“板凳龙”的运动学模拟与碰撞检测研究

摘要

“板凳龙”是一种传统的民俗活动，其中涉及到首尾相连的长方形板凳的运动轨迹问题以及碰撞问题。为此，本文对板凳龙的运动过程进行数学建模，建立微分方程模型预测了板凳龙在各种轨迹下不同时刻的位置和速度，并对其运动轨迹进行了优化。

问题一中，本文首先建立了**极坐标系**下的**微分方程模型**，求解后确定了龙头一定时刻在螺线上的位置。然后，本文通过将半径为把手间距的圆的方程与螺线的极坐标**方程联立**，得出了**龙身位置的递推方程**，从头到尾依次确定了板凳龙在各个时刻各把手的位置。最后，本文利用**求导的链式法则和隐函数定理**，基于此前得到的龙身位置的递推方程得出了**龙身速度的递推方程**，求出了板凳龙在各个时刻各把手的速度还使用了**差分法**再次计算速度，验证了速度迭代公式的正确性。**板凳龙位置与速度的数据见文中表1，表2和文件result1.xlsx**。

问题二中，本文首先建立了粗略判断板凳是否可能发生碰撞的**条带模型**，即通过计算板凳龙在运动过程中扫过的区域，将其抽象为落在螺线上的条带，并计算条带区域发生重叠的时刻，确定了板凳可能发生碰撞的起始时刻。随后，本文继续建立了更准确的**碰撞检测模型**，检测碰撞最先发生的龙头板凳的角点是否落在其它板凳的矩形范围内，最后，本文采用**变步长的遍历求解法**，求得在终止时刻**t=412.473838s**时，板凳龙即将发生碰撞，无法再向内盘入，**此时的板凳龙位置与速度的数据见表3,表4和文件result2.xlsx**。

问题三中，本文首先使用**条带模型**计算龙头与龙身在盘入螺线与掉头空间的切点（即进入调头空间点）处碰撞时螺距的大小。并在条带模型计算出的螺距附近使用更准确的**碰撞检测模型**进行搜索求解，通过使用足够小的螺距步长，准确求得最小螺距。最终求得最小螺距为**0.449810m**。

问题四中，本文首先在问题所给的约束条件下尝试调整了调头曲线的圆弧，发现在约束条件下，无论如何调整圆弧，调头曲线的总长度不变，**无法使调头曲线变短**。随后，本文在螺线上使用**极坐标系**，在调头曲线上使用**直角坐标系**，对舞龙队的运动轨迹进行了描述。然后，考虑到各段曲线无法用统一的方程表示，本文对不同情况**分类讨论**，确定了龙头在一定时刻的位置，并构建了不同情况下的**龙身位置的递推方程**，求出了板凳龙在各个时刻各把手的速度。最后，本文使用**差分法**计算得出了板凳龙各时刻把手的速度，**位置与速度的详细见表5，表6和文件result4.xlsx**。

问题五中，由于**板凳龙下一节点的位置仅由上一节点的位置唯一确定**，本文认为在任一时刻，**板凳龙上任一节点的速度与龙头速度成比例关系**。因此，基于问题四中得出的各把手的速度数据，寻找出龙身上最大速度出现的大致时间范围后，本文继续缩小时间步长，找出了**龙头前进速度为1m/s时**产生的最大速度，计算出**允许的龙身最大速度与该速度的比值**，与原本**1m/s**的龙头速度相乘，最终得出龙头的最大速度为**v=1.23246201969456m/s**

关键词: 螺线模型　叉积法　微分方程　递推关系　差分法

一、问题重述

1.1 问题背景

近年来，随着文化强国战略的提出，保护与发扬各地传统民俗文化，提高文化软实力的重要性日渐凸显。其中，在浙闽地区有一种古老的民俗活动“板凳龙”，又称“盘龙”，因其独特的观赏性和技巧性得到关注。盘龙时讲究一定的技巧，若能在舞龙队能自由的盘入盘出的前提下尽可能行进得更快，盘龙使用的面积更小，则观赏性越好。

为了响应保护与发扬民俗文化的号召，以数学建模的方式解决其中的技巧问题，不仅可以使其观赏性得到提高，也能让更多人了解这一优秀的民俗文化。

1.2 问题重述

板凳龙由223张板凳首尾相连组成。其中领头前进的第一张板凳为龙头，其余作为龙身和龙尾的相同的板凳相随盘旋，整体呈圆盘状。每张板凳均为长方形，前后各有一个孔，盘龙时，板凳之间由穿过孔的把手相连，且各把手的中心均位于一等距螺线上。题目给出了各张板凳的长和宽，板凳上的孔的中心位置以及孔径，要求解决以下问题：

问题一：题目给定舞龙队沿螺距为**55cm**的螺线盘入，且给定了龙头的行进速度为**1m/s**以及龙头的初始位置，要求计算出从初始时刻到300s为止每秒钟的整个舞龙队的位置和速度，并按要求将特定时刻特定序号的板凳的位置和速度填入表1和表2。

问题二：在与问题一相同的条件下，计算出舞龙队盘入的终止时刻，即板凳即将发生碰撞的时刻，给出整个舞龙队此时的位置和速度，并在论文中给出特定序号的板凳的位置和速度。

问题三：舞龙队盘出需要预留一定调头空间，给定调头空间为以螺线中心为圆心直径**9m**的圆形，给出能使龙头前把手沿螺线盘入掉头空间边界而不发生碰撞的最小螺距。

问题四：调头空间与问题三相同，给定盘入螺线的螺距为**1.7m**，龙头行进的速度为**1m/s**，盘入螺线与盘出螺线关于螺线中心呈中心对称。舞龙队在调头空间内沿S形曲线路径进行调头，且此曲线的形状以及约束条件由题目给定，要求判断是否存在符合条件的更短的路径。以调头开始时间为零时刻，给出-100s到100s内每秒的整个舞龙队的位置和速度，并在论文中特定时刻的特定序号板凳的位置和速度。

问题五：沿问题四设定的路径前进，龙头的行进速度为常数，确定使舞龙队各把手速度均不超过**2m/s**的最大龙头行进速度。

二、问题分析

2.1 问题一的分析

问题一要求我们给出板凳龙在不同时刻的各把手的位置和速度，要实现这一目的，需要建立一个板凳龙的运动模拟模型。首先，题目给出了板凳龙的运动轨迹为等距螺线，故建立极坐标系进行描述最为方便。然后，板凳龙头的速度和初始位置是给定的，结合已知的轨迹方程，即可建立微分方程求解得出任一时间龙头的位置的表达式。在确定龙头位置后，可以用递推的方式诸葛确定之后的龙身和龙尾的位置。递推时需要寻找轨迹上离前点距离为一定值的后点，这可以通过将轨迹方程和以前点为圆心，板凳上两把手中心间距为半径的圆的方程联立的方法得到。在得到

板凳龙上所有把手的位置后，我们首先想到可以用差分法求点的速度，但也可以对位置的递推方程进一步求导得到速度的递推方程，也由龙头的速度推出所有点的速度。

2.2 问题二的分析

问题二要求对板凳龙盘入的过程进行碰撞检测。板凳龙盘入时几何形状复杂，且时间跨度长，就该问题而言需要进行400秒以上的运动模拟及碰撞检测，且为满足碰撞检测所需精度，时间步长需要足够小。为避免巨大的计算开销，我们计划对模型进行一定程度的简化，确定碰撞时间所在区间，并辅以精确的运动过程模拟，确定碰撞的精确时间。进行分析后，我们认为板凳龙更可能在螺线靠内（曲率半径小）的位置发生碰撞，即对于螺线上某处位置，若板凳龙无法在此处发生碰撞，那么对于比该点更靠外（曲率半径更大）的位置，板凳龙更加不可能发生碰撞。另外，板凳龙首次发生碰撞时，应为板凳龙龙头与龙身发生碰撞。为了简化运动模型中板凳龙复杂的几何形状，我们将木板抽象为由龙身内侧偏移部分与龙头外侧偏移部分扫过的轨迹组成的**条带**，并计算第i+1层内层条带厚度与第i层外层条带厚度之和大于螺距的时刻。为了求得更加准确的碰撞时间，我们通过在条带模型计算出的时刻附近使用严格模拟板凳龙几何形状的**碰撞检测模型**，进行模拟，求解出精确的第一次碰撞时间。

2.3 问题三的分析

问题三与问题二相似，同样是对板凳龙盘入的运动过程进行模拟，并进行碰撞检测。该问不同之处在于由第二问的给定运动轨迹求碰撞时间改为了给出板凳龙不发生碰撞的区域（即掉头空间外的螺线区域）求最小螺距的优化问题。考虑到板凳龙在曲率半径更小的螺线处等容易发生碰撞，则在该问情形下板凳龙更容易在靠近掉头空间处发生碰撞，那么在螺距最小时,板凳龙将会在**盘入螺线与调头空间的切点**后发生碰撞。因此我们使用类似第二问中的方法，即先用**条带法**确定板凳龙在切点处恰好发生碰撞时螺距的大小，并在该螺距大小附近使用严格模拟的**碰撞检测**的方法进行搜索，找出符合要求的最小螺距大小。

2.4 问题四的分析

问题四要求我们调整板凳龙的调头曲线使其尽可能变短，这看起来是一个优化问题，但根据约束条件调整调头曲线时，调头曲线的总长不会发生变化，故不能调整调头曲线。随后，问题四又要求我们求出给定的以给定的螺线和调头曲线为轨迹的板凳龙的位置和速度，对此，我们采用和此前一致的思路，先确定龙头位置再递推龙身的位置。然而，问题四的轨迹更加复杂，调头曲线和螺线不能用统一的轨迹方程表示，需要进行分类讨论。为了计算方便，我们需要建立一个旋转后的直角坐标系来表示调头曲线上的位置，并使用旋转变换矩阵描述新坐标系和原本的直角坐标系的关系。找到分类的条件后，求出不同条件下的龙头位置随时间的表达式以及龙身位置的递推方程，即可得到板凳龙所有把手的位置。对于速度，对龙身位置的递推方程求导得到速度递推方程的方式显得过于复杂，故应直接采用差分法计算板凳龙所有把手的速度。

2.5 问题五的分析

问题五要求我们在保证板凳龙所有把手的速度均不超过2m/s，且龙头速度恒定的情况下，求出龙头允许的最大速度。这看似是一个优化问题，但其实存在更简单的解决方法。解决问题的关键在于：板凳龙各点的位置均由上一点的位置唯一确定。这说明在前点经过相同距离时，不论花费时间为多少，后点经过的距离都不会改变，即前点与后点速度成比例。又因为龙头速度确定，故龙头和龙身所有点的速度都成比例。在这种情况下，只需要设龙头速度为定值，找到出现最大速度的时刻和点，即可依靠此最大速度和允许的最大速度的比值，求得龙头允许的最大速度。

三、模型假设

1.不考虑舞龙队的人为误差，即每个板凳把手的中心准确处于给定的轨迹上。
2.板凳龙在起始时刻都盘起，即不考虑盘起之前部分龙身呈一条直线的情况。
3.两个板凳的连接处可以自由转动，不考虑卡死的情况。
4.板凳龙在即将碰撞的时刻速度不会受到碰撞的影响。
5.不考虑板凳的厚度，仅在二维空间内进行分析。

四、符号说明

符号	说明	单位
ρ_i	第i个把手在极坐标中的极径	m
θ_i	第i个把手在极坐标中的极角	rad
x_i	第i个把手的x坐标	m
y_i	第i个把手的y坐标	m
k	螺距（米）与2π的比值	m
k_b	板凳的方向斜率	-
l_h	龙头板凳的两个孔中心的距离	m
l_b	龙身板凳的两个孔中心的距离	m
l	两个孔中心的距离的统称	m

五、模型的建立与求解

5.1 问题一模型的建立与求解

5.1.1 龙头位置微分方程模型的建立

问题一要求我们提供板凳龙所有把手在不同时间点的位置和速度，这需要我们对板凳龙的运动过程进行模拟。

问题给出了板凳龙各把手的运动轨迹为螺距为55cm的等距螺线，即阿基米德螺线[1]，故以螺线的中心为极点建立极坐标系，螺线的极坐标方程为：

$$\rho = k\theta \tag{1}$$

其中，k 为螺距(单位为米) 与 2π 的比值，$k = \frac{0.55}{2\pi}$。

问题给定龙头的运动速度固定为$1m/s$，方向为沿螺线向内，表示为：

$$\frac{ds}{dt} = -1 \tag{2}$$

其中$\frac{ds}{dt}$表示点在曲线上的运动速度，由于方向向内，速度为负数。将速度$\frac{ds}{dt}$由链式法则求导进一步展开得：

$$\frac{ds}{dt} = \frac{ds}{d\theta} \cdot \frac{d\theta}{dt} \tag{3}$$

由于极坐标下的弧微元$ds = \sqrt{\rho^2 + \left(\frac{d\rho}{d\theta}\right)^2} \cdot d\theta$，且$\rho = k\theta$，有如下推导：

$$\frac{ds}{d\theta} = \sqrt{\rho^2 + \left(\frac{d\rho}{d\theta}\right)^2} = \sqrt{k^2\theta^2 + k^2} = k\sqrt{\theta^2 + 1} \tag{4}$$

结合以上四个式子，可得到如下微分方程：

$$k\sqrt{\theta^2 + 1}\frac{d\theta}{dt} = -1 \tag{5}$$

该微分方程可用分离变量法求得解析解：

$$\int k\sqrt{\theta^2 + 1}\, d\theta = \int -dt + C \tag{6}$$

$$\frac{k}{2}\left[\sqrt{\theta^2 + 1} + \ln\left(\theta + \sqrt{\theta^2 + 1}\right)\right] = -t + C \tag{7}$$

问题给出了龙头在零时刻的位置为第16圈，即初值条件。代入初值条件$\theta = 32\pi, t = 0$，即可求出积分常数 C，由此便可得到任意时刻t龙头的极角θ，确定龙头的位置。

5.1.2 龙身位置递推模型的建立

在确定了龙头在任意时刻的位置后，还需要确定龙身和龙尾在对应时刻的位置。由于题目已经给出了板凳龙前后两个把手的距离，我们可以以上一个点为圆心，画一个半径为两点间距的圆，在直角坐标系下表示为：

$$\begin{cases} (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 = l_h^2 & i = 1, \\ (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 = l_b^2 & i > 1. \end{cases} \tag{8}$$

其中l_h^2为龙头板凳两把手之间的距离，l_b^2为龙头以外的板凳两把手之间的距离，为表示方便，此后统一记作l。

从直角坐标系转换到极坐标系，需要进行如下变换：

$$\begin{cases} x = \rho \cos \theta \\ y = \rho \sin \theta \end{cases} \tag{9}$$

得到在极坐标下的圆方程：

$$\rho_{i+1}^2 + \rho_i^2 - 2\rho_{i+1}\rho_i \cos(\theta_{i+1} - \theta_i) = l^2 \tag{10}$$

由于这些点都在螺线$\rho = k\theta$上，可以进行方程联立消去ρ，得到前后两点的极角的递推方程：

$$k^2 \left(\theta_{i+1}^2 - 2\theta_i\theta_{i+1} \cos(\theta_{i+1} - \theta_i) + \theta_i^2\right) = l^2 \tag{11}$$

需要注意此方程有多个根，需要根据实际情况选用其中一个根。由于此问题中的板凳龙正在向内盘入，下一个点相对于上一个点的极角更大，故应取较大且极角大小最接近上一个点的根。由此递推方程，任意时刻的整个板凳龙的位置都可以被确定。

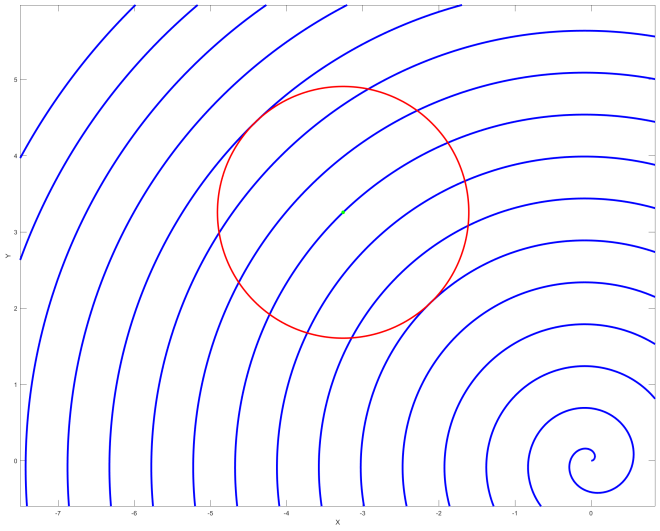


图 1: 距离圆和螺线的相交情况

5.1.3 计算龙身各点速度的两种方法

最后，还需要解决板凳龙各点的速度问题。对此有两种方法可解决该问题，首先是递推法，对于前后两点的速度，根据求导的链式法则有如下关系：

$$\frac{ds_{i+1}}{dt} = \frac{ds_{i+1}/d\theta_{i+1}}{ds_i/d\theta_i} \cdot \frac{d\theta_{i+1}}{d\theta_i} \cdot \frac{ds_i}{dt} \tag{12}$$

其中的$\frac{ds_{i+1}}{dt}$和$\frac{ds_i}{dt}$分别为前后两点的速度，$ds_i/d\theta_i$此前已经由公式(4)给出，只有$\frac{d\theta_{i+1}}{d\theta_i}$暂时未知。对此，对此前求出的递推方程变形：

$$F(\theta_{i+1},\theta_i)=k^2(\theta_{i+1}^2-2\theta_i\theta_{i+1}\cos(\theta_{i+1}-\theta_i)+\theta_i^2)-l^2=0 \tag{13}$$

根据隐函数定理，有：

$$\frac{d\theta_{i+1}}{d\theta_i}=-\frac{\partial F/\partial\theta_i}{\partial F/\partial\theta_{i+1}}=\frac{\theta_{i+1}\cos(\theta_{i+1}-\theta_i)+\theta_i\theta_{i+1}\sin(\theta_{i+1}-\theta_i)-\theta_i}{\theta_{i+1}+\theta_i\theta_{i+1}\sin(\theta_{i+1}-\theta_i)-\theta_i\cos(\theta_{i+1}-\theta_i)} \tag{14}$$

最后得到前后两点速度的递推公式为：

$$\frac{ds_{i+1}}{dt}=\frac{\sqrt{\theta_{i+1}^2+1}}{\sqrt{\theta_i^2+1}}\cdot\frac{\theta_{i+1}\cos(\theta_{i+1}-\theta_i)+\theta_i\theta_{i+1}\sin(\theta_{i+1}-\theta_i)-\theta_i}{\theta_{i+1}+\theta_i\theta_{i+1}\sin(\theta_{i+1}-\theta_i)-\theta_i\cos(\theta_{i+1}-\theta_i)}\cdot\frac{ds_i}{dt} \tag{15}$$

利用以上的递推公式，已知上一点的速度，前后两点的位置，即可得到下一点的速度，最后可得出板凳龙上所有把手的速度。

对于速度的计算，第二种方法为差分法。首先瞬时速度的定义式为：

$$v=\lim_{\Delta t\to 0}\frac{\Delta s}{\Delta t} \tag{16}$$

将Δt取一个足够小但不为零的值，由于时间很短，在此时间内走过的轨迹Δs可近似为直线，由此得到用差分法计算板凳龙上第i个点在时刻t的运动速度的公式为：

$$v_i(t)=\frac{\sqrt{(x_i(t+\Delta t)-x_i(t-\Delta t))^2+(y_i(t+\Delta t)-y_i(t-\Delta t))^2}}{2\Delta t} \tag{17}$$

5.1.4 模型的求解

将以上模型编写为MATLAB代码后进行求解，首先解出龙头在t=0s到t=300s中每秒的位置。由于龙头的极角与时间的关系是解析但非线性的，我们使用了MATLAB中的fsolve函数对方程进行求解，得到了各个时间下龙头的极角，从而得到各个时间下龙头的位置。

接下来使用递推方程确定每个时间下的龙身位置，由于递推方程同样是非线性的且具有多个根，我们在使用MATLAB中的fsolve函数时将初始值设定为上一个点的极角加一个合适的值0.15，这样可以使得出的下一个点的极角大于上一个点，且极角大小最接近上一个点，符合实际情况。

最后我们用速度的递推公式，计算出了各个时间下龙身的速度，并且我们还使用差分法再次计算了速度，两种方法得出的数据经比对，差距均小于1.2e-6，相当于验证了速度递推模型的正确性。

计算得到的部分时间的板凳龙部分把手的位置见文中表1，得到的部分时间的板凳龙部分把手的速度见文中表2，0s到300s内板凳龙每秒钟的所有把手的位置见附件result1.xlsx。

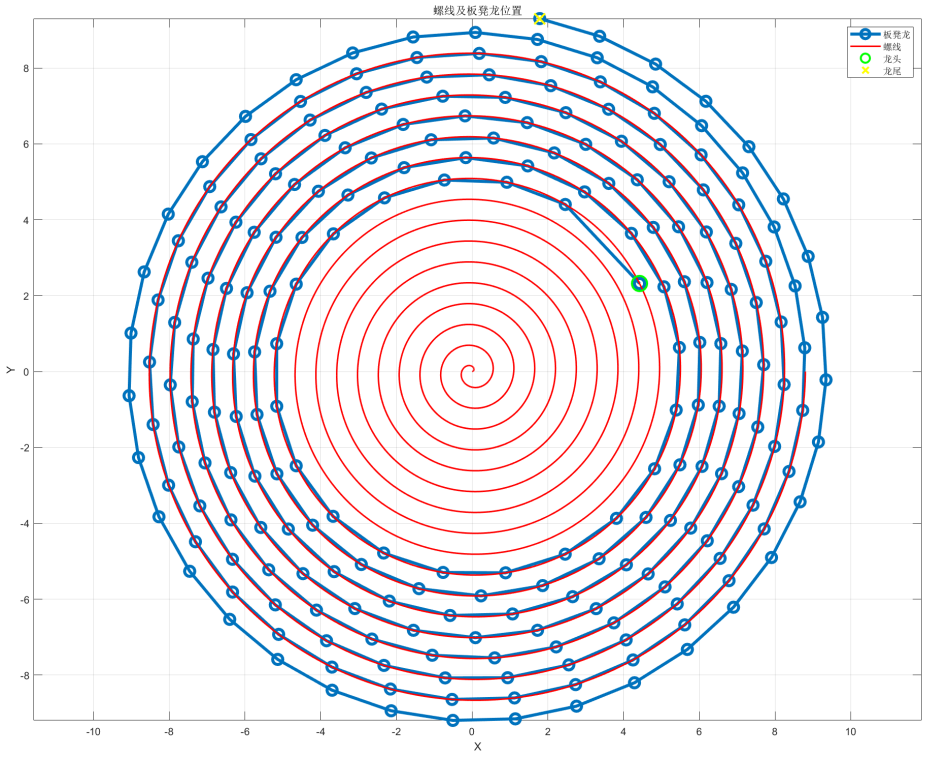


图 2: 时刻300s的板凳龙的位置图

表 1: 板凳龙部分把手的位置

	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)	8.363824	7.456758	-1.445473	-5.237118	4.821221	2.459488
第 1 节龙身 y (m)	2.826544	-3.440399	-7.405883	4.359627	-3.561949	4.402476
第 51 节龙身 x (m)	-9.518732	-8.686317	-5.543150	2.890455	5.980011	-6.301346
第 51 节龙身 y (m)	1.341137	2.540108	6.377946	7.249289	-3.827758	0.465829
第 101 节龙身 x (m)	2.913984	5.687116	5.361939	1.898794	-4.917371	-6.237722
第 101 节龙身 y (m)	-9.918311	-8.001384	-7.557638	-8.471614	-6.379874	3.936007
第 151 节龙身 x (m)	10.861726	6.682311	2.388757	1.005154	2.965378	7.040740
第 151 节龙身 y (m)	1.828754	8.134544	9.727411	9.424751	8.399721	4.393013
第 201 节龙身 x (m)	4.555102	-6.619664	-10.627211	-9.287720	-7.457151	-7.458662
第 201 节龙身 y (m)	10.725118	9.025570	1.359847	-4.246673	-6.180726	-5.263384
龙尾（后）x (m)	-5.305444	7.364557	10.974348	7.383896	3.241051	1.785032
龙尾（后）y (m)	-10.676584	-8.797992	0.843473	7.492371	9.469337	9.301164

表 2: 板凳龙部分把手的速度

	0 s	60 s	120 s	180 s	240 s	300 s
龙头 (m/s)	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
第 1 节龙身 (m/s)	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

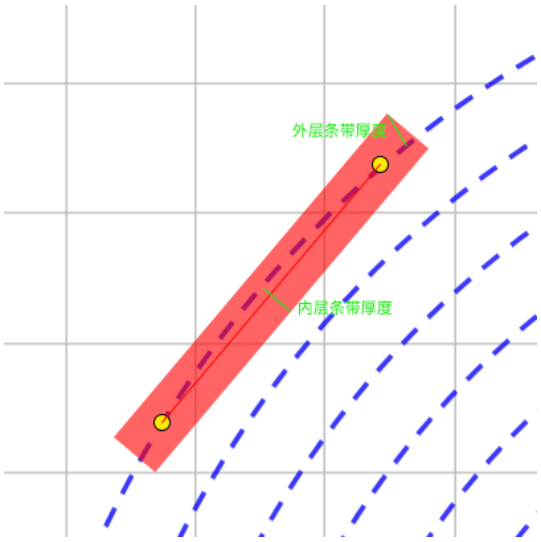
5.2 问题二模型的建立与求解

5.2.1 条带法与碰撞检测模型的说明

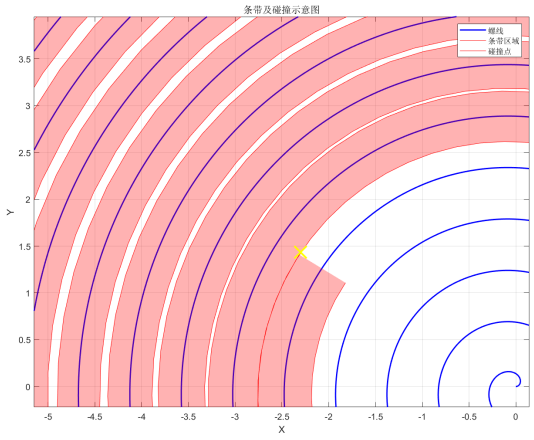
舞龙队之间发生碰撞时仅可能为第i+1层舞龙队板凳靠内侧边与位于第i层的龙头靠外的角点发生碰撞。为此，本文使用**碰撞检测模型**对龙头及其后一层的龙身进行精确的运动模拟，计算出其中每个把手的位置，并对该时刻的龙头和龙身进行碰撞检测，判断龙头角点是否进入了龙身的矩形区域内。

为求得精确碰撞时间，本文设置**碰撞检测模型**阶段使用的时间步长$dt = 0.0000001s$，且舞龙队终止时刻$t = 442.5903s$，若未知碰撞时间范围，直接使用**碰撞检测模型**，则计算量过大。

为求得舞龙队碰撞大致时间点，对模型进行简化。因舞龙队龙身几何形状相同且沿相同路径运动，故拥有一定周期性。故将单个龙身和龙头所经过区域扫过形成的**条带**作为碰撞判断区域，若相邻两层螺线上的条带区域相交则视为发生碰撞。



(a) 单个板凳示意图



(b) 条带及碰撞示意图

图 3: 条带示意图

条带代表了板凳龙所有可能经过的区域，故该方法计算出的碰撞时间为最早可能发生碰撞的时间点，基于改时间点向后采用**碰撞检测模型**进行精确计算，可以得到准确的碰撞时间，并大大减少计算量。

5.2.2 条带法模型的建立

条带法模型通过计算单个龙头及龙身在螺线上任意位置时，板凳向内偏移的距离（内层条带厚度）与板凳头向外偏移的距离（外层条带厚度），得到舞龙队在整个螺线上的条带厚度。

利用问题一中建立的龙身位置**递推模型**，给定第i块板凳的前把手的位置(ρ_i, θ_i)利用结合**递推**关系式(11)，即可计算出第i+1块板凳的前把手的位置$(\rho_{i+1}, \theta_{i+1})$，进而确定第i块板凳的位置和方向。

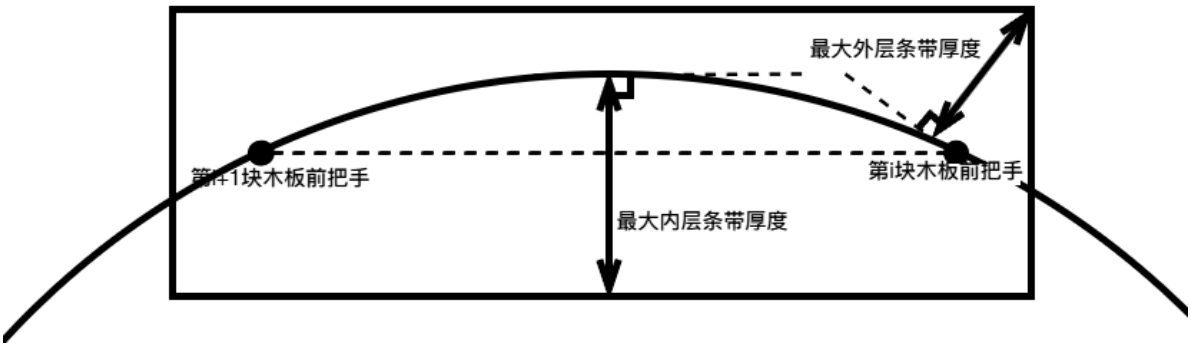


图 4: 最大条带厚度示意图

根据示意图中所示几何关系，对给定位置的板凳，其最大内层条带厚度位于螺线与板凳方向平行处，即螺线切线斜率等于前后把手间连线的斜率：

板凳所在直线解析式：

$$y - y_i = k_b(x - x_i) \tag{18}$$

$$y = k_b x - k_b x_i + y_i = k_b x - k_b k\theta_i cos\theta_i + k\theta_i sin\theta_i \tag{19}$$

其中板凳方向斜率k_b：

$$k_b = \frac{k\theta_i sin\theta_i - k\theta_{i+1} sin\theta_{i+1}}{k\theta_i cos\theta_i - k\theta_{i+1} cos\theta_{i+1}} \tag{20}$$

最大内层条带厚度处对应极角θ_{inner}：

$$\theta_{inner} = arctan(k_{inner}) = arctan(\frac{-1}{k_b}) = arctan(-\frac{k\theta_i cos\theta_i - k\theta_{i+1} cos\theta_{i+1}}{k\theta_i sin\theta_i - k\theta_{i+1} sin\theta_{i+1}}) \tag{21}$$

最大内层条带厚度d_{inner}为极角对应半径$r_{\theta_{inner}}$与板凳所在直线到原点距离d_l之差加板凳宽度w的一半：

$$d_{inner} = r_{\theta_{inner}} - d_l + \frac{w}{2} \tag{22}$$

$$d_{inner} = k\theta_{inner} - \frac{|-k_b k\theta_i cos\theta_i + k\theta_i sin\theta_i|}{\sqrt{1 + k_b^2}} + \frac{w}{2} \tag{23}$$

对于最大外层条带厚度，其为板凳前端靠外的角点(x_{corner}, y_{corner})至原点的距离减去极角对应的半径：

$$d_{outer} = d_{corner} - r_{\theta_{outer}} \tag{24}$$

$$d_{outer} = \sqrt{x_{corner}^2 + y_{corner}^2} - karctan(\frac{y_{corner}}{x_{corner}}) \tag{25}$$

5.2.3 条带法模型的求解

条带法模型求解算法流程图：

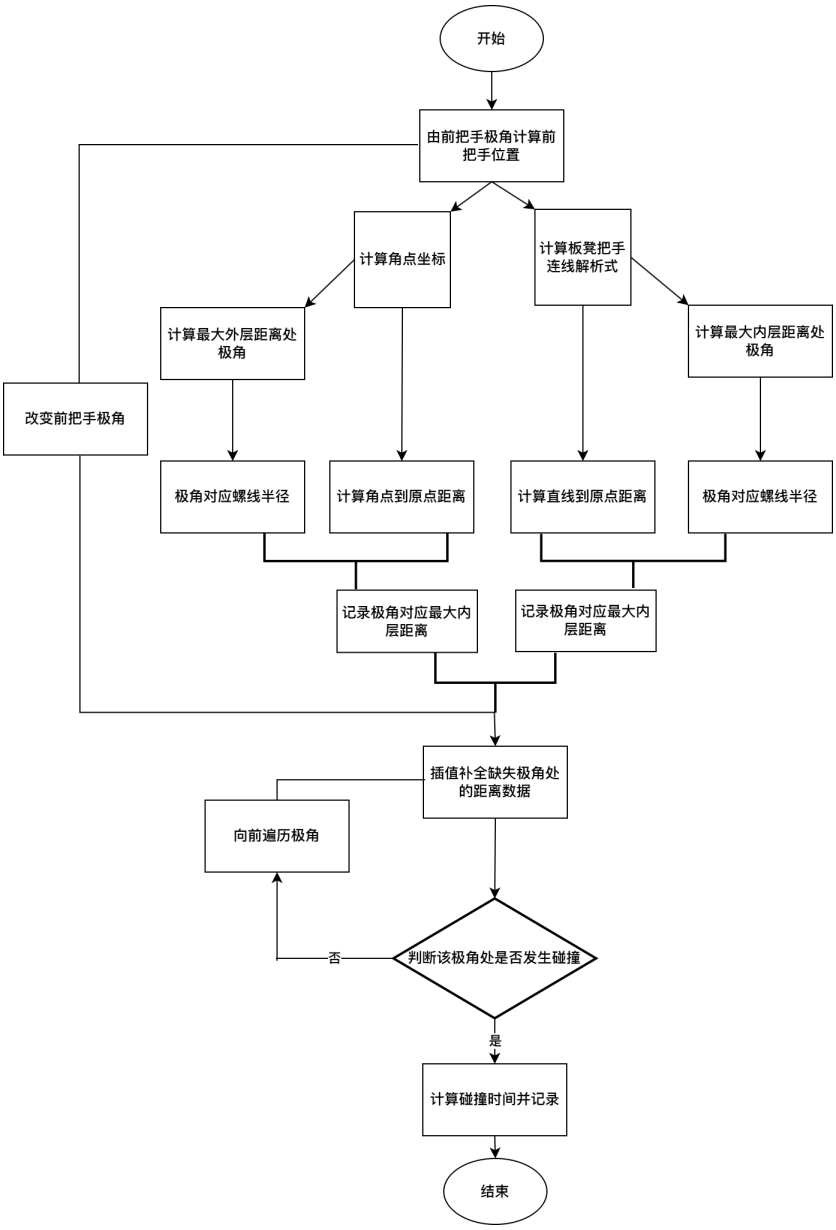


图 5: 条带法模型流程图

为减小计算量，计算时设置前把手极角位置范围为 $(0, 100rad)$（约16圈），步长 $0.1rad$，碰撞时间遍历范围为 $(235.6822s, 442.5888s)$。

若某极角缺失对应厚度数据，使用MATLAB内置的线性插值进行补全。

条带法模型计算结果为：

碰撞时刻：408.7694s

碰撞位置对应极角：27.7168rad

计算得到的结果及碰撞点如图：

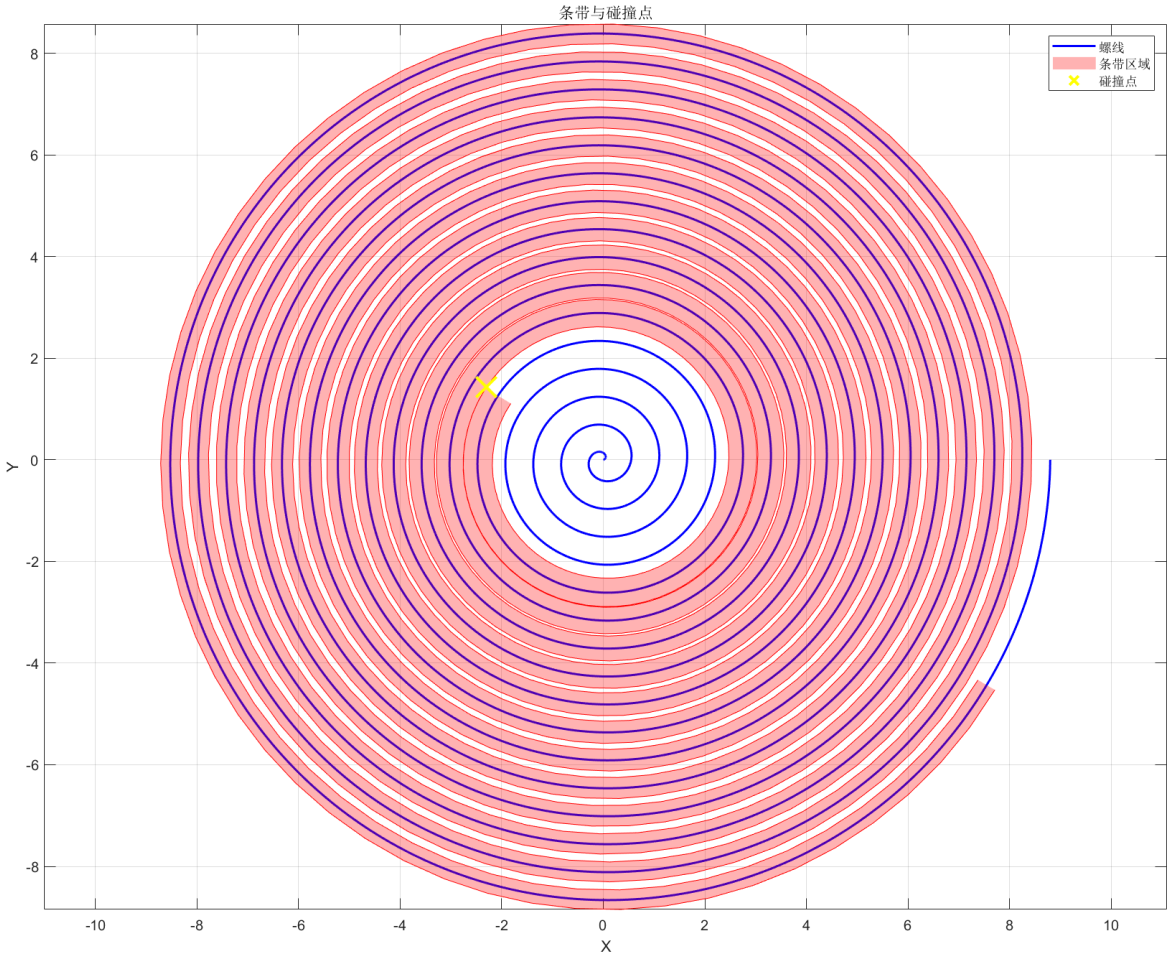


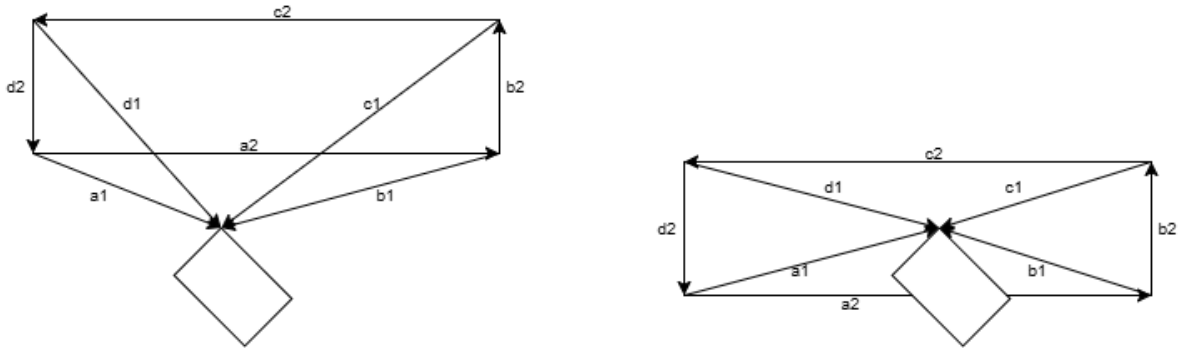
图 6: 结果及碰撞点

利用该简化模型计算出的碰撞时刻作为起点，使用**碰撞检测模型**向该时间点后进行遍历求精确解。

5.2.4 碰撞检测模型的建立

针对问题二，可以考虑使用**叉积法**判断是否发生碰撞。根据螺线模型，可以显然地判断出碰撞只可能发生在**龙头与相邻外层螺线的龙身之间**，因此只需要计算在板凳龙盘入的某一时刻，龙头矩形的前端角点位于龙身矩形的范围内即可。

对于两个向量$\mathbf{a}(ax, ay), \mathbf{b}(bx, by)$而言，二者的叉积$\mathbf{a}\times\mathbf{b} = ax\cdot by - ay\cdot bx$。结果的符号表示b向量相对于a向量的方向: **1.正值：b在a的逆时针方向； 2.负值：b在a的顺时针方向； 3.零：a和b共线。** 那么利用这些性质可以判断出点是否在矩形内，选择逆时针设置矩形边向量如图7a，只需要计算$\mathbf{a_2}\times\mathbf{a_1}, \mathbf{b_2}\times\mathbf{b_1}, \mathbf{c_2}\times\mathbf{c_1}, \mathbf{d_2}\times\mathbf{d_1}$的结果是否都同号即可，由图7b可知，当龙头矩形的角点位于龙身矩形内时，所有的龙身矩形角点到龙头角点的向量都在相对应的边向量的左边，即$\mathbf{a_1}, \mathbf{b_1}, \mathbf{c_1}, \mathbf{d_1}$分别在$\mathbf{a_2}, \mathbf{b_2}, \mathbf{c_2}, \mathbf{d_2}$的逆时针方向，因此当叉积的值都大于0的时候，龙头的角点位于龙身矩形范围内，此时发生碰撞。



(a) 边向量与龙身角点到龙头角点向量示意图　　(b) 龙头角点位于龙身矩形内示意图

图 7: 叉积示意图

针对矩形的角点，我们可以通过**斜率方程，几何关系和题目所给龙头、龙身和龙尾的俯视图模型**进行位置计算。如图8，根据螺线模型我们可以得到龙头前把手和第一条龙身的前把手的位置，由此我们可以计算出龙头的斜率$\theta = \arctan\frac{y_1-y_2}{x_1-x_2}$，然后我们计算出C点位置$y_C = y_1 + 0.15\cdot\sin(90-\theta), x_C = x_1 - 0.15\cdot\cos(90-\theta)$，最后我们可以计算出角点D的位置$y_D = y_C + 0.275\cdot\sin(\theta), x_D = x_C + 0.275\cdot\cos(\theta)$，同理我们可以求出E点的位置，以及龙身各点的位置，以此我们可以将位置信息放入上述**叉积法**中，将龙头角点位置信息与各个龙身的角点位置信息遍历计算，判断龙头是否与龙身相碰。

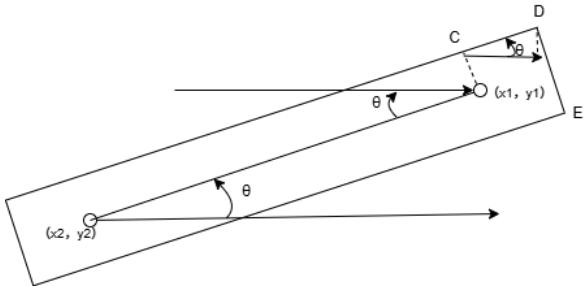


图 8: 龙头角点位置示意图

5.2.5 碰撞检测模型的求解

我们通过**迭代法**，对碰撞终止时刻和各个部位的位置与速度进行更精确化地计算，使其符合题目的精确到6位小数。如图9所示。

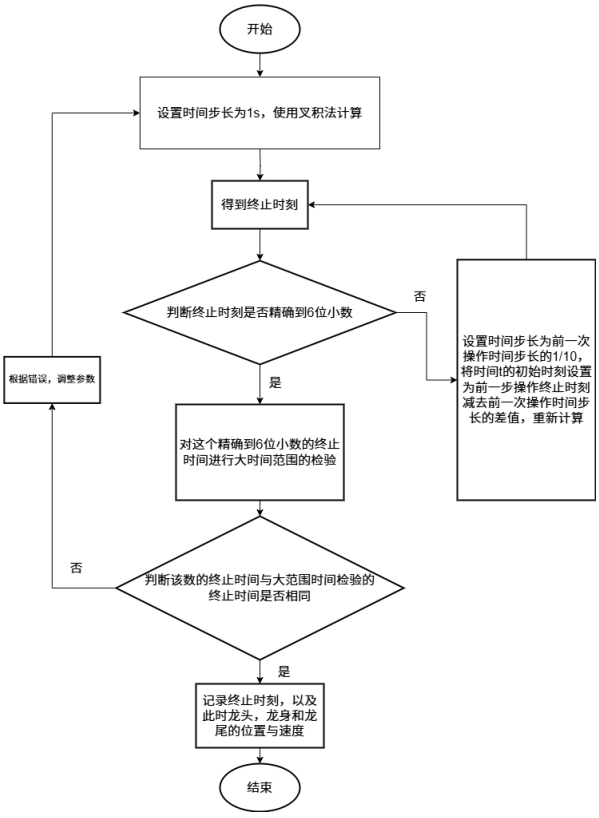


图 9: 迭代法计算流程图

最终得到的位置和速度结果如表3和表4所示。图10是碰撞时刻板凳龙所有节点所在的位置。

表 3: 终止时刻板凳龙部分把手的位置

	412.473838s
龙头 x (m)	1.209931
龙头 y (m)	1.942784
第 1 节龙身 x (m)	-1.643792
第 1 节龙身 y (m)	1.753399
第 51 节龙身 x (m)	1.281201
第 51 节龙身 y (m)	4.326588
第 101 节龙身 x (m)	-0.536246
第 101 节龙身 y (m)	-5.880138
第 151 节龙身 x (m)	0.96884
第 151 节龙身 y (m)	-6.957479
第 201 节龙身 x (m)	-7.893161
第 201 节龙身 y (m)	-1.230764
龙尾（后）x (m)	0.956217
龙尾（后）y (m)	8.322736

表 4: 终止时刻板凳龙部分把手的速度

	412.473838s
龙头 (m/s)	1.000000
第 1 节龙身 (m/s)	0.991551
第 51 节龙身 (m/s)	0.976858
第 101 节龙身 (m/s)	0.974550
第 151 节龙身 (m/s)	0.973608
第 201 节龙身 (m/s)	0.973096
龙尾（后）(m/s)	0.972938

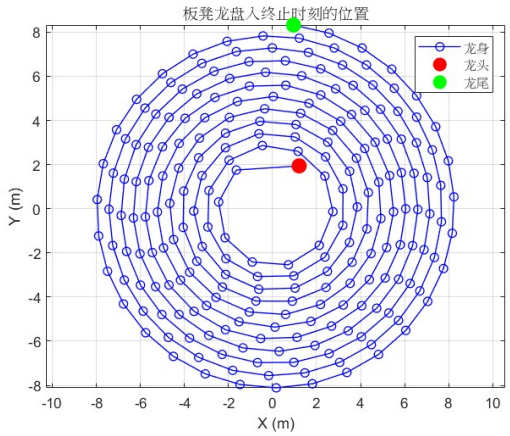


图 10: 碰撞终止时刻板凳龙的位置

5.3 问题三模型的建立与求解

5.3.1 模型说明

我们使用类似第二问中的方法，即先用**条带法**确定板凳龙在切点处恰好发生碰撞时螺距的大小，并在该螺距大小附近使用严格模拟的**碰撞检测**的方法进行搜索，找出符合要求的最小螺距大小。

5.3.2 条带法模型求解

条带法模型求解算法流程图：

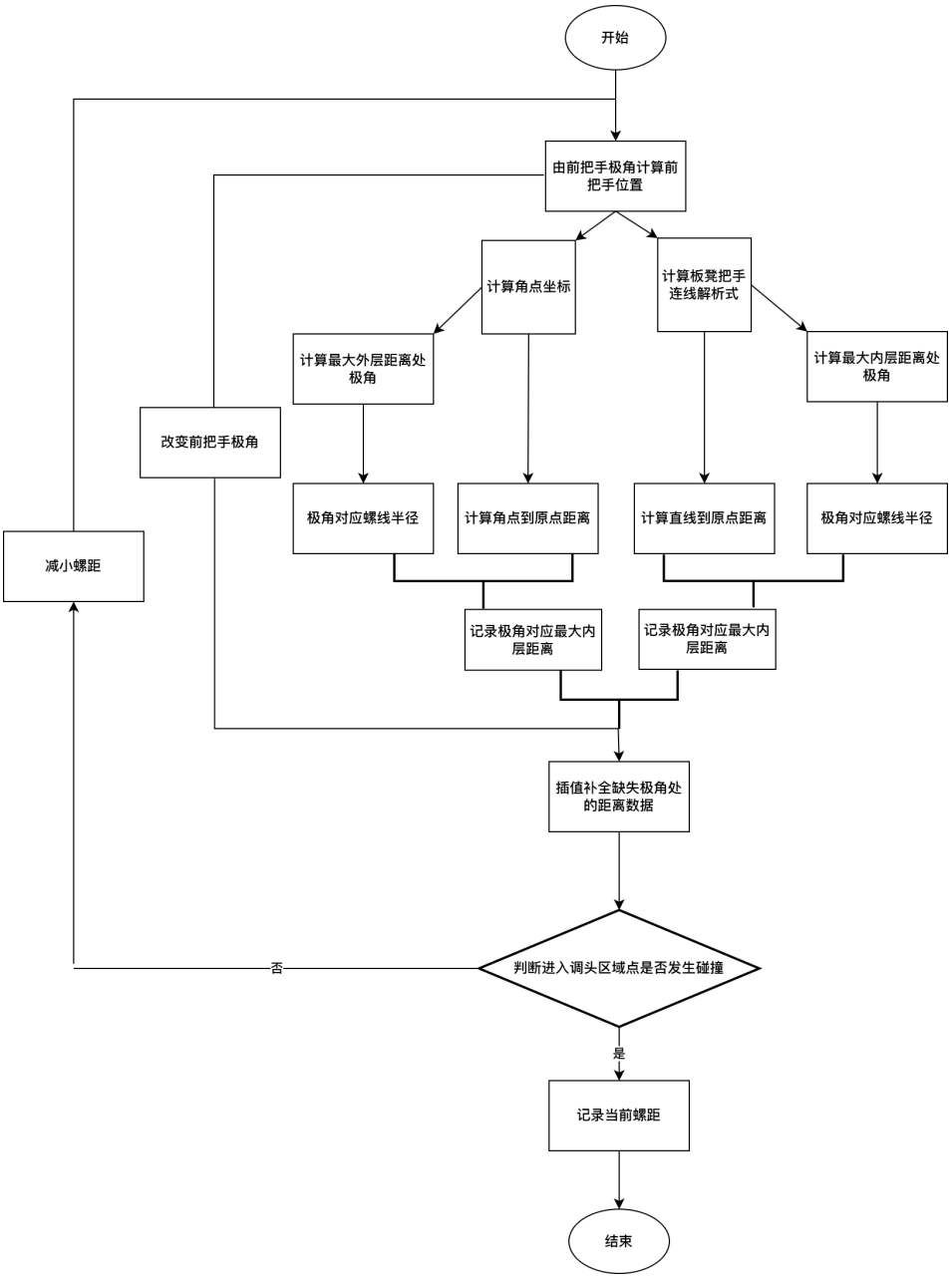


图 11: 条带法流程图

为减小计算量，计算时设置前把手极角位置范围为 $(0, 60rad)$，设置螺距遍历范围$(0, 0.55m)$，步长设置为0.001m。

因条带宽度随极角减小而单调递减，若在进入调头区域点（螺线与调头区域的切点）处不发生碰撞，则不可能在其之前的螺线处发生碰撞。因此只需要对进入调头区域点进行碰撞检测即可判断该螺距下舞龙队是否发生碰撞。

若某极角缺失对应厚度数据，使用MATLAB内置的线性插值进行补全。

条带法模型计算结果为：

最小螺距：0.4540m

计算得到的结果如图：

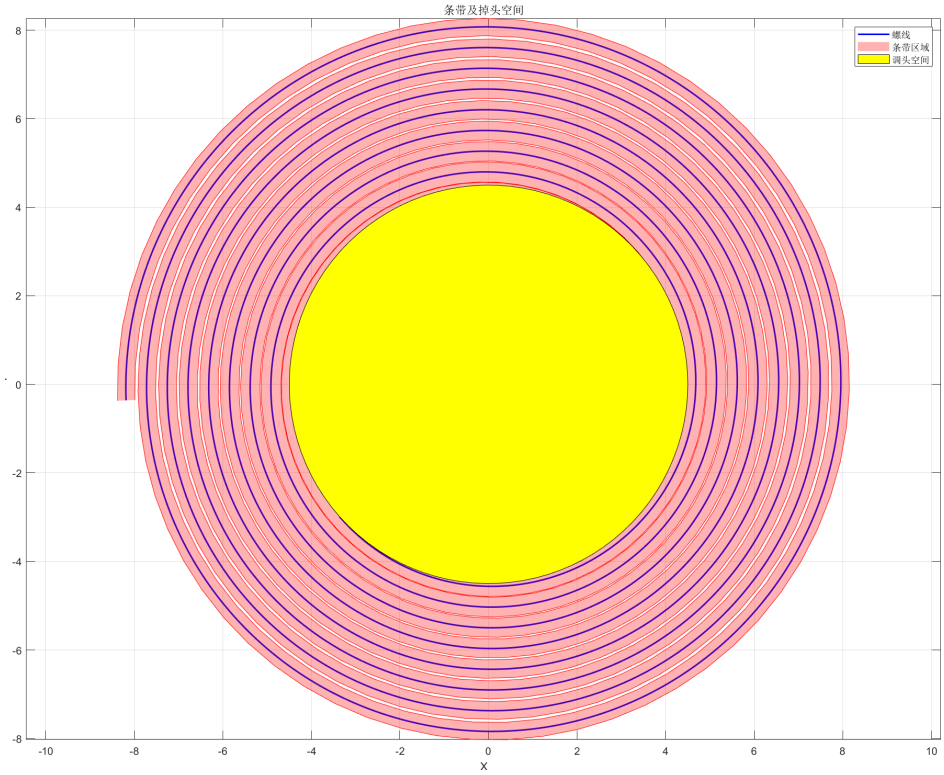


图 12: 条带及调头空间

利用该简化模型计算出的最小螺距作为起点，使用**碰撞检测模型**逐步减少螺距进行精确求解。

5.3.3 碰撞检测模型的求解

根据**条带模型**得到了最小螺距在0.454m附近之后，我们继续使用**叉积法**对最小螺距进行更为精确地求解。经过多次实验，我们发现将0.4499m的螺距代入**螺线模型**，龙头的碰撞终止位置距离原点4.62m，将 0.4498m的螺距代入模型，龙头的碰撞终止位置距离原点为4.40m，因此我们在0.4498m到0.4499m 范围内使用步长为0.00001的**迭代法**去进行计算，最终我们得到在螺距为0.449810m的时候，龙头的碰撞终止位置距离原点为4.44m，之后龙头的碰撞终止位置就会有很大的变化，与原点的距离会超过0.45m，因此我们选择使用0.449810m作为最小螺距。

5.4 问题四模型的建立与求解

5.4.1 调头曲线的调整

题目给定调头空间的直径为9m，盘入螺线与盘出螺线中心对称，即进入调头空间的点与离开调头空间的点的连线是经过坐标原点的。在此情况下题目还要求调头曲线为两段圆弧，圆弧之间

相切，且圆弧与螺线也相切。

以上约束条件在下图中表示：

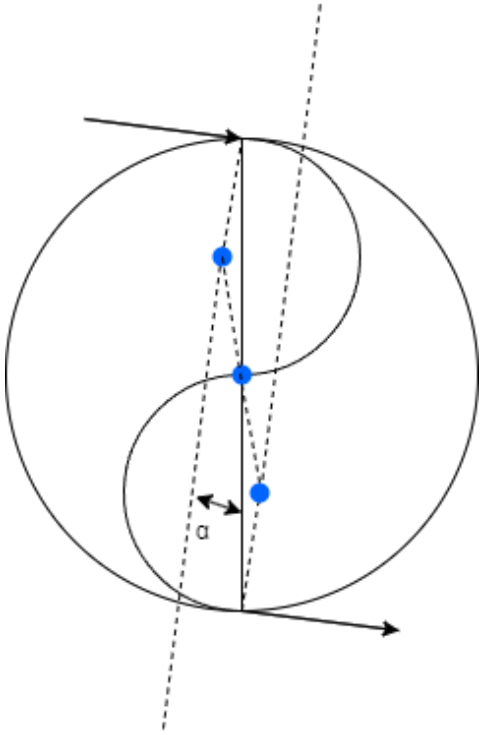


图 13: 调头曲线满足约束条件的一种情况

其中角度α为定值，由螺线的参数可以确定，为满足约束条件，两段圆弧的圆心必须处在与调头区域大圆的中间的直径夹角为α的虚线上，且若两段圆弧需要相切，切点必须在大圆的直径上，故若设两圆弧半径分别为r_1，r_2，大圆半径为R，则有以下关系：

$$(r_1 + r_2)\, cos\alpha = R \tag{26}$$

整段调头曲线的长度为：

$$(\pi + 2\alpha)\,(r_1 + r_2) \tag{27}$$

由于R和α均为定值，调整圆弧只能调整r_1和r_2，但根据上述关系式，无论怎么调整，最终长度都不变，**故无法调整圆弧使得调头曲线变短。**

5.4.2 求解位置时调头曲线的建模和分类讨论

由于题目中所给的情况下，夹角α很小，经计算仅有$0.06rad$，因此我们在此将调头曲线进行简化，认为调头曲线的圆心在调头区域的圆的直径上，以方便接下来的建模和计算。

首先需要确定龙头的位置，对此，需要将整条轨迹分为四段：

case1.位于盘入螺线

case2.位于曲线大圆弧

case3.位于曲线小圆弧

case4.位于盘出螺线

其中分类的时间条件如下：

$$\begin{cases} t < 0 & case1, \\ 0 \le t < 3\pi/v & case2, \\ 3\pi/v \le t < 4.5\pi/v & case3, \\ t \ge 4.5\pi/v & case4. \end{cases} \tag{28}$$

这里的v为龙头的运动速度，单位为m/s。

在不同的条件下，我们对龙头的位置进行不同的计算。首先，对于盘入螺线上的情况，可以沿用问题一中的微分方程的解(7)。另外，由于盘入螺线与盘出螺线是中心对称的，且龙头速度大小不变，故任何时刻的盘出螺线上的位置可以由对应时刻的盘入螺线上的位置经中心对称得到。

对于在调头曲线上的情况，则需要建立新的坐标系进行表示，这样方便进行计算。以螺线中心为原点，连接进入调头空间和离开调头空间的调头空间圆的直径方向为x'轴方向，与其垂直的方向为y'轴方向建立旋转后的直角坐标系，如下图所示：

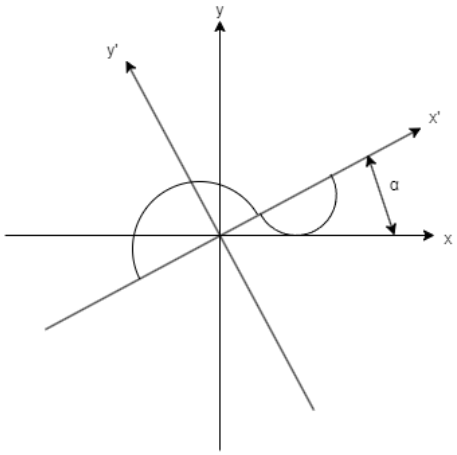


图 14: 旋转后的坐标系示意图

其与原先坐标系的变换关系为：

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} \tag{29}$$

在问题条件下，此处的旋转角度$\alpha = \pi/3.4$，在新的坐标系下，龙头只是在圆上匀速运动，有了圆心和半径，我们很容易得到龙头的位置：

$$\begin{cases} x' = -1.5 - 3cos\,(vt/3)\,, y' = 3sin\,(vt/3) & 0 \le t < 3\pi/v, \\ x' = 3 - 1.5cos\,(vt/6)\,, y' = -1.5sin\,(vt/6) & 3\pi/v \le t < 4.5\pi/v. \end{cases} \tag{30}$$

只需将得到的坐标向量左乘旋转矩阵，即可还原回到原先的坐标系。至此，所有时间下的龙头位置都已经能够确定。

接下来需要找出龙身上各个点的位置，这同样需要分类讨论，不同类别下所需要联立的轨迹方程由不同。对此，以下一个点处在轨迹的哪一段为依据，我们同样分出了四种情况：

1.下一点在盘出螺线

2.在盘入螺线

3.在调头曲线大圆弧和

4.在调头曲线小圆弧

由于是递推方程，我们以上一个点的位置作为分类条件。首先考虑最简单的情况，上一个点在盘入螺线上，则下一点必然在盘出螺线上。

然后，若上一个点在盘出螺线上，此时有两种可能，一是下一个点仍在盘出螺线上，二是下一个点在调头区域内。

最后，对于上一个点处在调头区域内的最复杂的情况，此处有三种可能，一是下一点在调头曲线小圆弧上，二是下一个点在调头曲线大圆弧上，三是下一个点在盘入螺线上。对于这些情况，需要找到合适的判断依据。

对于上一点在盘出螺线上的情况，需要找到下一点是否在调头区域的判断方法。在这里可以先假设下一点仍在螺线上，使用此前得到的螺线上板凳龙的前后把手位置的递推方程(11)得到下一点。若下一点不在调头区域，则通过假设得到的位置就是下一点的位置。若下一点的位置在调头区域，则下一点一定在调头曲线小圆弧上的另一处，需要进一步计算，计算公式如下：

$$\left[(2x_0' - 6)^2 + 4y_0'^2\right]x'^2 + \left[(4x_0' - 12)(l^2 - x_0'^2 - y_0'^2 + 6.75) - 24y_0'^2\right]x' + 27y_0'^2 + (l^2 - x_0'^2 - y_0'^2 + 6.75)^2 = 0 \tag{31}$$

$$y' = -\sqrt{-x'^2 + 6x' - 6.75} \tag{32}$$

其中，x_0'和y_0'为上一点在旋转后的坐标系中的坐标，l为两点间距即板凳上两把手间距，此方程是由上一点的距离圆与调头曲线的小圆弧的方程联立得到的二次方程，有两根，取位置合适的根x'，代回到小圆弧的方程求出y'，再由坐标变换即可得到下一点的位置。

上一点在调头区域内的情况最为复杂，需要找到几个临界条件，临界条件如下图所示：

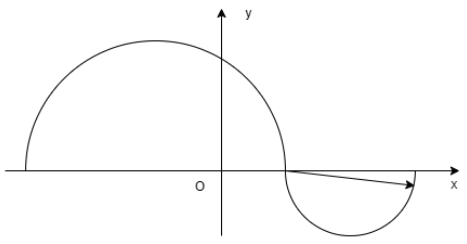


图 15: 下一个点在小圆弧和大圆弧之间的临界条件，箭头为上一个点

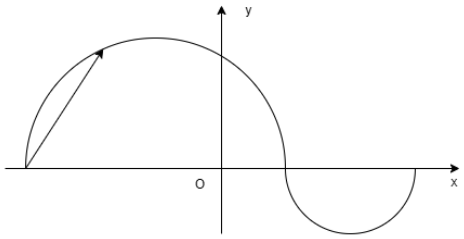


图 16: 下一个点在螺线和大圆弧之间的临界条件，箭头为上一个点

接下来需要求出上述图中临界条件发生时上一点的x'坐标，如下图所示

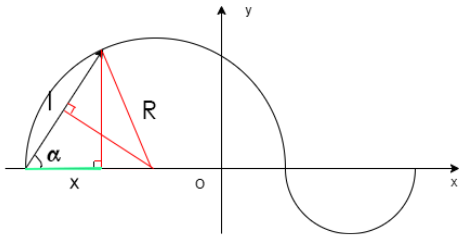


图 17: 临界位置的推导示意图

根据图中的几何关系，可以得到图中绿色线段的长度x，其中l为前后点间距，R为半径

$$x = l\cos\alpha = l \cdot \frac{l}{2R} = \frac{l^2}{2R} \tag{33}$$

由此即可得到临界条件时的前点的坐标。对于另外一个临界条件，以上公式仍适用。在计算得到临界条件后，我们最终可以确定在调头区域内的情况的三种分类：

case1.下一点在小圆弧上

case2.下一点在大圆弧上

case3.下一点在盘入螺线上

$$\begin{cases} 1.5 + \dfrac{l^2}{3} < x' < 4.5 & case1, \\ -4.5 + \dfrac{l^2}{6} < x' \le 1.5 + \dfrac{l^2}{3}x' & case2, \\ -4.5 \le x' \le -4.5 + \dfrac{l^2}{6} & case3, \end{cases} \tag{34}$$

对于第一种分类，由于需要联立的方程还是小圆弧的方程，求前点坐标的公式可沿用之前得到的公式(31)(32)。

对于第二种分类，需要联立的方程是大圆弧的方程，联立后求坐标的公式如下：

$$\left[(2x_0' + 3)^2 + 4y_0'^2\right]x'^2 + \left[12y_0'^2 - (4x_0 + 6)(6.75 + x_0'^2 + y_0'^2 - l^2)\right]x' + (6.75 + x_0'^2 + y_0'^2 - l^2)^2 - 27y_0'^2 = 0 \tag{35}$$

$$y' = \sqrt{-x'^2 - 3x' + 6.75} \tag{36}$$

最后对于第三种分类，后点在螺线上，只需将前点坐标转化成极坐标（注意极角范围此时为4π到6π），使用之前得到得螺线上的位置递推方程即可得到后点位置。

5.4.3 差分法计算各点速度

如果继续使用递推法求解各点的速度，因为分类过多，加之联立后的方程复杂且涉及坐标变换，推导显得过于复杂。在这里，我们放弃了解析的速度递推式，采用差分法来计算所有点的速度。差分法的原理在问题一中已经叙述，公式为(17)

5.4.4 模型的求解

我们将题目给出的条件和以上建模得到的坐标系旋转矩阵，不同情况下的分类条件，联立方程得到的坐标求解方程，以及差分方程输入了MATLAB软件进行求解。

其中，对于问题一中得到的螺线上的板凳龙前后点位置的非线性递推方程使用fsolve函数求解。此处需要特别注意点是处于盘入状态还是盘出状态，因为盘入时下一点的极角大于上一点，但盘出时是相反的，需要特别注意fsolve的初值，才能得到正确结果。我们在MATLAB代码中加入了与盘入盘出状态检测有关的代码，在不同的情况下采用不同的初值，可以得到正确的解。

对于问题四中得到的在调头空间上使用的位置递推方程，由于这是一个二次方程，我们可以直接由MATLAB的solve函数得到解析解。然而这里同样需要注意，可能会产生两个根，需要进行检验以保证选择的根是正确的。根的检验涉及两步，首先检验求出的下一个点到上一个点的距离，距离不符合的结果予以排除。接下来取所有根中x'值更小的一个，因为后点x'坐标必然小于前点。进行检验后即可保证求解正确。

以下为模型求解得到的，某个时刻的整个板凳龙位置的示意图，圆点位置为板凳把手位置：

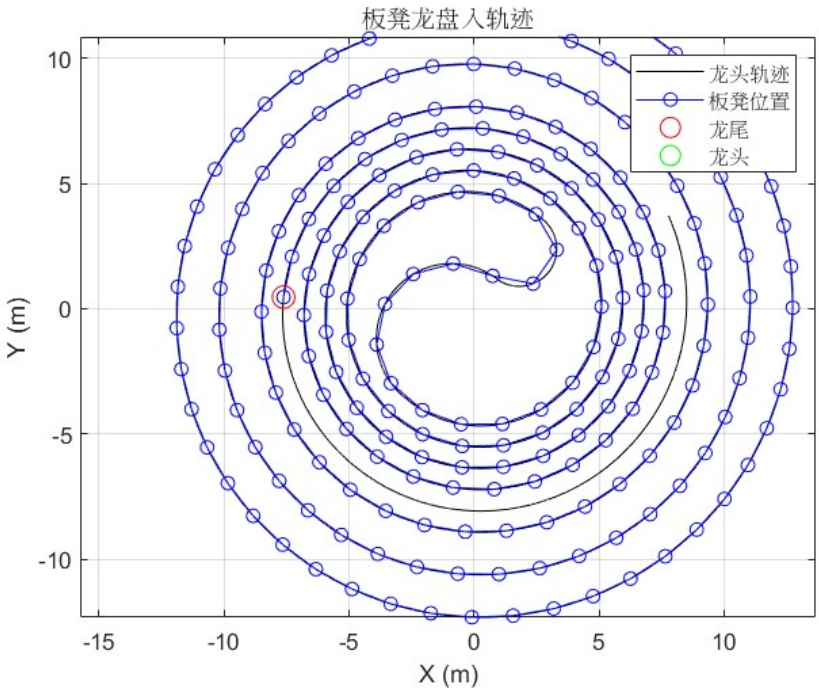


图 18: 某时刻的板凳龙位置

最后我们使用差分法计算了板凳龙上各点的速度，设置差分法的时间$\Delta t = 0.00001s$。

部分时刻板凳龙上部分把手的位置如下表所示，从-100s到100s的每一秒的板凳龙上所有把手的位置和速度见文件**result4.xlsx**

表 5: 板凳龙部分把手的位置

	-100 s	-50 s	0 s	50 s	100 s
龙头x (m)	7.778034	6.608301	-2.711856	1.826002	-2.669481
龙头y (m)	3.717164	1.898865	-3.591078	6.024740	7.716408
第1节龙身x (m)	6.209273	5.366911	-0.063534	4.236659	0.168878
第1节龙身y (m)	6.108521	4.475403	-4.670888	4.485801	8.067570
第51节龙身x (m)	-10.608038	-3.629945	2.459962	-1.695558	2.523569
第51节龙身y (m)	2.831491	-8.963800	-7.778145	-6.068881	3.743315
第101节龙身x (m)	-11.922761	10.125787	3.008493	-7.576885	-7.293817
第101节龙身y (m)	-4.802378	-5.972246	10.108539	5.196000	2.048162
第151节龙身x (m)	-14.351032	12.974784	-7.002789	-4.628084	9.468690
第151节龙身y (m)	-1.980993	-3.810357	10.337482	-10.376136	-3.525069
第201节龙身x (m)	-11.952942	10.522509	-6.872842	0.311599	8.512917
第201节龙身y (m)	10.566998	-10.807425	12.382609	-13.177713	8.618782
龙尾（后）x (m)	-1.011059	0.189809	-1.933627	5.881864	-10.971794
龙尾（后）y (m)	-16.527573	15.720588	-14.713128	12.601748	-6.784208

表 6: 板凳龙部分把手的速度

	-100 s	-50 s	0 s	50 s	100 s
龙头 (m/s)	1.000000	1.000000	1.002704	1.000000	1.000000
第1节龙身 (m/s)	0.999904	0.999762	1.010864	1.000368	1.000125
第51节龙身 (m/s)	0.999348	0.998641	1.007268	0.904333	1.004037
第101节龙身 (m/s)	0.999092	0.998248	1.006574	0.902950	0.959922
第151节龙身 (m/s)	0.998945	0.998045	1.006279	0.902527	0.959086
第201节龙身 (m/s)	0.998850	0.997923	1.006117	0.902321	0.958760
龙尾（后） (m/s)	0.998818	0.997883	1.006066	0.902261	0.958673

5.5 问题五模型的建立与求解

5.5.1 龙身各点的速度成比例

问题五要求我们在保证板凳龙所有把手的速度不超过2m/s，且龙头速度恒定的情况下，求出龙头允许的最大速度。这看似是优化问题，但其实存在更简单的方法。

我们现在取其中一张板凳进行分析，示意图如下：

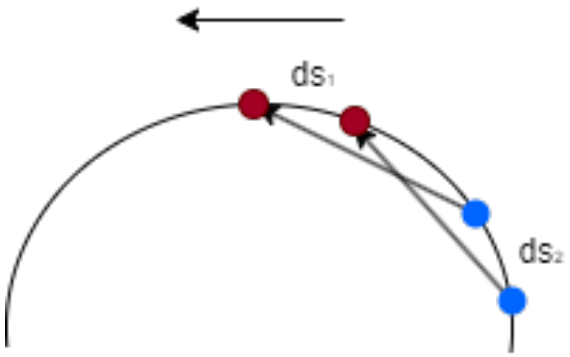


图 19: 速度成比例示意图

板凳前点在时间微元dt内走过了弧长ds_1，与此同时板凳后点在时间微元dt内走过了弧长ds_2，两点的速度分别为$\frac{ds_1}{dt}$和$\frac{ds_2}{dt}$。现在将前点走过弧长ds_1的时长乘以n（n为任意正实数），则前点的速度变为$\frac{ds_1}{n \cdot dt}$，即原来的1/n倍。由于轨迹已经确定，不论n取何值，前点走过弧长ds_1时，后点必定走过弧长ds_2，其速度为$\frac{ds_1}{n \cdot dt}$，也为原本的1/n倍。

因此可以得出结论，板凳龙前一点和后一点的移动速度一定成比例，通过递推，可以推广为板凳龙上任意一点的速度都与龙头的速度成比例。

5.5.2 利用比例关系求得最大速度

既然龙身上任何一点的速度都与龙头的速度成比例，我们可以采用如下方法来确定龙头允许的最大速度。首先，在龙身速度固定为1m/s时计算出任何时刻下龙身上所有点的速度。然后，找出这些速度中的最大值。最后，用允许的速度2m/s除以该最大速度，得到一比值，用该比值乘以假设的龙头速度1m/s，即可得到龙头的最大允许速度。

5.5.3 模型的求解

利用问题四的MATLAB代码，可以得到-100s到100s内每一秒板凳龙所有把手的速度。然而问题四的时间间隔为1s，无法精确求出最大速度。对此我们采用了变步长的遍历求解法，先粗步确定最大速度出现的时间点，再逐步缩小搜索的时间范围，减小时间步长。最后，在时间步长为0.001s时，得到最大速度出现在t=14.855s时的第六个节点，由此计算出龙头的最大允许速度为v=1.23246201969456m/s。

六、模型的分析与检验

6.1 针对碰撞终止的灵敏度分析

根据问题二的条件，我们可以改变螺距，然后使用**条带模型**和**螺线模型叉积法**得到碰撞终止时刻，分析各个终止时刻之间的差异。我们分别对0.45m,0.50m,0.55m,0.60m,0.65m的螺距进行分析，通过**条带模型**，我们得到的不同螺距的终止时刻如下表7，而使用**螺线模型叉积法**我们得到的不同螺距的终止时刻如下表8。

表 7: 条带模型下计算的不同螺距的终止时刻

螺距（m）	终止时刻（s）
0.45	214.2548
0.50	335.5459
0.55	408.7694
0.60	465.1788
0.65	520.8400

表 8: 螺线模型叉积法下计算的不同螺距的终止时刻

螺距（m）	终止时刻（s）
0.45	226.337228
0.50	342.346765
0.55	412.473838
0.60	466.508653
0.65	517.026286

根据表格数据我们可以看出，在两个不同模型下计算出来的终止时刻十分相近，但是条带模型由于模型特性得到的结果精确性较低，而使用螺线模型叉积法可以得到准确到6位小数的结果，因此在获取精确数据的时候我们可以选择**螺线模型叉积法**，在获取大致数据时选择**条带模型**。两者模型数据的比对可以得出模型对不同的螺距具有一定灵敏度，可以正确地分析出不同螺距下的碰撞终止时刻。

6.2 针对螺线模型叉积法的误差分析

螺线模型叉积法涉及到角度的计算，如果龙头，龙身和龙尾的位置存在一定的误差的话，那么角度可能会存在一部分偏差，但是这个误差是在可接受范围内的，因为该模型的时间步长设置为$1 \times 10^{-7}s$，则时间上的误差在$1 \times 10^{-7}s$内，而我们的龙头的速度为$1m/s$，因此龙头，龙身和龙尾的位置误差也应该在$1 \times 10^{-7}m$内，误差非常小，符合题目的要求。

6.3 针对问题四中的简化和差分法带来的的误差分析

因为在对问题四的模型进行简化时，认为调头空间内的圆心都在调头空间的圆的直线上，调头空间内的轨迹存在一定误差，但此误差较小，为了简化模型，可以接受这种程度的误差。

另外，差分法计算速度也会带来一定的误差，对于在螺线上的点的速度误差，在问题一中已经进行了验证，误差数量级在1e-6。对于其它处，尤其是在不同段的轨迹连接处，差分法可能会带来较为明显的误差。例如上表中时间为0s时龙头的速度为1.002704s。然而龙头的速度应固定为1m/s，误差数量级在1e-3。对于其它正好经过轨迹连接点的把手的速度，类似的误差应该也会出现。

七、模型的评价、改进与推广

7.1 模型的优点

1.用微分方程求出了龙头位置与时间的解析的关系式，也求出了龙身位置的准确的迭代方程，计算速度快，开销较小。

2.条带模型计算的速度非常高效，螺线模型叉积法的精确度很高，两个模型充分结合，可以既快速又高效地计算出结果。

3.条带模型和螺线模型叉积法都具有一定稳定性，可以很好地适用于不同的螺距。

7.2 模型的缺点

1.问题四的简化带来的误差实际上虽然很小，但相对于本文中其它模型的误差较大。

2.实验过程没有加入随机噪声，因此无法完全适用于现实情况。

7.3 模型的改进

1.可以对问题四的轨迹进行更加准确的建模，并求出所有的速度递推关系式，替代差分法。

2.可以在之后模型的运动模拟中加入随机噪声，使其更符合现实的情况。

7.4 模型的推广

该模型可以在确定轨迹的条件下，应用于相互连接可旋转的刚体的位置与速度预测，例如规划具有观赏性的板凳龙轨迹，预测轨道上的火车运动，模拟未来太空电梯的连接体运动。

参考文献

[1] 阿基米德螺线. 新世纪智能, (45):49, 2024.

附录

附录**1**：支撑材料的文件列表
result1.xlsx
result2.xlsx
result4.xlsx
q1.m
q2.m
q3.m
q4_q5.m
A2.m
A3.m
lingmindu.m
lingmindu1.m
lingmindu2.m
lingmindu3.m
lingmindu4.m
lingmindu5.m

附录2: 问题1代码(q1.m)

```
clc
clear
format long

% 初始化参数
k = 0.55/(2*pi);   % 螺线参数
v_h = 1;           % 龙头速度 (m/s)
theta_s = 32*pi;   % 初始角度
len_h = 2.86;      % 龙头长度 (m)
len_b = 1.65;      % 龙身长度 (m)

% 初始化结果数组
ans1 = zeros(448,301); % 位置结果（和坐标）xy
ans2 = zeros(224,301); % 速度结果

% 主循环：计算到秒的位置和速度0300
for t = 0:300
theta = zeros(223,1); % 初始化角度数组
pos = zeros(224,2); % 初始化位置数组
v = zeros(224,1);   % 初始化速度数组

% 计算总弧长
tot_s = k/2*(theta_s*(1+theta_s^2)^0.5+log(theta_s+(1+theta_s^2)^0.5));

% 计算龙头角度
func_sh = @(theta)
    k/2*(theta*(1+theta^2)^0.5+log(theta+(1+theta^2)^0.5)) - tot_s +
    v_h*t;
theta_h = fsolve(func_sh, 0);

% 计算第一个龙身节点的角度
func_th = @(dtheta) k^2*(theta_h+dtheta)^2 -
    2*k^2*theta_h*(theta_h+dtheta)*cos(dtheta) + k^2*theta_h^2 - len_h^2;
dtheta = fsolve(func_th, 0.15);
theta(1) = theta_h + dtheta;

% 计算其余龙身节点的角度
for i = 2:223
func_tb = @(dtheta) k^2*(theta(i-1)+dtheta)^2 -
    2*k^2*theta(i-1)*(theta(i-1)+dtheta)*cos(dtheta) + k^2*theta(i-1)^2
    - len_b^2;
dtheta = fsolve(func_tb, 0.15);
theta(i) = theta(i-1) + dtheta;
end
```

附录2: 问题1代码(q1.m)

```
% 计算速度
v(1) = v_h; % 龙头速度
% 计算第一个龙身节点的速度
v(2) = (theta(1)^2+1)^0.5/(theta_h^2+1)^0.5 *
    (theta(1)*theta_h*sin(theta(1)-theta_h)+theta(1)*
cos(theta(1)-theta_h)-theta_h) ...
/ (theta(1)*theta_h*sin(theta(1)-theta_h)-theta_h*
cos(theta(1)-theta_h)+theta(1));
% 计算其余节点的速度
for i = 2:223
v(i+1) = ((theta(i)^2+1)^0.5/(theta(i-1)^2+1)^0.5 *
(theta(i)*theta(i-1)*sin(theta(i)-theta(i-1))+theta(i)*
cos(theta(i)-theta(i-1))-theta(i-1)) ...
/ (theta(i)*theta(i-1)*sin(theta(i)-theta(i-1))-theta(i-1)*
cos(theta(i)-theta(i-1))+theta(i)))*v(i);
end

% 计算位置
pos(1,1)= k*theta_h*cos(theta_h);
pos(1,2)= k*theta_h*sin(theta_h);
for i = 2:224
pos(i,1)= k*theta(i-1)*cos(theta(i-1));
pos(i,2)= k*theta(i-1)*sin(theta(i-1));
end

% 存储结果
for i = 1:224
ans1(i*2-1,t+1) = pos(i,1);
ans1(i*2,t+1) = pos(i,2);
ans2(i,t+1) = v(i);
end
end

% 绘图
figure;
l1 = plot(pos(:, 1), pos(:, 2), '-o', 'LineWidth', 3, 'MarkerSize', 10);
hold on;
theta2 = linspace(0, theta_s, 10000);
r_plt = k * theta2;
x = r_plt .* cos(theta2);
y = r_plt .* sin(theta2);
```

附录2: 问题1代码(q1.m)

```
l2 = plot(x, y, 'r', 'LineWidth', 1.5);
l3 = plot(pos(1,1), pos(1,2), 'o', 'Color', 'g', 'MarkerSize', 15,
    'LineWidth', 2);
l4 = plot(pos(224,1), pos(224,2), 'x', 'Color', 'y', 'MarkerSize', 15,
    'LineWidth', 2);
axis equal;
xlabel('X');
ylabel('Y');
legend([l1, l2, l3, l4], {'板凳龙', '螺线', '龙头', '龙尾'});
title('螺线及板凳龙位置');
grid on;
```

附录2: 问题2代码(A2.m)

```
clc
clear

gap = 0.55;   %螺距
k = gap/(2*pi);
v_h = 1;      %龙头速度
theta_s = 32*pi;
len_h = 2.86; %龙头长
len_b = 1.65; %龙身长
w = 0.15;     %板凳半宽
len_e = 0.275; %把手前长度

inner = zeros(1002,1); %条带内层厚度
outer = zeros(1002,1); %条带外层厚度
tmp = zeros(1002,1);

func_len = @(theta)
    k/2*(theta*(1+theta^2)^0.5+log(theta+(1+theta^2)^0.5));
tot_s = func_len(theta_s); %总长

for theta_a = 0:0.1:100 %前把手极角
pos = zeros(2,2);
func_tb = @(dtheta) k^2*(theta_a+dtheta)^2 -
    2*k^2*theta_a*(theta_a+dtheta)*cos(dtheta) + k^2*theta_a^2 - len_b^2;
dtheta = fsolve(func_tb, 0.15);
theta_b = theta_a + dtheta; %后把手极角
pos(1,1) = k*theta_a*cos(theta_a); %前把手x
pos(1,2) = k*theta_a*sin(theta_a); %前把手y
pos(2,1) = k*theta_b*cos(theta_b); %后把手x
pos(2,2) = k*theta_b*sin(theta_b); %前把手y
kq = (pos(1,2)-pos(2,2))/(pos(1,1)-pos(2,1)); %把手间连线斜率
d = abs(kq*pos(1,1)-pos(1,2)) / sqrt(kq^2+1);
kq = -1/kq;
theta_i = atan(kq);
if theta_i<0
theta_i = theta_i + pi;
end
theta_i = theta_i + pi*floor(theta_a/pi); %内层条带最厚处对应极角
r = k*theta_i;
inner(round(theta_i*10)+1) = max(inner(round(theta_i*10)+1),r-d+w);
end
inner(inner==0) = NaN;
inner = fillmissing(inner, "linear"); %线性插值补全缺失值
```

附录2: 问题2代码(A2.m)

```

for theta_a = 0:0.1:100
pos = zeros(4,2);
func_th = @(dtheta) k^2*(theta_a+dtheta)^2 -
    2*k^2*theta_a*(theta_a+dtheta)*cos(dtheta) + k^2*theta_a^2 - len_h^2;
dtheta = fsolve(func_th, 0.15);
theta_b = theta_a + dtheta;
pos(1,1) = k*theta_a*cos(theta_a);
pos(1,2) = k*theta_a*sin(theta_a);
pos(2,1) = k*theta_b*cos(theta_b);
pos(2,2) = k*theta_b*sin(theta_b);
pos(3,1) = pos(1,1) + (pos(1,1)-pos(2,1))*len_e/len_h;
pos(3,2) = pos(1,2) + (pos(1,2)-pos(2,2))*len_e/len_h;
pos(4,1) = pos(3,1) + (pos(1,2)-pos(3,2))*w/len_e; %角点x
pos(4,2) = pos(3,2) - (pos(1,1)-pos(3,1))*w/len_e; %角点y
d = (pos(4,1)^2+pos(4,2)^2)^0.5;
theta_o = atan2(pos(4,2),pos(4,1));
if theta_o<0
theta_o = theta_o + 2*pi;
end
theta_o = theta_o + 2*pi*floor(theta_a/pi/2); %外层条带最厚处对应极角
if (pos(1,1)>0) && (pos(1,2)>0) && (pos(4,1)>0) && (pos(4,2)<0)
theta_o = theta_o - 2*pi;
end
r = k*theta_o;
if theta_o < 0
theta_o = 0;
end
outer(round(theta_o*10)+1) = max(outer(round(theta_o*10)+1),d-r);
end
outer(outer==0) = NaN;
outer = fillmissing(outer, "linear");

for theta = 75:-0.1:2*pi %从后向前搜索碰撞点
tmp(round(theta*10)+1) = inner(round(theta*10)+1) +
    outer(round((theta-2*pi)*10)+1);
if (inner(round(theta*10)+1) + outer(round((theta-2*pi)*10)+1)) >= gap
theta = theta - 2*pi;
ans = (tot_s - func_len(theta))/v_h;
disp(theta)
disp(ans)
break
end
```

附录2: 问题2代码(A2.m)

```
        disp(theta)
        disp(ans)
        break
        end
        end
        coli_x = (k*theta+outer(round(theta*10)+1))*cos(theta); %碰撞点
        coli_y = (k*theta+outer(round(theta*10)+1))*sin(theta);

        %画图部分
        theta1 = linspace(0, theta_s, 10000);
        r_plt = k * theta1;
        x_plt = r_plt .* cos(theta1);
        y_plt = r_plt .* sin(theta1);

        r_i = zeros(1, 1000);
        r_o = zeros(1, 1000);
        theta2 = linspace(0, 100, 1000);
        for i = 1:1000
        r_i(i) = k * (i - 1) / 10 - inner(i);
        r_o(i) = k * (i - 1) / 10 + outer(i);
        end

        r_i = r_i(theta2 >= theta);
        r_o = r_o(theta2 >= theta);
        theta2 = theta2(theta2 >= theta);

        x_i = r_i .* cos(theta2);
        y_i = r_i .* sin(theta2);
        x_o = r_o .* cos(theta2);
        y_o = r_o .* sin(theta2);

        figure;
        l1 = plot(x_plt, y_plt, 'b', 'LineWidth', 1.5);
        hold on;
        plot(x_i, y_i, 'r', 'LineWidth', 0.5);
        plot(x_o, y_o, 'r', 'LineWidth', 0.5);

        fill_x = [x_i, fliplr(x_o)];
        fill_y = [y_i, fliplr(y_o)];
        l2 = fill(fill_x, fill_y, 'r', 'EdgeColor', 'none', 'FaceAlpha', 0.3);
        l3 = plot(coli_x, coli_y, 'x', 'Color', 'y', 'MarkerSize', 20,
            'LineWidth', 2);
```

附录2: 问题2代码(A2.m)

```
hold off;
axis equal;
xlabel('X');
ylabel('Y');
legend([l1, l2, l3], {'螺线', '条带区域', '碰撞点'});
title('条带与碰撞点');
grid on;
```

附录2: 问题2代码(q2.m)

```
clc
clear

% 开始计时
tic;

% 初始化参数
k = 0.55 / (2*pi);
v = 1; % m/s
dt = 0.0000001; % 时间步长
n_segments = 223;
lengths = [286, repmat(165, 1, 222)]; % cm
bench_width = 30; % cm
hole_distance = 27.5; % cm

% 定义螺线和弧长函数
spiral = @(theta) k * theta .* [cos(theta), sin(theta)];
S = @(theta) k * (theta/2 * sqrt(theta^2 + 1) + 1/2 * log(theta +
    sqrt(theta^2 + 1)));
S_tot= k * (32*pi/2 * sqrt((32*pi)^2 + 1) + 1/2 * log(32*pi+
    sqrt((32*pi)^2 + 1)));


% 主循环
t = 412.47383;
positions = zeros(n_segments+1, 2);
velocities = zeros(n_segments+1, 1);
angles = zeros(n_segments+1, 1);
collision_detected = false;

while ~collision_detected
t = t + dt;

% 计算龙头位置
theta_head = solve_theta(t, k, v,S,S_tot);
positions(1,:) = spiral(theta_head);
velocities(1) = v ;
angles(1) = theta_head;
```

附录2: 问题2代码(q2.m)

```
% 计算龙身和龙尾
for i = 2:n_segments+1
d = lengths(i-1) / 100; % 转换为米
func_tb = @(dtheta) k^2*(angles(i-1)+dtheta)^2 -
    2*k^2*angles(i-1)*(angles(i-1)+dtheta)*cos(dtheta) +
    k^2*angles(i-1)^2 - d^2;
dtheta = fsolve(func_tb, 0.15);
angles(i) = angles(i-1) + dtheta;
positions(i,:) = spiral(angles(i));
velocities(i) = ((angles(i)^2+1)^0.5/(angles(i-1)^2+1)^0.5 *
(angles(i)*angles(i-1)*sin(angles(i)-angles(i-1))+angles(i)*
cos(angles(i)-angles(i-1))-angles(i-1)) ...
/ (angles(i)*angles(i-1)*
sin(angles(i)-angles(i-1))-angles(i-1)*cos(angles(i)-angles(i-1))
+angles(i)))*velocities(i-1);
end

% 碰撞检测
[collision_detected, min_distance] = detect_collision(positions, angles);

end

% 输出结果
fprintf('终止时刻: %.6f 秒\n', t);
fprintf('最小距离: %.6f cm\n', min_distance * 100);

% 输出特定节点的位置和速度
special_nodes = [1, 2, 52, 102, 152, 202, 223];
for i = special_nodes
fprintf('节点 %d:\n', i);
fprintf(' 位置: (%.6f, %.6f)\n', positions(i,1), positions(i,2));
fprintf(' 速度: %.6f\n', velocities(i));
end
% 结束计时并显示运行时间
elapsedTime = toc;
fprintf('程序运行总时间: %.2f 秒\n', elapsedTime);

% 保存结果到文件Excel
results = [positions, velocities];
n_columns = size(results, 2);
headers = cell(1, n_columns);
```

附录2: 问题2代码(q2.m)

```
% 确保我们为每一列创建一个头部
headers{1} = sprintf('x');
headers{2} = sprintf('y');
headers{3} = sprintf('v');


% 检查是否所有的头部都已填充
if any(cellfun(@isempty, headers))
error('未能为所有列创建头部。请检查 results 数组的列数。');
end

T = array2table(results, 'VariableNames', headers);
writetable(T, 'result2.xlsx');
% 绘图
figure;
plot(positions(:,1), positions(:,2), 'b-o','LineWidth', 0.8);
hold on;
plot(positions(1,1), positions(1,2), 'ro', 'MarkerSize', 10,'LineWidth',
    0.8, 'MarkerFaceColor', 'r');
plot(positions(end,1), positions(end,2), 'go', 'MarkerSize',
    10,'LineWidth', 0.8, 'MarkerFaceColor', 'g');
title('板凳龙盘入终止时刻的位置');
xlabel('X (m)');
ylabel('Y (m)');
legend('龙身', '龙头', '龙尾');
axis equal;
grid on;


% 求解的函数theta
function theta = solve_theta(t, k, v, S,S_tot)
f = @(th) S_tot - S(th) - v*t;
theta = fsolve(f, 0);
end

% 碰撞检测函数
function [collision, min_distance] = detect_collision(positions, angles)
n = size(positions, 1);
collision = false;
min_distance = inf;

theta_h=atan2(positions(1,2)-positions(2,2),positions(1,1)-positions(2,1));
```

附录2: 问题2代码(q2.m)

```
% 龙头节点的长度和宽度
head_length = 0.275; % 米
head_width = 0.30; % 米

% 龙身节点的长度和宽度
body_length = 0.275; % 米
body_width = 0.30; % 米

% 龙头前端两个角点的位置
head_front_left = positions(1,:) + [cos(theta_h)*head_length -
    sin(theta_h)*head_width/2, ...
sin(theta_h)*head_length + cos(theta_h)*head_width/2];
head_front_right = positions(1,:) + [cos(theta_h)*head_length +
    sin(theta_h)*head_width/2, ...
sin(theta_h)*head_length - cos(theta_h)*head_width/2];

% 检查龙头与每个龙身节点的碰撞
for i = 2:n-1
% 龙身节点的四个角点
theta=atan2(positions(i,2)-positions(i+1,2),positions(i,1)-positions(i+1,1));

body_corners = [
positions(i+1,:) + [-cos(theta)*body_length + sin(theta)*body_width/2,
    -sin(theta)*body_length - cos(theta)*body_width/2];
positions(i,:) + [cos(theta)*body_length + sin(theta)*body_width/2,
    sin(theta)*body_length - cos(theta)*body_width/2];
positions(i,:) + [cos(theta)*body_length - sin(theta)*body_width/2,
    sin(theta)*body_length + cos(theta)*body_width/2];
positions(i+1,:) + [-cos(theta)*body_length - sin(theta)*body_width/2,
    -sin(theta)*body_length + cos(theta)*body_width/2]

];

% 检查龙头前端两个角点是否在龙身矩形内
if is_point_in_rectangle(head_front_left, body_corners) || ...
is_point_in_rectangle(head_front_right, body_corners)
collision = true;
d = min(min(pdist2(head_front_left, body_corners)),
    min(pdist2(head_front_right, body_corners)));
min_distance = min(min_distance, d);
end
end
end
```

附录2: 问题2代码(q2.m)

```
function in_rect = is_point_in_rectangle(point, corners)
% 使用叉积判断点是否在矩形内
in_rect = true;
for i = 1:4
v1 = corners(mod(i,4)+1,:) - corners(i,:);
v2 = point - corners(i,:);
if cross2d(v1, v2) < 0
in_rect = false;
break;
end
end
end

function z = cross2d(a, b)
z = a(1)*b(2) - a(2)*b(1);
end
```

附录2: 问题3代码(A3.m)

```
clc
clear

v_h = 1;      %龙头速度
theta_s = 32*pi;

len_h = 2.86; %龙头长
len_b = 1.65; %龙身长
w = 0.15;     %板凳半宽
len_e = 0.275; %把手前长度

inner = zeros(702,1);
outer = zeros(702,1);

for gap = 0.55:-0.001:0 %螺距
k = gap/(2*pi);
theta_t = 4.5/k;
for theta_a = 0:0.1:60 %前把手极角
pos = zeros(2,2);
func_tb = @(dtheta) k^2*(theta_a+dtheta)^2 -
    2*k^2*theta_a*(theta_a+dtheta)*cos(dtheta) + k^2*theta_a^2 - len_b^2;
dtheta = fsolve(func_tb, 0.15);
theta_b = theta_a + dtheta; %后把手极角
pos(1,1) = k*theta_a*cos(theta_a); %前把手x
pos(1,2) = k*theta_a*sin(theta_a); %前把手y
pos(2,1) = k*theta_b*cos(theta_b); %后把手x
pos(2,2) = k*theta_b*sin(theta_b); %后把手y
kq = (pos(1,2)-pos(2,2))/(pos(1,1)-pos(2,1)); %把手间连线斜率
d = abs(kq*pos(1,1)-pos(1,2)) / sqrt(kq^2+1);
kq = -1/kq;
theta_i = atan(kq);
if theta_i<0
theta_i = theta_i + pi;
end
theta_i = theta_i + pi*floor(theta_a/pi); %内层条带最厚处对应极角
r = k*theta_i;
inner(round(theta_i*10)+1) = max(inner(round(theta_i*10)+1),r-d+w);
end
inner(inner==0) = NaN;
inner = fillmissing(inner, "linear"); %线性插值补全缺失值
```

附录2: 问题3代码(A3.m)

```
for theta_a = 0:0.1:60
pos = zeros(4,2);
func_th = @(dtheta) k^2*(theta_a+dtheta)^2 -
    2*k^2*theta_a*(theta_a+dtheta)*cos(dtheta) + k^2*theta_a^2 - len_h^2;
dtheta = fsolve(func_th, 0.15);
theta_b = theta_a + dtheta;
pos(1,1) = k*theta_a*cos(theta_a);
pos(1,2) = k*theta_a*sin(theta_a);
pos(2,1) = k*theta_b*cos(theta_b);
pos(2,2) = k*theta_b*sin(theta_b);
pos(3,1) = pos(1,1) + (pos(1,1)-pos(2,1))*len_e/len_h;
pos(3,2) = pos(1,2) + (pos(1,2)-pos(2,2))*len_e/len_h;
pos(4,1) = pos(3,1) + (pos(1,2)-pos(3,2))*w/len_e; %角点x
pos(4,2) = pos(3,2) - (pos(1,1)-pos(3,1))*w/len_e; %角点y
d = (pos(4,1)^2+pos(4,2)^2)^0.5;
theta_o = atan2(pos(4,2),pos(4,1));
if theta_o<0
theta_o = theta_o + 2*pi;
end
theta_o = theta_o + 2*pi*floor(theta_a/pi/2); %外层条带最厚处对应极角
if (pos(1,1)>0) && (pos(1,2)>0) && (pos(4,1)>0) && (pos(4,2)<0)
theta_o = theta_o - 2*pi;
end
r = k*theta_o;
if theta_o < 0
theta_o = 0;
end
outer(round(theta_o*10)+1) = max(outer(round(theta_o*10)+1),d-r);
end
outer(outer==0) = NaN;
outer = fillmissing(outer, "linear");

theta = 4.5/k; %进入调头区域点极角
if (inner(round(theta*10)+1) + outer(round((theta-2*pi)*10)+1)) >= gap
break
end
end
disp(gap)
```

附录2: 问题3代码(q3.m)

```
clc
clear

% 初始化参数
v = 1; % m/s
dt = 0.01; % 时间步长
n_segments = 223;
lengths = [286, repmat(165, 1, 222)]; % cm
bench_width = 30; % cm
hole_distance = 27.5; % cm

for c = 0.4498:0.00001:0.4499
% c=0.4498;
k = c / (2*pi);

% 定义螺线和弧长函数
spiral = @(theta) k * theta .* [cos(theta), sin(theta)];
S = @(theta) k * (theta/2 * sqrt(theta^2 + 1) + 1/2 * log(theta +
    sqrt(theta^2 + 1)));
S_tot= k * (32*pi/2 * sqrt((32*pi)^2 + 1) + 1/2 * log(32*pi+
    sqrt((32*pi)^2 + 1)));


% 主循环
t = 211;
positions = zeros(n_segments+1, 2);
velocities = zeros(n_segments+1, 2);
angles = zeros(n_segments+1, 1);
collision_detected = false;

while ~collision_detected
t = t + dt;

% 计算龙头位置
theta_head = solve_theta(t, k, v,S,S_tot);
positions(1,:) = spiral(theta_head);
velocities(1,:) = v * [sin(theta_head), -cos(theta_head)];
angles(1) = theta_head;
```

附录2: 问题3代码(q3.m)

```
% 计算龙身和龙尾
for i = 2:n_segments+1
d = lengths(i-1) / 100; % 转换为米
func_tb = @(dtheta) k^2*(angles(i-1)+dtheta)^2 -
    2*k^2*angles(i-1)*(angles(i-1)+dtheta)*cos(dtheta) +
    k^2*angles(i-1)^2 - d^2;
dtheta = fsolve(func_tb, 0.15);
angles(i) = angles(i-1) + dtheta;
positions(i,:) = spiral(angles(i));
velocities(i,:) = v * [sin(angles(i)), -cos(angles(i))];
end

% 碰撞检测
[collision_detected, min_distance] = detect_collision(positions, angles);

end
if c/(2*pi)*angles(1)<=4.5
fprintf('角长度rou: %.6f m\n', c/(2*pi)*angles(1));
fprintf("C:%.6f\n",c);
% break
end
end




% 输出结果
fprintf('终止时刻: %.6f 秒\n', t);
fprintf('最小距离: %.6f cm\n', min_distance * 100);
fprintf('角长度rou: %.6f m\n', c/(2*pi)*angles(1));
fprintf("C:%.6f",c);

% 输出特定节点的位置和速度
special_nodes = [1, 2, 52, 102, 152, 202, 223];
for i = special_nodes
fprintf('节点 %d:\n', i);
fprintf(' 位置: (%.6f, %.6f)\n', positions(i,1), positions(i,2));
fprintf(' 速度: (%.6f, %.6f)\n', velocities(i,1), velocities(i,2));
end
```

附录2: 问题3代码(q3.m)

```
% 保存结果到文件Excel
results = [positions, velocities];
n_columns = size(results, 2);
headers = cell(1, n_columns);

% 确保我们为每一列创建一个头部
for i = 1:n_columns/4
headers{4*i-3} = sprintf('x%d', i);
headers{4*i-2} = sprintf('y%d', i);
headers{4*i-1} = sprintf('vx%d', i);
headers{4*i} = sprintf('vy%d', i);
end

% 检查是否所有的头部都已填充
%if any(cellfun(@isempty, headers))
%    error未能为所有列创建头部。请检查(' results 数组的列数。');
%end

%T = array2table(results, 'VariableNames', headers);
%writetable(T, 'result2.xlsx');
% 绘图
figure;
plot(positions(:,1), positions(:,2), 'b-');
hold on;
plot(positions(1,1), positions(1,2), 'ro', 'MarkerSize', 10);
plot(positions(end,1), positions(end,2), 'go', 'MarkerSize', 10);
title('板凳龙盘入终止时刻的位置');
xlabel('X (m)');
ylabel('Y (m)');
legend('龙身', '龙头', '龙尾');
axis equal;
grid on;


% 求解的函数theta
function theta = solve_theta(t, k, v, S,S_tot)
f = @(th) S_tot - S(th) - v*t;
theta = fsolve(f, 0);
end
```

附录2: 问题3代码(q3.m)

```
% 碰撞检测函数
function [collision, min_distance] = detect_collision(positions, angles)
n = size(positions, 1);
collision = false;
min_distance = inf;

theta_h=atan2(positions(1,2)-positions(2,2),positions(1,1)-positions(2,1));

% 龙头节点的长度和宽度
head_length = 0.275; % 米
head_width = 0.30; % 米

% 龙身节点的长度和宽度
body_length = 0.275; % 米
body_width = 0.30; % 米

% 龙头前端两个角点的位置
head_front_left = positions(1,:) + [cos(theta_h)*head_length -
    sin(theta_h)*head_width/2, ...
sin(theta_h)*head_length + cos(theta_h)*head_width/2];
head_front_right = positions(1,:) + [cos(theta_h)*head_length +
    sin(theta_h)*head_width/2, ...
sin(theta_h)*head_length - cos(theta_h)*head_width/2];

% 检查龙头与每个龙身节点的碰撞
for i = 2:n-1
% 龙身节点的四个角点
theta=atan2(positions(i,2)-positions(i+1,2),positions(i,1)-positions(i+1,1));

body_corners = [
positions(i+1,:) + [-cos(theta)*body_length + sin(theta)*body_width/2,
    -sin(theta)*body_length - cos(theta)*body_width/2];
positions(i,:) + [cos(theta)*body_length + sin(theta)*body_width/2,
    sin(theta)*body_length - cos(theta)*body_width/2];
positions(i,:) + [cos(theta)*body_length - sin(theta)*body_width/2,
    sin(theta)*body_length + cos(theta)*body_width/2];
positions(i+1,:) + [-cos(theta)*body_length - sin(theta)*body_width/2,
    -sin(theta)*body_length + cos(theta)*body_width/2]

];
```

附录2: 问题3代码(q3.m)

```
        % 检查龙头前端两个角点是否在龙身矩形内
        if is_point_in_rectangle(head_front_left, body_corners) || ...
        is_point_in_rectangle(head_front_right, body_corners)
        collision = true;
        d = min(min(pdist2(head_front_left, body_corners)),
            min(pdist2(head_front_right, body_corners)));
        min_distance = min(min_distance, d);
        end
        end
        end

        function in_rect = is_point_in_rectangle(point, corners)
        % 使用叉积判断点是否在矩形内
        in_rect = true;
        for i = 1:4
        v1 = corners(mod(i,4)+1,:) - corners(i,:);
        v2 = point - corners(i,:);
        if cross2d(v1, v2) < 0
        in_rect = false;
        break;
        end
        end
        end

        function z = cross2d(a, b)
        z = a(1)*b(2) - a(2)*b(1);
        end
```

附录2: 问题4和问题5代码(q4_q5.m)

```
clc
clear

% 初始化参数
k = 1.7/(2*pi); % 螺线参数
v_h = 1; % 龙头速度(m/s)
theta_s = 9/1.7*pi; % 初始角度
len_h = 2.86; % 龙头长度(m)
len_b = 1.65; % 龙身长度(m)

% 定义旋转矩阵
rot = [cos(0.5/1.7*pi), -sin(0.5/1.7*pi);
sin(0.5/1.7*pi), cos(0.5/1.7*pi)];

% 时间参数设置
t_ini = -100; % 初始时间
t_iter = 1; % 时间步长
t_num = 201; % 时间点数量
dt = 0.00002; % 微小时间步长，用于计算速度

% 初始化位置、角度和速度数组
pos_h = zeros(2, t_num); % 龙头位置
pos = zeros(2, t_num, 224); % 所有节点位置
pos_before = zeros(2, t_num, 224); % 计算速度用的前一时刻位置
pos_after = zeros(2, t_num, 224); % 计算速度用的后一时刻位置
pos_state = zeros(t_num, 224); % 位置状态
theta = zeros(t_num, 224); % 角度
rou = zeros(t_num, 224); % 到原点的距离
v = zeros(t_num, 224); % 速度

% 主循环，计算每个时间点的位置
for j = 1:t_num
t=t_ini+(j-1)*t_iter;
if t<=0
tot_s = k/2*(theta_s*(1+theta_s^2)^0.5+log(theta_s+(1+theta_s^2)^0.5));
func_sh = @(theta)
    k/2*(theta*(1+theta^2)^0.5+log(theta+(1+theta^2)^0.5)) - tot_s +
    v_h*t;
theta_h = fsolve(func_sh, 0);
pos_h(:,j) = [k*theta_h*cos(theta_h);
k*theta_h*sin(theta_h)];
```

附录2: 问题4和问题5代码(q4_q5.m)

```
        elseif t>0&&t<=3*pi/v_h
        pos_h(:,j) = rot*[-1.5-3*cos(v_h*t/3);3*sin(v_h*t/3)];
        elseif t>3*pi/v_h&&t<=4.5*pi/v_h
        pos_h(:,j) = rot*[3-1.5*cos(v_h*(t-3*pi/v_h)/1.5);
        -1.5*sin(v_h*(t-3*pi/v_h)/1.5)];
        else
        tot_s = k/2*(theta_s*(1+theta_s^2)^0.5+log(theta_s+(1+theta_s^2)^0.5));
        func_sh = @(theta)
            k/2*(theta*(1+theta^2)^0.5+log(theta+(1+theta^2)^0.5)) - tot_s +
            v_h*(4.5*pi/v_h-t);
        theta_h = fsolve(func_sh, 0);
        pos_h(:,j) = [-k*theta_h*cos(theta_h);
        -k*theta_h*sin(theta_h)];
        end


        pos(1,j,1)=pos_h(1,j);
        pos(2,j,1)=pos_h(2,j);
        rou(j,1) = sqrt((pos_h(1,j))^2+(pos_h(2,j))^2);
        if rou(j,1)>4.5
        func_th = @(dtheta) k^2*(theta_h+dtheta)^2 -
            2*k^2*theta_h*(theta_h+dtheta)*cos(dtheta) + k^2*theta_h^2 - len_h^2;
        dtheta = fsolve(func_th,-0.15*sign(t-7.50000101));
        theta(j,2) = theta_h + dtheta;
        if theta(j,2)>=9/1.7*pi
        pos(1,j,2)=-k*theta(j,2)*cos(theta(j,2))*sign(t-7.50000101);
        pos(2,j,2)=-k*theta(j,2)*sin(theta(j,2))*sign(t-7.50000101);
        pos_state(j,2)=sign(t-7.50000101);
        else
```

附录2: 问题4和问题5代码(q4_q5.m)

```
        temp_pos=rot\[pos(1,j,1);pos(2,j,1)];
        temp_x0=temp_pos(1);
        temp_y0=temp_pos(2);
        syms sym_x;

        eqn = ((2*temp_x0-6)^2+4*temp_y0^2)*sym_x^2 +
            (-24*temp_y0^2+(4*temp_x0-12)*(len_h^2-temp_x0^2-temp_y0^2+6.75))*sym_x
            + 27*temp_y0^2+(len_h^2-temp_x0^2-temp_y0^2+6.75)^2 == 0;
        sym_x = solve(eqn,sym_x);
        temp_xs = double(sym_x);
        temp_x = min(real(temp_xs));
        temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
        if abs(sqrt((temp_x-temp_x0)^2+(temp_y-temp_y0)^2) - len_h) > 0.01
        temp_x = max(real(temp_xs));
        temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
        end
        final_pos = rot*[temp_x;temp_y];
        pos(1,j,2) = final_pos(1);
        pos(2,j,2) = final_pos(2);
        end
        else
        temp_pos=rot\[pos(1,j,1);pos(2,j,1)];
        temp_x0=temp_pos(1);
        temp_y0=temp_pos(2);
        if temp_x0>1.5+(len_h)^2/3&&temp_x0<=4.5
        syms sym_x;
```

附录2: 问题4和问题5代码(q4_q5.m)

```
        eqn = ((2*temp_x0-6)^2+4*temp_y0^2)*sym_x^2 +
            (-24*temp_y0^2+(4*temp_x0-12)*(len_h^2-temp_x0^2-temp_y0^2+6.75))*sym_x
            + 27*temp_y0^2+(len_h^2-temp_x0^2-temp_y0^2+6.75)^2 == 0;
        sym_x = solve(eqn,sym_x);
        temp_xs = double(sym_x);
        temp_x = min(real(temp_xs));
        temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
        if abs(sqrt((temp_x-temp_x0)^2+(temp_y-temp_y0)^2) - len_h) > 0.01
        temp_x = max(real(temp_xs));
        temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
        end
        final_pos = rot*[temp_x;temp_y];
        pos(1,j,2) = final_pos(1);
        pos(2,j,2) = final_pos(2);
        elseif temp_x0>-4.5+(len_h)^2/6 && temp_x0<=1.5+(len_h)^2/3
        syms sym_x;
        eqn = ((2*temp_x0+3)^2+4*temp_y0^2)*sym_x^2 +
            (12*temp_y0^2-(4*temp_x0+6)*(6.75+temp_x0^2+temp_y0^2-len_h^2))*sym_x
            + (6.75+temp_x0^2+temp_y0^2-len_h^2)^2-27*temp_y0^2 == 0;
        sym_x = solve(eqn,sym_x);
        temp_xs = double(sym_x);
        temp_x = min(real(temp_xs));
        temp_y = sqrt(-(temp_x)^2-3*temp_x+6.75);
        if abs(sqrt((temp_x-temp_x0)^2+(temp_y-temp_y0)^2) - len_h) > 0.01
        temp_x = max(real(temp_xs));
        temp_y = sqrt(-(temp_x)^2-3*temp_x+6.75);
        end
        final_pos = rot*[temp_x;temp_y];
        pos(1,j,2) = final_pos(1);
        pos(2,j,2) = final_pos(2);
        else
        func_cl = @(temp_theta)
            k^2*temp_theta^2-2*k*temp_theta*(pos(1,j,1)*cos(temp_theta)+pos(2,j,1)*
        sin(temp_theta))+(pos(1,j,1))^2+(pos(2,j,1))^2-len_h^2;
        temp_theta = fsolve(func_cl,5.5*pi);
        pos(1,j,2) = k*temp_theta*cos(temp_theta);
        pos(2,j,2) = k*temp_theta*sin(temp_theta);
        theta(j,2) = temp_theta;
        pos_state(j,2) = -1;
        end
        end
```

附录2: 问题4和问题5代码(q4_q5.m)

```
rou(j,2) = sqrt((pos(1,j,2))^2+(pos(2,j,2))^2);
for l=2:224
if rou(j,l)>4.5
func_th = @(dtheta) k^2*(theta(j,l)+dtheta)^2 -
    2*k^2*theta(j,l)*(theta(j,l)+dtheta)*cos(dtheta) + k^2*theta(j,l)^2
    - len_b^2;
dtheta = fsolve(func_th,-0.15*pos_state(j,l));
theta(j,l+1) = theta(j,l) + dtheta;
if theta(j,l+1)>=9/1.7*pi
pos(1,j,l+1)=-k*theta(j,l+1)*cos(theta(j,l+1))*pos_state(j,l);
pos(2,j,l+1)=-k*theta(j,l+1)*sin(theta(j,l+1))*pos_state(j,l);
pos_state(j,l+1)=pos_state(j,l);
else
temp_pos=rot\[pos(1,j,l);pos(2,j,l)];
temp_x0=temp_pos(1);
temp_y0=temp_pos(2);
syms sym_x;
eqn = ((2*temp_x0-6)^2+4*temp_y0^2)*sym_x^2 +
    (-24*temp_y0^2+(4*temp_x0-12)*(len_b^2-temp_x0^2-temp_y0^2+6.75))*sym_x
    + 27*temp_y0^2+(len_b^2-temp_x0^2-temp_y0^2+6.75)^2 == 0;
sym_x = solve(eqn,sym_x);
temp_xs = double(sym_x);
temp_x = min(real(temp_xs));
temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
if abs(sqrt((temp_x-temp_x0)^2+(temp_y-temp_y0)^2) - len_b) > 0.01
temp_x = max(real(temp_xs));
temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
end
final_pos = rot*[temp_x;temp_y];
pos(1,j,l+1) = final_pos(1);
pos(2,j,l+1) = final_pos(2);
end
else
temp_pos=rot\[pos(1,j,l);pos(2,j,l)];
temp_x0=temp_pos(1);
temp_y0=temp_pos(2);
if temp_x0>1.5+(len_b)^2/3&&temp_x0<=4.5
syms sym_x;
eqn = ((2*temp_x0-6)^2+4*temp_y0^2)*sym_x^2 +
    (-24*temp_y0^2+(4*temp_x0-12)*(len_b^2-temp_x0^2-temp_y0^2+6.75))*sym_x
    + 27*temp_y0^2+(len_b^2-temp_x0^2-temp_y0^2+6.75)^2 == 0;
```

附录2: 问题4和问题5代码(q4_q5.m)

```
        sym_x = solve(eqn,sym_x);
        temp_xs = double(sym_x);
        temp_x = min(real(temp_xs));
        temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
        if abs(sqrt((temp_x-temp_x0)^2+(temp_y-temp_y0)^2) - len_b) > 0.01
        temp_x = max(real(temp_xs));
        temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
        end
        final_pos = rot*[temp_x;temp_y];
        pos(1,j,l+1) = final_pos(1);
        pos(2,j,l+1) = final_pos(2);
        elseif temp_x0>-4.5+(len_b)^2/6 && temp_x0<=1.5+(len_b)^2/3
        syms sym_x;
        eqn = ((2*temp_x0+3)^2+4*temp_y0^2)*sym_x^2 +
            (12*temp_y0^2-(4*temp_x0+6)*(6.75+temp_x0^2+temp_y0^2-len_b^2))*sym_x
            + (6.75+temp_x0^2+temp_y0^2-len_b^2)^2-27*temp_y0^2 == 0;
        sym_x = solve(eqn,sym_x);
        temp_xs = double(sym_x);
        temp_x = min(real(temp_xs));
        temp_y = sqrt(-(temp_x)^2-3*temp_x+6.75);
        if abs(sqrt((temp_x-temp_x0)^2+(temp_y-temp_y0)^2) - len_b) > 0.01
        temp_x = max(real(temp_xs));
        temp_y = sqrt(-(temp_x)^2-3*temp_x+6.75);
        end
        final_pos = rot*[temp_x;temp_y];
        pos(1,j,l+1) = final_pos(1);
        pos(2,j,l+1) = final_pos(2);
        else
        func_cl = @(temp_theta)
            k^2*temp_theta^2-2*k*temp_theta*(pos(1,j,l)*cos(temp_theta)
        +pos(2,j,l)*sin(temp_theta))+(pos(1,j,l))^2+(pos(2,j,l))^2-len_b^2;
        temp_theta = fsolve(func_cl,5.5*pi);
        pos(1,j,l+1) = k*temp_theta*cos(temp_theta);
        pos(2,j,l+1) = k*temp_theta*sin(temp_theta);
        pos_state(j,l+1) = -1;
        theta(j,l+1) = temp_theta;
        end
        end
        rou(j,l+1) = sqrt((pos(1,j,l+1))^2+(pos(2,j,l+1))^2);
        end
        end
```

附录2: 问题4和问题5代码(q4_q5.m)

```
1
2           for j = 1:t_num
3           t=t_ini+(j-1)*t_iter-0.5*dt;
4           if t<=0
5           tot_s = k/2*(theta_s*(1+theta_s^2)^0.5+log(theta_s+(1+theta_s^2)^0.5));
6           func_sh = @(theta)
                k/2*(theta*(1+theta^2)^0.5+log(theta+(1+theta^2)^0.5)) - tot_s +
                v_h*t;
7           theta_h = fsolve(func_sh, 0);
8           pos_h(:,j) = [k*theta_h*cos(theta_h);
9           k*theta_h*sin(theta_h)];
10          elseif t>0&&t<=3*pi/v_h
11          pos_h(:,j) = rot*[-1.5-3*cos(v_h*t/3);3*sin(v_h*t/3)];
12          elseif t>3*pi/v_h&&t<=4.5*pi/v_h
13          pos_h(:,j) = rot*[3-1.5*cos(v_h*(t-3*pi/v_h)/1.5);
14          -1.5*sin(v_h*(t-3*pi/v_h)/1.5)];
15          else
16          tot_s = k/2*(theta_s*(1+theta_s^2)^0.5+log(theta_s+(1+theta_s^2)^0.5));
17          func_sh = @(theta)
                k/2*(theta*(1+theta^2)^0.5+log(theta+(1+theta^2)^0.5)) - tot_s +
                v_h*(4.5*pi/v_h-t);
18          theta_h = fsolve(func_sh, 0);
19          pos_h(:,j) = [-k*theta_h*cos(theta_h);
20          -k*theta_h*sin(theta_h)];
21          end
22
23          pos_before(1,j,1)=pos_h(1,j);
24          pos_before(2,j,1)=pos_h(2,j);
25          rou(j,1) = sqrt((pos_h(1,j))^2+(pos_h(2,j))^2);
26          if rou(j,1)>4.5
27          func_th = @(dtheta) k^2*(theta_h+dtheta)^2 -
                2*k^2*theta_h*(theta_h+dtheta)*cos(dtheta) + k^2*theta_h^2 - len_h^2;
28          dtheta = fsolve(func_th,-0.15*sign(t));
29          theta(j,2) = theta_h + dtheta;
30          if theta(j,2)>=9/1.7*pi
31          pos_before(1,j,2)=-k*theta(j,2)*cos(theta(j,2))*sign(t-7.50000101);
32          pos_before(2,j,2)=-k*theta(j,2)*sin(theta(j,2))*sign(t-7.50000101);
33          pos_state(j,2)=sign(t-7.50000101);
34          else
35          temp_pos=rot\[pos_before(1,j,1);pos_before(2,j,1)];
36          temp_x0=temp_pos(1);
37          temp_y0=temp_pos(2);
38          syms sym_x;
```

附录2: 问题4和问题5代码(q4_q5.m)

```
        eqn = ((2*temp_x0-6)^2+4*temp_y0^2)*sym_x^2 +
            (-24*temp_y0^2+(4*temp_x0-12)*(len_h^2-temp_x0^2-temp_y0^2+6.75))*sym_x
            + 27*temp_y0^2+(len_h^2-temp_x0^2-temp_y0^2+6.75)^2 == 0;
        sym_x = solve(eqn,sym_x);
        temp_xs = double(sym_x);
        temp_x = min(real(temp_xs));
        temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
        if abs(sqrt((temp_x-temp_x0)^2+(temp_y-temp_y0)^2) - len_h) > 0.01
        temp_x = max(real(temp_xs));
        temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
        end
        final_pos = rot*[temp_x;temp_y];
        pos_before(1,j,2) = final_pos(1);
        pos_before(2,j,2) = final_pos(2);
        end
        else
        temp_pos=rot\[pos_before(1,j,1);pos_before(2,j,1)];
        temp_x0=temp_pos(1);
        temp_y0=temp_pos(2);
        if temp_x0>1.5+(len_h)^2/3&&temp_x0<=4.5
        syms sym_x;
        eqn = ((2*temp_x0-6)^2+4*temp_y0^2)*sym_x^2 +
            (-24*temp_y0^2+(4*temp_x0-12)*(len_h^2-temp_x0^2-temp_y0^2+6.75))*sym_x
            + 27*temp_y0^2+(len_h^2-temp_x0^2-temp_y0^2+6.75)^2 == 0;
        sym_x = solve(eqn,sym_x);
        temp_xs = double(sym_x);
        temp_x = min(real(temp_xs));
        temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
        if abs(sqrt((temp_x-temp_x0)^2+(temp_y-temp_y0)^2) - len_h) > 0.01
        temp_x = max(real(temp_xs));
        temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
        end
```

附录2: 问题4和问题5代码(q4_q5.m)

```
        final_pos = rot*[temp_x;temp_y];
        pos_before(1,j,2) = final_pos(1);
        pos_before(2,j,2) = final_pos(2);
        elseif temp_x0>-4.5+(len_h)^2/6 && temp_x0<=1.5+(len_h)^2/3
        syms sym_x;
        eqn = ((2*temp_x0+3)^2+4*temp_y0^2)*sym_x^2 +
            (12*temp_y0^2-(4*temp_x0+6)*(6.75+temp_x0^2+temp_y0^2-len_h^2))*sym_x
            + (6.75+temp_x0^2+temp_y0^2-len_h^2)^2-27*temp_y0^2 == 0;
        sym_x = solve(eqn,sym_x);
        temp_xs = double(sym_x);
        temp_x = min(real(temp_xs));
        temp_y = sqrt(-(temp_x)^2-3*temp_x+6.75);
        if abs(sqrt((temp_x-temp_x0)^2+(temp_y-temp_y0)^2) - len_h) > 0.01
        temp_x = max(real(temp_xs));
        temp_y = sqrt(-(temp_x)^2-3*temp_x+6.75);
        end
        final_pos = rot*[temp_x;temp_y];
        pos_before(1,j,2) = final_pos(1);
        pos_before(2,j,2) = final_pos(2);
        else
        func_cl = @(temp_theta)
            k^2*temp_theta^2-2*k*temp_theta*(pos_before(1,j,1)*cos(temp_theta)+pos_before(2,j,1)*sin(temp
        temp_theta = fsolve(func_cl,5.5*pi);
        pos_before(1,j,2) = k*temp_theta*cos(temp_theta);
        pos_before(2,j,2) = k*temp_theta*sin(temp_theta);
        theta(j,2) = temp_theta;
        pos_state(j,2) = -1;
        end
        end
        rou(j,2) = sqrt((pos_before(1,j,2))^2+(pos_before(2,j,2))^2);
        for l=2:224
        if rou(j,l)>4.5
        func_th = @(dtheta) k^2*(theta(j,l)+dtheta)^2 -
            2*k^2*theta(j,l)*(theta(j,l)+dtheta)*cos(dtheta) + k^2*theta(j,l)^2
            - len_b^2;
        dtheta = fsolve(func_th,-0.15*pos_state(j,l));
        theta(j,l+1) = theta(j,l) + dtheta;
        if theta(j,l+1)>=9/1.7*pi
        pos_before(1,j,l+1)=-k*theta(j,l+1)*cos(theta(j,l+1))*pos_state(j,l);
        pos_before(2,j,l+1)=-k*theta(j,l+1)*sin(theta(j,l+1))*pos_state(j,l);
        pos_state(j,l+1)=pos_state(j,l);
        else
        temp_pos=rot\[pos_before(1,j,l);pos_before(2,j,l)];
```

附录2: 问题4和问题5代码(q4_q5.m)

```
        temp_x0=temp_pos(1);
        temp_y0=temp_pos(2);
        syms sym_x;
        eqn = ((2*temp_x0-6)^2+4*temp_y0^2)*sym_x^2 +
            (-24*temp_y0^2+(4*temp_x0-12)*(len_b^2-temp_x0^2-temp_y0^2+6.75))*sym_x
            + 27*temp_y0^2+(len_b^2-temp_x0^2-temp_y0^2+6.75)^2 == 0;
        sym_x = solve(eqn,sym_x);
        temp_xs = double(sym_x);
        temp_x = min(real(temp_xs));
        temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
        if abs(sqrt((temp_x-temp_x0)^2+(temp_y-temp_y0)^2) - len_b) > 0.01
        temp_x = max(real(temp_xs));
        temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
        end
        final_pos = rot*[temp_x;temp_y];
        pos_before(1,j,l+1) = final_pos(1);
        pos_before(2,j,l+1) = final_pos(2);
        end

        else
        temp_pos=rot\[pos_before(1,j,l);pos_before(2,j,l)];
        temp_x0=temp_pos(1);
        temp_y0=temp_pos(2);
        if temp_x0>1.5+(len_b)^2/3&&temp_x0<=4.5
        syms sym_x;
        eqn = ((2*temp_x0-6)^2+4*temp_y0^2)*sym_x^2 +
            (-24*temp_y0^2+(4*temp_x0-12)*(len_b^2-temp_x0^2-temp_y0^2+6.75))*sym_x
            + 27*temp_y0^2+(len_b^2-temp_x0^2-temp_y0^2+6.75)^2 == 0;
        sym_x = solve(eqn,sym_x);
        temp_xs = double(sym_x);
        temp_x = min(real(temp_xs));
        temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
        if abs(sqrt((temp_x-temp_x0)^2+(temp_y-temp_y0)^2) - len_b) > 0.01
        temp_x = max(real(temp_xs));
        temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
        end
        final_pos = rot*[temp_x;temp_y];
        pos_before(1,j,l+1) = final_pos(1);
        pos_before(2,j,l+1) = final_pos(2);
        elseif temp_x0>-4.5+(len_b)^2/6 && temp_x0<=1.5+(len_b)^2/3
        syms sym_x;
```

附录2: 问题4和问题5代码(q4_q5.m)

```
        eqn = ((2*temp_x0+3)^2+4*temp_y0^2)*sym_x^2 +
            (12*temp_y0^2-(4*temp_x0+6)*(6.75+temp_x0^2+temp_y0^2-len_b^2))*sym_x
            + (6.75+temp_x0^2+temp_y0^2-len_b^2)^2-27*temp_y0^2 == 0;

        sym_x = solve(eqn,sym_x);
        temp_xs = double(sym_x);
        temp_x = min(real(temp_xs));
        temp_y = sqrt(-(temp_x)^2-3*temp_x+6.75);
        if abs(sqrt((temp_x-temp_x0)^2+(temp_y-temp_y0)^2) - len_b) > 0.01
        temp_x = max(real(temp_xs));
        temp_y = sqrt(-(temp_x)^2-3*temp_x+6.75);
        end
        final_pos = rot*[temp_x;temp_y];
        pos_before(1,j,l+1) = final_pos(1);
        pos_before(2,j,l+1) = final_pos(2);
        else
        func_cl = @(temp_theta)
            k^2*temp_theta^2-2*k*temp_theta*(pos_before(1,j,l)*cos(temp_theta)
        +pos_before(2,j,l)*sin(temp_theta))+(pos_before(1,j,l))^2+(pos_before(2,j,l))^2-len_b^2;
        temp_theta = fsolve(func_cl,5.5*pi);
        pos_before(1,j,l+1) = k*temp_theta*cos(temp_theta);
        pos_before(2,j,l+1) = k*temp_theta*sin(temp_theta);
        pos_state(j,l+1) = -1;
        theta(j,l+1) = temp_theta;
        end
        end
        rou(j,l+1) = sqrt((pos_before(1,j,l+1))^2+(pos_before(2,j,l+1))^2);
        end
        end



        for j = 1:t_num
        t=t_ini+(j-1)*t_iter+0.5*dt;
        if t<=0
        tot_s = k/2*(theta_s*(1+theta_s^2)^0.5+log(theta_s+(1+theta_s^2)^0.5));
        func_sh = @(theta)
            k/2*(theta*(1+theta^2)^0.5+log(theta+(1+theta^2)^0.5)) - tot_s +
            v_h*t;
        theta_h = fsolve(func_sh, 0);
        pos_h(:,j) = [k*theta_h*cos(theta_h);
        k*theta_h*sin(theta_h)];
```

附录2: 问题4和问题5代码(q4_q5.m)

```
        elseif t>0&&t<=3*pi/v_h
        pos_h(:,j) = rot*[-1.5-3*cos(v_h*t/3);3*sin(v_h*t/3)];
        elseif t>3*pi/v_h&&t<=4.5*pi/v_h
        pos_h(:,j) = rot*[3-1.5*cos(v_h*(t-3*pi/v_h)/1.5);
        -1.5*sin(v_h*(t-3*pi/v_h)/1.5)];
        else

        tot_s = k/2*(theta_s*(1+theta_s^2)^0.5+log(theta_s+(1+theta_s^2)^0.5));
        func_sh = @(theta)
            k/2*(theta*(1+theta^2)^0.5+log(theta+(1+theta^2)^0.5)) - tot_s +
            v_h*(4.5*pi/v_h-t);
        theta_h = fsolve(func_sh, 0);
        pos_h(:,j) = [-k*theta_h*cos(theta_h);
        -k*theta_h*sin(theta_h)];
        end

        pos_after(1,j,1)=pos_h(1,j);
        pos_after(2,j,1)=pos_h(2,j);
        rou(j,1) = sqrt((pos_h(1,j))^2+(pos_h(2,j))^2);
        if rou(j,1)>4.5
        func_th = @(dtheta) k^2*(theta_h+dtheta)^2 -
            2*k^2*theta_h*(theta_h+dtheta)*cos(dtheta) + k^2*theta_h^2 - len_h^2;
        dtheta = fsolve(func_th,-0.15*sign(t-7.50000101));
        theta(j,2) = theta_h + dtheta;
        if theta(j,2)>=9/1.7*pi
        pos_after(1,j,2)=-k*theta(j,2)*cos(theta(j,2))*sign(t-7.50000101);
        pos_after(2,j,2)=-k*theta(j,2)*sin(theta(j,2))*sign(t-7.50000101);
        pos_state(j,2)=sign(t-7.50000101);
        else
        temp_pos=rot\[pos_after(1,j,1);pos_after(2,j,1)];
        temp_x0=temp_pos(1);
        temp_y0=temp_pos(2);
        syms sym_x;
        eqn = ((2*temp_x0-6)^2+4*temp_y0^2)*sym_x^2 +
            (-24*temp_y0^2+(4*temp_x0-12)*(len_h^2-temp_x0^2-temp_y0^2+6.75))*sym_x
            + 27*temp_y0^2+(len_h^2-temp_x0^2-temp_y0^2+6.75)^2 == 0;
        sym_x = solve(eqn,sym_x);
        temp_xs = double(sym_x);
        temp_x = min(real(temp_xs));
```

附录2: 问题4和问题5代码(q4_q5.m)

```
        temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
        if abs(sqrt((temp_x-temp_x0)^2+(temp_y-temp_y0)^2) - len_h) > 0.01
        temp_x = max(real(temp_xs));
        temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
        end
        final_pos = rot*[temp_x;temp_y];
        pos_after(1,j,2) = final_pos(1);
        pos_after(2,j,2) = final_pos(2);
        end

        else
        temp_pos=rot\[pos_after(1,j,1);pos_after(2,j,1)];
        temp_x0=temp_pos(1);
        temp_y0=temp_pos(2);
        if temp_x0>1.5+(len_h)^2/3&&temp_x0<=4.5
        syms sym_x;
        eqn = ((2*temp_x0-6)^2+4*temp_y0^2)*sym_x^2 +
            (-24*temp_y0^2+(4*temp_x0-12)*(len_h^2-temp_x0^2-temp_y0^2+6.75))*sym_x
            + 27*temp_y0^2+(len_h^2-temp_x0^2-temp_y0^2+6.75)^2 == 0;
        sym_x = solve(eqn,sym_x);
        temp_xs = double(sym_x);
        temp_x = min(real(temp_xs));
        temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
        if abs(sqrt((temp_x-temp_x0)^2+(temp_y-temp_y0)^2) - len_h) > 0.01
        temp_x = max(real(temp_xs));
        temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
        end
        final_pos = rot*[temp_x;temp_y];
        pos_after(1,j,2) = final_pos(1);
        pos_after(2,j,2) = final_pos(2);
        elseif temp_x0>-4.5+(len_h)^2/6 && temp_x0<=1.5+(len_h)^2/3
        syms sym_x;
        eqn = ((2*temp_x0+3)^2+4*temp_y0^2)*sym_x^2 +
            (12*temp_y0^2-(4*temp_x0+6)*(6.75+temp_x0^2+temp_y0^2-len_h^2))*sym_x
            + (6.75+temp_x0^2+temp_y0^2-len_h^2)^2-27*temp_y0^2 == 0;
        sym_x = solve(eqn,sym_x);
        temp_xs = double(sym_x);
        temp_x = min(real(temp_xs));
        temp_y = sqrt(-(temp_x)^2-3*temp_x+6.75);
```

附录2: 问题4和问题5代码(q4_q5.m)

```
        if abs(sqrt((temp_x-temp_x0)^2+(temp_y-temp_y0)^2) - len_h) > 0.01
        temp_x = max(real(temp_xs));
        temp_y = sqrt(-(temp_x)^2-3*temp_x+6.75);
        end
        final_pos = rot*[temp_x;temp_y];
        pos_after(1,j,2) = final_pos(1);
        pos_after(2,j,2) = final_pos(2);
        else
        func_cl = @(temp_theta)
            k^2*temp_theta^2-2*k*temp_theta*(pos_after(1,j,1)*cos(temp_theta)
        +pos_after(2,j,1)*sin(temp_theta))+(pos_after(1,j,1))^2+(pos_after(2,j,1))^2-len_h^2;
        temp_theta = fsolve(func_cl,5.5*pi);
        pos_after(1,j,2) = k*temp_theta*cos(temp_theta);
        pos_after(2,j,2) = k*temp_theta*sin(temp_theta);
        theta(j,2) = temp_theta;
        pos_state(j,2) = -1;
        end
        end

        rou(j,2) = sqrt((pos_after(1,j,2))^2+(pos_after(2,j,2))^2);
        for l=2:224
        if rou(j,l)>4.5
        func_th = @(dtheta) k^2*(theta(j,l)+dtheta)^2 -
            2*k^2*theta(j,l)*(theta(j,l)+dtheta)*cos(dtheta) + k^2*theta(j,l)^2
            - len_b^2;
        dtheta = fsolve(func_th,-0.15*pos_state(j,l));
        theta(j,l+1) = theta(j,l) + dtheta;
        if theta(j,l+1)>=9/1.7*pi
        pos_after(1,j,l+1)=-k*theta(j,l+1)*cos(theta(j,l+1))*pos_state(j,l);
        pos_after(2,j,l+1)=-k*theta(j,l+1)*sin(theta(j,l+1))*pos_state(j,l);
        pos_state(j,l+1)=pos_state(j,l);
        else
        temp_pos=rot\[pos_after(1,j,l);pos_after(2,j,l)];
        temp_x0=temp_pos(1);
        temp_y0=temp_pos(2);
        syms sym_x;
        eqn = ((2*temp_x0-6)^2+4*temp_y0^2)*sym_x^2 +
            (-24*temp_y0^2+(4*temp_x0-12)*(len_b^2-temp_x0^2-temp_y0^2+6.75))*sym_x
            + 27*temp_y0^2+(len_b^2-temp_x0^2-temp_y0^2+6.75)^2 == 0;
```

附录2: 问题4和问题5代码(q4_q5.m)

```
        sym_x = solve(eqn,sym_x);
        temp_xs = double(sym_x);
        temp_x = min(real(temp_xs));
        temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
        if abs(sqrt((temp_x-temp_x0)^2+(temp_y-temp_y0)^2) - len_b) > 0.01
        temp_x = max(real(temp_xs));
        temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
        end
        final_pos = rot*[temp_x;temp_y];
        pos_after(1,j,l+1) = final_pos(1);
        pos_after(2,j,l+1) = final_pos(2);
        end
        else
        temp_pos=rot\[pos_after(1,j,l);pos_after(2,j,l)];
        temp_x0=temp_pos(1);
        temp_y0=temp_pos(2);
        if temp_x0>1.5+(len_b)^2/3&&temp_x0<=4.5
        syms sym_x;
        eqn = ((2*temp_x0-6)^2+4*temp_y0^2)*sym_x^2 +
            (-24*temp_y0^2+(4*temp_x0-12)*(len_b^2-temp_x0^2-temp_y0^2+6.75))*sym_x
            + 27*temp_y0^2+(len_b^2-temp_x0^2-temp_y0^2+6.75)^2 == 0;
        sym_x = solve(eqn,sym_x);
        temp_xs = double(sym_x);
        temp_x = min(real(temp_xs));
        temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
        if abs(sqrt((temp_x-temp_x0)^2+(temp_y-temp_y0)^2) - len_b) > 0.01
        temp_x = max(real(temp_xs));
        temp_y = -sqrt(6*temp_x-(temp_x)^2-6.75);
        end
        final_pos = rot*[temp_x;temp_y];
        pos_after(1,j,l+1) = final_pos(1);
        pos_after(2,j,l+1) = final_pos(2);
        elseif temp_x0>-4.5+(len_b)^2/6 && temp_x0<=1.5+(len_b)^2/3
        syms sym_x;
        eqn = ((2*temp_x0+3)^2+4*temp_y0^2)*sym_x^2 +
            (12*temp_y0^2-(4*temp_x0+6)*(6.75+temp_x0^2+temp_y0^2-len_b^2))*sym_x
            + (6.75+temp_x0^2+temp_y0^2-len_b^2)^2-27*temp_y0^2 == 0;
        sym_x = solve(eqn,sym_x);
```

附录2: 问题4和问题5代码(q4_q5.m)

```
            temp_xs = double(sym_x);
            temp_x = min(real(temp_xs));
            temp_y = sqrt(-(temp_x)^2-3*temp_x+6.75);
            if abs(sqrt((temp_x-temp_x0)^2+(temp_y-temp_y0)^2) - len_b) > 0.01
            temp_x = max(real(temp_xs));
            temp_y = sqrt(-(temp_x)^2-3*temp_x+6.75);
            end

            final_pos = rot*[temp_x;temp_y];
            pos_after(1,j,l+1) = final_pos(1);
            pos_after(2,j,l+1) = final_pos(2);
            else
            func_cl = @(temp_theta)
                k^2*temp_theta^2-2*k*temp_theta*(pos_after(1,j,l)*cos(temp_theta)+
            pos_after(2,j,l)*sin(temp_theta))+(pos_after(1,j,l))^2+(pos_after(2,j,l))^2-
            len_b^2;
            temp_theta = fsolve(func_cl,5.5*pi);
            pos_after(1,j,l+1) = k*temp_theta*cos(temp_theta);
            pos_after(2,j,l+1) = k*temp_theta*sin(temp_theta);
            pos_state(j,l+1) = -1;
            theta(j,l+1) = temp_theta;
            end
            end
            rou(j,l+1) = sqrt((pos_after(1,j,l+1))^2+(pos_after(2,j,l+1))^2);
            end
            end
```

附录2: 问题4和问题5代码(q4_q5.m)

```
for j=1:224
v(:,j)=sqrt((pos_before(1,:,j)-pos_after(1,:,j)).^2+
(pos_before(2,:,j)-pos_after(2,:,j)).^2)/dt;
end

out_pos = zeros(201,448);
out_v = zeros(201,224);
for m=1:224
out_pos(:,2*m-1) = pos(1,:,m);
out_pos(:,2*m) = pos(2,:,m);
end

%xlswrite('result4.xlsx',round(out_pos.',6)位置,'','B2:GT449');
%xlswrite('result4.xlsx',round(v.',6)速度,'','B2:GT225');

% 计算最大允许速度和实际最大速度
v_max_h = 2/max(v(:));
max_v = max(v(:))
```