

---

# 承诺书

我们授权全国大学生数学建模竞赛组委会,可将我们的论文以任何形式进行公开展示(包括进行网上公示,在书籍、期刊和其他媒体进行正式或非正式发表等)。

(指导教师签名意味着对参赛队的行为和论文的真实性负责)

日期: 2019 年 09 月 15 日

(请勿改动此页内容和格式。此承诺书打印签名后作为纸质论文的封面，注意电子版论文中不得出现此页。以上内容请仔细核对，如填写错误，论文可能被取消评奖资格。)

赛区评阅编号（由赛区组委会填写）：

---

**2019 高教社杯全国大学生数学建模竞赛**

**编 号 专 用 页**

赛区评阅记录（可供赛区评阅时使用）：

评 阅 人						
备 注						

送全国评阅统一编号（由赛区组委会填写）：

全国评阅随机编号（由全国组委会填写）：

（请勿改动此页内容和格式。此编号专用页仅供赛区和全国评阅使用，参赛队打印后装订到纸质论文的第二页上。注意电子版论文中不得出现此页。）

# 燃油系统的压力控制及方案设计

## 摘要

如何通过对高压油泵与喷油嘴等开关的控制,有效维持高压油管内的压力从而保证喷油量的稳定,提高发动机的工作效率是燃油系统设计的关键问题.本文对燃油在整个高压燃油系统中的传递过程进行模拟,确定不同条件下燃油的质量分布,从而获得使高压燃油系统保持稳定的最优控制方案.

**关于问题一:** 问题一通过忽略每次泵油后,高压油管内压力改变导致的泵油速率改变,在一个恒定的压强差下获得了单向阀的最佳单次开启时长  $t = 0.284 \text{ ms}$ ,使得一个周期内泵入的油量与喷出的油量相等.第二问在稳压的基础上新增了一个增压的变速泵油过程.对于增压过程,需要考虑不同压力下泵油速率的改变,采用计算机迭代的方法来模拟这个过程.本题保持每次泵入的燃油质量不变,控制泵油时间,每次泵油时间均不相同,其变化规律如图11所示.

**关于问题二:** 问题二更精细地考虑了高压油泵与喷油嘴的结构.通过解耦合的方法,在保证一段时间内泵油量与喷油量相等的基础上,分开计算泵油与喷油.当角速度取  $\omega = 0.0275 \text{ rad/ms}$  时,可以使高压油管内质量的相对变化接近 0.

**关于问题三:** 问题三通过两个喷油器与减压阀共同作用,减小油管内气压波动.喷油器在一个周期内平均地交替工作,视作一个周期减半的等效喷油器.新增的减压阀则在油管内压力大于一定阈值时打开.通过比较前后的方差,可以得知减压阀的最佳压力阈值在 105 MPa.

本文通过忽略燃油压强改变的弛豫过程等次要条件,将系统各部分部件解耦合得到了一个简化模型,该模型具有均能较好地满足题目要求,且具有较高的精度与稳定性.

**关键字:** 燃油系统 压力控制 解耦合 计算机模拟

## 目录

一、问题重述 .....	4
1.1 背景介绍 .....	4
1.2 具体问题 .....	4
二、问题的分析 .....	4
2.1 问题的总分析 .....	4
2.2 问题一的分析 .....	4
2.3 问题二的分析 .....	5
2.4 问题三的分析 .....	5
三、模型的假设 .....	5
四、名词解释与符号说明 .....	6
4.1 名词解释 .....	6
4.2 符号说明 .....	6
五、模型的建立与求解 .....	6
5.1 数据预处理 .....	6
5.2 问题一分析 .....	7
5.3 问题二分析 .....	11
5.4 问题三分析 .....	14
六、模型评价与改进 .....	15
6.1 模型的优点 .....	15
6.2 模型的缺点 .....	16
参考文献 .....	16
附录 A 题目一第一小问 – Mathematica 源代码 .....	17
附录 B 第一题第二小问 – Python 源代码 .....	17
附录 C 题目二喷油过程 – Python 源代码 .....	19
附录 D 题目二泵入油过程 – Python 源代码 .....	21

附录 E	题目三 – Python 源代码 .....	23
附录 F	数据处理 – MATLAB 源代码 .....	27

# 一、问题重述

## 1.1 背景介绍

提高功率、节约能源与改善排放已经成为当今燃油机发展的必然趋势. 为了与高性能燃油机相配合, 必然要求高压燃油系统也具有极高的性能.

高压油泵、高压油管与喷射嘴是高压燃油系统中最为重要的组成部分. 燃油进入与喷出的间歇性工作过程会导致高压油管内压力的变化, 并使得喷油量出现偏差, 这些偏差会影响发动机的工作效率. 通过对高压油泵与喷油嘴等开关的控制, 可以有效地维持高压油管内的压力从而保证喷油量的稳定, 提高发动机的工作效率.

## 1.2 具体问题

(1) 已知高压油泵压力及其单向阀休息时长、喷油器的工作状态、高压油管尺寸以及压力的变化要求, 确定高压油泵单向阀的单次开启时长, 使得油管内的压力波动尽可能较小.

(2) 结合实际情况分析, 将问题一中的恒压高压轴泵换为由凸轮驱动的柱塞, 喷油嘴则由针阀周期性控制, 其他条件均与第一题相同. 在此基础上, 确定凸轮的角速度, 使得高压油管内的压力尽量稳定在 100 MPa.

(3) 在问题二的基础上, 增加了一个喷油规律相同的喷油嘴与单向减压阀, 使得管内的燃油可以回流到外部的低压油路中, 进一步减小压力, 从而给出更优化的高压油泵与减压阀控制方案.

# 二、问题的分析

## 2.1 问题的总分析

本题通过泵入, 喷出和回流来改变高压油管内的压强, 要求通过改变泵入和回流的油量来保持压强稳定. 本文的基本思路为通过计算机模拟, 获得在不同条件下使得高压燃油系统保持稳定的最优控制方案.

## 2.2 问题一的分析

第一问是在压力差恒定的匀速泵油条件下, 获得高压油管内压强尽可能稳定在 100 MPa 的单向阀最佳单次开启时长, 本小题 1 s 内喷油量是已知的. 由于油管内压强始终在 100 MPa 上下浮动, 可以忽略每次泵油后由于高压油管内压力改变导致的泵油速率改变. 问题变为寻找合适的高压油泵单向阀开启时长, 使得一个周期内泵入的油量与喷出的油量相等.

而第二问是在压力差减小的变速泵油条件下, 寻找能将高压油管内压强增加至 150 MPa 并在规定时长内达到稳定的单向阀开启时长方案. 该方案分为增压和稳压两个不同的部分, 其中的稳压部分同第一问类似. 对于增压过程, 需要考虑不同压力下泵油速率的改变, 我们选择采用计算机迭代的方法模拟这个过程.

### 2.3 问题二的分析

问题二更精细地考虑了高压油泵与喷油嘴的结构. 与问题一第一问一样, 由于油管内压力始终在 100 MPa 上下浮动, 可以忽略每次泵油后由于高压油管内压力改变导致的泵油速率改变. 基于此, 就可以分开计算泵油与喷油两个过程, 并保证一段时间内泵油量等于喷油量即可.

对于泵入油的过程, 设初始时是极径最短的状态. 对于泵出油的过程, 将其分为两个过程: 油从高压油管进入喷油器喷嘴的过程和油从喷油器喷嘴进入环境的过程.

### 2.4 问题三的分析

问题三增加了一个喷油器和减压阀. 新旧两个喷油器将交替喷油, 以减小油管内气压波动. 为了燃油系统的出油过程相对稳定, 需要将两个喷油器平均分配, 因此可以将其视作一个周期减半为 50 ms 的等效喷油器.

考虑高压油管内的压力变化, 使新增的减压阀在油管内压力大于初始压力 100 MPa 时打开. 计算发现这样可以有效降低油管内油量波动.

## 三、模型的假设

- 忽略燃油压强改变的弛豫过程

由于压力波在液体中的传递速度可以达到音速 [1], 而在燃油中音速的范围大约在 1200 mm/ms~1600 mm/ms, 因此每次泵入的燃油在 1 ms 的时间内就能弥散在整个油管内. 基于此, 流体的扩散过程可以在一定程度上被忽略, 从而得到一个各部分燃料都均匀分布的简化模型.

- 视高压油管内的燃油为单相液质 [1]
- 燃油在系统的各部分内分别均匀分布, 需要将喷油器喷嘴内部小圆台视作一个单独的部分
- 当要求高压油管压力稳定 100 MPa 时, 忽略单次进油与喷油过程中导致的压力微小波动
- 取喷嘴外部环境气压为标准大气压

## 四、名词解释与符号说明

### 4.1 名词解释

- Aa 处体积: 高压油泵柱塞体积
- A 处面积, 压强: 供油入口面积, 与高压油泵内的压强
- B 处面积: 针阀下端与密封座形成的圆环的面积
- B 处体积, 压强: 针阀下端与喷孔上端之间的圆台的体积, 压强
- C 处压强: 外界环境的压强, 即标准大气压

### 4.2 符号说明

符号	意义
$E$	弹性模量 (MPa)
$m_B$	从 B 处每次泵入油的质量 (mg)
$m_r$	高压油管相较于初始时的相对质量 (mg)
$\delta_{m_r}^2$	相对质量的方差
$\theta$	极角 (rad)
$R$	极径 (mm)
$d_1$	针阀抬升的高度 (mm)
$d_2$	针阀升程为 0 时与最下端小孔的距离 (mm)
$d_3$	针阀最下端小孔到圆心距离 (mm)
$\omega$	凸轮转动角速度 (rad/ms)

## 五、模型的建立与求解

### 5.1 数据预处理

附件 1 将凸轮在旋转过程中的图像从竖直方向上抽离出, 我们联想到了三角函数, 因此将极径  $R$  与  $\theta$  用三角函数进行拟合, 得到了:

$$R = 4.826 + 2.413 \cos \theta, \quad R^2 = 1 \quad (1)$$

液体的压强与密度的关系无简单的解析式, 弹性模量一般是通过试验来确定的, 因



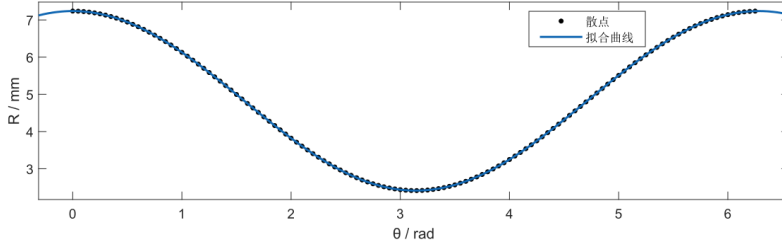


图 1 极径与极角关系图

此直接选择用相关性最高的指数函数进行拟合得到式 1, 拟合结果如图2.

$$E(P) = 1495e^{0.0039P}, \quad R^2 = 0.9955 \quad (2)$$

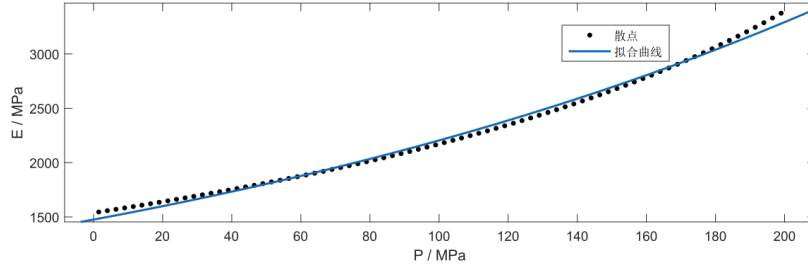


图 2 弹性模量与压强关系图

我们观测到针阀运动曲线的两端为 S 型曲线, S 型曲线可以使用 logistic 回归进行拟合, 使用此公式对针阀上升与下降过程进行分别的拟合, 得到一个分段的函数

$$d_1 = \begin{cases} \frac{2.225}{1 + 537.9e^{-19.3t}}, & t \in [0, 0.45), & R^2 = 0.9996 \\ 2, & t \in [0.45, 2), & R^2 = 1 \\ \frac{2.267}{1 + 456.4e^{18.63(t-2.45)}}, & t \in [2, 2.45), & R^2 = 0.9997 \\ 0, & t \in [2.45, 100), & R^2 = 1 \end{cases} \quad (3)$$

## 5.2 问题一分析

问题一共分两个小问.

对于第一个小问, 高压油管内压力始终在 100 MPa 上下浮动. 对问题初步分析可以发现, 单次泵入, 喷出的燃油质量比油管内燃油质量低 3 个数量级. 因此忽略每次泵入、喷出燃油后油管内部的压力变化是合理的. 进行这样的假设后, 泵入流量  $Q = CA\sqrt{\frac{2\Delta P}{\rho}} = 15.547 \text{ mm}^3/\text{ms}$  将是一个恒定值. 又由于每次喷出的燃油体积  $V_{out} = 440 \text{ mm}^3$  已知. 于是每秒需要泵入燃油的时间在修正密度变化后为  $t_s = \frac{440 \text{ mm}^3}{Q} \times \frac{0.85 \text{ mg/mm}^3}{0.87 \text{ mg/mm}^3} = 27.65 \text{ ms}$ .

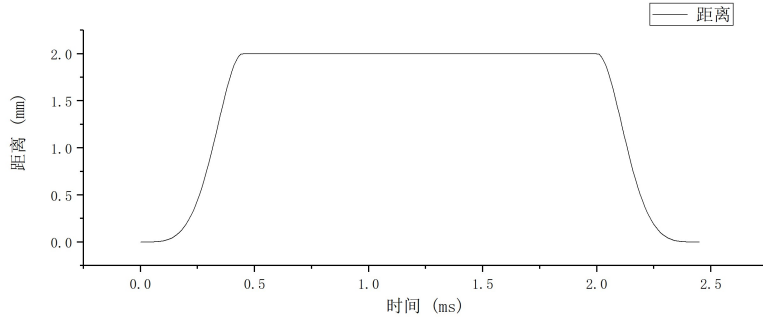


图3 针阀运动曲线

为了让压力尽量稳定,需要多次少量泵入燃油,即使每次泵入燃油的间隔时间均为 10 ms. 考虑这个条件之后,设每次泵油时间为  $t$ ,便可写出 1000 ms 内喷油次数

$$N = \frac{1000}{t + 10} = \frac{t_s}{t}$$

根据该方程即可得到每 10 ms 泵入一次油,每次泵入燃油的时间:

$$t = 0.284 \text{ ms}$$

对于第二个小问,可以分解为油管内压力在规定时间内逐渐从 100 MPa 增加到 150 MPa 的过程一和使油管内压力保持在 150 MPa 的过程二.

对于过程一,随着油管内压力逐渐增高,泵入油的速率会逐渐降低,但喷出油的速率保持不变. 为了保持油管内压力上升速率稳定,应保持每次泵入的油量不变,这会使每次开启泵的时间逐渐增加. 通过调节每次泵入的油的质量  $m_B$  的大小可以改变油管内压强的增加速度,进而改变压强到达 150 MPa 的时间. 实际计算中采用时间迭代的方法,,计算泵入和喷出两个过程中每一个  $dt$  内油量的微小变化. 其计算过程如图4所示. 2 s, 5 s, 10 s 内压强逐渐增加到 150 MPa 的过程如图5, 7, 9所示.

最终将得到不同时间达到 150 MPa 所需的每次泵入的油的质量  $m_B$ :

2 s	$m_B = 7.31 \text{ mg}$
5 s	$m_B = 5.04 \text{ mg}$
10 s	$m_B = 4.31 \text{ mg}$

不同稳定时间下每次阀门开启时间如图6, 8, 10所示:

对于过程二,此时问题简化为稳恒压力下喷出油策略,将第一小问内部压强从 100 MPa 改变为 150 MPa,再将此时的密度替换为 150 MPa 下的密度即可.

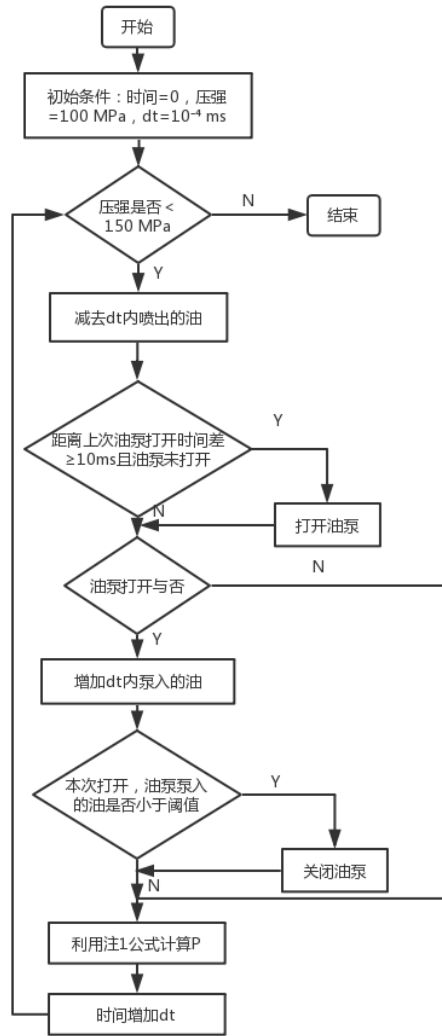


图 4 问题一第二小问过程一流程图: 设定初值后, 开始按照  $dt = 10^{-4} \text{ ms}$  逐渐增加时间  $t$  进行循环, 循环进行条件为压强小于  $150 \text{ MPa}$ , 在循环内, 先计算  $dt$  内喷出的油, 再判断此时是否需要打开阀门, 如果阀门已经打开, 则计算  $dt$  内泵入的油, 并判断此时是否需要关闭阀门. 结束后, 输出停止循环, 即得到  $150 \text{ MPa}$  需要的时间.

计算密度可利用注 1 中的公式

$$dP = \frac{E(P)}{\rho} d\rho$$

$$\frac{dP}{E(P)} = d(\ln \rho)$$

代入数据处理中得到的式2:

$$E(P) = 1495e^{0.0039P}$$

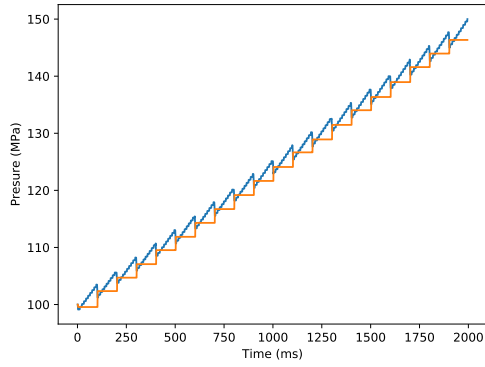


图 5 2s 稳定

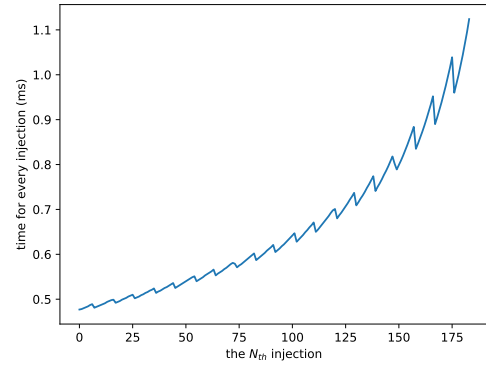


图 6 2s 稳定

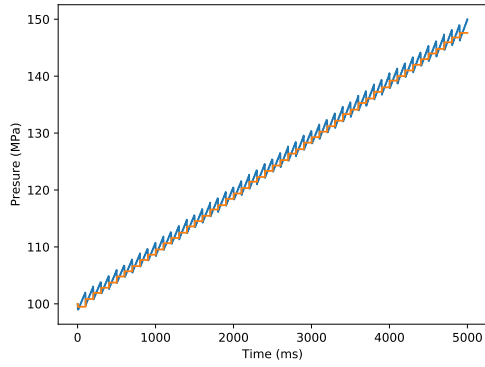


图 7 5s 稳定

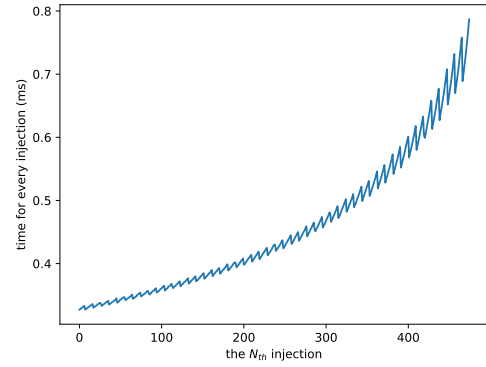


图 8 5s 稳定

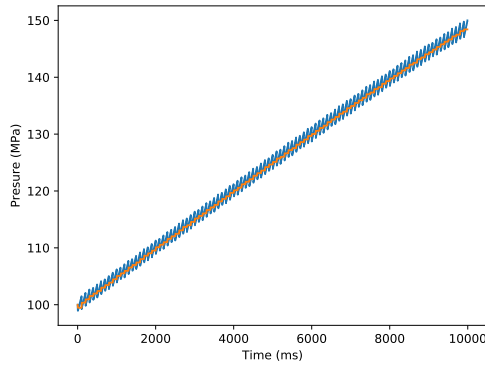


图 9 10s 稳定

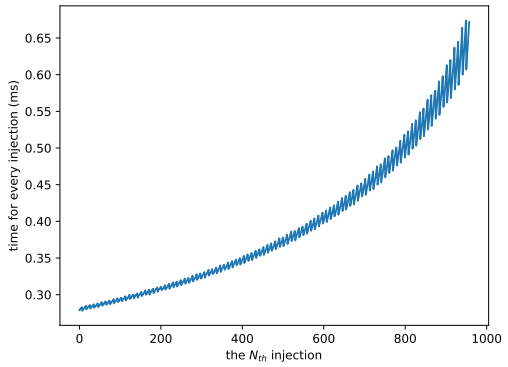


图 10 10s 稳定

图 11 左侧三图: 压强逐渐增加到 **150 MPa** 的过程. 蓝线: 油管内压强, 橙线: 去除小范围波动后压强上升情况.

右侧三图: 第  $N$  次泵入油的阀门开启时间随  $N$  的变化情况.

积分并代入条件  $\rho = 0.85 \text{ mg} \cdot \text{mm}^{-3}$ ,  $P = 100 \text{ MPa}$  可以得到

$$\rho = \frac{1}{20.02} e^{-\frac{1}{5.831}} e^{-0.0039P} \quad (4)$$

当  $P = 150 \text{ MPa}$ , 密度  $\rho = 0.868 \text{ mg} \cdot \text{mm}^{-3}$  时, 每  $10 \text{ ms}$  泵入一次油, 利用与第一小问相同的计算方法, 可得在  $150 \text{ MPa}$  时每次泵油的时间为:

$$t = 0.75 \text{ ms}$$

### 5.3 问题二分析

对于问题二, 由于油管内压力始终在  $100 \text{ MPa}$  上下小范围波动, 近似地认为油管内压力不变, 这个近似可以将泵油和喷油两个过程解耦合, 从而可以分开考虑这两个过程.

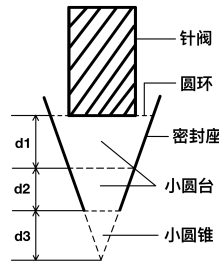


图 12 喷油嘴形状示意图: 将油从油管内进入小圆台内和从小圆台内进入外部环境看作两个出油过程

对于喷油的过程, 喷油量的变化是固定的. 喷油嘴的形状如图12所示. 可以将油从油管内穿过针阀底部和密封座形成的圆环进入圆台内, 和从圆台内穿过喷孔进入环境看做两个出油过程. 计算过程如图13所示. 在实际计算中发现, 几乎每个  $dt$  内喷入圆台的燃油都可以立刻从圆台完全喷出到环境中.

对于前一个过程, 首先利用数据预处理中计算的针阀位移关系 (式3):

$$d_1 = \begin{cases} \frac{2.225}{1 + 537.9e^{-19.3t}}, & t \in [0, 0.45) \\ 2, & t \in [0.45, 2) \\ \frac{2.267}{1 + 456.4e^{18.63(t-2.45)}}, & t \in [2, 2.45) \\ 0, & t \in [2.45, 100) \end{cases}$$

于是可以得到小圆台的体积和上述圆环面积:

$$V_B = \frac{\pi d_1}{3} (r_1^2 + r_3^2 + r_1 r_3), \quad r_1 = (d_1 + d_2 + d_3)\theta, \quad r_3 = (d_3)\theta \quad (5)$$

$$A_B = \pi r_1^2 - \pi r_2^2, \quad r_1 = (d_1 + d_2 + d_3)\theta, \quad r_2 = (d_2 + d_3)\theta \quad (6)$$

进而可以得到  $dt$  内进入圆台的燃油质量  $dm_B$ :

$$\begin{aligned} dm_B &= \rho_B Q(A_B, 100 - P_B, \rho_B) dt \\ &= \rho_B C A_B \sqrt{\frac{2(100 - P_B)}{\rho_B}} dt \end{aligned}$$

上述  $C$  为常数,  $A_B, P_B$  与时间  $t$  或密度  $\rho_B$  的表达式已于式6, 4给出. 而  $\rho_B = \frac{m_B}{V_B}$  是时间  $t$  和质量的函数, 于是上述表达式化为  $dm_B$  和  $dt$  的微分方程, 迭代时间  $t$  即可得到  $m_B$  变化情况. 图14展示了计算出的喷出的油的总质量随时间变化, 可以看到燃油以 100 ms 为周期向外泵出的过程.

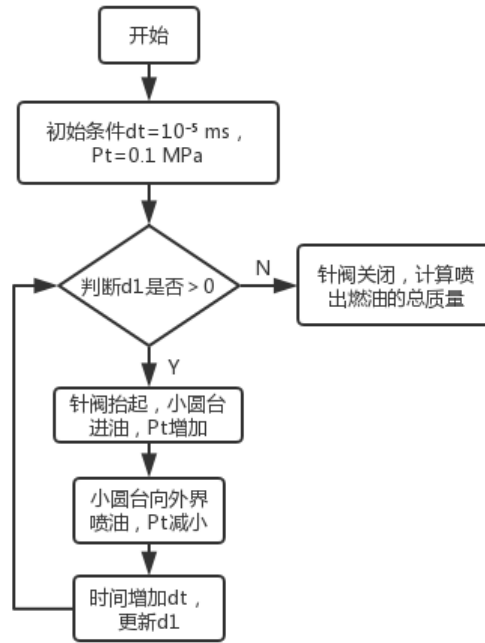


图 13 喷油嘴形状示意图: 将油从油管内进入圆台内和从圆台内进入环境看做两个出油过程

对于泵油过程, 可以通过改变凸轮旋转的角速度来改变泵油的周期和每次泵油的油量. 可以将泵油过程分解成两种情况, 即油泵压力小于 100 MPa 不会向外泵油的情况和压力大于 100 MPa 会向油管泵油的情况.

利用几何关系计算出图12中的  $d_2, d_3$ :

$$\begin{aligned} d_2 + d_3 &= \frac{R_{\text{针阀}}}{\theta} \\ d_3 &= \frac{R_{\text{喷孔}}}{\theta} \end{aligned}$$

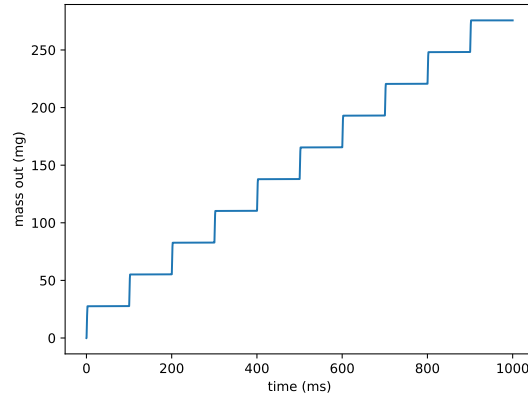


图 14 泵出的油的总质量随时间变化

假设开始时油泵在下止点, 即极径最小的位置, 油泵内将充满压强为 0.5 MPa 的低压燃油, 随着柱塞向上运动压缩燃油, 直到在  $\theta = 0.515 \text{ rad}$  时燃油密度达到 100 MPa 对应的  $0.85 \text{ mg/mm}^3$ . 图15展示了在刚开始时油泵内压力较低 ( $< 100 \text{ MPa}$ ) 时油泵内压强随角度变化. 图16展示了当压力较大 ( $> 100 \text{ MPa}$ ), 且  $\omega = 0.3 \text{ rad/ms}$  时油泵内压强随时间变化.

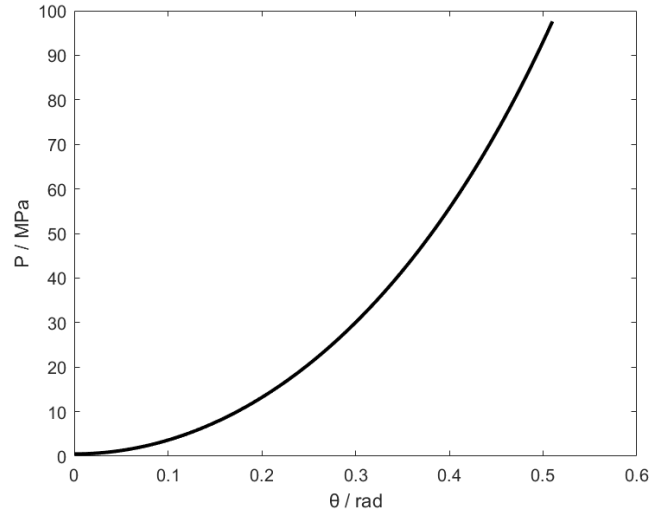


图 15 油泵内压强随时间变化

由于喷油过程喷油量是不可控的, 但泵油过程可以通过改变角速度改变泵入油的速度, 我们的目标是通过改变角速度使进油量 = 出油量. 为了描述进油量与出油量的关系, 我们绘制出高压油管内燃油质量相较于初始质量的关系图17, 并计算其平均值  $\langle m_r \rangle$ .

$$\langle m_r \rangle = \frac{\int_0^{1000 \text{ s}} m_r(t) dt}{1000 \text{ s}}$$

计算发现当角速度取

$$\omega = 0.0275 \text{ rad/ms}$$

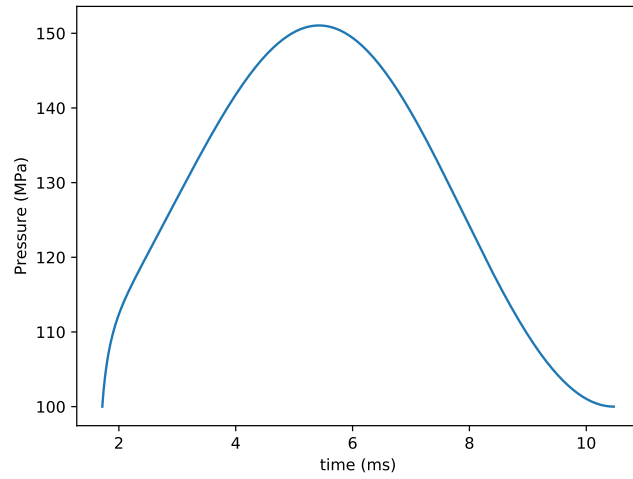


图 16 油泵内压强随时间变化

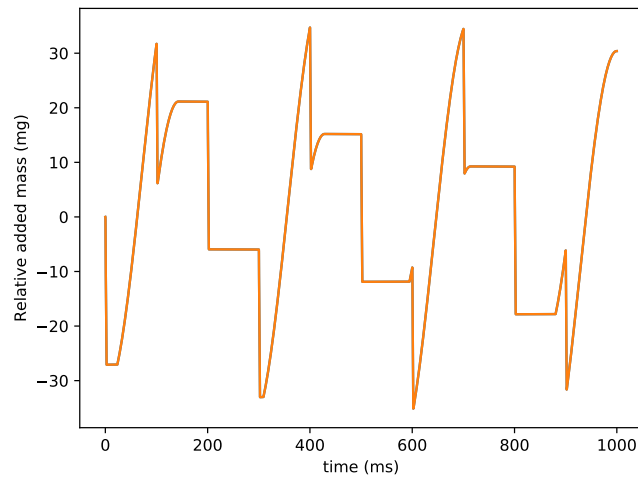


图 17 高压油管内相对燃油质量变化: 相对燃油质量 = 某一时刻油管内燃油质量 - 初始燃油质量

时可以使  $\langle m_r \rangle$  接近 0.

#### 5.4 问题三分析

问题三在问题二的基础上增加了一个喷油嘴和减压阀. 用与上一题同样的方式对各部分解耦合, 分别计算各部分对系统总油量的影响.

增加一个喷油嘴后, 让两个喷油嘴按相同的间隔交替喷油是一个减小波动的自然且直观的想法. 也就是让第二个喷油嘴比第一个喷油嘴滞后 50 ms 喷油.

利用与第二题相同的方法, 写出高压油管内相较于初始 100 MPa 时相对质量  $m_r(t) =$



$m(t) - m_0$  随时间的变化

$$m_r(t) = m_{\text{油泵}}(t) - m_{\text{喷油}}(t) - m_{\text{减压阀}}(t)$$

要使压力稳定在 100 MPa, 可以使一段时间内平均  $m_r(t) = 0$ . 这段时间取 1000 ms, 则需平均相对质量  $\langle m_r(t) \rangle$

$$\begin{aligned}\langle m_r(t) \rangle &= \frac{\int_0^{1000 \text{ s}} m_r(t) dt}{1000 \text{ s}} \\ &= \frac{\int_0^{1000 \text{ s}} m_{\text{油泵}}(t) dt}{1000 \text{ s}} - \frac{\int_0^{1000 \text{ s}} m_{\text{喷油}}(t) dt}{1000 \text{ s}} - \frac{\int_0^{1000 \text{ s}} m_{\text{减压阀}}(t) dt}{1000 \text{ s}} \\ &= 0\end{aligned}$$

通过刚才的分析可以知道,  $m_{\text{油泵}}(t)$  由一个参数: 转动角速度  $\omega$  确定,  $m_{\text{喷油}}(t)$  是一个确定的函数, 而对于减压阀  $m_{\text{减压阀}}(t)$ , 题目中没有提到除减压阀流量外的任何条件. 最直接的,  $m_{\text{减压阀}}(t)$  将由 0-1000 ms 每一时刻  $t_i$  阀门是否打开所确定. 也就是函数  $m_{\text{减压阀}}(t)$  有无穷多个参数  $t_1, t_2 \dots$  我们这样设计减压阀的控制方案以减小其自由度:

测量或计算预测当前高压油管内压强, 若压强 > 阈值, 则打开减压阀, 若压强 < 阈值, 则关闭泄压阀.

因此, 如何设计压强阈值, 使系统平均相对质量  $\langle m_r \rangle$  接近 0, 同时又使系统尽量稳定是这道题需要考虑的关键因素. 为了衡量系统的稳定程度, 引入  $m_r$  的方差:

$$\sigma_{m_r}^2 = \langle m^2 \rangle - \langle m \rangle^2 = \frac{1}{N} \left( \sum_{i=1}^N (m_r)_i^2 - N \langle m_r \rangle^2 \right)$$

在实际计算中,  $dt = 10^{-4} \text{ ms}$ , 总时间为  $t = 10^3 \text{ ms}$ , 因此上述  $N = 10^7$ . 经过简单的分析可以发现, 当阈值过高时, 减压阀难以开启, 起到的效果不大, 无法使  $\sigma_{m_r}^2$  大幅降低. 当阈值较低接近 100 MPa 时, 油罐内压强难以超过 100MPa, 无法使  $\langle m_r \rangle$  接近 0. 计算得到的不同压力阈值时  $\langle m_r \rangle$  和  $\sigma_{m_r}^2$  如表1所示. 发现当阈值压强取  $P_0 = 105 \text{ MPa}$ , 角速度取  $\omega = 0.11 \text{ rad/ms}$  时,  $\langle m_r \rangle$  最小, 为 4.85,  $\sigma_{m_r}^2$  最小, 为 132.7.

最小时不同时间  $m_r$  变化如图18所示

## 六、模型评价与改进

### 6.1 模型的优点

(1) 在模型的建立过程中, 忽略燃油压强改变的弛豫过程等次要条件, 将系统各部分部件解耦合得到了一个简化模型, 该模型具有均能较好地满足题目要求, 且具有较高的精度与稳定性.

(2) 问题三采用变步长多次枚举法遍历求得, 高压油管内压强稳定性得到显著提升.

角速度 $\omega$ (rad/ms)	阈值压强 $P_0$ (MPa)	$\langle m_r \rangle$	$\sigma_{m_r}^2$
0.0445	$\infty$	-0.79	296.8
0.055	110	-11.2	234.3
0.06	105	-12.5	199.1
0.09	100	-11.9	142.1
0.11	105	4.85	132.7

表 1 不同压力阈值时  $\langle m_r \rangle$  和  $\sigma_{m_r}^2$

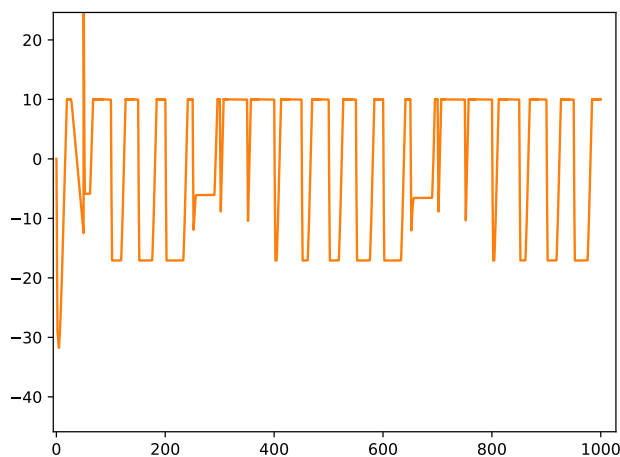


图 18  $\langle m_r \rangle$  和  $\sigma_{m_r}^2$  最小时  $m_r$  变化, 横轴为时间 (ms), 纵轴为  $m_r$ (mg)

## 6.2 模型的缺点

- (1) 求解压强稳定的最佳策略时, 忽略了压强的微小波动, 具有一定的局限性.
- (2) 查阅文献时发现, 高压油管内是否是单相液质尚未有统一的定论, 为了模型的简便, 本文未考虑双相可能会带来的微小扰动.

## 参考文献

- [1] 殷子嘉. 高压油管中燃油压力波传递速度的研究 [J]. 上海交通大学学报, 1992(01):47-51.

## 附录 A 题目一第一小问 – Mathematica 源代码

```
1 c = 0.85;  
2 a = 0.7^2 Pi ;  
3 Δp = 60;  
4 rho = 0.85;  
5 q = c a Sqrt[2 Δp/rho];  
6 t = 440*(0.85/0.87)/q  
7 Solve[1000 t0/(t0 + 10) == t, t0]
```

## 附录 B 第一题第二小问 – Python 源代码

```
1 import numpy as np  
2 from scipy import integrate  
3 import math  
4 import matplotlib.pyplot as plt  
5 fig = plt.figure()  
6  
7 C = 0.85  
8 A = 0.7**2 * math.pi  
9 vIn = 500 * 25 * math.pi  
10 mIn = vIn * 0.85  
11 rho_at_160 = 0.87  
12 p = 100  
13 ma = 7.31 #2000ms - 7.31mg --- 5000ms - 5.04mg --- 10000ms - 4.31mg  
14 open = True  
15 trange = 0.001  
16 ma_0 = 0  
17 top = p  
18 bottom = p  
19 storet = []  
20 storep = []  
21 storetop = [p]  
22  
23  
24 def q(dp, rho):  
25     return C * A * math.sqrt(2 * dp / rho)  
26  
27  
28 def rho(mIn):  
29     return (mIn / vIn)
```

```

30
31
32 def E(p):
33     return (1495 * math.exp(0.0039 * p))
34
35
36 last_in_oil_time = 0
37 opentime = 0
38 storeopentime = []
39 for ti in np.arange(1, 15000, trange):
40     if p > 150: break
41
42     rho0 = rho(mIn)
43     temp = ti % 100
44     if temp < trange: top = p
45     if temp < 0.2:
46         mIn -= trange * (temp * 10) * rho(mIn)
47     elif temp < 2.2:
48         mIn -= 20 * trange * rho(mIn)
49     elif temp < 2.4:
50         mIn -= (2.4 - temp) * trange * 10 * rho(mIn)
51         bottom = p
52     else:
53         pass
54
55     if (ti - last_in_oil_time) >= 10 and not open:
56         mIn_last_open = mIn
57         opentime = ti
58         open = True
59     if open:
60         ma_0 += trange * q(160 - p, rho_at_160)
61         mIn += trange * q(160 - p, rho_at_160)
62         if ma_0 >= ma:
63             open = False
64             last_in_oil_time = ti
65             ma_0 = 0
66             storeopentime += [ti - opentime]
67         drho = rho(mIn) - rho0
68         p += E(p) / rho(mIn) * drho
69         storet += [ti]
70         storep += [p]
71         if ti % 100 < 2.4:
72             storetop += [storetop[-1]]
73         else:
74             storetop += [(top + bottom) / 2]
75 print('ma =', ma, 'ti =', ti)
76 plt.xlabel("Time (ms)")

```

```

77 plt.ylabel("Pressure (MPa)")
78 plt.plot(storet, storep)
79 plt.plot(storet, storetop[1:])
80 plt.show()
81 fig = plt.figure()
82 plt.plot(storeopentime[3:])
83 plt.xlabel("the $N_{th}$ injection")
84 plt.ylabel("time for every injection (ms)")
85 plt.show()

```

## 附录 C 题目二喷油过程 – Python 源代码

```

1  import numpy as np
2  import math
3  import matplotlib.pyplot as plt
4  fig = plt.figure()
5
6  C = 0.85
7  vIn = 500 * 25 * math.pi
8  mIn = vIn * 0.85
9  p = 100
10 open = True
11 trange = 0.001
12 Ac = 0.7**2 * math.pi
13 storet = []
14 storep = []
15
16
17 def q(A, dp, rho):
18     if dp == rho == 0: return 0
19     return C * A * math.sqrt(2 * dp / rho)
20
21
22 def rho(mIn):
23     return (mIn / vIn)
24
25
26 def E(p):
27     return (1495 * math.exp(0.0039 * p))
28
29
30 def distance(t):
31     if t <= 0.45:

```

```

32         return 2.225 / (1 + 537.9 * math.exp(-19.3 * t))
33     elif t <= 2:
34         return 2
35     elif t <= 2.45:
36         return 2.267 / (1 + 456.4 * math.exp(18.63 * (t - 2.45)))
37     elif t <= 100:
38         return 0
39     else:
40         print("Error")
41
42
43 def Ab(t):
44     return ((distance(t) + 7.95775) * 9 * math.pi /
45             180)**2 * math.pi - 1.25**2 * math.pi
46
47
48 print(distance(1e-3), Ab(1e-3))
49
50 pb = 0.1
51 mb = 0.8 * 10.74
52 trange = 1e-3
53 penchuqu = 0
54 rho0 = rho(mIn)
55 pball = []
56 rhob0 = 0.8
57 rhob = 0.8
58
59 for ti in np.arange(0, 100, trange):
60     vb = math.pi * (distance(ti) + (7.95775 - 4.45634)) / 3 * (
61         ((distance(ti) + 7.95775) * 9 * math.pi / 180)**2 + .7**2 + .7 *
62         ((distance(ti) + 7.95775) * 9 * math.pi / 180))
63     rhob0 = rhob
64     if pb >= p:
65         rhob = rho0
66         mb = rho0 * vb
67         pb = 100
68     elif pb < 0.1 or rhob < 0.8:
69         pb = 0.1
70         rhob = 0.8
71         rhob0 = 0.8
72     else:
73         #print(q(Ab(ti), p - pb, rho0))
74         mb += q(Ab(ti), p - pb, rho0) * trange * rho0
75         penchuqu += q(Ab(ti), p - pb, rho0) * trange * rho0
76         rhob = mb / vb
77         pb += E(p) / rhob * (rhob - rhob0)
78     if pb < 0.1 or rhob < 0.8:

```

```

79         pb = 0.1
80         rhob = 0.8
81         rhob0 = 0.8
82         rhob0 = rhob
83         #print(pb,rhob)
84         mb -= q(Ac, pb, rhob) * trange * rhob
85         pball += [penchuqu]
86         if mb < 0.8 * vb:
87             mb = 0.8 * vb
88             pb = 0.1
89             rhob = mb / vb
90             if rhob != 0.8:
91                 pb -= E(pb) / rhob * (rhob0 - rhob)
92             if pb < 0.1:
93                 pb = 0.1
94             rhob0 = rhob
95         pball = np.array(pball)
96         print("penchuqu =", penchuqu)
97         pball = np.array([i * penchuqu + pball for i in range(10)]).flat
98         np.savetxt('D:\\Onedrive\\prog\\PhysicsNotes\\cumcm\\pball.txt', pball)
99         np.savetxt('D:\\Onedrive\\prog\\PhysicsNotes\\cumcm\\time.txt',
100                 np.arange(0, 1000, trange))
101         plt.plot(np.arange(0, 1000, trange), pball)
102         plt.xlabel('time (ms)')
103         plt.ylabel('mass out (mg)')
104         plt.show()

```

## 附录 D 题目二泵入油过程 – Python 源代码

```

1     import numpy as np
2     import math
3     import matplotlib.pyplot as plt
4     fig = plt.figure()
5     Aaa = 2.5**2 * math.pi
6     Aa = 0.7**2 * math.pi
7     omega = 0.3
8     x = 5.5315
9     V = x * Aaa
10    m = 92.31943
11    rho = 0.85
12    p = 100
13    theta = .515075
14    dt = 1e-4

```

```

15 C = 0.85
16 storep = []
17 storet = []
18
19
20 def Q(dp, rho):
21     return C * Aa * math.sqrt(2 * dp / rho)
22
23
24 def E(p):
25     return 1495 * math.exp(0.0039 * p)
26
27
28 def v(t):
29     return -omega * 2.413 * math.sin(omega * t)
30
31
32 mall = 0
33
34 storep += [p]
35 storet += [theta / omega]
36 for t in np.arange(theta / omega, 1000, dt):
37     if p < 100:
38         break
39     dx = v(t) * dt
40     x += dx
41     dV = Aaa * dx
42     V += dV
43     dm = -Q(p - 100, rho) * rho * dt
44     m += dm
45     mall += dm
46     drho = dm / V - m * dV / V**2
47     rho += drho
48     dp = E(p) / rho * drho
49     p += dp
50     storep += [p]
51     storet += [t]
52 plt.plot(storet, storep)
53 plt.xlabel('time (ms)')
54 plt.ylabel('Pressure (MPa)')
55 plt.show()
56 print(mall)

```



## 附录 E 题目三 – Python 源代码

```
1 import numpy as np
2 from scipy import integrate
3 import math
4 import matplotlib.pyplot as plt
5 from tqdm import tqdm
6 fig = plt.figure()
7
8 ##### in #####
9 Aaa = 2.5**2 * math.pi
10 Aa = 0.7**2 * math.pi
11 omega = 0.11
12 p0 = 10
13 '''
14 When omega = 0.0445 and p0 = 100 , We will have average oil = -0.7888128422628126 Variance =
    = 296.8055590776664
15 When omega = 0.055 and p0 = 10 , We will have average oil = -11.165893056647555 Variance =
    234.27149260504498
16 When omega = 0.06 and p0 = 5 , We will have average oil = -12.53827327326307 Variance =
    199.07614140826723
17 When omega = 0.08 and p0 = 0 , We will have average oil = -13.075589479119037 Variance =
    155.3312421142647
18 When omega = 0.09 and p0 = 0 , We will have average oil = -11.908094180614148 Variance =
    142.1443762712971
19 When omega = 0.11 and p0 = 5 , We will have average oil = -4.848550471253415 Variance =
    132.67064087579712
20 '''
21 x = 5.5315
22 V = x * Aaa
23 m = 92.31943
24 rho0 = 0.85
25 p = 100
26 theta = .515075
27 dt = 1e-4
28 C = 0.85
29 storep = []
30 storet = []
31
32
33 def Q(dp, rho):
34     return C * Aa * math.sqrt(2 * dp / rho)
35
36
37 def E(p):
```

```

38     return 1495 * math.exp(0.0039 * p)
39
40
41 def v(t):
42     return -omega * 2.413 * math.sin(omega * t)
43
44
45 mall = 0
46
47 storep += [0 for i in np.arange(0, theta / omega, dt)]
48 storet += [i for i in np.arange(0, theta / omega, dt)]
49 for t in np.arange(theta / omega, 1000, dt):
50     if p < 100:
51         tfinal = float(t)
52         break
53     dx = v(t) * dt
54     x += dx
55     dV = Aaa * dx
56     V += dV
57     dm = -Q(p - 100, rho0) * rho0 * dt
58     m += dm
59     mall += dm
60     drho = dm / V - m * dV / V**2
61     rho0 += drho
62     dp = E(p) / rho0 * drho
63     p += dp
64     storep += [mall]
65 period = 2 * math.pi / omega
66 storep += [mall for i in np.arange(tfinal, period, dt)]
67 storep = np.array(storep)
68 repeatN = int(1000 / period) + 1
69 storet = np.array([i for i in np.arange(0, repeatN * period + dt, dt)])
70 storep = np.array([storep + i * mall for i in np.arange(0, repeatN)]).flatten()
71
72 print('mass in =', -mall)
73
74 #####
75
76 ##### out 1 #####
77
78 C = 0.85
79 vIn = 500 * 25 * math.pi
80 mIn = vIn * 0.85
81 p = 100
82 open = True
83 trange = 0.001
84 Ac = 0.7**2 * math.pi

```

```

85     storet = []
86
87
88     def q(A, dp, rho):
89         if dp == rho == 0: return 0
90         return C * A * math.sqrt(2 * dp / rho)
91
92
93     def rho(mIn):
94         return (mIn / vIn)
95
96
97     def distance(t):
98         if t <= 0.45:
99             return 2.225 / (1 + 537.9 * math.exp(-19.3 * t))
100         elif t <= 2:
101             return 2
102         elif t <= 2.45:
103             return 2.267 / (1 + 456.4 * math.exp(18.63 * (t - 2.45)))
104         elif t <= 100:
105             return 0
106         else:
107             print("Error")
108
109
110     def Ab(t):
111         return ((distance(t) + 7.95775) * 9 * math.pi /
112                180)**2 * math.pi - 1.25**2 * math.pi
113
114
115     pb = 0.1
116     mb = 0.8 * 10.74
117     trange = dt
118     penchuqu = 0
119     rho0 = rho(mIn)
120     pball = []
121     rhob0 = 0.8
122     rhob = 0.8
123
124     for ti in np.arange(0, 100, trange):
125         vb = math.pi * (distance(ti) + (7.95775 - 4.45634)) / 3 * (
126             ((distance(ti) + 7.95775) * 9 * math.pi / 180)**2 + .7**2 + .7 *
127             ((distance(ti) + 7.95775) * 9 * math.pi / 180))
128         rhob0 = rhob
129         if pb >= p:
130             rhob = rho0
131             mb = rho0 * vb

```

```

132     pb = 100
133     elif pb < 0.1 or rhob < 0.8:
134         pb = 0.1
135         rhob = 0.8
136         rhob0 = 0.8
137     else:
138         #print(q(Ab(ti), p - pb, rho0))
139         mb += q(Ab(ti), p - pb, rho0) * trange * rho0
140         penchuqu += q(Ab(ti), p - pb, rho0) * trange * rho0
141         rhob = mb / vb
142         pb += E(p) / rhob * (rhob - rhob0)
143     if pb < 0.1 or rhob < 0.8:
144         pb = 0.1
145         rhob = 0.8
146         rhob0 = 0.8
147     rhob0 = rhob
148     #print(pb, rhob)
149     mb -= q(Ac, pb, rhob) * trange * rhob
150     pball += [penchuqu]
151     if mb < 0.8 * vb:
152         mb = 0.8 * vb
153         pb = 0.1
154     rhob = mb / vb
155     if rhob != 0.8:
156         pb -= E(pb) / rhob * (rhob0 - rhob)
157     if pb < 0.1:
158         pb = 0.1
159     rhob0 = rhob
160     pball = np.array(pball)
161     print("mass out =", penchuqu)
162     pball = np.array([i * penchuqu + pball for i in range(10)]).flatten()
163
164     #####
165     shift_pball = np.append(np.arange(0, 50, trange), pball)
166     final = -(storep[:int(1000 / trange)] + pball +
167               shift_pball[:int(1000 / trange)])
168
169     open2 = False
170     mass_out_per_dt = 20.02 * dt * rho0
171     plt.plot(np.arange(0, 1000, trange), final)
172     count = 0
173     final = np.array(final)
174     for i in tqdm(range(final.size)):
175         if not open2 and final[i] > p0:
176             open2 = True
177         if open2 and final[i] < p0:
178             open2 = False

```

```

179     if open2:
180         count += 1
181         final[i:] -= mass_out_per_dt
182
183     print('When omega =', omega, 'and p0 =', p0, ', We will have average oil =',
184           np.mean(final), 'Variance =', np.var(final))
185     plt.plot(np.arange(0, 1000, trange), final)
186     plt.show()

```

## 附录 F 数据处理 – MATLAB 源代码

```

1     c=exp(-exp(-0.0039*100)/0.0039/1495)/0.85;
2     rho=exp(-exp(-0.0039*0.5)/0.0039/1495)/c; %0.5MPa时的燃油密度
3     s=pi*2.5^2;
4     z=20/s;
5     m=rho*(z+4.826)*s;
6     v=m/0.85;
7     shendu=v/s;
8     jijing=z+2.413*3-shendu;
9     theta=acos((4.826-jijing)/2.413); %高压油泵单向阀打开时的数据
10
11     ta=0:0.001:0.51;
12     v=(1.0186+2.413)*s+2.413.*s.*cos(ta);
13     rho=m./v;
14     p=-log(-0.0039.*1495.*log(c.*rho))./0.0039;
15     plot(ta,p) %高压油泵单向阀打开前极径和压强之间的关系
16
17     Q=0.85*0.7^2*pi*sqrt(2*99.5/0.85); %D口流量

```