Introduction 簡介

Does not cover microcontroller 不包含微控制器

Demo 展示

Body design Physics 機構設計原理

Sensor & Alignment 感測器與電腦鼠姿態和位置的校正

PID control 比例-積分-微分控制

Speed Profile 速度命令曲線

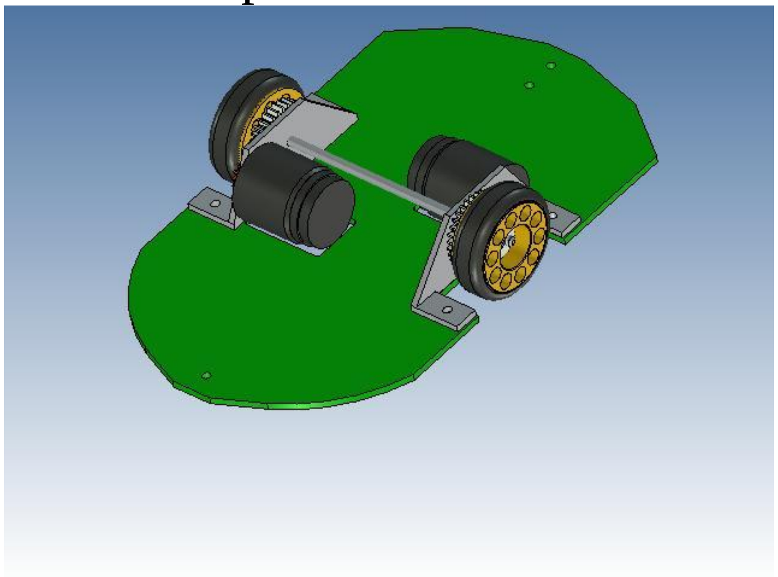
DC Motor sizing 直流馬達的選擇

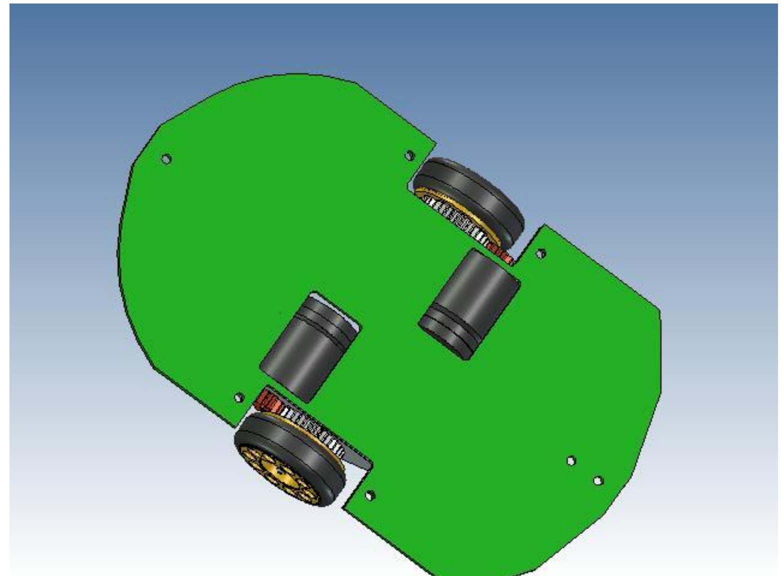
DC Motor driver 直流馬達驅動電路

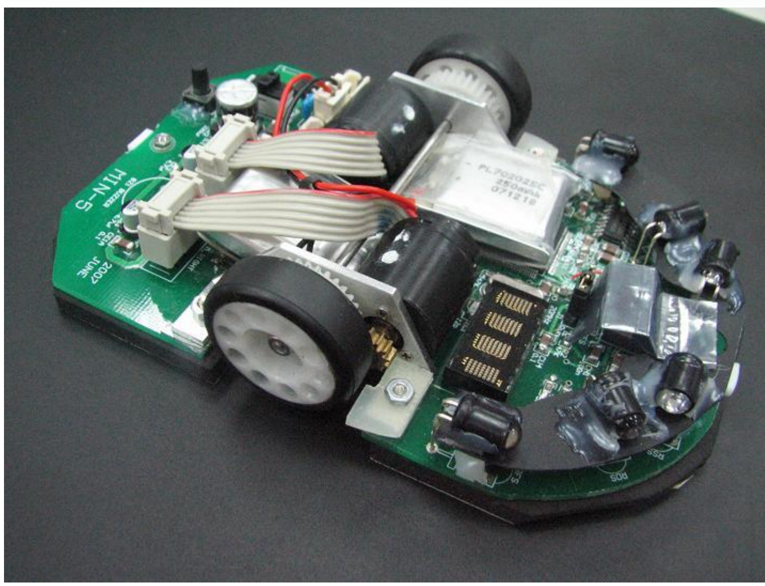
Battery 電池

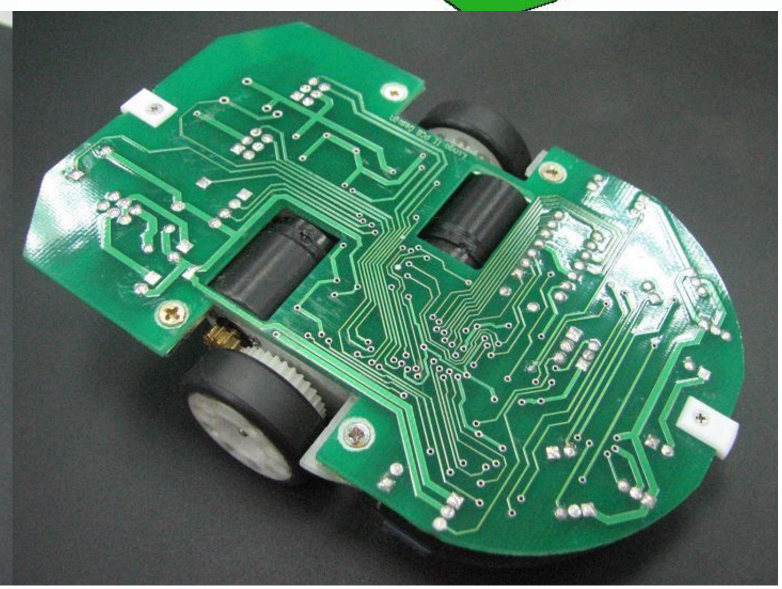
~~Search Algorithm???迷宮搜尋演算法則~~

Photo of top and bottom of Min5









- Minimum 最低 mechanical 機械 design 設計& construction 建造
- PCB 電路板 used as robot body (玻璃纖維)
- PCB is tough 堅韌 and strong
- Good soldering 焊接 to withstand crash 撞擊
- Sensors must be well supported (hot glue 熱膠)

DEMO

1.0 Body Design Physics (物理)(機構設計原理)

- Centre of gravity 重心& weight transfer 重量调动
- Moment of inertia (转动惯量)
- Robot weight (重量) – heavier or lighter?
- Robot wheel (輪子) － bigger or smaller?

1.1 Centre of gravity & weight transfer (重心&重量轉移)

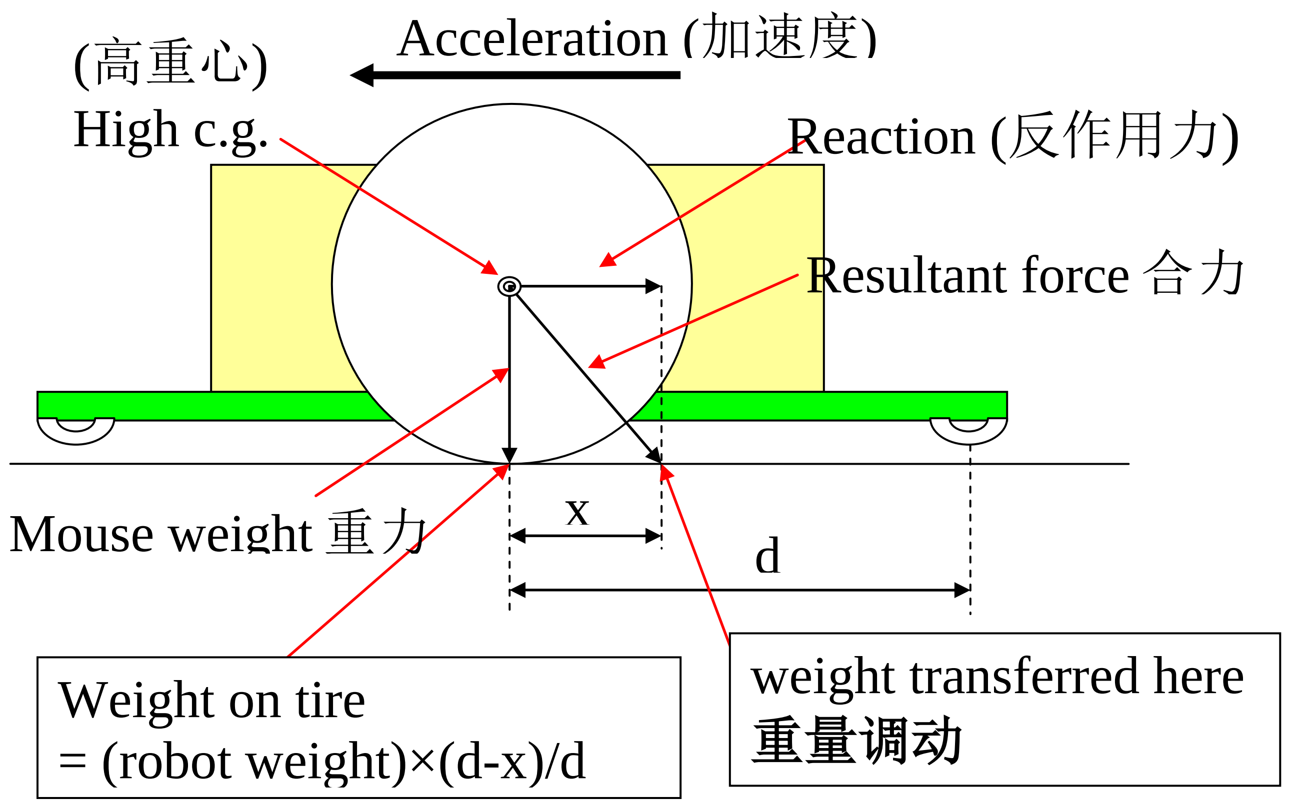


Fig 1: Weight transfer when accelerating

- When Robot is not accelerating/decelerating, all weight is on tires
- If there is no weight on tires -> no grip

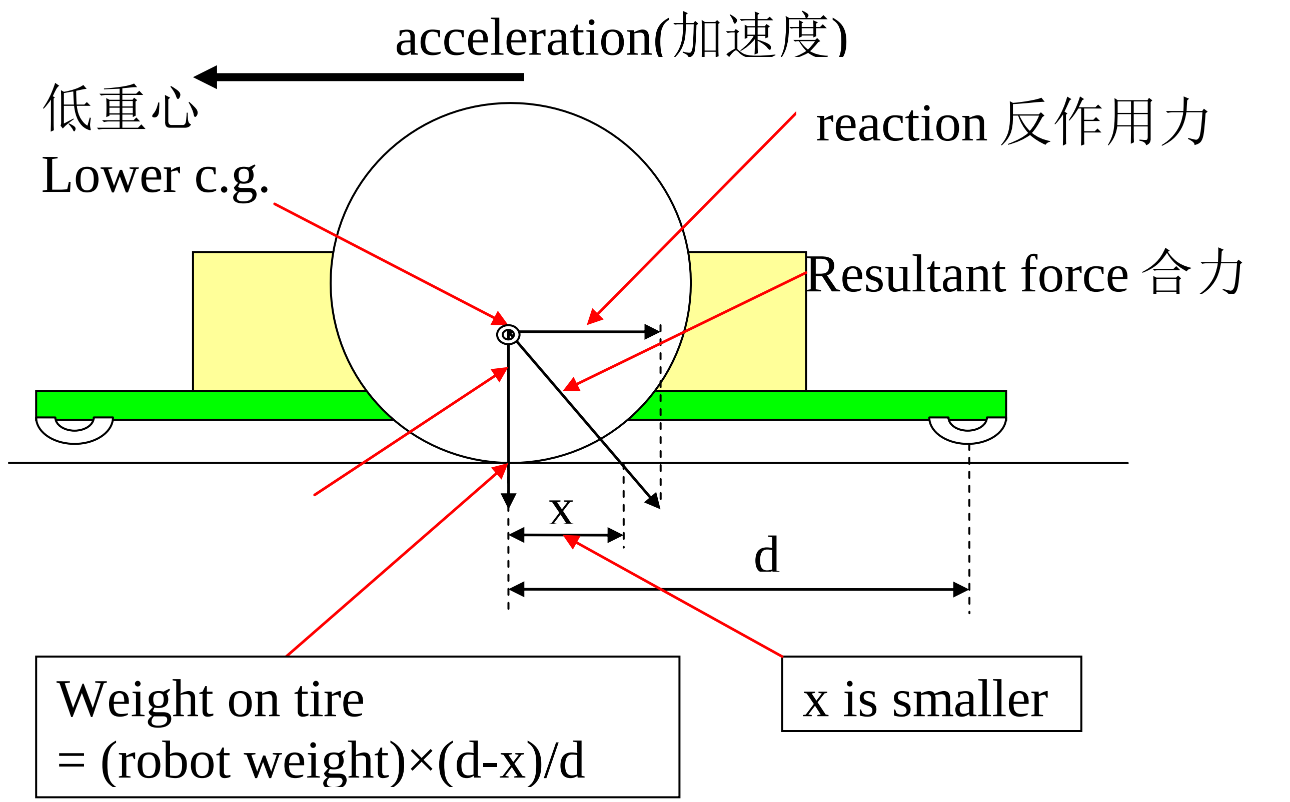


Fig 2: Weight transfer when accelerating (low cg)

- Lighter or slimmer motors to lower c.g.
- More weight on tires, hence more grip for acceleration
- Resultant force acts through c.g.

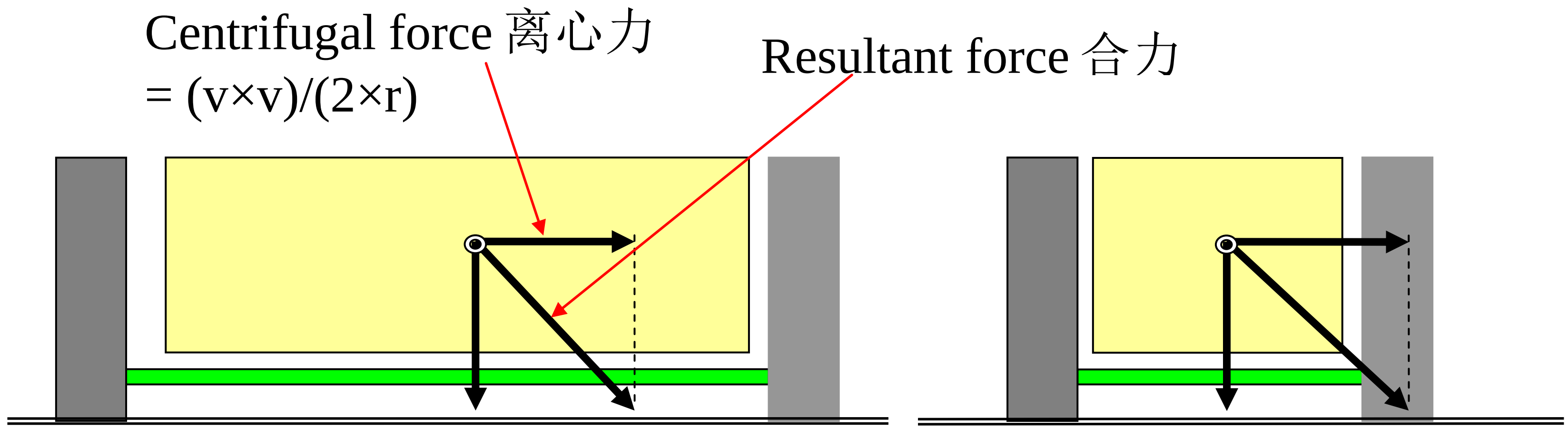
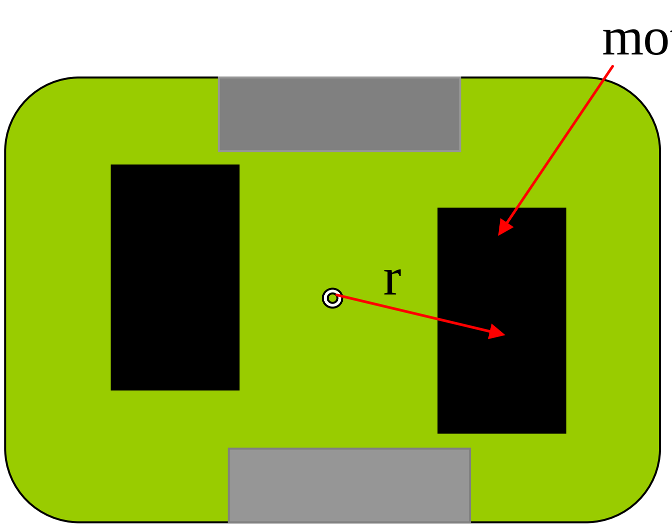
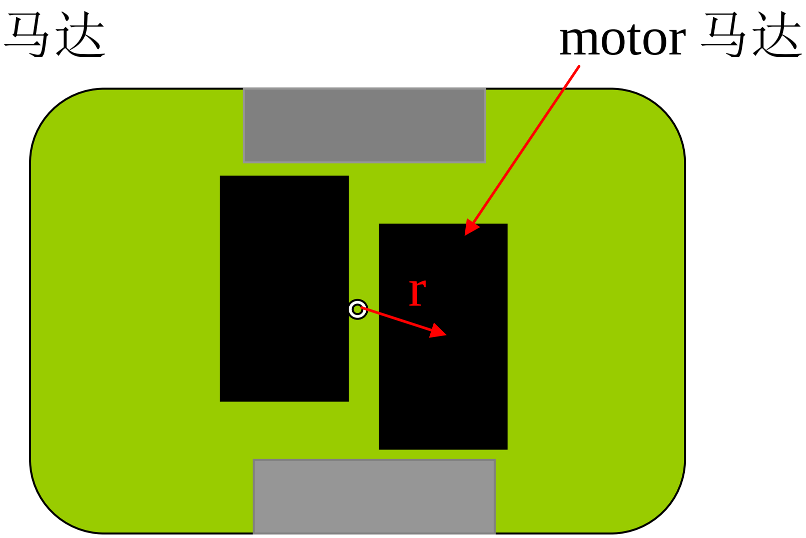


Fig 3 : Weight transfer 重量调 when cornering(電腦鼠向畫面的左邊轉)

- Stepper motor 步進馬達 based micromouse has high c.g.
- Has a higher tendency to roll over 翻轉 if cornering at high speed.
  由於離心力與轉彎速度的平方成正比，倘使迴轉半徑沒有改變的情況下，容易因上圖”車重轉移(重力與離心力的合力)”效應的影響而翻車。
- Always keep the robot c.g. as low as possible
  電腦鼠的重心設計，越低越好。
- Make robot as wide 宽 as possible (愈寬愈不會翻車)
- Make your robot like F1 car

1.2 Moment of inertia (转动惯量)





<u>High moment of inertia</u>

<u>Low moment of inertia</u>

$I = m \times r^2$

m = motor weight

- When moment of inertia is high, rotational acceleration (旋转的加速度) is reduced for the same torque 扭矩.

  Torque = I × α (F = m×a)

  當轉動慣量大時，由於馬達的出力固定，因此在不考慮阻力 而且轉動慣量×旋轉加速度(角加速度) = 馬達輸出力矩 的情形下，電腦鼠的旋轉加速度(角加速度)會降低，轉彎因此會花更多的時間。

- Keep heavy parts like motors, battery etc as close to robot centre as possible

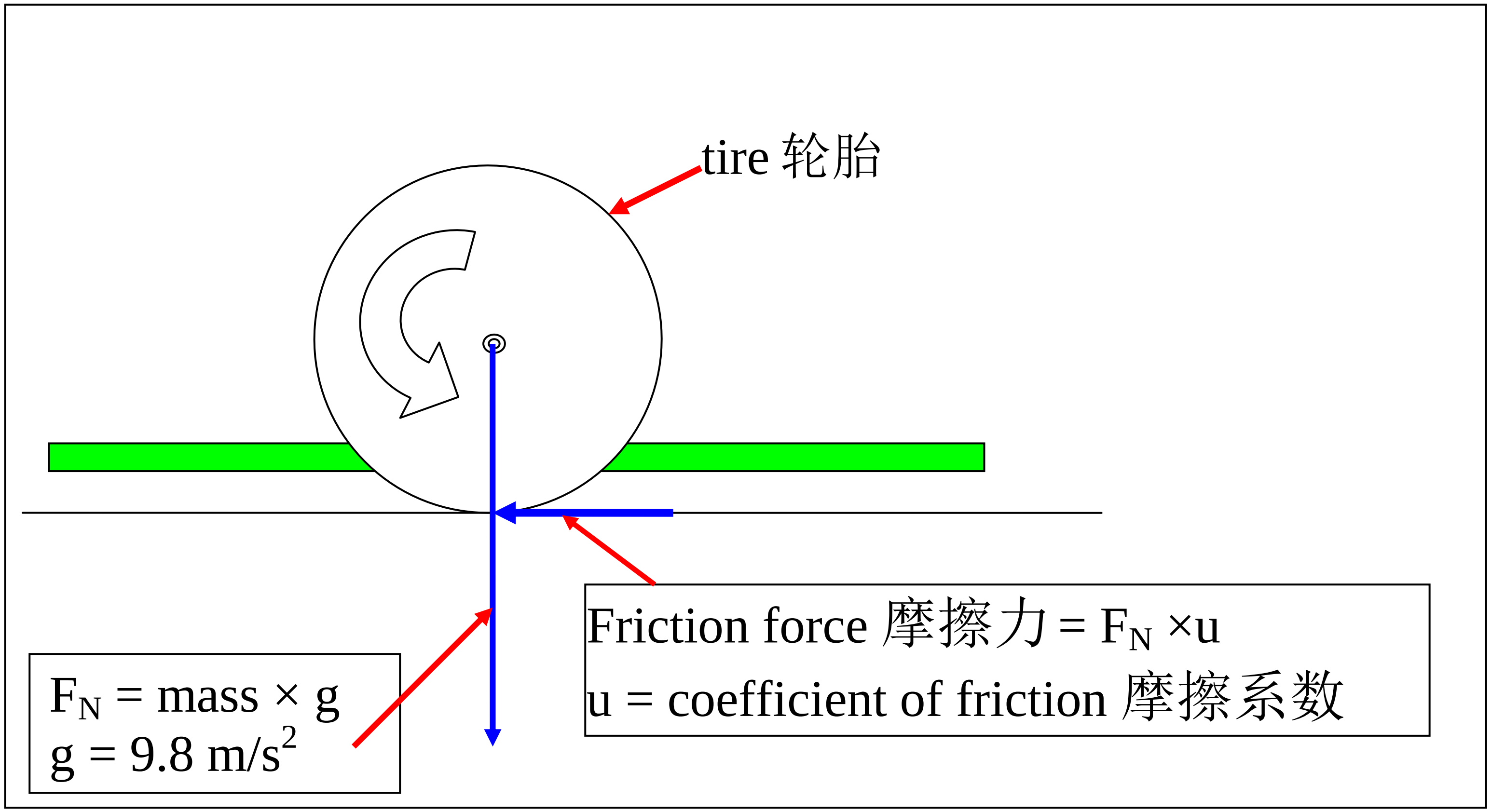
  因此電腦鼠上較重的元件(如馬達、電池等)，愈靠近旋轉中心(重心)擺放愈好。

- If moment of inertia is high, stronger motor is required, -> bigger battery too!

1.3 Robot weight (重量)

Question : Does a heavier mouse provides more acceleration (has more grip 抓地力)?
Question: 重量較重的電腦鼠可以提供較大的加速度嗎(提供較大的抓地力)?

Force (Newton) = mass(kg) × acceleration(m/s^2)



- Force available to propel the robot forward depends on motor power 功率 and friction force 摩擦力.

- If motor is powerful enough, then friction is the limiting factor (限制因素)
如果馬達夠力的話，地面的摩擦力便是主要的限制因素。

- If motor exerts a force greater than friction force, tires will skid (滑行).

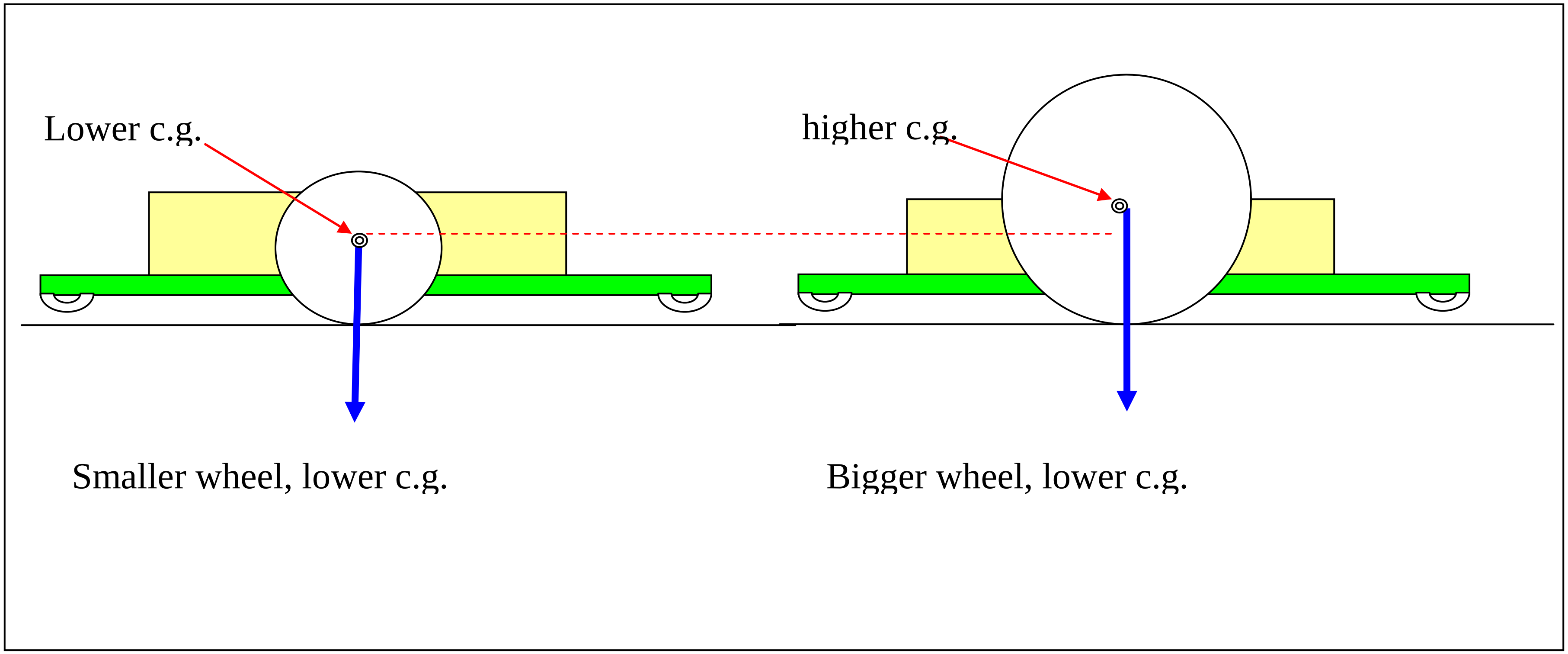
- To prevent skidding, driving force < friction force
  為了避免滑動，馬達出力必須小於地面所能提供的摩擦力
  - Friction $= F_N \times u$
    $= \text{mass} \times g \times u$
  - Therefore acceleration limit = $g \times u$
    因此加速度的限制就是 重力加速度×摩擦係數
  - If u=0.7, then acceleration limit is about $7 m/s^2$.
    假如摩擦係數 u = 0.7，那麼電腦鼠加速度的上限就是 $7 m/s^2$。
- Acceleration is independent of mass. Limited by tires.
  電腦鼠加速度的上限與重量無關

- Keep robot weight low -> reduce motor size -> reduce battery capacity.

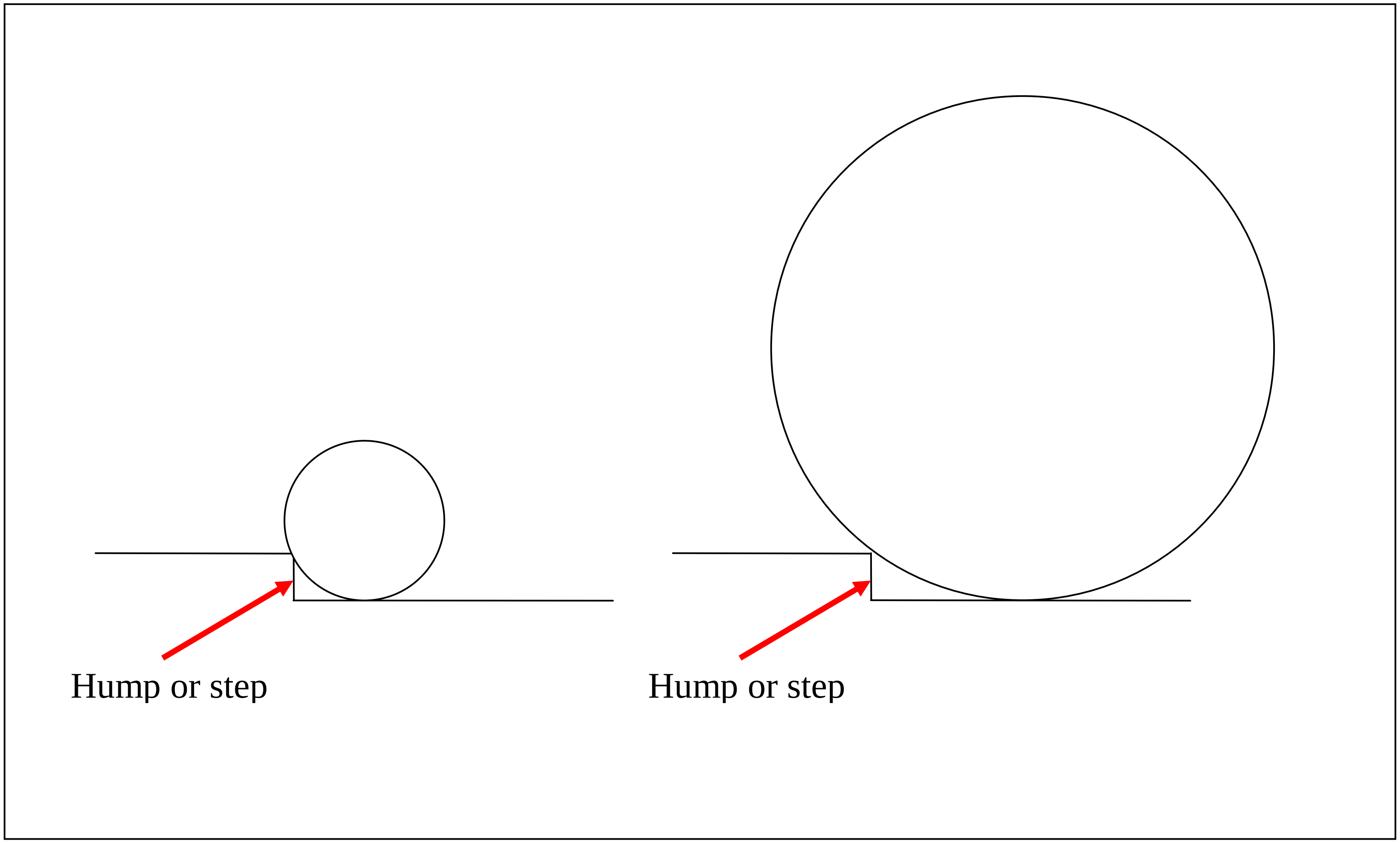
1.4 Robot wheel

Question : A bigger wheel or a smaller wheel?

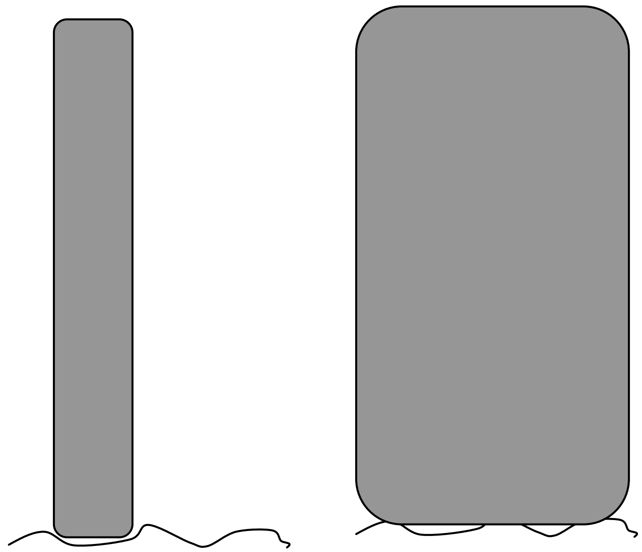
- Tire friction is reduced by dirt (灰塵)
- Clean tires before run
- A smaller wheel would need to rotate more times for the same distance
  - Pick up more dirt(容易積聚較多的灰塵)
  - If maze if dirty, performance will be affected
- Bigger wheels raise the c.g. (大車輪會提高電腦鼠的重心)



- Weight of wheels and shaft 軸 affect the robot's c.g.
- Keep wheels & shaft as light as possible
- Use light and thin material if possible



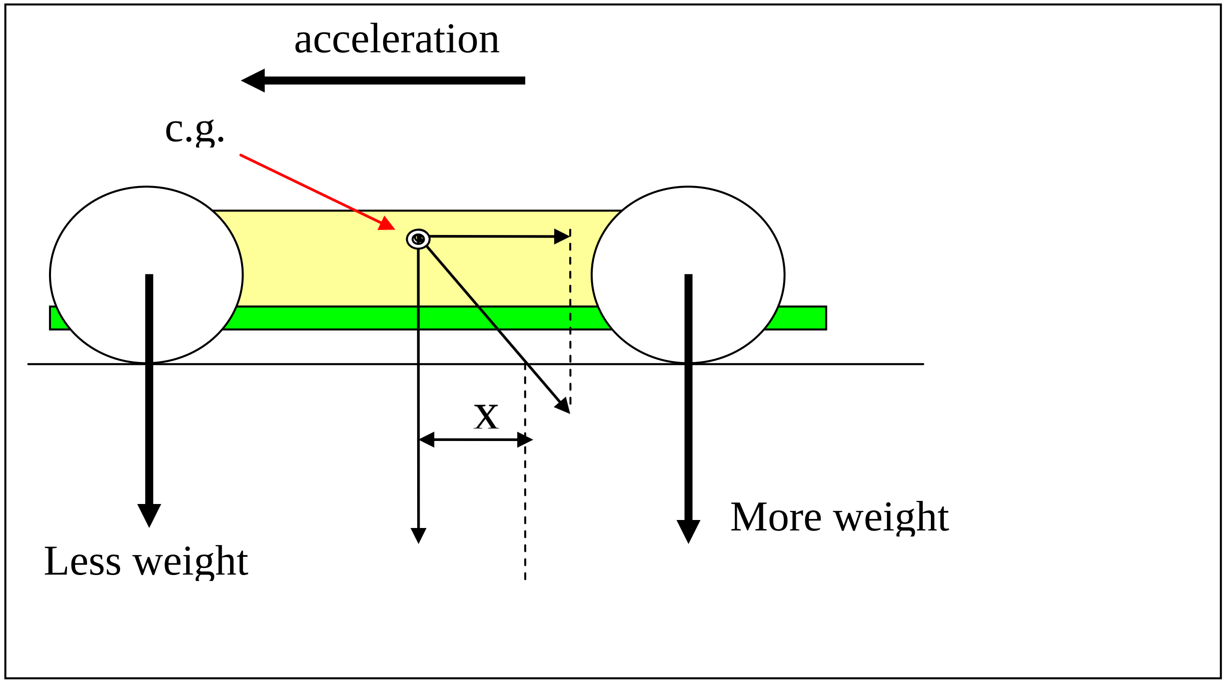
- Big wheel will roll easily over steps or humps(大車輪比起小車輪更容易越過地面的突起處)



A bigger surface allows the wheel to ride over unevenness

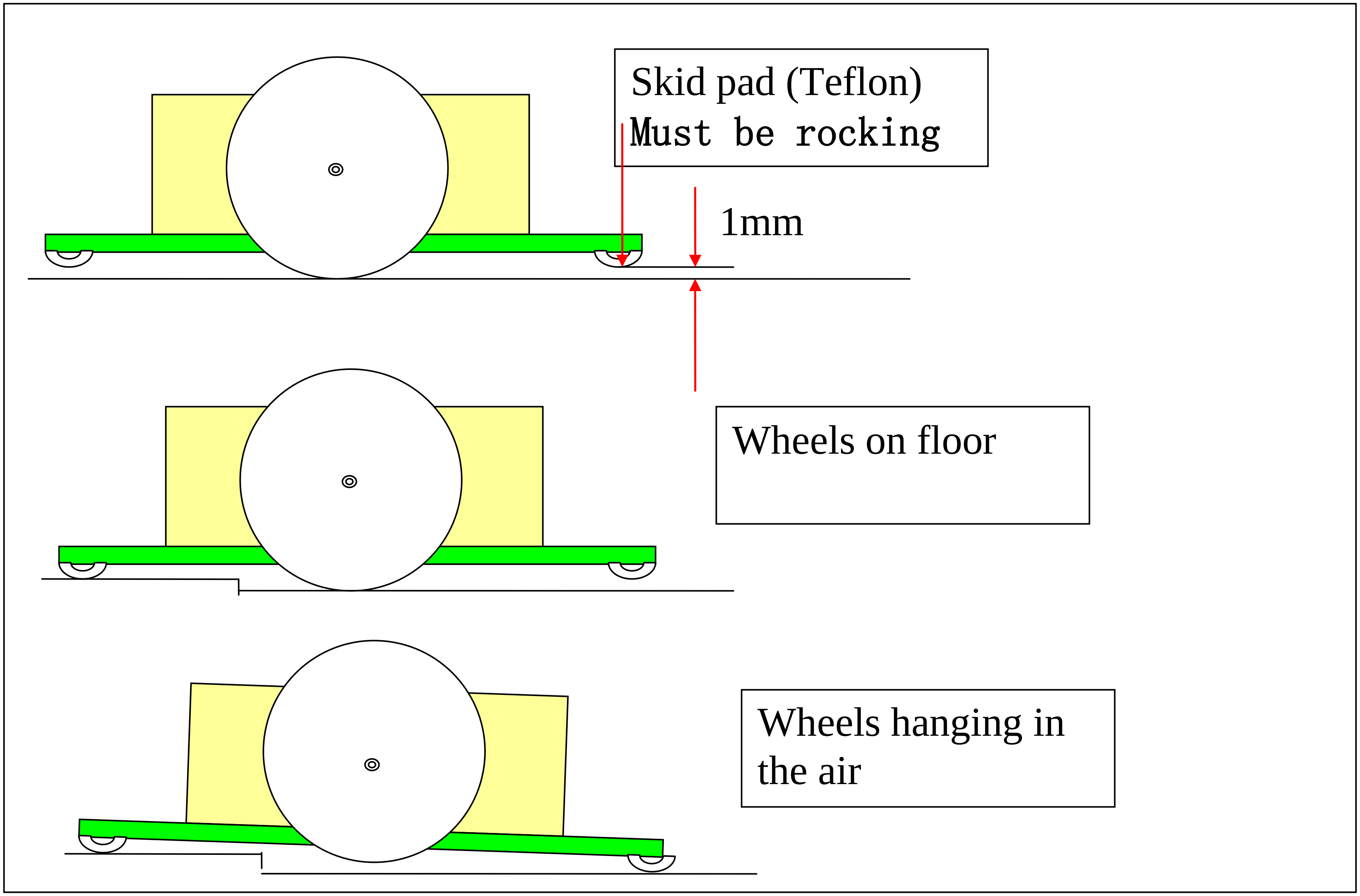
- Wider wheel will also be less affected by unevenness. (寬車輪比起窄車輪較不受地面不平的影響)
- On a flat and clean surface, tire diameter and width has negligible impact on performance. (但在平坦而乾淨的地面上，車輪的寬度與大小對性能的影響很小)
- Too wide and it’s hard to turn

1.5 Four wheel drive ?

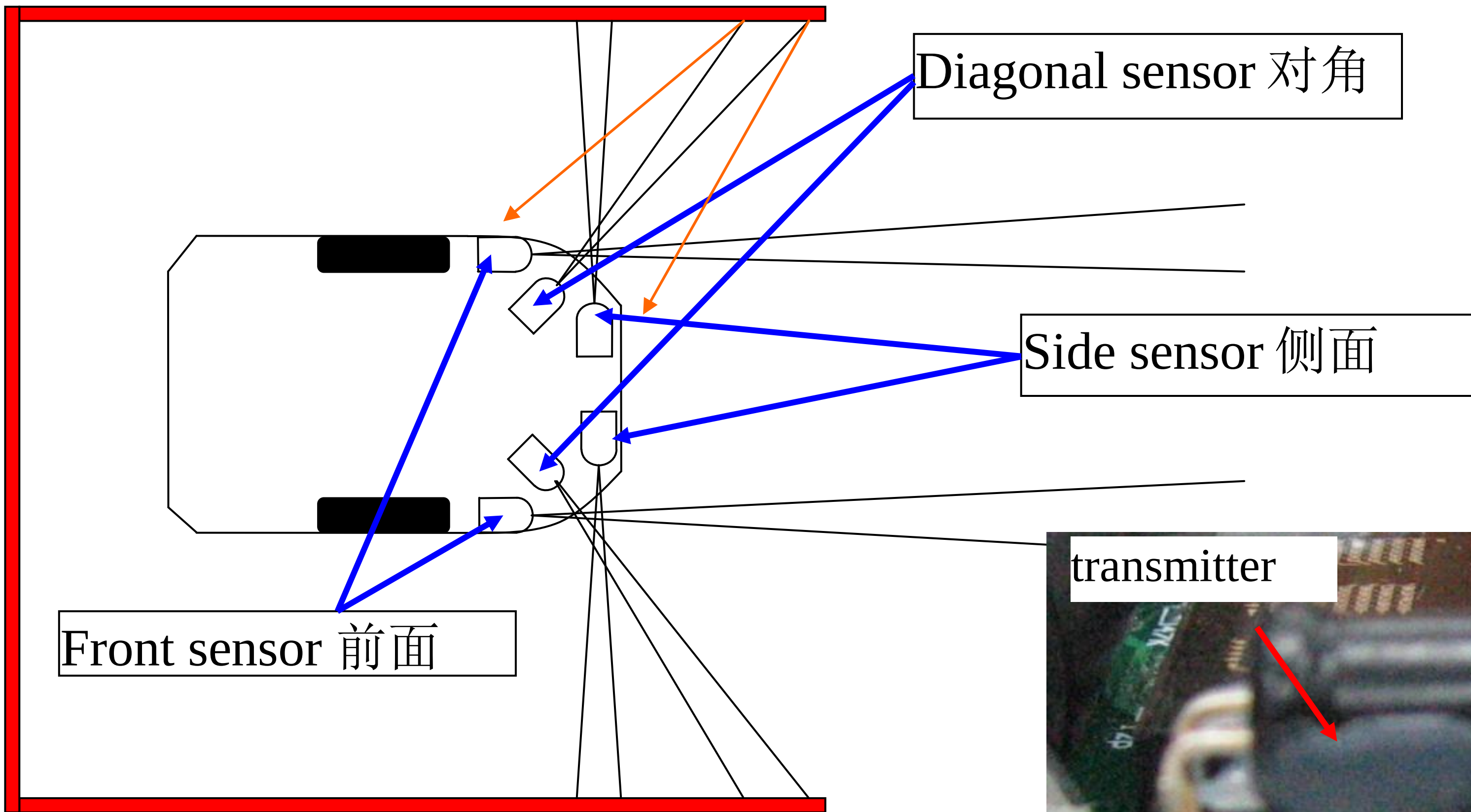


- During acceleration, weight is transferred to back wheels (加速時，重量移轉至後輪)
- During deceleration, weight is transferred to front wheels
- Good for straight acceleration/deceleration (對直線加減速不錯)
- Complex and hard to build (但複雜而不好做)
- My dream mouse !!!

1.6 Skid Pads


Skid pad (Teflon)
Must be rocking
1mm
Wheels on floor
Wheels hanging in the air

2.0 Sensors 感測器 & Alignment 位置的校正



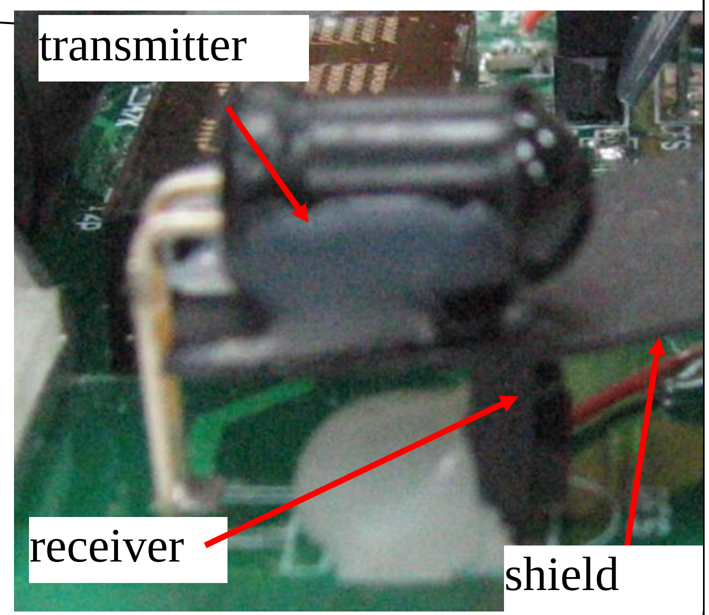


Figure: Mouse sensors layout 佈局

2.1 Sensor circuit and operation

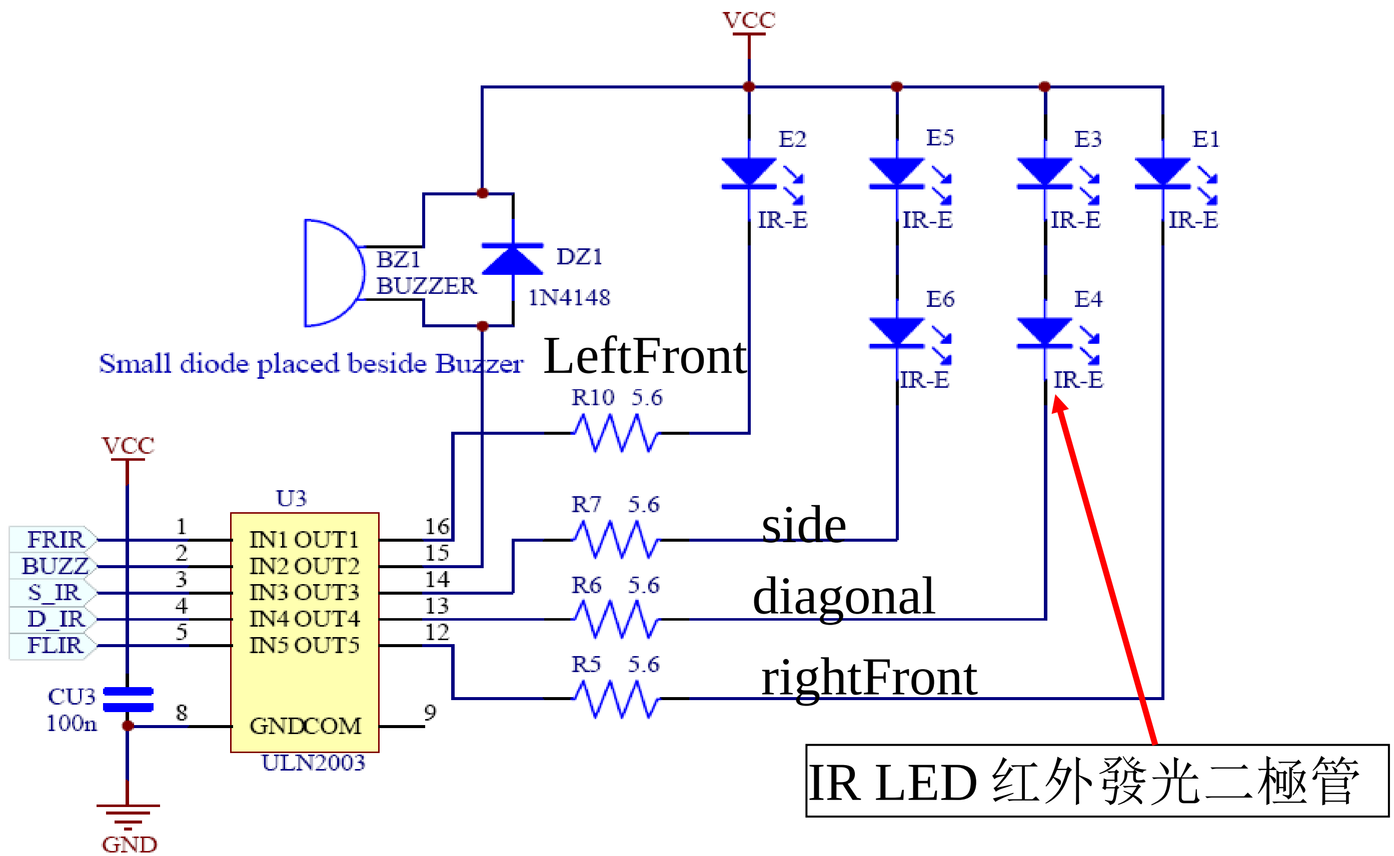


Figure : Transmitter 发送器 circuit diagram

- Left and right front sensors pulsed separately to avoid interference 干擾

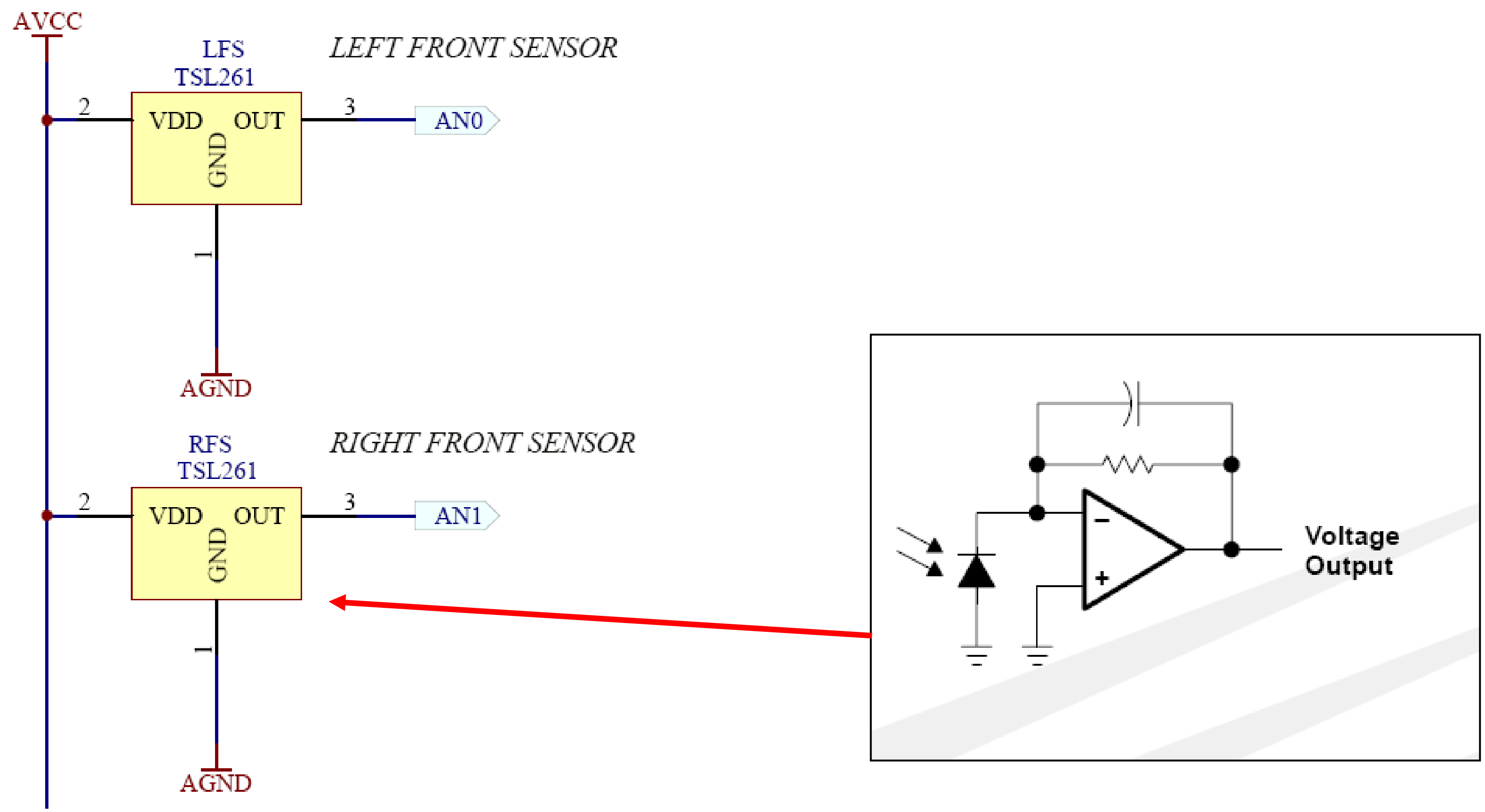


Figure : Receiver 接收器 circuit

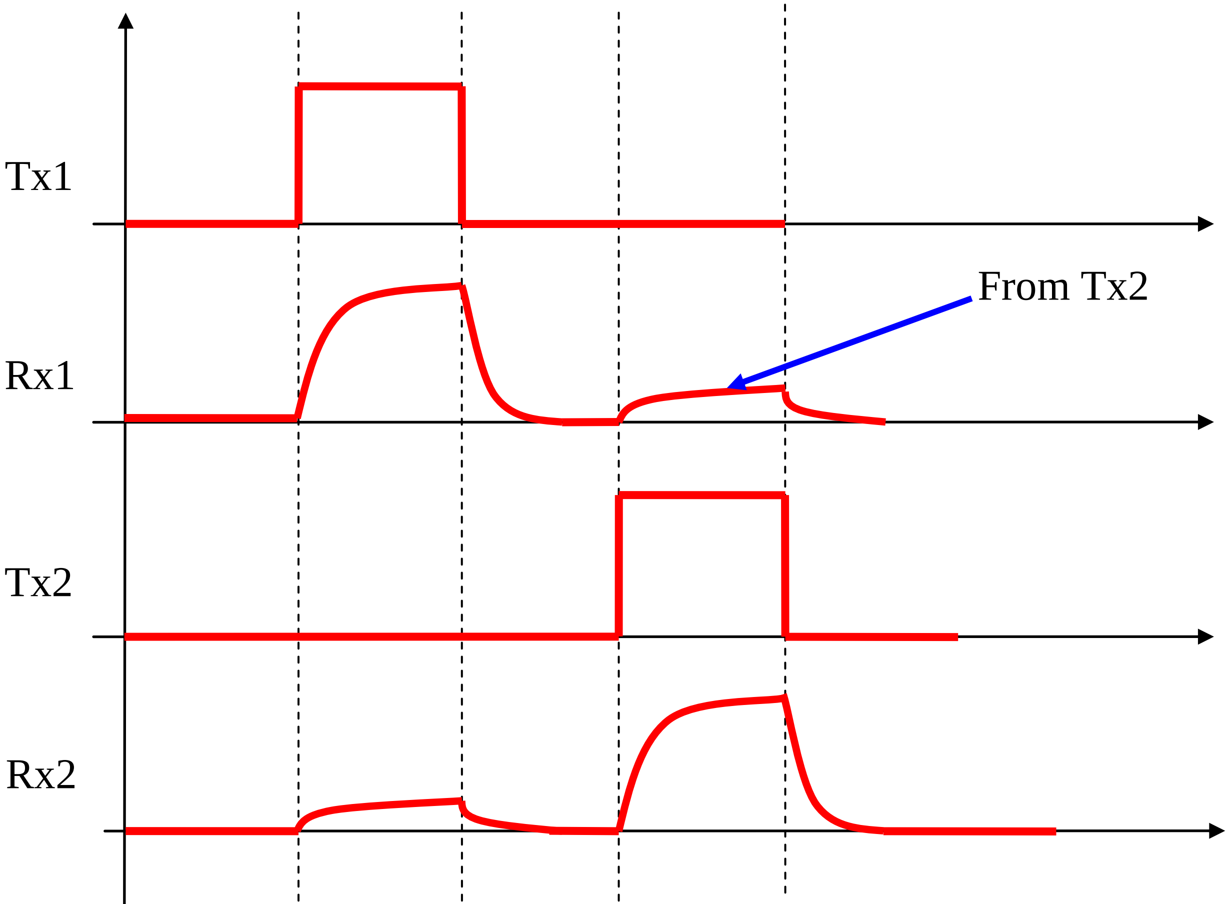


Figure : Timing diagram of sensors' signals(感測器信號的時序圖)

- IR LED 红外發光二極管
- Pulsing allows higher current to swarm interference 干扰
  (脈衝式的信號可以提高電流大小以對抗環境的干擾)
  LED on < 10% of time only -> can stand higher current
- Pulsing minimize interference from neighbouring sensors
- Receiver should be less sensitive 敏感 -> allows high transmitter output
  (接收器不需要太敏感)
- Improve signal noise ratio 信号噪声比率

2.2 Alignment 位置校正

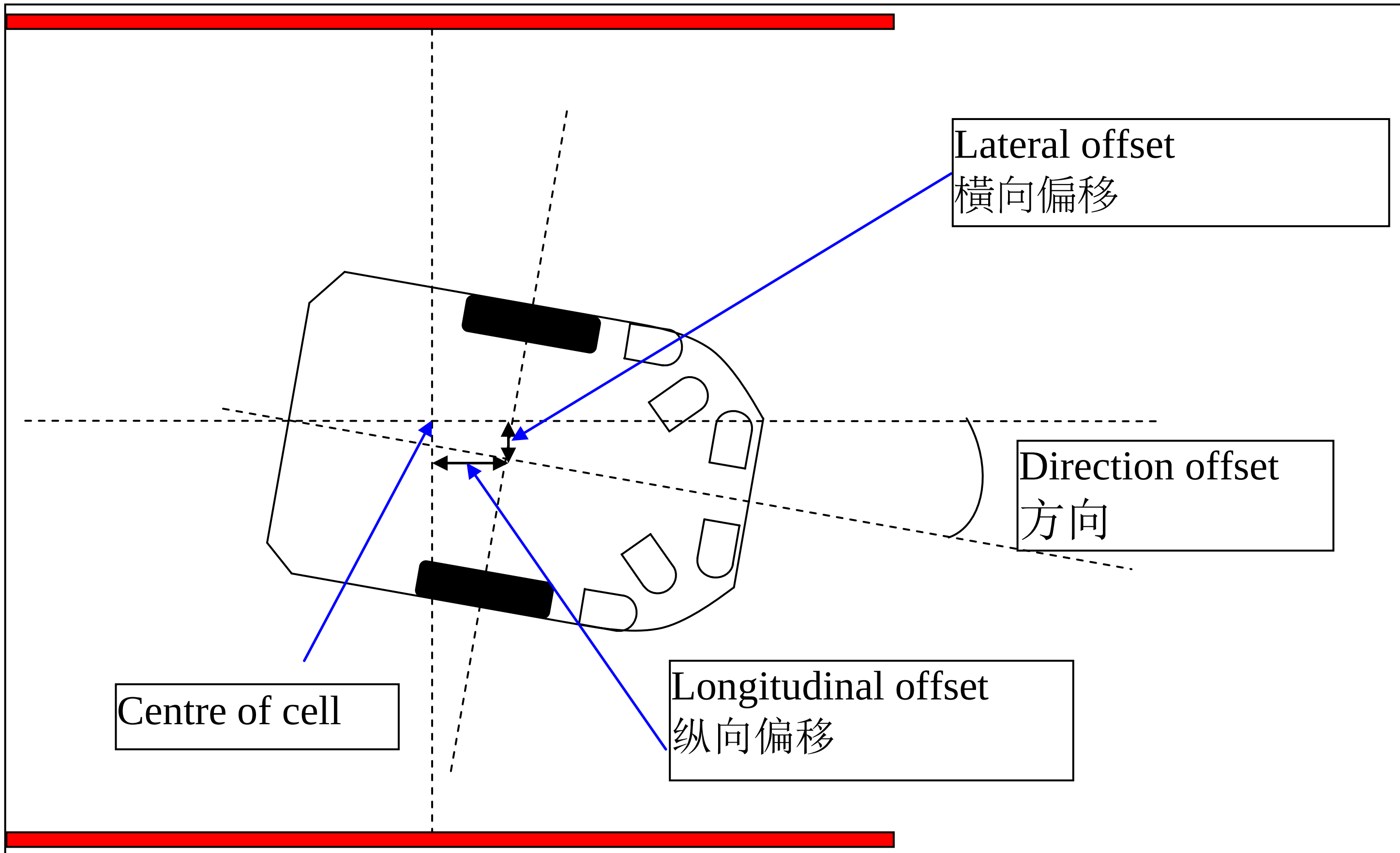
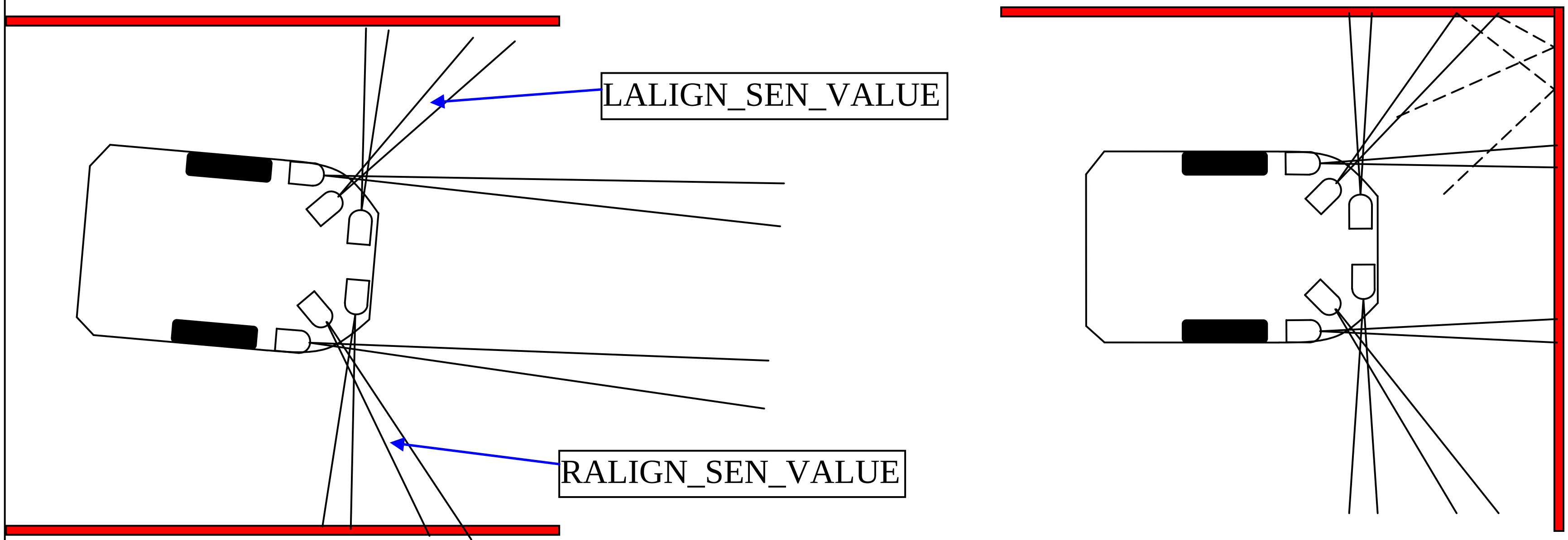
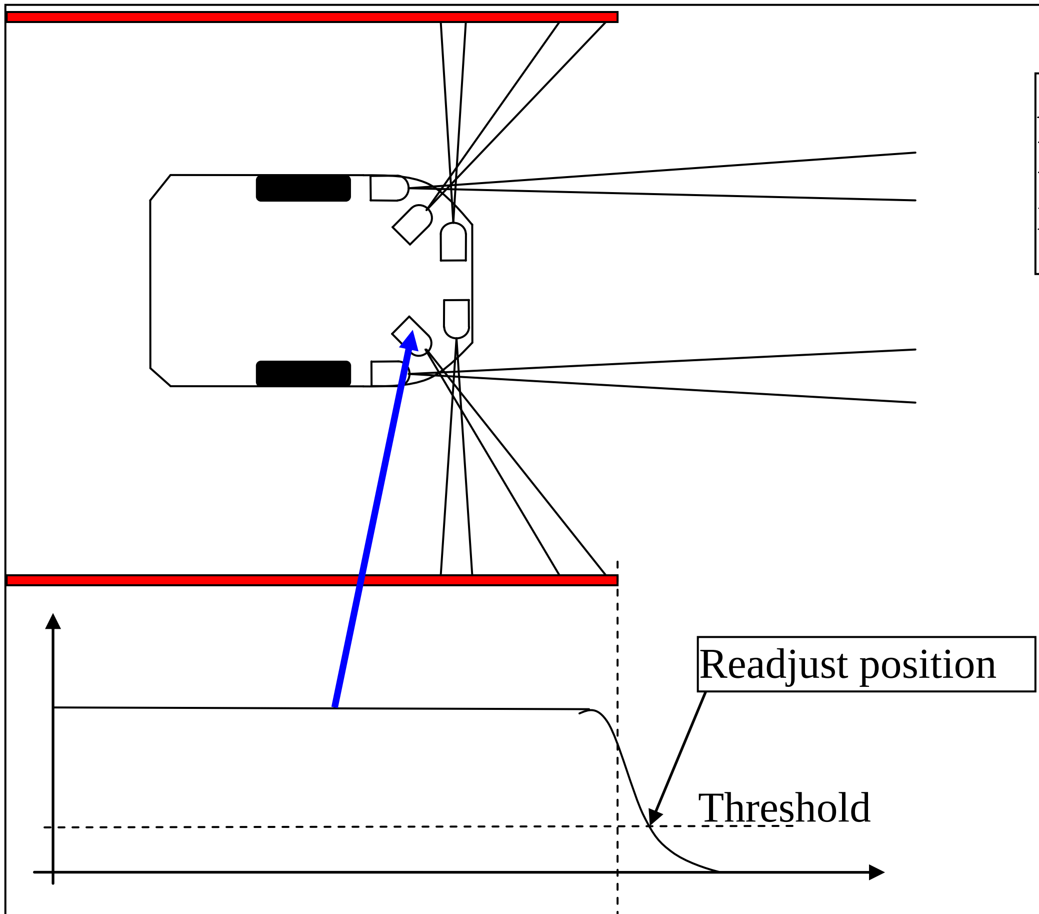


Figure: Types of positional error 位置差錯




```
if( (LALIGN_SEN_VALUE > laSenV3) && (RALIGN_SEN_VALUE < raSenV3)) {
        //Too left, adjust right
        alignSpeed= (LALIGN_SEN_VALUE - laSenV3)*maxAlignSpeed/laNearLeftRange ;
}
else    if( (RALIGN_SEN_VALUE > raSenV3) && (LALIGN_SEN_VALUE < laSenV3) ) {
        // Too right, adjust left
        alignSpeed = -(RALIGN_SEN_VALUE - raSenV3)*maxAlignSpeed/raNearRightRange);
}
```

Figure: Correcting direction 方向 & lateral 横向 offset with diagonal sensor
(利用斜 45 度的感測器來校正電腦鼠方向與橫向上的位置計算誤差)



Assumption :
Lateral and angular offset
is small.

Figure: Correcting longitudinal 縱向 offset with pole 柱 detection 偵測
(偵測迷宮柱的位置來修正縱向的位置計算誤差)

*Almost all turns are preceded by a pole to no-pole transisition

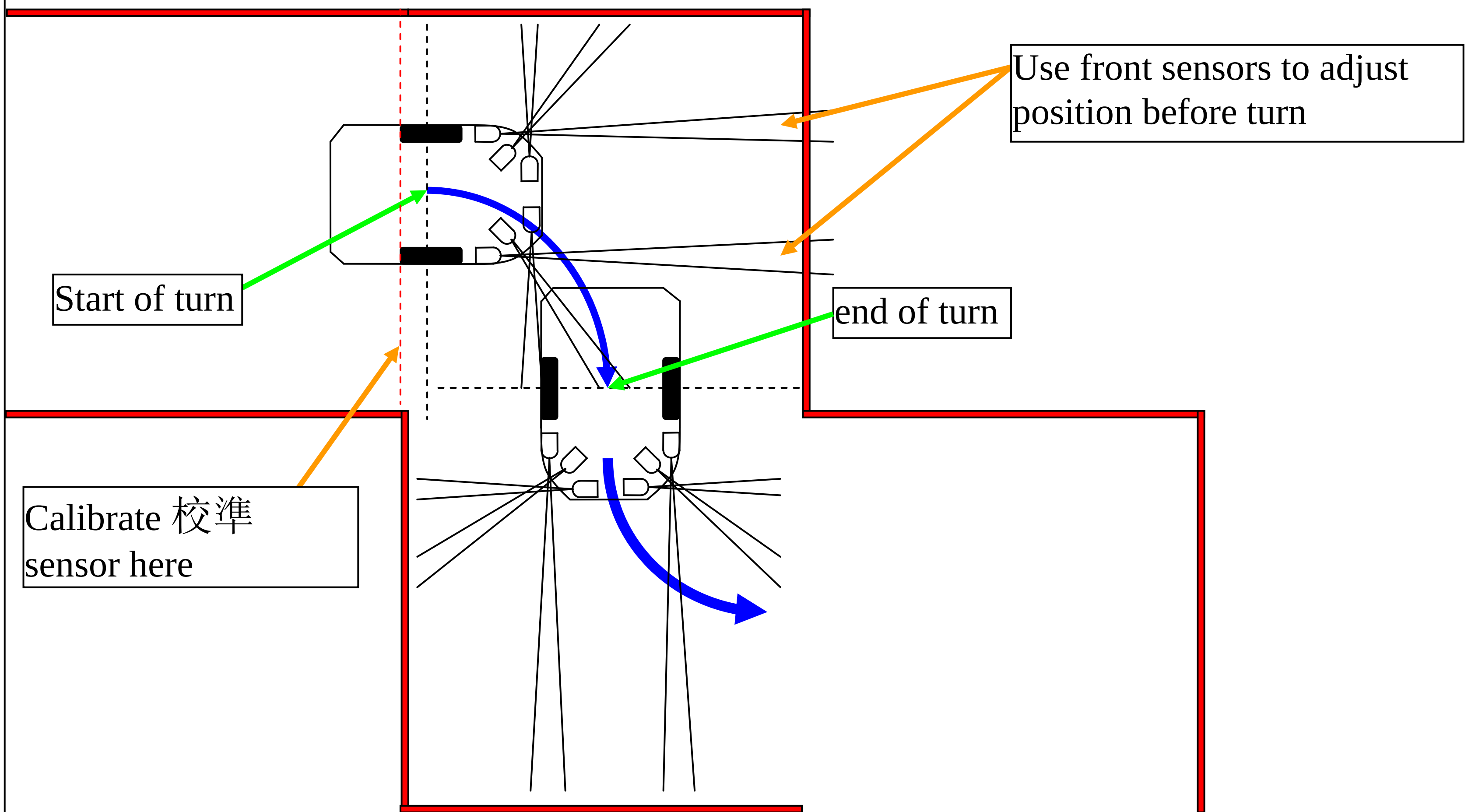


Figure: Correcting longitudinal offset with front wall (exploration)

(轉彎前如果前方有牆，可以利用往前的光感測器來修正縱向的位置計算誤差)

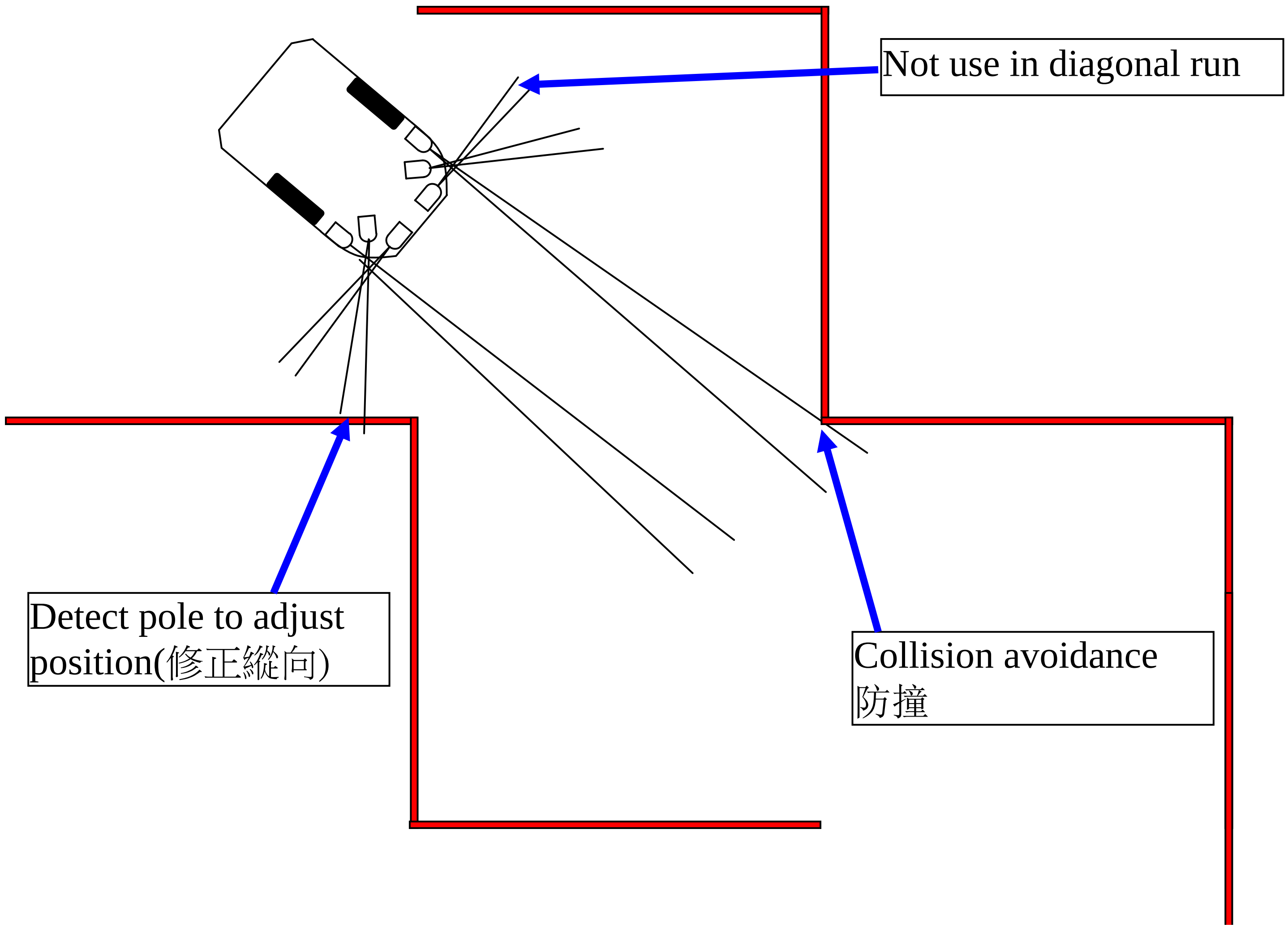
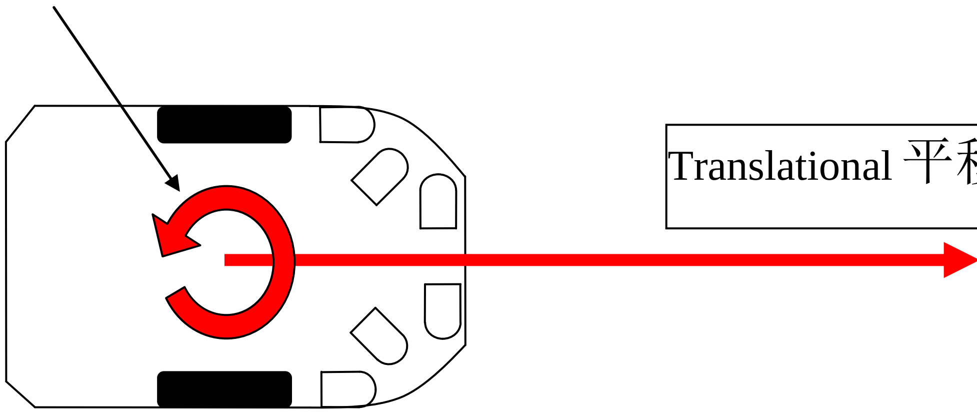


Figure: Diagonal 斜线 run. Collision avoidance and position adjustment

3.0 Speed Profile 速度命令曲線

Rotational 旋转 speed(wSpeed)

Translational 平移 speed (xSpeed)

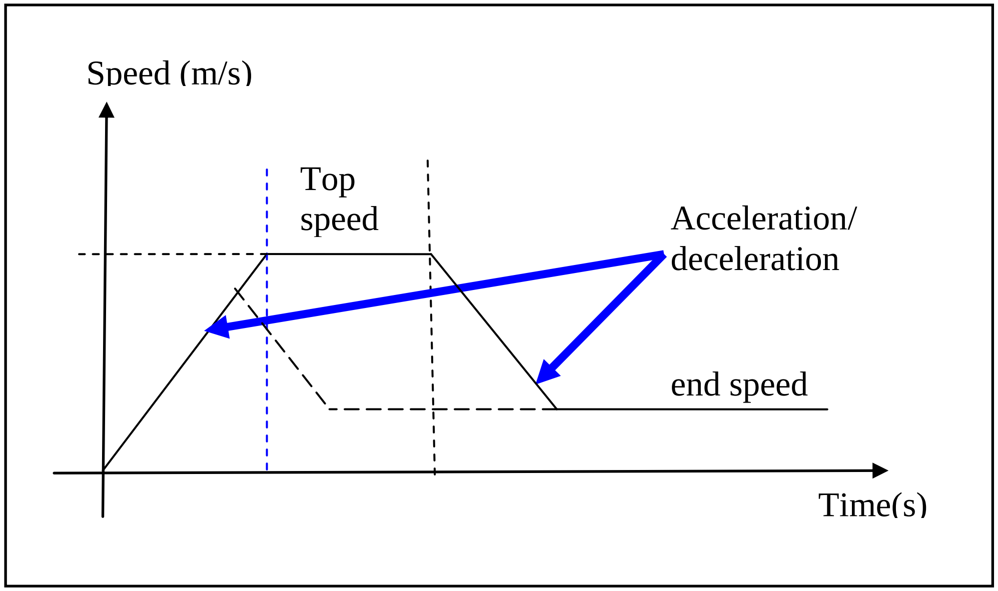


leftWheelSpeed = xSpeed – wSpeed;
rightWheelSpeed = xSpeed + wSpeed;

Figure : Speed components of a 2 wheels robot

3.1 Two basic speed profiles

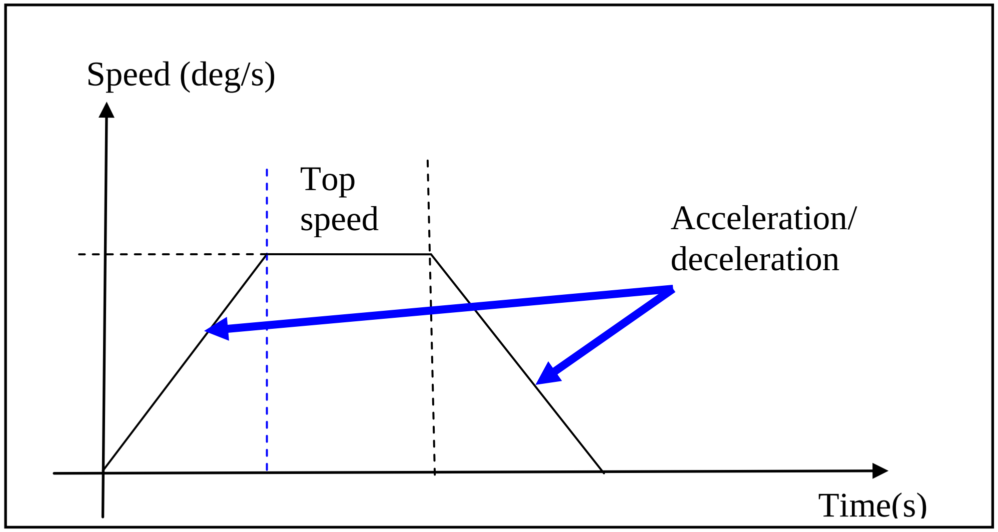
Translational speed 平移



Parameters 參數

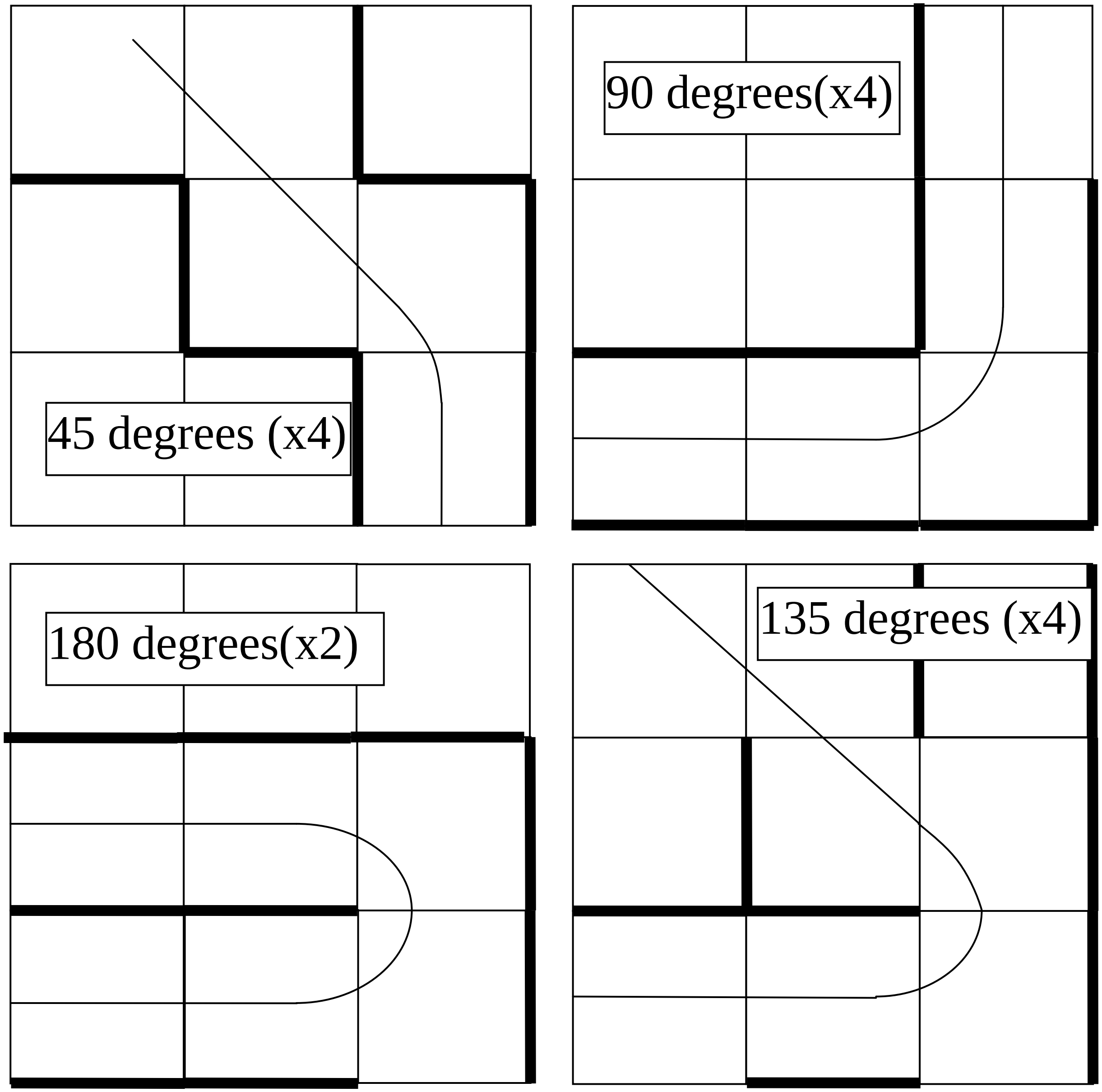
- Top speed
- Acceleration/deceleration
- End speed (= curve turn speed)
- Distance 距離

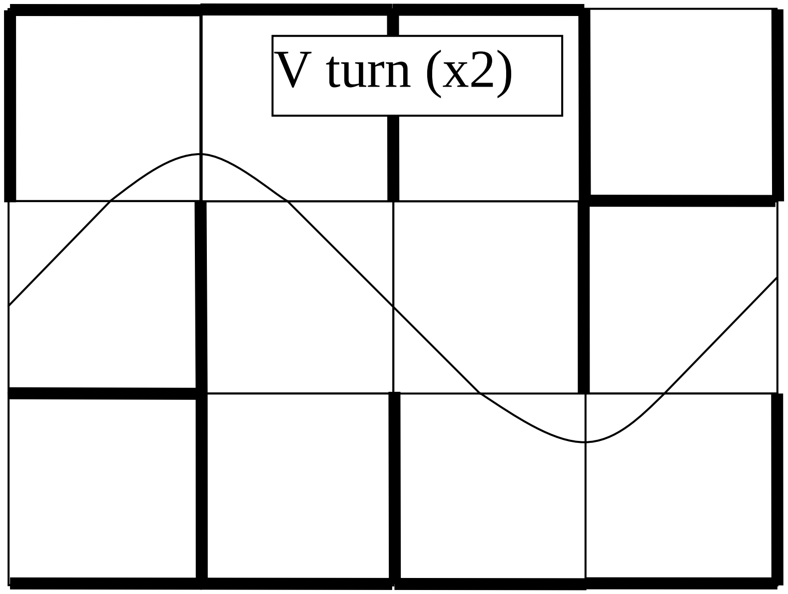
Rotational speed (旋轉速度)



3.2 Basic movements

- Diagonal & non-diagonal straight runs (wSpeed=0) 對角或非對角的直線運動
- Pivot turns (xSpeed=0) (原地旋轉)
- Curve turns 90, 180(U), 135(J), 45,V(xSpeed=constant, + wSpeed)


45 degrees (x4)
90 degrees(x4)
180 degrees(x2)
135 degrees (x4)


V turn (x2)

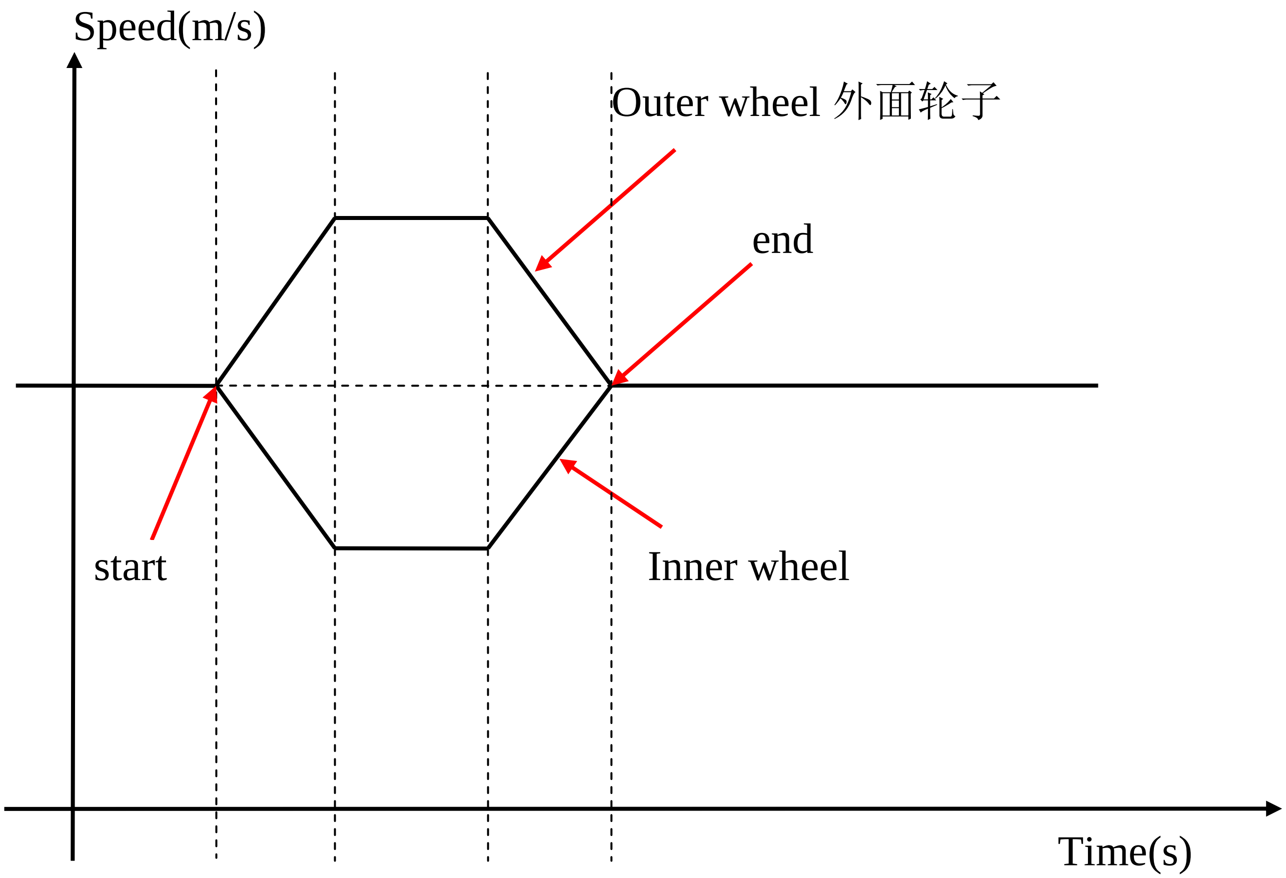


Figure : Curve turn speed profile

3.3 Profile generator and PD loop

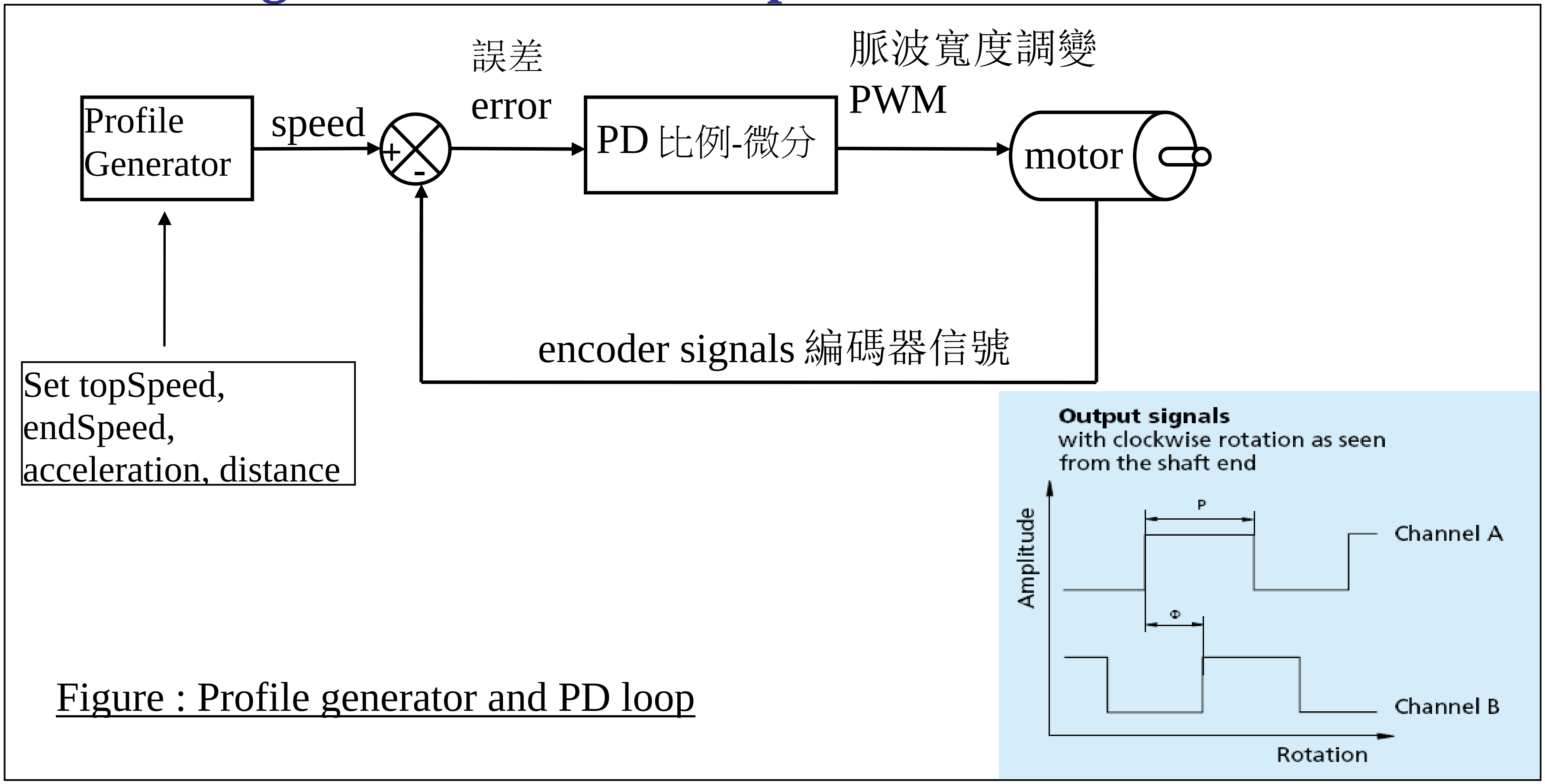


Figure : Profile generator and PD loop

- No integral 積分 term
- PD control & Profile generator is executed every msec 毫秒
- PWM frequency 頻率 is 10kHz 千赫
- PWM duty cycle = $kp \times error_n + kd \times (error_n - error_{n-1})$

3.4 Straight speed profile C code

```
// At the beginning of profile, targetSpeed = topSpeed;
// Keep checking for deceleration condition
void SpeedProfile() {

    decelerationRequired = (curSpeed² – endSpeed²)/(2×distance);

    if (decelerationRequired>deceleration) targetSpeed = endSpeed;

    if (curSpeed<targetSpeed) curSpeed += acceleration;

    if (curSpeed>targetSpeed) curSpeed -= deceleration;

    distance += curSpeed;
}
```

distance - 距離; curSpeed -目前的速度; endSpeed - 終點速度;
acceleration - 加速度; deceleration - 減速度

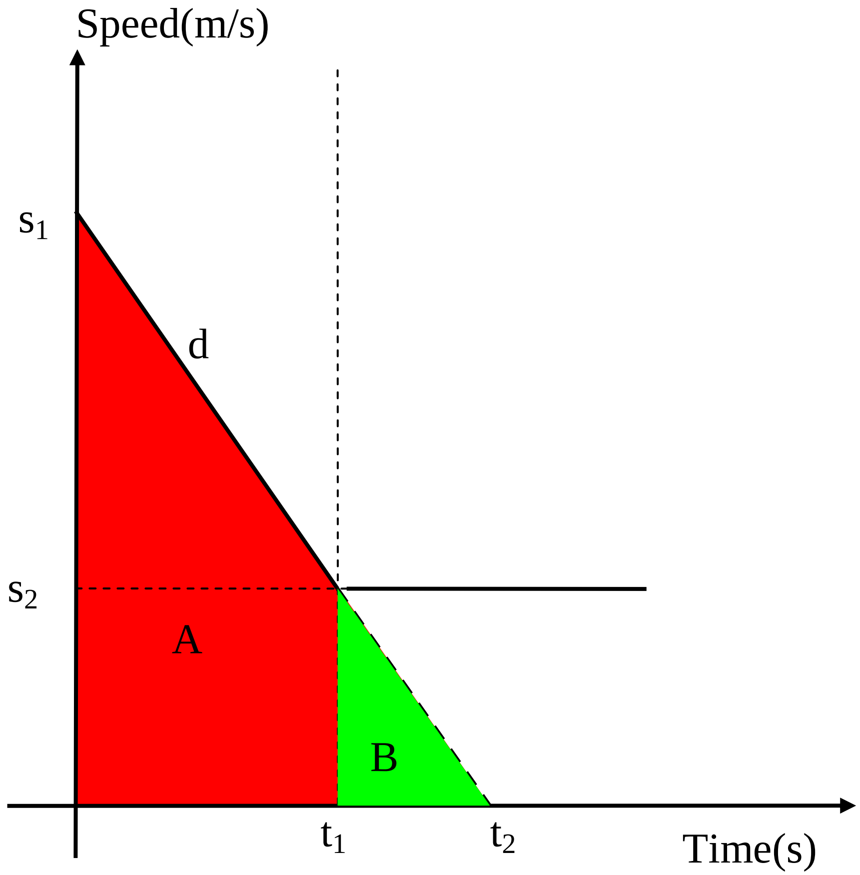


Figure : Speed profile

Calculating deceleration:

$A+B = (S_1 t_2)/2$

$B = S_2(t_2 - t_1)/2$

$d = S_1/t_2 = S_2/(t_2 - t_1)$

- $t_2 = S_1/d$
- $(t_2 - t_1) = S_2/d$

$$
\begin{aligned}
A &= (S_1 t_2)/2 - B \\
&= (S_1 t_2)/2 - S_2(t_2 - t_1)/2 \\
&= (S_1^2 - S_2^2)/2d
\end{aligned}
$$

$$d = (S_1^2 - S_2^2)/2A$$

3.5 PD loop and gyroscope

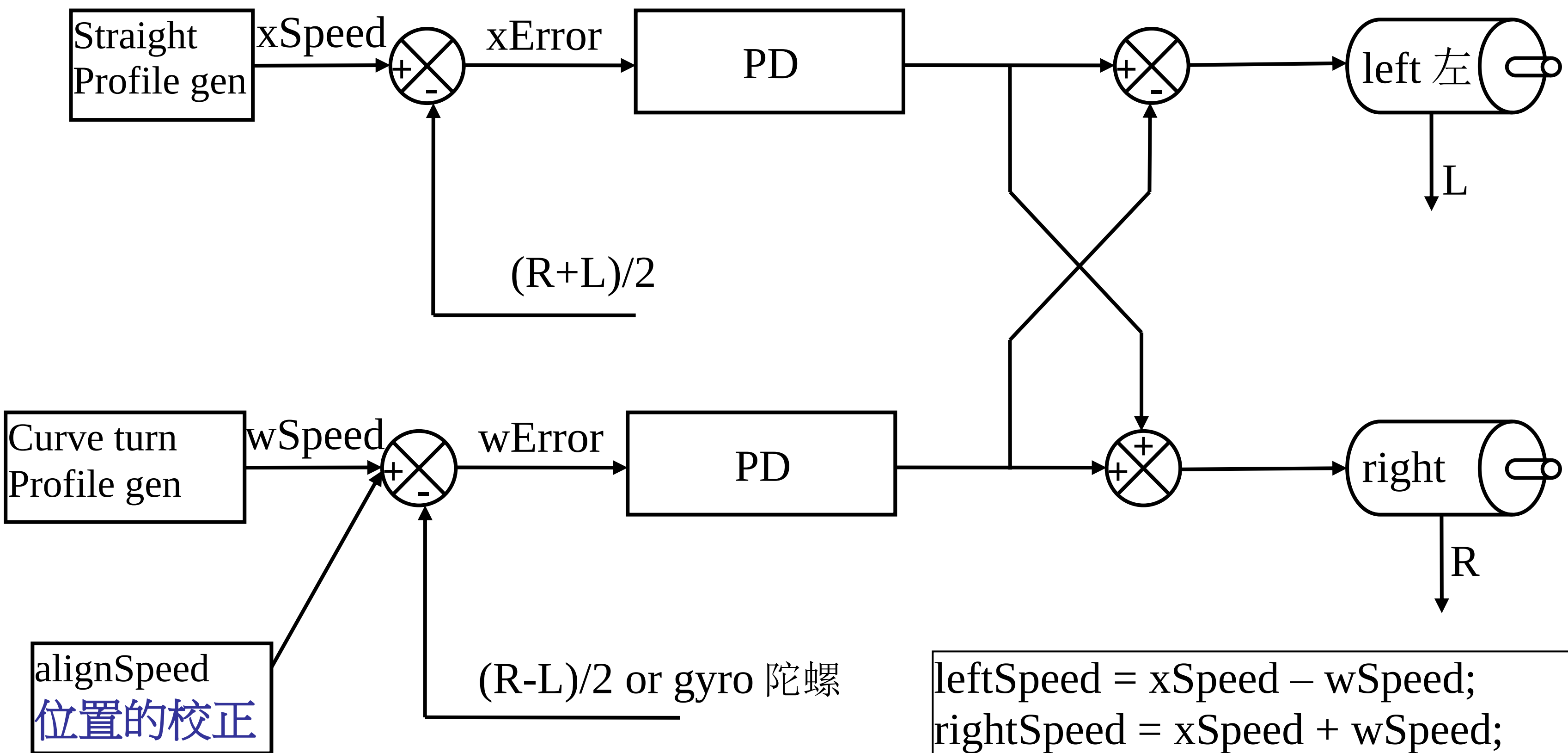
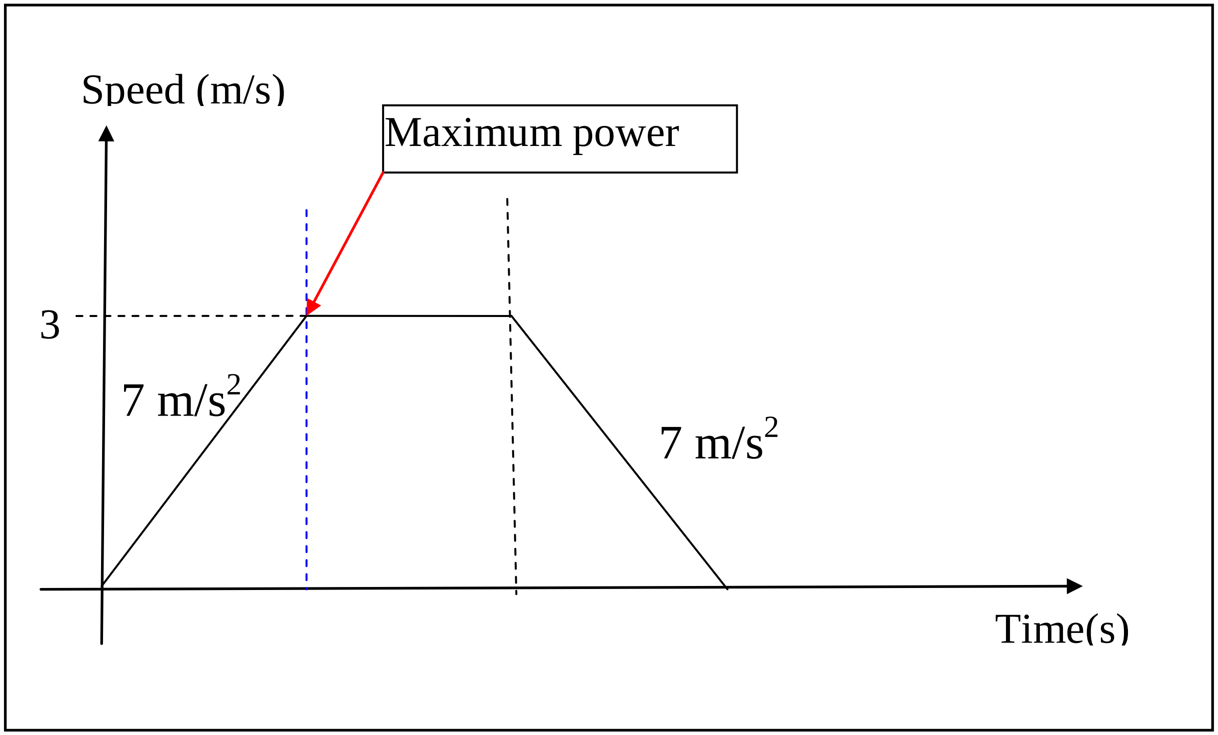


Figure : rotational and translation control loop

leftSpeed = xSpeed – wSpeed;
rightSpeed = xSpeed + wSpeed;
xSpeed = (rightSpeed+leftSpeed)/2
wSpeed = (rightSpeed-leftSpeed)/2

- Gyro 陀螺 is fast in response
- Accurate for fast curve turn (not for very slow turn) / opposite of optical encoder 編碼器信號
- Need to calibrate 校準 before run due to DC drift 漂移?
- Not affected by tires size change

4.0 DC motor sizing 直流馬達的選擇



Step 1 : Calculate maximum power 功率 required

- Force(N) = mass(kg) × acceleration(m/s^2) 動力 =重量×加速度
  - Desired acceleration = 7 m/s^2 // experience required

- Force = 0.12kg × 7m/s_2 = 0.84N
- Power (W) = force(N)× speed(m/s) (功率 = 動力× 速度)
- Maximum power is required when robot is accelerating at 7 m/s_2 near top speed = 3m/s.
- Max power required = 0.84N×3 m/s = 2.52 watt.
- Since there are 2 motors, power required per motor = 1.26 W.

Step 2 : Select motor (Faulhaber 1717SR)

- Choose a motor with maximum Power 1.5 times to 2 times 1.26W = 1.89W to 2.52 W,
- Which is less than the Pmax of 1.96 W (Faulhaber 1717SR).

Series 1717 ... SR

		1717 T		003 SR	006 SR	012 SR	018 SR	024 SR	
1	Nominal voltage	U_N		3	6	12	18	24	Volt
2	Terminal resistance	R		1,07	4,30	17,1	50,1	68,8	Ω
3	Output power	$P_{2\,max.}$		1,97	1,96	1,97	1,50	1,96	W
4	Efficiency	$\eta_{max.}$		69	69	70	68	70	%
5	No-load speed	n_o		14 000	14 000	14 000	12 300	14 000	rpm
6	No-load current (with shaft ø 1,5 mm)	I_o		0,091	0,046	0,023	0,013	0,011	A
7	Stall torque	M_H		5,37	5,34	5,38	4,66	5,36	mNm
8	Friction torque	M_R		0,18	0,18	0,18	0,18	0,17	mNm

- Note that I also overdrive the motor. 2x Lithium Polymer = 8.4V. Average around 7.8 volts
- Maximum power out at 7.8 V = $(7.8/6)^2 \times 1.96$ = 3.3 watts

Step 3 : Select gear ratio齒輪比

- Calculate torque and speed
  - Tires diameter 直徑 is 24mm
  - Torque required = Force×radius (扭矩 = 動力 × 半徑)

$= 0.84N \times 12mm = 10.08\ mNm$

- Torque required per motor = 5.04mNm.
- Wheel circumference 圓週= Pi×diameter = 0.0754m
- At 3m/s, wheel revolution 旋轉= speed/circumference

  = 3/0.0754m = 39.8 rps = 2387 rpm

- Therefore, near top speed, motor must be able to provide torque output of 5.04mNm at speed of 2387 rpm.

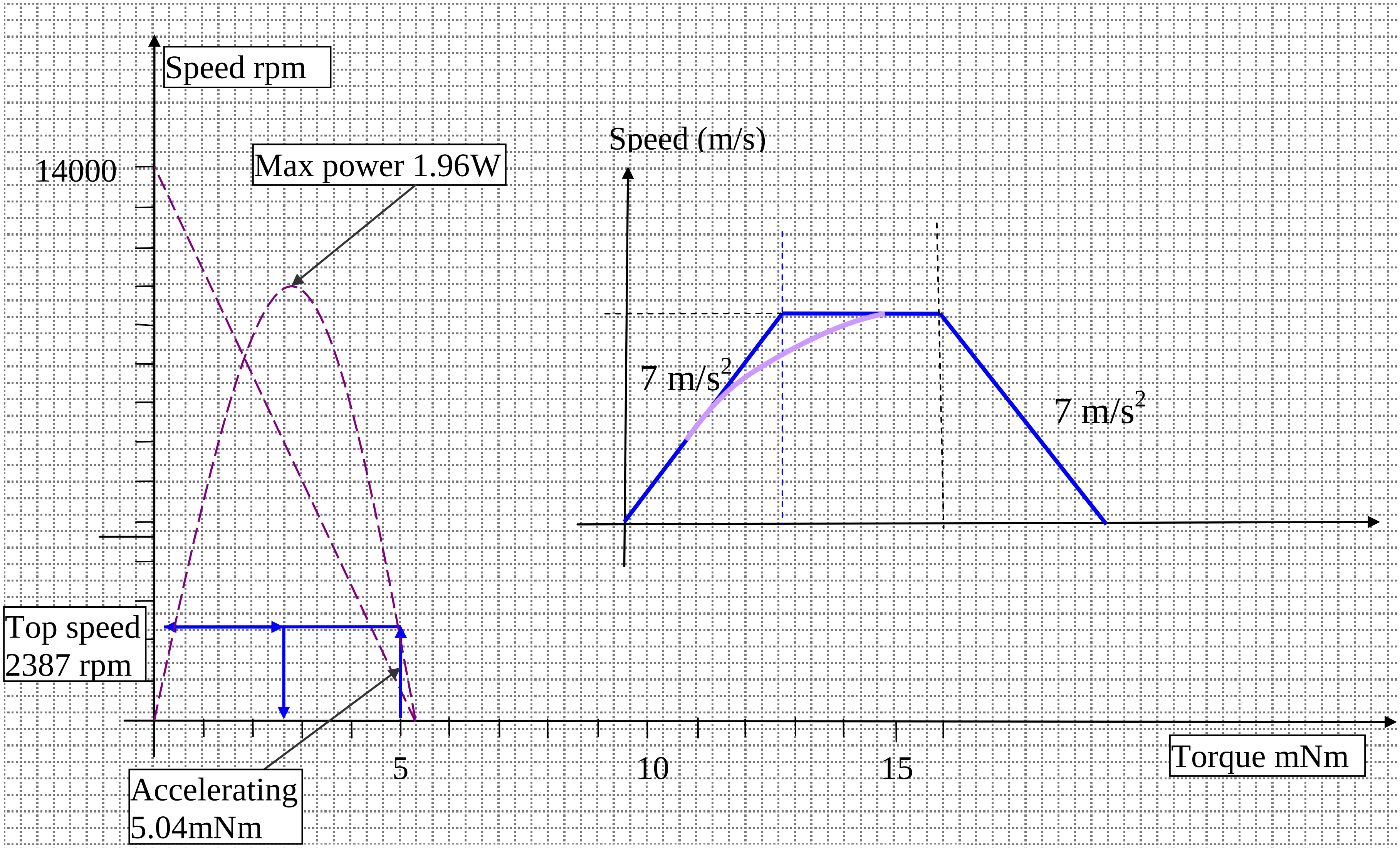


Figure : DC motor speed-torque 扭矩 characteristic 特徵

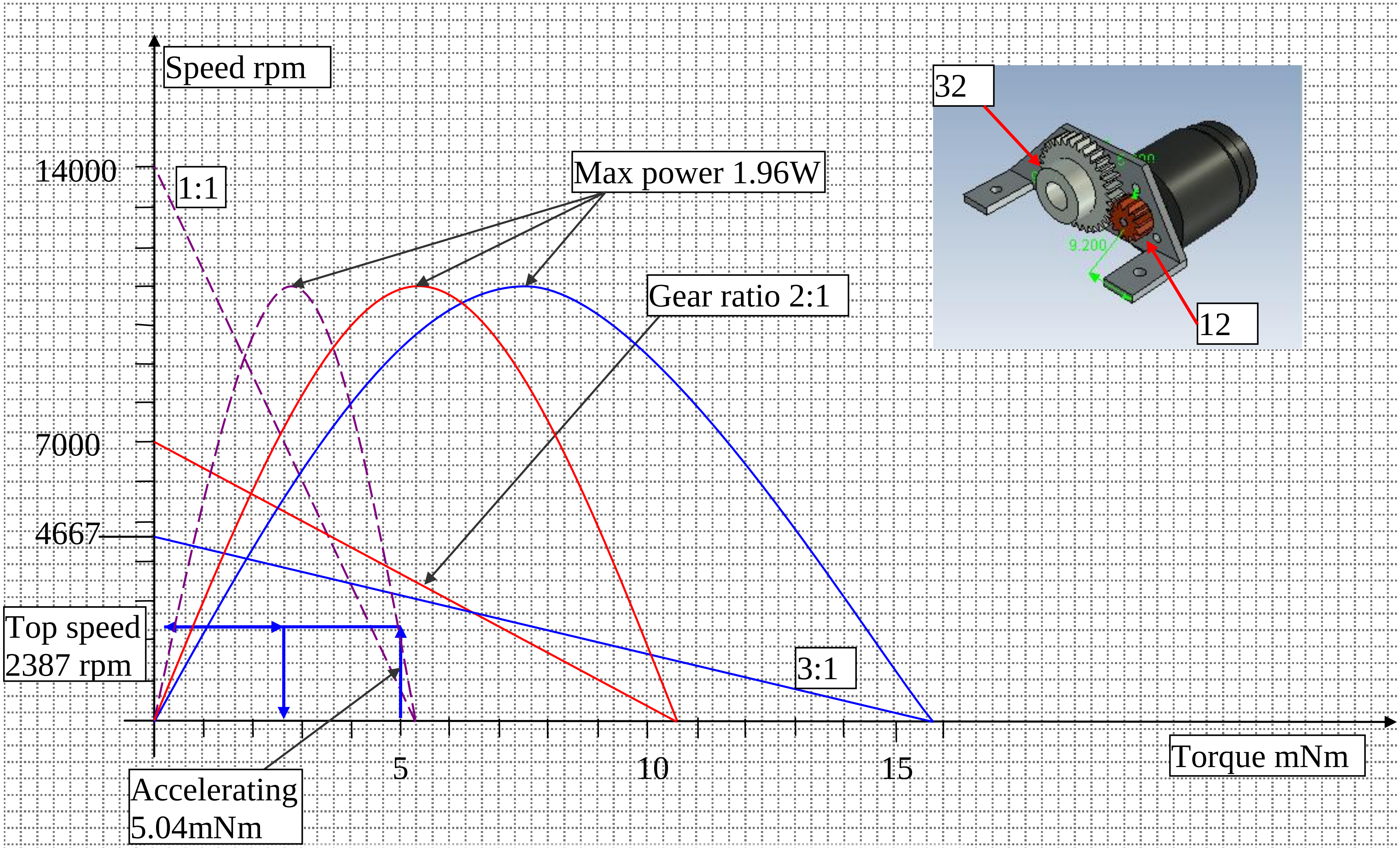


Figure : DC motor speed-torque 扭矩 characteristic 特徵

- There is a range of gear ratio to select from
- The higher the gear ratio, the more energy efficient 效率 the motor is.
- Allows spare power for alignment

Prefer long & slim 細長 motors to short and fat 肥短 motor. Lower c.g.

4.1 DC motor driver

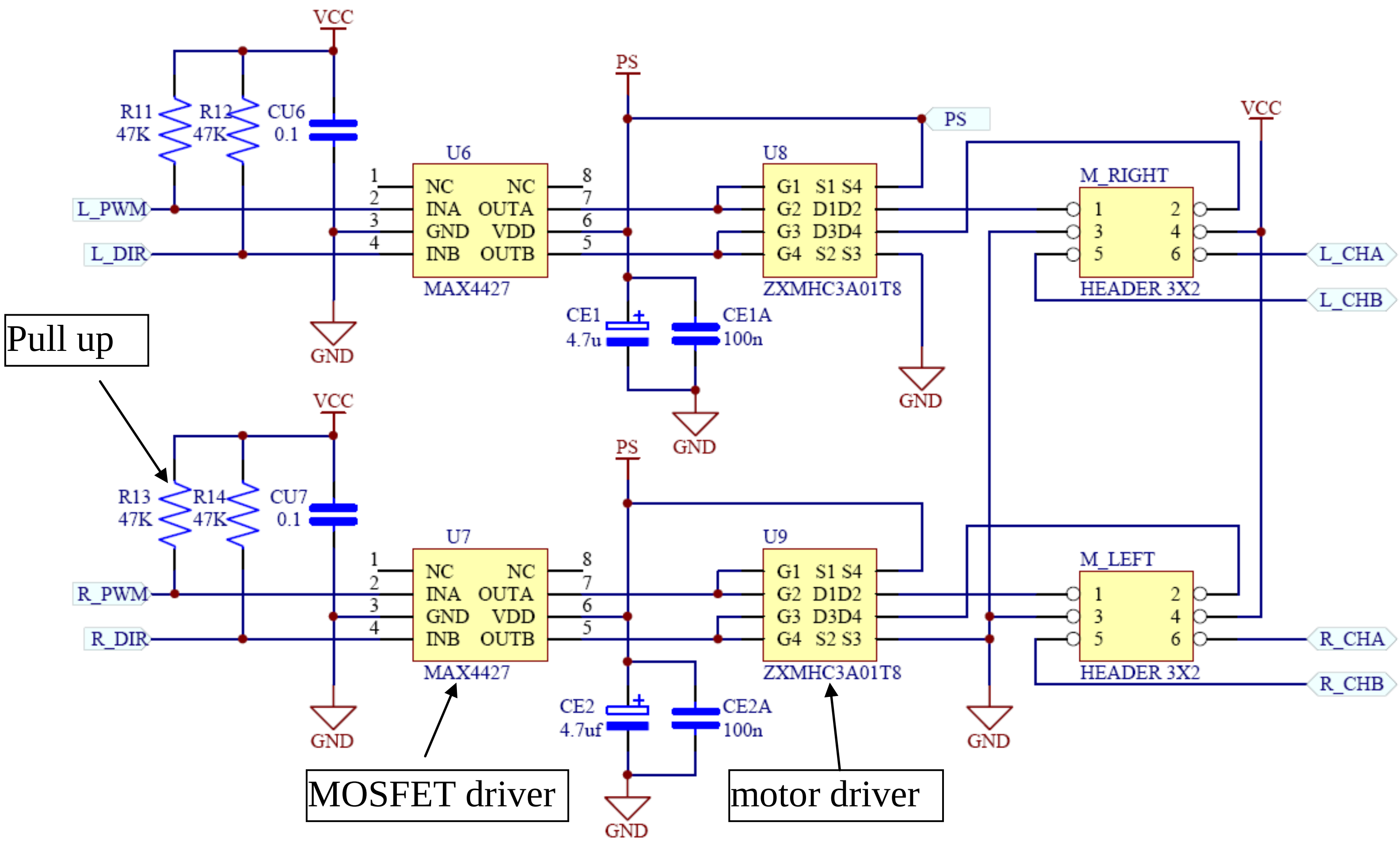


Figure : Min5 motor driver circuit

-Choose low turn-On resistance MOSFET
-Pull up input to prevent conduction

User interface

- Need feedback to tell what the mouse is doing
- LED lights
- Sound
- Menu driven (dotmatrix display & input switch)

Websites

- Visit my website at www.np.edu.sg/alpha/nbk
- (A* Pathfinding) http://www.policyalmanac.org/games/aStarTutorial.htm
- (Nakashima website) http://homepage1.nifty.com/hfd01577/index.html
- (Pete Harrison) http://micromouse.cannock.ac.uk

Good Luck!