Introduction

Robot: A programmable machine that can perform tasks and interacts with its environment autonomously, without the aid of humans

Robot: Components: Control system (Brain), Actuators (Muscles), Sensors (Perception), Mechanism/Drivetrain (Bones/Tendons)

Robot System: sensing (observation)-> logic(command) -> motor -> new observation

Mobile Robots

Wheeled robot: balancing problem - increase payload 增加负载, adapt terrain 适应地形

Legged robot: balancing problem, gait problem(leg coordination), but can adapt to more complex terrain compared to wheeled robot

Aerial robot: diverse flying mechanisms, battery problem, can work avoid obstacles in the air and can search

Underwater Robots: Diverse bio-inspiration, Localization problem – light, current (accuracy decreases)

Manipulator robots (mostly generate object's spatial displacements)

- **Rigid robot arm:** High precision and stability; High payload; Limited flexibility and adaptability
- **Soft robot arm:** High flexibility and adaptability; Inherent safety and compliance; Low precision and payload

Applications

Safety: Bomb defusal ; Transportation; Surveillance 1. Protect people 2. Reduce labor

Service: Robots to vacuum/clean floors; Robots to assist in household, commercial building, and sports; Robots for homecare; Sewer/pipes inspection(1. Operate in confined spaces 2. Operate in hazardous environments)

Agriculture **Medical Care**:Surgery, rehabilitation, assistive robots

Control systems

Microcontrollers are mostly used for low-level behaviors such as: Driving motors, Reading sensors, General digital I/Os, Simple logic (e.g. counting), Serial communication

Often described as single-chip computers

Smaller, cheaper, slower, etc.

Microcontroller units (MCU) are developed for **embedded applications**

- turn off electronic devices under a certain condition (e.g. after a period of time or when certain temperature is reached)

- They are found in home appliances, consumer electronics, automobiles, smartphones/tablets

Programming a Microcontroller

Programs are generally written:– Assembly language (more efficient but hard to understand) – "C-like" programming language (easier to understand, but cannot run as fast as with assembly language)

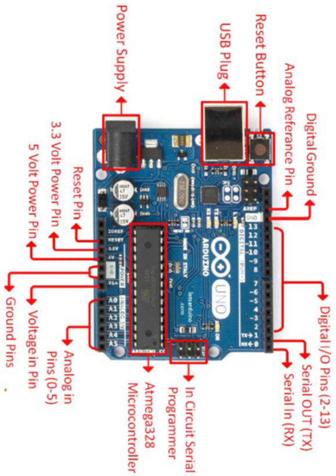
Arduino as a Platform

-Easy to learn and use; Widely used

- **Open-source** electronics prototyping platform (open hardware, open software)

Specifications:– Digital I/O Pins: 14 (6 with PWM output) – Analog Input Pins: 6 – USB port (to load programs) – Multiple OS (Windows, OS X, Linux)

The word "Arduino" can mean: A physical piece of hardware ;A programming environment ; A community & philosophy



Basic Structure

All programs have at least two main parts:

– **setup()** is called only once when your program starts. Use it to initialise pin modes, begin serial communication, etc.

– **loop()** repeats (i.e., loops) continuously after running setup(). Use it to read sensors, enable output pins, drive motors, etc.

x++; x+1 and returns the old value

++x; x+1 and returns the new value

x = 2;

y = ++x; // x = 3, y = 3

y = x--; // x = 2, y = 3

Predefined Constants

– **HIGH** and **LOW** (defines the state of pins)

– **INPUT** and **OUTPUT** (configures pin modes)

– **LED_BUILTIN** (is the pin attached to the LED)

– **true** and **false** (mean 1 and 0, respectively)

Configure Pins as Inputs or Outputs

Digital pins can be used as both inputs and outputs

Use **pinMode** (inside the setup function) to configure the pins:

pinMode (PinNumber, INPUT or OUTPUT)

Pin Level HIGH

-When a pin is configured as INPUT (i.e., measurement), Arduino will report as HIGH if: Voltage > 3V at the pin

-When a pin is configured as OUTPUT (e.g., for control) and set to HIGH, the pin is at 5V

Pin Level LOW

-When a pin is configured as INPUT (i.e. measurement), Arduino will report as LOW if: Voltage < 1.5V at the pin

-When a pin is configured as OUTPUT (e.g. for control) and set to LOW, the pin is at: 0V

Variable's Qualifiers

- **static**: persist beyond the function call, preserving their data between function calls.
- **const**: defines a variable as "read only", i.e., it can be used but its value cannot change.

1.Functions

A modular pieces of code that performs a task

Once the instructions in the function are executed, the program returns to the area of code from which the function is called. Enables the same action to be performed **multiple times**.

Advantages of Functions

- Help the program stay organized
- Make the main program smaller and more efficient
- Make it easier to reuse code in other programs (create libraries)

2. Digital Inputs/Outputs

What is I/O?

– Stands for Input and Output

– Input: Reading (measuring) an external signal

– Output: Sending out a control signal or action

Logic Levels

- High, or 1, or true; Low, or 0, or false

Two popular voltage levels:

– TTL (Transistor-Transistor Logic) and CMOS (Complementary Metal-Oxide-Semiconductor)

	L voltage	H voltage	Notes
CMOS	0 V to 1/3 V_{DD}	2/3 V_{DD} to V_{DD}	V_{DD} = supply voltage
TTL	0 V to 0.8 V	2 V to V_{CC}	V_{CC} = 5 V ±10%

Arduino's Digital I/O Functions

– **pinMode()** configures pins to behave as either inputs or outputs

– **digitalRead()** measures a discrete input (e.g., from a push-button)

– **digitalWrite()** activates a discrete output (e.g., to control a motor)

```
void setup() {
pinMode(out_Pin, OUTPUT);
pinMode(in_Pin, INPUT); }
void loop() {
val = digitalRead(in_Pin);
if (val == true) digitalWrite(out_Pin, HIGH);
else digitalWrite(out_Pin, LOW); }
```

Robot Behaviour with Discrete I/Os

-Simple motion rules can be implemented

Individual level: use the discrete contact information (input) to change direction and scape from obstacles/predators (output)

3. Analogue to Digital Converter (ADC)

-Microcontrollers (and thus robots) use digital signals; they speak like this:

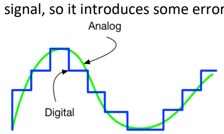
10110010 (binary), instead of using decimal or natural languages

-An ADC is a device that converts a continuous signal (usually voltage) to a digital number that represents its amplitude

Examples of Analogue Signals: Position, Orientation, Velocity, Contact force, Temperature, Air pressure, Humidity

Conversion Error

- The conversion involves quantization of the signal, so it introduces some error.

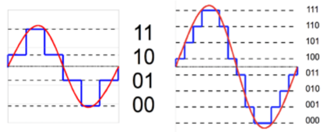


Resolution of the Converter

- The resolution of the converter indicates the **number of discrete values it can** produce over the range of analogue values

- ADC **approximates** the signal with a finite set of levels

2-bit and 3-bit Resolution



Arduino's analogRead()

- The board has a built-in function to read analogue signals

- Arduino Uno has a **10-bit ADC**.

- Return integers from [0–1023]

- It can tell a voltage difference of 4.9e-3Volt when the voltage reference is at 5Volt

– Why? 5/1024

10-bit → 2^{10} → 5Volt / 2^{10} = 0.0048828125

Example of Analog Input

```
int analoguePin = 0, val = 0; // declare variables
void setup() {
Serial.begin(9600); // setup serial}
void loop() {
val = analogRead(analoguePin);
Serial.println(val);
delay(500);
}
```

4. Pulse-Width Modulation (PWM)

A Quasi-Analogue Output

- A technique for getting analogue results by digital means - Creates a square wave (方形) by switching ON / OFF - Simulates voltages in the range of full 5V and 0V by modulating the time the signal is ON

Average voltage is used for controlling devices:

- 100% duty cycle: full signal
- 50% duty cycle: "medium" signal
- 0% duty cycle: no signal

Duty cycle (%) = time high/ total period * 100

Applications of PWM Signals: Servomotors (position control), DC motor speed control, Servo-valves, Hydraulics, LED brightness, Telecommunications, Voltage regulation, Audio synthesis

Control brightness of a LED

-PWM wave of the specified duty cycle

-Syntax: **analogWrite**(pin_number, duty_cycle)

– Pins (UNO): 3, 5, 6, 9, 10, 11

– Duty cycle: from 0 (always off) to 255 (always on)

Example: analogWrite(5,127); // 255/2 ~= 127, 50% duty cycle

Example of PWM Output

```
int ledPin = 9; // declare variables
void setup() { }
void loop() {
for (int fade = 0 ; fade <= 255; fade += 5) { // fade min to max
analogWrite(ledPin, fade); // write value
delay(30); // wait }
```

5. Serial Communication

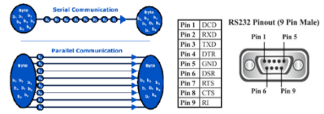
-The process of sending data one bit at a time, sequentially, over a communication channel

- General format for **serial transmission**:

– Start bit + Data bits + Parity bit (optional) + Stop bit

Serial vs Parallel Communication

In contrast, parallel communication sends binary digits simultaneously



Serial Port: RS-232 connector (industrial standard)

Arduino's Serial Communication

- Used for communication of the board with computers and other devices
- Set-up the port with: **Serial.begin(**speed)

– speed determines the data rate (usually 9600 bps)

- Print data to serial port with: **Serial.println(**var)

– Prints the variable var followed by a carriage return \r (i.e., prints data in a new line)

Serial.print 不会换行(human readable ASCII)

- **Serial.write(…)** Sends the raw binary value(s)
- Check if data arrives with: **Serial.available()**

– Gets the number of characters; 0 if none arrives

- Read incoming data with: **Serial.read()**

– Returns the 1st byte of data available, or -1 if none

Sensors

➢ responds to a physical stimulus (such as heat, light, sound, pressure, magnetism, or a particular motion)

➢ transmits a resulting signal (as for observation or measurement)

Sensors in Robot

Detect physical quantities such as

– Position, orientation, velocity, acceleration (e.g., Inertial Measurement Unit (IMU))

– Distance, size, height – Force, torque, pressure

– Temperature, luminance, weight, etc.

Transforms them into (typically) electrical signals (e.g., voltage)

Classification in function: distance, vision, sound, touch

Roles of sensor in robotics: understand the configuration of robot and the external env

Sensor classification:

Proprioception/internal (measure the robot's state eg, Position,v, acceleration, orientation, joint torque)

Exteroception/external (vision, range, contact, acoustic, temperature, humidity, radiation

Sensing measurement can be:

Inaccurate provide a measurement that is not the real value; drifting), **Noisy** (unwanted variation; needs filtering),**Nonlinear** (requires calibration)

->Combining measurements of several sensors is beneficial.

Characteristics

All sensors are characterized by:

– Sensitivity: how "small" it can detect changes in the input

– Linearity: how "straight" the input-output relation is

– Accuracy: difference between real quantity and the measured one

– Resolution: smallest unit measurement to output/display

– Response time: how fast it can respond to change in the input

– Measurement range: the whole spectrum it can detect

Internal

Robot Joint Position

For robots, we need to measure the rotation angle of the motor shaft (Potentiometers, optical encoders). For that, there are (typically) two types of **joint position sensors**:

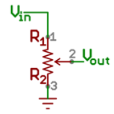
1.Potentiometer = **variable resistance**

-We measure the joint's position by measuring the **varying voltage**

-**Mechanically couple** the pot to the motor's shaft

-V out is proportional to the motor's position

Disadvantage: Finite measurement range and resolution



原理

2.Optical Encoder

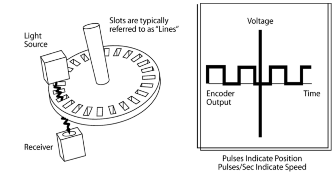
- It consists of a light beam, a light detector, and a rotating disc with a radial grating

- The disk has black lines and clear spaces

– **Line:** cuts the beam → LOW signal

– **Space:** allows the beam to pass → HIGH signal

● Measure the angle by **counting pulses**



Quadrature Optical Encoders

– Channel A: train of pulses

– Channel B: train of pulses

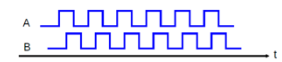
– Channel C: single pulse per turn

– The phase difference between A and B is 90 deg.

– The C pulse is used as a reference to detect absolute position

If Chanel A is leading, clockwise; if Chanel B is leading, anti-clockwise

– If Channel A is leading, the rotation is clockwise



Resolution: s = 360 / number of lines

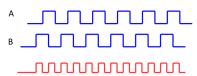
- The smaller the resolution, the better the measurement

How to improve the resolution?

– Increase the number of lines →manufacturing cost increases

– Evaluate the two train of pulses (i.e., Channels A and B) by computing Boolean operations (XOR) on its signals → The cheap way since it can be done with software (The resolution is **doubled**)

A	B	Out
0	0	0
0	1	1
1	1	0
1	0	1



Torque Sensors

- A device used for measuring the torque (moment) of a joint

• Torque can be measured by detecting the strain changes on the motor's shaft where T, r, and G are torque, shaft radius, and shear modulus

$$\varepsilon = \frac{T}{\pi r^3 G}$$

• In practice, the magnitude of measured strain is very small.

Strain: Definition and Measurement

-Strain is the amount of **deformation** of a body due to an applied force. More specifically, strain (ε) is defined as the **fractional change** in length

• Strain gauge is a device whose electrical resistance varies in proportion to the amount of strain in the device

Strain Gauge: Working Principle

• As the object deforms, its **resistance changes**

• Gage factor GF is the ratio of electrical strain (fractional change in resistance) to mechanical strain

$$GF = \frac{\frac{\Delta R}{R}}{\frac{\Delta L}{L}} = \frac{\frac{\Delta R}{R}}{\varepsilon}$$

where R is the resistance and L is the strain gage length. The higher GF means it is **easier** to detect resistance changes

• **Metallic strain gauges:** the Gage Factor is typically around 2

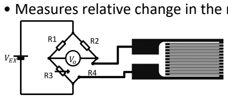
• **Semiconductor strain gauges:** the Gage Factor can be 50 or more

• Measuring strain is **challenging** because the strain rarely involve quantities larger than a few millistrain (ϵ × 10−3) and temperature affects. $\Delta R = GF \cdot R \cdot \epsilon$

Wheatstone Bridge

• Two parallel voltage divider circuits

• Measures relative change in the resistance



$$V_o = [\frac{R_4}{R_2+R_4} - \frac{R_3}{R_1+R_3}] \times V_{EX}$$

• When the wheatstone bridge is balanced, $R1/R3$=R2/R4, $Vo = 0$

• When there is a strain-induced change in resistance, ΔR,

$$\frac{V_o}{V_{EX}} = -\frac{GF \bullet \varepsilon}{4}\left(\frac{1}{1 + GF \bullet \frac{\varepsilon}{2}}\right)$$

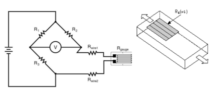
R4 = ΔR + R if R4=R originally

Quarter Bridge Strain Gage

Type I:

• Measure axial or bending strain
• R4 is the only active strain gage used to measure tensile strain
• The other three resistors in the wheatstone bridge are dummy resistors with known resistances.
• Assuming $R \gg \Delta R$,

$$V_{out} = \frac{\Delta R}{4R} V_{EX} = \frac{V_{EX}}{4} \cdot \frac{\Delta R}{R} = \frac{V_{EX}}{4} \cdot GF \cdot \varepsilon$$

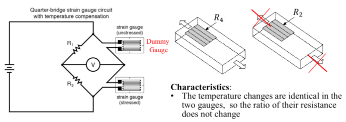


Characteristics:
• Prone to errors caused by temperature variations

Type II:

• R2 is not bonded to the substrate (dummy strain gauge)
• R4 is still the only active strain gage
• Compensate for effect of temperature

$$V_{out} = \frac{\Delta R}{4R} V_{EX} = \frac{V_{EX}}{4} \cdot GF \cdot \varepsilon$$



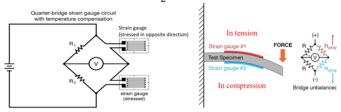
Characteristics:
• The temperature changes are identical in the two gauges, so the ratio of their resistance does not change

Half Bridge Strain Gage

• R2 and R4 are both active and experience strain of the same magnitude but different directions.
• Compensate for effect of temperature
• Double the sensitivity compared with quarter bridge

$$V_{out} = \frac{V_{EX}}{4} \cdot \frac{\Delta R_2 - \Delta R_4}{R} = \frac{V_{EX}}{4} \cdot GF \cdot (\varepsilon_2 - \varepsilon_4)$$

$$V_{out} = \frac{V_{EX}}{2} \cdot GF \cdot \varepsilon$$

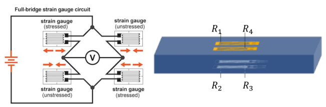


Full Bridge Strain Gage

• 4 active strain gauges
• Strain gauges of same strain direction placed diagonally.
• Compensate for effect of temperature
• Double the sensitivity compared with half bridge

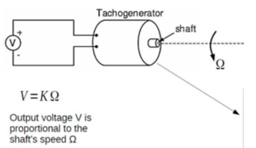
$$V_{out} = \frac{V_{EX}}{4} \cdot (\frac{\Delta R_1}{R_1} - \frac{\Delta R_2}{R_2} + \frac{\Delta R_4}{R_4} - \frac{\Delta R_3}{R_3}) = \frac{V_{EX}}{4} \cdot GF \cdot (\varepsilon_1 - \varepsilon_2 + \varepsilon_4 - \varepsilon_3)$$

$$V_{out} = V_{EX} \cdot GF \cdot \varepsilon$$



Velocity Measurement

A **tachometer** is a sensor to measure speed (the simplest case is just a voltage generator)
Attach a **tachometer** to the motor's shaft



Contact-type Tachometer

• A device used to measure the **speed of a rotating object**, such as a motor or shaft
• A rotating coil between permanent magnets
• According to Faraday's Law, voltage produced is proportional to coil speed which in turn is proportional to the shaft speed
• $EMF = -N\frac{\Delta\Phi}{\Delta t} = -N\frac{\Delta BA}{\Delta t}$
  • N = number of turns
  • Φ = magnetic flux
  • B = external magnetic field
  • A = area of coil

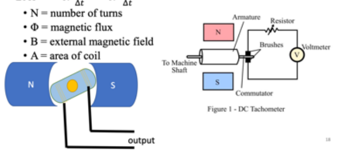


Figure 1 - DC Tachometer

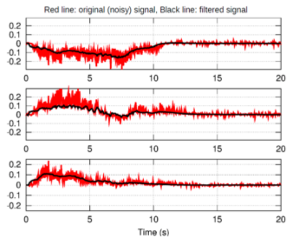
Inertial Measurement Unit (IMU)

An Inertial Measurement Unit (IMU) typically consists of the following key components:
1. Accelerometer(s): Measures linear acceleration along one or more axes (typically x, y, and z).
2. Gyroscope(s): Measures angular velocity (rotation) around one or more axes
3. Magnetometer (optional, in some IMUs): Measures the magnetic field around the device to help correct orientation, especially in the absence of a reference (like gravity).
2. Typically used for heading or compass functionality.

Velocity: Numerical Differentiation

- Advantages: Simple, without using additional sensors
- Disadvantages: Noisy signals
– Can be improved by using low-pass filters

Signal Filtering: Low-Pass Filtering



External Sensors

Vision Sensors:

-Cameras • Camera images provide environmental information at a pixel level

• Using segmentation and classification algorithms to understand environments

-Multi-camera Motion Capture Systems

• Using multiple **synchronized cameras** to track **reflective markers** placed on a subject, reconstruct 3D motion through triangulation and advanced algorithms

•Offers **high precision**, **real-time data capture**, and the ability to analyze **complex movements**

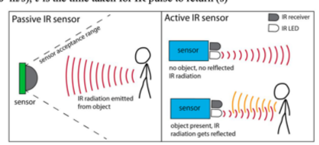
Range (position) Sensors

Infrared (IR) Sensors

• Infrared (IR) sensors detect **infrared radiation** (wavelengths typically from 700 nm to 1 mm) emitted or reflected by objects. They are widely used for proximity sensing and motion detection
• For an IR sensor using time-of-flight (ToF) principle:

$$d = \frac{c \cdot t}{2}$$

where d is the distance to the object (m), c is the speed of light (3.0×10^8m/s), t is the time taken for IR pulse to return (s)



Range (position) Sensors: Ultrasonic Sensors

• The ultrasonic sensor emits a **high-frequency sound wave** (typically in the range of 20 kHz to several MHz) from a piezoelectric transducer
• The basic equation for calculating distance using ultrasonic sensors is based on time-of-flight and the speed of sound:

$$d = \frac{v \cdot t}{2}$$

where d is the distance to the object (m), v is the speed of sound (343 m/s in air), t is the time taken for ultrasonic pulse to travel to the object and back (s)

• Range between 12 cm up to 5 m
• Resolution of around 2 cm
• Sound beam propagates in a cone
• Soft surfaces that absorb most of the sound energy

Should the strain gauges be mounted at a specific angle such as 45 degree? Why

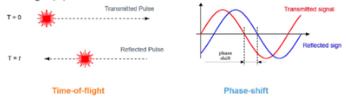
Yes. Under pure torsion a circular shaft is subjected only to shear stresses (no normal stress), and the principal strains are at ±45° to the shaft axis. Mounting the gauges at 45° maximizes their sensitivity to the torsional shear; gauges aligned with the shaft axis or circumferentially would register zero normal strain under pure torsion.

Range (position) sensors: Laser Range Finder

• In some laser range sensors, the emitted laser light is modulated at a known frequency. When the laser beam travels to the target and reflects back, it experiences a **delay**, which results in **a shift in the phase** of the modulated wave
• Phase shift sensors use modulated laser light and measure the phase difference between the emitted and received light:

$$d = \frac{\Delta\phi \cdot \lambda}{2\pi}$$

where d is the distance to the object (m), $\Delta\phi$ is the phase shift between the emitted and reflected laser light (radians), λ is the wavelength of the emitted laser light (m)



Comparison Between Time-of-Flight (ToF) and Phase-Shift Laser Range Sensors

Feature	Time-of-Flight (ToF)	Phase-Shift
Principle	Measures the time taken for a laser pulse to travel to the object and return	Measures the phase shift between the emitted and received modulated laser signal
Accuracy	Moderate to high, but affected by noise at longer distances	Very high, as phase shift measurements can detect small changes in distance
Resolution	Limited by the precision of time measurement (typically in nanoseconds)	Higher resolution since phase shifts can be measured with great precision
Range	Suitable for long-range measurements (meters to kilometers)	More suitable for short-to-medium range (up to a few hundred meters)
Speed	Very fast, capable of measuring distances in real-time	Slower compared to ToF due to complex signal processing

Contact Sensors

Tactile Feedback

• A tactile sensor is a device that measures information arising from physical interaction with its environment.

• What does it sense? Deformation of bodies (strain)

• How does it work? It detects changes in physical properties (e.g., resistance, capacitance, or optical signals) caused by external forces or deformations. Then contact force can be estimated based on the changes

Optical Tactile Sensors

• The sensor typically uses light sources (such as LEDs) and light detectors (like photodiodes or photodetectors) embedded within the sensor.

• When pressure or force is applied to the sensor surface, it can alter the optical path or change the reflection or transmission of light.

Robotic Applications

Robot Application

• With a tactile sensor, the gripper can understand how much it is applying to the egg and stop closing the gripper before breaking the egg

• The tactile sensor could also precisely sense the position of the egg and the contacting area, which can be for monitoring slipping

Actuators

• A device that converts a controller command signal into a change in a **physical parameter** • The change is usually mechanical (e.g., (joint) position or (joint) velocity) • An actuator is usually activated by a low-level command signal, so an amplifier may be required to provide sufficient power to drive the actuator

Rotary Actuators

•Convert **energy into a rotary motion** through a shaft that controls the attached equipment's position and speed• Continuous rotational motion • The resulting motion is **not restricted** by the distance they travel (infinite rotation angle)

• An **electric motor** is a classic example: - High and constant torque during full angle rotation (may need a gear) - Any degree of rotation, from 0 degree to full 360 degrees, is achievable

Linear Actuator

• Linear actuators move an object along a straight line - usually in a **back- and-forth** manner (前后动). • Translational motion • The resultant motion **is restricted** by the distance they travel

• Linear actuators are normally applied when loads need to be pushed, pulled, lifted, lowered, positioned, or rotated. Such movements are often required in the following industries:
  • High repeatability and positioning accuracy
  • Mounting, integration, and operation
  • Can withstand harsh and adverse conditions and environments

Convert Motion Between Rotary and Linear

• Generate linear motion from rotational motion: screw-and-nut system, belt and pulley, rack and pinion, cam-follow mechanism, worm and worm wheel, crank shaft system

• Generate rotary into linear motion: crank shaft system, rack and pinion

Actuators in Robotics

We need actuators to create force and displacement in the robot. – Change its configuration – Manipulate objects – Navigate around the environment

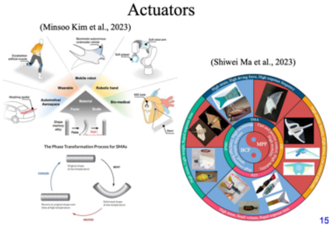
• Typically controlled with **electrical signals**; they generate **mechanical motion**

• A "bridge" between the numerical algorithms and the physical world algo –(control sig)> actuator –(mechanical motion)>physical world

Classification of Actuators

– Electric actuators
– Hydraulic actuators
– Pneumatic actuators
– Smart actuators (Shape memory alloys (SMA), Piezo (ultrasonic) motors, Magneto-rheological or electro-rheological fluid)

Robotic Application Example on Smart Actuators



Comparison Between Electric and Hydraulic Motors

Aspect	Electric Motor	Hydraulic Motor
Power Source	Electricity (AC or DC)	Hydraulic fluid (pressurized oil)
Energy Efficiency	High efficiency (90-95%)	Lower efficiency (60-70%) due to fluid losses
Torque Output	Consistent torque at low speeds	High torque at low speeds
Speed Range	Wide speed range, easily controllable	Limited speed range, depends on fluid flow
Precision Control	High precision with electronic control systems	Less precise, depends on hydraulic system
Maintenance	Low maintenance, fewer moving parts	Higher maintenance, fluid leaks, and wear

Pneumatic (氣動) Actuators

Advantages: Simplicity and Reliability; Fast Response and High Speed; Safety and Clean Operation; Cost-Effective
Disadvantages: Limited Precision and Control; Energy Inefficiency; Noise and Air Leaks; Limited Force and Speed Control

Meshing Gears

• Gear diameter(d) = # of teeth(n) x Module (m)

• For gears to mesh, they must have the **same module**. (module 表示齿轮齿的大小 mm)否則會導致 poor meshing, increased wear, noise, and mechanical failure

• To prevent slippage, the **instantaneous linear velocities** at the point of contact must be the **same** for meshing gears.

Gear ratio = output gear teeth/input gear teeth
= output gear radius / input gear radius
= output torque / input torque
= input angular velocity / output angular velocity

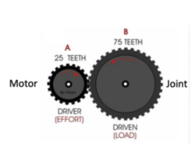
Gear Reduction 减速 e.g., a gear box

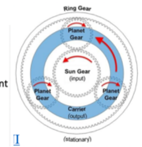
• Decrease output speed

• Increase output torque 牺牲速度换取更强的力量

• The output encoder's resolution will be increased n times, where n is the gear ratio

• n turns of the motor shaft 只等於 one turn of the joint.





Planetary Gear (Epicyclic Gear)

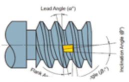
• Input = sun gear • Output = carrier

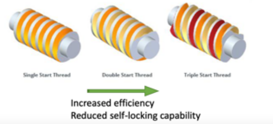
• Gear ratio = $sum\ of\ \#\ teeth\ of\ sun\ and\ ring\ gears\ /\ \#\ of\ teeth\ on\ sun\ gear$
$=angular\ velocity\ of\ sun\ gear\ /\ angular\ velocity\ of\ carrier$

Could have six different gear ratios!

Worm Drive 通过滑动接触传递动力

- Worm wheel (worm gear) and worm
- Compact and used for **right-angled** applications
- **Gear ratio** = # *teeth on worm wheel / # of starts on the worm* 蜗杆上的**螺旋线数量**
- Wide range of gear ratios from 4:1 to 300:1
- Transmission via sliding between the flanks of the worm and the worm gear
- Higher magnitude but less efficient power transmission due to sliding contact between teeth
- Does not backdrive (self-locking)
  - typically in higher ratios
  - low lead angle





Electric Actuators

-Machine that converts electrical energy into mechanical energy

-Operate with DC/AC power

Common Types of Electric Actuators

- DC motors (Brush/ Brushless)
- AC motors
- Stepper motors
- RC servo motors

-Advantages: Easy to control; Accurate; Fast response; Clean

-Disadvantages: Relatively low power output (compared to hydraulics)

DC motor

Permanent Magnet DC Motors

- Input: DC (direct current, or voltage)
- Structure – Stator: fixed part of the motor with magnets – Rotor: rotating part of the motor with coils

•Underlying Principle: An electric current is passed through a conductor (usually a coil of wire) placed within a magnetic field; a force is exerted on the wire causing it to move.

Brushless DC Motors

- A brushless motor uses a permanent magnet as its rotor and used a triple phase of coils. The sensory tracks rotor positioning and sends signals to the controller which in turn activates the coil in a structured manner.
- **Advantages (vs. brush)**
- long service life; low noise level;
- high-efficiency indicators - about 90%;
- maximum rotation speed is achieved quickly;
- do not spark form during operation;
- additional cooling resources are not required;
- no mechanisms that need regular maintenance.

Comparison Between Brush and Brushless DC Motors

Parameter	Brush DC Motors	Brushless DC Motors (BLDC)
Commutation	Mechanical (using brushes and commutator)	Electronic (using sensors and controllers)
Efficiency	Lower efficiency due to friction and losses	Higher efficiency (no brush friction)
Maintenance	High (brushes wear out and need replacement)	Low (no brushes, minimal wear)
Lifespan	Shorter (due to brush wear)	Longer (no brushes to wear out)
Speed Range	Limited by brush and commutator	Wider speed range
Noise	Noisier (brush friction and arcing)	Quieter operation
Cost	Lower initial cost	Higher initial cost (due to electronics)
Control Complexity	Simple (no external controller needed)	Requires electronic controller
Size and Weight	Larger and heavier for the same power	Compact and lightweight

Back Electromotive Force (EMF) in DC Motors

- Steady-state voltage equation:

$$V = I_a R_a + V_{EMF}$$

- V = *supply voltage* • Ia = *armature current*
- Ra = *internal armature resistance*

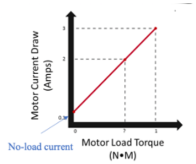
$$V_{EMF} = \frac{P\emptyset Z}{60A}\omega = k_e\omega$$

- $V\ EMF$= *induced EMF*
- ω = *motor speed* • P = *number of poles*
- $\emptyset$ = *useful flux per pole* • Z = *total number of conductors in armature* • A = *number of coils in armature*

Practical scenarios:• At no load, torque is low,I is low.VEMF is close to V.• When applying sudden load, speed reduces; V reduces; I increases; torque increases to support the load.

Motor torque

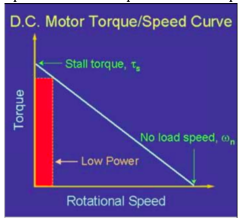
The output torque can be approximated with torque constant kt. $\tau \approx kti$



- No load current is used to overcome internal resistance.

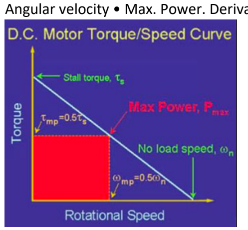
Torque-Speed Curve

- Relation between Torque and Rotational Speed. • Stall Torque • No load Speed



Motor Power

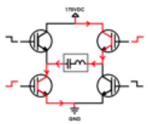
- Power = Voltage × Current; Power = Torque × Angular velocity • Max. Power. Derivation???



H Bridge

- Circuit (4 switches) that inverts the polarity of the (fixed) control voltage

By activating S1–S4, the bridge can drive the motor in both directions



S1	S2	S3	S4	Turn
1	0	0	1	Left
0	1	1	0	Right

Control of DC Motors

– Open-loop control (no measurement)

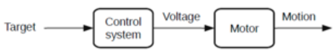
– Feedback control (regulation of a variable)

Open-loop Control

- No sensors are used

- Try to control the speed or torque based on model estimations

- Problem: We cannot precisely set the speed or torque

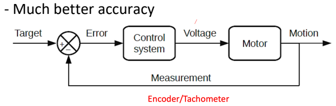


Closed-loop Control

- We use sensor feedback (i.e., measurements) to correct the control action

- We can control speed, position, and torque

- Much better accuracy

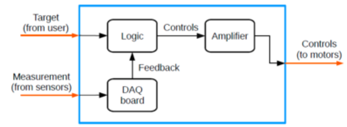


Control Systems for DC Motors

A generic motion control system may have:

– Control logic/computing (PC or even Arduino)

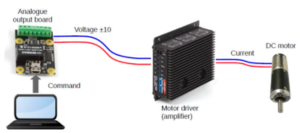
– Power amplifier/motor driver (to drive the motor) – Measurement board (to read sensors)



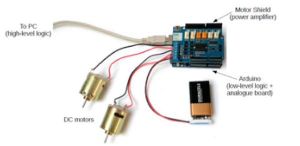
Industrial Motor Driver

-Power amplifier (digital controls are small)

-Driving current is proportional to the input voltage



Arduino Solution



Stepper Motors （方形）

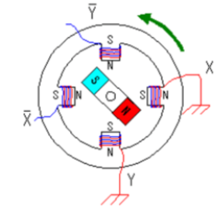
- Electromechanical device that converts a series of pulses into discrete rotational motion
- Recognizable for its distinctive square shape• Rotors (permanent magnet, variable reluctance, and hybrid) • Stator (2-phase, 3 phase, etc.)

Feature	Permanent Magnet (PM)	Variable Reluctance (VR)	Hybrid
Rotor Material	Permanent magnet	Soft magnetic material (no permanent magnet)	Permanent magnet + toothed structure
Step Angle	Larger (e.g., 7.5° or 15°)	Smaller (e.g., 1.8° or less)	Very small (e.g., 1.8° or less)
Resolution	Low	Moderate	High
Low-Speed Torque	High	Moderate	High
High-Speed Performance	Poor	Good	Good
Cost	Low	Moderate	High
Applications	Low-speed, low-precision devices	High-speed, moderate-precision devices	High-precision, high-performance devices

Working Principle

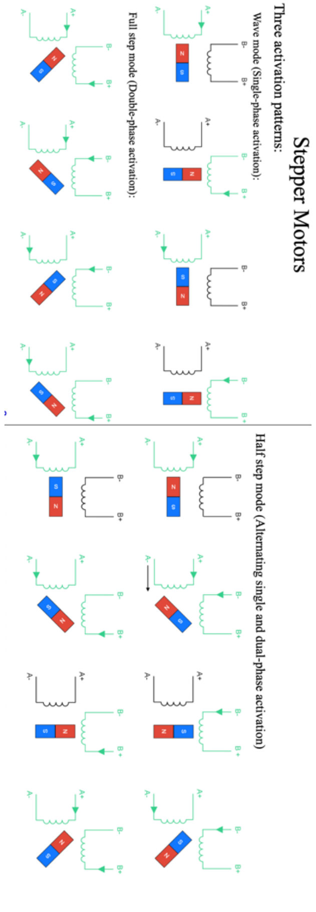
- Controlled by a **combination of digital pulses**
- Coils in the stator are turned ON/OFF

– The rotor will align with the magnetic field – The shaft rotates with a fixed angle



X	X̄	Y	Ȳ
1	0	1	0
0	1	1	0
0	1	0	1
1	0	0	1

•High number of teeth on rotor (50 teeth) and stator (48 teeth) • Typically feature 200 steps for 1 revolution • 1.8° per step • Each phase/coil set takes a turn to be activated • Each time a coil is energized, the magnetic field direction reverses (polarity changes).



Advantages

- Precise positioning and repeatability (with no sensor feedback) - Error is no cumulative with rotations

- Excellent starting/stopping response - Simple to control (using digital signals) -Easy to achieve low-speed motion with load

Disadvantages

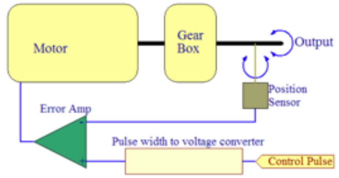
- No good for continuous, high-speed motions
- Low efficiency (compared to DC motor, current is independent of external load) • Heavy • Resonance can occur if not properly controlled (~200 Hz)
- No sensing feedback

Servo Motors

- Low-cost rotational motors for positioning
- Limited range

-Components • A DC motor • Gear trains

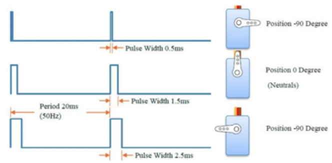
- A position sensor (potentiometer) • A built-in controller(black: ground, white: control signal, red: power supply 4.8V)



Working Principle

- The servo is controlled by PWM signals of 20ms

- The width of the pulse corresponds to the desired angle (pulse width =0.5ms: -90 度; 1.5ms : 0 度, 2.5ms: 90 度)



AC Motors (Induction motors)

- Input: Alternating current (AC)
- Consists of stator and rotor
- **no brushes, no commutator.**
- Basic operation: Stator windings are energized in sequence to create rotating magnetic field
- 3-phase power supply for the stator coils
- Types of AC motors: Synchronous motor (rotor with fixed polarity); Induction motor (rotor with magnetic field; resulted from induced current)
- Compared to DC motors, **advantages** are:

1. Mechanically simple and sturdy 2. Less maintenance, thus low cost 3. Better energy efficiency and less heating

- **Disadvantages**: 1. Can be heavy and bulky

Mechanisms

Robot Motion

Definition: The dynamic change of the robot's configuration

Robot Manipulators (Arms)

- Are electro-mechanical systems that are used for moving objects
- They consist in a series of segments (or links) joined by rotational/translational joints

Purpose of Manipulators

- Are intended to perform the tasks of a human arm (i.e. to reach and interact with objects)

-Parts:

base, joints, links, end-effector

-A manipulator forms a kinematic chain: Multiple links are interconnected by joints

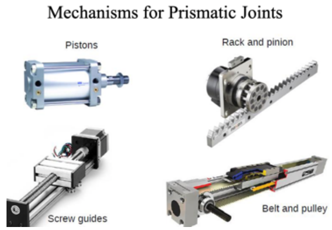
Base

- The fixed (static) part of the manipulator
- Supports the whole structure of the robot
- A reference frame is typically defined at the base
- Think of the base as the 0-th link

Joints

- Mechanical components that allow relative motions in the structure of a manipulator
- Two main types: revolute (i.e. rotational) and prismatic (i.e. translational)
- There is a joint range in practice

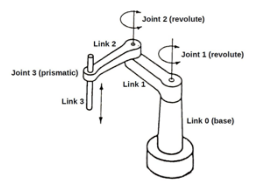
Mechanisms for Revolute Joints **Direct drive**: Pros: Accurate transmission; Cons: Heavy links; **Distant belt drive**: Pros: Lightweight Cons: Backlash



Links

- The mechanical components that are connected to / moved by the robot joints

- These are passive components (usually rigid)

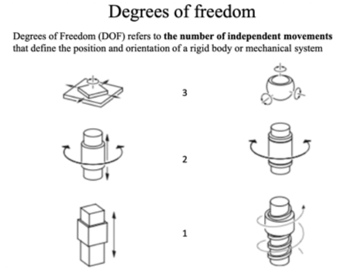


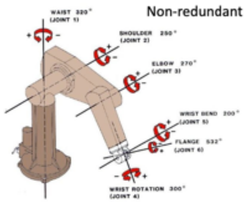
End-Effector

- Analogous to the human hand

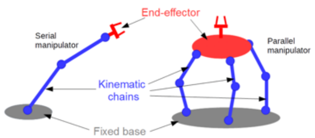
- Some examples of end-effector tools:
  – Grippers or fingered hands
  – Spray guns and welding torches
  – Surgical tools

Degrees of Freedom (DOF) refers to the number of independent movements that define the position and orientation of a rigid body or mechanical system





Two Main Types of Manipulators

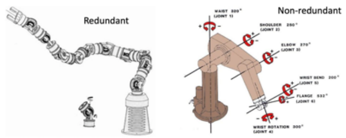


Serial manipulators (open kinematic chains)

- Commonly used in industry (pick-and-place)
- **Advantage:** Slim structures and compact base
- **Disadvantages:** – Errors accumulate (amplify) from link to link – Have to carry the weight of most actuators – Parallel manipulators (closed kinematic chains)

Number of Joints

- To control the end-effector's Cartesian position (x-y-z) and orientation (3 angles), a manipulator must have at least **6** joints
- Manipulators with **more than 6-DOF** are called **redundant** if the goal is to control the end-effector's Cartesian position (x-y-z) and orientation (3 angles)



Anthropomorphic Configuration

-Human-like arm (in terms of its motion)
-It has 7-DOF (shoulder 3, elbow 1, wrist 3) :Structure that allows to reconfigure the kinematic chain (null space motion)

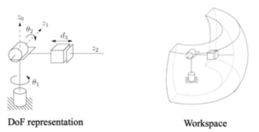
Null Space Motion

-Maintain the hand position and orientation while reconfiguring the internal kinematic chain
-Only achievable with **redundant** manipulators（可以繞過障礙物）

Workspace/Task space

-The space that can be reached with the manipulator's end-effector
-Determined by: • Link length • Joint arrangement • Joint limits

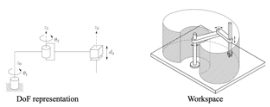
Spherical Manipulator (RRP)



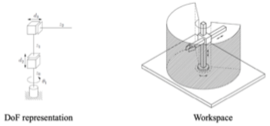
Read ultrasonic sensors

```
const int trigPin1 = 2; const int echoPin1 = 3; const int trigPin2 = 4; const int echoPin2 = 5;
void setup() {Serial.begin(9600); pinMode(trigPin1, OUTPUT); pinMode(echoPin1, INPUT); pinMode(trigPin2, OUTPUT);pinMode(echoPin2, INPUT);}
void loop() { float distance1 = readUltra (trigPin1, echoPin1); float distance2 = readUltra (trigPin2, echoPin2);
float diff = distance1-distance2;  Serial.print("Sensor 1: ");Serial.print(distance1, 1); Serial.print(" cm, Sensor 2: "); Serial.print(distance2, 1); Serial.print(" cm, Difference: ");Serial.print(diff, 1); Serial.println(" cm"); delay(200); }
float readUltrasonicCM(int trigPin, int echoPin) {  digitalWrite(trigPin, LOW); delayMicroseconds(2);digitalWrite(trigPin, HIGH); delayMicroseconds(10); digitalWrite(trigPin, LOW); // 读取返回脉冲持续时间
long duration = pulseIn(echoPin, HIGH, 30000);
float distance = (duration * 0.0343) / 2.0; return distance;}
```

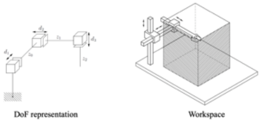
SCARA Manipulator (RRP)



Cylindrical Manipulator (RPP)



Cartesian Manipulator (PPP)



Parallel Manipulators

- Use several serial manipulators
- **Advantages:** carry heavy loads, higher structural rigidity, precise motion
- **Disadvantages:** large footprint, small workspace 占地面积大、工作空间小

Stewart platform

Generation of Motion

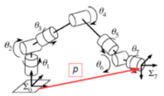
-The end-effector trajectory is determined by the total contribution of the joints' motion

Forwards Kinematics (FK)

-Compute the end-effector position
- A map of static relation between the joint and the end-effector motions
-Require knowledge of 1) joint arrangement/type, 2) link distances

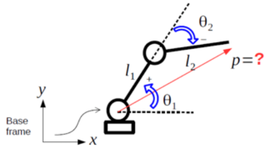
$\theta=[\theta_1,\dots,\theta_n]^T$ joint positions (measured)

$p=h(\theta)$ $\quad h:R^n \rightarrow R^m$ known mapping



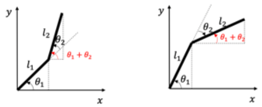
A 2-DOF Example

- FK: Given the vector of joint angles θ, find the vector of end-effector position p
- Consider the planar manipulator:



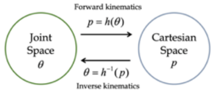
Geometric Solution

- The position p can be solved using basic **trigonometric functions**
  – e.g., decomposing p with triangles



$$p=\begin{bmatrix}x\\y\end{bmatrix}=h(\theta)=\begin{bmatrix}l_1\cos\theta_1+l_2\cos(\theta_1+\theta_2)\\l_1\sin\theta_1+l_2\sin(\theta_1+\theta_2)\end{bmatrix}$$

Forward Kinematics vs. Inverse Kinematics



Inverse Kinematics (IK)

- Inverse problem: Given p, find θ to achieve it
- IK are needed to enable the robot to follow a desired trajectory
- **Multiple** joint solutions may exist ➔ solve the problem at vertical level

Differential Kinematics

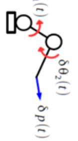
• Physically: It models the dynamic relation between the joint and the end-effector motions
• Mathematically: It represents the 1st order system obtained by differentiating the FK

$$\underbrace{\frac{dh}{dt}}_{\text{End-effector velocity}}=\dot p(t)=\frac{\partial h}{\partial\theta}(\theta)\underbrace{\dot\theta(t)}_{\text{Joint velocity}}$$

Jacobian Matrix (Analytical)

- Map the instantaneous joint velocities into end-effector velocities

$$\dot p(t)=J(\theta)\dot\theta(t)$$

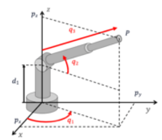


- It is the **partial derivatives** of the FK mapping function $h(\theta)$

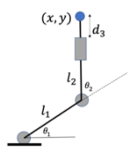
$$h(\theta)=\begin{bmatrix}l_1\cos\theta_1+l_2\cos(\theta_1+\theta_2)\\l_1\sin\theta_1+l_2\sin(\theta_1+\theta_2)\end{bmatrix}$$

$$J(\theta)=\frac{\partial h}{\partial\theta}=\begin{bmatrix}\partial p_x/\partial\theta_1 & \partial p_x/\partial\theta_2\\ \partial p_y/\partial\theta_1 & \partial p_y/\partial\theta_2\end{bmatrix}$$

$$J=\begin{bmatrix}-l_1\sin\theta_1-l_2\sin(\theta_1+\theta_2) & -l_2\sin(\theta_1+\theta_2)\\ l_1\cos\theta_1+l_2\cos(\theta_1+\theta_2) & l_2\cos(\theta_1+\theta_2)\end{bmatrix}$$



$$P_x=q_3\cdot cosq_2\cdot cosq_1$$
$$P_y=q_3\cdot cosq_2\cdot sinq_1$$
$$P_z=q_3\cdot sinq_2+d_1$$



$$x=l_1\cdot cos\theta_1+(l_2+d_3)\cdot\cos(\theta_1+\theta_2)$$
$$y=l_1\cdot sin\theta_1+(l_2+d_3)\cdot\sin(\theta_1+\theta_2)$$

What Is this Matrix Useful For?

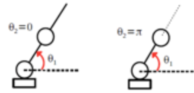
- To compute the necessary joint velocities that produce a desired end-effector velocity for tracking a desired Cartesian space trajectory

$\dot\theta(t)=J^{-1}(\theta)\dot P(t)$

- This only works if the inverse matrix exists

Manipulator Singularities

• Configurations where the Jacobian matrix loses rank (thus, it cannot be inverted)
– The determinant is zero |J(θ)| = 0
• Singularities of the planar 2-DOF manipulator:



Jacobian-based Method (Open-Loop)

Recall the equation for the open-loop Jacobian approach: $\theta(t_{k+1})=\theta(t_k)+J^{-1}(\theta(t_k))\dot P(t_k)\Delta t$

If at one time, the Jacobian is not computed correctly or has some small error, the joint space commanded will not be computed correctly at this instant.

How will it affect the rest of the Cartesian space trajectory tracking? Drifting!!!

Closed-Loop

• Drawback for the open-loop approach:
• Accumulate integration error, we call it "drifting"
• We need to come up a method to manage the error accumulation
• Error in Cartesian space: $e=P_d-P=P_d-h(\theta)$
• Time derivative of the error: $\dot e=\dot P_d-\dot P$
• Differential error in Cartesian space can also be written as: $\dot e=\dot x_d-J(\theta)\dot\theta$
• The goal is to find an expression for θ such that the error converges to zero over time

- Assuming that the Jacobian matrix is square and non-singular, the choice of $\dot\theta$ is

$$\dot\theta=J^{-1}(\theta)(\dot P_d+Ke)$$

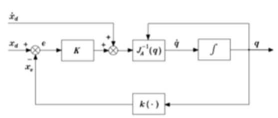
- Substitute this equation into $\dot e=\dot P_d-J(\theta)\dot\theta$, we have

$$\dot e+Ke=0$$

- If K is a positive definite matrix, the Cartesian space error is asymptotically stable. Why? What does it imply?

Since K is positive definite, the equation is a linear differential equation with solutions that exhibit exponential decay. This ensures that the error e diminishes over time. (ensures robust stability)

Jacobian-based Method (Closed-Loop)



Block diagram for the inverse kinematics algorithm using the differential method

Mobile Robot

• Electro-mechanical systems that are capable of locomotion(运动)
• To move around the environment, they can use multiple types of mechanisms

Applications

Logistics, Home cleaning, Autonomous cars, Museum tour guides, Building inspection, Aerial photo capturing

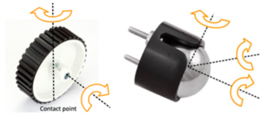
Classification of mobile robots Wheeled; Tracked; Legged

Wheeled Robots

- For locomotion in a **flat environment**
- Not suitable for rough terrain
- Easiest to design and build

Types of Wheels

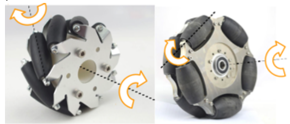
Standard wheels can rotate around the motorized axis and around the contact point



Ball wheels can rotate in 3 directions around its spherical axis (center of the wheel) ; Typically used as **passive wheels** for support (i.e. with no motor)

Mecanum (Swedish) Wheels

Rollers are attached to a larger wheel with a **45 degrees** configuration; Used for **omni-directional** platforms with 4 wheels



Omni Wheels

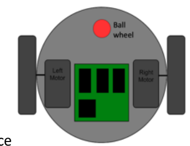
Rollers as also attached to the wheel, but in a perpendicular direction (90 degrees);Used for omni-directional platforms with 3 wheels

Wheels on a Robot

- How many wheels do we need to stabilize a robot? **Three**
- Four-wheel structure: Improves the system's stability; However, a flexible suspension might be needed (due to the redundant contact point)

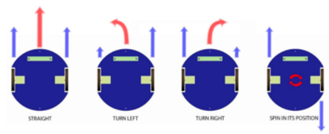
Differential Drive Robot

Two motorized "standard" wheels with an additional passive wheel (sometimes two) to keep balance



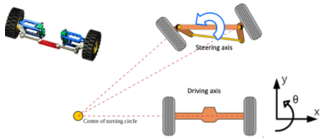
Motion of Differential Drive Robots

The resulting motion depends on the speed difference and the direction of the rotation



Steering-Drive Robot (Car)

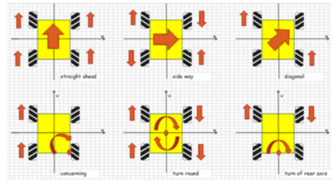
- **Two motors:** one controls the **speed**, the other the **steering angle**
- Motion is constrained by the center of rotation



Omnidirectional Robot(with Mecanum Wheels)

Omnidirectional means in all directions
Controls two positions (x, y) and one rotation (θ)

Four motorized mecanum wheels



Omnidirectional Robot

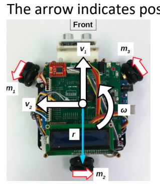
• Platform with **three motorized** Omni wheels
• Allows motion in all directions (x, y, θ)
• Simple to build and control

Robot's Configuration

Typically, the axes of the wheels are separated by **120 degrees** at a distance r from the center.

Velocity Analysis

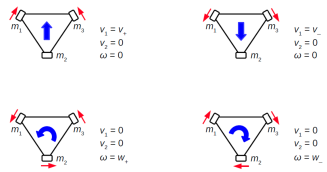
Define the front along one wheel axis (e.g. the sensor): m_k: velocity (control input) of the k-th motor; v1: robot's forward velocity; v2: robot's lateral velocity; ω: robot's angular velocity; r: distance from the robot's center to the wheels; The arrow indicates positive (+) direction



The **desired robot's velocities** (i.e., forward, lateral, angular) are transformed into the **required motors' velocities** as follows:

$$\begin{bmatrix}m_1\\m_2\\m_3\end{bmatrix}=\begin{bmatrix}-\sin(60) & \cos(60) & r\\0 & -1 & r\\ \sin(60) & \cos(60) & r\end{bmatrix}\begin{bmatrix}v_1\\v_2\\\omega\end{bmatrix}$$

Simple Motion Commands
(with zero lateral velocity v_2)



Simple Lateral Motions
(with zero frontal velocity v_1)

where:
✓ m_2 always moves in the **same direction** as v_2
✓ m_1 and m_3 move at **half the speed** of m_2

Wheel's Motion Kinematics

- Note that m_k represents the linear velocity that is generated by the k-th motor