

# 基于多自由度振动系统的波浪能发电装置建模与功率优化

## 摘要

如何建立波浪能收集转化装置模型，以准确描述浮子在波浪中运动的情形，从而对转化装置深入研究，是能源需求与环境污染双重挑战的当下面临的一大问题。本文结合实际情况，对波浪能装置工作状态做出合理假设，建立有阻尼受迫振动的微分方程组模型。在题设要求下，利用粒子群优化算法确定最优阻尼系数以使系统平均输出功率最大化，为实际发电与测试提供理论参考。

**针对问题一** 波浪能转化装置是一个双自由度有阻尼受迫振动的模型，我们将海浪视作一个非线性弹簧与阻尼器的组合，根据受力分析、牛顿第二定律与合理假设，使用二阶常微分方程组描述浮子与振子随波浪振动过程，并通过降阶法将其转化为一阶线性方程组后，采用代数精度为4、5阶Runge-Kutta法，代入题目要求中两种不同阻尼系数求解。本文中给出波浪激励力下，浮子与振子前40个波浪周期内速度及位移图像，部分时刻数据由表格呈现。定常阻尼情况下，浮子和振子的最大垂荡位移分别为**0.622230m**和**0.676844m**；非定常阻尼情况下，其值分别为**0.628063m**和**0.687847m**。

**针对问题二** 外界环境参数已知情况下，根据题目给出的两种界限确定阻尼系数或阻尼系数相关参数，实际上是多元函数求极值问题。本文给出平均输出功率与阻尼系数的关系式，使用粒子群优化算法在可行域中求解。

最终得到：当阻尼系数为常数时，最大平均输出功率  $P=240.0065W$ ，其对应的阻尼系数为  $c = 37775.7914N \cdot s/m$ ；当阻尼系数和浮子与振子的相对速度的绝对值的幂成正比时，最大平均输出功率  $P = 240.5146W$ ，相关参数  $c_0 = 90787.6639$ ,  $k = 0.36917$ ，即阻尼系数为  $90787.6639 \cdot |\dot{x}_1 - \dot{x}_2|^{0.36917}$ 。

**针对问题三** 当考虑转化装置纵摇运动时，本题是一个四自由度系统，同样需要借助受力分析与转动平衡，使用二阶常微分方程组分别描述浮子和振子绕质心轴转动和垂荡运动。方程组同样采用降阶法与代数精度为4、5阶的Runge-Kutta法求解。本文中给出在波浪激励力和波浪激励力矩作用下前40个周期内浮子和振子的垂荡位移与速度以及纵摇角位移与角速度图像，在表格中呈现部分数据。其中浮子和振子的最大垂荡位移分别是**0.831092m**和**0.915018/m**；最大纵摇角位移分别是**0.077527/rad**和**0.077552/rad**。

**针对问题四** 基于问题二，建立平均输出功率与直线阻尼器、旋转阻尼器的阻尼系数的关系模型，通过粒子群优化算法求解该多元函数的极值问题，得出结论：当直线阻尼器与旋转阻尼器的阻尼系数分别为  $51166.9986N \cdot s/m$  和  $79728.5923N \cdot s/m$  时，装置达到最大平均输出功率：**268.5576W**

**关键字：** 多自由度振动系统；二阶常微分方程组；降阶法；Runge-Kutta法；粒子群优化算法；

# 一、问题重述

## 1.1 问题背景

经济与社会高度发展的当下，为迎接能源需求与环境污染的双重挑战，作为重要海洋可再生能源之一的波浪能应运而生，并且具有可观的应用前景，正逐渐被世界各国采纳使用。波浪能是海洋表面波浪运动所产生的能量，如何提升此种能量的转换效率是波浪能规模化利用的关键问题之一。<sup>[1]</sup>

## 1.2 问题提出

图 1 为一种波浪能装置示意图，由浮子、振子、中轴以及能量输出系统（PTO，包括弹簧和阻尼器）构成，其中振子、中轴及 PTO 被密封在浮子内部；浮子由质量均匀分布的圆柱壳体和圆锥壳体组成；两壳体连接部分有一个隔层，作为安装中轴的支撑面；振子是穿在中轴上的圆柱体，通过 PTO 系统与中轴底座链接。在波浪的作用下，浮子运动并带动振子运动，通过两者的相对运动驱动阻尼器做功，并将所做的功作为能量输出。考虑海水是无粘和无旋的，浮子在线性周期微幅播作用下会受到波浪激励力（矩）、附加惯性力（矩）、兴波阻尼力（矩）和静水恢复力（矩）。

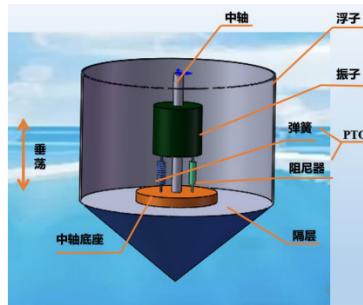


图 1 波浪能装置示意图

本文拟根据上述条件，在忽略中轴、底座、隔层及 PTO 的质量和各种摩擦的情况下，建立数学模型以研究一下四个问题：

**问题 1** 如图 1 所示，中轴底座固定于隔层的中心位置，弹簧和直线阻尼器一段固定在振子上，一端固定在中轴底座上，振子沿中轴做往复运动。直线阻尼器的阻尼力与浮子和振子的相对速度成正比，比例系数为直线阻尼器的阻尼系数。考虑浮子在波浪中只做垂荡运动，建立浮子与振子的运动模型。初始时刻浮子和振子平衡与静水中，利用附件 3 和附件 4 提供的参数值（其中波浪频率取，这里及以下出现的频率均指圆频率，角度均采用弧度制），分别对以下两种情况计算浮子和振子在波浪激励力（为波浪激励

力振幅, 为波浪频率) 作用下前 40 个波浪周期内时间间隔为的垂荡位移和速度: (1) 直线阻尼器的阻尼系数为  $10000N \cdot s/m$  (2) 直线阻尼器的阻尼系数与浮子和振子的相对速度的绝对值的幂成正比, 其中比例系数取 10000 , 幂指数取 0.5 。将结果存放在 result1-1.xlsx 和 result1-2.xlsx 中。在论文中给出 10 s , 20 s , 40 s, 60 s, 100 s 时, 浮子与振子的垂荡位移和速度。

**问题 2** 仍考虑浮子在波浪中只做垂荡运动, 分别对一下两种情况建立确定直线阻尼器的最有阻尼系数的数学模型, 使得 PTO 系统的平均输出功率最大:

(1) 阻尼系数为常量, 阻尼系数在区间  $[0,100000]$  内取值;

(2) 阻尼系数与浮子和振子的相对速度的绝对值的幂成正比, 比例系数在区间  $[0,100000]$  内取值, 幂指数在区间  $[0,1]$  内取值。利用附件 3 和附件 4 提供的参数值 (波浪频率取  $2.2143s^{-1}$ ) , 分别计算两种情况的最大输出功率及相应的最优阻尼系数。

**问题 3** 如图 2 所示, 中轴底座固定于隔层的中心位置, 中轴架通过转轴铰接于中轴底座中心, 中轴绕转轴转动, PTO 系统连接振子和转轴架, 并处于中轴和转轴所在的平面。除了直线阻尼器, 在转轴上还安装了旋转阻尼器和扭转弹簧, 直线阻尼器和旋转阻尼器共同做功输出能量。在波浪的作用下, 浮子进行摇荡运动, 并通过转轴及扭转弹簧和旋转阻尼器带动中轴转动。振子随中轴转动, 同时沿中轴进行滑动。扭转弹簧的扭矩与浮子和振子的相对角位移成正比, 比例系数为扭转弹簧的刚度。旋转阻尼器的扭矩与浮子和振子的相对角速度成正比, 比例系数为旋转阻尼器的旋转阻尼系数。考虑浮子只做垂荡和纵摇运动, 建立浮子与振子的运动模型。初始时刻浮子和振子平衡于静水中, 利用附件 3 和附件 4 提供的参数值 (波浪频率取  $1.7152s^{-1}$ ), 假定直线阻尼器和旋转阻尼器的阻尼系数均为常量, 分别为  $10000N \cdot s/m$  和  $1000N \cdot s/m$ , 计算浮子与振子在波浪激励力和波浪激励力矩  $f \cdot \cos \omega t, L \cdot \cos \omega t$  ( $f$  为波浪激励力振幅,  $L$  为波浪激励力矩振幅,  $\omega$  为波浪频率) 作用下前 40 个波浪周期内时间间隔为的垂荡位移与速度和纵摇角位移与角速度。将结果存放在 result3.xlsx 中。在论文中给出 10 s , 20 s , 40 s, 60 s, 100 s 时, 浮子与振子的垂荡位移与速度和纵摇角位移与角速度。

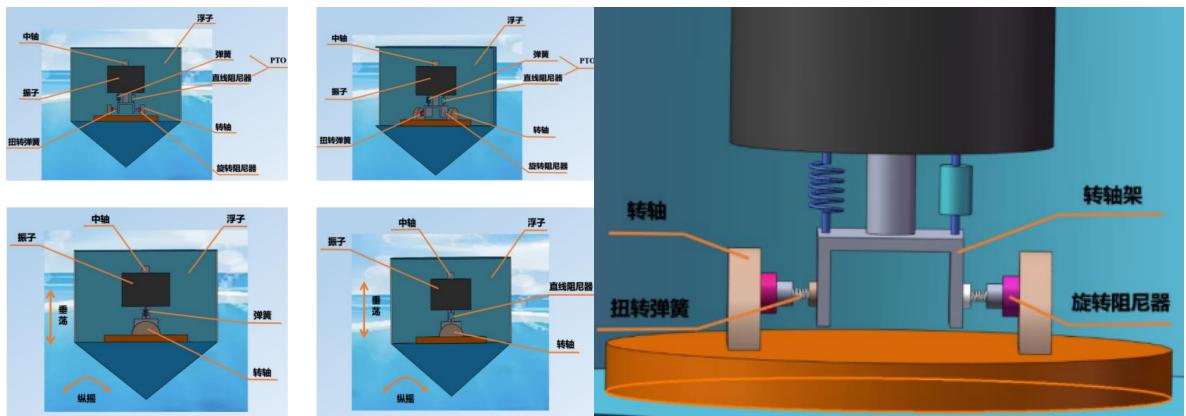


图 2 波浪能装置不同侧面的示意图

**问题4** 考虑浮子在波浪中只做垂荡和纵摇的情形，针对直线阻尼器和旋转阻尼器的阻尼系数均为常量的情况，建立确定直线阻尼器和旋转阻尼器最优阻尼系数的数学模型，直线阻尼器和旋转阻尼器的阻尼系数均在区间 [0,100000] 内取值。利用附件3和附件4提供的参数值（波浪频率取）计算最大输出功率及相应的最优阻尼系数。

**术语** 浮体在波浪的作用下做垂荡运动时，会受到海水的作用，包括附加惯性力（矩）、兴波阻尼力（矩）和静水恢复力（矩）。

**附加惯性力（矩）** 推动浮体做摇荡运动的力（矩）不仅要推动浮体运动，还要推动浮体周围的流体运动。因此，要使浮体在海水中获得（角）加速度，需要施加额外的力（矩），称为附加惯性力（矩）。附加惯性力（矩）对应产生一个虚拟质量（虚拟转动惯量），即为附加质量（附加转动惯量）。

**兴波阻尼力（矩）** 浮体在海水中做摇荡运动时，会兴起波浪，从而产生对浮体摇荡运动的阻力（矩），称为兴波阻尼力（矩）。兴波阻尼力（矩）与摇荡运动的（角）速度成正比，方向相反，比例系数称为兴波阻尼系数。

**静水恢复力** 浮体在海水中做垂荡运动时，会受到使浮体回到平衡位置的作用力，称为静水恢复力。静水恢复力实际上是由浮体在垂荡运动时所受到的浮力变化引起的。

**静水恢复力矩** 浮体在海水中做纵摇运动时，会受到使浮体转正的力矩，称为静水恢复力矩，其大小与浮体相对于静水面的转角成正比，比例系数称为静水恢复力矩系数

## 二、模型假设

在建立模型之前，我们给出以下几点合理假设：

- 1、假设海水是无粘无旋的，并且本文将海平面视作水平面，以便于坐标系的建立。
- 2、波浪能转化装置各部分材料密度相同、质量分布均匀、结构稳定。
- 3、海浪的效果视作非线性弹簧振子与阻尼器的组合，从而模拟波浪激励力与波浪激励力矩。<sup>[2]</sup>
- 4、由于  $\theta$  角度较小，因此浮力  $F^*$  不受系统转动影响，从而简化兼有垂荡运动与纵摇运动的模型。

### 三、 符号说明

表 1 符号说明

符号	符号描述	单位
$F^*$	浮力大小	N
$m_1$	浮子质量	kg
$m_2$	振子质量	kg
$m'$	附加惯性力产生的虚拟质量	kg
$k_1$	将静水恢复力视作的非线性弹簧劲度系数	N/m
$C_1$	由兴波阻尼力带来的阻尼系数	N·s/m
$\omega$	波浪频率	s <sup>-1</sup>
$f_c^{(1)}$ 、 $f_k^{(1)}$	浮子收到的阻尼力、恢复力	N
$f_c^{(2)}$ 、 $f_k^{(2)}$	振子收到的阻尼力、恢复力	N
$x_1$ 、 $x_2$	浮子、振子相对于海平面的位移	m
$\theta_1$ 、 $\theta_2$	浮子、振子角位移	rad
$\dot{x}_1$ 、 $\dot{x}_2$	浮子、振子垂荡运动速度	m/s
$J_1$ 、 $J_2$	浮子、振子转动惯量	kg·m <sup>2</sup>
$P$	装置平均输出功率	w
$C_{PTOz}$	直线阻尼器阻尼系数	N·s/m
$C_{PTOx}$	旋转阻尼器阻尼系数	N·s/m
$v_z$	垂荡运动中浮子与振子相对速度	m/s
$v_x$	纵摇运动中浮子与振子相对角速度	rad/s

## 四、问题分析

### 4.1 问题一分析

问题一属于双自由度有阻尼受迫振动的数学问题，通常采用受力分析以及牛顿第二定律的方式建立微分方程组进行求解。本题要求计算两种不同情况下浮子和振子的垂荡位移和速度，具体需要：

- (1) 假设海水无粘无旋，同时浮子在波浪中只做垂荡运动。依题目描述，我们将整个系统合理简化成四个部分，分别是浮子、振子、海浪和 PTO 系统；
- (2) 绘制受力分析图，建立系统运动的微分方程组；
- (3) 将两种不同情况下的阻尼系数代入微分方程组，并通过 Runge-Kutta 法求解；

### 4.2 问题二分析

在问题二中，需要根据已知条件确定最优阻尼系数，以实现最大平均输出功率。该问题可以转化为多元函数求极值的问题，对第一种定常阻尼的情况，我们采用在  $[0,100000]$  中等间隔取点；对第二种非定常阻尼的情况，我们在  $[0,1]$  和  $[0,100000]$  之间等间隔取点，通过建立 P-C 曲线观察趋势，并找到最大值存在的区间。在此基础上，通过粒子群算法代入两种不同情况，在估计区间内求出输出功率最优值以及相应的阻尼系数或相关参数。

### 4.3 问题三分析

在问题一的基础上，浮子与振子增加了绕质心轴的转动，使问题三成为四自由度系统。因此在本题中同样需要通过受力分析与力矩平衡相关知识求解。本题要求对浮子与振子的垂荡和纵摇两种不同运动状态进行分析求解，具体需要：

- (1) 通过寻找质心求出浮子和振子对应的转动惯量；
- (2) 根据力矩平衡，给出浮子与振子绕质心轴转动与垂荡运动的微分方程组；
- (3) 参考问题一中，将二阶微分方程组降阶后，采用代数精度为 4、5 阶的 Runge-Kutta 法对微分方程组进行求解。

### 4.4 问题四分析

在问题二情况一的基础上，浮子与振子增加纵摇运动，通过增加计算纵摇系统的平均输出功率即可将问题转化为多元函数求极值的问题。采用粒子群算法使直线阻尼器和旋转阻尼器均为  $[0,100000]$  的可行域内求近似最优解，并得到相应的直线阻尼器与旋转阻尼器的阻尼系数。

## 五、问题求解

### 5.1 问题一求解

#### 5.1.1 模型建立

对于整个系统的受力分析如图所示

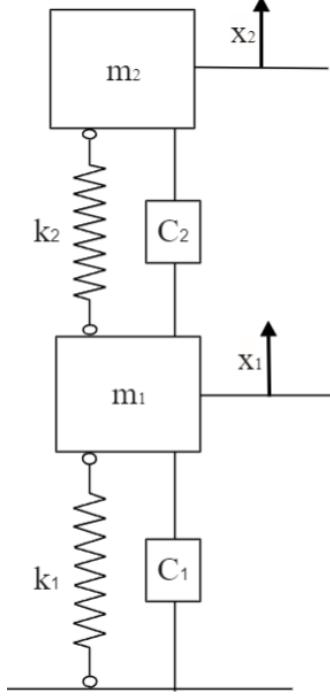


图3 系统受力分析图

对  $m_1, m_2$  进行受力分析:

$$\begin{cases} m_2 \cdot \ddot{x}_2 + f_c^{(2)} + f_k^{(2)} = 0 \\ (m' + m_1) \cdot \ddot{x}_2 + f_c^{(1)} + f_k^{(1)} = F(t) = f \cdot \cos \omega t \end{cases} \quad (1)$$

其中  $m'$  为附加惯性力所产生的虚拟质量,  $f \cdot \cos \omega t$  为波浪激励力。波浪视为非线性弹簧与阻尼器的组合,  $k_1$  来源为静水恢复力,  $c_1$  来源于兴波阻尼力。[3] [4]

振子所受到的阻尼力与回复力为:

$$\begin{cases} f_c^{(2)} = c_2 \cdot (\dot{x}_2 - \dot{x}_1) \\ f_k^{(2)} = k_2 \cdot (x_2 - x_1) \end{cases} \quad (2)$$

浮子所受的阻尼力与恢复力:

$$\begin{cases} f_c^{(1)} = c_1 \cdot \dot{x}_1 - c_2 \cdot (\dot{x}_2 - \dot{x}_1) = (c_1 + c_2) \cdot \dot{x}_1 - c_2 \cdot \dot{x}_2 \\ f_k^{(1)} = -k_2 \cdot (x_2 - x_1) + F^* - G \end{cases} \quad (3)$$

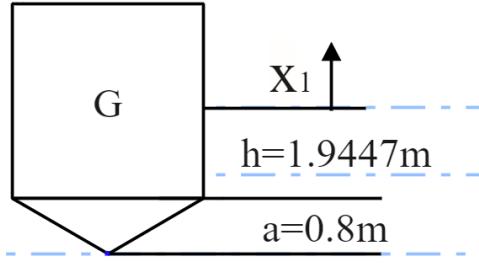


图4 所受浮力示意图

对系统所受浮力进行分析, 如图四所示, 系统所受浮力是随所处水位不同分段变化的, 我们假设海平面静止,  $x_1$  是相对于初始海平面的位移, 从而得到  $F^*$  的方程组:

$$F^* = \begin{cases} \rho g \cdot \left[ \frac{\pi}{3} \cdot R^2 a + \pi R^2 \cdot (h - x_1) \right] & x_1 < h \\ \rho g \cdot \frac{\pi}{3} \cdot \frac{(a + h - x_1)^3}{a^2} \cdot R^2 & x_1 > h \end{cases}$$

### 5.1.2 模型求解

对于上述二阶微分方程组 (3), 采用降阶法将其转为一阶的微分方程组:

令  $y_1 = x_1, \quad y_2 = \dot{x}_1, \quad y_3 = x_2, \quad y_4 = \dot{x}_2$

$$\begin{cases} dy_1 = y_2 \\ dy_2 = -\frac{(c_1 + c_2)}{m_1 + m'} \cdot y_2 + \frac{c_2}{m_1 + m'} \cdot y_4 + \frac{k_2 \cdot (y_3 - y_1) - F^* + G}{m_1 + m'} + \frac{f \cdot \cos \omega t}{m_1 + m'} \\ dy_3 = y_4 \\ dy_4 = -\frac{c_2}{m_2} \cdot (y_4 - y_2) - \frac{k_2}{m_2} \cdot (y_3 - y_1) \end{cases} \quad (4)$$

对于上述常微分方程组 (4), 采用代数精度为 4、5 阶的 Runge-Kutta 法求解, 并代入两种情况, 得到结果如下所示:

- (1) 直线阻尼器的阻尼系数为  $10000 N \cdot s/m$
- (2) 直线阻尼器的阻尼系数与浮子和振子的相对速度的绝对值幂成正比, 比例系数取 10000, 幂指数取 0.5:  $c = 10000 \sqrt{|\dot{x}_1 - \dot{x}_2|}$

具体数值见表二表三, 从结果来看, 系统中浮子和振子呈现出持续时间较短的暂态振动, 这是含阻尼的振动系统中真实存在的现象, 可以验证我们模型的科学性。

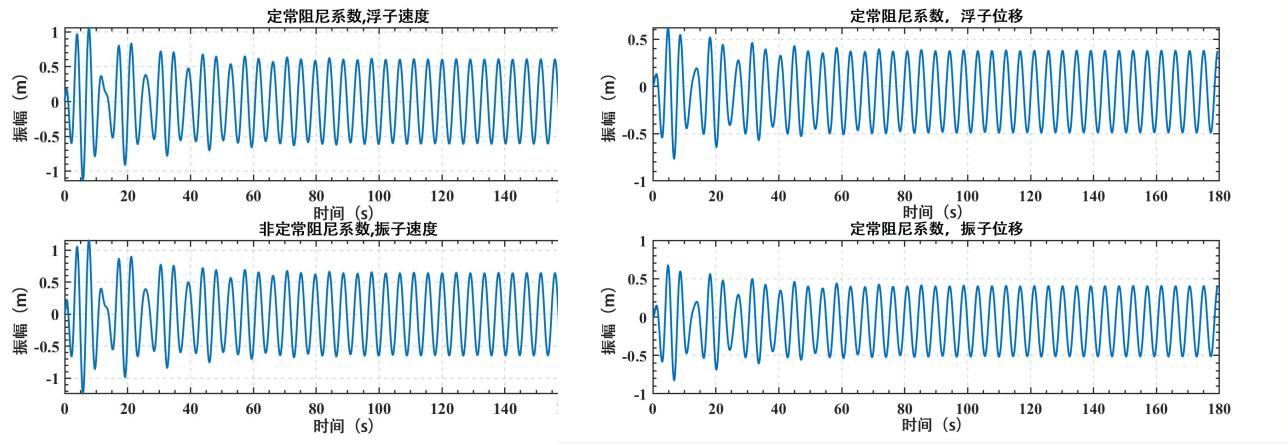


图 5 定常阻尼系数速度及垂荡位移

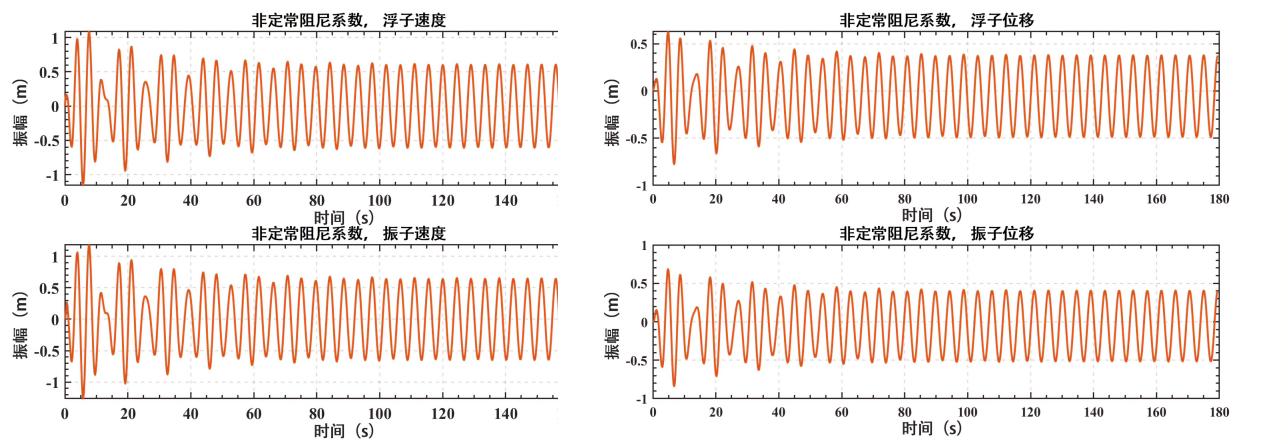


图 6 非定常阻尼系数速度及垂荡位移

表 2 定常阻尼系数下浮子和振子的垂荡位移及速度

时间 (s)	浮子垂荡位移 (m)	浮子速度 (m/s)	振子垂荡位移 (m)	振子速度 (m/s)
10	-0.211162	-0.638925	-0.228173	-0.690102
20	-0.623272	-0.237993	-0.664420	-0.267312
40	0.239556	0.316016	0.251752	0.336580
60	-0.365837	-0.477379	-0.382367	-0.513582
100	-0.138245	-0.603617	-0.138628	-0.642276

表 3 非定常阻尼系数下浮子和振子的垂荡位移及速度

时间 (s)	浮子垂荡位移 (m)	浮子速度 (m/s)	振子垂荡位移 (m)	振子速度 (m/s)
10	-0.225004	-0.649025	-0.249396	-0.694883
20	-0.641673	-0.249305	-0.688586	-0.270871
40	0.225012	0.300473	0.237661	0.318463
60	-0.377269	-0.488443	-0.398745	-0.519827
100	-0.142541	-0.608606	-0.147808	-0.648179

## 5.2 问题二求解

### 5.2.1 模型建立

阻尼系数的大小对系统的运动性能有显著影响，同时会引起输出功率的变化。由于系统在前 20 个周期中会出现上文提到的暂态振动现象，我们计算的是系统稳定后的平均输出功率，因此本题中采用了第 20 至第 40 个周期内振子与浮子的运动情况，平均输出功率即为垂荡浮子的输出功率值，即阻尼力在单位时间内做功的大小：[5]

$$P = \frac{1}{T} \cdot \int_0^T F v dt = \frac{1}{T} \cdot \int_0^T C_{PTO_z} v_z^2 dt \quad (5)$$

以情况一为例，我们在阻尼系数区间  $[0,100000]$  中等间隔选取 2000 个点，建立平均输出功率与阻尼系数的关系曲线（P-C 曲线），观察期趋势并估计出最优解存在的区间为  $[30000,40000]$ ，其中  $v_z = |\dot{x}_1 - \dot{x}_2|$  为垂荡运动中浮子与振子的相对速度：

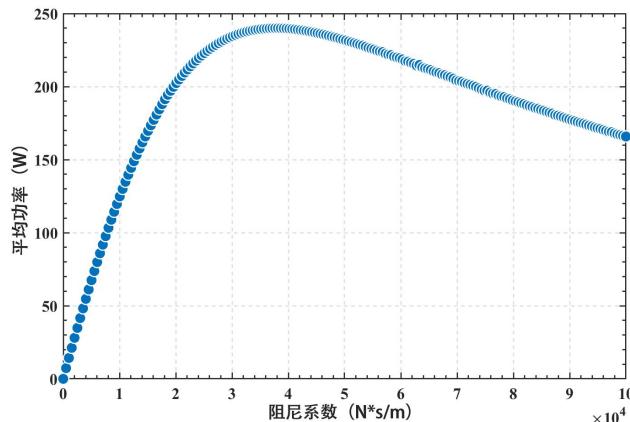


图 7 输出功率-阻尼系数趋势曲线

从上图中，我们发现当系统参数和外界环境条件不变时，平均输出功率随着阻尼系数呈现先增后减的趋势，从而可以得出存在最佳阻尼系数值与最大平均功率。

### 情况二中的阻尼系数

$$C_{PTOz} = c_0 \cdot |\dot{x}_1 - \dot{x}_2|^k \quad (6)$$

其中  $c_0 \in [0, 100000]$ ,  $k \in [0, 1]$

### 5.2.2 粒子群优化算法

在编程解决情况一和情况二中的优化问题时，由于情况二是多元函数全局极值求解，粒子群优化算法适用于连续函数极值问题，具有参数少、算法简单并且对于非线性、多峰问题均有较强的全局搜索能力，同时能够更快地收敛于最优解。因此，我们采用粒子群算法在可行域中进行求解，情况二的具体算法流程图如下所示。其中  $x_1$  是  $C_0$  的坐标,  $x_2$  是  $k$  的坐标。

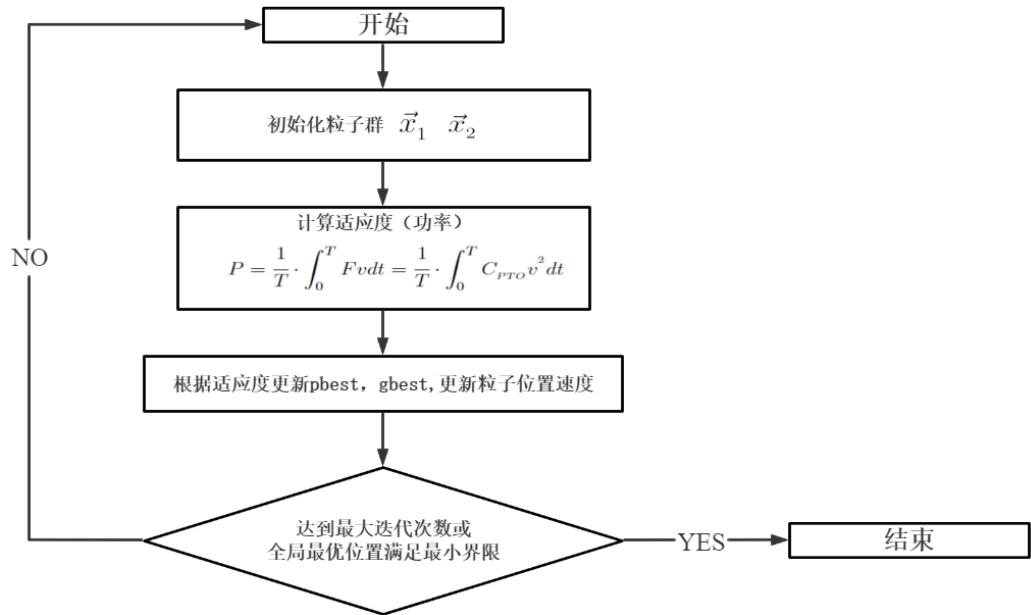
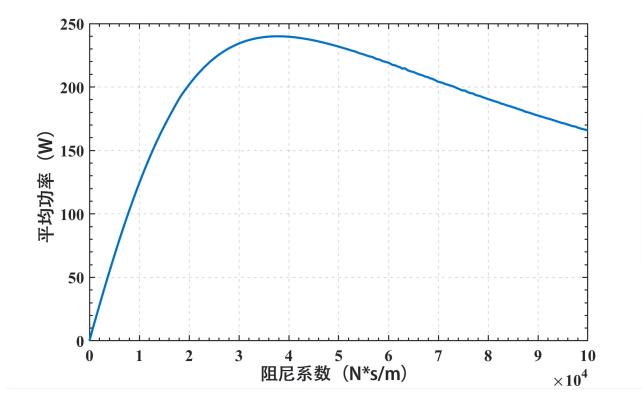


图 8 粒子群优化算法流程图

### 5.2.3 模型求解

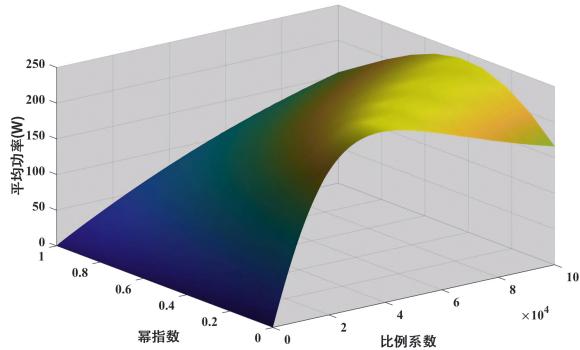
将题目中相关数据代入上述模型中，运用粒子群优化算法求解分别得出两种情况的结论。对于情况一：



平均输出功率最优值 (w)	相应阻尼系数值 (N*s/m)
240.0065	37775.7914

图 9 定常阻尼系数 P-C 关系图 + 定常阻尼系数结果

对于情况二：



平均输出功率最优值 (w)	比例系数 $C_0$	幂指数 k
240.5146	90787.6639	0.36917

图 10 非定常阻尼系数 P-C0、k 关系图 + 非定常阻尼系数结果

其中，情况二在迭代 30 次后收敛，因此算法速度较快。算法中平均功率（适应度）与迭代次数曲线如下所示：

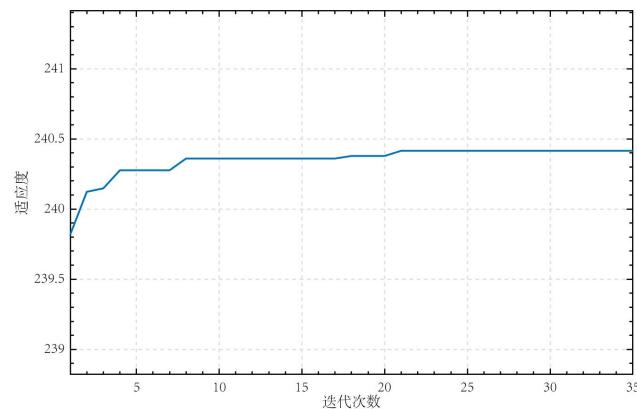


图 11 平均功率-迭代次数曲线 1

## 5.3 问题三求解

### 5.3.1 模型建立

我们先对浮子与振子绕质心轴转动和质心垂直运动分别建立模型 [6][7]，并在建立微分方程模型后，再计算浮子与振子的转动惯量。本题中规定顺时针方向为正方向，浮子角位移为  $\theta_1$ ，振子角位移为  $\theta_2$ ，如图 12 所示：

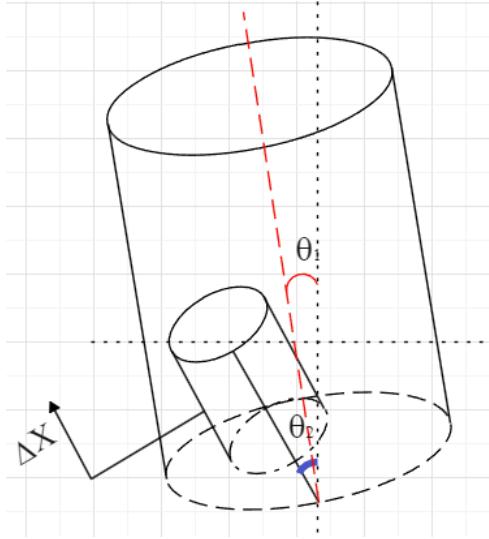


图 12 振子浮子受力分析图

对浮子分析：浮子转动后，存在惯性矩  $(J_1 + J_3)\ddot{\theta}_1$  静水恢复力矩为  $C_H \cdot \theta_1$ ，摇荡兴波阻尼力矩为  $C_Y \cdot \dot{\theta}_1$ ，振子对浮子的扭转弹簧力矩为

$$\tau_s = k_N \cdot (\theta_1 - \theta_2) \quad (7)$$

扭转阻尼力矩为

$$\tau_z = k_z (\dot{\theta}_1 - \dot{\theta}_2) \quad (8)$$

外力矩：

$$\tau_{water} = L \cdot \cos \omega t \quad (9)$$

因此浮子转动的常微分方程为：

$$(J_1 + J_3) \cdot \ddot{\theta}_1 + C_H \cdot \theta_1 + C_Y \cdot \dot{\theta}_1 + k_N (\theta_1 - \theta_2) + k_z (\dot{\theta}_1 - \dot{\theta}_2) = L \cdot \cos \omega t \quad (10)$$

上式中要用到浮子对自身质心的转动惯量  $J_1$ ，附加转动惯量  $J_3$

如图所示，以原点 O 为中心建立空间直角坐标系，我们仅计算浮子外壳表面积质量，具体将其分为三个部分：

$$S_{11} = \pi r^2 = 3.14 \quad (11)$$

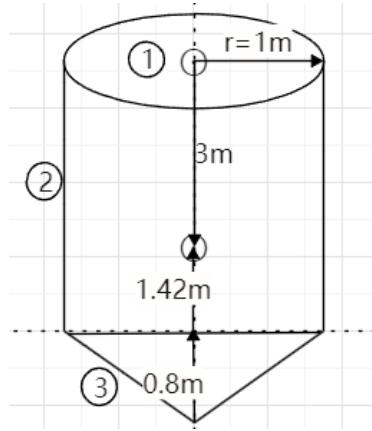


图 13 浮子简化后的质心图

$$S_{12} = 2\pi rh = 2 \times 3.14 \times 3 = 18.84 \quad (12)$$

$$S_{13} = \pi rL = 3.14 \times 1.28 = 4.02 \quad (13)$$

因此总表面积:

$$S_t = S_{11} + S_{12} + S_{13} = 26 \quad (14)$$

$$m_{11} = m \times \frac{S_1}{S} = 578.66 \quad (15)$$

$$m_{12} = m \times \frac{S_2}{S} = 3525.98 \quad (16)$$

$$m_{13} = m \times \frac{S_3}{S} = 753.36 \quad (17)$$

从而有: 第一部分重心在空间直角坐标系的坐标为:(0,0,3), 相应地第二第三部分的重心分别为:(0,0, $\frac{3}{2}$ ) 和 (0,0, $-\frac{1}{5}$ )。

$$x_m = \frac{\sum_i x_i m_i}{\sum_i m_i} = 0 \quad (18)$$

$$y_m = \frac{\sum_i y_i m_i}{\sum_i m_i} = 1.42 \quad (19)$$

从而得到浮子的质心  $z_c$  为 (0, 0, 1.42), 对于浮子上一点 (x,y,z), 其对于质心的位置矢量

$$\vec{r} = (x, y, z - z_c), \quad \vec{e}_\omega = (1, 0, 0) \quad (20)$$

从而

$$J_1 = \oint (\vec{e}_w \times \vec{r})^2 \cdot \rho dS = \oint [y^2 + (z - z_c)^2] \cdot \rho dS \quad (21)$$

对振子分析，振子转动时产生惯性矩  $J_2 \cdot \ddot{\theta}_2$ ，其中  $J_2$  为此时振子相对 O 点所在轴的转动惯量。振子与浮子存在相互作用，其产生的：

扭弹簧力矩为：

$$\tau_s = k_N \cdot (\theta_2 - \theta_1) \quad (22)$$

扭阻尼力矩为：

$$\tau_Z = k_Z \cdot (\dot{\theta}_2 - \dot{\theta}_1) \quad (23)$$

先前计算中，我们考虑科里奥利力——进行直线运动的质点由于惯性相对于旋转体系偏移的一种描述，源于物体运动所具有的惯性。在振子系统中科氏力也存在，如下图所示：

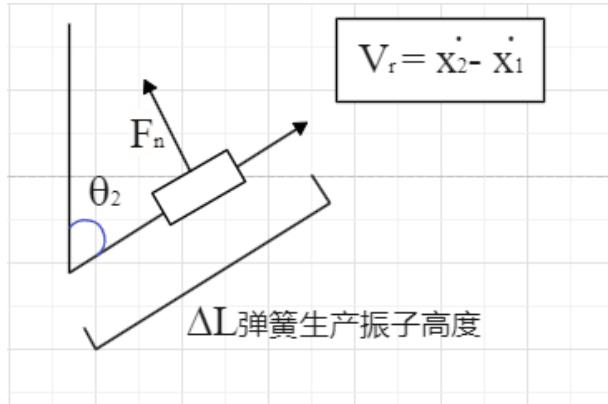


图 14 科氏力示意图

而由于我们参考系是垂直于海平面的定轴，因此不存在科氏力。

基于上述计算得到浮子与振子绕质心转动的微分方程组：

$$J_2 \cdot \ddot{\theta}_2 + k_N(\theta_2 - \theta_1) + k_Z(\dot{\theta}_2 - \dot{\theta}_1) = 0 \quad (24)$$

下求振子对质心的转动惯量，考虑振子运动状态与初始情况，弹簧原长为 0.5m，受到振子重力影响的形变量

$$l' = \frac{m_2 g}{k} = \frac{2433 \times 9.8}{80000} = 0.2980425m \quad (25)$$

原点到振子的距离

$$L_0 = 0.5 - 0.2980425 = 0.2019575m \quad (26)$$

$x_2$  为振子以向上为正方向的位移，即弹簧伸长量为  $(x_2 - x_1)$ ，

$$\Delta L = L_0 + x_2 - x_1 \quad (27)$$

从而振子对 O 点所在轴的转动惯量

$$J_2 = \Delta L^2 \cdot m_2 + m_2 \cdot y_m^2 = (L_0 + x_2 - x_1)^2 \cdot m_2 + m_2 \cdot y_m^2 \quad (28)$$

基于上述计算得到浮子与振子绕质心转动的微分方程组：

$$\begin{cases} (J_1 + J_3)\ddot{\theta}_1 + C_h\dot{\theta}_1 + C_y\dot{\theta}_1 + k_n(\theta_1 - \theta_2) + k_z(\dot{\theta}_1 - \dot{\theta}_2) = L \cos \omega t \\ J_2\ddot{\theta}_2 + k_n(\theta_2 - \theta_1) + k_z(\dot{\theta}_2 - \dot{\theta}_1) = 0 \end{cases} \quad (29)$$

### 5.3.2 模型求解

对于上述二阶微分方程组 (29)，同样采用降阶法将其转为一阶微分方程组：

$$\text{令 } y_1 = x_1, y_2 = \dot{x}_1, y_3 = x_2, y_4 = \dot{x}_2$$

$$y_5 = \theta_1, y_6 = \dot{\theta}_1, y_7 = \theta_2, y_8 = \dot{\theta}_2$$

从而微分方程组可以转化为：

$$\begin{cases} dy_1 = y_2 \\ dy_2 = -\frac{(c_1 + c_2)}{m_1 + m'} \cdot y_2 + \frac{c_2}{m_1 + m'} \cdot y_4 + \frac{k_2 \cdot (y_3 - y_1) - F^* + G}{m_1 + m'} + \frac{f \cdot \cos \omega t}{m_1 + m'} \\ dy_3 = y_4 \\ dy_4 = -\frac{c_2}{m_2} \cdot (y_4 - y_2) - \frac{k_2}{m_2} \cdot (y_3 - y_1) \\ dy_5 = y_6 \\ dy_6 = \frac{-c_h \cdot y_5}{J_1 + J_3} - \frac{c_y y_6}{J_1 + J_3} - \frac{k_n(y_5 - y_7)}{J_1 + J_3} - \frac{k_z(y_6 - y_8)}{J_1 + J_3} + \frac{L \cos \omega t}{J_1 + J_3} \\ dy_7 = y_8 \\ dy_8 = \frac{-1}{J_2} \cdot (k_n(y_7 - y_5) + k_z(y_8 - y_6) + 2m_2(y_4 - y_2) \cdot y_8 \cdot (L_0 + y_3 - y_1)) \end{cases} \quad (30)$$

对于上述常微分方程组 (30)，同样采用代数精度为 4、5 阶的 Runge-Kutta 法求解，得到浮子与振子的垂荡位移与速度和纵摇角位移与角速度如图 15, 16 所示：

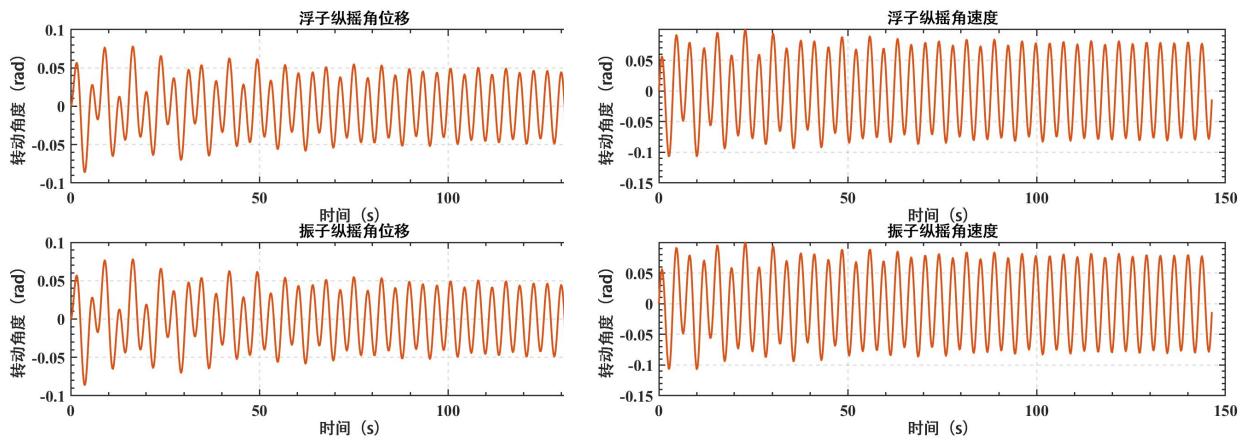


图 15 浮子纵摇角位移及纵摇角速度

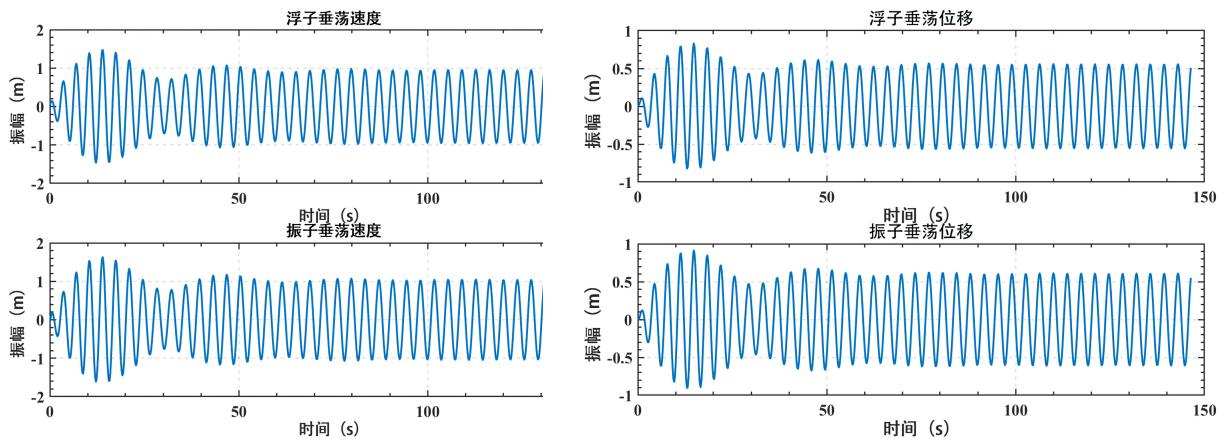


图 16 浮子垂荡位移及速度

具体数值如表 4、表 5 所列：

表 4 浮子位移速度与角位移速度

时间 (s)	浮子位移 (m)	浮子速度 (m/s)	浮子角位移 (rad)	浮子角速度 (rad/s)
10	-0.528276001	0.969851308	0.013066196	-0.106154590
20	-0.704814679	-0.269305682	0.018167709	0.006556911
40	0.369361258	0.757579522	-0.032216291	-0.023021602
60	-0.320675265	-0.721761812	0.035109418	0.042901134
100	-0.050153973	-0.946689651	0.019943543	0.073055580

**表 5 振子位移速度与角位移速度**

时间 (s)	振子位移 (m)	振子速度 (m/s)	振子角位移 (/rad)	振子角速度 (rad/s)
10	-0.598729109	1.038200629	0.013063118	-0.106333723
20	-0.772293806	-0.319025579	0.018188681	0.006574767
40	0.392653033	0.844962276	-0.032269625	-0.023069101
60	-0.341426215	-0.799321187	0.035142817	0.043167895
100	-0.042611762	-1.036525949	0.019964200	0.072701985

## 5.4 问题四求解

### 5.4.1 模型建立

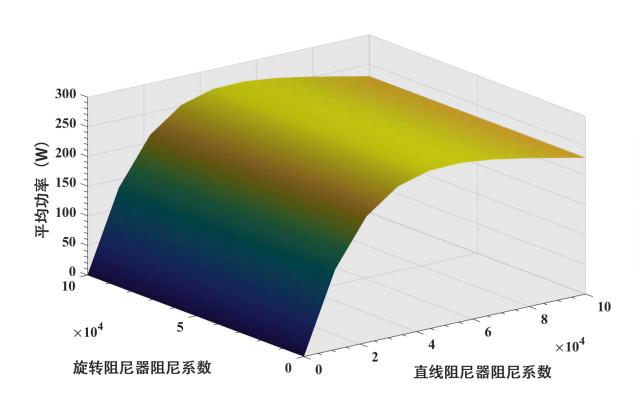
在问题二的模型基础上，波浪能转化装置增加纵摇状态的运动情况，在考虑平均输出功率时，增加纵摇系统的平均功率。参考问题二中相关学者给出的公式，该平均功率即为相对切向速度与外力的乘积，而相对切向速度  $v_x$  为相对角速度与转动半径的乘积，从而得到新的平均输出功率与直线阻尼器阻尼系数、旋转阻尼器阻尼系数的关系式如下：

$$P = P_1 + P_2 = \frac{1}{T} \cdot \int_0^T C_{PTOz} v_z^2 dt + \frac{1}{T} \cdot \int_0^T C_{PTOx} v_x^2 dt \quad (31)$$

其中  $C_{PTOx}$  为旋转阻尼器的阻尼系数，相对切向速度  $v_x = |(\dot{\theta}_1 - \dot{\theta}_2) \cdot \Delta L|$

### 5.4.2 模型求解

同问题二的情况二中采用粒子群优化算法，在直线阻尼器与旋转阻尼器的阻尼系数均为 [0,100000] 的可行域中搜索最优解，其结果如图 17 所示：



输出功率最优值 (w)	直线阻尼系数	旋转阻尼系数
268.5576	51166.9986	79728.5923

图 17 非定常阻尼系数 P-C0、k 关系图 + 非定常阻尼系数结果

其中，该算法在迭代 10 次后较快地收敛于最优解，如图 18 所示：

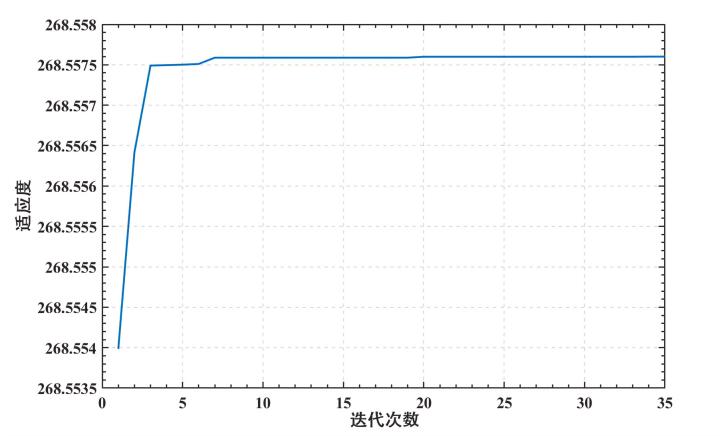


图 18 平均功率-迭代次数曲线 2

## 六、模型评价

### 6.1 模型优点

- 1、模型建立过程中充分考虑了实际情况，使模型更加真实、可靠、科学。
- 2、求解过程中采用降阶法对复杂微分方程系统进行简化，并使用 4、5 阶的 Runge-Kutta 法进行数值解的求解，具有高精度、高运行速度、稳定性强的优点。
- 3、本文采用使用粒子群优化算法，其算法简单、收敛速度快，有较强的全局搜索能力。

### 6.2 模型缺点

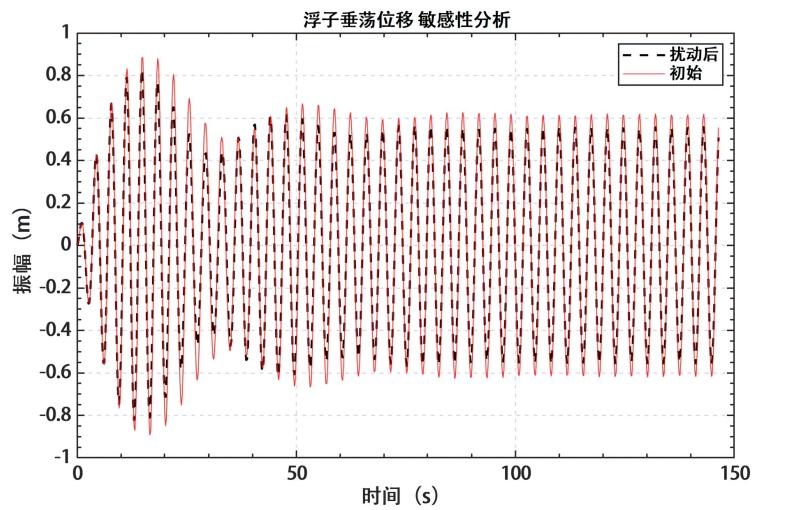
问题四优化的过程中没有考虑转动对垂荡系统的影响，主要为转动产生的离心力和振子重力在沿轴方向的分力变化。但是由于转动速度与角度过小，产生极小的向心加速度；重力也可近似于始终沿轴作用，对于最后结果的求解影响并不大。

### 6.3 稳定性分析

本文采用第三问中的复杂模型，由于微分方程系统是相互作用的，因此只对浮子的垂荡位移进行稳定性分析，将波浪能装置的核心参数以及海浪的阻尼系数进行了 5% 的扰动，从而得到最终结果：在平稳后，位移变化了 12% ( $0.6457/0.5500$ )。此外，波形图仍旧保持原来的形状，未发生突变，因此我们的模型具有很强的稳定性。

表 6 模型敏感性分析参数范围

	阻尼系数	转动惯量	力矩振幅	弹簧劲度系数	扭转弹簧刚度
扰动分布	$C \pm 5\%$	$J \pm 5\%$	$L \pm 5\%$	$k \pm 5\%$	$k_N \pm 5\%$



**图 19 浮子垂荡位移敏感性分析**

因此根据微分方程系统的连续性和相关性，我们可以推断出在扰动之下其它系统也会保持一个稳定状态。

#### 6.4 模型推广

- (1) 本文中未考虑纵摇对垂荡系统的影响，后续研究中将对转动产生的离心力和振子重力在沿轴方向上的分力变化加入到垂荡系统中，从而使我们的模型更具有普适性，以应对更加复杂的海面情况。
- (2) 参考微幅波理论并将其运用到海浪的建模中，从而获得更真实的效果。
- (3) 对波浪能发电装置本身参数的设置进行调整与选优。

## 参考文献

- [1] Antonio, F. D. O. (2010). Wave energy utilization: A review of the technologies. *Renewable and sustainable energy reviews*, 14(3), 899-918.
- [2] 张弘超. 海洋波浪发电系统振荡浮子结构的优化研究 [D]. 清华大学, 2010.
- [3] 李增亮, 范梦浩, 王萌, 郑杰. 浮子式波浪能转换装置功率匹配试验研究 [J]. 石油机械, 2019, 47(05):85-90.
- [4] 王帝. 振荡浮子式波浪能发电装置水动力性能数值研究 [D]. 上海交通大学, 2019. DOI:10.27307/d.cnki.gsjtu.2019.000765.
- [5] 王淇. 一种新型浮式风浪能混合利用系统概念设计与性能分析 [D]. 哈尔滨工程大学, 2016.
- [6] 郑永红, 游亚戈, 沈永明, 吴必军, 刘荣. 状态空间模型在海洋波能转换系统中的应用 ii. 单个矩形振荡浮子装置运动受力分析的解析方法 [J]. 海洋学报 (中文版), 2004(01):113-124.
- [7] 王树春, 车仁炜, 胡长胜, 张月红, 游玮. 浮筒式海浪发电机理研究 [J]. 太阳能学报, 2007(11):1266-1271.

## 附录 A 问题一 matlab 代码

```
close all
clc;
clear;
global m1 m2 k omega f m3
m3 = 1335.535;
m1 = 4866 + m3;
m2 = 2433;
k = 80000;
omega = 1.4005;
f = 6250;
Time = 2*pi/omega;
Timetotal = 40*Time;
dt = 0.2;
timespan = 0:dt:Timetotal;
y0 = [0;0;0;0];
[timespan1,solution1] = ode45(@constantode1,timespan,y0);
[timespan2,solution2] = ode45(@noconstantode2,timespan,y0);
figure(1)
set (gcf,'Position',[100 100 800 400])
subplot(211)
plot(timespan1,solution1(:,1));
xlabel('time/s')
ylabel('displacement/m')
title('定常阻尼系数,浮子位移')
subplot(212)
plot(timespan1,solution1(:,3));
xlabel('time/s')
ylabel('displacement/m')
title('非定常阻尼系数,振子位移')
figure(2)
set (gcf,'Position',[100 100 800 400])
subplot(211)
plot(timespan1,solution1(:,2));
xlabel('time/s')
ylabel('displacement/m')
title('定常阻尼系数,浮子速度')
subplot(212)
plot(timespan1,solution1(:,4));
xlabel('time/s')
ylabel('displacement/m')
title('非定常阻尼系数,振子速度')
for time =[10 20 40 60 100]
n = time/dt +1;
d1_fuzi = solution1(n,1);
```

```

v1_fuzi = solution1(n,2);
d1_zhenzi = solution1(n,3);
v1_zhenzi = solution1(n,4);
fprintf('\t时间为%ds时, 浮子位移为%f/m, 速度为%f/(m/s); 振子位移为%f/m, 速度为%f/(m/s).\n', time,d1_fuzi,v1_fuzi,
end
disp('第二种情况: 非定常阻尼系数')
for time =[10 20 40 60 100]
    n = time/dt +1;
    d1_fuzi = solution2(n,1);
    v1_fuzi = solution2(n,2);
    d1_zhenzi = solution2(n,3);
    v1_zhenzi = solution2(n,4);
    fprintf('\t时间为%ds时, 浮子位移为%f/m, 速度为%f/(m/s); 振子位移为%f/m, 速度为%f/(m/s).\n', time,d1_fuzi,v1_fuzi,
end
a = max(solution2(:,1));
b = max(solution2(:,2));
c = max(solution2(:,3));
d = max(solution2(:,4));
fprintf('非常阻尼浮子最大位移为%f/m, 最大速度为%f/(m/s); 振子最大位移为%f/m, 最大速度为%f/(m/s).\n', a,b,c,d)
a = max(solution1(:,1));
b = max(solution1(:,2));
c = max(solution1(:,3));
d = max(solution1(:,4));
fprintf('常阻尼浮子最大位移为%f/m, 最大速度为%f/(m/s); 振子最大位移为%f/m, 最大速度为%f/(m/s).\n', a,b,c,d)
function differentitaly =constantode1(t,y)
global m1 m2 k omega f m3
c2=10000;
a=0.8;
c1=656.3616;
h=1.9447;
differentitaly = zeros(4,1);
differentitaly(1) = y(2);
if y(1)> h
    F=1025*9.8/(3*(a^2))*pi*(a+h-y(1))^3;
else
    F=1025*9.8*(pi*a/3+pi*(h-y(1)));
end
F=-F+(m1+m2-m3)*9.8;
differentitaly(2) = -(c2+c1)/m1*y(2)+c2/m1*y(4)+(k*(y(3)-y(1))-F)/m1+f/m1*cos(omega*t);
differentitaly(3) = y(4);
differentitaly(4) = -c2/m2*(y(4)-y(2))-k/m2*(y(3)-y(1));
end
function differentitaly =noconstantode2(t,y)
global m1 m2 k omega f m3
c2=10000*sqrt(abs(y(2)-y(4)));
a=0.8;
c1=656.3616;

```

```

h=1.9447;
differentitaly = zeros(4,1);
differentitaly(1) = y(2);
if y(1)> h
    F=1025*9.8/(3*(a^2))*pi*(a+h-y(1))^3;
else
    F=1025*9.8*(pi*a/3+pi*(h-y(1)));
end
F=-F+(m1+m2-m3)*9.8;
differentitaly(2) = -(c2+c1)/m1*y(2)+c2/m1*y(4)+(k*(y(3)-y(1))-F)/m1+f/m1*cos(omega*t);
differentitaly(3) = y(4);
differentitaly(4) = -c2/m2*(y(4)-y(2))-k/m2*(y(3)-y(1));
end

```

## 附录 B 问题二 (1) 求解代码

```

close all
clc;
clear;
global m1 m2 kzuni omega m3 c2 f1
m3 = 1165.992;
m1 = 4866 + m3;
m2 = 2433;
kzuni = 80000;
omega = 2.2413;
f1 = 4890;
T = 2*pi/omega;
Timetotal = 40*T;
dt = 0.1;
timespan = Timetotal/2:dt:Timetotal;
y0 = [0;0;0;0];
nc=200;
bilixishu=linspace(0,100000,nc);
P=zeros(size(bilixishu));
for i =1:nc
    c2=bilixishu(i);
    [tt1,yy1]=ode45(@constantode1,timespan,y0);
    vr=yy1(:,2)-yy1(:,4);
    P(i)=2*c2*trapz(tt1,vr.^2)/Timetotal;
end
scatter(bilixishu,P,'filled')%画一个散点图
nc = 100;
nmi =10;
[Canshu1,Canshu2]=meshgrid(linspace(0,100000,nc),linspace(0,1,nmi));
P2=zeros(size(Canshu1));

```

```

global bilixishu mizhishu
for i= 1:nmi
    for j = 1:nc
        bilixishu=Canshu1(i,j);
        mizhishu=Canshu2(i,j);
        [tt2,yy2]=ode45(@noconstantode2,timespan,y0);
        vr = yy2(:,2)-yy2(:,4);
        c2=bilixishu*(abs(vr)).^mizhishu;
        P2(i,j)=2*trapz(tt2,c2.*vr.^2)/Timetotal;
    end
    i
end
figure(2)
surf(Canshu1,Canshu2,P2);%画一个立体图
shading interp
camlight
lighting phong

function differentialy =constantode1(t,y)
global m1 m2 kzuni omega f1 m3 c2
a=0.8;
c1=167.8395;
h=1.9447;
differentialy = zeros(4,1);
differentialy(1) = y(2);
if y(1)> h
    F=1025*9.8/(3*(a^2))*pi*(a+h-y(1))^3;
else
    F=1025*9.8*(pi*a/3+pi*(h-y(1)));
end
F=-F+(m1+m2-m3)*9.8;
differentialy(2) = -(c2+c1)/m1*y(2)+c2/m1*y(4)+(kzuni*(y(3)-y(1))-F)/m1+f1/m1*cos(omega*t);
differentialy(3) = y(4);
differentialy(4) = -c2/m2*(y(4)-y(2))-kzuni/m2*(y(3)-y(1));
end
function differentialy =noconstantode2(t,y,bilixishu,mizhishu)
global m1 m2 kzuni omega f1 m3 c2 mizhishu bilixishu
a=0.8;
c1=167.8395;
h=1.9447;
differentialy = zeros(4,1);
differentialy(1) = y(2);
if y(1)> h
    F=1025*9.8/(3*(a^2))*pi*(a+h-y(1))^3;
else
    F=1025*9.8*(pi*a/3+pi*(h-y(1)));
end

```

```

F=-F+(m1+m2-m3)*9.8;
c2=bilixishu*abs(y(2)-y(4))^mizhishu;
differentialy(2) = -(c2+c1)/m1*y(2)+c2/m1*y(4)+(kzuni*(y(3)-y(1))-F)/m1+f1/m1*cos(omega*t);
differentialy(3) = y(4);
differentialy(4) = -c2/m2*(y(4)-y(2))-kzuni/m2*(y(3)-y(1));
end

```

## 附录 C 问题二 (2) 求解代码

```

close all;
clc;
format long
tic;
c23 = 2;
c13 = 0.75;
w = 0.2;
xlimit = [0,100000;...
           0,1];
popsize = 50;
maxgen = 35;
vrange = 0.3;
SolveTimes = 1;

popdimension = size(xlimit,1);
vlimit = vrange*[-xlimit(:,2)+xlimit(:,1),xlimit(:,2)-xlimit(:,1)];
xrecode = zeros(SolveTimes,popdimension);
fitrecord = zeros(SolveTimes,maxgen);

for times = 1:1:SolveTimes

    pop=
        repmat(xlimit(:,1)',popsize,1)+repmat(diff(xlimit'),popsize,1).*rand(popsize,popdimension);
    v =
        repmat(vlimit(:,1)',popsize,1)+repmat(diff(vlimit'),popsize,1).*rand(popsize,popdimension);

    fitness = zeros(popsize,1);
    for i=1:popsize
        fitness(i)=fcn(pop(i,:));
    end

    [Firstbestfitness,Firstbestindex]=max(fitness,[],'all','linear');
    gbestnum = pop(Firstbestindex,:);
    gbesfitness = Firstbestfitness;

```

```

pbestnum = pop;
pbestfitness = fitness;
clear Firstbestfitness Firstbestindex

%
for m=1:maxgen

    v = w*v + ...
        c13*rand(popsize,1).*(pbestnum -pop)+...
        c23*rand(popsize,1).*( repmat(gbestnum,popsize,1) - pop);

    vlimitbelow = repmat(vlimit(:,1)',popsize,1);
    vlimitup = repmat(vlimit(:,2)',popsize,1);
    bigger = v > vlimit(:,2)';
    v(bigger) = vlimitup(bigger);
    smaller = v < vlimit(:,1)';
    v(smaller) = vlimitbelow(smaller);

    pop = pop +v;

    poplimitbelow = repmat(xlimit(:,1)',popsize,1);
    poplimitup = repmat(xlimit(:,2)',popsize,1);
    bigger = pop > xlimit(:,2)';
    pop(bigger) = poplimitup(bigger);
    smaller = pop < xlimit(:,1)';
    pop(smaller) = poplimitbelow(smaller);

    for i=1:popsize
        fitness(i)=fcn(pop(i,:));
    end

    toreplacepop = (fitness > pbestfitness);
    pbestfitness(toreplacepop) = fitness(toreplacepop);
    pbestnum(toreplacepop,:) = pop(toreplacepop,:);

    [maxfitness,maxfitnessindex] = max(fitness,[],'all','linear');
    if maxfitness > gbesfitness
        gbesfitness = maxfitness;
        gbestnum = pop(maxfitnessindex,:);
    end

```

```

fitrecord(times,m)=gbesfitness;
xrecode(times,:)=gbestnum;
end
end
[PSObestfit,bestfitindex] = max(fitrecord(:,end),[],'all','linear');
PSObestnum = xrecode(bestfitindex,:);
disp('%%%%%%%%%(粒子群算法)%%%%%%');
disp(['最优值为：',num2str(PSObestfit)]);
disp(['最优解为：',num2str(PSObestnum)]);
toc
for i=1:1
    Fontsize = 10.5;
    LineWidth = 1.2;
    f1 = figure;
    f1.Visible = 'on';
    f1.Units = 'centimeters';
    f1.Position = [20 8 16 9];
    f1.Color = 'w';
    f1.NumberTitle = 'off';
    f1.Name = '迭代收敛图';
    f1.Resize = 'off';

    ax1=axes(f1);
    ax1.Box = 'off';
    ax1.Visible = 'on';
    ax1.NextPlot = 'add';
    ax1.Color= 'w';
    ax1.TickDir = 'in';
    ax1.FontUnits = 'points';
    ax1.FontSize = Fontsize;
    ax1.LineWidth = LineWidth;
    ax1.FontName = 'Times New Roman';
    ax1.FontWeight = 'bold';
    ax1.TitleFontSizeMultiplier = 1;
    ax1.LabelFontSizeMultiplier = 1;
    ax1.XColor = 'k';
    ax1.YColor = 'k';
    ax1.YAxisLocation = 'left';
    ax1.XAxisLocation = 'bottom';
    ax1.TickLength = [0.01,0.035];
    ax1.XDir = 'normal';
    ax1.YDir = 'normal';
    grid(ax1,'on');
    ax1.GridLineStyle = '--';
    ax1.GridAlpha = 0.85;

```

```

ax1.GridColor = [0.85,0.85,0.85];

x1 = 1:1:maxgen;
y1 = fitrecord(bestfitindex,:);
p1 = plot(ax1,x1,y1);
p1.Visible = 'on';
p1.LineWidth = LineWidth;
p1.Color = 'k';
p1.LineStyle = '-';
p1.LineJoin = 'round';
p1.Marker = 'none';
p1.MarkerIndices = 1:round(length(x1)/10):length(x1);
p1.MarkerSize = 6;
p1.MarkerEdgeColor = 'auto';
p1.MarkerFaceColor = 'none';
ax1.XScale = 'linear';
ax1.YScale = 'linear';
axis(ax1,[min(x1,[],'all'),max(x1,[],'all'),min(y1,[],'all') - 1,max(y1,[],'all') + 1]);
ax1.XTickMode = 'auto';
ax1.YTickMode = 'auto';
ax1.XTickLabelMode = 'auto';
ax1.YTickLabelMode = 'auto';
xlabel('迭代次数','FontSize',Fontsize,'FontName','宋体');
ylabel('适应度','FontSize',Fontsize,'FontName','宋体');
end

%适应度函数
function y=fcn(x)
global m1 m2 k omega f1 m3 c2 bilixishu mizhishu
m3 = 1165.992;
m1 = 4866 + m3;
m2 = 2433;
k = 80000;
omega = 2.2413;
f1 = 4890;
T = 2*pi/omega;
Ttotal = 40*T;
dt = 0.1;
tspan = Ttotal/2:dt:Ttotal;
y0 = [0;0;0;0];
bilixishu=x(:,1);
mizhishu=x(:,2);
[tt2,yy2]=ode45(@noconstantode2,tspan,y0);
vr=yy2(:,2)-yy2(:,4);
c2=bilixishu*abs(vr)).^mizhishu;
summ=2*trapz(tt2,c2.*vr.^2)/Ttotal;
y = summ;
end

```

```

function differentitaly =noconstantode2(t,y,bilixishu,mizhishu)
global m1 m2 k omega f1 m3 bilixishu mizhishu
a=0.8;
c1=167.8395;
h=1.9447;
differentitaly = zeros(4,1);
differentitaly(1) = y(2);
if y(1)> h
    F=1025*9.8/(3*(a^2))*pi*(a+h-y(1))^3;
else
    F=1025*9.8*(pi*a/3+pi*(h-y(1)));
end
F=-F+(m1+m2-m3)*9.8;
c2=bilixishu*abs(y(2)-y(4))^mizhishu;
differentitaly(2) = -(c2+c1)/m1*y(2)+c2/m1*y(4)+(k*(y(3)-y(1))-F)/m1+f1/m1*cos(omega*t);
differentitaly(3) = y(4);
differentitaly(4) = -c2/m2*(y(4)-y(2))-k/m2*(y(3)-y(1));
end

```

## 附录 D 问题三 matlab 代码

```

close all
clc;
clear;
global m1 m2 mfujia c1 k omega f seadensity gravity radius J1 J_fu C_huifu C_xingbo k_niuzhuan
k_zuni L
m1=4866;
mfujia=1028.876;
J1=zhuandongguanliangfuzi;
J_fu=7001.914;
c1=683.4558;
C_xingbo=654.3383;
C_huifu=8890.7;
m2=2433;
k=80000;
k_niuzhuan=250000;
k_zuni=1000;
omega=1.7152;
f=3640;
L=1690;
seadensity=1025;
gravity=9.8;
radius=1;
Time=2*pi/omega;
Timetotal=40*Time;

```

```

dt=0.2;
timespan=0:dt:Timetotal;
y0=[0;0;0;0;0;0;0];
[timespan1,solution1]=ode45(@constantode1,timespan,y0);
figure(1)
subplot(211)
set(gcf,'Position',[100 100 700 400])
plot(timespan1,solution1(:,1));
xlabel('时间 (s) ','FontName','宋体')
ylabel('振幅 (m) ','FontName','宋体')
title('浮子垂荡位移')
subplot(212)
plot(timespan1,solution1(:,3));
xlabel('时间 (s) ','FontName','宋体')
ylabel('振幅 (m) ','FontName','宋体')
title('振子垂荡位移')
figure(2)
set(gcf,'Position',[100 400 700 400])
subplot(211)
plot(timespan1,solution1(:,5));
xlabel('时间 (s) ','FontName','宋体')
ylabel('转动角度 (rad) ','FontName','宋体')
title('浮子纵摇角位移')
subplot(212)
plot(timespan1,solution1(:,7));
xlabel('时间 (s) ','FontName','宋体')
ylabel('转动角度 (rad) ','FontName','宋体')
title('振子纵摇角位移')
figure(3)
subplot(211)
set(gcf,'Position',[100 100 700 400])
plot(timespan1,solution1(:,2));
xlabel('时间 (s) ','FontName','宋体')
ylabel('振幅 (m) ','FontName','宋体')
title('浮子垂荡速度')
subplot(212)
plot(timespan1,solution1(:,4));
xlabel('时间 (s) ','FontName','宋体')
ylabel('振幅 (m) ','FontName','宋体')
title('振子垂荡速度')
figure(4)
set(gcf,'Position',[100 400 700 400])
subplot(211)
plot(timespan1,solution1(:,6));
xlabel('时间 (s) ','FontName','宋体')
ylabel('转动角度 (rad) ','FontName','宋体')
title('浮子纵摇角速度')

```

```

subplot(212)
plot(timespan1,solution1(:,8));
xlabel('时间 (s)', 'FontName', '宋体')
ylabel('转动角度 (rad)', 'FontName', '宋体')
title('振子纵摇角速度')
for time=[10 20 40 60 100]
n=time/dt+1;
d_fuzi=solution1(n,1);
v_fuzi=solution1(n,2);
d_zhenzi=solution1(n,3);
v_zhenzi=solution1(n,4);
angle_fuzi=solution1(n,5);
anglev_fuzi=solution1(n,6);
angle_zhenzi=solution1(n,7);
anglev_zhenzi=solution1(n,8);
fprintf('\t时间为%ds时，浮子位移为%f/m,速度为%f/(m/s);  

振子位移为%f/m,速度为%f/(m/s);\n\t\t浮子角位移为%f/rad,  

角速度为%f/(rad/s);振子角位移为%f/rad,角速度为%f/(rad/s).\n',...  

time,d_fuzi,v_fuzi,d_zhenzi,v_zhenzi,angle_fuzi,anglev_fuzi,angle_zhenzi,anglev_zhenzi);
end
a = max(solution1(:,1));
b = max(solution1(:,2));
c = max(solution1(:,3));
d = max(solution1(:,4));
e = max(solution1(:,5));
f123 = max(solution1(:,6));
g = max(solution1(:,7));
h = max(solution1(:,8));
fprintf('常阻尼浮子最大位移为%f/m,最大速度为%f/(m/s);振子最大位移为%f/m,最大速度为%f/(m/s);  

\n\t\t浮子最大角位移为%f/rad,最大角速度为%f/(rad/s);振子最大角位移为%f/rad,  

最大角速度为%f/(rad/s).\n',a,b,c,d,e,f123,g,h)
function differentialty=constantode1(t,y)
global m1 m2 mfujia c1 k omega f seadensity gravity radius J1 J_fu C_huifu C_xingbo k_niuzhuan
k_zuni L
h0=2;
c2=10000;
differentialty=zeros(8,1);
differentialty(1)=y(2);
differentialty(2)=-c1/(m1+mfujia)*y(2)-(pi*radius^2*seadensity*gravity/(m1+mfujia)*y(1)*(y(1)>=h0-3)+pi*radius^2*seadensity*gravity/(m1+mfujia)*y(1));
differentialty(3)=y(4);
differentialty(4)=-c2/m2*(y(4)-y(2))-k/m2*(y(3)-y(1));
differentialty(5)=y(6);
differentialty(6)=(L*cos(omega*t)-(C_huifu*y(5)+C_xingbo*y(6)+k_niuzhuan*(y(5)-y(7))+(k_zuni*(y(6)-y(8))))/(J1+J_fu));
differentialty(7)=y(8);
J2=(0.5-m2*gravity/k+y(3)-y(1))^2*m2;
differentialty(8)=-(k_niuzhuan*(y(7)-y(5))+k_zuni*(y(8)-y(6)))/J2;%
```

```

+2*m2*(y(4)-y(2))*y(8)
*(0.5-m2*gravity/k+y(3)-y(1)))/J2; 不考慮科力
end
function J1=zhuandongguanliangfuzi()
m1=4866;
hshangdinggai=2/3*0.8;
hyuanzhuti=2.3;
hxiahuiti=3.8;
Sshangdinggai=pi*sqrt(1+0.8^2);
Syuanzhuti=6*pi;
Sxiadinggai=pi;
Sfuti=Sshangdinggai+Syuanzhuti+Sxiadinggai;
zc=Sshangdinggai/Sfuti*hshangdinggai+Syuanzhuti/Sfuti*hyuanzhuti+Sxiadinggai/Sfuti*hxiahuiti;
seadensity=m1/Sfuti;
Ja=seadensity*sqrt(1+0.8^2)*(1/2+0.8^2)/4*2*pi+seadensity*sqrt(1+0.8^2)*pi*zc^2-
2*seadensity*sqrt(1+0.8^2)*zc*0.8/3*2*pi;
Jb=seadensity*(pi*3+2*pi*zc^2*3+2*pi/3*54.36-2*pi*zc*13.8);
Jc=seadensity*(3.8-zc)^2*pi+seadensity*2*pi/2/4;
J1=Ja+Jb+Jc;
end

```

## 附录 E 问题四 matlab 代码

```

close all;
clc;
clear;
format long
tic;
c23 = 2;
c13 = 0.75;
w = 0.2;
xlimit = [0,100000;...
           0,100000];
popsize = 50;
maxgen = 35;
vrangle = 0.3;
SolveTimes = 1;

popdimension = size(xlimit,1);
vlimit = vrangle*[-xlimit(:,2)+xlimit(:,1),xlimit(:,2)-xlimit(:,1)];
xrecode = zeros(SolveTimes,popdimension);
fitrecord = zeros(SolveTimes,maxgen);

for times = 1:1:SolveTimes

```

```

pop=
    repmat(xlimit(:,1)',popsize,1)+repmat(diff(xlimit'),popsize,1).*rand(popsize,popdimension);
v =
    repmat(vlimit(:,1)',popsize,1)+repmat(diff(vlimit'),popsize,1).*rand(popsize,popdimension);

fitness = zeros(popsize,1);
fprintf('in process\n')
for i=1:popsize
    fitness(i)=fcn(pop(i,:));
end

[Firstbestfitness,Firstbestindex]=max(fitness,[],'all','linear');
gbestnum = pop(Firstbestindex,:);
gbestfitness = Firstbestfitness;
pbestnum = pop;
pbestfitness = fitness;
clear Firstbestfitness Firstbestindex

%
for m=1:maxgen

v = w*v + ...
    c13*rand(popsize,1).*(pbestnum -pop)+...
    c23*rand(popsize,1).*(repmat(gbestnum,popsize,1) - pop);

vlimitbelow = repmat(vlimit(:,1)',popsize,1);
vlimitup = repmat(vlimit(:,2)',popsize,1);
bigger = v > vlimit(:,2)';
v(bigger) = vlimitup(bigger);
smaller = v < vlimit(:,1)';
v(smaller) = vlimitbelow(smaller);

pop = pop +v;

poplimitbelow = repmat(xlimit(:,1)',popsize,1);
poplimitup = repmat(xlimit(:,2)',popsize,1);
bigger = pop > xlimit(:,2)';
pop(bigger) = poplimitup(bigger);
smaller = pop < xlimit(:,1)';
pop(smaller) = poplimitbelow(smaller);

```

```

for i=1:popsiz
    fitness(i)=fcn(pop(i,:));
end

toreplacepop = (fitness > pbestfitness);
pbestfitness(toreplacepop) = fitness(toreplacepop);
pbestnum(toreplacepop,:) = pop(toreplacepop,:);

[maxfitness,maxfitnessindex] = max(fitness,[],'all','linear');
if maxfitness > gbesfitness
    gbesfitness = maxfitness;
    gbestnum = pop(maxfitnessindex,:);
end

fitrecord(times,m)=gbesfitness;
xrecode(times,:)=gbestnum;
end
end

[PSObestfit,bestfitindex] = max(fitrecord(:,end),[],'all','linear');
PSObestnum = xrecode(bestfitindex,:);
disp('%%%%%%%%%(粒子群算法)%%%%%%');
disp(['最优值为：',num2str(PSObestfit)]);
disp(['最优解为：',num2str(PSObestnum)]);
toc

for i=1:1
    Fontsize = 10.5;
    LineWidth = 1.2;
    f1 = figure;
    f1.Visible = 'on';
    f1.Units = 'centimeters';
    f1.Position = [20 8 16 9];
    f1.Color = 'w';
    f1.NumberTitle = 'off';
    f1.Name = '迭代收敛图';
    f1.Resize = 'off';

    ax1=axes(f1);
    ax1.Box = 'off';
    ax1.Visible = 'on';
    ax1.NextPlot = 'add';
    ax1.Color= 'w';
    ax1.TickDir = 'in';
    ax1.FontUnits = 'points';

```

```

ax1.FontSize = Fontsize;
ax1.LineWidth = LineWidth;
ax1.FontName = 'Times New Roman';
ax1.FontWeight = 'bold';
ax1.TitleFontSizeMultiplier = 1;
ax1.LabelFontSizeMultiplier = 1;
ax1.XColor = 'k';
ax1.YColor = 'k';
ax1.YAxisLocation = 'left';
ax1.XAxisLocation = 'bottom';
ax1.TickLength = [0.01,0.035];
ax1.XDir = 'normal';
ax1.YDir = 'normal';
grid(ax1,'on');
ax1.GridLineStyle = '--';
ax1.GridAlpha = 0.85;
ax1.GridColor = [0.85,0.85,0.85];

x1 = 1:1:maxgen;
y1 = fitrecord(bestfitindex,:);
p1 = plot(ax1,x1,y1);
p1.Visible = 'on';
p1.LineWidth = LineWidth;
p1.Color = 'k';
p1.LineStyle = '-';
p1.LineJoin = 'round';
p1.Marker = 'none';
p1.MarkerIndices = 1:round(length(x1)/10):length(x1);
p1.MarkerSize = 6;
p1.MarkerEdgeColor = 'auto';
p1.MarkerFaceColor = 'none';
ax1.XScale = 'linear';
ax1.YScale = 'linear';
axis(ax1,[min(x1,[],'all'),max(x1,[],'all'),min(y1,[],'all') - 1,max(y1,[],'all') + 1]);
ax1.XTickMode = 'auto';
ax1.YTickMode = 'auto';
ax1.XTickLabelMode = 'auto';
ax1.YTickLabelMode = 'auto';
xlabel('迭代次数','FontSize',Fontsize,'FontName','宋体');
ylabel('适应度','FontSize',Fontsize,'FontName','宋体');
end
%适应度函数
function y=fcn(x)
global m1 m2 mfujia c1 k omega f seadensity gravity radius J1 J_fu C_huifu C_xingbo k_niuzhuan
k_zuni L c2
m1=4866;
mfujia=1091.099;

```

```

J1=zhuandongguanliangfuzi;
J_fu=7142.493;
c1=528.5018;
C_xingbo=1655.909;
C_huifu=8890.7;
m2=2433;
k=80000;
k_niuzhuan=250000;
omega=1.9806;
f=1760;
L=2140;
seadensity=1025;
gravity=9.8;
radius=1;
Time=2*pi/omega;
Timetotal=40*Time;
dt=0.2;
timespan=Timetotal/2:dt:Timetotal;
y0=[0;0;0;0;0;0;0;0];
c2=x(:,1);
k_zuni=x(:,2);
[tt1,yy1]=ode45(@constantode1,timespan,y0);
vr=yy1(:,2)-yy1(:,4);
vr2=yy1(:,6)-yy1(:,8);
summ=2*trapz(tt1,c2.*vr.^2)/Timetotal;
summ2=2*trapz(tt1,k_zuni.*vr2.^2)/Timetotal;
y = summ+summ2;
end
function differentialaly=constantode1(t,y)
global m1 m2 mfujia c1 k omega f seadensity gravity radius J1 J_fu C_huifu C_xingbo k_niuzhuan
k_zuni L c2
h0=2;
differentialaly=zeros(8,1);
differentialaly(1)=y(2);
differentialaly(2)=-c1/(m1+mfujia)*y(2)-(pi*radius^2*seadensity*gravity/(m1+mfujia)*y(1)*(y(1)>=h0-3)+pi*radius^2*seadensity*gravity/(m1+mfujia)*y(1));
differentialaly(3)=y(4);
differentialaly(4)=-c2/m2*(y(4)-y(2))-k/m2*(y(3)-y(1));
differentialaly(5)=y(6);
differentialaly(6)=(L*cos(omega*t)-(C_huifu*y(5)+C_xingbo*y(6)+k_niuzhuan*(y(5)-y(7))+k_zuni*(y(6)-y(8))))/(J1+J2);
differentialaly(7)=y(8);
J2=(0.5-m2*gravity/k+y(3)-y(1))^2*m2;
differentialaly(8)=-(k_niuzhuan*(y(7)-y(5))+k_zuni*(y(8)-y(6))+2*m2*(y(4)-y(2))*y(8)*(0.5-m2*gravity/k+y(3)-y(1)));
end
function J1=zhuandongguanliangfuzi()
m1=4866;
hshangdinggai=2/3*0.8;
hyuanzhuti=2.3;

```

```
hxiachuiti=3.8;
Sshangdinggai=pi*sqrt(1+0.8^2);
Syuanzhuti=6*pi;
Sxiadinggai=pi;
Sfuti=Sshangdinggai+Syuanzhuti+Sxiadinggai;
zc=Sshangdinggai/Sfuti*hshangdinggai+Syuanzhuti/Sfuti*hyuanzhuti+Sxiadinggai/Sfuti*hxiachuiti;
seadensity=m1/Sfuti;
Ja=seadensity*sqrt(1+0.8^2)*(1/2+0.8^2)/4*2*pi+seadensity*sqrt(1+0.8^2)*pi*zc^2-2*seadensity*sqrt(1+0.8^2)*zc*0
Jb=seadensity*(pi*3+2*pi*zc^2*3+2*pi/3*54.36-2*pi*zc*13.8);
Jc=seadensity*(3.8-zc)^2*pi+seadensity*2*pi/2/4;
J1=Ja+Jb+Jc;
end
```

---