

波浪能装置输出功率最大化模型

摘要

本文从分析波浪能装置在海水中垂荡和纵摇运动出发,通过分别对浮子和振子进行受力分析、建立微分方程,获得了浮子和振子从静止到进行垂荡和纵摇运动各时刻的运动状态;而后通过调节直线阻尼器和旋转阻尼器的阻尼系数的相关参数,计算出使得 PTO 系统均输出功率最大的阻尼系数的相关参数,提升了波浪能装置的能量转换效率。

针对问题 1,我们仅考虑装置做垂荡运动的情况,不计海水纵摇激励力等的影响,可认为这是一个多自由度线性系统。根据题意,初始状态装置平衡于静水中,假设此后浮子进行垂荡运动时不会完全浸没水中或露出圆锥部分,则可得到静水恢复力的线性表达式。对浮子和振子分别进行受力分析、建立多元微分方程,利用 MATLAB 中的 ode45 功能函数求解,最终得到浮子和振子在 40 个波浪周期内的垂荡位移和速度,两小问各自条件下 10s、20s、40s、60s、100s 时的数据如下文 5.1 中表 1 和表 2 所示。

针对问题 2,由于是求使得 PTO 系统平均输出功率最大的阻尼系数,则不妨利用稳态下浮子和振子的运动来简化运算。在阻尼系数为某一常量时,可以利用稳态时浮子和振子的运动频率同海浪相等,以此得到稳态下二者的相对速度的解析解,从而写出 PTO 系统平

均输出功率的表达式: $PTO = \frac{1}{2} \frac{\omega^6 m_z^2 f^2 \eta_l}{(C - D\eta_l)^2 + \omega^2 [E\eta_l + B\lambda_l]^2}$, 由此可知: $\eta_{l, best} = \sqrt{\frac{C^2 + \omega^2 B^2 \lambda_l^2}{D^2 + \omega^2 E^2}}$ 时有最大

功率。除此之外,我们还可以利用已经得到的数值结果计算每 0.2 秒 PTO 系统的做功,利用 trapz 梯形数值积分后再除以总时间,得到某一阻尼系数下的近似平均输出功率,再遍历阻尼系数的取值,可以得到最大输出功率为 229.334W,此时的阻尼系数取值为 37193.812。当阻尼系数和相对速度绝对值的幂成正比时,难以得到其解析解,参考第一小问的结果,可见 trapz 梯形数值积分的精度满足要求,因此直接使用前者进行计算,得到最大输出功率为 230.127W,阻尼比例系数为 9.75×10^4 ,幂值为 0.370。

针对问题 3,浮子和振子不仅做垂荡运动,还进行纵摇运动。仔细审题发现,题目并未明确纵摇激励力矩、兴波阻尼力矩等力矩的相对转轴,我们在对转轴和中轴上的作用力与作用点进行分析之后,发现过于繁琐且诸多信息未知,需要对模型进行合理近似。通过对附表中给出数据的初步估算,得到浮子和振子的角位移为小量,因此在考虑垂荡运动时,可以忽略由于纵摇运动带来的影响;在考虑纵摇运动时,由于计算得到的浮子和振子的相对垂荡位移很小,因此可以将浮子和振子视为一个整体计算其在海水中的转动,此后再构建以转轴为原点的平动非惯性参考系,分析振子的转动,最终得到浮子和振子在 40 个波浪周期内的垂荡位移和速度以及纵摇角位移和角速度,10s、20s、40s、60s、100s 时的数据如下文 5.3 中表 3 和表 4 所示。

针对问题 4,仅需考虑阻尼系数为常量的情况,参考问题 2 的解法,可认为 trapz 梯形数值积分法足够精确,使用 MATLAB 做出全局遍历图并得到该条件下最大输出功率为 322.128W,此时直线阻尼器的阻尼系数为 5.93×10^4 ,旋转阻尼器的阻尼系数为 100000。

经过模型稳定性分析,对于大多数波浪激励力、波浪激励力矩下模型均成立。模型具有较好鲁棒性。总体而言,该模型原创度高,简化较为合理,结果吻合初始假设;在第三、四问的模型中,由于做了小量近似,可能导致当波浪较大时出现一定误差,精度有所下降。该模型还可推广至研究弹簧刚度和扭转弹簧刚度、或浮子、振子的质量比对于输出功率最大化的影响,以提升波浪能装置的转换效率。

关键词: 多自由度线性系统、多元微分方程、平动非惯性系



一、问题重述

1.1 问题背景

当今全球化石燃料日益枯竭，环境问题也日益严重。波浪能具有可再生、储量多、易获取的特点，有望成为人类解决当前所面临的能源危机与环境污染的一大助力。而要想规模化使用波浪能这一重要的海洋可再生能源，则需要解决波浪能装置的能量转化效率问题。

一种利用波浪能的装置的原理为，在波浪作用下浮子的运动带动其内部振子的运动，通过由弹簧和阻尼器所构成的 PTO（能量输出系统），利用两者的相对运动驱动阻尼器做功，从而实现向其他能源的转化。

1.2 问题提出

某波浪能装置由四个部分组成，分别为质量均匀的空心壳体浮子以及被密封在浮子内部的中轴、穿在中轴上的振子与包含弹簧和阻尼器的 PTO。其中，浮子由一个有盖无底的圆柱壳体和圆锥壳体共同组成，中轴的底座固定于两壳体连接部分的隔层中心处，并通过 PTO 与振子连接。在波浪的作用下，浮子在海面做摇荡运动，浮子和振子的相对运动使得阻尼器做功，将波浪能转化为可利用的能量，同时通过调节阻尼器的阻尼系数有望提升此装置的能量转化效率。

基于以上背景信息与附件内容，需建立数学模型解决以下问题：

问题一：考虑浮子只做垂荡运动的情况，且中轴固定，振子由 PTO 与底座连接并只沿中轴运动，其受到阻尼力的大小与其同浮子的相对速度正相关。装置在初始状态平衡于静水中，此后受垂荡激励力影响开始运动。建立数学模型，分别计算在阻尼器的阻尼系数固定为 $10000 \text{ N}\cdot\text{s}/\text{m}$ 时，以及阻尼系数正比于浮子和振子相对速度绝对值的幂（取比例系数为 10000 ，幂指数 0.5 ）两种情况下，装置受垂荡激励力作用后 40 个波浪周期内浮子与振子分别的垂荡位移和速度。

问题二：其他条件与（1）中相同，求解最优阻尼使得装置转化效率最佳，分别对 $[0, 100000]$ 内的常量阻尼系数以及比例系数处于 $[0, 100000]$ 内、幂指数处于 $[0, 1]$ 区间内的阻尼系数计算出最大平均输出功率和相应阻尼系数。

问题三：考虑装置同时进行垂荡和纵摇运动的情况。此时中轴与底座采取铰接，并在转轴处增置了旋转阻尼器和扭转弹簧，各项参数见附件。要求建立数学模型并算出平衡于水中的装置在此情况下受垂荡激励力和纵摇激励力矩共同作用后在 40 个周期内浮子和振子分别的垂荡位移和速度以及纵摇角位移和角速度。

问题四：对阻尼系数在 $[0, 100000]$ 内直线阻尼器和旋转阻尼器取值，计算出浮子同时进行垂荡和纵摇的情况下最大输出功率与相应阻尼系数。



二、问题分析

本文主要解决的是一种波浪能装置的能量转化效率最优化问题。问题一、问题二要解决的是在该装置只做垂荡运动时浮子与振子的运动状况以及如何确定最优的阻尼系数使得平均输出功率最高，而问题三、问题四要解决的是在该设备只做垂荡和纵摇运动时浮子与振子的运动状况以及通过调节阻尼系数实现平均输出功率的最优化。

2.1 问题一的分析

问题一考虑浮子在波浪中只做垂荡运动，中轴与底座固定连接。 $t=0$ 时，装置平衡于静水中； $t>0$ 时，浮子受垂荡激励力作用带动振子开始运动，并在一段时间后达到稳态。由于已经假定浮子仅做垂荡运动，故不考虑纵摇激励力的影响。依据题目所给条件首先对装置内各部分进行受力分析，建立关于浮子和振子的动力学方程。

第一小问：题目给定阻尼固定的情况下，取阻尼系数为 $10000 \text{ N}\cdot\text{s/m}$ 带入所列的两个联立微分方程，使用 MATLAB 算出数值解即可。

第二小问：题目给定阻尼系数正比于浮子、振子两者之间的相对速度绝对值的幂，则修正动力学方程，带入比例系数 10000 和幂指数 0.5 ，同样使用 MATLAB 求解。

2.2 问题二的分析

问题二要求确定在问题一条件下最优的阻尼系数以实现 PTO 系统最大平均功率。对于最优的阻尼系数，一方面可以利用复数解法求解响应函数，得到 PTO 功率和阻尼系数的关系式，确定其解析解；另一方面，也可通过计算机枚举以取得最佳阻尼系数的数值解。

第一小问：求得其解析解和数值解，对比两个解，差距较小则可认定两种方法均能准确求出最优阻尼系数。

第二小问：由于复数解法较难解决非线性方程，在比例系数和幂指数均未知的情况下可能会出现解析解无法写出的情况，此时则需考虑用数值解代替解析解。若第一小问中两个解差距很小，则可认定数值解也足够准确，取计算机枚举得到的数值解为本小问情形下的最优解。

2.3 问题三的分析

问题三考虑浮子在水中只做垂荡和纵摇运动，中轴与底座铰接，并在转轴处增加了旋转阻尼器与旋转弹簧，其余条件与问题一相同。

假定浮子和振子的角位移很小，因此在考虑垂荡运动时，可以忽略由于纵摇运动带来的影响；在考虑纵摇运动时，由于计算得到的浮子和振子的相对垂荡位移很小，因此可以将浮子和振子视为一个整体计算转动，此后再构建以转轴为原点的平动非惯性参考系，分析振子的转动。

分别列出 2 条动力学方程和 2 条转动方程，利用 MATLAB 求解微分方程。

2.4 问题四的分析

问题四需要对阻尼系数在 $[0, 100000]$ 内直线阻尼器和旋转阻尼器取值，得到浮子同时进行垂荡和纵摇的情况下最大输出功率与相应阻尼系数。类似于问题二，利用数值积分法求解出系统 PTO，再对直线阻尼器和旋转阻尼器最优阻尼系数从 $0-100000$ 进行遍历，计算出不同条件下输出 PTO。通过多次细分目标区间，找出最优参数。

三、模型假设

1. 假设海水是无粘且无旋的；
2. 底座、中轴、隔层、PTO 的质量与所有的摩擦不计；
3. 忽略附件未给出的有关底座、中轴、中轴架、转轴以及其他结构的大小、厚度、高度等；
4. 海平面足够大且浮子的运动对海平面高度不产生影响；
5. 浮子顶端不会浸没且圆锥部分不露出水面；
6. 在解决问题二时，假设模型将在有限时间内趋于稳定，最终所有广义坐标的圆频率与激励力圆频率相等；
7. 在解决问题三时，相比于浮子的垂荡运动，振子与浮子的垂荡运动的相对位移可视为小量；
8. 在解决问题三时，浮子与振子的纵摇运动的角位移为小量；

四、符号说明

| 符号 | 意义 | 单位 |
|----------|----------|-------------------|
| m_f | 浮子质量 | kg |
| R_f | 浮子底半径 | m |
| h_y | 浮子圆柱部分高度 | m |
| h_z | 浮子圆锥部分高度 | m |
| m_z | 振子质量 | kg |
| R_z | 振子半径 | m |
| H_z | 振子高度 | m |
| ρ | 海水的密度 | kg/m ³ |
| g | 重力加速度 | m/s ² |
| k | 弹簧刚度 | N/m |
| L_0 | 弹簧原长 | m |
| C_k | 扭转弹簧刚度 | N·m |
| C_w | 静水恢复力矩系数 | N·m |
| I_z | 振子转动惯量 | kg·m ² |
| I_f | 浮子转动惯量 | kg·m ² |
| ω | 入射波浪频率 | s ⁻¹ |
| μ | 垂荡附加质量 | kg |

| | | |
|-------------|---------------------------|--------------------------------------|
| J | 纵摇附加转动惯量 | $\text{kg}\cdot\text{m}^2$ |
| λ_i | 垂荡兴波阻尼系数 | $\text{N}\cdot\text{s}/\text{m}$ |
| λ_j | 纵摇兴波阻尼系数 | $\text{N}\cdot\text{s}/\text{m}$ |
| f | 垂荡激励力振幅 | N |
| L | 纵摇激励力矩振幅 | $\text{N}\cdot\text{m}$ |
| η_i | 直线阻尼器阻尼系数 | $\text{N}\cdot\text{s}/\text{m}$ |
| η | 直线阻尼器阻尼比例系数 | \backslash |
| η_j | 旋转阻尼器阻尼系数 | $\text{N}\cdot\text{s}\cdot\text{m}$ |
| h_0 | 初始状态($t=0$)时浮子顶端超过水面的距离 | m |
| h_1 | 初始状态振子质心距转轴的距离 | m |
| h_2 | 初始状态浮子质心距转轴的距离 | m |
| h_3 | 初始状态浮子和振子二者质心距转轴的距离 | m |

五、模型建立与求解

5.1 问题一模型的建立与求解

5.1.1 模型的建立

考虑到波浪能装置在海面做垂荡运动，首先需要排除振幅过大导致海平面高于浮子顶板或者浮子圆锥部分露出水面的情况，因为此情况下静水恢复力的线性条件不再适用。故首先计算平衡状态下浮子顶端距离海平面的距离用以比对。设浮子顶端距离水面的距离为 h_0 ，假设浮子正常做垂荡运动，则振幅需同时小于 h_0 以及圆柱壳体部分长度与 h_0 的差。

初始状态，即 $t=0$ 时，由于海面足够大，可认为浮子的运动不会引起海平面发生变化。由于此状态下受力平衡，因此有以下等式：

$$\rho g \left[\frac{1}{3} \pi R_f^2 h_z + \pi R_f^2 (h_y - h_0) \right] = (m_f + m_j) g$$

得到： $h_0=0.99999\text{m}$ 。之后将用以比对振幅以验证浮子是否符合假设。

接下来，需要分析浮子和振子的受力与运动情况。设浮子竖直向上位移为 x_1 ，振子竖直向上的位移为 x_2 。

在 40 个波浪周期内，浮子受到周期性的垂荡激励力的作用，其运动的同时还会受到阻碍其运动的静水恢复力（与其位移 x_1 正相关）、垂荡兴波阻尼力（与其速度正相关）以及垂荡附加惯性力。在装置内部，由于浮子与振子并非刚性连接，故二者的相对运动会产生相应的弹簧弹力（与二者相对位移正相关）以及阻尼力（与二者的相对运动速度正相关）。

考虑到在垂直方向上，浮子收到波浪激励力、静水回复力、兴波阻尼力、附加惯性力

以及 TPO 系统的弹力与阻尼力，取浮子的位移为 x_1 ，振子的位移为 x_2 ，则受力分析后得到该系统的运动方程。

阻尼系数固定时，浮子与振子的运动方程如下：

$$\begin{cases} (m_f + \mu)\ddot{x}_1 + \eta_l(\dot{x}_1 - \dot{x}_2) + \lambda_1\dot{x}_1 + k(x_1 - x_2) + \rho g \pi R_f^2 x_1 = f \\ m_z\ddot{x}_2 + \eta_l(\dot{x}_2 - \dot{x}_1) + k(x_2 - x_1) = 0 \end{cases} \quad (1)$$

阻尼系数正比于浮子和振子相对速度绝对值的幂时，浮子与振子的运动方程如下：

$$\begin{cases} (m_f + \mu)\ddot{x}_1 + \eta(\dot{x}_1 - \dot{x}_2)|\dot{x}_1 - \dot{x}_2|^{\frac{1}{2}} + \lambda_1\dot{x}_1 + k(x_1 - x_2) + \rho g \pi R_f^2 x_1 = f \\ m_z\ddot{x}_2 + \eta(\dot{x}_2 - \dot{x}_1)|\dot{x}_1 - \dot{x}_2|^{\frac{1}{2}} + k(x_2 - x_1) = 0 \end{cases} \quad (2)$$

对以上两组方程分别求解即可得到浮子与振子位移和速度随时间的变化情况，从而解决本问题。

5.1.2 第一小问模型的求解（阻尼系数固定时）

ode45 是 MATLAB 中一种专门用于求解微分方程的功能函数，是一种中阶、自适应步长、用以求解非刚性常微分方程的方式。针对已经得到的公式（1），对其进行代码编写并得到数值解，使用 ode45 求解非刚性微分方程，得到数据见附件 result1-1.xlsx。

对得到的结果进行可视化以直观体现浮子与振子随时间的运动情况，得到下图：

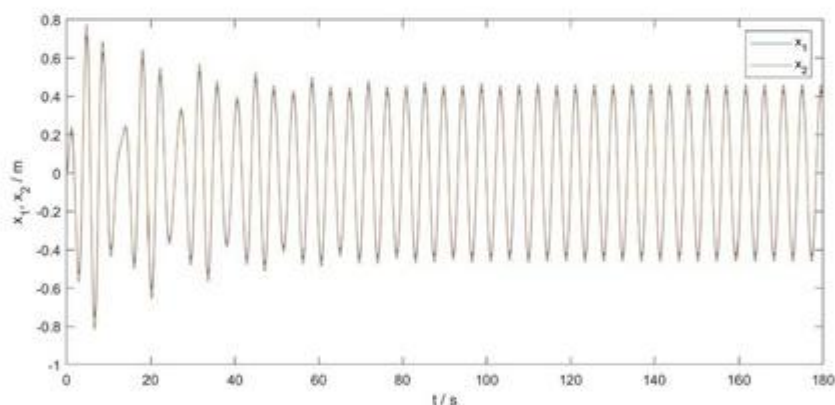


图 1 浮子与振子位移 x_1 、 x_2 随时间 t 的变化曲线

其中，在 10s、20s、40s、60s、100s 时浮子与振子的垂荡位移和速度如下表所示：

表 1 浮子与振子的垂荡位移和速度表(其中时间 t 单位为 s，位移单位为 m，速度单位为 m/s)

| t | x_1 | v_1 | x_2 | v_2 | $x_1 - x_2$ |
|-----|--------------|--------------|--------------|-------------|-------------|
| 10 | -0.190591947 | -0.640564278 | -0.211549275 | -0.69360835 | 0.020957328 |
| 20 | -0.590531163 | -0.240467502 | -0.634068234 | -0.27228741 | 0.043537071 |
| 40 | 0.285456229 | 0.313435653 | 0.296609879 | 0.333247927 | -0.01115365 |
| 60 | -0.31443603 | -0.479107064 | -0.33135498 | -0.51555596 | 0.01691895 |
| 100 | -0.083590128 | -0.604065973 | -0.084036856 | -0.64304386 | 0.000446728 |

5.1.3 第二小问模型的求解（阻尼系数正比于浮子和振子相对速度绝对值的幂）

同理，使用 ode45 求解公式（2），得到的数据见附件 result1-2.xlsx。

对得到的结果进行可视化以直观体现浮子与振子随时间的运动情况，得到下图：

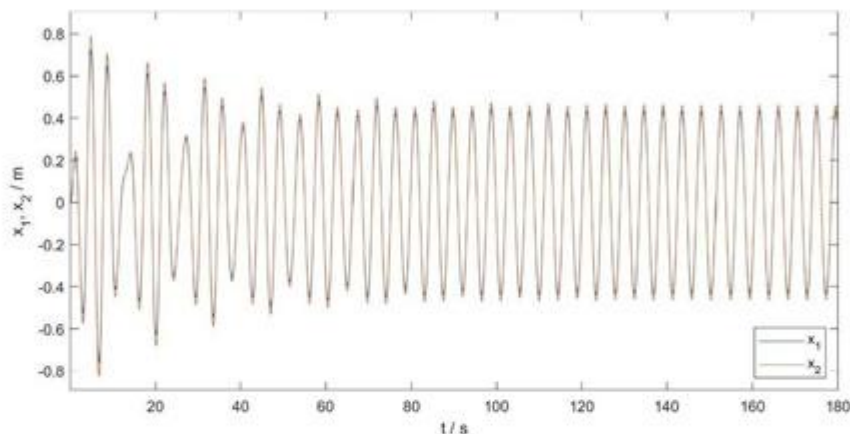


图 2 浮子与振子位移 x_1 、 x_2 随时间 t 的变化曲线

表 2 浮子与振子的垂荡位移和速度表(其中时间 t 单位为 s，位移单位为 m，速度单位为 m/s)

| t | x_1 | v_1 | x_2 | v_2 |
|-----|----------|----------|----------|----------|
| 10 | -0.20583 | -0.65279 | -0.23463 | -0.69823 |
| 20 | -0.61125 | -0.25419 | -0.66077 | -0.27589 |
| 40 | 0.268708 | 0.295805 | 0.280324 | 0.313936 |
| 60 | -0.32725 | -0.49125 | -0.3493 | -0.52414 |
| 100 | -0.08832 | -0.60974 | -0.09376 | -0.64955 |

不难发现 x_1 的最大振幅均小于 h_0 ，满足水面不会没过浮子顶盖以及浮子圆锥壳体部分不会露出水面的假设。

5.2 问题二模型的建立与求解

由于是求使得 PTO 系统平均输出功率最大的阻尼系数，则不妨利用稳态下浮子和振子的运动来简化运算。在阻尼系数为某一常量时，可以利用稳态时浮子和振子的运动频率同海浪相等¹，以此得到稳态下二者的相对速度的解析解，从而写出 PTO 系统平均输出功率的表达式，得到令输出功率最大的阻尼系数的解析解。除此之外，我们还可以利用已经得到的数值结果计算每 0.2 秒 PTO 系统的做功，利用 trapz 梯形数值积分后再除以总时间，得到某一阻尼系数下的近似平均输出功率，再遍历阻尼系数的取值，可以得到最大输出功率和此时的阻尼系数的取值。当阻尼系数和相对速度绝对值的幂成正比时，难以得到其解析解，参考第一小问的结果，如果 trapz 梯形数值积分的精度满足要求，便可直接使用前者进行计算，得到最大输出功率、阻尼比例系数和幂值。

¹ 详情可见附录 2

5.2.1 第一小问模型的求解（阻尼系数为某一定值时）

5.2.1.1 第一小问的解析解

由于阻尼做功只与阻尼系数以及浮子和振子间的相对运动有关，取 PTO 为装置的能量转化功率，则在阻尼系数为某一固定值时，其关系可表示为：

$$PTO = \frac{1}{2} \eta_i \omega^2 |x_1 - x_2|^2$$

其中，此时的 x_1 、 x_2 表示复振幅。

因此首先需要确定浮子与振子相对运动位移的方程。

首先尝试公式解。利用复数解法求解响应函数。由 5.1 可知，阻尼系数固定时的运动方程如 (1) 所示，对于 (1)，利用稳态时浮子和振子的运动频率同海浪相等，可将其改写为：

$$\begin{bmatrix} -\omega^2(m_f + \mu_1) + i\omega(\lambda_1 + \eta_i) + k + \rho g \pi R_f^2 & -i\omega\eta_i - k \\ -i\omega\eta_i - k & -\omega^2 m_z + i\omega\eta_i + k \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}$$

取：

$$Y = \begin{bmatrix} -\omega^2(m_f + \mu_1) + i\omega(\lambda_1 + \eta_i) + k + \rho g \pi R_f^2 & -i\omega\eta_i - k \\ -i\omega\eta_i - k & -\omega^2 m_z + i\omega\eta_i + k \end{bmatrix}$$

则有：

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = Y^{-1} \begin{bmatrix} f \\ 0 \end{bmatrix} = \frac{f}{|Y|} \begin{bmatrix} -\omega^2 m_z + i\omega\eta_i + k \\ i\omega\eta_i + k \end{bmatrix}$$

$$\text{由此得到 } x_1 - x_2 = \frac{f}{|Y|} [-\omega^2 m_z]$$

$$\text{取 } A = -\omega^2(m_f + \mu_1) + k + \rho g \pi R_f^2, \quad B = -\omega^2 m_z + k$$

$$\text{则有 } x_1 - x_2 = \frac{-\omega^2 m_z f}{AB - k^2 - \omega^2 \lambda_1 \eta_i + i\omega[A\eta_i + B(\lambda_1 + \eta_i) - 2k\eta_i]}$$

对 $x_1 - x_2$ 取模长并平方，并令 $C = AB - k^2$ ， $D = -\omega^2 \lambda_1$ ， $E = A + B - 2k$ ，则可简化得到：

$$|x_1 - x_2|^2 = \frac{\omega^4 m_z^2 f^2}{(C - D\eta_i)^2 + \omega^2 [E\eta_i + B\lambda_1]^2}$$

$$\text{带入后得到： } PTO = \frac{1}{2} \frac{\omega^6 m_z^2 f^2 \eta_i}{(C - D\eta_i)^2 + \omega^2 [E\eta_i + B\lambda_1]^2}$$

$$\text{对此式求导可知： } \eta_{i_best} = \sqrt{\frac{C^2 + \omega^2 B^2 \lambda_1^2}{D^2 + \omega^2 E^2}} \text{ 时有最大功率。}$$

带入数据，得到最佳的 $\eta_{i_best}=37193.8119 \text{ N}\cdot\text{s}/\text{m}$ ，对应的最大功率为 229.3339W。

5.2.1.2 第一小问的数值解

数值解采取 trapz 梯形法执行数值积分运算，通过将一个区域分为包含多个更容易计算的区域的梯形，对区间内的积分计算近似值。

考虑到波浪能装置在初始状态后一段时间内未达到垂荡稳定状态，因此在采样区间应当从某一适当的时刻开始。

取积分区间为 0-400s，采样区间为 200-400s，步长为 0.2s。得到 η_i 在[0,100000]的范围内时 PTO 关于 η_i 的函数关系如图所示：

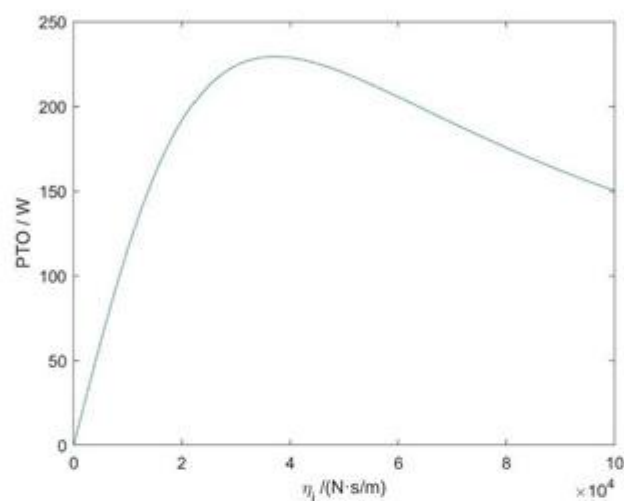


图 3 PTO 关于 η_i 的函数图 1

观察图像可知，PTO 的峰值出现在 η_i 位于[30000,40000]的区间，在此区间作图得：

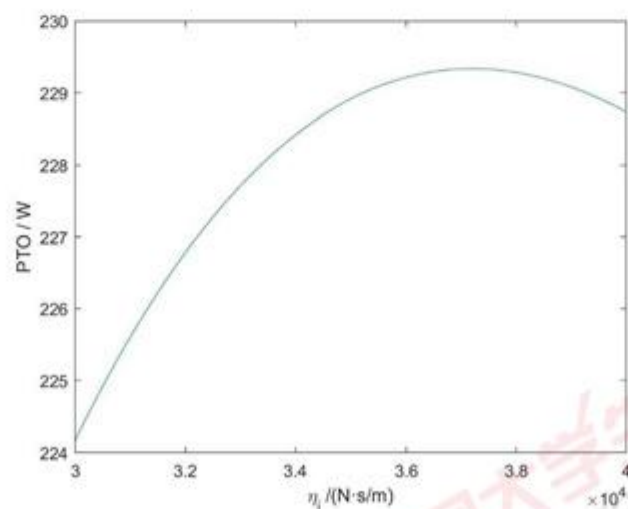


图 4 PTO 关于 η_i 的函数图 2

对峰值采样，得到最大功率为 229.497276437679W，此时的阻尼系数为 3.72×10^4 N·s/m。

5.2.2 第二小问模型的求解（阻尼系数正比于浮子和振子相对速度绝对值的幂）

使用与 5.2.1 中相同的方式建立模型，则在阻尼系数正比于浮子和振子相对速度绝对值的幂时，令幂指数为 α ，比例系数为 η ，则其关系可表示为：

$$PTO = \frac{1}{2} \eta \omega^2 |x_1 - x_2|^{2+\alpha}$$

由于幂指数为 α 未知，很难求出该方程的解析解。

5.2.1 中，经对比可知，由 trapz 梯形法积分后枚举取得的结果 229.49728W 相比于此前采取数学方法获得的解析解得到的 229.33394W 十分接近，因此可以认为，使用数值法得到的结果具有相当的精确性，在难以得到解析解的时候可以采用数值法得出结果。

考虑用数值解来解决本问题。

使用与 5.2.1 相同的思路，取积分区间为 0-400s，采样区间为 200-400s，步长为 0.2s，用 MATLAB 进行二层遍历，做出 PTO 关于比例系数 η 和幂指数 α 的函数。

经可视化后，得到 η 在 [0,100000]、 α 在 [0,1] 的范围内时 PTO 关于两者的函数关系如图所示：

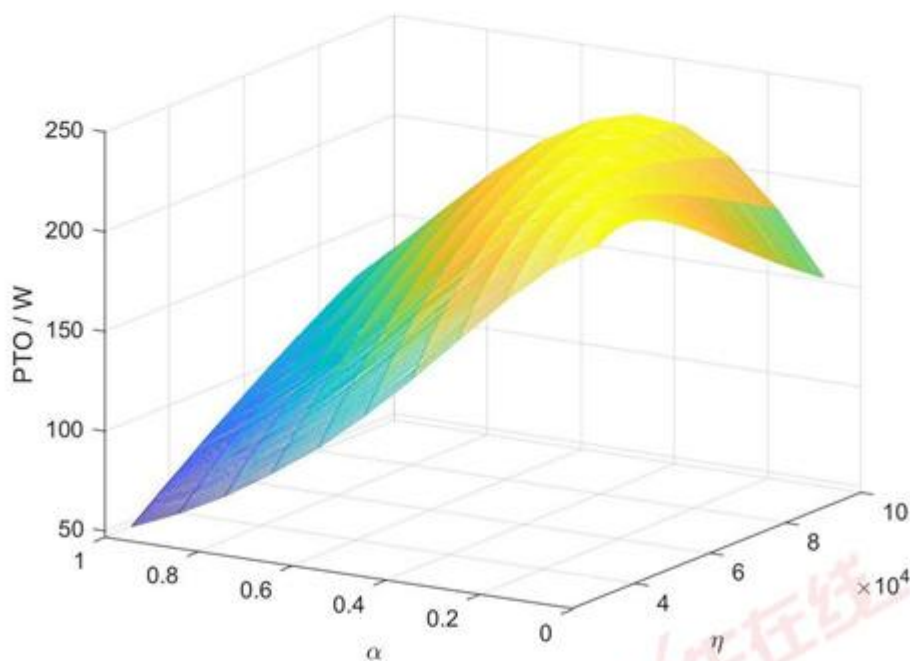


图 5 PTO 关于比例系数 η 和幂指数 α 的函数图 1

观察图像可知，得到 η 在 [20000,100000]、 α 在 [0.3,0.5] 的范围内时 PTO 取得峰值。

故在此范围内重新作图，得到图像如下所示：

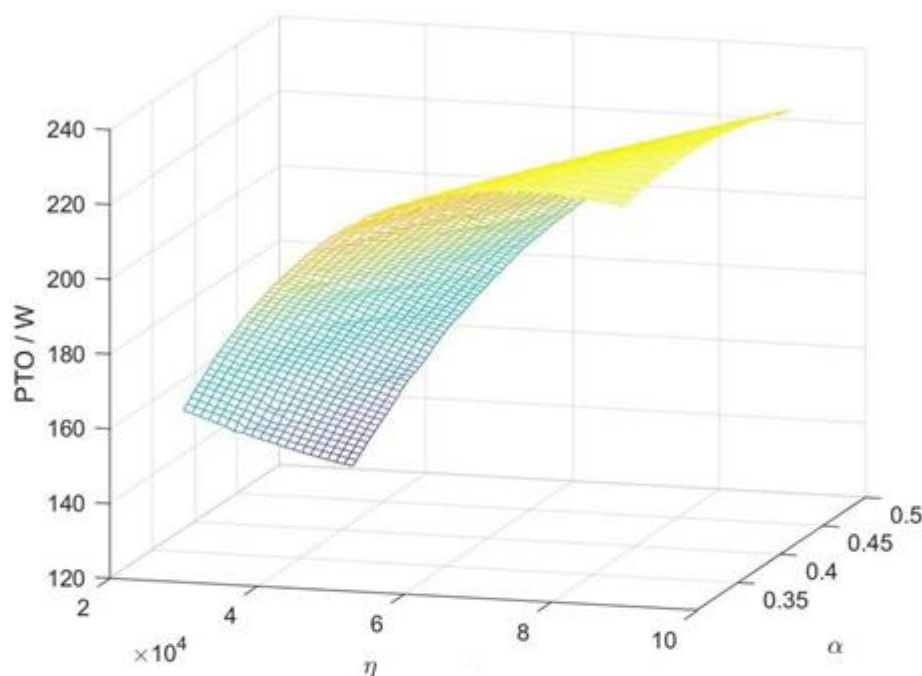


图 6 PTO 关于比例系数 η 和幂指数 α 的函数图 2

最后对所得的数据进行排序，最终得到最大 $PTO=230.12739W$ 。

此时有幂值 $\alpha=0.370$ ，比例系数 $\eta=9.75 \times 10^4$ 。

5.3 问题三模型的建立与求解

5.3.1 各部分转动惯量的确定

计算转动惯量和平衡拽台下质心距离需要用到浮子壳体各部分的面积，故首先需计算得到以下三部分的面积：

浮子圆筒部分侧面积： $S_1 = \pi R_f h_y = 6\pi$

浮子圆锥部分的面积： $S_2 = \pi R_f \sqrt{R_f^2 + h_z^2} = \frac{\sqrt{41}}{5} \pi$

浮子桶盖部分的面积： $S_3 = \pi R_f^2 = \pi$

为简化表述，在之后的算式中均直接以其数值代替符号表示，不再出现面积。

平衡状态下振子质心距离转轴 O 的距离：

$$h_1 = L_0 + \frac{1}{2} H_z - \frac{m_z g}{k}$$

平衡状态下浮子质心距离转轴 O 的距离：

$$h_2 = \frac{1}{35 + \sqrt{41}} \left[\sqrt{41} \left(-\frac{h_z}{3} \right) + 30 \left(\frac{h_y}{2} \right) + 5h_y \right]$$

平衡状态下振子和浮子共同的质心距离转轴 O 的距离:

$$h_3 = \frac{1}{m_z + m_f} (m_z h_1 + m_f h_2)$$

振子相对于振子质心的水平轴的转动惯量:

$$I_z = \frac{1}{4} m_z R_z^2 + \frac{1}{12} m_z H_z^2$$

浮子相对于浮子质心的水平轴的转动惯量:

$$I_f = \frac{m_f}{35 + \sqrt{41}} \left[\sqrt{41} \left(\frac{h_z^2}{6} + \frac{R_f^2}{4} + \frac{2}{3} h_z h_2 \right) + 30 \left(\frac{R_f^2}{2} + \frac{h_f^2}{12} \right) + 5 \left(\frac{R_f^2}{4} + (h_y - h_z)^2 \right) \right]$$

5.3.2 垂荡运动和纵摇运动的求解

经过对附表中给出数值的初步估算, 假设浮子和振子的角位移为小量, 因此在计算浮子的垂荡运动时, 无需考虑纵摇运动的影响。由此可得:

$$\begin{cases} (m_f + \mu) \ddot{x}_1 + \lambda_1 \dot{x}_1 + \rho g \pi R_f^2 x_1 = f \cos \omega t + kx_2 + \eta_1 \dot{x}_2 \\ m_z \ddot{x}_2 + y_1 \dot{x}_2 + kx_2 = m_z [g(1 - \cos \theta_2) - \ddot{x}_1 \cos \theta_2 + \ddot{\theta}_1 h_3 \sin(\theta_2 - \theta_1) - h_3 \dot{\theta}_1^2 \cos(\theta_2 - \theta_1)] \end{cases}$$

其中, x_1 的定义于 5.1 中一样, 表示浮子在垂直方向上的位移, 正方向定为垂直向上; x_2 的定义略有变化 (如图 9 所示), 定义为以转轴 O 为原点的平动非惯性系中振子沿中轴的位移, 正方向沿中轴向上, 可以近似理解为浮子和振子的相对位移。具体求解结果结果见附件 result3.xlsx。解得 x_1 、 x_2 随时间的变化:

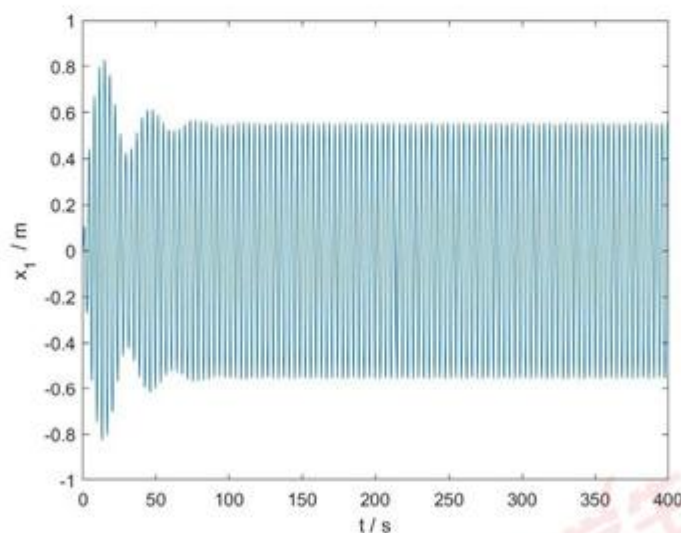


图 7 x_1 关于时间的变化

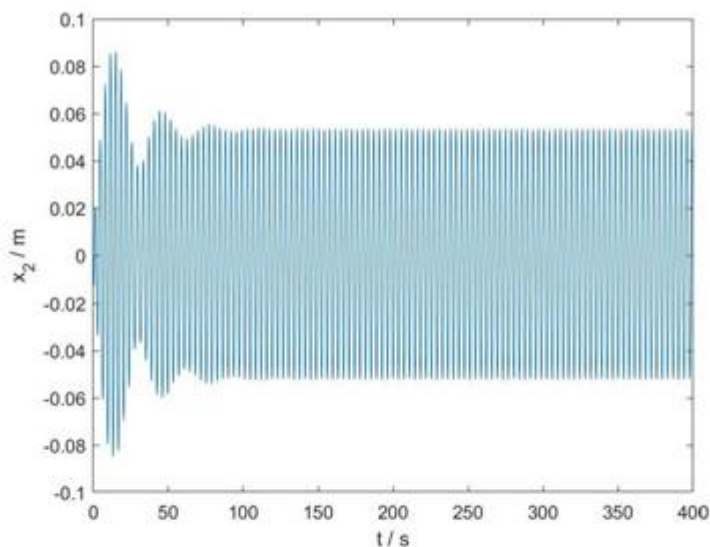


图 8 x_2 关于时间的变化

不难发现浮子和振子之间的相对位移 x_2 是一个小量，因此在计算浮子的纵摇运动时，不妨将浮子和振子视为一个整体进行计算，得到浮子的位移、速度、角位移与角速度后，建立以转轴 O 为原点的平动非惯性系，以计算在此系中的振子的运动情况。取浮子偏转角度为 θ_1 ，振子偏转角度（等于中轴摆动角度） θ_2 ，浮子位移 x_1 、振子位移 x_2 ，正方向如图。

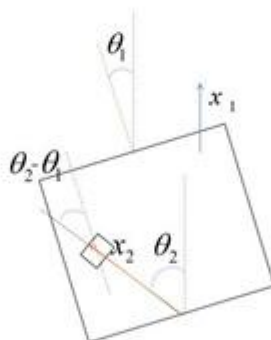


图 9 以中轴与底座的连接转轴 O 建立平动非惯性系

考虑此系中分析浮子和振子的受力情况和力矩平衡，得到以下联立方程组：

$$\begin{cases} [I_1 + I_f + M_1(h_3 - h_1)^2 + M_f(h_3 - h_2)^2] \ddot{\theta}_1 + \lambda_j \dot{\theta}_1 + C_w \theta_1 = L \cos \omega t \\ [I_2 + m_2(h_1 + x_2)^2] \ddot{\theta}_2 + C_k(\theta_2 - \theta_1) + \eta_j(\dot{\theta}_2 - \dot{\theta}_1) \\ = m_2[(g + \ddot{x}_1) \sin \theta_2 + h_3 \dot{\theta}_1^2 \sin(\theta_2 - \theta_1) + \ddot{\theta}_1 h_3 \cos(\theta_2 - \theta_1)](h_1 + x_2) \end{cases}$$

采用 MATLAB 中的 ode45 功能函数来求解非刚性微分方程，其结果见附件 result3.xlsx。

最终得到 θ_1 、 θ_2 随时间的变化如下图所示。

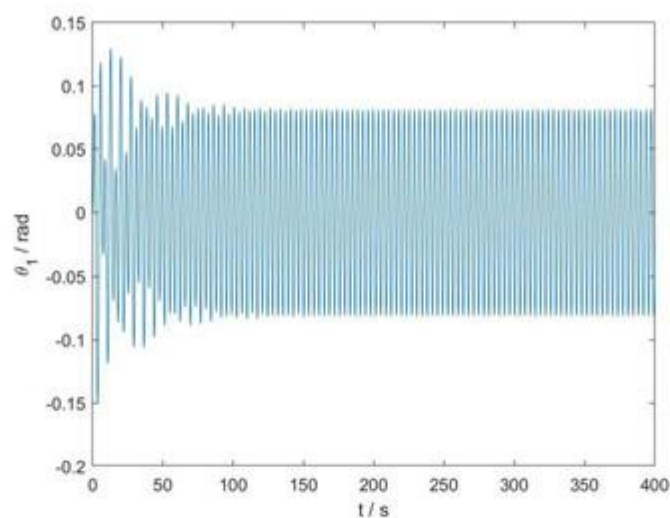


图 10 θ_1 关于时间的变化

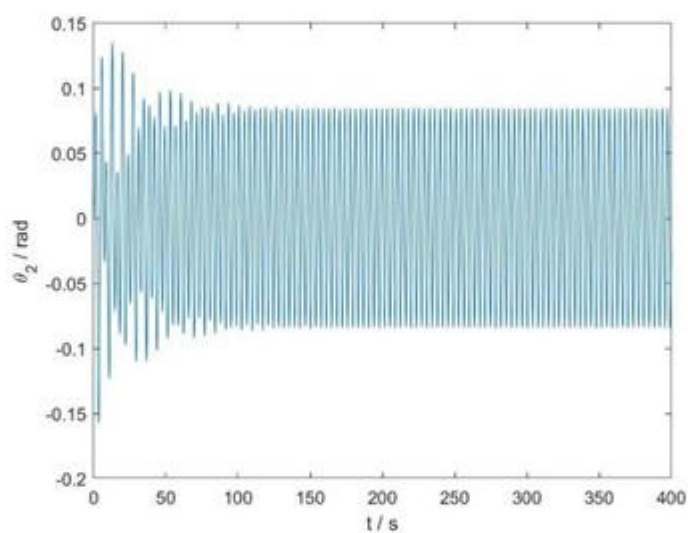


图 11 θ_2 关于时间的变化

不难发现 θ_1 、 θ_2 均为小量，满足初始假设。

其中，浮子和振子在 10s、20s、40s、60s、100s 时的垂荡位移和速度以及纵摇角位移和角速度分别如表 3 和表 4 所示，其中各数值的单位为标准单位。

表 3 浮子垂荡位移和速度以及纵摇角位移和角速度

| t | x_1 | v_1 | θ_1 | ω_1 |
|-----|----------|----------|------------|------------|
| 10 | -0.52313 | 0.98326 | -0.05299 | -0.13555 |
| 20 | -0.69838 | -0.24489 | 0.12253 | 0.02588 |
| 40 | 0.37500 | 0.76586 | -0.05116 | -0.06105 |
| 60 | -0.31531 | -0.72160 | 0.07359 | 0.09148 |
| 100 | -0.04792 | -0.94736 | 0.03163 | 0.13157 |

表 4 振子垂荡位移和速度以及纵摇角位移和角速度

| t | x_2 | v_2 | θ_2 | ω_2 |
|-----|----------|----------|------------|------------|
| 10 | -0.59350 | 1.05454 | -0.05529 | -0.14150 |
| 20 | -0.76358 | -0.28834 | 0.12784 | 0.02742 |
| 40 | 0.39883 | 0.85610 | -0.05293 | -0.06373 |
| 60 | -0.33537 | -0.79616 | 0.07664 | 0.09485 |
| 100 | -0.04062 | -1.03652 | 0.03294 | 0.13739 |

同样，不难发现 x_1 的最大振幅均小于 h_0 ，满足水面不会没过浮子顶盖以及浮子圆锥壳体部分不会露出水面的假设。

5.4 问题四模型的建立与求解

问题四的思路与问题二相同。由于同样难以得到解析解，因此采取 trapz 梯形法执行数值积分运算，对区间内的积分计算近似值，得到数值解。

做出 PTO 关于 η_i 和 η_j 的全局遍历图如下：

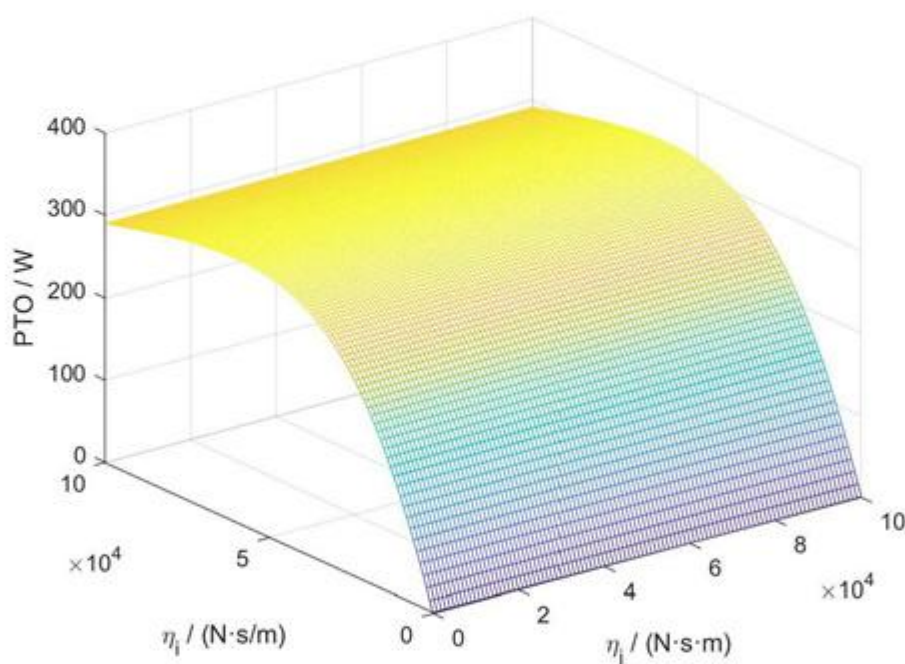


图 12 PTO 关于 η_i 和 η_j 的全局遍历图

根据 PTO 关于 η_i 、 η_j 的数据输出表²可以得出，经过研究数据发现 PTO 对于 η_j 单调变化，因此取得最大 PTO 时旋转阻尼系数 η_j 的值为 100000。再对 η_j 取 100000 时的局部遍历、求解。得到局部图如下：

² 详见支撑材料“问题 4 数据表”

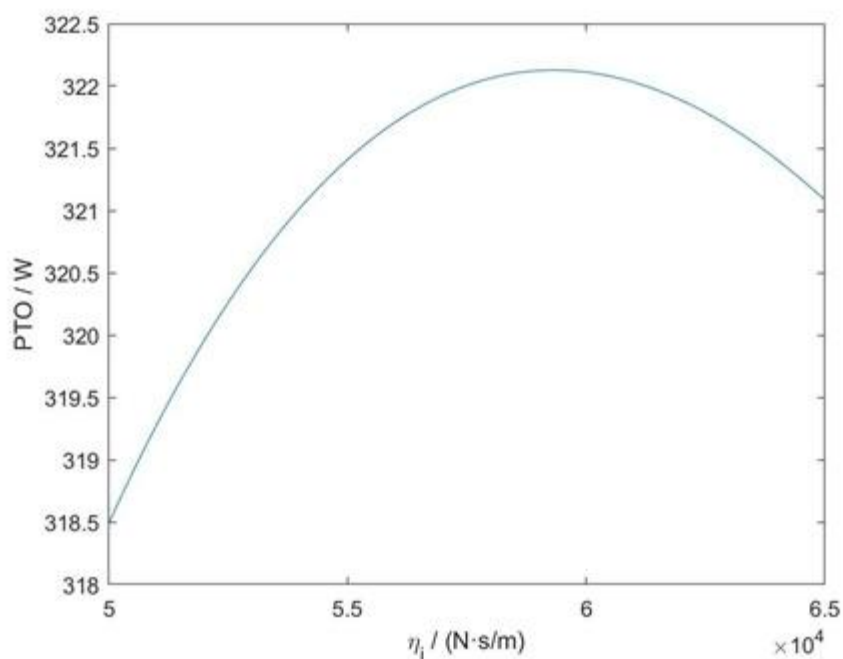


图 13 PTO 关于 η_l 的局部遍历图

对峰值采样，得到的最大功率为：322.128795508752W
 此时直线阻尼系数 $\eta_l = 5.93 \times 10^4$ ，旋转阻尼系数为 $\eta_j = 100000$ 。

六、模型的分析与检验

6.1 模型检验

6.1.1 对“浮子顶端不会浸没且圆锥部分不露出水面”的检验

由 4 个问题中得到的浮子位移数据，均小于起平衡时顶端距离水面的高度（1.00 m）。假设成立。

6.1.2 对“系统在有限时间内趋于稳态”的检验

利用 MATLAB 的信号分析器，对问题一中垂荡位移数据进行频谱分析，得到功率谱。取运动后半段（趋于稳定），得到信号圆频率为 $0.0891 \times 5 \times \pi = 1.399 \text{ rad/s}$ ，与激励力圆频率近似相等。假设成立。

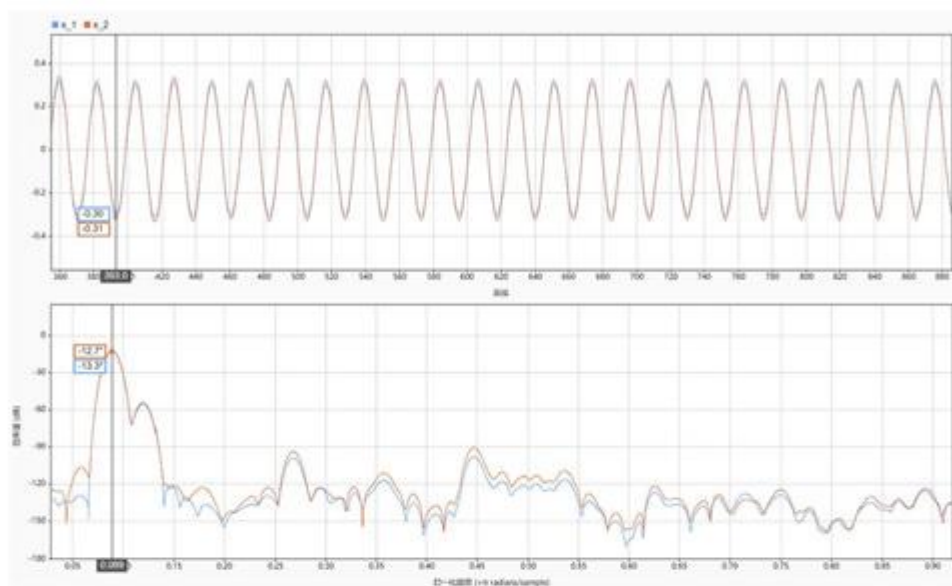


图 14 频率分析

6.1.3 对于“振子与浮子的垂荡运动的相对位移可视为小量”的检验

由 4 个问题的数据可知，相对位移与浮子垂荡位移的比值在 $[0.09, 0.10]$ 区间内，可视为小量。

6.1.4 对于“浮子与振子的纵摇运动的角位移为小量”的检验

由问题三的数据可知，浮子角位移最大为 6.88° ，振子角位移最大为 7.73° ，均可视为小角。

6.2 模型稳定性分析

6.2.1 模型对波浪激励力的稳定性分析

对于不同大小的波浪激励力，计算振子相对于浮子的位移。

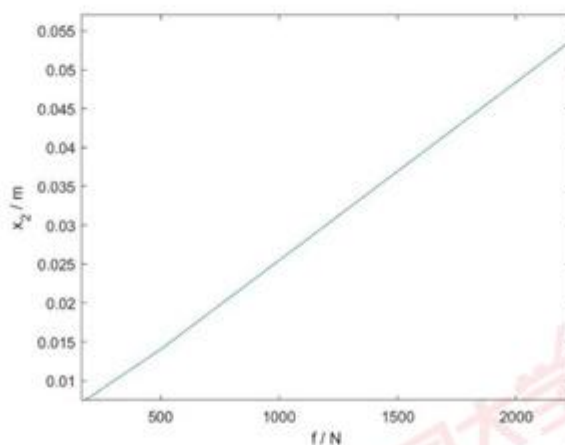


图 15 波浪激励力的稳定性分析图

可得波浪激励力大小在一定范围内变动时，振子相对于浮子位移均满足小量的假设。

6.2.2 模型对波浪激励力矩的稳定性分析

对于不同大小的波浪激励力矩，计算振子相对于浮子的角位移

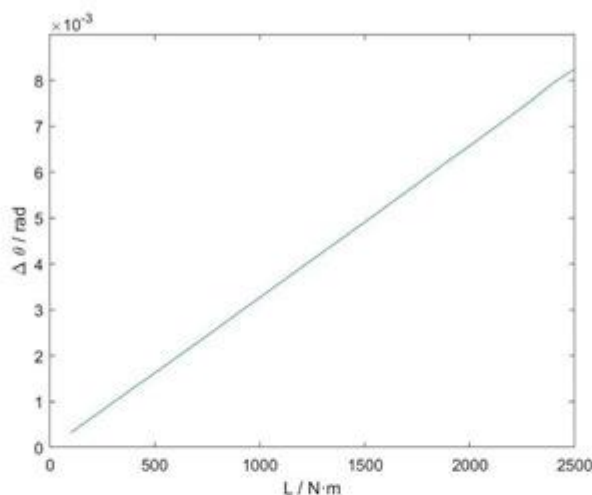


图 16 波浪激励力矩的稳定性分析图

可得，波浪激励力矩大小在一定范围内变动，振子相对于浮子位移均满足小量的假设。

6.2.3 模型对扭转弹簧刚度的稳定性分析

对于不同大小的扭转弹簧刚度，计算输出功率。

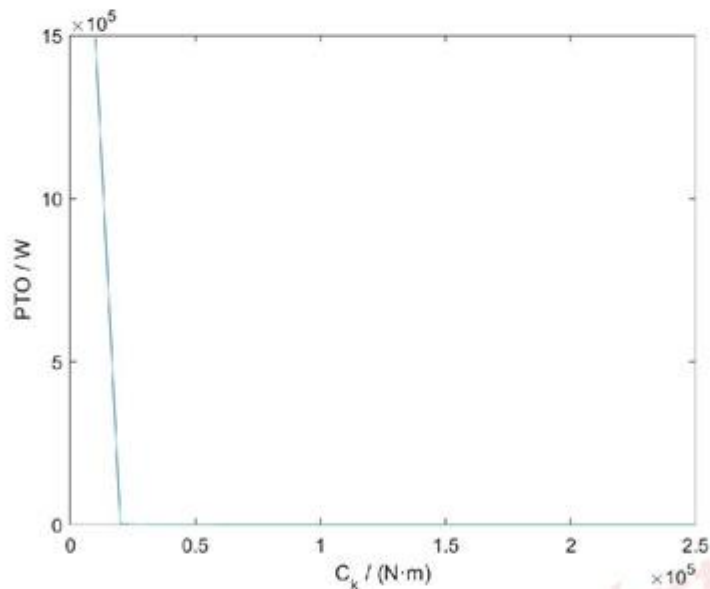


图 17 扭转弹簧刚度的稳定性分析图

可得，扭转弹簧刚度在题目给出数据的领域内变动，模型均成立，系统输出正常；扭转弹簧刚度小于一定值之后，模型崩溃。

6.2.4 模型对弹簧刚度的稳定性分析

对于不同大小的弹簧刚度，计算输出功率。

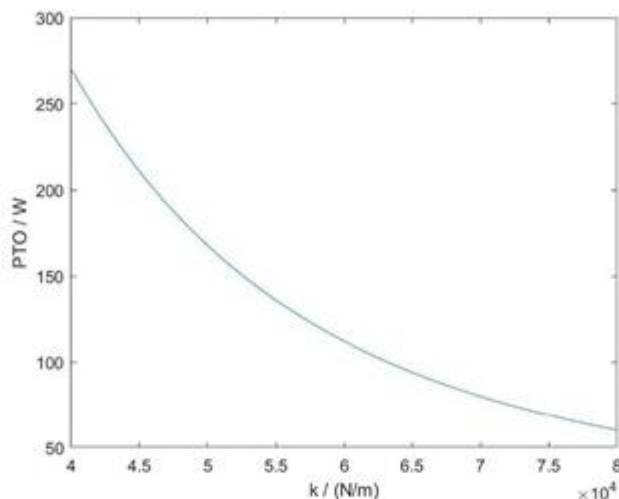


图 18 弹簧刚度的稳定性分析图

可得，弹簧刚度在题目给出数据的领域内变动，模型均成立，系统输出正常。

七、模型的评价

7.1 模型优点

1. 原创性高，本文中所有模型均为自主建立；
2. 在问题二中，给出解析解和积分解两种解法，准确性更高；
3. 在问题三中，对模型进行了合理简化，避免了对转轴和中轴受力（矩）的讨论；
4. 在问题三中，巧妙地选取了平动非惯性系，使模型便于理解也便于计算；

7.2 模型缺点与改进

1. 模型忽略了该波浪能装置中次要部件的体积、质量以及摩擦带来的影响，因此和生产生活中的实际情况存在差异；
2. 问题三和问题四的模型建立在浮子和振子的纵摇为小角度、相对位移为小量的基础上，当波浪较大时可能会出现较大误差，详见“六、模型的分析与检验”；
3. 问题二和问题四中采取的是积分法求数值解，未能给出解析解的公式，可能会略微降低精度，同时在模型引申使用时需要重新计算；

可能的改进方案是：

1. 加入次要部件的体积、质量以及摩擦的影响；
2. 充分分析转轴和中轴处的作用力（矩），将振子与浮子分开分析，重新列得微分方程；

3. 尽可能给出更精确的积分算法, 或寻找更高的理论解得解析解;

7.3 模型推广

本文对一种波浪能装置的能量转化过程作了理论分析, 并构建了数学模型。通过对直线阻尼系数和旋转阻尼系数不同取值的分析, 得到了使得输出功率最大化的阻尼系数及其形式。同理, 该模型可以用于分析, 当弹簧刚度和扭转弹簧刚度、或浮子、振子的质量比发生变化时, 寻求使得输出功率最大化的弹簧刚度或质量比, 以提升波浪能装置的转换效率。

参考文献

- [1]李慧彬. 振动理论与工程应用. 北京理工大学出版社. 2006 年 9 月第一版.
- [2]郑雄波, 张亮, 马勇. 双浮体波能装置的水动力计算与能量转换特性分析[J]. 科技导报, 2014, 32(19): 26-30.

附录

| 附录 1 | | | |
|------------|---------------------|----------------|------------------------|
| 支撑文件清单 | | | |
| 文件夹名 | | 文件名 | 含义 |
| 数据 | | result1-1.xlsx | 问题一(1)结果 |
| | | result1-2.xlsx | 问题一(2)结果 |
| | | result3.xlsx | 问题三结果 |
| 代码 | 1 | code1.m | 主函数 |
| | | func1.m | 微分方程 |
| | 2 | code2.1_积分解.m | 积分法求解 |
| | | code2.1_解析解.m | 公式求解 |
| | | code2.2-积分解.m | 积分求解 |
| | | func2.m | 微分方程 |
| | 3 | code3.m | 主函数 |
| | | func3.m | 微分方程 |
| | 4 | code4.m | 主函数 |
| | | func4.m | 微分方程 |
| 其他 支撑文件 | 敏 感 性 检 验 代 码 | 问题 4 数据表 | 支撑问题 4 对于 η_j 的假设 |
| | | code5.m | 主函数 |
| | | func5.m | 微分方程 |

| 附录 2 |
|---|
| 关于问题 2-1 解析解 ω 相等的引证 |
| 李慧彬. 振动理论与工程应用. 北京理工大学出版社. 2006 年 9 月第一版. 第 20 页-第 22 页-2.3.2 线性阻尼系统对简弦激励的响应 由牛顿运动定律得振动微分方程: $m\ddot{x} + c\dot{x} + kx = F_0 \sin \omega t$ |

引入记号:

$$p^2 \equiv \frac{k}{m}, X_0 \equiv \frac{F_0}{k}, \zeta \equiv \frac{c}{2pm} = \frac{c}{2\sqrt{mk}} = \frac{c}{c_c}$$

得到对应于初始条件的解为:

$$x = e^{-\zeta pt} \left(\frac{\dot{x}_0 + \zeta p x_0}{q} \sin qt + x_0 \cos qt \right) - X_0 e^{-\zeta pt} \left(\frac{\zeta p \sin \psi + \omega \cos \psi}{q} \sin qt + \sin \psi \cos qt \right) + X \sin(\omega t - \psi)$$

不难发现, 经过有限长时间后, 方程前两项由于阻尼存在衰减为0, 故可得稳态解圆频率即为激励力圆频率 ω 。

附录 3

问题 1 程序 code1.m

```
1. %% argument
2. m_f = 4866;
3. mu_1 = 1335.535;
4. eta_i = 10000;
5. lambda_i = 656.3616;
6. k = 80000;
7. rho = 1025;
8. g = 9.8;
9. R_f = 1;
10. f = 6250;
11. omega = 1.4005;
12. m_z = 2433;
13.
14. x_0 = [0 0 0 0];
15. tspan = [0:0.2:180];
16.
17. %% run
18. [t, x] = ode45(@(t, x) func1(t, x), tspan, x_0);
19.
20. %% plot
21. plot(t, x(:, 1), '-', t, x(:, 3), '-')
```

附录 4

问题 1 程序 func1.m

```
1. function dxdt = func1(t, x)
2.     m_f = 4866;
3.     mu_1 = 1335.535;
4.     eta_i = 10000;
5.     lambda_i = 656.3616;
6.     k = 80000;
7.     rho = 1025;
8.     g = 9.8;
9.     R_f = 1;
10.    f = 6250;
11.    omega = 1.4005;
12.    m_z = 2433;
13.
14.    dxdt = zeros(4, 1);
```

```

15. dxdt(1)=x(2);
16. dxdt(2)=(f*cos(omega*t)-eta_i*(x(2)-x(4))*sqrt(abs(x(2)-x(4)))-lambda_i*x(2)-k*(
x(1)-x(3))-rho*g*pi*R_f^2*x(1))/(m_f+mu_1);
17. dxdt(3)=x(4);
18. dxdt(4)=(-eta_i*(x(4)-x(2))*sqrt(abs(x(2)-x(4)))-k*(x(3)-x(1)))/m_z;
19.
20. end

```

附录 5

问题 2 程序 code2.1_积分解.m

```

1. % 积分法求解 PTO
2.
3. %% argument
4. m_f=4866;
5. mu_1=1165.992;
6. eta_i=eta_i_best;
7. lambda_i=167.8395;
8. k=80000;
9. rho=1025;
10. g=9.8;
11. R_f=1;
12. f=4890;
13. omega=2.2143;
14. m_z=2433;
15.
16. alpha=0;
17.
18. x_0=[0 0 0 0];
19. tspan=[0:0.2:400];
20.
21. %% run
22. [t,x]=ode45(@(t,x) func2(t,x,eta_i,alpha),tspan,x_0);
23. % plot(t,x(:,1),'-t,x(:,3),'-')
24. % plot(t,x(:,2)-x(:,4),'-')
25.
26.
27. %% v
28. delta_v=x(1000:2000,2)-x(1000:2000,4)
29.
30. pto=eta_i.*delta_v.^2;
31. PTO=trapz(pto)/(1000)

```

附录 6

问题 2 程序 code2.1_解析解.m

```

1. %% argument
2.
3. m_f=4866;
4. mu_1=1165.992;
5. % eta_i=10000;
6. lambda_i=167.8395;
7. k=80000;
8. rho=1025;
9. g=9.8;
10. R_f=1;
11. f=4890;

```

```

12. omega = 2.2143;
13. m_z = 2433;
14.
15. %% A, B
16. A = -omega^2 * (m_f + mu_1) + k + rho * g * pi * R_f^2;
17. B = -omega^2 * m_z + k;
18. C = A * B - k^2;
19. D = omega^2 * lambda_i;
20. E = A + B - 2 * k;
21.
22. %% eta_i
23. eta_i_best = sqrt((C^2 + omega^2 * B^2 * lambda_i^2) / (D^2 + omega^2 * E^2));
24. eta_i_best
25.
26. eta_i = eta_i_best;
27.
28. PTO = 0.5 * (omega^6 * m_z^2 * f^2 * eta_i) / ((C - D * eta_i)^2 + omega^2 * (E * eta_i + B * lambda_i)^2);
29.
30. %% plot
31. eta_i = [30000:1:40000];
32.
33. PTO = 0.5 * (omega^6 * m_z^2 * f^2 * eta_i) / ((C - D * eta_i)^2 + omega^2 * (E * eta_i + B * lambda_i)^2);
34.
35. plot(eta_i, PTO, '-')

```

附录 7

问题 2 程序 code2.2-积分解.m

```

1. % 积分法求解 PTO
2.
3. %% argument
4. m_f = 4866;
5. mu_1 = 1165.992;
6. lambda_i = 167.8395;
7. k = 80000;
8. rho = 1025;
9. g = 9.8;
10. R_f = 1;
11. f = 4890;
12. omega = 2.2143;
13. m_z = 2433;
14.
15. x_0 = [0 0 0];
16. tspan = [0:0.2:400];
17.
18. %% run-local
19. PTO = zeros(21, 141);
20.
21. for alpha_index = 1:21
22.     alpha = (alpha_index - 1) * 0.05;
23.
24.     for eta_i_index = 1:141
25.         eta_i = (eta_i_index - 1) * 500 + 30000;
26.         [t, x] = ode45(@(t, x) func2(t, x, eta_i, alpha), tspan, x_0);
27.
28.         delta_v = x(1000:2000, 2) - x(1000:2000, 4);
29.
30.         pto = eta_i * (abs(delta_v)).^(2 + alpha);
31.         PTO(alpha_index, eta_i_index) = trapz(pto) / (1000);
32.     end

```



```

33.
34.     end
35.     % 通过不断改变参数区间，即可精确化极值点。此处只放了第一个版本
36.
37.     %% PTO
38.     PTO
39.
40.     mesh([30000:500:100000], [0:.05:1], PTO)

```

附录 8

问题 2 程序 func2.m

```

1.     function dxdt = func2(t, x, eta_i, alpha)
2.         m_f = 4866;
3.         mu_1 = 1165.992;
4.         lambda_i = 167.8395;
5.         k = 80000;
6.         rho = 1025;
7.         g = 9.8;
8.         R_f = 1;
9.         f = 4890;
10.        omega = 2.2143;
11.        m_z = 2433;
12.
13.        dxdt = zeros(4, 1);
14.        dxdt(1) = x(2);
15.        dxdt(2) = (f * cos(omega * t) - eta_i * (x(2) - x(4)) * (abs(x(2) - x(4)))^alpha - lambda_i * x(2) - k
16.        * (x(1) - x(3)) - rho * g * pi * R_f^2 * x(1)) / (m_f + mu_1);
17.        dxdt(3) = x(4);
18.        dxdt(4) = (-eta_i * (x(4) - x(2)) * (abs(x(2) - x(4)))^alpha - k * (x(3) - x(1))) / m_z;
19.    end

```

附录 9

问题 3 程序 code3.m

```

1.     %% argument
2.     h_y = 3;
3.     h_z = 0.8;
4.     R_z = 0.5;
5.     H_z = 0.5;
6.     L_0 = 0.5;
7.     C_k = 250000;
8.     C_w = 8890.7;
9.     m_f = 4866;
10.    m_z = 2433;
11.    rho = 1025;
12.    g = 9.8;
13.    R_f = 1;
14.    k = 80000;
15.
16.    J = 7001.914;
17.    lambda_i = 683.4558;
18.    lambda_j = 654.3383;
19.    omega = 1.7152;
20.    mu = 1028.876;
21.    f = 3640;
22.    M_j = 1690;

```

```

23.
24. eta_i = 10000;
25. eta_j = 1000;
26.
27. h_1 = L_0 + 0.5 * H_z - m_z * g / k;
28. h_2 = (sqrt(41) * (-h_z / 3) + 30 * (h_y / 2) + 5 * h_y) / (35 + sqrt(41));
29. h_3 = (m_z * h_1 + m_f * h_2) / (m_z + m_f);
30.
31. l_z = 0.25 * m_z * R_z^2 + (1/12) * m_z * H_z^2;
32. l_f = (m_f / (35 + sqrt(41))) * (sqrt(41) * (0.5 * h_z^2 + 0.25 * R_f^2 - (2 * h_z / 3)^2 + (h_z / 3 + h_2)^2) + 30 * (0.5 * R_f^2 + (1/12) * h_y^2) + 5 * (0.25 * R_f^2 + (h_y - h_2)^2));
33.
34. %% run
35. x_0 = [0 0 0 0 0 0 0];
36. tspan = [0:0.2:400];
37.
38. [t, x] = ode45(@(t, x) func3(t, x, eta_i, eta_j), tspan, x_0);
39.
40. plot(t, x(:, 1), '-');
41. plot(t, x(:, 5), '-');
42. plot(t, x(:, 3), '-');
43. plot(t, x(:, 7), '-');
44.
45. %% pto
46. xx_2_dot = x(1000:2000, 6);
47. delta_theta = x(1000:2000, 8) - x(1000:2000, 4);
48.
49. pto = eta_i .* x_2_dot.^2 + eta_j .* delta_theta.^2;
50. PTO(:, eta_i_index) = trapz(pto) / (1000);

```

附录 10

问题 3 程序 func3.m

```

1. function dxdt = func3(t, x, eta_i, eta_j)
2.
3.     h_y = 3;
4.     h_z = 0.8;
5.     R_z = 0.5;
6.     H_z = 0.5;
7.     L_0 = 0.5;
8.     C_k = 250000;
9.     C_w = 8890.7;
10.    m_f = 4866;
11.    m_z = 2433;
12.    rho = 1025;
13.    g = 9.8;
14.    R_f = 1;
15.    k = 80000;
16.
17.    J = 7001.914;
18.    lambda_i = 683.4558;
19.    lambda_j = 654.3383;
20.    omega = 1.7152;
21.    mu = 1028.876;
22.    f = 3640;
23.    M_j = 1690;
24.    eta_i = 10000;
25.    eta_j = 1000;
26.
27.    h_1 = L_0 + 0.5 * H_z - m_z * g / k;
28.    h_2 = (sqrt(41) * (-h_z / 3) + 30 * (h_y / 2) + 5 * h_y) / (35 + sqrt(41));

```

```

29.     h_3 = (m_z * h_1 + m_f * h_2) / (m_z + m_f);
30.
31.     l_z = 0.25 * m_z * R_z^2 + (1/12) * m_z * H_z^2;
32.     l_f = (m_f / (35 + sqrt(41))) * (sqrt(41) * (0.5 * h_z^2 + 0.25 * R_f^2 - (2 * h_z / 3)^2 + (h_z / 3 +
h_2)^2) + 30 * (0.5 * R_f^2 + (1/12) * h_y^2) + 5 * (0.25 * R_f^2 + (h_y - h_2)^2));
33.
34.     dxdt = zeros(8, 1);
35.
36.     dxdt(1) = x(2);
37.     dxdt(2) = (f * cos(omega * t) + k * x(5) + eta_i * x(6) - rho * g * pi * R_f^2 * x(1) - lambda_i * x(
2)) / (m_f + mu);
38.     dxdt(3) = x(4);
39.     dxdt(4) = (M_j * cos(omega * t) - C_w * x(3) - lambda_j * x(4)) / (l_z + l_f + m_z * (h_3 - h_1)^2
+ m_f * (h_3 - h_2)^2);
40.     dxdt(5) = x(6);
41.     dxdt(6) = g * (1 - cos(x(7))) - dxdt(2) * cos(x(7)) + dxdt(4) * h_3 * sin(x(7) - x(3)) -
eta_i * x(6) / m_z - k * x(5) / m_z - (x(4)^2 * h_3 * cos(x(7) - x(3)));
42.     dxdt(7) = x(8);
43.     dxdt(8) = (m_z * ((g + dxdt(2)) * sin(x(7)) + (x(4))^2 * h_3 * sin(x(7) - x(3)) + dxdt(4) * h_3 * cos
(x(7) - x(3))) * (h_1 + x(5)) - eta_j * (x(8) - x(4)) - C_k * (x(7) - x(3))) / (l_z + m_z * (h_1 + x(5))^2);
44.
45.     end

```

附录 11

问题 4 程序 func4.m

```

1. %% argument
2. h_y = 3;
3. h_z = 0.8;
4. R_z = 0.5;
5. H_z = 0.5;
6. L_0 = 0.5;
7. C_k = 250000;
8. C_w = 8890.7;
9. m_f = 4866;
10. m_z = 2433;
11. rho = 1025;
12. g = 9.8;
13. R_f = 1;
14. k = 80000;
15.
16. J = 7142.493;
17. lambda_i = 528.5018;
18. lambda_j = 1655.909;
19. omega = 1.9806;
20. mu = 1091.099;
21. f = 1760;

```

```

22. M_j = 2140;
23. % eta_i = 10000;
24. % eta_j = 1000;
25.
26. h_1 = L_0 + 0.5 * H_z - m_z * g / k;
27. h_2 = (sqrt(41) * (-
    h_z / 3) + 30 * (h_y / 2) + 5 * h_y) / (35 + sqrt(41));
28. h_3 = (m_z * h_1 + m_f * h_2) / (m_z + m_f);
29.
30. I_z = 0.25 * m_z * R_z^2 + (1/12) * m_z * H_z^2;
31. I_f = (m_f / (35 + sqrt(41))) * (sqrt(41) * (0.5 * h_z^2 + 0.25 * R_f^2 -
    (2 * h_z / 3)^2 + (h_z / 3 + h_2)^2) + 30 * (0.5 * R_f^2 + (1/12) * h_y^2)
    + 5 * (0.25 * R_f^2 + (h_y - h_2)^2));
32.
33. %% run_global
34. x_0 = [0 0 0 0 0 0 0 0];
35. tspan = [0:0.2:400];
36.
37.
38. PTO = zeros(101, 101);
39.
40. for eta_i_index = 1:101
41.     eta_i = (eta_i_index - 1) * 1000;
42.
43.     for eta_j_index = 1:101
44.         eta_j = (eta_j_index - 1) * 1000;
45.         [t, x] = ode45(@(t, x) func4(t, x, eta_i, eta_j), tspan, x_0);
46.
47.         xx_2_dot = x(1000:2000, 6);
48.         delta_theta_dot = x(1000:2000, 8) - x(1000:2000, 4);
49.
50.         pto = eta_i .* xx_2_dot.^2 + eta_j .* delta_theta_dot.^2;
51.         PTO(eta_i_index, eta_j_index) = trapz(pto) / (1000);
52.     end
53.
54. end
55.
56. %% mesh
57. mesh([0:1000:100000], [0:1000:100000], PTO)

```



```

58.
59. %% run-local
60.
61. x_0 = [0 0 0 0 0 0 0 0];
62. tspan = [0:0.2:400];
63.
64. eta_j = 100000;
65. PTO = zeros(151, 1);
66.
67. for eta_i_index = 1:151
68.     eta_i = (eta_i_index - 1) * 100 + 50000;
69.
70.     [t, x] = ode45(@(t, x) func4(t, x, eta_i, eta_j), tspan, x_0);
71.
72.     xx_2_dot = x(1000:2000, 6);
73.     delta_theta_dot = x(1000:2000, 8) - x(1000:2000, 4);
74.
75.     pto = eta_i .* x_2_dot.^2 + eta_j .* delta_theta_dot.^2;
76.     PTO(eta_i_index) = trapz(pto) / (1000);
77.
78. end
79.
80. %% PTO
81. PTO
82.
83. plot([35000:100:37000], PTO)
84. [maxPTO, maxPTO_index] = max(PTO)

```

附录 12

问题 4 程序 func4.m

```

85. function dxdt = func4(t, x, eta_i, eta_j)
86.
87.     h = 3;
88.     h_z = 0.8;
89.     R_z = 0.5;
90.     H_z = 0.5;
91.     L_0 = 0.5;
92.     C_k = 250000;
93.     C_w = 8890.7;
94.     m_f = 4866;
95.     m_z = 2433;

```

```

96.     rho = 1025;
97.     g = 9.8;
98.     R_f = 1;
99.     k = 80000;
100.
101.     J = 7142.493;
102.     lambda_i = 528.5018;
103.     lambda_j = 1655.909;
104.     omega = 1.9806;
105.     mu = 1091.099;
106.     f = 1760;
107.     M_j = 2140;
108.     % eta_i = 10000;
109.     % eta_j = 1000;
110.
111.     h_1 = L_0 + 0.5 * H_z - m_z * g / k;
112.     h_2 = (sqrt(41) * (-h_z / 3) + 30 * (h_y / 2) + 5 * h_y) / (35 + sqrt(41));
113.     h_3 = (m_z * h_1 + m_f * h_2) / (m_z + m_f);
114.
115.     l_z = 0.25 * m_z * R_z^2 + (1/12) * m_z * H_z^2;
116.     l_f = (m_f / (35 + sqrt(41))) * (sqrt(41) * (0.5 * h_z^2 + 0.25 * R_f^2 - (2 * h_z / 3)^2 + (h_z / 3 + h_2)^2) + 30 * (0.5 * R_f^2 + (1/12) * h_y^2) + 5 * (0.25 * R_f^2 + (h_y - h_2)^2));
117.
118.     dxdt = zeros(8, 1);
119.
120.     dxdt(1) = x(2);
121.     dxdt(2) = (f * cos(omega * t) + k * x(5) + eta_i * x(6) - rho * g * pi * R_f^2 * x(1) - lambda_i * x(2)) / (m_f + mu);
122.     dxdt(3) = x(4);
123.     dxdt(4) = (M_j * cos(omega * t) - C_w * x(3) - lambda_j * x(4)) / (l_z + l_f + m_z * (h_3 - h_1)^2 + m_f * (h_3 - h_2)^2);
124.     dxdt(5) = x(6);
125.     dxdt(6) = g * (1 - cos(x(7))) - dxdt(2) * cos(x(7)) + dxdt(4) * h_3 * sin(x(7) - x(3)) - eta_i * x(6) / m_z - k * x(5) / m_z - (x(4)^2 * h_3 * cos(x(7) - x(3)));
126.     dxdt(7) = x(8);
127.     dxdt(8) = (m_z * ((g + dxdt(2)) * sin(x(7)) + (x(4))^2 * h_3 * sin(x(7) - x(3)) + dxdt(4) * h_3 * cos(x(7) - x(3))) * (h_1 + x(5)) - eta_j * (x(8) - x(4)) - C_k * (x(7) - x(3))) / (l_z + m_z * (h_1 + x(5))^2);
128.
129.     end

```