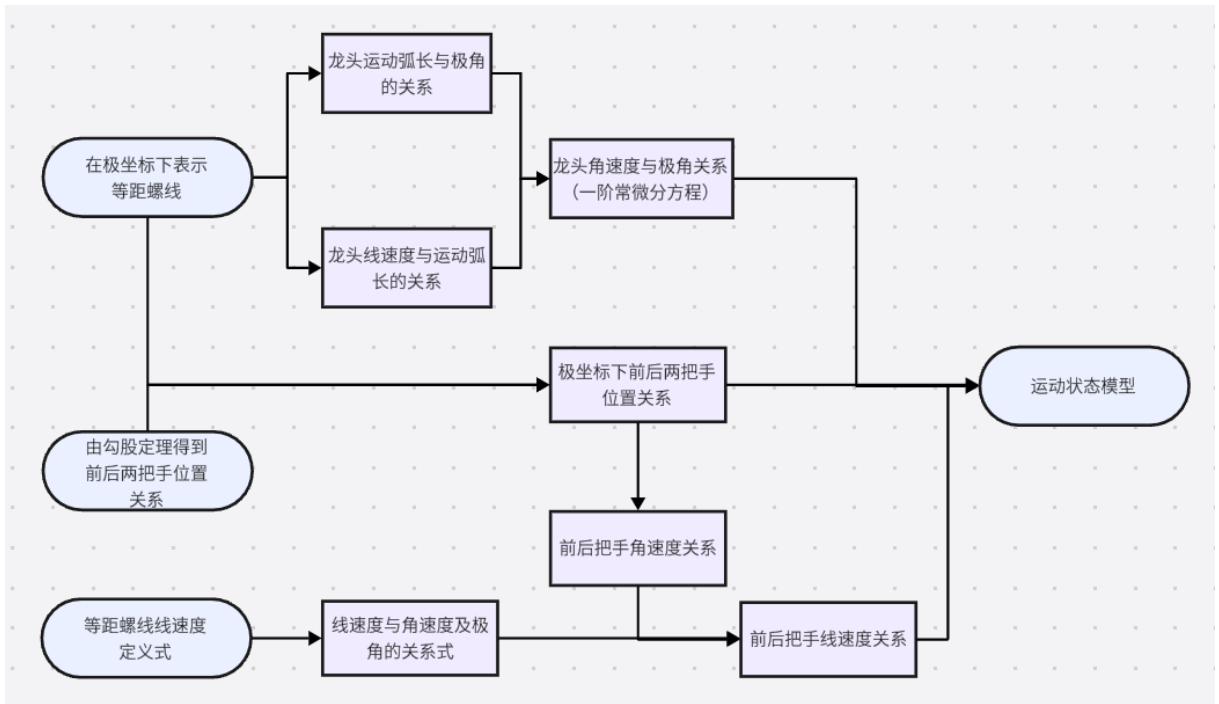


图 1 运动状态模型建立流程图

### 5.1.2 舞龙队的运动状态模型的求解

#### 位置求解

##### (1) 龙头位置求解

龙头前把手沿等距螺线运动的一阶常微分方程如式 (6) 所示, 已知  $\theta_1(0)=32\pi$ , 使用 ODE45 方法, 步长为 0.01s, 得到龙头 0~300s 的极角位置。

##### (2) ODE45 方法

ODE45 是 MATLAB 中用于求解常微分方程初值问题的最常用指令之一。其原理是用 4 阶和 5 阶 Runge-Kutta 公式各算一次下一步的近似值, 通过比较两者差异来估计局部截断误差, 再据此自适应地放大或缩小步长, 兼顾精度和效率。

### (3) 龙身位置求解

将龙身位置迭代公式即式 (8) 转换为：

$$\left(\frac{0.55}{2\pi}\right)^2(\theta_{i+1}^2(t) + \theta_i^2(t) - 2\theta_{i+1}(t)\theta_i(t)\cos(\theta_{i+1}(t) - \theta_i(t))) - l_i^2 = 0 \quad (18)$$

其中  $l_1=2.86$ ,  $l_i = 1.65$  ( $i = 2, 3, \dots, 224$ )

利用二分法, 在同一时刻下对 (18) 在 0~300 秒范围内每隔 1 秒龙头后面的每一节龙身前把手和龙尾后把手的极角, 转化为二维直角坐标系。部分结果在结果展示与分析中给出。

### 速度求解

#### (1) 龙身速度求解

通过推导出的把手线速度递推公式 (式 (15)), 结合已知条件 ( $V_1 = 1$ ) 和上面求得的各项极角, 可以通过下式 (19) 来对 300 秒内每秒各把手的线速度求解。部分结果见结果展示与分析。

$$\begin{cases} V_{i+1}(t) = \sqrt{\frac{1+\theta_{i+1}^2}{1+\theta_1^2}} \prod_{j=1}^{i+1} |g_j| \\ V_1 = 1 \\ g_{i+1} = \frac{-\theta_i(t)+\theta_{i+1}(t)\cos(\theta_{i+1}(t)-\theta_i(t))+\theta_{i+1}(t)\theta_i(t)\sin(\theta_{i+1}(t)-\theta_i(t))}{\theta_{i+1}(t)-\theta_i(t)\cos(\theta_{i+1}(t)-\theta_i(t))+\theta_{i+1}(t)\theta_i(t)\sin(\theta_{i+1}(t)-\theta_i(t))} \end{cases} \quad (19)$$



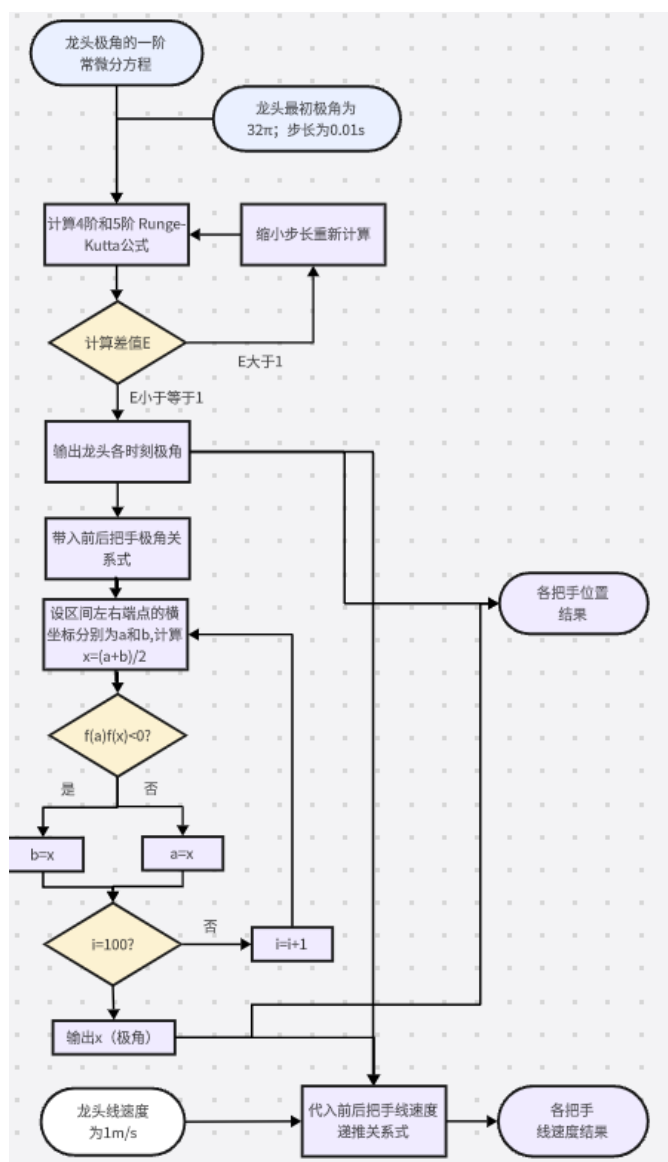


图2 运动状态模型求解流程图

## 结果展示与分析

第一问求解结果如下：

**step1:Runge-Kutta 方法求出各时刻龙头前把手极角。**以  $32\pi$  为初值,  $0.01s$  为步长计算龙头极角常微分方程的 4 阶和 5 阶 Runge-Kutta 公式结果, 若差值大于 1 则缩小步长重新计算, 若差值小于或等于 1, 输出结果为极角 (精度  $10^{-6}$ )。

**step2: 二分法求出各时刻各把手极角。**先将上一步结果代入前后把手极角关系式, 再用二分法在龙头极角与龙头极角  $+\pi$  范围内逼近零点。

**step3: 达到搜索次数后, 结束二分法。**当搜索区间更新 100 次后, 本文认为精度符合要求。此时, 我们把第 100 次搜索区间中点近似作待求零点, 输出该值并结束二分法。

**step4: 各把手位置求解。**将 step1 和 step3 的结果按坐标转换公式, 转换成二维平面直角坐标系的位置坐标, 填入结果表格。

**step5: 各把手线速度求解。**将 step1 和 step3 的结果代入前后把手速度迭代公式, 已知龙头前把手线速度为  $1m/s$ , 依次迭代可得各把手线速度结果, 填入对应表格。

表 2 板凳龙第一问位置结果

位置	0 s	60 s	120 s	180 s	240 s	300 s
龙头 x (m)	8.800000	5.799209	-4.084887	-2.963609	2.594494	4.420274
龙头 y (m)	0.000000	-5.771092	-6.304479	6.094780	-5.356743	2.320429
第 1 节龙身 x (m)	8.363825	7.456757	-1.445474	-5.237117	4.821220	2.459491
第 1 节龙身 y (m)	2.826540	-3.440401	-7.405882	4.359628	-3.561949	4.402474
第 51 节龙身 x (m)	-9.518734	-8.686321	-5.543166	2.890468	5.979992	-6.301349
第 51 节龙身 y (m)	1.341126	2.540094	6.377932	7.249284	-3.827786	0.465790
第 101 节龙身 x (m)	2.914024	5.687102	5.361975	1.898747	-4.917405	-6.237770
第 101 节龙身 y (m)	-9.918300	-8.001393	-7.557613	-8.471624	-6.379848	3.935935
第 151 节龙身 x (m)	10.861712	6.682329	2.388646	1.005193	2.965431	7.040723
第 151 节龙身 y (m)	1.828844	8.134529	9.727440	9.424746	8.399701	4.393042
第 201 节龙身 x (m)	4.555040	-6.619636	-10.627231	-9.287707	-7.457186	-7.458651
第 201 节龙身 y (m)	10.725145	9.025590	1.359696	-4.246703	-6.180683	-5.263400
龙尾（后） x (m)	-5.305385	7.364528	10.974337	7.383885	3.241114	1.785032
龙尾（后） y (m)	-10.676614	-8.798016	0.843639	7.492381	9.469314	9.301164

表 3 板凳龙第一问速度结果

位置	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)	0.999971	0.999961	0.999945	0.999917	0.999859	0.999709
第 51 节龙身 (m/s)	0.999742	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

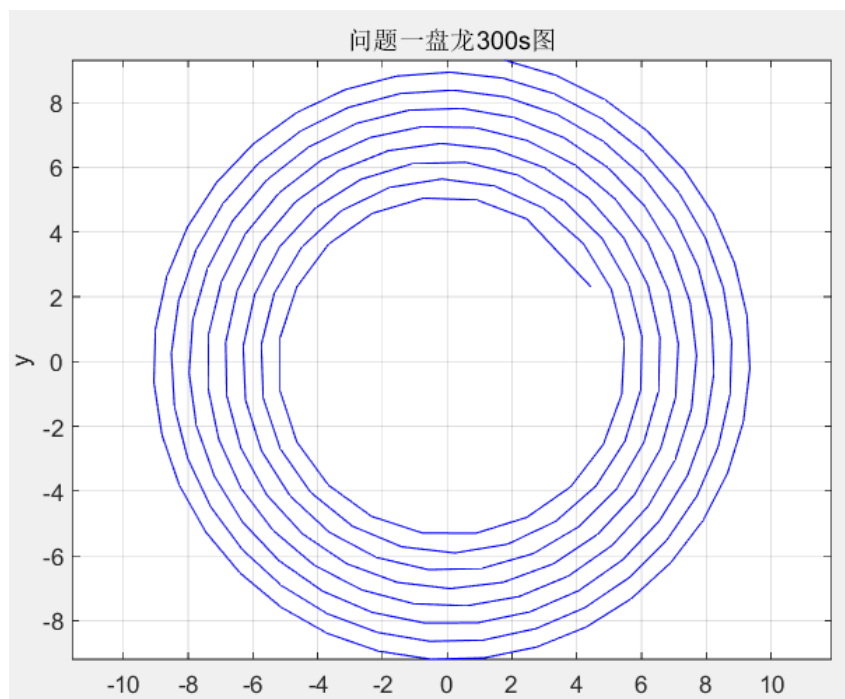


图3 问题一盘龙第300秒情况

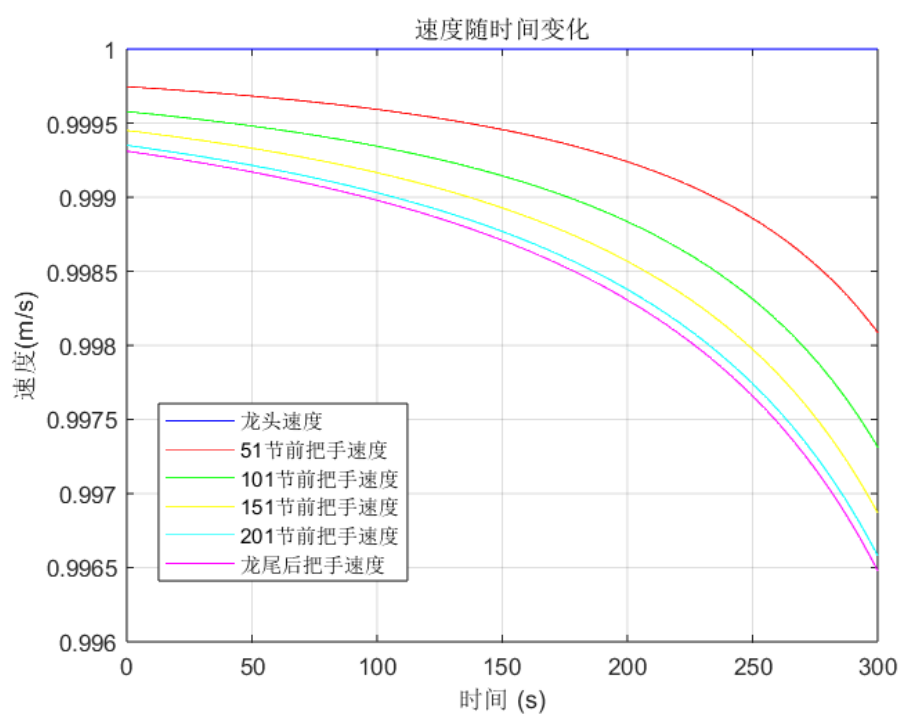


图4 部分把手速度随时间的变化

## 5.2 问题二: 碰撞模型的建立与求解

### 5.2.1 碰撞模型的建立

#### (1) 碰撞位置证明

首次碰撞必然发生在龙头与龙身之间。下面用反证法证明此结论：

反设：假设首次碰撞不发生在龙头与某龙身上。

归谬：不妨设与把手  $M_i(r_i(t), \theta_i(t))$  所对应的第  $i$  个非龙头板凳在  $t$  时刻与其它板凳发生了碰撞。由于龙头后面的节点运动具有滞后性, 即后一个把手的节点总是在重复前一节点运动过的轨迹。因此, 如果  $t$  时刻第  $i$  个非龙头板凳发生了碰撞, 那么在该板凳到达碰撞点前, 必有第  $i-1$  个非龙头板凳在  $t$  时刻之前先发生了碰撞。在此逻辑下不断递推, 可得出碰撞最先发生于第 1 个板凳, 即龙头。因此, 该假设并不成立。

结论：首次碰撞必然发生在龙头与龙身上。

推论：由于舞龙队顺时针盘入, 可以根据几何关系得到碰撞发生在龙头左前顶点与某龙身的右侧长边。

#### (2) 龙头四顶点表示

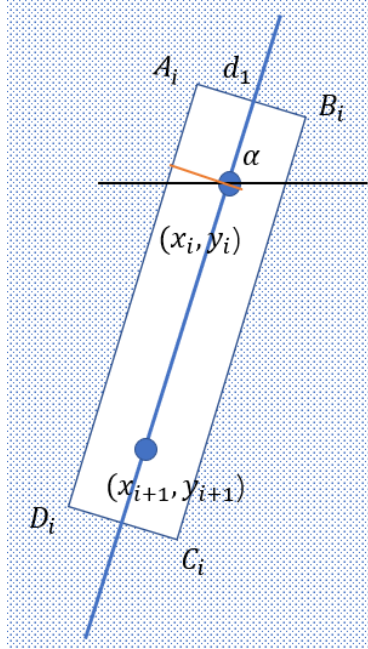


图5 极坐标系中的阿基米德螺旋线

为了方便研究板凳碰撞问题, 需要以前把手坐标  $(x_i, y_i)$  和  $\alpha$  为基准计算四个顶点 (如??所示) 的坐标。根据三角函数和坐标变换矩阵相关的几何知识得到如下四个顶点坐标计算矩阵：

•  $A_i$  :

$$\begin{pmatrix} x_{Ai} - x_i \\ y_{Ai} - y_i \end{pmatrix} = \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} d_2 \\ d_1 \end{pmatrix}$$

•  $B_i$  :

$$\begin{pmatrix} x_{Bi} - x_i \\ y_{Bi} - y_i \end{pmatrix} = \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} d_2 \\ -d_1 \end{pmatrix}$$

•  $C_i$  :

$$\begin{pmatrix} x_{Ci} - x_{i+1} \\ y_{Ci} - y_{i+1} \end{pmatrix} = \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} -d_2 \\ -d_1 \end{pmatrix}$$

•  $D_i$  :

$$\begin{pmatrix} x_{Di} - x_{i+1} \\ y_{Di} - y_{i+1} \end{pmatrix} = \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} -d_2 \\ d_1 \end{pmatrix}$$

将以上四个矩阵化简为下面四个顶点坐标方程组：

•

$$\begin{cases} x_{A_i} = x_i + d_2 \cos \alpha + d_1 \sin \alpha \\ y_{A_i} = y_i - d_2 \sin \alpha + d_1 \cos \alpha \end{cases} \quad \begin{cases} x_{B_i} = x_i + d_2 \cos \alpha - d_1 \sin \alpha \\ y_{B_i} = y_i - d_2 \sin \alpha - d_1 \cos \alpha \end{cases}$$

•

$$\begin{cases} x_{C_i} = x_i - d_2 \cos \alpha - d_1 \sin \alpha \\ y_{C_i} = y_i + d_2 \sin \alpha - d_1 \cos \alpha \end{cases} \quad \begin{cases} x_{D_i} = x_i - d_2 \cos \alpha + d_1 \sin \alpha \\ y_{D_i} = y_i + d_2 \sin \alpha + d_1 \cos \alpha \end{cases}$$

$$\text{其中: } \begin{cases} i = 1, 2, \dots, 224 \\ d_1 = 0.15m \\ d_2 = 0.275m \\ \text{偏转角: } \alpha = \arctan \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = \arctan \frac{v_{i+1}(t) - v_i(t)}{u_{i+1}(t) - u_i(t)} \end{cases}$$

因为碰撞发生在龙头左前顶点  $A_1$  与某个龙身板凳的右侧  $B_i C_i$ ，所以求出  $A_1$  的坐标表达式：

$$\begin{cases} x_{A_1} = u_1(t) + d_2 \cos \alpha + d_1 \sin \alpha = \frac{11}{40\pi} \theta_1(t) \cos \theta_1(t) + d_2 \cos \arctan k + d_1 \sin \arctan k \\ y_{A_1} = v_1(t) - d_2 \sin \alpha + d_1 \cos \alpha = \frac{11}{40\pi} \theta_1(t) \sin \theta_1(t) - d_2 \sin \arctan k + d_1 \cos \arctan k \end{cases} \quad (20)$$

### (3) 碰撞检测

当  $A_1$  到某节龙身的  $B_i C_i$  距离为 0 时，即  $A_1$  到某节龙身中轴线的距离为龙身板凳宽度的一半时，发生碰撞。

因为中轴线过  $(x_i, y_i)$  且斜率为  $\frac{y_{i+1} - y_i}{x_{i+1} - x_i}$  故  $t$  时刻第  $i$  节龙身中轴线的方程为：

$$y - y_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} (x - x_i) \quad (21)$$

即

$$y - v_i(t) = k(x - u_i(t)) \quad (22)$$

其中， $k = \frac{v_{i+1}(t) - v_i(t)}{u_{i+1}(t) - u_i(t)} = \frac{\theta_{i+1}(t) \sin \theta_{i+1}(t) - \theta_i(t) \sin \theta_i(t)}{\theta_{i+1}(t) \cos \theta_{i+1}(t) - \theta_i(t) \cos \theta_i(t)}$  (代入式 (1) 和式 (2))

$$\text{第 } i \text{ 节板凳的前把手坐标 } \begin{cases} u_i(t) = \frac{11}{40\pi} \theta_i(t) \cos \theta_i(t) \\ v_i(t) = \frac{11}{40\pi} \theta_i(t) \sin \theta_i(t) \end{cases} \quad (23)$$

龙头左前顶点  $A_1$  到第  $i$  节龙身中轴线的距离为：

$$d(t) = \frac{|k(x_{A_1} - u_i(t)) - y_{A_1} + v_i(t)|}{\sqrt{1 + k^2}} \quad (24)$$

综上，得到用距离检测碰撞的距离计算方程组

$$\left\{ \begin{array}{l} \text{第 } i \text{ 节板凳的前把手坐标:} \\ u_i(t) = \frac{11}{40\pi} \theta_i(t) \cos \theta_i(t) \\ v_i(t) = \frac{11}{40\pi} \theta_i(t) \sin \theta_i(t) \\ \text{中轴线 } K \text{ 与极角 } \theta \text{ 的关系:} \\ k = \frac{\theta_{i+1}(t) \sin \theta_{i+1}(t) - \theta_i(t) \sin \theta_i(t)}{\theta_{i+1}(t) \cos \theta_{i+1}(t) - \theta_i(t) \cos \theta_i(t)} \\ \text{龙头 } A_1 \text{ 极坐标表达式:} \\ x_{A_1} = \frac{11}{40\pi} \theta_1(t) \cos \theta_1(t) + d_2 \cos \arctan k + d_1 \sin \arctan k \\ y_{A_1} = \frac{11}{40\pi} \theta_1(t) \sin \theta_1(t) - d_2 \sin \arctan k + d_1 \cos \arctan k \\ \text{两把手极角与两把手距离关系式:} \\ (\frac{11}{40\pi})^2 (\theta_{i+1}^2(t) + \theta_i^2(t) - 2\theta_{i+1}(t)\theta_i(t)\cos(\theta_{i+1}(t) - \theta_i(t))) = l_i^2 \\ \text{龙头左前顶点 } A_1 \text{ 到第 } i \text{ 节龙身中轴线的距离公式:} \\ d(t) = \frac{|k(x_{A_1} - u_i(t)) - y_{A_1} + v_i(t)|}{\sqrt{1+k^2}} \end{array} \right. \quad (25)$$

#### (4) 碰撞模型结果

建立以碰撞时间最大为目标函数，以碰撞检测和碰撞发生范围为约束条件的单目标优化模型。

目标函数：

$$f(t) = \max(t) \quad (26)$$

约束条件：

$$s.t. \begin{cases} d(t) \geq 0.15 \\ \pi \leq \theta_i - \theta \leq 3\pi \end{cases}$$

#### (5) 碰撞模型建立流程图

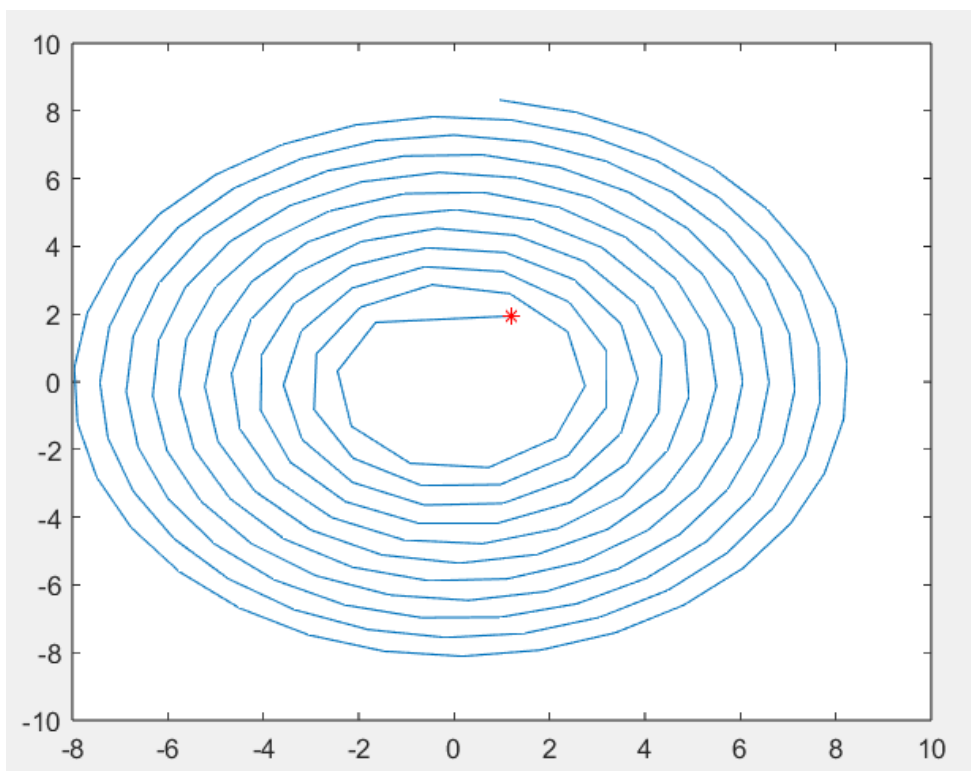


图7 盘龙动画模拟结果

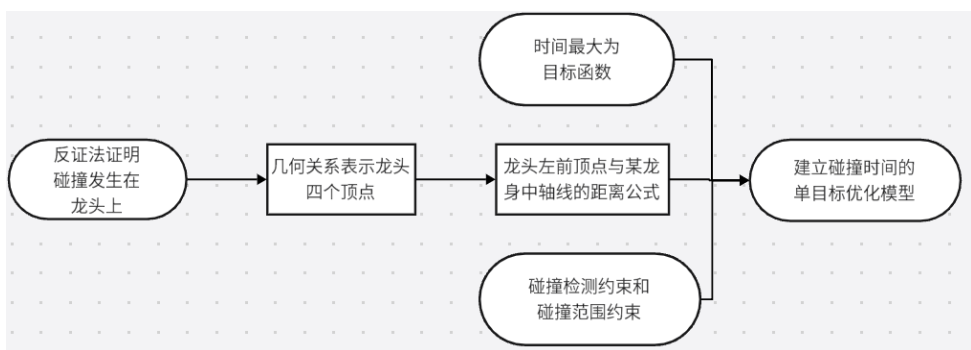


图6 碰撞模型建立流程图

### 5.2.2 碰撞模型求解

#### (1) 预测碰撞位置

通过问题一的运动状态模型进行动画模拟。通过盘龙动画模拟估算得出龙头左前顶点

$A_1$  会与第 8 节龙身 ( $i = 9$ ) 碰撞, 因此距离计算式改为:

$$\left\{ \begin{array}{l} \text{第九节板凳的前把手坐标:} \\ u_9(t) = \frac{11}{40\pi} \theta_9(t) \cos \theta_9(t) \\ v_9(t) = \frac{11}{40\pi} \theta_9(t) \sin \theta_9(t) \\ \text{第九节板凳的中轴线斜率:} \\ k_9 = \frac{\theta_{10}(t) \sin \theta_{10}(t) - \theta_9(t) \sin \theta_9(t)}{\theta_{10}(t) \cos \theta_{10}(t) - \theta_9(t) \cos \theta_9(t)} \\ \text{龙头左前顶点坐标:} \\ x_{A_1} = \frac{11}{40\pi} \theta_1(t) \cos \theta_1(t) + d_2 \cos \arctan k + d_1 \sin \arctan k \\ y_{A_1} = \frac{11}{40\pi} \theta_1(t) \sin \theta_1(t) - d_2 \sin \arctan k + d_1 \cos \arctan k \\ \text{第九节板凳极角递推公式:} \\ (\frac{11}{40\pi})^2 (\theta_{10}^2(t) + \theta_9^2(t) - 2\theta_{10}(t)\theta_9(t)\cos(\theta_{10}(t) - \theta_9(t))) = l_9^2 \\ \text{龙头左前顶点与第九节板凳中轴线距离:} \\ d(t) = \frac{|k(x_{A_1} - u_9(t)) - y_{A_1} + v_9(t)|}{\sqrt{1+k^2}} \end{array} \right. \quad (27)$$

由问题一可知,  $\theta_{10}(t)$  和  $\theta_9(t)$  可根据  $\theta_1(t)$  迭代得到, 因此, 上述方程组只与  $\theta_1(t)$  有关通过二分法求第 9 节板凳的极角。

## (2) 碰撞时间和碰撞时龙头极角求解

对问题一中常微分方程式 (6), 通过 Runge-Kutta 算法先在 0~420s 内步长为 1, 求解得碰撞发生在 412 与 413 秒之间。再次使用 Runge-Kutta 算法在 412 与 413 之间步长为  $10^{-6}$  求出碰撞时间为 **412.47384s** 及龙头碰撞时极角 = 26.146535rad。

## (3) 其他板凳的位置和速度求解

运用问题一的运动状态模型, 带入时间 = 412.47384s 得到所有板凳在碰撞时的位置和距离。部分结果见结果展示。

## (4) 求解流程图

结果展示



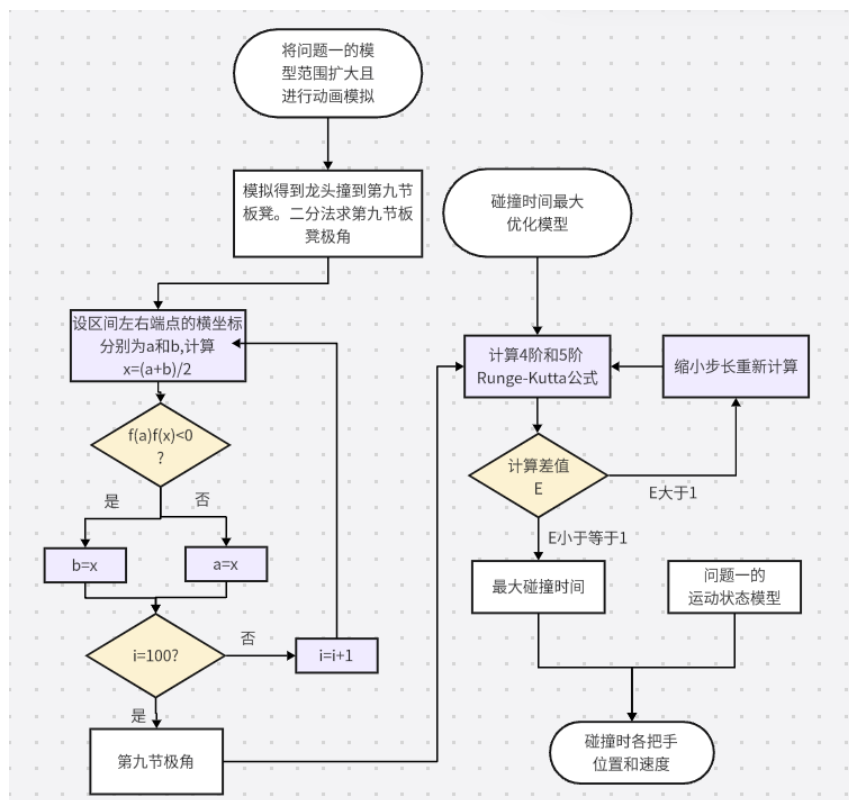


图 8 问题二求解流程图

表 4 舞龙各把手位置与速度数据

节段名称	横坐标 $x$ (m)	纵坐标 $y$ (m)	速度 (m/s)
龙头	1.20993	1.942784	1.0
第 1 节龙身	-1.643792	1.753399	0.991551
第 51 节龙身	1.281201	4.326588	0.976858
第 101 节龙身	-0.536246	-5.880138	0.97455
第 151 节龙身	0.968841	-6.957479	0.973608
第 201 节龙身	-7.893161	-1.230764	0.973096
龙尾（后）	0.956216	8.322736	0.972938

### 5.3 问题三：最小螺距优化模型

#### 5.3.1 最小螺距优化模型的建立

##### (1) 顶点重新表示

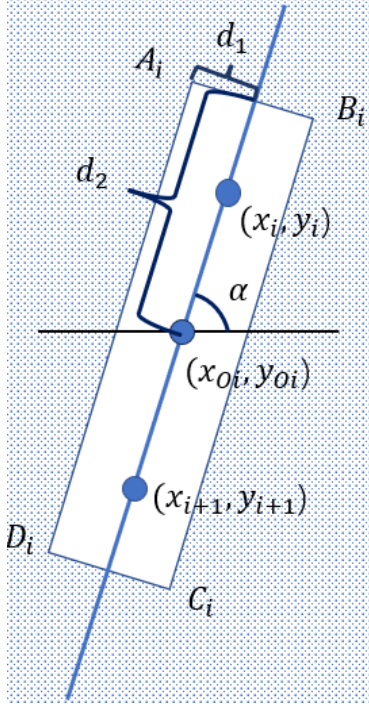


图9 四顶点示意图

为了方便计算顶点坐标，以板凳几何中心  $(x_{O_i}, y_{O_i})$  为原点，重新表达顶点坐标。 $\alpha$  为板凳的宽对应的中轴线的旋转角， $d_1$  为板凳的宽度的一半， $d_2$  为板凳长度的一半，如??所示。现在四顶点的坐标转换公式为：

令第  $i$  节板凳中点  $O_i(\frac{x_i+x_{i+1}}{2}, \frac{y_i+y_{i+1}}{2})$

•  $A_i$  :

$$\begin{pmatrix} x_{A_i} - x_{O_i} \\ y_{A_i} - y_{O_i} \end{pmatrix} = \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} d_2 \\ d_1 \end{pmatrix}$$

•  $B_i$  :

$$\begin{pmatrix} x_{B_i} - x_{O_i} \\ y_{B_i} - y_{O_i} \end{pmatrix} = \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} d_2 \\ -d_1 \end{pmatrix}$$

•  $C_i$  :

$$\begin{pmatrix} x_{C_i} - x_{O_i} \\ y_{C_i} - y_{O_i} \end{pmatrix} = \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} -d_2 \\ -d_1 \end{pmatrix}$$

•  $D_i$  :

$$\begin{pmatrix} x_{D_i} - x_{O_i} \\ y_{D_i} - y_{O_i} \end{pmatrix} = \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} -d_2 \\ d_1 \end{pmatrix}$$

化简后四顶点计算式为：

$$\begin{cases} x_{A_i} = x_{O_i} + d_2 \cos \alpha + d_1 \sin \alpha \\ y_{A_i} = y_{O_i} - d_2 \sin \alpha + d_1 \cos \alpha \end{cases} \quad \begin{cases} x_{B_i} = x_{O_i} + d_2 \cos \alpha - d_1 \sin \alpha \\ y_{B_i} = y_{O_i} - d_2 \sin \alpha - d_1 \cos \alpha \end{cases}$$

$$\begin{cases} x_{C_i} = x_{O_i} - d_2 \cos \alpha - d_1 \sin \alpha \\ y_{C_i} = y_{O_i} + d_2 \sin \alpha - d_1 \cos \alpha \end{cases} \quad \begin{cases} x_{D_i} = x_{O_i} - d_2 \cos \alpha + d_1 \sin \alpha \\ y_{D_i} = y_{O_i} + d_2 \sin \alpha + d_1 \cos \alpha \end{cases}$$

$$\text{其中} \begin{cases} i = 1, 2, \dots, 223 \\ d_1 = 0.15m \\ d_2 = \begin{cases} 1.705m & i = 1 \\ 1.1m & i = 2, 3, \dots, 223 \end{cases} \\ \alpha = \arctan \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = \arctan \frac{v_{i+1}(t) - v_i(t)}{u_{i+1}(t) - u_i(t)} \end{cases}$$

## (2) 基于分离轴定理的碰撞检测

### 分离轴单位向量

以顶点  $A_1$  和  $D_1$  为例

$$\vec{e}_{A_1 D_1} = \frac{(x_{D_1}(t) - x_{A_1}(t), y_{D_1}(t) - y_{A_1}(t))}{|l_{A_1 D_1}|}$$

### 投影点

以  $\vec{OA_1}$  在  $\vec{e}_{A_1 D_1}$  的投影为例

$$\vec{OA_1} \cdot \vec{e}_{A_1 D_1} = \frac{x_{A_1}(x_{D_1}(t) - x_{A_1}(t)) + y_{A_1}(y_{D_1}(t) - y_{A_1}(t))}{|l_{A_1 D_1}|}$$

### 碰撞检测步骤

1. 表示投影点：对于龙头和某个可能发生碰撞的第  $i$  节龙身板凳，求得两组投影轴单位向量  $\vec{e}_{A_1 D_1}, \vec{e}_{A_1 B_1}, \vec{e}_{A_i D_i}, \vec{e}_{A_i B_i}$ ，再用向量积的算法求得两组顶点在投影轴向量的投影。
2. 比较投影区间：比较龙头和第  $i$  节龙身板凳在投影轴上的投影区间  $R$  和  $Q$ ，若有交点，则更换投影轴比较，直至遍历四条投影轴。
3. 碰撞检测：经过四次比较，若每个投影轴上的两个投影区间均有交点，则发生碰撞；反之，则未发生碰撞。

### 碰撞检测模型

$$P = \begin{cases} 1 & \text{任一投影轴上的两个投影向量满足 } Q_{max} > R_{min} \text{ 且 } R_{max} > Q_{min} \\ 0 & \text{其他} \end{cases}$$

当龙头发生碰撞时， $P=1$ ；反之， $P=0$ 。

### (3) 几何模型

对式 (1) 进行简化得到 等距螺旋线简化方程：

$$\rho = b\theta \quad (b = \frac{d}{2\pi}, d \text{ 为螺距}) \quad (28)$$

将式 (5) 对  $t$  求导后代入上式 (28) 得 龙头运动方程：

$$\begin{cases} V_1 = 1m/s \\ \frac{d\theta_1}{dt} = -\frac{V_1}{b\sqrt{1+\theta_1^2}} = -\frac{1}{b\sqrt{1+\theta_1^2}} \\ \text{初始条件: } \theta_1(0) = 32\pi \end{cases} \quad (29)$$

龙头到达掉头区域边界，即  $\rho = 4.5m$  时

$$\begin{cases} \rho = b\theta_1 = 4.5m \implies \theta_{\text{边界}} = \frac{4.5}{b} \\ T_{\text{reach}} = \int_{32\pi}^{\theta_{\text{边界}}} \frac{b\sqrt{1+\theta^2}}{v_0} d\theta \end{cases} \quad (30)$$

**龙身迭代方程：**

$$b^2(\theta_{i+1}^2(t) + \theta_i^2(t) - 2\theta_{i+1}(t)\theta_i(t)\cos(\theta_{i+1}(t) - \theta_i(t))) = l_i^2$$

$$\text{其中, } l_i = \begin{cases} 2.86 & i = 1 \\ 1.65 & i = 2, 3, \dots, 223 \end{cases}$$

**碰撞约束条件：**

在龙头到达掉头区域之前不发生碰撞，即龙头左前顶点 A 到第 i 节板凳中轴线的距离  $d_i(t)$  大于 0.15m

$$\min_{t \in [0, T_{\text{reach}}]} d_i(t) \geq 0.15m$$

求出板凳龙盘入终止时刻  $t_{\text{碰撞}}$ ：

$$t_{\text{碰撞}} \geq T_{\text{reach}}$$

**螺距定义域：**

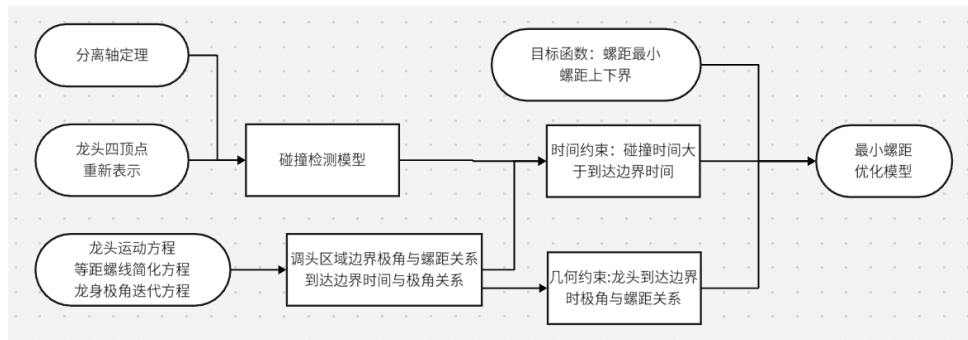
$$d \in [0.3, 0.55]m \quad (31)$$

**综上，螺距 d 的优化模型为：**

以螺距 d 最小为目标函数，以几何约束，时间约束，螺距上下界为约束条件，建立单目标优化模型。

$$\begin{aligned} & \min d \\ & \text{约束条件} \begin{cases} \text{几何约束: } \theta_1(T_{\text{reach}}) = \frac{4.5}{b} \\ \text{时间约束: } t_{\text{碰撞}} \geq T_{\text{reach}} \\ \text{螺距上下界: } 0.3m \leq d \leq 0.55m \end{cases} \end{aligned} \quad (32)$$

其中，掉头区域为  $D: x^2 + y^2 \leq 20.25$



**图 10 问题三建模流程图**

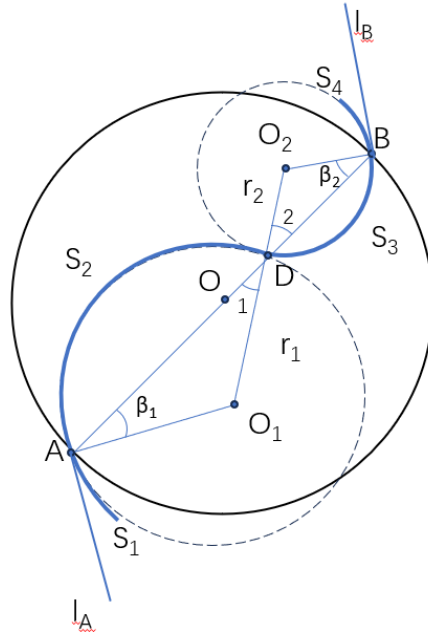


图 11 运动曲线示意图

### 5.3.2 螺距最小优化模型求解

1. 对于每一个  $d_{mid}$  ( $d_{mid} = \frac{d_{high} - d_{low}}{2}$ ), 计算龙头到达边界时间  $T_{reach}$
2. 运用问题二的模型检测在  $[0, T_{reach}]$  内是否发生碰撞。
3. 若发生碰撞, 增大螺距, 即  $d_{low} = d_{mid}$ ;  
若未发生碰撞, 减小螺距, 即  $d_{high} = d_{mid}$
4. 当  $|d_{high} - d_{low}| \leq 10^{-6}$  时, 输出结果  $d_{high}$  我们得到最小螺距为 **0.417137m**

## 5.4 问题四: s 形曲线长度优化模型与含调头的舞龙队运动状态模型

### 5.4.1 s 形曲线长度优化模型的建立

#### (1) 调头区域几何分析

图 11 中舞龙队经过以下四个区域:

1. 盘入螺线  $S_1$ ; 2. 大圆弧  $S_2$ ; 3. 小圆弧  $S_3$ ; 4. 盘出螺线  $S_4$

$O$  为掉头区域的圆心, 半径为 4.5m。  $O_1$  为大圆弧的圆心, 半径为  $r_1$ ,  $AO_1$  与  $AO$  夹角为  $\beta_1$ ;  $O_2$  为小圆弧的圆心, 半径为  $r_2$ ,  $AO_2$  与  $AO$  夹角为  $\beta_2$ 。  $S_1$  与  $S_2$  相切与  $A$  点, 切线为  $l_A$ ;  $S_2$  与  $S_3$  相切与  $D$  点;  $S_3$  与  $S_4$  相切与  $B$  点, 切线为  $l_B$ 。

#### (2) 分析题目条件, 得到以下结论:

1. 由于盘入螺线和盘出螺线分别与大圆弧和小圆弧相切, 因此  $AO_1$  和  $BO_2$  分别垂直于  $l_A$  和  $l_B$ 。
2. 由于盘出螺线与盘入螺线关于螺线中心  $O$  呈中心对称, 因此线段  $AB$  为掉头区

域的直径

3. 由于大圆弧  $S_2$  和小圆弧相切于  $S_3$ ，因此  $O_1$ 、 $D$  和  $O_2$  共线

4. 由以上三点可知： $AO_1 \parallel BO_2$ ，因此  $\beta_1 = \beta_2$

**推论：**由于  $\triangle AO_1D$  与  $\triangle BO_2D$  均为等腰三角形，且  $\beta_1 = \beta_2$ ，因此  $\angle 1 = \angle 2$ ，因此  $A$ 、 $D$  和  $B$  共线

### (3) 曲线长度优化

假设两圆弧的半径比为  $k$ ，即  $r_1 = k r_2$ ，另  $\beta = \beta_1 = \beta_2$ ，延长  $AO_1$  与圆  $O$  交于点  $C$ ，连接  $B$  和  $C$ ， $\triangle ABC$  是直角三角形。过  $O_2$  作  $AC$  的垂线段，交  $AC$  于点  $E$ 。

$S_2$  和  $S_3$  的总弧长为：

$$s_{\text{总}} = (k+1)r_2(\pi - 2\beta) \quad (33)$$

在  $Rt\triangle ABC$  中，

$$\begin{cases} BC = AB \sin \beta \\ AC = AB \cos \beta \end{cases} \quad (34)$$

因此，在  $Rt\triangle O_1EO_2$  中，

$$\begin{cases} O_1E = AC - (r_1 + r_2) = AB \cos \beta - (k+1)r_2 \\ O_2E = BC = AB \sin \beta \end{cases} \quad (35)$$

由勾股定理得， $O_1O_2^2 = O_1E^2 + O_2E^2$ ，即：

$$(k+1)^2 r_2^2 = (AB \cos \beta - (k+1)r_2)^2 + (AB \sin \beta)^2 \quad (36)$$

联立式 (33) 和式 (36) 得：

$$s_{\text{总}} = \frac{AB r_2 (\pi - 2\beta)}{2 \cos \beta} \quad (37)$$

由此发现，两圆弧总长与半径比  $k$  无关，因此不可以通过调整圆弧，使得调头曲线变短。

本文在后续的求解过程中设定两圆弧的半径比为 2: 1

## 5.4.2 含调头的舞龙队运动状态模型的建立

### (1) 确定关键位置

已知盘入螺线螺距为 1.7m，掉头区域半径为 4.5m，两圆弧半径比为 2: 1，通过计算得到：

$$r_1 = 3.005418 \text{ m}$$

$$r_2 = 1.502709 \text{ m}$$

切入点  $A$  的坐标为：(-2.711856, -3.591078)

大圆弧圆心  $O_1$  的坐标为：(-0.760009, -1.305726)

小圆弧圆心  $O_2$  的坐标为: (1.735932,2.448402)

两圆弧切点  $D$  的坐标为: (0.903952,1.197026)

切出点  $B$  的坐标为: (2.711856,3.591078)

## (2) 分类讨论

- 1) 全部位于  $S_1$ ;
- 2) 龙头位于  $S_2$ , 其余位于  $S_1$ ;
- 3) 龙头位于  $S_2$ , 部分龙身位于  $S_2$ , 部分位于  $S_1$ ;
- 4) 龙头位于  $S_3$ , 部分龙身位于  $S_2$ , 部分位于  $S_1$ ;
- 5) 龙头位于  $S_3$ , 部分龙身位于  $S_3$ , 部分龙身位于  $S_2$ , 部分位于  $S_1$ ;
- 6) 龙头位于  $S_4$ , 部分龙身位于  $S_3$ , 部分龙身位于  $S_2$ , 部分位于  $S_1$ ;
- 7) 龙头位于  $S_4$ , 部分龙身位于  $S_4$ , 部分龙身位于  $S_3$ , 部分龙身位于  $S_2$ , 部分位于  $S_1$ ;
- 8) 龙头位于  $S_4$ , 部分龙身位于  $S_4$ , 部分龙身位于  $S_3$ , 部分龙身位于  $S_2$ ;
- 9) 龙头位于  $S_4$ , 部分龙身位于  $S_4$ , 部分龙身位于  $S_3$ ;
- 10) 全部位于  $S_4$ 。

## (3) 龙头前把手中心位置的坐标

根据四段弧线的方程求解龙头前把手中心位置的坐标  $(x, y)$

$$\begin{cases} (\frac{17}{20\pi}\theta(t) \cos \theta(t), \frac{17}{20\pi}\theta(t) \sin \theta(t)) & -100 \leq t \leq 0 \\ (x_{O_1} + r_1 \cos \theta(t), y_{O_1} + r_1 \sin \theta(t)) & 0 \leq t \leq \frac{\alpha r_1}{v_0} \\ (x_{O_2} - r_2 \cos \theta(t), y_{O_2} - r_2 \sin \theta(t)) & \frac{\alpha r_1}{v_0} \leq t \leq (\frac{\alpha r_2}{v_0} + \frac{\alpha r_1}{v_0}) \\ (\frac{17}{20\pi}\theta(t) \cos(\theta(t) + \pi), \frac{17}{20\pi}\theta(t) \sin(\theta(t) + \pi)) & (\frac{\alpha r_2}{v_0} + \frac{\alpha r_1}{v_0}) \leq t \leq 100 \end{cases}$$

其中,  $\alpha$  为大圆弧和小圆弧的圆心角

## (4) 各把手所在区域判断

根据前把手位置中心判断后把手所处区域, 假设前把手位置为  $(x_i, y_i)$ , 前后把手距离为  $l_i$

(1) 前把手位于  $S_2$ , 判断前把手中心与切入点  $A$  的距离  $d_A$  与  $l_i$  的关系。若  $d_A > l_i$ , 则后把手在  $S_2$ ; 反之, 则在  $S_1$

(2) 前把手位于  $S_3$ , 判断前把手中心与切点  $D$  的距离  $d_D$  与  $l_i$  的关系。若  $d_D > l_i$ , 则后把手在  $S_3$ ; 反之, 则在  $S_2$

(3) 前把手位于  $S_4$ , 判断前把手中心与切出点  $B$  的距离  $d_B$  与  $l_i$  的关系。若  $d_B > l_i$ , 则后把手在  $S_4$ ; 反之, 则在  $S_3$

## (5) 不同运动区域的把手运动状态模型

### 模型 1

$S_1$  上两点位置与速度迭代模型

$$\begin{cases} \frac{d\theta}{dt} = \left(-\frac{2\pi}{1.7}\right) \frac{1}{\sqrt{1+\theta^2}} \\ \theta_{i+1}^2 + \theta_i^2 - 2\theta_i\theta_{i+1} \cos(\theta_{i+1} - \theta_i) = \frac{4\pi^2 l_i^2}{17^2} \\ V_{i+1}(t) = \sqrt{\frac{1+\theta_{i+1}^2}{1+\theta_i^2}} |g_{i+1}| V_i(t) \end{cases} \quad (38)$$

其中,

$$g_{i+1} = \frac{-\theta_i(t) + \theta_{i+1}(t) \cos(\theta_{i+1}(t) - \theta_i(t)) + \theta_{i+1}(t)\theta_i(t) \sin(\theta_{i+1}(t) - \theta_i(t))}{\theta_{i+1}(t) - \theta_i(t) \cos(\theta_{i+1}(t) - \theta_i(t)) + \theta_{i+1}(t)\theta_i(t) \sin(\theta_{i+1}(t) - \theta_i(t))} \quad (39)$$

#### 5.4.3 含调头的舞龙队运动状态模型的求解

所有把手都在  $S_1$  上的把手位置求解

##### (1) 龙头位置求解

利用式 (38) 中的龙头极角常微分方程, 用 ODE45 算法 ( $h=1s$ ) 求 -100~0 秒, 即第 1230.0425491~1330.0425491 秒每秒极角大小。然后通过式 (40) 转化为二维直角坐标系下的位置坐标。

$$\begin{cases} x_1 = \frac{17}{20\pi} \theta(t) \cos \theta(t) \\ y_1 = \frac{17}{20\pi} \theta(t) \sin \theta(t) \end{cases} \quad (40)$$

##### (2) 龙身位置求解

利用式 (38) 中的相邻两把手极角关系式, 将其转化为:

$$\begin{cases} \theta_{i+1}^2 + \theta_i^2 - 2\theta_i\theta_{i+1} \cos(\theta_{i+1} - \theta_i) - \frac{4\pi^2 l_i^2}{17^2} = 0 \\ l_i = 2.86, (i = 2) \\ l_i = 1.65, (i = 2, 3, \dots, 223) \end{cases} \quad (41)$$

运用二分法 (精度为  $10^{-6}$ ) 求出 -100~0s 每秒各把手极角。然后通过使 (40) 转化为二维直角坐标系下的位置坐标。

##### (3) 各把手速度求解

利用式 (38) 中的速度迭代公式, 已知  $V_1 = 1m/s$ , 结合前面求得的各把手极角, 计算各把手速度。

**结果展示**



表 5 舞龙各节段坐标 (单位: m)

节段	-100 s		-50 s		0 s	
	$x$	$y$	$x$	$y$	$x$	$y$
龙头	7.778034	3.717163	6.608301	1.898863	-2.711856	-3.591077
第 1 节龙身	6.209273	6.108521	5.366912	4.475402	-0.063535	-4.670888
第 51 节龙身	-10.608038	2.831492	-3.629947	-8.963799	2.459961	-7.778145
第 101 节龙身	-11.922761	-4.802377	10.125786	-5.972248	3.008494	10.108539
第 151 节龙身	-14.351032	-1.980992	12.974783	-3.810359	-7.002788	10.337483
第 201 节龙身	-11.952942	10.566998	10.522507	-10.807426	-6.872841	12.382609
龙尾 (后)	-1.01106	-16.527573	0.189811	15.720588	-1.933628	-14.713128

表 6 舞龙各把手速度 (单位: m/s)

节段	-100 s	-50 s	0 s
龙头	1.000000	1.000000	1.000000
第 1 节龙身	0.999904	0.999762	0.998687
第 51 节龙身	0.999346	0.998642	0.995134
第 101 节龙身	0.999091	0.998248	0.994448
第 151 节龙身	0.998944	0.998047	0.994156
第 201 节龙身	0.998849	0.997925	0.993994
龙尾 (后)	0.998817	0.997885	0.993944

## 六、模型评价

### 6.1 模型的优点

1. 本文采用阿基米德螺线精确描述盘入/盘出螺线, 较准确地计算出板凳龙各个时刻的位置和速度

2. 利用分离轴定理进行矩形碰撞检测，准确判断非连续节间的碰撞
3. 对不同的优化模型，选择适配的算法，如二分法、自适应步长 ODE 求解器等，大大提高了准确度和精度

## 6.2 模型的不足

1. 模型假设是在理想条件下的，忽略了板凳倾斜、板凳变形、摩擦、舞者操作等因素带来的影响

## 参考文献

- [1] 刘娜, 毛晓菊. 基于分离轴定理的碰撞检测算法 [J]. 数字技术与应用, 2012, (08): 102. DOI: 10.19695/j.cnki.cn12-1369.2012.08.070.

## 附录 A 模板所用的宏包

表 7 宏包罗列

模板中已经加载的宏包				
amsbsy	amsfonts	amsgen	amsmath	amsopn
amssymb	amstext	appendix	array	atbegshi
atveryend	auxhook	bigdelim	bigintcalc	bigstrut
bitset	bm	booktabs	calc	caption
caption3	CJKfntef	cprotect	ctex	ctexhook
ctexpatch	enumitem	etexcmds	etoolbox	everysel
expl3	fix-cm	fontenc	fontspec	fontspec-xetex
geometry	getttitlestring	graphics	graphicx	hobsub
hobsub-generic	hobsub-hyperref	hopatch	hxdetex	hycolor
hyperref	ifluatex	ifpdf	ifthen	ifvtex
ifxetex	indentfirst	inwarerr	intcalc	keyval
kvdefinekeys	kvoptions	kvsetkeys	l3keys2e	letltxmacro
listings	longtable	lstmisc	ltxcaption	ltxcmds
multirow	nameref	pdfescape	pdfetexcmds	refcount
rerunfilecheck	stringenc	suffix	titletoc	tocloft
trig	ulem	uniquecounter	url	xcolor
xcolor-patch	xeCJK	xeCJKfntef	xeCJK-listings	xparse
xtemplate	zhnumber			

以上宏包都已经加载过了，不要重复加载它们。

## 附录 B 支撑材料文件列表

文件名	文件内容
problem_1.m	计算 0-300 秒（秒为单位）的精确计算位置和速度，简易动画
problem_1_t_theta.m	精确求 $\theta(t)$ 的数值解（没有用到）
problem_2.m	精确计算相撞时间及当时精确计算位置和速度
problem_2_huatu.m	表示龙头左前顶点到直线距离的示意图
problem_3.m	计算最小螺距
problem_4_1.m	计算关键点坐标
problem_4_2.m	精确计算-100 到 0 秒位置和速度
erfenfa_12.m	对龙头与第一个龙身使用二分法
erfenfa_3.m	对第三个龙身及后面的龙身使用二分法
solve_d.m	求解龙首顶点与龙身的距离
sat_rect_overlap.m	判断两个矩形是否相交
fun3.m	给定螺距判断是否相撞
result1.xlsx	结果表一
result2.xlsx	结果表二
result4.xlsx	结果表四

## 附录 C 问题一代码

```

clc,clear,close all

format long g

dtheta_dt = @(t, theta) -40*pi./(11*sqrt(1 + theta.^2));
theta0 = 32*pi;
tspan = linspace(0, 300, 301);
options = odeset('RelTol', 1e-9, 'AbsTol', 1e-9, 'Refine', 1); % 精度控制
[t, theta] = ode45(dtheta_dt, tspan, theta0, options);

l = [2.86, 1.65];

```

```

theta_matrix = zeros(224,301);

for i = 1:length(theta)
    theta_matrix(1,i) = theta(i);
end

n = ceil(6*log2(10) - log2(180-0));

for j = 1:length(tspan)
    for i = 2
        youbian=@(x)((40*pi/11)^2)*l(1)^2;
        zuobian=@(x)theta_matrix(i-1,j)^2+x^2-2*theta_matrix(i-1,j)*x*cos(-theta_matrix(i-1,j)+x);
        a = theta_matrix(i-1,j);
        b = theta_matrix(i-1,j) + pi;
        for ii = 1:22 % 22
            c = (a+b)/2;
            aa = youbian(a) - zuobian(a);
            cc = youbian(c) - zuobian(c);
            if aa*cc<0
                a = a;
                b = c;
            else
                a = c;
                b = b;
            end
        end
        theta_matrix(i,j) = c;
    end
end

for j = 1:length(tspan)
    for i = 3:224
        youbian=@(x)((40*pi/11)^2)*l(2)^2;
        zuobian=@(x)theta_matrix(i-1,j)^2+x^2-2*theta_matrix(i-1,j)*x*cos(-theta_matrix(i-1,j)+x);
        a = theta_matrix(i-1,j);
        b = theta_matrix(i-1,j) + pi;
        for ii = 1:22
            c = (a+b)/2;
            aa = youbian(a) - zuobian(a);
            cc = youbian(c) - zuobian(c);
            if aa*cc<0
                a = a;
                b = c;
            else
                a = c;
                b = b;
            end
        end
    end
end

```

```

        end
        theta_matrix(i,j) = c;
    end
end

rho_matrix = (11/(40*pi)) .* theta_matrix;

result_matrix = zeros(448,length(tspan));
for i = 1:size(rho_matrix,1)
    x = rho_matrix(i,:).*cos(theta_matrix(i,:));
    y = rho_matrix(i,:).*sin(theta_matrix(i,:));
    result_matrix(2*i-1,:) = x;
    result_matrix(2*i,:) = y;
end

result_matrix = round(result_matrix, 6);
writematrix(result_matrix,'result1.xlsx','Sheet','位置','Range','B2:KP449')

V=ones (224,length(tspan));

for j = 1:length(tspan)
    for i=1:223
        u=theta_matrix(i,j);
        v=theta_matrix(i+1,j);
        G = (u*v*sin(v-u)+v*cos(v-u)-u)/(u*v*sin(v-u)-u*cos(v-u)+v);
        V(i+1,j) = V(i,j)*abs(G)*sqrt((1+v^2)/(1+u^2));
    end
end

V = round(V, 6);
writematrix(V,'result1.xlsx','Sheet','速度','Range','B2:KP225')

% for i = 1:length(tspan)
%     x = rho_matrix(:,i).*cos(theta_matrix(:,i));
%     y = rho_matrix(:,i).*sin(theta_matrix(:,i));
%     plot(x,y,"b-")
%     grid on
%     pause(1)
%     axis equal
% end
figure; hold on; grid on; axis equal;
title('等距螺线上的“板凳龙”运动模拟','FontSize',14,'FontWeight','bold');

% 初始化图形对象（只创建一次）
h_plot = plot(NaN, NaN, 'b-', 'LineWidth', 1.5);
h_time = text(0.05, 0.95, '', 'Units', 'normalized', ...
    'FontSize', 12, 'FontWeight', 'bold', ...

```

```

        'VerticalAlignment', 'top');

% 预计算坐标轴范围 (避免闪烁)
all_x = rho_matrix .* cos(theta_matrix);
all_y = rho_matrix .* sin(theta_matrix);
xlim([min(all_x(:))-5, max(all_x(:))+5]);
ylim([min(all_y(:))-5, max(all_y(:))+5]);

for i = 1:length(tspan)
    x = rho_matrix(:,i) .* cos(theta_matrix(:,i));
    y = rho_matrix(:,i) .* sin(theta_matrix(:,i));

    % 更新图形数据 (不重新绘制)
    set(h_plot, 'XData', x, 'YData', y);

    % 更新时间标签
    set(h_time, 'String', sprintf('时间: %.1f s', tspan(i)));

    % 刷新图形
    drawnow;
    pause(0.5); % 控制动画速度
end

```

## 附录 D 问题二代码

```

clc,clear,close all

format long g % 26.352603633422080 25.915795186647780 0-420
              % 26.352604039865460 25.915795468562592 0-412

a = 26; % 26.3-26.8
b = 40;
for ii = 1:20000

    c = (a+b)/2;
    aa = slope_d(a,0.55) - 0.15;
    cc = slope_d(c,0.55) - 0.15;
    if aa*cc<0
        a = a;
        b = c;
    else
        a = c;
        b = b;
    end
end
end

```

```

res = c;

dtheta_dt = @(t, theta) -40*pi./(11*sqrt(1 + theta.^2));
theta0 = 32*pi;
tspan = linspace(0, 420, 421); % 0-420s
options = odeset('RelTol', 1e-9, 'AbsTol', 1e-9, 'Refine', 1); % 精度控制
[t,theta] = ode45(dtheta_dt, tspan, theta0, options);

theta0 = theta(413) % 第412秒
tspan = linspace(0, 1, 100001);
options = odeset('RelTol', 1e-9, 'AbsTol', 1e-9);
[t, theta] = ode45(dtheta_dt, tspan, theta0, options); % 26.146526312045
[i,j] = min(abs(theta - res));
tot_time = t(j) + 412 % 412.4738598
res_theta = theta(j)

theta_tot = ones(224,1);
theta_tot(1) = theta(j);
theta_tot(2) = erfenfa_12(theta_tot(1),0.55);
for i = 3:224
    theta_tot(i) = erfenfa_3(theta_tot(i-1),0.55);
end

V = ones(224,1);
for i=1:223
    u=theta_tot(i);
    v=theta_tot(i+1);
    G = (u*v*sin(v-u)+v*cos(v-u)-u)/(u*v*sin(v-u)-u*cos(v-u)+v);
    V(i+1) = V(i)*abs(G)*sqrt((1+v^2)/(1+u^2));
end

x = (11/(40*pi)).*theta_tot.*cos(theta_tot);
y = (11/(40*pi)).*theta_tot.*sin(theta_tot);
plot(x,y)
hold on
xx = [(11/(40*pi))*26.1465965779221*cos(26.14659657792212),
      (11/(40*pi))*26.1465965779221*sin(26.1465965779221)];
plot(xx(1),xx(2),'r*')
result = [x,y,V];
result = round(result, 6);
writematrix(result,'result2.xlsx','Range','B2:D225')

```

## 附录 E 问题三代码

```

clc,clear,close all

```



```

format long g

for d = 0.4171375975 % 0.430337 % 0.4171375975
    res = fun3(d)
end

```

## 附录 F 问题三 fun3 函数代码

```

function [res] = fun3(d)
    res = 0;
    theta_tot = ones(224,1);
    for theta1 = 9*pi/d %:100
        rho = (d/(2*pi))*theta1;
        theta_tot(1) = theta1;
        theta_tot(2) = erfenfa_12(theta_tot(1),d);
        for i = 3:224
            theta_tot(i) = erfenfa_3(theta_tot(i-1),d);
        end
        p = (d/(2*pi));
        x1 = p*theta_tot(1)*cos(theta_tot(1));
        y1 = p*theta_tot(1)*sin(theta_tot(1));
        x2 = p*theta_tot(2)*cos(theta_tot(2));
        y2 = p*theta_tot(2)*sin(theta_tot(2));
        rect1_center = [(x1+x2)/2; (y1+y2)/2];
        k1 = (y2-y1)/(x2-x1);
        alpha1 = atan(k1);
        M1 = [cos(alpha1), -sin(alpha1); sin(alpha1), cos(alpha1)];
        rect1 = zeros(2,4);
        rect1(:,1) = rect1_center + M1*[3.41/2; 0.15];
        rect1(:,2) = rect1_center + M1*[3.41/2; -0.15];
        rect1(:,3) = rect1_center + M1*[-3.41/2; -0.15];
        rect1(:,4) = rect1_center + M1*[-3.41/2; 0.15];
        ((rect1(1,:)-x1).^2 + (rect1(2,:)-y1).^2).^0.5;
        (0.275^2 + 0.15^2)^0.5;
        ((2.86+0.275)^2 + 0.15^2)^0.5;
        for i = 2:223 % 排除第二个
            x11 = p*theta_tot(i)*cos(theta_tot(i));
            y11 = p*theta_tot(i)*sin(theta_tot(i));
            x22 = p*theta_tot(i+1)*cos(theta_tot(i+1));
            y22 = p*theta_tot(i+1)*sin(theta_tot(i+1));
            k = (y22-y11)/(x22-x11);
            alpha = atan(k);
            M = [cos(alpha), -sin(alpha); sin(alpha), cos(alpha)];
            rect2_center = [(x11+x22)/2; (y11+y22)/2];

```

```

rect2 = zeros(2,4);
rect2(:,1) = rect2_center + M*[1.1;0.15];
rect2(:,2) = rect2_center + M*[1.1;-0.15];
rect2(:,3) = rect2_center + M*[-1.1;-0.15];
rect2(:,4) = rect2_center + M*[-1.1;0.15];
result = sat_rect_overlap(rect1',rect2');

plot([rect1(1,:),rect1(1,1)],[rect1(2,:),rect1(2,1)],"b")
hold on
plot(x1,y1,"b*")
plot(x2,y2,"b*")
plot([rect2(1,:),rect2(1,1)],[rect2(2,:),rect2(2,1)],"b")
% plot(rect2(1,:),rect2(2,:),"r*")
plot(x11,y11,"b*")
plot(x22,y22,"b*")
axis equal

if result == true
    if i==2
        continue
    end
function [res] = fun3(d)
res = 0;
theta_tot = ones(224,1);
for theta1 = 9*pi/d %:100
    rho = (d/(2*pi))*theta1;
    theta_tot(1) = theta1;
    theta_tot(2) = erfenfa_12(theta_tot(1),d);
    for i = 3:224
        theta_tot(i) = erfenfa_3(theta_tot(i-1),d);
    end
    p = (d/(2*pi));
    x1 = p*theta_tot(1)*cos(theta_tot(1));
    y1 = p*theta_tot(1)*sin(theta_tot(1));
    x2 = p*theta_tot(2)*cos(theta_tot(2));
    y2 = p*theta_tot(2)*sin(theta_tot(2));
    rect1_center = [(x1+x2)/2;(y1+y2)/2];
    k1 = (y2-y1)/(x2-x1);
    alpha1 = atan(k1);
    M1 = [cos(alpha1),-sin(alpha1);sin(alpha1),cos(alpha1)];
    rect1 = zeros(2,4);
    rect1(:,1) = rect1_center + M1*[3.41/2;0.15];
    rect1(:,2) = rect1_center + M1*[3.41/2;-0.15];
    rect1(:,3) = rect1_center + M1*[-3.41/2;-0.15];
    rect1(:,4) = rect1_center + M1*[-3.41/2;0.15];
    ((rect1(1,:)-x1).^2 + (rect1(2,:)-y1).^2).^(0.5);
    (0.275^2 + 0.15^2)^0.5;

```

```

((2.86+0.275)^2 + 0.15^2)^0.5;
for i = 2:223 % 排除第二个
    x11 = p*theta_tot(i)*cos(theta_tot(i));
    y11 = p*theta_tot(i)*sin(theta_tot(i));
    x22 = p*theta_tot(i+1)*cos(theta_tot(i+1));
    y22 = p*theta_tot(i+1)*sin(theta_tot(i+1));
    k = (y22-y11)./(x22-x11);
    alpha = atan(k);
    M = [cos(alpha),-sin(alpha);sin(alpha),cos(alpha)];
    rect2_center = [(x11+x22)/2;(y11+y22)/2];
    rect2 = zeros(2,4);
    rect2(:,1) = rect2_center + M*[1.1;0.15];
    rect2(:,2) = rect2_center + M*[1.1;-0.15];
    rect2(:,3) = rect2_center + M*[-1.1;-0.15];
    rect2(:,4) = rect2_center + M*[-1.1;0.15];
    result = sat_rect_overlap(rect1',rect2');

    plot([rect1(1,:),rect1(1,1)],[rect1(2,:),rect1(2,1)],"b")
    hold on
    plot(x1,y1,"b*")
    plot(x2,y2,"b*")
    plot([rect2(1,:),rect2(1,1)],[rect2(2,:),rect2(2,1)],"b")
    % plot(rect2(1,:),rect2(2,:),"r*")
    plot(x11,y11,"b*")
    plot(x22,y22,"b*")
    axis equal

    if result == true
        if i==2
            continue
        end
        res = 1;
        fprintf("当d=%.12f,在theta=%.6f时,龙首刚好与第%d个龙身相撞\n",d,theta1,i)
        % break
    end
end
end
end
res = 1;
fprintf("当d=%.12f,在theta=%.6f时,龙首刚好与第%d个龙身相撞\n",d,theta1,i)
% break
end
end
end
end
end
end
end

```

## 附录 G 问题四关键位置求解代码

```
clc,clear,close all

hold on
d = 1.7;
theta1 = 9*pi/d;
rho = (d/(2*pi))*theta1;
xa = rho*cos(theta1) ; % -2.71185586370666
ya = rho*sin(theta1) ; % -3.59107752276107
alpha = atan( -( d/(2*pi))*cos(theta1) - rho*sin(theta1) )/( (d/(2*pi))*sin(theta1) +
    rho*cos(theta1) ) )
k = tan(alpha);
xb = -rho*cos(theta1) ; % 2.71185586370666
yb = -rho*sin(theta1) ; % 3.59107752276107
% dd = ((xb-xa)^2 + (yb - ya)^2)^0.5
xq = xa + (2/3)*(xb - xa); % 0.903951954568886
yq = ya + (2/3)*(yb - ya) ; % 1.19702584092036
alpha0 = atan((yb-ya)./(xb-xa))
k0 = tan(alpha0);
alpha1 = alpha0 - alpha ; %弧度制，角度制为3.44077805009858
r1 = 3/cos(alpha1) ; % 3.00541766778904
r2 = r1/2 ; % 1.50270883389452
x_circle1 = xa + r1*cos(alpha) % -0.760009116655529
y_circle1 = ya + r1*sin(alpha) % -1.30572642634626
x_circle2 = xb - r2*cos(alpha) % 1.73593249018109
y_circle2 = yb - r2*sin(alpha) % 2.44840197455367

s = r1*(pi - 2*alpha1) + r2*(pi - 2*alpha1); % 13.6212449068211
x = linspace(-3,3,100);

x_circle = 4.5*cos(linspace(0,2*pi,100));
y_circle = 4.5*sin(linspace(0,2*pi,100));
plot(x_circle, y_circle, 'b');
plot([xa,xa+4*cos(alpha)], [ya,ya+4*sin(alpha)], "b-")
plot([xb,xb-2*cos(alpha)], [yb,yb-2*sin(alpha)], "b-")
plot([xq+2*cos(alpha0+alpha1),xq-4*cos(alpha0+alpha1)], [yq+2*sin(alpha0+alpha1),yq-4*sin(alpha0+alpha1)], "b")
plot([xa,xb], [ya,yb], "b-")
plot(xa,ya,"b*")
plot(xb,yb,"b*")
plot(xq,yq,"b*")
plot(x_circle1,y_circle1,"r*")
plot(x_circle2,y_circle2,"r*")
axis equal
```

## 附录 H 问题四位置速度求解代码

```
clc,clear,close all

format long g

% theta0 = 32*pi; % 16圈
% dtheta_dt = @(t, theta) (-2*pi/1.7)/sqrt(1 + theta.^2);
% tspan = linspace(0, 2000, 2001);
% options = odeset('RelTol', 1e-9, 'AbsTol', 1e-9);
% [t, theta] = ode45(dtheta_dt, tspan, theta0, options);
% [i,j] = min(abs(theta - 16.6319611072401)) % 1330-1331 秒

% theta01 = theta(j) % - 100);
% dtheta_dt = @(t, theta) (-2*pi/1.7)/sqrt(1 + theta.^2);
% tspan = linspace(0, 1, 10000001);
% options = odeset('RelTol', 1e-9, 'AbsTol', 1e-9);
% [t, theta] = ode45(dtheta_dt, tspan, theta01, options);
%
% [i,j] = min(abs(theta - 16.6319611072401)) % 1330-1331 秒
% t_res = t(j) + 1330 % 1330.0425491

% 从开始计时来看, 在第1330.0425491秒时, 龙头进入圈内
% 所求-100-0秒, 即第1230.0425491-1330.0425491秒
%对应rho是 31.8617426274912-16.6319611072401

dtheta_dt = @(t, theta) (-2*pi/1.7)/sqrt(1 + theta.^2);
theta0 = 31.8617426274912;
tspan = linspace(0, 100, 101);
options = odeset('RelTol', 1e-9, 'AbsTol', 1e-9, 'Refine', 1); % 精度控制
[t, theta] = ode45(dtheta_dt, tspan, theta0, options);

l = [2.86, 1.65];
theta_matrix = zeros(224,length(tspan));

for i = 1:length(theta)
    theta_matrix(1,i) = theta(i);
end

% n = ceil(6*log2(10) - log2(180-0));
```

```

for j = 1:length(tspan)
    theta_matrix(2,j) = erfenfa_12(theta_matrix(1,j),1.7);
end

for j = 1:length(tspan)
    for i = 3:224
        theta_matrix(i,j) = erfenfa_3(theta_matrix(i - 1,j),1.7);
    end
end

rho_matrix = (1.7/(2*pi)) .* theta_matrix;

result_matrix = zeros(448,length(tspan));
for i = 1:size(rho_matrix,1)
    x = rho_matrix(i,:).*cos(theta_matrix(i,:));
    y = rho_matrix(i,:).*sin(theta_matrix(i,:));
    result_matrix(2*i-1,:) = x;
    result_matrix(2*i,:) = y;
end

result_matrix = round(result_matrix, 6);
writematrix(result_matrix,'result4.xlsx','Sheet','位置','Range','B2:CX449')

V=ones (224,length(tspan));

for j = 1:length(tspan)
    for i=1:223
        u=theta_matrix(i,j);
        v=theta_matrix(i+1,j);
        G = (u*v*sin(v-u)+v*cos(v-u)-u)/(u*v*sin(v-u)-u*cos(v-u)+v);
        V(i+1,j) = V(i,j)*abs(G)*sqrt((1+v^2)/(1+u^2));
    end
end

V = round(V, 6);
writematrix(V,'result4.xlsx','Sheet','速度','Range','B2:CX225')

for i = 1:length(tspan)
    x = rho_matrix(:,i).*cos(theta_matrix(:,i));
    y = rho_matrix(:,i).*sin(theta_matrix(:,i));
    plot(x,y,"b")
    grid on
    pause(0.0005)
    axis equal
end
hold on

```

```

plot(rho_matrix(1,length(tspan)).*cos(theta_matrix(1,length(tspan))),rho_matrix(1,length(tspan)).*sin(theta_mat
x_circle = 4.5*cos(linspace(0,2*pi,100));
y_circle = 4.5*sin(linspace(0,2*pi,100));
plot(x_circle, y_circle, 'r');
axis equal

```

## 附录 I 代码 *problem\_1\_t\_theta.m*

```

clc,clear,close all

theta_list = [];
theta0 = 32*pi;
dtheta_dt = @(t, theta) -40*pi./(11*sqrt(1 + theta.^2));
tspan = linspace(0, 420, 421);
options = odeset('RelTol', 1e-9, 'AbsTol', 1e-9);
[t, theta1] = ode45(dtheta_dt, tspan, theta0, options); % 26.1465268090075

for i = 1:420
    theta0 = theta1(i);
    dtheta_dt = @(t, theta) -40*pi./(11*sqrt(1 + theta.^2));
    tspan = linspace(0, 1, 100001);
    options = odeset('RelTol', 1e-9, 'AbsTol', 1e-9);
    [t, theta] = ode45(dtheta_dt, tspan, theta0, options); % 26.146526312045
    theta_list = [theta_list;theta(1:end - 1)];
end

% theta_list = load("theta.mat");
% theta_list = theta_list.theta_list;
% theta_list(41247387)

```

## 附录 J 代码 *problem\_2\_huatu.m*

```

clc,clear,close all

for theta1 = 26.0:0.1:26.2%26.1465965779221
    theta_1_10 = ones(10,1);
    theta_1_10(2) = erfenfa_12(theta1,0.55);
    for i = 3:10

```

```

        theta_1_10(i) = erfenfa_3(theta_1_10(i-1),0.55);
    end
    theta2 = theta_1_10(2);
    theta9 = theta_1_10(9);
    theta10 = theta_1_10(10);
    d2 = 0.275;
    d1 = 0.15;
    u9t = (11/(40*pi)).*theta9*cos(theta9);
    v9t = (11/(40*pi)).*theta9*sin(theta9);
    k = (theta10*sin(theta10)-theta9*sin(theta9)) ./
        (theta10*cos(theta10)-theta9*cos(theta9));
    k1 = ((11/(40*pi))*theta2*sin(theta2) - (11/(40*pi))*theta1*sin(theta1)) ./ (
        (11/(40*pi))*theta2 ...
        *cos(theta2) - (11/(40*pi))*theta1*cos(theta1));
    xa1 = (11/(40*pi))*theta1*cos(theta1) + d2*cos(atan(k1)) - d1*sin(atan(k1));
    ya1 = (11/(40*pi))*theta1*sin(theta1) + d2*sin(atan(k1)) + d1*cos(atan(k1));
    d = abs(k*(xa1-u9t)-ya1+v9t) ./ sqrt(1+k^2) ;
    yy = [(11/(40*pi))*theta10*sin(theta10), (11/(40*pi))*theta9*sin(theta9)];
    xx = [(11/(40*pi))*theta10*cos(theta10), (11/(40*pi))*theta9*cos(theta9)];
    plot(xx,yy)
    hold on
    scatter(xa1,ya1)
    pause(0.2)
end

```

## 附录 K 代码 *erfenfa\_12.m*

```

function [theta9] = erfenfa_12(theta,d)

youbian=@(x)((2*pi)/d)^2*2.86^2; % 2.86 1.65
zuobian=@(x)theta^2+x^2-2*theta*x*cos(-theta+x);
a = theta;
b = theta + pi;
for ii = 1:100
    c = (a+b)/2;
    aa = youbian(a) - zuobian(a);
    cc = youbian(c) - zuobian(c);
    if aa*cc<0
        a = a;
        b = c;
    else
        a = c;
        b = b;
    end
end

```



```

end
theta9 = c;
end

```

## 附录 L 代码 *erfenfa3.m*

```

function [theta9] = erfenfa_3(theta,d)

youbian=@(x)((2*pi)/d)^2*1.65^2; % 2.86 1.65
zuobian=@(x)theta^2+x^2-2*theta*x*cos(-theta+x);
a = theta;
b = theta + pi;
for ii = 1:220
    c = (a+b)/2;
    aa = youbian(a) - zuobian(a);
    cc = youbian(c) - zuobian(c);
    if aa*cc<0
        a = a;
        b = c;
    else
        a = c;
        b = b;
    end
end
theta9 = c;
end

```

## 附录 M 代码 *slove\_d.m*

```

function [d] = slove_d(theta1,d)
theta_1_10 = ones(10,1);
theta_1_10(2) = erfenfa_12(theta1,0.55);
for i = 3:10 % 5-20
    theta_1_10(i) = erfenfa_3(theta_1_10(i-1),0.55);
end
theta2 = theta_1_10(2);
theta9 = theta_1_10(9);
theta10 = theta_1_10(10);
d2 = 0.275;
d1 = 0.15;
u9t = (11/(40*pi)).*theta9*cos(theta9); % 撞9
v9t = (11/(40*pi)).*theta9*sin(theta9);

```

```

k = (theta10*sin(theta10)-theta9*sin(theta9)) ./
    (theta10*cos(theta10)-theta9*cos(theta9));
k1 = ((11/(40*pi))*theta2*sin(theta2) - (11/(40*pi))*theta1*sin(theta1)) ./ (
    (11/(40*pi))*theta2 ...
    *cos(theta2) - (11/(40*pi))*theta1*cos(theta1));
xa1 = (11/(40*pi))*theta1*cos(theta1) + d2*cos(atan(k1)) - d1*sin(atan(k1));
ya1 = (11/(40*pi))*theta1*sin(theta1) + d2*sin(atan(k1)) + d1*cos(atan(k1));

d = abs(k*(xa1-u9t)-ya1+v9t) ./ sqrt(1+k^2) ; % 0.15

end

```

## 附录 N 代码 *sat\_rect\_overlap.m*

```

function isOverlap = sat_rect_overlap(rect1, rect2)

% 输入参数:
% rect1: 4x2矩阵, 表示矩形1的四个顶点坐标 (顺时针或逆时针顺序)
% rect2: 4x2矩阵, 表示矩形2的四个顶点坐标
% drawFlag: 可选参数, true时绘制图形 (默认false)
% 输出:
% isOverlap: 布尔值, true表示重叠

% 获取所有边的法线方向 (分离轴)
axes = get_all_axes(rect1, rect2);

% 检查所有分离轴上的投影是否重叠
isOverlap = true;
for i = 1:size(axes, 1)
    axis = axes(i, :);
    [min1, max1] = project_vertices(rect1, axis);
    [min2, max2] = project_vertices(rect2, axis);

    % 若存在投影不重叠, 则不相交
    if max1 < min2 || max2 < min1
        isOverlap = false;
        break;
    end
end
end

%% 子函数: 获取所有边的法线方向 (分离轴)
function axes = get_all_axes(verts1, verts2)
    axes = [];
    % 处理矩形1的边
    for i = 1:4

```

```

    v1 = verts1(i, :);
    v2 = verts1(mod(i,4)+1, :); % 循环连接最后一个顶点到第一个
    edge = v2 - v1;
    normal = [-edge(2), edge(1)]; % 法向量 (垂直边)
    axes = [axes; normal / norm(normal)]; % 单位化
end
% 处理矩形2的边
for i = 1:4
    v1 = verts2(i, :);
    v2 = verts2(mod(i,4)+1, :);
    edge = v2 - v1;
    normal = [-edge(2), edge(1)];
    axes = [axes; normal / norm(normal)];
end
% 去重 (避免重复轴)
axes = unique(axes, 'rows', 'stable');
end

%% 子函数: 投影顶点到指定轴
function [minVal, maxVal] = project_vertices(verts, axis)
    proj = zeros(size(verts, 1), 1);
    for i = 1:size(verts, 1)
        proj(i) = dot(verts(i, :), axis); % 点积计算投影标量
    end
    minVal = min(proj);
    maxVal = max(proj);
end

```