

# 龙门吊问题的数学建模

## 摘要

对于港口装卸货物集装箱的龙门吊的吊运过程的求解与优化问题,本文建立了货物运动的动力学模型和摆角、效率的优化模型。首先,对于货物在吊运过程中的运动问题,由于这是一种完整约束下的具有一个自由度的平面运动,可以通过使用分析力学中的拉格朗日方程,解出在吊车处于4段不同运动状态下的货物的以摆角 $\theta$ 为广义坐标的运动微分方程。使用 Matlab 软件数值求解微分方程可以得到货物在吊运过程中的运动状态。

对于前三问提出的不同的优化问题,我们可以通过不同的优化方法解出各问题的较优解。对于第一问最大摆角的优化问题,我们使用多起点寻找局部最小值的方法搜索可行的较优解,并取了其中摆角最小的一个作为答案:当 $(t_1, t_2, t_3, t_4) = (5.57s, 15.97s, 5.2s, 93.02s)$ 时,货物在BC段的最大摆角 $\theta_{max} = 0.54^\circ$ 。第二问在第一问的基础上增加了对吊运效率的考虑,由于货物质量固定,我们把吊运时间看作效率,通过对第一问搜出的许多局部较优解使用逼近理想解排序 (TOPSIS) 法和熵权法确定其中的较优解,我们得到的一组较优解为: $(t_1, t_2, t_3, t_4) = (5.57s, 16.00s, 5.08s, 61.25s)$ ,  $\theta_{max} = 0.96^\circ, t = 87.9s$ 。第三问将加速度的取值考虑在内,且吊运过程中缆绳拉力不得超过其最大承载力。由于该问题中货物质量为可变值,我们重定义吊运效率为最大起吊货物质量与吊运所需时间之比。我们采用改进的非支配排序遗传算法 (NSGA-II) 在限制条件内搜索目标函数的帕累托最优解,对于解集中的每个解,也采用与第二问相同的熵权法和逼近理想解排序法确定其在解集中排序。根据排序结果我们得到一组较优解为: $a = 0.745m/s^2, (t_1, t_2, t_3, t_4) = (8s, 2.01s, 7.55s, 50s)$ , BC段最大摆角和吊运时间分别为 $\theta_{max} = 0.86^\circ, t = 61.69s$  最大起吊货物重量 $m_{max} = 19784kg$ 。

第四问要求设计一个可以直观演示吊运过程的图形界面,我们通过使用 Matlab 的图形界面设计工具,设计了一个能够在给定 $a, t_i$ 的情况下给出货物的运动演示图的图形窗口。

**关键字:** 分析力学 微分方程数值求解 多起点优化 遗传算法 GUI 演示设计

## 一、问题重述

某公司一货运码头，配备带有动力驱动的货运吊车。现需将货物从货运船的起吊点吊运到货运码头的终点，并且按照规定的方式进行运动，提出如下问题：

(1) 在不考虑吊车缆绳承载力的情况下，确定吊车运行方式，即建立模型，确定时间的大小，使得货物从岸边到终点，其间摆动最小，且整个吊运时间不超过 2 分钟；

(2) 在第一问条件下，如果同时计及吊运效率，建立模型确定时间的大小，使货物从岸边到终点不仅摆动幅度小而且整体效率高；

(3) 考虑缆绳的最大承载力，并兼顾摆动大小和效率，请建立模型，确定吊车运行方式，即确定加速度的大小、时间，并确定最大起吊货物重量；

(4) 考虑实际问题的需要，为该公司设计一套图形计算机演示系统，直观演示龙门吊吊运过程。

## 二、符号说明

符号	意义
$a$	吊车加速度大小 ( $m/s^{-2}$ )
$g$	重力加速度, $9.8m/s^{-2}$
$x(t), y(t)$	货物在时刻 $t$ 的位置坐标
$v_x(t), v_y(t)$	货物在时刻 $t$ 的水平、竖直速度
$\theta(t)$	吊绳在时刻 $t$ 与竖直方向夹角，顺时针为正
$\theta_{max}$	从 B 点到 C 点货物运动过程中的最大摆角
$v_{4xmax}$	货物到终点处的水平速度
$x_a(t), v_a(t)$	吊车在时刻 $t$ 时的位置坐标，运动速度
$t_{1,2,3,4}$	第 1-4 段运动分别持续的时间 ( $s$ )
$T_i, i = 1, 2, 3, 4$	第 $i$ 段运动结束的时刻 ( $s$ )
$F(t)$	吊绳在时刻 $t$ 时的拉力大小 ( $N$ )
$F_{max}$	吊绳的最大拉力，值为 $20000g \quad N$

### 三、 问题分析

题目给出的龙门吊问题可以看作二维的质点运动问题，因为吊车只沿直线运动，货物没有初速度，在吊车前进的过程中货物只会在包含吊车运动轨迹的与地面垂直的平面内运动。货物的运动也不是平面内的自由运动，而是受到了来自吊车和吊绳的约束，因此货物的运动自由度实际上只有一个，可以用吊绳与竖直方向的夹角  $\theta$  表示。由于货物的运动受到的约束是完整约束，可以使用分析力学中的拉格朗日方程可以解出其运动微分方程，微分方程可以用软件进行数值求解，这样可以建立货物运动的动力学模型，可以在给定所有参数时解出整个过程的运动状态。

前三问是在不同的参数、目标下的优化问题，可以使用优化模型求解。第一问在加速度给定时仅要求对货物的摆动角度进行优化，第二问在第一问的基础上同时要对整个过程的效率进行优化，第三问比第二问增加了加速度这一参数，并且要求给出能够吊运货物的最大质量。首先考虑效率的定义，由于写出的运动微分方程与货物的质量无关，在相同的吊运方式下，货物质量仅仅受到吊绳能够承受的最大拉力的限制，因此我们认为效率只与时间相关，吊运时间越短效率越高。我们可以根据运动的动力学模型写出最大摆角、吊运时间和实际吊绳最大拉力与  $t_{1,2,3,4}$  的程序函数，然后通过调用不同的优化算法来求解不同的优化问题。对于较为简单的前两问，可以使用多起点搜索最小值的方法；对于第三问，可以使用遗传算法、退火算法等优化算法进行求解。

对于第四问要求的设计展示龙门吊吊运过程的图形界面 (GUI), 我们可以使用 `matlab` 的图形设计工具进行设计。

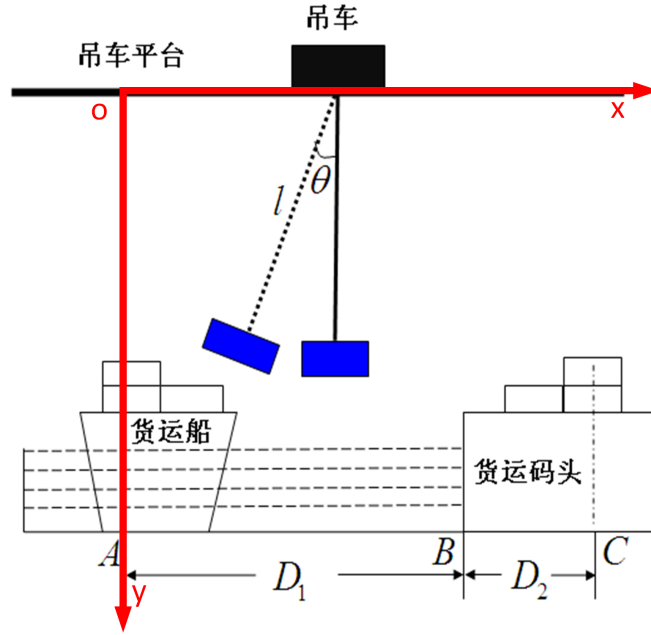
### 四、 模型假设

1. 把货物当作质点；
2. 货物只做平面运动；
3. 吊绳的质量忽略不计；
4. 货物摆动过程中不计各种阻力；
5. 整个过程中吊绳始终处于拉紧状态，且长度不变；
6. 吊运的效率仅取决于吊运过程耗费的时间；
7. 吊车到达终点后会立刻刹车将速度降至 0。

### 五、 建立模型

#### 5.1 货物运动的动力学模型

如图所示建立坐标系：



设吊车位置坐标为  $x_a$ , 速度为  $v_a$ ; 货物的位置  $(x, y)$ , 速度  $v$ , 水平速度  $v_x$ , 缆绳与竖直方向的角度为  $\theta$  (顺时针为正)。令  $T_1 = t_1$ ,  $T_2 = t_1 + t_2$ ,  $T_3 = t_1 + t_2 + t_3$ ,  $T_4 = t_1 + t_2 + t_3 + t_4$ 。吊绳能承受的最大拉力  $T_{max} = 20000g$ ,  $g$  取  $9.8m/s^2$ 。AB 间距离为  $D_1$ , CD 间距离为  $D_2$ 。

取货物为研究对象, 用分析力学方法, 取广义坐标  $\theta$

(1) 当  $0 \leq t \leq T_1$  时, 吊车匀加速运动, 对于货物有如下拉格朗日函数:

$$L_1 = \frac{m}{2} \left( l^2 \dot{\theta}^2 - 2al\dot{\theta} \cos \theta + a^2 t^2 \right) + mgl \cos \theta$$

代入拉格朗日方程

$$\frac{d}{dt} \left( \frac{\partial L_1}{\partial \dot{\theta}} \right) - \frac{\partial L_1}{\partial \theta} = 0$$

得到运动微分方程:

$$l\ddot{\theta} + a\dot{\theta} \sin \theta - a \cos \theta + g \sin \theta = 0$$

$$\text{初始条件 } \theta|_{t=0} = 0, \quad \dot{\theta}|_{t=0} = 0$$

由  $\theta(t), 0 \leq t \leq T_1$ , 有

$$\begin{cases} x = x_a - l\theta \sin \theta = \frac{a}{2}t^2 - l\sin \theta \\ v_x = v_a - l\dot{\theta} \cos \theta = at - l\dot{\theta} \cos \theta \end{cases}$$

(2) 当  $T_1 \leq t \leq T_2$  时, 吊车匀速运动, 对于货物同上可得运动微分方程:

$$l\ddot{\theta} - v_{a2}\dot{\theta} \sin \theta + (v_{a2} + g) \sin \theta = 0$$

$$\text{初始条件 } \begin{cases} \theta|_{t=T_1^+} = \theta|_{t=T_1^-} \\ \dot{\theta}|_{t=T_1^+} = \dot{\theta}|_{t=T_1^-} \end{cases}$$

此时有:

$$\begin{cases} x = x_a - l\theta \sin \theta = \frac{a}{2}T_1^2 + aT_1(t - T_1) - l \sin \theta \\ v_x = v_a - l\dot{\theta} \cos \theta = aT_1 - l\dot{\theta} \cos \theta \end{cases}$$

(3) 当  $T_2 \leq t \leq T_3$  时, 吊车匀减速运动, 对于货物同 (1) 可得运动微分方程:

$$l\ddot{\theta} + a(T_1 + T_2 - t)\dot{\theta} \sin \theta + a \cos \theta + g \sin \theta = 0$$

$$\text{初始条件} \begin{cases} \theta|_{t=T_2^+} = \theta|_{t=T_2^-} \\ \dot{\theta}|_{t=T_2^+} = \dot{\theta}|_{t=T_2^-} \end{cases}$$

此时有:

$$\begin{cases} v_x = a(T_1 + T_2 - t) - l\dot{\theta} \cos \theta \\ x = x_2 + aT_1(t - T_2) - \frac{a}{2}(t - T_2)^2 - l \sin \theta \end{cases}$$

并且需要满足在匀减速运动结束时吊车刚好到达 B 点, 即

$$\frac{a}{2}t_1^2 + at_1t_2 + \frac{a}{2}t_3(2t_1 - t_3) = D_1$$

(4) 当  $T_3 \leq t \leq T_4$  时, 吊车匀速运动, 对于货物同 (1) 可得运动微分方程:

$$l\ddot{\theta} - v_{a4}\dot{\theta} \sin \theta + (v_{a4} + g) \sin \theta = 0$$

$$\text{初始条件} \begin{cases} \theta|_{t=T_3^+} = \theta|_{t=T_3^-} \\ \dot{\theta}|_{t=T_3^+} = \dot{\theta}|_{t=T_3^-} \end{cases}$$

此时有:

$$\begin{cases} v_x = a(T_1 + T_2 - T_3) - l\dot{\theta} \cos \theta \\ x = D_1 + a(T_1 + T_2 - T_3)(t - T_3) - l \sin \theta \end{cases}$$

从 B 点到 C 点货物运动过程中的最大摆角  $\theta_{\max} = \max \theta(t)$ ,  $T_3 \leq t \leq T_4$ ,

对于货物最终的水平速度, 取第四段匀速过程中货物水平速度绝对值的最大值  $v_{4x\max} = \max \{v_x, T_3 \leq t \leq T_4\}$ , 要求  $v_{4x\max} \leq 0.5m/s$ 。

运动全过程中货物的竖直速度  $v_y = -l\dot{\theta} \sin \theta$ ,  $0 \leq t \leq T_4$ , 对速度求导可得水平、竖直方向的加速度, 由此可以计算整个运动过程中每一时刻的拉力:

$$\begin{cases} F_x = ma_x \\ F_y = mg - ma_y \\ F = \sqrt{F_x^2 + F_y^2} \end{cases}$$

要求  $F \leq F_{\max}, 0 \leq t \leq T_4$

## 5.2 对摆角、效率的优化模型

### 5.2.1 仅考虑摆角的优化模型

针对第一问提出的问题，在加速度确定的情况下，考虑到题目中对运动的限制，即在匀减速运动结束时吊车要正要运动到 B 点：

$$x_{a1} + x_{a2} + x_{a3} = \frac{a}{2}t_1^2 + at_1t_2 + \frac{a}{2}t_3(2t_1 - t_3) = D_1$$

可以把参数  $t_{1,2,3,4}$  简化为只需要两个参数  $t_1, t_3$ 。通过上节建立的动力学模型可以求出在 BC 段运动过程中的最大摆角  $\theta_{max}$ 。由此可以建立单目标优化模型如下：

$$\begin{aligned} & \min \theta_{max} \\ & s.t. \begin{cases} v_{4xmax} < 0.5m/s \\ T_4 \leq 120s \\ F < F_{max}, 0 \leq t \leq T_4 \end{cases} \end{aligned}$$

### 5.2.2 考虑摆角、效率的优化模型

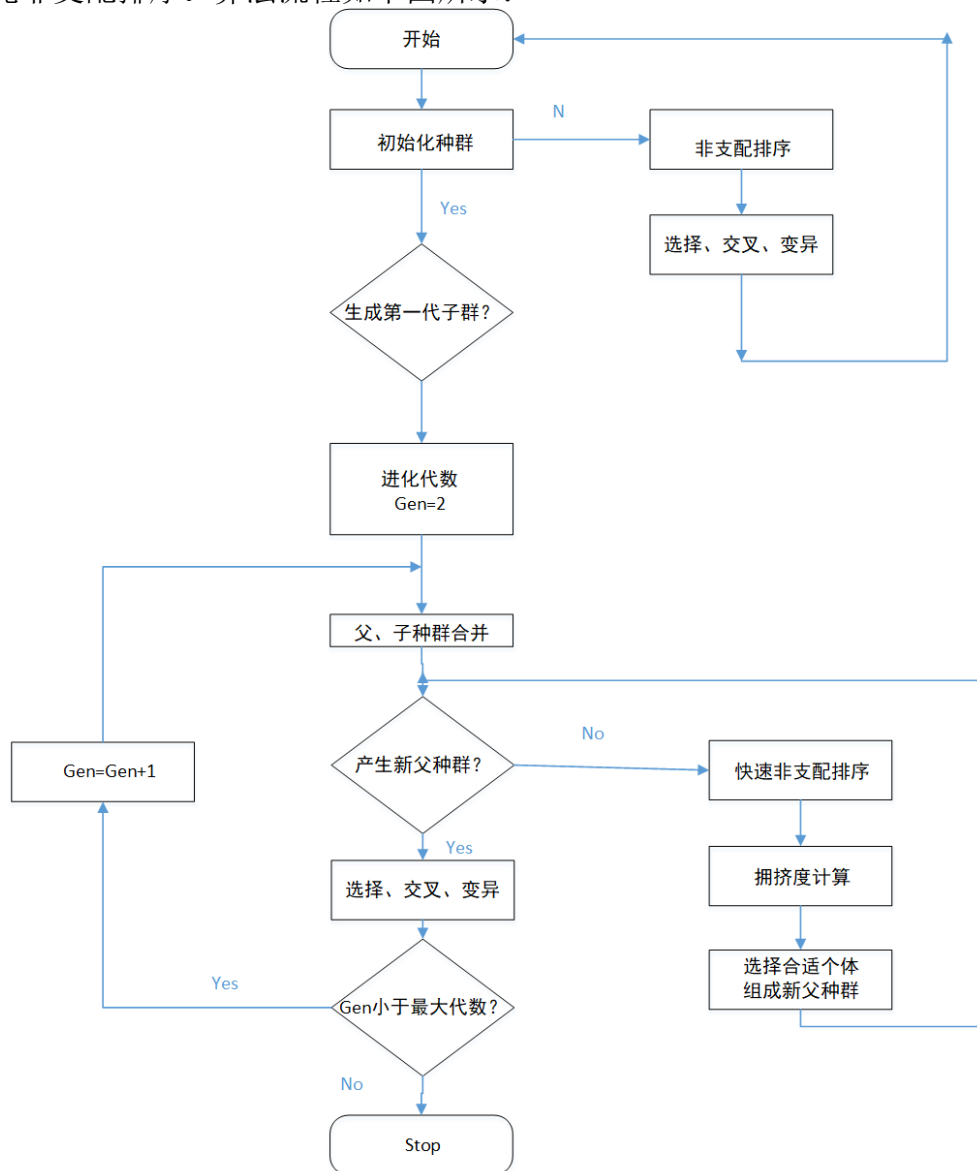
第二问在第一问的基础上同时考虑 BC 段的最大摆角  $\theta_{max}$  和吊运效率的，优化模型由单目标变为多目标。由于该问题中的货物质量固定为  $m = 6000kg$ ，我们便把效率定义为时间的线性函数，吊运所用的总时间越短，调运效率越高。在用第一问的方法已经求得若干有不同的 BC 段最大摆角、吊运时间的解的情况下，可以通过逼近理想解排序模型（TOPSIS）找出相对较优的一种解。我们可以首先找到所有解当中摆角  $\theta_{max}$  和时间  $t$  的最大值与最小值，使用极差标准化方法得到这若干种情形下的摆角和时间的效益型属性值，取理想解为两属性都取最小值，负理想解为两属性都取最大值，调整两个目标的权重  $w_1, w_2$ ，解出每种情形下与理想解的贴近度  $\frac{d^+}{d^+ + d^-}$ ，取与最理想解最贴近近的解作为该问题的答案。

### 5.2.3 考虑加速度时对摆角、效率的优化模型

对于第三问提出的问题，自变量比前两问增加了一个加速度  $a, < a \leq 1$ ，且货物运输的过程中缆绳拉力不得超过最大承载力。我们定义效率  $\eta$  为平均单位内时间吊运货物的最大质量，即  $\eta = m_{max}/T_4$ ，则优化模型如下所示：

$$\begin{aligned} & \max \eta \\ & \min \theta \\ & s.t. \begin{cases} F_m < F_{max}, 0 \leq t \leq T_4 \\ T_4 \leq 120s \\ v_4 < 0.5m/s \end{cases} \end{aligned}$$

由 5.1 中提出的货物运动的动力学模型可知，货物的运动方程与货物质量  $m$  无关，在货物运动的加速度已经确定的情况下，吊绳拉力的大小正比于质量  $m$ 。因此当运动过程中吊绳的最大拉力等于  $T_{max}$  时，对应的货物质量就是当前吊运方式能吊运的最大质量。即  $m_{max} = \frac{T_{max}}{a_{max}}$ ，其中  $a_{max}$  是整个运动过程中去除重力加速度后的加速度最大值。对于该多目标问题，我们在 5.1 中模型的基础上采用 NSGA-II（Nondominated Sorting Genetic Algorithm-II，非支配排序遗传算法二）进行求解。对于现有规模为种群，其通过帕累托支配关系将种群分为  $k$  个非支配层，每个非支配层里的个体互无支配关系，并使用联赛选择法选择较优个体，其规则为：非支配等级越小的个体越优，若两个个体非支配等级相同，则拥挤度越大的越优。优质个体选作新父种群，其产生的子群与该新父种群一起进行下一轮非支配排序。算法流程如下图所示：



最后一轮迭代结束后，我们能够得到一组互不支配的帕累托占优解。类似问题二的求解方法，我们使用熵权法确定各目标的权值，并通过逼近理想排序在解集种找出相对较优的一组解作为该问题答案。

## 六、模型求解

我们首先将货物运动的动力学模型使用 matlab 编写为一个函数脚本，其中四段运动过程的常微分方程使用 ode45 求解器进行数值求解，当给定的参数  $a, t_1, t_3$  满足题目中的限制要求，则计算出这种运动样式下的最大摆角和吊运时间，否则返回一个设定的极大值。我们针对前三问不同的优化目标对函数脚本进行对应的修改，以满足不同的求解需求。函数脚本 func.m 见附录。

### 6.1 第一问的求解

考虑到动力学模型的最大摆角—— $(t_1, t_3)$  不是线性关系，在特定区间的取值由于不满足模型限制而返回极大值，因此很难通过一次简单的优化过程找到最优解。为此我们使用了在多个起点运行区域极小值的搜索函数，具体而言是使用了 matlab 的 fmin-search 函数，在  $t_1 = [3, 12]$ ,  $t_3 = [t_1 - 2, t_1]$  的范围均匀取初始点多次运行，找到若干局部较优解，再从中取  $\theta_{max}$  最小的解作为最终解。运行后得到当  $(t_1, t_2, t_3, t_4) = (5.57s, 15.97s, 5.2s, 93.02s)$  时，货物在 BC 段的最大摆角  $\theta_{max} = 0.54^\circ$ 。这种情况下的货物  $\theta(t), x(t)$  曲线如下图所示。(不同颜色的线条代表 4 段不同的运动过程)

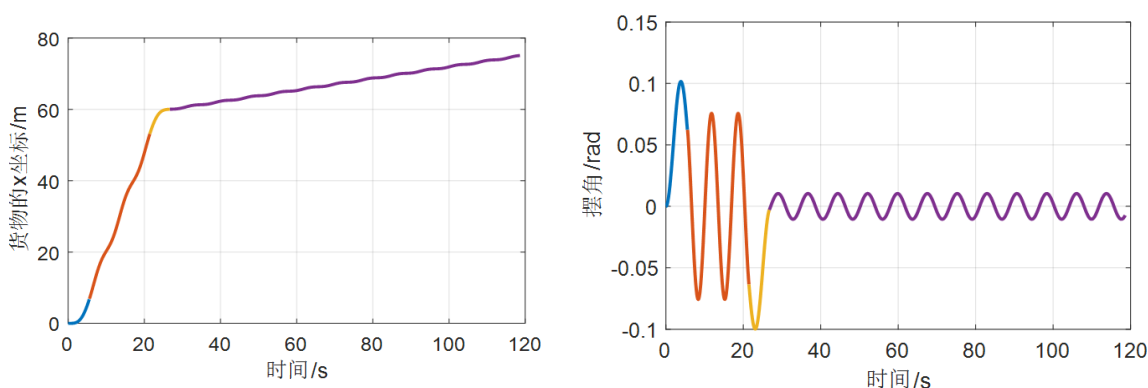


图 1 问题一解的  $\theta(t)$   $x(t)$  图像

### 6.2 第二问的求解

第二问在第一问的基础上增加了对吊运效率的考虑，也就是使吊运时间越短越好。为此，我们采取了先找出最大摆角接近第一问解得的最小值的若干解，再对这些解的摆角和时间使用逼近理想解排序法进行排序，取结果最好的一个作为问题的解。我们使用第一问的程序找出所有 BC 段最大摆角小于  $1.89^\circ$  的解，即找到  $\theta_{max}$  在三倍最小值的范围内的所有解，共得到了 30 组解，观察发现其中摆角最大的是  $1.43^\circ$ ，且其用时为两分钟；其余的解都要比这个更好；我们又发现在所有的解当中，用时最短的是  $82.14s$ 。因



此我们以最小值  $\theta_{max} = 0.63^\circ, t = 82.14s$ , 和最大值  $\theta_{max} = 1.43^\circ, t = 120s$  使用极差标化解出每种解的效益性指标属性值  $a_{\theta i} = \frac{\theta_i - 0.63^\circ}{1.43^\circ - 0.63^\circ}$ ,  $a_{ti} = \frac{t_i - 82.14s}{120s - 82.14s}$ 。令理想解为两属性值都取最小值, 负理想解为两属性值都取最大值。取对摆角和时间的权值分别为  $(w_\theta, w_t) = (0.6, 0.4), (0.5, 0.5), (0.7, 0.3)$  三种权值组合, 求得每组解与理想解与负理想解的加权欧氏距离  $d^+, d^-$ , 最后得到每组解与理想解的贴近度  $c_i = \frac{d^-}{d^+ + d^-}$ , 对其排序后得到贴近度最大的一组解如下:

$$(t_1, t_2, t_3, t_4) = (5.57s, 16.00s, 5.08s, 61.25s), \quad \theta_{max} = 0.96^\circ, t = 87.9s$$

。该吊运过程的  $\theta(t), x(t)$  曲线如下图所示。

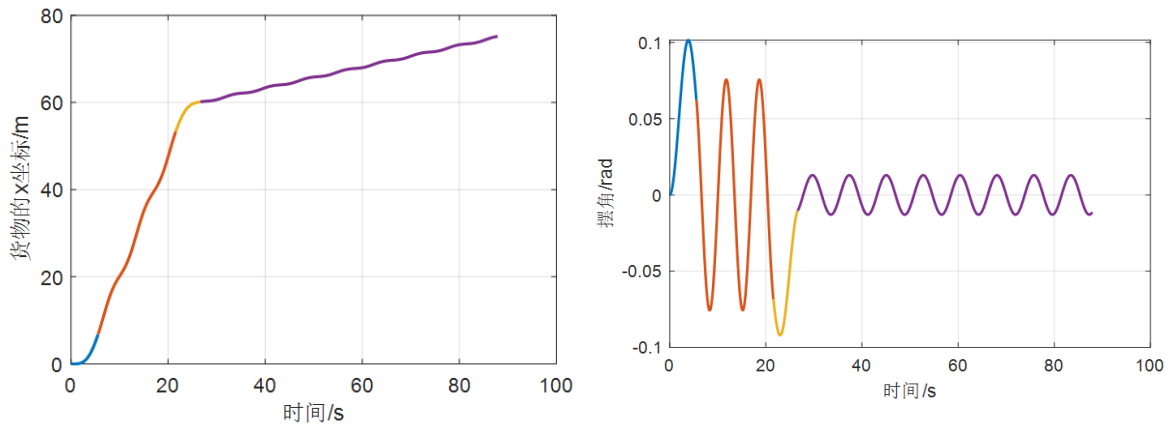
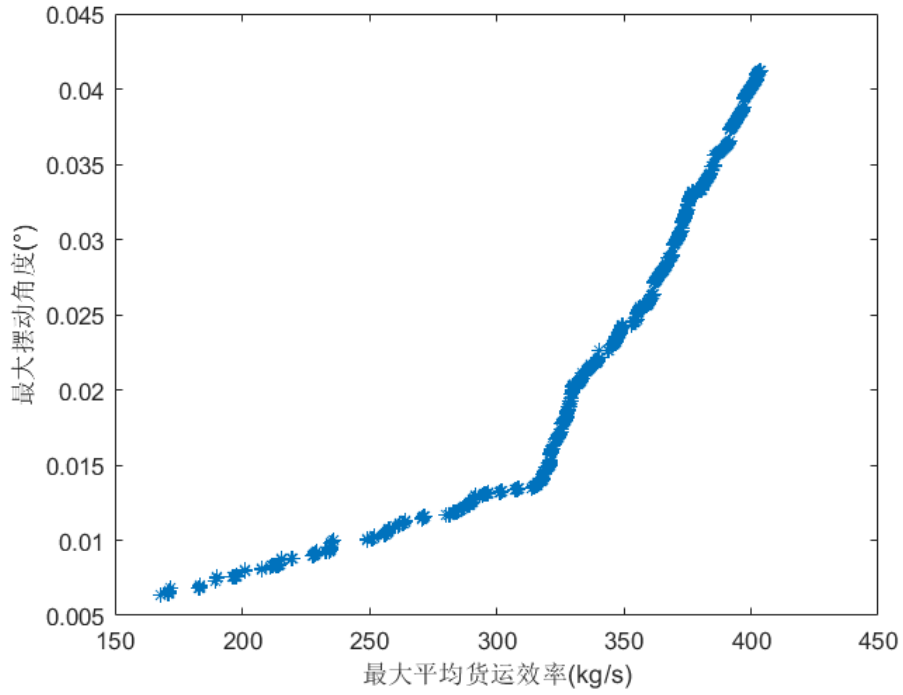


图 2 问题二解的  $\theta(t)$   $x(t)$  图像

### 6.3 第三问的求解

考虑到当  $(t_1, t_3)$  取值不当时模型会由于不满足限制而返回极大值, 因此我们取  $t_1 = [3, 10], t_2 = [3, 10], a = [0, 1]$  作为搜索空间, 设置种群规模  $n = 1000$ , 最大遗传代数 100 代, 迭代求出帕累托解集如下图所示:



我们用熵权法确定各目标权值。设解集中有  $n$  个解，有  $m$  个目标函数。第  $i$  个解的第  $j$  个指标  $d_{ij}$ ，我们对其进行向量归一化：

$$p_{ij} = \frac{d_{ij}}{\sum_{i=1}^n d_{ij}}$$

其中各指标的熵值为

$$e_j = -k \sum_{i=1}^n p_{ij} \ln p_{ij}$$

其中  $k = 1/\ln n$ 。由此，各指标的权重为：

$$h_j = \frac{1 - e_j}{\sum_{k=1}^m (1 - e_k)}$$

我们首先将解集中的角度转化由成本型指标转化为效益性指标，即令  $\theta'_i = -\theta_i$ 。随后对每种指标采用  $Z$  标准化，即

$$\begin{cases} \theta_{zi} = \frac{\theta_i - \mu_\theta}{\sigma_{\theta_i}} \\ \eta_{zi} = \frac{\eta_i - \mu_\eta}{\sigma_{\eta_i}} \end{cases}$$

其中  $\mu_\theta, \mu_\eta, \sigma_{\theta_i}, \sigma_{\eta_i}$  分别为  $\theta_i, \eta_i$  的均值和标准差。由此得到最大平均货运效率和最大摆动角度的权值分别为  $w_\eta = 0.554, w_\theta = 0.447$ 。使用 TOPSIS 法和对加权后的解进行排序，排序前十的结果如下表：

$a(m/s^2)$	$t_1/s$	$t_3/s$	$\eta_i$	$\theta_i/rad$	rank
0.74	8.01	7.55	320.77	0.0150	1.00
0.74	8.01	7.55	320.70	0.0150	2.00
0.77	7.94	7.50	318.27	0.0141	3.00
0.77	7.94	7.50	318.10	0.0140	4.00
0.74	8.01	7.56	320.50	0.0149	5.00
0.74	8.01	7.56	320.89	0.0151	6.00
0.77	7.94	7.51	318.31	0.0141	7.00
0.74	8.02	7.56	320.53	0.0149	8.00
0.74	8.01	7.56	320.85	0.0151	9.00
0.74	8.02	7.56	320.97	0.0151	10.00

由此可知，当考虑缆绳最大承载力和加速度时，令  $a = 0.75m/s^2, (t_1, t_2, t_3, t_4) = (8.00s, 2.01s, 7.55s, 50s)$  可在吊运效率和最大吊运角度间进行较好权衡，此时 BC 段最大摆角和吊运时间分别为  $\theta_{max} = 0.86^\circ, t = 61.69s$  最大起吊货物重量  $m_{max} = 19784kg$ 。该解的  $\theta(t), x(t)$  图像如下所示。

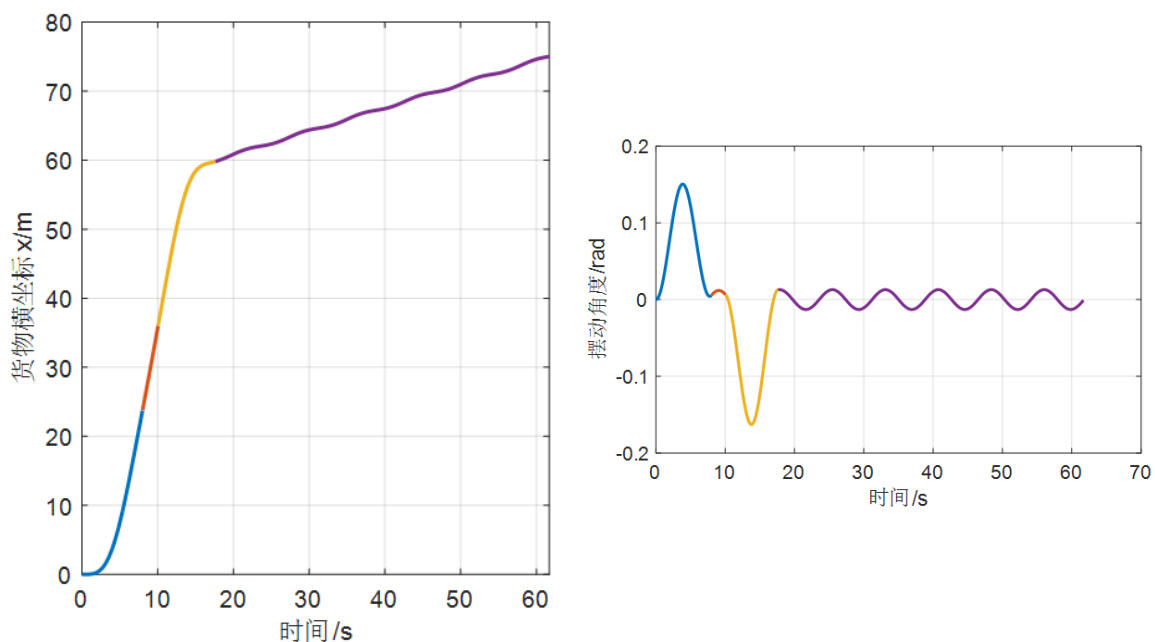
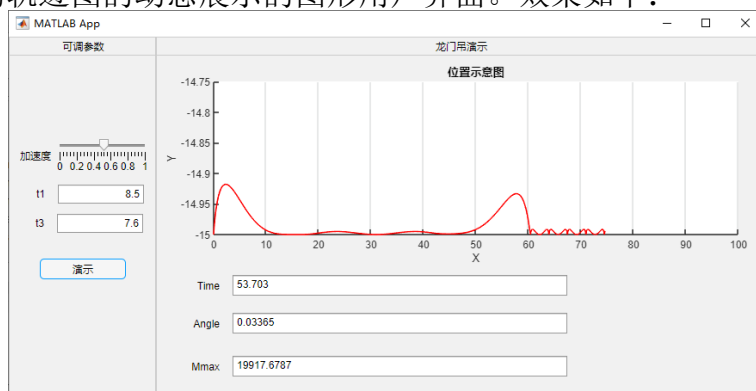


图 3 问题三解的  $\theta(t)$   $x(t)$  图像

#### 6.4 问题四的求解

我们可以使用 matlab 的图形设计工具进行设计,得到了可以输入  $(a, t_1, t_3)$  参数便可求得货物运动的轨迹图的动态展示的图形用户界面。效果如下:



### 七、模型总结

本文通过建立龙门吊的动力学模型,使用优化算法对吊运过程的摆角和效率进行了优化,得到了比较好的解。本文模型的优点在于使用了分析力学方法,直接以摆角为广义坐标得到了一个自由度的运动微分方程,但由于求解时使用的是数值法求解微分方程,且后一段依赖前一段的结果,导致精准程度不是很理想。对于优化问题,本文对不同规模的问题使用了不同的优化方法,得到的结果都通过作图的方式得到了初步验证。最后,本文通过设计 GUI 界面使整个吊运过程可视化,便于检验求解结果。

## 附录 A func.m

```
function [Time, Angle, Mmax]=func(a,t1,t3)
inf=10000000;
D1=60; D2=15; l=15; m=6000;
g=9.8; Tmax=20000*g;
%a=0.5; t1=5; t2=20; t3=4;%此时t为每段的时间间隔
xa1=a*t1*t1/2; xa3=D1; xa4=D1+D2; xa2=xa3-0.5*a*t3*(2*t1-t3);
t2=(xa2-xa1)/(a*t1); t4=(xa4-xa3)/(a*(t1-t3)); %xai为第i段结束后吊车的位移
t2=t2+t1; t3=t2+t3; t4=t3+t4;%此时t为每段结束的时刻
t=cell(4,1); theta=cell(4,1); x=cell(4,1); y=cell(4,1); %时间, 角度, 横纵坐标
va=cell(4,1); va{2}=a*t1; va{4}=a*(t1+t2-t3);%吊车速度
v=cell(4,1); Tx=cell(4,1); Ty=cell(4,1); T=cell(4,1); %水平速度, 拉力,
ax=cell(4,1); ay=cell(4,1);%水平竖直加速度
thetamax=0; Tm=0; %实际最大的角度, 拉力

if t2<0 || (t3-t2)>=t1 || (t2-t1)<0.1 || t4>120
    Time=inf; Angle=inf; Mmax=inf;
    return;
end

%第一段加速
dydt=@(t,y) [ y(2) ; (-a*y(2)*t*sin(y(1))+a*cos(y(1))-g*sin(y(1)))/l ];
options = odeset('RelTol',1e-5,'AbsTol',[1e-5 1e-5]);
[t{1},theta{1}] = ode45(dydt,[0:0.01:t1],[0 0],options);
x{1}=0.5*a*t{1}.^2-l*sin(theta{1}(:,1));
v{1}=a*t{1}-l*theta{1}(:,2).*cos(theta{1}(:,1));

%第二段匀速
dydt=@(t,y) [ y(2) ; (1/l)*(va{2}*y(2)*sin(y(1))-(va{2}+g)*sin(y(1))) ];
options = odeset('RelTol',1e-5,'AbsTol',[1e-5 1e-5]);
[t{2},theta{2}] = ode45(dydt,[t1:0.01:t2],[theta{1}(length(theta{1}(:,1)),1)
    theta{1}(length(theta{1}(:,2)),2)],options);
x{2}=xa1+a*t1*(t{2}-t1)-l*sin(theta{2}(:,1));
v{2}=a*t1-l*theta{2}(:,2).*cos(theta{2}(:,1));

%第三段减速
dydt=@(t,y) [ y(2) ; (-a*y(2)*(t1+t2-t)*sin(y(1))-a*cos(y(1))-g*sin(y(1)))/l ];
options = odeset('RelTol',1e-5,'AbsTol',[1e-5 1e-5]);
[t{3},theta{3}] = ode45(dydt,[t2:0.01:t3],[theta{2}(length(theta{2}(:,1)),1)
    theta{2}(length(theta{2}(:,2)),2)],options);
x{3}=xa2+a*t1*(t{3}-t2)-0.5*a*(t{3}-t2).^2-l*sin(theta{3}(:,1));
v{3}=a*(t1+t2-t{3})-l*theta{3}(:,2).*cos(theta{3}(:,1));

%第四段匀速
dydt=@(t,y) [ y(2) ; (1/l)*(va{4}*y(2)*sin(y(1))-(va{4}+g)*sin(y(1))) ];
```

```

options = odeset('RelTol',1e-5,'AbsTol',[1e-5 1e-5]);
[t{4},theta{4}] = ode45(dydt,[t3:0.01:t4],[theta{3}(length(theta{3}(:,1)),1)
    theta{3}(length(theta{3}(:,2)),2)],options);
x{4}=xa3+a*(t1+t2-t3)*(t{4}-t3)-l*sin(theta{4}(:,1));
v{4}=a*(t1+t2-t3)-l*theta{4}(:,2).*cos(theta{4}(:,1));
thetamax=max(theta{4}(:,1));

for i=1:4
    y{i}=l*cos(theta{i}(:,1));
    ax{i}=gradient(v{i},0.05);
    Tx{i}=m*ax{i};
    ay{i}=-1*(gradient(theta{i}(:,2),0.05).*sin(theta{i}(:,1)) +
        cos(theta{i}(:,1)).*theta{i}(:,2).^2);
    Ty{i}=m*(g-l*ay{i});
    T{i}=(Tx{i}.^2 + Ty{i}.^2).^0.5;
    Tm=max(Tm,max(T{i}));
    p=plot(x{i},-y{i},'linewidth',1.5);
    hold on;
end

E4 = 0.5*m*(v{4}(length(v{4}))^2 +
    (l*theta{4}(length(theta{4}(:,2)),2)*sin(theta{4}(length(theta{4}(:,1)),1)))^2);
vcx=(2*E4/m + 2*g*l*(1-cos(theta{4}(length(theta{4}(:,1)),1))))^0.5;
thetamax=max(thetamax,acos(1-E4/(m*g*l)));

vcx=max(vcx,max(v{4}));
if Tm>Tmax || vcx>0.5 %若拉力超限，最终速度超限
    Time=inf; Angle=inf;
    return;
end

Time=t4; Angle=thetamax; Mmax=m*Tmax/Tm;

```

## 附录 B func1.m

```

function Angle=func1(input)
inf=10000000;
D1=60; D2=15; l=15; m=6000;
g=9.8; Tmax=20000*g;
a=0.5; t1=input(1); t3=input(2);%此时t为每段的时间间隔
xa1=a*t1*t1/2; xa3=D1; xa4=D1+D2; xa2=xa3-0.5*a*t3*(2*t1-t3);
t2=(xa2-xa1)/(a*t1); t4=(xa4-xa3)/(a*(t1-t3)); %xai为第i段结束后吊车的位移
t2=t2+t1; t3=t2+t3; t4=t3+t4;%此时t为每段结束的时刻
t=cell(4,1); theta=cell(4,1); x=cell(4,1); y=cell(4,1); %时间，角度，横纵坐标
va=cell(4,1); va{2}=a*t1; va{4}=a*(t1+t2-t3);%吊车速度

```

```

v=cell(4,1); Tx=cell(4,1); Ty=cell(4,1); T=cell(4,1); %水平速度, 拉力,
ax=cell(4,1); ay=cell(4,1); %水平竖直加速度
thetamax=0; Tm=0; %实际最大的角度, 拉力

if t2<0 || (t3-t2)>=t1 || (t2-t1)<0.1 || t4>120
    Time=inf; Angle=inf; Tm=inf;
    return;
end

%第一段加速
dydt=@(t,y)[ y(2) ; (-a*y(2)*t*sin(y(1))+a*cos(y(1))-g*sin(y(1)))/l ];
options = odeset('RelTol',1e-5,'AbsTol',[1e-5 1e-5]);
[t{1},theta{1}] = ode45(dydt,[0:0.05:t1],[0 0],options);
x{1}=0.5*a*t{1}.^2-l*sin(theta{1}(:,1));
v{1}=a*t{1}-l*theta{1}(:,2).*cos(theta{1}(:,1));

%第二段匀速
dydt=@(t,y)[ y(2) ; (1/l)*(va{2}*y(2)*sin(y(1))-(va{2}+g)*sin(y(1))) ];
options = odeset('RelTol',1e-5,'AbsTol',[1e-5 1e-5]);
[t{2},theta{2}] = ode45(dydt,[t1:0.05:t2],[theta{1}(length(theta{1}(:,1)),1)
    theta{1}(length(theta{1}(:,2)),2)],options);
x{2}=xa1+a*t1*(t{2}-t1)-l*sin(theta{2}(:,1));
v{2}=a*t1-l*theta{2}(:,2).*cos(theta{2}(:,1));

%第三段减速
dydt=@(t,y)[ y(2) ; (-a*y(2)*(t1+t2-t)*sin(y(1))-a*cos(y(1))-g*sin(y(1)))/l ];
options = odeset('RelTol',1e-5,'AbsTol',[1e-5 1e-5]);
[t{3},theta{3}] = ode45(dydt,[t2:0.05:t3],[theta{2}(length(theta{2}(:,1)),1)
    theta{2}(length(theta{2}(:,2)),2)],options);
x{3}=xa2+a*t1*(t{3}-t2)-0.5*a*(t{3}-t2).^2-l*sin(theta{3}(:,1));
v{3}=a*(t1+t2-t{3})-l*theta{3}(:,2).*cos(theta{3}(:,1));

%第四段匀速
dydt=@(t,y)[ y(2) ; (1/l)*(va{4}*y(2)*sin(y(1))-(va{4}+g)*sin(y(1))) ];
options = odeset('RelTol',1e-5,'AbsTol',[1e-5 1e-5]);
[t{4},theta{4}] = ode45(dydt,[t3:0.05:t4],[theta{3}(length(theta{3}(:,1)),1)
    theta{3}(length(theta{3}(:,2)),2)],options);
x{4}=xa3+a*(t1+t2-t3)*(t{4}-t3)-l*sin(theta{4}(:,1));
v{4}=a*(t1+t2-t3)-l*theta{4}(:,2).*cos(theta{4}(:,1));
thetamax=max(theta{4}(:,1));

for i=1:4
    y{i}=l*cos(theta{i}(:,1));
    ax{i}=gradient(v{i},0.05);
    Tx{i}=m*ax{i};
    ay{i}=-1*(gradient(theta{i}(:,2),0.05).*sin(theta{i}(:,1)) +
        cos(theta{i}(:,1)).*theta{i}(:,2).^2);

```

```

    Ty{i}=m*(g-l*ay{i});
    T{i}=(Tx{i}.^2 + Ty{i}.^2).^0.5;
    Tm=max(Tm,max(T{i}));
    %plot(t{i},theta{i}(:,1));
    %hold on;
end

E4 = 0.5*m*(v{4}(length(v{4}))^2 +
    (1*theta{4}(length(theta{4}(:,2)),2)*sin(theta{4}(length(theta{4}(:,1)),1)))^2);
vcx=(2*E4/m + 2*g*l*(1-cos(theta{4}(length(theta{4}(:,1)),1))))^0.5;
thetamax=max(thetamax,acos(1-E4/(m*g*l)));

vcx=max(vcx,max(v{4}));
if Tm>Tmax || vcx>0.5 %若拉力超限，最终速度超限
    Time=inf; Angle=inf;
    return;
end

Time=t4; Angle=thetamax;

```

## 附录 C 演示代码

```

classdef app1_exported < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure          matlab.ui.Figure
        GridLayout         matlab.ui.container.GridLayout
        LeftPanel          matlab.ui.container.Panel
        t1EditFieldLabel   matlab.ui.control.Label
        t1EditField        matlab.ui.control.NumericEditField
        t3EditFieldLabel   matlab.ui.control.Label
        t3EditField        matlab.ui.control.NumericEditField
        Button             matlab.ui.control.Button
        Label_2            matlab.ui.control.Label
        Slider             matlab.ui.control.Slider
        RightPanel         matlab.ui.container.Panel
        UIAxes             matlab.ui.control.UIAxes
        TimeTextAreaLabel  matlab.ui.control.Label
        Time               matlab.ui.control.TextArea
        AngleTextAreaLabel matlab.ui.control.Label
        Angle              matlab.ui.control.TextArea
        MmaxTextAreaLabel  matlab.ui.control.Label
        Mmax               matlab.ui.control.TextArea
    end

```



```

% Properties that correspond to apps with auto-reflow
properties (Access = private)
    onePanelWidth = 576;
end

properties (Access = public)
    a1 % Description
    t_1
    t_3
    h
end

% Callbacks that handle component events
methods (Access = private)

    % Code that executes after component creation
    function startupFcn(app, a1, t_1, t_3)
        app.h = animatedline(app.UIAxes);
    end

    % Changes arrangement of the app based on UIFigure width
    function updateAppLayout(app, event)
        currentFigureWidth = app.UIFigure.Position(3);
        if(currentFigureWidth <= app.onePanelWidth)
            % Change to a 2x1 grid
            app.GridLayout.RowHeight = {480, 480};
            app.GridLayout.ColumnWidth = {'1x'};
            app.RightPanel.Layout.Row = 2;
            app.RightPanel.Layout.Column = 1;
        else
            % Change to a 1x2 grid
            app.GridLayout.RowHeight = {'1x'};
            app.GridLayout.ColumnWidth = {171, '1x'};
            app.RightPanel.Layout.Row = 1;
            app.RightPanel.Layout.Column = 2;
        end
    end

% Callback function: Angle, Button, Mmax
function ButtonPushed(app, event)
    %clf(app.UIAxes);
    app.h.LineWidth=0.7;
    app.h.Color='r';
    [app.Time.Value,app.Angle.Value,app.Mmax.Value]=funcapp(app.a1,app.t_1,app.t_3,app.h);

```

```

end

% Callback function
function EditFieldValueChanged(app, event)
    value = app.EditField.Value;
    app.a1=value;
end

% Value changed function: t1EditField
function t1EditFieldValueChanged(app, event)
    value = app.t1EditField.Value;
    app.t_1=value;
end

% Value changed function: t3EditField
function t3EditFieldValueChanged(app, event)
    value = app.t3EditField.Value;
    app.t_3=value;
end

% Value changed function: Slider
function SliderValueChanged(app, event)
    value = app.Slider.Value;
    app.a1=value;
end

% Callback function
function UITableDisplayDataChanged(app, event)

end

% Callback function
function UITableCellEdit(app, event)

end

% Value changed function: Time
function TimeValueChanged(app, event)

end

end

% Component initialization
methods (Access = private)

```

```

% Create UIFigure and components
function createComponents(app)

    % Create UIFigure and hide until all components are created
    app.UIFigure = uifigure('Visible', 'off');
    app.UIFigure.AutoResizeChildren = 'off';
    app.UIFigure.Position = [100 100 640 480];
    app.UIFigure.Name = 'MATLAB App';
    app.UIFigure.SizeChangedFcn = createCallbackFcn(app, @updateAppLayout, true);

    % Create GridLayout
    app.GridLayout = uigridlayout(app.UIFigure);
    app.GridLayout.ColumnWidth = {171, '1x'};
    app.GridLayout.RowHeight = {'1x'};
    app.GridLayout.ColumnSpacing = 0;
    app.GridLayout.RowSpacing = 0;
    app.GridLayout.Padding = [0 0 0 0];
    app.GridLayout.Scrollable = 'on';

    % Create LeftPanel
    app.LeftPanel = uipanel(app.GridLayout);
    app.LeftPanel.TitlePosition = 'centertop';
    app.LeftPanel.Title = '可调参数';
    app.LeftPanel.Layout.Row = 1;
    app.LeftPanel.Layout.Column = 1;

    % Create t1EditFieldLabel
    app.t1EditFieldLabel = uilabel(app.LeftPanel);
    app.t1EditFieldLabel.HorizontalAlignment = 'right';
    app.t1EditFieldLabel.Position = [17 263 25 22];
    app.t1EditFieldLabel.Text = 't1';

    % Create t1EditField
    app.t1EditField = uieditfield(app.LeftPanel, 'numeric');
    app.t1EditField.ValueChangedFcn = createCallbackFcn(app,
        @t1EditFieldValueChanged, true);
    app.t1EditField.Position = [57 263 100 22];

    % Create t3EditFieldLabel
    app.t3EditFieldLabel = uilabel(app.LeftPanel);
    app.t3EditFieldLabel.HorizontalAlignment = 'right';
    app.t3EditFieldLabel.Position = [16 226 25 22];
    app.t3EditFieldLabel.Text = 't3';

    % Create t3EditField
    app.t3EditField = uieditfield(app.LeftPanel, 'numeric');
    app.t3EditField.ValueChangedFcn = createCallbackFcn(app,

```

```

        @t3EditFieldValueChanged, true);
app.t3EditField.Position = [56 226 100 22];

% Create Button
app.Button = uibutton(app.LeftPanel, 'push');
app.Button.ButtonPushedFcn = createCallbackFcn(app, @ButtonPushed, true);
app.Button.Position = [36 166 100 24];
app.Button.Text = '演示';

% Create Label_2
app.Label_2 = uilabel(app.LeftPanel);
app.Label_2.HorizontalAlignment = 'right';
app.Label_2.Position = [8 311 41 22];
app.Label_2.Text = '加速度';

% Create Slider
app.Slider = uislider(app.LeftPanel);
app.Slider.Limits = [0 1];
app.Slider.ValueChangedFcn = createCallbackFcn(app, @SliderValueChanged, true);
app.Slider.Position = [59 332 99 3];

% Create RightPanel
app.RightPanel = uipanel(app.GridLayout);
app.RightPanel.TitlePosition = 'centertop';
app.RightPanel.Title = '龙门吊演示';
app.RightPanel.BackgroundColor = [0.9412 0.9412 0.9412];
app.RightPanel.Layout.Row = 1;
app.RightPanel.Layout.Column = 2;

% Create UIAxes
app.UIAxes = uiaxes(app.RightPanel);
title(app.UIAxes, '位置示意图')
xlabel(app.UIAxes, 'X')
ylabel(app.UIAxes, 'Y')
app.UIAxes.XLim = [0 100];
app.UIAxes.YLim = [-15 -14.75];
app.UIAxes.XGrid = 'on';
app.UIAxes.TitleFontWeight = 'bold';
app.UIAxes.Position = [10 197 438 250];

% Create TimeTextAreaLabel
app.TimeTextAreaLabel = uilabel(app.RightPanel);
app.TimeTextAreaLabel.HorizontalAlignment = 'right';
app.TimeTextAreaLabel.Position = [42 155 32 22];
app.TimeTextAreaLabel.Text = 'Time';

% Create Time

```

```

app.Time = uitextarea(app.RightPanel);
app.Time.ValueChangedFcn = createCallbackFcn(app, @TimeValueChanged, true);
app.Time.Tag = 'Time';
app.Time.Position = [89 155 150 24];

% Create AngleTextAreaLabel
app.AngleTextAreaLabel = uilabel(app.RightPanel);
app.AngleTextAreaLabel.HorizontalAlignment = 'right';
app.AngleTextAreaLabel.Position = [38 99 36 22];
app.AngleTextAreaLabel.Text = 'Angle';

% Create Angle
app.Angle = uitextarea(app.RightPanel);
app.Angle.ValueChangedFcn = createCallbackFcn(app, @ButtonPushed, true);
app.Angle.Tag = 'Angle';
app.Angle.Position = [89 99 150 24];

% Create MmaxTextAreaLabel
app.MmaxTextAreaLabel = uilabel(app.RightPanel);
app.MmaxTextAreaLabel.HorizontalAlignment = 'right';
app.MmaxTextAreaLabel.Position = [36 36 38 22];
app.MmaxTextAreaLabel.Text = 'Mmax';

% Create Mmax
app.Mmax = uitextarea(app.RightPanel);
app.Mmax.ValueChangedFcn = createCallbackFcn(app, @ButtonPushed, true);
app.Mmax.Tag = 'Mmax';
app.Mmax.Position = [89 36 150 24];

% Show the figure after all components are created
app.UIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

% Construct app
function app = app1_exported(varargin)

% Create UIFigure and components
createComponents(app)

% Register the app with App Designer
registerApp(app, app.UIFigure)

% Execute the startup function

```

```

runStartupFcn(app, @(app)startupFcn(app, varargin{:}))

if nargin == 0
    clear app
end
end

% Code that executes before app deletion
function delete(app)

    % Delete UIFigure when app is deleted
    delete(app.UIFigure)
end
end
end

```

## 附录 D 第二问代码

```

thetamin=min(Ans(:,3)); thetamax=max(Ans(:,3));
tmin=min(Ans(:,4)); tmax=max(Ans(:,4));
a=zeros(30,5);
for i=1:133
    a(i,1)=(Ans(i,3)-thetamin)/(thetamax-thetamin);
    a(i,2)=(Ans(i,4)-tmin)/(tmax-tmin);
end
abest=[min(a(:,1)), min(a(:,2))];
aworst=[max(a(:,1)), max(a(:,2))];
w1=0.6; w2=0.4;
for i=1:133
    a(i,3)=(w1*(a(i,1)-abest(1))^2 + w2*(a(i,2)-abest(2))^2)^0.5;
    a(i,4)=(w1*(a(i,1)-aworst(1))^2 + w2*(a(i,2)-aworst(2))^2)^0.5;
    a(i,5)=a(i,3)/(a(i,3)+a(i,4));
end

for i=1:133
    Ans(i,3)=Ans(i,3)*180/pi;
end

```