

# “同心协力”策略研究

## 摘要

本题在于考察分析同心球项目在不同情境下，通过调整团队队员发力的时机及力度，找出团队协作的最佳策略，使排球在鼓面上能持续稳定颠起。

- **针对问题一分析** 首先对于人数的一般值进行分析，确定绳长及人相对同心鼓的最佳站位和分布。我们将一次完整的颠球过程（鼓从初始位置出发到回到初始位置的过程）分为三个阶段，建立竖直平面内的运动模型。以排球和同心鼓的理想运动过程是稳定周期运动为切入点，对于周期内各个阶段的位移、时间关系设列运动关系方程。然后根据对同心鼓建立空间受力分析模型，使用得到的牛顿第二定律方程与前面的运动关系方程联立，得到各物理量值。计算出在最佳策略下，每位队员只需施加的大小为  $5.83214N$  的力，使鼓与排球碰撞，在碰撞结束  $0.0903086s$  后，施加  $0.166724s$  的大小为  $6.65350N$  的力，使鼓回到原位置，完成一次周期。
- **针对问题二分析** 该问题是一个旋转运动问题。首先需要对鼓和绳子进行几何分析，找出每个固定点的倾斜角度与鼓的倾斜角度，以及固定点的倾斜角度与此处牵拉绳子的旋转角度之间的关系。由于鼓是一个旋转对称体，所以可将八个固定点分成四组，每两个对称固定点为一组，建立力矩与转动惯量模型。分析序号 1 到序号 9 的施力力度及时机，找出每种情况对应的旋转主轴。再利用 MATLAB 编程，改变不同的参数，求出不同时刻鼓的倾斜角度。
- **针对问题三分析** 该问题是基于问题二对于问题一的理想状态进行调整，首先将问题二中产生的误差类型分为三类：1. 发力时间提前 2. 发力大小误差 3. 发力时间提前和大小误差同时出现。对于前两者，我们根据对称性关系相应提出了补偿性不发力或对称力增大，以及补偿性力减小的调整策略，对于第三种情况则采取两种调整方式综合使用的方法。最后我们讨论了现实情况下，人的用力大小和时机相对于理想值有一个正态分布的扰动，得到同心鼓鼓面  $0.5^\circ$  的倾斜角是不可避免的结论。
- **针对问题四分析** 根据碰撞的对称性，我们需要在将鼓面从  $-0.5^\circ$  倾斜至  $0.5^\circ$ ，根据到再次碰撞时的排球运动时间与鼓面完成自由落体和加速度向上的运动时间相同，联立鼓面竖直方向的运动方程，可以得到两个运动阶段分别的时间。在加速度向上的过程中，最高效的力量作用方式是增大与转轴角度最大的一对力，来控制鼓面的倾斜角度。最后运用计算机对力度的大小进行小步长的搜索，对每一个力度模拟鼓面的倾斜角度变化，来得到可行解。

**关键词** 运动方程 MATLAB 同心鼓

# 1 问题重述

## 1.1 问题背景

“同心协力”是一项需要团队高度协作的项目，不仅考察队员之间的默契程度，还锻炼了队员在压力环境中的心理承受能力，近年来受到了人们的热烈追捧。该项目的道具是一面在鼓身沿圆周均匀固定了多根绳子的牛皮双面鼓，以及一个富有弹性、气密性好的专用排球。项目开始时，团队成员每人牵拉一根绳子的末端，使鼓面保持水平，然后将球从鼓面正中心放下，队员找准时机同时用力将球颠起，使其能在鼓面上持续跳动而不落地，最终颠球次数最多的队伍获胜。

## 1.2 问题的提出（题目重述）

项目的目标是使球连续颠起次数尽可能多。现给定相关参数，包括所用双面鼓及排球的质量，鼓面直径，鼓身高度。同时要求团队队员人数不少于 8 人，各队员之间最小距离不小于  $60\text{cm}$ 。排球最开始从鼓面中心正上方  $40\text{cm}$  处竖直下落，之后被颠起高度离鼓面距离应不小于  $40\text{cm}$ ，否则项目终止。试通过数学建模，给出最佳策略。

- (1) 假设每个队员都可以精确控制用力方向、时机和力度，讨论在此理想状态下团队的最佳协作策略，并给出该策略下的颠球高度。
- (2) 现实情况下，由于各队员之间用力不同，鼓面会出现一定程度的倾斜。试通过给定的参考数据，建立模型考察队员的发力时机和力度与某一特定时刻的鼓面倾斜角度关系。
- (3) 根据问题 2 中考虑了现实情况的模型，对问题 1 的策略做出适当调整。
- (4) 实际中，当鼓面发生倾斜时，球的跳动方向也不再竖直。现给定相关数据，计算出在可精确控制条件下所有队员的发力时机及力度，使球调整为竖直状态弹跳，并分析在现实情况中这种调整策略的实施效果。

# 2 模型假设

1. 假设每根绳子均为柔性绳，每根绳子长度相同。
2. 不考虑风力以及空气阻力对排球运动的影响。
3. 当地重力加速度为  $9.8\text{kg}/\text{m}^2$ 。
4. 排球是一个直径为  $204\text{mm}$  的五号标准排球。
5. 将排球视为一个均匀球壳。
6. 不考虑鼓在运动过程中绳子的角度变化。
7. 假设每次碰撞都在同一水平高度。
8. 每位队员所隔距离相同，且手离地面高度相同（假定为  $1.2\text{m}$ ），即各力作用位置沿鼓均匀分布。

### 3 符号说明

表1列出了本文需要的符号。

表 1: 符号说明

符号	符号描述	符号	符号描述
$m_0$	鼓的质量	$m_1$	排球质量
$l$	绳子长度	$h$	双面鼓底面距离地面高度
$r$	鼓面半径	$F_i$	第 $i$ 位队员施加力的大小
$\varphi$	绳子与水平方向夹角	$x_{10}$	球从最高处落下到碰撞时的高度差
$\gamma$	鼓面半径之间夹角	$x_{00}$	鼓从最低位置到碰撞时的高度差
$\theta$	绳子改变角度	$x_{01}$	鼓自由落体的高度差
$\alpha$	鼓倾斜后与水平方向夹角	$x_{02}$	鼓减速回到最低位置的高度差
$v_{00}$	碰撞前鼓的速度	$v_{02}$	鼓自由落体结束时的高度差
$v_{01}$	碰撞后鼓的速度	$a_2$	鼓减速下降时的加速度
$v_{10}$	碰撞前球的速度	$a_1$	鼓加速上升时的加速度
$v_{11}$	碰撞后球的速度		

### 4 问题分析

#### 4.1 问题 1 分析

题一是一个循环运动过程分析问题，以鼓和球的运动周期相同为切入点。因为我们考虑绳子为柔性绳，那么施加力的方向沿绳方向。理想情况下，我们可认为每位队员同时施加大小相等的力。设在鼓减速下降阶段，每位队员施加力的大小为  $F_1$ ，在加速上升阶段，每位队员施加  $F_2$  的力。假设在初始条件下鼓处于最低位置，记为  $s_0$ 。将一次颠球过程分为三个阶段，步骤如下：

1. 在第一阶段每位队员施加竖直向上的力  $F_1$  使鼓向上运动，到达  $s_1$  与排球碰撞，此阶段鼓上升高度为  $x_{00}$ ，球下降的高度为  $x_{10}$ ，两者运动时间相等。在碰撞前鼓有一个竖直向上的速度  $v_{00}$ ，球有一个竖直向下的速度  $v_{10}$ 。我们考虑两者碰撞过程在瞬间完成，碰撞后鼓速度记为  $v_{01}$ ，球速度记为  $v_{11}$ ，方向均为竖直向上。
2. 在第二阶段，碰撞后鼓做自由落体运动，运动方向为先上升后下降，到达  $s_2$ ，此时鼓速度为  $v_{02}$ ，与碰撞处的高度差为  $x_{01}$ ，而排球继续上升。
3. 在第三阶段，队员施加  $F_2$  的力使鼓做减速运动从  $s_2$  回到最低位置，同时排球到达最高处，该阶段结束时鼓与排球的速度都为 0。

运动分析图如图 (1) 所示：

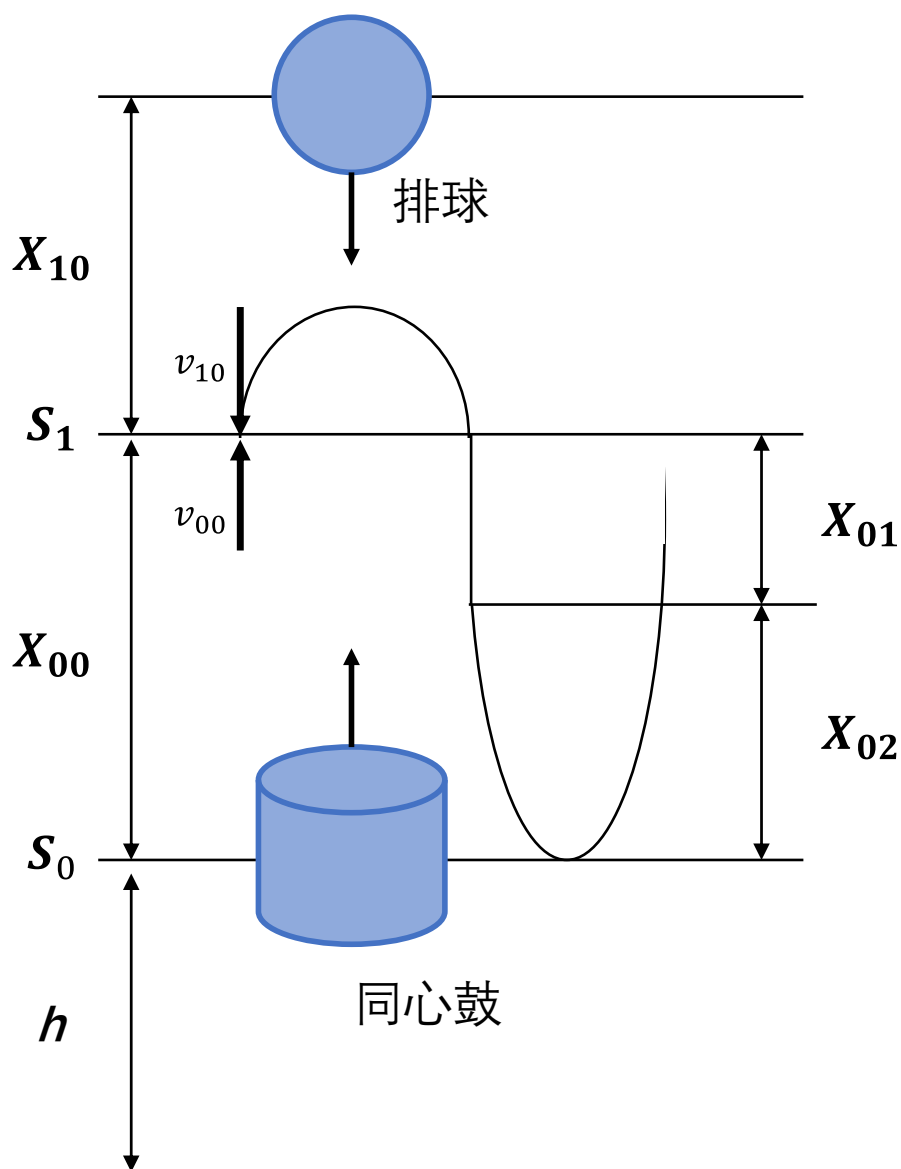


图 1: 问题一运动过程示意图 (未考虑尺寸比例)

## 4.2 问题 2 分析

题二是一个旋转运动过程研究问题，首先通过空间分析，找出旋转角度之间的关系，然后建立转动模型。分析步骤如下：

1. 任意选取鼓面上两条半径  $AD, AH$ ，长度为  $r$ ，记它们之间夹角为  $\gamma$ ，如图 (2) 所示。

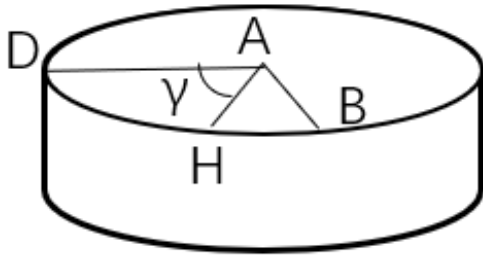


图 2: 鼓面倾斜示意图 (一)

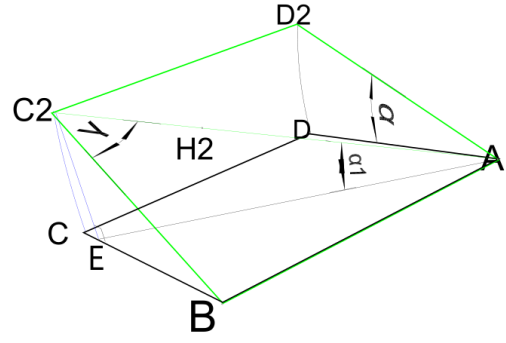


图 3: 鼓面倾斜示意图 (二)

假设鼓面沿着与半径  $AD$  垂直方向的线  $AB$  为旋转轴倾斜, 鼓平面从平面  $ABCD$  转到了平面  $ABC_2D_2$  (将两平面均看成长方形分析), 旋转角为  $\alpha$ , 如图 (3) 所示。  $AC$  为  $AH$  所在半径的延长线, 则  $AC$  与  $CB$  之间夹角为  $\gamma$ ,  $AC_2$  与  $C_2B$  之间夹角为  $\gamma$ 。点  $E$  为点  $C_2$  在平面  $ABCD$  的投影, 即  $C_2E$  垂直于  $CB$ , 角  $\alpha_1$  为倾斜后直线  $AC_2$  与原平面所夹角。则有:

$$AC_2 = \frac{C_2B}{\cos\gamma}$$

$$C_2E = C_2B \sin\alpha$$

$$C_2E = AC_2 \sin\alpha_1$$

由以上三式可得出  $\alpha_1$  与  $\alpha$ 、 $\gamma$  之间关系为:

$$\sin\alpha_1 = \sin\alpha \cos\gamma \quad (1)$$

2. 当鼓身上某一绳子的固定端与圆心的连线倾斜了  $\alpha$  角时, 我们记该条绳角度倾斜了  $\theta$  角, 如下图 (4) 所示。因为  $\alpha$  与  $\theta$  很小, 所以我们近似可得:

$$l\theta = r\alpha$$

不失一般性, 我们可推广得到:

$$l\theta_i = r\alpha_i \quad (2)$$

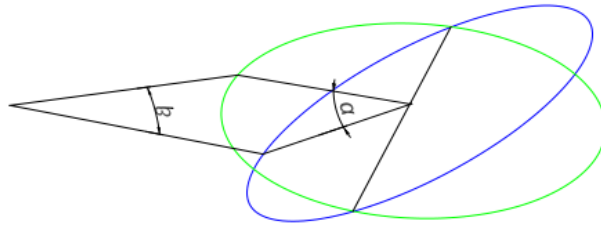


图 4: 绳子改变角度示意图

3. 由于鼓是一个旋转对称体, 所以可将八个固定点分成四组, 每两个对称固定点为一组。我们首先考察其中一个固定端。在鼓面为水平时, 绳子与水平方向夹角为  $\varphi$ , 当鼓面倾斜了  $\alpha$  角时, 绳子与水平方向角度改变  $\theta_1$ , 则该固定点处绳子与水平方向夹角变为  $(\varphi - \theta_1)$ , 绳子所施加力  $F_1$  的作用

点在平面  $AC_2E$  上点  $H_2$  的下方某处, 我们可将力分解为竖直方向和沿  $AC$  方向的力。竖直方向力的大小为  $F_1 \sin(\varphi - \theta_1)$ , 与直线  $AB$  距离为  $r \cos \gamma \cos \alpha$ ; 沿  $AC$  方向力的大小为  $F_1 \cos(\varphi - \theta_1)$ , 再将其分解为与  $AB$  直线和与  $AD$  直线平行的两个力。  $F_{AD}$  大小为  $F_1 \cos(\varphi - \theta_1) \sin \eta$ , 与直线  $AB$  距离为  $r \sin \alpha_1$ ; 因为  $F_{AB}$  与  $AB$  平行, 所以在鼓倾斜时对力矩不会产生影响; 而关于该固定点关于圆心中心对称处的另一固定端处, 绳子与竖直方向夹角变为  $(\varphi + \theta_1)$ , 设该处力的大小为  $F_2$ , 所能产生影响的两分解力大小分别为  $F_2 \sin(\varphi + \theta_1)$ ,  $F_2 \cos(\varphi + \theta_1) \sin \eta$ , 与直线  $AB$  距离分别为  $r \cos \gamma \cos \alpha$ ,  $r \sin \alpha_1$ , 建立出转动方程。结合每组的转动方程, 可得到关于鼓的转动模型。

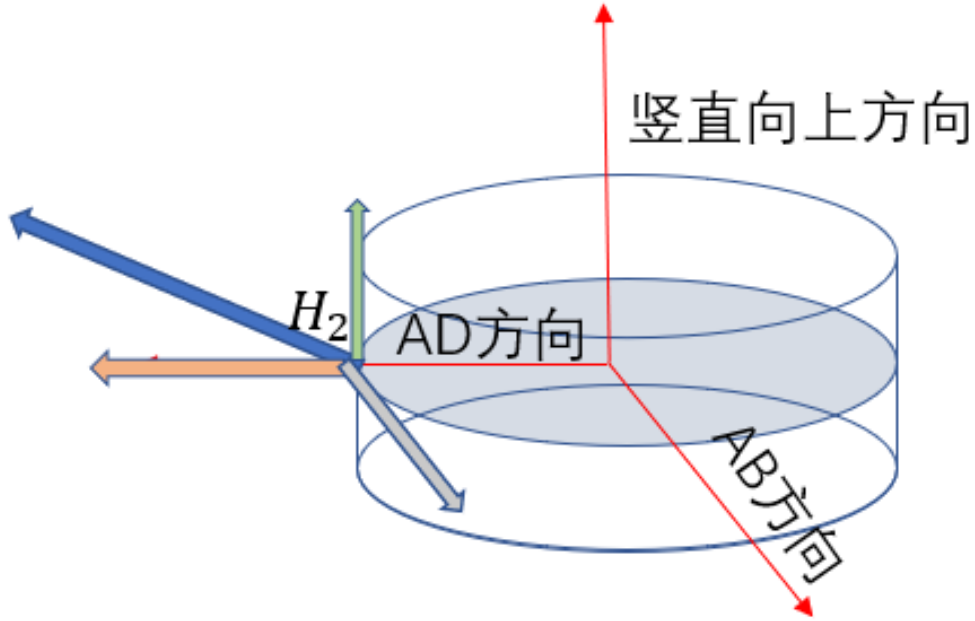


图 5: 双面鼓受力示意图

4. 分析题目给定数据, 选取旋转主轴: 某一时刻的旋转主轴方向, 即为与此时刻合力方向垂直的鼓面直径所在方向。如图 (8) 所示。在某时刻鼓上受到八个力的作用, 根据牛顿第一定律, 我们可对这八个力进行抵消,  $F_1$  和  $F_2$  为抵消后残余的力,  $F$  为两残余力的合力, 它在平面  $P$  内, 则此时的旋转主轴方向就为与平面  $P$  垂直的  $ab$  直径所在方向。

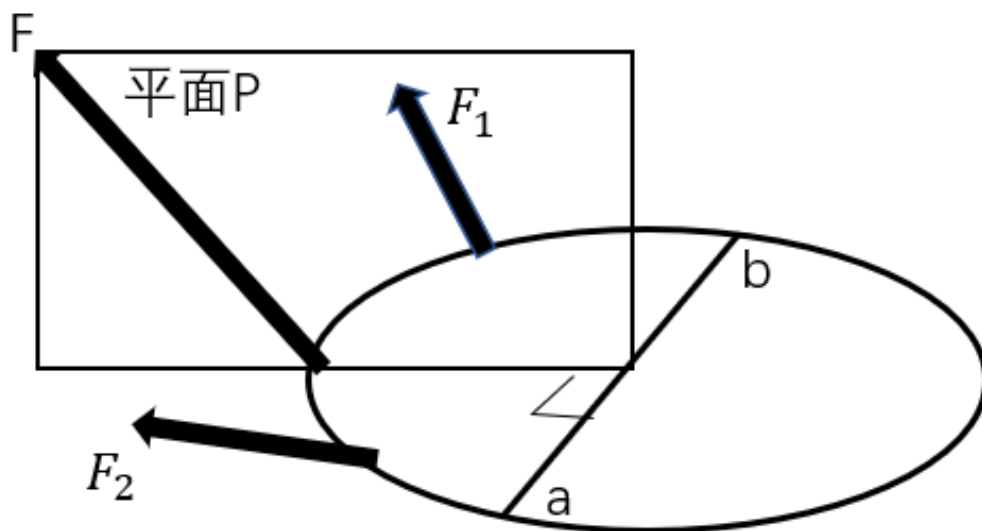


图 6: 旋转主轴示意图

5. 对序号 1 到序号 9 给定的序号进行分析, 我们发现 9 种情况可分为两种类型: 对于序号 1 到序号 7, 以及序号 9, 旋转主轴只有一个, 旋转主轴可由上述方法得到。而序号 8 在  $-0.1s$  和  $0s$  分别有两条不同的旋转主轴。我们首先通过 MATLAB 对另外八题的鼓面倾角求解, 发现在同一时刻由于几位队员多施加了  $10N$  的力而引起的鼓面倾角变化, 相对于队员由于施力时机提早了  $0.1$  而引起的鼓面倾角变化来说, 可以忽略不计, 所以我们仍可认为序号 8 的旋转主轴方向始终为  $-0.1s$  的旋转主轴方向。

### 4.3 问题 3 分析

通过查阅治疗, 人的反应时间为  $0.2s$ 。根据问题二所给数据, 结合现实情况, 我们将模型分为三种类型讨论, 通过 MATLAB 进行模拟分析。其中,  $F$ 、 $F_1$ 、 $F_2$  关系式为:  $F_1 + F_2 = 2F$ ,  $t_1$ 、 $t_2$ 、 $t_3$  关系式为:  $t_3 - t_2 = t_1$ 。

- **类型一**  $n$  位队员发力大小相同, 一位队员提早发力  $t_1$ 。针对此种情况, 我们给出两种解决方案: 方案一该队员在  $0.2s$  后意识到自己早发力, 于是第  $t_2$  到第  $t_3$  期间选择不发力; 方案二, 对面队员在  $0.2s$  后意识到队友的错误, 于是选择在第  $t_2$  到第  $t_3$  期间增大自己的发力大小来弥补错误。
- **类型二**  $n$  位队员同时在第  $0s$  发力  $F_1$ , 一位队员由于未掌控力度, 发力大小为  $F$ 。给出解决方案: 该队员在  $0.2s$  后意识到自己的错误, 选择在第  $0.2s$  到第  $0.4s$  减少自己的发力, 大小为  $F_2$ 。
- **类型三**  $(n-1)$  位队员在第  $0s$  同时发力  $F$ , 一位队员早发力  $t_1$ , 且发力力度为  $F_1$ 。给出解方案: 该队员在发力  $0.2s$  后将自己的施力力度变为  $F_2$ , 同时其对面队员意识到队友错误增大自己的施力力度。

### 4.4 问题 4 分析

结合问题 (1)、(2)、(3) 三题的分析解答过程, 综合对问题四进行分析。

1. 我们首先进行运动过程分析。因为鼓对排球的作用力始终是垂直鼓面向上的, 所以当排球反弹时相对竖直方向产生  $1^\circ$  倾斜角, 说明鼓与排球碰撞时的运动方向夹  $0.5^\circ$  角, 则鼓相对水平面的倾

斜角度为  $0.5^\circ$ ，我们记此时倾角为  $-0.5^\circ$ 。所以若想要在下一次碰撞时将排球调整为竖直状态弹跳，则应该使排球倾斜方向变为  $+0.5^\circ$ 。如图 (7) 所示。

2. 对排球在空中的运动进行分析，可计算得到两次碰撞的间隔时间。通过问题一的分析，我们知道在此次碰撞后到下一次碰撞前，鼓经历自由落体运动，减速下降，以及加速上升三个阶段。在此题中为简化问题，我们做如下假设：

- (a) 因为我们无需考虑排球上升高度，所以可假定减速加速两过程所施加的力是一样的。那么在变速运动阶段，鼓由于受力不均而发生倾斜，由于我们认为在此过程中每个人施力情况不变，那么鼓做匀速翻转。我们想要使鼓的倾斜角度从  $-0.5^\circ$  变为  $+0.5^\circ$ ，所以当鼓水平时刚好为变速运动中间时刻。
- (b) 由于碰撞时鼓的速度不可知，我们可假定碰撞后鼓立即以速度  $0m/s$  自由落体。通过对排球运动时间分析，可计算得到队员作用力的时机、作用时间。
- (c) 假定鼓面初始位置较绳子水平时仍下降  $11cm$ 。

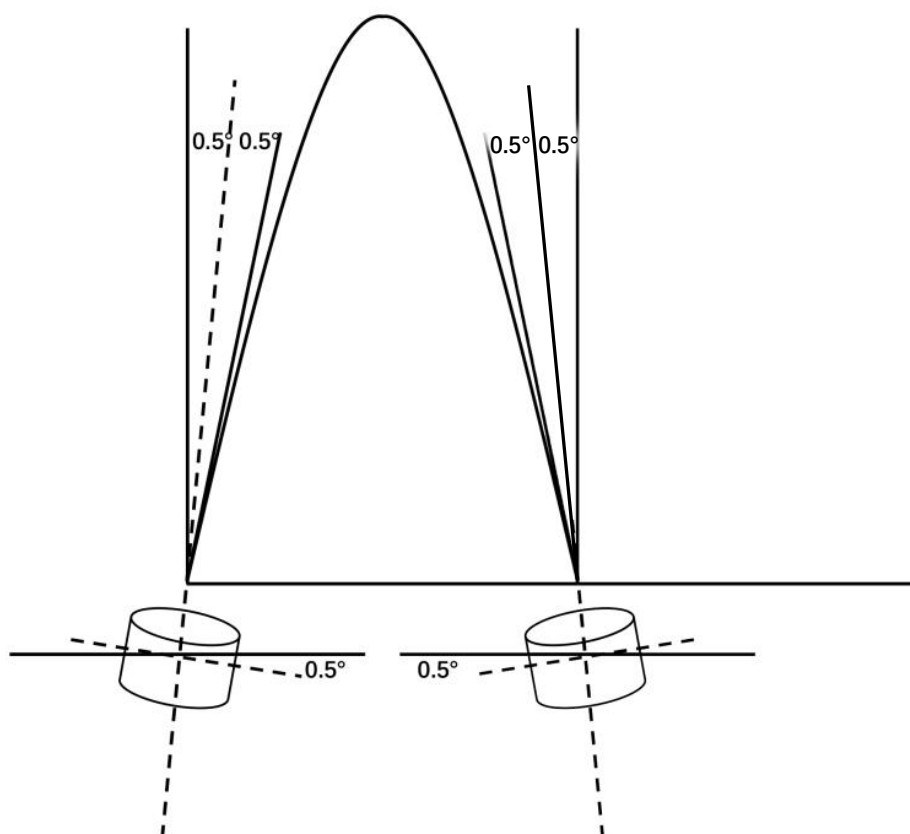


图 7: 旋转主轴示意图



3. 开始时，我们假设 10 位队员每人施加力的时机和力度都相同，通过分析计算可发现在这种情况下，第二次碰撞时鼓的倾斜角度并不为  $0.5^\circ$ 。然后在保持原有的合力大小不变的基础上，逐渐增加几位队员施加力，而减少其余队员施加力，利用 MATLAB 对该过程进行逐步模拟分析，直至找出最终使鼓倾斜角度为  $0.5^\circ$  时，每位队员施加力的情况。
4. 模拟在现实情况下，由于队员的用力时机和大小都无法完全遵从理想值，我们在理想值的基础上，对施力时间和大小上都加上了一个正态分布的扰动，并且多次进行实验模拟随机情况下纪律扰动带来的角度变化（如图）。由图可以明显看出，绝大部分倾斜角度的振幅都大于  $0.5^\circ$ ，而现实情况下补救措施中发力的误差也不可避免，极大概率也会带来超过  $0.5^\circ$  的扰动，这说明鼓面  $0.5^\circ$  的倾斜在现实情况下是不可避免的。

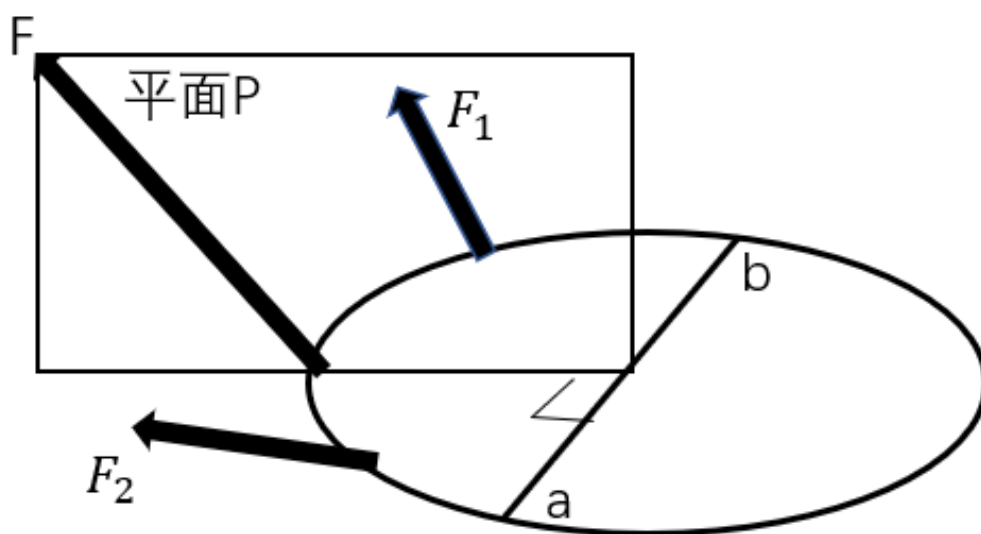


图 8: 旋转主轴示意图

## 4.5 问题 1 的解答

### 1. 绳长计算

我们假设团队队员共有  $n$  人，每人之间的距离为最小距离  $60cm$ 。队员牵拉绳子的水平投影线的延长线交于绳子固定端所在圆的圆心  $A$ ，则该圆心与两队员队员所站位置可构成水平的等腰三角形，三角形顶角为  $\frac{360^\circ}{n}$ ，三角形底角角度为  $(90^\circ - \frac{180^\circ}{n})$ ，如下图所示：

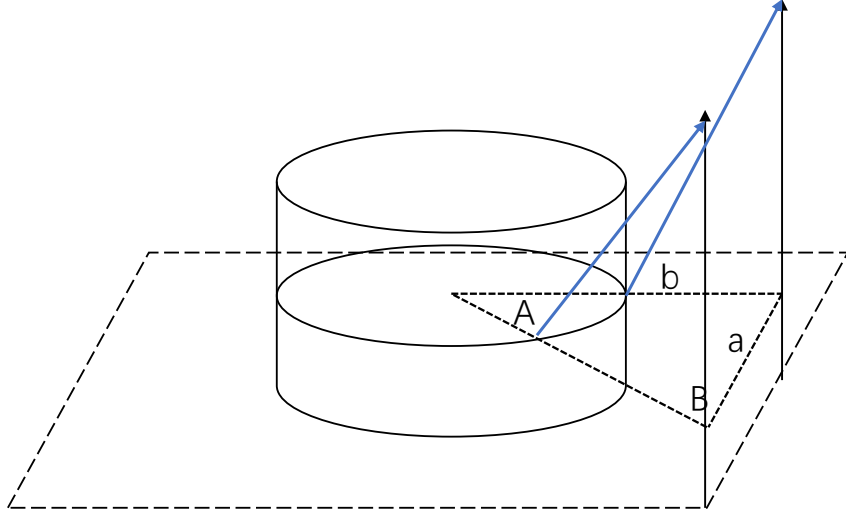


图 9: 绳长计算示意图

由正弦定理可得:

$$\frac{a}{\sin A} = \frac{b}{\sin B}$$

即:

$$\frac{0.6}{\sin(\frac{360^\circ}{n})} = \frac{b}{\sin(90^\circ - \frac{180^\circ}{n})}$$

解得  $b$  的长度为:

$$b = \frac{0.3}{\sin(\frac{180^\circ}{n})}$$

又因为绳子与水平方向夹角为  $\varphi$ , 所以绳子长度可以表示为:

$$l = \frac{b - r}{\cos \varphi} = \frac{\frac{0.3}{\sin(\frac{180^\circ}{n})} - r}{\cos \varphi}$$

在此题中, 假设  $n = 8$  人。所以有:

$$l = \frac{0.3\sqrt{2} - 0.2}{\cos \varphi} \quad (3)$$

在最佳策略下每人施加力度应尽可能小, 那么应尽量减小绳子与竖直方向夹角, 所以此时设定  $h = 0$ , 即  $s_0$  在水平地面上。手的作用高度与绳子在鼓身固定端的高度差为  $1.2 - (\frac{0.22}{2}) = 1.09m$ 。

$$\sin \varphi = \frac{1.09}{l} \quad (4)$$

结合式子 (3)、(4), 计算得到绳子长度为  $l = 1.16661m$ 。

## 2. 运动计算

在最佳协作策略下, 每位队员都想要使自己施加的力尽可能小, 持续时间尽可能短。这样我们可以假设排球所能到达的最高高度恰好为  $40cm$ 。在第一阶段中, 排球从最高处自由落到碰撞位置的时间与鼓从  $s_0$  上升到碰撞位置的时间相等, 可列公式:

$$\frac{v_{00}}{a_1} = \frac{v_{10}}{g} \quad (5)$$

且有  $x_{00} + x_{10} = 0.4$ , 即:

$$\frac{v_{10}^2}{2g} + \frac{v_{00}^2}{2a_1} = 0.4 \quad (6)$$

在二、三、阶段中, 排球从碰撞位置回到最高点的时间与鼓从碰撞位置回到  $s_0$  的时间相等, 可列公式:

$$\frac{v_{11}}{g} = \frac{v_{02} + v_{01}}{g} + \frac{v_{02}}{a_2} \quad (7)$$

且有  $x_{10} + x_{01} + x_{02} = 0.4$ , 则:

$$\frac{v_{11}^2}{2g} + \frac{(v_{02}^2 - v_{01}^2)}{g} + \frac{v_{02}^2}{a_2} = 0.4 \quad (8)$$

排球做自由落体运动, 在同一高度时速度大小相同, 则有:

$$v_{11} = v_{10} \quad (9)$$

碰撞时, 由动量守恒可列公式:

$$m_0 v_{00} - m_1 v_{11} = m_0 v_{01} + m_1 v_{11} \quad (10)$$

通过查阅相关资料, 我们设定碰撞恢复系数大小为 0.74, 所列公式为:

$$e = \frac{v_{11} - v_{01}}{v_{00} + v_{11}} \quad (11)$$

### 3. 计算求解

结合方程 (5) 到方程 (11), 利用 MATLAB 求解, 得到结果:

$$v_{00} = 0.593537m/s$$

$$v_{01} = 0.2157m/s$$

$$v_{02} = 0.669324m/s$$

$$v_{11} = v_{10} = 2.51891m/s$$

$$F_1 = 5.83214N$$

$$F_2 = 6.65350N$$

算得第一阶段时间为  $0.257032s$ , 第二阶段时间为  $0.0903086s$ , 第三阶段时间为  $0.166724s$ 。所以最佳策略为当球从最高处下落时, 每位队员施加  $5.83214N$  的力, 持续时间为  $0.257032s$ , 与排球碰撞, 使排球弹起, 最高点仍能回到  $40cm$ , 碰撞  $0.0903086s$  后, 每位队员施加  $6.65350N$  大小的力, 持续  $0.166724s$ , 使鼓减速回到  $s_0$ 。

## 4.6 问题 2 的解答

在问题二中,  $\sin\varphi$  计算结果为  $0.0647$ 。

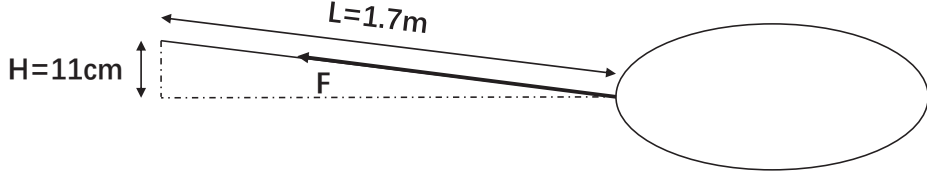


图 10: 绳长计算示意图

通过对问题二的分析，我们可将八个固定端分成四组，每两个相对称的固定端为一组，某一固定端转动方程为：

$$J\beta_{1i} = F_{1i}\sin(\varphi - \theta_i)r\cos\gamma\cos\alpha - F_{1i}\cos(\varphi - \theta_i)r\sin\eta\sin\varphi_i \quad (12)$$

其对称处转动方程为：

$$J\beta_{2i} = F_{2i}\sin(\varphi + \theta_i)r\cos\gamma\cos\alpha + F_{2i}\cos(\varphi + \theta_i)r\sin\eta\sin\varphi_i \quad (13)$$

综合所有固定端，得到鼓的转动方程如下：

$$J\beta = \sum_{i=1}^4 (J\beta_{i1} - J\beta_{i2}) \quad (14)$$

结合公式 (1)、(2)、(12)、(13)、(14)，利用 MATLAB 进行小步长求解，得到结果如表2所示。由于初始时，每位队员施加力的力度和时机不同，我们假定在开始时由于施力不均而造成鼓面旋转的方向为正方向。

表 2: 鼓面倾斜角度结果

序号	用力参数	1	2	3	4	5	6	7	8	鼓面倾角（度）
1	发力时机	0	0	0	0	0	0	0	0	0.198926°
	用力大小	90	80	80	80	80	80	80	80	
2	发力时机	0	0	0	0	0	0	0	0	0.361033°
	用力大小	90	90	80	80	80	80	80	80	
3	发力时机	0	0	0	0	0	0	0	0	0.154996°
	用力大小	90	80	80	90	80	80	80	80	
4	发力时机	-0.1	0	0	0	0	0	0	0	-2.33284°
	用力大小	80	80	80	80	80	80	80	80	
5	发力时机	-0.1	-0.1	0	0	0	0	0	0	-3.90435°
	用力大小	80	80	80	80	80	80	80	80	
6	发力时机	-0.1	0	0	-0.1	0	0	0	0	-1.9634°
	用力大小	80	80	80	80	80	80	80	80	
7	发力时机	-0.1	0	0	0	0	0	0	0	-2.49741°
	用力大小	90	80	80	80	80	80	80	80	
8	发力时机	0	-0.1	0	0	-0.1	0	0	0	-1.95983°
	用力大小	90	80	80	90	80	80	80	80	
9	发力时机	0	0	0	0	-0.1	0	0	-0.1	-2.14966°
	用力大小	90	80	80	90	80	80	80	80	

## 4.7 问题 3 解答

针对类型一、二、三给出的策略，利用 MATLAB 模拟分析，有如下结果 (假定人数  $n = 8$ ):

### • 类型一策略一

对于类型一，所有队员发力大小相同 (令  $F_1 = 80N$ )，但某一队员提早  $t_1$  发力。给出策略：该队员在反应了  $0.2s$  后意识到自己的错误，于是选择在  $0.2s$  后持续  $t_1$  不发力。假定该队员提早发力  $0.1s$ ，于是在第  $0.1s$  到第  $0.3s$  期间选择不发力。通过 MATLAB 模拟分析，给出了在有策略 (不发力) 和无策略 (保持原有力) 两种情况下，鼓面倾斜角度变化，如图 (11) 所示。发现两种情况下鼓面倾斜角度无明显区别，该策略无效。

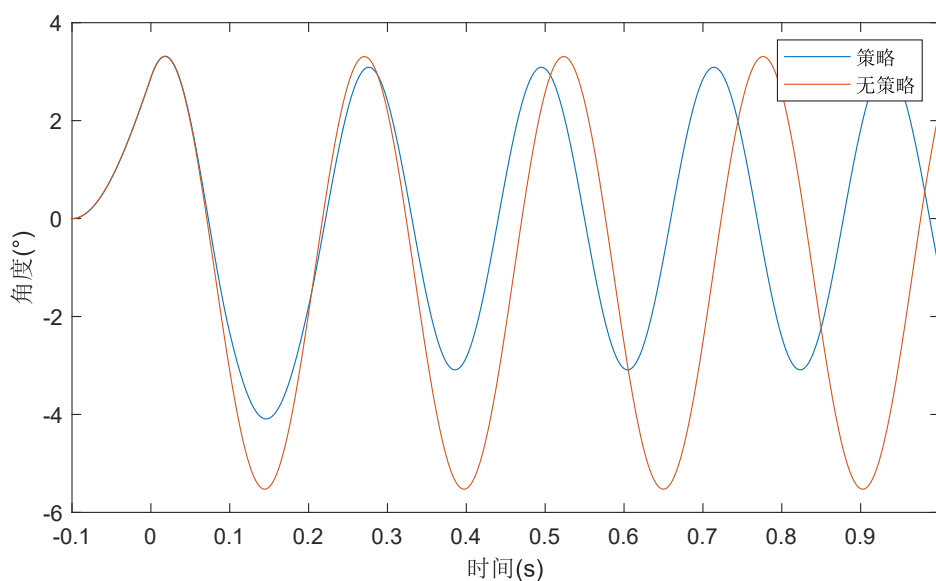


图 11: 类型一策略一结果图

### • 类型一策略二

对于类型一，所有队员发力大小相同 (令  $F_1 = 80N$ )，但某一队员提早发力。给出策略：对面队员在反应了  $0.2s$  后，通过改变自己施力大小弥补队友错误。我们假定一队员提早发力  $0.1s$ ，那么对面队友在第  $0.1s$  到第  $0.2s$  期间改变自己的用力。利用 MATLAB 对未实施策略、对面队友实施力改变为  $160N$ 、 $240N$ 、 $320N$ 、 $400N$  五种情况进行分析，得到如图 (12) 所示结果。发现当对面队友逐渐增加自己的实施力，鼓面倾斜程度波动范围逐渐变小， $400N$  时我们认为鼓面维持在水平。

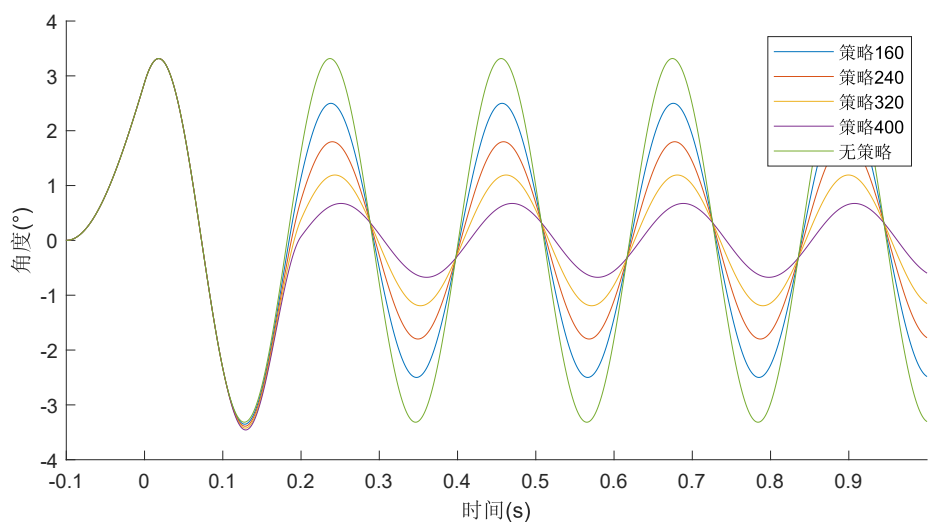


图 12: 类型一策略二结果图

#### • 类型二策略

对于类型二，所有队员同时发力，但某一队员未能掌控施力力度情况。给出策略：该队员在反应了  $0.2s$  后，改变自己的施力力度。令  $F_1 = 90N, F = 80N$  带入分析，通过 MATLAB 对实施策略和未实施策略两种情况进行分析，得到如图 (13) 所示结果。发现当未实施策略时，鼓面倾斜角度会以  $0.1^\circ$  的均值，在  $0^\circ$  到  $0.2^\circ$  之间进行大幅度波动。而当实施了策略时，鼓面倾斜角度在  $0^\circ$  附近进行小幅波动，说明可以使鼓面维持水平的效果更好，该策略有效。

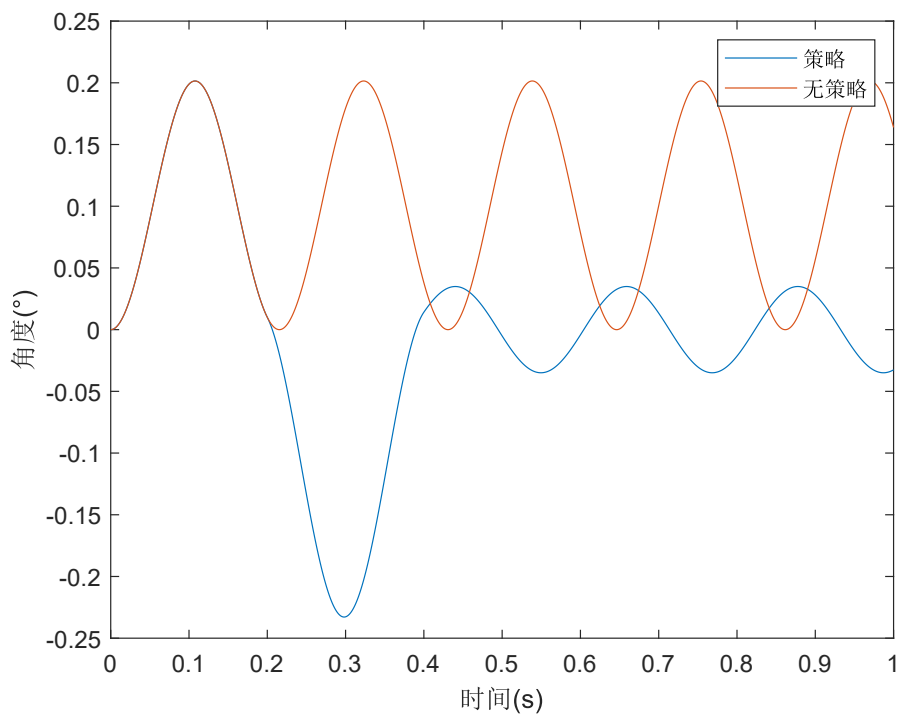


图 13: 类型二策略结果图

### • 类型三策略

对于类型三，7 位队员同时发力  $F$ ，但某一队员未能掌控发力时机及力度。给出策略：该队员在反应了  $0.2s$  后，改变自己的施力力度，同时其对面队友也改变自己的施力力度来弥补队友的错误。假定具体数据，一队员提早发力  $0.1s$ ，且施力力度为  $90N$ 。在第  $0.1s$  意识到错误，于是第  $0.1s$  到第  $0.3s$  将自己的施力改变为  $70N$ 。同时在第  $0.1s$  到第  $0.2s$  对面队员改变自己的施力力度（有类型一策略二可知该方法有效）。通过 MATLAB 对实施策略和未实施策略两种情况进行分析，同时将有策略情况分为对面队友分别实施  $160N$ 、 $240N$ 、 $320N$ 、 $400N$  五种情况，得到如图 (14) 所示结果。发现在实施了策略后，鼓面倾斜角度有一定减小，且对面队员施力大小越大效果越明显。

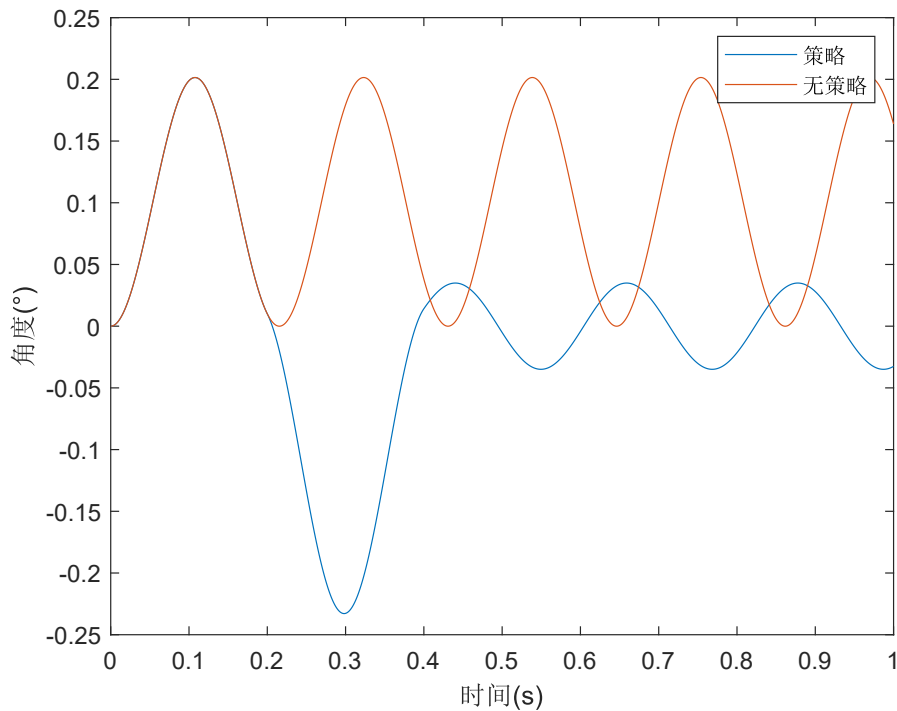


图 14: 类型三策略结果图

## 4.8 问题 4 解答

### 1. 排球运动分析

排球在空中做加速度恒为  $g = 9.8m/s$  的运动，且能达到的最高高度为  $h = 60cm$ ，所以根据运动方程：

$$\frac{1}{2}gt^2 = h$$

得到在空中运动时间为  $2t = 0.670s$ ，即两次碰撞间隔时间为  $0.670s$ 。

### 2. 鼓运动计算

我们假设鼓自由落体时间为  $t_1$ ，变速运动时间为  $t_2$ ，加速度为  $a$ ，由于自由落体与变速运动两阶段位移相等。所以有：

$$\frac{1}{2}gt^2 = -gt_1t_2 + \frac{1}{2}at_2^2 \quad (15)$$

两次碰撞间隔，鼓运动时间与排球运动时间相等，可得等式：

$$t_1 + t_2 = 2t \quad (16)$$

我们仍假设鼓面初始位置较绳子水平时下降  $11\text{cm}$ 。

$$a = \frac{F \sin \varphi - mg}{m} \quad (17)$$

综合式子 (15) 到 (17), 可计算得到  $t_1$  为  $0.1804\text{s}$ ,  $t_2$  为  $0.5193\text{s}$ 。

### 3. 排球受力分析

现在，我们对鼓的旋转主轴进行分析。现有十位队员参与比赛，我们分别标记为  $1, 2, 3, \dots, 10$ ，那么每两位队员所拉绳子的固定端与圆心的夹角为  $36^\circ$ 。根据题意，排球倾斜方向在水平面的投影指向某两位队员之间，且与这两位队员的夹角之比为  $1:2$ 。我们假设排球倾斜方向的投影落在队员 1 与队员 2 之间，与队员 1 的夹角为  $24^\circ$ ，与队员 2 的夹角为  $12^\circ$ 。图 (15) 为鼓面俯视图， $oc$  为排球倾斜方向投影，那么鼓的旋转主轴方向应与  $oc$  垂直，可找出过圆心的直线  $ab$  为旋转主轴方向。而由于离旋转主轴越远，对旋转的影响越大，所以我们增加队员 2、7 施加的力，减少其余队员的施加力，逐步分析直至第二次碰转角度为  $0.5^\circ$ 。

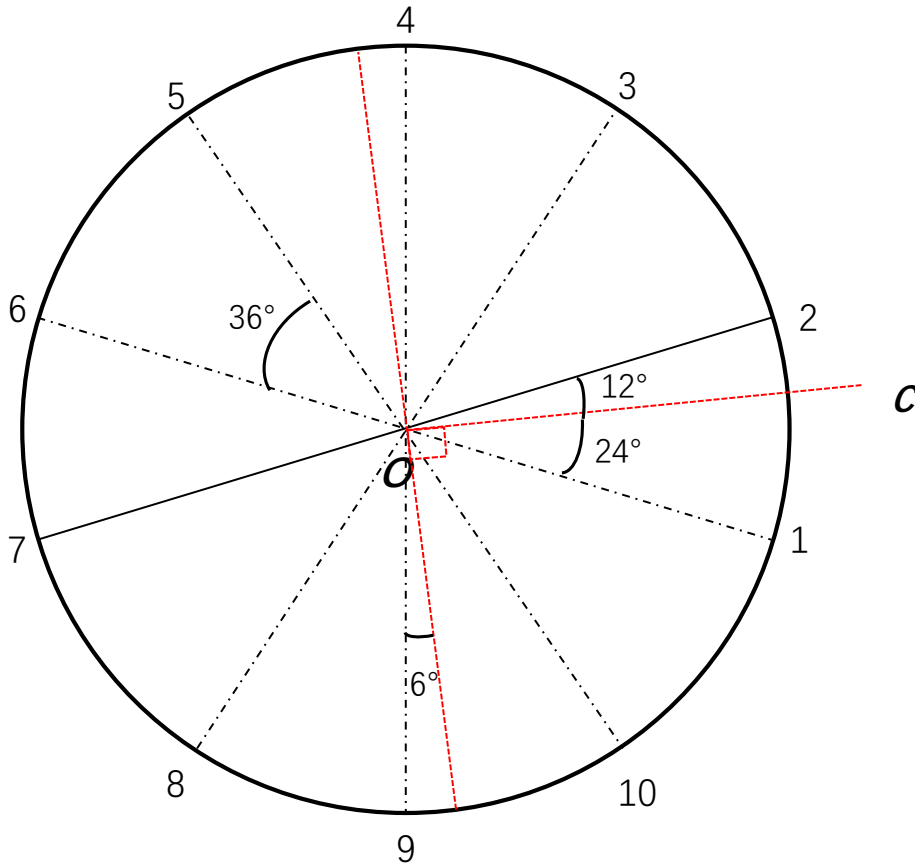


图 15: 鼓面俯视图

### 4. 扰动情况分析

- 为了保证鼓面不围绕其他转轴发生转动，并且使用最小的力达到转动鼓面的效果，将图 (15) 中 2、7 两对称点的力数值等比例增大，为了尽可能保证鼓面在竖直方向上的位移不受影响，假设所有队员用力的数值和为  $800\text{N}$ 。于是，可得下式：



然后对不同的  $F$  进行受力分析得到角度方程如下：

运用计算机进行模拟可以得到不同  $F$  下的倾斜角度，进行小步长搜索得到目标可行解如下。

- 现在考虑现实情况下的调整措施的可行性，考虑到现实情况下队员的发力时机和大小无法完全遵从理想值，故在发力时机和大小的理想值基础上加上正态分布的扰动，由  $3\sigma$  原则确定相应的  $\sigma$  值，得到发力时机和发力大小表达式如下：

根据上面的每一组发力时机和大小，运用计算机模拟鼓面的倾斜角度变化，得到多组实验图像如下：

由图可知，现实情况下队员用力误差产生鼓面倾斜的角度大多数情况下大于  $0.5^\circ$ ，鼓面倾斜的调整策略产生的误差大于  $0.5^\circ$ ，所以在现实情况下没有意义。

## 5 模型总结

### 5.1 模型优点

1. 在问题一中，我们在满足题设条件的情况下，提出了一个在理想条件下适用的物理模型，使得每位队员施加力的时间及力度可达到最小，不费精力。
2. 对于问题二的模型，我们运用空间几何分析，考虑了由于鼓面的倾斜导致各个绳子固定点的高度不同，进而导致绳子上作用力与转轴之间的角度变化，导致产生的力矩发生了变化，提升了分析的精度和准确性。
3. 对于问题三，我们对于队员产生误差的情况进行了分类，并且根据各个情况的具体情况运用对称性原理，提出了合理的调整策略，并且对比了使用策略与否的鼓面倾斜情况验证了策略的可行性。
4. 在问题四中，我们利用计算机对所有队员的力度进行小步长的搜索，来获取队员力度的可行解。并且通过模拟显示在现实生活中正态分布的误差，对鼓的倾斜角度变化情况进行模拟，结合结论说明了一定程度的倾斜在现实情况下是难以避免的。

### 5.2 模型缺点

1. 问题一模型：忽略了由于鼓面的高度变化带来的绳子作用力方向的变化引起的角加速度的变化。
2. 问题二模型：在题目第八组的分析中，我们忽略了与原转动方向相异的轻微转动带来的扰动在垂直转轴方向上产生的转动分量，导致结果相较于真实情况有一定的误差。
3. 问题三模型：未能考虑现实情况中多个成员发力时机与力度产生变化带来的误差的调整策略。
4. 问题四模型：未能考虑到高度变化带来的所有绳子作用力方向与竖直方向间的角度变化。

## 参考文献

[1]

[2]

## A 问题一求解代码

```

1  clc
2  clear
3  syms n h step F1 F2 v00 v11 v01 v02 h11 n
4  % e 为鼓与球碰撞的恢复系数。
5  e = 0.74; m0 = 3.6; m1 = 0.27; g = 9.8; d = 0.4; r = d / 2;
6  % h_min 为球颠起后需要距离鼓的最小距离40cm。
7  h_min = 0.4;
8  n = input('Please input the number of students:');
9  h = input('Please input the height of the hand:');
10 % h0 为鼓距离地面的距离。
11 h0 = 0.11;
12 % alpha 人与人之间的角度。
13 alpha = 360/n;
14 % 绳长
15 lx = sqrt(0.6^2 / (2 * (1 - cos(alpha)))) - r;
16 ly = h - h0;
17 l = sqrt(lx^2 + ly^2);
18
19 phi = asin(ly / l);
20 % a1 为鼓上升时候的加速度, a2 为鼓下降第二段的加速度
21 a1 = (n * F1 * sin(phi) - m0 * g)/m0;
22 a2 = (n * F2 * sin(phi) - m0 * g)/m0;
23
24 equ1 = m0 * v00 - m1 * v11 == m0 * v01 + m1 * v11;
25 equ2 = (v11 - v01) / (v00 + v11) == e;
26 equ3 = a1 * v11 == g * v00;
27 equ4 = v11/g == (v02 + v01)/g + v02/a2;
28 equ5 = v11^2/g + v00^2/a1 == 0.8;
29 equ6 = v11^2/g + (v02^2 - v01^2)/g + v02^2/a2 == 0.8;
30 equ = [equ1, equ2, equ3, equ4, equ5, equ6];
31 [F1, F2, v00, v01, v02, v11] = solve(equ, [F1, F2, v00, v01, v02, v11]);
32 fprintf('施加的力F1为: %.5f\n施加的力F2为: %.5f\n绳子长度为: %g\n',F1(2), F2(2), l);
33 fprintf('v00 = %g\nv01 = %g\nv02 = %g\nv11 = %g\n', v00(2), v01(2), v02(2), v11(2));
34 a1 = (n * F1(2) * sin(phi) - m0 * g)/m0;
35 a2 = (n * F2(2) * sin(phi) - m0 * g)/m0;
36 t1 = v00(2) / a1;
37 t2 = (v01(2) + v02(2)) / g;

```

```

38 t3 = v02(2) / a2;
39 fprintf('上升时间: %g\n下降第一段时间: %g\n下降第二段时间: %g\n', t1, t2, t3);

```

## B 问题二代码

### B.1 子问题一

```

1  clc
2  clear
3  d = 0.4; r = d/2; m = 3.6; h = 0.22;
4  F1 = 90; F2 = 80;
5  l = 1.7;
6  phi = asin(0.11/1.7); alpha = 0; omega = 0;
7  gamma1 = 0; gamma2 = pi / 4;
8  J = 1/2 * m * r^2 + 1/12 * m * h^2; %圆环的转动惯量。
9  step = 1000000;
10 t = 0.1 / step; % 0.1 秒的步长。
11
12 for i=1:step
13     alpha1 = asin(sin(alpha) * cos(gamma1));
14     alpha2 = asin(sin(alpha) * cos(gamma2));
15     Eta1 = acos(sin(gamma1) / cos(alpha1));
16     Eta2 = acos(sin(gamma2) / cos(alpha2));
17     theta1 = alpha1 * r / l;
18     theta2 = alpha2 * r / l;
19     % 转动方程求此时的角加速度。
20     beta = (r * (F1 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1)) - ...
21         F2 * (sin(phi + theta1) * cos(alpha) * cos(gamma1) + cos(phi + theta1) * sin(
        alpha1) * sin(Eta1))) + ...
22         2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2
        ) * sin(alpha2) * sin(Eta2) - ...
23         sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2
        ) * sin(Eta2))) / J;
24     alpha = alpha + omega * t + 1/2 * beta * t^2;
25     omega = omega + beta * t;
26 end
27 fprintf('鼓面倾角: %g°\n', rad2deg(alpha));

```

### B.2 子问题二

```

1  clc
2  clear
3  d = 0.4; r = d/2; m = 3.6; h = 0.22;
4  F1 = 90; F2 = 80;
5  l = 1.7;
6  phi = asin(0.11/1.7); alpha = 0; omega = 0;
7  gamma1 = pi / 8; gamma2 = 3 * pi / 8;
8  J = 1/2 * m * r^2 + 1/12 * m * h^2; %圆环的转动惯量。
9  step = 1000000;
10 t = 0.1 / step; % 0.1 秒的步长。
11
12 for i=1:step
13     alpha1 = asin(sin(alpha) * cos(gamma1));
14     alpha2 = asin(sin(alpha) * cos(gamma2));
15     Eta1 = acos(sin(gamma1) / cos(alpha1));
16     Eta2 = acos(sin(gamma2) / cos(alpha2));
17     theta1 = alpha1 * r / l;
18     theta2 = alpha2 * r / l;
19     % 转动方程求此时的角加速度。
20     beta = (2 * r * (F1 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1)) - ...
21         F2 * (sin(phi + theta1) * cos(alpha) * cos(gamma1) + cos(phi + theta1) * sin(
        alpha1) * sin(Eta1))) + ...
22         2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2
        ) * sin(alpha2) * sin(Eta2) - ...
23         sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2
        ) * sin(Eta2))) / J;
24     alpha = alpha + omega * t + 1/2 * beta * t^2;
25     omega = omega + beta * t;
26 end
27 fprintf('鼓面倾角: %g°\n', rad2deg(alpha));

```

### B.3 子问题三

```

1  clc
2  clear
3  d = 0.4; r = d/2; m = 3.6; h = 0.22;
4  F1 = 90; F2 = 80;
5  l = 1.7;
6  phi = asin(0.11/1.7); alpha = 0; omega = 0;
7  gamma2 = pi / 8; gamma1 = 3 * pi / 8;

```

```

8 J = 1/2 * m * r^2 + 1/12 * m * h^2; %圆环的转动惯量。
9 step = 1000000;
10 t = 0.1 / step; % 0.1 秒的步长。
11
12 for i=1:step
13     alpha1 = asin(sin(alpha) * cos(gamma1));
14     alpha2 = asin(sin(alpha) * cos(gamma2));
15     Eta1 = acos(sin(gamma1) / cos(alpha1));
16     Eta2 = acos(sin(gamma2) / cos(alpha2));
17     theta1 = alpha1 * r / l;
18     theta2 = alpha2 * r / l;
19     % 转动方程求此时的角加速度。
20     beta = (2 * r * (F1 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1)) - ...
21         F2 * (sin(phi + theta1) * cos(alpha) * cos(gamma1) + cos(phi + theta1) * sin(
        alpha1) * sin(Eta1))) + ...
22         2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2
        ) * sin(alpha2) * sin(Eta2) - ...
23         sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2
        ) * sin(Eta2))) / J;
24     alpha = alpha + omega * t + 1/2 * beta * t^2;
25     omega = omega + beta * t;
26 end
27 fprintf('鼓面倾角: %g°\n', rad2deg(alpha));

```

#### B.4 子问题四

```

1 clc
2 clear
3 d = 0.4; r = d/2; m = 3.6; h = 0.22;
4 F1 = 90; F2 = 80;
5 l = 1.7;
6 phi = asin(0.11/1.7); alpha = 0; omega = 0;
7 gamma1 = 0; gamma2 = pi / 4;
8 J = 1/2 * m * r^2 + 1/12 * m * h^2; %圆环的转动惯量。
9 step = 1000000;
10 t = 0.1 / step; % 0.1 秒的步长。
11
12 for i=1:step
13     alpha1 = asin(sin(alpha) * cos(gamma1));
14     Eta1 = acos(sin(gamma1) / cos(alpha1));

```

```

15     theta1 = alpha1 * r / l;
16     % 转动方程求此时的角加速度。
17     beta = r * F2 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi - theta1)
        ) * sin(alpha1) * sin(Eta1)) / J;
18     alpha = alpha + omega * t + 1/2 * beta * t^2;
19     omega = omega + beta * t;
20 end
21 fprintf('0.1s前的鼓面倾角: %g°\n', rad2deg(alpha));
22 for i=1:step
23     %     if mod(i,10) == 0
24     %         fprintf('%d\n', i/10);
25     %     end
26     alpha1 = asin(sin(alpha) * cos(gamma1));
27     alpha2 = asin(sin(alpha) * cos(gamma2));
28     Eta1 = acos(sin(gamma1) / cos(alpha1));
29     Eta2 = acos(sin(gamma2) / cos(alpha2));
30     theta1 = alpha1 * r / l;
31     theta2 = alpha2 * r / l;
32     % 转动方程求此时的角加速度。
33     beta = (F2 * r * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1) - ...
34         sin(phi + theta1) * cos(alpha) * cos(gamma1) - cos(phi + theta1) * sin(alpha1)
        ) * sin(Eta1)) + ...
35         2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2)
        ) * sin(alpha2) * sin(Eta2) - ...
36         sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2)
        ) * sin(Eta2))) / J;
37     alpha = alpha + omega * t + 1/2 * beta * t^2;
38     omega = omega + beta * t;
39 end
40 fprintf('鼓面倾角: %g°\n', rad2deg(alpha));

```

## B.5 子问题五

```

1  clc
2  clear
3  d = 0.4; r = d/2; m = 3.6; h = 0.22;
4  F1 = 90; F2 = 80;
5  l = 1.7;
6  phi = asin(0.11/1.7); alpha = 0; omega = 0;
7  gamma1 = pi / 8; gamma2 = 3 * pi / 8;

```

```

8 J = 1/2 * m * r^2 + 1/12 * m * h^2; %圆环的转动惯量。
9 step = 1000000;
10 t = 0.1 / step; % 0.1 秒的步长。
11
12 for i=1:step
13     alpha1 = asin(sin(alpha) * cos(gamma1));
14     Eta1 = acos(sin(gamma1) / cos(alpha1));
15     theta1 = alpha1 * r / l;
16     % 转动方程求此时的角加速度。
17     beta = 2 * r * F2 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1)) / J;
18     alpha = alpha + omega * t + 1/2 * beta * t^2;
19     omega = omega + beta * t;
20 end
21 fprintf('0.1s前的鼓面倾角: %g°\n', rad2deg(alpha));
22
23 for i=1:step
24     alpha1 = asin(sin(alpha) * cos(gamma1));
25     alpha2 = asin(sin(alpha) * cos(gamma2));
26     Eta1 = acos(sin(gamma1) / cos(alpha1));
27     Eta2 = acos(sin(gamma2) / cos(alpha2));
28     theta1 = alpha1 * r / l;
29     theta2 = alpha2 * r / l;
30     % 转动方程求此时的角加速度。
31     beta = (2 * F2 * r * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1) - ...
32         sin(phi + theta1) * cos(alpha) * cos(gamma1) - cos(phi + theta1) * sin(alpha1
        ) * sin(Eta1)) + ...
33         2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2
        ) * sin(alpha2) * sin(Eta2) - ...
34         sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2
        ) * sin(Eta2))) / J;
35     alpha = alpha + omega * t + 1/2 * beta * t^2;
36     omega = omega + beta * t;
37 end
38 fprintf('鼓面倾角: %g°\n', rad2deg(alpha));

```

## B.6 子问题六

```

1 clc
2 clear

```

```

3 d = 0.4; r = d/2; m = 3.6; h = 0.22;
4 F1 = 90; F2 = 80;
5 l = 1.7;
6 phi = asin(0.11/1.7); alpha = 0; omega = 0;
7 gamma2 = pi / 8; gamma1 = 3 * pi / 8;
8 J = 1/2 * m * r^2 + 1/12 * m * h^2; %圆环的转动惯量。
9 step = 1000000;
10 t = 0.1 / step; % 0.1 秒的步长。
11
12 for i=1:step
13     alpha1 = asin(sin(alpha) * cos(gamma1));
14     Eta1 = acos(sin(gamma1) / cos(alpha1));
15     theta1 = alpha1 * r / l;
16     % 转动方程求此时的角加速度。
17     beta = 2 * r * F2 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1)) / J;
18     alpha = alpha + omega * t + 1/2 * beta * t^2;
19     omega = omega + beta * t;
20 end
21 fprintf('0.1s前的鼓面倾角: %g°\n', rad2deg(alpha));
22
23 for i=1:step
24     % if mod(i,10) == 0
25     %     fprintf('%d\n', i/10);
26     % end
27     alpha1 = asin(sin(alpha) * cos(gamma1));
28     alpha2 = asin(sin(alpha) * cos(gamma2));
29     Eta1 = acos(sin(gamma1) / cos(alpha1));
30     Eta2 = acos(sin(gamma2) / cos(alpha2));
31     theta1 = alpha1 * r / l;
32     theta2 = alpha2 * r / l;
33     % 转动方程求此时的角加速度。
34     beta = (2 * F2 * r * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1) - ...
35         sin(phi + theta1) * cos(alpha) * cos(gamma1) - cos(phi + theta1) * sin(alpha1
        ) * sin(Eta1)) + ...
36         2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2
        ) * sin(alpha2) * sin(Eta2) - ...
37         sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2
        ) * sin(Eta2)))) / J;
38     alpha = alpha + omega * t + 1/2 * beta * t^2;

```



```

39     omega = omega + beta * t;
40 end
41 fprintf('鼓面倾角: %g°\n', rad2deg(alpha));

```

## B.7 子问题七

```

1  clc
2  clear
3  d = 0.4; r = d/2; m = 3.6; h = 0.22;
4  F1 = 90; F2 = 80;
5  l = 1.7;
6  phi = asin(0.11/1.7); alpha = 0; omega = 0;
7  gamma1 = 0; gamma2 = pi / 4;
8  J = 1/2 * m * r^2 + 1/12 * m * h^2; %圆环的转动惯量。
9  step = 1000000;
10 t = 0.1 / step; % 0.1 秒的步长。
11
12 for i=1:step
13     alpha1 = asin(sin(alpha) * cos(gamma1));
14     Eta1 = acos(sin(gamma1) / cos(alpha1));
15     theta1 = alpha1 * r / l;
16     % 转动方程求此时的角加速度。
17     beta = r * F1 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi - theta1)
        * sin(alpha1) * sin(Eta1)) / J;
18     alpha = alpha + omega * t + 1/2 * beta * t^2;
19     omega = omega + beta * t;
20 end
21 fprintf('0.1s前的鼓面倾角: %g°\n', rad2deg(alpha));
22
23 for i=1:step
24     alpha1 = asin(sin(alpha) * cos(gamma1));
25     alpha2 = asin(sin(alpha) * cos(gamma2));
26     Eta1 = acos(sin(gamma1) / cos(alpha1));
27     Eta2 = acos(sin(gamma2) / cos(alpha2));
28     theta1 = alpha1 * r / l;
29     theta2 = alpha2 * r / l;
30     % 转动方程求此时的角加速度。
31     beta = (r * (F1 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1)) - ...
32         F2 * (sin(phi + theta1) * cos(alpha) * cos(gamma1) + cos(phi + theta1) * sin(
        alpha1) * sin(Eta1)))) + ...

```

```

33         2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2
           ) * sin(alpha2) * sin(Eta2) - ...
34         sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2
           ) * sin(Eta2))) / J;
35     alpha = alpha + omega * t + 1/2 * beta * t^2;
36     omega = omega + beta * t;
37 end
38 fprintf('鼓面倾角: %g°\n', rad2deg(alpha));

```

## B.8 子问题八

```

1  clc
2  clear
3  d = 0.4; r = d/2; m = 3.6; h = 0.22;
4  F1 = 90; F2 = 80;
5  l = 1.7;
6  phi = asin(0.11/1.7); alpha = 0; omega = 0;
7  gamma1 = 3 * pi / 8; gamma2 = pi / 8;
8  J = 1/2 * m * r^2 + 1/12 * m * h^2; %圆环的转动惯量。
9  step = 1000000;
10 t = 0.1 / step; % 0.1 秒的步长。
11
12 for i=1:step
13     alpha1 = asin(sin(alpha) * cos(gamma1));
14     Eta1 = acos(sin(gamma1) / cos(alpha1));
15     theta1 = alpha1 * r / l;
16     % 转动方程求此时的角加速度。
17     beta = 2 * r * F2 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
           theta1) * sin(alpha1) * sin(Eta1)) / J;
18     alpha = alpha + omega * t + 1/2 * beta * t^2;
19     omega = omega + beta * t;
20 end
21 fprintf('0.1s前的鼓面倾角: %g°\n', rad2deg(alpha));
22
23 for i=1:step
24     %     if mod(i,10) == 0
25     %         fprintf('%d\n', i/10);
26     %     end
27     alpha1 = asin(sin(alpha) * cos(gamma1));
28     alpha2 = asin(sin(alpha) * cos(gamma2));
29     Eta1 = acos(sin(gamma1) / cos(alpha1));

```

```

30     Eta2 = acos(sin(gamma2) / cos(alpha2));
31     theta1 = alpha1 * r / l;
32     theta2 = alpha2 * r / l;
33     % 转动方程求此时的角加速度。
34     beta = (r * (F2 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1)) - ...
35         F1 * (sin(phi + theta1) * cos(alpha) * cos(gamma1) + cos(phi + theta1) * sin(
        alpha1) * sin(Eta1))) + ...
36         F2 * r * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi - theta1) *
        sin(alpha1) * sin(Eta1) - ...
37         sin(phi + theta1) * cos(alpha) * cos(gamma1) - cos(phi + theta1) * sin(alpha1
        ) * sin(Eta1)) + ...
38         r * (F1 * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2) *
        sin(alpha2) * sin(Eta2)) - ...
39         F2 * (sin(phi + theta2) * cos(alpha) * cos(gamma2) + cos(phi + theta2) * sin(
        alpha2) * sin(Eta2))) + ...
40         F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2) *
        sin(alpha2) * sin(Eta2) - ...
41         sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2
        ) * sin(Eta2))) / J;
42     alpha = alpha + omega * t + 1/2 * beta * t^2;
43     omega = omega + beta * t;
44 end
45 fprintf('鼓面倾角: %g°\n', rad2deg(alpha));

```

## B.9 子问题九

```

1  clc
2  clear
3  d = 0.4; r = d/2; m = 3.6; h = 0.22;
4  F1 = 90; F2 = 80;
5  l = 1.7;
6  phi = asin(0.11/1.7); alpha = 0; omega = 0;
7  gamma1 = 3 * pi / 8; gamma2 = pi / 8;
8  J = 1/2 * m * r^2 + 1/12 * m * h^2; %圆环的转动惯量。
9  step = 1000000;
10 t = 0.1 / step; % 0.1 秒的步长。
11
12 for i=1:step
13     alpha1 = asin(sin(alpha) * cos(gamma1));
14     Eta1 = acos(sin(gamma1) / cos(alpha1));

```

```

15     theta1 = alpha1 * r / l;
16     % 转动方程求此时的角加速度。
17     beta = 2 * r * F2 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1)) / J;
18     alpha = alpha + omega * t + 1/2 * beta * t^2;
19     omega = omega + beta * t;
20 end
21 fprintf('0.1s前的鼓面倾角: %g°\n', rad2deg(alpha));
22
23 for i=1:step
24     %     if mod(i,10) == 0
25     %         fprintf('%d\n', i/10);
26     %     end
27     alpha1 = asin(sin(alpha) * cos(gamma1));
28     alpha2 = asin(sin(alpha) * cos(gamma2));
29     Eta1 = acos(sin(gamma1) / cos(alpha1));
30     Eta2 = acos(sin(gamma2) / cos(alpha2));
31     theta1 = alpha1 * r / l;
32     theta2 = alpha2 * r / l;
33     % 转动方程求此时的角加速度。
34     beta = (2 * r * (F2 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1)) - ...
35         F1 * (sin(phi + theta1) * cos(alpha) * cos(gamma1) + cos(phi + theta1) * sin(
        alpha1) * sin(Eta1))) + ...
36         2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2
        ) * sin(alpha2) * sin(Eta2) - ...
37         sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2
        ) * sin(Eta2))) / J;
38     alpha = alpha + omega * t + 1/2 * beta * t^2;
39     omega = omega + beta * t;
40 end
41 fprintf('鼓面倾角: %g°\n', rad2deg(alpha));

```

## C 问题三求解代码

### C.1 类型一策略一

```

1 clc
2 clear
3 d = 0.4; r = d/2; m = 3.6; h = 0.22;
4 F1 = 90; F2 = 80;

```

```

5 l = 1.7;
6 phi = asin(0.11/1.7); alpha = 0; omega = 0;
7 gamma1 = 0; gamma2 = pi / 4;
8 J = 1/2 * m * r^2 + 1/12 * m * h^2; % 圆环的转动惯量。
9 step = 1000000;
10 t = 0.1 / step;
11 alphadata = [];
12
13 % 有策略情况下
14 % -0.1 - 0 s 提前发力
15 for i=1:step
16     alpha1 = asin(sin(alpha) * cos(gamma1));
17     Eta1 = acos(sin(gamma1) / cos(alpha1));
18     theta1 = alpha1 * r / l;
19     beta = r * F2 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi - theta1)
        ) * sin(alpha1) * sin(Eta1)) / J;
20     alpha = alpha + omega * t + 1/2 * beta * t^2;
21     omega = omega + beta * t;
22     if mod(i,100) == 0
23         alphadata = [alphadata, rad2deg(alpha)];
24     end
25 end
26 % 0 - 0.1 s 正常发力
27 for i=1:step
28     alpha1 = asin(sin(alpha) * cos(gamma1));
29     alpha2 = asin(sin(alpha) * cos(gamma2));
30     Eta1 = acos(sin(gamma1) / cos(alpha1));
31     Eta2 = acos(sin(gamma2) / cos(alpha2));
32     theta1 = alpha1 * r / l;
33     theta2 = alpha2 * r / l;
34     beta = (F2 * r * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1) - ...
35         sin(phi + theta1) * cos(alpha) * cos(gamma1) - cos(phi + theta1) * sin(alpha1)
        ) * sin(Eta1)) + ...
36         2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2)
        ) * sin(alpha2) * sin(Eta2) - ...
37         sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2)
        ) * sin(Eta2))) / J;
38     alpha = alpha + omega * t + 1/2 * beta * t^2;
39     omega = omega + beta * t;
40     if mod(i,100) == 0

```

```

41     alphadata = [alphadata, rad2deg(alpha)];
42 end
43 end
44
45 % 0.1 - 0.2 s 反应过来停止发力
46 for i=1:step
47     alpha1 = asin(sin(alpha) * cos(gamma1));
48     alpha2 = asin(sin(alpha) * cos(gamma2));
49     Eta1 = acos(sin(gamma1) / cos(alpha1));
50     Eta2 = acos(sin(gamma2) / cos(alpha2));
51     theta1 = alpha1 * r / l;
52     theta2 = alpha2 * r / l;
53     beta = (F2 * r * (- sin(phi + theta1) * cos(alpha) * cos(gamma1) - cos(phi +
        theta1) * sin(alpha1) * sin(Eta1)) + ...
54         2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2
        ) * sin(alpha2) * sin(Eta2) - ...
55         sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2
        ) * sin(Eta2))) / J;
56     alpha = alpha + omega * t + 1/2 * beta * t^2;
57     omega = omega + beta * t;
58     if mod(i,100) == 0
59         alphadata = [alphadata, rad2deg(alpha)];
60     end
61 end
62
63 % 0.2 - 1 s 正常发力
64 step = 0.8 / t;
65 for i=1:step
66     alpha1 = asin(sin(alpha) * cos(gamma1));
67     alpha2 = asin(sin(alpha) * cos(gamma2));
68     Eta1 = acos(sin(gamma1) / cos(alpha1));
69     Eta2 = acos(sin(gamma2) / cos(alpha2));
70     theta1 = alpha1 * r / l;
71     theta2 = alpha2 * r / l;
72     beta = (F2 * r * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1) - ...
73         sin(phi + theta1) * cos(alpha) * cos(gamma1) - cos(phi + theta1) * sin(alpha1
        ) * sin(Eta1)) + ...
74         2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2
        ) * sin(alpha2) * sin(Eta2) - ...

```

```

75         sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2
           ) * sin(Eta2))) / J;
76     alpha = alpha + omega * t + 1/2 * beta * t^2;
77     omega = omega + beta * t;
78     if mod(i,100) == 0
79         alphadata = [alphadata, rad2deg(alpha)];
80     end
81 end
82 x = -0.1 + (t * 100): (t * 100): 1;
83 plot(x, alphadata);
84 hold on
85
86
87 alpha = 0; omega = 0;
88 alphadata = [];
89 % 无策略
90 % -0.1 - 0 s
91 step = 0.1 / t;
92 for i=1:step
93     alpha1 = asin(sin(alpha) * cos(gamma1));
94     Eta1 = acos(sin(gamma1) / cos(alpha1));
95     theta1 = alpha1 * r / l;
96     beta = r * F2 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi - theta1
           ) * sin(alpha1) * sin(Eta1)) / J;
97     alpha = alpha + omega * t + 1/2 * beta * t^2;
98     omega = omega + beta * t;
99     if mod(i,100) == 0
100         alphadata = [alphadata, rad2deg(alpha)];
101     end
102 end
103 % 0 - 1 s
104 step = 1 / t;
105 for i=1:step
106     alpha1 = asin(sin(alpha) * cos(gamma1));
107     alpha2 = asin(sin(alpha) * cos(gamma2));
108     Eta1 = acos(sin(gamma1) / cos(alpha1));
109     Eta2 = acos(sin(gamma2) / cos(alpha2));
110     theta1 = alpha1 * r / l;
111     theta2 = alpha2 * r / l;
112     beta = (F2 * r * (- sin(phi + theta1) * cos(alpha) * cos(gamma1) - cos(phi +
           theta1) * sin(alpha1) * sin(Eta1)) + ...

```

```

113         2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2
            ) * sin(alpha2) * sin(Eta2) - ...
114         sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2
            ) * sin(Eta2))) / J;
115     alpha = alpha + omega * t + 1/2 * beta * t^2;
116     omega = omega + beta * t;
117     if mod(i,100) == 0
118         alphadata = [alphadata, rad2deg(alpha)];
119     end
120 end
121 plot(x, alphadata);
122 xlabel('时间(s)');
123 ylabel('角度(°)');
124 legend('策略','无策略');

```

## C.2 类型一策略二

```

1  clc
2  clear
3  d = 0.4; r = d/2; m = 3.6; h = 0.22; l = 1.7;
4  F2 = 80;
5  phi = asin(0.11/1.7);
6  gamma1 = 0; gamma2 = pi / 4;
7  J = 1/2 * m * r^2 + 1/12 * m * h^2; %圆环的转动惯量。
8  step = 1000000;
9  t = 1 / step;
10
11 x = -0.1 + (t * 100): (t * 100): 1;
12 hold on
13 for F1 = 160:80:400
14     alphadata = question3_type1method2_fun(F1);
15     plot(x, alphadata);
16 end
17
18 alpha = 0; omega = 0;
19 alphadata = [];
20 step = 0.1 / t;
21 % -0.1 - 0 先发力的时间
22 for i=1:step
23     alpha1 = asin(sin(alpha) * cos(gamma1));
24     Eta1 = acos(sin(gamma1) / cos(alpha1));

```



```

25     theta1 = alpha1 * r / l;
26     beta = r * F2 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi - theta1
        ) * sin(alpha1) * sin(Eta1)) / J;
27     alpha = alpha + omega * t + 1/2 * beta * t^2;
28     omega = omega + beta * t;
29     if mod(i,100) == 0
30         alphadata = [alphadata, rad2deg(alpha)];
31     end
32 end
33 % 0 - 1 正常发力
34 step = 1 / t;
35 for i=1:step
36     alpha1 = asin(sin(alpha) * cos(gamma1));
37     alpha2 = asin(sin(alpha) * cos(gamma2));
38     Eta1 = acos(sin(gamma1) / cos(alpha1));
39     Eta2 = acos(sin(gamma2) / cos(alpha2));
40     theta1 = alpha1 * r / l;
41     theta2 = alpha2 * r / l;
42     beta = (F2 * r * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1) - ...
43         sin(phi + theta1) * cos(alpha) * cos(gamma1) - cos(phi + theta1) * sin(alpha1
        ) * sin(Eta1)) + ...
44         2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2
        ) * sin(alpha2) * sin(Eta2) - ...
45         sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2
        ) * sin(Eta2))) / J;
46     alpha = alpha + omega * t + 1/2 * beta * t^2;
47     omega = omega + beta * t;
48     if mod(i,100) == 0
49         alphadata = [alphadata, rad2deg(alpha)];
50     end
51 end
52 plot(x, alphadata);
53 xlabel('时间(s)');
54 ylabel('角度(°)');
55 legend('策略160', '策略240', '策略320', '策略400', '无策略');

```

### C.3 类型一策略二辅助代码

```

1 function [alphadata] = question3_type1method2_fun(F1)
2 % 计算不同的 F1 对之后的转动角度造成的影响。

```

```

3 d = 0.4; r = d/2; m = 3.6; h = 0.22; F2 = 80;
4 l = 1.7;
5 phi = asin(0.11/1.7); alpha = 0; omega = 0;
6 gamma1 = 0; gamma2 = pi / 4;
7 J = 1/2 * m * r^2 + 1/12 * m * h^2; %圆环的转动惯量。
8 step = 1000000;
9 t = 1 / step; % 0.1 秒的步长。
10 alphadata = [];
11
12 % -0.1 - 0 s 先发力 80 N
13 step = 0.1 / t;
14 for i=1:step
15     alpha1 = asin(sin(alpha) * cos(gamma1));
16     Eta1 = acos(sin(gamma1) / cos(alpha1));
17     theta1 = alpha1 * r / l;
18     beta = r * F2 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi - theta1)
        ) * sin(alpha1) * sin(Eta1)) / J;
19     alpha = alpha + omega * t + 1/2 * beta * t^2;
20     omega = omega + beta * t;
21     if mod(i,100) == 0
22         alphadata = [alphadata, rad2deg(alpha)];
23     end
24 end
25 % 0 - 0.1 s 一起发力 80 N
26 for i=1:step
27     alpha1 = asin(sin(alpha) * cos(gamma1));
28     alpha2 = asin(sin(alpha) * cos(gamma2));
29     Eta1 = acos(sin(gamma1) / cos(alpha1));
30     Eta2 = acos(sin(gamma2) / cos(alpha2));
31     theta1 = alpha1 * r / l;
32     theta2 = alpha2 * r / l;
33     beta = (F2 * r * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1) - ...
34         sin(phi + theta1) * cos(alpha) * cos(gamma1) - cos(phi + theta1) * sin(alpha1)
        ) * sin(Eta1)) + ...
35         2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2)
        ) * sin(alpha2) * sin(Eta2) - ...
36         sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2)
        ) * sin(Eta2))) / J;
37     alpha = alpha + omega * t + 1/2 * beta * t^2;
38     omega = omega + beta * t;

```

```

39     if mod(i,100) == 0
40         alphadata = [alphadata, rad2deg(alpha)];
41     end
42 end
43 % 0.1 - 0.2 s 对面发力增大
44 for i=1:step
45     % if mod(i,10) == 0
46     %     fprintf('%d\n', i/10);
47     % end
48     alpha1 = asin(sin(alpha) * cos(gamma1));
49     alpha2 = asin(sin(alpha) * cos(gamma2));
50     Eta1 = acos(sin(gamma1) / cos(alpha1));
51     Eta2 = acos(sin(gamma2) / cos(alpha2));
52     theta1 = alpha1 * r / l;
53     theta2 = alpha2 * r / l;
54     beta = (r * (F2 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1)) - ...
55         F1 * (sin(phi + theta1) * cos(alpha) * cos(gamma1) + cos(phi + theta1) * sin(
        alpha1) * sin(Eta1))) + ...
56         2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2
        ) * sin(alpha2) * sin(Eta2) - ...
57         sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2
        ) * sin(Eta2)))) / J;
58     alpha = alpha + omega * t + 1/2 * beta * t^2;
59     omega = omega + beta * t;
60     if mod(i,100) == 0
61         alphadata = [alphadata, rad2deg(alpha)];
62     end
63 end
64
65 % 0.2 - 1 s 恢复正常
66 step = 0.8 / t;
67 for i=1:step
68     % if mod(i,10) == 0
69     %     fprintf('%d\n', i/10);
70     % end
71     alpha1 = asin(sin(alpha) * cos(gamma1));
72     alpha2 = asin(sin(alpha) * cos(gamma2));
73     Eta1 = acos(sin(gamma1) / cos(alpha1));
74     Eta2 = acos(sin(gamma2) / cos(alpha2));
75     theta1 = alpha1 * r / l;

```

```

76     theta2 = alpha2 * r / l;
77     beta = (F2 * r * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1) - ...
78         sin(phi + theta1) * cos(alpha) * cos(gamma1) - cos(phi + theta1) * sin(alpha1
        ) * sin(Eta1)) + ...
79         2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2
        ) * sin(alpha2) * sin(Eta2) - ...
80         sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2
        ) * sin(Eta2)))) / J;
81     alpha = alpha + omega * t + 1/2 * beta * t^2;
82     omega = omega + beta * t;
83     if mod(i,100) == 0
84         alphadata = [alphadata, rad2deg(alpha)];
85     end
86 end

```

## C.4 类型二策略

```

1  % 针对类型二,队员同时发力, 但是有队员未掌控力度。
2  clc
3  clear
4  d = 0.4; r = d/2; m = 3.6; h = 0.22;
5  F1 = 90; F2 = 80;
6  l = 1.7;
7  phi = asin(0.11/1.7); alpha = 0; omega = 0;
8  gamma1 = 0; gamma2 = pi / 4;
9  J = 1/2 * m * r^2 + 1/12 * m * h^2; %圆环的转动惯量。
10 step = 1000000;
11 t = 1 / step;
12 alphadata = []; % 0.1 秒的步长。
13
14 % 有策略情况
15 % 0 - 0.2s
16 step = 0.2 / t;
17 for i=1:step
18     alpha1 = asin(sin(alpha) * cos(gamma1));
19     alpha2 = asin(sin(alpha) * cos(gamma2));
20     Eta1 = acos(sin(gamma1) / cos(alpha1));
21     Eta2 = acos(sin(gamma2) / cos(alpha2));
22     theta1 = alpha1 * r / l;
23     theta2 = alpha2 * r / l;

```

```

24     % 转动方程求此时的角加速度。
25     beta = (r * (F1 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1))) - ...
26         F2 * (sin(phi + theta1) * cos(alpha) * cos(gamma1) + cos(phi + theta1) * sin(
        alpha1) * sin(Eta1))) + ...
27         2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2
        ) * sin(alpha2) * sin(Eta2) - ...
28         sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2
        ) * sin(Eta2)))) / J;
29     alpha = alpha + omega * t + 1/2 * beta * t^2;
30     omega = omega + beta * t;
31     % 记录数据
32     if mod(i,100) == 0
33         alphadata = [alphadata, rad2deg(alpha)];
34     end
35 end
36 % 0.2 - 0.4s
37 F1 = 70;
38 for i=1:step
39     alpha1 = asin(sin(alpha) * cos(gamma1));
40     alpha2 = asin(sin(alpha) * cos(gamma2));
41     Eta1 = acos(sin(gamma1) / cos(alpha1));
42     Eta2 = acos(sin(gamma2) / cos(alpha2));
43     theta1 = alpha1 * r / l;
44     theta2 = alpha2 * r / l;
45     % 转动方程求此时的角加速度。
46     beta = (r * (F1 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1))) - ...
47         F2 * (sin(phi + theta1) * cos(alpha) * cos(gamma1) + cos(phi + theta1) * sin(
        alpha1) * sin(Eta1))) + ...
48         2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2
        ) * sin(alpha2) * sin(Eta2) - ...
49         sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2
        ) * sin(Eta2)))) / J;
50     alpha = alpha + omega * t + 1/2 * beta * t^2;
51     omega = omega + beta * t;
52     if mod(i,100) == 0
53         alphadata = [alphadata, rad2deg(alpha)];
54     end
55 end
56 % 0.4 - 1s

```

```

57 step = 0.6 / t;
58 F1 = 80;
59 for i=1:step
60     alpha1 = asin(sin(alpha) * cos(gamma1));
61     alpha2 = asin(sin(alpha) * cos(gamma2));
62     Eta1 = acos(sin(gamma1) / cos(alpha1));
63     Eta2 = acos(sin(gamma2) / cos(alpha2));
64     theta1 = alpha1 * r / l;
65     theta2 = alpha2 * r / l;
66     % 转动方程求此时的角加速度。
67     beta = (r * (F1 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1)) - ...
68         F2 * (sin(phi + theta1) * cos(alpha) * cos(gamma1) + cos(phi + theta1) * sin(
        alpha1) * sin(Eta1))) + ...
69         2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2
        ) * sin(alpha2) * sin(Eta2) - ...
70         sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2
        ) * sin(Eta2))) / J;
71     alpha = alpha + omega * t + 1/2 * beta * t^2;
72     omega = omega + beta * t;
73     if mod(i,100) == 0
74         alphadata = [alphadata, rad2deg(alpha)];
75     end
76 end
77 x = (t * 100): (t * 100): 1;
78 plot(x, alphadata);
79 hold on
80 % 无策略
81 alpha = 0; omega = 0;
82 F1 = 90;
83 alphadata = [];
84 step = 1 / t;
85 for i=1:step
86     alpha1 = asin(sin(alpha) * cos(gamma1));
87     alpha2 = asin(sin(alpha) * cos(gamma2));
88     Eta1 = acos(sin(gamma1) / cos(alpha1));
89     Eta2 = acos(sin(gamma2) / cos(alpha2));
90     theta1 = alpha1 * r / l;
91     theta2 = alpha2 * r / l;
92     % 转动方程求此时的角加速度。

```

```

93     beta = (r * (F1 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1))) - ...
94     F2 * (sin(phi + theta1) * cos(alpha) * cos(gamma1) + cos(phi + theta1) * sin(
        alpha1) * sin(Eta1))) + ...
95     2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2
        ) * sin(alpha2) * sin(Eta2) - ...
96     sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2
        ) * sin(Eta2))) / J;
97     alpha = alpha + omega * t + 1/2 * beta * t^2;
98     omega = omega + beta * t;
99     if mod(i,100) == 0
100         alphadata = [alphadata, rad2deg(alpha)];
101     end
102 end
103 plot(x, alphadata);
104 xlabel('时间(s)');
105 ylabel('角度(°)');
106 legend('策略','无策略');

```

## C.5 类型三策略

```

1  clc
2  clear
3  d = 0.4; r = d/2; m = 3.6; h = 0.22;
4  F1 = 90; F2 = 80;
5  l = 1.7;
6  phi = asin(0.11/1.7); alpha = 0; omega = 0;
7  gamma1 = 0; gamma2 = pi / 4;
8  J = 1/2 * m * r^2 + 1/12 * m * h^2; %圆环的转动惯量。
9  step = 1000000;
10 t = 1 / step;
11 alphadata = [];
12
13 % 无策略的情况
14 % -0.1 - 0 s
15 step = 0.1 / t;
16 for i=1:step
17     alpha1 = asin(sin(alpha) * cos(gamma1));
18     Eta1 = acos(sin(gamma1) / cos(alpha1));
19     theta1 = alpha1 * r / l;

```

```

20     beta = r * F1 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi - theta1
        ) * sin(alpha1) * sin(Eta1)) / J;
21     alpha = alpha + omega * t + 1/2 * beta * t^2;
22     omega = omega + beta * t;
23     if mod(i,100) == 0
24         alphadata = [alphadata, rad2deg(alpha)];
25     end
26 end
27
28 % 0 - 1 s
29 step = 1 / t;
30 for i=1:step
31     alpha1 = asin(sin(alpha) * cos(gamma1));
32     alpha2 = asin(sin(alpha) * cos(gamma2));
33     Eta1 = acos(sin(gamma1) / cos(alpha1));
34     Eta2 = acos(sin(gamma2) / cos(alpha2));
35     theta1 = alpha1 * r / l;
36     theta2 = alpha2 * r / l;
37     beta = (r * (F1 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1)) - ...
38         F2 * (sin(phi + theta1) * cos(alpha) * cos(gamma1) + cos(phi + theta1) * sin(
        alpha1) * sin(Eta1))) + ...
39         2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2
        ) * sin(alpha2) * sin(Eta2)) - ...
40         sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2
        ) * sin(Eta2))) / J;
41     alpha = alpha + omega * t + 1/2 * beta * t^2;
42     omega = omega + beta * t;
43     if mod(i,100) == 0
44         alphadata = [alphadata, rad2deg(alpha)];
45     end
46 end
47 x = -0.1 + (t * 100): (t * 100): 1;
48 hold on
49 plot(x, alphadata);
50
51 for F3 = 160: 80: 400
52     alphadata = question3_type3_fun(F3);
53     plot(x, alphadata);
54 end
55 xlabel('时间(s)');

```



```

56 ylabel('角度(°)');
57 legend('无策略', '策略160', '策略240', '策略320', '策略400');

```

## C.6 类型三策略辅助代码

```

1 function [alphadata] = question3_type3_fun(F3)
2 d = 0.4; r = d/2; m = 3.6; h = 0.22; F1 = 90; F2 = 80;
3 l = 1.7;
4 phi = asin(0.11/1.7); alpha = 0; omega = 0;
5 gamma1 = 0; gamma2 = pi / 4;
6 J = 1/2 * m * r^2 + 1/12 * m * h^2; %圆环的转动惯量。
7 step = 1000000;
8 t = 1 / step;
9 alphadata = [];
10
11 % 有策略情况
12 % -0.1 - 0 s 先发力 90 N
13 step = 0.1 / t;
14 for i=1:step
15     alpha1 = asin(sin(alpha) * cos(gamma1));
16     Eta1 = acos(sin(gamma1) / cos(alpha1));
17     theta1 = alpha1 * r / l;
18     beta = r * F1 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi - theta1)
19         ) * sin(alpha1) * sin(Eta1)) / J;
20     alpha = alpha + omega * t + 1/2 * beta * t^2;
21     omega = omega + beta * t;
22     if mod(i,100) == 0
23         alphadata = [alphadata, rad2deg(alpha)];
24     end
25 % 0 - 0.1 s 继续 90 N发力
26 for i=1:step
27     alpha1 = asin(sin(alpha) * cos(gamma1));
28     alpha2 = asin(sin(alpha) * cos(gamma2));
29     Eta1 = acos(sin(gamma1) / cos(alpha1));
30     Eta2 = acos(sin(gamma2) / cos(alpha2));
31     theta1 = alpha1 * r / l;
32     theta2 = alpha2 * r / l;
33     beta = (r * (F1 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
34         theta1) * sin(alpha1) * sin(Eta1)) - ...

```

```

34      F2 * (sin(phi + theta1) * cos(alpha) * cos(gamma1) + cos(phi + theta1) * sin(
          alpha1) * sin(Eta1))) + ...
35      2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2
          ) * sin(alpha2) * sin(Eta2) - ...
36      sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2
          ) * sin(Eta2))) / J;
37      alpha = alpha + omega * t + 1/2 * beta * t^2;
38      omega = omega + beta * t;
39      if mod(i,100) == 0
40          alphadata = [alphadata, rad2deg(alpha)];
41      end
42  end
43  % 0.1 - 0.2 s 意识到发力过大, 自己以 70 N 发力, 对方以 F3 发力
44  F1 = 70;
45  for i=1:step
46      alpha1 = asin(sin(alpha) * cos(gamma1));
47      alpha2 = asin(sin(alpha) * cos(gamma2));
48      Eta1 = acos(sin(gamma1) / cos(alpha1));
49      Eta2 = acos(sin(gamma2) / cos(alpha2));
50      theta1 = alpha1 * r / l;
51      theta2 = alpha2 * r / l;
52      beta = (r * (F1 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
          theta1) * sin(alpha1) * sin(Eta1)) - ...
53      F3 * (sin(phi + theta1) * cos(alpha) * cos(gamma1) + cos(phi + theta1) * sin(
          alpha1) * sin(Eta1))) + ...
54      2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2
          ) * sin(alpha2) * sin(Eta2) - ...
55      sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2
          ) * sin(Eta2))) / J;
56      alpha = alpha + omega * t + 1/2 * beta * t^2;
57      omega = omega + beta * t;
58      if mod(i,100) == 0
59          alphadata = [alphadata, rad2deg(alpha)];
60      end
61  end
62  % 0.2 - 0.3 s 对方发力回归正常 80 N, 自己仍然是 70 N
63  for i=1:step
64      alpha1 = asin(sin(alpha) * cos(gamma1));
65      alpha2 = asin(sin(alpha) * cos(gamma2));
66      Eta1 = acos(sin(gamma1) / cos(alpha1));
67      Eta2 = acos(sin(gamma2) / cos(alpha2));

```

```

68     theta1 = alpha1 * r / l;
69     theta2 = alpha2 * r / l;
70     beta = (r * (F1 * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1)) - ...
71         F2 * (sin(phi + theta1) * cos(alpha) * cos(gamma1) + cos(phi + theta1) * sin(
        alpha1) * sin(Eta1))) + ...
72         2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2
        ) * sin(alpha2) * sin(Eta2) - ...
73         sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2
        ) * sin(Eta2))) / J;
74     alpha = alpha + omega * t + 1/2 * beta * t^2;
75     omega = omega + beta * t;
76     if mod(i,100) == 0
77         alphadata = [alphadata, rad2deg(alpha)];
78     end
79 end
80 % 0.3 - 1 s 自己发力恢复正常
81 step = 0.7 / t;
82 for i=1:step
83     alpha1 = asin(sin(alpha) * cos(gamma1));
84     alpha2 = asin(sin(alpha) * cos(gamma2));
85     Eta1 = acos(sin(gamma1) / cos(alpha1));
86     Eta2 = acos(sin(gamma2) / cos(alpha2));
87     theta1 = alpha1 * r / l;
88     theta2 = alpha2 * r / l;
89     beta = (F2 * r * (sin(phi - theta1) * cos(alpha) * cos(gamma1) - cos(phi -
        theta1) * sin(alpha1) * sin(Eta1) - ...
90         sin(phi + theta1) * cos(alpha) * cos(gamma1) - cos(phi + theta1) * sin(alpha1
        ) * sin(Eta1)) + ...
91         2 * F2 * r * (sin(phi - theta2) * cos(alpha) * cos(gamma2) - cos(phi - theta2
        ) * sin(alpha2) * sin(Eta2) - ...
92         sin(phi + theta2) * cos(alpha) * cos(gamma2) - cos(phi + theta2) * sin(alpha2
        ) * sin(Eta2))) / J;
93     alpha = alpha + omega * t + 1/2 * beta * t^2;
94     omega = omega + beta * t;
95     if mod(i,100) == 0
96         alphadata = [alphadata, rad2deg(alpha)];
97     end
98 end

```

## D 问题四解答代码

```
1  clc
2  clear
3  d = 0.4; r = d/2; m = 3.6; h = 0.22; g = 9.8; L = 2;
4  phi = asin(0.11/L); F1 = 80; F2 = 80;
5  F = F1 * 10;
6  az = (F * sin(phi) - m * g) / m;
7  lambada = deg2rad(0.5) * r / L;
8
9  syms t1 t2
10 equ1 = -g * t1 * t2 + 1/2 * az * t2^2 == 1/2 * g * t1^2;
11 equ2 = t1 + t2 == 2 * sqrt(1.2 / g);
12 [t1, t2] = solve([equ1, equ2], [t1, t2]);
13 fprintf('下降时间: %g\n减速与上升时间: %g\n', t1(2), t2(2));
14
15 gammai = [deg2rad(12), deg2rad(24), deg2rad(48), deg2rad(60), deg2rad(84)];
16 J = 1/2 * m * r^2 + 1/12 * m * h^2; % 圆环的转动惯量。
17 step = 10000;
18 t = double(t2(2) / 2 / step);
19
20 alphai = zeros(1,5);
21 Etai = zeros(1,5);
22 betai = zeros(1,5);
23 while 1
24     alpha = deg2rad(-0.5);
25     omega = 0;
26     alphadata = [];
27     for i = 1 : step
28         for j = 1 : 5
29             alphai(j) = - asin(sin(alpha) * cos(gammai(j)));
30             Etai(j) = acos(sin(gammai(j)) / cos(alphai(j)));
31             betai(j) = alphai(j) * r / L;
32         end
33         temp = F1 * r * sin(Etai(1)) * sin(alphai(1)) * (cos(phi + lambada - betai(1))
            ) + cos(phi - lambada + betai(1)));
34         for j = 2 : 5
35             temp = temp + F2 * r * sin(Etai(j)) * sin(alphai(j)) * (cos(phi + lambada
                - betai(j)) + cos(phi - lambada + betai(j)));
36         end
37         beta = temp / J;
```

```

38     alpha = alpha + omega * t + 1/2 * beta * t^2;
39     omega = omega + beta * t;
40     alphadata = [alphadata, rad2deg(alpha)];
41 end
42 if abs(rad2deg(alpha) - 0.5) <= 0.5 * 0.001
43     break
44 else
45     F1 = F1 - 0.001; % 这里的运算符号为加或者减，根据具体情况定。在这里是减。
46     F2 = (F - F1 * 2) / 8;
47 end
48 end
49 fprintf('F1: %g\nF2: %g\n', F1, F2);
50 fprintf('alpha: %g\n', rad2deg(alpha));

```