建议配合韩京清的书食用
如无特殊说明，下文“自控“均指程鹏的自动控制原理第二版

1.4 安排过渡过程的作用
二阶系统状态方程：

$$\begin{cases} \ddot{x} = -a_1(x - v_0) - a_2\dot{x} \\ y = x \end{cases} \tag{1.4.1}$$

解释 1：为什么$a_1 = r^2, a_2 = 2r, r > 0$的时候过渡过程没超调？
转化成传递函数的形式：

$$\begin{cases} s^2X(s) = -a_1[X(s) - V_0] - a_2sX(s) \\ \quad Y(s) = X(s) \end{cases} \tag{1.4.1-1}$$

此时$X(s)$是状态，V_0是输入控制，$Y(s)$是输出

$$s^2X(s) + a_2sX(s) + a_1X(s) = a_1V_0 \tag{1.4.1-2}$$

$$\frac{X(s)}{V_0} = \frac{Y(s)}{V_0} = \frac{a_1}{s^2 + a_2s + a_1} \tag{1.4.1-3}$$

对比二阶系统闭环传递函数一般形式：

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{1.4.1-4}$$

在阻尼比$\zeta = 1$（临界阻尼）的情况下，二阶系统对单位阶跃响应的跟踪**必定**没有超调（参考自控 P67），系统上升时间完全由ω_n确定。
对比 1.4.1-3 和 1.4.1-4，发现a_1和a_2可以由一个参数确定，也就是$a_1 = \omega_n^2, a_2 = 2\omega_n$，系统的$\omega_n$必定是大于 0 的。

Matlab: r_change.m

解释 2：过渡过程时间（上升时间）如何计算
对闭环传递函数进行拉普拉斯反变换，变换后时域公式必定带有指数项，输出效果到达 1 的时间是$+\infty$，假设取 5%误差带，也就是把输出上升到 0.95 所需要的时间当成过渡过程时间，就能算出结果。

PID 中 P 的增益$k_1(k_P)$和 D 的增益$k_2(k_D)$是用来调整系统的a_1和a_2使得$\overline{a_1} = a_1 + k_1$和$\overline{a_2} = a_2 + k_2$，当$\overline{a_1}$和$\overline{a_2}$满足解释 1 的条件时，经过 PD 控制的系统就可以实现响应快而且无超调的效果。

解释 3：讨论系统参数（a_1和a_2）摄动对 PD 控制后系统对单位阶跃相应的影响
当系统参数不改变的时候，只要 PD 调参调得好，控制效果肯定不错，但是如果系统参数改变，原来的 PD 参数就**大概率**使得控制后系统出现超调或者相应慢的情况。
系统参数改变对于我们来说有可能是飞机加东西了，比如 3S 电池换成了 4S 电池，或者加了个云台，或者飞一段时间后电池电压下来了，电机驱动力不足。

视频 https://www.bilibili.com/video/BV1et41157tZ

解释 4：为什么工程中往往达不到快而无超调的效果
书上 P21 说的是没有获取微分信号的合适办法，我觉得对于阶跃响应来说应该是零时刻倒数无穷大，实际上只能 PI 调节，由于$k_2 = 0$，k_1要满足解释 1 的关系的话，会被系统参数a_1和a_2定死，加大k_1的话系统就超调了，但是不超调的情况下过渡过程时间就太长，也不符合我们控制的需求。

解释 5：引起超调的原因
其实可以很明显的看出，如果目标状态和系统状态误差太大，PI 控制就会发生超调，我们要做的就是减缓两者之间的差值，误差越小，我们调参的范围越大。因为系统到达目标值是需要时间的，也就是所说的过渡过程时间，只要在过渡过程时间之内，v_0到达 1（对于阶跃相应），系统的控制效果就会很好。

问题 1：为什么过渡过程要用正弦曲线而不用一条一次函数的斜线？
猜测是在过渡过程早期，两者误差可以比较大，以加快系统的上升速度，到过渡过程后期，已经接近目标值了，两者误差要比较小，使得调参范围更大，而一次函数的斜线在过渡过程后期的转折点附近误差会突然增大，从而引起超调。

解释 6：安排过渡过程的过渡过程时间如何确定
这个也是工程取法，因为系统到达指定状态需要一定的时间，这是由系统的极限和我们的需要决定的，假设我们要求飞机的横滚在 1ns 内转动 90°，这是不现实的，因为很明显超出了系统的物理极限，以无人机为例，我们程序控制是 10ms 改变一次 PWM 的值，这也是近似于一个阶跃状态，因此过渡过程时间可以定为 10ms。

解释 7：为什么安排过渡过程时间后，k_1会那么大
过渡过程和系统状态之间的误差始终很小，k_1不大的话系统无法在过渡过程结尾到达指定状态。书 P23 所说的快速性和超调并不对立是因为误差小，k_1可以设置得很大，同时，较小的误差也使得系统参数a_1和a_2的摄动对系统影响不大。

解释 8：过渡过程与微分信号的关系
通过对过渡过程求微分：

$$trns(T_0,t) = \begin{cases} \frac{1}{2}\sin\left(\pi\left(\frac{t}{T_0}-\frac{1}{2}\right)\right), t \le T_0 \\ 1, \qquad\qquad\qquad\qquad t > T_0 \end{cases} \tag{1.4.7}$$

$$\frac{d}{dt}[trns(T_0,t)] = \begin{cases} \frac{\pi}{2T_0}\cos\left(\pi\left(\frac{t}{T_0}-\frac{1}{2}\right)\right), t \le T_0 \\ 0, \qquad\qquad\qquad\qquad t > T_0 \end{cases} \tag{1.4.9}$$

现在就相当于输入v_0可导。
我们写一下阶跃函数的微分形式：

$$\varepsilon(\mathrm{t}) = \begin{cases} 0, t = 0 \\ 1, t > 0 \end{cases} \tag{1.4.9 - 1}$$

$$\frac{d}{dt}[\varepsilon(\mathrm{t})] = \begin{cases} +\infty, t = 0 \\ 0, \quad t > 0 \end{cases} \tag{1.4.9 - 2}$$

会发现阶跃函数的微分毫无意义，我们怎么设置k_2效果都不好。安排过渡过程之后，对于单位阶跃函数来说，实际上$trns(T_0,t)$就是$\bar{\varepsilon}(t)$，此时修正后的单位阶跃函数就有意义了，假设

系统输出的微分信号$\dot{y}$（也是系统状态的微分信号$\dot{x}$）能获取，解释 2 和解释 3 的 PD 控制的参数$k_1(k_P)$和$k_2(k_D)$就能合理设置且满足解释 1 的条件，此时的 PD 反馈系统如下：

$$\begin{cases} \overline{a_1} = a_1 + k_1 \\ \overline{a_2} = a_2 + k_2 \\ \ddot{x} = -\overline{a_1}\big(x - v_0 trns(T_0, t)\big) - \overline{a_2}\left(\dot{x} - v_0 \frac{d}{dt}[trns(T_0, t)]\right) \\ y = x \end{cases} \tag{1.4.10}$$

公式 1.4.10 第三行的形式，就是标准的 PD 控制形式，$\overline{a_1}$是经过$k_1(k_P)$修正后的系统参数，$\overline{a_2}$是经过$k_2(k_D)$修正后的系统参数，$x - v_0 trns(T_0, t)$是系统输出信号的误差，

$\dot{x} - v_0 \frac{d}{dt}[trns(T_0, t)]$是系统输出的微分信号的误差。

解释 9：如何对系统安排过渡过程

按照原理安排过渡过程其实非常简单，但是我们需要确定我们的系统需要对几阶安排过渡过程，以无人机为例，我们需要控制的是角度和角速度，那么就需要安排两个阶次的过渡过程，为了保证角速度的过渡过程足够平滑，我们需要再降一阶，角加速度即使不平滑，积分后的角速度过渡过程也是平滑可导的。

解释 10：为什么公式 1.4.7 用正弦函数安排过渡过程

对于**连续**系统来说，用正弦波安排过渡过程是比较好的，第一，足够平滑，第二，过渡过程前期追踪快（仅限一阶系统，二阶及以上变成中期追踪快），第三，任意阶可导。但是也有其缺点，如果需要安排过渡过程的阶次太多，就会导致最高阶的过渡过程中期突变非常厉害，从而又引起超调。

Matlab:sin_tp.m

解释 11：为什么 P28 安排过渡过程变成了方波

当高阶系统所有状态都需要跟踪的时候，最低阶用正弦波安排会使得v_2近似一个冲激函数，此时v_1就又变成了阶跃函数，不利于系统跟踪。假设系统需要跟踪 n 阶状态，最低阶的过渡过程设为v_n，最高阶过渡过程为v_1，那么v_{n+1}安排成方波的话，保证了v_n分部可导，而低阶状态对系统影响并不大，所以高阶系统安排过渡过程可以用方波。

2.1 小时间常数惯性环节

解释 1：P48 公式如何近似的

$$\frac{1}{1+Ts} \approx e^{-Ts} \tag{2.1.6}$$

公式 2.1.6 实际上就是欧拉公式，令$s = j\omega$：

$$e^{-Ts} = e^{-Tj\omega} = \frac{1}{\cos(T\omega) + jsin(T\omega)} \tag{2.1.6-1}$$

由于此时T较小，$\cos(T\omega) \approx 1, sin(T\omega) \approx T\omega$，便可得到 2.1.6 的公式。而$e^{-Ts}$这个环节在自动控制里面是延时环节，在时域上的表示就是一个延时的阶跃函数，而 T 较小的一阶惯性环节效果恰好近似。

2.2 经典微分器

解释 1：小时间常数惯性环节在微分器中的应用以及对噪声的影响
经典微分环节一般如下：

$$y = \frac{1}{T}\left(v - \frac{1}{Ts+1}v\right) \tag{2.2.2}$$

在时间常数较小的情况下，按照 2.1 的解释 1，$\frac{1}{Ts+1}v$可以认为是对v信号滞后T时间后的$\bar{v}(t)$，从而得出：

$$y(t) = \frac{1}{T}[v(t) - \bar{v}(t)] \approx \frac{1}{T}[v(t) - v(t-T)] \approx \dot{v}(t) \tag{2.2.4}$$

在没有噪声的情况下，微分时间越短，微分信号的近似度越高。然而，加入白噪声之后情况却大为不同，假设输入信号中包含了均值为零的白噪声，公式 2.2.4 变成如下形式：

$$y(t) = \frac{1}{T}[v(t) + n(t) - \bar{v}(t) - \bar{n}(t)] \tag{2.2.6}$$

经过 P50 的积分计算（不会算），噪声的延时信号没了，公式 2.2.6 变成下式：

$$y(t) = \frac{1}{T}[v(t) + n(t) - \bar{v}(t)] \approx \dot{v}(t) + \frac{1}{T}n(t) \tag{2.2.7}$$

可见时间常数越小，对噪声的放大越严重。

解释 2：新的微分近似公式为什么效果会好
先说回过渡过程的问题，我们的系统不可能对突变信号进行无差跟踪，只要在能接受的较短的时间内到达目标值就可以了，如果我们对阶跃信号安排一个一阶惯性环节的过渡过程，然后对这个信号进行延时（实际上用的都是一阶惯性环节），就可以得到新的微分近似公式。

$$y = \frac{1}{\tau_2 - \tau_1}\left(\frac{1}{\tau_1 s + 1} - \frac{1}{\tau_2 s + 1}\right)v = \frac{s}{\tau_2\tau_1 s^2 + (\tau_2 + \tau_1)s + 1}v \tag{2.2.9}$$

将公式 2.2.9 转化成传递函数形式：

$$\tau_2\tau_1\ddot{y} + (\tau_2 + \tau_1)\dot{y} + y = \dot{v} \tag{2.2.9-1}$$

$$\ddot{y} + \frac{\tau_2 + \tau_1}{\tau_2\tau_1}\dot{y} + \frac{1}{\tau_2\tau_1}y = \frac{\dot{v}}{\tau_2\tau_1} \tag{2.2.9-2}$$

$$\dot{X} = \begin{bmatrix}\dot{x}_1 \\ \dot{x}_2\end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{\tau_2\tau_1} & -\frac{\tau_2 + \tau_1}{\tau_2\tau_1}\end{bmatrix}\begin{bmatrix}x_1 \\ x_2\end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{\tau_2\tau_1}\end{bmatrix}v \tag{2.2.9-3}$$

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{1}{\tau_2\tau_1}(x_1 - v) - \frac{\tau_2 + \tau_1}{\tau_2\tau_1}x_2 \\ y = x_2 \end{cases} \tag{2.2.10}$$

对公式 2.2.10 离散化：

$$\begin{cases} \frac{x_1(k+1) - x_1(k)}{h} = x_2(k) \\ \frac{x_2(k+1) - x_2(k)}{h} = -\frac{1}{\tau_2\tau_1}\big(x_1(k) - v(k)\big) - \frac{\tau_2 + \tau_1}{\tau_2\tau_1}x_2(k) \\ y(k) = x_2(k) \end{cases} \tag{2.2.12}$$

求公式 2.2.2 的状态方程：

$$y = \frac{s}{Ts+1}v \tag{2.2.2-1}$$

$$\dot{y} + \frac{y}{T} = \frac{\dot{v}}{T} \tag{2.2.2-2}$$

$$\begin{cases} \dot{x}_1 = v - \frac{1}{T}x_1 \\ y = \frac{1}{T}\dot{x}_1 \end{cases} \tag{2.2.2-3}$$

离散化可得：

$$-\begin{cases} \frac{x_1(k+1) - x_1(k)}{h} = v(k) - \frac{1}{T}x_1(k) \\ y(k) = \frac{1}{T}\frac{x_1(k+1) - x_1(k)}{h} \end{cases}$$

将噪声$v_0(t) + \gamma n(t)$代入 2.2.12：

$$\begin{cases} \frac{x_1(k+1) - x_1(k)}{h} = x_2(k) \\ \frac{x_2(k+1) - x_2(k)}{h} = -\frac{1}{\tau_2\tau_1}\Big(x_1(k) - \big(v_0(t) + \gamma n(t)\big)\Big) - \frac{\tau_2 + \tau_1}{\tau_2\tau_1}x_2(k) \\ y(k) = x_2(k) \end{cases} \tag{2.2.15}$$

可以发现$x_2(k+1)$（也就是系统下一时刻的输出）的噪声项为：

$$\frac{h\gamma n(t)}{\tau_2\tau_1} \tag{2.2.15-1}$$

噪声项和$\tau_2\tau_1$成反比，和积分步长h成正比，因此，小时间常数所带来的噪声增益可以由较短的积分步长来抵消。

解释 3：1.4 解释 1 的快而无超调条件在跟踪微分器中的应用

如果τ_1和τ_2很接近，我们新的微分近似公式 2.2.9 就可以化为以下形式：

$$w(s) = \frac{s}{\tau^2 s^2 + 2\tau s + 1} \tag{2.2.16}$$

$$w(s) = s\frac{\frac{1}{\tau^2}}{s^2 + 2\frac{1}{\tau}s + \frac{1}{\tau^2}} \tag{2.2.16-1}$$

实际上这就是一个微分环节乘一个二阶震荡环节，且二阶震荡环节满足快而无超调的条件，因此跟踪微分器可以通过小时间常数的一阶惯性环节快速跟踪系统状态，同时能用另一个一阶惯性环节组合获得系统的微分信号（噪声靠比时间常数数量级更小的积分步长抑制）。

2.3 跟踪微分器的一般形式

简介：前面说完跟踪微分器的效果之后，我们需要考虑在单片机中实际能实现的跟踪微分器形式（实际上就是离散形式），TD 的重头戏是在最优控制综合函数的推导中，不然你就只能直接用现成的公式了，不能深入理解。

补充解释 1：高阶微分功能如何体现

从 2.2 我们知道当τ_1和τ_2很接近，两个小时间常数惯性环节的并联框图可以近似为二阶震荡环节，因此也赋予了二阶震荡环节有一阶问分的功能。如果是多个小时间常数惯性环节并联，其实是类似的，二阶震荡环节只是阶次为 2 的特例。

$$y = w(s)v = \frac{r^m}{(s+r)^m}v \tag{2.2.19}$$

当$m = 2$时，就变成了二阶震荡环节，且阻尼比$\zeta = 1$。

高阶微分其实很好体现，输入给一个正弦波就能看出来了，相位差 90 度。

解释 1：线性跟踪微分器

一般情况来说，我们获得输入信号的微分信号已经很足够了，所以这个线性跟踪微分器考虑对二阶震荡环节离散化来获得。

$$W(s) = \frac{r^2}{s^2 + 2rs + r^2} \tag{2.3.3}$$

阻尼比$\zeta = 1$，那就不用说了，r 越大，跟踪越快，而且必定无超调，然后我们转化成输入为v_0的系统的离散状态方程形式：

$$\begin{cases} \dfrac{x_1(k+1) - x_1(k)}{h} = x_2(k) \\ \dfrac{x_2(k+1) - x_2(k)}{h} = -r^2\big(x_1(k) - v_0(k)\big) - 2rx_2(k) \end{cases} \tag{2.3.4}$$

这个就是离散系统的线性微分跟踪器，P58 用它来跟踪正弦波信号，$r = 50$的时候跟踪有点超前，随着r增大，跟踪曲线基本和正弦波贴在一起了。微分信号就是相位落后 90°。

重点：最优控制综合函数的含义和推导

看书的时候注意字母是否为斜体，斜体其实是变量，非斜体是函数名字，例如：

$$u(x_1, x_2) = -r\,\mathrm{sign}\left(x_1 + \frac{x_2|x_2|}{2r}\right) \tag{2.3.6}$$

书上的r和sign之间是没空格的，实际上是$r * \mathrm{sign}(\mathrm{XXXX})$，$\mathrm{sign}$是符号函数。

考虑到实际使用和容易理解，我们推导有两个状态变量的离散系统的最优控制综合函数：

$$\begin{cases} \dfrac{x_1(k+1) - x_1(k)}{h} = x_2(k) \\ \dfrac{x_2(k+1) - x_2(k)}{h} = u, \quad |u| \le r \end{cases} \tag{2.7.1-1}$$

式子中u为控制量，$|u| \le r$的意思就是说我们对系统的控制能力并不是无限大的，上面说的r越大，跟踪效果越好，但是这受到了系统控制能力的限制，例如无人机要向前飞行，那就意味着俯仰要摆到一个角度（无超调地到达这个角度），最极端的情况就是从零时刻到某一

时刻后面两个电机满速转，前面两个电机停转，这都不能在指定时间内无超调到达目标角度，就是超出我们控制能力。

首先最优控制综合函数要求系统能在有限的步数内通过控制序列$u(0), u(1), \cdots, u(k)$到达目标状态，公式$(2.7.1-1)$换回书上原来的形式会比较好理解：

$$\begin{cases} x_1(k+1) = x_1(k) + hx_2(k) \\ x_2(k+1) = x_2(k) + hu(k), |u| \le r \end{cases} \tag{2.7.1}$$

假设我们一步就到达目标状态了：

$$\begin{cases} x_1(1) = x_1(0) + hx_2(0) \\ x_2(1) = x_2(0) + hu(0), |u| \le r \end{cases} \tag{2.7.1-2}$$

写成矩阵形式：

$$\begin{bmatrix} x_1(1) \\ x_2(1) \end{bmatrix} = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} + \begin{bmatrix} 0 \\ h \end{bmatrix} u(0) \tag{2.7.1-3}$$

再假设我们两步到达目标状态：

$$\begin{aligned} &\begin{cases} x_1(2) = x_1(1) + hx_2(1) \\ x_2(2) = x_2(1) + hu(1), |u| \le r \end{cases} = \\ &\begin{cases} x_1(2) = x_1(0) + hx_2(0) + h\big(x_2(0) + hu(0)\big) \\ x_2(2) = x_2(0) + hu(0) + hu(1), \qquad\qquad |u| \le r \end{cases} = \\ &\begin{cases} x_1(2) = x_1(0) + 2hx_2(0) + h^2u(0) \\ x_2(2) = x_2(0) + hu(0) + hu(1), \quad |u| \le r \end{cases} \end{aligned} \tag{2.7.1-4}$$

写成矩阵形式：

$$\begin{bmatrix} x_1(2) \\ x_2(2) \end{bmatrix} = \begin{bmatrix} 1 & 2h \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} + \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ h \end{bmatrix} u(0) + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ h \end{bmatrix} u(1) \tag{2.7.1-5}$$

那么可以推导出 k 步到达目标状态的矩阵形式：

$$\begin{aligned} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} = &\begin{bmatrix} 1 & kh \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} \\ &+ \begin{bmatrix} 1 & (k-1)h \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ h \end{bmatrix} u(0) \\ &+ \cdots \\ &+ \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ h \end{bmatrix} u(k-2) \\ &+ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ h \end{bmatrix} u(k-1) \end{aligned} \tag{2.7.2}$$

然后我们让我们的目标状态等于 0（其实就是到达原点之后，系统的位置$x_1 = 0$，速度$x_2 = 0$，也就是到达原点停下来了）。

将公式 2.7.2 右边第一项移到左边：

$$\begin{aligned} -\begin{bmatrix} 1 & kh \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = &\begin{bmatrix} 1 & (k-1)h \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ h \end{bmatrix} u(0) \\ &+ \cdots \\ &+ \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ h \end{bmatrix} u(k-2) \\ &+ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ h \end{bmatrix} u(k-1) \end{aligned} \tag{2.7.2}$$

$-\begin{bmatrix} 1 & kh \\ 0 & 1 \end{bmatrix}$可逆，逆矩阵是$\begin{bmatrix} -1 & kh \\ 0 & -1 \end{bmatrix}$，于是得到式子：

$$\begin{aligned}\begin{bmatrix}x_1(0)\\x_2(0)\end{bmatrix}=&\begin{bmatrix}-1 & kh\\0 & -1\end{bmatrix}\begin{bmatrix}1 & (k-1)h\\0 & 1\end{bmatrix}\begin{bmatrix}0\\h\end{bmatrix}u(0)\\&+\cdots\\&+\begin{bmatrix}-1 & kh\\0 & -1\end{bmatrix}\begin{bmatrix}1 & h\\0 & 1\end{bmatrix}\begin{bmatrix}0\\h\end{bmatrix}u(k-2)\\&+\begin{bmatrix}-1 & kh\\0 & -1\end{bmatrix}\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}0\\h\end{bmatrix}u(k-1)\end{aligned}\tag{2.7.2-1}$$

$$\begin{aligned}\begin{bmatrix}x_1(0)\\x_2(0)\end{bmatrix}=&\begin{bmatrix}-1 & kh\\0 & -1\end{bmatrix}\begin{bmatrix}1 & (k-1)h\\0 & 1\end{bmatrix}\begin{bmatrix}0\\h\end{bmatrix}u(0)\\&+\cdots\\&+\begin{bmatrix}-1 & kh\\0 & -1\end{bmatrix}\begin{bmatrix}1 & h\\0 & 1\end{bmatrix}\begin{bmatrix}0\\h\end{bmatrix}u(k-2)\\&+\begin{bmatrix}-1 & kh\\0 & -1\end{bmatrix}\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}0\\h\end{bmatrix}u(k-1)\end{aligned}\Rightarrow\begin{aligned}\begin{bmatrix}x_1(0)\\x_2(0)\end{bmatrix}=&\begin{bmatrix}h^2\\-h\end{bmatrix}u(0)\\&+\begin{bmatrix}2h^2\\-h\end{bmatrix}u(1)\\&+\cdots\\&+\begin{bmatrix}(k-1)h^2\\-h\end{bmatrix}u(k-2)\\&+\begin{bmatrix}kh^2\\-h\end{bmatrix}u(k-1)\end{aligned}\tag{2.7.3}$$

说一下 P98 那个网格图的意思：

由于我们只有两个（阶）状态，因此可以画在 xy 坐标轴下，x 轴表示x_1，y 轴表示x_2，到达原点就是位置和速度都归零。这个玩意叫最优控制，其实就是说我们的控制量拉满的情况下，能以最少的步长到达我们的目标状态，假设我们一步到达：

$$\begin{bmatrix}x_1(0)\\x_2(0)\end{bmatrix}=\begin{bmatrix}h^2\\-h\end{bmatrix}u(0)=\begin{bmatrix}h^2r\\-hr\end{bmatrix}\text{或}\begin{bmatrix}-h^2r\\hr\end{bmatrix}\tag{2.7.3-1}$$

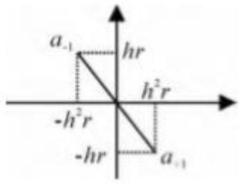


图 1　1 步到达原点的初值构成的图形

假设我们两步到：

$$\begin{aligned}a_{+2}&=\begin{bmatrix}x_1(0)\\x_2(0)\end{bmatrix}=\begin{bmatrix}h^2\\-h\end{bmatrix}u(0)+\begin{bmatrix}2h^2\\-h\end{bmatrix}u(1)=\begin{bmatrix}3h^2r\\-2hr\end{bmatrix},u(0)=r,u(1)=r\\b_{+2}&=\begin{bmatrix}x_1(0)\\x_2(0)\end{bmatrix}=\begin{bmatrix}h^2\\-h\end{bmatrix}u(0)+\begin{bmatrix}2h^2\\-h\end{bmatrix}u(1)=\begin{bmatrix}-h^2r\\0\end{bmatrix},u(0)=r,u(1)=-r\\b_{-2}&=\begin{bmatrix}x_1(0)\\x_2(0)\end{bmatrix}=\begin{bmatrix}h^2\\-h\end{bmatrix}u(0)+\begin{bmatrix}2h^2\\-h\end{bmatrix}u(1)=\begin{bmatrix}h^2r\\0\end{bmatrix},u(0)=-r,u(1)=r\\a_{-2}&=\begin{bmatrix}x_1(0)\\x_2(0)\end{bmatrix}=\begin{bmatrix}h^2\\-h\end{bmatrix}u(0)+\begin{bmatrix}2h^2\\-h\end{bmatrix}u(1)=\begin{bmatrix}-3h^2r\\2hr\end{bmatrix},u(0)=-r,u(1)=-r\end{aligned}\tag{2.7.3-2}$$

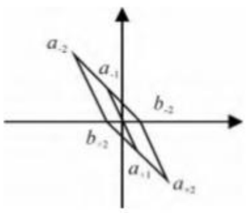


图 2　2 步内到达原点的初值构成的图形

也就是说，控制序列为 k 的情况下，有2^k个点，按照特定的控制序列就能最快到达目标状态。

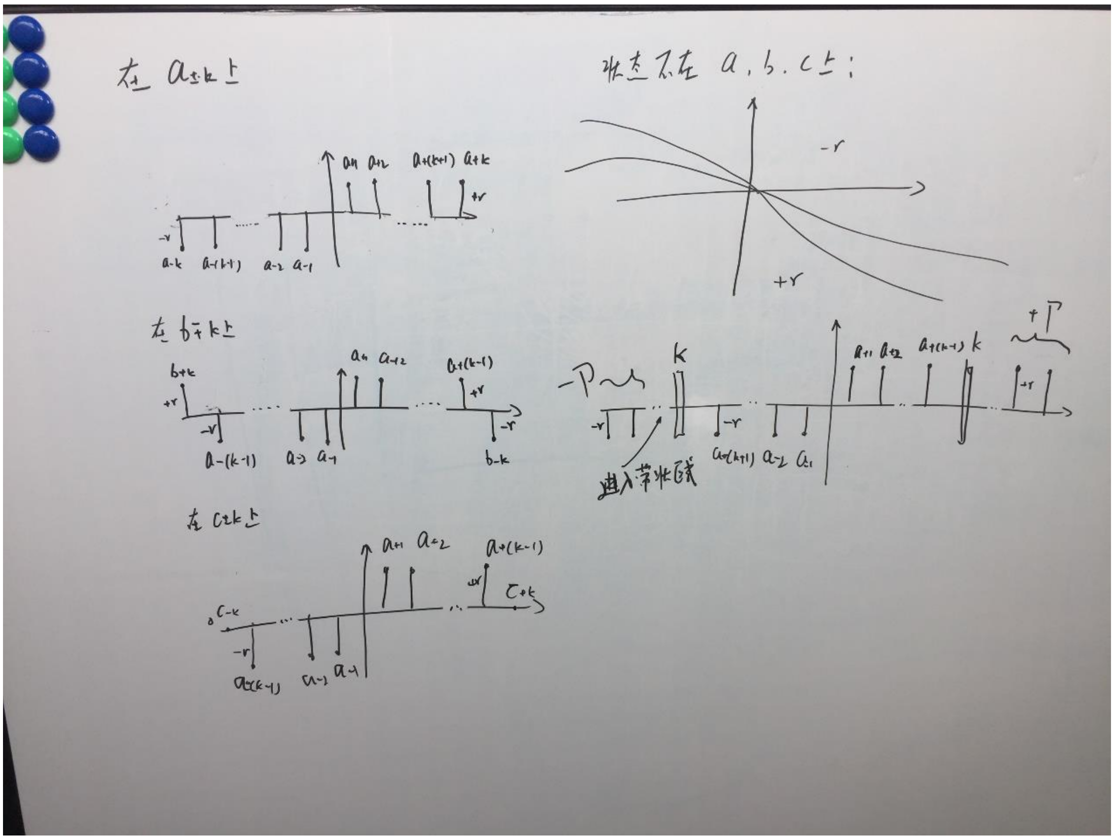
理论上，假设我们的初始状态在$a_{\pm k}, b_{\mp k}, c_{\pm k}$上，我们可以通过第一步分别取$\pm r$（针对$a_{\pm k}$）

,$\mp r$（针对$b_{\mp k}$）,0（针对$c_{\pm k}$）到达$a_{\pm(k-1)}$，然后控制量r拉满就完事了。

但是，我们的状态并不一定在点上，假设我们的点在带状区域外，我们可以速度先拉满，进入到带状区域内之后第一步的 r 控制一下，让它到达$a_{\pm k}, b_{\mp k}, c_{\pm k}$的点上，再经过一步控制到达$a_{\pm(k-1)}$，然后控制量$r$拉满到达目标点。

这种情况下就要求我们的控制步数大于两步，最极端情况是我们到最后两步才到达带状区域，也就是当$k \geq 3$的时候控制量是$\pm r$，当$k = 2$时控制量是$\pm(2a-1)r$到达$a_{\pm 1}$，然后控制量变成$\pm r$到达原点。

其实说$a_{\pm 1}$不是$\pm r$也不是不可以，但是最后一步用$\pm(2a-1)r$貌似就不满足最优控制综合函数的要求了。



虽然我们控制是离散的，但是我们要在程序里用一条（或多条）式子表示出无限个点的序列，也就是说只需要折线和曲线在点上重合就好。

下面推导拟合的曲线，这个直接给结果就算了。

$$x_1 = \frac{-\text{sign}(x_2)x_2^2 - hrx_2}{2r}, a_{\pm i}, i \geq 2 \tag{2.7.4}$$

$$x_1 = \frac{-\text{sign}(x_2)2h^2r^2 - 5hrx_2 - \text{sign}(x_2)x_2^2}{2r}, b_{\pm i}, i \geq 2 \tag{2.7.5}$$

$$x_1 = \frac{-3hrx_2 - \text{sign}(x_2)x_2^2}{2r}, c_{\pm i}, i \geq 2 \tag{2.7.6}$$

我们的最终目标是将三个对接抛物线最好能用一样的式子表达出来，于是先移项：

$$\begin{cases}2r(x_1+hx_2)=-\mathrm{sign}(x_2)x_2^2+hrx_2\\2r(x_1+hx_2)=-\mathrm{sign}(x_2)x_2^2-3hrx_2-\mathrm{sign}(x_2)2h^2r^2\\2r(x_1+hx_2)=-\mathrm{sign}(x_2)x_2^2-hrx_2\end{cases}\tag{2.7.6-1}$$

发现三条对接抛物线在$G(2)$之外的部分都位于$|x_1+hx_2|\ge h^2r$之外，记$y=x_1+hx_2$：

$$\begin{cases}\left(x_2+\mathrm{sign}(x_2)\dfrac{hr}{2}-\mathrm{sign}(x_2)hr\right)^2=\dfrac{h^2r^2-8ry\mathrm{sign}(x_2)}{4}\\\left(x_2+\mathrm{sign}(x_2)\dfrac{hr}{2}+\mathrm{sign}(x_2)hr\right)^2=\dfrac{h^2r^2-8ry\mathrm{sign}(x_2)}{4}\\\left(x_2+\mathrm{sign}(x_2)\dfrac{hr}{2}\right)^2=\dfrac{h^2r^2-8ry\mathrm{sign}(x_2)}{4}\end{cases}\tag{2.7.6-2}$$

开方：

$$\begin{cases}x_2-\mathrm{sign}(x_2)\dfrac{\sqrt{h^2r^2-8ry\mathrm{sign}(x_2)}-hr}{2}=\mathrm{sign}(x_2)hr\\x_2-\mathrm{sign}(x_2)\dfrac{\sqrt{h^2r^2-8ry\mathrm{sign}(x_2)}-hr}{2}=-\mathrm{sign}(x_2)hr\\x_2-\mathrm{sign}(x_2)\dfrac{\sqrt{h^2r^2-8ry\mathrm{sign}(x_2)}-hr}{2}=0\end{cases}\tag{2.7.8}$$

现在等式左边形式完全一样了，然后把符号函数也替换成和y相关的：

$$\mathrm{sign}(x_2)=-\mathrm{sign}(y)\tag{2.7.9}$$

$$\begin{cases}x_2+\mathrm{sign}(y)\dfrac{\sqrt{h^2r^2-8ry\mathrm{sign}(y)}-hr}{2}=-\mathrm{sign}(y)hr\\x_2+\mathrm{sign}(y)\dfrac{\sqrt{h^2r^2-8ry\mathrm{sign}(y)}-hr}{2}=\mathrm{sign}(y)hr\\x_2+\mathrm{sign}(y)\dfrac{\sqrt{h^2r^2-8ry\mathrm{sign}(y)}-hr}{2}=0\end{cases}\tag{2.7.10}$$

因为$y\mathrm{sign}(y)=|y|$：

$$\begin{cases}x_2+\mathrm{sign}(y)\dfrac{\sqrt{h^2r^2-8r|y|}-hr}{2}=-\mathrm{sign}(y)hr\\x_2+\mathrm{sign}(y)\dfrac{\sqrt{h^2r^2-8r|y|}-hr}{2}=\mathrm{sign}(y)hr\\x_2+\mathrm{sign}(y)\dfrac{\sqrt{h^2r^2-8r|y|}-hr}{2}=0\end{cases}\tag{2.7.10-1}$$

记$a(x_1,x_2,r,h)=x_2+\mathrm{sign}(y)\frac{\sqrt{h^2r^2-8r|y|}-hr}{2}$，（$y$和$x_1,x_2$相关），然后顺便把等式右边的$\mathrm{sign}(y)$消了，这个很简单，因为$a_{+i}$和$b_{+i}$用同一个抛物线公式，$a_{-i}$和$b_{-i}$用另一个：

$$\begin{cases}a(x_1,x_2,r,h)|_{a_{+i},b_{+i},i\ge2}=-hr,\\a(x_1,x_2,r,h)|_{a_{-i},b_{-i},i\ge2}=hr\\a(x_1,x_2,r,h)|_{c_{+i},c_{+i},i\ge2}=0\end{cases}\tag{2.7.12}$$

a_{+i}和b_{+i}拟合出来的曲线称为$\tilde{\Gamma}^{+}$，a_{-i}和b_{-i}拟合出来的曲线称为$\tilde{\Gamma}^{-}$，c_{-i}和c_{-i}拟合出来的曲线称为$\tilde{\Gamma}^{0}$。

也就是说有以下几种情况：

点在$\tilde{\Gamma}^{+}$下方，就用$+r$控制，点在$\tilde{\Gamma}^{-}$上方，就用$-r$控制

点在$\tilde{\Gamma}^{+}$线上，如果是a_{+i}，就用$+r$控制，如果是b_{+i}，就用$-r$控制

（经过这步控制都会到达$a_{+(i-1)}$，然后就是一路$+r$）

点在$\tilde{\Gamma}^-$线上，如果是a_{-i}，就用$-r$控制，如果是b_{-i}，就用$+r$控制
（经过这步控制都会到达$a_{-(i-1)}$，然后就是一路$-r$）
点在$\tilde{\Gamma}^0$线上，就用0控制
（经过这步控制都会到达$a_{+(i-1)}$或者$a_{-(i-1)}$，然后就是一路或者$+r-r$）
如果是在$\tilde{\Gamma}^+$和$\tilde{\Gamma}^-$之间，稍微对r进行控制，也就是这一步不拉满，让状态跳到$a_{+(i-1)}$或者$a_{-(i-1)}$，然后就是一路或者$+r-r$。
因此，在等时区$G(2)$之外的部分的最优控制综合函数就出来了：

$$u = -r\mathrm{sat}(a(x_1,x_2,r,h),hr), |y| \geq h^2 r \qquad (2.7.15)$$

sat是一个饱和函数，形式如下：

$$\mathrm{sat}(x,\delta) = \begin{cases} \mathrm{sign}(x), |x| \geq \delta \\ \dfrac{x}{\delta}, \qquad |x| \leq \delta \end{cases} \qquad (2.7.15-1)$$

简单来说就是$a(x_1,x_2,r,h) \geq hr$或者$\leq -hr$就用$\mathrm{sign}\big(a(x_1,x_2,r,h)\big)$，$\mathrm{sign}(x)$是符号函数，也就是只有$\pm r$，如果$-hr \leq a(x_1,x_2,r,h) \leq hr$就用$-r\frac{a(x_1,x_2,r,h)}{hr}$，然后你看$a(x_1,x_2,r,h)$的值域就会发现，$\frac{a(x_1,x_2,r,h)}{hr}$刚好在$\pm 1$之间，绝了。

下面来考虑等时区$G(2)$之内的情况：
这里也可以直接抛出结论，有以下几种情况：
如果点在$a_{+2}b_{+2}$线上，$u=+r$之后就会到达$a_{+1}a_{-1}$线上。
如果点在$a_{-2}b_{-2}$线上，$u=-r$之后就会到达$a_{+1}a_{-1}$线上。
如果点在$c_{+2}c_{-2}$线上，$u=0$之后就会到达$a_{+1}a_{-1}$线上。
然后控制最后一步就完事了

那么怎么求解最后一步的控制量呢？
见 2.7.3-2，为了方便看我复制下来了：

$$\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} h^2 \\ -h \end{bmatrix} u(0) + \begin{bmatrix} 2h^2 \\ -h \end{bmatrix} u(1) \qquad (2.7.3-2)$$

$$\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 2h^2 & h^2 \\ -h & -h \end{bmatrix} \begin{bmatrix} u(1) \\ u(0) \end{bmatrix} \qquad (2.7.3-3)$$

给$\begin{bmatrix} 2h^2 & h^2 \\ -h & -h \end{bmatrix}$求个逆就行了：

$$\begin{bmatrix} 2h^2 & h^2 \\ -h & -h \end{bmatrix}^{-1} = \frac{\begin{bmatrix} -h & -h^2 \\ h & 2h^2 \end{bmatrix}}{-h^3} \qquad (2.7.3-4)$$

$$\begin{bmatrix} u(1) \\ u(0) \end{bmatrix} = \frac{\begin{bmatrix} -h & -h^2 \\ h & 2h^2 \end{bmatrix}}{-h^3} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} \dfrac{-hx_1(0) - h^2x_2(0)}{-h^3} \\ \dfrac{hx_1(0) + 2h^2x_2(0)}{-h^3} \end{bmatrix} = \begin{bmatrix} \dfrac{-x_1(0) - hx_2(0)}{-h^2} \\ \dfrac{x_1(0) + 2hx_2(0)}{-h^2} \end{bmatrix} \qquad (2.7.3-5)$$

这里是顺便把$u(1)$算出来了，但是这个是没鸟用的，因为$u(1)$我们上面已经定好了。
注：2.7.3-5 正负号反了，没检查出来哪儿的问题。
以上这个玩意就是 P106 公式 2.7.18 的离散形式，写代码的时候直接用公式 2.7.18
这里我们考虑的是$G(2)$之内的情况，如果状态处于曲线拟合的区域和$G(2)$区域之外，按照上

面带状区域的要求，应该是取$+r$或者$-r$，所以这里也要来一个饱和函数：

$$u = -r\text{sat}\left(x_2 + \frac{y}{h}, hr\right), |y| \le h^2 r \tag{2.7.20}$$

所以我们整个最优控制综合函数就出来了：

$$\begin{cases} d = rh \\ d_0 = hd \\ y = x_1 + hx_2 \\ a_0 = \sqrt{h^2r^2 - 8r|y|} = \sqrt{d^2 - 8r|y|} \\ a = \begin{cases} x_2 + \text{sign}(y)\dfrac{\sqrt{h^2r^2 - 8r|y|} - hr}{2} = x_2 + \text{sign}(y)\dfrac{a_0 - d}{2}, G(2)\text{之外} \\ x_2 + \dfrac{y}{h}, \qquad G(2)\text{之内} \end{cases} \\ \text{fhan} = -\begin{cases} r\text{sign}(\text{a}), \text{带状区域内} \\ r\dfrac{a}{d}, \qquad \text{带状区域内} \end{cases} \end{cases} \tag{2.7.23}$$

后面就是线性化，感觉上程序直接用这个也可以了，加几个判断语句就好。