Robot Operating System

Chapter 7

How to build Autonomous Mobile Robot

Pyae Soan Aung
RHCSA
Rom Robitics Co.ltd

Outline

- Weight of Robot?
- Torque
- Friction
- Calculating motor torque
- Calculating motor rpm
- Motor Control

Weight of Robot

We nee this.

Maximum payload = 2 Kg

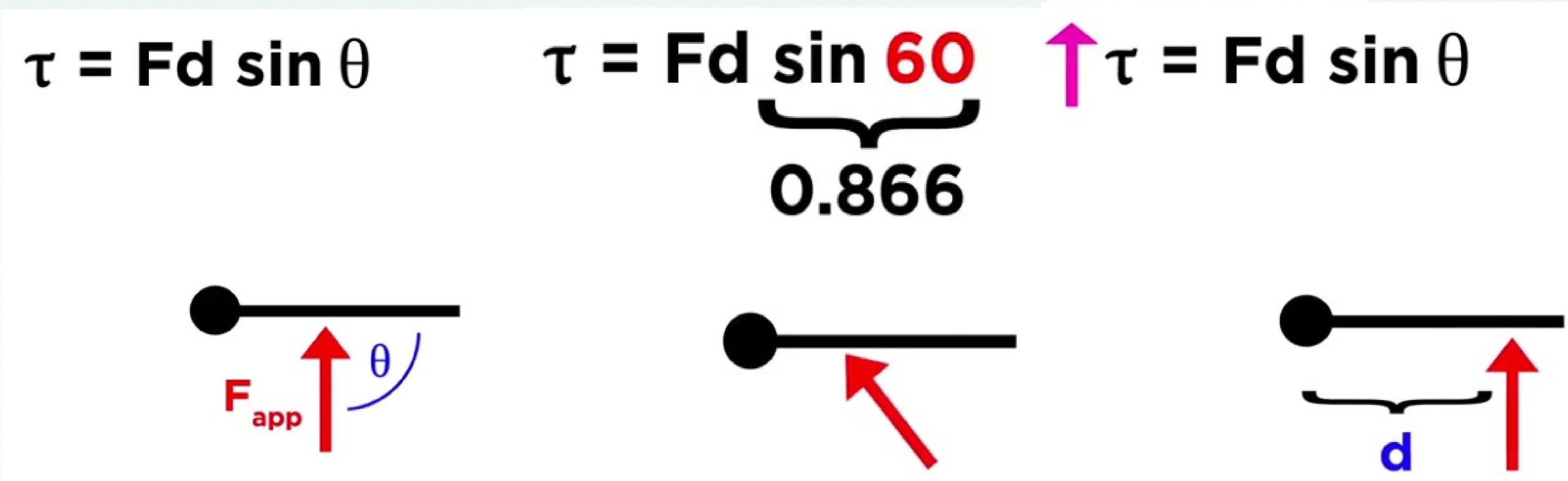
Body Weight = 2 to 2.5 Kg

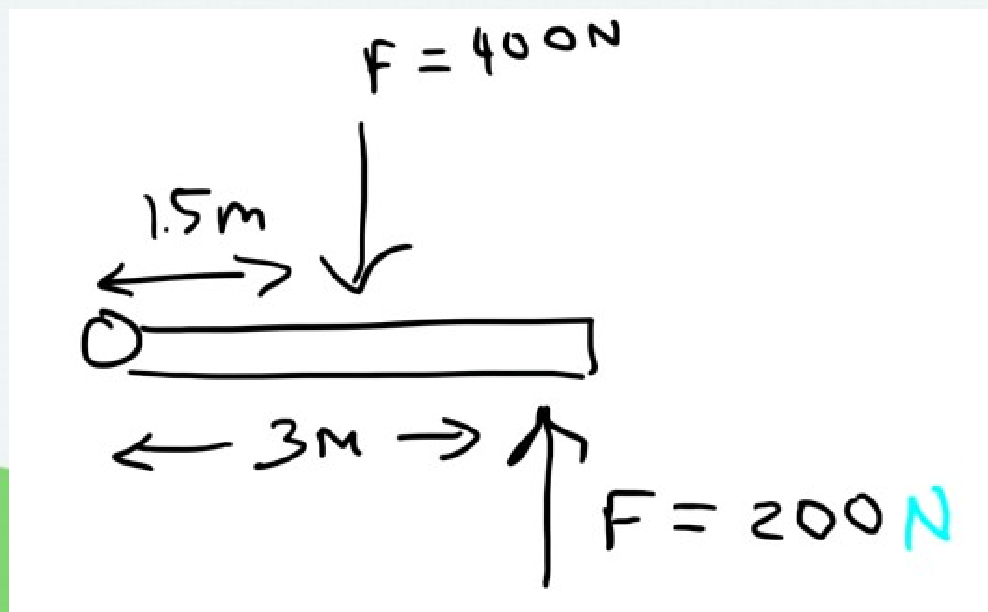
Maximum speed = 0.35 m/s

Ground Clearance = 3 cm

What type of motor?

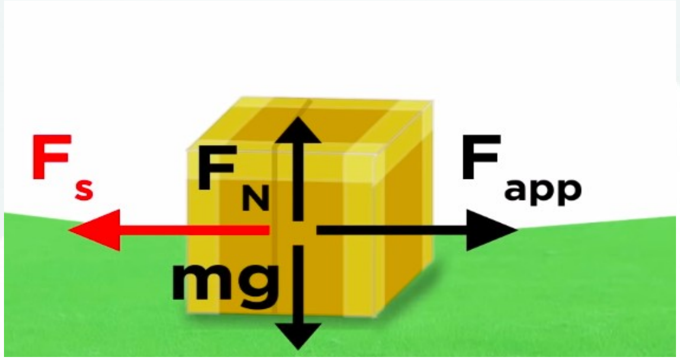
Torque



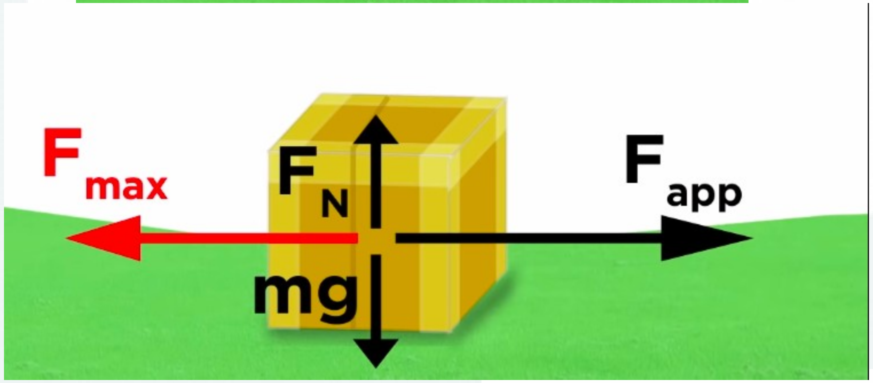


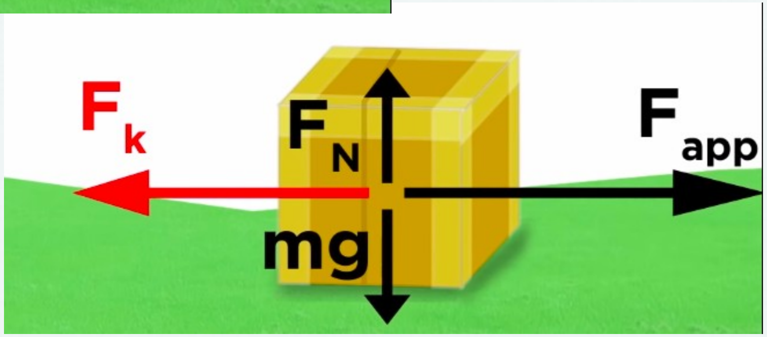
N.m (or) Kg.cm

Friction



$F_s = \mu_s F_N$





$F_k = \mu_k F_N$

$\mu_k = F_k / F_N$

Friction

surfaces	μ_s	μ_k
glass on glass	0.94	0.4
steel on steel	0.74	0.57
copper on steel	0.53	0.36
ice on ice	0.1	0.03
teflon on steel	0.04	0.04

Calculating motor torque

Wheel diameter = 7 cm
Coefficient of friction = 0.6

$$Total_{\text{Weight of Robot}} = Weight_{\text{Robot}} + Weight_{\text{Payload}} = (2.5\ Kg * 9.8) + (2\ Kg * 9.8) = 44.1\ N$$

$$F_{\text{rictionForce}} = \mu * N_{\text{normal}}$$

$$\tau = F * r$$

$$\tau = \mu * N * r$$

$$\tau = 0.6 * 44.1/2\ N * 0.035\ m = 0.46305\ N.m = 4.721\ Kg.cm$$

OK

Gear Motor parameter list	
Rated voltage	DC12.0V
No-load speed	320RPM 0.15A
Max efficiency	Load 1.7kg.cm/253rpm/4.2W/0.6A
Max power	Load 4.0kg.cm/160rpm/6.7W/1.2A
Stall	STALL TORQUE 7.5kg.cm STALL CURRENT 3.4A
Retarder reduction ratio	1 : 30
Holzer resolution	Motor Holzer11×ratio30=330PPR
Gear Motor Outline Drawing	

Calculating motor rpm

Max speed = 0.3 m/s
Wheel Diameter = 0.07 m

RPM = 0.3 m/s * 60 s/1m * πd m/1rev

RPM = (0.3*60) / (3.14*0.07) = 82 rpm

Our motor is maximum 320! it’s ok.

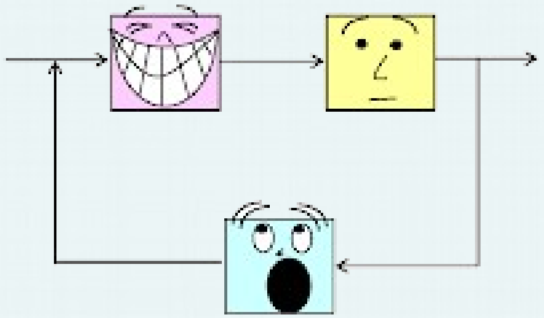
Design Summary

Motor rpm = 80
Motor torque = 5 kg.cm
Wheel Diameter = 7 cm

Motor Control

- Motor
- Control

+

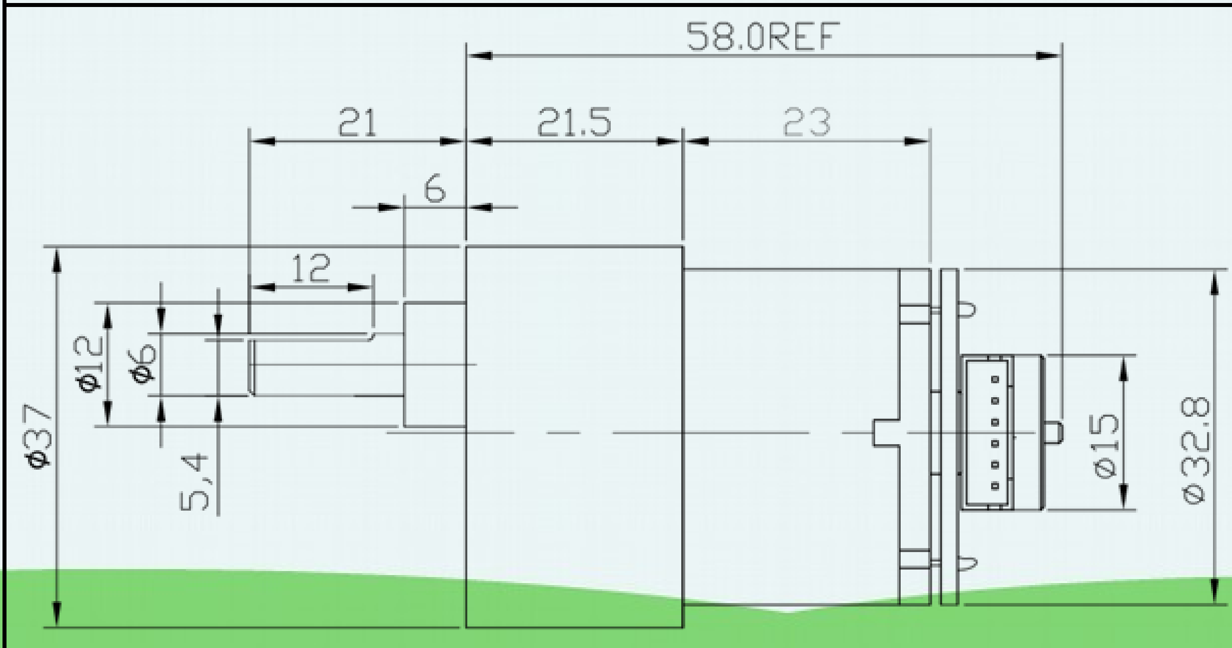


Motor

မော်တာ နမူနာ
အမျိုးအစား GM37-520 12V 320rpm

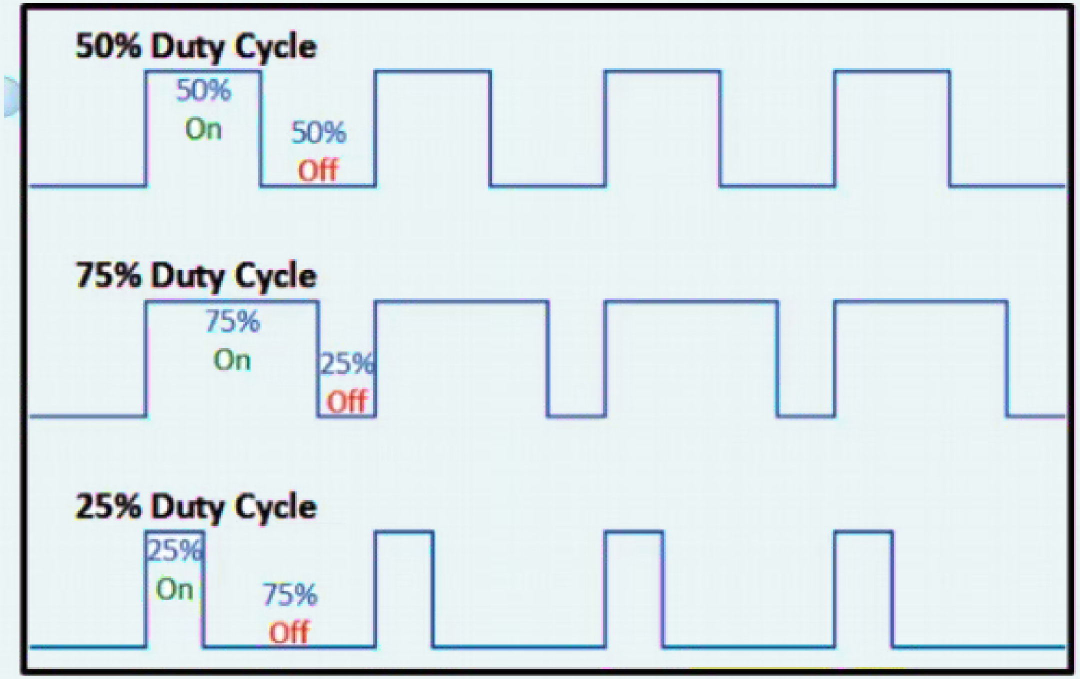
Gear Motor parameter list	
Rated voltage	DC12.0V
No-load speed	320RPM 0.15A
Max efficiency	Load 1.7kg.cm/253rpm/4.2W/0.6A
Max power	Load 4.0kg.cm/160rpm/6.7W/1.2A
Stall	STALL TORQUE 7.5kg.cm STALL CURRENT 3.4A
Retarder reduction ratio	1 : 30
Holzer resolution	Motor Holzer11×ratio30=330PPR

Gear Motor Outline Drawing



Pulse Width Modulation(PWM)

1) Duty Cycle



Duty Cycle = 100% = 12V
Duty Cycle = 50% = 6V
Duty Cycle = 25% = 3V
Duty Cycle = 0% = 0V

2) PWM Resolution

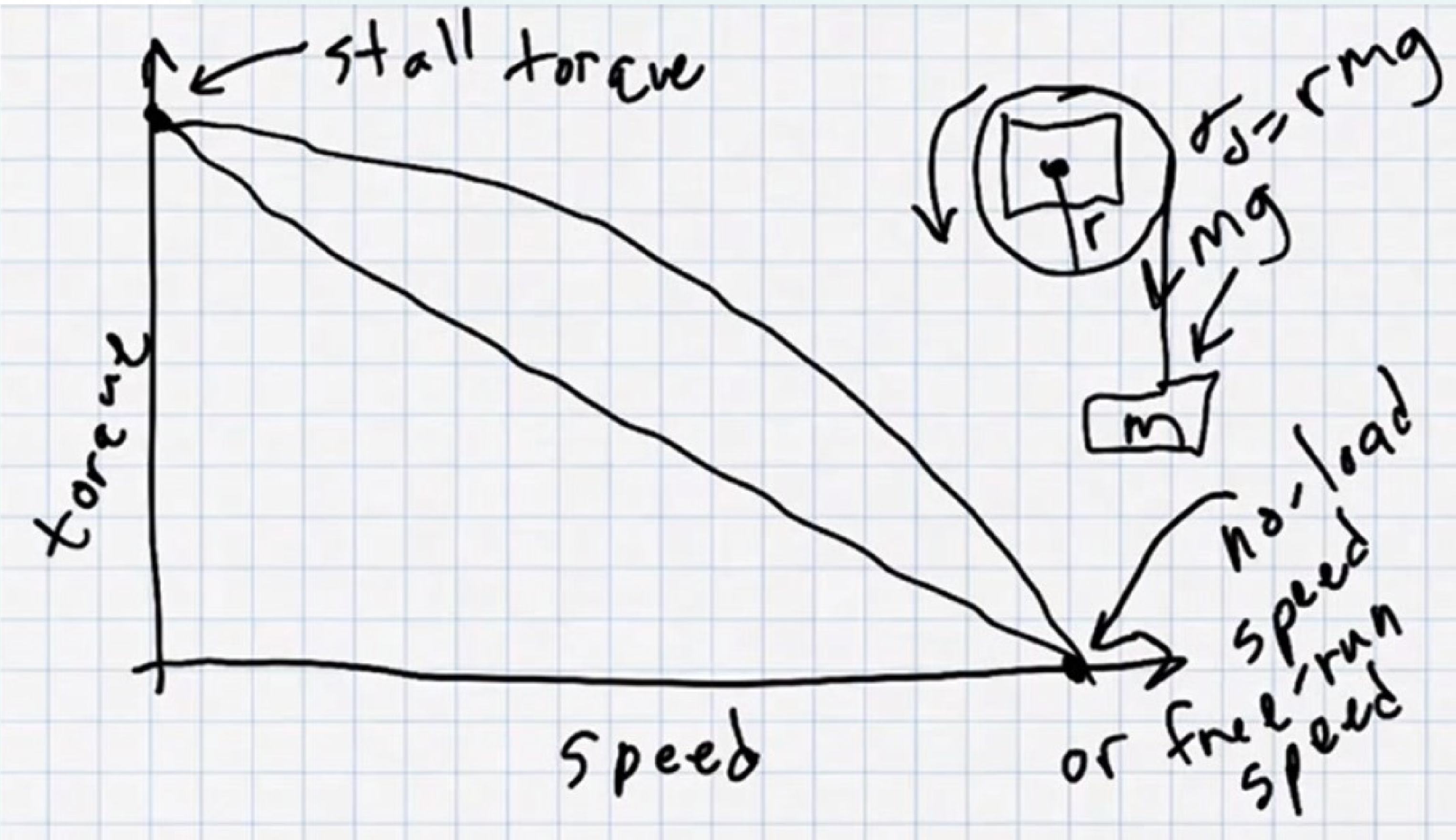
PWM 8bit Resolution

Duty Cycle = 100% = 255 = 12V
Duty Cycle = 50% = 128 = 6V
Duty Cycle = 25% = 64 = 3V
Duty Cycle = 0.4% = 1 = 0.048V

PWM 10bit Resolution needs (20MHz crystal)

Duty Cycle = 100% = 1023 = 12V
Duty Cycle = 50% = 512 = 6V
Duty Cycle = 25% = 128 = 3V
Duty Cycle = 0.1% = 1 = 0.012V

Torque and Speed



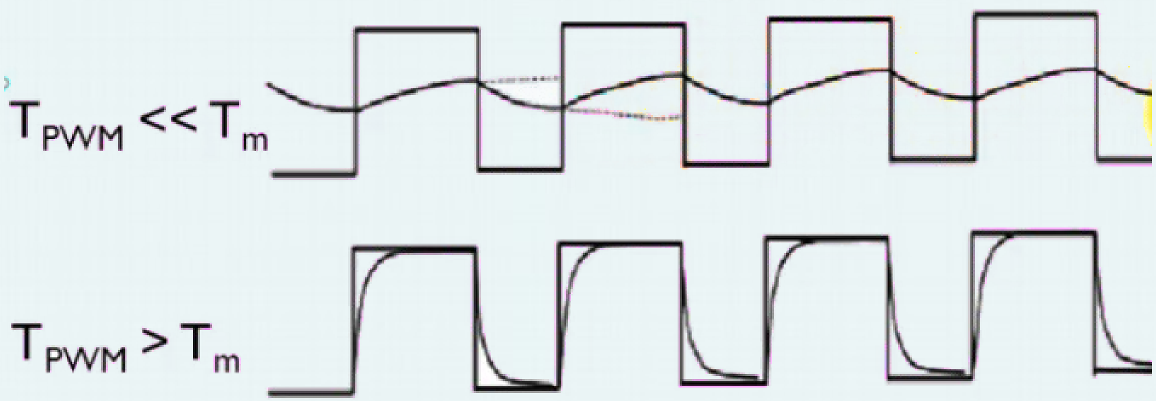
$$T = F * r * \sin(\theta)$$

$$T = F * r * 1$$

$$W = F = m * g$$

$$T = r * m * g$$

3) PWM Frequency



T_m = Motor time constant (second)
T_{PWM} = PWM signal period (second)

Frequency of PWM signal
$F = 1/T_{PWM}$

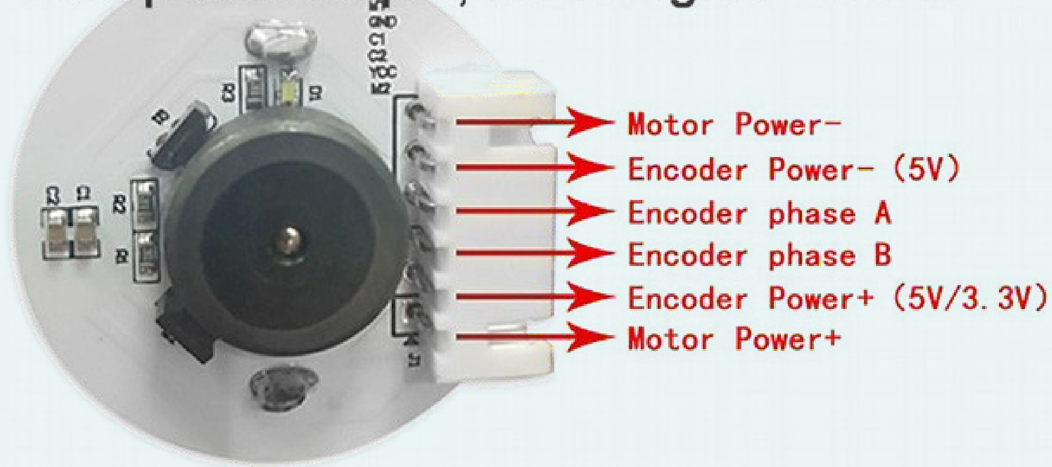
$T_m = 0.025$ s
$T_{PWM} = 0.001$ s
$F = 1/T_{PWM} = 1/0.001 = 1000$ Hz

မော်တာ **pwm signal** ကိုထုတ်ဖို့ **frequency** တခုထုတ်ပေးဖို့လိုသည်။ **motor time constant** ရဲ့ လေးပုံတစ်ပုံထက်ငယ်သင့်ပါတယ်။

Encoder

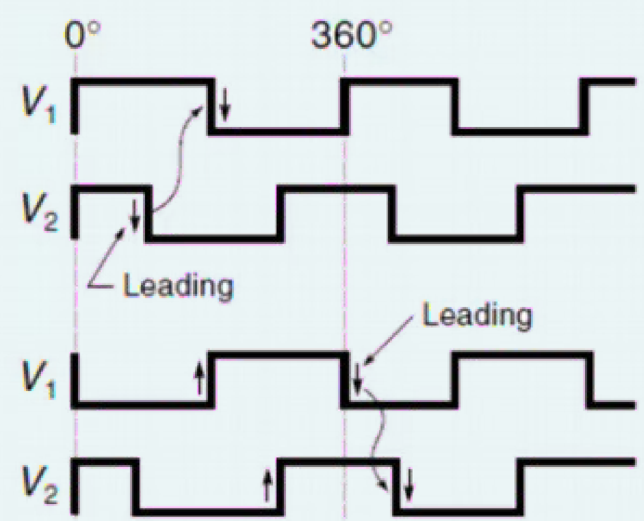
High and low level magnetic Hall encoder.

dual phase output, basic signal 11PPR

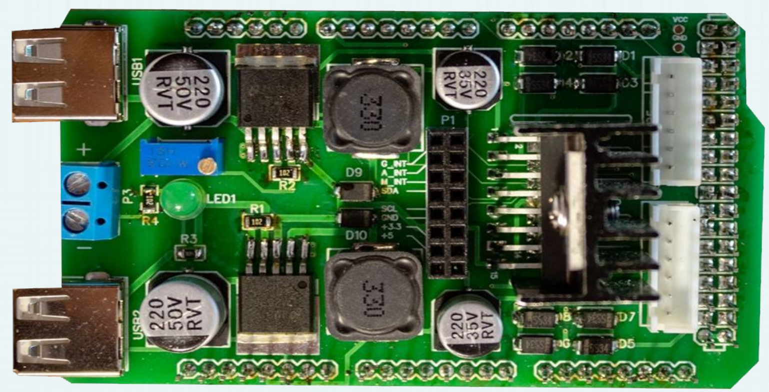


(b) CCW—Photocell waveforms for counterclockwise

(c) CW—Photocell waveforms for clockwise



Micro Controller & Motor Driver



1) PWM

2) Digital pin

3) Digital pin

Speed & Volt
Torque & Ampere

e.g 12V-->320rpm

$$\omega_{noLoad} = K_v * V$$

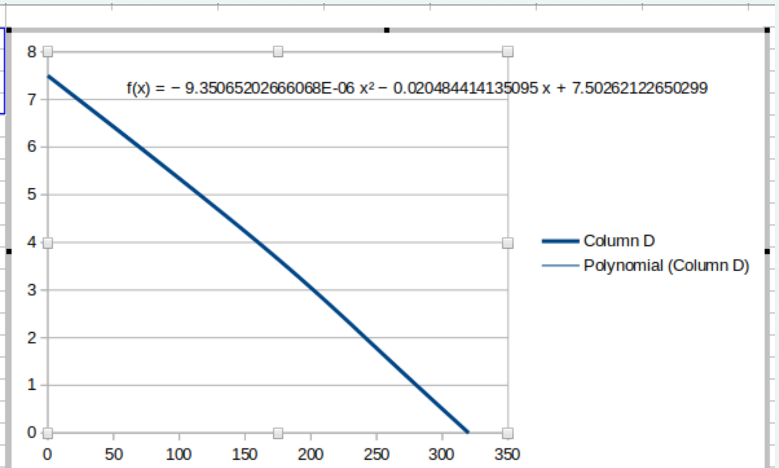
Velocity Constant

e.g 7.5Kg.cm-->3.4A

$$\tau = K_t * I$$

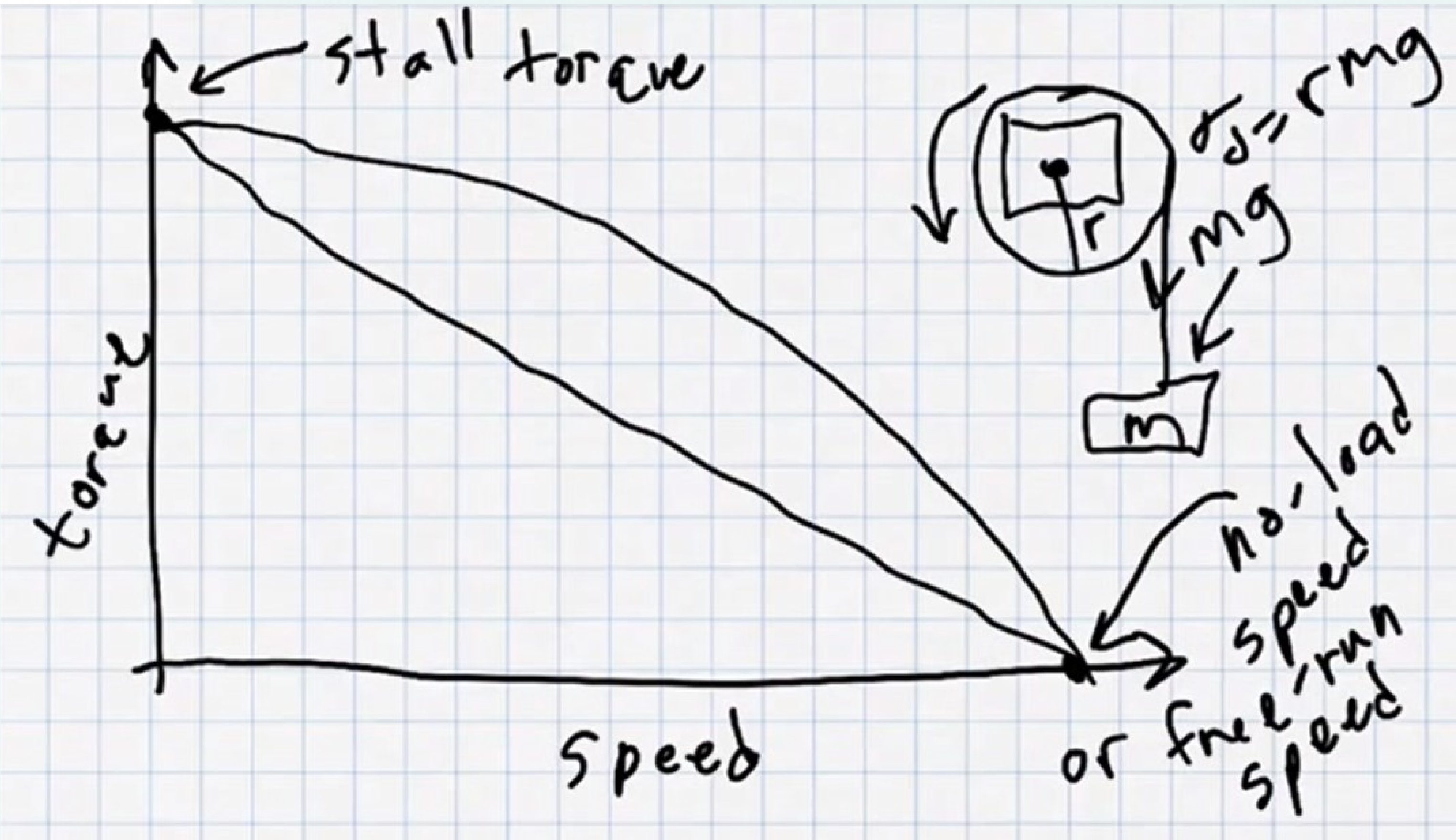
Torque Constant

RPM	Torque
0	7.5
160	4
253	1.7
320	0



Torque Speed Curve

Torque and Speed



$T = F * r * \sin(\theta)$

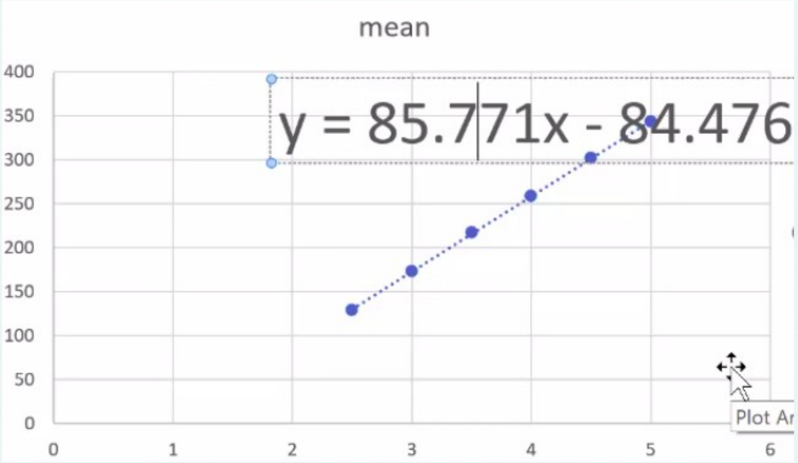
$T = F * r * 1$

$W = F = m * g$

$T = r * m * g$

Calculate Speed (RPM)

B	C	D	E	F	G
Voltage	Speed1	Speed2	Speed3	mean	st. dev
5	339	345	346	343.3333	3.785939
4.5	298	303	304	301.6667	3.21455
4	258	259	260	259	1
3.5	218	217	217	217.3333	0.57735
3	172	174	173	173	1
2.5	128	129	129	128.6667	0.57735
2	0	0	85	28.33333	49.07477
1.5	0	0	0	0	0



Counting ticks

```
  attachInterrupt(digitalPinToInterrupt(right_encoderA), calculate_right_A, CHANGE);
  attachInterrupt(digitalPinToInterrupt(right_encoderB), calculate_right_B, CHANGE);
}

void calculate_right_A() {
  if (digitalRead(right_encoderA) == digitalRead(right_encoderB)) {
    right_count = right_count - 1;
  }
  else                                                       {
    right_count = right_count + 1;
  }
}
void calculate_right_B() {
  if (digitalRead(right_encoderA) == digitalRead(right_encoderB)) {
    right_count = right_count + 1;
  }
  else                                                       {
    right_count = right_count - 1;
  }
}
```

```
void getMotorData(unsigned long time)  {
  RPM_act_right = double((right_count - prev_right_count) * 60000) / double(time * enc_ticks);
  prev_right_count = right_count;
}
```

Source code
https://github.com/GreenGhostMan/calculator_rpm/tree/master/calculator_rpm_counter

Speed Control with PID Controller

$$K_\text{p} e(t) + K_\text{i} \int_0^t e(t')\,dt' + K_\text{d} \frac{de(t)}{dt}$$

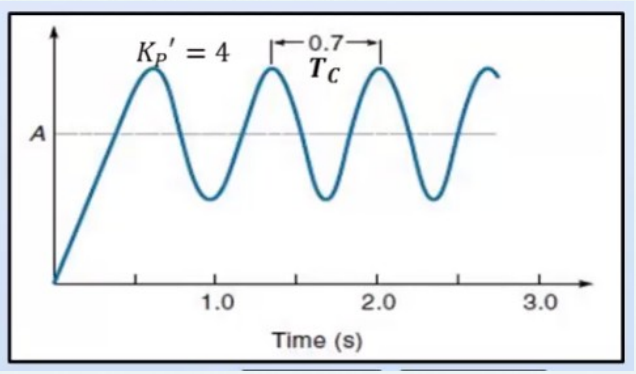
Closed-Loop Response	Rise Time	Overshoot	Settling Time	Steady-State Error	Stability
Increasing K_P	Decrease	Increase	Small Increase	Decrease	Degrade
Increasing K_I	Small Decrease	Increase	Increase	Large Decrease	Degrade
Increasing K_D	Small Decrease	Decrease	Decrease	Minor Change	Improve

pidTerm = Kp * error + Ki * int_error + Kd * (error - last_error)

Ziegler-Nichols Tuning Method(1940)

1. Continuous-cycle method (closed-loop method)

Continuous-cycle method ဟာ System ကို oscillation ဖြစ်တဲ့အထိ gain ကို tunning လုပ်ရမှာဖြစ်တဲ့အတွက် oscillation ဖြစ်တဲ့ ဒဏ်ကို ခံနိုင်တဲ့ system တွေမှာပဲ ဒီ method ကို အသုံးပြုနိုင်ပါတယ်။



The tuning procedure

Step 1 $K_P = 1$, $K_I = 0$, and $K_D = 0$ ထားပါ။ PID controller နဲ့ system ကို ချိတ်ဆက်ထားပါ။ Set point ကို rated value ရဲ့တဝက် မှာထားပါ။

Step 2 Output response ကို Amplitude တူ oscillation ဖြစ်တဲ့အထိ K_P ကို ဖြေးဖြေးချင်းတိုးတိုးပေးပါ။ ထို့နောက် K_P' နဲ့ T_C တန်ဖိုးတွေ ရပါမည်။

Step 3 ရလာတဲ့ K_P' နဲ့ T_C တန်ဖိုးတွေကိုသုံးပြီး K_P , K_I , K_D gain တန်ဖိုးတွေကို တွက်ပါ။

Step 4 တွက်ချက်ရရှိလာတဲ့ K_P , K_I , K_D gain တန်ဖိုးတွေကို PID controller ထဲထည့်ပြီး output respone ကိုကြည့်ပါ။ output response ကိုကြည့်ပြီး gain တွေကို လိုတိုးပိုလျှော့ အနည်းငယ်ချိန်ညှိပါ။ fine tunning

Gain estimator chart

Controller parameters for the Ziegler-Nichols frequency response method which gives controller parameters in terms of critical gain Kcr and critical period **Pcr**

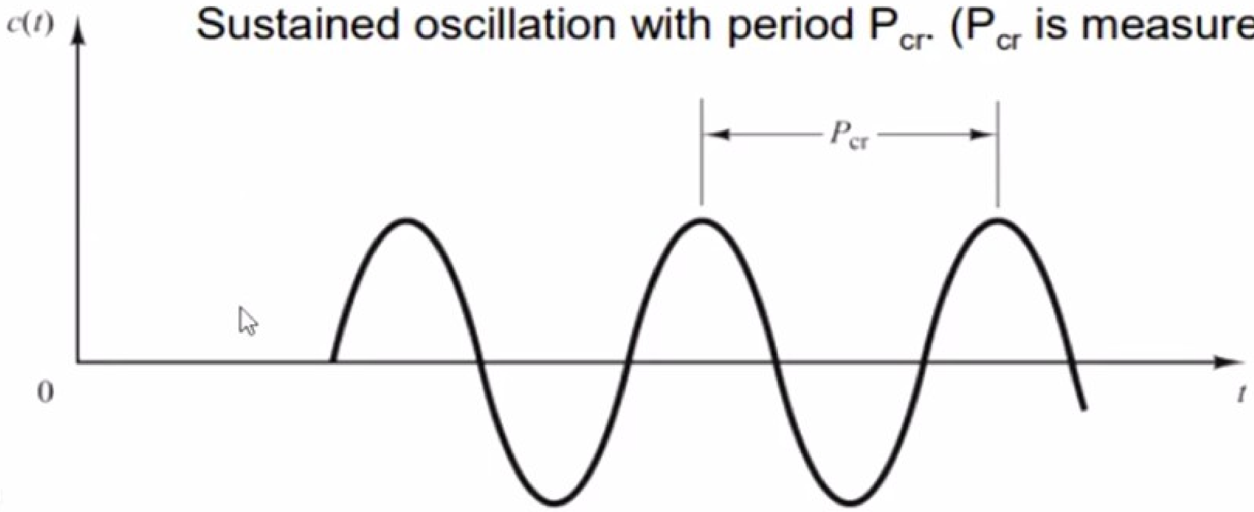
Type of Controller	K_p	T_i	T_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$\frac{1}{1.2}P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

Ziegler–Nichols Tuning, First Method

Start with Closed-loop system with a proportional controller.

1. Begin with a low value of gain, Kp
2. Reduce the integrator and derivative gains to 0.
3. Increase Kp from 0 to some critical value Kp=Kcr at which sustained oscillations occur. If it does not occur then another method has to be applied.
4. Note the value Kcr and the corresponding period of sustained oscillation, Pcr

Sustained oscillation with period P_{cr}. (P_{cr} is measured in sec.)



Code

PID Tuning

```
int updatePid(int old_pwm, double targetRPM, double currentRPM) {
  double pidTerm = 0;
  double error = 0;
  double new_rpm = 0;
  double new_pwm = 0;
  static double last_error = 0;
  static double int_error = 0;

  error = targetRPM - currentRPM;
    int_error += error;
    if(int_error > 1000) { int_error = 1000;}
    else if(int_error < -1000) {int_error=-1000;}
    pidTerm = Kp * error + Ki * int_error + Kd * (error - last_error);
    last_error = error;
  new_pwm = constrain(double(old_pwm) + pidTerm, -MAX_RPM, MAX_RPM);

  return int(new_pwm);
}
```

Source Code
https://github.com/GreenGhostMan/calculator_rpm/tree/master/calculator_pid_tester

Thank you!

rom robotics