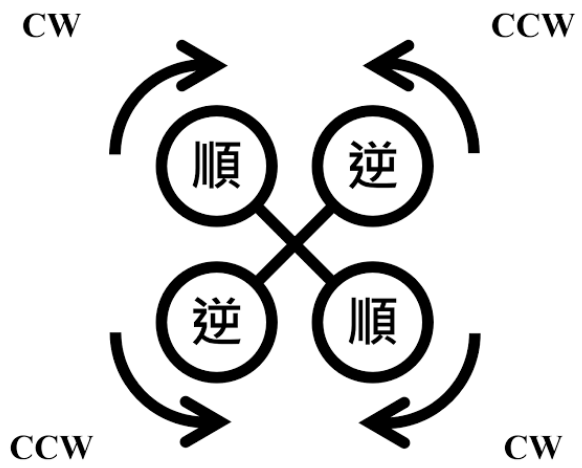
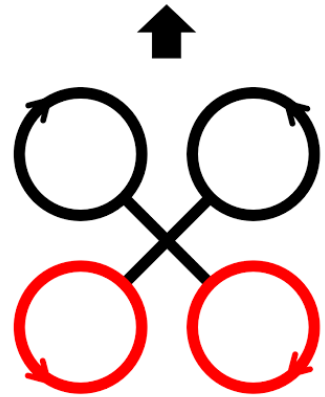
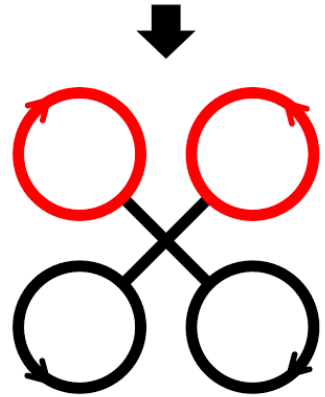
Motion Planning

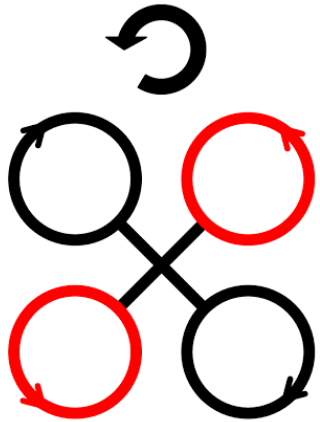




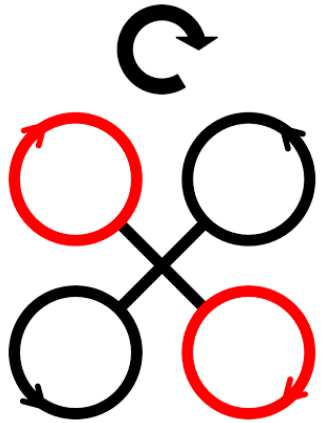
向前運動



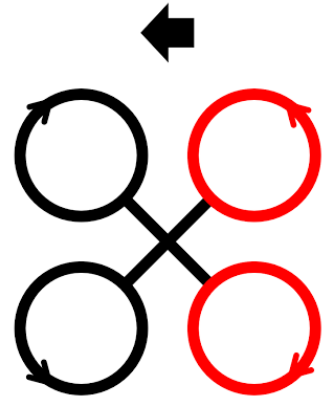
向後運動



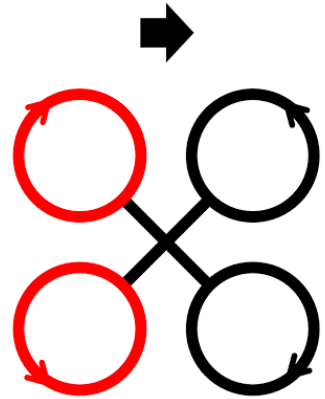
逆時針旋轉



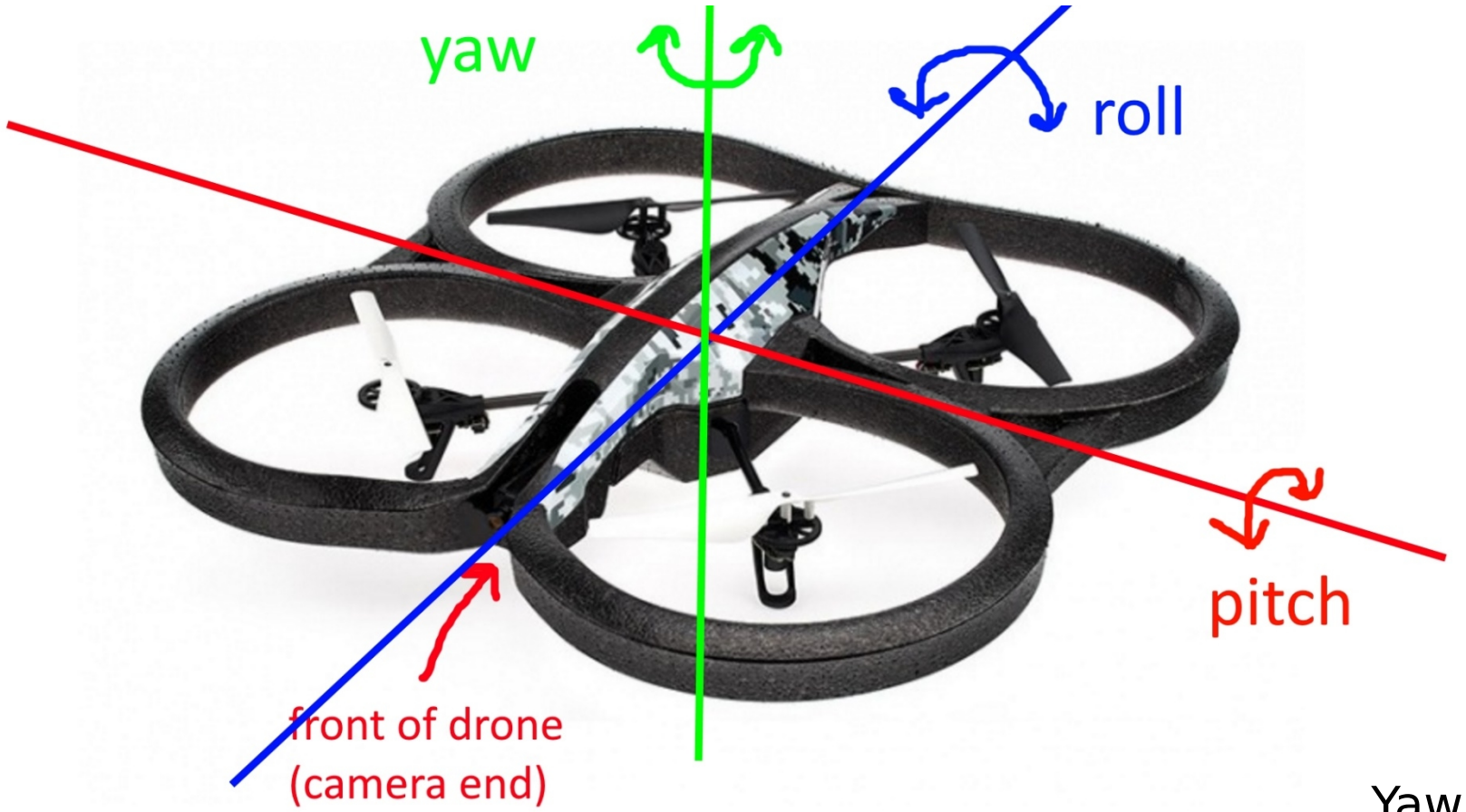
順時針旋轉



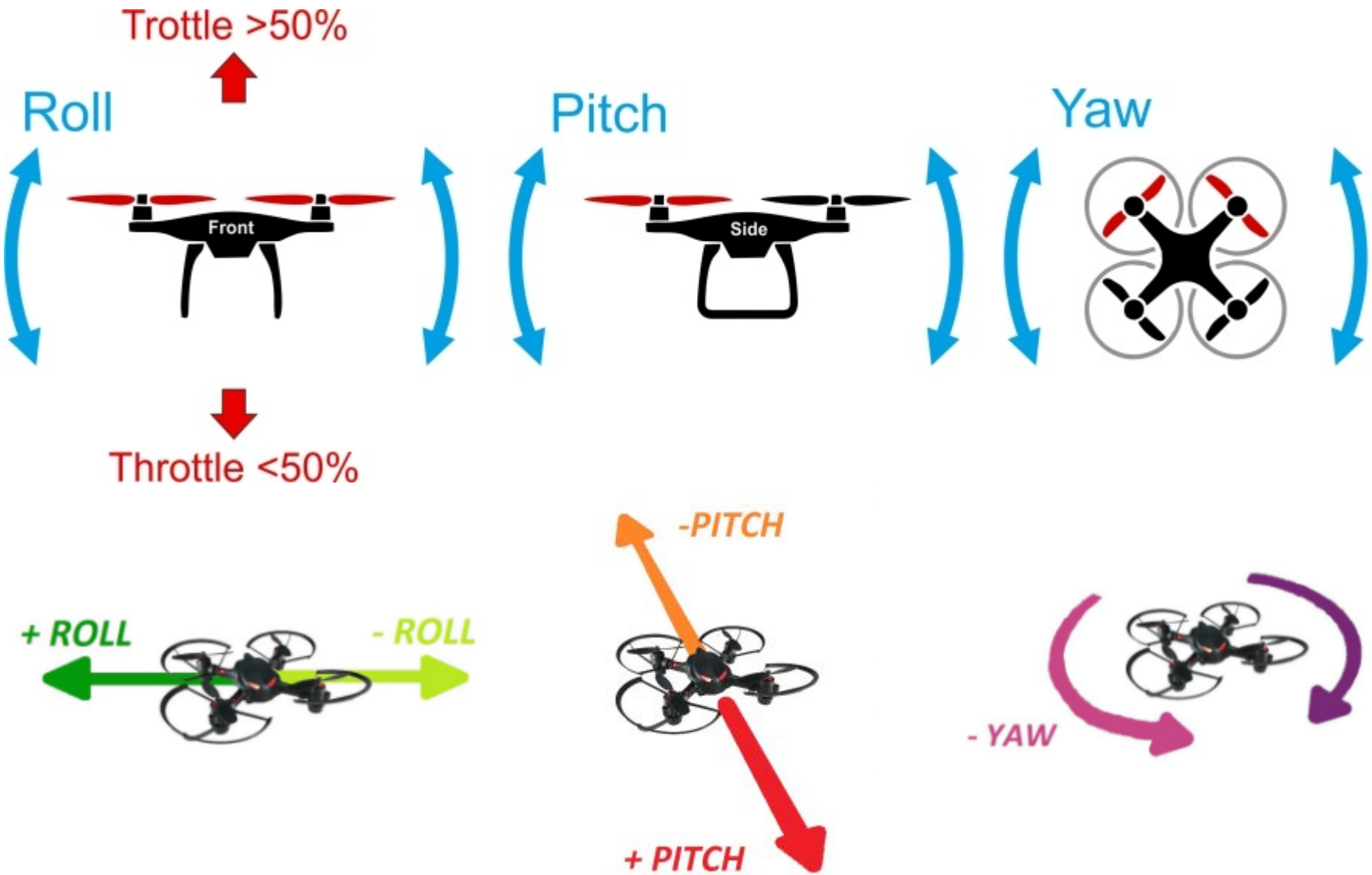
向左運動

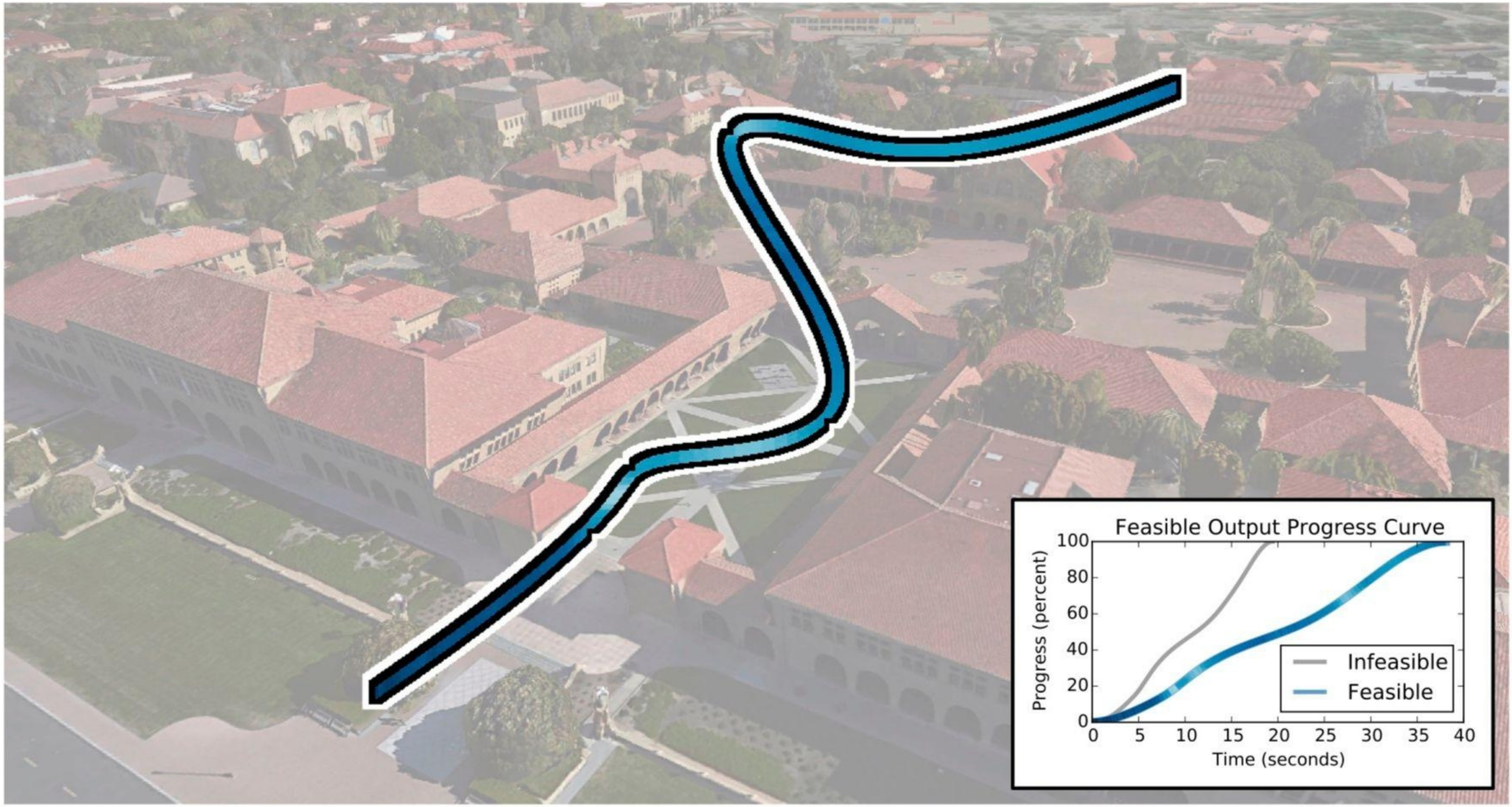


向右運動



Yaw： 繞物體的 z 軸旋轉
Pitch： 繞物體的 y 軸旋轉
Roll： 繞物體的 x 軸旋轉

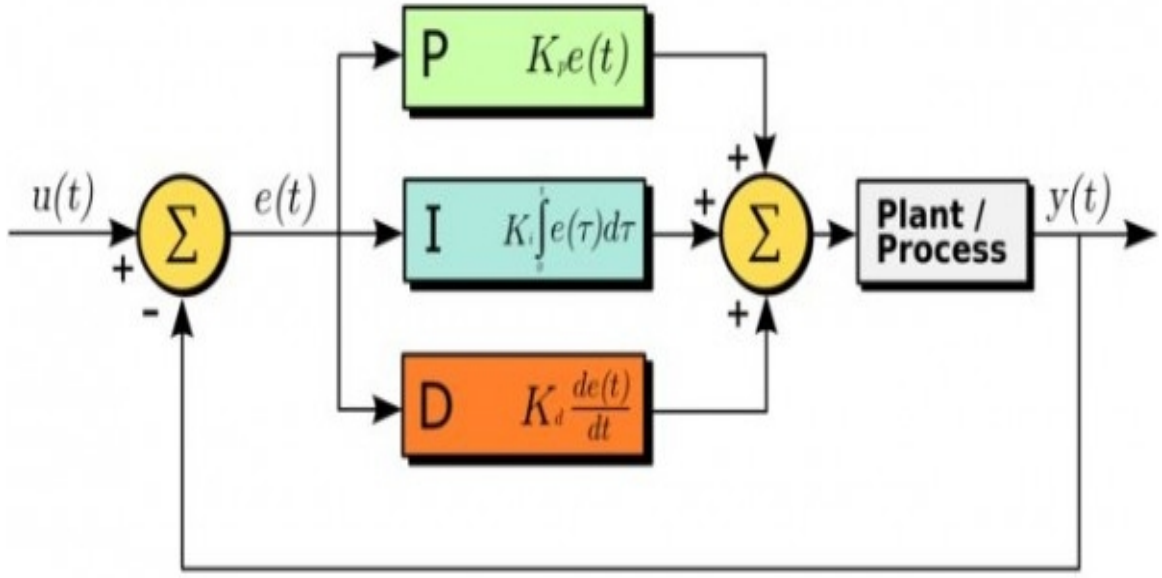

Trottle >50%
Roll
Pitch
Yaw
Front
Side
Throttle <50%
+ ROLL
- ROLL
-PITCH
+ PITCH
- YAW
+ YAW


Feasible Output Progress Curve
Progress (percent)
Time (seconds)
Infeasible
Feasible
0
20
40
60
80
100
0
5
10
15
20
25
30
35
40

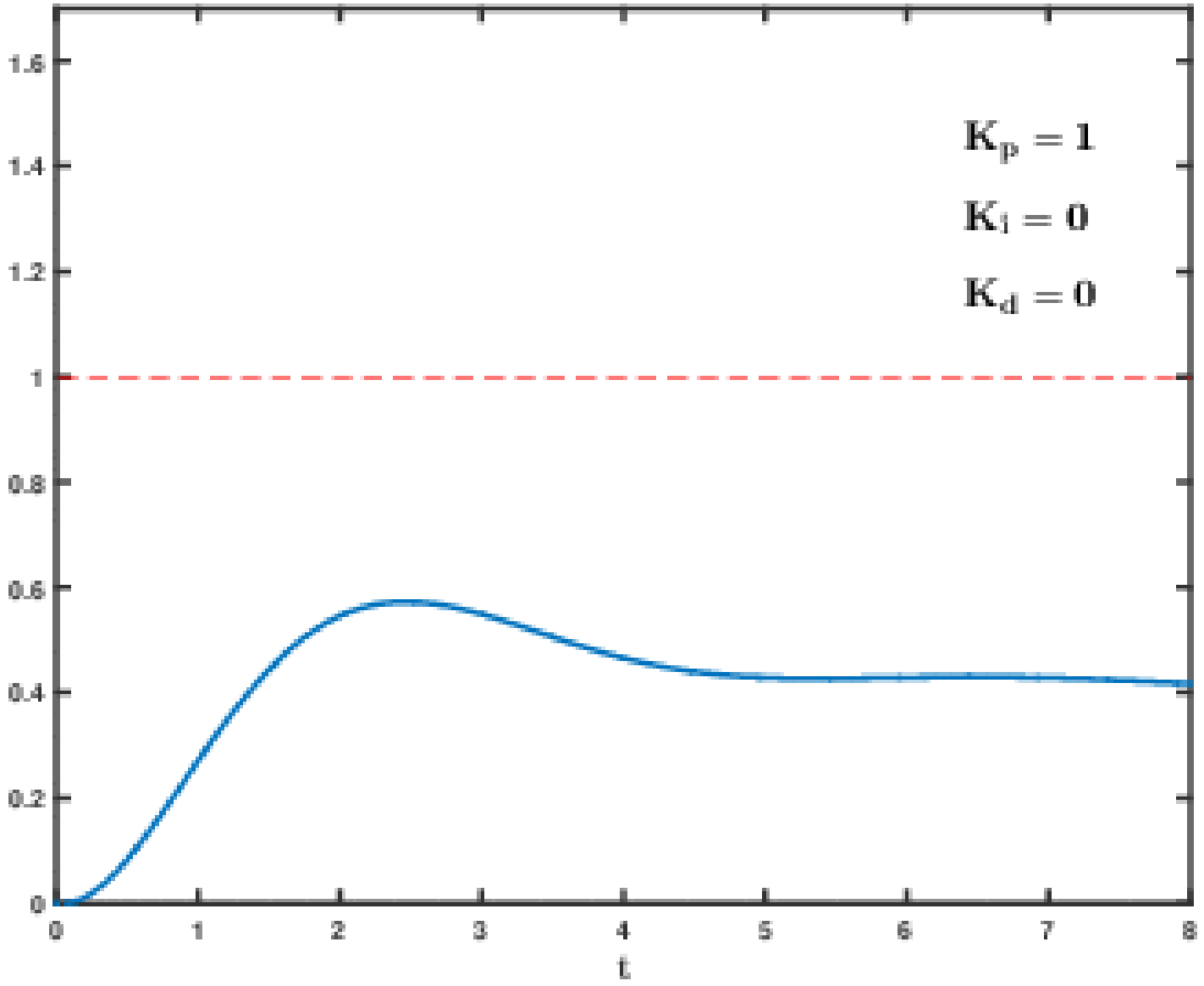
PID Control

Theory

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



調整方式	上升時間	超調量	穩態誤差	穩定性
↑ K_p	減少 ↓	增加 ↑	減少 ↓	變差 ↓
↑ K_i	小幅減少 ↘	增加↑	大幅減少↓↓	變差↓
↑ K_d	小幅減少 ↘	減少↓	變動不大→	變好 ↑


$K_p = 1$
$K_i = 0$
$K_d = 0$
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
1
2
3
4
5
6
7
8
t

Algorithm

```
#Note: Set dt to small value, tuning constants Kp, Ki and Kd to appropiate values, and goal_speed to the desired flight speed.
 previous_error = 0
 integral = 0
 while true{
   actual_speed = getGroundSpeed()
   error = goal_speed - actual_speed
   integral = integral + error*dt
   derivative = (error - previous_error)/dt
   output = Kp*error + Ki*integral + Kd*derivative
   previous_error = error
   setThrottleLevel(output)
   wait(dt)
}
```

Landing


N
W
E
S
H
crazepony


H

Lab06

PID Control

PID Control (100%)

- 目標 ： 調整 PID 參數使無人機能流暢穩定地移動，且在追蹤 marker
  時能保持一定距離並讓 marker 在畫面中間

- 評分: １～１０分由助教依各組無人機表現狀況給分，而實際總
  分則為７０＋得分＊３


Camera Image
Camera Image_t

Midterm Project