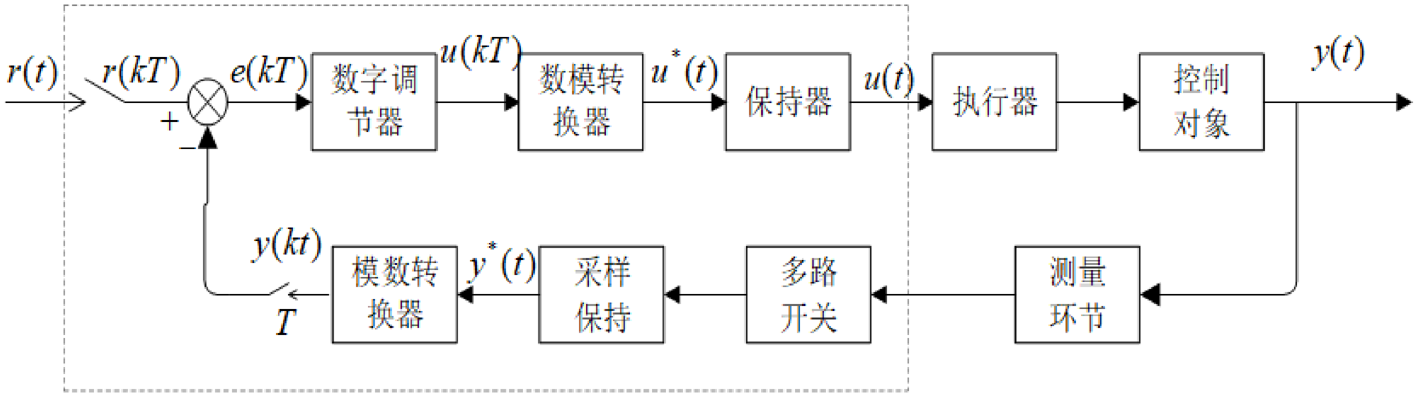
间接设计法

将计算机控制系统近似地看成是一个连续变化的模拟系统，用模拟系统的理论和方法进行分析和设计，得到模拟控制器，然后再将模拟控制器进行离散化，得到数字控制器，这种设计方法叫间接设计法，其中典型的算法为PID算法

这里我们只是提到对控制器进行离散化，那么对控制对象用不用离散化?
答案是不用，因为我们的控制对象前面会接入D/A转换器，他输入的是连续量而非离散量，如下图所示



常用的离散化方法

对于模拟控制器，我们要把他进行离散化，很多人会说那我直接采样不就离散化? 实际上，这是一种离散化办法(脉冲响应不变法)，这里还有其他的离散化办法:

- 前向差分变换(实际一般不用)

$$s=\frac{z-1}{T}$$

- 后向差分变换(常用)

$$s=\frac{1-z^{-1}}{T}$$

- 双线性变化法

$$s=\frac{2}{T}\frac{z-1}{z+1}$$

- 脉冲响应不变法
  实际上就是采样然后z变换过程，给采样周期直接做z变换即可得到
  所谓脉冲响应不变法就是将连续滤波器离散得到离散滤波器后，它的脉冲响应与连续滤波器的脉冲响应在各采样时刻的值是相等的
- 阶跃响应不变法
  有点类似于加上零阶保持器然后z变换的过程，前面带零阶保持器的z变换怎么求，这里就怎么求

所谓阶跃响应不变法就是将连续滤波器离散后得到的离散滤波器，保证其阶跃响应与原连续滤波器的阶跃响应在各采样时刻的值是相等的。

$$G(z) = \frac{z-1}{z}\mathcal{Z}[\frac{G_p(s)}{s}]$$

- 零极点匹配法

$$D(s) = \frac{k\prod_m (s+z_i)}{\prod_n (s+p_i)} \rightarrow z = \mathrm{e}^{sT} \rightarrow D(z) = \frac{k_1\prod_m (z-\mathrm{e}^{-z_iT})}{\prod_n (z-\mathrm{e}^{-p_iT})}(z+1)^{n-m}$$

$$\lim_{s\to 0} sG_c(s)E(s) = \lim_{z\to 1}(z-1)D(z)E(z)$$

确定$\boldsymbol{D(z)}$的增益$\mathbf{k_1}$的方法：

- 按右式来匹配 $D(s)\big|_{s=0} = D(z)\big|_{z=1}$
- 若$\boldsymbol{D(s)}$分子有$\boldsymbol{s}$因子，可依高频段增益相等原则确定增益**,**即

$$\left|D(s)\big|_{s=\infty}\right| = \left|D(z)\big|_{z=-1}\right|$$

- 也可选择某关键频率处的幅频相等，即

$$\left|D(\mathrm{j}\omega_1)\right| = \left|D(\mathrm{e}^{j\omega_1 T})\right|$$

小结

1. 后向差分与双线性变换较为常用
2. 在采样周期T较小时，各种变换方法区别不大
3. 零极点匹配法效果最好，但比较复杂，在要求较高时使用

数字PID控制器

数字式PID是在模拟型PID上离散化过来的，那么对于模拟型PID我们有以下式子:

$$u(t) = K_p[e(t) + \frac{1}{T_I}\int_0^t e(t)d(t) + T_D\frac{de(t)}{dt}]$$

实际上对于上述所有的离散方法，这里都能用，但是一般在工程上我们只用后向差分这一种方法，这里都是基于后向差分得到的，具体推导过程如下:

1. 对于时间t,则变为$t = KT$，通常我们省略T不写，只留K
2. 比例环节不变，积分环节就变成了长度为采样周期，高度为采样值的长方形面积加在一起(就是微元法)，即: $\int e(t)dt = T\sum_{j=0}^{k} e(jT)$
3. 微分环节就变成了后向差分，即: $\frac{de(t)}{dt} = \frac{e(kT)-e[(k-1)T]}{T}$

我们将上述式子替换过去，就得到了位置式PID控制的方程

位置式PID控制

$$
\begin{aligned}
u(k) &= K_p[e(k) + \frac{T}{T_1}\sum_{j=0}^{k} e(j) + \frac{T_D}{T}[e(k) - e(k-1)]] \\
&= K_p e(k) + K_I \sum_{j=0}^{k} e(j) + K_D[e(k) - e(k-1)]
\end{aligned}
$$

$$
\begin{aligned}
K_p\text{比例系数} \quad K_I\text{积分系数} \quad K_D\text{微分系数} \\
t = kT(\text{省略}T) \quad u(k)PID\text{输出} \quad e(k)PID\text{输入}
\end{aligned}
$$

传递函数

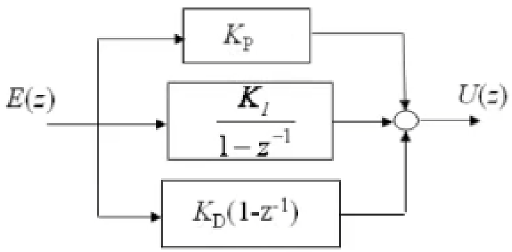
若对上式两边同时取z变换，那么就有(离散化是离散化，z变换是z变换，你不离散化，我怎么Z变换):

$$
U(z) = K_p E(z) + K_I \frac{E(Z)}{1 - z^{-1}} + K_D(1 - z^{-1})E(z)
$$

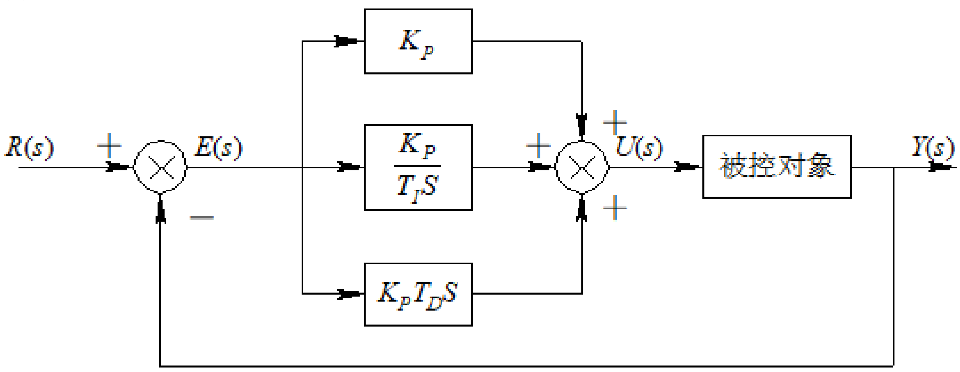
我们将$E(z)$除过去，就可以得到离散PID传递函数:

$$
D(z) = \frac{U(z)}{E(z)} = K_p + \frac{K_I}{1 - z^{-1}} + K_D(1 - z^{-1})
$$

对比离散和连续系统下的PID，如图所示:



模拟PID传递函数为:
$\frac{U(s)}{E(s)} = K_p(1 + \frac{1}{T_1 s} + T_D s)$



有人可能会问，既然是后向差分得到的，何必那么麻烦，我直把$s = \frac{1-z^{-1}}{T}$带入模拟PID传递函数中，不就好了？答案是可以的，求出来的结果与我们上述求得的完全相同

增量式PID算法

所谓增量，就是求在前后两个采样周期之间数值增加了多少，那么就有：

$$\Delta u(k) = u(k) - u(k-1) = \\ K_p[e(k) - e(k-1)] + K_I e(k) + K_D[e(k) - 2e(k-1) + e(k-2)]$$

可以看到当前的增量值与三个采样周期的采样值有关，我们整理一下就有：

$$\Delta u(k) = a_0 e(k) - a_1 e(k-1) + a_2(k-2) \\ a_0 = K_p(1 + \frac{T}{T_I} + \frac{T_D}{T}) \\ a_1 = K_p(1 + 2\frac{T_D}{T}) \\ a_2 = K_p\frac{T_D}{T}$$

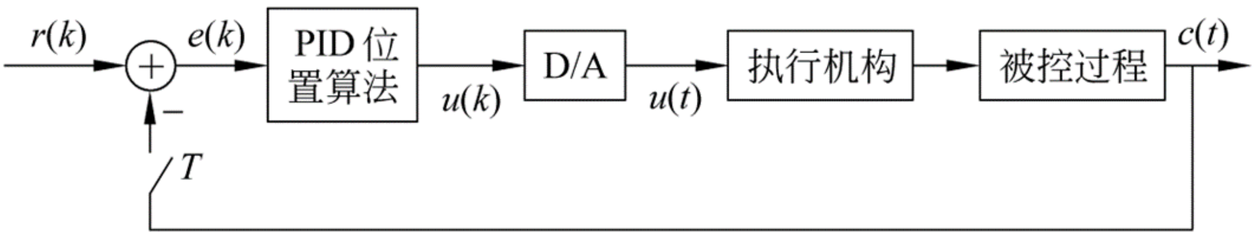
所谓增量式，就是我当前应该在上一个基础上增加多少，那么就有：

$$u(k) = \Delta u(k) + u(k-1)$$

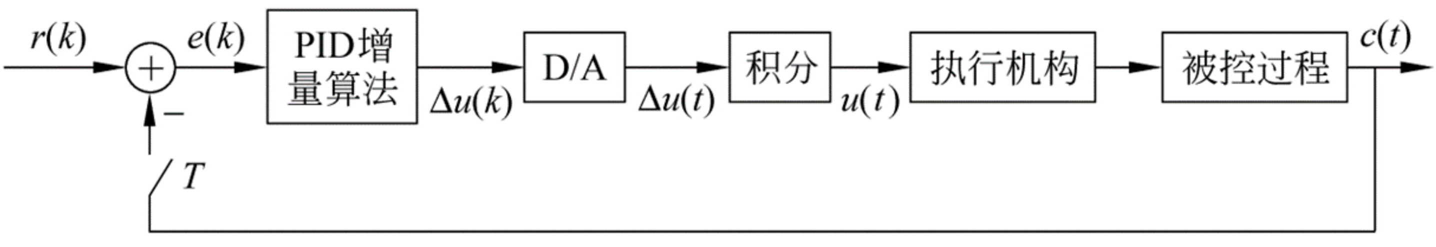
即增加上述过程中我们求出的$\Delta u(t)$

控制结构图

对于上述两种算法，他们控制的结构图如下：



(a)



(b)