PID 介紹

Theory

通常先把 P, I, D 設為 0。

調完 P 之後再依次開 I, D

output：速度

input：當人机和目標位置的差

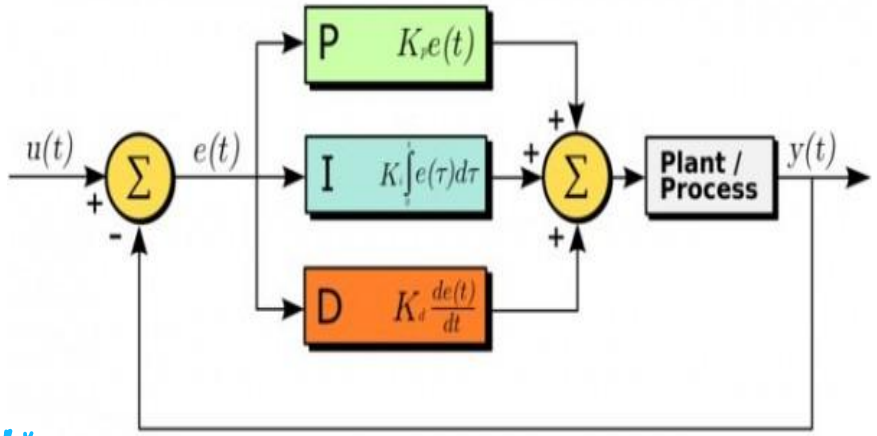
$$u = K_p e + K_i \int_0^t e\,dt + K_d \frac{d}{dt} e$$

P　I　D

Proportional Term　Integral Term　Differential Term

比例項　累積過去的誤差　未來的誤差

（PN小，開得慢）



目標位置

離目標還很遠時

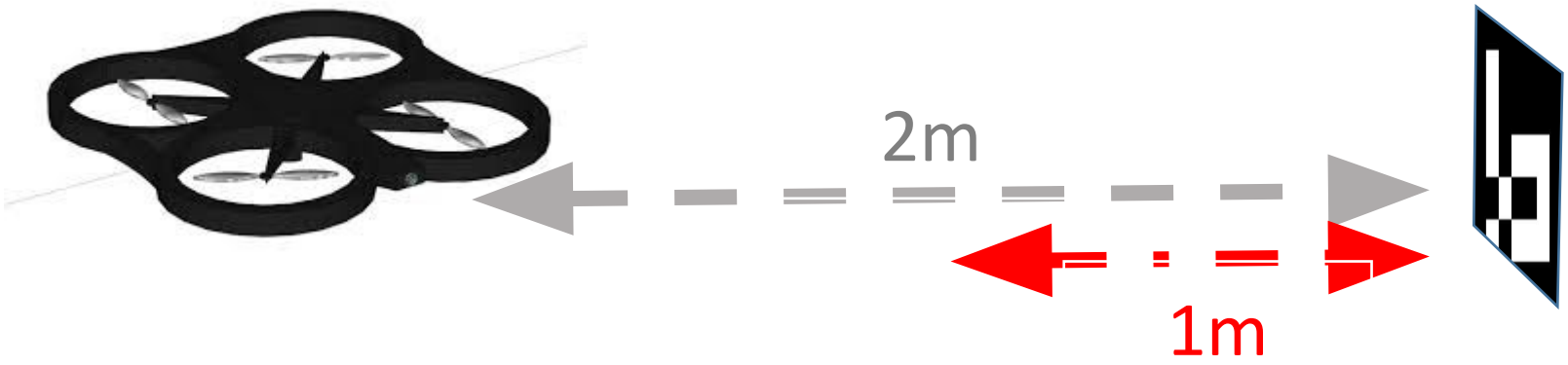
速度大

往前飛，越來越接近目標後，
誤差變小，速度也變小

速度小

$$u = K_p e$$

Step1:



欲修正的誤差(error): 2m – 1m = 1m

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

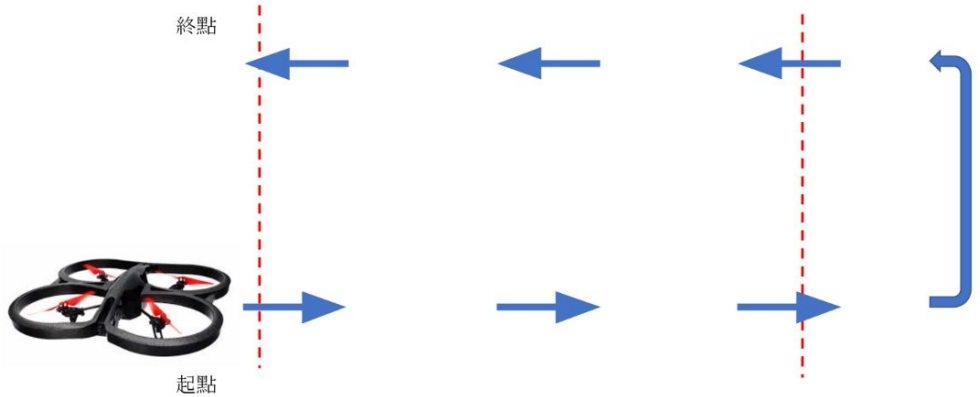
1. 先把I, D設為0, 先調整P
2. 再調整I, D

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

旋轉向量

(1) rvecs ⟶ rotation matrix R

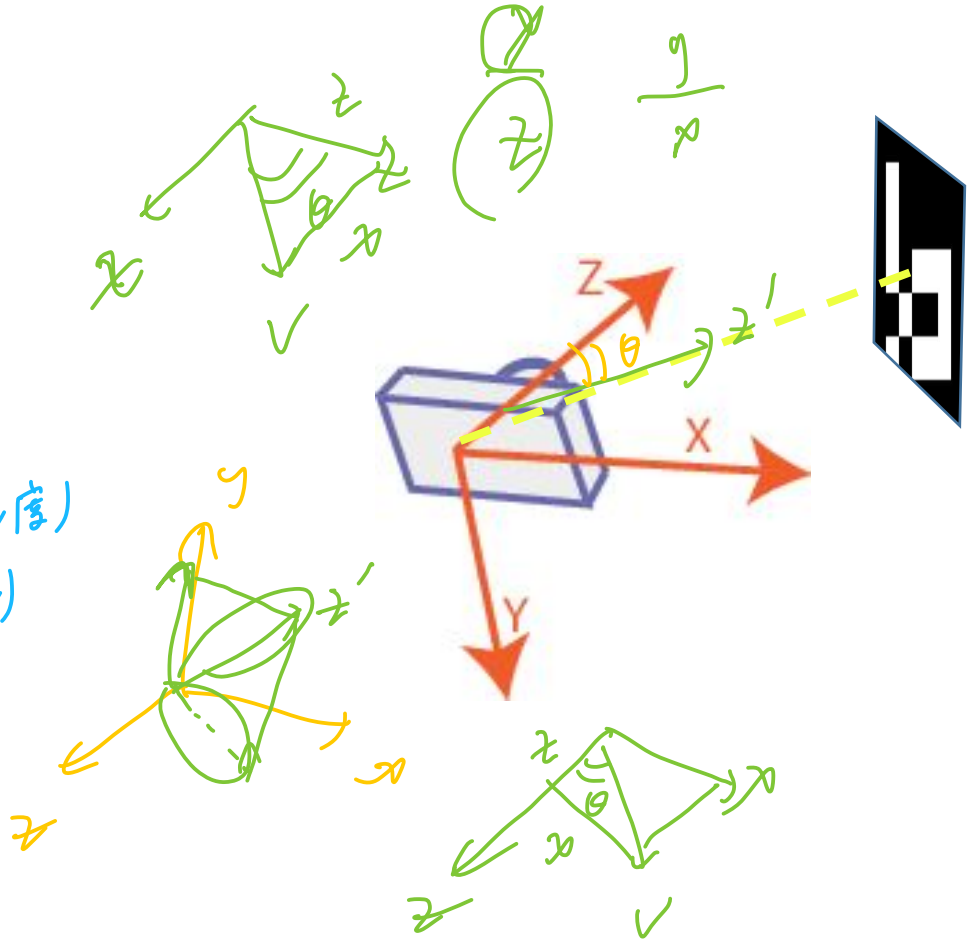
(2) 用R乘以Z軸(0, 0, 1)得到Z'

(3) 將Z'投影到XZ平面得到向量V

(4) 求出Z與V的夾角 (aruco旋轉了幾度)

(5) rad轉換成degree (再付給無人机)

- 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。