高压油管的压力控制

摘　要

本文以高压油管为研究对象，探究了入油、出油的周期时长、周期比，以及单向阀/油泵凸轮、喷油嘴、减压阀等各可控制单元之间的相位差对高压油管内油压的影响。

关于题目一:我们从整体压力变化、局部压力波动入手,在保证油压整体稳定在 100MPa 的前提下，最小化用于刻画波动程度的损失函数。由于油管内油压影响泵油、喷油的速率，而速率又反作用于油管，改变其油压，变量耦合度较高，获得解析解需要较多的近似处理，影响精确度；因此我们决定采取遍历的方式，并选取合适的步长，遍历可能的进油持续时间-相位组合，确定满足前提限制并最小化损失函数的最优解。

关于题目二：为了更好地了解喷油嘴的内部构造，我们首先查阅了相关文献，更加细致地了解了喷油嘴针阀的工作原理，以及限制喷油嘴燃油流量的存在的两个瓶颈界面；同时我们还运用 COMSOL 软件进行物理仿真模拟，直观地定位两个界面所在的位置（一是针阀与喷嘴内壁形成的圆环面，二是喷油嘴出口的圆面）。接着分析剖面，确定圆环面的流体速率场分布，进而确定两个瓶颈位置有效面积函数，根据临界条件分段计算得到了喷嘴喷速公式中除了压强与密度以外的系数。之后，沿用题目一的优化求解方式得到不同喷嘴外压强下的凸轮角速度，同时还发现该角速度与喷嘴外压强 (MPa) 有线性关系，拟合表达式为 $w = -0.1340p + 26.34$，R^2 为 0.9999，并从理论计算上说明了该线性结果之使然。

关于题目三：我们在喷油嘴由一个变为两个时，相较于问题二多设置了两个喷油嘴之间的相位差并对其进行遍历。由于复杂度较高，我们将步长从 0.01ms 调整为 1ms，最后发现角速度与喷嘴外压强（MPa）也存在线性关系，拟合表达式为 $w = -0.2664p + 52.51$，R^2 为 0.9996；得到两喷油嘴之间的相位差应为 50ms 左右，即周期的一半，符合对称性。在引入减压阀之后，将减压阀的功能视作：使油泵凸轮工作周期与喷油器相同（100ms）以便控制，并使管内压强更稳定。在假设减压阀周期也为 100ms、每周期仅开启一次的情况下，以 0.01ms 为步长遍历了油泵与某一喷油器的相位差、两喷油器之间的相位差、减压阀与某喷油器之间的相位差、减压阀的单次开启持续时间，得到针对 0.1MPa、1MPa、4MPa 的喷嘴外气压的最优控制策略。

目录

1 问题的重述

已知高压油管的基本规格参数，求解不同场景下能稳定油管内压力的解决方案。

问题 1：喷油嘴（出油端）定期向外喷油，为了维持油管内油压的稳定，求解单向阀（供油端）每次供油的时长。

问题 2：在高压油管工作过程中，单向阀、喷油嘴、凸轮需根据参数精确分析：

- **单向阀**（供油端）：由压力差控制开闭凸轮驱动活塞上下运动，若活塞上行，会压缩燃油的体积，造成油压上升，当油泵与油管的压力大小相等时，到达临界情况，单向阀打开，进一步的压缩，使燃油进入到油管，实现油料补充；若下行，则会补充油泵内的油料并停止向油管内供油。
- **喷油嘴**（出油端）：针阀在升程大于 0 时开启，与圆锥截面形成一个圆环通道，燃油可通过此截面，进入到下方的锥状腔内，再通过喷油嘴雾化喷出。在一定的压力下，喷油嘴单位时间喷出的燃油量是一定的，而在一次往复运动中油泵向油管中补充的油量是一定的，因此只需保证喷油与补充速率相同，即可实现稳压。
- **凸轮**：在一个周期内往复运动，向油管中泵入燃油；通过“出 = 入”等式，可以确定为补充喷出的油料，凸轮应有的角速度。

2 模型假设

为了让模型更便于处理，我们在不影响效果的前提下，做以下假设：

- **全局假设**：
    1. 体系温度不发生改变
    2. 高压油管体积不改变
- **局部假设**：
    1. 问题 2 中当油泵向高压油管供油时，油泵与油管内的燃油压强梯度极小
    2. 问题 2 中针阀底部与密封座内壁形成圆环状的截面，将环状截面的面积作为该局部燃油通过的有效面积，即“此截面处经过的流体为该截面的法向流”。
    3. 问题 2 中针阀底部与密封座内壁形成圆环状的截面，将环状截面的面积作为该局部燃油通过的有效面积，即“此截面处经过的流体为该截面的法向流”。

4. 问题 2、3 的题干未给出喷油嘴外的具体压强，我们用 p_0 代表，令 p_0 遍历一个压强列表

3 参数对应表

符号	含义
$\mathbf{A}$	A 处小孔面积
$\mathbf{p_t}$	t 时刻时高压油管内压力（以 Δt 为单位时间）
$\mathbf{p_{start}}$	每个泵油、喷油周期开始时，高压油管的压力
$\mathbf{p_{end}}$	每个泵油、喷油周期结束时，高压油管的压力
$\mathbf{p_{in}}$	油泵泵入燃油造成的压力变化
$\mathbf{p_{out}}$	喷嘴喷出燃油造成的压力变化
$\mathbf{p_{pump}}$	高压油泵内的燃油压力
$\rho_{\mathbf{p=160MPa}}$	在 $160MPa$ 压力下燃油的密度
$\mathbf{E_{p_t}}$	在 t 时刻的压力下，燃油的弹性模量
$\mathbf{Q_{in}}$	自油泵流入的燃油流量
$\mathbf{Q_{out}}$	自喷嘴流出的燃油流量
$\mathbf{V}$	高压油管的容积
$\mathbf{f_{in}}$	油泵入油的频率
$\mathbf{f_{out}}$	喷油嘴出油的频率
$\Delta\phi$	高压油泵内的燃油压力
ρ_t	t 时刻油管内燃油密度
$\mathbf{S}$	t 高压油泵柱塞腔横截面积
$\mathbf{p_0}$	t 喷油嘴外部的外界压力
ω	t 凸轮角速度

4 模型的建立与求解

4.1 问题一

4.1.1 问题分析

从压力变化结果、压力变化过程两个方面入手，我们做出了较为全面的考虑：

1. 每一次入油、出油结束后，末状态的压力与初状态的压力偏离尽可能的小；保障在每个周期结束后压力基本不变。

2. 在入油、出油过程中，压力波动的方差尽可能的小，减小喷油过程中高压油管内的压力波动幅度；保障喷油嘴压力稳定，喷油均匀。

根据题目所给信息，我们不难发现：

1. 高压油管压力 p 在求解中占据关键的位置，一方面，压力 p 决定了单次入油流量及出油流量，另一方面，入油（Δp_{in}）、出油（Δp_{out}）过程也使油管内压力 p 波动。综上所述，高压油管压力 p 在模型中既是各个过程的自变量，也是最终由各个过程所左右的因变量。

2. 我们最终的目标是使得高压油管压力 p 在整个过程中更加稳定，所以需要得知每一个时刻 t 所对应的高压油管压力 p，并尝试优化相关参数使得压力随时间改变的 $P(t)$ 函数更加平稳，符合预期。

3. 题目所给数据精确到 10^-5

4.1.2 模型计算

综合**模型分析**部分，我们不妨使用**递推模拟 + 逐步逼近**的方法 (以为步长计算每一时刻的高压油管压力值)，来解决“模型建立”部分提出的两方面问题，我们导出如下公式：

- **油管压力递推**：$t+1$ 时刻的管内压力 p_{t+1} 是在 t 时刻的管内压力 p_t 的基础上，考虑由油泵入油引起的 p_{in} 变化，以及由喷油嘴出油引起的 p_{out} 变化，由此我们可以得到压力递推公式：
$$p_{t+1} = p_t + p_{in} + p_{out} \tag{1}$$

- **出入油压力变化推导**：油泵入油引起的压力变化 p_{in} 可通过注 1（燃油的压力变化量与密度变化量成正比 d，比例系数为 $\frac{E}{\rho}$ ）得到：
$$p_{in} = \frac{E_{p_t}}{\rho_t} \times \frac{Q_{in}\Delta t}{V} \times \rho_t = \frac{E_{p_t} Q_{in}\Delta t}{V} \tag{2}$$
同理可得喷油嘴出油引起的压力变化 p_{out}：
$$p_{out} = -\frac{E_{p_t}}{\rho_t} \times \frac{Q_{out}\Delta t}{V} \times \rho_t = \frac{E_{p_t} Q_{out}\Delta t}{V} \tag{3}$$

其中，Q_{in} 为 t 时刻至 $t+1$ 时刻间高压油泵泵入的燃油流速，可由注 2 的公式导出：

$$Q_{in} = CA\sqrt{\frac{2(p_{pump} - p_t)}{\rho_{p=160MPa}}} \tag{4}$$

同时，Q_{out} 为 t 时刻至 $t+1$ 时刻间喷油嘴喷出的燃油流速，由题目图 2 给出。

4.1.3 优化目标

- **整体压力恒定**：保障每周期内大部分时间，也就是不喷油且不入油的时间段内，压力稳定在 $100MPa$ 附近，允许偏移量为 Δp，即：

$$|p_{stable} - 100| < \Delta p \tag{5}$$

  在本题所给参数的精度下，为避免无效的计算，我们取 $\Delta p = 0.01MPa$。

- **局部压力稳定**：尽量减小入油出油时的压力波动，让入油、出油量动态平衡。以每一时刻与目标稳定值 100MPa 作差取平方之和作为损失函数（如下式），并使该损失值取到最小。

$$LOSS = \sum_{t=start}^{end} (p_t - 100)^2 \tag{6}$$

4.1.4 优化变量

- **单位周期入油时长**：合适的单位周期内入油时长保证了整体的压力恒定，也即让出油量与入油量在较长时间内保持一致。

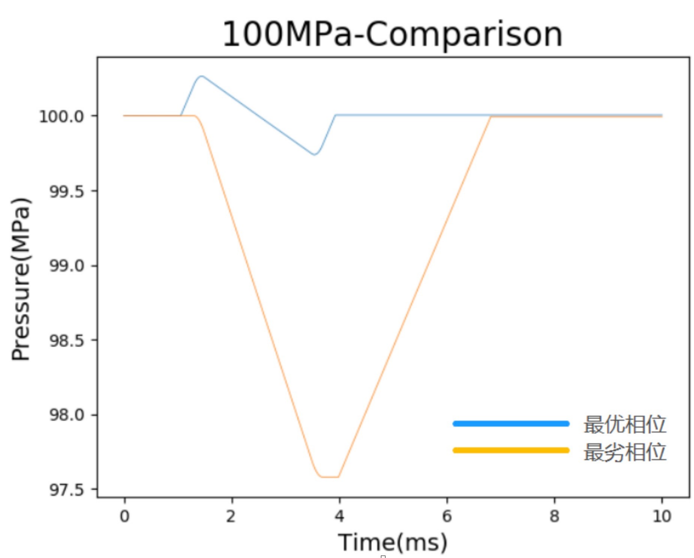


图 1: 相位不同时的 $p(t)$ 图像对比

- **入油相位**（每周期内入油开始时间较出油开始时间的提前量）：为稳定高压油管内的压力，应尽量使泵油、喷油的相位一致，这样可以达到最小化压力波动的目标。例如在上图中，两条曲线所对应的单位周期入油时长都可以使得每个周期的末状态压力稳定在 100MPa 附近，但是未优化的相位会有 2.4MPa 的压力波动峰值，对喷油量的稳定性有较大影响；而优化后的波动峰值则控制在 0.3MPa 以内。

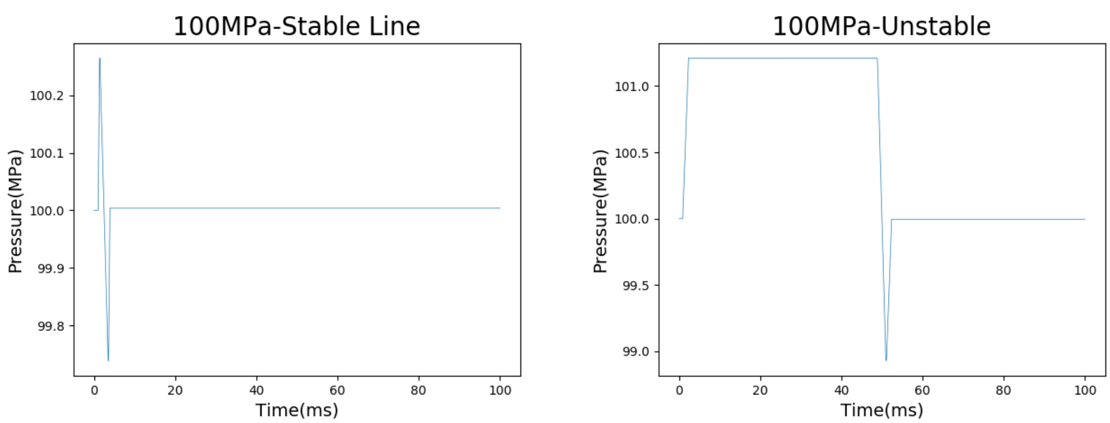
  命题 1 单向阀开启的相位在很大程度上影响了压力随时间的波动。

4.1.5 模型求解过程

1. **入油出油周期比讨论**：在模型求解过程中，我们还考虑到了入油出油周期比的取值，并得出结论、分析如下。

   命题 2 入油周期与出油周期一致才能最小化压力波动。

   实际情况中，出油周期为 100ms，而在这 100ms 中只有 2.4ms 在喷油，而出油流速与入油流速也相去无几，为了平衡这 2.4ms 内喷油带来的压力减小，应该在喷油的同时入油以抵消出油带来的压降。也就是说，最优选择是在每段出油过程的同时入油，这也就要求了出油与入油的周期一致。为了进一步加强我们的分析，我们做了入油出油周期比为 2:1 的实验得到相应的 $p(t)$ 图像，并与入油出油周期比为 1:1 的 $p(t)$ 图像进行对比，如下图：



(a) 周期比 1:1　　(b) 周期比 2:1

图 2: 周期不同时的 $p(t)$ 对比

   不难发现，在周期比为 1:1 时入油能立刻抵消出油造成的压降，并迅速使压力稳定至 100MPa，而周期比为 2:1 时有很长一段时间压力与 100MPa 相去甚远，且压力波动很大。

2. **求取最优过程**：之前提到过，我们通过改变单位周期入油时长以及入油相位来使得油管内压力波动达到最小（即满足优化目标中的两条）。在本道题目中我们用程序模拟并计算每一时刻的压力变化（运用之前列出的递推公式），并设置步长 $\Delta t = 0.01ms$，即每 0.01ms 更新一次当前油管内压力。通过遍历单位周期内单向阀开启时长（从 0.01ms 到 5.00ms），同时遍历相位差（从-3.00ms 到 3.00ms），筛选出满足优化目标 1 且最小化优化目标 2 中损失函数的时长及相位差组合。

3. **模型结果**：经过对参数大小的遍历以及模型的计算，我们得到的最优的单位周期（100ms）内单向阀开启时长为 2.88ms，同时在同周期内入油较出油提前 0.24ms。在这样的条件下，高压油管压力关于时间的曲线如下图所示：

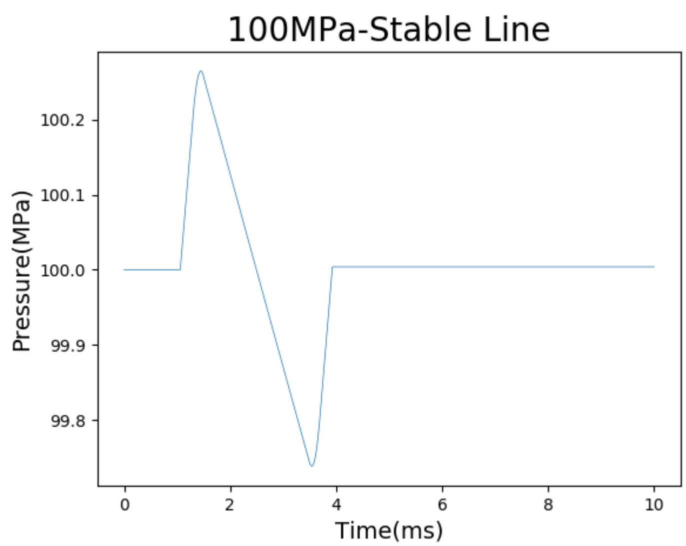


图 3: 最优情形的 $p(t)$ 图

4.1.6 针对后一小问的模型改进

在后一个小问题中，我们需要求得新的单向阀开启策略使得经过约 2s、5s 和 10s 的调整过程后油管内压力从 100MPa 增加到 150MPa，并在之后阶段稳定在 150MPa。所以，我们可以将总体过程分为两个阶段，首先是**压力提升阶段**，之后是 **150MPa 压力稳定阶段**，在两个阶段内单向阀有不同的开启策略。

在压力提升阶段，我们引入两个自变量：单向阀开启周期 T 以及每周期内开启时长 C，通过优化这两个变量使得在相应时间段能将压力近乎线性并准确地提升到 150MPa，也即最小化如下损失函数：

$$LOSS = \sum_{t=start}^{end} (p_t - (100 + \frac{50}{end - start}(t - start)))^2 \tag{7}$$

其中，$start = 0$，对于 2s 的压力提升，$end = 20000$；对于 5s 的压力提升，$end = 50000$；对于 10s 的压力提升，$end = 100000$。注意到，在这一小问题中为了降低问题的复杂度我们改取时间 t 的步长为 0.1ms。同时，我们还增加了一个约束条件，就是在 2s/5s/10s 末，油管压力被精确地提升至 150MPa。接下来，我们通过一个例子阐释增加此约束条件的必要性。如图，

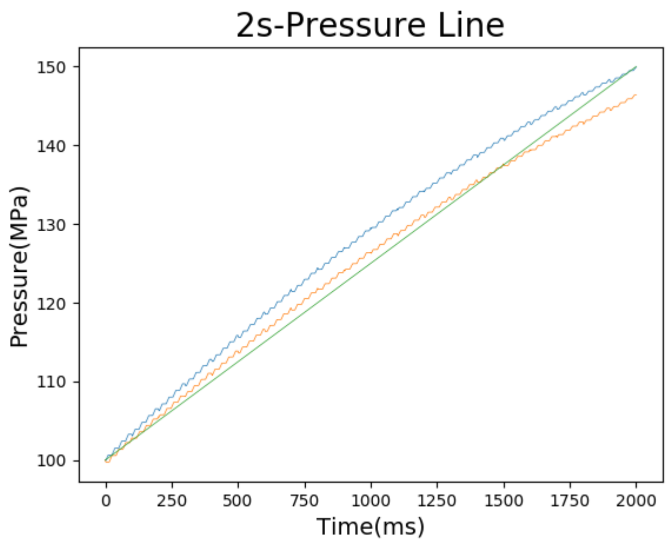


图 4: 有无该约束条件的的 $p(t)$ 图对比

绿色直线为在 2s 内压力由 100MPa 纯线性提升至 150MPa 的理想 $p(t)$ 曲线，红色、蓝色两条曲线分别为 $\{T = 25.9ms, C = 9.7ms\}$ 以及 $\{T = 25.5ms, C = 10.5ms\}$ 情况下的 $p(t)$ 曲线。容易发现，红色曲线在整个过程中更加接近理想 $p(t)$ 曲线（即绿色直线），但它在最终 2000ms 时刻并没有将油管内压力提升到 150MPa。蓝色曲线虽然与理想 $p(t)$ 曲线相去更远，但它在 2s 末将油管压力被精确地提升到了 150MPa。所以，我们最终选择蓝色曲线所对应的 $\{T, C\}$ 参数，而舍去红色曲线。可见，我们需要以下约束条件：

$$|p_{end} - 150| < \Delta p \tag{8}$$

在本题模型中，由于计算精度所限，我们取 $\Delta p = 0.1ms$。
在 150MPa 压力稳定阶段，模型与前一小问中基本一致，只需改变优化目标 1 中公式：

$$|p_{stable} - 150| < \Delta p \tag{9}$$

并改变优化目标 2 中损失函数：

$$LOSS = \sum_{t=start}^{end} (p_t - 150)^2 \tag{10}$$

4.1.7 针对后一小问的模型求解过程

1. **降低复杂度**：在求解压力提升阶段时的单向阀开启策略时，由于需要同时遍历单向阀周期 T（从 0.1ms 到 200.0ms）以及单向阀在一个周期内的开启时间（从 0.1ms 到 50.0ms），这样的计算复杂度巨大，所以我们先以 1ms 为步长遍历周期及开启时间，并计算各 $\{T, C\}$ 对下的损失函数（由式（7）给出），利用 MATLAB 绘制出了如下图像：

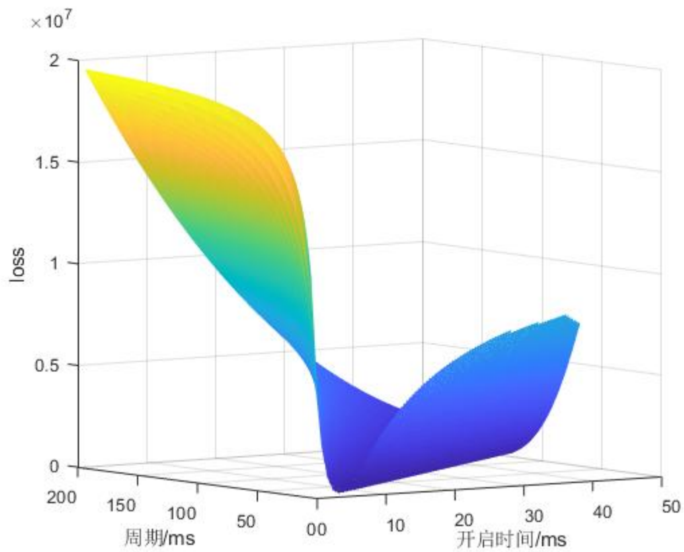


图 5: LOSS 关于 T,C 的图像（2ms 提升过程）

不难发现，在全 $T-C$ 空间内 LOSS 最小的的 $\{T, C\}$ 对集中在一条 T 关于 C 的直线附近。其实这也正和我们预期相符合，考虑到单向阀周期 T 远小于总时长 2s，也就是说在 2s 内总计入油量近似正比于 $\frac{C}{T}$，而要将油管压力在 2s 内从 100MPa 提升至 150MPa 所需总油量大致为一个定值，所以较优 $\{T, C\}$ 解的 $\frac{C}{T}$ 均相近。

命题 3 能最小化损失函数的较优 $\{T, C\}$ 解具有相近的 T 与 C 的比值。

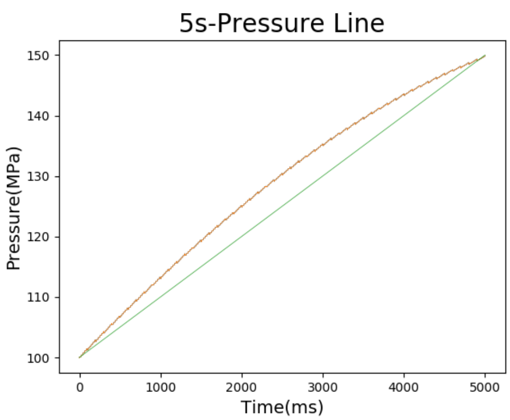
由此结论，我们便可以减小搜索空间，从而降低问题求解的复杂度。

2. **求取最优过程**：在求取压力提升阶段的最优解时，我们用程序模拟并计算每一时刻的压力变化，并设置步长 $\Delta t = 0.1ms$，即每 0.1ms 更新一次当前油管内压力，遍历 T 与 C 值并筛选出满足公式（8）且最小化公式（7）中损失函数的单向阀周期及每周期内开启时长组合。而在求取 150MPa 稳定阶段的最优解时，运用与前一问一样的方法，在方程（9）的约束下最小化方程（11）中的损失函数，得到最优的单向阀在单周期内开启时长以及相位差。

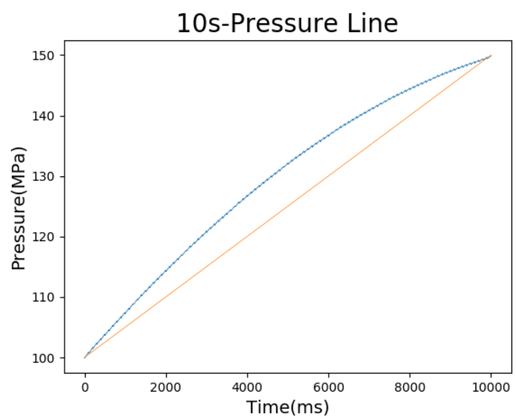
3. **模型结果**：经过对参数大小的遍历，我们得到的最优单向阀周期及每周期内开启时长组合如下表所示：

提升时间 t（s）	单向阀周期 T（ms）	开启时长 C（ms）
2	25.5	10.5
5	20.9	4.0
10	24.5	2.9

其中 2ms 时最优 $\{T, C\}$ 组合对应的 $p(t)$ 曲线如图 5 的蓝色曲线所示，5ms、10ms 时最优 $\{T, C\}$ 组合对应的 $p(t)$ 曲线如下图所示：



(a) 5ms 提升过程 $p(t)$ 曲线



(b) 10ms 提升过程 $p(t)$ 曲线

图 6

同时我们得到 150MPa 压力稳定阶段的最优单位周期（100ms）内单向阀开启时长为 7.03ms，同时在同周期内入油较出油提前 2.31ms。在这样的条件下，高压油管压力关于时间的曲线如下图所示：

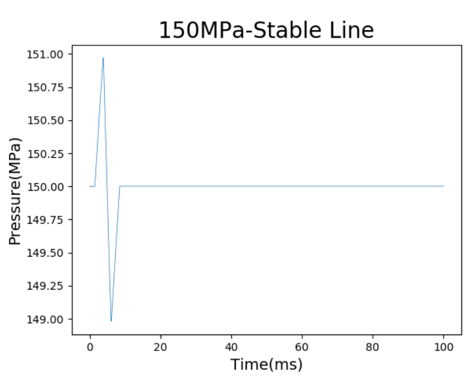


图 7: 150MPa 下 $p(t)$ 稳定曲线

4.2 问题二

4.2.1 问题分析

工作原理:

1. 供油端凸轮转动，不同角位置的极径不同，进而驱动柱塞做周期性往复运动，油泵向高压油管供油。
2. 喷油端喷油嘴内的针阀升程大于 0 时，高压油管内燃油通过环形截面，进入到针阀下方的锥形腔内，从喷嘴雾化喷出。

工作组件:

1. **凸轮**
   凸轮的角速度为 ω，设极径 r 关于极角 θ 的函数为

$$r = f(\omega)$$

   由题意，我们可以得到以下结论:

   (a) 柱塞下行阶段，油泵补充低压燃油，单向阀关闭。
   (b) 柱塞上行且油泵压强未达到油管压强值阶段：单向阀关闭，泵内燃油增压。
   (c) 柱塞上行且油泵压强不小于油管压强值阶段：在极短的时间 Δt 内，柱塞的位移 Δh=Δr=$\Delta f(\theta)$

$$\Delta V = S \times \Delta f(\theta)$$

$$Q_{in} = \frac{\Delta V}{\Delta t}$$

   由此，我们可以计算出油泵的供油速率。

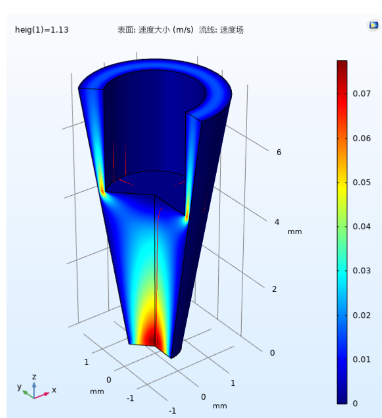


图 8: 针阀上升 h = 1.13mm 喷油嘴内部流速场

2. **喷油嘴**

运用 COMSOL Multiphysics 5.4 软件，我们成功地模拟了如上图在理想情况下喷油嘴内流体的速度场。通过可视化，我们可以初步确定，喷油嘴有两个限制瓶颈，一个是环状截面，另一个是出油口。

我们猜想：针阀抬起时，密封座与针阀之间可供流体流动部分的截面积限制了从喷孔喷油的速度；当该截面积较大时，则由于喷孔面积较小，实际限制喷速的是下方的喷孔。使用 COMSOL 软件流体模块对针阀升起距离为 0.5mm、1mm、1.5mm、2mm 的情况进行模拟，验证该猜想。

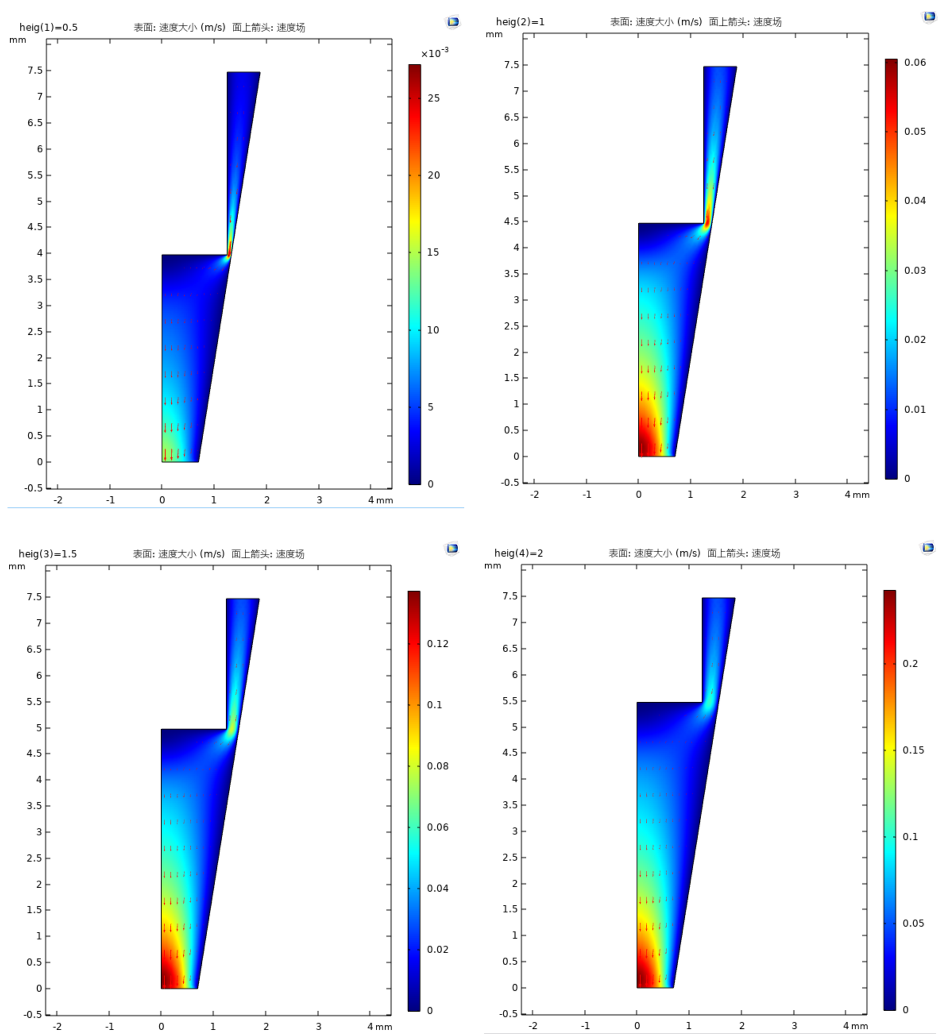


图 9: 在不同针阀升程下流体场的模拟结果

模拟的几何形体与喷嘴放大后示意图一致，模拟过程中在入口处（最上方）施加压强 100MPa，在出口处（最下方）施加压强 4MPa，温度为 293.15K，选取流体为软件内置的 Diesel fuel，在层流条件下得到稳态结果如上图所示 (图片左上标识 heig(i)=k 中 k 表示针阀升起距离 (mm))。

4.2.2 模型计算

与问题一相似，我们同样采用**递推模拟 + 逐步逼近**的方法，采用以极小间隔分割时间并递推方法推出每个时刻的油管压力值 p，具体推导公式如下：

- **油管压力递推**：与上一问相同，

$$p_{t+1} = p_t + p_{in} + p_{out} \tag{11}$$

- **出入油压力变化推导**：同样地，有

$$p_{in} = \frac{E_{p_t} Q_{in} \Delta t}{V} \tag{12}$$

$$p_{out} = \frac{E_{p_t} Q_{out} \Delta t}{V} \tag{13}$$

  其中，Q_{in} 为 t 时刻至 $t+1$ 时刻间高压油泵泵入的燃油流速：

$$\begin{aligned} Q_{in} &= \frac{S(r(\theta_{t+1}) - r(\theta_t))}{\Delta t} \\ &= \frac{S(r(\omega(t+1)) - r(\omega t))}{\Delta t} \end{aligned} \tag{14}$$

  同时，Q_{out} 为 t 时刻至 $t+1$ 时刻间喷油嘴喷出的燃油流速：

$$Q_{out} = CA\sqrt{\frac{2(p_t - p_0)}{\rho_t}} \tag{15}$$

  其中，A 为限制喷速的瓶颈面积，即当喷孔面积更小时，A 为喷孔面积，当密封座与针阀间截面面积更小时，A 为密封座与针阀间截面面积。

4.2.3 优化目标

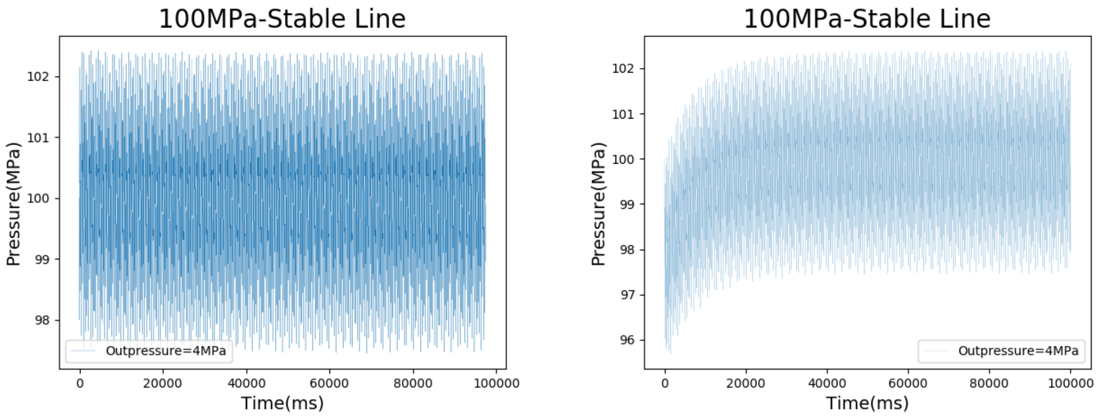
- **整体压力稳定**：尽量减小压力波动，让入油、出油量动态平衡。以每一时刻与目标稳定值 100MPa 作差取平方之和并求平均作为损失函数（如下式），并使该损失值取到最小。

$$LOSS = \frac{\sum_{t=start}^{end} (p_t - 100)^2}{end - start} \tag{16}$$

4.2.4 优化变量

- **凸轮转动角速度** ω：调整凸轮转动角速度可以保证平均入油速度与平均出油速度持平，使油管内压力始终在 100MPa 上下周期性浮动。

- **初始相位差**：定义凸轮从最低点状态开始旋转较喷油嘴开启时刻提前的时间为初始相位差。调整初始相位差，可以使得压力波动稳定下来的时间提前。如下图中凸轮角速度完全一致，但相位差不一样导致了明显的压力波动幅度的不同，当然这种不同主要体现在压力波动稳定所需时长，如 (a) 图压力波动很快进入稳定状态，而 (b) 图中过了约 20s 才进入稳定状态。



(a) 初始相位差为 113ms　　(b) 初始相位差为 0ms

图 10: 角速度 $\omega = 25.80 rad/s$

4.2.5 模型求解过程

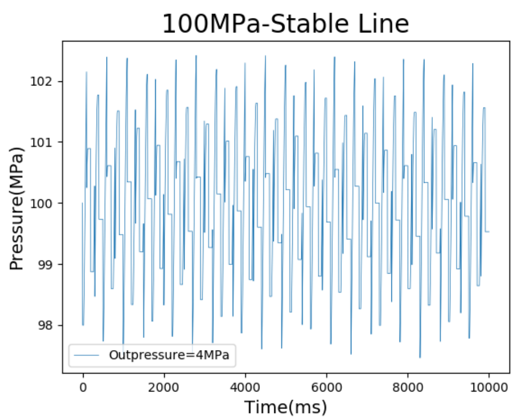
- **喷油嘴外界压力**：由上述计算公式可知，在计算喷油嘴喷油流量时需要喷油嘴外界压力的信息，经查阅相关资料我们得到喷油嘴外界压力在 0.1MPa 至 10MPa 范围内，所以我们分别取 0.1,0.5,1,2,3,4,5,10MPa 作为喷油嘴外界压力分别求解相应的最优解。

- **求取最优过程**：我们通过改变凸轮角速度以及初始相位差来使得油管内压力波动达到最小（即满足优化目标）。我们仍然采用程序模拟的方法计算每一时刻的压力变化（运用之前列出的递推公式），并设置步长 $\Delta t = 0.01ms$。通过遍历一段相对较长时间内（20s ~ 100s）凸轮转动一圈的周期（从 200ms 到 300ms），同时遍历相位差（从-100ms 到 150ms），筛选出最小化优化目标 1 中损失函数的周期及初始相位差组合。

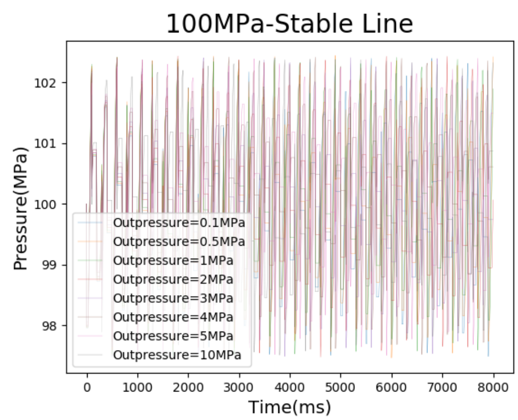
外界压力 p_0（MPa）	凸轮角速度 ω（rad/s）	初始相位差（ms）	方差（MPa）
0.1	26.32	111	1.10
0.5	26.27	113	1.10
1	26.20	114	1.10
2	26.07	113	1.10
3	25.94	113	1.10
4	25.80	113	1.10
5	25.68	111	1.09
10	24.99	112	1.08

- **模型结果**：针对不同的喷油嘴外压力值（0.1MPa, 0.5MPa, 1MPa, 2MPa, 3MPa, 4MPa, 5MPa, 10MPa），我们各求得一组最优的凸轮角速度及初始相位差，如上表所示。为了说明采取各个方案时压力波动的稳定性，我们在每一种方案后给出了此时压力 p 波

动的方差（以 1ms 为单位离散化后进行计算)，并给出各种情况下压力关于时间波动图 $p(t)$：



(a) 单条曲线最优波动图像



(b) 多条曲线最优波动图像

图 11: $p(t)$ 函数图像

同时，我们还有一个新的发现，就是凸轮角速度 ω 与外界压力 p 近似线性关系，如下图：

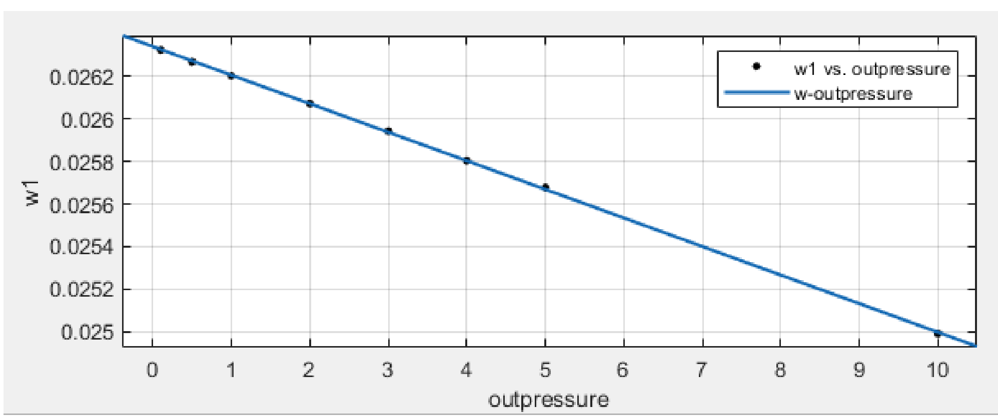


图 12: ω 与 p 近似线性关系

接下来，我们尝试用一些近似计算去解释此结果。由于高压油管内的燃油油压保持稳定，因此，油管系统应保证入油量 = 出油量，我们考察 t 为时间间隔，T_{out} 为一个喷油循环的周期时长，Q_{out} 为一个喷油周期内喷出的油量，T_{in} 为一个泵油循环的周期时长，Q_{in} 为一个泵油周期内泵入的油量。我们可以作如下推导：

$$Q_{out} = CA\sqrt{\frac{2(p-p_0)}{\rho}} \tag{17}$$

$$V_{out} = Q_{out} \times \frac{t}{T_{out}} \tag{18}$$

$$T_{out} = \frac{1}{\omega} \tag{19}$$

$$Q_{in} = S \times \frac{(Cam_{max} - Cam_{min})}{T_{in}} \tag{20}$$

$$V_{in} = \frac{1}{4}\pi {D_{cam}}^2 \times \frac{(R_{max} - R_{min})}{T_{in}} * t \tag{21}$$

$$Q_{out} \times \frac{t}{T_{out}} = \frac{1}{4}\pi {D_{cam}}^2 \times \frac{R_{max} - R_{min}}{T_{in}} \times t \tag{22}$$

化简可得:

$$k \times \omega_{in} = \sqrt{p - p_0} \tag{23}$$

由于 $p >> p_0$:

$$\sqrt{p - p_0} = \sqrt{p} \times \sqrt{1 - \frac{p_0}{p}} \tag{24}$$

而 $\frac{p_0}{p}$ 趋近于 0，由二项展开，我们可以做以下近似:

$$\sqrt{p} \times \sqrt{1 - \frac{p_0}{p}} \sim \sqrt{p} \times (1 - \frac{1}{2}\frac{p_0}{p}) \tag{25}$$

由此易证:

$$p = \lambda\omega + \delta \tag{26}$$

即 p 与 ω 呈良好的线性关系。

4.3 问题三

4.3.1 问题分析

在增加一个喷油嘴后，我们不仅需要考虑喷油嘴的开启时长变化，还应考虑两个喷油嘴的协调工作；通过分析我们发现，当两个喷油嘴的工作时段均匀地将时间轴分割时，能达到最好的稳压效果。

在一个油泵、两个喷油嘴的情况下，油泵的周期略长于喷油嘴的周期；而引入一个减压阀之后，则会导致自由度过多而难以求解的情况。为此，我们从实际情况考虑，希望减压阀能做到让油泵的工作周期与喷油嘴相同（即 100ms)，从而便于控制；而减压阀工作周期也需要与喷油嘴相同。同时我们还假设减压阀在一个周期中只开启一次，以便于计算。

4.3.2 优化目标

类似于问题二，我们仍以最小化 $MSE\ Loss$ 为优化目标，遍历求解。

4.3.3 模型计算

p 为更新后的高压油管内压强，p_0 为高压油管内原始压强，Q 为单位时间内从油泵流向油管的燃油量（mm^3/ms），V 为油管容积，Δp_{pump} 为油泵泵入燃油造成的油压变化，Δp_q 为喷油嘴 A 喷油造成的压力变化，Δp_{q2} 为喷油嘴 B 喷油造成的压力变化，Δp_{q3} 为减压阀造成的压力变化，$time$ 为当前时刻。迭代式为：

$$p = p_0 + \Delta p_{pump} + \Delta p_q + \Delta p_{q2} + \Delta p_{q3} \tag{27}$$

$$\Delta p_{pump} = Q \times \frac{E[time-1]}{V} \tag{28}$$

$$\Delta p_q = -q \times \Delta t \times \frac{E[time-1]}{V} \tag{29}$$

$$\Delta p_{q2} = -q_2 \times \Delta t \times \frac{E[time-1]}{V} \tag{30}$$

$$\Delta p_{q3} = -q_3 \times \frac{E[time-1]}{V} \tag{31}$$

4.3.4 优化变量

类似于之前的两问，我们的优化变量仍然包括：油泵和喷油嘴 A 的相差 $\Delta phase_{pump-A}$，凸轮周期 T（表格中转为角速度）。

在第二问中，固定凸轮周期为 100ms，同时增加了减压阀和喷油嘴 A 的相位差 $\Delta phase_{valve-A}$，以及减压阀每次持续开启的时长 T_0。

4.3.5 模型求解过程

- **求取最优过程**：将先前的喷油嘴命名为喷油嘴 A，新增的喷油嘴命名为喷油嘴 B，记泵为 pump。记 A、B 的相位差为 $\Delta phase_{B-A}$，油泵凸轮、喷油嘴 A 之间相位差为 $\Delta phase_{pump-A}$。为了最小化 $MSE\ Loss$，我们以 0.01ms 为步长，遍历了如下变量：油泵和喷油嘴 A 的相位差 $\Delta phase_{pump-A}$，喷油嘴 B 和喷油嘴 A 的相位差 $\Delta phase_{B-A}$，减压阀和喷油嘴 A 的相位差 $\Delta phase_{valve-A}$，减压阀每次持续开启的时长 T_0. 以取得最优解。

表 2: P-T 关系结果

外界压力 p_0(MPa)	凸轮角速度ω(rad/s)	$\Delta phase_{pump-A}$	$\Delta phase_{B-A}$	$LOSS$
0.1	52.49	54	51	1.11
0.5	52.36	55	51	1.11
1	52.23	58	49	1.10
2	51.97	57	50	1.12
3	51.71	56	50	1.09
4	51.46	58	50	1.10
5	51.21	55	51	1.11
10	49.83	55	50	1.10

- **模型结果**:

  由结果表格，我们不难发现 w 和 p 存在线性关系

  $$w = -0.2664p + 52.51$$

  最优解情况下，$\Delta phase_{B-A}$ 为 50 左右，这一特性也并非偶然，从直观上的对称性而言，我们也可以理解（两个喷油嘴各占一半时间，相位差占周期的 50%）。

- **后一问模型结果**:

外界压力 (MPa)	泵的相位	喷油嘴 B 的相位	阀的相位	减压阀开启时长	loss
0.1	61.11	17.99	30.76	0.70	0.41
1	61.75	17.16	29.37	0.73	0.62
4	61.75	17.21	29.30	0.77	0.37

喷油得以平均化导致 loss 小了很多，之前情况 loss 没有相较于问题二情形明显减少可能是因为遍历步长为 1ms，较为粗糙。观察一个周期内的情况可知，其喷油时间分布在泵油时间内，且较为均匀，而这正是减少方差的重要方式。至于图像中最开始略有偏移，则是有开始节点的原因：如果刚开始就是喷油，那么整体压强会略有下移；如果刚开始就是泵油，则整体压强会略上移；但最后都会渐渐趋于 100MPa。

5 模型优缺点及展望

- **精确性**：我们通过以步长迭代各个时刻的油管压力 p 成功地将各个变量解耦，避免了求多个耦合微分方程的连续解。同时由于步长设定非常微小 (0.01ms)，保证了模型所得结果的精确性。不过这种方法导致计算复杂度很高，也即模型求解过程比较复杂，

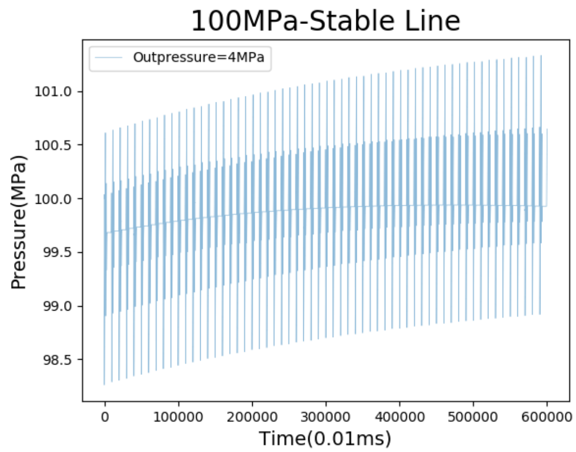


图 13: 压强 4MPa 时 $p(t)$ 曲线

我们用了多种方法减小模型的复杂度，比如第一问中先以较大步长找到全局最优再在全局最优附近精确搜寻最优点，再比如第二问中利用近似计算对模型进行预处理，较为精确地预测到最优结果值，再用小步长进行精确搜索。当然，受限于算力，我们的模型若得到更加强大算力的加持还能得到更加精确的最优解。

- **相位差**：除去问题设问中的开启时长、角速度等要求外，我们还对单向阀/油泵凸轮、喷油嘴、减压阀等可控制单元之间的相位差进行了研究，以获得在同样的开启时长、角速度等要求下相位意义上的最优解。对于问题三，相位差的研究更是体现出重要性，因为这是控制策略的重要组成部分。
- **物理过程**：将喷油嘴喷油等过程进行了简化，如在模拟中视为层流进行研究，而未考虑湍流等实际中会发生的现象，可能会影响模型的精确性。若我们有更加专业的知识以及更精确的数据则可建立一个更加全面的物理模型。

参考文献

[1] 高阳邓俊李理光，《小喷油量情况下喷油孔截面积对喷油器喷射性能的影响》, `http://qikan.cqvip.com/Qikan/Article/Detail?id=672609129&from=Qikan_Search_Index`

6 附录

以下是我们的代码（Python）

6.1 第一题前一问代码

```
import numpy as np
import matplotlib.pyplot as plt
import math
import csv

p2E = {}  #由p查表得到E
with open("") as csvfile:  #绝对路径名已省去
    csv_reader = csv.reader(csvfile)
    header = next(csv_reader)  # 读取第一行每一列的标题
    for row in csv_reader:
        p2E[float(row[0])] = float(row[1])
        #p2E.append(row)

min_loss = 10000000
min_start = 0
min_end = 0
count = 0
start_out = 130
end_out = 370
V = 500*3.1416*25
for start_in in range(1, 250):  #开始入油时间
    for end_in in range(start_in+200, 501):  #停止入油时间
        p = {0: 100}  # 压强p(t)
        E = {0: 2171.4}  # 模量E(t)
        for time in range(1, 1001):
            p0 = p[time-1]
            if time > start_in and time < end_in:  #正在进油
                Q = 0.85*3.1416*0.49*math.sqrt(2*(160-p0)/0.8713)
                p0 += Q*0.01*E[time-1]/V  #入油导致压升
            if time > start_out and time < end_out:  #正在出油
                if (time-start_out) < 20:
                    q = time-start_out
                elif (time-start_out) > 220:
                    q = end_out-time
                else:
                    q = 20
                p0 -= q*0.01*E[time-1]/V  #出油导致压降
            p[time] = p0
            E[time] = p2E[int(p0*10+0.5)/10]
        loss = 0
        for t in p:
            loss += (p[t]-100)*(p[t]-100)
        count += 1
        #print(str(count)+":done!")
        if loss < min_loss and abs(p[900]-100) < 0.01:
            min_loss = loss
            print(min_loss)
            min_start = start_in
            min_end = end_in
```

6.2 第一题后一问代码

```
import math
import csv
p2E = {}  #由p查表得到E
with open("") as csvfile:  #绝对路径名已省去
    csv_reader = csv.reader(csvfile)
    header = next(csv_reader)  # 读取第一行每一列的标题
    for row in csv_reader:
        p2E[float(row[0])] = float(row[1])
        #p2E.append(row)

min_loss = 100000000
graph = []
```

```
min_c = 0
min_t = 0
min_s = 0
count = 0
V = 500*3.1416*25
for C in range(1, 40):#遍历单向阀在每一周期内的开启时间，以0.1ms为单位
    print(C)
    for T in range(7*C, 11*C):  #遍历单向阀周期，以0.1ms为单位
        for S in range(0, 1):
            p = {0: 100}  # 压强p(t)
            E = {0: 2171.4}  # 模量E(t)
            start = S
            end = S+C
            all_in = []
            while end < 100000:
                one_in = [start, end]
                all_in.append(one_in)
                start += T
                end += T
            time_in = []  #各个时刻是否在进油
            for i in range(100001):
                time_in.append(0)
            for one_in in all_in:
                start = one_in[0]
                end = one_in[1]
                for t in range(start, end):
                    time_in[t] = 1

            start = 0
            end = 24
            all_out = []
            while end < 100000:
                one_out = [start, end]
                all_out.append(one_out)
                start += 1000
                end += 1000
            time_out = []  # 各个时刻是否在进油
            for i in range(100001):
                time_out.append(0)
            for one_out in all_out:
                start = one_out[0]
                end = one_out[1]
                time_out[start+1] = 2
                time_out[end-1] = 2
                for t in range(start+2, end-1):
                    time_out[t] = 1
            judge = False
            for time in range(1, 100001):
                p0 = p[time-1]
                if p0 > 160:
                    judge = True
                    break
                if time_in[time] == 1:  #正在进油
                    Q = 0.85*3.1416*0.49*math.sqrt(2*(160-p0)/0.8713)
                    p0 += Q*0.01*E[time-1]/V
                if time_out[time] != 0:  #正在出油
                    if time_out[time]==1:
                        q = 20
                    else:
                        q = 10
                    p0 -= q*0.01*E[time-1]/V
                p[time] = p0
                E[time] = p2E[int(p0*10+0.5)/10]
            #print(p[2500])
            loss = 0
            if judge:
                continue
            for t in p:
                loss += (p[t]-100-0.0005*t)*(p[t]-100-0.0005*t)
```

```
            #更改:0.0005对应10s提升, 0.001对应5s提升, 0.0025对应2s提升
            #print(loss)
            count += 1
            if loss < min_loss and abs(p[99999]-150) < 0.1:
                min_loss = loss
                #print(str(count)+":done!")
                print(min_loss)
                print(C, T, S)
                min_c = C
                min_t = T
                min_s = S
```

6.3 第二题代码

```
import matplotlib.pyplot as plt
import math
import csv

theta2h = []  #由theta查表得到h
with open("") as csvfile:  #绝对路径名已省去
    csv_reader = csv.reader(csvfile)
    header = next(csv_reader)  # 读取第一行每一列的标题
    for row in csv_reader:
        theta2h.append(float(row[1]))

p2rho = []  #由p查表得到rho
with open("") as csvfile:  #绝对路径名已省去
    csv_reader = csv.reader(csvfile)
    header = next(csv_reader)  # 读取第一行每一列的标题
    for row in csv_reader:
        p2rho.append(float(row[2]))

p2E = {}  #由p查表得到E
with open("") as csvfile:  #绝对路径名已省去
    csv_reader = csv.reader(csvfile)
    header = next(csv_reader)  # 读取第一行每一列的标题
    for row in csv_reader:
        p2E[float(row[0])] = float(row[1])

t2A = {}  #由t查表得到A
with open("") as csvfile:  #绝对路径名已省去
    csv_reader = csv.reader(csvfile)
    header = next(csv_reader)  # 读取第一行每一列的标题
    for row in csv_reader:
        t2A[float(row[0])] = float(row[1])

out_p = [0.1, 0.5, 1, 2, 3, 4, 5, 10]  #喷油嘴外界压力
V = 500*3.1416*25
min_T_ls = []
min_phase_ls = []
min_loss = 10000000

def Qn(pressure,w,n,phase):  #入油流速
    """w is the change on theta after delta t"""
    theta = w*(n+phase)
    left = int(100*(theta%6.28)+0.000001)
    leftpre = int(100*((theta-w)%6.28)+0.000001)
    right = int(10*pressure+0.000001)
    length = theta2h[left]
    rholeft0 = 0.80455*114.7583/(3.1415926*(7.239-length)*2.5*2.5+20)
    rhoright = p2rho[right]
    if rholeft0 <= rhoright:
        return 0
    else:
        deltalength = theta2h[left]-theta2h[leftpre]
        Qn_delta_t = deltalength*3.1415926*2.5*2.5
```

```
        if Qn_delta_t > 0:
            return Qn_delta_t
        else:
            return 0

def qn_delta_p(inpressure,time,outpressure,phase):  #出油流速
    time = time%100
    time -= 1
    if time*100 < phase or time*100 > phase+245:
        return 0
    delta_p = 0
    for t in range(time*100, (time+1)*100):
        if t < phase or t > phase+245:
            continue
        elif t > phase+45 and t < phase+200:
            right = int(10 * inpressure + 0.000001)
            rhoright = p2rho[right]
            q = 1.8505*math.sqrt((inpressure-outpressure)/rhoright)
            delta_delta_p = 0.01*q*p2E[int(inpressure*10+0.5)/10]/V
            inpressure -= delta_delta_p
            delta_p += delta_delta_p
        else:
            right = int(10 * inpressure + 0.000001)
            rhoright = p2rho[right]
            q = t2A[(t-phase)/100]*
                math.sqrt((inpressure-outpressure)/rhoright)
            delta_delta_p = 0.01 * q *
                p2E[int(inpressure * 10 + 0.5) / 10] / V
            inpressure -= delta_delta_p
            delta_p += delta_delta_p
    return delta_p


for outpressure in out_p:
    print(outpressure)
    min_T = 0
    min_phase = 0
    min_loss = 10000000
    for phase in range(80, 121):  #遍历初始相位差, 单位1ms
        for T in range(2510, 2525):  #遍历凸轮周期, 单位0.1ms
            T = T/10
            #for phase
            w = 2*3.1415926/T
            p = {0: 100}  # 压强p(t)
            E = {0: 2171.4}  # 模量E(t)
            for time in range(1, int(T*100)):
                p0 = p[time-1]
                Q = Qn(p0, w, time, phase)
                q = qn_delta_p(p0, time, outpressure, 0)
                p0 = p0+Q*E[time-1]/V-q  #下一时刻的油管压力
                p[time] = p0
                E[time] = p2E[int(p0*10+0.5)/10]
            loss = 0
            for t in p:
                loss += (p[t]-100)*(p[t]-100)
            loss = math.sqrt(loss/int(T*100))
            #print(str(count)+":done!")
            #print(loss)
            if loss < min_loss:
                min_loss = loss
                print(min_loss, T, phase)
                min_T = T
                min_phase = phase
    min_T_ls.append(min_T)
    min_phase_ls.append(min_phase)
```

6.4 第三题前一问代码

Command invalid in math modeCommand invalid in math mode

```python
import math
import csv

theta2h = []
with open("cam.csv") as csvfile:
    csv_reader = csv.reader(csvfile)  # 使用csv.reader读取csvfile中的文件
    header = next(csv_reader)  # 读取第一行每一列的标题
    for row in csv_reader:
        theta2h.append(float(row[1]))

p2rho = []
with open("p_E_rho_afterinterp.csv") as csvfile:
    csv_reader = csv.reader(csvfile)  # 使用csv.reader读取csvfile中的文件
    header = next(csv_reader)  # 读取第一行每一列的标题
    for row in csv_reader:
        p2rho.append(float(row[2]))

p2E = {}
with open("p_E_rho_afterinterp.csv") as csvfile:
    csv_reader = csv.reader(csvfile)  # 使用csv.reader读取csvfile中的文件
    header = next(csv_reader)  # 读取第一行每一列的标题
    for row in csv_reader:
        p2E[float(row[0])] = float(row[1])

t2A = {}
with open("sprayer_cAsqrt2.csv") as csvfile:
    csv_reader = csv.reader(csvfile)  # 使用csv.reader读取csvfile中的文件
    header = next(csv_reader)  # 读取第一行每一列的标题
    for row in csv_reader:
        t2A[float(row[0])] = float(row[1])



out_p = [4]
V = 500*3.1416*25
min_T_ls = []
min_phase0_ls = []
min_phase_ls = []
min_loss = 10000000

def Qn(pressure,w,n,phase):  #入油流速
    """w is the change on theta after delta t"""
    theta = w*(n+phase)
    left = int(100*(theta%6.28)+0.000001)
    leftpre = int(100*((theta-w)%6.28)+0.000001)
    right = int(10*pressure+0.000001)
    length = theta2h[left]
    rholeft0 = 0.80455*114.7583/(3.1415926*(7.239-length)*2.5*2.5+20)
    rhoright = p2rho[right]
    if rholeft0 <= rhoright:
        return 0
    else:
        deltalength = theta2h[left]-theta2h[leftpre]
        Qn_delta_t = deltalength*3.1415926*2.5*2.5
        if Qn_delta_t > 0:
            return Qn_delta_t
        else:
            return 0

def qn_delta_p(inpressure,time,outpressure,phase):  #出油流速
    time = time%100
    time -= 1
    if time*100 < phase or time*100 > phase+245:
        return 0
    delta_p = 0
```

```
    for t in range(time*100, (time+1)*100):
        if t < phase or t > phase+245:
            continue
        elif t > phase+45 and t < phase+200:
            right = int(10 * inpressure + 0.000001)
            rhoright = p2rho[right]
            q = 1.8505*math.sqrt((inpressure-outpressure)/rhoright)
            delta_delta_p = 0.01*q*p2E[int(inpressure*10+0.5)/10]/V
            inpressure -= delta_delta_p
            delta_p += delta_delta_p
        else:
            right = int(10 * inpressure + 0.000001)
            rhoright = p2rho[right]
            q = t2A[(t-phase)/100]*math.sqrt((inpressure-outpressure)/rhoright)
            delta_delta_p = 0.01 * q * p2E[int(inpressure * 10 + 0.5) / 10] / V
            inpressure -= delta_delta_p
            delta_p += delta_delta_p
    return delta_p

for outpressure in out_p:
        min_T = 0
        min_phase = 0
        min_phase0 = 0
        print(outpressure)
        min_loss = 10000000
        for phase0 in range(50,60):
            print('phase0:',phase0)
            for phase in range(49,52):
                for T in range(1220,1223): #以1ms为单位
                    T = T/10
                    #for phase
                    w = 2*3.1415926/T
                    p = {0: 100}  # 压强p(t)
                    E = {0: 2171.4}  # 模量E(t)
                    try:
                        for time in range(1, int(T*100)):
                            p0 = p[time-1]
                            Q = Qn(p0, w, time, phase0)
                            q = qn_delta_p(p0, time, outpressure, 0)
                            q2= qn_delta_p(p0, time, outpressure, phase*100)
                            p0 = p0+Q*E[time-1]/V-q-q2  #原压强加上进油增压减去两嘴降压
                            p[time] = p0
                            E[time] = p2E[int(p0*10+0.5)/10]
                        loss = 0
                        for t in p:
                            loss += (p[t]-100)*(p[t]-100)
                        loss = math.sqrt(loss/int(T*100))
                        #print(str(count)+":done!")
                        #print(loss)
                        if loss < min_loss:
                            min_loss = loss
                            print(min_loss, T, phase0,phase)
                            min_T = T
                            min_phase = phase
                            min_phase0 = phase0
                    except:
                        pass
        min_phase0_ls.append(min_phase0)
        min_T_ls.append(min_T)
        min_phase_ls.append(min_phase)
```

126

6.5 第三题后一问代码

Command invalid in math modeCommand invalid in math mode

```
import math
import csv

theta2h = []
with open(”cam.csv”) as csvfile:
    csv_reader = csv.reader(csvfile)  # 使用csv.reader读取csvfile中的文件
    header = next(csv_reader)  # 读取第一行每一列的标题
    for row in csv_reader:
        theta2h.append(float(row[1]))

p2rho = []
with open(”p_E_rho_afterinterp.csv”) as csvfile:
    csv_reader = csv.reader(csvfile)  # 使用csv.reader读取csvfile中的文件
    header = next(csv_reader)  # 读取第一行每一列的标题
    for row in csv_reader:
        p2rho.append(float(row[2]))

p2E = {}
with open(”p_E_rho_afterinterp.csv”) as csvfile:
    csv_reader = csv.reader(csvfile)  # 使用csv.reader读取csvfile中的文件
    header = next(csv_reader)  # 读取第一行每一列的标题
    for row in csv_reader:
        p2E[float(row[0])] = float(row[1])

t2A = {}
with open(”sprayer_cAsqrt2.csv”) as csvfile:
    csv_reader = csv.reader(csvfile)  # 使用csv.reader读取csvfile中的文件
    header = next(csv_reader)  # 读取第一行每一列的标题
    for row in csv_reader:
        t2A[float(row[0])] = float(row[1])



out_p = [1]
V = 500*3.1416*25
min_T_ls = []
min_phase0_ls = []
min_phase_ls = []
min_phaseout_ls = []#减压阀
min_opentime_ls = []#减压阀
min_loss = 10000000

def Qn(pressure,w,n,phase):  #入油流速
    ”””w is the change on theta after delta t”””
    theta = w*(n+phase)
    left = int(100*(theta%6.28)+0.000001)
    leftpre = int(100*((theta-w)%6.28)+0.000001)
    right = int(10*pressure+0.000001)
    length = theta2h[left]
    rholeft0 = 0.80455*114.7583/(3.1415926*(7.239-length)*2.5*2.5+20)
    rhoright = p2rho[right]
    if rholeft0 <= rhoright:
        return 0
    else:
        deltalength = theta2h[left]-theta2h[leftpre]
        Qn_delta_t = deltalength*3.1415926*2.5*2.5
        if Qn_delta_t > 0:
            return Qn_delta_t
        else:
            return 0

def qn(inpressure,time,outpressure,phase):  #出油流速
    time = (time-phase)%10000
    if time > 245:
        return 0
    elif time > 45 and time < 200:
        right = int(10 * inpressure + 0.000001)
        rhoright = p2rho[right]
```

```
        return 1.8505*math.sqrt((inpressure-outpressure)/rhoright)
    else:
        right = int(10 * inpressure + 0.000001)
        rhoright = p2rho[right]
        return t2A[time/100]*math.sqrt((inpressure-outpressure)/rhoright)

def qn3(inpressure,time, phase, T0):  #出油流速
    time = time%10000
    if time > T0 + phase or time < phase:
        return 0
    else:
        right = int(10 * inpressure + 0.000001)
        rhoright = p2rho[right]
        return 1.8505 * math.sqrt((inpressure-0.5)/rhoright)

for outpressure in out_p:
        min_T = 0
        min_phase = 0
        min_phase0 = 0
        min_phaseout=0
        min_opentime=0
        print(outpressure)
        min_loss = 10000000
        for phase0 in range(6174,6177):
            print('phase0:',phase0)
            for phase in range(1715,1718):
                print('phase:',phase)
                for phaseout in range(2935,2939):
                    for opentime in range(64,68):
                        for T in range(10000,10001):
                            T = T*1#以0.01ms为单位
                            w = 2*3.1415926/T
                            p = {0: 100}  # 压强p(t)
                            E = {0: 2171.4}  # 模量E(t)
                            try:
                                for time in range(1, int(T*10)):
                                    p0 = p[time-1]
                                    Q = Qn(p0, w, time, phase0)
                                    q = qn(p0, time, outpressure, 0)
                                    q2= qn(p0, time, outpressure, phase)
                                    q3= qn3(p0,time,phaseout,opentime)
                                    p0 = p0+Q*E[time-1]/V-q*0.01*E[time-1]/V-q2*0.01
                                    *E[time-1]/V-q3*0.01*E[time-1]/V
                                    #原压强加上进油增加压强减去两嘴和减压阀压强
                                    p[time] = p0
                                    E[time] = p2E[int(p0*10+0.5)/10]
                                loss = 0
                                for t in p:
                                    loss += (p[t]-100)*(p[t]-100)
                                loss = math.sqrt(loss/int(T*10))
                                if loss < min_loss:
                                    min_loss = loss
                                    print(min_loss, T, phase0,phase,phaseout,opentime)
                                    min_T = T
                                    min_phase = phase
                                    min_phase0 = phase0
                                    min_phaseout = phaseout
                                    min_opentime=opentime
                            except:
                                pass
        min_phase0_ls.append(min_phase0)
        min_T_ls.append(min_T)
        min_phase_ls.append(min_phase)
        min_phaseout_ls.append(min_phaseout)
        min_opentime_ls.append(min_opentime)
```