

# “FAST”工作抛物面的优化设计

## 摘要

本文基于FAST的工作原理,通过机理分析、坐标变换、非线性最小二乘优化等方法,建立了反射面板调节优化模型,并利用BFGS算法、蒙特卡洛积分算法等算法,对不同条件下反射光线吸收比率进行了研究。

问题一中,首先基于固定的仰角观测目标S、圆心C和焦点P,利用旋转抛物面的中心对称性,选取焦距作为自由度控制变量,构建在极坐标系下开口竖直向上的二维抛物线方程,得到不同偏转角度下原点到抛物线的距离,进而导出三维下的旋转抛物面方程。其次,以焦距为决策变量,将口径300米的抛物面作为积分域,将理想抛物面到原点的距离与基准球面半径差值平方作为被积函数进行积分作为最小化目标函数,建立了确定理想抛物面的优化模型。最后,使用二分法求得目标函数导函数在定义区间上的零点,得到理想抛物面焦距的精确值为280.854,误差平方积分的最小值为10.1120,此时对应理想抛物面的解析式为 $z = (Q + \#)^2/561.708300.841$ 。

问题二中,首先利用球坐标下不同轴线方向抛物面的旋转不变性,在原坐标系和问题一的坐标系之间建立了双向可逆的变换关系,得到了不同方位角下理想抛物面到原点的距离。其次,以主索节点的工作坐标和促动器的伸缩长度为决策变量,以积分域覆盖的主索节点到原点的距离与理想抛物面到原点的距离之差的平方和为最小化目标函数,分别考虑下拉索长度固定、相邻节点的距离变化幅度不超过0.07%,促动器的伸缩范围在 $\pm 0.6\text{m}$ 为约束条件,建立反射面板调节优化模型。最后,使用拉格朗日乘子法和BFGS算法进行求解,得到误差平方在抛物面口径上的积分的最小值为 $5.1353 \times 10^9$ ,理想抛物线的顶点坐标为(-49.392, -36.943, -294.450),调节后反射面300米口径内的主索节点编号、位置坐标、各促动器的伸缩量等结果见文件result.xlsx。

问题三中,首先通过旋转变换,将反肘问题的倾斜入射光线转化为垂直入射光线。其次,通过求解线性方程组,依次确定出入射光线与三角形面板的相交判定式、交点坐标、三角形面板的法线向量,并利用光线垂直入射的性质,使用法线向最简化计算得到出射光线的方向角。再次,通过联立射线方程与馈源舱所在的目标高度,得出射线方程的步长和出射光到达目标高度时的坐标,并与馈源舱的有效区域进行比对,作为入射光线是否被有效接收的判定式。最后,将300米口径内实际接收区域作为积分域,将入射光线有效接收判别式作为被积函数,使用蒙特卡洛算法进行积分,得到有效接收的光源面积,并计算出调节后接收比0.811%,调节后接收比1.103%,提升了36%,调节工作抛物面的过程使有效光源的光斑可以尽量完整地出现在每个三角形面板内(图1)。

本文的特色在于将机理分析与非线性最小二乘优化相结合,并灵活采用二分法、BFGS算法和蒙特卡洛积分算法进行求解,在大规模、高维且带有非线性约束的求解中仍然以接近二阶的速度收敛至最优解,为FAST在不同情况下的调节与设计提供了参考依据。

关键词:坐标旋转、非线性最小二乘、BFGS算法、蒙特卡洛积分

## 一、问题重述

### 1.1 问题背景

中国天眼(FAST)由主动反射面、馈源舱及其他系统组成,其中主动反射面是由主索网、反射面板、下拉索、促动器及支承结构等构成的可调节球面。主索网由柔性主索按短程线三角网格方式构成,每个主索节点连接一根下端与固定在地表的促动器连接的下拉索。促动器沿基准球面径向安装,可沿径向伸缩。

主动反射面有两个状态：基准态时反射面为半径约 300 米、口径为 500 米的球面；工作状态时反射面为一个 300 米口径的近似旋转抛物面。馈源接收中心在与基准球面同心、半径差为一个球面上移动。当观测某个方向的目标 S 时，馈源接收中心被移动到直线 SC 与焦面的交点 P 处，调节基准球面上的反射板形成以直线 SC 为对称轴、以 P 为焦点的近似旋转抛物面。从而将来自 S 的平行电磁波汇聚到有效区域。

## 1.2 问题提出

在反射面板调节约束下，确定一个理想抛物面，调节促动器，使工作抛物面尽量贴近理想抛物面，以获得反射后的最佳接收效果。建立模型解决以下问题：

- 问题一：当待观测天体 S 位于基准球面正上方，即  $\theta = 0^\circ$ ， $\phi = 90^\circ$  时，结合考虑反射面板调节因素，确定理想抛物面。
- 问题二：当待观测天体 S 位于  $\theta = 36.765^\circ$ ， $\phi = 78.169^\circ$  时，确定理想抛物面。建立反射面调节模型，调节相关促动器的伸缩量，使反射面尽量贴近该理想抛物面 O。
- 问题三：基于问题二方案，计算调节后馈源能接收比，与基准球面接收比比作比较。

## 二、 问题假设

1. 假设光线在反射面不存在二次反射。
2. 假设不考虑板间间隙给反射带来的影响。
3. 假设电磁波在大气中的传播为直线传播，
4. 假设反射面板不会发生弯曲，始终维持一个平面。
5. 假设电磁波的反射为全反射，不考虑反射时的损耗。
6. 假设主索节点调节后，相邻节点之间的距离会发生微小变化，变化幅度不超过 0.07%。

## 三、 符号说明

序号	符号	意义	单位
1		抛物线及旋转抛物面的焦距	<i>m</i>
2		从原点出发到理想抛物面的距离	<i>m</i>
3	心	序号为 e 的促动器的伸缩量	<i>m</i>
4	乡	序号为 i 的下拉索的长度	<i>m</i>
5	<i>I</i>	所有主索节点的下标集合	—
6	<i>J</i>	一块反射面板所用到的下标集合	—
7	<i>E</i>	所有三角形的所有边的集合	—
8	( <i>o(i), t/o(i), z_o(i)</i> )	序号为 i 的主索节点的基准坐标	—
9	<i>Q(i), M(i), z(i)</i>	序号为 i 的主索节点的工作坐标	—
10	( <i>z(*), l(i), z(i)</i> )	序号为 i 的促动器底座的坐标	—
11	<i>GG(i), z(i)</i>	序号为 e 的促动器顶端的坐标	—
12	<i>3G(i), z°G(i)</i>	序号为 i 的促动器顶端基准状态的坐标	—
13	<i>S(i), sG(i), M(i)</i>	序号为 e 的主索节点旋转后的工作坐标	—

## 四、问题分析

### 4.1 问题一的分析

问题一要求当  $\alpha = 0^\circ, \beta = 90^\circ$  时, 结合反射面板调节因素, 确定理想抛物面。

基于  $\beta = \pi/2$ , 且观测目标  $S$ 、圆  $I$  心  $C$  和焦点  $P$  固定, 确定所求剖面抛物线的自由度只有 1。注意到旋转抛物面的中心对称性, 故所求抛物面是其剖面上的抛物线的复合。故电利用焦距作最优化控制变量, 进而构建在直角坐标系下开口竖直向上的二维抛物线方程, 将其代入极坐标进行变换, 可导出旋转抛物面的方程。最后对理想抛物面到原点的距离与基准球面半径的差值平方进行积分, 并进行最小二乘最优化求解, 使得基准球面与最终确定的抛物面之间的总调节距离的平方和最小, 从而求出理想抛物面。

### 4.2 问题二的分析

问题二要求当  $\alpha = 36.795^\circ, \beta = 78.169^\circ$  时, 确定理想抛物面。并建立反射面板调节模型, 调节相关促动器的伸缩量, 使反射面尽量贴近该理想抛物面。

注意到球坐标下不同轴线方向的抛物面为各同向性这一性质, 本文考虑在原有空间坐标系和问题一已建立的坐标系之间建立交互的旋转变换关系, 使得问题二中可部分沿用问题一中所得有关理想抛物面的数学关系, 进而可以确定本问的目标函数和决策变量, 然后引入下拉索固定长度、节点之间的伸缩变动率、促动器的伸缩量等约束, 再引入相邻节点的距源变化幅度的约束和促动器伸缩范围的约束, 求出相关促动器的伸缩量: 优化解。

### 4.3 问题三的分析

问题三要求基于问题二的反射面调节方案, 计算调节后馈源舱的接收比, 并与基准球面接收比作比较。

本文拟采用与问题二相同的坐标旋转方法, 将入射的光线从斜射变为垂直于水平面入射。接着, 本文拟通过构建以  $I$  为反射面板上三个主索节点坐标为基准的参数方程, 利用参数方程判断入射光与任意一块反射面板的几何关系, 从而确定交点坐标。然后, 本文拟利用待定系数法确定反射平面的法向量, 从而利用向量方向角的关系确定反射光的指向。然后, 可以利用方程联立求得的反射光与接收平面的交点判断该光线是否被有效接收。最终, 可将入射光的照射区域作为积分域, 是否接收到光线的判别式作为被积函数进行积分, 将积出的结果与照射面积做比值, 从而计算出馈源舱的接收比。

## 五、问题一模型的建立与求解

### 5.1 问题一模型的建立

#### 5.1.1 直角坐标系下二维抛物线方程的构建

题目要求给出考瑞反射面板调节因素下的理想抛物面。根据描述, FAST 的抛物面由二维抛物线旋转得来, 因此本文首先基于  $\beta = \pi/2$ , 讨论在二维平面内, 抛物线开口方向竖直向上的抛物线方程, 再通过旋转确定三维的旋转抛物面。

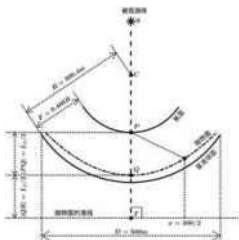


图 1: FAST 剖面示意图

本文基于题目附件 8 给出的空间直角坐标, 以 C 为坐标原点, 构建剖面图上的二维直角坐标系。则水平向右方向为  $x$  轴正向, 竖直向上为  $z$  轴正向。图 1 为 FAST 的剖面示意图, 其中抛物线的焦点为点 F 由于讨论的是开口方向竖直向上的抛物线, 故其焦点 F 的坐标为:

$$(S_z^F) = (0, -(B-F)) \quad (5-1)$$

根据抛物线的几何性质, 设该抛物线的焦距为  $f$  则焦点与顶点的距离为  $0.5L$  则该抛物线顶点。坐标为:

$$G_0, ZQ) = (0, -(R-F) - 0.5L) \quad (5-2)$$

由于抛物线焦点与准线的距离为一倍焦距, 故可得抛物线准线方程为:

$$z = -(12 - F) - L_f \quad (5-3)$$

根据抛物线的几何性质，可知抛物线上的任意一点到准线的距离等于该点到焦点的距离。利用该性质构建抛物线方程：

$$(Z - Z_T)^2 = x^2 + (Z - Z_P)^2 \quad (5-4)$$

其中， $Z_T = (R - F) - L_f$  为点 T 的 z 轴坐标，由准线方程(5-3)得出：

$Z_P = ?$  为点 P 的 z 轴坐标。

接着，将孙和给的值代入式(5-4)，可得：

$$[z + 0 - F] + L_f]^2 = x^2 + [z + (R - F)]^2 \quad (5-5)$$

两边平方项展开后，移项整理可得方程：

以上，得到了二维直角坐标系下，开口竖直向上的抛物线方程。

### 5.1.2 极坐标系下二维抛物线方程的构建

由于需要将计算得到的抛物面与球面进行对比，使得工作抛物面尽量贴近理想抛物面。因此为方便求解，将上述抛物线方程转换成极坐标形式。

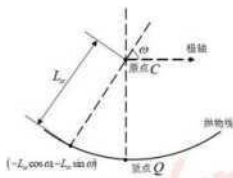


图 2: 极坐标系下抛物线示意图

如图 2,以 C 点为原点，以原 Z 轴为极轴，构建极坐标系，有：

$$(x, z) = (-L_0 \cos \theta, -L_0 \sin \theta) \quad (5-7)$$

其中， $L_0$  是从原点出发，到抛物线的距离。将式(5-7)代入直角坐标下的抛物线方程并整理可得：

$$\frac{L_0^2}{R} - L_0 \cos \theta + L_0 \sin \theta = 0 \quad (5-8)$$

为了得到不同偏转角度  $\theta$  与  $L_0$  的表达式，需要对式(5-8)的方程进行求解。

(1)当  $\theta > 0$  时，可将式(5-8)看作以  $L_0$  为未知数的一元二次方程，故有

$$A = \sin^2 u; + 4^{**} + \ll - F^{\wedge} > 0 \quad (5-9)$$

此时方程始终有两个不同的根。根据韦达定理：

$$L_3 I + L_5 = - \frac{-2L_f \sin \omega}{\cos^3 \omega} = \frac{2L_f \sin \omega}{\cos^3 \omega} > 0 \quad (5-10)$$

$$(5-11)$$

故可知，此时对于每一个  $\theta$  都会产生取值一正一负的两个根。舍去负根，得到正根的解析式：

$$-L/\sin \theta + \frac{L_f}{\cos^3 \omega} + (R - F) \cos^2 \theta = \cos^2 \theta \quad (5-12)$$

(2)当  $\theta = \theta'$  时，可将式(5-9)看作以  $\theta$  为未知量的  $\theta$  次方程，故有：

$$L_{\omega} = \frac{L_f}{2 \sin \omega} + \sin^3 \theta \quad (5-13)$$

经过化简，可得基于偏转角度  $\theta$  与心的表达式：

$$L_{\omega} = \begin{cases} L/\sin \theta + X/\sin \theta + \frac{L_f}{\cos^3 \omega} + (R - F) \cos^2 \theta \\ 5' \\ \frac{L_f}{2 \sin \omega} + \sin^3 \theta \end{cases} \quad (5-14)$$

由于公式(5.14)中含有未知参数，则将该函数记为：

$$f(\theta) = \frac{L_f}{2 \sin \omega} + \sin^3 \theta + (R - F) \cos^2 \theta \quad (5-15)$$

以上，得到了在不同的偏转角度下，二维抛物线上的点到原点的距离  $L$  的表达式，接洽将此二维抛物线进行旋转即可得到三维的旋转抛物面。

### 5.1.3 理想抛物面与基准球面的相似度计算

将 5.1.2 节得到的二维抛物线以极轴为中轴线旋转，得到反射面板的旋转抛物面，该抛物面可使得沿着中轴线平行射入的电磁波可以被反射到焦点 P 处。为比较这一理想抛物面与基准球面的相似程度，需要在球坐标系下，以口径为 300 米的抛物面在基准球面上的投影所围成的区域为积分域，对理想抛物面到原点的距离与基准球面半径的差值平方进行积分，即：

$$EA \int_{\Omega} (z - \sqrt{R^2 - x^2 - y^2})^2 dx dy \quad (5-16)$$

积分数值的大小是评价相似程度的准则。其中，R为基准球面半径；z为抛物面上点与原点的连线和中垂线所形成的空间角的大小；db为曲面微元；Ω为积分域，可表示如下：

$$\Omega = \{(x, y, z) | z = \sqrt{R^2 - x^2 - y^2}, x^2 + y^2 \leq R^2\} \quad (5-17)$$

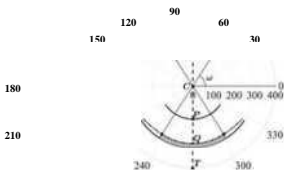


图 3: 极坐标系下 FAST 剖面及积分区域示赢图

由于抛物面的中轴线是竖直的，所以在球坐标系下可转化为二重积分。故该积分式可化简为：

$$\int_{\Omega} (z - \sqrt{R^2 - r_s^2})^2 r_s dr_s d\theta \quad (5-18)$$

其中，θ为方位角；φ为仰角； $r_s = R \cos \phi$ 是在φ角给定时，θ角在基准球面上的轨迹投影出的圆形的半径。根据题目要求的工作口径，可知，θ的取值范围满足：

$$|\theta - \theta_0| = 150^\circ \quad (5-19)$$

根据几何关系，易知式(5-19)有两个解θ<sub>1</sub>和θ<sub>2</sub>，有θ<sub>1</sub>和θ<sub>2</sub>这两个解的均值为θ<sub>0</sub>。为防止对θ指向的圆环垂叠积分，可设θ的取值范围为θ<sub>0</sub>±150°。结合上述分析，对式(5-18)进行逐步化简得：

$$2 \int_{\Omega} (z - \sqrt{R^2 - r_s^2})^2 r_s dr_s \quad (5-20)$$

以上，得到了球坐标系下计算理想抛物面和基准球面之间相似程度的准则“

#### 5.1.4 理想抛物面优化模型的建立

由于在现有约束条件下，可求解出多个不同焦距的理想抛物面，故需要对于理想抛物面进行优化的选取。故构建目标函数如下：

$$L_f^* = \operatorname{argmin}_{L_f} \int_{R_m}^{R^*/2} 2^{L_f/(\wedge E)} \cdot R^2 dO \quad (5-21)$$

结合式(5-6).确定形成咬优理想抛物面的抛物线方程为:

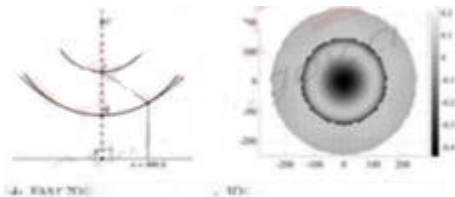
$$(5-22)$$

## 5.2 问题一模型的求解

### 5.2.1 算法设计

首先对目标函数(5-20)的数值特性进行定性分析,可知焦距与决定了理想抛物面和基准球面之间的相似程度,焦距过大·则理想抛物面会整体低于基准球面;焦距过小,则理想抛物面会整体高于基准球面。易知目标函数的梯度函数在区间内为一个单调连续函数,且区间两端对应的梯度值异号,即目标函数(5-20)&区间(100, 400)±是一个存在极小值点的凸函数。因此,本文基于该性质设计针对梯度函数零点的二分查找法对极小值点进行求解。具体求解算法如下: 算法 1: 二分查找第△

- 1: 选定边界值。首先根据题目设定与实际情况,选取 100 与 400 作为二分查找区间的左右端点。
- 2: 取二分查找区间的中点坐标代入公式(5-23)的梯度函数中,计算得到该点所对应的梯度值:



- 3: 若梯度值远大于 0,将二分查找区间的右·端点设置为当前中点,并返回第 2 行。
- 4: 若梯度值远小于 0,将二分查找区间的左端点设置为当前中点,并返回第 2 行。
- 5: 若梯度值的绝对值小于机器精度 epsilon,则中点即为待求解的极小值点,程序结束。

### 5.2.2 理想抛物面的求解与分析

基于算法 1.在 MATLAB 中编程进行求解,给出求解出的理想抛物面的剖面视图(图 4 左)及理想抛物面与基准球面的径向高度高低差(图 4 右):

图 4: FAST 2D fit 面最优抛物线(左)、3D 最优抛物面径向高度高低差(右)



最终结果求解出理想抛物面的焦距的精确值为 280.854, 误差平方在口径上的积分的最小值为 10.112. 将  $\rho = 280.854$ ,  $R = 300.4$ ,  $F = 0.466R$  带入式(5-6)可得剖切平面上的抛物线方程为:

$$z = 0.000708x^2 - 300.841 \quad (5-23)$$

将抛物线绕 z 轴旋转一周后, 可知其旋转抛物面的方程为:

$$Z = 561.708(Z + y)^2 - 300.841 \quad (5-24)$$

### 5.2.3 结果检验

基于 5.2.2 节中的精确结果, 对所得理想抛物面的剖面上的径向高度与基准球面半径的高低差进行检验, 可得最优抛物面与基准球面的偏移量:

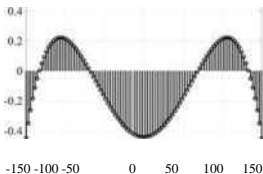


图 5: 最优抛物面与基准球面的偏移量

由图 5 可知, 最优抛物面与基准球面的偏移量在  $(-0.6, +0.4)$  的范围内。因此此最优抛物面在实际应用中, 促动器顶端上下拉动的拉索的长度范围也较小。结合反射板调节效率因素, 可得本结果合理且较优。

## 六、问题二模型的建立与求解

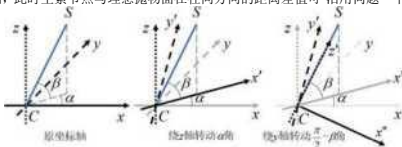
### 6.1 问题二模型的建立

问题二要求基于问题一所建立的模型, 求解当  $\theta = 36.795^\circ$ ,  $\theta = 78.169^\circ$  时的理想抛物面, 并以工作抛物面尽量贴近理想抛物面为目标。在下拉索固定长度、节点之间的伸缩变动率、促动器的伸缩量: 约束下, 优化相关促动器的伸缩量。

为简化计算, 本文首先建立在标准坐标系上旋转得到的新坐标系, 使得主索节点与理想抛物面在径向方向的距离差值可沿用问题一中的式(5-15)。从而构建本问的目标函数。然后通过分析促动器底端坐标、基准状态与工作状态的促动器顶端坐标、主索节点的基准坐标与工作坐标以及促动器伸缩量: 等变之间的几何关系, 构建本问题的约束条件, 得到反射面板调节优化模型。最后, 设计 BFGS 算法进行模型求解。

### 6.1.1 基于坐标系旋转的理想抛物面与反射面板相似度计算

本题中，由于观测天体 S 的方位对于基准球心 C 存在与竖直方向的倾角，这给构建新的理想抛物面带来了一定的困难。但在求解理想抛物面时，对于不同轴线方向的抛物面来说，其焦距、焦点等性质都是不变的，故将现有空间坐标系，以基准球面的球心 C 为中心、沿基准球面的剖面方向进行旋转，使得问题二中理想抛物面的轴线与旋转后空间坐标系 z 轴所在 H 线重合，此时主索节点与理想抛物面在径向方向的距离差值可沿用问题一中的式(5-15)。



设第 i 个主索节点的工作坐标为  $(z(i), y(i), zG(i))$ 。将原先的坐标轴绕看 z 轴沿正方向旋转角度 (正向旋转方向与坐标轴指向满足右手定则，下文同义)，则坐标相对于坐标系绕 z 轴反向旋转角度再将此时的坐标轴绕着 y 轴沿正方向旋转角度  $\pi/2$  (3, 则坐标相对于坐标系绕 z 轴反向旋转 0、根据三维坐标中旋转矩阵的定义，可旋转坐标系下，第 i 个主索节点的工作坐标 (砲(., 如 %G)) 为:

$$\begin{pmatrix} x_w(i) \\ y_w(i) \\ z_w(i) \end{pmatrix} = \begin{pmatrix} \cos\left(\frac{\pi}{2} - \beta\right) & 0 & -\sin\left(\frac{\pi}{2} - \beta\right) \\ 0 & 1 & 0 \\ \sin\left(\frac{\pi}{2} - \beta\right) & 0 & \cos\left(\frac{\pi}{2} - \beta\right) \end{pmatrix} \begin{pmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x(i) \\ y(i) \\ z(i) \end{pmatrix} \quad (6-1)$$

如图 6, 不难看出旋转后坐标系的 z 轴正方向指向被观测体球心 C 为原点，在新的空间直角坐标系上建立空间极坐标系。再将变换后的主索节点的直角坐标转化为新的空间极坐标系下的坐标，得到第 i 个主索节点的坐标为  $(L(i)^{\wedge}(i), ft(i))$ , 具体形式为:

$$1(\infty) / \text{珀 } 6)2 + \text{加}^n + *(\text{序}$$

BG)

。且

接希，根据问题•中理想抛物面上偏转角度 3 与儿的衣达式(5-15), 得到在新的坐

标系中,理想抛物面上仰角为PG)的点与原点。的距歷:

$$i \cdot SMG) \langle \text{餃十 } 2(R F\%88" G) \text{ 严} -, \text{院蒲錦'句G)"。} \rangle 2 \quad (6-3)$$

为保证反射面在工作状态下与理想抛物面尽量贴合,需要使得在同一径向方向上的主索节点与理想抛物面上的点距磯原点。的距离差尽量小。故可构建如下优化目标:

(\*) $(y^*(\alpha), 2^n(0) \operatorname{argmin} \xi$  (知6)-乙。尸 (4) 其中 $\bullet Z$ 为所有主索节点所用到的下标集合: ①, 为新的极坐标下仰角收值的最小值。S 于理想抛物面具有照明X域的限制 5.1.3 节中已经据此给出了仰角的取值范围式。这在 本问题中也同样具有约束效果。注意到, 在基于促动器伸缩完成的主索节点位租变化过 程中, 第 i 个主索节点的坐标(伽(, ), 伽<(£)孫巧 6))可以由促动器的伸缩欣唯一确定。根 据附件 2 中数据, 可知促动器底端的坐标(z(t), y(t), z(i))与促动器顶端基准状态下的坐标 G°(i), y°G, zy)) · 设促动器在工作状态时的顶端坐标为(x(: ), !/(\*) , z(i)) 由于促动器底端与促动器顶端始终在一·条直线上, 故存在比例关系:

$$Z^{\circ}G) X(0) \frac{z(i) \cdot y(i)}{y(i) \cdot z(i)} = \frac{z(i) \cdot y(i)}{y(i) \cdot z(i)} \quad (6-5)$$

其中 $\bullet LG$ )为每个促动器的伸缩抵。同样地, y 轴和 Z 轴方向均同在相同的比例关系, 因此有:

$$\begin{aligned} Z(i) &= \frac{z(i) \cdot y(i)}{y(i) \cdot z(i)} \cdot z^{\circ}G) \\ &= \frac{z(i) \cdot y(i)}{y(i) \cdot z(i)} \cdot z^{\circ}G) \end{aligned} \quad (6-6)$$

以上, 经过推导最终得到了第 3 个促动器的伸缩量&G)与对应主索节点的坐标(zwG), ywG, zwG))之间的关系。因此加入 LG)作为式(6.4)中反射面与理想抛物面相 似度计算目标函数的决策变量, 整理可得:

$$= \operatorname{argmin} \xi \&G) \text{一珀} \quad (6-7)$$

### 6.1.2 约束条件的确定

#### (1) 节点之间的伸缩变动率约束

根据假设 6, 相邻节点之间的距离可能会发生微小变化, 变化幅度的极限为 0.07%, 故可构建如下约束条件:

$$\begin{aligned} & [珀。 , (\沛'-3\textcircled{g}) \text{己}[z\textcircled{z} (h) ]^{\circ} f \\ M) \text{与} (\text{切} \cdot [voGi) \text{如} (\text{冲} \cdot Mid MF ( \text{M} \cdot E \\ WO, (\text{曲} \text{J} 3 (\text{助} \text{讯} \text{可} \text{卜} \cdot [z\textcircled{z} (测 2 \text{M} \cdot F \\ \% (\text{冲} \text{砲} \text{蚰} \text{尸} \cdot Mk) \% (\text{湖} Q \cdot 0.07\%) , (\text{m} E \end{aligned}$$

其中,  $F=(h, i_2) \in Z \times Z \setminus c$

## (2) 促动器伸缩范围约束

根据题设, 促动器伸缩量只能在 0.6m 到+0.6m 之间, 故具有如下约束条件:

$$-0.6 \leq \Delta L_i \leq 0.6 \quad (6-9)$$

## (3) 下拉索长度约束

主索节点与促动器顶端是由固定长度的下拉索连接的, 可通过基准状态下的主索节点的坐标与促动器顶端基准坐标, 计算出每一段下拉索的长度。设每一段下拉索的长度为  $L_i$ , 基准状态下主索节点切标为  $(S_0, L_0(i), Z_0(i))$ , 促动器顶端基准坐标为  $(G_0, L_0, Z_0)$ , 则有:

$$L_i = \sqrt{(S_0 - G_0)^2 + (L_0(i) - L_0)^2 + (Z_0(i) - Z_0)^2} \quad (6-10)$$

设促动器在工作状态时的顶端坐标为  $(X(i), y(i), z(i))$ , 则存在如下约束条件:

$$X(i)^2 + y(i)^2 + z(i)^2 - L_i^2 = 0 \quad (6-11)$$

### 6.1.3 反射面板调节优化模型的构建

综上所述, 本文建立的反射面板调节优化模型为:

$$\begin{aligned} & \text{目标函数: } \min F(\mathbf{x}) = \sum_{i=1}^n \left( \frac{L_i}{L_{i0}} - 1 \right)^2 \\ & \text{约束条件: } \begin{cases} X(i)^2 + y(i)^2 + z(i)^2 - L_i^2 = 0 \\ -0.6 \leq \Delta L_i \leq 0.6 \\ L_i \geq 0 \end{cases} \end{aligned}$$

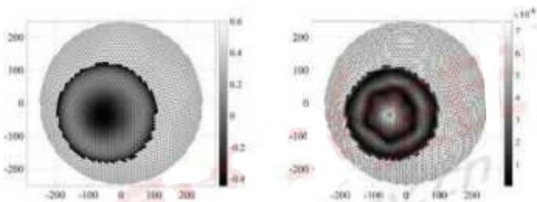
### 6.2.1 算法设计

由于问题二共计需要求解 2226 个主索节点的  $(x, y, z)$  坐标, 以及其对应的促动器伸缩量;  $L_i$ , 需要求解的决策变量数量: 庞大且其中带有非线性的等式和不等式约束, 因此必须寻找适应于高维大规模问题求解的算法。

对于模型中非线性约束, 首先基于拉格朗日乘子法思想, 引入一个乘数  $\lambda$ , 将等式非线性约束转化为目标函数的一部分。接着, 在不考虑不等式约束的情况下进行求解。当结果违反不等式约束时, 将其转化为等式约束并再此求解, 直到得到不违反约束的最优解。注意到待求解的模型中的非线性约束都能表示为决策变量的二次型, 具有光滑和凸的性质, 且容许在数值定义域上轻微违反约束时稳定求值, 因此此处使用拟牛顿方法中的 BFGS 方法, 以在不直接计算 Hessian 矩阵的情况下实现这...高维问题的超线性收敛。具体求解算法如下:

#### 算法 2: BFGS 算法

- 1: 初始化参数。为循环变量赋初始值  $k=1$ , 为切线刚度矩阵  $B_k$  赋初始值为单位矩阵, 设置初始解  $\mathbf{x}_k$  与主索网节点在附件中的基准坐标。
- 2: 循环: 根据条件  $\|B_k^{-1} \nabla F(\mathbf{x}_k)\| < \epsilon$ , 计算以
- 3: 由式  $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$  得到其中  $\alpha_k$  的值从 1.0 开始折半查找, 直到步长足够小时, 函数值低于上一步。
- 4: 计算  $\delta_k = \alpha_k \mathbf{p}_k^T \mathbf{y}_k$  和  $\mathbf{V}(\mathbf{x}_k)$
- 5: 计算新的切线刚度矩阵  $B_{k+1} = B_k + \frac{\mathbf{V}(\mathbf{x}_k) \mathbf{V}(\mathbf{x}_k)^T}{\delta_k}$
- 6: 判断: 将当前解代入原式验证, 若拉格朗日方程组等式两边的误差小于  $\epsilon$ , 则得到待求最优解; 否则返回第 2 步。



6.2.2 反射面调节情况的求解与分析

由于问题二中使用到了旋转的坐标系,故本问理想抛物面的结果可以在旋转坐标变换的条件下,得到了顶点坐标为 $(-49.392, -36.943, -294.450)$ 的抛物面。接着,基于算法2,在MATLAB中编程进行求解,给出求解出的3D的促动器提升量图(图7左)及3D的理想抛物面的拟合误差图(图7右):

图7: 3D促动器提升量(左)、3D理想曲面拟合误差(右)

由图7左可知,此状态下促动器的提升量处于 $-0.4\text{m}$ 到 $0.6\text{m}$ 的范围内,此时反射面较为贴近所求理想抛物面,且促动器顶端上下拉下拉索的长度范围也较小。由图7(右)可知,此状态下理想抛物面的拟合误差处于0到 $7 \times 10^{-6}$ 的范围内,拟合误差也较小,说明此时的结果为合理最优解。

### 6.2.3 结果检验

基于6.2.2节中的结果,求出主网节点与理想抛物面间的球面径向误差以及主网节点间距离的变化率:

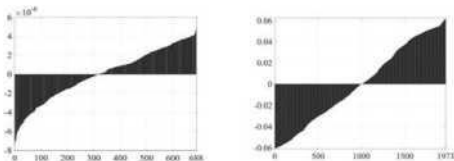


图8: 主网节点与理想抛物面球面径向误差(左)、主网节点间距离变化率(单位:百分数)(右)

由图8(左)可知,此状态下主网节点与理想抛物面间的球面径向误差处于 $-8 \times 10^{-6}$ 到 $6 \times 10^{-6}$ 的范围内,可见此径向误差非常小,此时反射面较为贴近所求理想抛物面。由图8(右)可知,此状态下主网节点间距离的变化率处于 $-0.06\%$ 到 $0.06\%$ 的范围内,充分利用而并没有违反 $0.07\%$ 的裕度,说明此时的结果是合理的。

## 七、问题三模型的建立与求解

### 7.1 问题三模型的建立

问题三要求基于问题二的反射面调节方案,计算调节后馈源舱的接收比,并与基准球面接

收比比。对于该问题,本文首先通过使用旋转变换,将倾斜入射光线转化为垂直入射光线。其次,通过求解线性方程组,依次确定出入射光线与三角形面板的相交判定式、交点坐标、三角形面板的法线向量,并利用光线垂直入射的性质,使用法线向量简化计算得到出射光线的方向角。再次,通过联立射线方程与馈源舱所在的目标高度,得出射线方程的步长和出射光到达目标高度时的坐标,并与馈源舱的有效区域进行比对,作为入射光线是否被有效接收的判定式。最后,将300米口径内实际接收区域作为积分域,将入射光线有效接收判别式作为被积函数,使用蒙特卡洛算法进行积分,计算有效接收的光源面积和调节前后的信号接收比。

### 7-1.1 反射光线几何模型的建立

首先,要在新坐标系中计算反射点在反射面板三角形区域内的坐标。设新坐标系的空间变量为 $(x, y, z)$ ,在球坐标系 $(r, \theta, \phi)$ 的范围内,沿着 $z$ 轴负方向均有来自于被观测物体的电磁波照射在反射面上。记反射面板上一三角形三个顶点1, 2与3的坐标分别为 $A_1(x_1, y_1, z_1)$ ,  $A_2(x_2, y_2, z_2)$ 和 $A_3(x_3, y_3, z_3)$ ,表示三角形访问的主索节点的坐标。其中编号 $i, j, k = 1, 2, 3$ 。

$$IX: f(4 \text{ 七 } 0) \diamond 5r(\Delta) + V. (.) \rightarrow jri = \wedge(Vr(A) + * S)) \quad )$$

y 協同

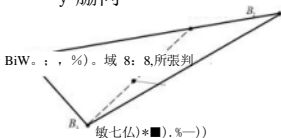


图 9: 反燒面板三角形区域示意图

设一点  $(ZwRw, Zw)$  在线段耳尻上, 则其在 Z 轴的坐标可以表示为:

$$Xw r_s x_w (Ji) + r_2 x_w \quad (2) \quad (7-1)$$

其中, 矩 1,  $[2 > 0]$  且  $[1] + [2] = 1$ . 接着, 设另一点  $(Xw, Vw, Zw)$  在点 (如, 加,  $Zw$ ) 与田 3 为端点的线段上, 该点在 Z 轴上的坐标可以表示为:

$$Xw S_i T_i X_w (j^A) ' 8i T_i X_w (j^A) S_{2lw}(j3) \quad (7-2)$$

其中,  $8[$  如  $>$ . 且  $S1 +$  如  $- 1 e$

记  $S, JH = ti, 81 r_2 = t_2, s_2 = t_3$ , 易证  $勾 + 0 - (3 1 \cdot$  且  $ii > 0, \langle 2 > 0, t_3 > 0 c$  同理可得该点在 g 轴与 z 轴的坐标. 以上, 得到了一点在以晶爲为顶点形成的  $\cdot$  角形内部平面上的坐标为:

$$\begin{pmatrix} x'_w \\ y'_w \\ z'_w \end{pmatrix} = \begin{pmatrix} Xw(jd) T_w (j_2) y_w \\ GO y_w(ji) 初 Gi) 咖. 3) 司 \\ Zw(02) \quad * 伝) 八 / \end{pmatrix} \quad (7-3)$$

将  $< 3 1 ti$  七代入式(7-3)约去 z 轴的坐标, 并根据矩阵运算法则可得:

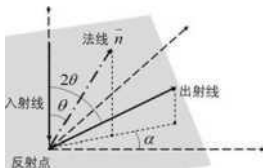
$$\begin{pmatrix} t_1 \\ t_2 \end{pmatrix} = \begin{pmatrix} 叫 G1) - Xw (j_3) Xw (撈-Xw (项 3)) ^ Xw-Xw (>3) \setminus y_w GO - y_w \\ 3) y_w (方) - y_w G) \setminus y_w - y_w (糸) \end{pmatrix} \quad (7-4)$$

注意到三个点  $B1B2B3$  为三角形的三个顶点, 式(7Y)中的逆矩阵总是存在. 因此, 当坐标 (成, 筵) 给定时, 可以根据式(7-4)计算勾和七的值, 进而判断该点的水平面投影 是否在三角形的水平面投影的内部' 根据实际条件, 对于任意在珞+诡 W1502 范围内 (的(阿, 姑), 都有一个顶 6 3 使得其在项所指代的三角形内部. 故可进一步根据式(7-3)求得(昭滅)对应的 z 轴坐标为:

$$Zw Gx) + (j2) ' hZw(ji) \quad (7-5)$$

另次, 要确定反射直线的解析表达式. 计算反馈面上: 个点  $\mathbf{J}^2$  为构成平面的法向量,  $= (e_j; e_2; e_3)$  有:

$$\begin{aligned} & 珈 Gi) _ (ji) y_w Gi) - y_w (J2) 叫 02) - \% (j1) _ Zw(j2) ' \\ Xw Gs) y_w 02) - y_w 伝) & \quad Zw(02) - Zw(03) \\ , 务 v(03) _ x_w (?i) y_w G) - y_w (Jd) & \quad 知 Gs) - 初 G1), \end{aligned} \quad (7-6)$$



图】0: 各角度三维示意图

注意到, 法向量没有归一化且式(7-6)左侧矩阵的秩为 2。且考虑到问题所求的抛物面开口向上, 即  $e_3 > 0$ 。故令免 1。将式(7·6)进行化简并根据矩阵运算法则可得:

$$\begin{aligned} \text{仲 } 1) = & (\text{知 } G1) - \text{一} (02) V_w G, J) - V_w 0^* 2) \backslash \quad \_Z W G I) \backslash \\ 2 J \quad & \backslash X_w (J2) - X_w (j_s) V W (j2) - V W G_s) / L z w G) - Z \#(, 2) J \end{aligned} \quad , ,$$

由于入射光垂直于水平面向下, 由方向向量  $rh = (0,0,1)$  因此二与云的夹角:

$$\theta = \arccos \frac{\vec{n} \cdot \vec{n}_1}{|\vec{n}| \cdot |\vec{n}_1|} \quad (7-8)$$

接着, 将法向量成投影到水平面后与 4 轴的方向向量  $M = (1,0,0)$  的夹角  $Q$  表示为:

$$\arccos \frac{z_2}{r} = \arccos \frac{e_2 \cdot 0}{g \text{ 端}} \quad (7-9)$$

$s < \circ$

另外, 由于出射光与入射光所在平面垂直于反射面, 可知出射光与 z 轴的夹角为  $2\theta$  因此出射光的方向向量:

$$Th = (\sin(2\theta)\cos\alpha, \sin(2\theta)\sin\alpha, \cos(2\theta)) \quad (7-10)$$

以上, 可得出射光所在的直线的标准方程为:

$$\begin{aligned} \frac{x}{x_w} \backslash \quad \frac{y}{y_w} \backslash \quad \frac{z}{z_w} \backslash \\ \text{(加 } = (\text{琅 } + r \text{ sm}(2^\wedge) \text{ Sina} \\ \backslash z w J \quad \backslash z^\wedge J \quad \backslash \cos(2d) J \end{aligned} \quad (7-11)$$

最后, 要确定馈源舱接收平面的解析表达式。根据题目, 馈源舱接收平面的法线方向指向理想抛物面的顶点, 距离原点的距离为  $r$ 。考虑到由于馈源舱接收信号有效区域为直径 1 米的中心圆盘, 因此其平面方程为



J 如= \_(R\_F)  
[瑞] 诡 WO\*

综上, 在新坐标系中构建了反射光线所在直线与馈源接收平面的解析表达式。

### 7.1.2 反射光线吸收与信号比计算模型的建立

在确定了反射光线与馈源接收平面的解析表达式后, 即可判断反射光线是否被接收并计算反射信号比。

首先, 将接收面平面方程与出射光所在直线方程进行联立, 可以得到出射光线命中 馈源舱所在 Z 平面的行进步长:

结合式(7-11)和(7-13).整理可得:

$$x_w = X_w - \frac{\sin(20)\cos\alpha}{\sin(20)\cos\alpha} \quad (7-14)$$

$$V_w = V_w + \frac{\sin(20)\cos\alpha}{\sin(20)\cos\alpha} \quad (20)008.$$

接着, 可以基于入射线指向的水平面上点的坐标 ( $3^\circ, \alpha <$ ) 并根据式(7-14)求得  $x_w$  和  $y_w$  因此可以通过:

$$\blacksquare 3 \text{ " 赢的 OS 且或诡 } N. * \quad (7, 5)$$

判断反射光线是否能够被接受面吸收。进一步, 对判断函数  $1(\text{加用 "})$  在劣  $3^\circ + \text{弗 } W 15.$  2 范围内积分, 可以得到被馈源接收的光线所占的面积:

$$\Delta^*, (\text{"}) \text{城烦} \quad \text{网}$$

接着, 注意到有效口径内的每个点都必然有且仅有一个相应的三角形, 则平面内接收到的光线面积为  $150^\circ 7T$  最后, 可得接收与反射信号之比:

$$\text{"} = \text{筋} \Delta \quad \text{伽} < 5$$

## 7.2 问题三模型的求解

### 7.2.1 算法设计

对于问题三所求的积分数值, 如果直接对整个口径圆面进行积分, 则对于每个输入的 光线坐标, 算法都将消耗大量时间遍历  $zOy$  平面上的各个投影三角形, 以确定这一坐标所属的三角形法线, 用于计算出射光线的射线方程。为了避免这一逐点依次进行三角形遍历的时间成本, 我们将圆形口径积分域分解为各个互不重叠的  $zOy$  投影三角形积分域, 通过对各个三角形积分域内均匀地采样各个光线的接收判定信息, 使得一次性遍历到各个投影三角形时, 可以大批量地使用同一个法线向量处理多个投影到该三角形上的入射光线, 甚至使用足够高密度的采样从入射光线的接收信息还原出馈源有效区域在 各个投影三角形上的像。为了实现三角形区域的均匀采样式积分, 我们设计了包含对边界条件精细修正的蒙特卡洛积分算皂算法细节如下:

算法 3: 嘴推 1: 洛耸沃

1: 设置接收面积近似值  $a: = 0;$

- 2: 遍历三角形 JG J:
  - 3: 设置接收次数初始值  $d:=0$ ;
  - 4: 抽样次数 10000.循环;
  - 5: 每次在 0 到 1 之间平均随机抽样生成一个  $t_i$  和一个有:
  - 6: 如果 +
  
  - 10: 使用式 (7-3) 从 0,坛. 计算出 (务  $y,z$ );
  - 11: 使用. 和 0 计算出E01/平面上的投影向量长度;
  - 12: 如果向量:长度超过口径的一半 150 米:
  - 13: 跳过这次接收计算, 跳转到第 16 行:
  - 14: 使用 ( $z,y,z$ ) 计算出是否接收, 如果接收:
  - 15:  $d:=d+1$ ;
  - 16: 如果抽样次数没满. 跳转到第 4 行循环:
  - 17: 计算出三角形项在  $a$ :!/平面上的投影面积:
  - 18:  $ai: = ai + d/10000 \cdot$  (阴影时)
  - 19: 如果週历三角形没结束. 跳转到第 2 行继续遍历
  - 20: 输出小/(1502 汗) 円为所求接收礼
- 

### 7.2.2 问题三的结果与分析

基于算法 3,在 MATLAB 中编程进行求解. 给出求解出的在相同区间下, 工作态 (左) 与基准态 (右) 中. 反射面板与有效入射点区域的纹理对比图. 其中. 黑点表示 光波经反射后. 可以被馈源舱接受平面的中心有效区域吸收的反射面板的入射点区域. 图 1】中所选取的区间横坐标范围为 (  $90^\circ, 30^\circ$  ).对照图 5.可得此区间基准态和工作 态之间偏移量的相对斜率最大, 夹角差异最大, 导致基准态时部分反射面板中的“有效 接受反射命中点”逐渐偏移至反射面板外部, 进而反射率也相应降低. 因此工作态和基 准态下·最终产生的反射效果相差较大. 最终计算得到的调节前接收比 0.811%.调节后 接收比 1.103%,提升了 36%

-30 -90 -80 -70 ■\* -50 -40 -JO\* -90 -80 -70 ^0 -50 -40 -M>

图 11: 在相同区间下, 工作态 (左) 与基准态 (右) 纹理对比  
 将计算得到的此情况下反射光线聚焦于焦点的状态进行可视化处理, 得到图 12。

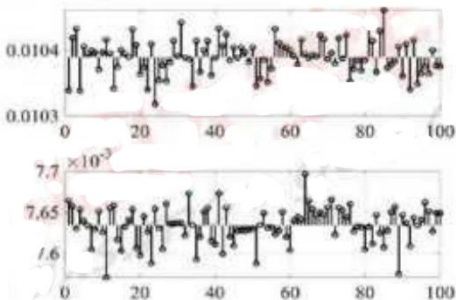


图 13: 工作态 (上) 与基准态 (下) 下多次蒙特卡洛积分的数值波动量

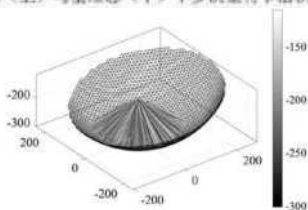


图 12: 反射光线聚焦于焦点的示意图

由图 12 可观察到, 在此情况下, 由观测点 S 方向射出的平行均匀光波被精准地反射到了馈源舱有效接收区域, 故所得结果符合题设。

### 7.2.3 问题三结果检验

基于 7.2.2 节中的结果, 对蒙特卡洛积分的稳定性进行检验。在工作态与基准态下 各积分 100 次, 得到数值波动量如图 12:

观察光斑分布偏移, 由图 13 可得工作态下与基准态下, 多次蒙特卡洛积分的数值波动量

范围都在  $10^{-3}$  的维度上, 波动量较小, 可以认为结果合理。

## 八、模型的评价与推广

### 8.1 模型的评价

本文构建的 FAST 射电望远镜的工作抛物面调整模型, 对于利用促动器调整工作抛物面的过程进行了合理假设, 在保证模型求解结果精确性的条件下, 有效降低了模型的复杂程度。在求解过程中, 利用了二分法, BFGS 法, 蒙特卡洛积分等方法对于模型进行求解, 使得算法的收敛速度快且结果精确。

在第一问的理想抛物面求解过程中, 考虑到抛物面是由抛物线沿轴线旋转得来, 故将空间的问题转化为平面坐标系下的问题, 使得模型描述更为简便。对于限定了焦距的抛物线方程, 本文以使其与基准球面的近似程度最大为最优化条件, 对于其焦距进行了优化, 从而确定了最优的抛物面方程。

在第二问的工作抛物面调整过程中, 考虑到抛物面的开口方向与原先的坐标轴方向并不相同, 本文将原先的直角坐标系进行旋转, 使得原先的坐标轴方向依然指向被测物体, 使得抛物面方程的描述更为简化, 构建出的模型更加简明。并且在此过程中, 通过极坐标变换, 利用了第一问得出的结论, 节省了公式推导的工作量。在求解时利用 BFGS 法, 将约束条件转化为二次型, 提高了算法的收敛速度与精确度。

在第三问的馈源舱接收比计算模型中, 本文将直角坐标, 方位角等概念结合使用, 在旋转后的直角坐标中通过构建参数方程判断三角形平面与直线的相交关系, 使得模型的推导过程更加简单。在求解有效接受的光源面积时, 使用蒙特卡洛积分的方法进行求解, 在一定程度上提高了解算的速度。

### 8.2 模型的改进

与实际情况相比, 题目中缺乏边界结构如何固定的相关信息, 如支撑结构等, 这就使得在确定整个工作抛物面时约束条件是不够充分的。若能够提供最外圈反射面板的固定方式, 并且合理假设在边界的节点之间的距离约束等条件, 可以对于工作抛物面的边界条件进行建模, 并且通过各个节点之间的连接关系间接地传递约束, 从而给出 FAST 射电望远镜所有促动器伸缩量的调整策略, 构建出可以投入实际调度的工作抛物面调整模型。

## 九、参考文献

- [1] 杨凡, 李广云, 王力. 三维坐标转换方法研究 [J]. 测绘通报, 2010(6):5-7
- [2] 龚纯, 王正林. 精通 MATLAB 最优化计算 [M]. 电子工业出版社, 2009.
- [3] Beavis B. Dobbs I. Optimisation and stability theory for economic analysis [M]. Cambridge university press. 1990.

## 十、附录

### 10.1 主要计算程序：基于 MATLAB R2019a 开发

#### 10.1.1 主要计算程序

probl.calc.m: 问题 1 的求解核心算法

```
function [Lf, angle300] probl_calc。  
%  
%问题 1 的核心计算部分  
%积分输出最优焦距  
%并且与 Constant 中固定的焦距做比较  
%  
warning('off', 'MATLAB: integral: NaxIncrvalCounlReachcd');  
warning('off', 'MATLAB: integral: MinStepSize');  
angle_min = acos(150/Constant.sphere radius);  
Lf_fzero = @(L)int dl/Lf, angle_min, (100, 400)];  
disp('问题 1 最优焦距 Lf: ' + mat2str(Lf_fzero));  
disp('响题 1 最小误差平方积分: ' + mat2str(int L(Lf, angle_min), 17));  
assert(abs(int dl/Lf - f_angle_min)) < sqrt(eps));  
assert(Lf < Constant.LQ);  
I, . . . angle I Constant, get rad angle。 :  
f = @(angle)abs(PolarPara(Lf, angle)*cos(aiangle))-150;  
angle300 = fzero(f, [pi/2, angle(end)]);  
assert(abs(Rangle300)) < sqrt(eps));  
end
```

```
function v = int_L(Lf, angle_min)  
v = integral(@C angle_min, pi/2);  
function err = "angle"  
[L, ] = PolarPara(Lf, angle);  
err = (L - Constant.sphere radius) * (I, -Constant.sphere radius);  
radius = Constant.sphere radius * cos(angle);  
err_err = (2 * pi * radius);  
aid  
end
```

```
Afunction v = int_dL/LC angle_min)  
v = integral(@f, angle_min, pi/2);  
function derr = RangleC  
[L, dL] = PolarPara(Lf, angle);  
derr = 2 * dL * (I, Constant.sphere radius); radius
```

```

        Const ant.sphereradius.*cos(angle);
        derr dcurr.*(2.*pi.*ra (lius);
    end
end
end

```

prob2\_calc.m:问题 2 的求解核心算法

```

function [fval, opt X, opt XYZ, opt Lz, opt ABR, index, W, Winv,...
    edge, edge distSQ, motor hi, motor unit, motor_dislSQ] prob2_calc() clffreset
mg('deiaull');
clear global

%初始化数据集
data = Data();
%初始化旋转变换矩阵
[W, Winv] Polar.compose_rotate(Constant.alpha, Constant .beta);
%选择 300 口径内的节点
[~, angle300] = probl calc();
main XYZ = data.inain node coordinate;
main ABR = Polar.xyz2abr(main XYZ*\V);
index main ABR(:,2) >= pi-angle300;
N nodes sum(index);
main_XYZ = main_XYZ(index,:); % 过滤主网节点 edge
data.edge(all(index(data.edge),2),:); % 过滤节点之间的连边 index2(index) -
1:suin(index); %构建过滤之后的新的节点下标顺序 edge indcx2(edgec):為使
用新的节点下标对连边节点序号定向 assert(all(all(edge)»); %连边节点
序号不能有空泡
%初始化邻边长度
edge distSQ = main_XYZ(edgec(:,1),:)-main_XYZ(edge(:,2),:);
edge distSQ sum(edge distSQ.*edge distSQ.2);
%促动器预处理
motor hi data.motor higlier base(index,:);
motor Io = data.motor lower(index,:);
motor unit motor hi-motorJo:
motor unil motor unil.'sqrt(suni(motor unit.*motor unit.2));
motor distSQ = niain XYZ-motor hi:

```

```

motor distSQ sum(motor distSQ.*motor distSQ. 2);

objective = @(X)obj_SSE(X, W, N_nodes);
nonIcon    @(X)constraints(X. edge, edge distSQ, motor hi. motor unit,
motor distSQ. N nodes);

f testkit(...
    pack(niain XYZ, rand(N nodes. 1))....
    objective,...
    nonIcon...

opts optimoptions(*fmincon', 'Display', 'iter-detailed');
opts. Fun ValChck 'on';
opts.SpecifyO ̄ eclivcGradicnl true; opts.Specif) <?onslrainlGradient = tnic;
opts.MaxFunctionEvaluations inf;
opts. Max Iterations inf;
opts. HonorBounds = false;
% opts. Subproblem Algorithm " 'eg':
opts. Hessian Approximation 'bfgs';
saved = load('prob2');
if true
    opt_X = saved.opt X;
else

    opt X finincon(objective, pack(niain XYZ, zeros(N nodes. 1))> (. ||. ||. pack(-
    inf; size(main XYZ)). repmat(-0.6, N nodes. 1)).... °o lb
    pack( inRsize(main_XYZ)). repmat( 0.6. N nodes. 1)),... % ub nonIcon,
    opts): %#ok<UNRCH>
end
ival obj SSE(opt X. W. N nodes);
(opt XYZ, opt Lz| unpack(opt_X, N nodes);
opt_ABR Polar.xyz2abr(opt_XYZ,W): end

function [SSE. dSSE_dXYZ] = obj_SSE(X. W, N nodes)
%
%目标函数

```

```

%球面径向投影 Lw (理想抛物面)和主网节点实际投影位置 L 之间的误差平方和
%
(XYZ. [N] = unpack(X, N nodes);
[abr. . db. dr] Polar.xyz2abr(XYZ*W');
(Lw, dangle) = Polar.para(Constant.IX abr(:,2,:));
err Lv-abr(:,3);
SSE = eiy*err;
dSSE_dXYZ 2.*crr.*(dangle.*db-dr);
dSSE dXYZ = dSSE_dXYZ*W;
dSSEJXYZ [dSSE_dXYZ(:); zeros(size(Lz))];
end

function (c.ceq.gc.gceq) constraints(...
    X,...
    edge, edge distSQ,...
    motor hi, motor unit, motor distSQ, N nodes)
%
%约束条件(二次型)
%条件 1:保持主网节点之间的连接索的距嘉保持固定
%条件 2:保持主网节点与相应的促动器之间的距离(下拉索)保持固定
%
(XYZ. Lz) = unpack(X, N nodes);
errdist = XYZ(edge(:,1):-XYZ(edge(:,2):-);
edisl sum(crrdist.*errdist,2);
c |edist-edge distSQ*( 1.0007* 1.0007); edge_distSQ*(0.9993*0.9993)-edist);
11 = sub2ind(size(XYZ), rq) mat(edge(:, 1). 1.3), repnlat( 1:3.numel(edist), 1));
12 sub2ind(size(XYZ), rcpnlat(edge(:,2),1,3). repn>at(1:3.numel(edist),1));
J = repmat((1:numel(edist)).', 1, 3);
gc = |
    sparse(...
    reshape([ll I2].|.IX—
    reshape@
    reshape(2.*[errdist -errdist].|1.1)...
    numel(X),numel(edist)),...
    -sparse(...
    reshape((ll 12 |]). 1> ...
    reshaped J |]), ...
    reshape(2.*[errdist -errdist].|.1)... numel(X).numel(edist)
);

```



```

motor hi new motor hi+Lz.*motor unit;
errdist = XYZ-motor hi new;
edist sum(errdist.*errdist,2);
ceq edist-motor distSQ;

J = repmat(( 1:numel(edist)).', 1, 4);
geeq sparse(...
    reshape( 1:numel(X). []. 1),...
    reshape(J,[],1),...
    reshape(2.*errdist -sum(errdist.*motor unit.2)).[],1),... nuinel(X),numel(edist));
end

function [SSE. dSSE] constr test ceq(X. constraints)
    [~,ccq,~,geeq] = constraints(X);
    SSE = ceq.*ceq;
    dSSE = full(2.*geeq.*ceq);
end

function [SSE. dSSE] constr test_c(X. constraints) (c,~,gc,-| constraints(X);
    SSE = c.*c;
    dSSE = full(2.*gc.*c);
end

function f_lcskit(dala. objective, constraints)
    %
    %函数微分正确性检查
    %
    f_0 = objective;
    fbr = 1:10
        index = randi(numel(data)/4*3X
        ratio = linspace(0.1, 10.0. 20);
        difT = arrayfun(@(r)derr(index), ratio);
        ana = arrayfun(@(r)dcT_aiaa(r. index), ratio);
        disp("信噪比: "+nomi(difl' -ana)'iionn(ana ));
        plot(ratio. difT -ana );
        hold on
    end
    f = @(X)constr_lcsl_ccq(X. constraints);
    for i = 1:10
        index = randi(nuinel(da(a));

```

```

ratio linspace(0.1, 10.0, 20);
ana    arrayfun(@(r)derr ana(r, index), ratio);
dift'  arrayfun(@(r)derr dift(r, index), ratio);
disp('信噪比: '+nonn(dift -ana) nonn(dift));
plot(ratio, dift -ana);
hold on
end
l' = @(X)conslr test c(X, constraints);
fori= 1:10
    index randi(numel(datay4*3));
    ratio = linspace(0.1, 10.0, 20);
    ana    arrayfun(@(r)derr ana(r, index), ratio);
    difT   arrayfun(@(r)derr dift(r, index), ratio);
    disp('信噪比: '+nonn(dift -ana) nonn(difT));
    plot(ratio, difT -ana);
    hold on
end
function dSSE = derr difT(ratio, index)
    assert(isscalar(ratio));
    dSSE (f(set(index, ratio*1e-4))-f( set(index, ratio-1e-4)))/2e-4;
end
function dSSE derr ana(ratio, index)
    assert(isscalar(ratio));
    [~, dSSE] = Rset(index, ratio);
    dSSE dSSE(index)*data(index);
end
function newdata = set(index, ratio)
    newdata data;
    newdata(index) - newdata(index)*ratio;
end
end
function X = pack(XYZ, Lz)
    X = [XYZ(:); Lz];
end
function [XYZ, Lz] = unpack(X, N nodes)
    assert(numel(X) == 4*N nodes);
    XYZ = reshape(X( 1:3*N_nodes), [3, 3]);
    Lz = X(3*N_nodes+1:end);
End

```

prob3\_calc.m:问题 3 的求解核心算法(含画图输出, 检验等)

```
cIR)
clearQ
mgCdefault');
close(air);
clear global
global opt XYZ index W
[.opt XYZ,index, W, I prob2 calc();cIR, reset');
%画图
clf^'reset');
    o plot working illu 3d(tnie);
cIR^'reset');
% plot 2d lcxture(tnie);
cl^'reset');
% plot 2d lcxture(false);

rec ratio work (0.011020300922443 0.0110003218576217 0.0110518782939506
```

0.0110643817438084	0.0109730660550652	0.0110257601080111	0.0110223227972293
0.0110485078071177	0.0110240151884694	0.0110130230869417	0.0110164104558229
0.0110550668326781	0.0110459270545307	0.0109882612985835	0.0110436769357324
0.0110367034749252	0.0110481766120668	0.0110321572590037	0.0110749365711392
0.0110386802631623	0.0110153292520637	0.0110192330970475	0.0109874238643901
0.0110296311887876	0.010975738534649】	0.0110192330970475	0.0110288022004406
0.0110296311887876	0.0110238540462687	0.0109755111105056	0.0110284372401636
0.0109799794453533	0.0110526144686468	0.0110453042446409	0.0109741922281317
0.0110669362584378	0.0110341899191246	0.0110334430375848	0.0110496603830398
0.0110478434113818	0.0110235600140406	0.0110517591813282	0.0110649714478939
0.0110046254841212	0.0110420096870245	0.0110550698235744	0.0110091725112358
0.0110474009766273	0.0110600521586413	0.0110365756602061	0.0110465102463241
0.0110382613793932	0.0110101135220489	0.0110028232503258	0.0110043363971838
0.0110136604630215	0.0】 10837873612865	0.0110183405377215	0.0110277008429329
0.0109958279906651	0.0110545986531244	0.011065917248056	0.0110429260222507
0.0110127664607068	0.011039406367263	0.0110345008277633	0.0110289593537192
0.0110505079293461	0.0110513385417147	0.0110103799922359	0.0110445898913447
0.0】 10541998462448	0.011022189720714	0.0110529854247001	0.0110285841846507
0.0110180088017255	0.0109907491678801	0.0110627343553963	0.0110311268543086
0.0110817414943682	0.0110255093771213	0.0110200180167431	0.0110777319262726
0.0110146584529961	0.0110789763845019	0.0110528637613645	0.0110068509917417
0.0110169034101532	(.0)110403474743263	0.011086907752962	0.0110162286817121
0.0110202757278154	0.0110068533880955	0.0110425429291141	0.0110996333441479
0.0110382443277745	0.0110230589753898	0.0110452395673006	0.0110296744899169
0.011(X)0327429337		0.0110471891212048	
0.0110090500347684):			
rec ratio_base	(0.00809957344004469	0.00816030276752031	0.00815114719028782
0.00809170932206714	0.00813359107499093	0.00815073014613017	0.00808926197249174
0.00808170453150241	0.00806468281488732	0.0081182224458027	0.00809664564432054
0.00803929331956928	0.00813862975913561	0.0(8)12651614405913	0.00807548662594117
0.00809838752451361	0.00809400286528839	0.008】 100908082638】	0.00810822712967782
0.00809315525807824	0.00806486974020875	0.008134630287794	0.00807892790322006
0.00805598714908852	0.00813180369790822	0.00810668350665907	0.00807828088140091
0.00811052825834018	0.00810927517071732	0.00811143502132163	0.00810309576115239
0.00810982854434356	0.00811344740473194	0.00814411524881943	0.00810455307681079
0.00806302524392838	0.00808600818409412	0.00812354564034997	0.00811186777478928
0.00809026737854994	0.00808663956574123	0.00814128338195877	0.00807940794169834
0.00812115941770167	0.0081024286210036	0.(X)80924217013025	0.(X)809501225610337
0.00809902994305578	0.00812940497617186	0.00809554341149597	0.00809631643018426
0.00804895426011689	0.00809575027496546	0.00812928273720676	0.008】 1334164853845
0.00812004359958459	0.00810587906967159	0.00807971230)25123	0.00810458060650855
0.00809081157217132	0.00807630129046536	0.00812220517987459	0.00811206634654426
0.00810709806891063	0.00815164906999222	0.00813766254713846	0.00814321524369861
0.00810908041014188	0.00812934844902824	0.00810621605696549	0.00811063002236979
0.00812537388975942	0.00809493844001929	0.00811284919748417	0.00812982625291347
0.00811444175548698	0.00812948462742588	0.00812471872392731	0.00809515370106578

```

0.00810070201885477 0.00809154975241786 0.00809924616852124 0.00808220053817524
0.00812402217146259 0.00811192024738402 0.00806217864697205 0.00809729197336589
0.00814243000329029 0.00814908482651054 0.00805549860700152 0.00812911963465325
0.00809820772628644 0.0080799192180059 0.00812099010446983 0.00811185636211377
0.00810677088931511 0.00812457458272667 0.00811715912091946 0.(X)810254234380183
0.00812512633851956];
subplot(2,1,1);
hold on
plot((1:100:1:100)Jrepmal(mcan(rec ratio work).size(rec ratio \work)):rec ratio work] ; k-
'.LineWidth\2);
plot(1:100.rec ratio work/ko'/LineWidth*^);
box on
set(gcf,'Name','fbntname*', 'FontSize', 'fbntsize');
%ylim([0.01033 0.01045]); subplot(2,2);
hold on
plot([1:100;l:100],[rq]mat(mean(rec ratio_base),size(rec ratio base));rec_ratio_base|,k
'.LineWidth\2);
plot(1:100.rec ratio base.'ko'. 'LineWidth\J);
box on
grid on
set(gcf,'Name'\ 'fbntname'\ 'fbntsize');
fasiprintC 图片/prob3.检轮.多次蒙特卡洛积分的数值波动量) ffunction [dala. XYZ. tri. Iri
area] = get data(WORKING)
%
%提取工作态或者非工作态下需要使用的所有数据
%
data = Data. :
global opt XYZ index W

XYZ data.main node coordinate: %产生一个对齐到光线入射角度的坐标格式 if
WORKING
XYZ(indcx,:) = opt XYZ;%主索网调整到工作抛物面
end
XYZ = XYZ»W;
%计算三角形的面积
triangle = data, triangle:
X reshape(X YZ(triangle. 1 ),size(tri;uigle)):
Y rcshapc(XYZ(trianglc.2),sizc( triangle)):
%Z reshapc(X Y Z(triaigle.3 Xsizc(trianglc));

```

```

triangle area 0.5.*abs(X(:,1).*(Y(:,2)-Y(:,3)) + X(:,2).*(Y(:,3)-Y(:,1)) +
X(:,3).*(Y(:,1),Y(:,2)));

```

```

%筛选出所有可能跟选中顶点有关系的三角形

```

```

tri = triangle(any(index(triangle),2).:);

```

```

tri_area = triangle_area(any(index(triangle),2));

```

```

end

```

```

(unction (rec_ratio, xyz, uvw, data, XYZ) % me integral(WORKING)

```

```

%

```

```

%提取出当前有效工作界面(由有效三角形定义)中的

```

```

%所有 xyz 映射到 uvw 的射线关系

```

```

%(蒙特卡洛法, 顺便就完成了积分)

```

```

%

```

```

[dala, XYZ, tri, tri_area] = get_data(WORKING);

```

```

area = zeros(size(tri, 1), 1);

```

```

xjz = cell(size(tri));

```

```

uvw = cell(size(tri));

```

```

N = 10000;

```

```

for i = 1:size(tri)

```

```

    X = XYZ(tri(i,:),1);

```

```

    Y = XYZ(tri(i,:),2);

```

```

    Z = XYZ(tri(i,:),3);

```

```

    t123 = rand(N,2);

```

```

    t123(t123(:,1)+t123(:,2)>1,:) | 1 - t123(t123(:,1)+t123(:,2)>1.); U23(:,end+1) = 1-t123(:,1)-t123(:,2); % «ok<AGROW>

```

```

    [xyz, uvw, ratio] = calc_t_fast(t123, X, Y, Z);

```

```

    xyz{i} = xyz_(~isnan(ratio),:);

```

```

    uvw{i} = uvw_(~isnan(ratio),:); arca(i) = mean(~isnan(ratio));

```

```

end

```

```

assert(ynyn(isnan(arca)));

```

```

area = area(~isnan(arca));

```

```

% tri_area = tri_area(~isnan(arca));

```

```

rec_ratio = sum(tri_area.*area)/(150*150*pi); if WORKING

```

```

    disp("工作接收比: ", mat2str(rec_ratio, 17)); else

```

```

    disp("基准接收比: " MnatZstiXrec_ratio. 17)); end

```

```

iimclion plot 2d texture(WORKING)

```

```

%
%画图
% 2d,放大界面纹理. 工作界面和基准态都要绘制
%
    卜. xyz, data. XYZ) mc_intcgm1(WORKING);
tri data.triangle(:,[1:3 1]);
p!ot(reshape(XYZ(lri.1),size(tri)).reshape(XYZ(tri.2),size(tri)), 'k'); hold on:
xyz - vcrctcat(xyz{:});
plot(x>-z(:,1).xyz(:,2)'.k');
axis equal
xlim([-90,-30]);
ylim([-30,30]);
sty('fblname'. 'fontsize');
if WORK ING
    fastprintC 图片/prob3. 结果. 纹理对比(工作态), );
else
    fastprintC 图片/prob3. 结果. 纹理对比(基准态), );
end
end

```

function plot working illu \_3d(WORKING)

```

%
%画图
% 3d.体现出工作界面的反射光聚焦于焦点
%这张图只在工作界面时绘制. 不在基准态绘制
%
assert(WORKING);
[ ~.xyz, uvw, data. XYZ] = me intcgral( WORKING);
xyz vcrctal(xyz{:});
uvw = vcrctal(uvw{ : });
xyz = xyz(1:100:end.);
uvw uvw(1: 100: end.);
p = plot3(...
    [xyz(: .1 Xx>-z(:,1)+uvw(:,1
    [xyz(: ,2>xyz(: ,2)*uvw(:,2)]. ' , ...
    [xyz(:,3),xyz(: ,3)*uvw(:,3)].V-');
color = mat2ce!(repmat(rand(umei(p),1 X 1.3),ones(nunicl(p),1 ) .3X

```

```
[p.Color] - color{:};
hold('on'X
```

```
X XYZ(:,1); Y XYZ(:, 2); Z = XYZ(:,3);
trisurf^data.lriangle. X,Y,Z, 'FaceColor'. 'None');
axis('equal');
colormap('gray');
colorbar();
grid('on');
box('on');
style('lntsize','f<) ntname');
```

rastprintC 图片/prob3. 结果. 反射光线聚焦于焦点的示意图)

end

```
hmclicion [xyz, uvw, ratio] calc t fast(tl23,X,Y,Z)
```

```
%
%寻找距离入射(x,y)最近的 XYZ 坐标点，并提取出以该点为顶点的所有三角
```

形

```
%依次求解它们的 t1,t2,t3
```

```
%
```

```
xyz = 1123 * [XYZ] ;
```

```
%分界线：上面是命中点，下面是法线
```

```
a = X(1)>X(2);b Y(1)>Y(2);
```

```
c = X(2)>X(3);d = Y(2)>Y(3);
```

```
uvw ((a b: c d)/[Z(2)-Z(1); Z(3)-Z(2)])•:
```

```
uvw(: ,end+1) = 1;
```

```
uvw abr Polar.xyz2abr(-uvw);
```

```
uvw abr(:,2) pi/2-(pi/2-uvw_abr(:,2)).*2;
```

```
uvw abr(:,3) - 1;
```

```
uvw -Polar.abr2xyz(uv xv abr);
```

```
ratio = -(1-Constant.FR_ratio).*Constant.sphere radius-xyz(:,3)./uv\(:,3):
```

```
ratio(ratio < 0) - NaN; %同向达不到馈源舱的情况下 pass
```

```
dst - xyz+ratio.*uvw;
```

```
assert(a)[(abs(dst(:,3))^(1-Constant.FR_ratio).*Constant.sphere_radius) < sqrt(eps)
```



```

isnan(ratio));
    ratio(sum(dst(:,1:2)?dst(:,1:2),2) > 0.5.*0.5) NaN: % 反射馈源舱平面时超过馈源舱圆圈
的情况下 pass
    ratio(sum(xyz(:,1:2)?xyz(:,1:2),2) < 0.5.*0.5) NaN: % 入射被馈源舱遮挡的情况 下
pass
    ratio(suin(xyz(:,1:2)*xyz(:,1:2),2) > 150? 150) NaN: % 超出 150 口径圆面 pass

    xyz(isnan(ratio),:) - NaN;
    uvw = ratio, ,u vw;
    assert(size(xyz, 1) == size(uv\ v. 1));
end

probl.m: 问题 1 的结果可视化、检验
%本文件原名 probl.m
财 ok<*DEFNU>
clear
clear global
close all
[ Lt angle300] = probl _calc():

cl f reset
illu2d(Lf. angle300);

elf reset
plot3d(Lf. angle300)

elf reset
error plot(Lf, angle300)

iiinclon error plot(Lf. aiigle300) % 横向剖面高低差
    angle = linspace(angle300, -angle300^pi. 100);
    Lw Polar.para(Lf. angle):
    err = Conslant.spherc_radiiis-Lw:
    plot([-Lw.*cos(angle); -Lw.*cos(angle)], [O.*err; err], '-k', 'Line\Width'.2); hold('on'X
    grid('on');
    plot(-Lw.*cos(a»glc). err. 'ok'. 'Line\Width'.2); xlim([-150. 150]);
    ylim(H) .6.0.6]);
    style('fbntsize', 'fbntnanc');
    %fastprintC 图片/probl.检验剖面上的高低差(径向高度 vs 基准球面半径), ); end

```

```
Hinction plot3d(Lf a»gle300) % 3d 高低差
```

```
d Data。 :
```

```
XYZ - d.main node coordinate:
```

```
ABR = Polar.xyz2abr(XYZ);
```

```
index = ABR(:,2) >= pi-angle300;
```

```
X XYZ(:,J); Y XYZ(:,2); Z = XYZ(:,3);
```

```
err = -Polar.para(LC ABR(:,2));
```

```
err(∞index) = NaN;
```

```
trisurf^d.triangle.X.Y.Z.err+Constant.sphere radius,'FaceColor','inteq,): view(2);
```

```
colorbar();
```

```
axis('equal');
```

```
style('fbntsize', 'fbntname');
```

```
colormapCgray');
```

```
box('on');
```

```
% fastprintfa 片/probl.结果.3D 最优掀物面高低差(径向高度), );
```

```
end
```

```
fiinction illu2d(Lf. angle300) % 横向剖面最优示意图
```

```
font opts = {'VerticalAlignment', 'middle*', 'HorizontalAlignment', 'center*', 'FontSize', 13.
```

```
'Interpreted, latex');
```

```
cl^'reset);
```

```
hold on
```

```
[sphcr rad, fbcral rad, angle] Constans.get rad angle。 :
```

```
p_S = ilhi.p S:
```

```
p_P = (0 -local rad I:
```

```
p_Q (0-fbcral_rad-0.5*Lfl:
```

```
p_T=[0 -focal rad-LQ;
```

```
hold('on');
```

```
%基准圆弧
```

```
plot(-sphcr rad*cos(angle),-sphcr rad*sin(angle), 'k', 'LineWidth', 2);
```

```
%焦面圆弧
```

```
plot(-fbcral rad*cos(angle),-fbcral rad*sin(angle). 'k'. 'LineWidth'. 2);
```

```
%标记定点点
```

```
plot(p S(1) p S(2): *k*. 'MarkerSize'. 14. 'LineWidth', 2);
```

```
plot(p P(1) p_P(2).'k'. 'MarkerSize'. 22. 'LineWidth', 2);
```

```
plot(p_Q(1). p_Q(2),'k'. 'MarkerSize'. 22. 'LineWidth'. 2);
```

```
plot(p T(1), p T(2).'k', 'MarkerSize', 22. 'LineWidth'. 2);
```

```
plot(0.0;.k. 'Markersize', 22. 'LineWidth', 2);
```

```
text(p_S(1)+20. p_S(2). ■ boldmathSS$. font_opts{:});
```

```
text(p_P(1)+15. p_P(2)+18. ■.boidmathSP$. font_opts{:});
```

```
text(p_Q(1)+15. p_Q(2)* 18. 'boldinathSQ$. font_opts{:});
```

```
text(p T(1)-18. p_T(2)+18, "\boldmath$TS', font_opts{:});
```

```

text(0+20. 0. *\boldmath$CS'. font opts{:}):
% SCPQT 中轴线
plot([p_S(1X p_T(1)|. |p_S(2). p_T(2)); -k; *LineWidth'. 2);
%准线
plot(pJT(l>+[-300 300J. p_T(2)+[0 0]. *k'. LineWidth'. 2);
plot(p T(1)+JO 30 30], p_T(2)+[30 30 0]. 'k'. Tine Width; 2); % 垂足
%计算抛物线
Lw = Polar.para(Lf. angle);
L300 Polar.para(Lf anglc300);
p_x [-L300.*cos(angle300) -L300.*sin(angle300)];
%抛物线
plot(-Lw.*cos(a>gle), -Lw.*sin(angle), '-.k\ 'LineWidth'. 2);
p!ot(p_x(1). p_x(2). Lk, . 'MarkerSize', 22); % 300 口径点
plot([p_P(1)p_x(1)L [p_P(2) P_X (2)], *k\ *LineWidth', 2); % 焦点连线
plot([p_x(D P_x(1)L (P_T(2)p_x(2))> *k\ ,LineWidth'. 2); % 准线连线
text(p_x(1). p T(2)>20. *\boldmath$x=300/2$, font opts{:});
%格式 axis('equal'): set(gca,Visibie,
% ylim([-300. 200]);
stylec('ibntsize'. 'fontnanc');
% festprint(, 图片/probl.结果.2D 剖面最优抛物线'):
end

```

prob2.m:问题 2 的结果可视化、检验

%本文件原名 prob2.m

clc

elf

clear

mg('default');

close all

clearCglobal');

data Data. :

(fval. opt X. opt XYZ. opt opt ABR. index. W. Winv....

edge, edge distSQ. motor hi. motor unit, motor distSQ] prob2 calc(): close all

xlsx output(opt XYZ. opt Lz. index, Winv) dispf 问题 2 最小误差平方和:Fval):

return

elf reset

errrplot 3d(data. opt XYZ, opt ABR. index);

```

elf reset
plot 3d Jz(dala. opt XYZ, opl Lz. index);
elf reset
plot 2d main node dist(opl XYZ. edge, edge distSQ);

```

```

elf reset
plot 2d fitcrr(opt ABR);

```

```

(unction xlsx output(opt XYZ. opt Lz. index, Winv)

```

```

    ifexistCresult.xlsx'. 'file')

```

```

        delctc('rcsul.xlsx');

```

```

    end

```

```

    copvfileC 数据/4.工作抛物面的主索节 A.xlsxVresult.xlsx*);

```

```

    %理想抛物面顶点坐标

```

```

    XYZ Polar.abr2xyz([0 pi/2 Polar.para(Constant.Lf pi 2));

```

```

    XYZ XYZ* Winv^1;

```

```

    t1 table(XYZ( 1), XYZ(2). XYZ(3));

```

```

    disp(" 理想抛物面顶点坐标 ");

```

```

    disp(head(t 1));

```

```

    writctabl(t1"rcsul.xlsx . . Sheet. , 理想抛物面顶点坐标 \ Range. . ,

```

```

A2:C2: WriteVariableNames. false);

```

```

    %调整后主索节点编号及其坐标

```

```

    data readtable(, 数据】.主索节点坐标和编号.csv)

```

```

    t2 "able(data.x_____(index). opt_XYZ(:,1), opt XYZ(:,2). opt XYZ(:,3));

```

```

    disp("i^调整后主索节点编号及坐标");

```

```

    disp(head(t2));

```

```

    writetable(t2, 'result.xlsx', 'Sheer. 调整后主索节点编号及坐标 L 'Range'. ['A2:D

```

```

num2str(hcight(t2)+1)]. 'WriteVhriableNames*, false);

```

```

    %促动器顶端伸缩 S:

```

```

    t3 二 table(data. x (i ndex), opt Lz);

```

```

    仙凤 “促动器顶端伸缩贸”);

```

```

    disp(head(t3));

```

```

    writetable(t3, 'result.xlsx', 'Sheet', '促动器顶端伸缩 ft', 'Range', ['A2:B,

```

```

num2str(hciglitt3)+1)]. 'WriteV_ariableNaines. false):

```

```

    xlsx check(X

```

```

end

```

```

function xlsx_check()

```

```

    t1 readtableCresult. xlsxVSheef.'理想抛物面顶点坐标 .);

```

```

    t2 readtablefresult.xlsx'. 'Sheef.'调整后主索节点编号及坐标, );

```

```

    l3 = readtableCresult.xls^'SheetV 促动器顶端伸缩 fif);

```

```

    [W. -] Po!ar.compose_rotate(Constant.alpha, Constant.beta);

```

```

    %抛物面顶端 W 旋转后的取值 Lw 要与 focal_rad.Lf>2 一样

```

```

ABR Polar. xyz2abKtl { 1,1:3} * W):
err \BR(3)-(0 -Constant. FR ratio)*ConstanL.sphere radius •ConstanL.Lf2);
disp( “检查: 抛物面顶端旋转后是否是 focal_rad+Lf/2 的取值(相减误差):f r);
assert(abs(err) < eps);

%重新构建 index
raw code = rcadtablc(, 数据 1.主索节点坐标和编号.csv, );
raw code = raw code.x_____ ;
index code string(t2.x_____);
index cellfiin(@(s)any(s = index code), raw code);
assert(all(string(raw code(index)) = index code));
dispf 构建 index 成功, );

% 0.6
assert(all(t3 {:,2}<0.6));
assert(all(t3{:,2}>0.6));
dispCLz 没有超过±0.6:成功, );

% 007
data = Data();

edge data.edge(all(index(data.edge),2),:);% 过滤节点之间的连边 index2(indcx) =
I:sum(index): %构建过滤之后的新的节点下标顺序 edge index2(edge):。使用新的节点下标对连边节点序号重定向 assert(all(all(edge)))。连边节点序号不能有空泡 raw
XYZ = data.main node coordinalc(index,:);
opt XYZ = t2{ :.2:4};
raw cdisl raw XYZ(edge(:, 1),:)-ravv XYZ(edge(:,2)j);
raw edist “ sqrt(sum(ra\v edist.*raw edist.2));
opt_edist = opt XYZ(edge(:, i),:)-opt XYZ(edge(:,2),:);
opt edist» sqrt(sum(opt_edist.*opt_edisl.2));
ratio (opt edisl./raw cdisl)* 100-100;
assert(all(ratio<0.07));
assert(all(ratio--0.07));

disp(, 节点之间的距离变化率没有超过±0.07%:成功.);

% motor 距离(下拉索) motor unit data.motor higher base(index,:)-data.motor
lower(in<lcx.): motor unit motor unit.’ sqrt(suni(niator unit.*inotor unit.2)):
assert(all(abs(sum(motor unit?motor unit.2)-1) < sqrt(eps)));

motor top data.motor_higher_base(index, :)+t3 {:,2}.*motor_unit; opt_pdist = opt XYZ-
motor top:
opt pdist = sqrt(sum(opt pdist.*optjxJisl,2));

```

```

raw_pdist = data.inain node coordinate(index:)-data.motor higher base(index:); raw_pdist
sqrt(sim(raw_pdist.*raw_p<iist,2));
assert(max(abs(ra\v pdisl-opl pdist)) sqrt(eps));
dispf 下拉索距离不变: 成功');

%最优贴合程度
oplABR = Polar.xj-z2abr(opt_XYZ*W);
err opt ABR(:,3)-Polar.para(Co»stait.Lf. opt ABR(:,2)); assert(max(abs(err)) 1e-5);
dispf 贴合程度至多 1e-5:成功');

end

function plot 2d filcrr(opt ABR) % ^ok<* DEFNU>
Lw Polar.para(Constant.LX opt_ABR(:,2));
err - opt ABR(:,3)-Lw: bar(sort(err). k): grid('on'): style('fbntsize'. 'fontname'); xticks([xticks
nuincl(err)]);
iastrprintC 图片/prob2.检验.主网节点与理想抛物面之间的球面径向误差 ·); end

function plot 2d main node dist(opt XYZ, edge, edge distSQ)
errdist opl_XYZ(edge(: J):-opt_XYZ(edge(:,2):);
cdisl 100*(sqrt(sum(crrdist.*errdist.2))/sqrt(edge distSQ)-!); bar(sort(edist). k);
gridCon');
styleCfbntsize'. 'fbntniune'); xticks([xticks nuincl(edist)]);
fastprint (, 图片/prob2.检验主网节点之间距离的变化率(百分比) ·);

end function plot 3d Iz(data, opt XYZ, opt Lz, index) % 三维促动器提升 M(Flat) raw XYZ
data.main node coordinate: raw_XYZ(index,:) = opt_XYZ;
X raw XYZ(:J); Y raw XYZ(:,2); Z raw XYZ(:,3);
Lz = NaN(size(index));
Lz(indcx) opt I-z: trisurRdata.triangle, X.Y.Z, Lz); colorbar();
view (2);
axis('equal');
style('fbntsize', 'fbnlname');
colomiapCgray');
cc - caxis();
cc(2) = 0.6; caxis(cc): box('on');
fastprint(*@片/prob2.结果.3D促动器提升量');

end

Ainction crrplot 3d(data, opt XYZ, opt ABR, index) % 三维拟合误差(intcrp) raw XYZ -

```

```

dala.main node coordinate:
raw XYZ(index,:) opt XYZ:
X raw XYZ(:,1); Y raw XYZ(:,2); Z raw XYZ(:,3); err = NaN(size(index));
err(index) abs(Polar.para(Constant.Lf. opt ABR(:,2)) - opt_ABR(:,3)); trisurfdata.triangle.
X.Y.Z. err, 'FaceColor*.'interp');
colorbar();
view(2);
axis('eqiia');
styleCfontsize', 'fbntnania');
colormapCgray'): box('on');
Ihstprint(, 图片/prob2.结果.3D理想曲面而拟合误差)
end

```

### 10.1.2 常数定义与数据处理

Constant, m: 程序用到的所有常量数值定义

%本文件原名 Constant.in

```
classdef Constant
```

```
    properties(Constant)

```

```
        FR_ratio = 0.466; % 焦径比

```

```
        sphere_radius 300.4; % 基准态球面半径, 单位是米

```

```
        spherejialfcaliber 500/2; % 基准态球面的口径的一半

```

```
        paraboloid_half_caliber = 300/2; % 工作态球面的口径的一半

```

```
        Lf= 280.85417567168855; % 第一问算出的最优焦距, 用作求解完的检查

```

```
        alpha deg2rad(36.795); % 第二问平面角

```

```
        beta = deg2rad(78.169); % 第二问仰角

```

```
    end

```

```
    methods(Static)

```

```
        fimction (sphcr rad. focal rad. angle] get rad_angle()

```

```
            sphcr rad ` Constant.sphere radius;

```

```
            sphcr cli Const am. sphere half^cal iber;

```

```
            focal rad = (1-Constant.FR_ratio) * sphcr rad;

```

```

    angle pi/2-asin(sphcr cli sphr rad): %使用半径和半口径计算出俯角, 俯角变成
仰角
    angle linspace(angle. pi-angle. 100);
    end
end
end

```

Data, m: 将题目附件 123 映射到程序内数组

%本文件原名 Data.m

```
classdef Data
```

```
    properties
```

```
        main node code %主索节点的编号. strings 转序数
```

```
        main node coordinate 0d K 索节点的坐标, XYZ motor_lower %促动器下
端点的坐标' XYZ motor higher base%促动器上端点的坐标, XYZ (基准态)
```

```
        triangle %三角形反射面板的顶点序号, string 转序数 edge%不車夏的连边顶点
序号, 按照左小右大表示, 格式是序数 end methods
```

```
    function d Data
```

```
        main node = readtable(, 数据/1. 主索节点坐标和编号.csv): d.maii node code
        string(inain node{:!});
```

```
        d.main node coordinate main nodc{:2:end};
```

```
        motor readtableC 数据/2. 促动器上下端点坐标和编号.csv) assert(all(d.main node
        code - string(motor{:1})));
```

```
        d.molor lower : motor2:4);
```

```
        d.niotor liiher base motor{:5:end};
```

```
        triangle = rcadtable(, 数据.3. 反射面板的顶点编号.csv, ); d.triangle
        string(triangle
```

```
        d.check()
```

```
        d d.convert node id());
```

```
        d.cdge = d.gcl edges. ;
```

```
    end
```

```
    function e = get edges(d)
```

```
        tri = d.triangle;
```

```
        e |tri{:1:2}:tri{:2:3}:tri{:L3}):
```



```

e(e(:,1)>e(:,2),:)=0<€(:,1)*(:,2),[2,1]);
e=unique(e.'ro\vs');
assert(size(e.1)-6525); assen(¬any(e(:,1)=e(:,2)));

end

function check(d)
    assert(isstring(d.main node code));

    assert(numel(d.main node code) == 2226);
    assert(size(d.main node coordinate. 1) - 2226);
    assert(size(d.main node coordinate.2) == 3);

    assert(ismatrix(d.motor lower)); assert(ismatrix(d.motor higher base));
    assert(size(d.motor lower. 1) == 2226);
    assert(size(d.motor lower.2) == 3);

    assert(size(d.motor higher .base. 1) == 2226); assert(size(d.motor higher base.2)
    3);

    assert(isstring(d.triangle))
    assert(size(d.triangle.1) == 4300);
    assert(size(d.triangle.2) == 3);

end

function d convert node id(d)
    %先给主节点排序, 确定一个顺序查找表 abc(i), 并且有 index(i)=[原 位]
    [abc, index] = sort(d.n main node code);

    %再给 triangle 排序, 确定一个顺序表 xyz(j), 并且有 index2(j) [tri 原位]
    [xyz, index2] = sort(d.triangle(:));
    j = 1;
    indexed = zeros(size(xyz));
    for i = 1:numel(abc)
        while abc(i) == xyz(j)
            indexed(i) = index(i);
            j = j + 1;
            if j > numel(xyz)
                break
            end
        end
    end
end
end
end

```

```

indexed(index2) indexed:
indexed = reshape(indcxd, 11, 3); assert(all(all(d.main_node_code(indexed)
d.triaijgle)));
d.main node code 1:nunel(d.niain node codc);
d.main node code d.main node code(:); d.triangle = indexed;
end

function length = get_pull_length(d)
    main d.inain node coordinate;
    higher d.motor higlier base: length sqrt(((main-higher).*(main-higher)), 11 1 !]);
end
end
end
end

```

Polar, m: 极坐标与直角坐标相互转换，抛物面径向距离

%本文件原名 Polar.m

```

classdef Polar
    methods(Static)
        function [L, dL dLf, dL dangle] para(Lf, angle)
            %
            %极坐标下，给定抛物线焦距，在特定的 omcga(angle)下 %输出相应的距离坐标 L
            %
            cos angle cos(angle); % discos) = -sin sin angle sin(angle): °odifl^sin) cos cos2
            angle cos angle.*cos angle:

            RF " (1-Constant.FR ratio),Consta»t.sphere radius; index cos angle 0 | abs(a»gle -
            pi/2) < cps:

            delta Lf.*Lf + 2.*RF.*Lf.*cos2_angle;
            L (-Lf.*sin angle+sqrt(delta))./cos2_angle; L(index) = (Lf./2+RF)./sin
            angle(index);

            ddeha_dLf 2.*Lf + 2.*RF.*cos2_angle; dL dLf -sin angle./cos2 angle:
            dL dLf= dL dLf+L./cos2 angle.*0.5./sqrt(della).• ddelta dLf;

```

```

dL_dLR(indcx) = 1 ./ (2 .* sin angle(indcx));

ddelta dangle = 2 .* RE * Lf * 2 .* cos_angle .* -sin_angle;
dL_dangle = -Lf .* cos2_angle .* cos_angle - L ./ cos2_angle .* 2 .* cos_angle .* -sin_angle;
dL_dangle = dL_dangle - t ./ cos2_angle .* 0.5 ./ sqrt(delta) .* ddelta dangle;
dL_dangle(indcx) = -L(indcx) ./ sin(uiangle(indcx)) .* cos_angle(indcx);
end

function [abr. dalpha. dbeta. dr] xyz2abr(XYZ)
%
%笛卡尔坐标转极坐标
%输出:
% abr : [alpha beta r]
% dalpha: d(alpha)/d{[X Y Z]}
% dbeta :d{beta}/d{[XYZ]}
% dr :d{r}/d{[XYZ]}
%
X = XYZ(:,1);
Y = XYZ(:,2);
Z = XYZ(:,3);
norm2_sq = X.*X+Y.*Y;
norm3_sq = norm2_sq+Z.*Z;
norm2 = sqrt(norm2_sq);
norm3 = sqrt(norm3_sq);
r = norm3;
beta = asin(-Z./r);
alpha = acos(-X./r);
alpha(Y>=0) = 2*pi-alpha(Y>=0);
alpha(mod(norm2,2)) = 0;
abr = [alpha, beta, r];
if nargin < 1
    return
end

ZEROS = zeros(size(Z));

dnonn2_sq = 2*[X Y ZEROS];

```

```

dnorm3_sq 2.*[XYZ];
dnonn2 0.5.*dnomi2 sq./nonn2;
dnorm3 = 0.5.*dnomi3 sq./norm3;
dr dnorm3;
dbeta r./nomi2.*(Z./r) + (Z./r).*(dr./r); dalpha =
abs(norm2./Y).*(-1./nomi2 ZEROS ZEROS)
(X. nonn2).*(dnonn2./nonn2);
dalpha(Y>=0.:) -dalpha(Y< 0.); dalpha( norm2.:) = 0;

```

end

function XYZ abr2xyz(abr)

```

%
%极坐标格式转笛卡尔坐标
%
alpha abr(:,1);
beta = abr(:,2);
r , abr(:,3);
Z -r.*sin(beta);
r = r.*cos(beta);
X -r.*cos(alpha);
Y -r.*sin(alpha);
XYZ = [X, Y, Z];

```

end

Hmction (W. WinvJ compose rotate(alpha. beta)

```

%
%组合旋转 (3d 旋转矩阵)
%先逆向旋转以消除 alpha 角(xOy 平面上的)
%再正向旋转以将 beta 补偿到 pi/2
%
Wxy -[
cos(alpha) sin(alpha) 0
-sin(alpha) cos(alpha) 0
0 0 1;
Wxyinv =[
cos(alpha) -sin(alpha) 0
sin(alpha) cos(alpha) 0
0 0 1;

```

```

Wxz [
    cos(pi/2-beta) 0 -sin(pi 2-beta)
    0 1 0
    sin(pi 2-beta) 0 cos(pi/2-beta)
];
Wxzinv = I
    cos(pi 2-beta) 0 sin(pi 2-beta)
    0 1 0
    -sin(pi/2-beta) 0 cos(pi/2-beta)
];
W = Wxz*Wxy;
Win\ Wxy inv*\Vxz inv: assert(all(abs(W*Winv-eye(3)) < eps()));
end
end
end
end

```

### 10.1.3 画图程序

illu.m: 与示意图有关的常量定义

```

%本文件原名 illu.m
classdef ilhi
    properties(Constant)
        p_S-[0 150];%被观测点的方位
    end
end
end

```

illutm: 剖面直角坐标示意图

%本文件原名 illul.m

```

font opts {'VerticalAlignment', 'middle'. 'HorizontalAlignment', 'center'. 'FontSize'. 10.
'Interpcter'. 'latex'};
close all
hold on

[sphcr rad. focal rad. angle] Constant.get rad ;u) gle();
p_spher (-spher_rad*cos(angle( 1)), -spher_rad*sin(angle( 1)));
p_spher end : [-sphcr rad*cos(angle(eiid)), -spher rad*sin(angle(end))];
p_fbcsl [-focal rad*cos(angle( 1)), -focal rad*sin(ang]e( 1)];
p_local end |-fbcsl rad*cos(angle(end)). -focal rad*sin(anglc(cnd))j;
v_sph [cos(pi/2+angk( I)), sin(pi/2+angle(I))J;
p_S ilhi.p_S;

```

```

p P [0 -focal rad];
p Q = [0-spher_rad+20];
P_T [0 P_P(2)*P_Q(2AP_P(2))];

hold('on');
%基准圆弧
draw 2d arc focal and base(spher rad, focal rad, angle);
%焦距比和 R300
standard distance_tag(angle. p_spher. p_spher end. p_focal, p_local end. v_sph. font_opts);

%标记定点
plot(p_S(1)>p_S(2),'k','MarkerSize', 14, 'LineWidth', 2);
plot(p_P(1)\p_P(2), 'k', 'MarkerSize', 22, 'LineWidth', 2);
plot(p_Q(1), p_Q(2), 'k', 'MarkerSize', 22, 'LineWidth', 2);
plot(p_T(1), p_T(2), 'k', 'MarkerSize', 22, 'LineWidth', 2);
plot(0,0, 'k', 'MarkerSize', 22, 'LineWidth', 2);

text(p_S(1)+10, p_S(2)+3 (V 被观测体\ font_opts(1:end-2)); text(p_S(1)*20, p_S(2),
'\boldmath$SS$, font_opts{:}); text(p_P(1)+15, p_P(2)+15. '\boldmath$P$, font_opts{:});
text(p_Q(1)+15, p_Q(2)+15. '\boldmath$Q$, font_opts{:});
text(p_T(1)+15, p_T(2)+10. '\boldmath$T$, font_opts{:});
text(0+15,0. '\boldmath$C$, font_opts{:});

% SCPQT 中轴线
plot([p_S(1), p_T(1)], [p_S(2), p_T(2)]); -k\ *LineWidth; 2);

%文字 tag
text(p_spher end(1)-40, p_spher end(2)-70, '基准球面

```

```

'Rotation'. rad2deg(angle(end-11)-pi/2), font opts {1:end-2});
text(p_focal_end(1), p_focal_end(2)-25, '魚面
'Rotation'. rad2deg(angle(end-4)-pi/2), font opts {1:end-2});
text(-160, p_T(2)+5, '抛物面的准线 font opts {1:end-2});

%PQ 长度
plot(p_P(1)+[-350 0], p_P(2)+[0 0], 'k', 'LineWidth', 2);
plot(p_Q(1)+[-350 0], p_Q(2), [0 0], 'k', 'LineWidth', 2); myquiver(8, [p_P(1)p_Q(1)]-325,
[p_P(2)p_Q(2)], 'k', 'LineWidth', 2);
text((p_P(1)+p_Q(1))/2-300, (p_P(2)+p_Q(2))/2, '\boldmath$|PQ|=L_P2$,... 'Rotation'. 90, font
opts{:});

% QT 长度以及垂直的准线
plot(p_T(1)+[-350 300], p_T(2)+[0 0], 'k', 'LineWidth', 2);
myquiver(8, [p_Q(1) p_T(1)]-325, [p_Q(2) p_T(2)], 'k', 'LineWidW', 2); text((p_Q(1)+p_T(1)),
2-300, (p_Q(2)+p_T(2))/2, '\boldmath$|QR|=L_f72$,...
'Rotation'. 90, font opts{:});
plot(p_T(1)+[0 30 30], p_T(2)-[30 30 0], 'k', 'Line Width', 2); % 垂直

%计算抛物线
Lf=2*(p_P(2)-p_Q(2));
L Polar.para(Lf, angle);

xangle fzero(@(angle)abs(Polar.para(Lf, angle))*cos(angle))-150, [pi 2 angle(end)]: xL
Polar.para(Lf, xangle);
p_x = [-xL, *cos(xangle) -xL, *sin(xangle)];

plot(-L*cos(angle), -L*sin(angle), '-k', 'LineWidth', 2);
plot(p_x(1), p_x(2), 'k', 'MarkerSize', 22);
plot([p_P(1)p_x(1)], [p_P(2)p_x(2)], 'k', 'LineWidth', 2);

plot([p_x(1) p_x(1)], [p_T(2) p_x(2)], 'k', 'LineWidth', 2);

text(p_spher_end(1)-50, p_spher_end(2)-10, '掀物面
'Rotation'. rad2deg(angle(end-14)-pi/2), font opts {1:end-2});
title'\boldmath$y^2={2L_f}-\frac{L_f}{2}-(1-0.446)RS'. Interpreter', latex^1); text(p
x(1), p_T(2)-15, '\boldmath$S_x=300/2S', font opts{:});

```

```

%
axis('eqiari):
% ylim([-525, 200]);
set(gca, 'Visible', 'off');
styleCfontnanie');
fastprintC 图片/prob 1. 示意图】. 准线等辅助线·);
(unction draw 2d arc local and base(spher rad. focal rad. angle)
%基准圆弧
plot(-spher rad*cos(angle).-spher rad*sin(ang!e). 'k', 'LineWidth', 2);
%魚面圆弧
plot(-focal rad*cos(aiiangle).-fbcal rad*sin(angle), k, 'LineWidth', 2);
end
Ainction standard distance tag(angle. p spher. p spher end. p focal, v sph. font opts)

%焦距比
plot(p_spher( 1)+(0 50»v_sph(1)], p_spher(2)+[0 50*v sph(2)]. ,:k*. 'LineWidth', 2); plol(p
jbcalf(1)+[0 50*v sph(1)]. p_fbcalf(2)+[0 50*v_sph(2)], ,:k\ 'LineWidth': 2);
myquiver(8.[p_sphcr( 1) p fbcalf( 1 )]+25*v sph( 1), [p sphcr(2)
p_focal(2)]+25*v_sph(2X': k'. 'LineWidth', , 2);
text(...
(p_spher(1)^p_focal(1)*80*v sph( 1))/2....
(p sphcr(2)+p focal(2)+80*v sph(2)) 2,...
'\boldmathSF=0.466RS', font opts{:}. 'Rotation'. rad2deg(angle( 1)));

%基准半径
plot(p_spher( 1)+100*(0 v_sph( 1)], p_spher(2)+100*[0 v sph(2)].:'k'. 'LineWidth'. 2);
plot( 0+100*(0 v_sph(1)], 0*100*(0 v_sph(2)). 'k', 'LineWidth'.
2);
myquiver( 15.(p_spher( 1) 0]+75*v_sph(1X |p sphcr(2) 0]+75*v_sph(2), 'k'. 'LineWidth'. 2);
text(...
(p_sphcr( 1)+0+180*v sph( 1))/2,...
(p_spher(2)+0+180*v sph(2))/2....
,\boldmathSR =300.4\mathrm{m}\$, fbnt opts. 'Rotation', rad2deg(anglc( 1)));

%D 500in

```



```

myquiver(20J-250 250). [-450 -450]; :k\ 'LineWidth'. 2);
plot([-250 p_spher(1)J. [-460 p_spher_end(2)].:k\ Tine Width'. 2): plot([ 250 p
spher end(1)], [-460 p_spher_end(2)],*:k', 'LineWidth', 2); text(0, -440.
'\bo!dmathSD 500\mathmi {m JS^1. fbnt_opls {:});
end
Hinction inyquivciXr.x.y.varargin)
rad atan2(y(1)-y(2),x(1)-x(2));
rad |rad+deg2rad(30). rad-dcg2rad(30);
plot(x,y,varargin{:});
L sqrt((x(1)-x(2))*(x(1)- x(2))+(y(1)-y(2))*(y(1)-y(2)))/r; plot(x(1)-
[0.L]*cos(rad( 1))y( 1)-[0.L]*sin(rad( 1 )),varargin{:});
plot(x(1)-[0.L]*cos(rad(2))x(1)-[0.L]*sin(rad(2)),varargin{:});
plot(x(2)+[0.L]*cos(rad( 1 )),y(2) [0.L]*sin(rad( 1 )),varargin{:});
plot(x(2)+[0.L]*cos(rad(2)),y(2)+[0.L]*sin(rad(2)),varargin{:});
end

```

## i i lu2.m:剖面极坐标示意图

```

%本文件原名 illu2.ni
%剖面示意图
font opts = {'VerticalAlignment'. 'middle', 'HorizontalAlignment'. 'center', 'FontSize'. 18.
'Interpreter', 'latex'};
close all
(spher rad. local rad. angle) - Constant.get rad aiigle());
p_S = [pi/2 ilhi.p S(2)];
p_P [-pi/2 focal rad];
p_Q = [-pi/2 spher rad-20];
p_T [-pi,2 p_P(2)+2*(P_Q(2)>p_P(2))];
%基准圆弧
polarplot(angle^pi,angle.,0^spher rad. 'k'. 'LineWidth'. 2); hold('oiT');
%焦面圆弧
polarplot(angle-pi,angle.*0+fbcal rad, k, 'LineWidth', 2);
%标记定点
polarplot(p_S(1), p_S(2)*. 'MarkerSize'. 14. 'LineWidth'. 2); polarplot(p_P(1), p_P(2)).'k',
'MarkerSize', 22. 'LineWidth'. 2); p< larplot(p_Q(1), p_Q(2)).'k'. 'MarkerSize', 22. 'LineWidth'.
2);
polarplot(p_T(1), p_T(2)).'k'. 'MarkerSize'. 22. 'LineWidth'. 2);
polarplot(0.0.'k'. 'MarkerSize; 22. 'LineWidth'. 2);
text(p S(1)>0.20. p S(2). "boldmathSSS'. font opts{:}); text(p P( 1 )+0.12. p P(2)A 18,

```

```

*boldmathSP$, font opts{:}); text(p_Q(1)+0.08, p_Q(2)-23, '^boldmathSQS', font_opts{:});
text(p_T(U+0.06, p_T(2)48, ^boldmathSTS*, font opts{:});
text(0.15, -30, ^boldmathSCS', font opts{:});
% SCPQT 中轴线
polarplot([p_S(1), p_T(1)], [p_S(2), p_T(2)]A-k\ tineWidth: 2);
%计算抛物线
Lf 2*(p_Q(2)-p_P(2));
I. = Polar.para(LC angle):
xangle lzero(@(angle)abs(Polar.para(LC angle)*cos(angle))-150. (pi/2*1.01.
anglc(end)J):
xL Polar.para(Lf, xangle);
theta linspace(-xangle, xangle-pi, 150);
r Polar.para(Lf, theta+pi);
r(2:2:end) = spher rad:
polarplot(reshape(theta,2,[],X reshape(r,2,[],) '-k', 'linewidth', 2, 'color'. [0.7.0.7.0.7]);
polarplot([0 0], [0 p T(2)], '*k', linewidth: 2);
polaqlot(linspace(0.-xangle+pi.100). repmat(60.I.100). 'k', 'linewidth'. 2); text((-xangle ^pi)^2-
O.O5. 90. ^boldmathS omegaS', font opts{:});
polarplot(angle-t-pi, L, '-.k', 'LineWidth'. 2);
polarplot(xangle^pi, xL, 'k', 'MarkerSize', 22);
polarplot(-xangle, xL, 'k', 'MarkerSize'. 22);
polarplot(-xangle xangle], [-p S(2) spher rad]. 'k', 'linewidth'. 2); polarplot([xangle xangle]+pi,
[-p S(2) spher rad]. 'k', 'linewidth'. 2);
% polarplot(-p_x(1), p_x(2), 'k', 'MarkerSize', 22);

set(gca,'R.AxisLocation(0);
rlin ] ([0, p T(2));
%极坐标
style('bntsize', 'bntname');
t'asprinl.图片/probl.示意图 2. 极坐标下的 omega 角与球面径向误差平方积分'):
style, m:为输出图片统一文字大小和字体
%本文件原名 style.m
Ainction style(varargin)

```

```

for v varargin
    switch lower(char(v)) case 'fbnsize' set(gca(). 'ibnsize'. 22); case 'fontname*' set(gca(),
        'fontname', 'times new roman*'); end
end
end fastprint, m: 简易绘图输出

%本文件原名 faslprint.in
function r faslprint(filename)
    assert(isequal(class(filename),'char') (isequal(class(filename),'string') &&
isscalar(filename)))
    if isstring(filename)
        filename = char(filename);
    end
    if numel(filename) >= 4 && isequal(filename(end-3:end),'.png')
        filename = filename(1 :end-4);
    end
    filename [filename., '.datcstr(no\v),yyyymm.dd. 1111 MM).'.png]:

```

```
print('-dpng'-'r3(X)'filename):  
disp(filename):  
r - @()system(('open 'filename]):  
end
```

E110 -131687 -0729 -269792 E134 -107292 96S43 -263744 D161 -699W -137485 -  
 257497  
 E111 -128661 10602 -271031 C147 69831 -126810 -263627 D162 - 79717 -130BCO -  
 2S8154  
 E112 125.417 21 933 \*271854 C148 60017 13336S -26278 0163 \*123616 -2S8380  
 fu3 -121976 33207 - 272263 CU9 S01S3 -13972S -261393 D164 -99778 -116222 -  
 fl4 -118203 44829 -2722S8 C150 40 096 -145962 -259.671 DIGS -109138 -109037 -257511  
 E115 -114368 55976 \*271847 C151 30114 -151914 -2S7579 D166 -118329 -101683 -2S64M  
 0167 -127455 94140 -254951 0196 -50160 -171307 241798 D240 -79883 -17274S -232873  
 DIe8 -136381 86451 -253046 0197 £0213 -165169 -243617 D241 -90243 -166185 -233728  
 0169 -145047 78753 \*250737 0198 -70169 -158794 -245062 D242 -ICO 462 -159371 -234213  
 D170 -153496 70937 -2480S2 0199 -80018 -1S2184 -246187 D243 -110578 -152172 -234 382  
 0171 -160152 62573 -246125 0200 -90254 -H52S7 -246788 D244 -120.516 -144 734 -234164  
 EIM -16661! 54127 ■243891 0201 -100340 -138090 -246984 D245 -130127 -137091 -233612  
 E1SS -16635SOS -246149 D2Q2 -U02S9 -  
 E1S« -165900 32827 -248095 0203 -119994 -123083 -246138 D247 -148675 -1216 鉗 -231136  
 £157 463661 21537 -250782 D204 129301 115743 244 994 D248 457590 -113960 -229255  
 E158 -161176 10212 -253067 D205 -138409 -108224 -243473 D249 -1G4 791 -106029 -2280TO  
 EIS9 -158 478 1260 -254986 0206 -147306 -100539 -241579 D2S0 -171794 -97961 -226607  
 EI60 -155536 12 730 \*256488 0207 155982 92693 -239322 D265 -110495 162601 -227711  
 E161 -152435 24087 -257.5S4 0208 -162912 -84 496 -237795 D266 -120526 -155271 -  
 E162 - 35.409 -258233 D2M 169636 -76185 -23S979 D267 -130 234 -147779 -227&  
 E163 -145419 47.257 \*258498 0210 176155 -67773 -233876 D268 -139.753 -140024 -226557  
 E164 -141498 59029 -258 X8 E191 -182469 -59277 -231490 D269 -149020 -132057 -  
 Ei6S -137602 70 187 -257793 EIW -182398 -48703 -233923  
 E166 -133.499 81247 -256882 E193 182C83 -38073 -236C80  
 D174 -200W -176.709 -242524 E1W -181548 -27396 -237938  
 0175 -30061 -172.749 -244147 E195 -180 789 -16688 -239498  
 0176 -40094 -166W1 -246545 Ei96 178395 -52\*0 -241779  
 D177 -SO 193 -160938 -248S50 E197 -175733 6215 -243696  
 0178 -60230 -154 674 -250208 E198 -172808 17663 -24S244  
 0179 70139 B8183 -251501 E199 -169630 29 090 -246419  
 0180 -79924 -U1593 -252350 E2Q0 166254 40987 -247114  
 0181 -90096 -J34.5S0 -2S2763 E201 -162.618 52.822 -247412  
 0182 -100133 127.250 -252762 0217 -60050 175 湖 -236 775  
 D183 -109967 -119743 -252340 0218 -70090 169078 -238407  
 D184 -119247 -112.462 -251.475 0219 -79.999 -162 569 -239.688  
 OJ85 -128464 -105032 -250150 0220 -90.303 155893 240381  
 0186 -137495 -97407 -248440 D221 -100 470 -148831 -240769  
 0187 \*6286 -89639 -246360 D222 1WS06 -UIS16 -240752  
 0188 -154851 -81875 -243861 0223 120343 134002 -240324  
 -161644 -73596 -242134 0224 -129850 126245 -239579  
 0190 -168245 -65199 -240110 D226 -139C66 -118804 -238237  
 \*174623 56726 -237803 0226 1\*8097 -111.175 -236522  
 \*174.468 M6136 -240147 0227 156894 -103398 -234 448  
 E174 -174 075 -35466 -242206 0228 -163967 -95368 -233076

E17S	-173458	-24777	-	0229	-170845	-87128	-231450
E176	171102	-13457	•24646	D23	177 527	-78757	-229541
E177	-168	-2002	•2485S	D23	-183991	-70313	227352
£178	-165754	9.482	-	E213	-190006	-40.617	-229671
E179	-162684	20922	-251616	£214	■189.54	-29928	231641
E180	-159479	32274	-	£215	-188862	-19222	-233309
E181	-155	44.145	-252990	£216	187892	-8569	-234.729
E182	-152170	55.976	•25306	£217	-185373	2895	-23684!
E183	-U8122	6768S	•	£218	-182582	W387	-238596
D195	-40023	-177194	-	£219	-1795*2	2S836	-239978