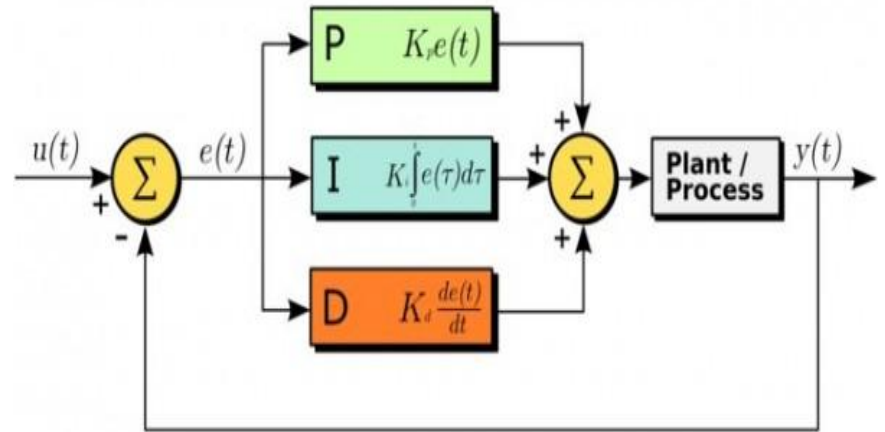


PID 介紹

Theory

$$u = \underbrace{K_p e}_{\text{Proportional Term}} + \underbrace{K_i \int_0^t e dt}_{\text{Integral Term}} + \underbrace{K_d \frac{d}{dt} e}_{\text{Differential Term}}$$



目標位置

離目標還很遠時

速度大

往前飛，越來越接近目標後，
誤差變小，速度也變小

$$u = K_p e$$

速度小

Step1:



欲修正的誤差(error): $2\text{m} - 1\text{m} = 1\text{m}$

Step2:



利用PID去smooth原本的誤差

1m → 0.4m

將誤差轉換成速度給無人機

Step3:



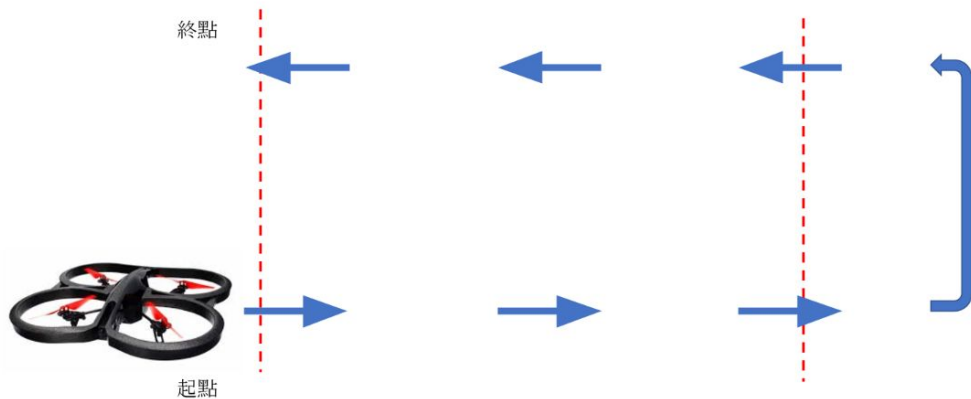
根據無人機飛行的
狀況調整PID

1. 先把I, D設為0, 先調整P
2. 再調整I, D

lab06

Demo

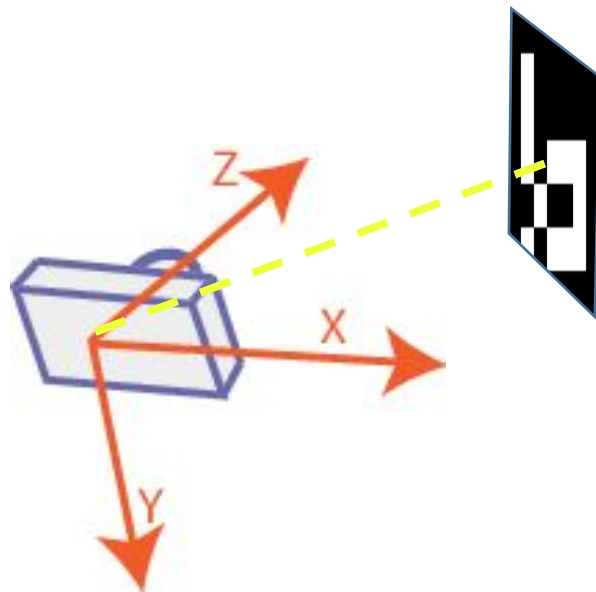
- 在101教室的投影幕前面區域
(無人機要會上下左右位移和左右旋轉)
- 要可以鍵盤控制



Rotation

- (1) rvecs \longrightarrow rotation matrix R
- (2) 用 R 乘以 Z 軸 $(0, 0, 1)$ 得到 Z'
- (3) 將 Z' 投影到 XZ 平面得到向量 V
- (4) 求出 Z 與 V 的夾角
- (5) rad轉換成degree

- `dst = cv2.Rodrigues(src)`
- `math.atan2(y, x)`



注意事項

- distance和dist的單位為公尺！
- 所以在測試時，建議distance和dist的數字**不要設超過"1"！**
- 撰寫自動飛行的程式碼時，**一定也要有 keyboard control** 功能，且要有最高優先權，確保自動飛行狀況不佳時仍能手動控制。