

PID 控制在医学麻醉过程血压控制中的应用

秦华谦*, 李中华老师†

中山大学 智能工程学院, 深圳 518000

【摘要】 本文基于 PID 控制系统开发了一个系统, 旨在自动测量和控制多种生命参数, 以保持其在适当范围内。其目标是提高手术中麻醉师的安全性。为了进一步优化系统性能, 我们进行了一项专门的研究, 通过对 k_p 、 k_i 、 k_d 三个参数的调整, 探究 PID 系统内不同参数对系统响应的影响。该系统实现了自动调节麻醉深度, 以减小波动和冲击响应。实验结果表明, 引入设计的控制器后, 血压控制系统对于给定输入信号的超调量减小, 调节时间缩短。在阶跃输入信号下的稳态误差为零, 并在阶跃扰动输入信号下能保持稳态误差在一定范围内, 最终实现稳态输出为零。

【关键词】 PID 控制, 麻醉, 血压控制

Application of PID control in blood pressure control in medical anesthesia process

Huaqian Chin, Prof. Li Zhonghua

School of Intelligent Engineering, Sun Yat-sen University, Shenzhen 518000, China

Abstract: This article has developed a system based on the PID control system, aimed at automatically measuring and controlling various vital parameters to maintain them within an appropriate range. The goal is to enhance the safety of anesthesiologists during surgery. To further optimize system performance, we conducted a dedicated study, investigating the impact of different parameters in the PID system on system response by adjusting the k_p , k_i , and k_d parameters. The system achieves automatic adjustment of anesthetic depth to minimize fluctuations and impact responses. Experimental results show that after introducing the designed controller, the overshoot of the blood pressure control system for a given input signal decreases, and the adjustment time shortens. The steady-state error under step input signals is zero, and under step disturbance input signals, it can maintain the steady-state error within a certain range, ultimately achieving a steady-state output of zero.

Key Words: PID control, Anaesthetization, Blood Pressure Pontrol

1 引言

手术中麻醉师需监测多种生命参数, 如: 麻醉深度、血压、心率、体温、血氧、呼气中二氧化碳浓度等, 并将它们控制在适当的范围内。能够自动测量、控制某些生命参数, 能够提高受术者的安

全。我们希望开发一个自动调节麻醉深度的系统, 病人的安全是最终目标。

许多麻醉师将平均动脉压作为麻醉深度最可靠的度量。根据临床经验和麻醉师所遵从的程序, 被控变量确定为平均动脉压 (MAP)。

本文主要是运用 PID 控制器的设计方法, 对麻醉过程中的血压控制模型进行控制器的设计, 通过对 k_p 、 k_i 、 k_d 三个参数的调整, 探究 PID 系统内不同参数对系统响应的影响。并且针对 PI, PD, PID 三种控制器在有无扰动两种情况下进行性能分析

实验时间: 2023-05-20

报告时间: 2023-5-30

† 指导教师: 李中华

*学号: 21312683

*E-mail: qinhq5@mail2.sysu.edu.cn

2 实验原理

2.1 PID 控制原理

PID 控制是一种常用的反馈控制算法，用于调节和稳定系统的输出。它的名称代表比例（Proportional）、积分（Integral）和微分（Derivative），分别表示控制器的三个主要组成部分。

比例控制（Proportional Control）是 PID 控制的基本部分之一。它通过将当前误差与一个比例常数相乘，产生一个控制量，用于调节系统输出。比例控制的作用是使系统对误差做出直接的响应，但它无法解决误差积累和系统超调等问题。

积分控制（Integral Control）是 PID 控制的第二个组成部分。它将误差的积分值与一个积分常数相乘，产生一个控制量。积分控制的作用是消除稳态误差，即系统输出与期望输出之间的差异。通过积分控制，系统可以逐渐减小并消除持续的误差。

微分控制（Derivative Control）是 PID 控制的最后一个组成部分。它通过将误差的变化率与一个微分常数相乘，产生一个控制量。微分控制的作用是抑制系统的超调和振荡，并提高系统的响应速度。通过微分控制，系统可以更快地调整输出，以适应误差变化的速率。

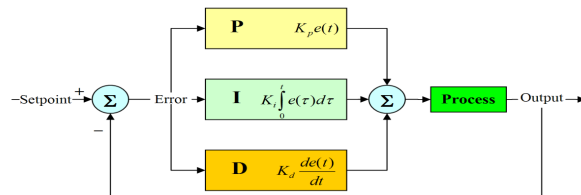
PID 控制器将比例、积分和微分控制相结合，通过对误差的不同方面进行调节，实现对系统输出的精确控制。PID 控制器的输出值由三个控制分量的加权和得出，即：

输出 = 比例控制 + 积分控制 + 微分控制

通过调整比例常数、积分常数和微分常数，可以对 PID 控制器进行参数调优，以满足不同系统的需求。调优过程通常包括实验、观察和调整，以获得最佳的控制效果。

总而言之，PID 控制是一种基于反馈的控制算法，通过比例、积分和微分控制三个部分的相互作用，实现对系统输出的精确调节。它广泛应用于自动控制领域，用于稳定系统、消除误差和提高响应速度。

2.2 系统设计流程图



2.3 系统参数介绍

PID 控制器的比例单元 (P)、积分单元 (I) 和微分单元 (D) 分别对应目前误差、过去累计误差及未来误差。借由调整 PID 控制器的三个参数，可以调整控制系统，设法满足设计需求。

PID 控制器（比例-积分-微分控制器）的参数选择取决于具体的应用场景和系统的动态特性。通常，这些参数需要通过一些调参方法（例如 Ziegler-Nichols 方法）进行调整，以达到系统的稳定性和性能需求。

PID 控制器的三个参数分别为：

1. P（比例）：该参数控制系统的响应速度。P 参数越大，系统对错误的反应越快，但如果过大，可能会导致系统不稳定。

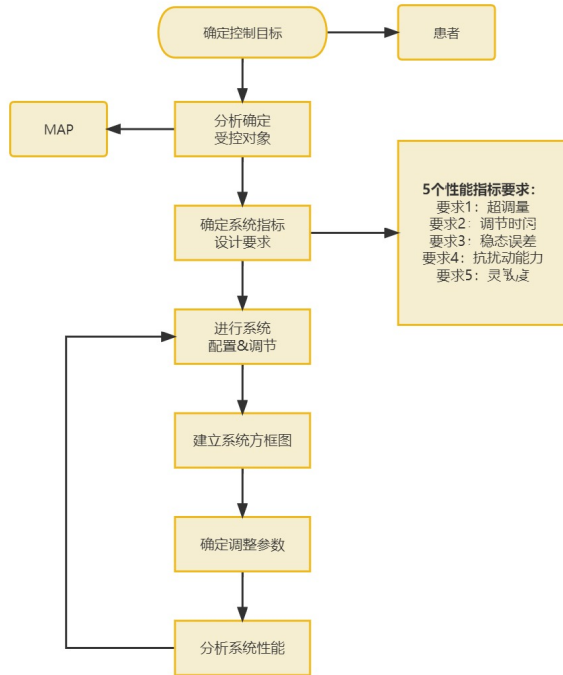
2. I（积分）：该参数控制系统是否能达到设定点（或称为参考点）。I 参数可以消除系统的静差，但如果过大，可能会导致系统超调和震荡。

3. D（微分）：该参数控制系统的稳定性。D 参数可以预测系统的行为，减小系统的超调，但如果过大，可能会引入噪声，使系统不稳定。

在实际应用中，PID 参数的调整是一项复杂的工程任务，需要根据系统的具体特性和性能需求进行优化。有时，人们也使用一些自适应的 PID 控制算法，如模糊 PID，遗传算法等，以自动地优化 PID 参数。

3 实验过程

3.1 系统设计流程图



3.2 控制目标与实验要求

3.2.1 麻醉过程中的血压控制系统控制目标

- 控制对象: 需麻醉的患者
- 控制任务: 将患者 MAP 调节到任意预期设定的水平, 并在存在干扰信号的情况下将 MAP 维持在预期设定的水平。
- 受控变量: MAP(平均动脉压)
- 系统输入: MAP 的期望值
- 干扰输入: 手术干扰, 噪声干扰
- 误差量: 稳态误差 = 输入 - 输出
- 系统输出: 实际的 MAP 量

3.2.2 该控制系统的实验设计要求

1. 取 $N(s)=0$, $Td(s)=0$

- 基于 PD、PI 控制器对系统进行 Matlab 仿真。
 - 在 PD 或 PI 控制器中, 固定一个参数, 调节另外一个参数, 观察输出结果的变化

- 基于 PID 控制器对系统进行 Matlab 仿真
 - 改变三个参数的值, 看在不同的组合下, 观察系统的输出的变化

2. 取 $N(s)=0$, $Td(s)=50/s$

- 重复上面的仿真过程

3.3 系统设计

其中, $R(s)$ 和 $Y(s)$ 分别为期望的平均动脉压变化和实际的平均动脉压变化, 两者的偏差被控制器用于确定对泵的阀门给定值, 泵/蒸发器给患者输送麻醉蒸汽。而 $Td(s)$ 和 $N(s)$ 分别为手术过程中的扰动和噪声, $G_c(s)$ 为待加入的控制器传递函数。另外, $G(s)$ 是被控对象, 其传递函数为

$$G(s) = 1/(s + p)^2 \quad (1)$$

传感器 $H(s)$ 传递函数为

$$H(s) = 1$$

泵的传递函数为 $G_p(s)$, 为

$$G_p(s) = \frac{U(s)}{V(s)} = \frac{1}{s}$$

3.4 传递函数设计与解析

通过对信号方框图的分析, 我将 R 与 Td 与 N 分别作为输入, 重绘信号方框图

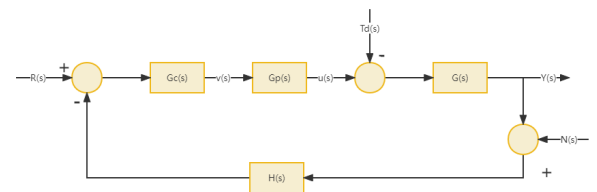


图1 $R(s)$ 作为输入

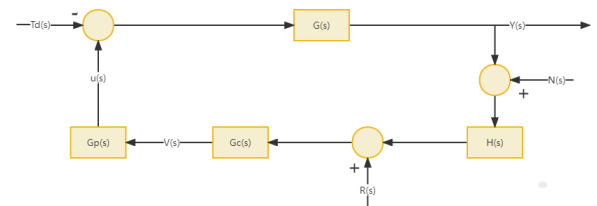
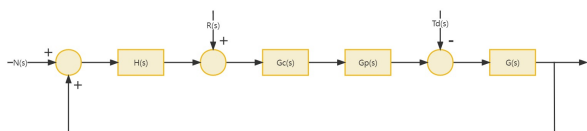


图2 $Td(s)$ 作为输入

图3 $N(s)$ 作为输入

根据三幅信号框图，写出闭环传递函数如下：

以 $R(s)$ 作为输入

$$\Phi_r(s) = \frac{G_c(s) * G_p(s) * G(s)}{1 + H(s) * G_c(s) * G_p(s) * G(s)}$$

以 $R(s)$ 作为输入

$$\Phi_{Td}(s) = \frac{G(s)}{1 + H(s) * G_c(s) * G_p(s) * G(s)}$$

以 $R(s)$ 作为输入

$$\Phi_n(s) = \frac{H(s) * G_c(s) * G_p(s) * G(s)}{1 + H(s) * G_c(s) * G_p(s) * G(s)}$$

3.5 PID 系统传递函数

$$G_c(s) = \frac{K_d s^2 + K_p s + K_i}{s}$$

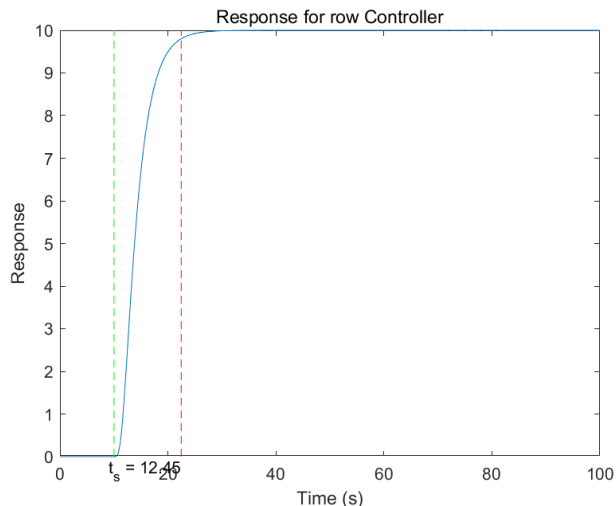
3.6 仿真细节与结果展示

3.6.1 原系统

取 $N(s)=0$, $Td(s)=0$, $R(s)=10/s$

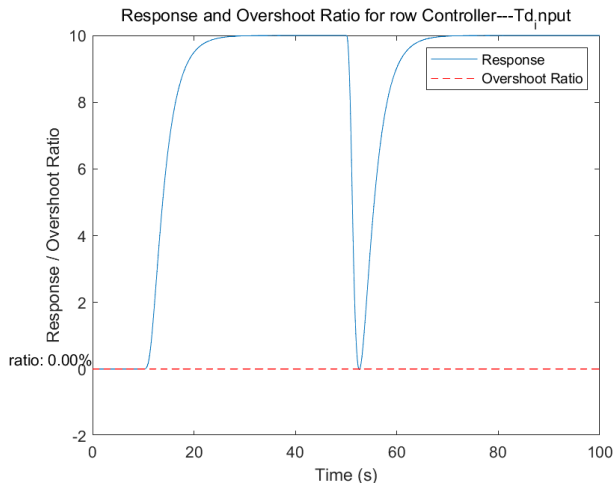
可以看到先对未加入 pid 控制器的系统进行响应分析，通过 matlab 运行程序，作图，得到原系统的阶跃输入

当到达绿线时，系统给予了一个 $R(s) = 10/s$ 的阶跃输入，可以看到几乎没有超调量，到达红线时输出进入 5% 的误差带，调节时间 $t_s = 12$



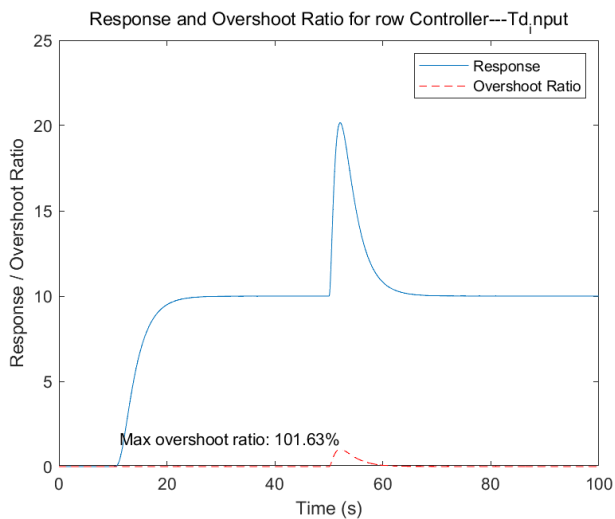
取 $N(s)=0$, $Td(s)=-50/s$, $R(s)=10/s$

测试系统在阶跃扰动的情况下的性能，在 $t=0$ 的时候给予一个 $R(s) = 10/s$ 的输入，在 $t=50$ 的时候给予一个 $Td(s) = -50/s$ 的阶跃扰动，观察超调量，作图如下



可以看到，超调量几乎为 0，调节时间也比较快，但是稳定性很差，响应范围过大。

为了方便观察， $Td(s)$ 取为正向输入的 $50/s$ ，作图，观察。



响应范围过大，接近 100%，无法满足要求。需要考虑加入控制器进行调节。

3.6.2 PI 控制器

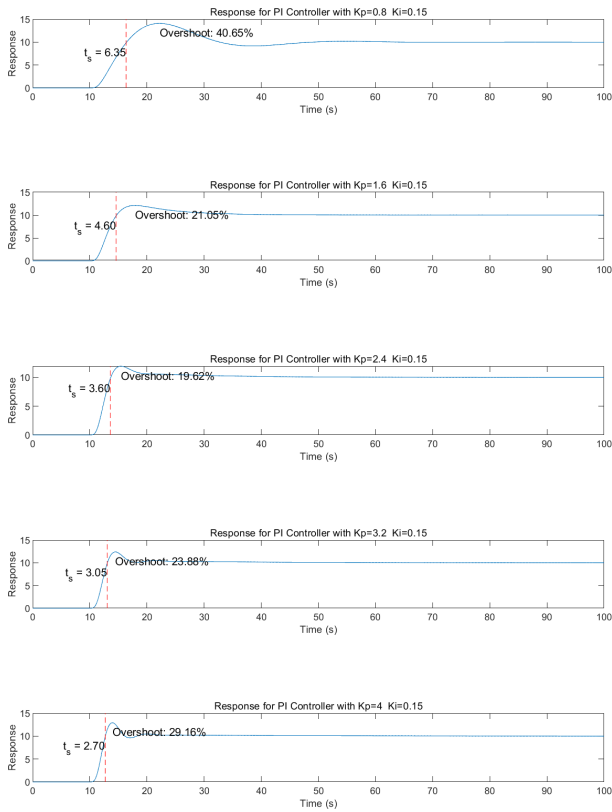
参数： $R = 10$ ($t > 10$); $N(s) = 0$; $Td(s) = 0$;

在这个模块中，将 PI 控制器接入了系统的 $G_c(s)$ 的位置

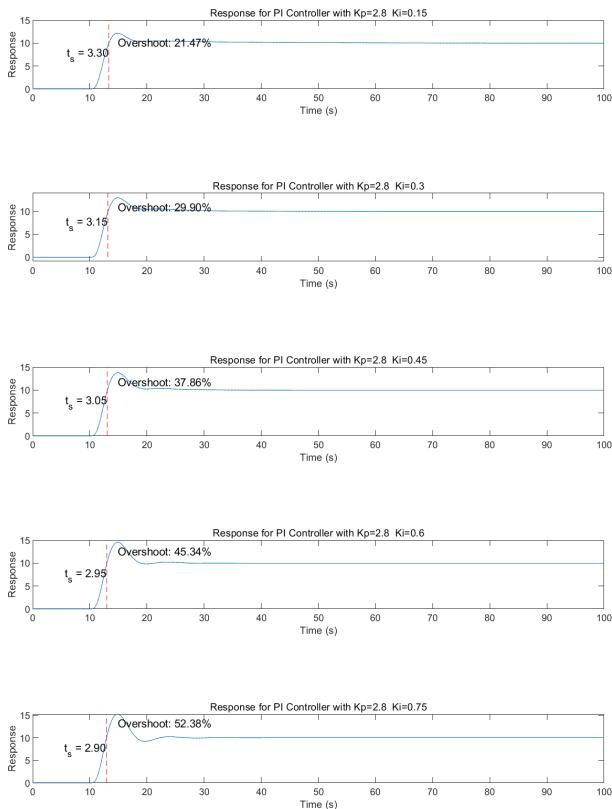
实现了对两个参数 (K_p & K_i) 进行了两组分析

1. 保持 K_i 不变，调整 K_p 的值——以 0.8 为

步长, 在 $[0.8, 4]$ 的范围内生成 5 张系统时域图



2. 保持 K_p 不变, 调整 K_i 的值——以 0.15 为步长, 在 $[0.15, 0.75]$ 的范围内生成 5 张系统时域图



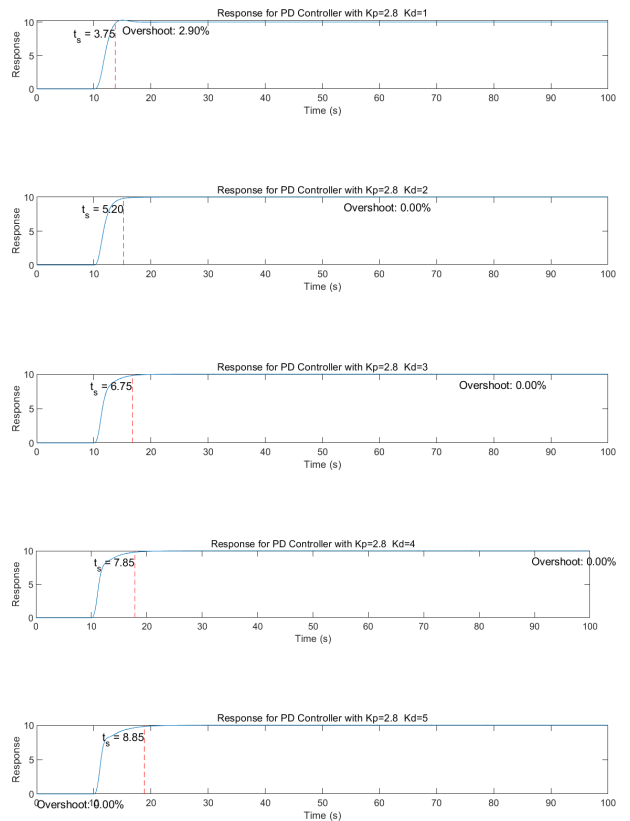
3.6.3 PD 控制器

参数: $R = 10$ ($t > 10$); $N(s) = 0$; $T_d(s) = 0$;

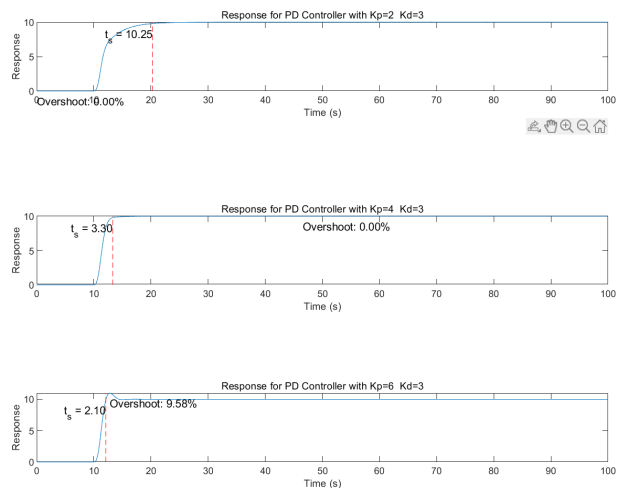
在这个模块中, 将 PD 控制器接入了系统的 $G_c(s)$ 的位置

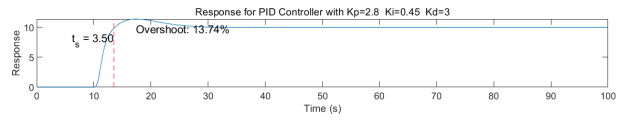
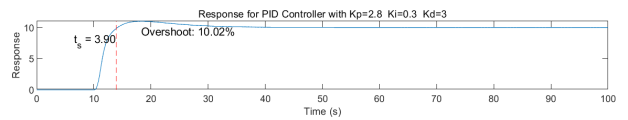
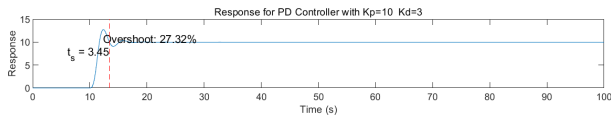
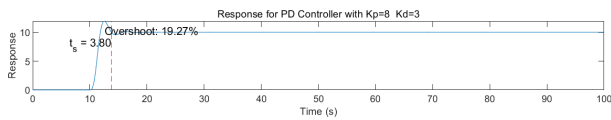
实现了对两个参数 (K_p & K_d) 进行了两组分析

1. 保持 K_p 不变, 调整 K_d 的值——以 1 为步长, 在 $[1, 5]$ 的范围内生成 5 张系统时域图



2. 保持 K_d 不变, 调整 K_p 的值——以 2 为步长, 在 $[2, 10]$ 的范围内生成 5 张系统时域图





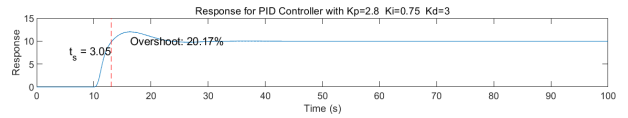
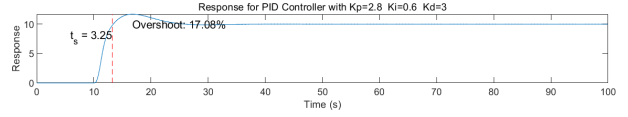
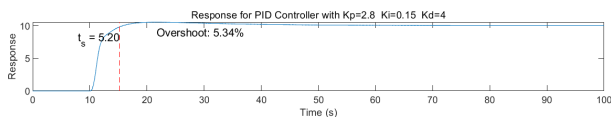
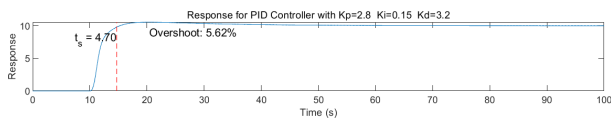
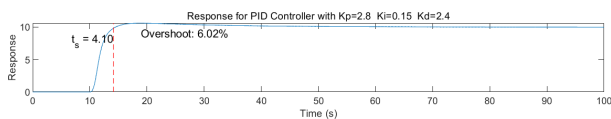
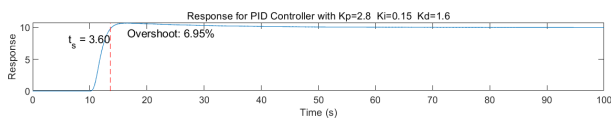
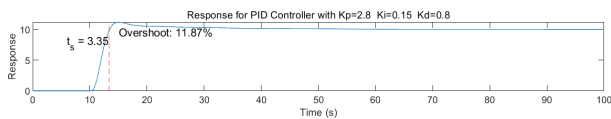
3.6.4 PID 控制器

参数: $R = 10$ ($t > 10$); $N(s) = 0$; $T_d(s) = 0$;

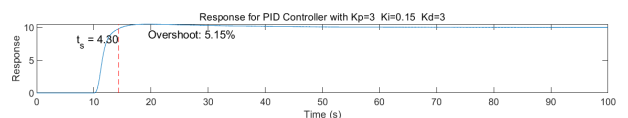
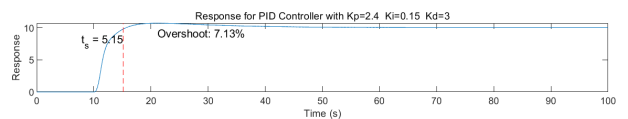
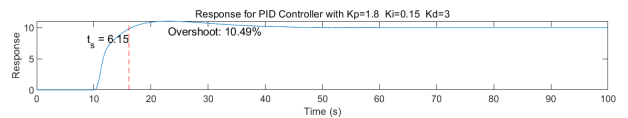
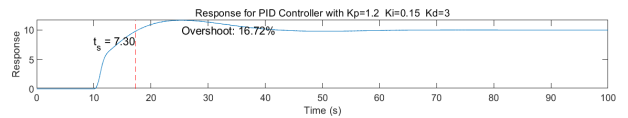
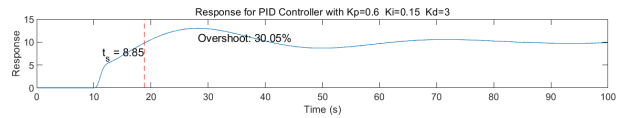
在这个模块中, 将 PID 控制器接入了系统的 $G_c(s)$ 的位置

实现了对三个参数 (K_p & K_i & K_d) 进行了两组分析

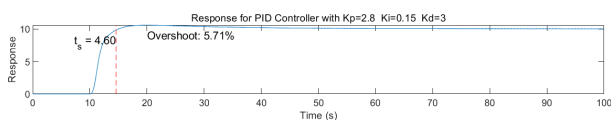
1. 保持 K_p & K_i 不变, 调整 K_d 的值———以 0.8 为步长, 在 $[0.8, 4]$ 的范围内生成 5 张系统时域图



3. 保持 K_i & K_d 不变, 调整 K_p 的值———以 0.6 为步长, 在 $[0.6, 3]$ 的范围内生成 5 张系统时域图



2. 保持 K_p & K_d 不变, 调整 K_i 的值———以 0.15 为步长, 在 $[0.15, 0.75]$ 的范围内生成 5 张系统时域图



3.6.5 阶跃扰动响应下的 PID 控制器

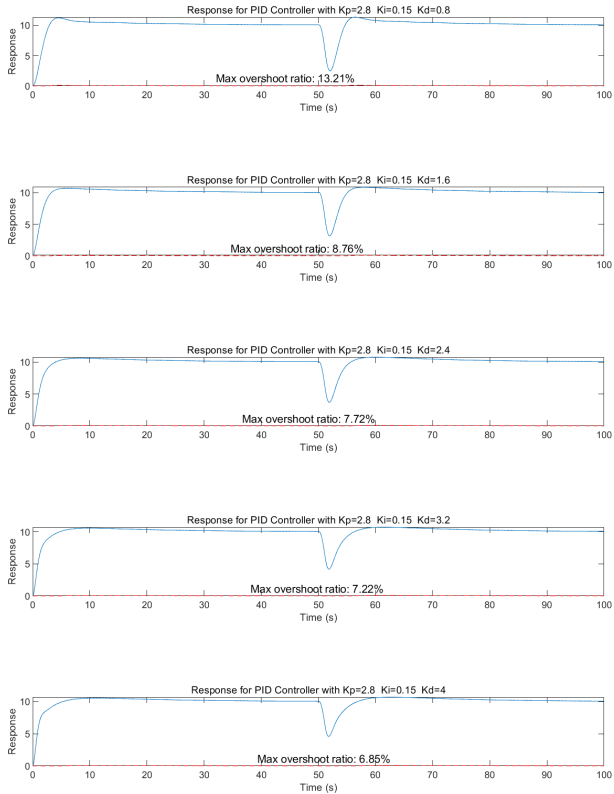
参数: $R = 10$ ($t > 10$); $N(s) = 0$; $T_d(s) = -50/s$ ($t > 50$);

在这个模块中, 将 PID 控制器接入了系统的 $G_c(s)$ 的位置

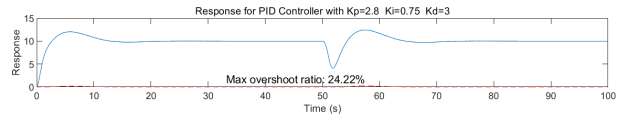
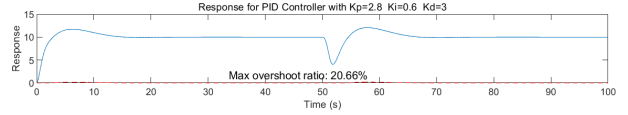
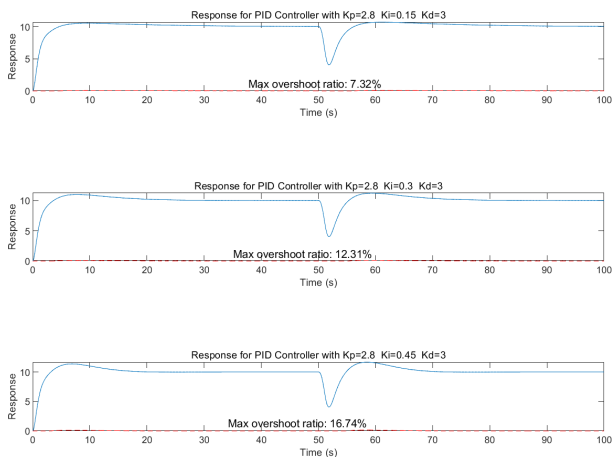
在 $t = 0$ 的时候系统对 R 阶跃信号产生响应, 当趋于稳定以后, 对系统添加一个 $T_d(s) = -50/s$ 的阶跃扰动响应信号观察系统的变化, 同时与没有接入 PID 控制器的信号进行对比

实现了对三个参数 (K_p & K_i & K_d) 进行了两组分析

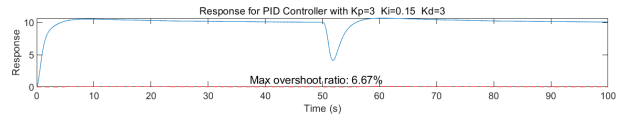
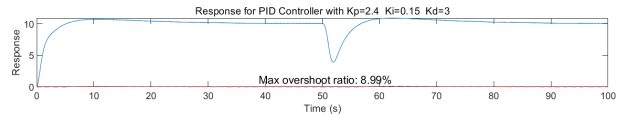
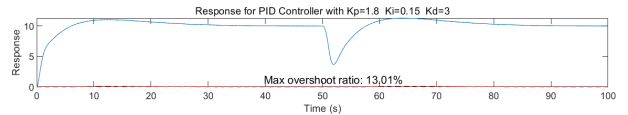
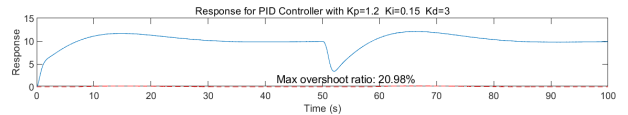
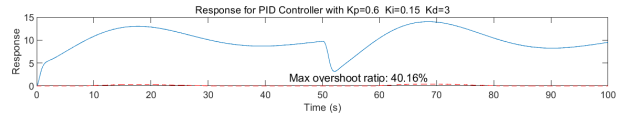
1. 保持 K_p & K_i 不变, 调整 K_d 的值——以 0.8 为步长, 在 $[0.8, 4]$ 的范围内生成 5 张系统时域图



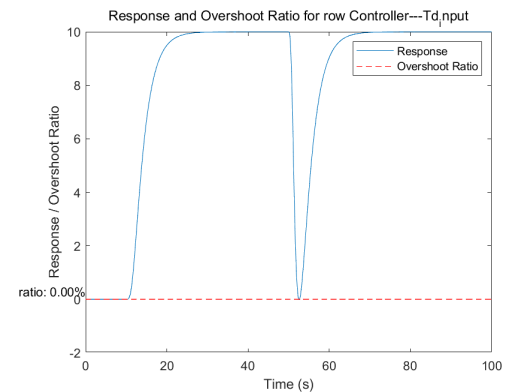
2. 保持 K_p & K_d 不变, 调整 K_i 的值——以 0.15 为步长, 在 $[0.15, 0.75]$ 的范围内生成 5 张系统时域图



3. 保持 K_i & K_d 不变, 调整 K_p 的值——以 0.6 为步长, 在 $[0.6, 3]$ 的范围内生成 5 张系统时域图



没有 PID 控制器的原系统响应为



可以看到, 当加入了 PID 系统以后, 整体的超调量有了明显的下降, 最重要的是响应范围过

大的问题有了很大的改善。这体现了通过加入 PID 控制器以后，系统的稳定性大大提高了。这对于手术过程中的病人麻醉状态的稳定有着至关重要的作用。

4 实验分析与结论

4.1 结论

利用控制变量法及相关资料，我在 PI/PD/PID 三种控制器的有无扰动两种情况下最终人工调整三个参数获得以下结论：

表 1 PID 参数对系统的影响

调整方式	上升时间	超调量	稳态误差	稳定性
↑ Kp	↓ 减少	↑ 增加	↓ 减少	↓ 变差
↑ Ki	↓ 小幅减少	↑ 增加	↓ 大幅减少	↓ 变差
↑ Kd	↓ 小幅减少	↓ 减少	→ 变动不大	↑ 变好

4.2 最优参数

人工调整有一种作法是先将 Ki 及 Kd 设为零，增加 Kp 一直到回路输出震荡为止，之后再再将 Kp 设定为“1/4 振幅衰减”（使系统第二次过冲量是第一次的 1/4）增益的一半，然后增加 Ki 直到一定时间后的稳态误差可被修正为止。不过若 Ki 可能会造成不稳定，最后若有需要，可以增加 Kd，并确认在负载变动后回路可以够快的回到其设定值，不过若 Kd 太会造成响应太快及过冲。一般而言快速反应的 PID 应该会有轻微的过冲，只是有些系统不允许过冲。因此需要将回授系统调整为过阻尼系统，而 Kp 比造成震荡 Kp 的一半还要小很多。^[1]

当然历史上 PID 的参数调试还有很多，例如齐格勒-尼克尔斯方法等，大部分现代的工业设备也实现了 PID 自动调试及最佳化软件。

写在最后

在完成自动控制原理课程作业的这个过程中，我体验到了丰富而深刻的学习过程，由衷地感谢所有帮助我、支持我的人。

首先，我要感谢李中华老师，您的热情教课和专业引导使我对 PID 控制和信号流图有了深入的理解和掌握。每次课堂上的讲解都是精心准备的，它们如照亮黑夜的明灯，指引我探索控制理论的广阔领域。您给予我们的这次作业机会，是我进一步理解自动控制原理的一个宝贵的实践场所。我

深深地感谢您的悉心教诲，也感谢您给我提供了这样一个展现所学知识的机会。

我也要特别感谢方桂安学长。虽然去年的作业要求与今年的不同，但学长去年在博客上分享的报告对我产生了深远的影响。在您的报告中，我看到了深入浅出的解析和清晰的逻辑思维。您的报告让我深受启发，也使我对 PID 控制和信号流图的学习有了更深的认识和理解。我把您的博客网址和引用都标在了文末，以表达我对您贡献的尊重和感谢。

回顾完成这份作业的整个过程，我深感欣喜。通过这次实践，我得以将所学的理论知识与实际操作相结合，实现了知识与实践的完美结合。我深深感受到自动控制原理的独特魅力和应用价值。

本着开源精神，我也将本次作业的 MATLAB 代码和完成报告所用的 L^AT_EX 源文件全部托管在了 GitHub 仓库中，以便老师和同学们进行浏览和学习。这不仅是我对自己工作的一种展示，也是希望能通过大家的交流和互动，一起探索学习，一起进步。

我诚挚地邀请每一位对此感兴趣的老师和同学，前往我的 GitHub 仓库查阅代码，提出您的宝贵建议和批评。任何形式的反馈和建议，无论是关于代码的优化，还是对于实验方法和结果的讨论，我都将热烈欢迎。通过大家的共同努力，我们可以一起推进学术的发展，也可以一起提升我们自己。

GitHub repository URL:

<https://github.com/pbcn2/PID-control-in-blood-pressure-control-in-anesthesia-process>

参考文献

- [1] ENDERFGA. Application of pid control in blood pressure control in medical anesthesia process[EB/OL]. <https://blog.enderfga.cn/2022/05/04/pid/>.