

第十八届全国大学生结构设计竞赛总决赛理论方案

《乌篷船模型结构设计与制作》

第十八届全国大学生结构设计竞赛组织委员会

2025 年 10 月

目 录

第一部分：模型选项

1 方案构思	1
1.1 乌篷船	1
1.2 船型减阻设计	2
1.2.1 航行阻力分析	3
1.2.2 船体各部位减阻设计要点	3
1.3 赛题基本要求	5
1.3.1 模型尺寸及边界条件	5
1.3.2 待定参数	6
1.3.3 评分标准	6
1.4 方案比选	7
1.4.1 缩尺船型比选	7
1.4.2 结构方案比选	12
1.4.3 加载方案比选	16

第二部分：设计计算

2 试验方面	22
2.1 材料测试	22
2.2 构件测试	23
2.3 结构测试	34
3 结构建模及主要参数	36
3.1 模型构造	36
3.2 建模参数	37
3.2.1 材料部分	37
3.2.2 几何信息部分	37
3.2.3 荷载工况	40
3.2.4 边界条件	41
4 受力分析	43
4.1 强度分析	43

4.1.1 第一级加载	43
4.1.2 第二级荷载	46
4.1.3 第三级荷载	50
4.2 刚度分析	53
4.3 稳定分析	57
4.4 小结	61
5 模型尺寸图	62
6 节点构造与施工	64
6.1 节点构造	64
6.2 组装工艺	65
6.3 粘贴工艺	67

第三部分：实训过程总结

7 参赛心得与体会	68
-----------------	----

作品总结

附录 A：电机数据解析代码

附录 B：电机试验代码

附录 C：卡尔曼滤波代码

第一部分：模型选型

1 方案构思

1.1 乌篷船

“轻舟八尺，低篷三扇，占断蘋洲烟雨。”一提到江南，乌篷船是绕不开的印象。乌篷船作为江南水乡的典型交通工具，其造型与结构特征经过长期实践沉淀，蕴含了丰富的船舶设计经验。在复杂水域条件下，乌篷船能够兼顾稳定性、适航性与轻便性，为本次赛题的模型方案构思提供了重要参考价值。

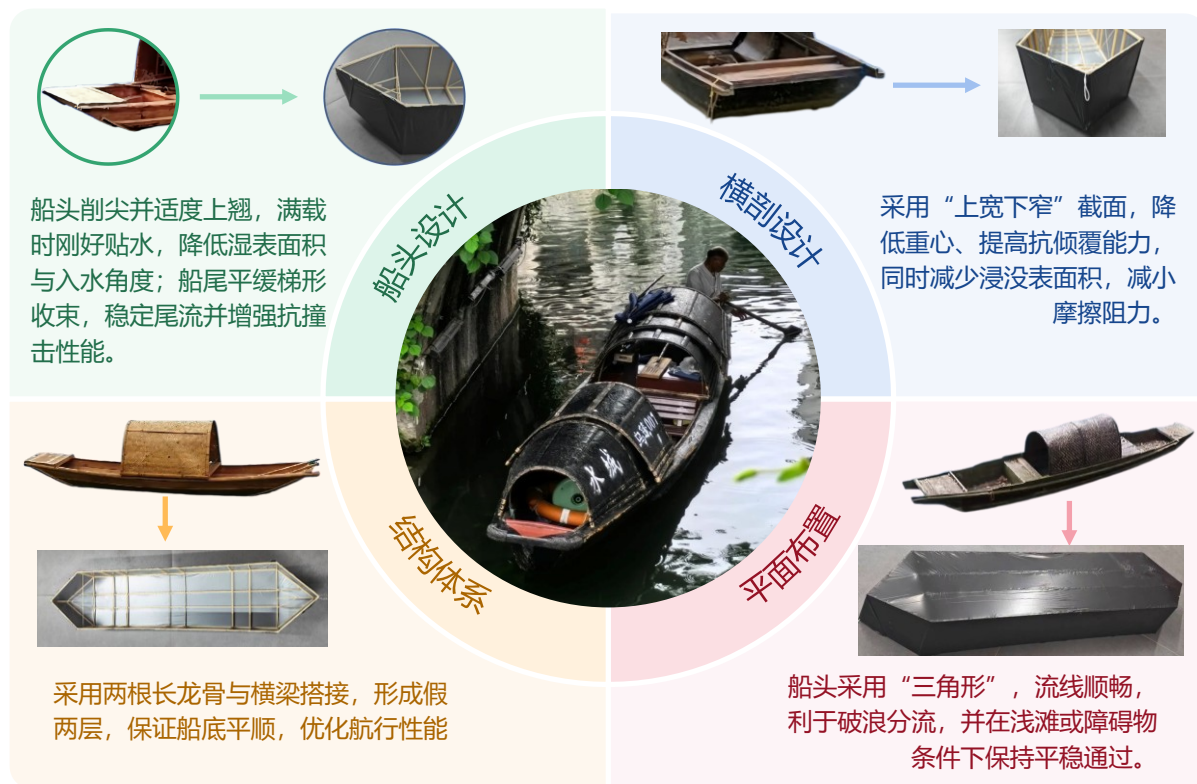


图 1-1 模型设计思路

通过对图 1-1 乌篷船结构与几何特征的分析，我们获得了以下灵感：

(1) **整体外形**：乌篷船“中间宽、首尾窄”的构型兼顾了承载能力与稳定性，同时优化了流线型，减少水流阻力和波浪扰动，提高了适应不同水域环境的能力。

(2) **船首设计**：乌篷船普遍采用上翘的船首形态，以降小入水角度，利于局部流体分离，从而在破浪与波浪扰动环境下能保持很好的航行性能。

(3) **船身比例**：船身纵向线型细长、长宽比大的特点保证了其优良的直线航行性能和抗偏航能力，这对模型在牵引航行中的速度和稳定性具有重要意义。

(4) **船底布置**：平顺而浅吃水的底部设计在保证浮力的同时，能有效降低水阻力，适合浅水和低速航行条件，提高了模型在浅水池条件下的适应性。

基于以上分析，本次方案将在参考传统乌篷船成熟经验的基础上，结合结构力学与船舶流体力学方法，进一步开展工程化优化与方案设计。

1.2 船型减阻设计

船舶在航行过程中影响其动能的主要因素是水的阻力。水阻力不仅取决于船体的几何外形，还与船速、流体介质特性以及航行状态密切相关，详见图 1-2、图 1-3。

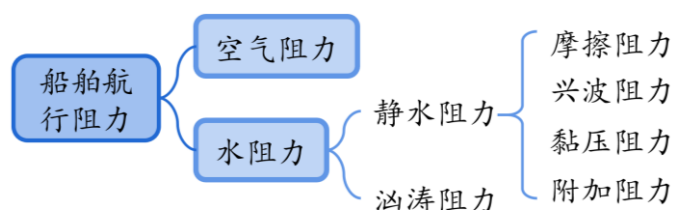


图 1-2 船舶航行阻力分类

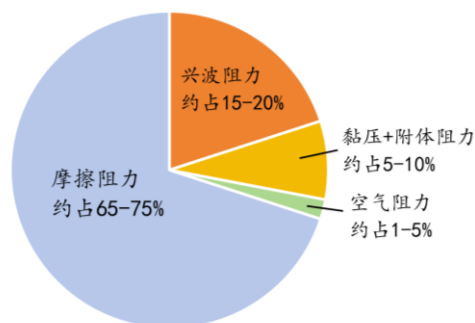


图 1-3 模型船航行各阻力比例

根据船舶设计原理，模型船的航行阻力可以分为以下三类：摩擦阻力 R_f 、兴波阻力 R_w 和附加阻力 R_a 。其中，摩擦阻力源于船体湿表面与水体之间的黏性作用；兴波阻力由船体在航行中扰动自由液面所产生的波浪引起；附加阻力则主要包括涡流、船尾回流不均及局部流动分离等带来的能量损耗。船舶总阻力可表示为：

$$R_T = R_f + R_w + R_a$$

其中，摩擦阻力常通过摩擦阻力系数 C_f 来计算：

$$R_f = \frac{1}{2} \rho V^2 S C_f$$

$$C_f = \frac{0.075}{(\log_{10} Re - 2)^2}, \quad Re = \frac{VL}{\nu}$$

式中 ρ 为水体密度， V 为航速， S 为船体湿表面积， L 为特征长度， ν 为水的运动黏度。由此可见，船体尺度、速度及表面光顺性对摩擦阻力均具有显著影响。

1.2.1 航行阻力分析

本次赛题要求的模型船航速较低,船体弗劳德数 $Fn = \frac{v}{\sqrt{gL}}$ 在 0.2~0.3 范围内,属于低速船。

在此速度区间内,摩擦阻力通常占主导,而兴波阻力次之,附加阻力则与船体局部线型和加工工艺相关。通过对阻力分量的估算,我们发现摩擦阻力约占总阻力的 60%以上,因此在船型设计中,减小湿表面积、保持船底平顺和避免不必要的波浪生成成为关键。

1.2.2 船体各部位减阻设计要点

船体各局部构造在减阻中发挥着不同作用:

(1) **船首设计:** 船首是与水体接触的第一部分,其几何形状直接决定入流分离及波浪生成情况。通过适度的上翘设计,可以使船首在入水时逐渐抬升,减小迎浪面积,从而降低兴波阻力。同时,上翘角度避免了波浪附着在船体表面,改善流场分布,详见图 1-4、图 1-5。然而,上翘过大则会导致船体有效长度增加,摩擦阻力上升,因此需在兴波与摩擦之间找到平衡点。理论上,船首的棱角与曲率半径控制了自由液面的扰动强度,合理的“尖削”比例能够显著降低兴波阻力 R_w 。



图 1-4 直立型首

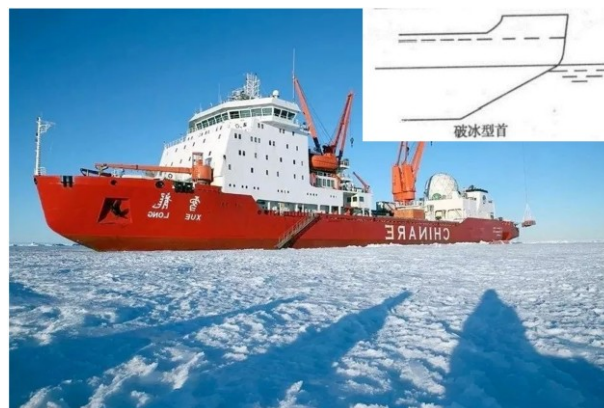


图 1-5 破冰型首

(2) **船身中部设计:** 船身中部是湿表面积最大的部分,其丰满度系数 C_m 与横剖面积直接影响摩擦阻力大小。过宽的船体虽然能提高静力稳定性,但湿表面积显著增加,摩擦阻力急剧上升;反之,过窄的船体虽可降低摩擦力,但稳性不足,易于横倾。因此,通过适度缩减船身中部宽度,使其既能维持必要的稳定性,又能在一定程度上减少湿表面积。这样,摩擦阻力项 R_f 得以有效控制。

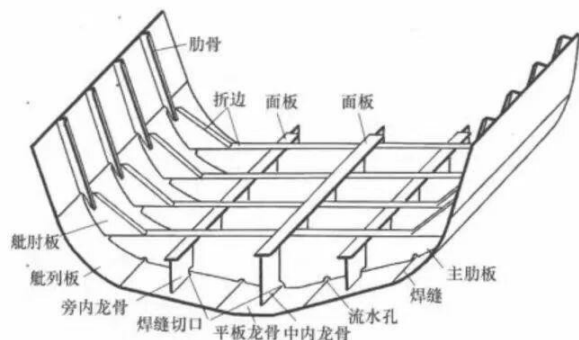


图 1-6 横骨架式单底结构



图 1-7 方型船尾

(3) **船尾设计**：船尾区影响尾流分布和能量损失，是附加阻力的重要来源。若船尾收缩不顺畅，易引起回流和涡流，从而增加附加阻力。合理的船尾曲率设计能够平缓地引导水流闭合，减小尾涡和局部分离，使得阻力降低。同时，顺畅的尾流改善了推进器工作环境（虽然本模型为无动力航行，但这一原理依旧适用）。理论上，船尾线型直接关系到阻力系数中的附加部分 C_a ，设计良好的船尾可使 C_a 显著减小。

(4) **船底设计**：船底作为与水体接触面积最大的部分，其刚性和平顺性对摩擦阻力起决定性作用。采用刚性船底能够避免因船底变形引起的局部波浪和附加摩擦；同时，平顺的船底能够维持层流或延迟湍流转捩，从而降低摩擦系数 C_f 。此外，平整的底板有助于减小附着水体的紊动层厚度，进一步削减摩擦阻力。

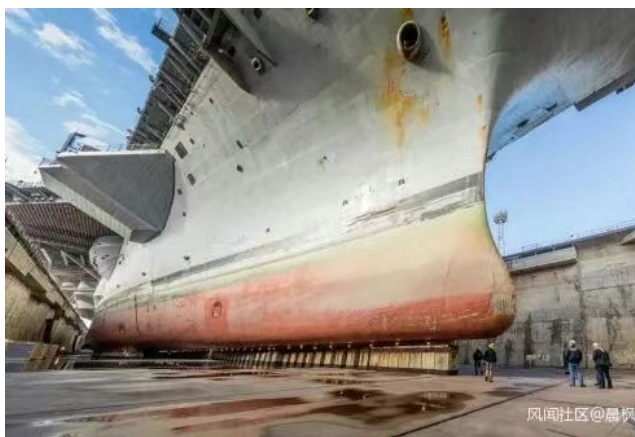


图 1-8 平顺船底

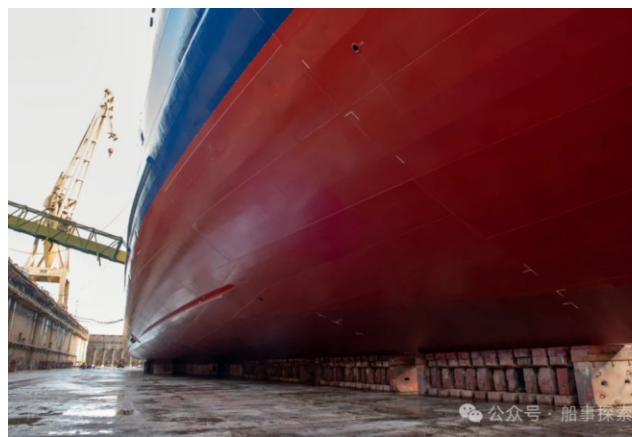


图 1-9 光滑船底

综上所述，船首的劈波性、船中的丰满度、船尾的回流控制以及船底的光滑性共同决定了船舶的阻力特性。通过对各部位线型的合理优化，可在理论上有效降低摩擦与兴波阻力，进而提升船舶航行的能效与稳定性。

1.3 赛题基本要求

本届竞赛以江南水乡传统乌篷船为设计原型，要求参赛队设计并制作一个船体模型，以应对减阻、抗倾覆与轻质化等综合性能考验。不同于以往偏重塔式或桥梁受力的题目，本次赛题更加突出水动力与结构稳定性的耦合特征。具体要求如下：

1.3.1 模型尺寸及边界条件

船体结构必须满足“中间宽、两端窄，上宽下窄”的基本几何特征，总长约 1200 mm，最大宽度与高度分别为 320 mm 和 200 mm。船体内部需预留规避区 I（船舱）和 II/III（座位）作为加载与净空区域，任何构件不得侵入。

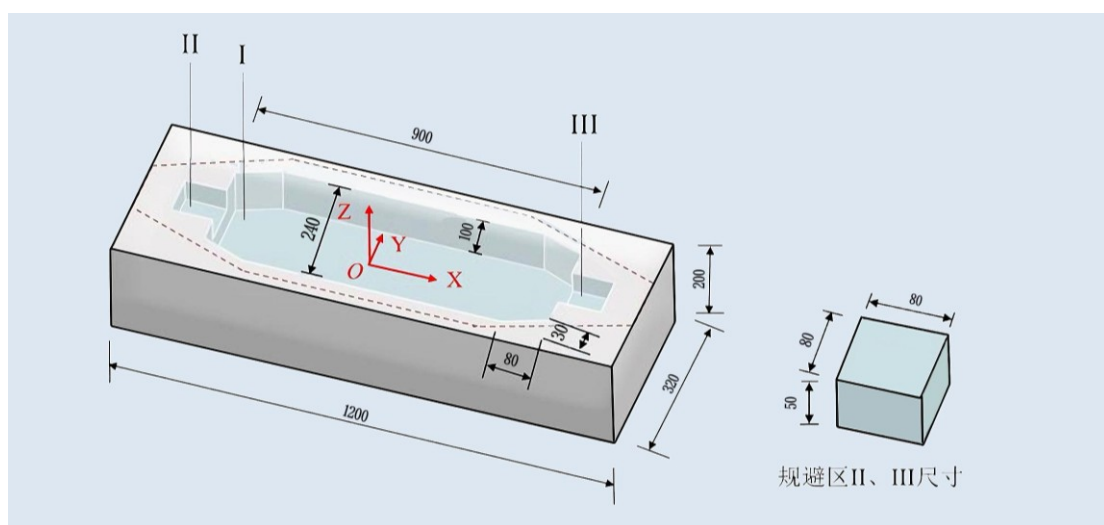


图 1-10 船体规避区示意图

船篷在施加荷载状态下需满足规避区 IV 与限高要求。

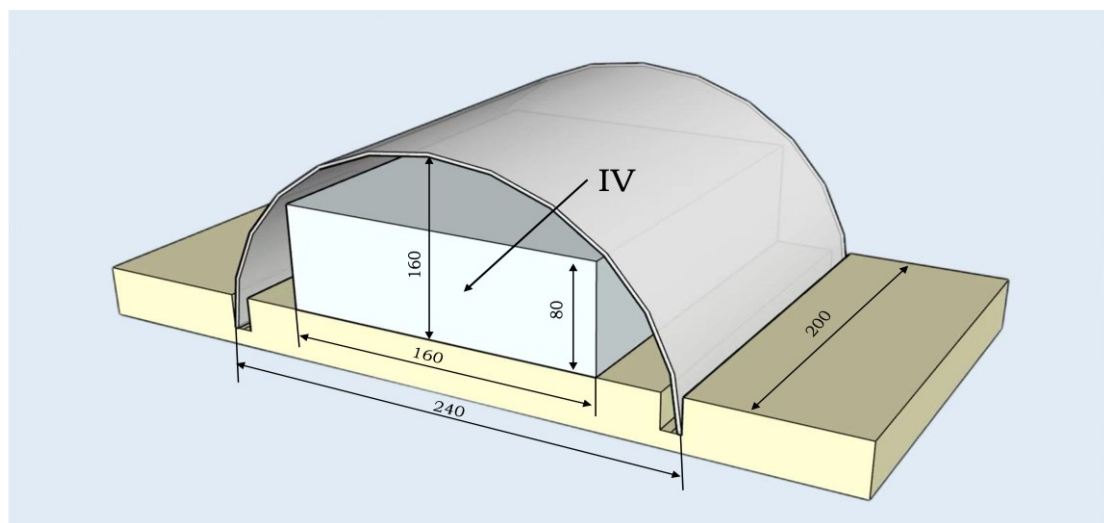


图 1-11 船篷检测示意图

1.2.2 待定参数

赛题中，第一级橘色指示棒位置由抽签决定，可选 6 或 8 kg 坛状砝码，第二级可选 1.0 kg 或 1.5 kg 侧向载荷，第三级可选 6、8 或 10 kg 坛状砝码，并需自定砂桶位置。抽签与自选参数使加载工况具有不确定性，因此设计需兼顾多种路径的安全性。尤其在第三级，坛状砝码布置方式对内力分布和上弦弯矩峰值影响显著。通过优化坛状砝码布置位置，可降低关键截面弯矩，提升结构的安全性与合理性。

1.2.3 评分标准

为了更直观地理解今年的赛题评分规则，我们将评分细则拆解为六个维度：模型质量效率、航行阻力、承载能力、操作与协同、抗撞性与防水性以及模型标准性。按照每个维度在检测、一级、二级和三级不同阶段所占的分值，按加分与扣分情况分别堆叠显示绘制了图 1-12 评分比重条形图。通过这种可视化方式，可以直观比较各个维度对总成绩的贡献与侧重点。

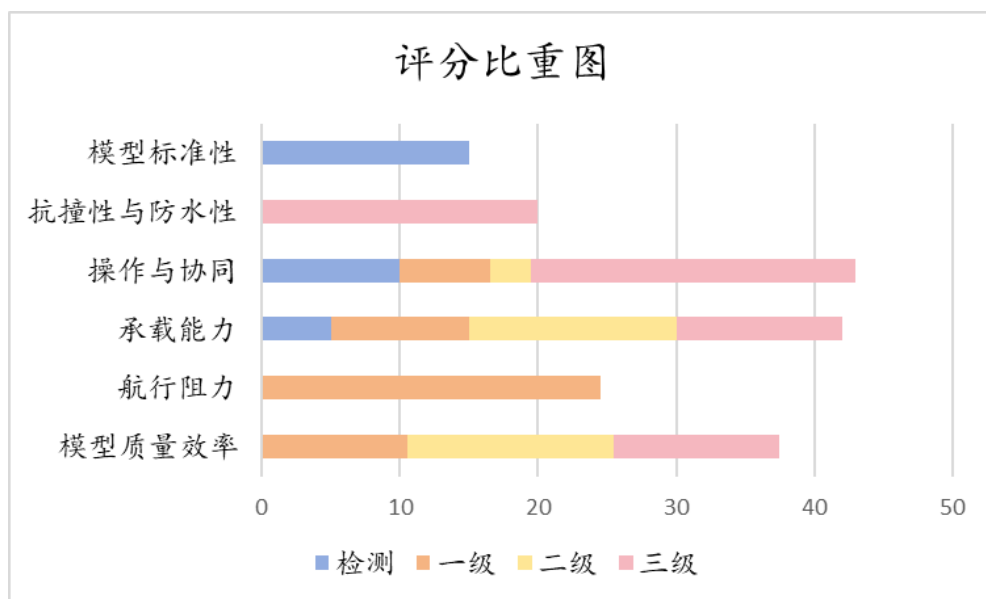


图 1-12 评分比重图

从图示的评分比重堆积条形图可以看出，本届赛题的评分维度更加多元，分值的分布也较往年有所变化。

模型检测阶段主要考察模型标准性与初步的承载能力、操作表现，这部分分值相对固定，体现了对基本规范与制作质量的重视。

一级加载分数构成出现明显倾斜，航行阻力与模型质量效率占据主导，说明在轻量化设计的同时，如何降低船体阻力、提升滑行速度成为评价重点。

二级加载则突出了承载能力与质量效率的乘积效应，强调在保证结构轻质化的同时具备较高的承重水平。

三级加载则最具综合性，既包含质量效率与承载能力，又增加了操作协同与抗撞性、防水性的比重，这表明在最终环节不仅要保证船体的力学性能与速度，还需兼顾结构鲁棒性与队员操作的稳定性。

总体而言，本届评分体系通过在不同阶段设置差异化的考察维度，使得最终得分呈现较强的不确定性：既要兼顾轻质化与阻力优化，又要平衡承载能力、航行速度和抗撞性。对比往年偏重单一荷质比的标准，今年的规则更注重多目标权衡与加载方案比选，要求参赛队在设计、制作与操作层面进行全方位的优化。

1.4 方案比选

根据赛题解读，本赛题对荷质比和航行阻力均提出了要求。因此在进行结构选型前，应首先对船身造型进行分析，确定航行阻力最小的选型方案，再依次对结构方案和加载方案进行对比分析。

1.4.1 缩尺船型比选

1) 试验背景

通过查阅资料，船舶阻力性能是船头、船身与船底等多构型耦合作用下的综合结果。为系统研究其影响，本研究从大量理论方案中，排除了存在明显性能冲突的设计，最终遴选出 13 种具有代表性的缩尺模型。

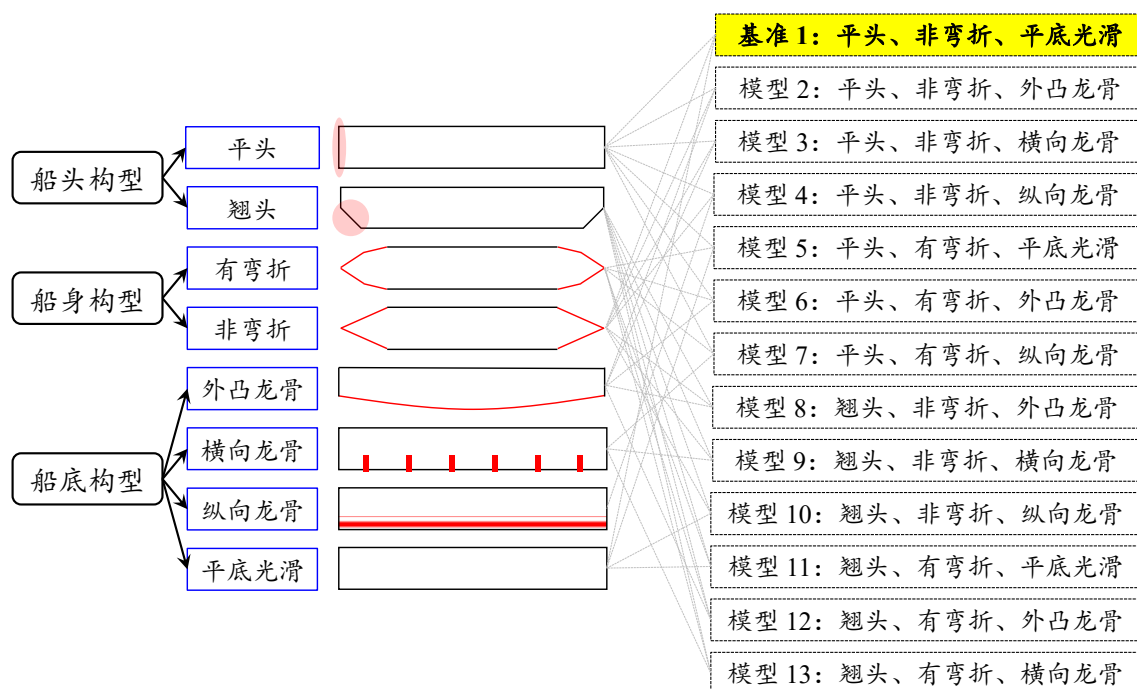


图 1-13 13 种具有代表性的缩尺模型

综上所述，为探明不同因素对航行阻力的主效应和交互效应，需对不同船身造型进行实验。然而，若需对上述 13 组方案进行同材料足尺模型试验，其工作量远超备赛期能力范围，因此需考虑缩尺实验方案。

2) 试验原理

要实现有效的结构相似，必须满足材料相似、时间相似、几何相似、动力学相似。

首先，由于模型试验与原型加载均在恒定水温下进行，故水的运动粘度相等，简化得：

$$v_p L_p = v_m L_m$$

定义：1) 几何缩尺比： $\lambda_L = L_m/L_p$ ；2) 速度缩尺比： $\lambda_v = v_m/v_p$ 。由此可推导出： $\lambda_v = 1/\lambda_L$ 。进而，时间缩尺比 λ_t （完成相同距离所需时间之比）为：

$$\lambda_t = \frac{\lambda_L}{\lambda_v} = \lambda_L^2$$

其次，模型与原型吃水深度应满足浮态相似，假设船体质量分布均匀，可简化为棱柱体进行浮力估算：

$$F_{\text{浮}} = \rho g h S = mg$$

其中 h 为吃水深度， S 为水线面面积。由于 $h \propto L, S \propto L^2$ ，有： $\rho g(k_1 L)(k_2 L^2) \propto mg \Rightarrow m \propto L^3$

因此，质量比例尺为：

$$\lambda_m = \lambda_L^3$$

最后，结构系统应动力学相似。加速度比例尺 λ_a 可由运动学关系得出：

$$\lambda_a = \frac{\lambda_L}{\lambda_t^2} = \frac{\lambda_L}{(\lambda_L^2)^2} = \lambda_L^{(-3)}$$

根据牛顿第二定律 $F = ma$ ，因此船身所受阻力的比例 λ_F 为：

$$\lambda_F = 1$$

本次缩尺模型使用 3D 打印技术进行制作，由于打印机尺寸限制，模型长度不能长于 30cm。综上所述，长度比尺取 $\lambda_L = 1:5$ ，此时 $\lambda_t = 1:25$ ， $\lambda_m = 1:25$ ， $\lambda_F = 1$ 。

3) 试验对象与设备

(1) 试验对象

缩尺实验采用 3d 打印工程塑料（ABS）制备缩尺模型方案，由于 3d 打印高效、自动化程度高的特性，且 ABS 轻质高强的特性，可以最大化消除模型自重和内部结构体系对实验结果的影响，因此使用该方案可以快速制备符合要求的缩尺实验模型，并探究船身造型对船体阻力的影响。缩尺试验模型如图 1-14 所示。



图 1-14 基准模型



图 1-15 缩尺模型船

(2) 试验设备

本试验采用一套自主搭建的拖曳式水动力学测试系统，旨在精确测量不同船模在静水中的航行阻力。该装置主要由驱动、传动、支承与测量四大模块构成，如图 1-16 所示，其核心工作原理是通过电机拖曳船模匀速行驶，并实时测量此时牵引绳上的张力，该张力数值即等效于船模所受的总阻力。

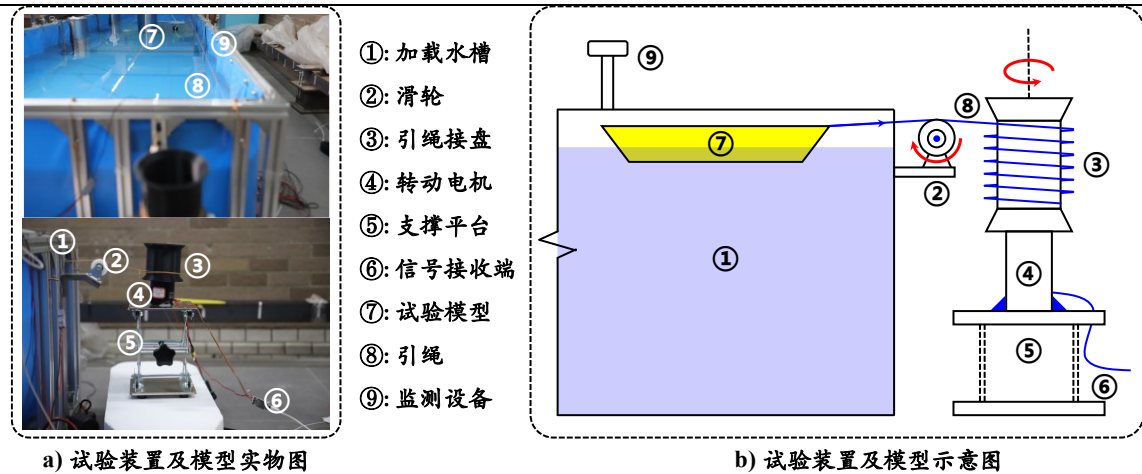


图 1-16 试验装置示意图

i) 驱动与传动系统

驱动源为一台力控 FOC 电机，提供稳定且可调的牵引速度，并通过电流环输出等效力矩。动力通过引绳传递，引绳一端缠绕于电机的引绳接盘上，另一端通过滑轮组件改变方向后，水平连接至船模。

ii) 试验环境与支承

试验在一个透明的加载水槽中进行，以提供可控的静水环境。水槽前方架设有一个支撑平台，用于固定电机、滑轮等整套拖曳设备，确保试验过程的稳定性与可重复性。

iii) 数据采集系统

监测设备安装于水槽外，用于实时监控并采集缩尺模型的运动状态。采集到的数据由信号接收端接收并记录，最终传输至计算机进行后续处理与分析。

4) 试验方案

本试验流程主要包括准备、测试与数据处理三个核心环节。

(1) 试验准备

清理水槽并注入足量清水，静置至水面平稳。安装目标船模，调整至标准浮态，检查牵引系统运转正常。

(2) 试验测试与数据采集

启动数据采集系统，在无负载状态下进行调零。通过控制程序设定拖曳速度，启动电机牵引船模匀速航行。待运动稳定后，采集并记录稳定的阻力数据。每种工况重复测试 3 次以确保结果可靠。

(3) 数据处理

去除船模初始加速航行段与最终减速段，选择平稳行驶的中间段，提取有效数据段，扣除系统零位偏差，并采用卡尔曼滤波算法去除传感器噪声，计算各航次平均阻力。

5) 试验结果

根据本研究的试验观测与数据分析,不同船底构型对模型阻力的影响显著,其结果与理论预期基本一致,如图 1-17 所示。

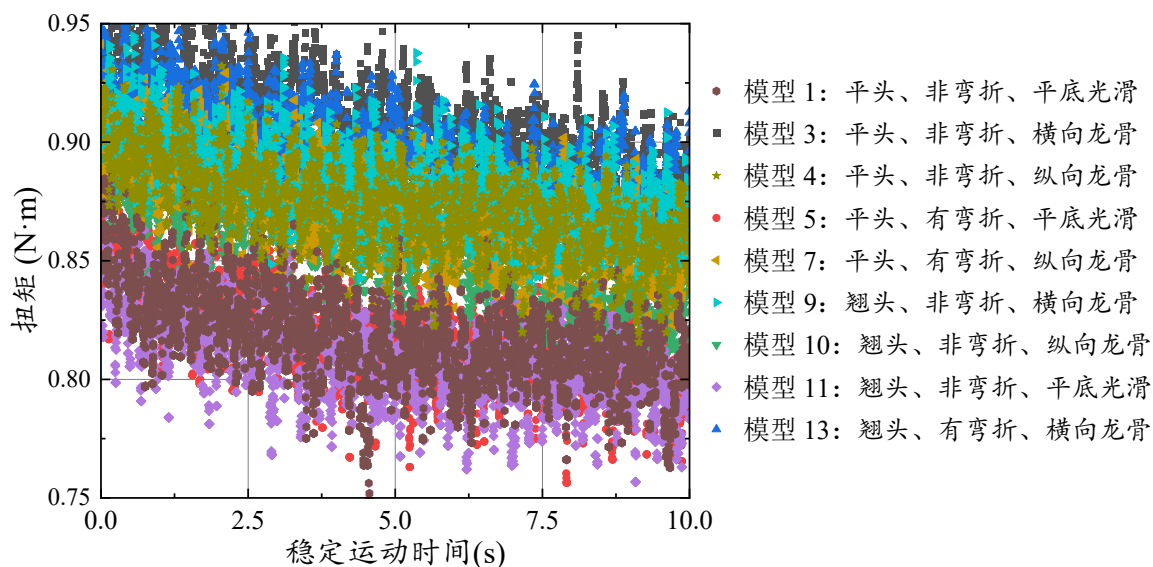


图 1-17 试验结果图

(1) 在全部测试方案中,“上翘船头-无弯折船身-光滑无龙骨船底”展现出最小的航行阻力,。这是由于在低速 ($Fr < 0.2$) 工况下,摩擦阻力占据主导地位,光滑船底拥有最小的湿表面积,从而有效降低了总阻力。

(2) 在加装龙骨的构型中,纵向龙骨模型的表现明显优于横向龙骨模型。横向龙骨因其构件方向与水流方向垂直,对水流边界层的破坏更为严重,产生了更大的形状阻力;而纵向龙骨对水流的干扰相对较小,因此阻力增加也相对较少。

(3) 此外,试验中发现所有外伸龙骨构型均出现倾覆现象,如图 1-18 所示,表明其水动力稳定性不足,在实际应用中不可行。

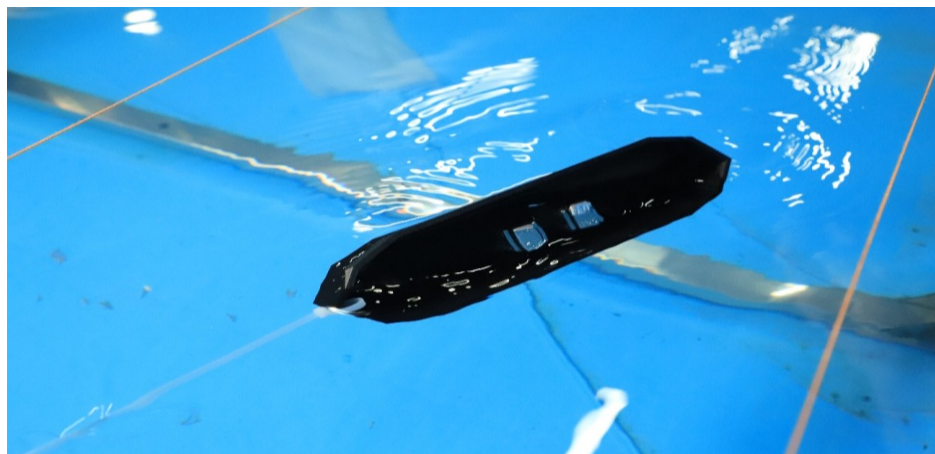


图 1-18 模型船倾覆图

(4) 船身是否弯折对阻力性能的影响在本试验的速度范围内并不显著，这是由于在低速状态下，船体兴起的波浪非常小，兴波阻力在总阻力占比极低。船身是否弯折的核心空气动力学优势在于破浪、减缓兴波以减少湿表面积，但这些效应在高速（高 Fr 数）时才显著。在本试验的低速条件下，这一优势无法得到有效发挥。

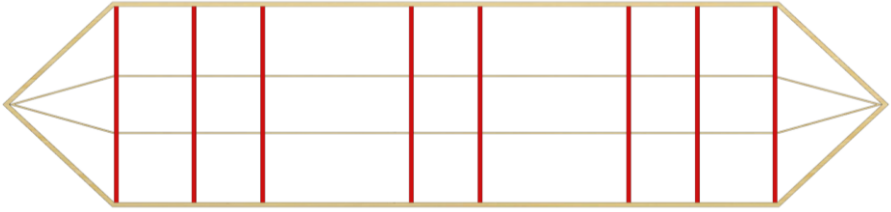
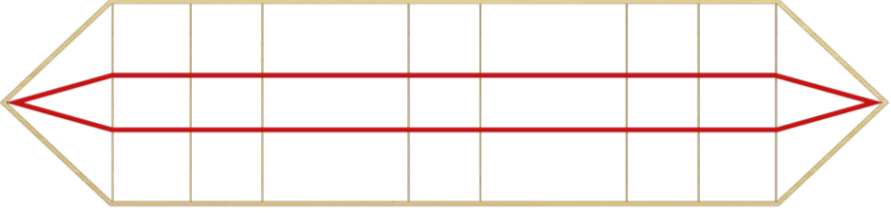
结论：综合阻力性能与稳定性，“上翘船头-无弯折船身-光滑无龙骨船底”为本试验中的最优方案，后续结构方案比选基于此船型展开。

1.4.2 结构方案比选

1) 船底骨架方案比选

船底骨架由纵向龙骨与横向加劲肋共同构成，分别构成纵向和横向刚度。纵向龙骨在提供纵向刚度的同时，还保证船体的整体连续性与航行速度；横向加劲肋则提供横向刚度和抗倾覆能力。为此，我们对两种结构联系方式进行了比选分析。

表 1-1 船底骨架体系对比与优缺点

体系对比	优点	缺点
	■ 主要受力杆件 ■ 次要受力杆件	
纵向刚度小 横向刚度大		横向刚度大;承载能力能满足要求; 船底横向变形, 水阻力大; 一级加 三级造浪工况稳定性好 载航速不足, 难达 T_3 要求
纵向刚度大 横向刚度小		船底纵向变形, 水阻小; 一级、三 模型质量大, 二级加载工况不易满 级加载工况滑行速度较快 足, 三级工况下易受波浪影响。

横向刚度大的体系在承载能力与稳定性上占优，但牺牲了一级航速；纵向刚度大的体系则相反，航速较优却难以保证二、三级的稳定性。结合赛题的评分公式与实际测试结果，我们后续需要在两者之间寻找折中：既保持纵向流线的低阻特性，又通过局部横向加固来提升二、三级的稳定性。从比选结果可见，单龙骨桁架虽承重能力充足，但稳性极差，易在二级和三级工

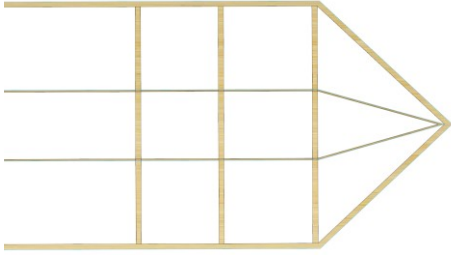
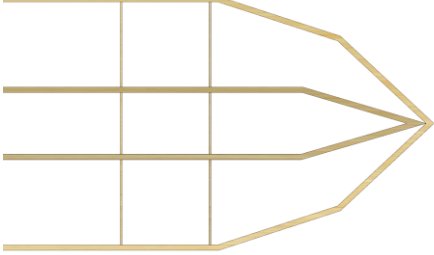
况下倾斜；而纵横单梁叠置骨架在一级工况水压作用下膜变形小，船底面流线顺畅，滑行阻力小，在二、三级工况下稳定性较好。因此，最终采用纵横单梁叠置骨架结构体系。该结构体系既能在一级加载下保证船底纵向流线的连贯性与较高航速，又能在三级满载与造浪撞击中保持稳定，不发生侧翻，在承重、稳性、速度三项指标间实现了最优平衡。

2) 船弦方案比选

在本赛题三级加载环节，模型需在造浪环境下完成航行并在终点处与阻挡墩发生撞击，同时保证船体结构完整稳定不倾覆。由此可见，船弦不仅是船体外形的边界，更直接承担了抗撞击、分散冲击载荷等关键功能。基于这一要求，我们将研究重点放在上船弦的优化设计上：通过改进船弦形状与受力路径，减弱碰撞瞬间的集中冲击，降低船体局部受损风险，并提升整体抗倾覆能力。

为此，我们对不同船弦布置形式进行了比选，力求在抗冲击稳定性、结构轻量化和航速影响之间取得合理平衡。

表 1-2 船弦方案的对比及优缺点

体系对比	双折弦体系	三折弦体系
示意图		
优点	节点数量少；简化受力路径，避免应力集中	折线更平顺，撞击时外力沿弦面滑移分散，过墩过程顺畅，减小正面冲击
缺点	抗撞击能力弱	节点数量多，连接失效风险增加；制作复杂度程度提升；但实际抗撞击效果并不显著

通过对比分析可见，双折弦体系结构简洁、重量较轻，尽管在三级加载的终点撞击中存在船体易受正面冲击、能量吸收与分散能力有限的情况，但综合实际应用场景来看，其性能已能满足基础需求。

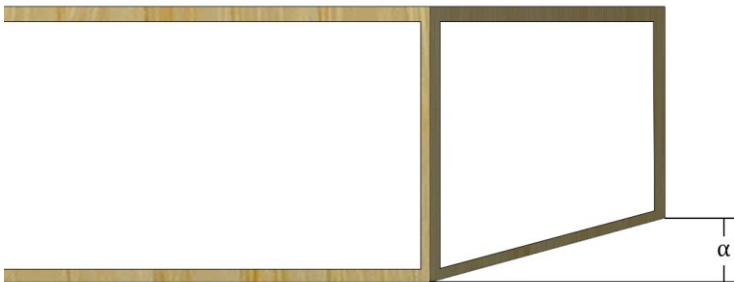
相比之下，三折弦体系虽理论上折线更平顺，可引导外力沿折线方向传递，但在实际试验中，其抗撞击效果并未呈现显著优势，且该体系节点数量较多导致制作难度大幅增加，复杂的连接结构反而带来了新的安全隐患，与设计预期的可靠性存在偏差。

综合考量性能表现、制作难度与安全风险后，最终选择双折舷体系作为本次方案的船舷构型，在保障基础抗撞击能力的同时，兼顾了结构简化、制作可行性与使用安全性。

3) 船头方案比选

根据缩尺实验结果可知，船头上翘将显著降低船体的航行阻力。但由于缩尺实验尺寸限制，无法对上翘角度进行定量分析。因此在结构比选阶段，实验设计了三组足尺不同上翘程度的船型，综合对比航行阻力与稳定性的平衡方案。

表 1-3 船头上翘程度对比及评价

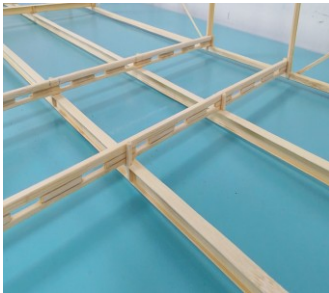
体系对比	$\alpha=15\text{ mm}$	$\alpha=25\text{ mm}$	$\alpha=35\text{ mm}$	
示意图				
	T_2-T_1	15.938 s	15.464 s	12.859 s
	T_3-T_1	29.347 s	28.828 s	22.564 s
现象与综合评价	船头离水适中，航迹稳定		与 15 mm 表现接近	满载后船头下沉几乎贴水，航速明显提升

综上所述，方案选择在空载状态上翘约 35 mm，以保证满载 8kg 时船头恰好贴近水面并保持轻微上翘的造型方案。该设计兼顾了一级加载的速度优势与二、三级加载的通过稳定性。

4) 主要受力杆件截面构型比选

本模型的纵向主承载体系直接影响一级与三级加载阶段的结构强度与稳定性。为在轻量化与承载能力之间取得平衡，我们设计了三种典型杆件截面形式：工字杆、轨道杆和箱形杆。同时，通过 dlubal 软件进行理论计算分析，并结合实体加载试验对三种构型的承载性能进行对比验证，以确定最优受力方案。

表 1-4 主要受力杆件截面构型对比与优缺点

杆件构型	工字杆	轨道杆	箱形杆
示意图			

“建行杯”第十八届全国大学生结构设计竞赛总决赛理论方案

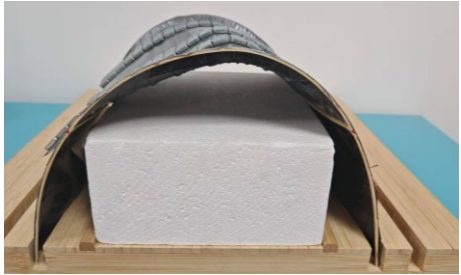
杆件构型	工字杆	轨道杆	箱形杆
理论计算结果	截面惯性矩大，抗扭刚度小	惯性矩最大，约为工字杆的 1.12 倍、箱形杆的 1.35 倍	抗扭刚度最大
实际加载表现	6 kg 承载稳定，破坏时腹板轻微弯折	8kg 承载下变形较小	胶水粘连部分易开裂，表面脆性破坏
优缺点综述	强度高、制作方便，但易失稳导致杆件失效	承载力高、韧性好，但质量较大	抗扭性能优，但易裂开、可靠性差

理论计算与实践结果一致表明：轨道杆的承载与抗弯性能最优，在 8 kg 等效加载下能保持整体稳定，无明显塑性变形；工字杆性能次之，在 6 kg 载荷下表现可靠；方杆最弱，易在角部产生开裂失效。因此，最终确定采用“中部轨道杆 + 两侧工字杆”的组合方案：轨道杆用于船体中央主承载区，承担 8 kg 荷载；工字杆布置于两侧，承担 6 kg 荷载。该布置方案既保证了纵向主梁的整体刚度和强度，又控制了结构重量，为三级加载阶段的抗撞击与稳定性提供了坚实基础。

5) 船篷主要杆件截面构型比选

根据赛题要求,我们的设计目标是在确保能顺利通过加载与规避区 IV 通行要求的前提下,实现结构轻量化与受力均衡。为此，我们采用最简“两竖两横”框架构型，并对中部横杆的截面形式进行比选，通过理论分析与试验验证综合确定最优方案。

表 1-5 船篷主要杆件截面构型对比与优缺点

杆件构型	2×2 竹杆	2×1 竹杆
示意图		
形变图		
优缺点综述	截面刚度较高，加载变形小，节点连接稳固，但质量偏大	在满足强度与刚度要求的同时，质量更轻

通过理论分析与多组加载试验,结果表明两种截面形式均能满足船篷加载的强度要求,但 1×2 mm 横杆在保证足够刚度的同时,质量显著更轻。综上所述,最终确定采用 1×2 mm 截面横杆作为船篷中横构件,兼顾轻量化与加载条件。

1.4.3 加载方案比选

1) 加载参数敏感性与方案比选

本次结构设计竞赛的评分规则与往届存在显著差异,本届竞赛的评分体系由模型质量系数、航行时间系数与载重系数等多项因素共同决定,各参数的设定及相互制约关系将直接影响总成绩。通过对赛题与评分细则的研读可发现,在加载评分体系中,时间性能的权重显著提升,尤其在第一级加载中,滑行时间系数占该级得分的 70%,而质量系数仅占 30%;在第三级加载中,航行时间亦为关键评分因素,时间权重可达 60%。公式中的 m_{min} 表示“通过该级加载的最小模型质量”,而非“满载工况下的最小质量”。这一设定具有重要的策略性含义:单纯追求满载并不必然带来得分优势。选择满载虽能提升承载力指标,但会迫使结构加重以保证通过,从而抬高整体质量,降低质量相关项的优势。

因此,为了量化分析三级加载间的耦合效应并确定最优加载策略,我们对包含多项影响因素的总得分函数进行了偏导计算,以明确各参数对总成绩的敏感性及主导作用,为结构设计与载荷分配提供理论依据。

赛题中的评分规则如下所示,其中, t_1 为一级航行时间, t_3 为三级航行时间, m_3 为三级加载质量, m 为模型总质量:

$$\begin{cases} D_{1i} = (w_{11} + w_{12} + w_{13}) \times 35 \times (0.3 \frac{m_{min}}{m} + 0.7 \frac{t_{1,min}}{t_1}) \\ D_{2i} = 15 \times \frac{m_{min}}{m} \times p_{2i} \\ D_{3i} = 30 \times (q_p \times \frac{m_{min}}{m} \times \frac{m_3}{m_{3,max}} + q_t \times \frac{t_{3,min}}{t_3}) \end{cases}$$

在多次实验中发现,一级加载的变形测试易通过,且 30 s 内航行规定距离容易实现,因而与此相挂钩的 $w_{11} + w_{12} + w_{13}$ 取 1。1.5 kg 的二级载重较易实现并稳定可靠,因此在后续分析中不考虑二级载荷变化的情况,即 p_{2i} 取 1。三级加载航行时间大概率在 30 s 内,因此权重系数 q_p 和 q_t 分别取 0.4 和 0.6。

根据经验和初步计算,可以假设在所有参赛队伍中,一级最快航行时间是 12 s,三级最大载重为 27 kg,最快航行时间是 20 s,加载成功的模型总质量最轻为 70 g。

综上,本队总得分函数可表示为包含变量一级航行时间 t_1 ,三级航行时间 t_3 ,三级载重 m_3 ,

模型总质量 m 的多元函数 $f(t_1, m_3, t_3, m)$:

$$f(t_1, m_3, t_3, m) = 35 \times \left(0.3 \times \frac{70}{m} + 0.7 \times \frac{12}{t_1} \right) + 15 \times \frac{70}{m} \\ + 30 \times \left(0.4 \times \frac{70}{m} \times \frac{m_3}{27} + 0.6 \times \frac{20}{t_3} \right)$$

经对比实验与初步试验，本队模型的合理取值为：一级航行时间 15 s，三级载重 19 kg，航行时间约 25 s，总质量约 90 g。因此，若假设各变量相互独立，我们在点 $(t_1, m_3, t_3, m) = (15, 19, 25, 90)$ 处分别计算出对各独立变量的偏导，以计算各独立变量对得分函数的影响：

$$\frac{\partial f}{\partial t_1} = -1.3067, \quad \frac{\partial f}{\partial t_3} = -0.5760, \quad \frac{\partial f}{\partial m_3} = +0.3457, \quad \frac{\partial f}{\partial m} = -0.2933$$

但是事实上，三级载重 m_3 的增加会引起模型总质量 m 的同步上升，因此我们还需考虑 m 与 m_3 之间的耦合关系。根据我们在试验阶段的测量结果可知，当第三级载重每增加 1 kg，模型总质量约增加 1 g，这一比例关系可近似认为线性，即：

$$f(t_1, m_3, t_3, g(m_3)) = 35 \times \left(0.3 \times \frac{70}{g(m_3)} + 0.7 \times \frac{12}{t_1} \right) + 15 \times \frac{70}{g(m_3)} \\ + 30 \times \left(0.4 \times \frac{70}{g(m_3)} \times \frac{m_3}{27} + 0.6 \times \frac{20}{t_3} \right) \\ m = g(m_3) = m_0 + m_3, \quad \frac{dm}{dm_3} = 1$$

因此， m_3 对得分函数 $f(t_1, m_3, t_3, m)$ 的影响不仅包括载荷系数项，还包含因质量变化间接引起质量系数项。因此，采用链式法则对 m_3 求偏导，从而考虑 m_3 的总体影响量：

$$\frac{df}{dm_3} = \frac{\partial f}{\partial m_3} + \frac{\partial f}{\partial m} \cdot \frac{dm}{dm_3} = \frac{\partial f}{\partial m_3} + \frac{\partial f}{\partial m}$$

代入数值可得：

$$\frac{df}{dm_3} = 0.3457 + (-0.2933) \approx 0.0523$$

因此，在考虑 m 随 m_3 增加带来的质量惩罚后，各因素的敏感性数值为：

$$\frac{\partial f}{\partial t_1} = -1.3067, \quad \frac{\partial f}{\partial t_3} = -0.5760, \quad \frac{df}{dm_3} \approx +0.0523, \quad \frac{\partial f}{\partial m} = -0.2933$$

故，各因素的敏感性排序为：

$$|\partial f / \partial t_1| > |\partial f / \partial t_3| > |df / dm_3| > |\partial f / \partial m|$$

我们可以得出结论：

表 1-6 各因素对总得分的影响

参数	一级航行时间 t_1	三级航行时间 t_3	三级载重 m_3	模型总质量 m
影响	时间越短 得分越高	时间越短 得分越高	增载对提升 分数效果有 限	越轻越好
重要程度	11	2	4	3

在三级加载方案的选取上，赛题允许 6 kg、8 kg 与 10 kg 三档自选载荷，为更直观地表示，我们绘制了三维数据图。经对比实验与初步试验，假定三级航行时间 t_3 为 25 s，且当取 6 kg 作为三级载重时，模型总质量约 90 g 作为参考点，并考虑质量惩罚。此时不同质量选择下，加载得分随一级航行时间 t_1 与三级航行时间 t_3 的关系如图所示。

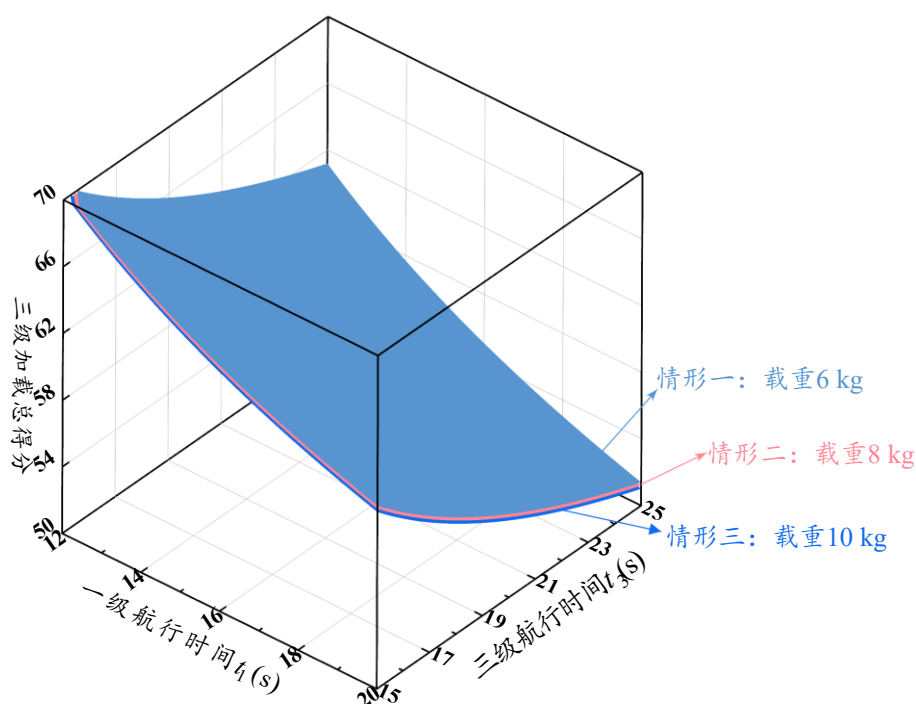


图 1-13 不同质量选择下，加载得分随 t_1 与 t_3 的关系图

综上，我们得出两点关键结论：

一级航行时间 t_1 对总得分的敏感性最高，是决定模型综合表现的关键因素；其次为三级航行时间 t_3 ，其在造浪与障碍物条件下的稳定性与速度表现具有显著影响。相比之下，载重项 m_3 的边际收益有限，且在与模型总质量 m 耦合后，其净增益几乎被质量惩罚项抵消。因此，模型优化的首要目标应放在提升一级航行性能的减阻设计，同时兼顾三级航行阶段的稳定性与速度效率。

特别地，三级航行时间主要取决于参赛队员的配合，经过实训，我们三级航行时间 t_3 一般为 19~20s，当我们固定 $t_3=20$ s 时，加载得分随一级航行时间 t_1 与三级航行时间 t_3 的关系如下图所示。

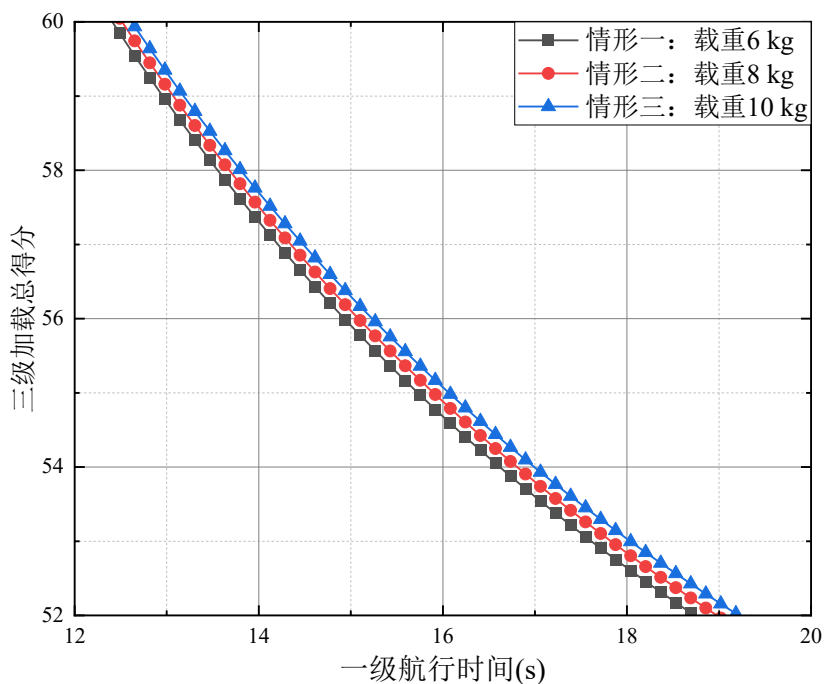


图 1-14 得分随一级航行时间 t_1 的关系

综上所述,通过计算模拟与敏感性对比可知,过大载荷会带来结构加重、刚度提升需求及弯矩增大,反而降低模型的整体得分。基于此,我队最终确定以 6 kg 为第三级主加载方案,在结构效率与得分潜力之间实现平衡。

2) 三级载重放置位置

在本次结构设计竞赛的第三级加载中,参赛队员需在保持船头二级砝码 1.5 kg 载重的基础上,继续于船尾定位膜 III 处施加 1.5 kg 砝码。同时,船舱内布置有 4 个坛状砝码,包括 2 个经比选确定的 6 kg 铁砂及 2 个 2 kg 水桶,布置方式如图 1-15 (a) 所示。

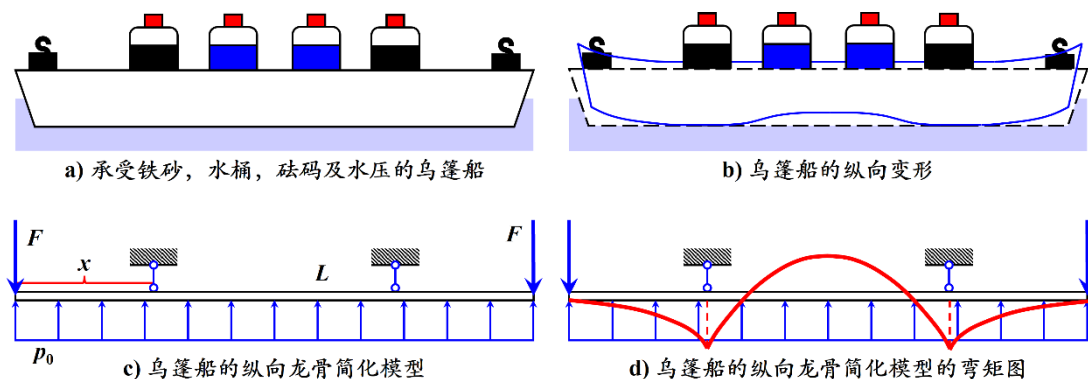


图 1-15 三级加载示意图

结合实训过程与有限元分析结果,纵向龙骨可视为乌篷船中承担铁砂、水桶、砝码及水压等荷载的主要受力构件,其变形曲线如图 1-15 (b) 所示。在受力分析中,水压可作为满跨均布荷载作用于整根纵向龙骨,引起构件的挠曲变形,构成船体底部变形的主要来源;船头与船尾的砝码则可简化为集中荷载,其作用方向与水压相反,从而部分抵消由水压引起的龙骨挠

曲；船舱内两处铁砂质量较大，使得该区段内纵向龙骨的挠度趋近于零，因此可将其视作连续梁模型中的跨中支座；而两处水桶因质量相对较小，且位于跨中区域力臂较短，其对挠曲变形的抵消作用可近似忽略。

据此，纵向龙骨的受力可简化为以下力学模型：一个两端自由的三跨连续梁，承受自由端集中荷载 F 与满跨均布荷载 P_0 的共同作用，如图 1-15 (c) 所示。第一跨和第三跨的长度设为 x ，中间跨的长度为 $L - 2x$ 。

其中自由端集中荷载 F 与满跨均布荷载设为 P_0 ，可由计算得出：

$$\text{a)} \quad F = 1.5\text{kg} \times 10\text{m/s}^2 = 15\text{N}$$

$$\text{b)} \quad P_0 = \frac{(2 \times 1.5 + 2 \times 6 + 2 \times 2)\text{kg} \times 10\text{m/s}^2}{1.2\text{m}} = \frac{380}{3}\text{N/m}$$

进而可求得连续梁模型中的跨中支座处的最大负弯矩为：

$$\text{c)} \quad M_1 = \frac{1}{2}P_0x^2 - Fx = \frac{190}{3}x^2 - 15x (\text{N}\cdot\text{m})$$

根据结构力学中的简支梁法，进而可求得连续梁模型中的中间跨最大正弯矩为：

$$\begin{aligned} M_2 &= \frac{1}{8}P_0(L-2x)^2 - M_1 = \frac{1}{8}P_0(L-2x)^2 - \left(\frac{1}{2}P_0x^2 - Fx\right) \\ &= \frac{1}{8} \times \frac{380}{3} \times (1.2-2x)^2 - \frac{190}{3}x^2 + 15x = \frac{114}{5} - 61x \end{aligned}$$

当连续梁模型中的中间跨最大正弯矩与跨中支座处最大负弯矩的绝对值相等时，可认为弯矩均匀分布，结构受力最合理，即：

$$\begin{aligned} M_1 - M_2 &= F(x) = 0 \\ \left(\frac{190}{3}x^2 - 15x\right) - \left(\frac{114}{5} - 61x\right) &= -\frac{114}{5} + 46x + \frac{190}{3}x^2 = 0 \end{aligned}$$

由二次函数的求根公式，得：

$$\Delta = b^2 - 4ac = 46^2 + 4 \times \frac{190}{3} \times \frac{114}{5} = 7892 > 0$$

$$x_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a} = \frac{-69 \pm 3\sqrt{1973}}{190}$$

由于 $x \in (0, 0.6)$ ，故取：

$$x = x_1 = \frac{-b + \sqrt{\Delta}}{2a} = \frac{-69 + 3\sqrt{1973}}{190} = 0.338\text{m}$$

综上所述,在近似忽略水桶抵消作用的前提下,理论分析表明,当两处铁砂布置于距船头 338 mm 位置时,结构受力最为合理,纵向龙骨弯矩达到最小。

然而,若进一步考虑中间跨水桶的抵消效应及坛状砧码的实际尺寸,中间跨最大弯矩将有所减小。此时,为优化整体受力状态,两处铁砂应适当向船头与船尾方向移动。因此,在本次结构设计竞赛的第三级加载中,参赛队最终将铁砂布置位置确定为距船头 300 mm 处,以综合平衡各类荷载作用的影响。

第二部分：设计计算

2 试验方面

2.1 材料测试

在此次赛题中，主要使用杆件为，对 $6\text{ mm} \times 1\text{ mm}$ 的竹条厚度打薄为 0.40 mm 用于制作模型的上下船弦，采用 $2\text{ mm} \times 6\text{ mm} \times 3\text{ mm}$ 的工字杆制作龙骨和肋骨。在前文对模型的建模计算以及塔柱壁厚的优化中，都需要用到竹材的弹性模量以及抗拉强度，因此，此次材料测试主要测量受拉杆件的抗拉强度。

本次试验采用比赛提供的 $2\text{ mm} \times 2\text{ mm}$ 的竹条作为测试样本，为保证试验的准确性，进行了三次平行试验，计算平均抗拉强度作为后续计算参数。



图 2-1 试验示意图



图 2-2 材料断裂图

表 2-1 中列出了 $2\text{ mm} \times 2\text{ mm}$ 竹条的极限抗拉强度。

表 2-1 竹条的极限抗拉强度

编号	杆件 1	杆件 2	杆件 3
极限抗拉强度	118.5 MPa	123.0 MPa	103.2 MPa

试验分析：测得竹材抗拉强度平均为 114.9 MPa 远大于赛题给定值 (60 MPa)，考虑到安全系数，取竹材抗拉强度为 90 MPa 。

2.2 构件测试

理论分析与系列试验结果表明,在本赛题给定工况下,该模型的主要失效模式为船模龙骨受弯强度破坏。因此,本小节将对模型核心构件的抗弯强度进行分析。为兼顾杆件质量、强度以及连接难易程度,采用了以下几种不同类型的杆件用于船体底部骨架的设计:

表 2-2 受弯杆件截面特性

编号	杆件类型	截面积	惯性矩
1	轨道杆	17 mm ²	67.0 mm ⁴
2	工字杆	7.5 mm ²	45.3 mm ⁴
3	箱型杆	7.8 mm ²	16.2 mm ⁴

如图 2-3 所示,本研究采用三点弯曲试验方法进行测试,对上述构件的极限弯曲强度(σ_{max})进行计算。



图 2-3 三点弯曲试验装置示意图

通过试验后,分别得到了不同试件条的极限破坏荷载如表 2-3 所示。

表 2-3 试件条极限破坏荷载汇总表

编号	构件 1		构件 2		构件 3	
	计算长度 (mm)	破坏载 (N)	计算长度 (mm)	破坏载(N)	计算长度 (mm)	破坏载 (N)
1	260	30.8	260	17	260	11
2	260	29.6	260	17	260	12
3	260	30.4	260	18	260	12
平均值	260	30.6	260	17.3	260	11.7

理想轴心受压构件发生分支点失稳，其临界荷载符合欧拉公式，且是构件承载力的上限值，而实际的构件通常具有一定的几何初始缺陷(包括几何初弯曲、材料不均匀和端部荷载偏心)，构件受压时发生极值点失稳，其临界荷载通常小于欧拉临界荷载。故本章对箱形截面构件的整体稳定性展开深入研究。

试验研究对象为 114 个 0.35mm 与 0.50mm 两种规格竹皮箱形截面受压柱（图 2-4），其实测的几何尺寸如表 2-4、表 2-5 所示；试件编号中“ B ”表示等边箱形截面柱；截面尺寸符号见图 2-5， L 为试件柱长， I 为截面惯性矩， A 为截面面积， i 为截面回转半径， λ 为柱长细比。考虑到结构设计竞赛中制作构件的长度不会太长，故长细比最大取到 100。



图 2-4 等边箱形截面柱实物图

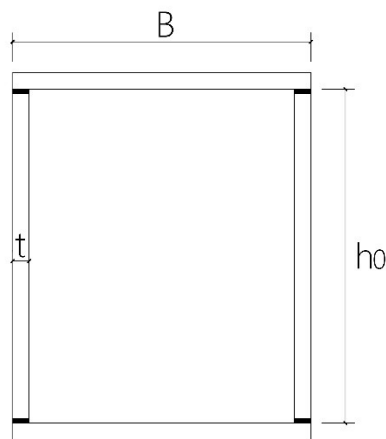


图 2-5 等边箱形截面尺寸符号示意图

表 2-4 0.35mm 规格竹皮试件截面尺寸

(a) 宽厚比为 12.5 的竹皮试件尺寸

编号	t/mm	B/mm	h_0/mm	L/mm	I/mm^4	A/mm^2	i/mm	λ
B12.5-0.35-1	0.4	5	4.2	18.9	26.15	7.36	1.89	10.0
B12.5-0.35-2	0.4	5	4.2	28.3	26.15	7.36	1.89	15.0
B12.5-0.35-3	0.4	5	4.2	37.7	26.15	7.36	1.89	20.0
B12.5-0.35-4	0.4	5	4.2	47.1	26.15	7.36	1.89	25.0
B12.5-0.35-5	0.4	5	4.2	56.6	26.15	7.36	1.89	30.0
B12.5-0.35-6	0.4	5	4.2	66.0	26.15	7.36	1.89	35.0
B12.5-0.35-7	0.4	5	4.2	75.4	26.15	7.36	1.89	40.0
B12.5-0.35-8	0.4	5	4.2	84.8	26.15	7.36	1.89	45.0
B12.5-0.35-9	0.4	5	4.2	94.3	26.15	7.36	1.89	50.0
B12.5-0.35-10	0.4	5	4.2	103.7	26.15	7.36	1.89	55.0
B12.5-0.35-11	0.4	5	4.2	113.1	26.15	7.36	1.89	60.0
B12.5-0.35-12	0.4	5	4.2	122.5	26.15	7.36	1.89	65.0
B12.5-0.35-13	0.4	5	4.2	132.0	26.15	7.36	1.89	70.0
B12.5-0.35-14	0.4	5	4.2	141.4	26.15	7.36	1.89	75.0
B12.5-0.35-15	0.4	5	4.2	150.8	26.15	7.36	1.89	80.0
B12.5-0.35-16	0.4	5	4.2	160.2	26.15	7.36	1.89	85.0
B12.5-0.35-17	0.4	5	4.2	169.7	26.15	7.36	1.89	90.0
B12.5-0.35-18	0.4	5	4.2	179.1	26.15	7.36	1.89	95.0
B12.5-0.35-19	0.4	5	4.2	188.5	26.15	7.36	1.89	100.0

(b) 宽厚比为 20 的竹皮试件尺寸

编号	t/mm	B/mm	h_0/mm	L/mm	I/mm^4	A/mm^2	i/mm	λ
B20-0.35-1	0.4	8	7.2	31.1	117.38	12.16	3.11	10.0
B20-0.35-2	0.4	8	7.2	46.6	117.38	12.16	3.11	15.0
B20-0.35-3	0.4	8	7.2	62.1	117.38	12.16	3.11	20.0
B20-0.35-4	0.4	8	7.2	77.7	117.38	12.16	3.11	25.0
B20-0.35-5	0.4	8	7.2	93.2	117.38	12.16	3.11	30.0
B20-0.35-6	0.4	8	7.2	108.7	117.38	12.16	3.11	35.0
B20-0.35-7	0.4	8	7.2	124.3	117.38	12.16	3.11	40.0
B20-0.35-8	0.4	8	7.2	139.8	117.38	12.16	3.11	45.0
B20-0.35-9	0.4	8	7.2	155.3	117.38	12.16	3.11	50.0
B20-0.35-10	0.4	8	7.2	170.9	117.38	12.16	3.11	55.0
B20-0.35-11	0.4	8	7.2	186.4	117.38	12.16	3.11	60.0

“建行杯”第十八届全国大学生结构设计竞赛总决赛理论方案

编号	t/mm	B/mm	h_0/mm	L/mm	I/mm^4	A/mm^2	i/mm	λ
B20-0.35-12	0.4	8	7.2	202.0	117.38	12.16	3.11	65.0
B20-0.35-13	0.4	8	7.2	217.5	117.38	12.16	3.11	70.0
B20-0.35-14	0.4	8	7.2	233.0	117.38	12.16	3.11	75.0
B20-0.35-15	0.4	8	7.2	248.6	117.38	12.16	3.11	80.0
B20-0.35-16	0.4	8	7.2	264.1	117.38	12.16	3.11	85.0
B20-0.35-17	0.4	8	7.2	279.6	117.38	12.16	3.11	90.0
B20-0.35-18	0.4	8	7.2	295.2	117.38	12.16	3.11	95.0
B20-0.35-19	0.4	8	7.2	310.7	117.38	12.16	3.11	100.0

(c) 宽厚比为 30 的竹皮试件尺寸

编号	t/mm	B/mm	h_0/mm	L/mm	I/mm^4	A/mm^2	i/mm	λ
B30-0.35-1	0.4	12	11.2	47.4	416.73	18.56	4.74	10.0
B30-0.35-2	0.4	12	11.2	71.1	416.73	18.56	4.74	15.0
B30-0.35-3	0.4	12	11.2	94.8	416.73	18.56	4.74	20.0
B30-0.35-4	0.4	12	11.2	118.5	416.73	18.56	4.74	25.0
B30-0.35-5	0.4	12	11.2	142.2	416.73	18.56	4.74	30.0
B30-0.35-6	0.4	12	11.2	165.8	416.73	18.56	4.74	35.0
B30-0.35-7	0.4	12	11.2	189.5	416.73	18.56	4.74	40.0
B30-0.35-8	0.4	12	11.2	213.2	416.73	18.56	4.74	45.0
B30-0.35-9	0.4	12	11.2	236.9	416.73	18.56	4.74	50.0
B30-0.35-10	0.4	12	11.2	260.6	416.73	18.56	4.74	55.0
B30-0.35-11	0.4	12	11.2	284.3	416.73	18.56	4.74	60.0
B30-0.35-12	0.4	12	11.2	308.0	416.73	18.56	4.74	65.0
B30-0.35-13	0.4	12	11.2	331.7	416.73	18.56	4.74	70.0
B30-0.35-14	0.4	12	11.2	355.4	416.73	18.56	4.74	75.0
B30-0.35-15	0.4	12	11.2	379.1	416.73	18.56	4.74	80.0
B30-0.35-16	0.4	12	11.2	402.8	416.73	18.56	4.74	85.0
B30-0.35-17	0.4	12	11.2	426.5	416.73	18.56	4.74	90.0
B30-0.35-18	0.4	12	11.2	450.2	416.73	18.56	4.74	95.0
B30-0.35-19	0.4	12	11.2	473.8	416.73	18.56	4.74	100.0

表 2-5 0.5mm 规格竹皮试件截面尺寸

(a) 宽厚比为 10 的竹皮试件尺寸

编号	t/mm	B/mm	h_0/mm	L/mm	I/mm^4	A/mm^2	i/mm	λ
B10-0.5-1	0.5	5	4	18.5	30.75	9	1.85	10.0
B10-0.5-2	0.5	5	4	27.7	30.75	9	1.85	15.0
B10-0.5-3	0.5	5	4	37.0	30.75	9	1.85	20.0
B10-0.5-4	0.5	5	4	46.2	30.75	9	1.85	25.0
B10-0.5-5	0.5	5	4	55.5	30.75	9	1.85	30.0
B10-0.5-6	0.5	5	4	64.7	30.75	9	1.85	35.0
B10-0.5-7	0.5	5	4	73.9	30.75	9	1.85	40.0
B10-0.5-8	0.5	5	4	83.2	30.75	9	1.85	45.0
B10-0.5-9	0.5	5	4	92.4	30.75	9	1.85	50.0
B10-0.5-10	0.5	5	4	101.7	30.75	9	1.85	55.0
B10-0.5-11	0.5	5	4	110.9	30.75	9	1.85	60.0
B10-0.5-12	0.5	5	4	120.1	30.75	9	1.85	65.0
B10-0.5-13	0.5	5	4	129.4	30.75	9	1.85	70.0
B10-0.5-14	0.5	5	4	138.6	30.75	9	1.85	75.0
B10-0.5-15	0.5	5	4	147.9	30.75	9	1.85	80.0
B10-0.5-16	0.5	5	4	157.1	30.75	9	1.85	85.0
B10-0.5-17	0.5	5	4	166.4	30.75	9	1.85	90.0
B10-0.5-18	0.5	5	4	175.6	30.75	9	1.85	95.0
B10-0.5-19	0.5	5	4	184.8	30.75	9	1.85	100.0

(b) 宽厚比为 16 的竹皮试件尺寸

编号	t/mm	B/mm	h_0/mm	L/mm	I/mm^4	A/mm^2	i/mm	λ
B16-0.5-1	0.5	8	7	30.7	141.25	15	3.07	10.0
B16-0.5-2	0.5	8	7	46.0	141.25	15	3.07	15.0
B16-0.5-3	0.5	8	7	61.4	141.25	15	3.07	20.0
B16-0.5-4	0.5	8	7	76.7	141.25	15	3.07	25.0
B16-0.5-5	0.5	8	7	92.1	141.25	15	3.07	30.0
B16-0.5-6	0.5	8	7	107.4	141.25	15	3.07	35.0
B16-0.5-7	0.5	8	7	122.7	141.25	15	3.07	40.0
B16-0.5-8	0.5	8	7	138.1	141.25	15	3.07	45.0
B16-0.5-9	0.5	8	7	153.4	141.25	15	3.07	50.0
B16-0.5-10	0.5	8	7	168.8	141.25	15	3.07	55.0
B16-0.5-11	0.5	8	7	184.1	141.25	15	3.07	60.0

“建行杯”第十八届全国大学生结构设计竞赛总决赛理论方案

编号	t/mm	B/mm	h_0/mm	L/mm	I/mm^4	A/mm^2	i/mm	λ
B16-0.5-12	0.5	8	7	199.5	141.25	15	3.07	65.0
B16-0.5-13	0.5	8	7	214.8	141.25	15	3.07	70.0
B16-0.5-14	0.5	8	7	230.1	141.25	15	3.07	75.0
B16-0.5-15	0.5	8	7	245.5	141.25	15	3.07	80.0
B16-0.5-16	0.5	8	7	260.8	141.25	15	3.07	85.0
B16-0.5-17	0.5	8	7	276.2	141.25	15	3.07	90.0
B16-0.5-18	0.5	8	7	291.5	141.25	15	3.07	95.0
B16-0.5-19	0.5	8	7	306.9	141.25	15	3.07	100.0

(c) 宽厚比为 24 的竹皮试件尺寸

编号	t/mm	B/mm	h_0/mm	L/mm	I/mm^4	A/mm^2	i/mm	λ
B24-0.5-1	0.5	12	11	47.0	507.92	23	4.70	10.0
B24-0.5-2	0.5	12	11	70.5	507.92	23	4.70	15.0
B24-0.5-3	0.5	12	11	94.0	507.92	23	4.70	20.0
B24-0.5-4	0.5	12	11	117.5	507.92	23	4.70	25.0
B24-0.5-5	0.5	12	11	141.0	507.92	23	4.70	30.0
B24-0.5-6	0.5	12	11	164.5	507.92	23	4.70	35.0
B24-0.5-7	0.5	12	11	188.0	507.92	23	4.70	40.0
B24-0.5-8	0.5	12	11	211.5	507.92	23	4.70	45.0
B24-0.5-9	0.5	12	11	235.0	507.92	23	4.70	50.0
B24-0.5-10	0.5	12	11	258.5	507.92	23	4.70	55.0
B24-0.5-11	0.5	12	11	282.0	507.92	23	4.70	60.0
B24-0.5-12	0.5	12	11	305.5	507.92	23	4.70	65.0
B24-0.5-13	0.5	12	11	329.0	507.92	23	4.70	70.0
B24-0.5-14	0.5	12	11	352.4	507.92	23	4.70	75.0
B24-0.5-15	0.5	12	11	375.9	507.92	23	4.70	80.0
B24-0.5-16	0.5	12	11	399.4	507.92	23	4.70	85.0
B24-0.5-17	0.5	12	11	422.9	507.92	23	4.70	90.0
B24-0.5-18	0.5	12	11	446.4	507.92	23	4.70	95.0
B24-0.5-19	0.5	12	11	469.9	507.92	23	4.70	100.0

试验的加载装置如图 2-6 所示,采用 FBS-5000N 微机控制电子万能试验机进行竖向加载。

试件两端为两个加载圆盘,由于试件端截面较小,圆盘对其转动约束较小,因此可近似看做两端铰接。

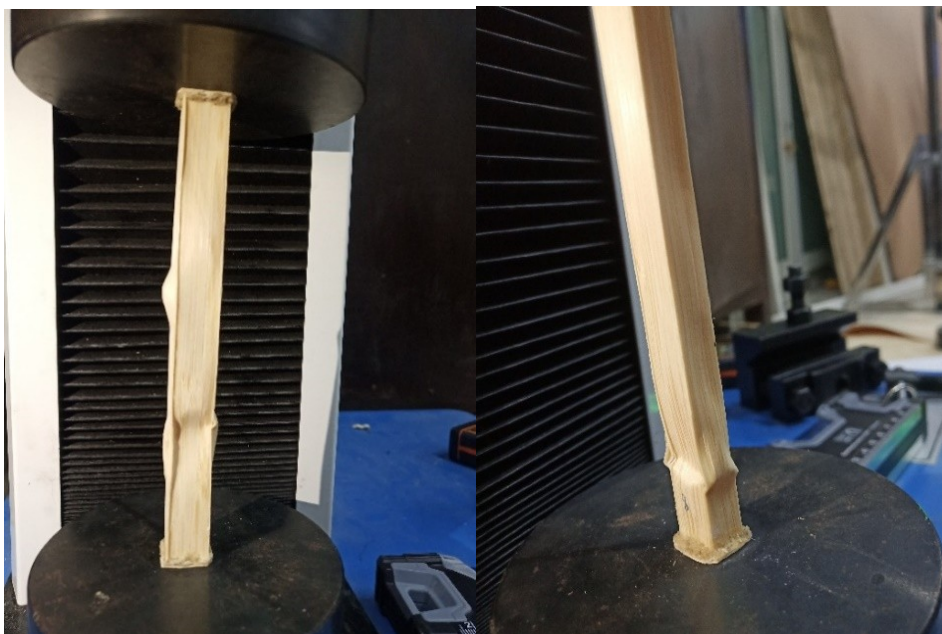


图 2-6 竹皮轴压箱形截面柱整体稳定试验装置

测量设备包括试验机配套的力传感器、位移测量系统、变形测量系统、位移计等。位移计为量程 25.4mm 的数显百分表，布置于试件跨中，测量试件失稳平面内的水平位移，试验装置自带位移测量系统可直接测量试件的竖向变形，即柱顶端千斤顶的加载点位移。

本试验以每分钟 1mm 的速度进行位移加载，试验机自行采集荷载峰值和位移值，并绘制荷载—位移曲线。当试验力值下降超过峰值的 50% 时停止加载。

试件典型的失稳破坏形态如图 2-7 所示。构件在受荷时同时存在局部稳定和整体稳定问题，但长细比小于临界长细比的试件，以局部失稳破坏为主，如图 2-7 (a) 所示。发生局部失稳时，试件局部呈波浪状，各截面均达到局部屈曲临界荷载。在加载初期荷载位移呈线性变化，随着荷载逐渐增大，组成构件的板件开始发生局部屈曲，板件从上至下呈波浪形，初始波幅较小，此时已有部分板件退出工作，但构件仍具有一定的屈曲后强度，随着荷载继续增大，板件波幅也继续增大，最终由于波幅过大引起试件破坏。长细比大于临界长细比的试件，以整体失稳破坏为主，如图 2-7 (b) 所示。试件一经加载即产生挠曲，而挠度 v 不是随着压力 N 按比例增加的，开始挠度增加慢，随后增加较快，如图 2-8、图 2-9 所示。当挠度增加到一定程度时，杆件中点截面在轴力 N 与弯矩 $N \cdot v$ 作用下，受压侧板件应力达到局部屈曲临界应力，此时，板件开始发生凹凸，即产生局部屈曲，当达到局部屈曲极限应力时，试件破坏，竖向承载力开始下降。



(a) 局部失稳破坏



(b) 整体失稳破坏

图 2-7 竹皮轴压箱形截面柱典型失稳破坏形态

图 2-8、图 2-9 和图 2-10 分别为试件 B16-0.5-18 的整体失稳荷载—竖向位移曲线、整体失稳跨中位移—时间曲线和整体失稳荷载—横向位移曲线。

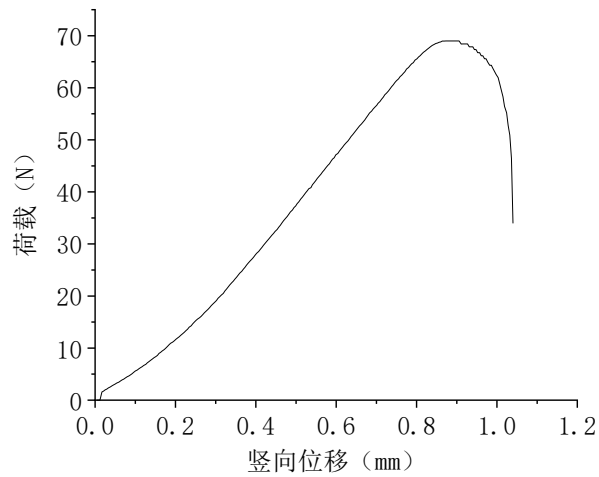


图 2-8 竹皮轴压箱形截面柱典型的荷载—竖向位移曲线

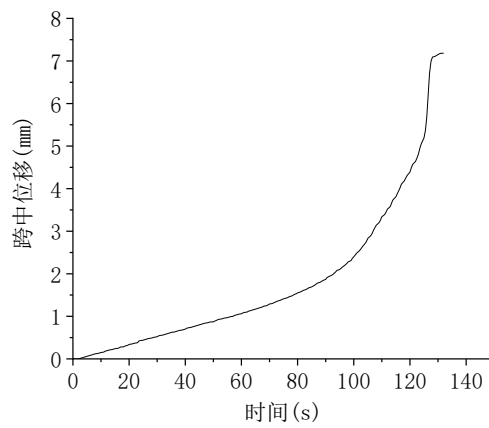


图 2-9 竹皮轴压箱形截面柱典型的跨中位移—时间曲线

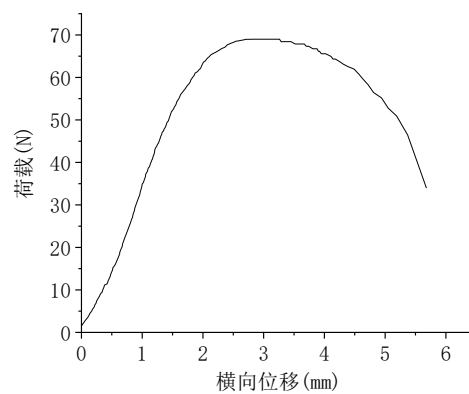


图 2-10 竹皮轴压箱形截面柱典型的荷载—横向位移曲线

表 2-6、表 2-7 汇总了所有试件的极限承载力 N （即轴向荷载的最大值）。

表 2-6 0.35mm 竹皮箱形截面柱极限承载力

编号	P_u / N	编号	P_u / N	编号	P_u / N
B12.5-0.35-1	95.44	B20-0.35-1	101.49	B30-0.35-1	130.22
B12.5-0.35-2	91.19	B20-0.35-2	118.21	B30-0.35-2	120.39
B12.5-0.35-3	86.76	B20-0.35-3	70.46	B30-0.35-3	121.44
B12.5-0.35-4	100.21	B20-0.35-4	113.44	B30-0.35-4	119.30
B12.5-0.35-5	103.82	B20-0.35-5	120.35	B30-0.35-5	97.30
B12.5-0.35-6	94.04	B20-0.35-6	104.46	B30-0.35-6	85.86
B12.5-0.35-7	78.91	B20-0.35-7	110.60	B30-0.35-7	85.72
B12.5-0.35-8	79.53	B20-0.35-8	84.66	B30-0.35-8	68.11
B12.5-0.35-9	80.75	B20-0.35-9	80.50	B30-0.35-9	51.22
B12.5-0.35-10	40.05	B20-0.35-10	96.53	B30-0.35-10	59.13
B12.5-0.35-11	69.29	B20-0.35-11	123.27	B30-0.35-11	60.48
B12.5-0.35-12	62.46	B20-0.35-12	66.59	B30-0.35-12	36.80
B12.5-0.35-13	78.08	B20-0.35-13	48.47	B30-0.35-13	53.45
B12.5-0.35-14	44.83	B20-0.35-14	31.90	B30-0.35-14	38.82
B12.5-0.35-15	31.08	B20-0.35-15	39.25	B30-0.35-15	15.11
B12.5-0.35-16	38.41	B20-0.35-16	77.78	B30-0.35-16	66.72
B12.5-0.35-17	41.14	B20-0.35-17	40.18	B30-0.35-17	57.01
B12.5-0.35-18	40.81	B20-0.35-18	40.29	B30-0.35-18	63.09
B12.5-0.35-19	35.21	B20-0.35-19	36.20	B30-0.35-19	36.20

表 2-7 0.5mm 竹皮箱形截面柱极限承载力

编号	P_u / N	编号	P_u / N	编号	P_u / N
B10-0.5-1	124.23	B16-0.5-1	211.93	B24-0.5-1	142.81
B10-0.5-2	178.05	B16-0.5-2	181.38	B24-0.5-2	136.01
B10-0.5-3	182.03	B16-0.5-3	170.03	B24-0.5-3	140.36
B10-0.5-4	164.22	B16-0.5-4	135.72	B24-0.5-4	98.96
B10-0.5-5	143.88	B16-0.5-5	158.44	B24-0.5-5	162.33
B10-0.5-6	155.30	B16-0.5-6	165.08	B24-0.5-6	141.33
B10-0.5-7	177.98	B16-0.5-7	188.55	B24-0.5-7	174.35
B10-0.5-8	163.31	B16-0.5-8	164.43	B24-0.5-8	126.81
B10-0.5-9	148.32	B16-0.5-9	173.72	B24-0.5-9	228.43
B10-0.5-10	64.13	B16-0.5-10	185.64	B24-0.5-10	104.47
B10-0.5-11	88.58	B16-0.5-11	130.38	B24-0.5-11	153.77
B10-0.5-12	111.24	B16-0.5-12	193.73	B24-0.5-12	102.8
B10-0.5-13	100.22	B16-0.5-13	82.71	B24-0.5-13	72.66
B10-0.5-14	85.64	B16-0.5-14	99.93	B24-0.5-14	138.07
B10-0.5-15	93.02	B16-0.5-15	97.54	B24-0.5-15	131.43
B10-0.5-16	69.90	B16-0.5-16	86.28	B24-0.5-16	59.37
B10-0.5-17	55.95	B16-0.5-17	69.11	B24-0.5-17	99.81
B10-0.5-18	41.02	B16-0.5-18	69.00	B24-0.5-18	141.44
B10-0.5-19	33.21	B16-0.5-19	62.30	B24-0.5-19	62.30

基于 114 个竹皮箱形截面柱进行了轴心受压试验，研究了 0.35mm 与 0.5mm 两种厚度规格竹皮制作的等边箱形截面柱在不同宽厚比下的整体稳定性能。基于试验结果，分析了竹皮箱形截面柱的失稳破坏形态。试验结果表明：箱形截面构件在受荷时同时存在局部失稳与整体弯曲失稳，其中长细比小于临界长细比的构件在受压时以局部失稳为主，长细比大于临界长细比的构件以整体失稳为主。

2.3 结构测试

本模型最终采用纵向联系体系，以主龙骨贯穿船体，结合横向肋骨连接，形成稳定的纵横协同受力结构。船舷采用两折舷设计，兼顾抗撞击性能与制造简便性。主承载体系由中部轨道杆与两侧工字杆组成，能够有效分散集中荷载并提升整体刚度。在一级加载中，中部轨道杆承担 8 kg 荷载；在三级加载阶段，模型采用中部两个 2 kg 与两端各 6 kg 沙桶的加载布局，工字杆承担 6 kg 荷载。模型实测总质量为 98.3 g，整体浮心居中，重心略低于中性面，稳定性良好。



图 2-11 模型实物图

经过多轮加载测试，本模型能够稳定通过赛题的三级加载要求，整体表现优良。

在第一级加载测试中，模型中部加载 8 kg 荷载后，滑行状态平稳，模型在起滑阶段加速度稳定上升，滑行时间约为 13 s，偶然性较大。轨道杆在承受集中荷载时表现出良好的抗弯刚度和稳定性，未出现明显挠曲或局部失稳现象。放置 8 kg 荷载时，模型上弦杆受压较为显著，出现横向内挠变形，局部杆件发生轻微压曲趋势，但这一趋势与理论计算结果相吻合，整体仍维持结构稳定，未出现失稳破坏。

在第二级加载测试中，模型在单侧加砝码的偏心荷载工况下进行了倾斜稳定性验证。测试时偏载质量为 1.5 kg，静态加载后测得模型最大吃水深度为 45 mm，船尾上翘离水面约 160 mm，于安全范围内，模型整体抗倾覆性能满足赛题要求。

在第三级加载测试中，模型加载中部 2×2 kg 坛状砝码与两端各 6 kg 铁砂、1.5 kg 砝码，总载重 19 kg。模型在航行末端撞击障碍物后，整体结构完整性良好。撞击瞬间，双折舷面有

效引导冲击力沿外壳分散滑移，避免了正面冲击的能量集中。加载数据表明，撞击过程中的最大加速度约为 2.4 m/s^2 ，模型无明显横向失稳。

经过对比分析发现，刚度与滑行性能之间存在一定的耦合关系。一级加载阶段，纵向刚度较大的结构有助于保持底部流线连续、提升滑行速度；而在三级加载中，过高的整体刚度则可能导致撞击时能量吸收不足、局部应力集中。因此，在本模型中，采取适度释放横向约束、保持纵向贯通的设计，使其在承载与稳定性之间取得平衡。

综上所述，通过理论计算（Dlubal 仿真分析）与多轮加载实测结果对比，本模型在承载、稳定与抗撞击性能方面均达到了赛题要求。纵向主承载体系及两折舷外形设计的合理性得到验证。

3 结构建模及主要参数

3.1 模型构造

利用 Auto CAD 建立三维空间线条模型, 转化为 DXF 文件后, 导入有限元分析软件 Dlubal RFEM6 建立数值计算分析模型, 结构骨架采用梁单元, 拉条采用仅受拉桁架单元, 最外层防水膜采用壳单元, 建立的有限元模型如图 3-1、图 3-2 所示:

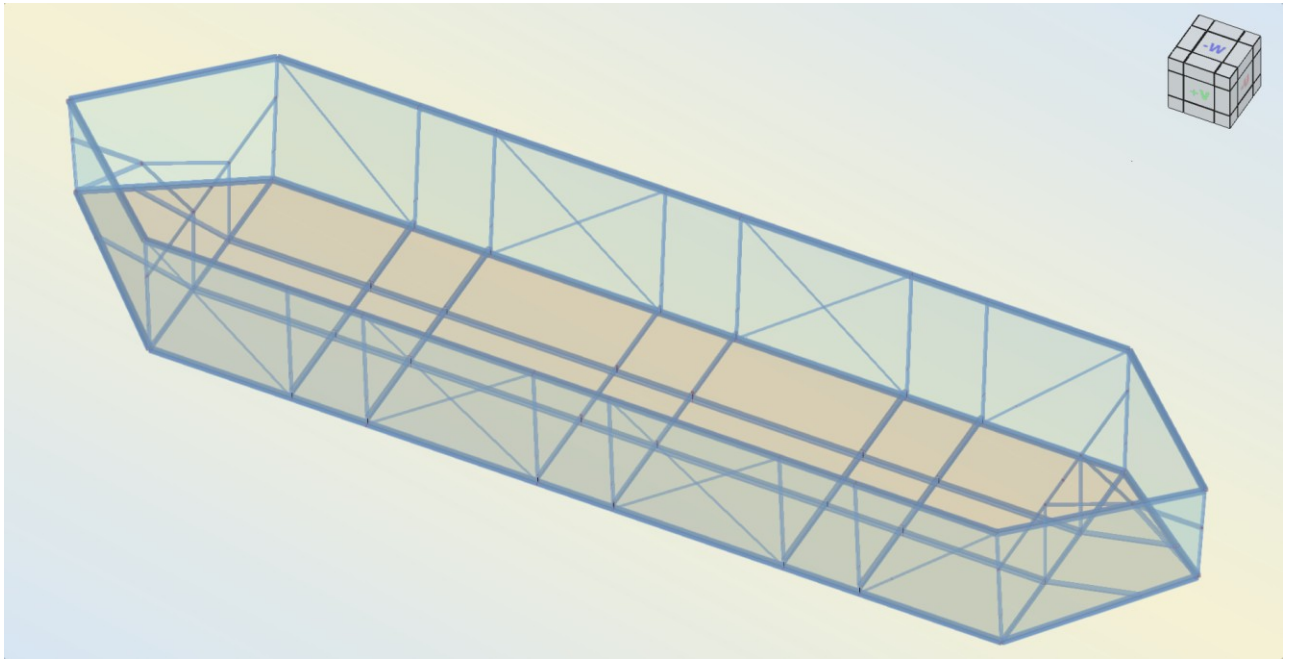
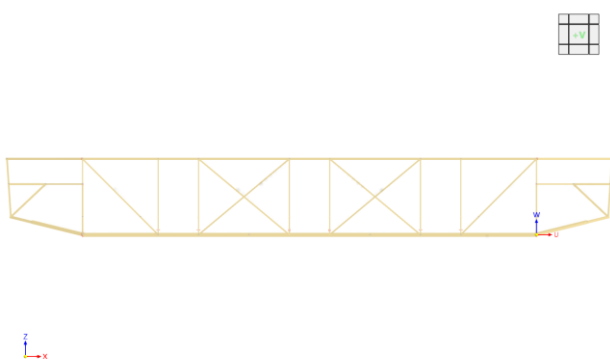
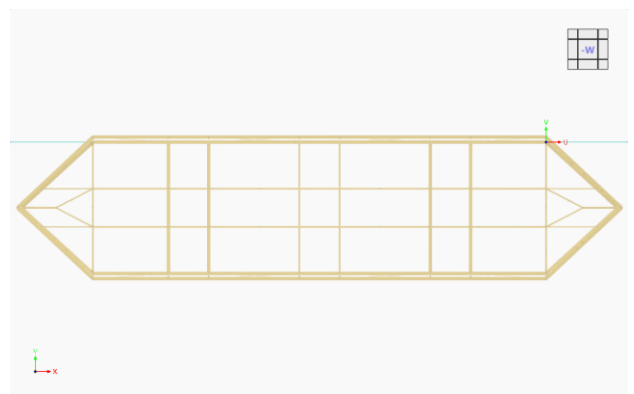


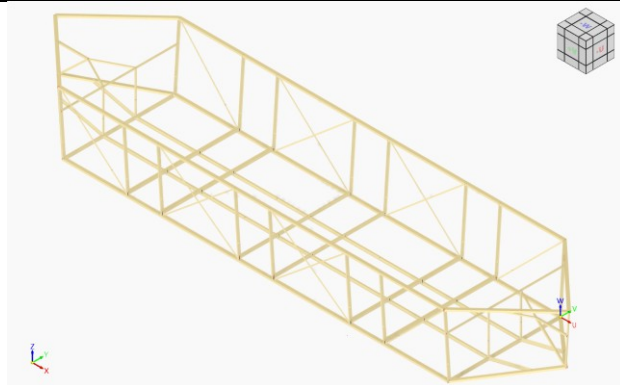
图 3-1 有限元模型透视图 (含防水膜)



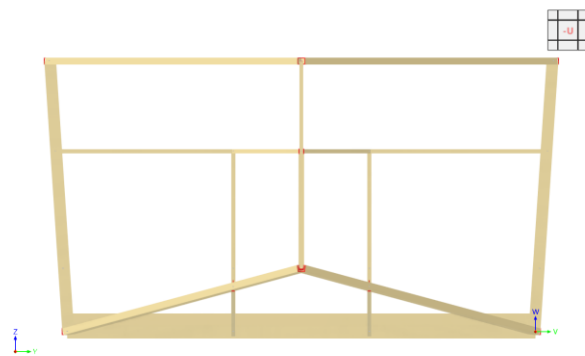
(a) 正视图



(b) 俯视图



(c)正等轴测图



(d)左视图

图 3-2 船体骨架计算模型图

3.2 建模参数

3.2.1 材料部分

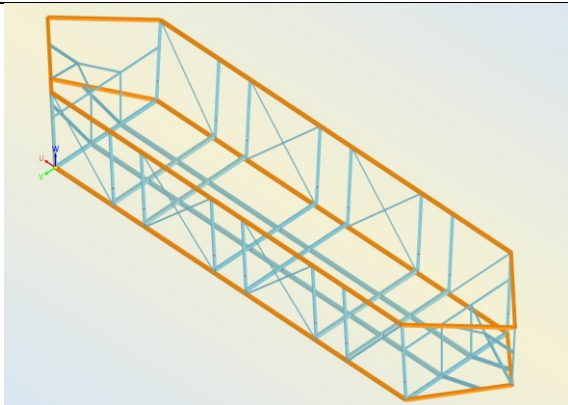
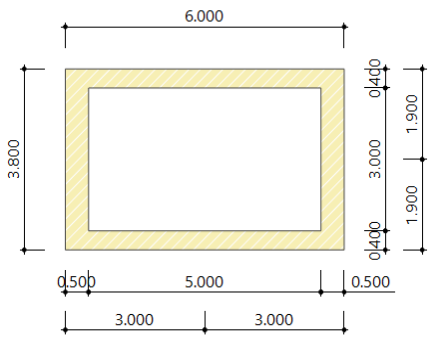
建模过程中所涉及的材料主要为集成竹材与防水膜，其中集成竹材为竹结构模型的主要材料。防水膜则附于结构外侧，使得结构得以在水中航行。材料的主要参数均为赛题提供的参考值。

竹材弹性模量 9.00 GPa，泊松比 0.30，剪切模量 3.46 GPa，容重 8.0 kN/m³；防水膜弹性模量 0.30 GPa，泊松比 0.40 GPa，剪切模量 0.11 GPa，容重 9.5 kN/m³

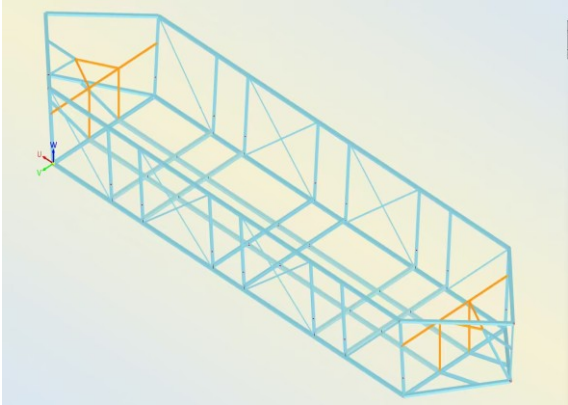
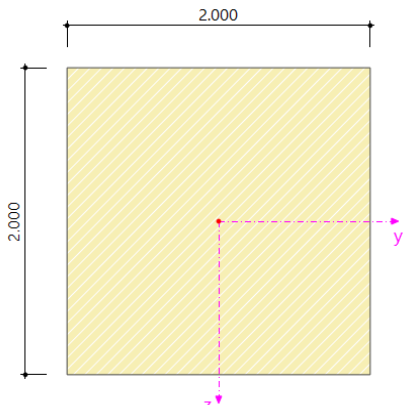
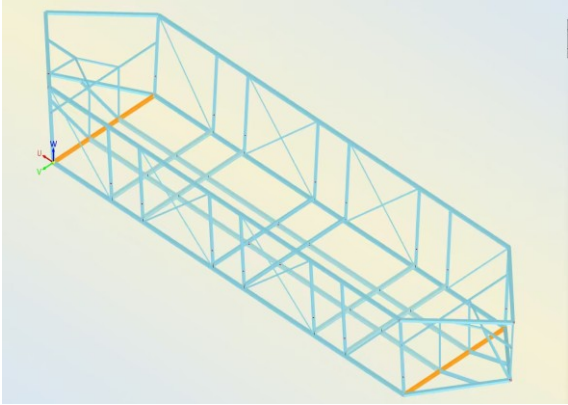
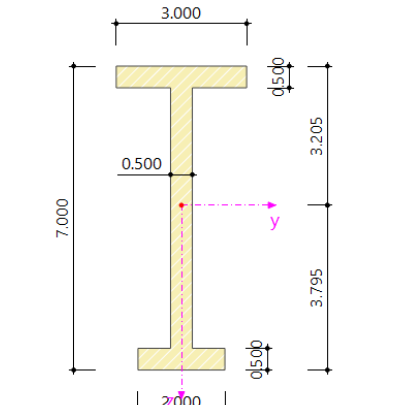
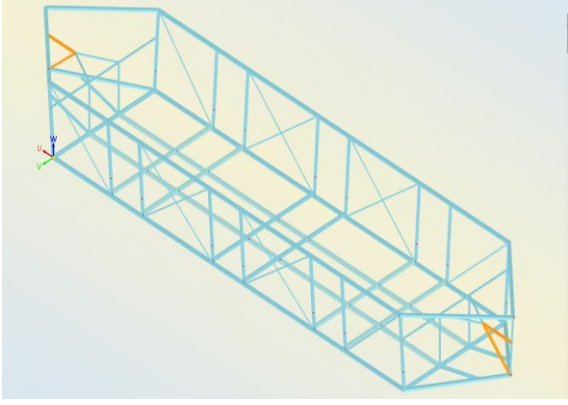
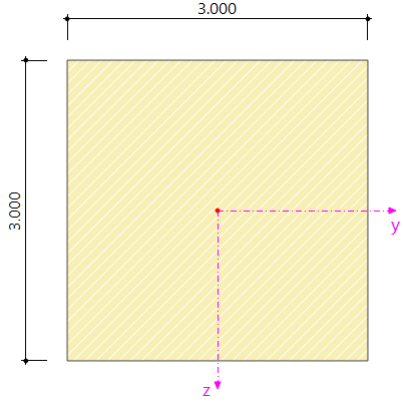
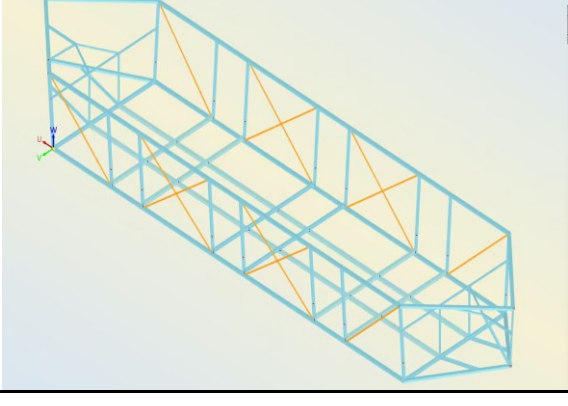
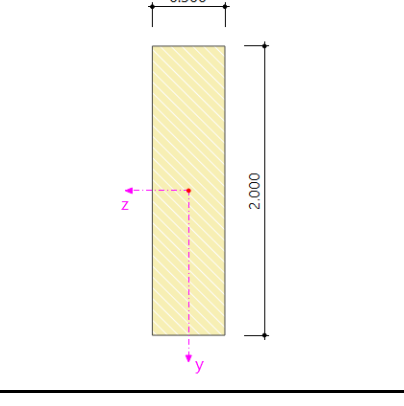
3.2.2 几何信息部分

各构件截面及尺寸按实际情况输入，表 3-1 中列出了所用构件的位置及截面尺寸。

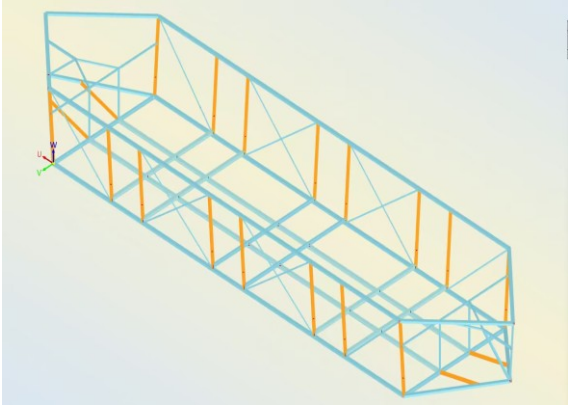
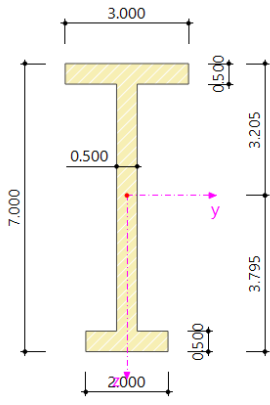
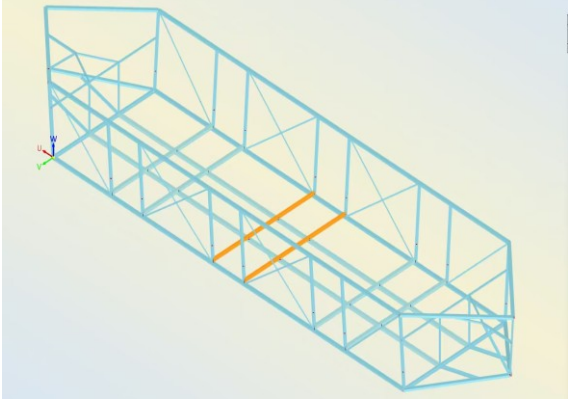
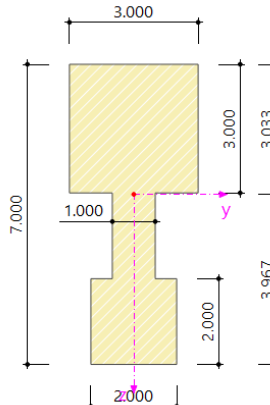
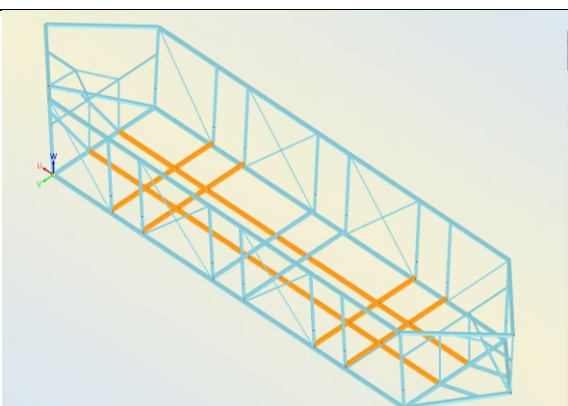
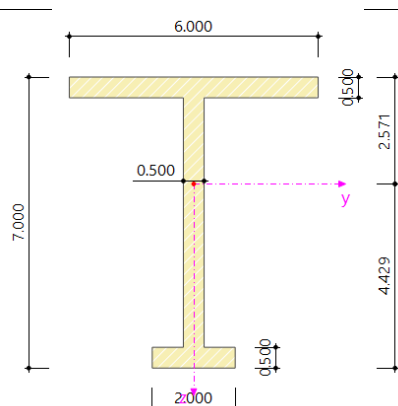
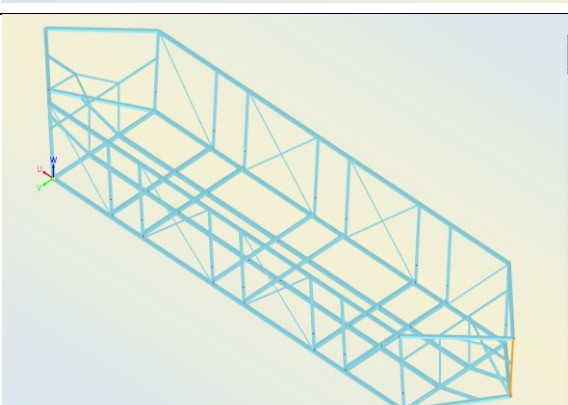
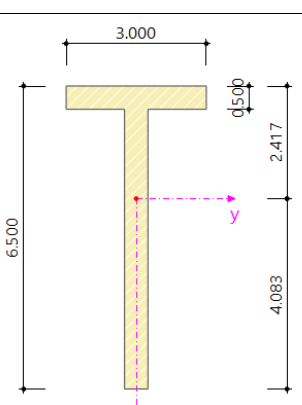
表 3-1 杆件位置及截面

编号	构件位置	截面形状	数量
L1			12

“建行杯”第十八届全国大学生结构设计竞赛总决赛理论方案

编号	构件位置	截面形状	数量
L2			10
L3			2
L4			4
L5			12

“建行杯”第十八届全国大学生结构设计竞赛总决赛理论方案

编号	构件位置	截面形状	数量
L6			20
L7			2
L8			6
L9			2

为真实模拟结构构件受力情况，结构单元分为梁单元（未定义一般单元）和拉杆单元（仅能承受拉力的单元），下图中橙黄色部分为拉杆单元，其余为梁单元。

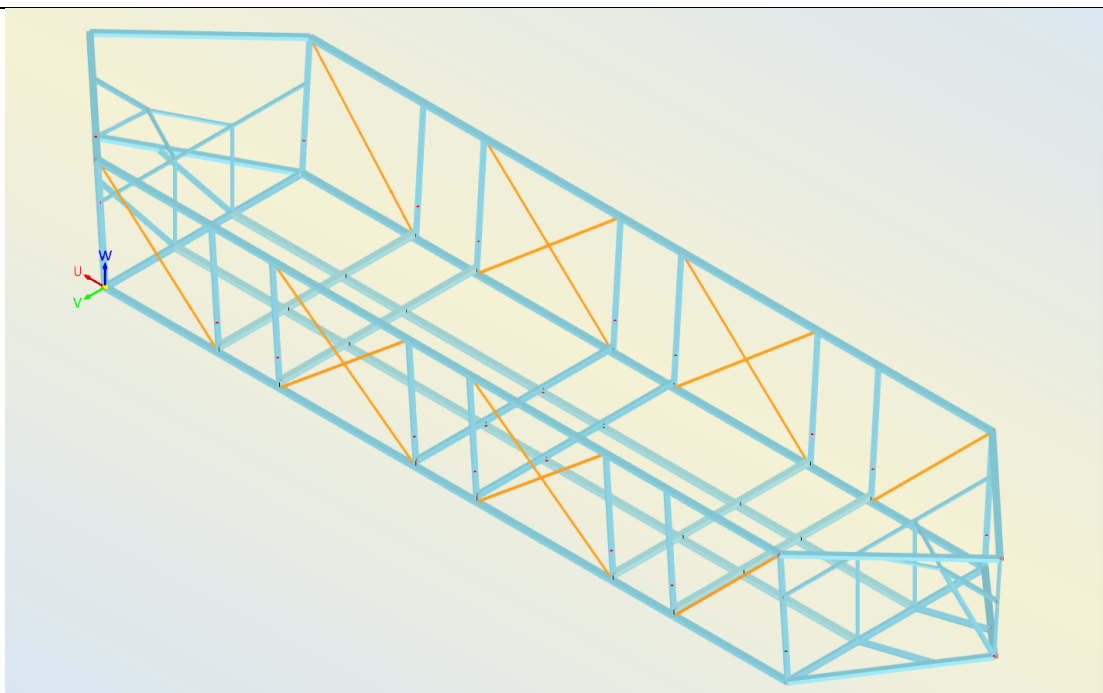


图 3-3 梁、拉杆单元设置图

参考胶粉比重 5%与构件补强 5%，模型的理论质量为 $\bar{M} = 96.5 \times 1.1 = 106.15\text{g}$ ，考虑几何性质，材料性能，制作环境的变异性，取 $\sigma = 10\% \times \bar{M}$ 为正态分布标准差。模型的估重结果为：

$$M \in [M - \sigma, M + \sigma] = [96.5 \times 1.1 \times 0.9, 96.5 \times 1.1 \times 1.1]$$

即模型质量分布在 $[95.535, 116.765]$ （单位：g）。

3.2.3 荷载工况

本次模型主要承受荷载有下方的水压力和上方砝码的重力。其中，装载铁砂的坛状砝码较为特殊，此处仅讨论如何给模型施加水压及一般砝码的重力。

水压部分，首先计算水位线的高度。该模型可以视为一个棱柱体和一个棱台的组合体，其中棱柱体底面积为 0.276 m^2 ，高为 0.15 m ；棱台下底底面积为 0.276 m^2 ，上底底面积为 0.234 m^2 ，高为 0.035 m ，并且假设棱台截面积随高度线性变化。因此，当吃水深度为 x 时，船体所受浮力 $F_{\text{浮}}$ 为：

$$F_{\text{浮}} = \begin{cases} 5880x^2 + 2293.2x, & 0 < x < 0.035 \\ 2704.8x - 7.203, & 0.035 < x \leq 0.185 \end{cases}$$

其中 $F_{\text{浮}}$ 的单位为 N， x 的单位为 m。

经过验算可以证明，在计算时因所做的假设引起的误差是可以忽略的。

代入与浮力相等的各阶段荷载，即可求出水位线的高度。在 RFEM6 中对水位线以下的船体表面施加面荷载，设置为 Z 方向线性，最小压强为 0 Pa，最大压强通过吃水深度计算得到，即可模拟水压对模型的影响。

此外，还可以验算水位线的正确性，详见下文结构支座。

对于较轻的砝码，在砝码受结构承载的位置设置对应数量的点荷载，将其重力均摊即可。

3.2.4 边界条件

由于本次模型整体浮在水中，模型的边界条件不易确定。下面以一级加载为例，介绍边界条件的确定方式。

注意到模型总重约 100 g，而坛状砝码重 8 kg，远远大于模型自重，因此可以尝试将坛状砝码视为支座，而将船体下部的水压视为荷载。如图 3-4 所示，图中绿色部分即为一级加载中砝码和船体接触部分，各分配一个支座。

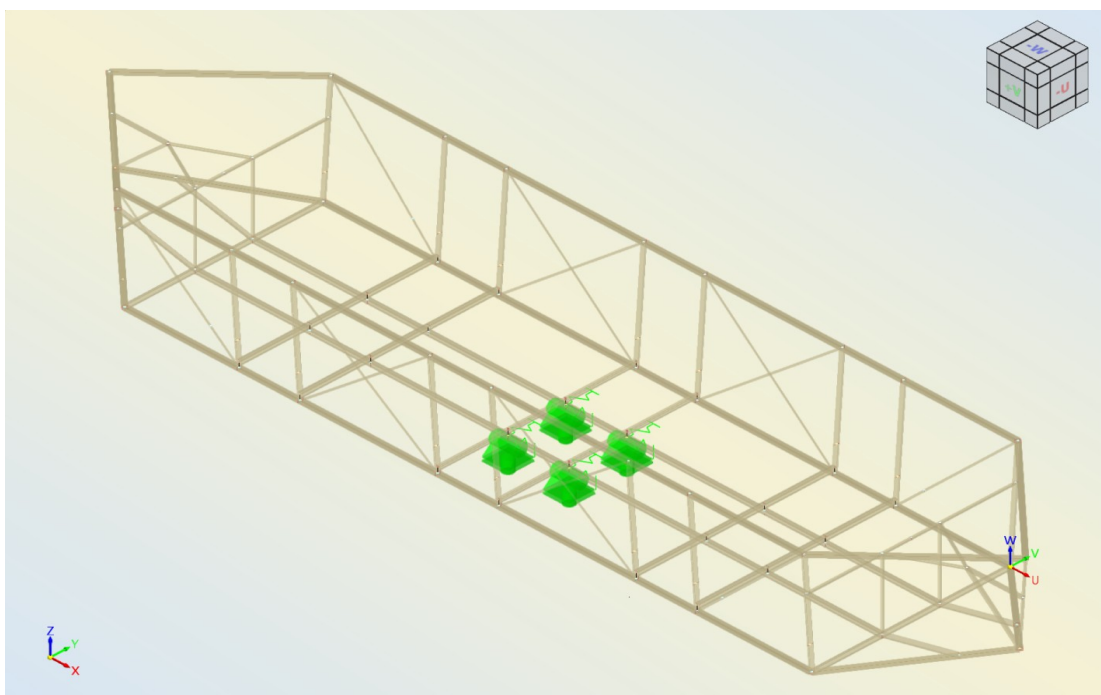


图 3-4 结构支座设置图

支座类型设计为铰接，约束 X，Y，Z 方向的平动。计算完成之后检验支座反力可以验算水位线的正确与否，如一阶段荷载为 8 kg，支座反力应为 78.4 N 左右。反力过大则应降低水位线，反之亦然。

同理，对于其他加载阶段，在荷载处设置支座，用支座反力校核其正确性。

荷载和支座设置完成后的模型如图 3-5 所示。

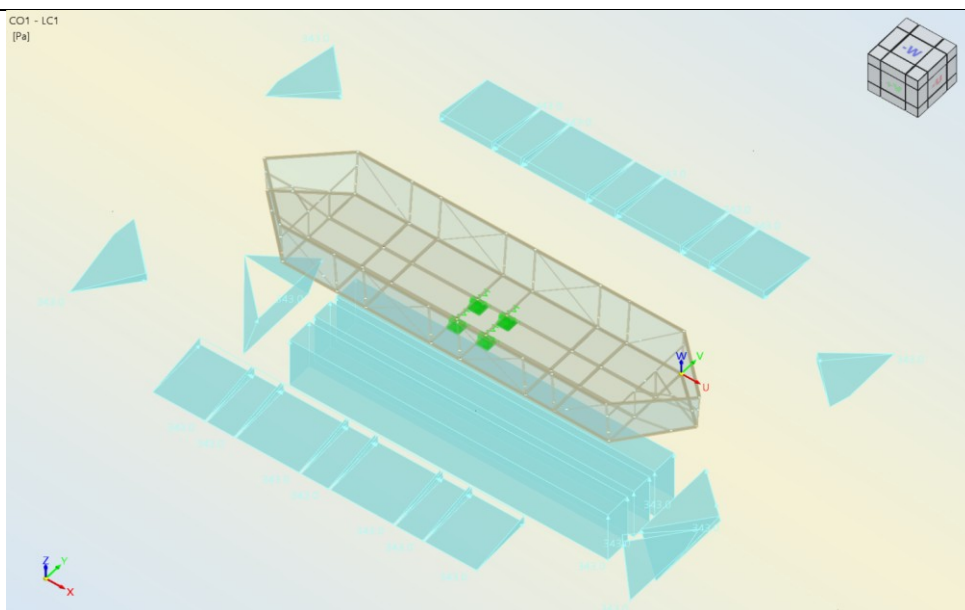


图 3-5 荷载-支座设置图

4 受力分析

4.1 强度分析

4.1.1 第一级加载

图 4-1~图 4-6 列出了模型在一级加载工况下的计算结果。从计算结果可知，在一级加载工况下，杆件最大拉应力出现在放置 8 kg 坛形砝码处，为 63.8 MPa。最大压应力出现在船头工字杆底部，为 32.6 MPa。杆件满足强度要求且有较大富余，可以继续完成二级、三级荷载。

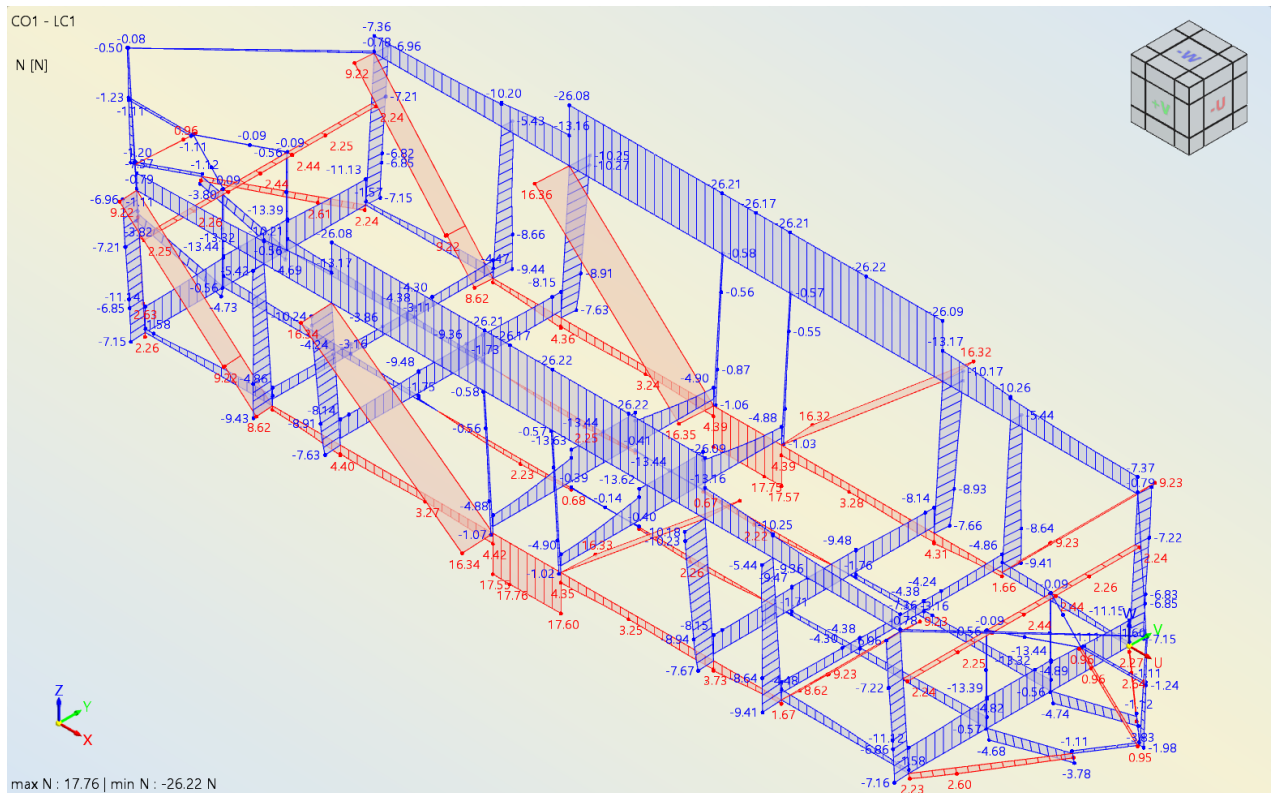


图 4-1 一级加载杆件轴力（单位：N）

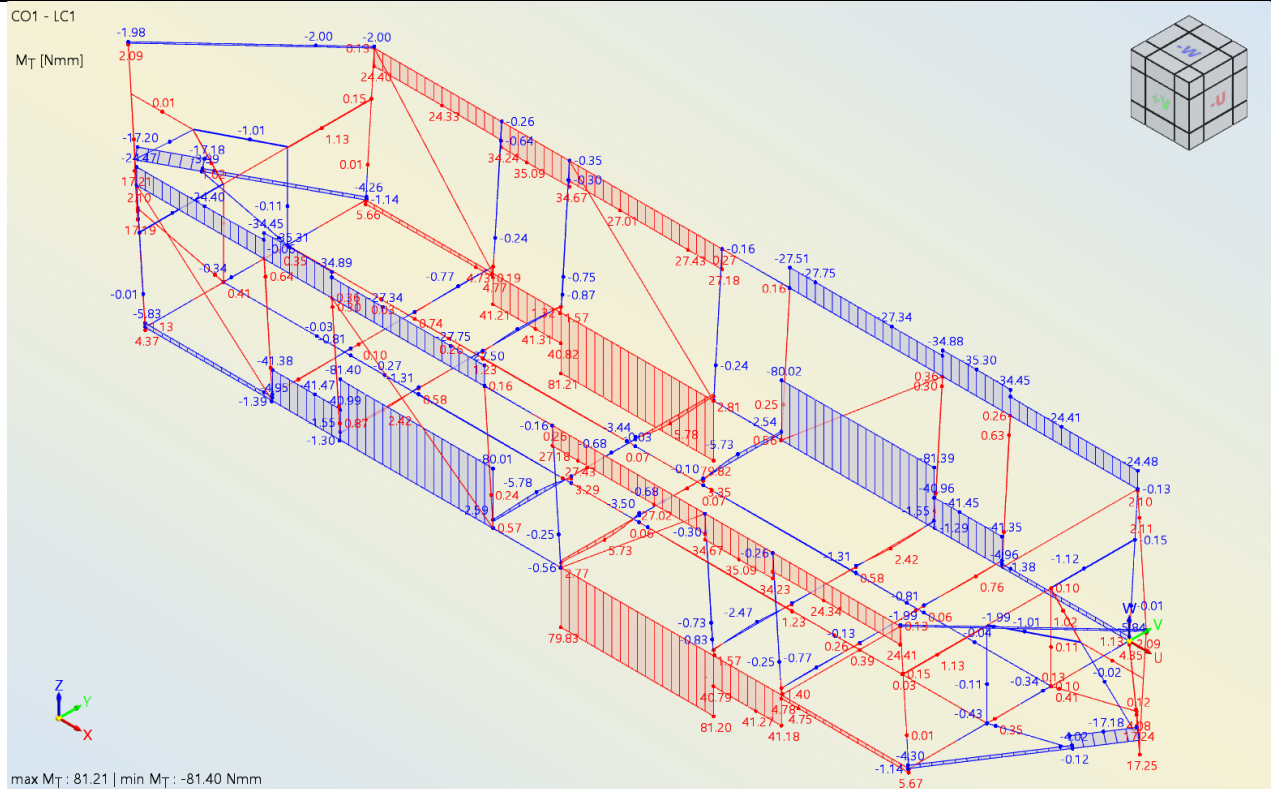


图 4-2 一级加载弯矩 M_x (单位: $N \cdot mm$)

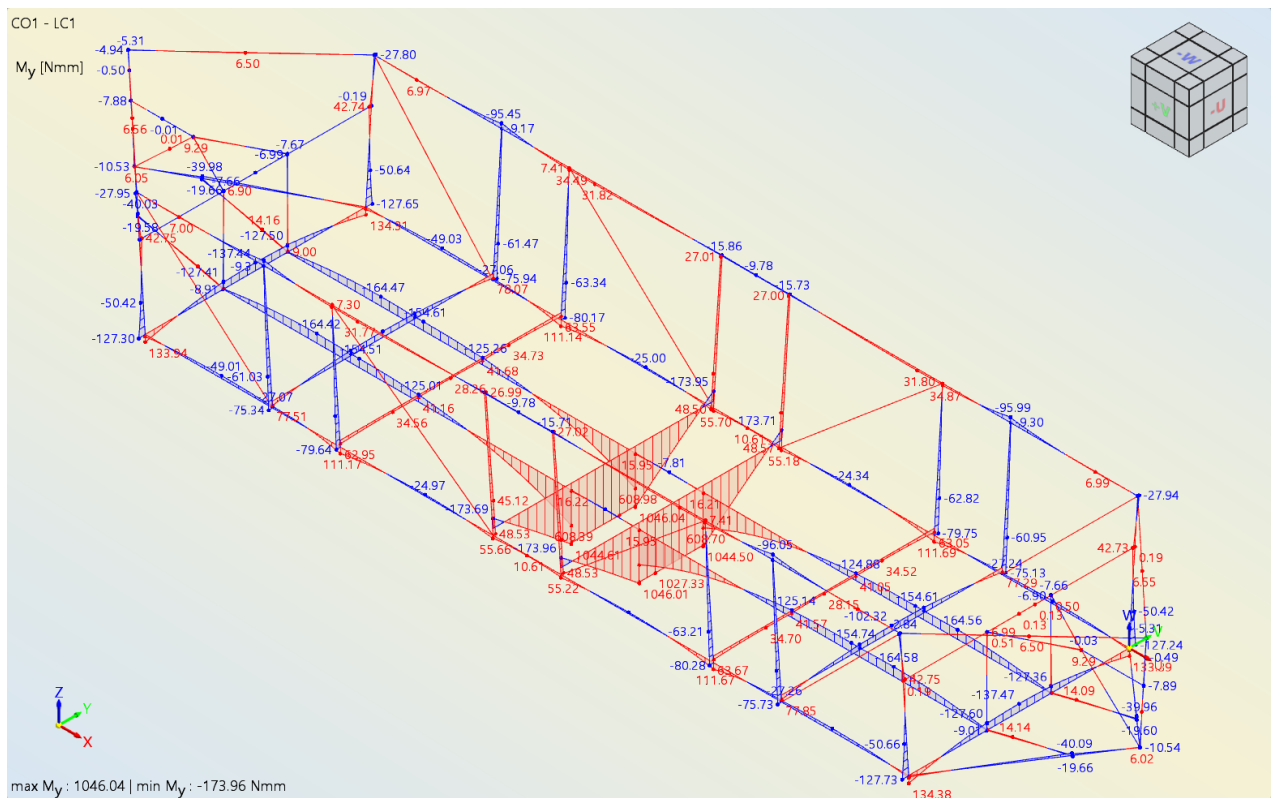


图 4-23 一级加载弯矩 M_y (单位: $N \cdot mm$)

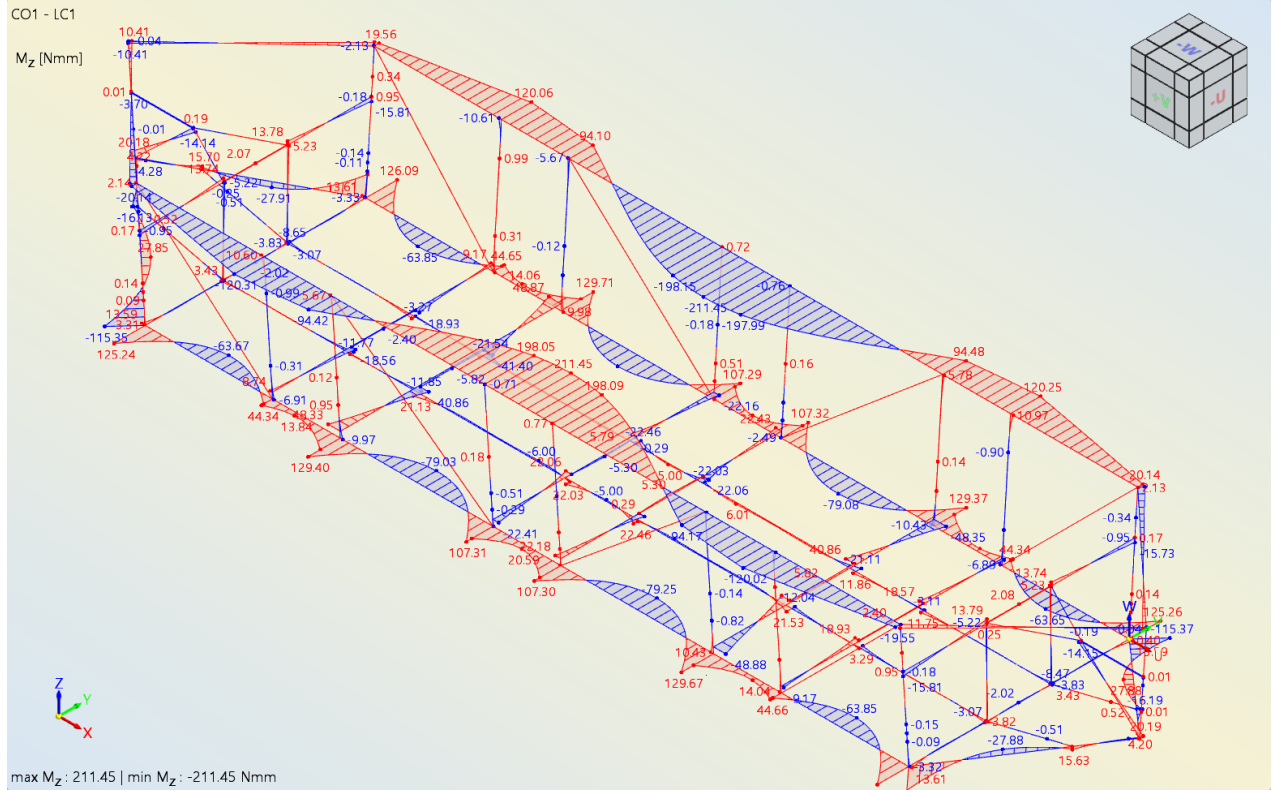


图 4-4 一级加载弯矩 M_z (单位: $N \cdot mm$)

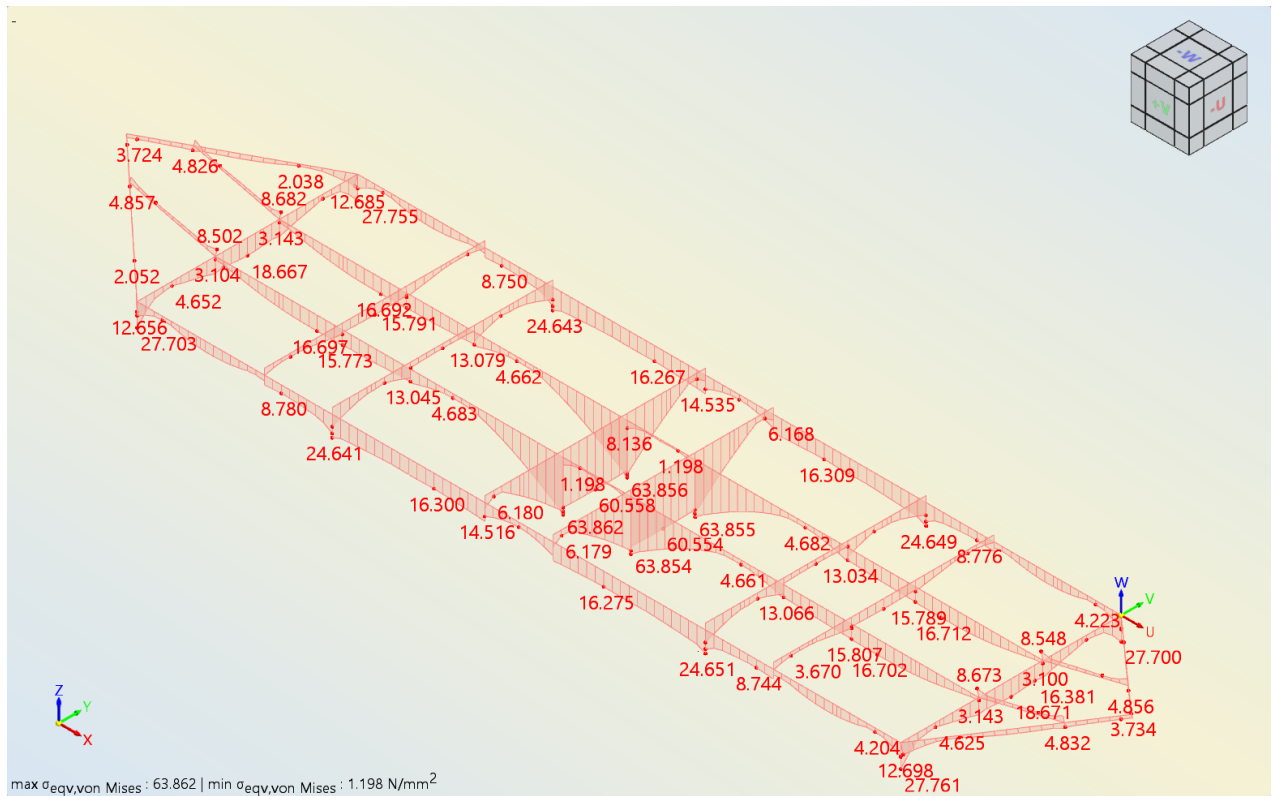


图 4-5 一级加载船底部龙骨等效应力 $\sigma_{eqv.vonMises}$ (单位: MPa)

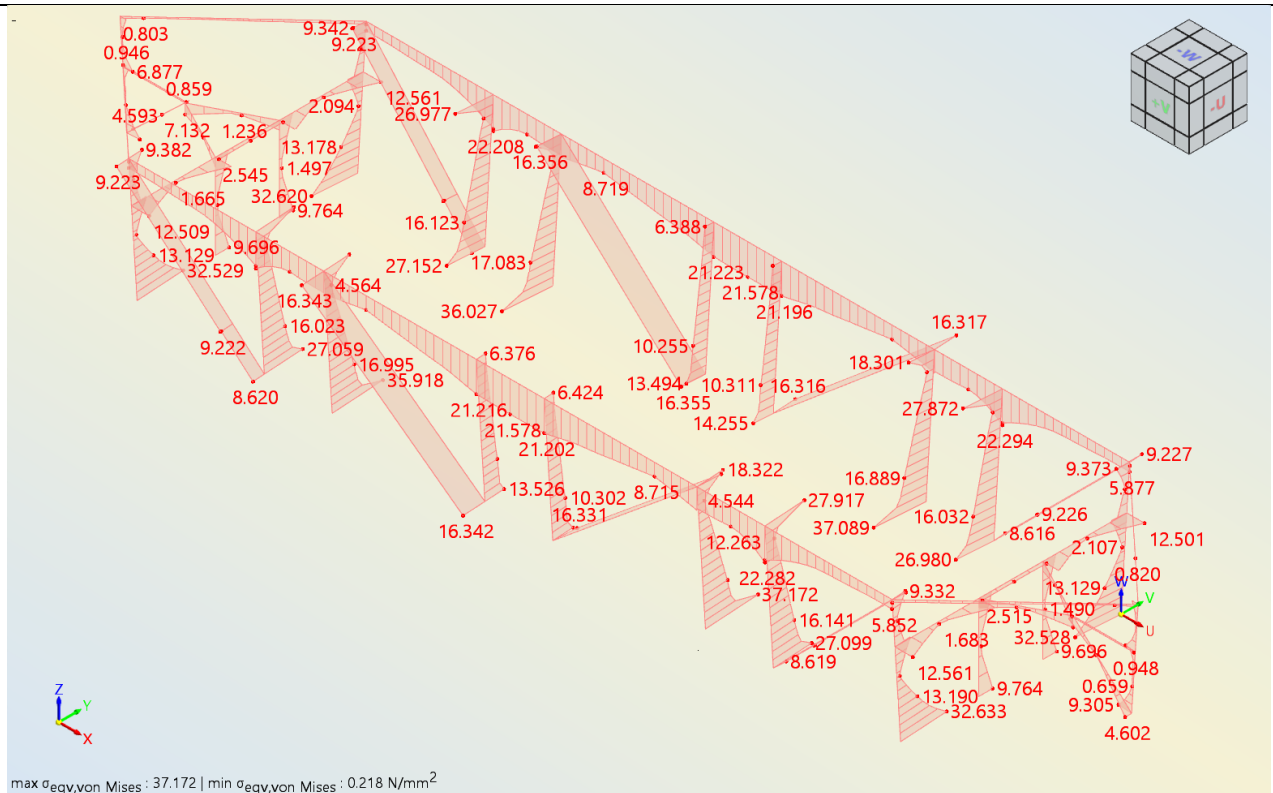


图 4-6 一级加载船舷等效应力 $\sigma_{eqv,vonMises}$ （单位：MPa）

4.1.2 第二级荷载

图 4-7~图 4-12 列出了模型在二级加载工况下的计算结果。从计算结果可知，在二级加载工况下，应力主要集中在近水端。最大压应力出现在船头 T 字杆处，为 85.8 MPa，最大拉应力出现在座位斜撑处，为 83.6 MPa。所有杆件满足强度要求，可以继续完成三级加载。

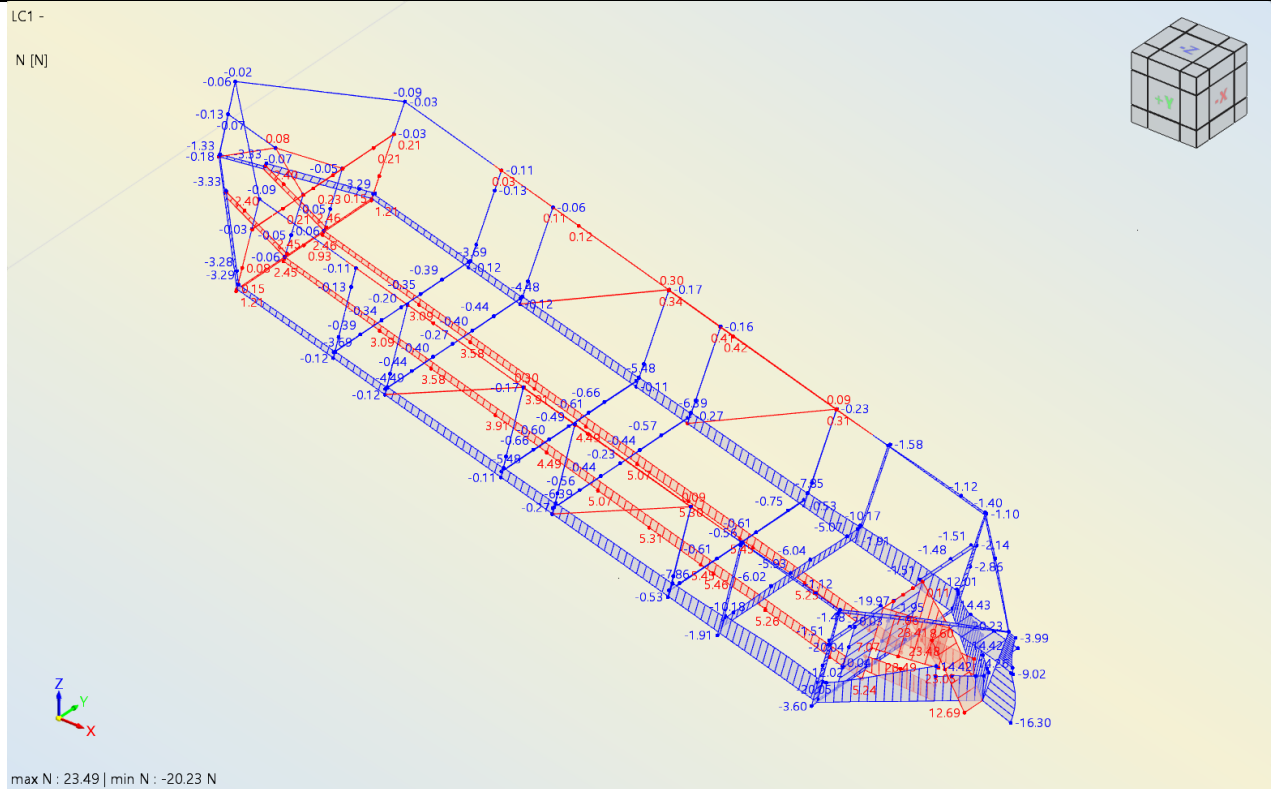


图 4-7 二级加载杆件轴力 N (单位: N)

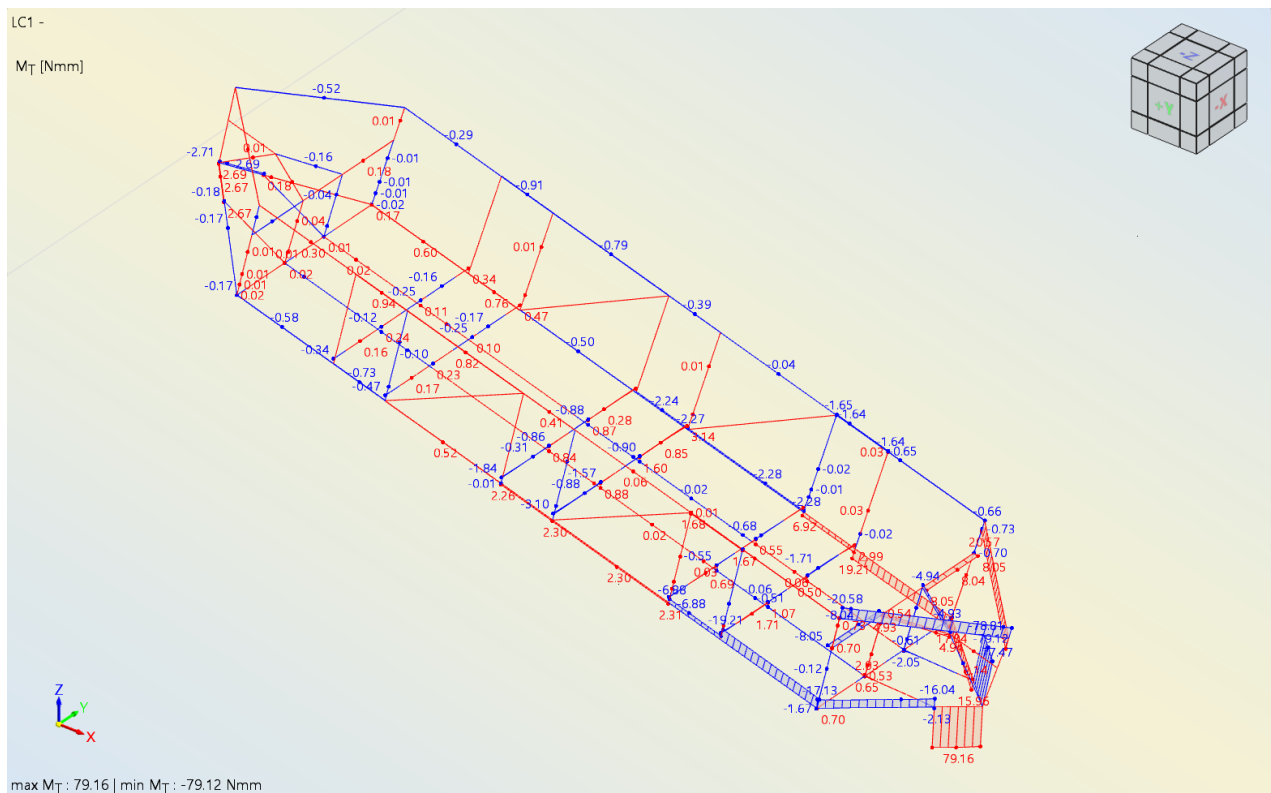


图 4-8 二级加载弯矩 M_T (单位: N · mm)

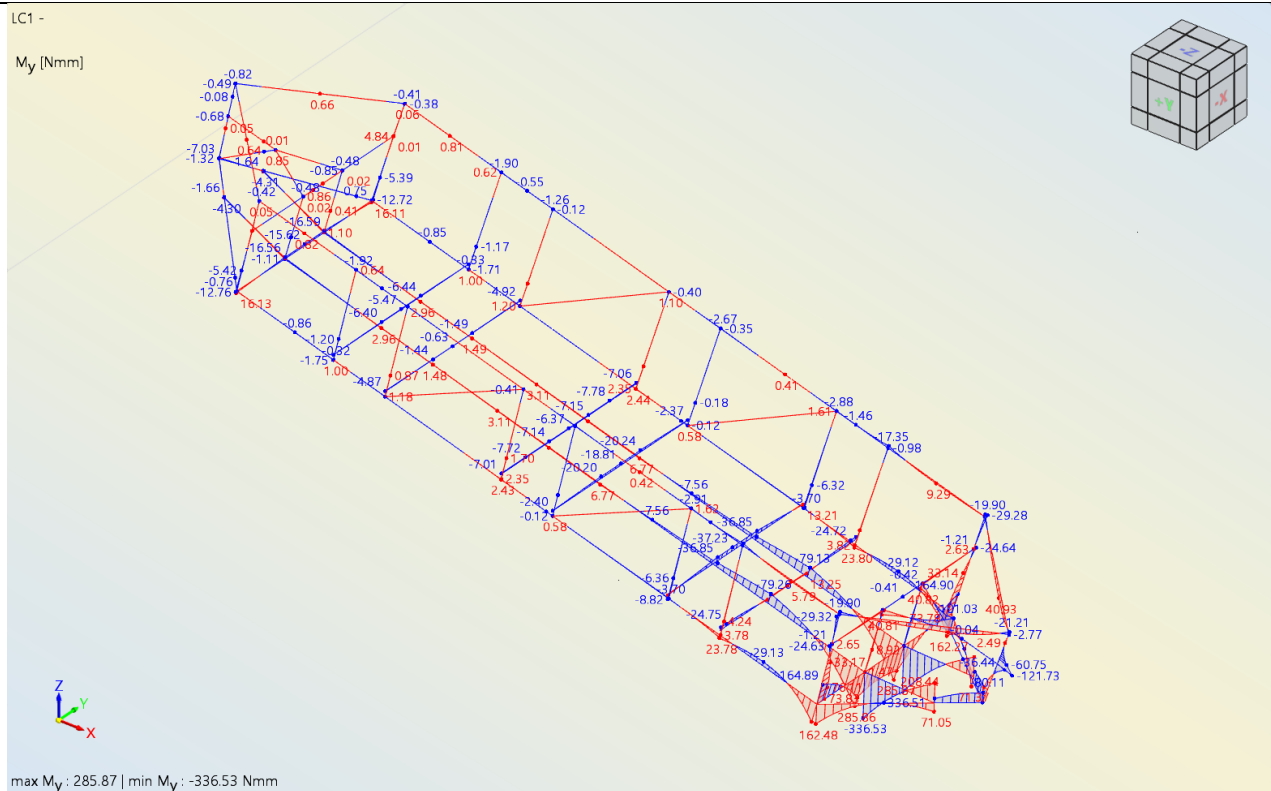


图 4-9 二级加载弯矩 M_y (单位: $N \cdot mm$)

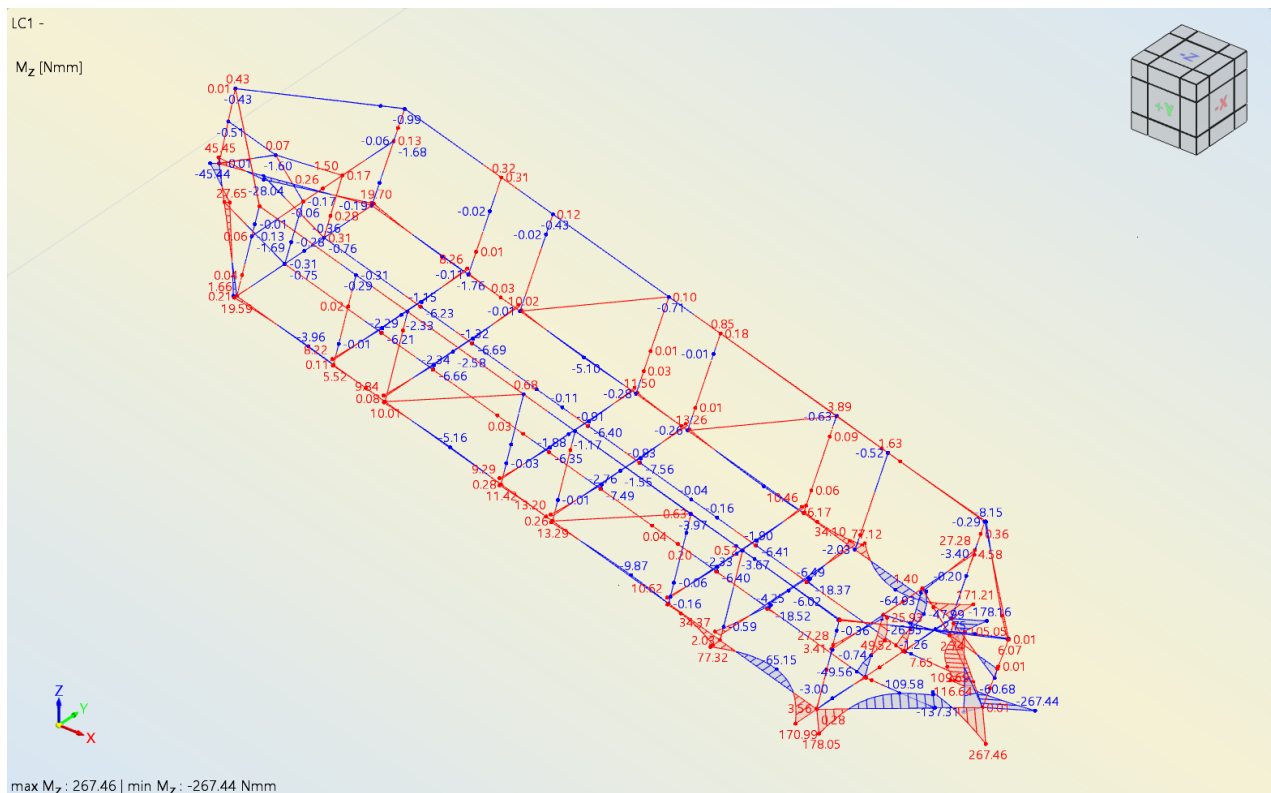


图 4-10 二级加载弯矩 M_x (单位: $N \cdot mm$)

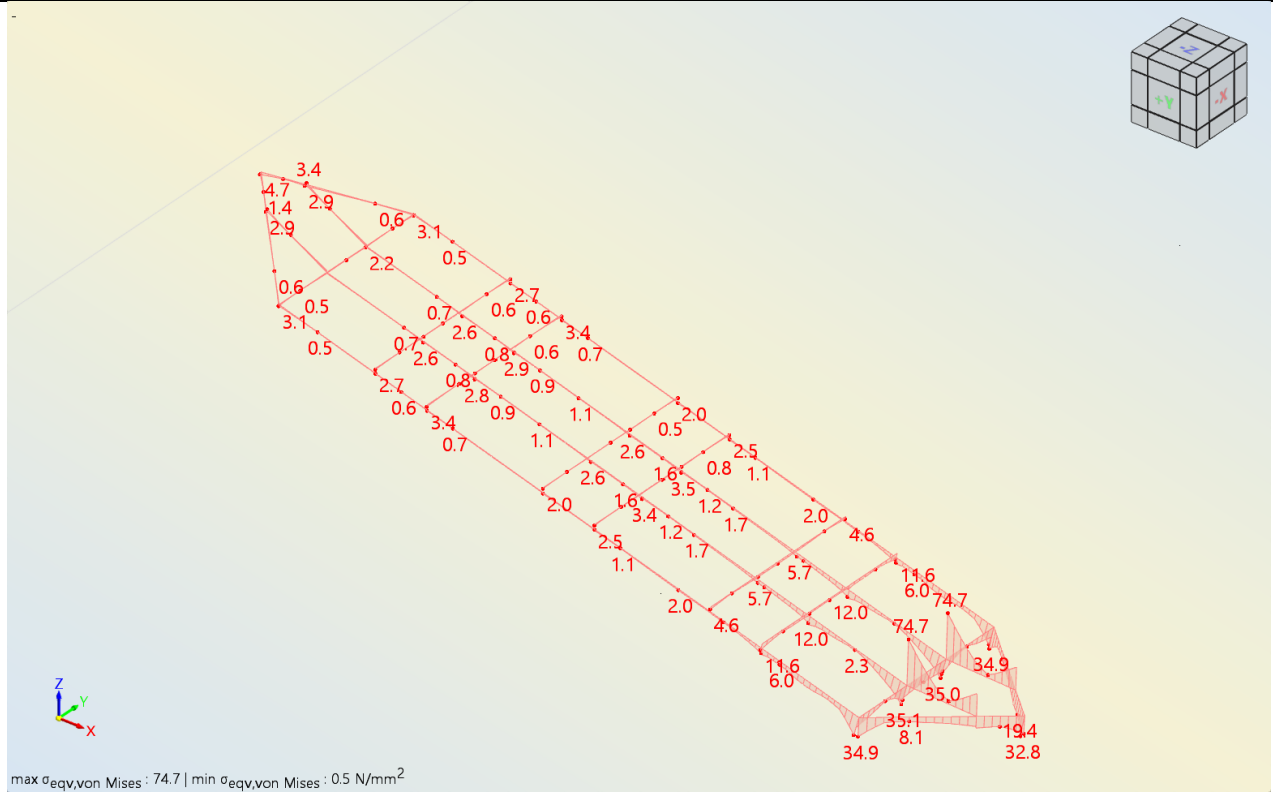


图 4-11 (a) 二级加载船底部龙骨等效应力 $\sigma_{eqv.vonMises}$ (单位: MPa)

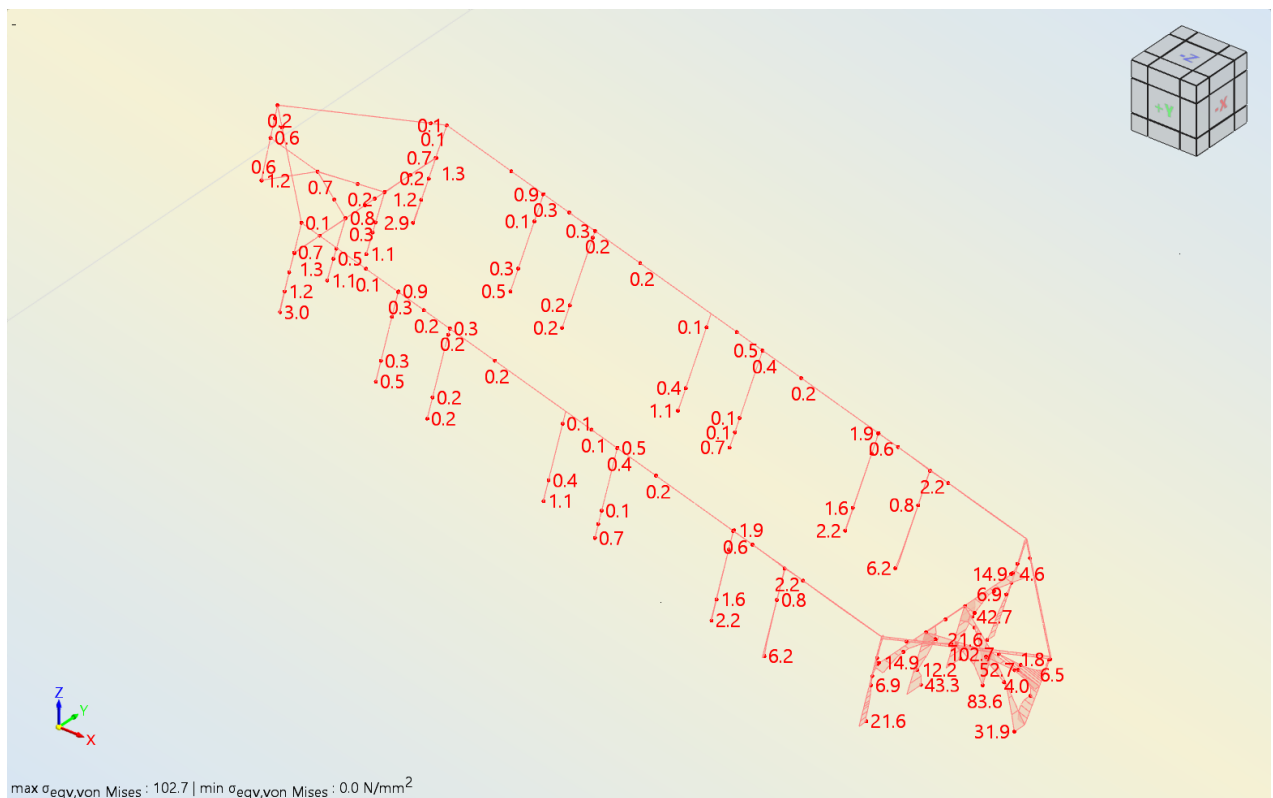


图 4-12 二级加载船舷等效应力 $\sigma_{eqv.vonMises}$ (单位: MPa)

4.1.3 第三级荷载

图 4-13~图 4-19 列出了模型在三级加载工况下的计算结果。从计算结果可知，在三级加载工况下，最大拉应力出现在 6 kg 坛状砝码放置处龙骨上，为 211.4 MPa；最大压应力出现在 6 kg 坛状砝码放置处横梁中，为 181.2 MPa。

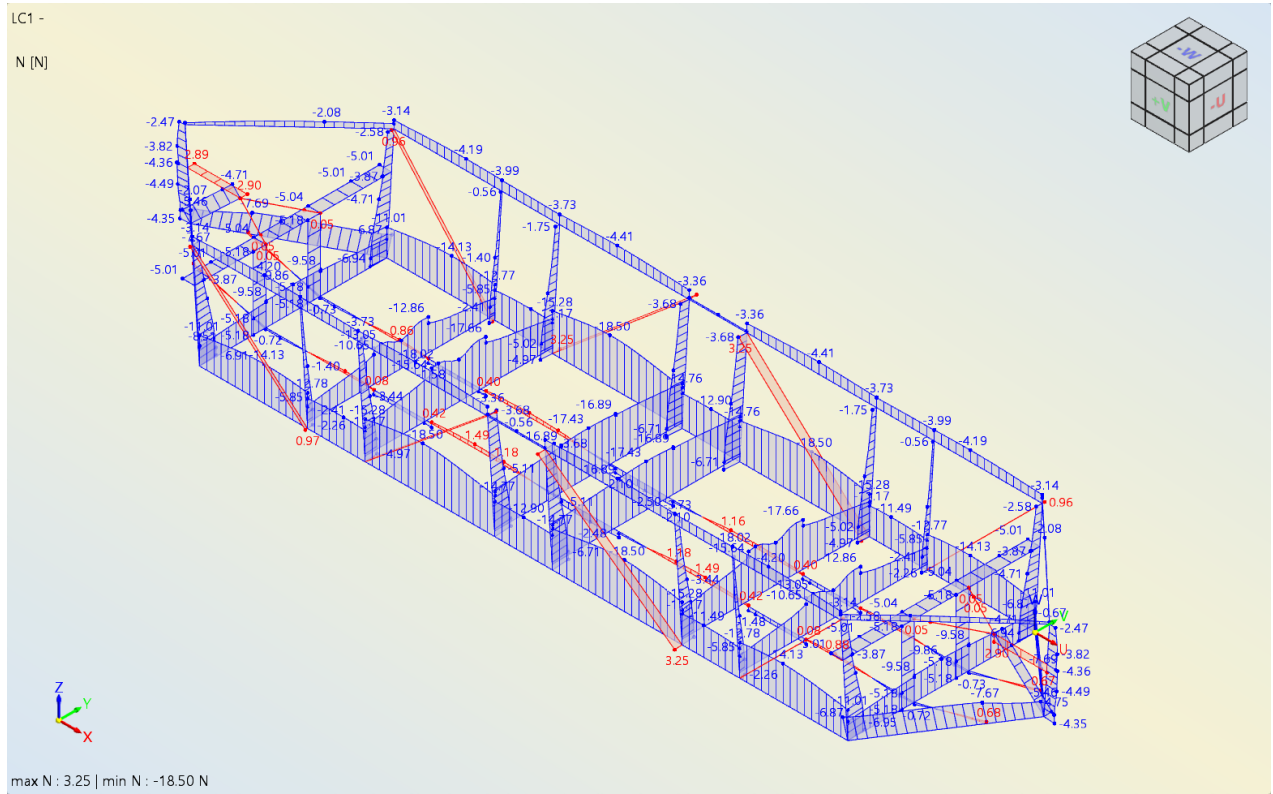


图 4-13 三级加载杆件轴力 N (单位: N)

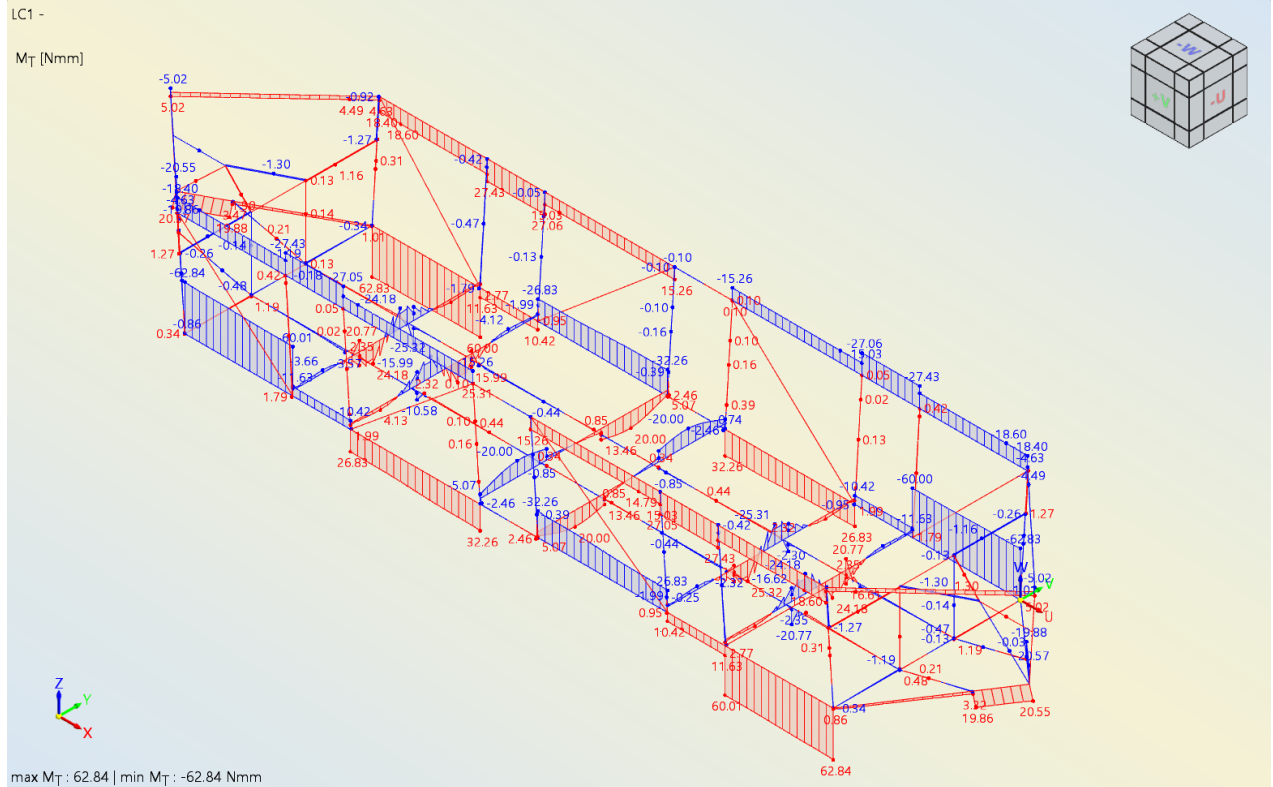


图 4-14 三级加载弯矩 M_T (单位: $N \cdot mm$)

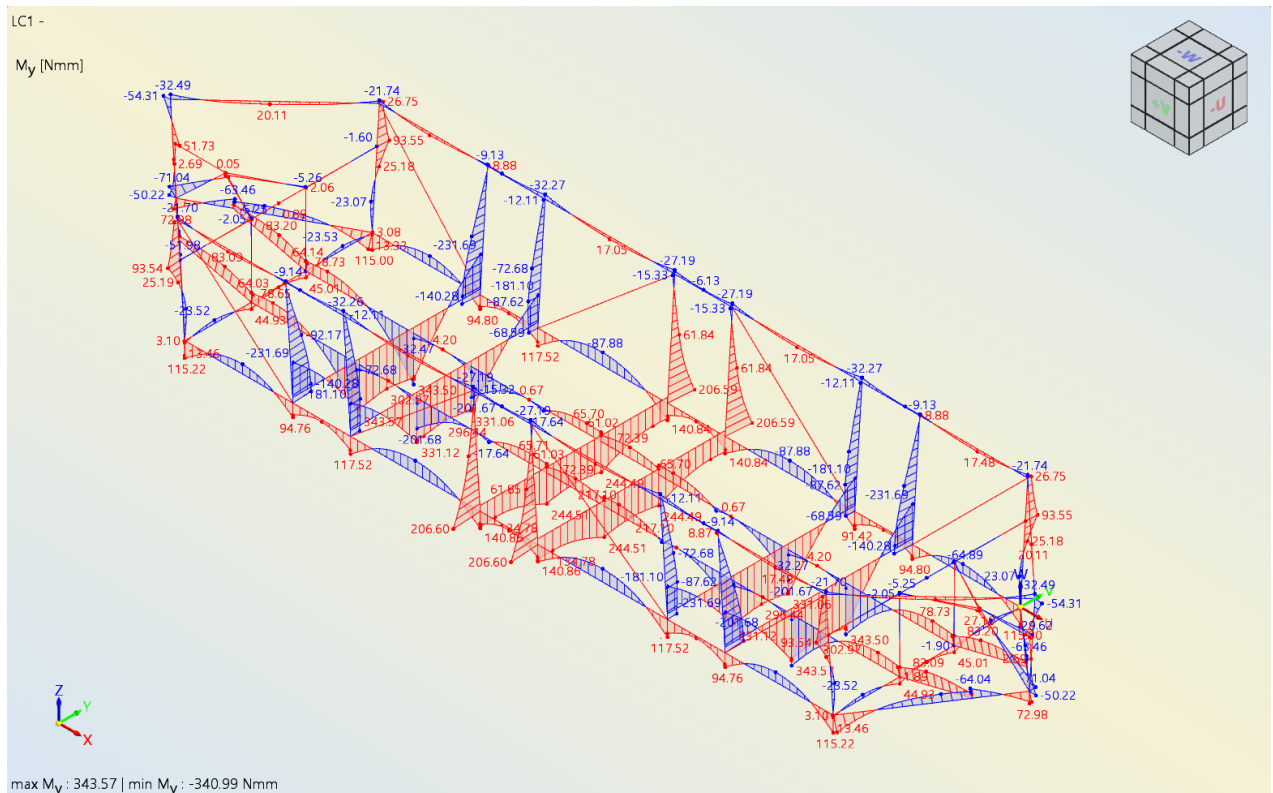


图 4-15 三级加载弯矩 M_y (单位: $N \cdot mm$)

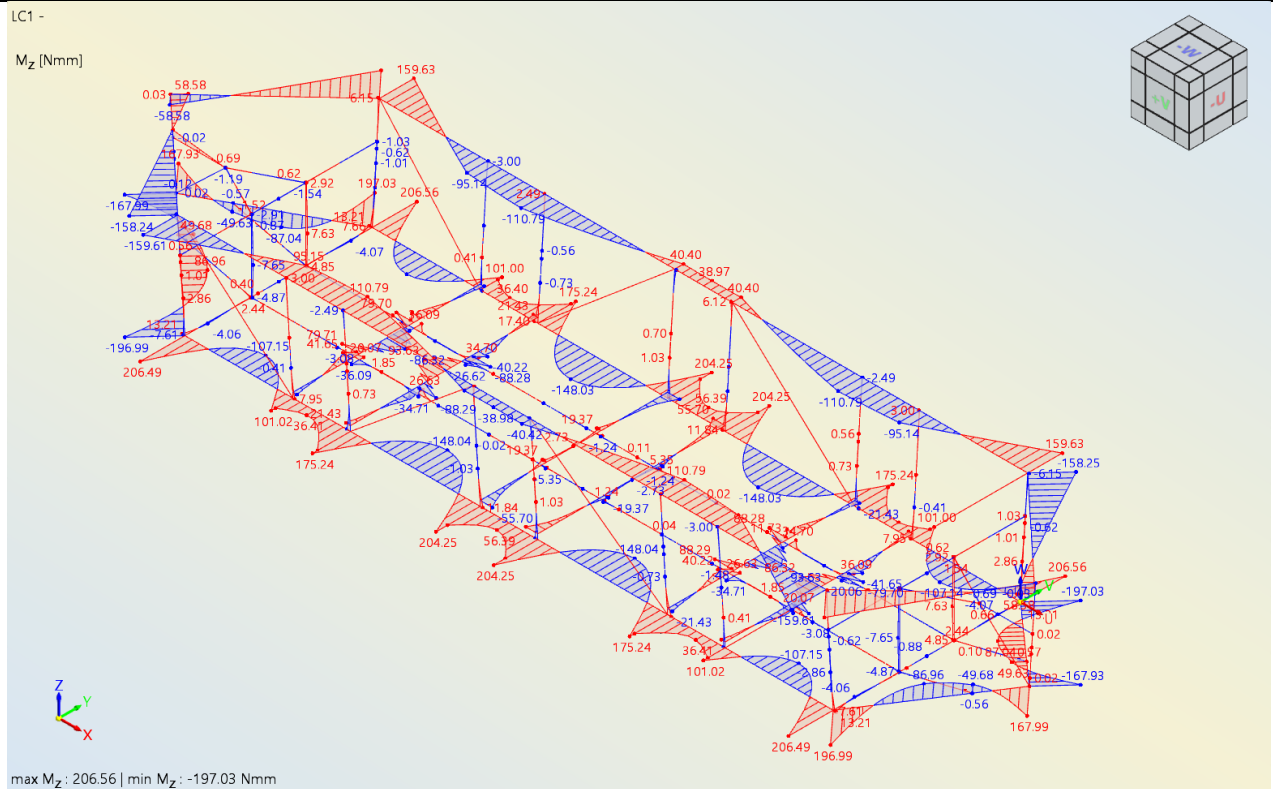


图 4-16 二级加载弯矩 M_z (单位: $N \cdot mm$)

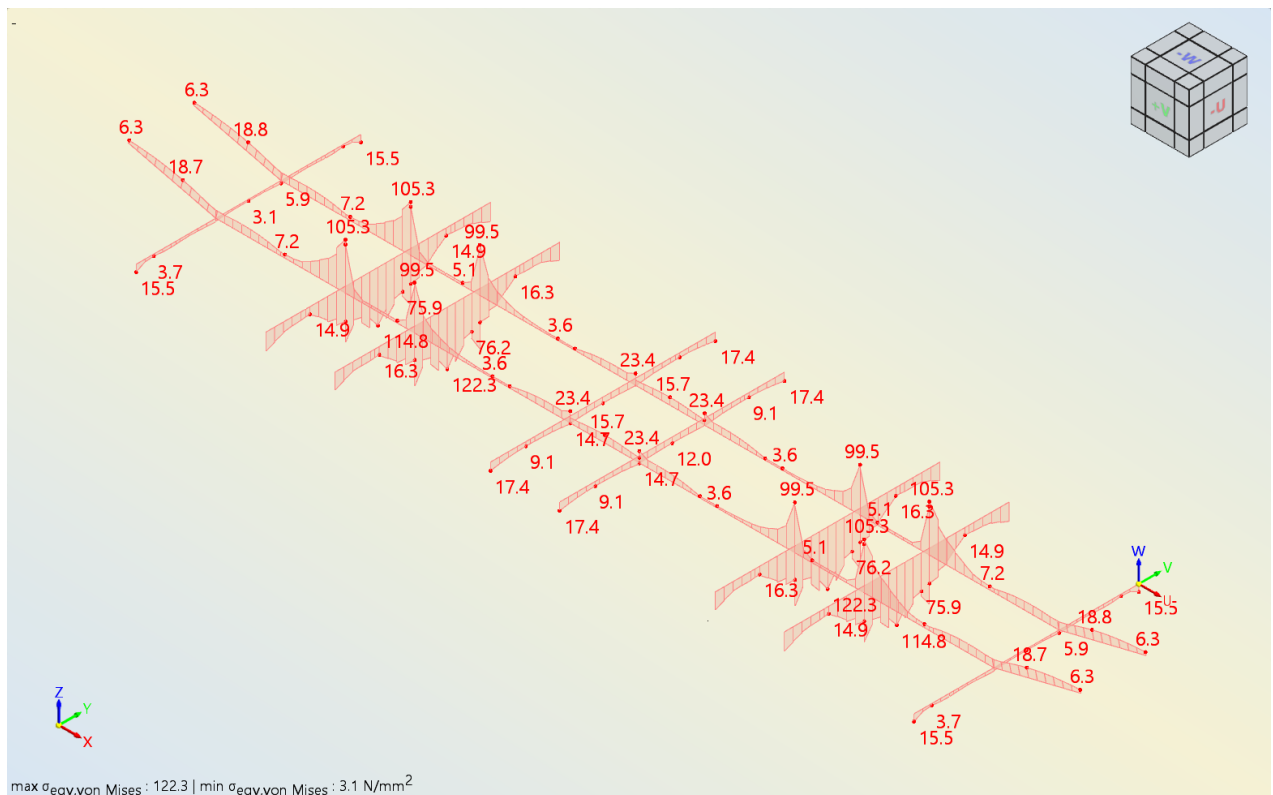


图 4-17 三级加载船底部龙骨等效应力 $\sigma_{eqv.vonMises}$ (单位: MPa)

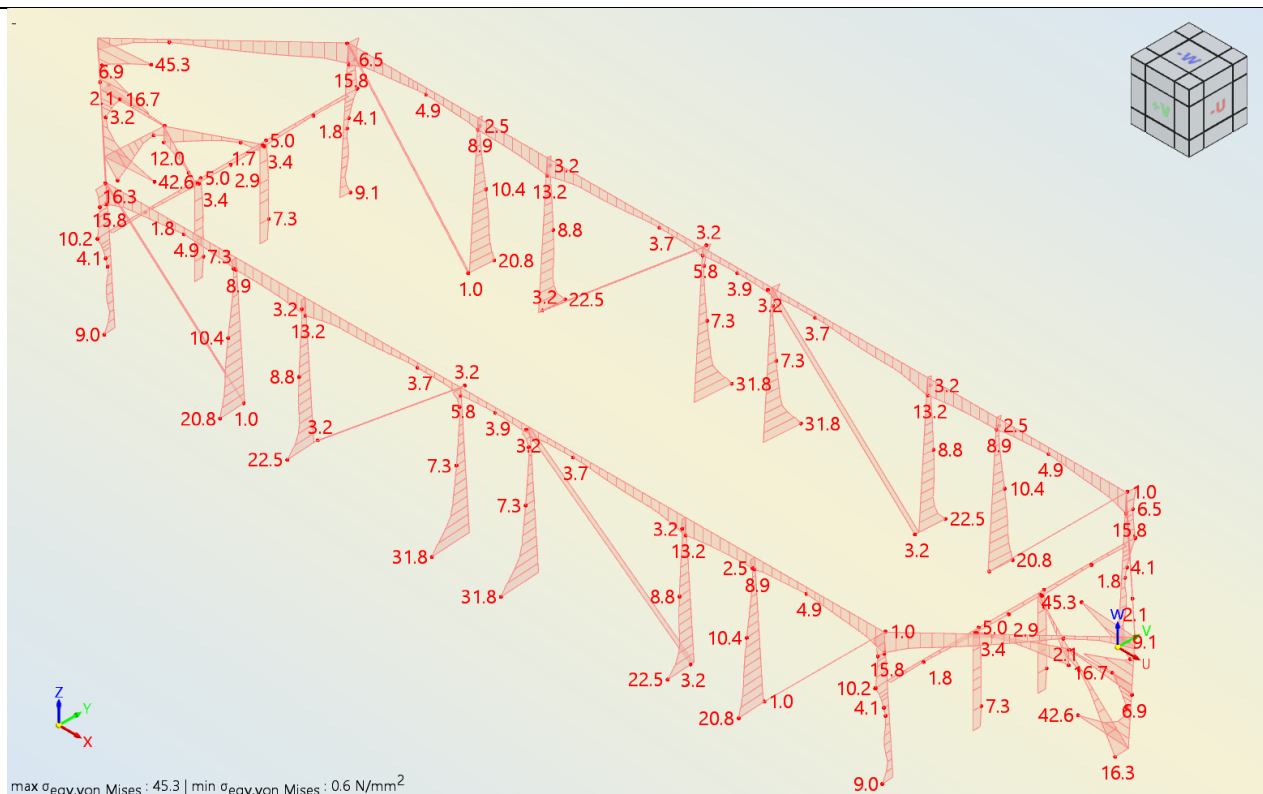


图 4-18 三级加载船舷等效应力 $\sigma_{eqv,vonMises}$ (单位: MPa)

从计算结果可知,最大拉、压应力均超过材料强度限制。然而计算中出现的危险截面均出现在结构支座处,支座反力是一个集中荷载,而实际上砣码对模型施加的应该是面荷载,同时,砣码底面对模型也有一定的约束效果,因此在结构支座处的应力将会较实际情况偏大,即存在应力集中现象。

除结构支座处以外,最大拉应力 45.0 MPa 出现在横梁上,最大压应力 48.6 MPa 出现在船头 T 字杆处,杆件强度均满足要求。

经过实验检验,放置砣码处模型强度满足要求。综上所述,本模型可以满足三级加载所有工况的要求。

4.2 刚度分析

一级加载工况变形计算结果详见图 4-19、图 4-20。在前文中已经提到,要减小船只在滑行阶段的阻力,应该使船头在一级荷载的工况下正好接触水面。因此需要根据船头在一级荷载时的变形情况调整设计。从图中可以看出,船头位移为 16.0 mm,这会使船头刚好到达水位线的位置。

模型最大位移出现在上舷中部,向内位移 20.4 mm。在横向上,船体各处形变较为均匀,不会影响船只的稳定性和正常使用。模型刚度满足一级荷载要求。

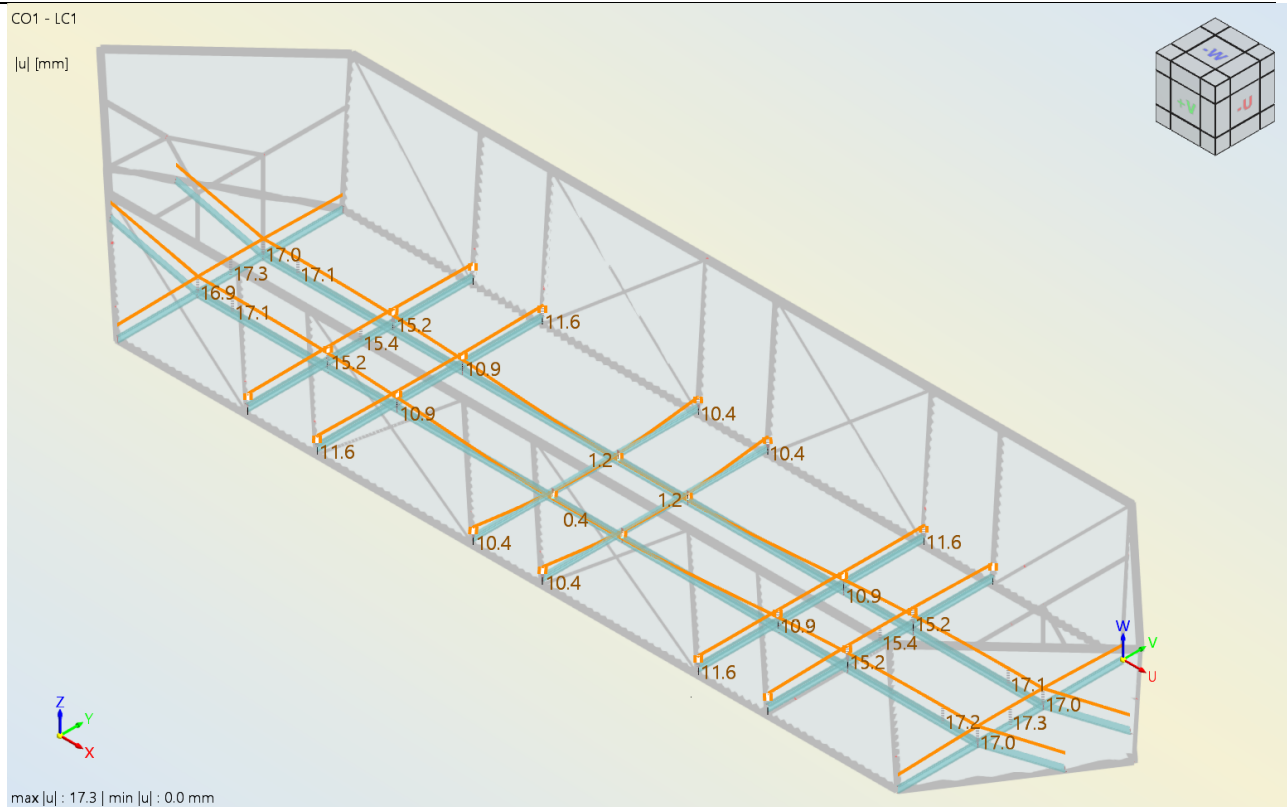


图 4-19 一级加载位移图（单位：mm）

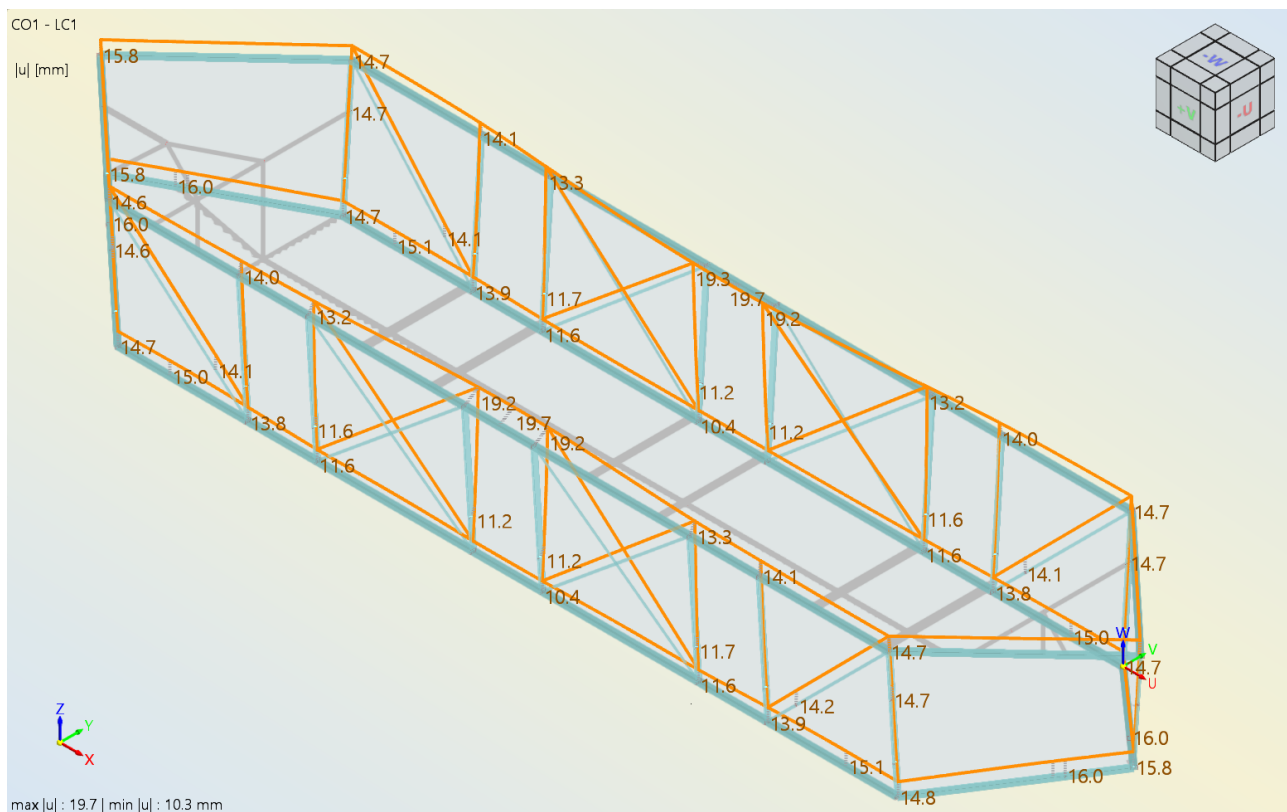


图 4-20 一级加载位移图（单位：mm）

二级加载工况变形计算结果详见图 4-21、图 4-22。由于二阶段船体支座设置在 1.5 kg 荷载平台处，船体不加荷载的一端船头（下称“远水端”）至支座距离 1040 mm，而施加荷载的一端（下称“近水端”）距支座 75 mm，杠杆效应明显，微小的误差在远水端会被放大 14 倍，

因此远水端的位移没有参考意义，下列图片为模型在一阶段工况下的各部件位移图，单位为毫米。可以发现模型虽然位移较大，但是这种位移主要是二级荷载工况的特殊性导致，各部件相对位移很小，形变不会影响正常使用。模型刚度满足二级荷载要求。

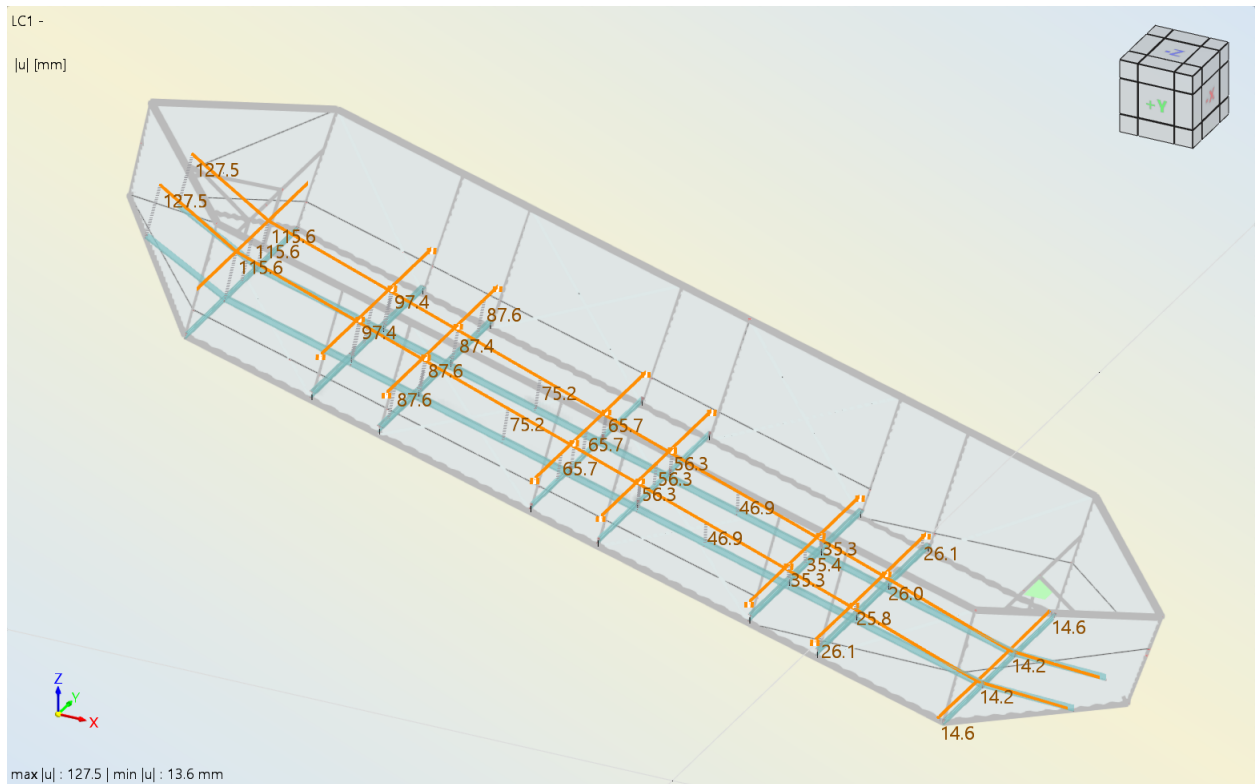


图 4-21 二级加载位移图（单位：mm）

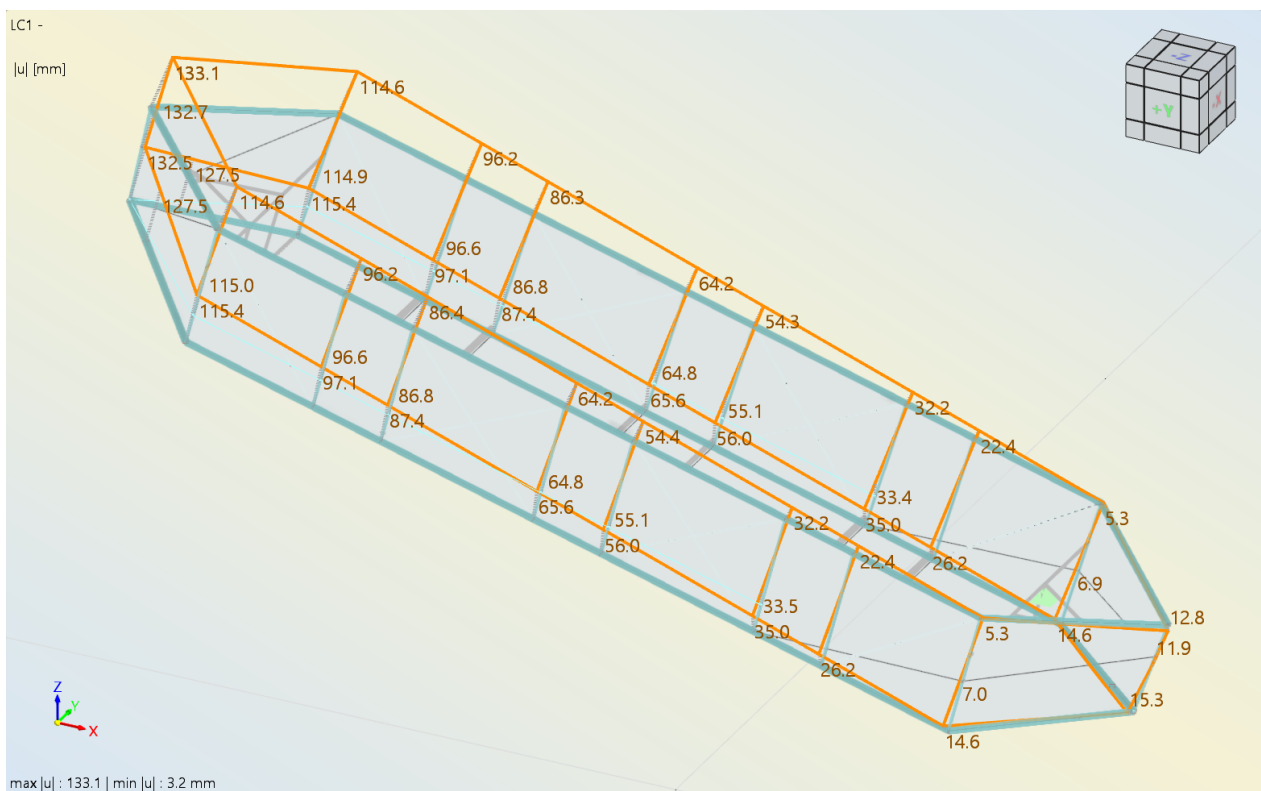


图 4-22 二级加载位移图（单位：mm）

三级加载工况变形计算结果详见图 4-23、图 4-24。模型最大位移出现在上舷中部，向内位移 20.5 mm。在横向上，船体各处形变较为均匀，不会影响船只的稳定性和正常使用。模型刚度满足三级荷载要求。

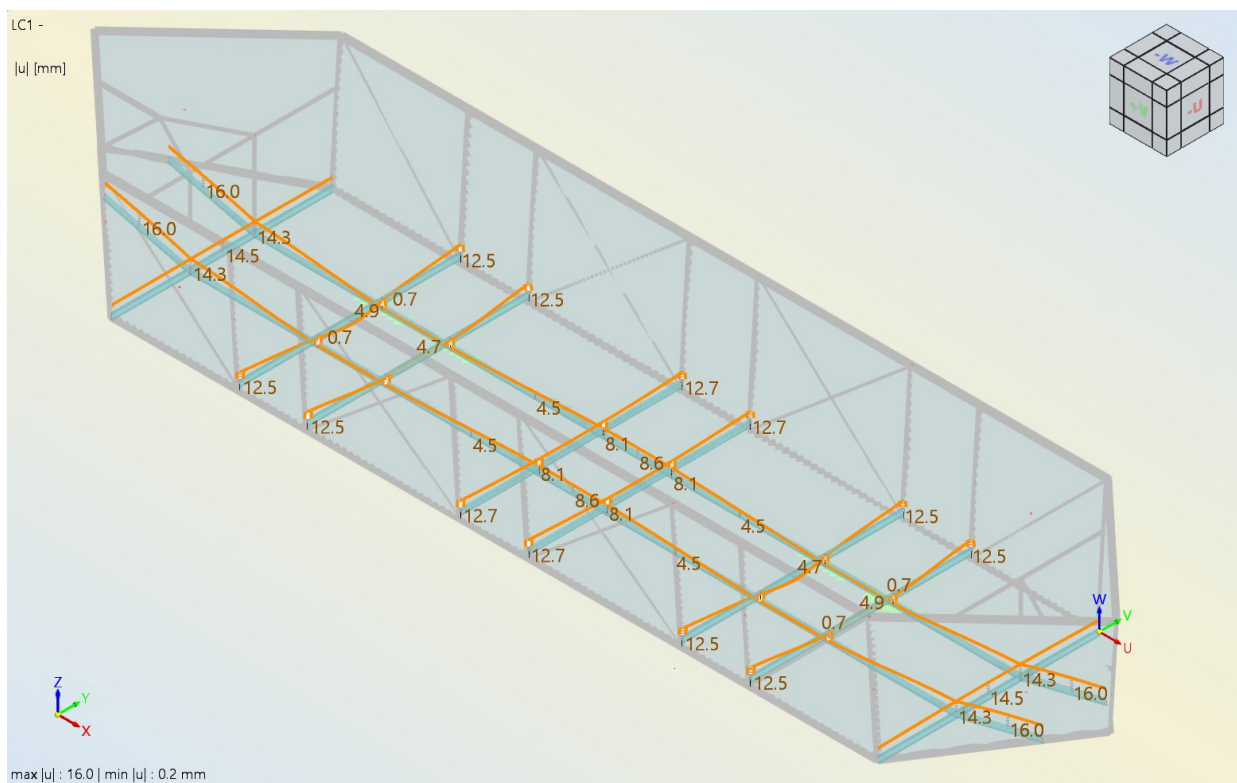


图 4-23 三级加载位移图（单位：mm）

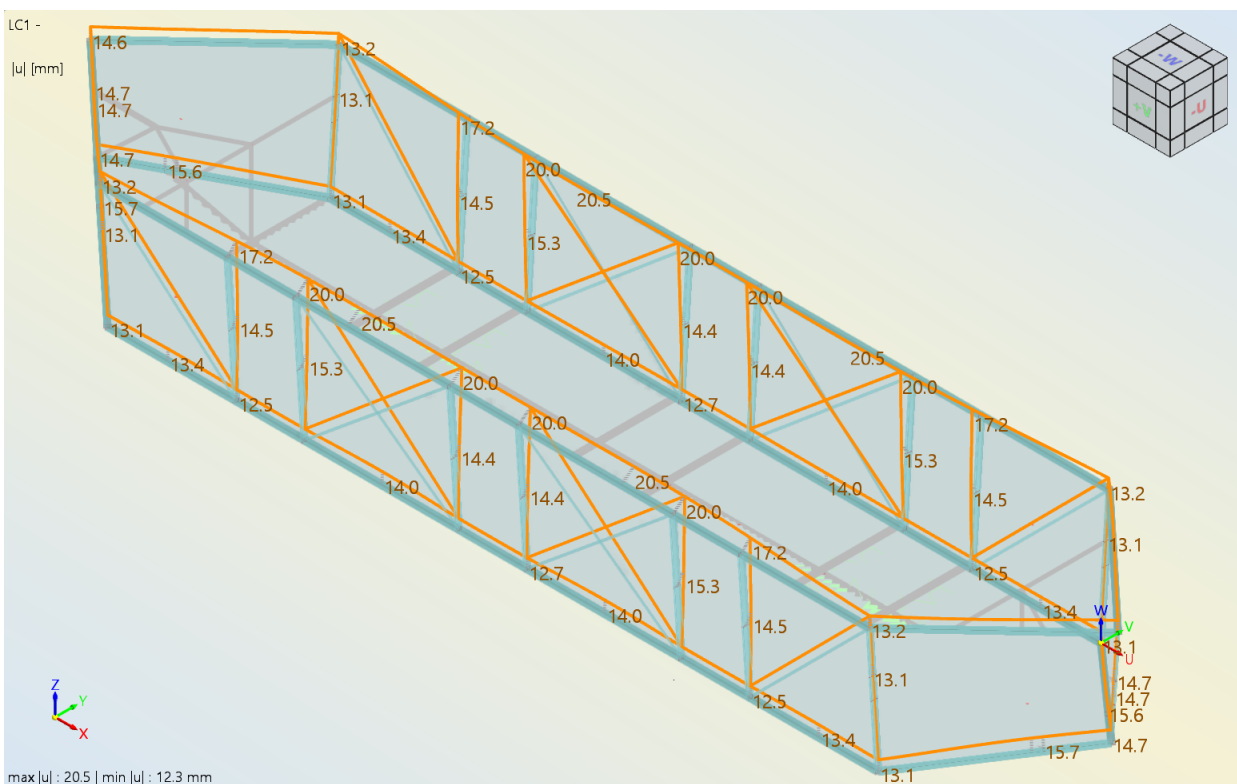


图 4-24 二级加载位移图（单位：mm）

4.3 稳定分析

采用有限元软件对该模型三种工况下的前四阶失稳模态进行了验算，计算结果见表 4-1。

表 2-6 各级荷载下的临界荷载系数与失稳模态图

荷载 工况	临界荷载 系数	失稳模态
一级 荷载	1.258	
	2.173	

“建行杯”第十八届全国大学生结构设计竞赛总决赛理论方案

荷载
工况

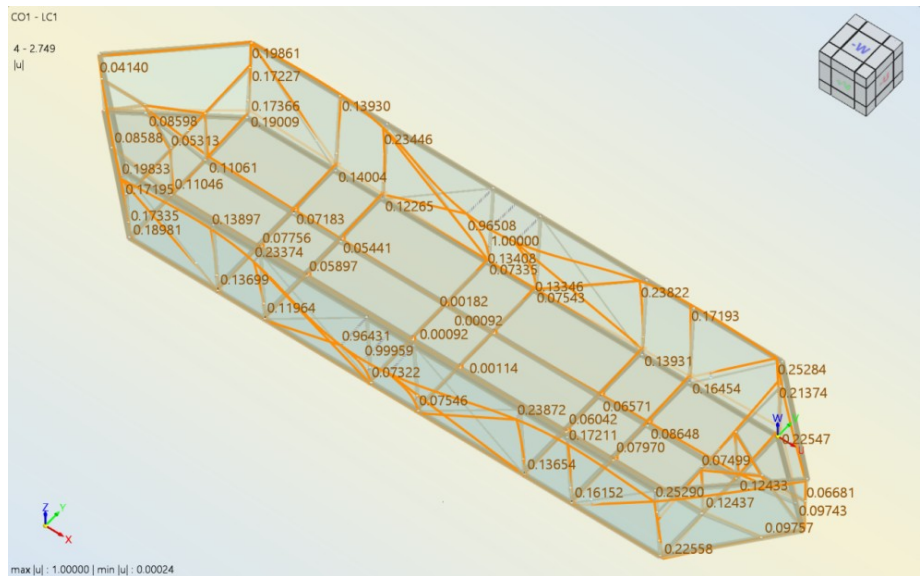
临界荷载
系数

失稳模态

2.419



2.749



二级
荷载

1.657



“建行杯”第十八届全国大学生结构设计竞赛总决赛理论方案

荷载
工况

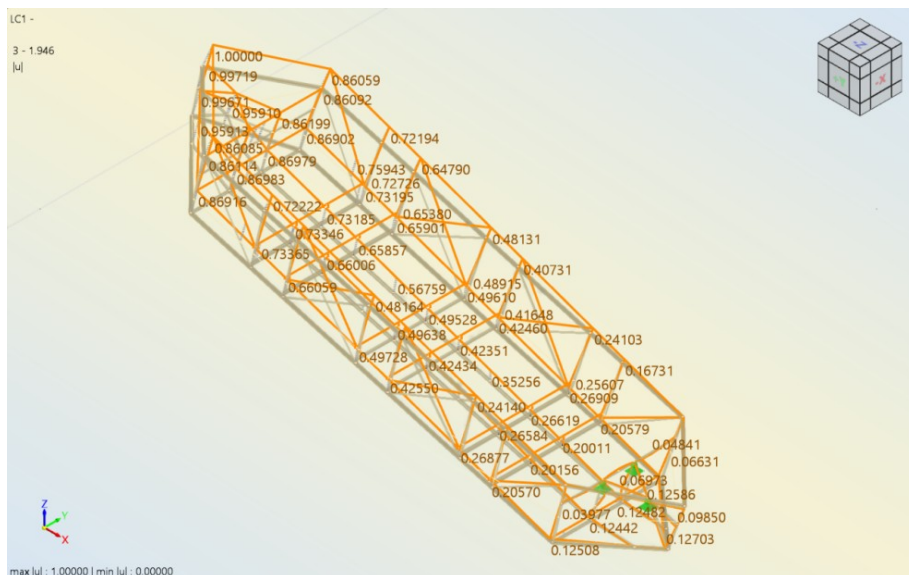
临界荷载
系数

失稳模态

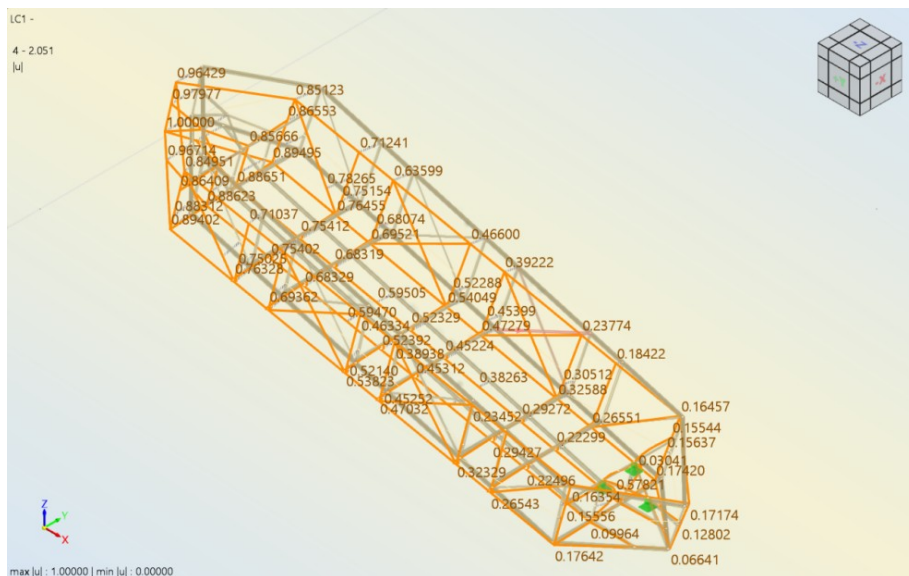
1.834



1.946



2.051



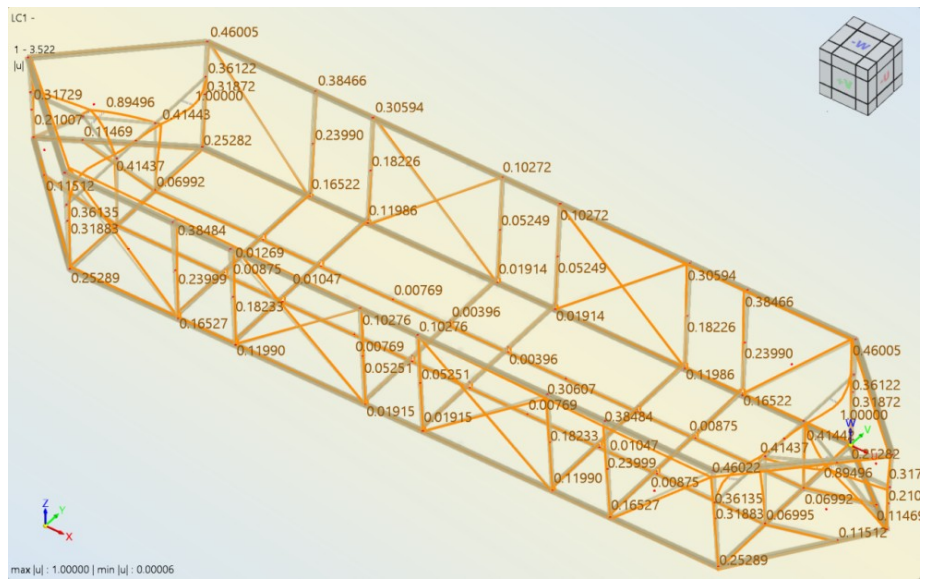
“建行杯”第十八届全国大学生结构设计竞赛总决赛理论方案

荷载
工况

临界荷载
系数

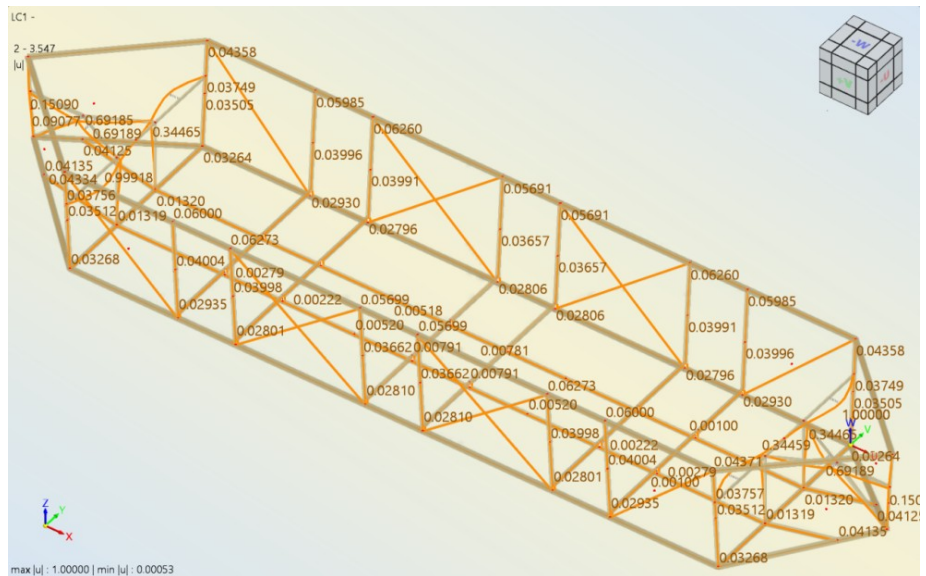
失稳模态

3.522

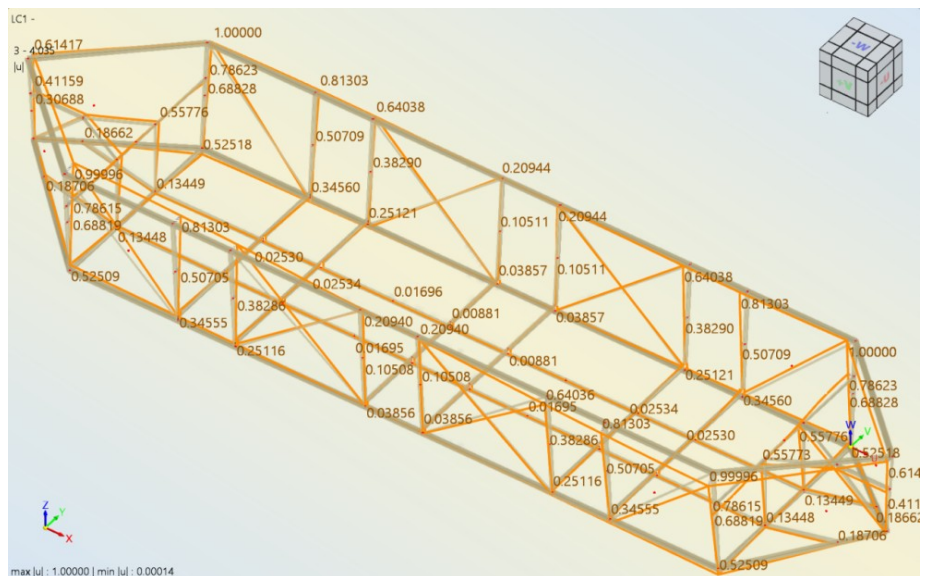


三级
荷载

3.547



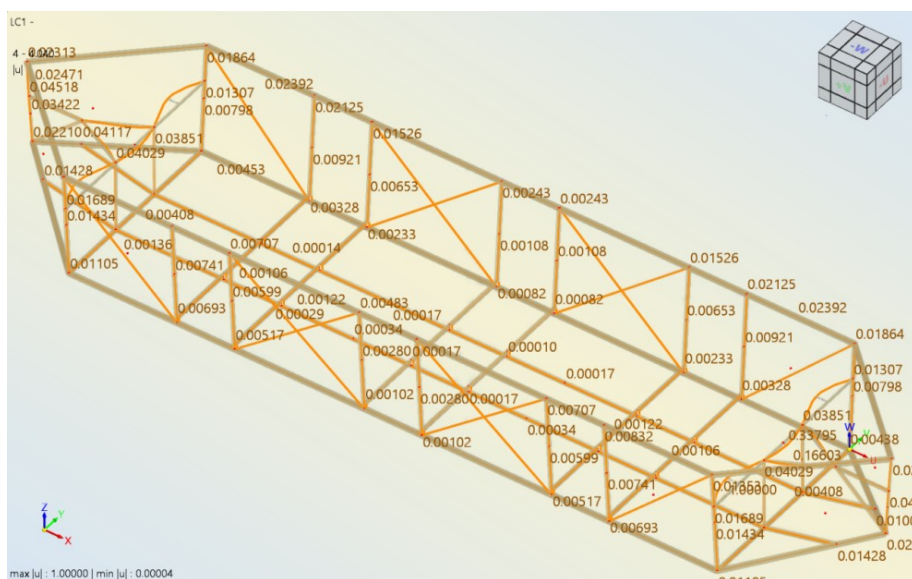
4.035



荷载 临界荷载
工况 系数

失稳模态

4.040



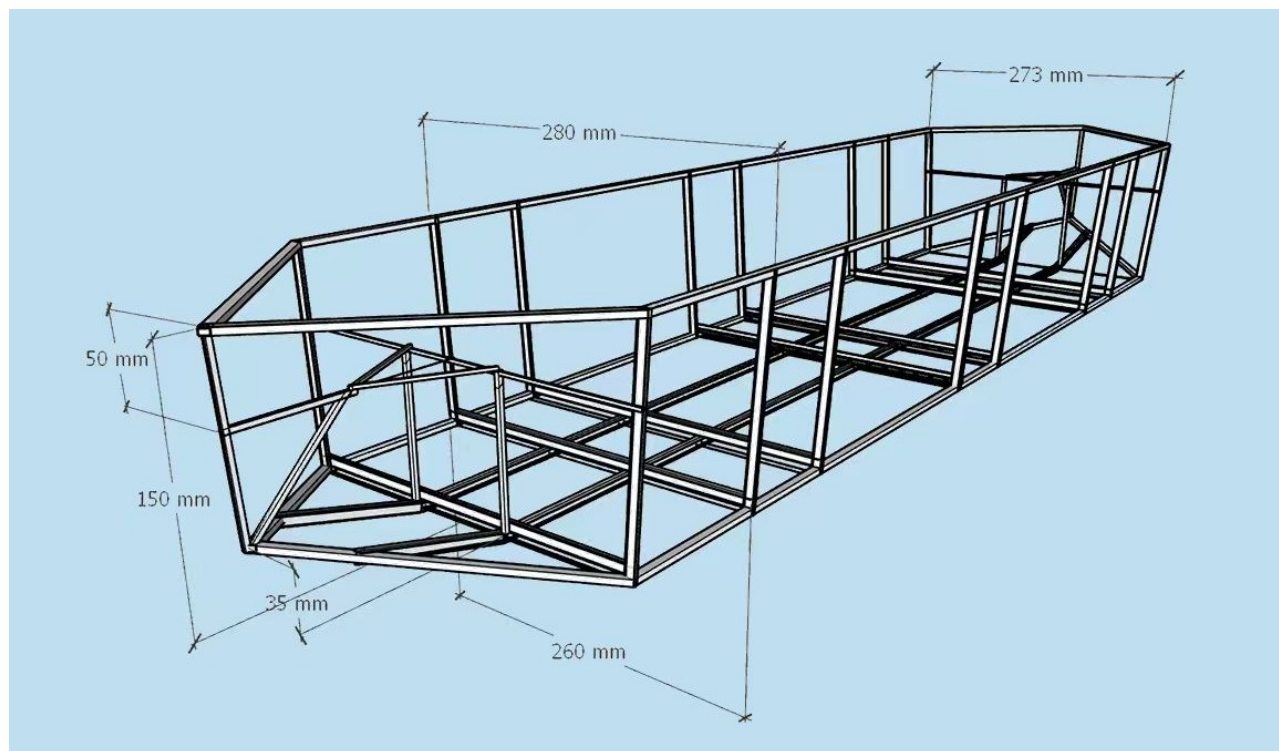
经过验算，各模态临界荷载系数均大于 1，模型可以通过稳定性验算。

4.4 小结

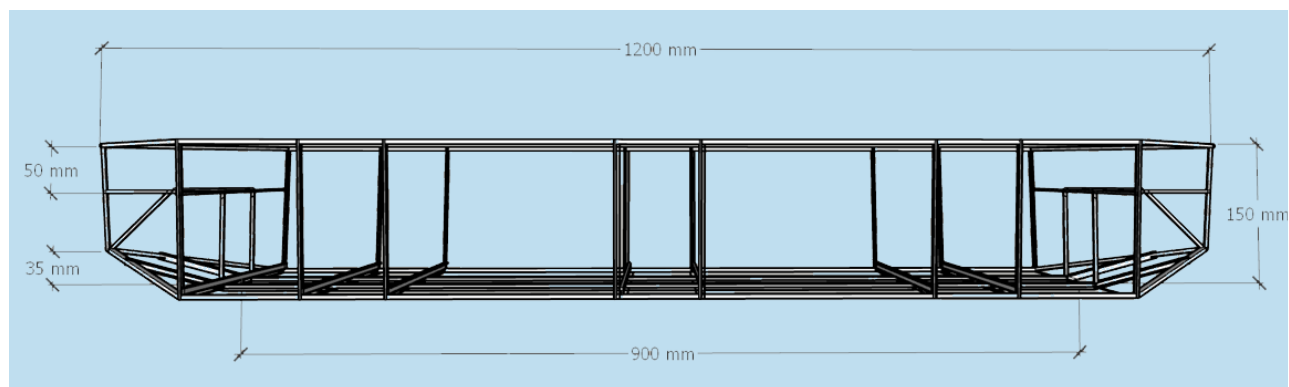
综合考虑三级荷载工况对模型所带来的各种效应，利用有限元分析和实验结合的方式，可以验证在最不利的工况下，结构的强度、刚度均满足工程要求，可以完成三级荷载。

5 模型尺寸图

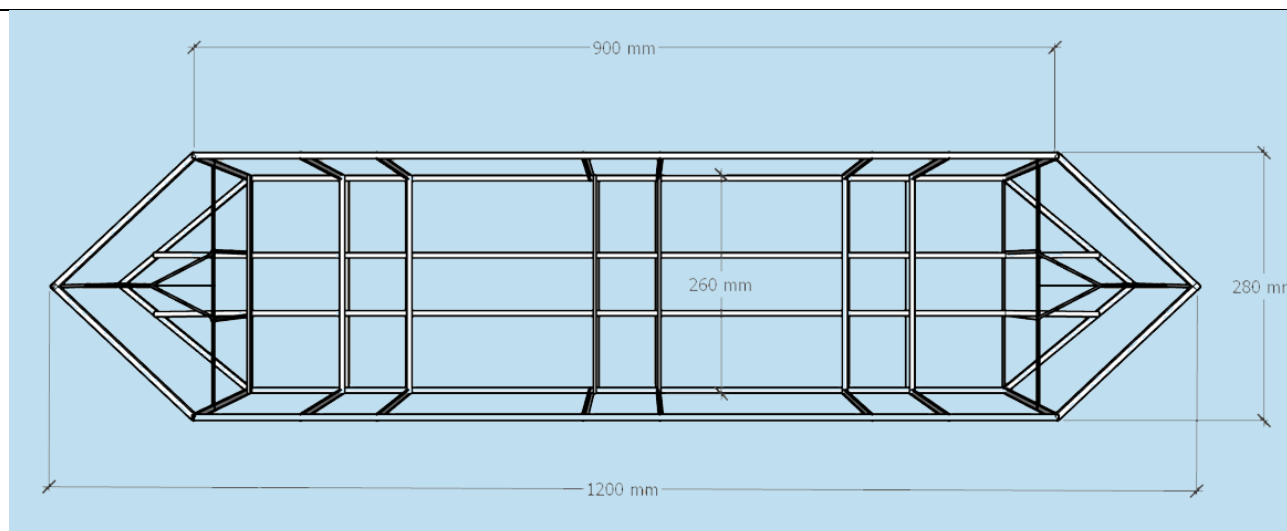
模型尺寸详见图 5-1。



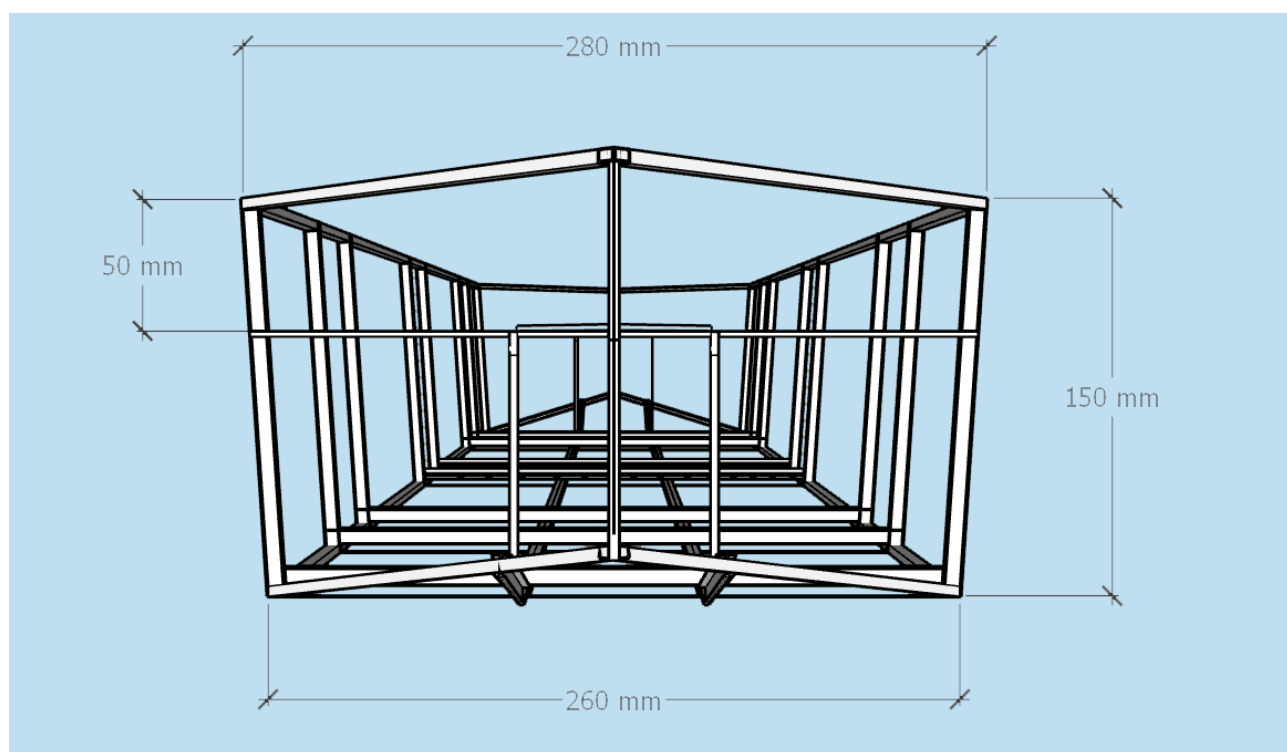
(a) 模型等轴视图



(b) 模型正立面图



(c) 模型轴俯视图



(d) 模型侧立视图

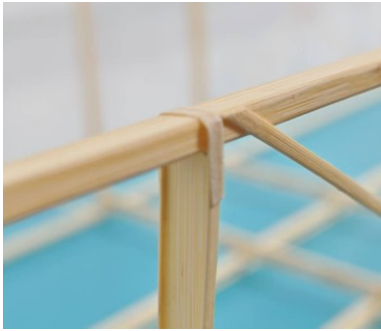
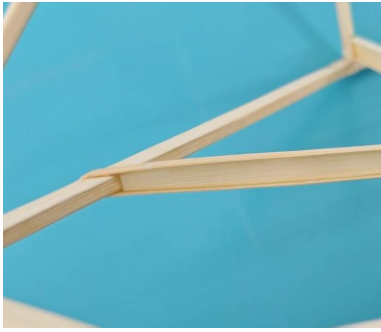
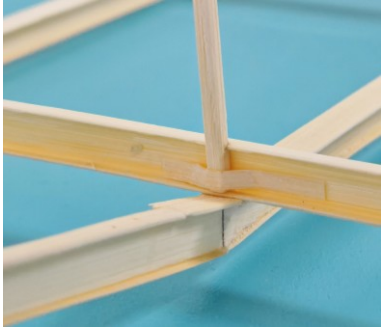
图 5-1 模型尺寸图

6 节点构造与施工

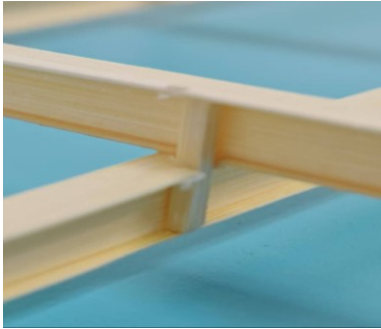
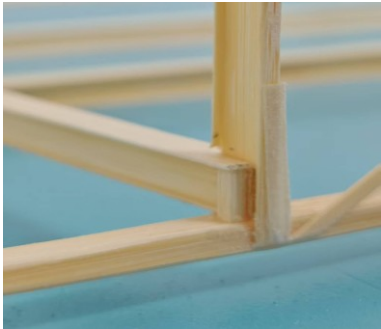
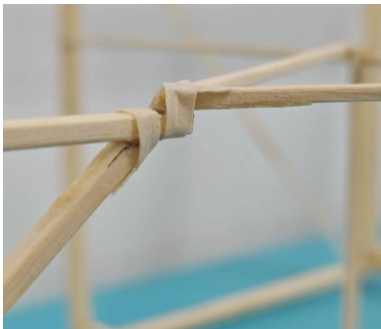
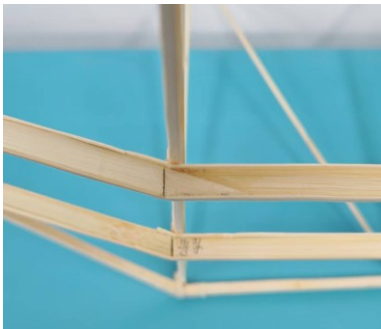
6.1 节点构造

节点连接是决定整体结构承载能力的重要因素之一。优良的节点不仅需具备足够的承载强度，还应兼顾轻量化。鉴于本模型采用桁架型骨架，且单一节点处粘连面积较小，无法采用简单的面贴式粘接。我们在节点构造上参考了若干成熟的桁架节点设计，并采用了多种互补性的加固方式：包括插接/榫卯式机械啮合、开槽与嵌插、腹板对腹板的贴合承压接触以及局部加劲板拼接，详见表 6-1。

表 6-1 主要节点参数表

编号	节点位置	说明	实物图	数量
J1	船舷工字杆与上下船弦链接处	将 0.2mm 竹皮裁成长方形细条状，按照顺纹方向进行半包围连接		32
J2	船底龙骨与下船弦链接处	采用插接的链接方式		4
J3	1.2 船头座位竖杆与工字杆连接处	将 0.2mm 竹皮裁成长方形细条状，按照顺纹方向进行半包围连接		4

“建行杯”第十八届全国大学生结构设计竞赛总决赛理论方案

编号	节点位置	说明	实物图	数量
J4	龙骨工字杆与横向承重工字杆连接处	将两根工字杆的翼缘部分开槽，插入 2*2 原杆使两腹板相粘合		24
J5	船舷工字杆与横向承重工字杆连接处	将两根工字杆的翼缘部分开槽，使两腹板直接粘合		16
J6	船头座位部分杆件连接处	采用搭接的粘连方式后将 0.2mm 竹皮裁成长方形细条状，按照顺纹方向进行包围加固链接		4
J7	船头龙骨转折处	采用腹板直接粘连的连接方式，增大粘连面积		4

6.2 组装工艺

主要受力的节点装配按“先粘合、后开槽、再插桩”的顺序进行。首先将两根工字杆的翼缘对齐并粘接，保持轴线一致。初固后，使用小锉刀在翼缘一侧沿腹板方向加工约 $3 \times 3 \text{ mm}$ 的矩形槽口，以保证槽位尺寸与小桩直径匹配。

随后，将加劲小桩插入槽内，使其与两侧工字杆的腹板贴合紧密，确保节点受力路径连续。防止工字杆在搭接处发生局部失稳。

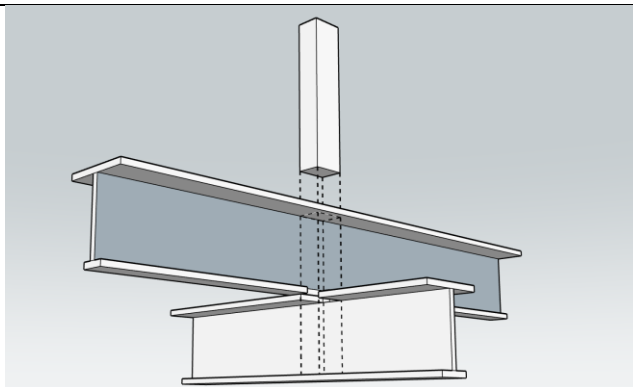


图 6-1 节点装配图

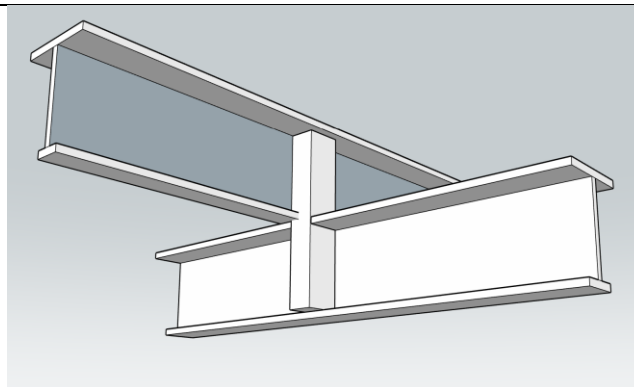


图 6-2 节点图

对于位于船体上弦与竖杆连接处的节点，竖杆的翼缘从两侧包覆方杆，使节点形成嵌合式连接结构，增加受力接触面积并提高抗弯刚度。同时，竖杆的腹板与方杆底面粘接，以形成连续的受力传递路径，增强节点的整体强度与连接稳定性。

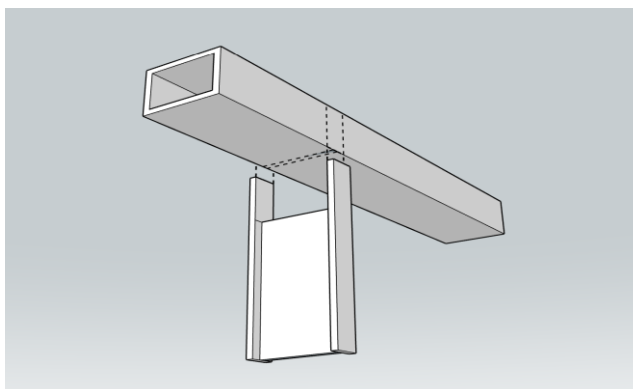


图 6-3 节点装配图

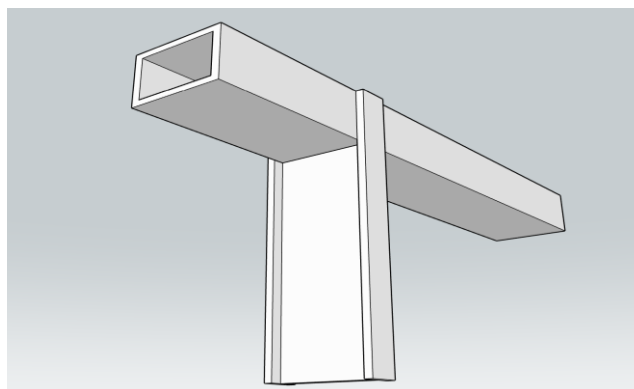


图 6-4 节点图

对于位于船舷的节点，采用两根相互垂直布置的工字杆。在粘接前，局部削除粘连区域的翼缘，使两腹板直接贴合，以增大有效粘接面积并提高节点的抗剪性能与整体刚度。同时，在另一侧增设加劲小桩，以防止工字杆失稳，进一步保证节点的稳定性。

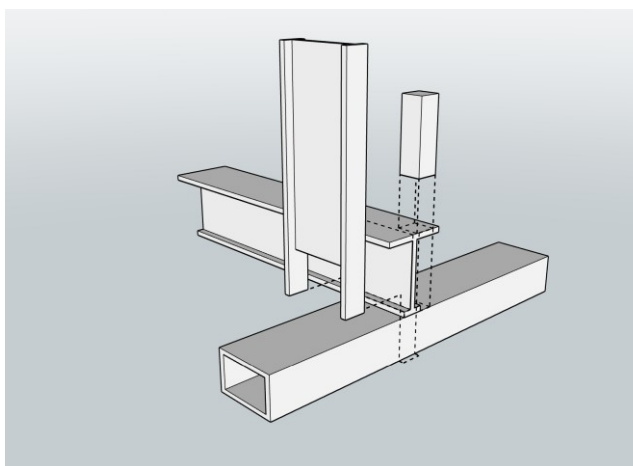


图 6-5 节点装配图

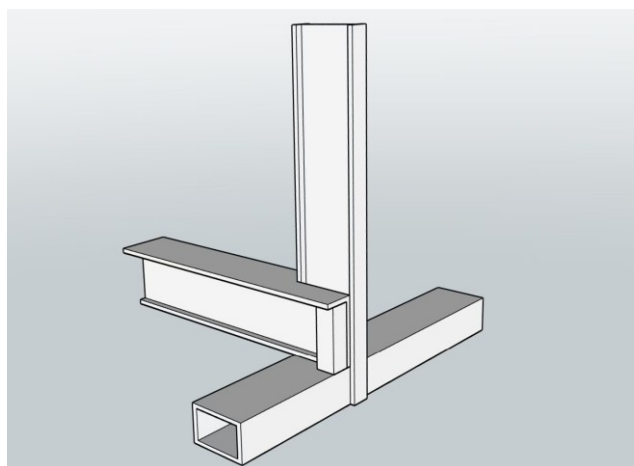


图 6-6 节点图

6.3 粘贴工艺

节点连接采用 502 瞬干胶进行粘接。粘贴过程中，表面处理与胶量控制是影响粘接质量的关键。所有粘接部位在操作前均经细砂纸打磨，确保竹材表面平整、无毛刺与浮尘，以提高胶液附着性能与结合强度。涂胶采用“点涂—薄涂”方式，胶量以均匀覆盖但不外溢为宜。过量用胶易导致节点打滑移位、固化后胶层脆弱并增加结构自重。制作时应严格控制胶量，先少量涂胶、快速对齐并轻压定位约 10 - 20 s 完成初固。对重要受力节点（如工字杆搭接、主龙骨连接等），可在初固后进行二次补胶以增强抗剪与抗剥性能。

第三部分：实训过程总结

7 参赛心得与体会

从四月组队结缘，到十月出征国赛，我们一路走来，经历了无数个日夜的打磨与坚持。这段漫长的备赛之路，几乎占据了我们整个学年。暑假、国庆假期，乃至每一个周末，我们都投入在模型设计、理论计算与试验验证中。起早贪黑成为常态，“结构竞赛”也成为我们这一年的关键词。

在校赛选拔之前，我们便主动走出校园，先后前往广州观摩省赛、上海观摩市赛，并赴上海交通大学学习交流经验。通过实地观摩不同高校的优秀作品与竞赛流程，我们深刻体会到结构设计竞赛的严谨与挑战，也逐渐明确了自身努力的方向。这些经历不仅开阔了我们的视野，更让我们意识到：结构设计的魅力在于将理论与实践相结合，让理性思考落地成形。

团队中有两位来自不同校区的成员，跨校区备赛带来了诸多不便。课程安排不同、时间难以统一、材料制作和试验环境也不在同一地点。但我们始终相互协调、共同克服困难，在反复的沟通与调整中建立了深厚的信任与默契。对于仍在学习基础力学知识的大一同学而言，这次竞赛不仅是一次挑战，更是一次快速成长的过程。

校赛阶段竞争激烈，每一支队伍都倾尽全力。我们深知自己并非起点最高的一组，却始终相信努力与坚持能弥补不足。无数个深夜，我们反复推敲结构计算、优化杆件布置、修改图纸模型。每一次加载成功都让我们欣喜若狂，每一次失败也让我们更坚定前行。

在整个备赛过程中，我们得到了来自老师和学长们无私而真挚的帮助。指导老师不仅在理论分析上给予悉心指导，更在方案优化、材料选择、工艺改进等细节上提供了宝贵建议。学长们也在百忙之中抽出时间，分享他们的比赛经验与技术要点，带领我们重新审视结构受力的本质。每一次加载结束，十几位老师与同学围坐在模型旁，围绕受力路径与节点布置展开热烈讨论——那一幕幕场景，成为我们记忆中最动人的画面。

从省赛到国赛，我们的备赛周期长达近一年。一路走来，我们经历过方案被否定的挫败，也迎来过加载成功的欢呼。过程艰辛，却充满意义。这场比赛，对我们而言，不只是一次竞技的经历，更是一场成长的洗礼。从陌生到默契，从迷茫到笃定，我们在一次次试验与推翻中，体会到了工程的严谨、协作的力量与坚持的意义。

回首这一路，我们深知“结构”的真正含义——它不仅是模型的骨架，更是我们团队信念的象征。未来的路仍然漫长，但我们相信，只要心中仍怀热爱，脚下的每一步都将成为通向更高峰的阶梯。

附录 A：电机数据解析代码

```
1. from time import sleep
2. import numpy as np
3. from enum import IntEnum
4. from struct import unpack
5. from struct import pack
6.
7.
8. class Motor:
9.     def __init__(self, MotorType, SlaveID, MasterID):
10.         """
11.         define Motor object 定义电机对象
12.         :param MotorType: Motor type 电机类型
13.         :param SlaveID: CANID 电机 ID
14.         :param MasterID: MasterID 主机 ID 建议不要设为 0
15.         """
16.         self.Pd = float(0)
17.         self.Vd = float(0)
18.         self.state_q = float(0)
19.         self.state_dq = float(0)
20.         self.state_tau = float(0)
21.         self.SlaveID = SlaveID
22.         self.MasterID = MasterID
23.         self.MotorType = MotorType
24.         self.isEnabled = False
25.         self.NowControlMode = Control_Type.MIT
26.         self.temp_param_dict = {}
27.
28.     def recv_data(self, q: float, dq: float, tau: float):
29.         self.state_q = q
30.         self.state_dq = dq
31.         self.state_tau = tau
32.
33.     def getPosition(self):
34.         """
35.         get the position of the motor 获取电机位置
36.         :return: the position of the motor 电机位置
37.         """
38.         return self.state_q
39.
40.     def getVelocity(self):
41.         """
42.         get the velocity of the motor 获取电机速度
```

```

43.         :return: the velocity of the motor 电机速度
44.         """
45.         return self.state_dq
46.
47.     def getTorque(self):
48.         """
49.         get the torque of the motor 获取电机力矩
50.         :return: the torque of the motor 电机力矩
51.         """
52.         return self.state_tau
53.
54.     def getParam(self, RID):
55.         """
56.         get the parameter of the motor 获取电机内部的参数，需要提前读取
57.         :param RID: DM_variable 电机参数
58.         :return: the parameter of the motor 电机参数
59.         """
60.         if RID in self.temp_param_dict:
61.             return self.temp_param_dict[RID]
62.         else:
63.             return None
64.
65.
66. class MotorControl:
67.     send_data_frame = np.array(
68.         [0x55, 0xAA, 0x1e, 0x03, 0x01, 0x00, 0x00, 0x00, 0x0a, 0x00, 0x00, 0x00,
69.          0x00, 0, 0, 0, 0, 0x00, 0x08, 0x00,
70.          0x00, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0x00], np.uint8)
71.     #           4310           4310_48           4340           4340_48
72.     Limit_Param = [[12.5, 30, 10], [12.5, 50, 10], [12.5, 8, 28], [12.5, 10, 28]
73. ,
74.                   # 6006           8006           8009           10010L           10
75.                   [12.5, 45, 20], [12.5, 45, 40], [12.5, 45, 54], [12.5, 25, 20
76. 0], [12.5, 20, 200],
77.                   # H3510           DMG6215           DMH6220
78.                   [12.5 , 280 , 1],[12.5 , 45 , 10],[12.5 , 45 , 10]]
79.
80.     def __init__(self, serial_device):
81.         """
82.         define MotorControl object 定义电机控制对象
83.         :param serial_device: serial object 串口对象
84.         """
85.         self.serial_ = serial_device

```

```

83.         self.motors_map = dict()
84.         self.data_save = bytes() # save data
85.         if self.serial_.is_open: # open the serial port
86.             print("Serial port is open")
87.             serial_device.close()
88.         self.serial_.open()
89.
90.     def controlMIT(self, DM_Motor, kp: float, kd: float, q: float, dq: float, tau
u: float):
91.         """
92.         MIT Control Mode Function 达妙电机 MIT 控制模式函数
93.         :param DM_Motor: Motor object 电机对象
94.         :param kp: kp
95.         :param kd: kd
96.         :param q: position 期望位置
97.         :param dq: velocity 期望速度
98.         :param tau: torque 期望力矩
99.         :return: None
100.        """
101.        if DM_Motor.SlaveID not in self.motors_map:
102.            print("controlMIT ERROR : Motor ID not found")
103.            return
104.        kp_uint = float_to_uint(kp, 0, 500, 12)
105.        kd_uint = float_to_uint(kd, 0, 5, 12)
106.        MotorType = DM_Motor.MotorType
107.        Q_MAX = self.Limit_Param[MotorType][0]
108.        DQ_MAX = self.Limit_Param[MotorType][1]
109.        TAU_MAX = self.Limit_Param[MotorType][2]
110.        q_uint = float_to_uint(q, -Q_MAX, Q_MAX, 16)
111.        dq_uint = float_to_uint(dq, -DQ_MAX, DQ_MAX, 12)
112.        tau_uint = float_to_uint(tau, -TAU_MAX, TAU_MAX, 12)
113.        data_buf = np.array([0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00], n
p.uint8)
114.        data_buf[0] = (q_uint >> 8) & 0xff
115.        data_buf[1] = q_uint & 0xff
116.        data_buf[2] = dq_uint >> 4
117.        data_buf[3] = ((dq_uint & 0xf) << 4) | ((kp_uint >> 8) & 0xf)
118.        data_buf[4] = kp_uint & 0xff
119.        data_buf[5] = kd_uint >> 4
120.        data_buf[6] = ((kd_uint & 0xf) << 4) | ((tau_uint >> 8) & 0xf)
121.        data_buf[7] = tau_uint & 0xff
122.        self.__send_data(DM_Motor.SlaveID, data_buf)
123.        self.recv() # receive the data from serial port
124.

```

```

125.     def control_delay(self, DM_Motor, kp: float, kd: float, q: float, dq: float
, tau: float, delay: float):
126.         """
127.         MIT Control Mode Function with delay 达妙电机 MIT 控制模式函数带延迟
128.         :param DM_Motor: Motor object 电机对象
129.         :param kp: kp
130.         :param kd: kd
131.         :param q: position 期望位置
132.         :param dq: velocity 期望速度
133.         :param tau: torque 期望力矩
134.         :param delay: delay time 延迟时间 单位秒
135.         """
136.         self.controlMIT(DM_Motor, kp, kd, q, dq, tau)
137.         sleep(delay)
138.
139.     def control_Pos_Vel(self, Motor, P_desired: float, V_desired: float):
140.         """
141.         control the motor in position and velocity control mode 电机位置速度控制
模式
142.         :param Motor: Motor object 电机对象
143.         :param P_desired: desired position 期望位置
144.         :param V_desired: desired velocity 期望速度
145.         :return: None
146.         """
147.         if Motor.SlaveID not in self.motors_map:
148.             print("Control Pos_Vel Error : Motor ID not found")
149.             return
150.         motorid = 0x100 + Motor.SlaveID
151.         data_buf = np.array([0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00], n
p.uint8)
152.         P_desired_uint8s = float_to_uint8s(P_desired)
153.         V_desired_uint8s = float_to_uint8s(V_desired)
154.         data_buf[0:4] = P_desired_uint8s
155.         data_buf[4:8] = V_desired_uint8s
156.         self.__send_data(motorid, data_buf)
157.         # time.sleep(0.001)
158.         self.recv() # receive the data from serial port
159.
160.     def control_Vel(self, Motor, Vel_desired):
161.         """
162.         control the motor in velocity control mode 电机速度控制模式
163.         :param Motor: Motor object 电机对象
164.         :param Vel_desired: desired velocity 期望速度
165.         """

```

```

166.         if Motor.SlaveID not in self.motors_map:
167.             print("control_VEL ERROR : Motor ID not found")
168.             return
169.         motorid = 0x200 + Motor.SlaveID
170.         data_buf = np.array([0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00], n
p.uint8)
171.         Vel_desired_uint8s = float_to_uint8s(Vel_desired)
172.         data_buf[0:4] = Vel_desired_uint8s
173.         self.__send_data(motorid, data_buf)
174.         self.recv() # receive the data from serial port
175.
176.     def control_pos_force(self, Motor, Pos_des: float, Vel_des, i_des):
177.         """
178.         control the motor in EMIT control mode 电机力位混合模式
179.         :param Pos_des: desired position rad 期望位置 单位为 rad
180.         :param Vel_des: desired velocity rad/s 期望速度 为放大 100 倍
181.         :param i_des: desired current rang 0-10000 期望电流标么值放大 10000 倍
182.         电流标么值: 实际电流值除以最大电流值, 最大电流见上电打印
183.         """
184.         if Motor.SlaveID not in self.motors_map:
185.             print("control_pos_vel ERROR : Motor ID not found")
186.             return
187.         motorid = 0x300 + Motor.SlaveID
188.         data_buf = np.array([0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00], n
p.uint8)
189.         Pos_desired_uint8s = float_to_uint8s(Pos_des)
190.         data_buf[0:4] = Pos_desired_uint8s
191.         Vel_uint = np.uint16(Vel_des)
192.         ides_uint = np.uint16(i_des)
193.         data_buf[4] = Vel_uint & 0xff
194.         data_buf[5] = Vel_uint >> 8
195.         data_buf[6] = ides_uint & 0xff
196.         data_buf[7] = ides_uint >> 8
197.         self.__send_data(motorid, data_buf)
198.         self.recv() # receive the data from serial port
199.
200.     def enable(self, Motor):
201.         """
202.         enable motor 使能电机
203.         最好在上电后几秒后再使能电机
204.         :param Motor: Motor object 电机对象
205.         """
206.         self.__control_cmd(Motor, np.uint8(0xFC))
207.         sleep(0.1)

```

```

208.         self.recv() # receive the data from serial port
209.
210.     def enable_old(self, Motor ,ControlMode):
211.         """
212.         enable motor old firmware 使能电机旧版本固件，这个是为了旧版本电机固件的兼容性
213.         可恶的旧版本固件使能需要加上偏移量
214.         最好在上电后几秒后再使能电机
215.         :param Motor: Motor object 电机对象
216.         """
217.         data_buf = np.array([0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc], np.uint8)
218.         enable_id = ((int(ControlMode)-1) << 2) + Motor.SlaveID
219.         self.__send_data(enable_id, data_buf)
220.         sleep(0.1)
221.         self.recv() # receive the data from serial port
222.
223.     def disable(self, Motor):
224.         """
225.         disable motor 失能电机
226.         :param Motor: Motor object 电机对象
227.         """
228.         self.__control_cmd(Motor, np.uint8(0xFD))
229.         sleep(0.01)
230.
231.     def set_zero_position(self, Motor):
232.         """
233.         set the zero position of the motor 设置电机 0 位
234.         :param Motor: Motor object 电机对象
235.         """
236.         self.__control_cmd(Motor, np.uint8(0xFE))
237.         sleep(0.1)
238.         self.recv() # receive the data from serial port
239.
240.     def recv(self):
241.         # 把上次没有解析完的剩下的也放进来
242.         data_recv = b''.join([self.data_save, self.serial_.read_all()])
243.         packets = self.__extract_packets(data_recv)
244.         for packet in packets:
245.             data = packet[7:15]
246.             CANID = (packet[6] << 24) | (packet[5] << 16) | (packet[4] << 8) | packet[3]
247.             CMD = packet[1]
248.             self.__process_packet(data, CANID, CMD)

```

```

249.
250.     def recv_set_param_data(self):
251.         data_recv = self.serial_.read_all()
252.         packets = self.__extract_packets(data_recv)
253.         for packet in packets:
254.             data = packet[7:15]
255.             CANID = (packet[6] << 24) | (packet[5] << 16) | (packet[4] << 8) |
packet[3]
256.             CMD = packet[1]
257.             self.__process_set_param_packet(data, CANID, CMD)
258.
259.     def __process_packet(self, data, CANID, CMD):
260.         if CMD == 0x11:
261.             if CANID != 0x00:
262.                 if CANID in self.motors_map:
263.                     q_uint = np.uint16((np.uint16(data[1]) << 8) | data[2])
264.                     dq_uint = np.uint16((np.uint16(data[3]) << 4) | (data[4] >>
4))
265.                     tau_uint = np.uint16(((data[4] & 0xf) << 8) | data[5])
266.                     MotorType_recv = self.motors_map[CANID].MotorType
267.                     Q_MAX = self.Limit_Param[MotorType_recv][0]
268.                     DQ_MAX = self.Limit_Param[MotorType_recv][1]
269.                     TAU_MAX = self.Limit_Param[MotorType_recv][2]
270.                     recv_q = uint_to_float(q_uint, -Q_MAX, Q_MAX, 16)
271.                     recv_dq = uint_to_float(dq_uint, -DQ_MAX, DQ_MAX, 12)
272.                     recv_tau = uint_to_float(tau_uint, -TAU_MAX, TAU_MAX, 12)
273.                     self.motors_map[CANID].recv_data(recv_q, recv_dq, recv_tau)
274.                 else:
275.                     MasterID=data[0] & 0x0f
276.                     if MasterID in self.motors_map:
277.                         q_uint = np.uint16((np.uint16(data[1]) << 8) | data[2])
278.                         dq_uint = np.uint16((np.uint16(data[3]) << 4) | (data[4] >>
4))
279.                         tau_uint = np.uint16(((data[4] & 0xf) << 8) | data[5])
280.                         MotorType_recv = self.motors_map[MasterID].MotorType
281.                         Q_MAX = self.Limit_Param[MotorType_recv][0]
282.                         DQ_MAX = self.Limit_Param[MotorType_recv][1]
283.                         TAU_MAX = self.Limit_Param[MotorType_recv][2]
284.                         recv_q = uint_to_float(q_uint, -Q_MAX, Q_MAX, 16)
285.                         recv_dq = uint_to_float(dq_uint, -DQ_MAX, DQ_MAX, 12)
286.                         recv_tau = uint_to_float(tau_uint, -TAU_MAX, TAU_MAX, 12)
287.                         self.motors_map[MasterID].recv_data(recv_q, recv_dq, recv_t
au)
288.

```

```

289.
290.     def __process_set_param_packet(self, data, CANID, CMD):
291.         if CMD == 0x11 and (data[2] == 0x33 or data[2] == 0x55):
292.             masterid=CANID
293.             slaveId = ((data[1] << 8) | data[0])
294.             if CANID==0x00: #防止有人把MasterID 设为0 稳一手
295.                 masterid=slaveId
296.
297.             if masterid not in self.motors_map:
298.                 if slaveId not in self.motors_map:
299.                     return
300.                 else:
301.                     masterid=slaveId
302.
303.             RID = data[3]
304.             # 读取参数得到的数据
305.             if is_in_ranges(RID):
306.                 #uint32 类型
307.                 num = uint8s_to_uint32(data[4], data[5], data[6], data[7])
308.                 self.motors_map[masterid].temp_param_dict[RID] = num
309.
310.             else:
311.                 #float 类型
312.                 num = uint8s_to_float(data[4], data[5], data[6], data[7])
313.                 self.motors_map[masterid].temp_param_dict[RID] = num
314.
315.
316.     def addMotor(self, Motor):
317.         """
318.         add motor to the motor control object 添加电机到电机控制对象
319.         :param Motor: Motor object 电机对象
320.         """
321.         self.motors_map[Motor.SlaveID] = Motor
322.         if Motor.MasterID != 0:
323.             self.motors_map[Motor.MasterID] = Motor
324.         return True
325.
326.     def __control_cmd(self, Motor, cmd: np.uint8):
327.         data_buf = np.array([0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, cmd], np
        .uint8)
328.         self.__send_data(Motor.SlaveID, data_buf)
329.
330.     def __send_data(self, motor_id, data):
331.         """
    
```

```

332.         send data to the motor 发送数据到电机
333.         :param motor_id:
334.         :param data:
335.         :return:
336.         """
337.         self.send_data_frame[13] = motor_id & 0xff
338.         self.send_data_frame[14] = (motor_id >> 8)& 0xff #id high 8 bits
339.         self.send_data_frame[21:29] = data
340.         self.serial_.write(bytes(self.send_data_frame.T))
341.
342.     def __read_RID_param(self, Motor, RID):
343.         can_id_l = Motor.SlaveID & 0xff #id low 8 bits
344.         can_id_h = (Motor.SlaveID >> 8)& 0xff #id high 8 bits
345.         data_buf = np.array([np.uint8(can_id_l), np.uint8(can_id_h), 0x33, np.u
346.         int8(RID), 0x00, 0x00, 0x00, 0x00], np.uint8)
347.         self.__send_data(0x7FF, data_buf)
348.
349.     def __write_motor_param(self, Motor, RID, data):
350.         can_id_l = Motor.SlaveID & 0xff #id low 8 bits
351.         can_id_h = (Motor.SlaveID >> 8)& 0xff #id high 8 bits
352.         data_buf = np.array([np.uint8(can_id_l), np.uint8(can_id_h), 0x55, np.u
353.         int8(RID), 0x00, 0x00, 0x00, 0x00], np.uint8)
354.         if not is_in_ranges(RID):
355.             # data is float
356.             data_buf[4:8] = float_to_uint8s(data)
357.         else:
358.             # data is int
359.             data_buf[4:8] = data_to_uint8s(int(data))
360.         self.__send_data(0x7FF, data_buf)
361.
362.     def switchControlMode(self, Motor, ControlMode):
363.         """
364.         switch the control mode of the motor 切换电机控制模式
365.         :param Motor: Motor object 电机对象
366.         :param ControlMode: Control_Type 电机控制模
367.         式 example:MIT:Control_Type.MIT MIT 模式
368.         """
369.         max_retries = 10
370.         retry_interval = 0.05 #retry times
371.         RID = 10
372.         self.__write_motor_param(Motor, RID, np.uint8(ControlMode))
373.         for _ in range(max_retries):
374.             sleep(retry_interval)
375.         self.recv_set_param_data()

```

```

373.         if Motor.SlaveID in self.motors_map:
374.             if RID in self.motors_map[Motor.SlaveID].temp_param_dict:
375.                 if self.motors_map[Motor.SlaveID].temp_param_dict[RID] == C
ontrolMode:
376.                     return True
377.                 else:
378.                     return False
379.         return False
380.
381.     def save_motor_param(self, Motor):
382.         """
383.         save the all parameter to flash 保存所有电机参数
384.         :param Motor: Motor object 电机对象
385.         :return:
386.         """
387.         can_id_l = Motor.SlaveID & 0xff #id low 8 bits
388.         can_id_h = (Motor.SlaveID >> 8) & 0xff #id high 8 bits
389.         data_buf = np.array([np.uint8(can_id_l), np.uint8(can_id_h), 0xAA, 0x00
, 0x00, 0x00, 0x00, 0x00], np.uint8)
390.         self.disable(Motor) # before save disable the motor
391.         self.__send_data(0x7FF, data_buf)
392.         sleep(0.001)
393.
394.     def change_limit_param(self, Motor_Type, PMAX, VMAX, TMAX):
395.         """
396.         change the PMAX VMAX TMAX of the motor 改变电机的 PMAX VMAX TMAX
397.         :param Motor_Type:
398.         :param PMAX: 电机的 PMAX
399.         :param VMAX: 电机的 VMAX
400.         :param TMAX: 电机的 TMAX
401.         :return:
402.         """
403.         self.Limit_Param[Motor_Type][0] = PMAX
404.         self.Limit_Param[Motor_Type][1] = VMAX
405.         self.Limit_Param[Motor_Type][2] = TMAX
406.
407.     def refresh_motor_status(self, Motor):
408.         """
409.         get the motor status 获得电机状态
410.         """
411.         can_id_l = Motor.SlaveID & 0xff #id low 8 bits
412.         can_id_h = (Motor.SlaveID >> 8) & 0xff #id high 8 bits
413.         data_buf = np.array([np.uint8(can_id_l), np.uint8(can_id_h), 0xCC, 0x00
, 0x00, 0x00, 0x00, 0x00], np.uint8)

```

“建行杯”第十八届全国大学生结构设计竞赛总决赛理论方案

```
414.         self.__send_data(0x7FF, data_buf)
415.         self.recv() # receive the data from serial port
416.
417.     def change_motor_param(self, Motor, RID, data):
418.         """
419.         change the RID of the motor 改变电机的参数
420.         :param Motor: Motor object 电机对象
421.         :param RID: DM_variable 电机参数
422.         :param data: 电机参数的值
423.         :return: True or False ,True means success, False means fail
424.         """
425.         max_retries = 20
426.         retry_interval = 0.05 #retry times
427.
428.         self.__write_motor_param(Motor, RID, data)
429.         for _ in range(max_retries):
430.             self.recv_set_param_data()
431.             if Motor.SlaveID in self.motors_map and RID in self.motors_map[Moto
r.SlaveID].temp_param_dict:
432.                 if abs(self.motors_map[Motor.SlaveID].temp_param_dict[RID] - da
ta) < 0.1:
433.                     return True
434.                 else:
435.                     return False
436.             sleep(retry_interval)
437.         return False
438.
439.     def read_motor_param(self, Motor, RID):
440.         """
441.         read only the RID of the motor 读取电机的内部信息例如 版本号等
442.         :param Motor: Motor object 电机对象
443.         :param RID: DM_variable 电机参数
444.         :return: 电机参数的值
445.         """
446.         max_retries = 20
447.         retry_interval = 0.05 #retry times
448.         self.__read_RID_param(Motor, RID)
449.         for _ in range(max_retries):
450.             sleep(retry_interval)
451.             self.recv_set_param_data()
452.             if Motor.SlaveID in self.motors_map:
453.                 if RID in self.motors_map[Motor.SlaveID].temp_param_dict:
454.                     return self.motors_map[Motor.SlaveID].temp_param_dict[RID]
455.             else:
```

```

456.             return None
457.         return None
458.
459.         # -----
460.         # Extract packets from the serial data
461.         def __extract_packets(self, data):
462.             frames = []
463.             header = 0xAA
464.             tail = 0x55
465.             frame_length = 16
466.             i = 0
467.             remainder_pos = 0
468.
469.             while i <= len(data) - frame_length:
470.                 if data[i] == header and data[i + frame_length - 1] == tail:
471.                     frame = data[i:i + frame_length]
472.                     frames.append(frame)
473.                     i += frame_length
474.                     remainder_pos = i
475.                 else:
476.                     i += 1
477.             self.data_save = data[remainder_pos:]
478.             return frames
479.
480.
481. def LIMIT_MIN_MAX(x, min, max):
482.     if x <= min:
483.         x = min
484.     elif x > max:
485.         x = max
486.
487.
488. def float_to_uint(x: float, x_min: float, x_max: float, bits):
489.     LIMIT_MIN_MAX(x, x_min, x_max)
490.     span = x_max - x_min
491.     data_norm = (x - x_min) / span
492.     return np.uint16(data_norm * ((1 << bits) - 1))
493.
494.
495. def uint_to_float(x: np.uint16, min: float, max: float, bits):
496.     span = max - min
497.     data_norm = float(x) / ((1 << bits) - 1)
498.     temp = data_norm * span + min
499.     return np.float32(temp)

```

```

500.
501.
502. def float_to_uint8s(value):
503.     # Pack the float into 4 bytes
504.     packed = pack('f', value)
505.     # Unpack the bytes into four uint8 values
506.     return unpack('4B', packed)
507.
508.
509. def data_to_uint8s(value):
510.     # Check if the value is within the range of uint32
511.     if isinstance(value, int) and (0 <= value <= 0xFFFFFFFF):
512.         # Pack the uint32 into 4 bytes
513.         packed = pack('I', value)
514.     else:
515.         raise ValueError("Value must be an integer within the range of uint32")
516.
517.     # Unpack the bytes into four uint8 values
518.     return unpack('4B', packed)
519.
520.
521. def is_in_ranges(number):
522.     """
523.     check if the number is in the range of uint32
524.     :param number:
525.     :return:
526.     """
527.     if (7 <= number <= 10) or (13 <= number <= 16) or (35 <= number <= 36):
528.         return True
529.     return False
530.
531.
532. def uint8s_to_uint32(byte1, byte2, byte3, byte4):
533.     # Pack the four uint8 values into a single uint32 value in little-
    endian order
534.     packed = pack('<4B', byte1, byte2, byte3, byte4)
535.     # Unpack the packed bytes into a uint32 value
536.     return unpack('<I', packed)[0]
537.
538.
539. def uint8s_to_float(byte1, byte2, byte3, byte4):
540.     # Pack the four uint8 values into a single float value in little-
    endian order
541.     packed = pack('<4B', byte1, byte2, byte3, byte4)

```

```

542.     # Unpack the packed bytes into a float value
543.     return unpack('<f', packed)[0]
544.
545.
546. def print_hex(data):
547.     hex_values = [f'{byte:02X}' for byte in data]
548.     print(' '.join(hex_values))
549.
550.
551. def get_enum_by_index(index, enum_class):
552.     try:
553.         return enum_class(index)
554.     except ValueError:
555.         return None
556.
557.
558. class DM_Motor_Type(IntEnum):
559.     DM4310 = 0
560.     DM4310_48V = 1
561.     DM4340 = 2
562.     DM4340_48V = 3
563.     DM6006 = 4
564.     DM8006 = 5
565.     DM8009 = 6
566.     DM10010L = 7
567.     DM10010 = 8
568.     DMH3510 = 9
569.     DMH6215 = 10
570.     DMG6220 = 11
571.
572.
573. class DM_variable(IntEnum):
574.     UV_Value = 0
575.     KT_Value = 1
576.     OT_Value = 2
577.     OC_Value = 3
578.     ACC = 4
579.     DEC = 5
580.     MAX_SPD = 6
581.     MST_ID = 7
582.     ESC_ID = 8
583.     TIMEOUT = 9
584.     CTRL_MODE = 10
585.     Damp = 11

```

```

586. Inertia = 12
587. hw_ver = 13
588. sw_ver = 14
589. SN = 15
590. NPP = 16
591. Rs = 17
592. LS = 18
593. Flux = 19
594. Gr = 20
595. PMAX = 21
596. VMAX = 22
597. TMAX = 23
598. I_BW = 24
599. KP_ASR = 25
600. KI_ASR = 26
601. KP_APR = 27
602. KI_APR = 28
603. OV_Value = 29
604. GREF = 30
605. Deta = 31
606. V_BW = 32
607. IQ_c1 = 33
608. VL_c1 = 34
609. can_br = 35
610. sub_ver = 36
611. u_off = 50
612. v_off = 51
613. k1 = 52
614. k2 = 53
615. m_off = 54
616. dir = 55
617. p_m = 80
618. xout = 81
619.
620.
621. class Control_Type(IntEnum):
622.     MIT = 1
623.     POS_VEL = 2
624.     VEL = 3
625.     Torque_Pos = 4
    
```

附录 B：电机试验代码

```
1. import math
2. from DM_CAN import *
3. import serial
4. import time
5.
6. # 创建加载数据文件
7. with open('7.txt', 'w') as f:
8.     f.write("Motor Data Log\n")
9.
10. Motor1=Motor(DM_Motor_Type.DM4340,0x05,0x15)
11. serial_device = serial.Serial('COM3', 921600, timeout=0.5)
12. MotorControl1=MotorControl(serial_device)
13. MotorControl1.addMotor(Motor1)
14.
15. # 切换速度控制模式
16. if MotorControl1.switchControlMode(Motor1,Control_Type.VEL):
17.     print("switch VEL success")
18.
19. # 电机自检输出
20. print("PMAx:",MotorControl1.read_motor_param(Motor1,DM_variable.PMAx))
21. print("MST_ID:",MotorControl1.read_motor_param(Motor1,DM_variable.MST_ID))
22. print("VMAx:",MotorControl1.read_motor_param(Motor1,DM_variable.VMAx))
23. print("TMAx:",MotorControl1.read_motor_param(Motor1,DM_variable.TMAx))
24.
25. MotorControl1.save_motor_param(Motor1)
26. # 电机使能
27. MotorControl1.enable(Motor1)
28. # 循环加载
29. i=0
30. while i<7000:
31.     q=math.sin(time.time())
32.     i=i+1
33.     MotorControl1.control_Vel(Motor1,3.6) #使电机以 3.6rad/s 的速度加载
34.     print("Motor1:", "POS:",Motor1.getPosition(), "VEL:",Motor1.getVelocity(), "TORQUE:",Motor1.getTorque())
35.     with open('7.txt', 'a') as f:
36.         f.write(f"Motor1: POS: {Motor1.getPosition()} VEL: {Motor1.getVelocity()} TORQUE: {Motor1.getTorque()}\n") #保存加载结果
37.     time.sleep(0.001)
38. MotorControl1.disable(Motor1)
39. # 关闭串口
40. serial_device.close()
```

附录 C：卡尔曼滤波代码

```
1. import numpy as np
2. import matplotlib.pyplot as plt
3. import pandas as pd
4.
5. class KalmanFilter:
6.     def __init__(self, process_variance=1e-5, measurement_variance=1e-2):
7.         """
8.         初始化卡尔曼滤波器
9.         process_variance: 过程噪声方差 (系统不确定性)
10.        measurement_variance: 测量噪声方差 (传感器噪声)
11.        """
12.        # 状态向量 [扭矩值, 扭矩变化率]
13.        self.state = np.zeros(2)
14.
15.        # 状态转移矩阵 (假设扭矩变化相对平滑)
16.        self.F = np.array([[1, 1],
17.                            [0, 1]])
18.
19.        # 观测矩阵 (我们只能观测到扭矩值)
20.        self.H = np.array([[1, 0]])
21.
22.        # 过程噪声协方差矩阵
23.        self.Q = np.array([[process_variance, 0],
24.                            [0, process_variance/10]])
25.
26.        # 测量噪声协方差
27.        self.R = np.array([[measurement_variance]])
28.
29.        # 误差协方差矩阵
30.        self.P = np.eye(2)
31.
32.    def predict(self):
33.        """预测步骤"""
34.        # 状态预测
35.        self.state = self.F @ self.state
36.        # 误差协方差预测
37.        self.P = self.F @ self.P @ self.F.T + self.Q
38.        return self.state[0] # 返回预测的扭矩值
39.
40.    def update(self, measurement):
41.        """更新步骤"""
42.        # 计算卡尔曼增益
```

```

43.         S = self.H @ self.P @ self.H.T + self.R
44.         K = self.P @ self.H.T @ np.linalg.inv(S)
45.
46.         # 状态更新
47.         y = measurement - self.H @ self.state # 残差
48.         self.state = self.state + K @ y
49.
50.         # 误差协方差更新
51.         I = np.eye(2)
52.         self.P = (I - K @ self.H) @ self.P
53.
54.         return self.state[0] # 返回更新后的扭矩值
55.
56.     def load_torque_data_from_file(filename):
57.         """
58.         从文件加载扭矩数据
59.         """
60.         torque_data = []
61.
62.         with open(filename, 'r') as f:
63.             for line in f:
64.                 if 'TORQUE:' in line:
65.                     # 提取TORQUE 后面的数值
66.                     parts = line.split('TORQUE:')
67.                     if len(parts) > 1:
68.                         torque_str = parts[1].strip()
69.                         # 提取第一个数值
70.                         torque_value = float(torque_str.split()[0])
71.                         torque_data.append(torque_value)
72.
73.         return np.array(torque_data)
74.
75.     def apply_kalman_filter_to_torque(data, process_var=1e-5, measurement_var=1e-
2):
76.         """
77.         对扭矩数据应用卡尔曼滤波
78.         data: 原始扭矩数据数组
79.         process_var: 过程噪声方差
80.         measurement_var: 测量噪声方差
81.         filtered_data: 滤波后的数据
82.         """
83.         kf = KalmanFilter(process_variance=process_var,
84.                             measurement_variance=measurement_var)
85.

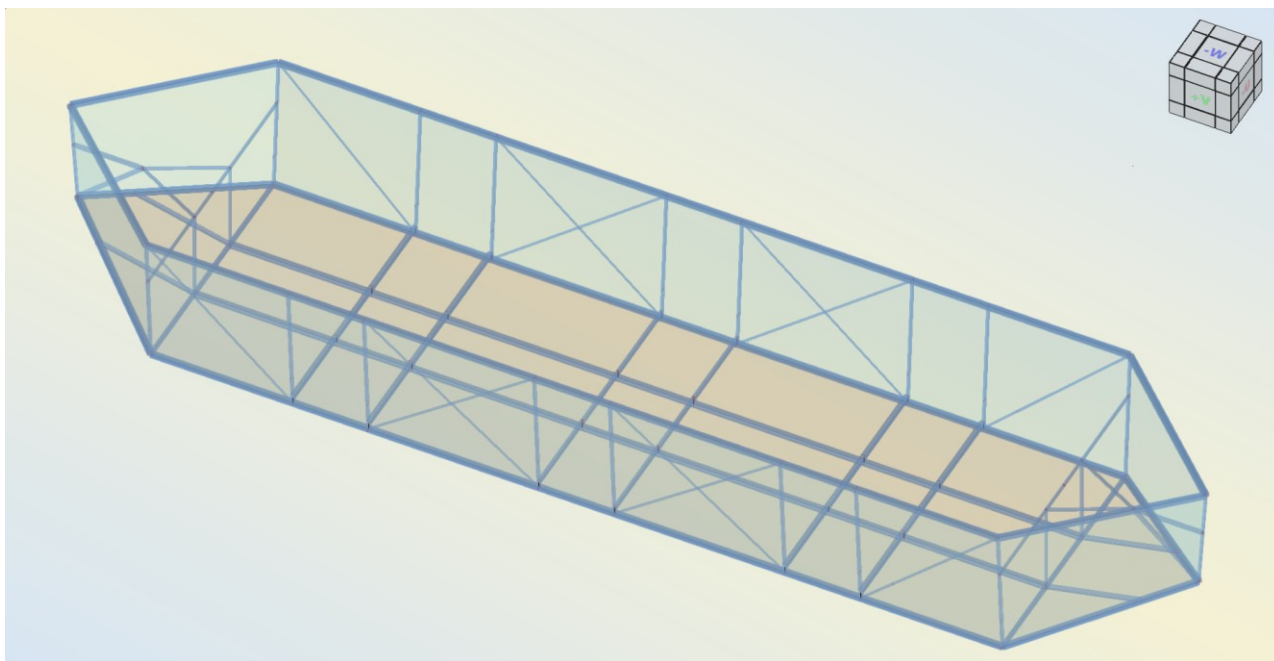
```

```
86.     filtered_data = []
87.
88.     for measurement in data:
89.         # 预测步骤
90.         kf.predict()
91.         # 更新步骤
92.         filtered_torque = kf.update(measurement)
93.         filtered_data.append(filtered_torque)
94.
95.     return np.array(filtered_data)
96.
97.
98. if __name__ == "__main__":
99.     # 方法1: 从文件加载数据
100.    filename = "12.txt" # 你的数据文件
101.    original_torque = load_torque_data_from_file(filename)
102.
103.    # 应用卡尔曼滤波
104.    filtered_torque = apply_kalman_filter_to_torque(original_torque)
105.
106.    print(f"滤波后统计: 均值={np.mean(filtered_torque):.4f}, 标准差
    ={np.std(filtered_torque):.4f}")
107.
108.    # 绘制结果
109.    plot_results(original_torque, filtered_torque, "电机扭矩数据卡尔曼滤波")
110.
111.    # 保存滤波后的数据
112.    np.savetxt('filtered_torque_data_12.txt', filtered_torque, fmt='%.6f')
113.    print("滤波后的数据已保存到 'filtered_torque_data.txt'")
```

作品总结

作品名称：神舟号

作品以“轻、快、稳”为核心设计理念，在外型选型阶段，通过大量缩尺实验确定了上翘船头与贯通长龙骨相结合的最优外形方案船体外型，可显著减小航行阻力。在结构设计部分，模型整体采用桁架体系，保证刚度的同时实现轻量化设计。模型内部横杆分层布置于纵向主杆中间及两侧，形成合理的受力分布。在节点处理方面，于主要受力节点插接加劲小桩，充分发挥杆件抗弯与抗剪性能，体现模型在实验验证与结构优化方面的创新。



模型示意图