板凳龙闹元宵空间几何运动学问题建模和求解

摘要

本文在围绕板凳龙的运动学问题，通过对把手位置和板凳形状的动态分析，建立了把手的**螺旋运动模型**，并通过欧拉算法将建立的一阶微分方程离散化处理，直接求得龙头前把手的运动位置方程，并递推求出了任一时间内各把手运动情况。同时对板凳之间的碰撞条件进行几何模型分析，建立**碰撞模型**，通过逐步逼近法缩紧参数，求得碰撞时刻各把手运动情况。并对模型进行了检验和误差分析。

针对问题一：首先，利用等距螺旋线方程$r=a+b\theta$并分解龙头速度为径向速度v_r和切向速度v_θ，建立龙头运动模型。其次，对板凳几何模型分析，通过余弦定理得到两把手的位置关系。最后，利用欧拉方法求解微分方程获得龙头位置并递推得到各节龙身（尾）位置;采用差分法估算不连续位置函数导数求解速度。

针对问题二：在板凳龙的运动模型基础上，确定舞龙队伍不能继续盘旋的临界时间和位置状态。通过分析龙头与龙身的几何模型，分析碰撞点位置和碰撞临界条件。建立碰撞模型：解三角形求出碰撞点的运动轨迹，利用点到直线的距离公式检测是否发生碰撞。确定首个碰撞时刻t_0为$\mathbf{413.2s}$，基于此计算结果，求出此时各把手的位置与速度。

针对问题三：基于板凳龙的运动模型和碰撞模型，将常数的螺距改为变量，推广碰撞模型。其次，观察几何运动模型的特点，得到碰撞点随螺距减小而外移的规律。通过迭代缩小螺距并重新评估碰撞条件，利用逐步逼近法，逐步减小查找范围并判断碰撞点位置，确保龙头能沿螺旋线进入4.5m半径的调头区。最后，根据迭代结果，确定**最小螺距为44.90cm**。

针对问题四：为解决板凳龙调头路径优化问题，建立**S形曲线运动模型**。首先，假设板凳龙在S形曲线上匀速运动，并通过几何分析确定了最短调头路径。将运动过程分为进入S形曲线前、在S形曲线上和离开S形曲线三个阶段。利用螺旋运动模型计算进入S形曲线前各把手的位置和速度；在S形曲线上，基于把手速度和曲线几何特性，通过解三角形确定把手位置；离开S形曲线后，再次应用螺旋模型预测把手位置和速度。通过迭代计算，最终得出各时间点把手的位置和速度信息。

针对问题五：根据S形曲线运动模型，通过迭代方法计算得出，龙头的最大行驶速度。从龙头速度1米/秒开始逐步增加，检查每节龙身把手速度，确保其在时间区间$[-100,100]$内均符合速度限制。最终，当龙头速度增加到**2.007米/秒**时，达到了最大允许速度，此时任一节龙身把手的速度刚好不超过2米/秒。

关键字：等距螺线曲线 螺旋运动模型 碰撞模型 S形曲线运动模型

一、问题重述

"板凳龙"是浙闽一带的传统民俗活动，也叫"盘龙"。村民们用数十至上百条板凳连接成长龙，龙头引导整条长龙盘旋，形成圆圈。舞龙者技艺越高超，表演所占空间越小、速度越快，视觉效果就越佳。

板凳龙几何模型：板凳龙由223节板凳组成，其中第1节是龙头，接下来的221节为龙身，最后一节为龙尾。从空中俯视时，每节板凳可被简化为一个长方形，龙头部分的长方形长度为341cm，而龙身和龙尾的长方形长度均为220cm，所有长方形的宽度统一为30cm。在每个长方形上均有两个直径为5.5cm的圆，圆的中心距离长方形的短边为27.5cm。相邻的两节板凳共用一个圆中的把手来连接。据此建立若干模型并求解如下问题。

问题一：已知舞龙队伍以55cm的等距螺旋线顺时针由A开始，龙头前把手始终以 1m/s 的速度前进。分别建立龙头和龙身（尾）运动模型，计算出 300 秒内每秒各把手的位置和速度。

问题二：在问题一的基础下，以舞龙队伍的各个板凳不发生碰撞为限制条件，建立龙头和龙身发生碰撞临界时的位置坐标模型，确定舞龙队伍不能继续盘入的终止时间并求得此时队伍的位置和速度大小。

问题三：在舞龙队伍顺时针盘入后，需要切换方向重新盘出。已知由盘入转向盘出的调头区域为以等距螺旋线中心为圆心、直径为9m的圆形区域。仍以舞龙队伍的各个板凳不发生碰撞为限制条件，依据给定的相关参数，确定最小螺距，使得龙头前把手能够沿着最小螺距的螺线盘入到该圆形调头空间的边界。

问题四：在盘入、盘出螺距为1.7m，螺线中心对称的情况下，考虑具体的调头路径，即由两段分别与盘入、盘出螺旋线相切的S形圆弧构成，前一段圆弧的半径是后一段的2倍。建立 S 形曲线运动模型，并据给定的相关参数，讨论能否在保证各个部分仍然相切的前提下，变化圆弧半径，规划调头路线，使得该S形调头线变短。并求出当龙头前把手前进速度保持1m/s时，以开始调头为时刻0，$[-100,100]$时刻内，每秒下舞龙队伍的位置和速度。

问题五：根据问题四中确定的路径行径，龙头前把手前进速度不变，以舞龙队伍各个把手的前进速度不超过2m/s为限制条件，计算可以使龙头最大行驶的速度。

二、问题分析

2.1 问题一分析：

问题一需要根据等距螺旋线方程对位置和速度进行动态分析，建立龙头和龙身（尾）各把手的**螺旋运动模型**并求解运动速度的微分方程。本题难点在于准确反映某时间下，各个板凳把手的运动位置。对此，选取等距螺旋线的中心为原点建立极坐标系，利用等距螺线方程对龙头前把手运动的径向和切向速度进行分解代换，建立角度对时间的一阶微分方程。继而对微分方程采用的**欧拉算法求解，**可知龙头前把手在t时刻下所处的准确位置。由龙头和各节龙身的几何位置关系，

通过余弦定理解三角形，基于龙头前把手位置递推出其余各节龙身（尾）在的位置。最后通过差分法近似求导解出各把手在t时刻的速度。

2.2 问题二分析：

在问题一的基础上，求出舞龙队伍不能继续盘入的情况下终止盘入的时刻。通过对龙头和龙身几何模型的分析，推断出龙头上的碰撞点的运动轨迹。根据各圈板凳的数量的递减规律，选定被碰撞的龙身的合理节数和碰撞发生的时间范围。建立龙头**碰撞模型**，使用点到直线距离公式，遍历若干节龙身，继而计算出被碰撞的龙身节数以及终止时刻。最后通过螺旋运动模型计算出各把手的位置及速度。

2.3 问题三分析：

本题需进一步考虑螺距对板凳是否发生碰撞的影响。对此基于问题 2 的点到直线距离公式，变更**碰撞模型**的参数，不断缩小螺距，在此过程中重新判断板凳的碰撞条件。通过螺距的不断迭代解得碰撞半径并比较，计算出使龙头沿对应螺距的螺旋线进入半径为4.5m圆形区域而不发生碰撞的最小螺距。

2.4 问题四分析：

本题需要考虑问题 2 中圆心区域内调头路线的具体的情况。需要先确定出满足各个约束条件的最短的 S 形圆弧，并求得其坐标方程。在此范围内，假设其上的点以匀速前进，确认初始速度，建立对应 S 形曲线运动模型。对 t 不断迭代，计算对应范围内各把手的位置与速度信息。S 形曲线之外的把手仍然通过螺旋运动模型求解相关信息。

2.5 问题五分析：

本问基于问题四中确定的板凳龙盘入和盘出路径，仍然利用 S 形曲线运动模型，仅变更龙头速度的参数。不断增大龙头速度，判断此过程中各节龙身把手 t 时刻的速度并比较。迭代出 t 在[-100,100]内能使任一节龙身速度不超过 2 的龙头最大速度。

三、模型假设

1. 扩展题中的终止于A点的螺旋线，即龙身在未经过A点前，也可以在对应时刻扩展的螺旋线上运动。

2. 假设S曲线的两相切圆弧为两个半径不同的半圆，且各把手在S曲线掉头的任意一点的速率恒定

3.忽路纵掘运动对垂荡运动的影响

四、符号说明

符号	意义	单位
x_i	第i个把手的横坐标	m
y_i	第i个把手的纵坐标	m
r_i	第i个把手的半径	m
θ_i	第i个把手的极角	rad
v_i	第i个把手的速度	m/s

五、模型建立与求解

5.1 问题一模型建立与求解

5.1.1 坐标系的转化和建模准备

为降低计算复杂程度，将题目中的直角坐标系转化为极坐标系并求解下述各个问题。这里仍然选用等距螺旋线的中心为原点O，用r表示某点到O的距离，θ表示该点与O的连线和题目中X轴正方向的夹角度数。转化坐标系的过程根据如下方程：

$$\begin{cases} x = r\cos\theta \\ y = r\sin\theta \end{cases} \tag{1}$$

查阅资料可知等距螺旋线是半径r随着角度θ的增加以指数方式增长的一种特殊螺旋线，任何一条从原点出发的射线与螺旋线相交的角度都是固定的。其极坐标方程通常表示为：

$$r = a + b\theta \tag{2}$$

其中，a为螺旋线的起始半径；b是螺旋线的螺距系数。根据题目已知条件，在A点由圈数n和螺径u的积可求解a，由螺距d除以2π可求解b，即：

$$\begin{cases} a = n \cdot u = 8.8 \\ b = \dfrac{d}{2\pi} = \dfrac{0.55}{2\pi} \end{cases} \tag{3}$$

得到r关于θ的线性方程r(θ)。

如图 1，在已知极坐标系上一点A的坐标(r_a,θ_a)和A与另一点B之间的距离时，可以通过余弦定理求出B在极坐标系上的位置(r_b,θ_b)，经过变换后，解得其在直角坐标系上的位置(x_b,y_b)：

$$\begin{cases} r_a = a - b\theta_a \\ r_b = a - b\theta_b \\ \theta a = \alpha + \theta_b \\ L^2 = r_a^2 + r_b^2 - 2r_a r_b \cos\alpha \\ x_b = r_b \cos\theta_b \\ y_b = r_b \sin\theta_b \end{cases} \tag{4}$$

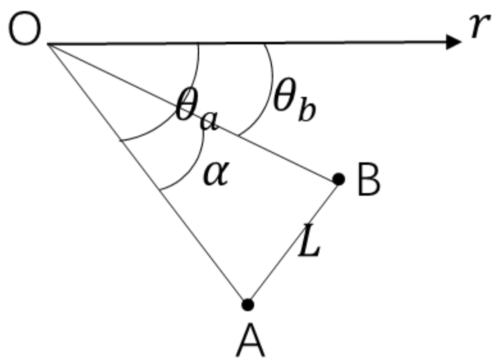


图 1 平面上两点的位置关系图

其中L是A和B之间的距离，α是点A和B与原点连线的夹角。

5.1.2 把手运动轨迹模型建立

1.龙头前把手螺旋运动模型

假设龙头前把手在平面直角坐标系中的位置为(x,y)，由题目已知信息，各把手中心均位于螺线上，满足（2）式，利用（1）式将其转化为极坐标，有：

$$\begin{cases} x(\theta) = r(\theta)\cos\theta \\ y(\theta) = r(\theta)\sin\theta \end{cases} \tag{5}$$

龙头的前进速度$v = 1m/s$，将其分解为沿内径的径向速度v_r和垂直内径的切向速度v_θ。在龙头前把手运动的矢量三角形中满足：

$$v = \sqrt{v_r^2 + v_\theta^2} \tag{6}$$

其中的径向和切向速度联立（2）式解得：

$$v_r = \frac{dr}{dt} = b\frac{d\theta}{dt} \tag{7}$$

$$v_\theta = \omega r = r(\theta)\frac{d\theta}{dt} = (a + b\theta)\frac{d\theta}{dt} \tag{8}$$

其中ω是龙头前把手运动的角速度。

联立（6）、（7）、8），解得：

$$v = \sqrt{b^2 + (a + b\theta)^2}\frac{d\theta}{dt} \tag{9}$$

而龙头前把手的前进速度v是固定不变的1m/ s，得角度关于时间的微分方程：

$$\frac{d\theta}{dt} = \frac{1}{\sqrt{b^2 + (a + b\theta)^2}} \tag{10}$$

将此微分方程求解结果记为θ(t)，联立（5）式，即可得到龙头前把手在t时刻下的位置方程x(t)、y(t) 。

2.龙身各把手运动位置递推模型

根据（4）式，已知某一点的具体位置和它与另一点的距离时，通过解三角形算出另一点的具体位置。在板凳龙模型中，相邻把手间均满足上述图 1 中的A、B点的几何位置关系。当求出龙头前把手t时刻的具体位置(x, y)时，代入（4）式，得到龙头后把手的具体位置。同样，龙头后把手同时也是第一节龙身的前把手，继续代入（4）式，得到其后把手的具体位置。以此类推，建立递推方程，求解t时刻下其后各节龙身（尾）把手的具体位置。

在求得t时刻下，各个把手的具体位置后，将其分别对t求导，即可得到对应时刻下X和Y方向上的分速度v_x、v_y。由于t是离散变量，无法直接求导，为求得x、y对t的导数值，该模型采用差分法近似估算求出。其估算导数值方法的公式为：

$$f'(x_i) = y_i' \approx \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \tag{11}$$

$$f'(x_i) = y_i' \approx \frac{y_{i+1} - y_{i-1}}{2h} \tag{12}$$

$$f'(x_i) \approx \frac{y_i - y_{i-1}}{x_i - x_{i-1}} \tag{13}$$

（11）式、（12）式和（13）式分别对应前向、后向、中心差分法。在计算0s时刻时采用前向差分进行求解；在计算300s时刻时采用后向差分进行求解；其余时刻采用中心差分求解。其中，$h = x_{i+1} - x_i = x_i - x_{i-1}$。

由上述计算结果得X和Y方向上的分速度v_x、v_y后，即可根据矢量三角形合成任一把手在t时刻的和速度v_t：

$$v_t = \sqrt{v_x^2 + v_y^2} \tag{14}$$

5.1.3 把手运动轨迹模型求解

1. 龙头前把手螺旋运动轨迹求解

Step1：初始化设置：由题目信息得，龙头前进速度为恒定的$1m/s$，螺线方程已知。初始时刻龙头位于螺线第16圈A点处，坐标为(8.8,0)。设定初始化条件令龙头前把手初始位移、初始速度与初始角度分别为(8.8,0)、1、0，即：

$$\begin{cases} x_1(0)=8.8 \\ y_1(0)=0 \\ v_1(0)=1 \\ \theta_1(0)=0 \\ r_1(0)=8.8 \end{cases} \tag{15}$$

Step2：微分方程的求解： 由于本题需要求解一元微分方程，故采用欧拉方法求解。欧拉方法的迭代公式为：

$$\theta(t_i+1)=\theta(t_i)+\frac{d\theta}{dt}(t_i,\theta(t_i))\cdot dt$$

Step3：求解龙头位置：设置t步长为0.1s，遍历3000个时刻，均代入$\theta(t)$计算出t在[0,300]中对应的$\theta(t)$，再根据（5）式转化为$x(t)$、$y(t)$，得到t时刻的具体位置。

2. 龙身各把手运动位置递推求解

Step1：初值条件的确定：仍取t步长为0.1s，遍历[0,300]。根据 1 中结果，已知任一t时刻龙头前把手的具体位置。据此结果递推t时刻各把手位置。

Step2：龙身（尾）位置算法的迭代：设置循环次数为板凳龙龙身和龙尾的个数，按式（4）依次递推，分别求解t时刻下各把手的位置(x_i,y_i)。t遍历[0,300]，求出每一t对应的各把手位置。

Step3：龙身（尾）速度求解：解得t时刻下龙头的具体位置信息$x(t)$、$y(t)$后，对时间求导可得t时刻速度大小。同 1 中第四步相同，根据t和$t-0.1$时刻龙尾后把手的位置函数、$t-0.1$和$t+0.1$时刻龙身各把手的位置函数，对不连续的函数$x(t)$、$y(t)$估算近似求解$v(t)$。

5.1.4 求解结果及结果分析

1. 龙头、龙身前把手和龙尾后把手位置求解结果

题目所求第0s、60s、120s、180s、240s、300s龙头、龙身前把手和龙尾后把的位置的如表 1 所示。300s内间隔1s的其他把手位置结果保存在 result1.xlsx 中。

	0s	60s	120s	180s	240s	300s
龙头x(m)	8.800000	5.825096	-4.018872	-3.081031	2.769910	4.286262
龙头y(m)	0.000000	-5.745516	-6.347854	6.038161	-5.271417	2.569170
第 1 节龙身x(m)	8.363824	7.472281	-1.368048	-5.320841	4.936857	2.211444
第 1 节龙身y(m)	2.826544	-3.407487	-7.421507	4.259721	-3.404885	4.537603
第 51 节龙身x(m)	-8.480995	-7.199919	-2.750749	5.503431	3.762251	-5.336066

第 51 节龙身y(m)	4.460853	5.430770	7.954450	5.482991	-5.976022	3.304600
第 101 节龙身x(m)	-8.601104	-6.487240	-6.623542	-8.234574	-5.960333	5.005268
第 101 节龙身y(m)	-5.529184	-7.209293	-6.300545	-2.296497	5.203551	5.204061
第 151 节龙身x(m)	-4.010344	2.461014	5.347018	5.084596	1.670871	-4.980922
第 151 节龙身y(m)	-10.059360	-10.036217	8.228199	-7.743174	-8.517110	-6.329260
第 201 节龙身x(m)	-4.589500	5.126659	9.517247	9.892647	9.347038	8.451712
第 201 节龙身y(m)	-10.409288	-9.625166	-4.226262	-0.303841	0.338637	-2.364091
龙尾（后）x(m)	9.568930	-0.495966	-7.886412	-9.939872	-9.623686	-9.038265
龙尾（后）y(m)	6.561728	11.130353	7.172764	2.088305	-0.269068	0.732749

表格 1 各秒把手位置

2. 龙头、龙身前把手和龙尾后把手速度求解结果

题目所求0s、60s、120s、180s、240s、300s对应的龙头、龙身前把手和龙尾后把的速度由表 2 给出。300s内间隔1s的其他把手速度结果保存在 result1.xlsx 中。

	0s	60s	120s	180s	240s	300s
龙头(m/s)	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
第 1 节龙身(m/s)	0.999909	0.999871	0.999838	0.999786	0.999689	0.999513
第 51 节龙身(m/s)	0.999689	0.999586	0.999451	0.999226	0.998812	0.997902
第 101 节龙身(m/s)	0.999538	0.999399	0.999209	0.998902	0.998352	0.997188
第 151 节龙身(m/s)	0.999420	0.999257	0.999031	0.998673	0.998048	0.996757
第 201 节龙身(m/s)	0.999329	0.999150	0.998901	0.998511	0.997841	0.996479
龙尾（后）(m/s)	0.999296	0.999110	0.998853	0.998453	0.997769	0.996386

表格 2 各秒把手速度

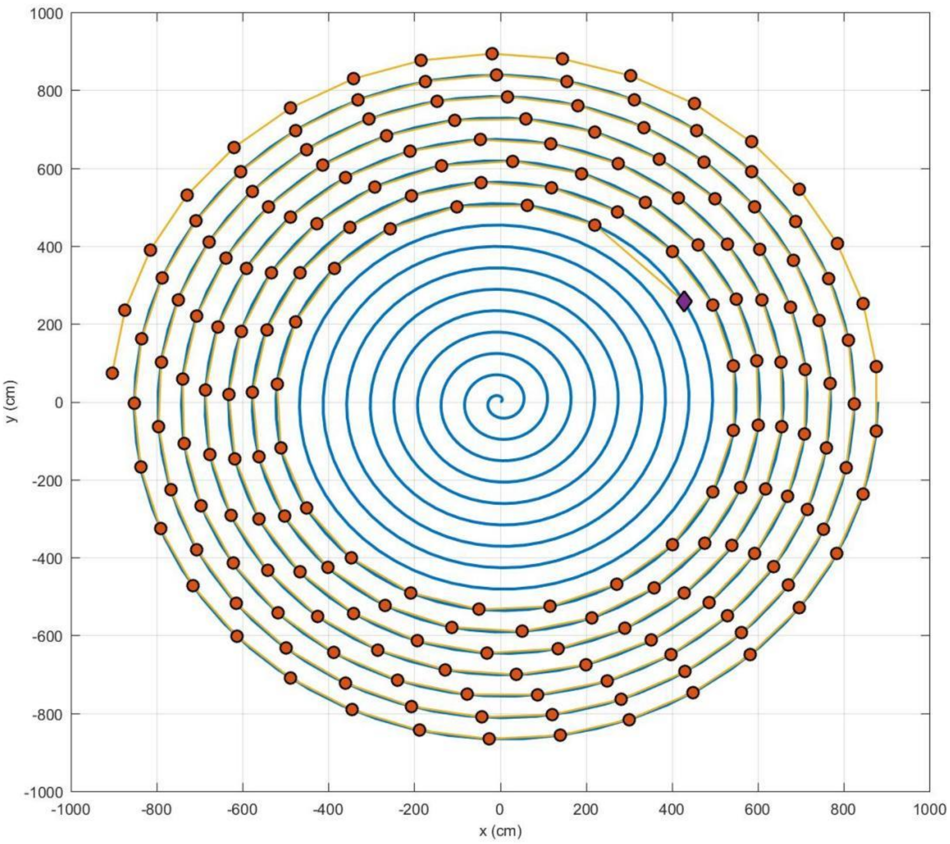


图 2 $t = 300$时舞龙队伍位置

板凳龙在$t = 300s$时的运动位置如图 2 所示。其中蓝色螺旋线是板凳龙的运动轨迹；紫色菱形为龙头前把手；橙红色圆形为各节龙身把手；橙黄色连线是各个板凳。可以直观地看到，在$t = 300$时刻下，板凳龙仍可以继续盘入，这对问题 2 的的研究有重要意义。

5.2 问题二模型建立与求解

5.2.1 碰撞点分析

基于图 3 中龙头和龙身的简化几何模型，在其顺时针盘入过程中，通过对板凳龙把手位置和板凳形状的分析，龙头上的 N 点将首先与某一节龙身接触发生碰撞。

由第一问结果可知，t在$[0,300]$的范围内龙头未与任何一节龙身碰撞。为降低计算量，观察t等于300时龙头和龙身的运动轨迹（图 2），根据由外向内每圈龙身数目递减的规律，合理选择可能发生碰撞的龙身节数范围n和时间范围，据此建立对应的碰撞检测模型，求出发生碰撞的时刻。

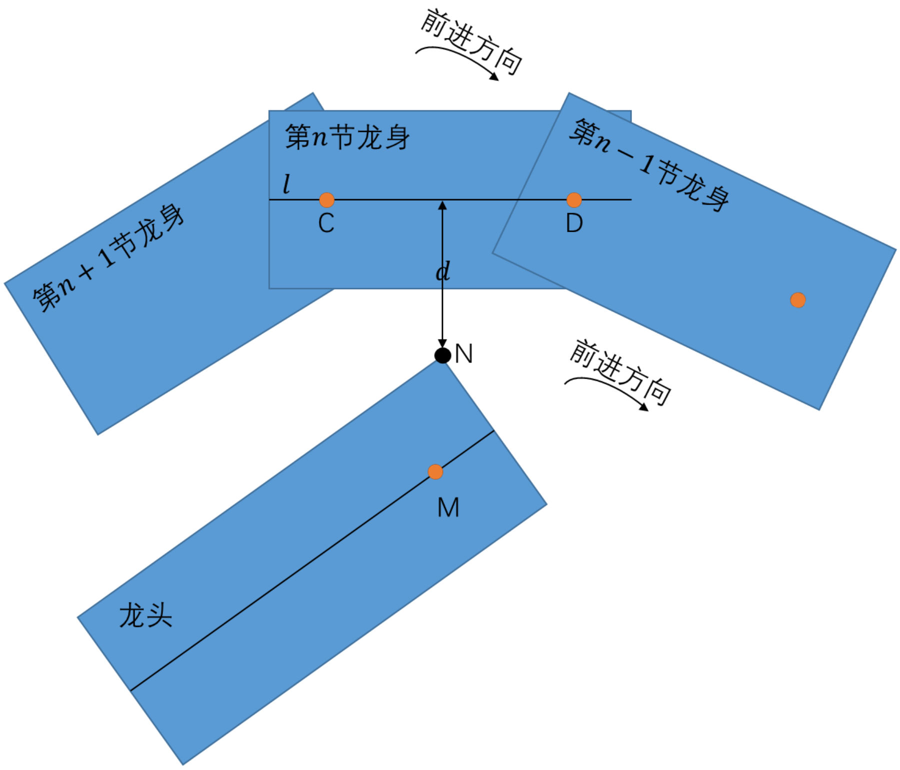


图 3 碰撞条件示意图

5.2.2 碰撞模型建立

在问题 1 中已经得到t时刻下任一把手的具体位置(x, y)。龙头几何模型如图 4，可以通过余弦定理解三角形，求出θ_N和NO的长度，继而确定出 N 点位置(x_N, y_N)：

$$\begin{cases} \cos\alpha = \dfrac{OM^2+MR^2-OR^2}{2\cdot OM\cdot MR} \\ \tan\beta = \dfrac{NK}{KM} \\ \theta = \pi - \alpha \\ \cos(\theta+\beta) = \dfrac{NM^2+MO^2-NO^2}{2\cdot NM\cdot MO} \\ \cos\gamma = \dfrac{OM^2+ON^2-MN^2}{2\cdot OM\cdot ON} \end{cases}$$

其中，NK、KM、MR由题目可知分别是0.15m、0.275m、2.86m。

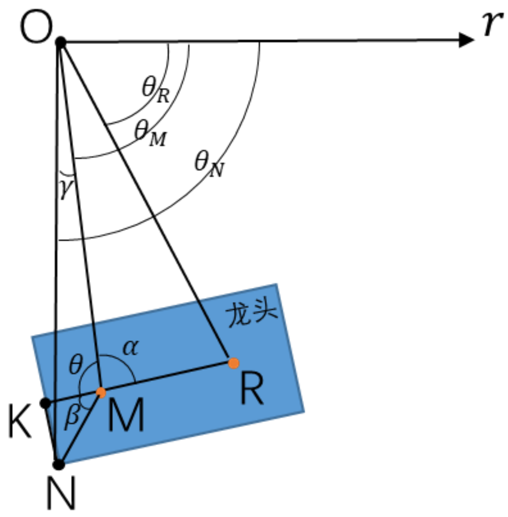


图 4 N 点位置推导图

由（1）式可以将t时刻龙头前把手M和后把手R的位置分别转化为极坐标。根据（）式求出N点的极坐标(r_{ON},θ_N)。转化为直角坐标后，确定(x_N,y_N)。此为任一时刻t下N点的轨迹方程。

已知第i节龙身前、后把手中心的位置C_i、D_i（如图 3），由点斜式确定过C_i、D_i点的线段l：

$$y-y_{C_i}=\frac{y_{C_i}-y_{D_i}}{x_{C_i}-x_{D_i}}\left(x-x_{C_i}\right) \tag{17}$$

此为t时刻下，l的轨迹方程。以l代表龙身中轴。

l的斜率k_i等于$\frac{y_{C_i}-y_{D_i}}{x_{C_i}-x_{D_i}}$。由点到直线距离公式可知，若龙头与这节龙身发生碰撞，满足N点与线段l间距离不大于0.15m，即：

$$d=\frac{\left|k_i x_N-y_N+y_{C_i}-k_i X_{C_i}\right|}{\sqrt{1+k_i^2}}\leq 0.15 \tag{18}$$

时将会发生碰撞，记此时的时间为t_0。其中d为N点与l间距离。

5.2.3 求解过程及结果分析

1.求解过程

Step1：初始化设置：在问题 1 的基础上，选择t遍历[400,500]的时间范围，步长为1；n遍历[0,80]的龙身范围，步长为1。以（18）式为终止条件开始检测是否碰撞。

Step2：遍历 t、n 查找碰撞点及龙身节数：从$t=400$时刻开始，计算当n遍

历范围$[0,160]$时是否满足碰撞条件。如果没有发生碰撞，那么时刻t将前进到下一时刻，并重新进行n的遍历计算，这个过程持续进行，直到找到碰撞点，从而确定碰撞发生的具体时间和与龙头发生碰撞的龙身体节编号。

Step3：停止时刻速度和位置求解： 得到发生碰撞时的t_0后，与问题 1 相同，由位置方程$x(t)$、$y(t)$可得各把手具体位置；由差分法可近似解出对应时刻的速度。

2. 求解结果

上述程序在 MATBAL 中运行求得在龙头与龙身发生碰撞的t_0为413.2s，此时舞龙队伍各把手的位置如图所示，其中紫色菱形代表龙头前把手。

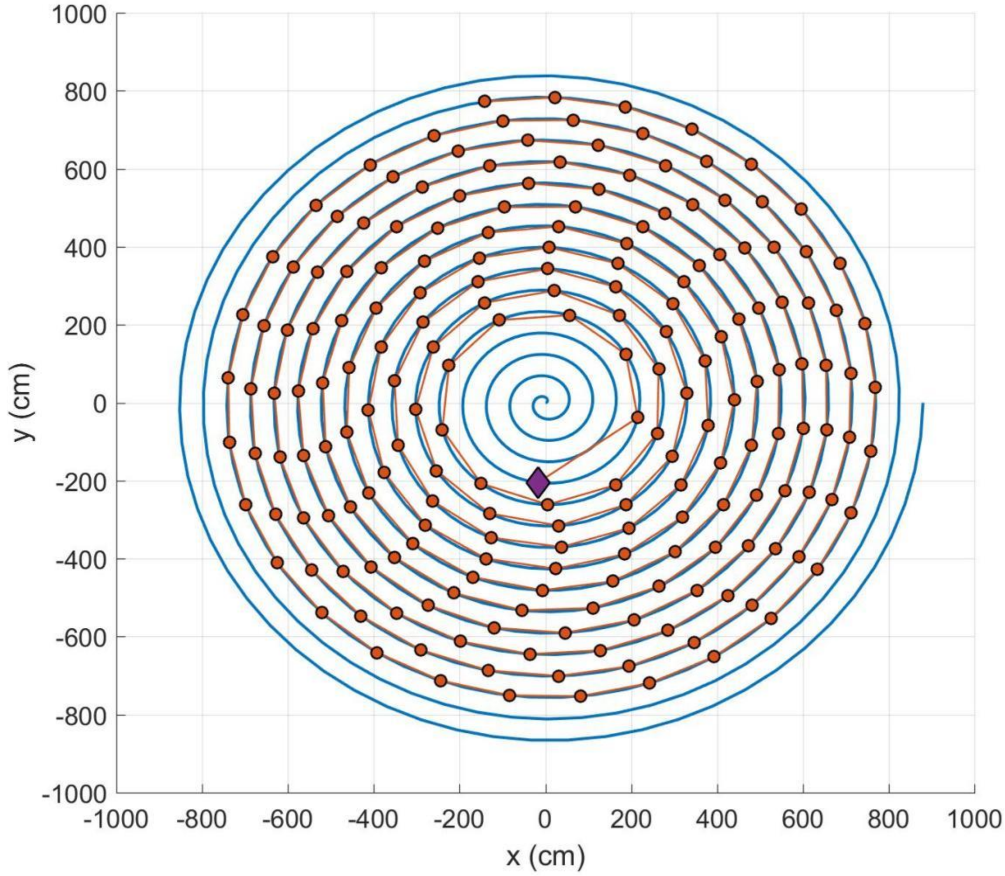


图 5 碰撞时舞龙队伍位置

题目所求碰撞时，龙头、龙身前把手和龙尾后把的位置如表 3 所示。其他各把手具体位置已保存在附件 result2.xlsx 中。

	$t_0=413.2s$
龙头x(m)	-1.376999
龙头y(m)	-0.133645
第 1 节龙身x(m)	1.412278
第 1 节龙身y(m)	-0.765731

第 51 节龙身x(m)	-3.589033
第 51 节龙身y(m)	1.943686
第 101 节龙身x(m)	3.559419
第 101 节龙身y(m)	-4.094230
第 151 节龙身x(m)	2.340142
第 151 节龙身y(m)	-6.058524
第 201 节龙身x(m)	-6.047813
第 201 节龙身y(m)	4.214998
龙尾（后）x(m)	7.572109
龙尾（后）y(m)	1.488316

表格 3 碰撞时把手位置

t_0对应的龙头、龙身前把手和龙尾后把的速度由表 4 给出。其他各把手速度结果保存在 result2.xlsx。

	$t_0 = 413.2s$
龙头(m/s)	1.000000
第 1 节龙身(m/s)	0.790549
第 51 节龙身(m/s)	0.758087
第 101 节龙身(m/s)	0.755801
第 151 节龙身(m/s)	0.754910
第 201 节龙身(m/s)	0.754452
龙尾（后）(m/s)	0.754313

表格 4 碰撞时把手速度

5.3 问题三模型建立与求解

5.3.1 板凳龙碰撞模型初始值的调整

在问题 1、2 中，舞龙队伍始终以既定的55cm为螺距向中心盘旋。但在问题 3 中，需要将螺距转化为本模型的内生变量。讨论在螺距递减的时，龙头可以沿着对应螺线行进到半径为$r_0 = 4.5$米的圆的边界而不发生碰撞的最小螺距。

观察图 6 和图 7，对于不同的螺距u_1和u_2，其他条件既定，易知，随螺距缩短，龙头与龙身的碰撞点不断外移。问题转变为求能使碰撞点外移至距圆心4.5m处的最小螺距。

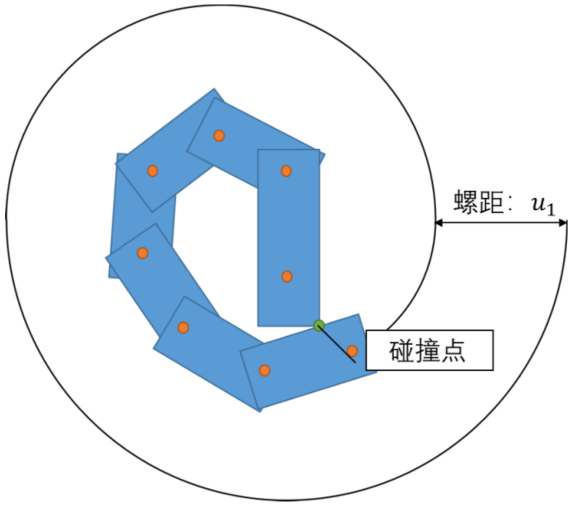


图 6 螺距与碰撞点关系示意图(a)

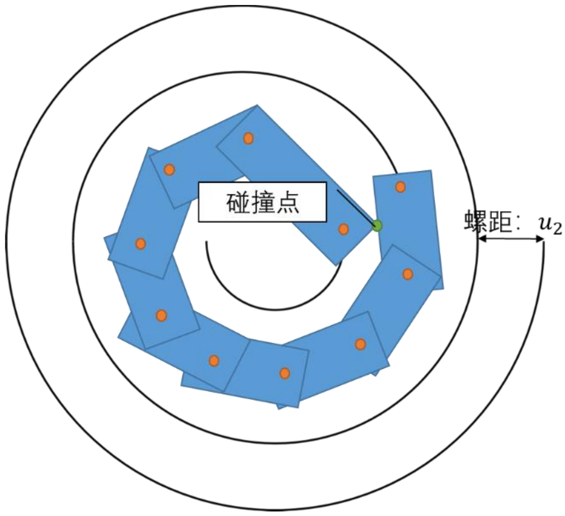


图 7 螺距与碰撞点关系示意图(b)

本题继续沿用问题一以及问题二中的板凳龙的运动模型。将螺距转化为变量并迭代求解。同时，重新选取合理的参数遍历范围进行计算。

5.3.2 求解过程及结果分析

1.求解过程

Step1：初始化设置：选取t遍历[1,430]， 步长为1；n遍历[0,60]，步长为1；在螺距u遍历范围的选取上，由于本题涉及t、n、u三个变量的迭代求解，为了缩短算法计算时间和降低计算量，先以[0,0.55]为范围，步长为0.05m进行求解。

Step2：粗略查找目标螺距：从$\mathrm{u}=0.55$开始，依次递减u值直到$\mathrm{u}=0$；在此过程中遍历t，对应每一螺距，从0依次递增直到$\mathrm{t}=430$；逐一查找每一t对应的n，得到$(\mathrm{u},\mathrm{t},\mathrm{n})$的这样一个参数点。由（18）式判断碰撞与否。记录停止时刻龙头前把手的极坐标，其半径为碰撞半径$\mathrm{r_k}$。比较$\mathrm{r_k}$与题目所给调头区域半径r_0的

大小，若$\mathrm{r_k} > r_0$，表明发生碰撞时未能进入调头区域，螺距过小；若$\mathrm{r_k} < r_0$，表明发生碰撞时已经进入调头区域。据此估算出满足题目要求的最小螺距大致范围。

Step3：精确查找目标螺距： 由上述过程推断出$\mathrm{r_k} = r_0$时的螺距范围大致在[0.4,0.5]之间，为提高精确度，仅在[0.4,0.5]之间查找u的精确取值并选取步长为0.001m，重复上述过程可得精确的目标螺距$\mathrm{r_k}$。

2.求解结果

由 MATBAL 运行上述程序，得到问题 3 使得龙头前把手能够沿着相应的螺线盘入到调头空间的边界的最小螺距为$\mathbf{0.4490m}$。

5.4 问题四模型建立与求解

5.4.1 S形曲线分析

1.前提假设

板凳龙进入调头空间可以开始“S”调头，在S形曲线上，板凳龙应满足如下前提：

（1） 板凳在S形曲线盘入和盘出阶段不相交。在板凳龙行进的过程中，考虑板凳宽度，使得出入的板凳不会发生碰撞；

（2） S形曲线应在9m为直径的调头区域内；

（3） 由于圆弧的圆心和半径难以确定，在本模型中，假设两条圆弧均为半圆，仍满足大半圆半径是小半圆半径的两倍的数量关系

（4） 为求解便利，假设各把手在S形曲线上匀速运动，大小为进入E点前一刻的瞬时速度。

2.最短弧长分析

如图 8 展示了板凳龙盘入与盘出的两条螺距为1.7m的螺旋线，分别为蓝色的盘入螺线和橙色盘出螺线，黄色圆形为直径9m的调头区域。

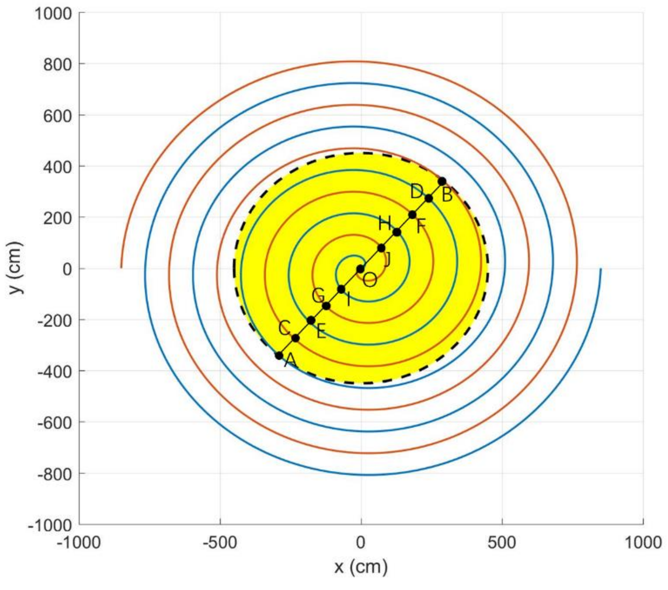


图 8 盘入盘出及调头示意图

由于S形曲线需要与盘入和盘出螺线相切，且两条螺线关于圆心O中心对称，所以S形曲线起始点和终止点仅能在过原点的l_{AB}与螺线交点中选择，A、B为调头区域与螺线的切点。则起始点-终止点有$A-B$、$C-D$、$E-F$、$G-H$、$I-J$共 5 种选择，且两两与O点中心对称。因此，可能的S形曲线分别有五个。由图，若选取S_{I-J}线或S_{G-H}线，观察其纵坐标区间，易知此时半径过小，不满足前提假设（1），故只能从S_{E-F}、S_{A-B}、S_{C-D}中选取。根据题目要求，选择S_{E-F}满足与盘入、盘出螺线均相切且使得调头曲线尽可能短。

3. 切点位置

如图 8，设切点E、F坐标分别是(x,y)和$(-x,-y)$。E、F到圆心O的距离为$r_E = r_F = 1.5u$，这两点均在盘入与盘出螺线上，满足（2）式，可得θ_E、θ_F。根据（1）将E、F的极坐标转化为直角坐标。

5.4.2 各把手运动状态分析及模型建立

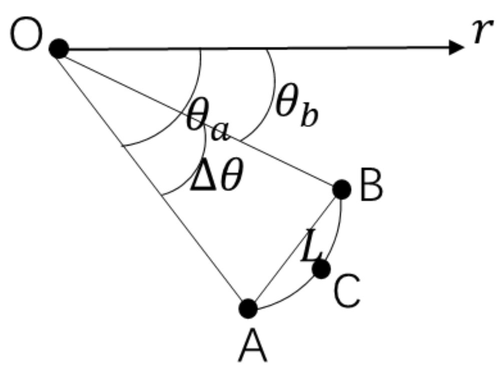
为简化模型，将各把手运动过程分为三段：进入S形曲线前、S形曲线上、离开S形曲线。

1.进入S形曲线前运动模型建立

此阶段以龙头前把手运动到E点为分界。在此之前，各把手均位于盘入螺线上，根据问题一所建立的螺旋运动模型，可求得 t 时刻各把手的位置和速度。根据板凳长度和龙头前把手运动速度，计算出相邻各把手大致以$2s$的时间间隔进入E点开始第二阶段，且该估计值在后续求解中误差较小。从龙头前把手经过E点为0时刻，第$2ns$时刻第n节龙身经过E点，若在此时刻，尽管该把手肯定未达到E点，本模型中会直接刷新该点的位置为E。

由于仅计算了把手$t = 2n$时的位置信息，需要估计$t = n$时的位置。这里根据图中所示，已知一点 A 的极坐标$(rA, 0A)$，将弧长ACB近似认为是相邻把手之间的距离LAB，A、B均在螺旋线上，有：

$$\begin{cases} \dfrac{L}{\Delta\theta} = \Delta r \\ \Delta r = a + b\Delta\theta \end{cases}$$



可近似解得 A 点把手前一秒的极坐标，即其位置信息。

2. S形曲线上点的运动模型建立

如图，把手进入 S 形曲线后，先在大圆弧上运动，后进入小圆弧。大圆弧弧

长为$s_2=\pi\rho$，小圆弧弧长为$s_1=\frac{1}{2}\pi\rho$。根据其速度v_t可以计算出 t 时刻在 S 形曲线上走过的弧长s_t，判断s_t与s_2的大小关系可确定把手所处的大致位置。

（1）大圆弧运动模型

已知某把手运动到大圆弧上T点，为求得此时T的极坐标，需要根据余弦定理解三角形。在图中确定调头区域直径EF为3倍的螺径；根据两段圆弧半径数量的倍数关系可知ER和RF；在某一 t 时刻下，由于把手在S形曲线匀速前进，可得弧ET、RJ 为含 t 的变量。对于ΔETO和ΔETO_2，满足：

$$\begin{cases} \cos\alpha_2=\dfrac{EO_2^2+TO_2^2-ET^2}{2\cdot EO_2\cdot TO_2} \\ \cos(\angle FO_2T)=\dfrac{OO_2^2+TO_2^2-OT^2}{2\cdot OO_2\cdot TO_2} \\ \cos\beta_2=\dfrac{OO_2^2+OT^2-TO_2^2}{2\cdot OT\cdot OO_2} \end{cases} \tag{20}$$

（2）小圆弧运动模型

当把手运动运动到小圆弧上后，与 T 点类似，需要根据余弦定理解三角形求得 J 点坐标。在图中，对于ΔOJO_1和ΔRJO_1满足:

$$\begin{cases} \cos\alpha_1=\dfrac{RO_1^2+JO_1^2-JR^2}{2\cdot RO_1\cdot JO_1} \\ \cos\beta_1=\dfrac{OO_1^2+OJ^2-JO_1^2}{2\cdot OJ\cdot OO_1} \\ \cos\alpha_1=\dfrac{OO_1^2+OJ^2-JO_1^2}{2\cdot OO_1\cdot OJ} \end{cases}$$

可以得出t时刻，某把手在大圆弧或小圆弧上运动的极坐标，带入（1）式，可得具体位置。在 S 形曲线上，把手运动速度匀速不变。

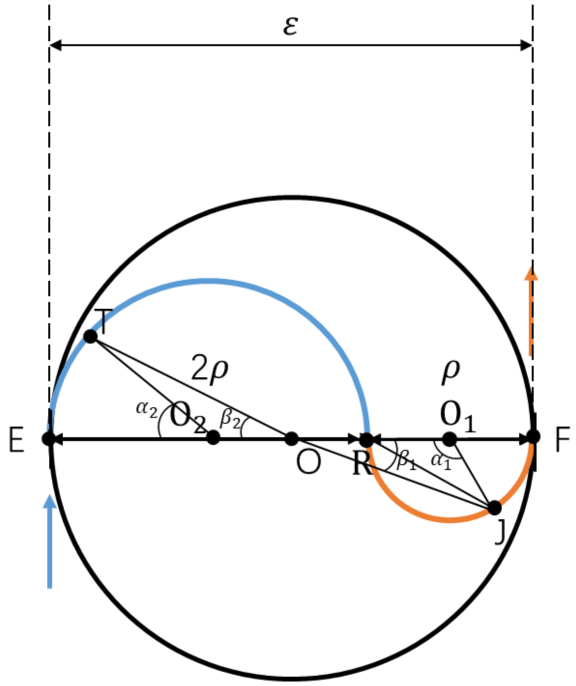


图 9 T、J 点位置推导

计算大小圆弧弧长和每秒运动速度的商，可以得到在同一时刻t，大圆弧上应有5个把手，小圆弧上应有3个把手。为简化模型，将T_5未到R点部分的弧长T_5R记入J_1对应的弧长RJ_1，以相应的计算方法可以得出小圆弧上一点的具体位置。

3.离开S形曲线的运动模型建立

此阶段由龙头前把手经过 F 点为分界。在此之后，经过F点后的把手全部位于盘出螺线上，根据螺旋运动模型求解出离开S形曲线后把手的位置和速度。由于每一时刻下，均有8个把手位于S形曲线上，根据1中确定的各把手进入时刻，可以推算出对应的离开时刻，即在$t = 2n + 8$时，第n节把手经过F点离开。引入参数i，记录t时刻经过F点离开的把手数目。

同样，由于仅计算了把手$t = 2n + 8$时的位置信息，需要估计$t = 2n + 9$时的位置。盘入、盘出螺线，E、F点中心对称，通过前式易得。

综合上述三个模型，可以解得 t 时刻各把手的位置和速度。

5.4.3 求解过程及结果分析

1. 求解过程

Step1：初始化设置：取t步长为1开始遍历，输出当龙头到达切入点的时刻为1348s。此时刻记为0，取t遍历[−100,100]步长为1开始程序。

Step2：进入S形曲线前运动把手位置和速度求解：根据 t 判断把手是否未经过E点，则其在螺旋线上。根据问题 1 建立的运动模型，其t时刻的位置和速度易得。若此时把手在 E 上，则通过中心差分法计算出其前一秒的位置和速度并保存此速度值。

Step3：S形曲线上把手位置和速度求解：根据 t 判断把手是否已处于 S 上，。

则具体位置由（22）或（23）式代入t求出。根据本模型假设，把手在S形曲线上的速度是恒定的，其速度为 2 中保存的对应速度值

Step4：离开S形曲线前运动把手位置和速度求解：根据 t 和 i 判断出把手是否经过 F 点，则其处于盘出螺旋线上，与 2 同理可得其位置和速度

2. 求解结果

在[−100,100]时刻内，题目所求对应的龙头、龙身前把手和龙尾后把的位置如表所示，其他各时刻的具体位置已保存在 result4.xlsx。

	−100s	−50s	0s	50s	100s
龙头x(m)	-721.919520	-432.434004	340.000000	384.499025	732.881255
龙头y(m)	-282.098537	379.742532	0.000000	-424.067430	248.429898
第 1 节龙身x(m)	-579.194117	-574.095667	-65.104363	-1843.643846	-2930.493119
第 1 节龙身y(m)	-529.940138	131.291383	-284.037936	2194.822150	-908.659812
第 51 节龙身x(m)	945.634508	49.115877	-197.831884	2116.664236	2659.367329
第 51 节龙身y(m)	411.009380	889.653030	702.605479	1763.403577	-497.121910
第 101 节龙身x(m)	169.269160	-1054.752989	434.436851	-817.662100	1448.279257
第 101 节龙身y(m)	1217.046175	-358.001755	-889.357726	2714.362156	2380.824882
第 151 节龙身x(m)	510.507546	-1043.068298	1178.640710	-2599.435615	-807.464568
第 151 节龙身y(m)	1295.376583	-762.547636	-139.108263	1306.858862	2746.352968
第 201 节龙身x(m)	1524.590108	-1435.397609	1236.398323	-2940.256353	-2344.415784
第 201 节龙身y(m)	158.450943	141.381745	-539.237157	-480.197722	1763.213297
龙尾（后）x(m)	-958.337739	814.413488	-561.010504	-2751.080501	-2746.759603
龙尾（后）y(m)	1268.739032	-1263.218524	1297.447001	-1218.950578	1113.151301

表格 5 [−100,100]把手位置

题目所求对应的龙头、龙身前把手和龙尾后把的速度在下表中列出，其他时刻的速度保存在 result4.xlsx。

	−100s	−50s	0s	50s	100s
龙头(m/s)	99.492933	99.073241	96.673173	0	0
第 1 节龙身(m/s)	99.528426	99.155952	95.949628	113.898640	81.231418
第 51 节龙身(m/s)	99.542863	99.178898	95.887756	111.368107	79.381501
第 101 节龙身(m/s)	99.550451	99.189348	95.865326	111.028608	79.236896
第 151 节龙身(m/s)	99.555121	99.195313	95.853761	111.443076	79.659381
第 201 节龙身(m/s)	99.556677	99.197229	95.850217	111.691595	79.893189
龙尾（后）(m/s)	99.492933	99.073241	96.673173	0	0

	$-100s$	$-50s$	$0s$	$50s$	$100s$
龙头(m/s)	99.492933	99.073241	96.6731735544915	0	0
第 1 节龙身(m/s)	99.528426	99.155952	95.9496287558008	113.898639999932	81.2314184461561
第 51 节龙身(m/s)	99.542863	99.178898	95.8877569597145	111.368106662618	79.3815014349805
第 101 节龙身(m/s)	99.550451	99.189348	95.8653269471249	111.028608080587	79.2368963653156
第 151 节龙身(m/s)	99.555121	99.195313	95.8537605703868	111.443076959631	79.6593817265074
第 201 节龙身(m/s)	99.556677	99.197229	95.8502165926692	111.691595734691	79.8931894106820
龙尾（后）(m/s)	99.492933	99.073241	96.6731735544915	0	0

表格 6 $[-100,100]$把手速度

5.5 问题五模型建立与求解

5.5.1 S 形曲线运动模型初始值调整

在问题四里，龙头以恒定的 v=1 米/秒的速度进行运动。但在问题五中，需要把龙头的速度作为一个可变参数纳入模型中。探讨当龙头速度逐渐增加时，在龙身各把手速度不超过 2 米/秒的约束条件下，其能够承受的最大龙头速度是多少。本问题依然采用问题四中板凳龙 S 形运动模型，并将通过迭代求解龙头速度作为变量时的情况。

5.5.2 求解过程及结果分析

1. 求解过程

Step1：初始化设置：给定龙头初始速度$v = 1m/s$，步长等于1。

Step2：龙头速度算法迭代：从$v = 1$开始遍历的值，查找每一时刻t下，其他各节龙身把手速度是否都不超过$2m/s$。在满足此条件的情况下使v尽可能大，当找到某节龙身在某一时刻t下速度大于$2m/s$ 时，确定最大龙头速度v。

2．求解结果

根据MATBAL求解结果为$v = 2.007m/s$。

六、模型分析

6.1 灵敏度分析

该题通过建立板凳龙运动模型以及碰撞模型求解问题，为检验上述模型的灵敏度，需要建立对关键参数对结果影响的评估模型，即考虑龙头初速度、螺距、初始半径或者螺距系数等参数的变化对最终结果的影响。

该评估模型采用定义参数扰动、重新运行模型以及对灵敏度的计算的方式进行评估。

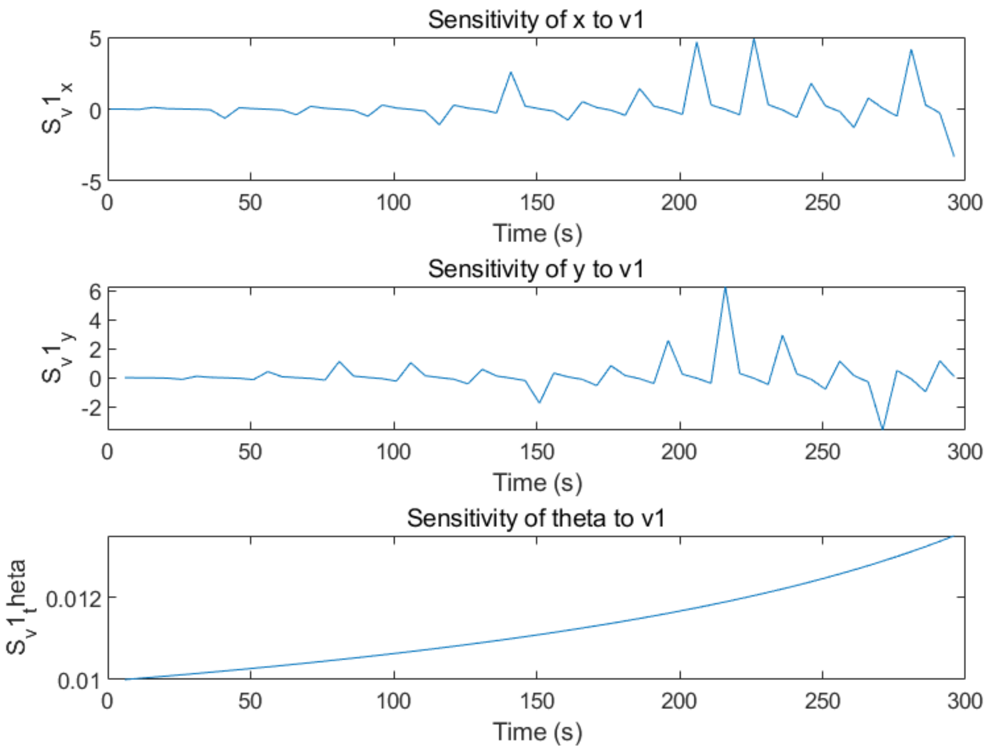
1.板凳龙运动模型：

（1）在该评估模型中，选取龙头初速度作为分析对象，设定$\Delta v = 0.01 \cdot v$

（2）在新的参数 $v + \Delta v$ 下重新运行模型，得到新的输出 $xnew, ynew, thetanew$。记录原始参数 v 下的输出 $x, y, theta$。

（3）计算灵敏度

计算输出的变化量 $\Delta y = ynew - y$。使用灵敏度公式计算灵敏度 Sp：$Sp =$（$\Delta y/\Delta p$）$*$（p/y），这里$\Delta y/\Delta p$）$*$（p/y)表示输出的变化量除以参数的变化量，即相对变化率；



2. 碰撞模型：

对于该模型采用与板凳龙运动模型相同的评价模型，通过定义参数扰动大小，重新运行模型，并计算其灵敏度，得到如下结果：

碰撞时间(s)	第一节龙身速	龙头 x(m)	龙头 y(m)

		度(m/s)		
原数据	414	97.6161	180.8365	134.9006
扰动数据	412	97.6287	174.3849	143.8139

6.2 误差分析

对于板凳龙运动模型和碰撞模型，误差主要是由求解微分方程是产生。该模型中采用欧拉方法进行求解微分方程，其局部截断误差为O（dt^2)，全局误差为O（dt）。

七、模型总结

7.1 模型优点

1. 本文基于等距螺旋线的方程，对坐标其进行适当处理，再跟据余弦定理，建立把手的螺旋运动模型。后续求出的数值解较为符合实际情况，能够较合理地描述任一时刻，各把手的位置和速度信息。

2.本文基于平面上点到直线距离公式，对板凳几何模型进行分析，建立碰撞模型。能较合理地计算出再多种约束条件下发生碰撞的时刻。

3.本文对目标函数做**离散化处理**，有效降低了数值计算的难度，为模型求解提供便利。

4.本文采用**递归算法**，求解复杂迭代条件下的多变量的最值问题。将原微分方程严谨变形、将算法科学改进后，实现了对问题的高精度求解，使结果更符合减小误差的要求。

7.2 模型缺点

1. 计算 S 形曲线上把手的速度时，为简化处理，忽略了其他圆弧形状。在曲线上认为是匀速运动，与实际速度存在偏差。

2.计算微分方程时，精度不够高，导致数据存在偏差。

7.3 模型推广与改进

本文较完整地分析了板凳龙在各种情况下各把手的运动过程。综合考虑了多种影响的因素，提供了板凳龙的螺旋运动模型和碰撞检测模型。本文献可以帮助推算在既定条件下板凳各把手的运动状态，并且能够较为精确地判断各个把手之间的位置关系。

然而在实际的运动中，不同板凳之间的运动会相互影响速度变化复杂，尤其是在复杂的“调头”运动中，各个板凳的运动模型用本文方法难以预测，之后会考虑结合更加精确的微分方程进行建模，并以为基础进行改进。

参考文献

[1] 赵黄磊,徐华兵.等距螺旋运动背景类试题的等效处理[J].高中数理化,2021,(17):44-45.

[2] 沈强,刘洁瑜,赵乾,等.非线性系统中心差分集员估计方法[J].控制理论与应用,2019,36(08):1239-1249.

[3] 涂德新,姜付锦.等距圆锥螺旋线数理性质的初探[J].物理通报,2018,(12):75-77+80.

[4] 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.

[5] 钱臻,齐英杰.迭代中心差分粒子滤波的 SLAM 算法[J].哈尔滨工程大学学报,2012,33(03):355-360.

[6] 白小虎.文化内生制度与经济发展的文化解释——鸡毛换糖、义乌兵与板凳龙[J].浙江社会科学,2006,(02):116-122.DOI:10.14167/j.zjss.2006.02.021.

[7] 汪国平,刘占平,华宣积,等.螺旋曲面的等距曲面[J].计算机辅助设计与图形学学报,2000,(05):321-324.

附录 A 问题 1

```
% 常量定义
v1 = 100;          % 线速度 cm/s
r0=55;             % 螺距   cm
a = 16*r0;   % 初始半径
b = r0 / (2*pi); % 螺距 b = P / (2*pi)
t_total = 301;    % 总时间
dt = 0.1;            % 时间步长
L1=341-27.5*2;
L2=220-27.5*2;
n_steps = t_total / dt;

% 初始化时间和角度
theta = zeros(n_steps, 224); % 存储每个时间步的角度，步数 X 第 i 个板凳
x=zeros(n_steps,224);
y=zeros(n_steps,224);
r=zeros(n_steps,224);
v=zeros(301,224);
v(:,1)=v1;
theta(1,1) = 0;                % 初始角度
time = (0:dt:t_total)';

% 数值积分：通过时间积分得到角度 theta(t)
for t_idx = 2:n_steps
    dtheta_dt = -v1 / sqrt(b^2 + (a + b * theta(t_idx-1,1))^2);
    theta(t_idx,1) = theta(t_idx-1,1) + dtheta_dt * dt;
end

% 计算轨迹
x(:,1) = (a + b .* theta(:,1)) .* cos(theta(:,1));
y(:,1) = (a + b .* theta(:,1)) .* sin(theta(:,1));
%计算第一个龙身位置
for t_idx = 1:n_steps
    r(t_idx,1)=sqrt(x(t_idx,1)^2+y(t_idx,1)^2);
    F=@(angle) r(t_idx,1)^2+(a+b*angle)^2-
2*r(t_idx,1)*(a+b*angle)*cos(theta(t_idx,1)-angle)-L1^2;
    if(t_idx==1)
        start0=L1/880;
    else
        start0=theta(t_idx-1,2);
    end
    theta(t_idx,2)=fsolve(F,start0);
    x(t_idx,2) = (a + b .* theta(t_idx,2)) .* cos(theta(t_idx,2));
```

```
    y(t_idx,2) = (a + b .* theta(t_idx,2)) .* sin(theta(t_idx,2));

end
%计算其余龙身
for i_loong = 3:224
    for t_idx = 1:n_steps
        r(t_idx,i_loong-1)=sqrt(x(t_idx,i_loong-
1)^2+y(t_idx,i_loong-1)^2);
        F=@(angle) r(t_idx,i_loong-1)^2+(a+b*angle)^2-
2*r(t_idx,i_loong-1)*(a+b*angle)*cos(theta(t_idx,i_loong-1)-angle)-
L2^2;
       if(t_idx==1)
              start0=L1/880+(i_loong-2)*L2/r(1,i_loong-1);
       else
           start0=theta(t_idx-1,i_loong);
       end
        theta(t_idx,i_loong)=fsolve(F,double(start0));
        x(t_idx,i_loong) = (a + b .* theta(t_idx,i_loong)) .* cos(th
eta(t_idx,i_loong));
        y(t_idx,i_loong) = (a + b .* theta(t_idx,i_loong)) .* sin(th
eta(t_idx,i_loong));
    end
end
for i_loong=2:224
    for t_idx=1:301
        if(t_idx==1)
            vx=abs(x(2,i_loong)-x(1,i_loong))/dt;
            vy=abs(y(2,i_loong)-y(1,i_loong))/dt;
        elseif(t_idx==301)
            vx=abs(x(3010,i_loong)-x(3009,i_loong))/dt;
            vy=abs(y(3010,i_loong)-y(3009,i_loong))/dt;
        else
            vx=abs(x(10*(t_idx-1)+10*dt,i_loong)-x(10*(t_idx-1)-
10*dt,i_loong))/(2*dt);
            vy=abs(y(10*(t_idx-1)+10*dt,i_loong)-y(10*(t_idx-1)-
10*dt,i_loong))/(2*dt);
        end
        v(t_idx,i_loong)=sqrt(vx^2+vy^2);
    end
end
```

附录 B 问题 2

```
v1 = 100;          % 线速度 cm/s
r0=55;            % 螺距   cm
a = 16*r0;  % 初始半径
b = r0 / (2*pi); % 螺距 b = P / (2*pi)
t_start = 401;   %起始时间
t_total = 501;   % 总时间
dt = 0.1;          % 时间步长
L1=341-27.5*2;
L2=220-27.5*2;
n_A2steps = (t_total-t_start) / dt;
% 初始化时间和角度
theta11 = zeros(n_A2steps, 224); % 存储每个时间步的角度，步数 X 第 i 个板
発
r11=zeros(n_A2steps,224);
x11=zeros(n_A2steps,224);
y11=zeros(n_A2steps,224);
v_A2=zeros(224,1);

x11(1,:)=x(401,:);             % 从 401-1 s 处计算
y11(1,:)=y(401,:);
theta11(1,:)=theta(401,:);
 %计算 400s 时 r 值 r11(1,:)=sqrt(x(401,:)^2+y(401,:)^2);
for i_loong1=1:224
   r11(1,i_loong1)=sqrt(x(401,i_loong1)^2+y(401,i_loong1)^2);
end
temp=0; %标记是否发生碰撞
%计算“龙头”对角点坐标位置
for i_A2steps=2:n_A2steps
    if(temp==1)
        break;
    end
    %基于当前位置对于后面把手位置及角度的计算
    %计算龙头位置
    dtheta_dt = -v1 / sqrt(b^2 + (a + b * theta11(i_A2steps-
1,1))^2);
    theta11(i_A2steps,1) = theta11(i_A2steps-1,1) + dtheta_dt * dt;
    x11(i_A2steps,1) = (a + b .* theta11(i_A2steps,1)) .* cos(theta1
1(i_A2steps,1));
    y11(i_A2steps,1) = (a + b .* theta11(i_A2steps,1)) .* sin(theta1
1(i_A2steps,1));
    r11(i_A2steps,1)=   a + b .* theta11(i_A2steps,1);
    %计算第一节龙身位置
```

```
    F=@(angle) r11(i_A2steps,1)^2+(a+b*angle)^2-
2*r11(i_A2steps,1)*(a+b*angle)*cos(theta11(i_A2steps,1)-angle)-L1^2;
    start0=theta11(i_A2steps-1,2);
    theta11(i_A2steps,2)=fsolve(F,start0);
    x11(i_A2steps,2) = (a + b .* theta11(i_A2steps,2)) .* cos(theta1
1(i_A2steps,2));
    y11(i_A2steps,2) = (a + b .* theta11(i_A2steps,2)) .* sin(theta1
1(i_A2steps,2));
    r11(i_A2steps,2)=   a + b .* theta11(i_A2steps,2);
    %计算其余龙身
    for i_loong = 3:224
            F=@(angle) r11(i_A2steps,i_loong-1)^2+(a+b*angle)^2-
2*r11(i_A2steps,i_loong-
1)*(a+b*angle)*cos(theta11(i_A2steps,i_loong-1)-angle)-L2^2;
            start0=theta11(i_A2steps-1,i_loong);
            theta11(i_A2steps,i_loong)=fsolve(F,double(start0));
            x11(i_A2steps,i_loong) = (a + b .* theta11(i_A2steps,i_l
oong)) .* cos(theta11(i_A2steps,i_loong));
            y11(i_A2steps,i_loong) = (a + b .* theta11(i_A2steps,i_l
oong)) .* sin(theta11(i_A2steps,i_loong));
            r11(i_A2steps,i_loong)=a + b .* theta11(i_A2steps,i_loon
g);
    end
    for t_idx2 = 1:160
        Rb = r11(i_A2steps,1); feib = theta11(i_A2steps,1);%当前时刻
龙头前把手
        Rc = r11(i_A2steps,2); feic = theta11(i_A2steps,2);%当前时刻
龙头后把手
        dfei1 =abs(feib - feic);
        %正弦定理
        feiabc = abs(asin(Rc*sin(dfei1)/L1));
        feiabd = pi - feiabc + atan(15/27.5);
        r_point = sqrt((15^2+27.5^2)+(Rb)^2-
2 * (15^2+27.5^2)^0.5 * Rb * cos(feiabd));

        %正弦定理
        feidab = abs(asin(((15^2+27.5^2)^0.5) * sin(feiabd)/r_point)
);
        %解得目标角度
        feid = -feidab + feib;

        %转化为直角坐标
        x_point = r_point * cos(feid);
        y_point = r_point * sin(feid);
```

```
        %建立可疑碰撞板凳的直线模型
        x21 = x11(i_A2steps,t_idx2+1); y21 = y11(i_A2steps,t_idx2+1)
;
        x22 = x11(i_A2steps,t_idx2+2); y22 = y11(i_A2steps,t_idx2+2)
;
        k=(y22-y21)/(x22-x21);
        %求出点到直线的距离
        Dec = abs(k*x_point-y_point+y21-k*x21 )/sqrt(1 + k^2 );
        if(Dec <= 15)
            %标志位置一
            disp('此时为: ');
            disp(Rb);
            disp(feib)
            disp('板凳碰撞! ');
            temp=1;
            R_confict=Rb;
            feib_confict=feib;
            T_confict=i_A2steps;
            break;
        end
    end
end
v_A2(1)=v1;
for i_loong2=2:224
      vx=abs(x11(T_confict,i_loong2)-x11(T_confict-1,i_loong2))/dt;
      vy=abs(y11(T_confict,i_loong2)-y11(T_confict-1,i_loong2))/dt;
      v_A2(i_loong2)=sqrt(vx^2+vy^2);
end
disp(T_confict);
```

附录 C 问题 3

```
% 常量定义
v1 = 100;          % 线速度 cm/s
r0=55;            % 螺距   cm
t_totalA3= 434;   % 总时间
dtA3 = 1;           % 时间步长
r_cricle=900/2;     %圆半径 cm
dr_change=0.01;     %螺距变化量由 5 逐步递减，求精确结果
L1=341-27.5*2;
L2=220-27.5*2;
n_stepsA3 = t_totalA3 / dtA3;
time = (0:dtA3:t_totalA3)';
R_confict=140;
while(R_confict<=r_cricle)
    r0 = r0 - dr_change;
    % 初始化时间和角度
    theta_A376 = zeros(n_stepsA3, 224); % 存储每个时间步的角度，步数 X
第 i 个板凳
    x_A376=zeros(n_stepsA3,224);
    y_A376=zeros(n_stepsA3,224);
    r_A376=zeros(n_stepsA3,224);
    theta_A376(1,1) = 0;                   % 初始角度
    a = 16*r0;  % 初始半径
    b = r0 / (2*pi); % 螺距 b = P / (2*pi)

    % 数值积分：通过时间积分得到角度 theta(t)
    % 解出当前螺距所有龙头在不同时间步的位置

    for t_idxA376 = 2:n_stepsA3
        dtheta_dt = -v1 / sqrt(b^2 + (a + b * theta_A376(t_idxA376-
1,1))^2);
        theta_A376(t_idxA376,1) = theta_A376(t_idxA376-
1,1) + dtheta_dt * dtA3;
    end

    % 计算龙头轨迹
    x_A376(:,1) = (a + b .* theta_A376(:,1)) .* cos(theta_A376(:,1))
;
    y_A376(:,1) = (a + b .* theta_A376(:,1)) .* sin(theta_A376(:,1))
;
    %计算所有时间步下第一个龙身位置
    for t_idxA376 = 1:n_stepsA3
```

```
        r_A376(t_idxA376,1)=sqrt(x_A376(t_idxA376,1)^2+y_A376(t_idxA
376,1)^2);
        F=@(angle) r_A376(t_idxA376,1)^2+(a+b*angle)^2-
2*r_A376(t_idxA376,1)*(a+b*angle)*cos(theta_A376(t_idxA376,1)-
angle)-L1^2;
        if(t_idxA376==1)
            start0_A3=L1/880;
        else
            start0_A3=theta_A376(t_idxA376-1,2);
        end
        theta_A376(t_idxA376,2)=fsolve(F,start0_A3);
        x_A376(t_idxA376,2) = (a + b .* theta_A376(t_idxA376,2)) .*
cos(theta_A376(t_idxA376,2));
        y_A376(t_idxA376,2) = (a + b .* theta_A376(t_idxA376,2)) .*
sin(theta_A376(t_idxA376,2));
    end
    %计算其余龙身
    for i_loongA3 = 3:61
        for t_idxA376 = 1:n_stepsA3
            r_A376(t_idxA376,i_loongA3-
1)=sqrt(x_A376(t_idxA376,i_loongA3-1)^2+y_A376(t_idxA376,i_loongA3-
1)^2);
            F=@(angle) r_A376(t_idxA376,i_loongA3-
1)^2+(a+b*angle)^2-2*r_A376(t_idxA376,i_loongA3-
1)*(a+b*angle)*cos(theta_A376(t_idxA376,i_loongA3-1)-angle)-L2^2;
           if(t_idxA376==1)
                   start0_A3=L1/880+(i_loongA3-
2)*L2/r_A376(1,i_loongA3-1);
           else
               start0_A3=theta_A376(t_idxA376-1,i_loongA3);
           end
            theta_A376(t_idxA376,i_loongA3)=fsolve(F,double(start0_A
3));
            x_A376(t_idxA376,i_loongA3) = (a + b .* theta_A376(t_idx
A376,i_loongA3)) .* cos(theta_A376(t_idxA376,i_loongA3));
            y_A376(t_idxA376,i_loongA3) = (a + b .* theta_A376(t_idx
A376,i_loongA3)) .* sin(theta_A376(t_idxA376,i_loongA3));
        end
    end
    temp=0;
    for t_idxA376 = 1:n_stepsA3
         if(temp==1)
            break;
         end
```

```
        for t_idx3 = 1:59
                Rb = r_A376(t_idxA376,1); feib = theta_A376(t_idxA376,1);%当前时刻龙头前把手
                Rc = r_A376(t_idxA376,2); feic = theta_A376(t_idxA376,2);%当前时刻龙头后把手
                dfei1 =abs(feib - feic);
                %正弦定理
                feiabc = asin(Rc*sin(dfei1)/L1);
                feiabd = pi - feiabc + atan(15/27.5);
                r_point = sqrt((15^2+27.5^2)+(Rb)^2-2 * (15^2+27.5^2)^0.5 * Rb * cos(feiabd));

                %正弦定理
                feidab = abs(asin(((15^2+27.5^2)^0.5) * sin(feiabd)/r_point));
                %解得目标角度
                feid = -feidab + feib;

                %转化为直角坐标
                x_point = r_point * cos(feid);
                y_point = r_point * sin(feid);

                %建立可疑碰撞板凳的直线模型
                x21 = x_A376(t_idxA376,t_idx3+1); y21 = y_A376(t_idxA376,t_idx3+1);
                x22 = x_A376(t_idxA376,t_idx3+2); y22 = y_A376(t_idxA376,t_idx3+2);
                k=(y22-y21)/(x22-x21);
                %求出点到直线的距离
                Dec = abs(k*x_point-y_point+y21-k*x21 )/sqrt(1 + k^2 );
                if(Dec <= 15)
                    %标志位置一
                    temp=1;
                    R_confict=Rb;
                    break;
                end
        end
    end
end
disp(r0);
```

附录 D 问题 4

```
%A4_1.m
% 常量定义
v1 = 100;           % 线速度 cm/s
r0=170;             % 螺距   cm
a = 16*r0;  % 初始半径                          原本是%;a = 2*r0
b = r0 / (2*pi); % 螺距 b = P / (2*pi)
t_total = 1500;   % 总时间
dt = 1;            % 时间步长
L1=341-27.5*2;
L2=220-27.5*2;
n_steps = t_total / dt;

% 初始化时间和角度
theta = zeros(n_steps, 224); % 存储每个时间步的角度，步数 X 第 i 个板凳
x=zeros(n_steps,224);
y=zeros(n_steps,224);
r=zeros(n_steps,224);
v=zeros(1500,224);
v(:,1)=v1;
theta(1,1) = 0; % 初始角度      %原本是 theta(1,1) =;-5 * pi/ 6
time = (0:dt:t_total)';

% 数值积分：通过时间积分得到角度 theta(t)
for t_idx = 2:n_steps
    dtheta_dt = -v1 / sqrt(b^2 + (a + b * theta(t_idx-1,1))^2); %原
本是 ;dtheta_dt = v1 / sqrt(b^2 + (a + b * theta(t_idx-1,1))^2)
    theta(t_idx,1) = theta(t_idx-1,1) + dtheta_dt * dt;
end

% 计算轨迹
x(:,1) = (a + b .* theta(:,1)) .* cos(theta(:,1));
y(:,1) = (a + b .* theta(:,1)) .* sin(theta(:,1));
%计算第一个龙身位置
for t_idx = 1:n_steps
    r(t_idx,1)=sqrt(x(t_idx,1)^2+y(t_idx,1)^2);
    if(r(t_idx,1) <= 255)
        break;
    end
    F=@(angle) r(t_idx,1)^2+(a+b*angle)^2-
2*r(t_idx,1)*(a+b*angle)*cos(theta(t_idx,1)-angle)-L1^2;
    if(t_idx==1)
        start0=L1/880;
```

```
    else
        start0=theta(t_idx-1,2);
    end
    theta(t_idx,2)=fsolve(F,start0);
    x(t_idx,2) = (a + b .* theta(t_idx,2)) .* cos(theta(t_idx,2));
    y(t_idx,2) = (a + b .* theta(t_idx,2)) .* sin(theta(t_idx,2));

end
%计算其余龙身
for i_loong = 3:224
    for t_idx = 1:n_steps
        r(t_idx,i_loong-1)=sqrt(x(t_idx,i_loong-
1)^2+y(t_idx,i_loong-1)^2);
        F=@(angle) r(t_idx,i_loong-1)^2+(a+b*angle)^2-
2*r(t_idx,i_loong-1)*(a+b*angle)*cos(theta(t_idx,i_loong-1)-angle)-
L2^2;
       if(t_idx==1)
              start0=L1/880+(i_loong-2)*L2/r(1,i_loong-1);
       else
           start0=theta(t_idx-1,i_loong);
       end
        theta(t_idx,i_loong)=fsolve(F,double(start0));
        x(t_idx,i_loong) = (a + b .* theta(t_idx,i_loong)) .* cos(th
eta(t_idx,i_loong));
        y(t_idx,i_loong) = (a + b .* theta(t_idx,i_loong)) .* sin(th
eta(t_idx,i_loong));
    end
end

%A4_2.m
dt = 1;
for i_loong=2:224
    for t_idx=578:677
        if(t_idx == 578)
        vx=abs(x(t_idx+dt,i_loong)-x(t_idx,i_loong))/dt;
        vy=abs(y(t_idx+dt,i_loong)-y(t_idx,i_loong))/dt;
        end

        if(t_idx == 677)
        vx=abs(x(t_idx,i_loong)-x(t_idx-dt,i_loong))/dt;
        vy=abs(y(t_idx,i_loong)-y(t_idx-dt,i_loong))/dt;
        end

        if(t_idx >= 579 && t_idx <= 676 )
```

```
        vx=abs(x(t_idx+dt,i_loong)-x(t_idx-dt,i_loong))/(2*dt);
        vy=abs(y(t_idx+dt,i_loong)-y(t_idx-dt,i_loong))/(2*dt);
        end
        v(t_idx,i_loong)=sqrt(vx^2+vy^2);
    end
end

%A4_3.m
% 常量定义
v1 = 100;         % 线速度 cm/s
r0=170;           % 螺距   cm
a_A42 = 16*r0;  % 初始半径                        原本是%;a = 2*r0
b_A42 = r0 / (2*pi); % 螺距 b = P / (2*pi)
t_total_A42 = 101;   % 总时间
dt_A42 = 1;          % 时间步长
L1=341-27.5*2;
L2=220-27.5*2;
n_steps = t_total_A42 / dt_A42;
t_end=1357;          %龙头前把手到达进入点时刻
R1=2*3*1.7/6;
R2=3*1.7/6;
L_round1=2*pi*3*1.7/6;
L_round2=pi*3*1.7/6;
% 初始化时间和角度
theta_A42 = zeros(n_steps, 224); % 存储每个时间步的角度，步数 X 第 i 个板凳
x_A42=zeros(n_steps,224);
y_A42=zeros(n_steps,224);
r_A42=zeros(n_steps,224);
v_A42=zeros(101,224);
time_A42 = (0:dt_A42:t_total_A42)';
%各把手进入时刻状态矩阵
t_statein=zeros(224,1);
%初始化各点位置
x_A42(1,:)=x(t_end,:);
y_A42(1,:)=y(t_end,:);
r_A42(1,:)=r(t_end,:);
v_A42(1,:)=v(t_end,:);
theta_A42(1,:)=theta(t_end,:); % 初始角度
%初始化龙头前把手位置
v_A42(:,1)=v1; x_A42(1,1)=340;y_A42(1,1)=0;r_A42(1,1)=340;
%初始化时刻状态矩阵
for t_idx_A4=1:101
   t_statein(t_idx_A4)=2*(t_idx_A4-1)+1;
```

```
end
%计算龙头盘出位置
% 数值积分：通过时间积分得到角度 theta(t)
theta_A42(12,1)=-28*pi;
x_A42(12,1)=255;y_A42(12,1)=0;
for t_idx = 13:n_steps
    dtheta_dt = v1 / sqrt(b_A42^2 + (a_A42 + b_A42 * theta_A42(t_idx
-1,1))^2);
    theta_A42(t_idx,1) = theta_A42(t_idx-1,1) + dtheta_dt * dt_A42;
    x_A42(t_idx,1) = -
(a_A42 + b_A42 .* theta_A42(t_idx,1)) .* cos(theta_A42(t_idx,1));
    y_A42(t_idx,1) = -
(a_A42 + b_A42 .* theta_A42(t_idx,1)) .* sin(theta_A42(t_idx,1));
end

cntin=1;cntout=0;
for t_idx_A4=2:101
    if(mod(t_idx_A4,2)==1)
        cntin=cntin+1;
       x_A42(t_idx_A4,cntin)=-255;
       y_A42(t_idx_A4,cntin)=0;
       theta_A42(t_idx_A4,cntin)=-29*pi;
       v_A42(t_idx_A4,cntin)=v_A42(t_idx_A4-1,cntin);
    else
        %计算第 cntin+1 个板凳的位置
        apf_change=L2/r_A42(t_idx_A4-1,cntin+1);
        theta_A42(t_idx_A4,cntin+1)=theta_A42(t_idx_A4-1,cntin+1)-
apf_change;
        r_A42(t_idx_A4,cntin+1)=-
(a_A42+b_A42*apf_change)+r_A42(t_idx_A4-1,cntin+1);
        x_A42(t_idx_A4,cntin+1)=r_A42(t_idx_A4,cntin+1)*cos(-
apf_change+theta_A42(t_idx_A4-1,cntin+1));
        y_A42(t_idx_A4,cntin+1)=r_A42(t_idx_A4,cntin+1)*sin(-
apf_change+theta_A42(t_idx_A4-1,cntin+1));
        vx=(x_A42(t_idx_A4,cntin+1)-x_A42(t_idx_A4-
1,cntin+1))/dt_A42;
        vy=(y_A42(t_idx_A4,cntin+1)-y_A42(t_idx_A4-
1,cntin+1))/dt_A42;
        v_A42(t_idx_A4,cntin+1)=sqrt(vx^2+vy^2);
    end
    %根据第 cnt+1 个点计算第 cnt+2 及以后板凳位置
    for i_outfront=cntin+1:224
        %迭代计算
```

```
        F=@(angle) r_A42(t_idx_A4,i_outfront-
1)^2+(a_A42+b_A42*angle)^2-2*r_A42(t_idx_A4,i_outfront-
1)*(a_A42+b_A42*angle)*cos(theta_A42(t_idx_A4,i_outfront-1)-angle)-
L2^2;
        start0=theta_A42(t_idx_A4-1,i_outfront);
        theta_A42(t_idx_A4,i_outfront)=fsolve(F,double(start0));
        x_A42(t_idx_A4,i_outfront) = (a_A42 + b_A42 .* theta(t_idx_A
4,i_outfront)) .* cos(theta_A42(t_idx_A4,i_outfront));
        y_A42(t_idx_A4,i_outfront) = (a_A42 + b_A42 .* theta(t_idx_A
4,i_outfront)) .* sin(theta_A42(t_idx_A4,i_outfront));
        r_A42(t_idx_A4,i_outfront)=sqrt(x_A42(t_idx_A4,i_outfront)^2
+y_A42(t_idx_A4,i_outfront)^2);
        vx=(x_A42(t_idx_A4,i_outfront)-x_A42(t_idx_A4-
1,i_outfront))/dt_A42;
        vy=(y_A42(t_idx_A4,i_outfront)-y_A42(t_idx_A4-
1,i_outfront))/dt_A42;
        v_A42(t_idx_A4,i_outfront)=sqrt(vx^2+vy^2);
    end
    if(t_idx_A4>8&&mod(t_idx_A4,2)==0)
       cntout=cntout+1;
%        x_A42(t_idx_A4,cntout)=-340;
%        y_A42(t_idx_A4,cntout)=0;
%        v_A42(t_idx_A4,cntout)=v_A42(t_idx_A4-11,cntout);
    end

    %计算圆弧中各点位置
    for i_in=1+cntout:cntin
        v_A42(t_idx_A4,i_in)=v_A42(t_idx_A4-1,i_in);
        L_now=v_A42(t_idx_A4,i_in)*(t_idx_A4-(2*i_in-1));
        if(L_now<L_round1)
           s_round1=1.5*1.7- R1;
           theta_round1=L_now/R1;
           r_round1=sqrt(s_round1^2+R1^2-2*s_round1*R1*cos(pi-
theta_round1));
           apf_round1=acos((s_round1^2+r_round1^2-
R1^2)/(2*r_round1*s_round1));
           x_A42(t_idx_A4,i_in)=r_round1*cos(apf_round1);
           y_A42(t_idx_A4,i_in)=r_round1*sin(apf_round1);
        else
            L_real=L_now-L_round1;
            s_round2=1.5*1.7- R2;
           theta_round2=L_real/R2;
           r_round2=sqrt(s_round2^2+R2^2-2*s_round2*R1*cos(pi-
theta_round2));
```

```
            apf_round2=acos((s_round2^2+r_round2^2-
R2^2)/(2*r_round2*s_round2));
            x_A42(t_idx_A4,i_in)=r_round2*cos(pi-apf_round2);
            y_A42(t_idx_A4,i_in)=r_round2*sin(pi-apf_round2);
        end
        %11 时刻后计算出去板凳位置
        if(t_idx_A4>8)
            if(cntout==1)
                continue;
            elseif(cntout==2)
                r_A42(t_idx_A4,1)=sqrt(x_A42(t_idx_A4,1)^2+y_A42(t_i
dx_A4,1)^2);
                F=@(angle) r_A42(t_idx_A4,1)^2+(a_A42+b_A42*angle)^2
+2*r_A42(t_idx_A4,1)*(a_A42+b_A42*angle)*cos(theta_A42(t_idx_A4,1)-
angle)-L1^2;
                start0=theta_A42(t_idx_A4-1,2);
                theta_A42(t_idx_A4,2)=fsolve(F,start0);
                x_A42(t_idx_A4,2) = -
(a_A42 + b_A42 .* theta_A42(t_idx_A4,2)) .* cos(theta_A42(t_idx_A4,2
));
                y_A42(t_idx_A4,2) = -
(a_A42 + b_A42 .* theta_A42(t_idx_A4,2)) .* sin(theta_A42(t_idx_A4,2
));
            else
                r_A42(t_idx_A4,1)=sqrt(x_A42(t_idx_A4,1)^2+y_A42(t_i
dx_A4,1)^2);
                F=@(angle) r_A42(t_idx_A4,1)^2+(a_A42+b_A42*angle)^2
+2*r_A42(t_idx_A4,1)*(a_A42+b_A42*angle)*cos(theta_A42(t_idx_A4,1)-
angle)-L1^2;
                start0=theta_A42(t_idx_A4-1,2);
                theta_A42(t_idx_A4,2)=fsolve(F,start0);
                x_A42(t_idx_A4,2) = -
(a_A42 + b_A42 .* theta_A42(t_idx_A4,2)) .* cos(theta_A42(t_idx_A4,2
));
                y_A42(t_idx_A4,2) = -
(a_A42 + b_A42 .* theta_A42(t_idx_A4,2)) .* sin(theta_A42(t_idx_A4,2
));
                for i_outlate=3:cntout
                    %迭代计算
                    F=@(angle) r_A42(t_idx_A4,i_outlate-
1)^2+(a_A42+b_A42*angle)^2+2*r_A42(t_idx_A4,i_outlate-
1)*(a_A42+b_A42*angle)*cos(theta_A42(t_idx_A4,i_outlate-1)-angle)-
L2^2;
                    start0=theta_A42(t_idx_A4-1,i_outlate);
```

```
                    theta_A42(t_idx_A4,i_outlate)=fsolve(F,double(st
art0));
                    x_A42(t_idx_A4,i_outlate) = -
(a_A42 + b_A42 .* theta(t_idx_A4,i_outlate)) .* cos(theta_A42(t_idx_
A4,i_outlate));
                    y_A42(t_idx_A4,i_outlate) = -
(a_A42 + b_A42.* theta(t_idx_A4,i_outlate)) .* sin(theta_A42(t_idx_A
4,i_outlate));
                    r_A42(t_idx_A4,i_outlate)=sqrt(x_A42(t_idx_A4,i_
outlate)^2+y_A42(t_idx_A4,i_outlate)^2);
                    vx=(x_A42(t_idx_A4,i_outlate)-x_A42(t_idx_A4-
1,i_outlate))/dt_A42;
                    vy=(y_A42(t_idx_A4,i_outlate)-y_A42(t_idx_A4-
1,i_outlate))/dt_A42;
                    v_A42(t_idx_A4,i_outlate)=sqrt(vx^2+vy^2);
                end
            end
        end
    end
end
```

附录 E 问题 5

```
%A5_1.m
% 常量定义
v1 = 200.7;          % 线速度 cm/s
r0=170;             % 螺距   cm
a = 16*r0;  % 初始半径                         原本是%;a = 2*r0
b = r0 / (2*pi); % 螺距 b = P / (2*pi)
t_total = 700;   % 总时间
dt = 1;            % 时间步长
L1=341-27.5*2;
L2=220-27.5*2;
n_steps = t_total / dt;

% 初始化时间和角度
theta = zeros(n_steps, 224); % 存储每个时间步的角度，步数 X 第 i 个板凳
x=zeros(n_steps,224);
y=zeros(n_steps,224);
r=zeros(n_steps,224);
v=zeros(1500,224);
v(:,1)=v1;
theta(1,1) = 0; % 初始角度      %原本是 theta(1,1) =;-5 * pi/ 6
time = (0:dt:t_total)';

% 数值积分：通过时间积分得到角度 theta(t)
for t_idx = 2:n_steps
    dtheta_dt = -v1 / sqrt(b^2 + (a + b * theta(t_idx-1,1))^2); %原
本是 ;dtheta_dt = v1 / sqrt(b^2 + (a + b * theta(t_idx-1,1))^2)
    theta(t_idx,1) = theta(t_idx-1,1) + dtheta_dt * dt;
end

% 计算轨迹
x(:,1) = (a + b .* theta(:,1)) .* cos(theta(:,1));
y(:,1) = (a + b .* theta(:,1)) .* sin(theta(:,1));
%计算第一个龙身位置
for t_idx = 1:n_steps
    r(t_idx,1)=sqrt(x(t_idx,1)^2+y(t_idx,1)^2);
    if(r(t_idx,1) <= 255)
        break;
    end
    F=@(angle) r(t_idx,1)^2+(a+b*angle)^2-
2*r(t_idx,1)*(a+b*angle)*cos(theta(t_idx,1)-angle)-L1^2;
    if(t_idx==1)
        start0=L1/880;
```

```
    else
        start0=theta(t_idx-1,2);
    end
    theta(t_idx,2)=fsolve(F,start0);
    x(t_idx,2) = (a + b .* theta(t_idx,2)) .* cos(theta(t_idx,2));
    y(t_idx,2) = (a + b .* theta(t_idx,2)) .* sin(theta(t_idx,2));

end
%计算其余龙身
for i_loong = 3:224
    for t_idx = 1:n_steps
        r(t_idx,i_loong-1)=sqrt(x(t_idx,i_loong-
1)^2+y(t_idx,i_loong-1)^2);
        F=@(angle) r(t_idx,i_loong-1)^2+(a+b*angle)^2-
2*r(t_idx,i_loong-1)*(a+b*angle)*cos(theta(t_idx,i_loong-1)-angle)-
L2^2;
       if(t_idx==1)
              start0=L1/880+(i_loong-2)*L2/r(1,i_loong-1);
       else
           start0=theta(t_idx-1,i_loong);
       end
        theta(t_idx,i_loong)=fsolve(F,double(start0));
        x(t_idx,i_loong) = (a + b .* theta(t_idx,i_loong)) .* cos(th
eta(t_idx,i_loong));
        y(t_idx,i_loong) = (a + b .* theta(t_idx,i_loong)) .* sin(th
eta(t_idx,i_loong));
    end
end

%A5_2.m
dt = 1;
for i_loong=2:224
    for t_idx=578:677
        if(t_idx == 578)
        vx=abs(x(t_idx+dt,i_loong)-x(t_idx,i_loong))/dt;
        vy=abs(y(t_idx+dt,i_loong)-y(t_idx,i_loong))/dt;
        end

        if(t_idx == 677)
        vx=abs(x(t_idx,i_loong)-x(t_idx-dt,i_loong))/dt;
        vy=abs(y(t_idx,i_loong)-y(t_idx-dt,i_loong))/dt;
        end

        if(t_idx >= 579 && t_idx <= 676 )
```

```
        vx=abs(x(t_idx+dt,i_loong)-x(t_idx-dt,i_loong))/(2*dt);
        vy=abs(y(t_idx+dt,i_loong)-y(t_idx-dt,i_loong))/(2*dt);
        end
        v(t_idx,i_loong)=sqrt(vx^2+vy^2);
    end
end
```

附录 F 灵敏度检验

```
v1 = 100;          % 线速度 cm/s
r0=55;             % 螺距   cm
a = 16*r0;  % 初始半径
b = r0 / (2*pi); % 螺距 b = P / (2*pi)
t_start = 401;   %起始时间
t_total = 501;   % 总时间
dt = 0.1;          % 时间步长
L1=341-27.5*2;
L2=220-27.5*2;
n_A2steps = (t_total-t_start) / dt;
% 初始化时间和角度
theta11 = zeros(n_A2steps, 224); % 存储每个时间步的角度，步数 X 第 i 个板凳
r11=zeros(n_A2steps,224);
x11=zeros(n_A2steps,224);
y11=zeros(n_A2steps,224);
v_A2=zeros(224,1);

x11(1,:)=x(401,:);              % 从 401-1 s 处计算
y11(1,:)=y(401,:);
theta11(1,:)=theta(401,:);
 %计算 400s 时 r 值 r11(1,:)=sqrt(x(401,:)^2+y(401,:)^2);
for i_loong1=1:224
   r11(1,i_loong1)=sqrt(x(401,i_loong1)^2+y(401,i_loong1)^2);
end
temp=0; %标记是否发生碰撞
%计算“龙头”对角点坐标位置
for i_A2steps=2:n_A2steps
    if(temp==1)
        break;
    end
    %基于当前位置对于后面把手位置及角度的计算
    %计算龙头位置
    dtheta_dt = -v1 / sqrt(b^2 + (a + b * theta11(i_A2steps-1,1))^2);
    theta11(i_A2steps,1) = theta11(i_A2steps-1,1) + dtheta_dt * dt;
```

```
    x11(i_A2steps,1) = (a + b .* theta11(i_A2steps,1)) .* cos(theta11(i
_A2steps,1));
    y11(i_A2steps,1) = (a + b .* theta11(i_A2steps,1)) .* sin(theta11(i
_A2steps,1));
    r11(i_A2steps,1)=   a + b .* theta11(i_A2steps,1);
    %计算第一节龙身位置
    F=@(angle) r11(i_A2steps,1)^2+(a+b*angle)^2-
2*r11(i_A2steps,1)*(a+b*angle)*cos(theta11(i_A2steps,1)-angle)-L1^2;
    start0=theta11(i_A2steps-1,2);
    theta11(i_A2steps,2)=fsolve(F,start0);
    x11(i_A2steps,2) = (a + b .* theta11(i_A2steps,2)) .* cos(theta11(i
_A2steps,2));
    y11(i_A2steps,2) = (a + b .* theta11(i_A2steps,2)) .* sin(theta11(i
_A2steps,2));
    r11(i_A2steps,2)=   a + b .* theta11(i_A2steps,2);
    %计算其余龙身
    for i_loong = 3:224
            F=@(angle) r11(i_A2steps,i_loong-1)^2+(a+b*angle)^2-
2*r11(i_A2steps,i_loong-1)*(a+b*angle)*cos(theta11(i_A2steps,i_loong-
1)-angle)-L2^2;
            start0=theta11(i_A2steps-1,i_loong);
            theta11(i_A2steps,i_loong)=fsolve(F,double(start0));
            x11(i_A2steps,i_loong) = (a + b .* theta11(i_A2steps,i_loon
g)) .* cos(theta11(i_A2steps,i_loong));
            y11(i_A2steps,i_loong) = (a + b .* theta11(i_A2steps,i_loon
g)) .* sin(theta11(i_A2steps,i_loong));
            r11(i_A2steps,i_loong)=a + b .* theta11(i_A2steps,i_loong);
    end
    for t_idx2 = 1:160
        Rb = r11(i_A2steps,1); feib = theta11(i_A2steps,1);%当前时刻龙头
前把手
        Rc = r11(i_A2steps,2); feic = theta11(i_A2steps,2);%当前时刻龙头
后把手
        dfei1 =abs(feib - feic);
        %正弦定理
        feiabc = abs(asin(Rc*sin(dfei1)/L1));
        feiabd = pi - feiabc + atan(15/27.5);
        r_point = sqrt((15^2+27.5^2)+(Rb)^2-
2 * (15^2+27.5^2)^0.5 * Rb * cos(feiabd));

        %正弦定理
        feidab = abs(asin(((15^2+27.5^2)^0.5) * sin(feiabd)/r_point));
        %解得目标角度
        feid = -feidab + feib;
```

```
        %转化为直角坐标
        x_point = r_point * cos(feid);
        y_point = r_point * sin(feid);

        %建立可疑碰撞板凳的直线模型
        x21 = x11(i_A2steps,t_idx2+1); y21 = y11(i_A2steps,t_idx2+1);
        x22 = x11(i_A2steps,t_idx2+2); y22 = y11(i_A2steps,t_idx2+2);
        k=(y22-y21)/(x22-x21);
        %求出点到直线的距离
        Dec = abs(k*x_point-y_point+y21-k*x21 )/sqrt(1 + k^2 );
        if(Dec <= 15)
            %标志位置一
            disp('此时为：');
            disp(Rb);
            disp(feib)
            disp('板凳碰撞！');
            temp=1;
            R_confict=Rb;
            feib_confict=feib;
            T_confict=i_A2steps;
            break;
        end
    end
end
v_A2(1)=v1;
for i_loong2=2:224
      vx=abs(x11(T_confict,i_loong2)-x11(T_confict-1,i_loong2))/dt;
      vy=abs(y11(T_confict,i_loong2)-y11(T_confict-1,i_loong2))/dt;
      v_A2(i_loong2)=sqrt(vx^2+vy^2);
end
disp(T_confict);
v1 = 100;         % 线速度 cm/s
r0=55;            % 螺距   cm
a = 16*r0;  % 初始半径
b = r0 / (2*pi); % 螺距 b = P / (2*pi)
t_start = 401;   %起始时间
t_total = 501;   % 总时间
dt = 0.1;          % 时间步长
L1=341-27.5*2;
L2=220-27.5*2;
n_A2steps = (t_total-t_start) / dt;
% 初始化时间和角度
theta11 = zeros(n_A2steps, 224); % 存储每个时间步的角度，步数 X 第 i 个板凳
```

```
r11=zeros(n_A2steps,224);
x11=zeros(n_A2steps,224);
y11=zeros(n_A2steps,224);
v_A2=zeros(224,1);

x11(1,:)=x(401,:);             % 从 401-1 s 处计算
y11(1,:)=y(401,:);
theta11(1,:)=theta(401,:)*1.01;
 %计算 400s 时 r 值 r11(1,:)=sqrt(x(401,:)^2+y(401,:)^2);
for i_loong1=1:224
   r11(1,i_loong1)=sqrt(x(401,i_loong1)^2+y(401,i_loong1)^2);
end
temp=0; %标记是否发生碰撞
%计算“龙头”对角点坐标位置
for i_A2steps=2:n_A2steps
    if(temp==1)
        break;
    end
    %基于当前位置对于后面把手位置及角度的计算
    %计算龙头位置
    dtheta_dt = -v1 / sqrt(b^2 + (a + b * theta11(i_A2steps-1,1))^2);
    theta11(i_A2steps,1) = theta11(i_A2steps-1,1) + dtheta_dt * dt;
    x11(i_A2steps,1) = (a + b .* theta11(i_A2steps,1)) .* cos(theta11(i
_A2steps,1));
    y11(i_A2steps,1) = (a + b .* theta11(i_A2steps,1)) .* sin(theta11(i
_A2steps,1));
    r11(i_A2steps,1)=   a + b .* theta11(i_A2steps,1);
    %计算第一节龙身位置
    F=@(angle) r11(i_A2steps,1)^2+(a+b*angle)^2-
2*r11(i_A2steps,1)*(a+b*angle)*cos(theta11(i_A2steps,1)-angle)-L1^2;
    start0=theta11(i_A2steps-1,2);
    theta11(i_A2steps,2)=fsolve(F,start0);
    x11(i_A2steps,2) = (a + b .* theta11(i_A2steps,2)) .* cos(theta11(i
_A2steps,2));
    y11(i_A2steps,2) = (a + b .* theta11(i_A2steps,2)) .* sin(theta11(i
_A2steps,2));
    r11(i_A2steps,2)=   a + b .* theta11(i_A2steps,2);
    %计算其余龙身
    for i_loong = 3:224
            F=@(angle) r11(i_A2steps,i_loong-1)^2+(a+b*angle)^2-
2*r11(i_A2steps,i_loong-1)*(a+b*angle)*cos(theta11(i_A2steps,i_loong-
1)-angle)-L2^2;
            start0=theta11(i_A2steps-1,i_loong);
            theta11(i_A2steps,i_loong)=fsolve(F,double(start0));
```

```
            x11(i_A2steps,i_loong) = (a + b .* theta11(i_A2steps,i_loon
g)) .* cos(theta11(i_A2steps,i_loong));
            y11(i_A2steps,i_loong) = (a + b .* theta11(i_A2steps,i_loon
g)) .* sin(theta11(i_A2steps,i_loong));
            r11(i_A2steps,i_loong)=a + b .* theta11(i_A2steps,i_loong);
    end
    for t_idx2 = 1:160
        Rb = r11(i_A2steps,1); feib = theta11(i_A2steps,1);%当前时刻龙头
前把手
        Rc = r11(i_A2steps,2); feic = theta11(i_A2steps,2);%当前时刻龙头
后把手
        dfei1 =abs(feib - feic);
        %正弦定理
        feiabc = abs(asin(Rc*sin(dfei1)/L1));
        feiabd = pi - feiabc + atan(15/27.5);
        r_point = sqrt((15^2+27.5^2)+(Rb)^2-
2 * (15^2+27.5^2)^0.5 * Rb * cos(feiabd));

        %正弦定理
        feidab = abs(asin(((15^2+27.5^2)^0.5) * sin(feiabd)/r_point));
        %解得目标角度
        feid = -feidab + feib;

        %转化为直角坐标
        x_point = r_point * cos(feid);
        y_point = r_point * sin(feid);

        %建立可疑碰撞板凳的直线模型
        x21 = x11(i_A2steps,t_idx2+1); y21 = y11(i_A2steps,t_idx2+1);
        x22 = x11(i_A2steps,t_idx2+2); y22 = y11(i_A2steps,t_idx2+2);
        k=(y22-y21)/(x22-x21);
        %求出点到直线的距离
        Dec = abs(k*x_point-y_point+y21-k*x21 )/sqrt(1 + k^2 );
        if(Dec <= 15)
            %标志位置一
            disp('此时为：');
            disp(Rb);
            disp(feib)
            disp('板凳碰撞！');
            temp=1;
            R_confict=Rb;
            feib_confict=feib;
            T_confict=i_A2steps;
            break;
```

```
        end
    end
end
v_A2(1)=v1;
for i_loong2=2:224
      vx=abs(x11(T_confict,i_loong2)-x11(T_confict-1,i_loong2))/dt;
      vy=abs(y11(T_confict,i_loong2)-y11(T_confict-1,i_loong2))/dt;
      v_A2(i_loong2)=sqrt(vx^2+vy^2);
end
disp(T_confict);
```