# Physical Interfaces and Sensors

CHAPTER 3 – EMBEDDED ROBOTICS

CHAPTER 7 - ROBOT PROGRAMMING A PRACTICAL GUIDE

# Aims and Objectives

Aim(s):

- To introduce different types of sensors, their operating characteristics and their application areas

Objective(s):

- Discuss what happens when sensors fail
- Introduce the six basic questions for sensor selection
- Classify and categorise sensors based on their applications and purpose
- Identify sensors for each classification and category
- Discuss operating characteristics for sensors in each classification and category

# Introduction

Robot programming involves sensors which there are a lot off

You as a programmer must develop a clear understanding of the sensors you use:

- How does the sensor work
  - Analogue/Digital output
  - Do you control the sensor using IIC or SPI or serial communication?
  - Should it be polled or does it work with interrupts?
- How does the sensor fail?
  - Failing open or closed
  - Randomly giving data
  - Not giving any data

It is often a balance between the algorithm and the sensors

- The algorithm dictates the type of sensors but
- The quality sensors dictate the algorithm

# Selecting sensors

Be aware of the following:

- What does the sensor actually measure?
- Example: Sonar sensor
  - We use sonar sensors to measure distance
  - BUT a sonar sensor ACTUALLY measures time!!
  - We interpret the time as a distance between objects
  - But under non-optimal circumstances the distance reported is not correct
- Answering the question of what a sensor actually senses leads directly to questions of under what conditions it is appropriate to use the particular sensor
  - Can we expect false positives?
  - Can we expect false negatives?
  - Can the robot determine when the sensor should not be trusted?
  - Can we ensure graceful degradation when the sensor supplies misleading data?

# Selecting sensors

Find the right sensor for a particular application taking the following into account:

- Measurements techniques
- Size
- Weight
- Operating temperature
- Power consumption
- Price

Data transfer from the sensor to the CPU can be:

- CPU-initiated (polling)
  - Much more time consuming
- Sensor-initiated (interrupt)

# Analog vs Digital Sensors

## ANALOG

A number of sensors produce analog signals

An ADC is required to connect these sensors to a microcontroller

Examples:
- Microphone
- Analog IR distance sensor (GP2DXX)
- Analog compass
- Barometer sensor

## DIGITAL

Digital sensors are usually more complex than analog sensors

They are usually more accurate

The same sensor might be available in analog and digital form

The digital output can have different forms:
- Parallel
- Serial (RS232)
- Synchronous serial
  - Means the converted value is read bit by bit from the sensor

# Sensor Outputs

From an engineer's point of view:

- Classify sensors according to their output signals
- It is important for interfacing to an embedded system

| Sensor Output | Sample Application |
| --- | --- |
| Binary Signal (0 or 1) | Tactile sensor |
| Analog signal (e.g. 0..5V) | Inclinometer |
| Timing Signal (e.g. PWM) | Gyroscope |
| Serial link (RS232 or USB) | GPS module |
| Parallel link | Digital camera |

# Sensor Categories

From a robot’s point of view it is more important to distinguish:

- Local or on-board sensors (sensors mounted on the robot)
- Global sensors (sensors mounted outside the robot in its environment and transmitting sensor data back to the robot)

For mobile robot systems it is also important to distinguish:

- Internal or proprioceptive sensors (sensor monitoring the robot’s internal state)
- External sensors (sensors monitoring the environment)

A Further distinction is between:

- Passive sensors (sensor monitoring the environment without disturbing it: e.g. Camera, gyroscope)
- Active sensors (sensors that stimulate the environment for their measurement: e.g. Sonar, laser scanners and IR sensors)

| | Local | Global |
|---|---|---|
| **Internal** | **Passive**<br>Battery sensor<br>Chip-temperature sensor<br>Shaft encoders<br>Accelerometers<br>Gyroscope<br>Inclinometer<br>Compass<br><br>**Active** | **Passive**<br><br>**Active** |
| **External** | **Passive**<br>On-board camera<br><br>**Active**<br>Sonar sensor<br>IR distance sensor<br>Laser scanner | **Passive**<br>Overhead camera<br>Satellite GPS<br><br>**Active**<br>Sonar or other global positioning system |

# Sensor Purposes

Sensors can also be grouped according to their purposes:

- Collision Sensors
  - Bumpers / Binary Sensors, Stall Sensors, Stasis Sensing
- Avoidance Sensors
  - IR Proximity Sensors, IR Range Sensors, Sonar Sensors
- Homing Sensors
  - Photocells, Phototransistors and Photodiodes, Coded Beacons, Pyro-electric Sensors, Colour Blob Sensors, Magnetic Sensors
- Dead Reckoning and Navigation Sensors
  - Shaft Encoders, Inertial Sensors, Compasses, GPS

# Collision Sensors

The most pertinent force with which a mobile robot must concern itself is the force between the robot and the environment

The robot/environment force is non-zero when a collision occurs

The robot generates the force of collision through the torque provided by its wheels

Three possible outcomes after a collision:

- If the wheel/floor traction is high, the wheels will stall during a collision
- If the wheel/floor traction is poor, the collision with a heavy object will have the wheels slipping on the floor
- If the object is light the robot might push the object but might move slower and use more current

Dealing with these 3 outcomes complicates your program because the situation might be more complex than the output from 1 sensor alone.

# Binary Sensor

It is the simplest type of sensor

They only return a single bit of information, either 0 or 1

Example: Bumper sensor or tactile sensor on robot using a micro switch

Interfacing to a microcontroller is done using a digital input on the controller or on a latch

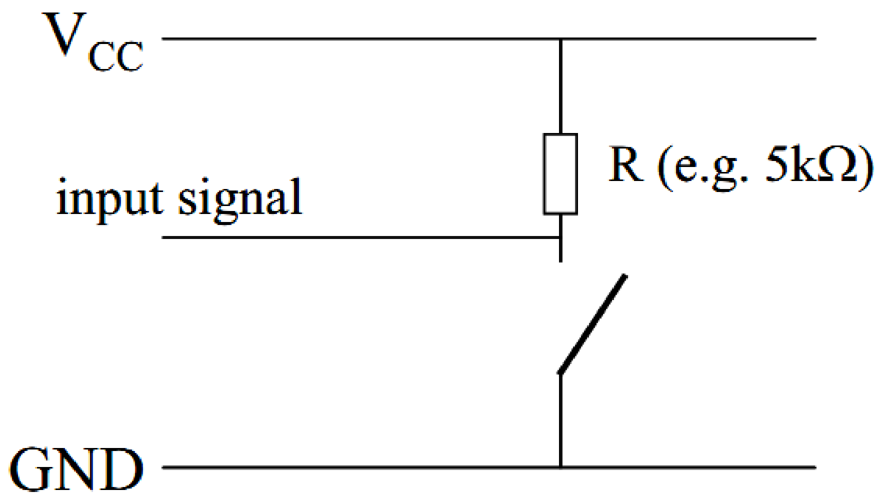



Figure 3.1 (Braunl, 2008. p51)

# Bumpers

An instrumented bumper provides the most direct method for detecting force between robot and environment.

Multiple bumper sensors can indicate if collision occurred on the left, front or right or even higher resolutions

A properly designed bumpers are reliable and gives an unambiguous answer to the question: Have I collided with an object?

Bumpers can sense only what it touches

The biggest problem with bumper design is poor or incomplete coverage (Fig 7.2)

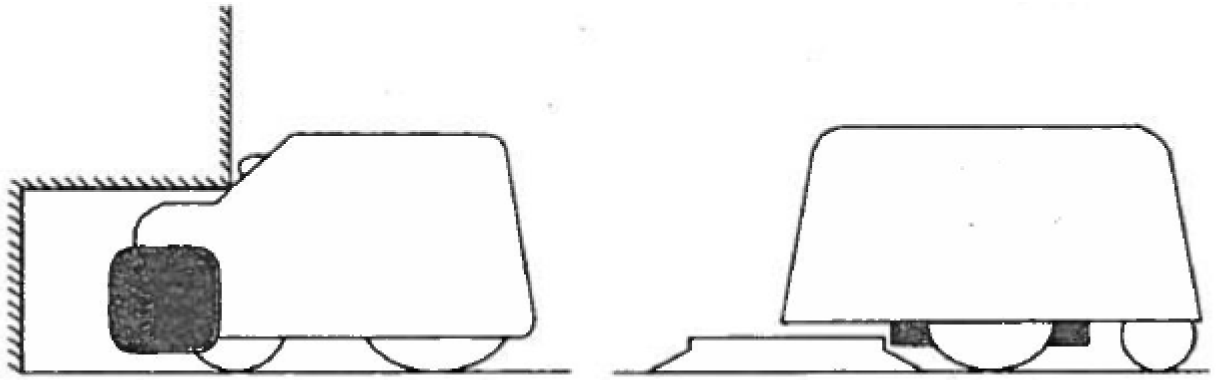


Figure 7.2 (Jones, 2004. p180)

# Bumpers Design Considerations

False Negatives due to incomplete bumper coverage (Robot collides but sensor does not trigger)

- Forward bumper but no rear bumper, risk of false negative when backing up
- Obstacles passing underneath bumper and snagging on chassis

False Positive due to bumper inertia (Triggered but should not have)

- Heavy bumpers and weak springs can lead to bumper oscillations
- Limit robot acceleration and deceleration
- Implement debouncing after rapid stops to eliminate bumper oscillations
- Weak springs allow for interaction with small forces, strong springs prevents false positives

A slippery floor might not allow enough traction to activate the bumper sensor

- Drive wheel stall detector
- Stasis sensing (detecting the absence of sensor value changes)
- Wheels with better traction

Do not fix bad bumper design with clever software, rather fix the bad designed bumper

# Stall Sensors

Determines if a collision occurred based on the measured current flowing through the motors

Small robots typically uses Permanent Magnet DC (PMDC) motors for drive wheels

PMDC characteristics (Fig 7.3)

- Maximum current flows when voltage is applied but the motor is not running
- Maximum current flows for an instance when the motor begins to turn from a dead stop
- Current flow increases as motor works harder

Maximum current flowing for a relatively long time might indicate a collision

Can give both false negative and false positives

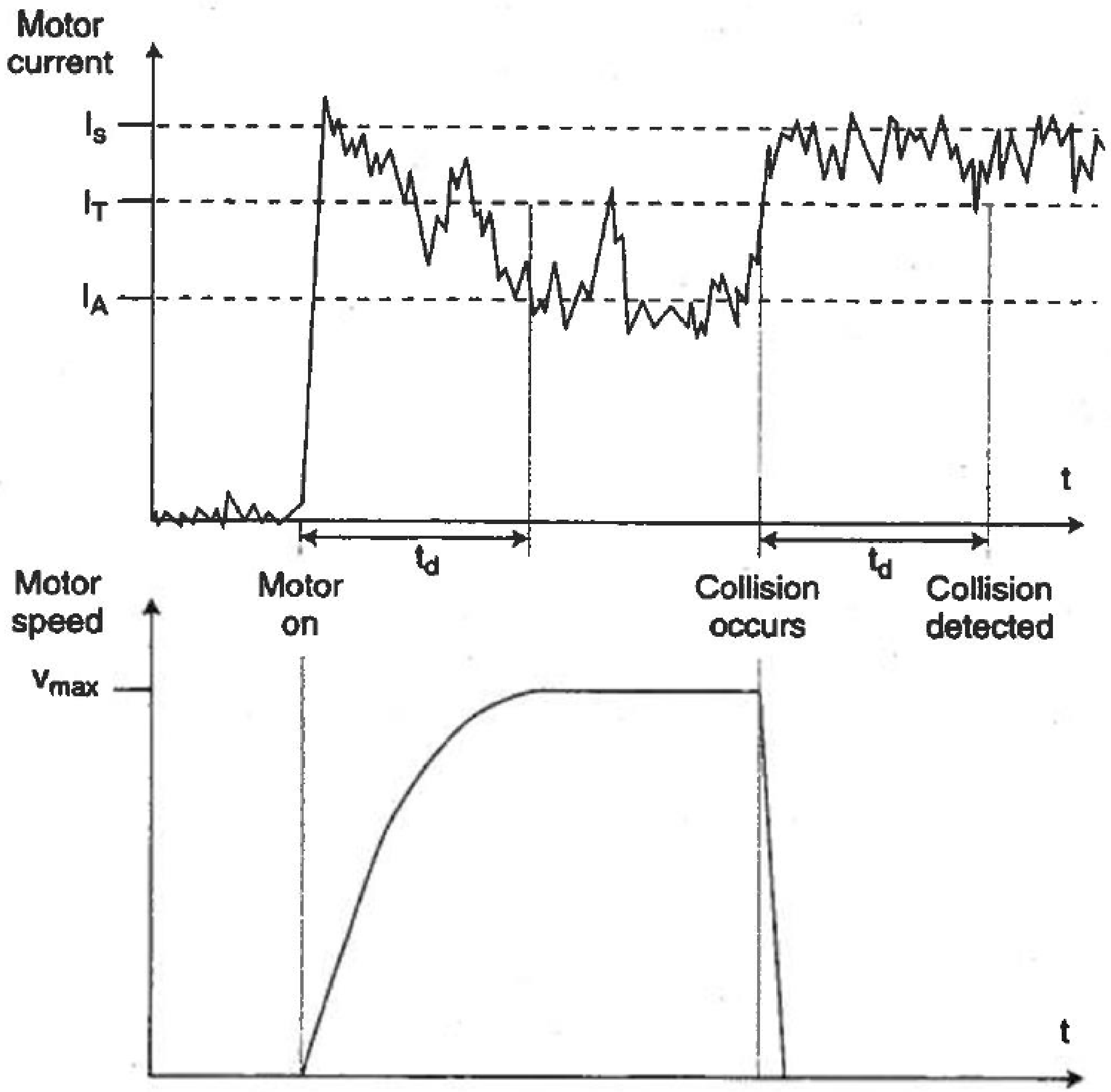



Figure 7.3 (Jones, 2004. p37)

# Stall Sensor Implementation

Select a threshold for safe operating current

- Terrain influences current usage
- Cheap PMDC noise is also a problem
- Poor robot traction can result in false negative

Above threshold current for an extended time equates to a collision

- Use a delay to determine if a collision occurred
- There is a trade off for the time to wait before a collision is called
  - A long delay will allow the motor to push against an object for an extended time
  - A short delay will often incorrectly interpret start-ups and temporary high load conditions as collision

Tune stall sensors conservatively to avoid false positives and rely on another system for occasional false negatives

# Stasis Sensing

A stasis sensor is used to detect whether a robot is moving by monitoring changes in sensor values

Can be virtual or physical

Virtual is easier

- Construct a behaviour that monitors one or more physical sensors to see whether their values change over time

Drive wheel shaft encoders are not useful since wheels can spin in low friction environments

# Avoidance Sensors

Relying on collision sensors only is not the best strategy

The better approach is to avoid hazards while they are some distance away

Light based and acoustic based sensors are commonly used for this purpose

# IR Proximity Sensors

These sensors are common and inexpensive

They consist of an IR emitter and detector

If an object is present in the intersecting fields from the emitter and detector, radiation reflected from the object will strike the detector

The absence of reflected radiation is interpreted as the absence of an object

To improve performance the emitter output is modulated with a carrier frequency of between 38kHz to 56kHz

These sensors can create false negatives when:

- Dark, shiny, too absorptive or too small cross section to reflect sufficient light
- Sunlight or other bright light sources can saturate the sensors

False positives are rare but can be because of IR noise from unexpected sources

- e.g. Fluorescent lamps

Multiple IR emitters and detectors can be used to reduce blind spots

# IR Range Sensors

These sensors return an analogue value relative to the distance measured

Does not make use of time-of-flight uses a triangulation array (Fig 7.6)

Also prone to errors (Disadvantages):

- Dark, shiny or too small objects
- Output not necessarily linear
  - use a lookup table (faster) or a mathematical procedure to convert the ADC value
- Objects near the limits of the sensor (too far or too close)
  - Below 6cm (sensor dependant) distances are ambiguous
  - Fix it mechanically

Require a significant time to operate

- GP2D12 produces a signal every 40ms
- A fast robot can run into stuff because sensor is ‘slow’

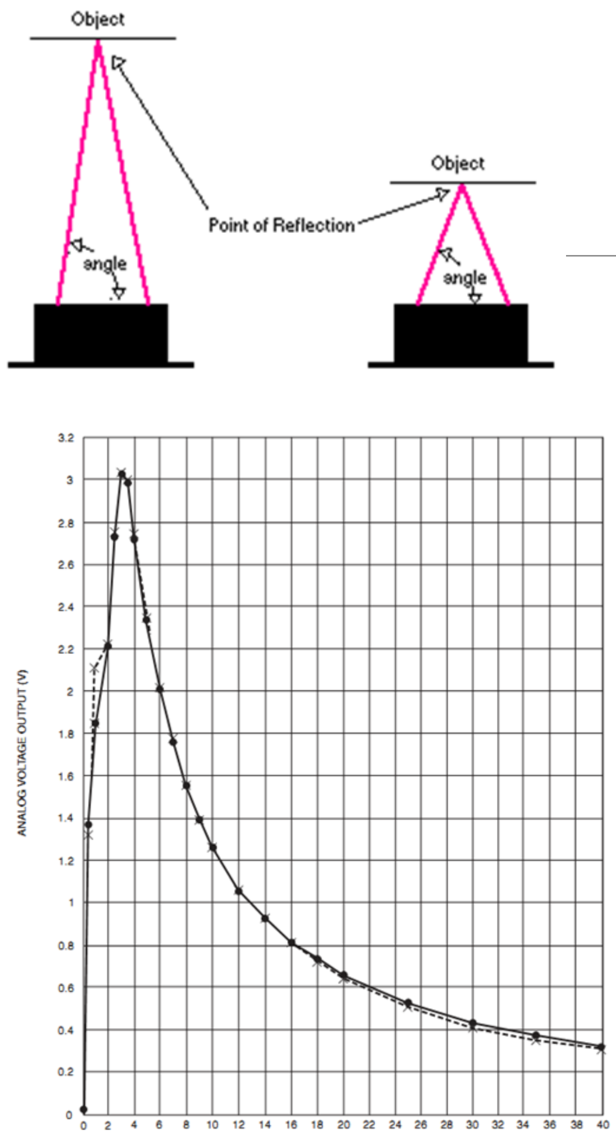



Figure 3.7: Sharp PSD sensor and sensor diagram (Braunl, 2008. P57)

# Sonar Sensors

Effective at sensing large objects perpendicular to the beam at a distance that is not too large and not too small

Has a relatively narrow acoustical 'cone' (15°)

Need about 24 sensors to cover the circumference of a round robot

Detection distance varies between sensors

Operating Principle

- Emits short ultrasonic (50kHz to 250kHz) bursts
- The time is measured from signal emission to reception
- The measured time-of-flight is proportional to twice the distance of the nearest obstacle in the sensor cone
- If no signal is received within a certain time, then no obstacle is detected within the corresponding distance
- Measurements are repeated at about 20Hz

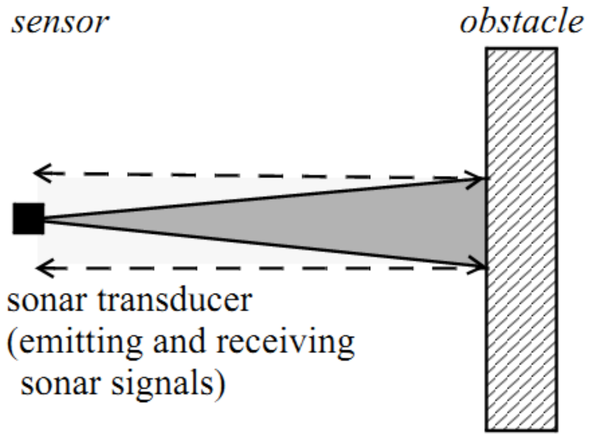



Figure 3.5: Sonar sensor (Braunl, 2008. P55)

# Sonar Sensor Errors

Disadvantages due to **Reflections**

- Beam strikes a smooth wall at a shallow angle and the return path is longer making an obstacle seem further away than it is or there is no return signal (c)

Disadvantages due to **Interference**

- When several sensors are operated at the same time the acoustical signals get mixed up resulting in incorrectly assuming an obstacle is closer than it is

False negatives

- The finite beam width missing obstacles just outside the acoustical cone
- Small objects return an echo that has an insufficient amplitude (b)
- Beam strikes a smooth wall at a shallow angle (c)

False positive

- Interference from other sources

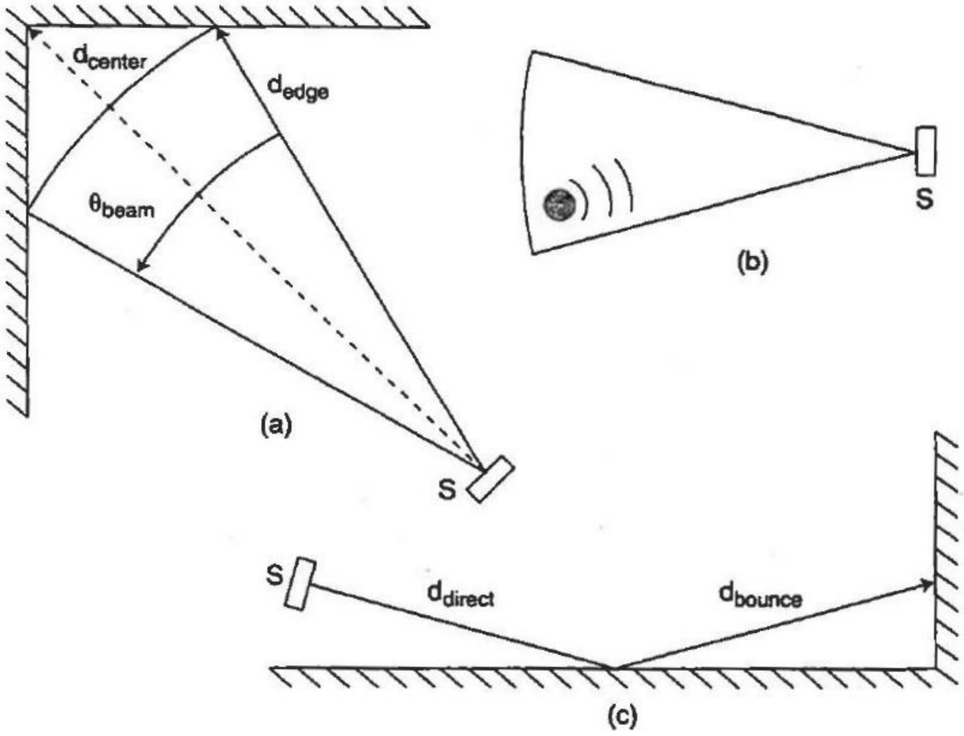



Figure 7.7 (Jones, 2004. p190)

# Range-Sensor Considerations

Ask the following question:

- What are max and min distances your range sensors should be able to measure?

If the sensor is only used for collision avoidance then the max distance is the robot's stopping distance plus safety margin

- Stopping distance = distance the robot moved after it decided it needs to stop
- This is a matter physics and sensor/system latency

If the purpose of the ranging sensor is to prevent collision it does not need to know about objects that are 3 meters away!

Do not purchase information you are not using.

# Homing Sensors

A homing sensor provides the robot with a means to reach some destination

The destination is typically marked in a unique way e.g. beacon, light, color, QR codes, barcodes, etc

# Photocells, Phototransistors and Photodiodes

- The simplest method of marking a location is to attach a light to it
- The simplest light sensor measures the intensity of light striking the element
- The output is an analogue voltage whose value is a function of light intensity
- The light intensity reaching the sensor is related to:
  - the intensity of the light source,
  - distance from source and
  - angle to the source, perpendicular is the best
- Simple light sensors are amongst the most reliable of sensors
- It is the interpretation of data that is problematic
  - Is the robot homing in on the lamp above the charging station or another light source e.g. lamp?
- All photocells are not the same and even a mismatched pair can home in on the target (Fig 7.9)

# Coded Beacons

A coded beacon can be made unique

Beacons typically consists of an omnidirectional IR source modulated in some particular way (Fig 7.10)

One or more receivers look only for the modulation pattern

This method is one of the most reliable since stray IR light will not fool the system

Coded beacons can fail when:

- The area is saturated by noisy IR emissions from other sources, such that the robot cannot identify the beacon amid the noise
- Obstacles in the environment prevents the robot from seeing the beacon (Light is line-of-sight)
- The sun or another source of bright light saturates the on-board sensor making it impossible to sense any signals

# Pyro electric Sensors

Converts radiant heat into electric signals

These sensors are sensitive to longer wavelength far-IR radiation emitted by warm-blooded animals

Can be used to detect a nearby person and is found in motion sensing burglar alarms

Affordable sensors detect differences in radiant heat rather than absolute levels of heat

Pyro sensors can fail when:

- the background temperature matches the object to detects’ temperature
- There are other hot sources in it’s field of view. Will home in on a fireplace rather than a human/animal

# Colour Blob Sensors

More and more inexpensive cameras make them suitable for colour tracking homing beacons on robots

Choose a colour and train the camera/algorithm for the colour

The camera then continuously output the coordinates of the centroid of the matching colour in its field of view

Caveats of vision

- Cameras have a finite frame rate – might lose the object
- Changes in illumination might influence the camera’s ability to detect colour and might lose the tracked object
- The colour of the tracking object must be unique

# Magnetic Sensors

Also called Magnetic detectors or Hall-effect sensors

Limited by range because magnetic field intensity reduces as the inverse cube of distance

Mostly used for docking, rather than homing

# Dead Reckoning and Navigation Sensors

Dead reckoning and other navigation sensors are used to drive the robot to some location without an explicit marker on the place where the robot is going

# Shaft Encoders

The drive wheel shaft encoder is the most commonly used position aid sensor

Shaft encoders measure the rotation of a motor or wheel very reliable and the signal is used to determine:

- Which direction the wheel turns: Forward or Reverse
- How fast the wheel rotates
- How far the wheel turned

Unfortunately the robots position using shaft encoders contains an unknown amount of error

The error component never decreases, the uncertainty of our knowledge of where the robot is, increases as the robot moves

Worst case scenario is when the wheel slips, then there is no relationship between wheel motion and robot motion

Shaft encoders are standard on robots for determining their position and orientation

**Study and IMPLEMENT: 'Interfacing Microcontrollers with Incremental Shaft Encoders' – downloadable from the CMS**

# Optical Shaft Encoders

Standard optical encoders use a sector disk with black and white segments together with an LED and a photo-diode

An IR emitter/detector pair is connected in order to detect the light and dark pulses as the wheel turns

- 16 black and 16 white segments will give us 16 pulses (encoder ticks) during 1 revolution

To determine direction 2x IR emitter/detector pairs are used, 90° out of phase

Wheel direction is determined by whether signal B leads or lags A

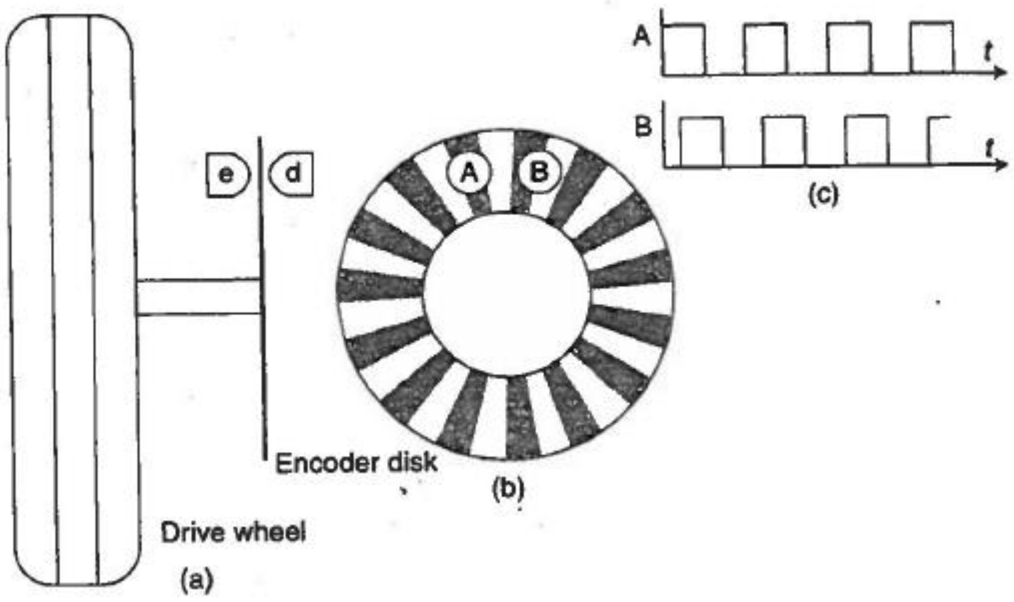



Figure 7.11 (Jones, 2004. p198)

# Magnetic Shaft Encoders

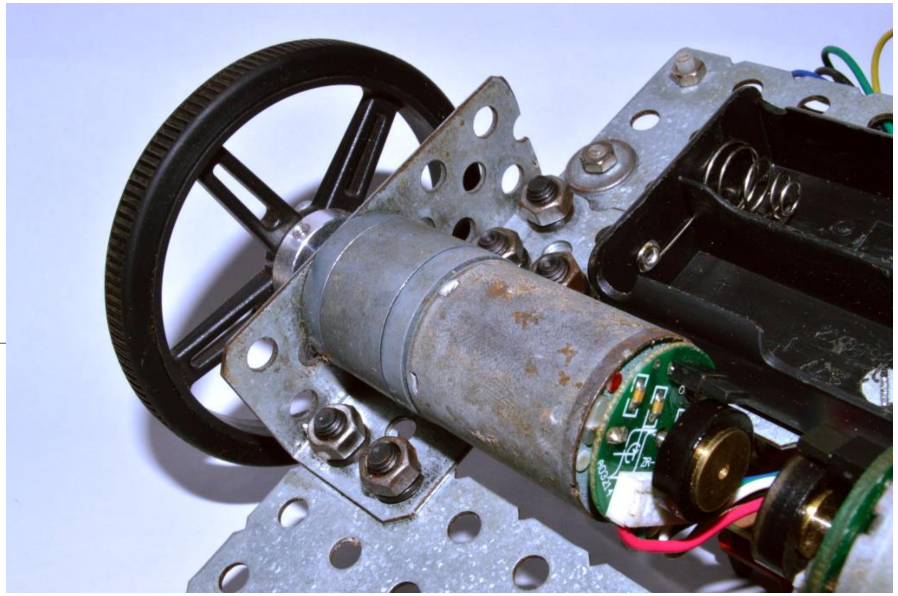
Shaft encoders can be built using magnetic or optical encoders.

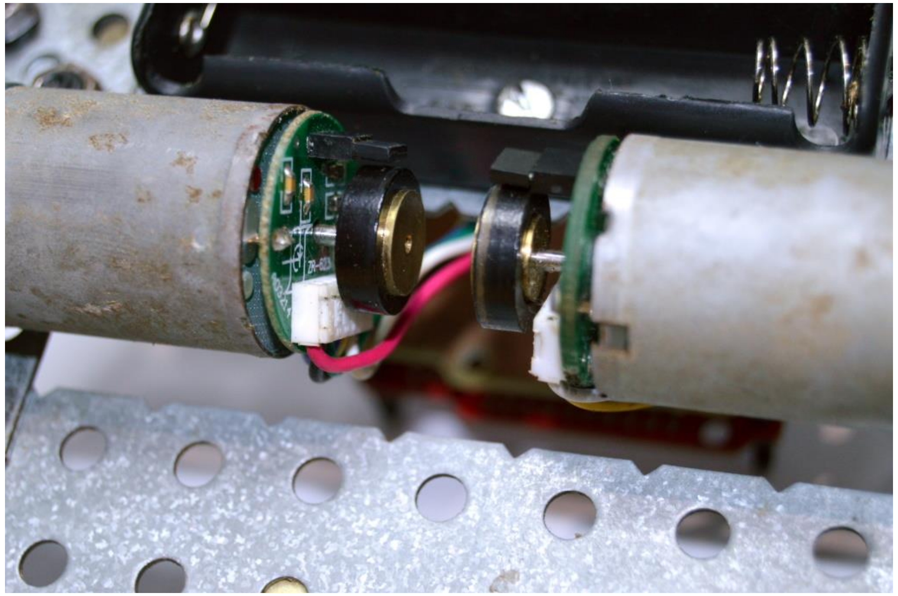
Magnetic encoders use a Hall-effect sensor and rotating disk on the motor shaft with one or more magnets mounted on the disc

Every pass of a magnet past the Hall-effect sensor results in a “tick” on the encoder line

Encoders are usually mounted on the motor shaft BEFORE the gearbox

- If we have an encoder which detects 16 pulses per revolution and a 100:1 gearbox we have 1600 ticks per wheel revolution





Source: OWN

# Inertial Sensors

Also known as accelerometers and gyroscopes

With expensive sensors, data can be integrated over time to determine position. Cheap sensors have to big an error

Cheap sensors can be used to determine the robot's attitude in relation to gravity

Cheap gyro's can faithfully report rapid changes in a robot's rotation rate for use in balancing robots

Always Remember: It is impossible for an inertial sensor to differentiate between forces caused by gravity and forces caused by changes in robot velocity due to the action of the robot motors

# Compasses

Most inexpensive electronic compasses are effective to determine the local vector toward the North magnetic pole

Problems with compass sensors:

- Interference from robot motors: Attach compass to mast
- Ferrous metals in the environment influences magnetic fields

Combining compass and shaft encoder data can generate information with a greater reliability than one sensor alone

- Use the compass together with the shaft encoders to determine orientation

Magnetic compasses tend to be globally accurate except in the presence of ferrous metals

Errors are normally very large, as much as 180°

Digital and analogue versions available

Usage: Use compass to compare and fix errors from shaft encoders when doing dead reckoning navigation

# Compass Example

Combine the heading sources to calculate the best possible estimate for $\theta_{rec}$, the heading we will use for dead reckoning computations

The two heading source we have is:

- $\theta_{encoders}$ = $(L_{enc\ value} - R_{enc\ value}).\ constant$
  - This tracks the rotation of the robot without any particular relationship with north
- $\theta_{compass}$ : The magnetic heading measured by an electronic compass

In order to calculate the robots true heading, we need to add a correction to $\theta_{rec}$ so that $\theta_{rec} = \theta_{encoders} + \theta_{correction}$

Where $\theta_{correction}$ is the difference between the true heading and the heading given by the encoders alone: $\theta_{correction} = \theta_{compass} - \theta_{encoders}$

By periodically updating $\theta_{correction}$ using the Running Average equation we can mitigate compass errors due to ferrous metals in the environment

Therefore $\theta_{rec} = \theta_{encoders} + \overline{\theta_{correction}}$ where $\overline{\theta_{correction}}$ is the average of the correction every time the robot has moved a certain distance, *d*

# Summary

This section introduced:

- Sensor classification using categories and usages
- Operating characteristics of sensors
- How sensors work and how they can fail

# Exercises

1. Consider a stasis sensor based on bumper activity. What factors determine how long the robot should wait after its most recent collision before it decides that it may be stuck?
2. Using many IR emitters and detectors mounted on the front of your robot, you can eliminate most (but not all) blind spots. Where, in relation to the robot, are the blind spots that cannot be eliminated in any practical way? What sorts of objects are most likely to penetrate undetected into the robot's blind spots?

# End of chapter

# References

Braunl, T., 2008. *Embedded Robotics. Mobile Robot Design And Applications With Embedded Systems*. 3rd ed. springer.

Jones, J., 2004. *Robot Programming: A Practical Guide To Behaviour-Based Robotics*. RR Donnelly.