# ENGR 3421: Robotics I

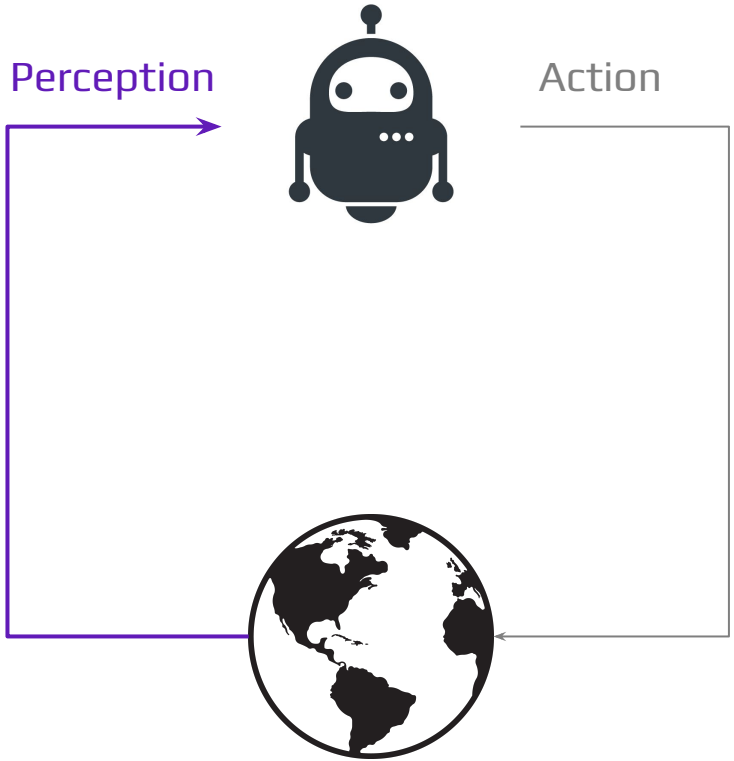
## Encoder

10/15/2024

# Outline

- Overview
- Quadrature Encoder
- Computations

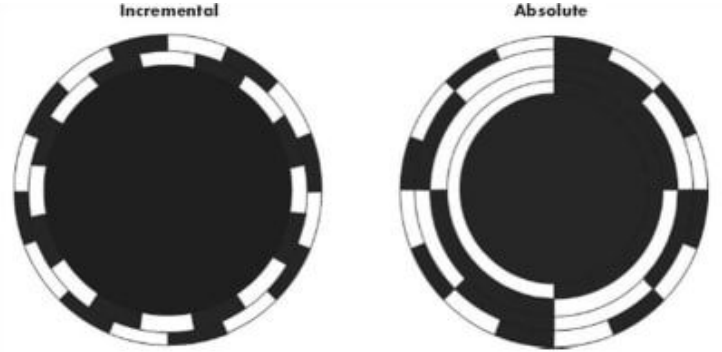
# A Robot Needs to Feel



# What is A (Rotary) Encoder

- (Rotary) Encoder measures angular movement.
- a common sensor for motors and other rotational devices.
- Provides closed-loop/feedback controls

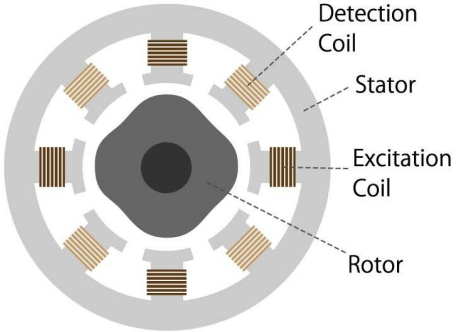
# Incremental vs. Absolute



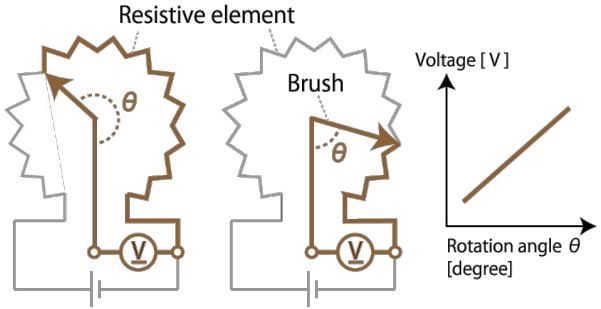

| Incremental | Absolute |
| --- | --- |
| Simple | Complicated |
| Cheap | Expensive |
| Measures angular displacement | Measures absolute position |
| Floating origin | Fixed origin |

# Types of Encoders

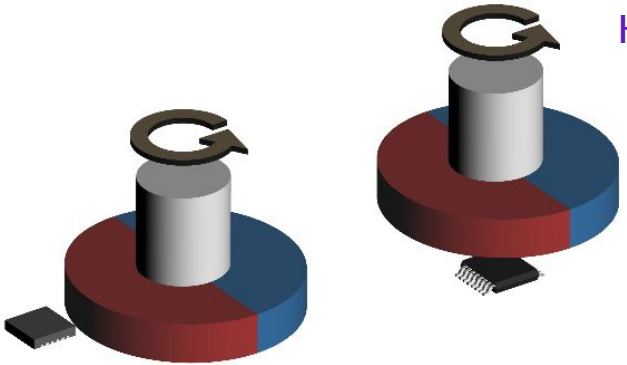
## Electromagnetic



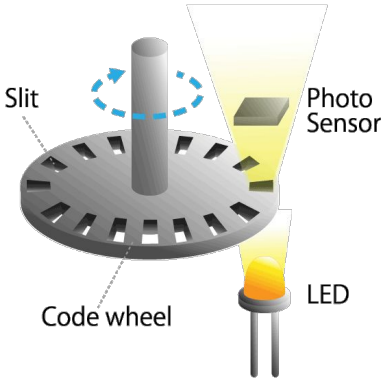

## Mechanical



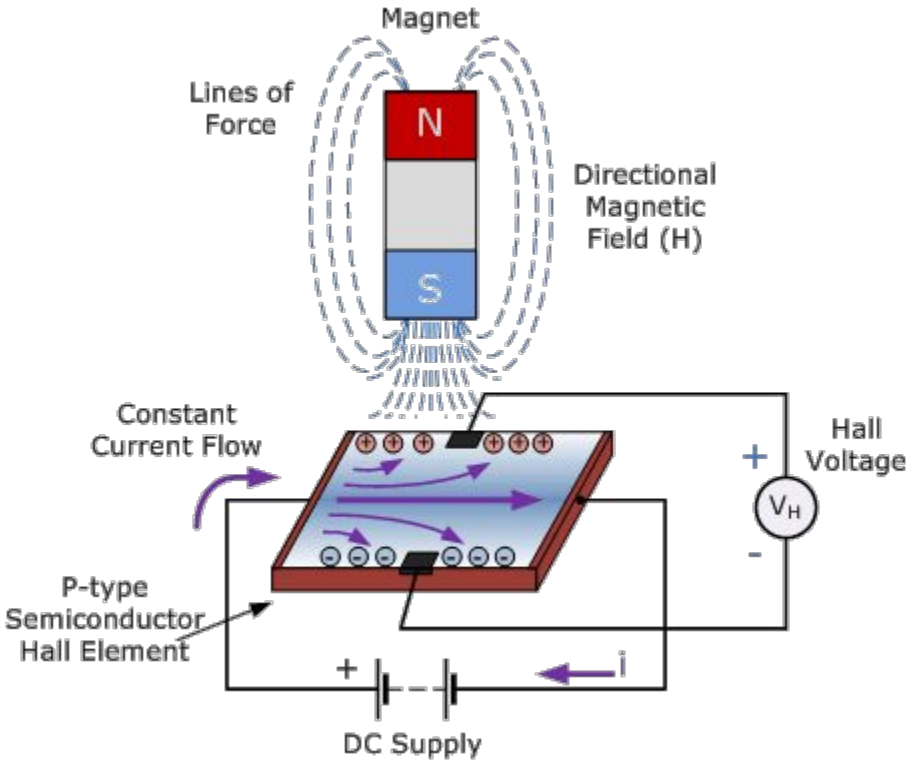

## Hall-Effect



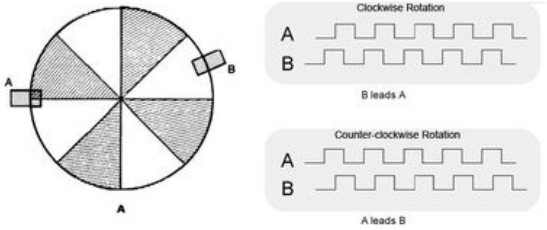
## Optical

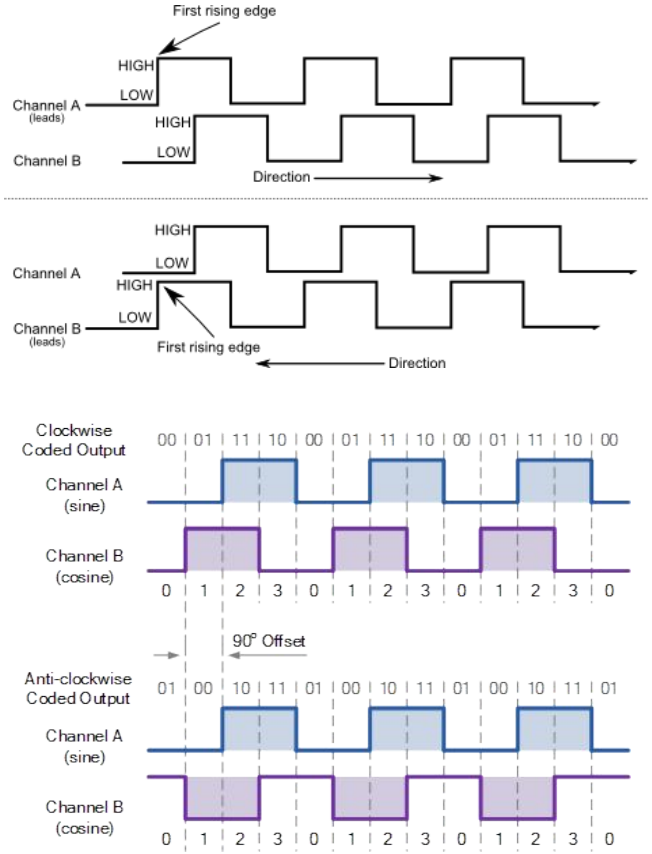


# Hall Effect

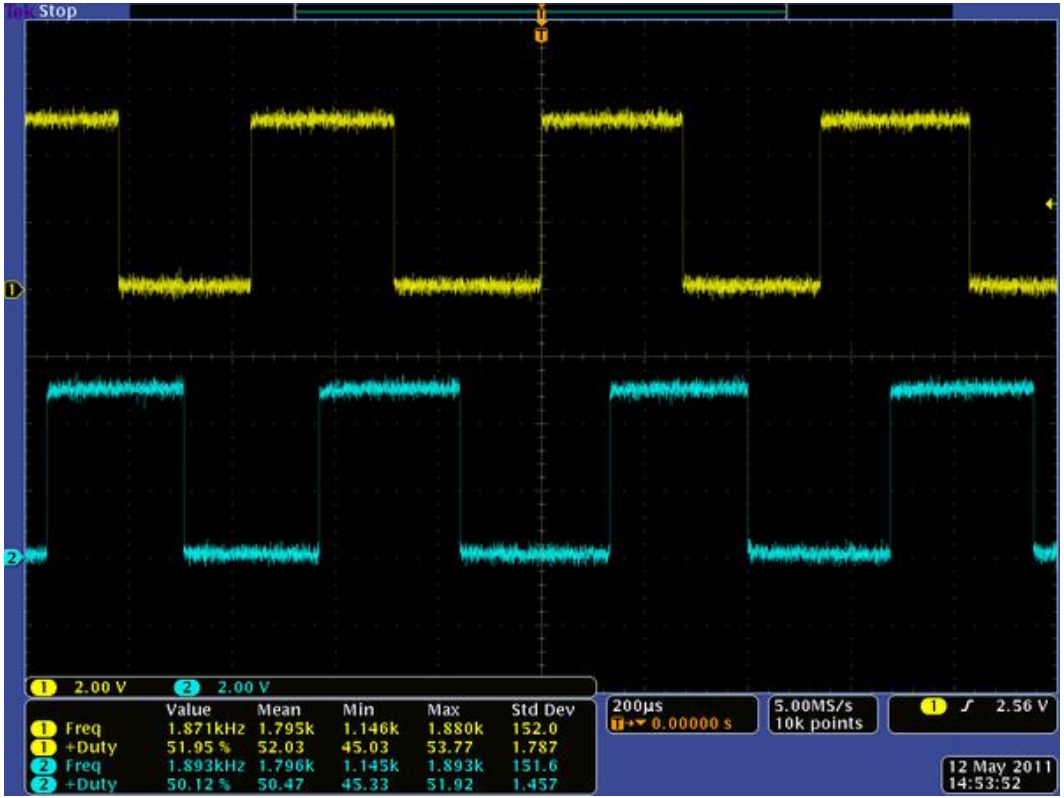

Magnet
Lines of Force
N
S
Directional Magnetic Field (H)
Constant Current Flow
Hall Voltage
+
$V_H$
-
P-type Semiconductor Hall Element
+
i
DC Supply

# Quadrature Encoder

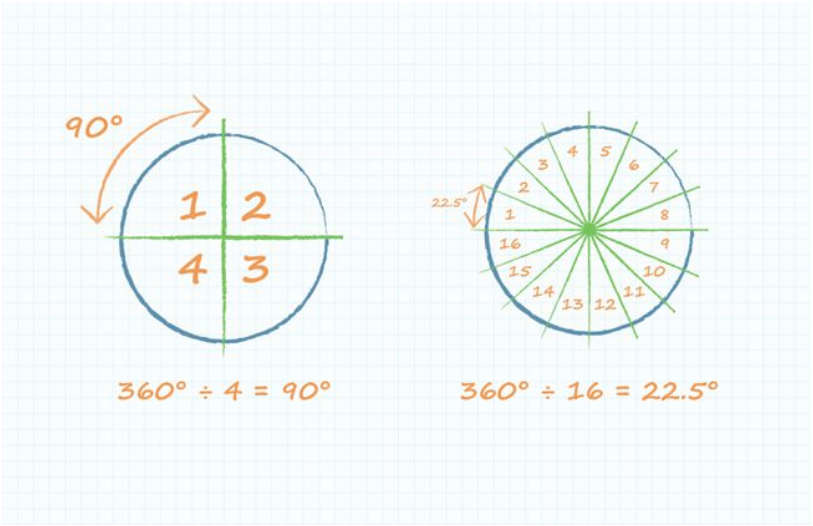


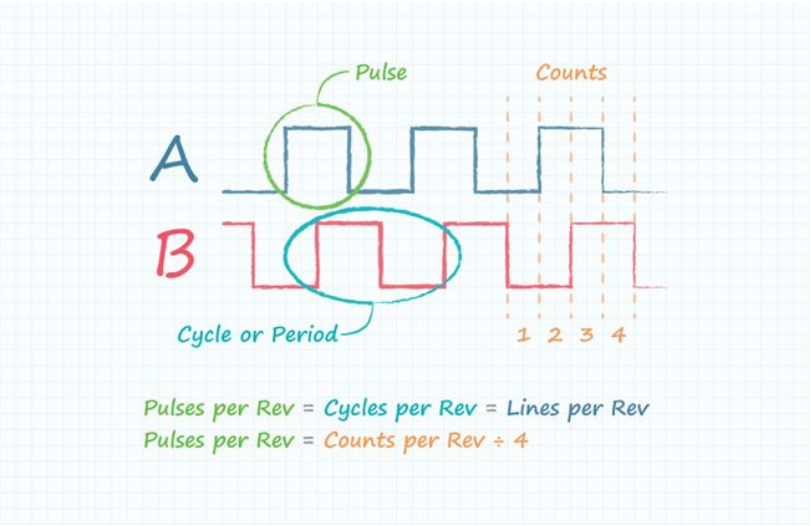



# Pololu 4805 Motor



# PPR & CPR




## Pulses Per Revolution:

describes the number of high pulses an encoder will have on either of its square wave outputs A or B over a single revolution.

## Counts Per Revolution:

refers to the number of quadrature decoded states that exist between the two outputs A and B

# Encoder Wiring

| Color | Function |
| --- | --- |
| Red | motor power (connects to one motor terminal) |
| Black | motor power (connects to the other motor terminal) |
| Green | encoder GND |
| Blue | encoder Vcc (3.5 V to 20 V) |
| Yellow | encoder A output |
| White | encoder B output |

# Wheel Speed Computation

1. Time “Counts Per Second”
2. Revolutions Per Second = Counts Per Second / Counts Per Revolution
3. Shaft Speed = Revolutions Per Second / Gear Ratio = Wheel Angular Speed
4. Wheel Linear Speed = Wheel Angular Speed * Wheel Radius

# Example: Measure Wheel Linear Speed

```
from machine import Pin, PWM
from time import sleep
from motor_driver import MotorDriver  # self developed

# SETUP
motor = MotorDriver(DIR_pin=4, PWM_pin=2)
enc_a = Pin(10, Pin.IN, Pin.PULL_UP)
enc_b = Pin(11, Pin.IN, Pin.PULL_UP)
counter = 0  # counts hall sensor trigger
def trig_handle_fn(pin):  # enc_a and enc_b share this function
    global counter
    counter += 1
enc_a.irq(
    trigger=Pin.IRQ_RISING | Pin.IRQ_FALLING,
    handler=trig_handle_fn
)
enc_b.irq(
    trigger=Pin.IRQ_RISING | Pin.IRQ_FALLING,
    handler=trig_handle_fn
)
```

# Example: Measure Wheel Linear Speed

```
# LOOP
from math import pi
prev_counter = 0
time_step = 0.1
CPR = 48  # counts per revolution, on original motor shaft
GEAR_RATIO = 46.85
WHEEL_RADIUS = 0.0325
motor.forward(speed=0.5)
for _ in range(50):  # let motor spin 50 * 0.1 = 5 seconds
    n_trigs = (counter - prev_counter)  # how many triggers happened after previous
loop
    rps_m = n_trigs / CPR / time_step  # motor revs per second
    rps_w = rps_m / GEAR_RATIO  # wheel revs per second
    ang_spd = rps_w * 2 * pi  # revs/s --> rad/s
    lin_spd = ang_spd * WHEEL_RADIUS
    prev_counter = counter  # IMPORTANT! get ready for next iteration
    print(f"wheel linear speed: {lin_spd} m/s")
    sleep(time_step)

motor.stop()
```