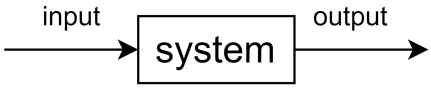
# 电机控制系统仿真实验

哈尔滨工程大学创梦之翼战队电控组

## 1 认识系统

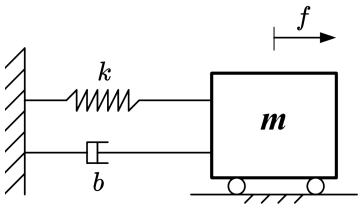
### 1.1 系统的基本概念

总的来说，系统是由相互联系的部分组成的一个更大更复杂的整体。为了直观，我们一般通过一个带有向内和向外箭头的方框表示系统：




其中指向方框的箭头表示系统的输入，由方框指向外的箭头表示系统的输出。系统的输出与输入并非相互独立，相反的，系统在输入信号的激励下会产生相应的输出响应。上图表示的系统只有单个的输入与输出，我们一般称这种系统为单输入单输出（single-input single-output SISO）系统。相应的，存在多个输入输出的系统被称之为多输入多输出（multi-input multi-output MIMO）系统：


以一个弹簧阻尼系统为例：




一个质量为 $m$ 的光滑物块通过弹性系数为 $k$ 的弹簧和阻尼系数为 $b$ 的阻尼与竖直墙面链接，施加给物体的外力 $f$ 正方向向右，物块位移正方形与外力 $f$ 相同。对物块进行受力分析，有：

$$f - kx - bv = ma$$

其中：

$$\begin{cases} v = \dot{x} \\ a = \ddot{x} \end{cases}$$

即：

$$f = m\ddot{x} + b\dot{x} + kx$$

不难发现，在这个系统中，力 $f$ 引起位移 $x$ 的变化。因此力 $f$ 便是系统的输入，系统在力 $f$ 的激励下会产生相应地输出响应，即位移 $x$ 的变化。

## 1.2 线性时不变系统

以下讨论均建立在系统是线性时不变系统的前提上，需先了解线性时不变系统（linear time-invariant LTI）的概念。

线性描述的是系统输入输出之间的关系是线性映射：如果输入信号 $x_1(t)$ 使系统产生输出响应 $y_1(t)$，而输入信号 $x_2(t)$ 使系统产生输出响应 $y_2(t)$，则输入信号 $\alpha_1 x_1(t)+\alpha_2 x_2(t)$ 使系统产生输出响应 $\alpha_1 y_1(t)+\alpha_2 y_2(t)$，即满足叠加定理。

时不变指系统输入延迟 $\tau$ 秒则其输出响应延迟 $\tau$ 秒：如果输入信号 $x(t)$ 使系统产生输出响应 $y(t)$，则输入信号 $x(t+\tau)$ 的输出响应为$y(t+\tau)$。

## 1.3 系统建模

系统建模即通过牛顿定律、基尔霍夫电压定律等物理定律建立描述动态系统的微分方程，上文中二阶系统的例子便是这个过程。对于一阶二阶线性微分方程而已，求其解析解尚且不是难事，但对于阶数更高的微分方程则难以直接求解，因此可以通过拉普拉斯变换将其由关于时间 $t$ 的线性微分方程变换为关于复数 $s=\sigma+j\omega$ 的多项式代数方程：

$$\mathcal{L}\{f\}=\mathcal{L}\{m\ddot{x}+b\dot{x}+kx\}$$

其中：

$$\begin{cases}\mathcal{L}\{f(t)\}=F(s)\\ \mathcal{L}\{x(t)\}=X(s)\\ \mathcal{L}\{\dot{x}(t)\}=sX(s)\\ \mathcal{L}\{\ddot{x}(t)\}=s^2X(s)\end{cases}$$

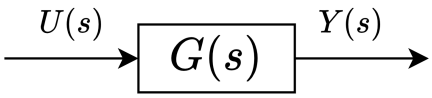
即：

$$F(s)=ms^2X(s)+bsX(s)+kX(s)$$

根据力 $f$ 引起位移 $x$ 的变化这一因果关系，我们可以确定系统输入 $U(s)=F(s)$ ，系统输出 $Y(s)=X(s)$ 。将上式变形为 $\frac{\text{输出}}{\text{输入}}$ 的形式以体现输入输出的因果关系，这样便得到了系统的传递函数：

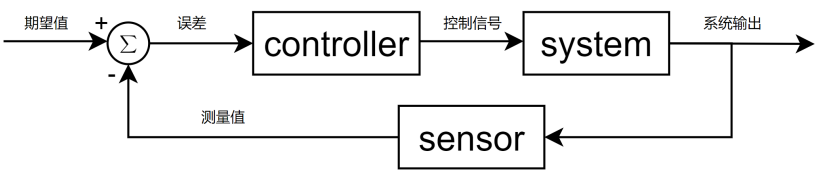
$$G(s)=\frac{Y(s)}{U(s)}=\frac{1}{ms^2+bs+k}$$

不难看出，传递函数描述的便是这个系统输入输出的关系，即：




## 1.4 系统控制

系统控制关注什么样的输入信号能使系统产生符合我们期望的输出响应。这要求我们设计合适的控制器和系统结构以得到恰当的系统输入。要得到符合我们期望的系统输出除了考虑动态系统本身，还需要考虑外界扰动的影响，因此反馈控制应用最为广泛。反馈控制系统的结构可表示为：

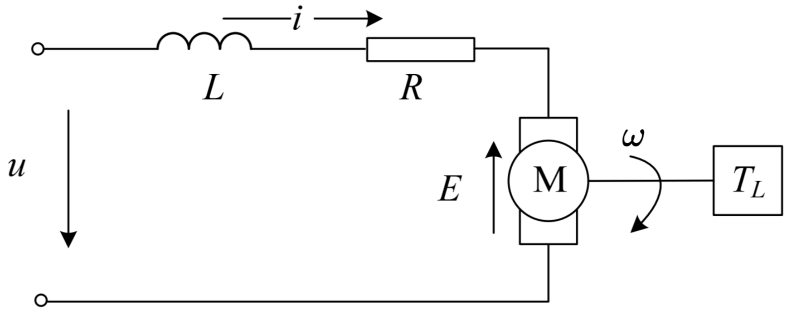


通过反馈回路，我们得以将系统输出的测量值与期望值作比较，以得到误差。误差反应的是当前系统实际表现与我期望的表现两者的差距，通过设计的控制器可以利用误差信号得到恰当的控制信号以减小误差信号，即使系统输出跟随期望值。

# 2 系统组成与基本原理

## 2.1 电机系统建模

采用电压控制时的直流电机等效电路下图所示。




其中，$L$ 和 $R$ 分别是绕组的电感和电阻，$T_L$为负载转矩，本实验中假设负载转矩为 0，即$T_L=0$。绕组通有的直流电流 $i$ 会在磁场中产生电磁转矩 $T=K_ti$，使电机转子旋转，其中 $K_t$ 为转矩常数；另外，电枢导体在定子磁场中以转速 $\omega$ 旋转切割磁力线，产生感应电动势 $E=K_e\omega$，其中 $K_e$ 成为反电动势常数。感应电动势 $E$ 的方向与电枢电流 $i$ 方向相反，称为反电势。

根据基尔霍夫电压定律，有：

$$u(t)=Ri(t)+L\frac{\mathrm{d}i(t)}{\mathrm{d}t}+E(t)$$

根据牛顿定律，有：

$$T=J\frac{\mathrm{d}\omega(t)}{\mathrm{d}t}+b\omega$$

其中 $b$ 为阻尼系数。带入反电动势方程 $E=K_e\omega$与 电磁转矩方程 $T=K_ti$，有：

$$u(t)=Ri(t)+L\frac{\mathrm{d}i(t)}{\mathrm{d}t}+K_e\omega(t)$$
$$K_ti(t)=J\frac{\mathrm{d}\omega(t)}{\mathrm{d}t}+b\omega(t)$$

通过拉普拉斯变换得到其 $s$ 域表达式：

$$\Omega(s)=\frac{K_tI(s)}{Js+b}$$
$$I(s)=\frac{U(s)-K_e\Omega(s)}{Ls+R}$$

两式合并可得到电压到转速的传递函数 $G_v(s)$：

$$G_v(s)=\frac{\Omega(s)}{U(s)}=\frac{K_t}{JLs^2+(JR+Lb)s+Rb+K_eK_t}$$

## 2.2 电机模型单片机仿真代码

代码包含motor_simulation.lib，motor_simulation.h，bsp_dwt.c，bsp_dwt.h四个文件。

- 首先创建电机对象：

```
motorObject_t Motor;
```

- 初始化电机对象：

```
/**
 * @brief Init motor object
 * @param[in]       *motor points to the motorObject_t struct
 */
void Motor_Object_Init(motorObject_t *motor);
```

- 电机仿真：

```
/**
 * @brief Motor simulation implementation
 * @param[in]       *motor points to the motorObject_t struct
 * @param[in]       input voltage
 * @param[in]       simulation period in s
 */
void Motor_Simulation(motorObject_t *motor, float input, float dt);
```

- 电机电流、速度与角度测量：

```
/**
 * @brief Get motor current
 * @param[in]       *motor points to the motorObject_t struct
 * @param[out]      motor current
 */
float Get_Motor_Current(motorObject_t *motor);

/**
 * @brief Get motor velocity
 * @param[in]       *motor points to the motorObject_t struct
 * @param[out]      motor velocity
 */
float Get_Motor_Velocity(motorObject_t *motor);

/**
 * @brief Get motor angle
 * @param[in]       *motor points to the motorObject_t struct
 * @param[out]      motor angle
 */
float Get_Motor_Angle(motorObject_t *motor);
```

例：

```
uint32_t DWT_CNT;
```

```c
float dt;
float Input;
motorObject_t Motor;
pid_t PID;
float VelocityRef = 0;
float Current, Velocity, Angle;
/**
 * @brief  The application entry point.
 * @retval int
 */
int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration--------------------------------------------------------*/

    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    HAL_Init();

    /* USER CODE BEGIN Init */

    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */

    /* USER CODE END SysInit */

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    /* USER CODE BEGIN 2 */
    // DWT定时器初始化：时钟频率[ ]MHz
    DWT_Init([ ]);
    // 电机初始化
    Motor_Object_Init(&Motor);
    // PID初始化
    PID_Init(&PID, 5, 25, 0);
    /* USER CODE END 2 */

    /* Infinite loop */
    /* USER CODE BEGIN WHILE */
    while (1)
    {
        // 利用DWT定时器获取仿真周期
```

请自行思考应该初始化DWT定时器为多少HZ？

```
        dt = DWT_GetDeltaT(&DWT_CNT);

        // 得到电机电流、速度与角度测量值
        Current = Get_Motor_Current(&Motor);
        Velocity = Get_Motor_Velocity(&Motor);
        Angle = Get_Motor_Angle(&Motor);

        // PID计算
        Input = PID_Calculate(&PID, Velocity, VelocityRef, dt);

        // 电机仿真
        Motor_Simulation(&Motor, Input, dt);
        /* USER CODE END WHILE */

        /* USER CODE BEGIN 3 */
        HAL_Delay(1);
    }
    /* USER CODE END 3 */
}
```

### 2.3 PID控制

请大家自行查阅资料，可参考：

1. RoboMaster电控入门（4）PID控制器 - sasasatori - 博客园 (cnblogs.com)
2. PID库与PID基本优化（一） - WangHongxi - 博客园 (cnblogs.com)
3. PID控制器开发笔记之一：PID算法原理及基本实现 - Moonan - 博客园 (cnblogs.com)
4. 串级 PID 为什么外环输入是内环的期望？ - 知乎 (zhihu.com)
5. PID的TRICK(一)简述五种PID积分抗饱和（ANTI-Windup）方法 - 知乎 (zhihu.com)
6. STM32应用(十)经典控制算法PID(单级和串级)原理与代码实现三木今天学习了嘛の博客-*CSDN*博客stm32pid算法

### 2.4 复合控制

请大家自行查阅资料。

## 3 实验内容与要求

### 3.1 控制系统设计与实现

#### 3.1.1 速度闭环

设计反馈控制系统实现电机速度闭环控制。要求掌握闭环控制基本原理与代码实现、三项参数如何影响系统表现。并了解参数整定基本方法。

系统性能检验：

1. 通过阶跃期望信号 velocityRef $= 10\ rad/s$ 检验闭环系统性能。
2. 通过正弦期望信号 velocityRef $= 10\ \sin(\omega_0 t)\ rad/s$ 检验闭环系统性能，其中 $\omega_0$ 使输出正弦信号幅值是期望速度幅值的0.707倍，即velocity $= 7.07sin(\omega_0 t + \psi)\ rad/s$。

### 3.1.2 角度闭环

分别设计单级反馈控制系统与串级反馈控制系统实现电机角度闭环控制，并对比分析两种控制系统优缺点（通过响应速度、精度与抗扰性能三方面分析）。要求明确串级控制系统结构，掌握串级控制系统相比单级反馈控制系统优缺点。并了解参数整定基本方法。

系统性能检验：

1. 阶跃响应：通过阶跃期望信号 angleRef $= 2\pi\ rad$ 对比两种闭环控制系统系统性能。
2. 频率响应：通过正弦期望信号 angleRef $= 2\pi \sin(\omega_0 t)\ rad$ 对比两种闭环控制系统系统性能，其中 $\omega_0$ 使串级控制系统输出正弦信号幅值是期望角度幅值的0.707倍，即angle $= 1.414\pi sin(\omega_0 t + \psi)\ rad$ 。
3. 抗扰性能：angleRef $= 0$，在电机电压输入端加入阶跃扰动输入 disturbance $= 10\ V$，对比单级反馈控制与串级反馈控制对扰动输入的响应。

### 3.1.3 复合控制

采用能想到的各种手段设计具有良好跟随速度、精度以及抗扰性能的控制系统实现电机角度闭环控制。

系统性能检验：

1. 阶跃响应：通过阶跃期望信号 angleRef $= 2\pi\ rad$ 对比复合控制与上一节中性能更好的闭环控制系统性能。
2. 频率响应：通过正弦期望信号 angleRef $= 2\pi \sin(\omega_0 t)\ rad$ 对比复合控制与上一节中性能更好的闭环控制系统性能，其中 $\omega_0$ 使复合控系统输出正弦信号幅值是期望角度幅值的0.707倍，即angle $= 1.414\pi sin(\omega_0 t + \psi)\ rad$。
3. 抗扰性能：angleRef $= 0$，在电机电压输入端加入扰动输入 disturbance $= 10\ V$，对比复合控制与上一节中性能更好的闭环控制系统对扰动输入的响应。

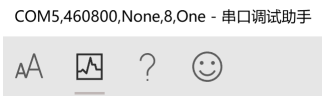
## 3.2 上位机通信程序设计与实现

串口调试助手下载连接：https://www.microsoft.com/store/productId/9NBLGGH43HDM

自行选择串口通信方式(轮询，中断，DMA)，并简单说明选择该方式原因及其优点。

### 3.2.1 串口发送打印变量波形

单片机通过串口向串口调试助手发送字符串，打印变量波形以观察系统动态过程：




打印变量波形协议：名称 = 数值 + ','，一行中包含多条数据可以用','分割。

例：添加两条曲线，名称分别为 Line1 Line2，需单片机向串口调试助手发送字符串：

```
Line1=10.0,Line2=200,
```

### 3.2.2 接收PID参数

串口调试助手向单片机发送字符串，以整定PID参数。PID调参通信协议：参数= 数值 + ','。一行中包含多各参数可以用“,”分割。

例1：设置Kp为10.0，需串口调试助手向单片机发送字符串：

Kp=10.0,

例2：设置Kp为2、Ki为10，Kd为0.1，需串口调试助手向单片机发送字符串：

Kp=2,Ki=10,Kd=0.1,