Introduction to Robotics

Felipe P. Vista IV

Class Admin Matters

Grading

➢ Attendance 5%

Name (Original Name)	User Email	Join Time	Leave Time	Duration (Minutes)
[redacted]		4/12/2021 9:12	4/12/2021 10:14	62
		4/12/2021 9:12	4/12/2021 9:14	3
		4/12/2021 9:12	4/12/2021 9:14	3
		4/12/2021 9:12	4/12/2021 9:14	3
		4/12/2021 9:12	4/12/2021 9:14	3
		4/12/2021 9:12	4/12/2021 9:14	3
		4/12/2021 9:13	4/12/2021 9:13	1
		4/12/2021 9:13	4/12/2021 9:14	2
		4/12/2021 9:14	4/12/2021 9:14	1
		4/12/2021 9:14	4/12/2021 9:14	1
		4/12/2021 9:14	4/12/2021 10:14	60

Bad ZOOM User Name (Absent)

- Iphone → Not your name
- SiAko 202100001 → Wrong order
- SiAko → Name only
- 202100001 → ID Num only

ZOOM User Name (Present)

- University ID Num_Name
- 202100001 SiAko → GOOD (Present)

Name (Original Name)	User Email	Total Duration (Minutes)
[redacted]		62
		63
		62
		62
		63
		62
		63

Student Responsibilities

- Download/Install ZOOM app for online lecture
  - *Zoom profile must be your **OASIS ID+name** similar to OASIS*
  - *Ex.: **202061234 YourName***
- Regularly login and check on-line learning system for updates, notifications
  - *https://ieilmsold.jbnu.ac.kr*
  - *Presentations & lecture videos will be uploaded after class*
- Regularly check Kakao Group Chat for class
  - *Everybody must have a Kakao talk account*
  - *Search & add account "**botjok**", introduce yourself and name of class ("Robotics"), then you will be added to the group chat*

Intro To Robotics

SENSORS

Sensors

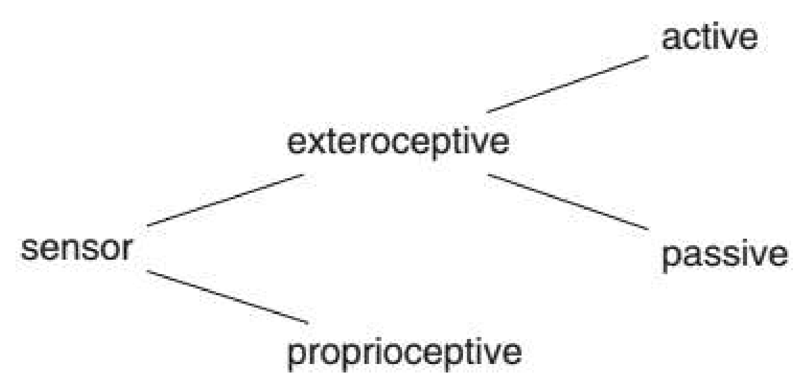
- Classification of sensors
- Distance sensors
- Cameras
- Other sensors
- Range, resolution, precision
- Nonlinearity

Sensors

- Make robot move specific distance at specific direction
  - *Even using same motor(s) & just controlling them is very hard*
  - *(Small differences actual power output) + (wheel & motor characteristics) + (surface unevenness) + (friction) → affect total dist & dir travelled*
- Ex: Robot move to wall 1 m away & stop 20 cm in front of it
  - *The robot must sense the wall and stop when it detects it is 20 cm away*
- Sensor
  - *Measure some aspect of the environment*
  - *Distance (cheap: infrared, ultrasound; expensive but accurate: lasers);*
  - *Images/ Videos for complex operations*
- Computer in robot
  - *Control actions of the robot based on sensor data*

Classification of Sensors

- General classification
  - ***Proprioceptive***
    - Internal to robot itself. Ex: speedometer
  - ***Exteroceptive***
    - External to the robot. Ex: distance



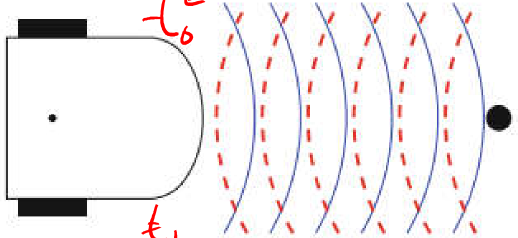
- Exteroceptive sub-class
  - ***Active***
    - Affects environment by usually emitting energy. Ex: sonar
  - ***Passive***
    - Does not affect the environment. Ex: camera record light reflected
- Robots usually use exteroceptives
  - *correct error due to proprioceptives; account for environmental changes*

Sensors

- Classification of sensors
- Distance sensors
- Cameras
- Other sensors
- Range, resolution, precision
- Nonlinearity

Distance Sensors

- Robots usually use it for **measuring** its distance from an object
  - Active, transmit a signal then receive any reflected signals
- One way is measuring time send→ receive bounced signal (t)

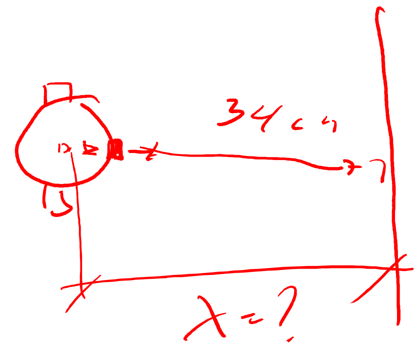


$$s = \frac{1}{2} vt$$

Where: ***s*** is distance, ***v*** is velocity of signal, ***t*** is time elapsed send→receive, **½** signal to travel (transmit →object→receiver)

- Some low-cost sensors measure intensity of reflected signal (i)
  - *Intensity of signal decrease with distance*
  - *Disadvantage*
    - factors affect intensity of returning signal (ex: object reflectivity)

Ultrasound Distance Sensors



- Sound whose frequency is **above** 20KHertz
  - *Higher than highest freq detectable by human ear*
- Sound **better** than vision
  - *At night and in water*
  - *Bats (navigating at night)*
  - *Ships/submarines (detect objects since sound travel better in water than air)*
    - We can try by going to sea, then compare how far we see to how far we can hear
- Ex: Speed of sound in air is **340 m/s** → (34 cm) = ½ (3.4×10³ cm/s) t
  - *If object is **34 cm** from robot, how long will it take for sound to travel?*

34

$$s=\frac{1}{2}vt \rightarrow 0.34\mathrm{m}=\frac{1}{2}(340\,\mathrm{m/s})(t) \rightarrow 2\mathrm{ms}$$

$$\frac{2(0.34)\,\mathrm{m}}{340\ \mathrm{m/s}} = 2\,\mathrm{ms}$$

**An electronic circuit can easily measure time in milliseconds.*

Ultrasound Distance Sensors

Advantages

- Not sensitive to:
  - *Change in object color or light reflectivity; Light intensity of environment*
- Relatively cheap & work outdoors (advantage)
  - *Ex: short distances such as aid to parking*

Disadvantages

- Sensitive to:
  - *Texture: fabric absorb some sound; wood/metal reflect almost all sounds*
- Measurement is relatively slow
  - *Speed of sound way lower than speed of light*
- Cannot be focused to measure distance of a specific object

Infrared Proximity Sensors

- Wavelength of IR light is longer than **red light**
  - *w/c is longest wavelength our eyes can see*
  - *Invisible to naked eye, remote control for electronic devices (TV, AirCon...)*
- Proximity sensors
  - *Detect presence of object by measuring intensity of reflected light*
  - *Light intensity decrease with square of the distance from source*

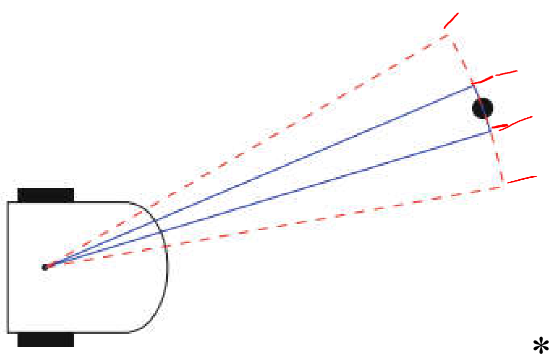
$$intensity \propto \frac{1}{distance^2}$$

* As distance from the light source increases, the intensity is equal to a value multiplied by **1/d²**

  - *Not very accurate, object reflectivity affects reflected intensity*
    - Black object reflects less light compared to white object at same distance
    - Cannot distinguish bet black object & white object farther away
  - *Which is why it is called "proximity sensors"*

Optical Distance Sensors

- Recall: distance computed through elapsed time trans → receipt
  - *Can be any type of light (ordinary or laser)*
  - *Laser* is **coherent**;*
    - Lasers for measuring distance usually use IR light, visible lights also
  - *Advantage of lasers:*
    - More powerful, can detect & measure long distances
    - Highly focused beam allow highly accurate measurement of angle to object

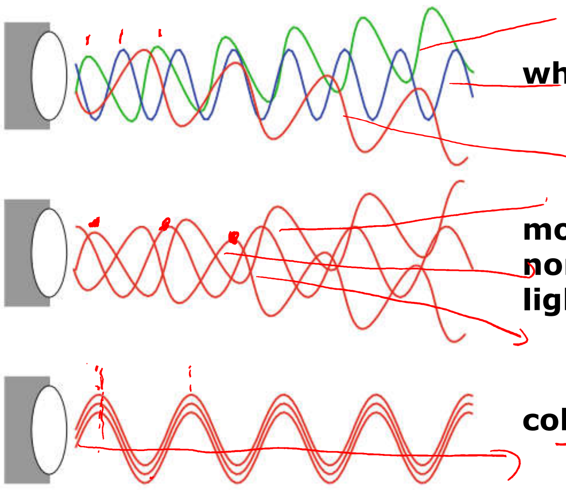


Beam width of laser light (*solid*) and non-coherent light (*dashed*)

* **l**ight **a**mplification by **s**timulated **e**mission of **r**adiation

Optical Distance Sensors

- White (sun, light bulb)
  - *Many different colors (freqs)*
  - *Different times (phases)*
  - *Different directions*
- Monochromatic (LEDs)
  - *Of a single color*
  - *Non-coherent*
    - Different phases & different directions
- Coherent (laser)
  - *All waves of same freq & same phase**
  - *All energy concentrated in one beam*



* Same phase – each wave start at the same time

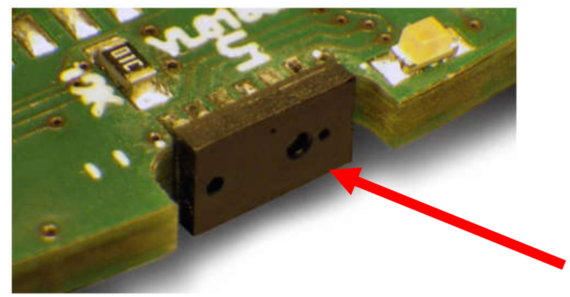
Optical Distance Sensors

- **Ex**: Pulse of light transmitted from robot, reflects on an object & then received by sensor on a robot. How long will it take?
  - *Speed of light in air:* ***3x10⁸ m/s (3x10¹⁰ cm/s)***
  - *Distance of object:* ***30 cm (0.3 m)***

$s = \frac{1}{2}vt \rightarrow t = \frac{2s}{v}$

$$\frac{(2)(30)}{3\times10^{10}} = \frac{2}{10^{9}} = 2\times10^{-9} = 0.002ms$$

A very short time period but can be electronically measured.

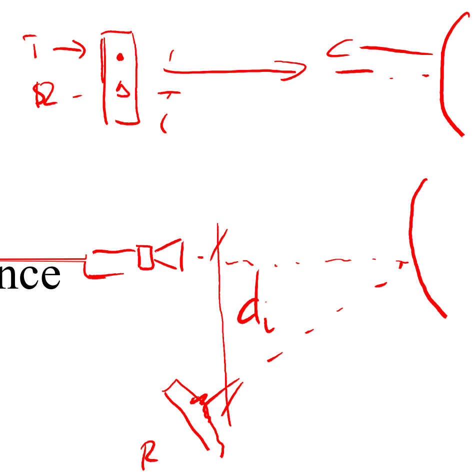


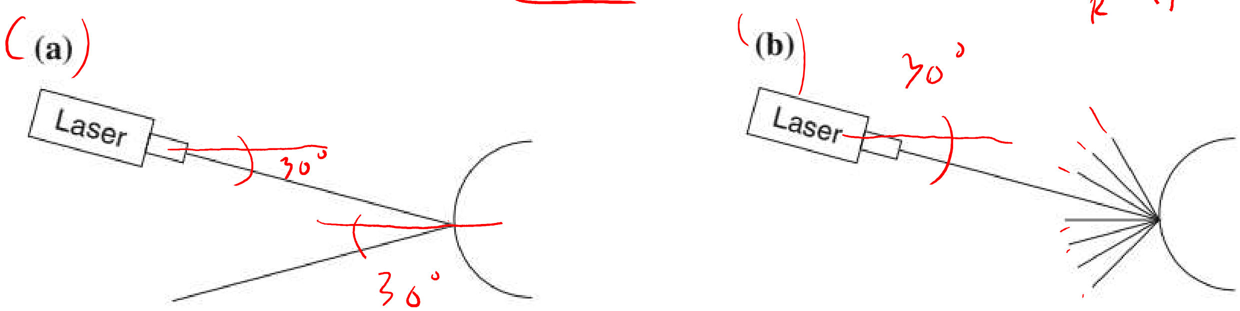
Time of flight sensor on a 1.6mm thick circuit board (sensor in black)

- Triangulation
  - *2nd principle of distance measurement via light beam*
  - *Transmitter and receiver are at different locations*

Triangulating Sensors

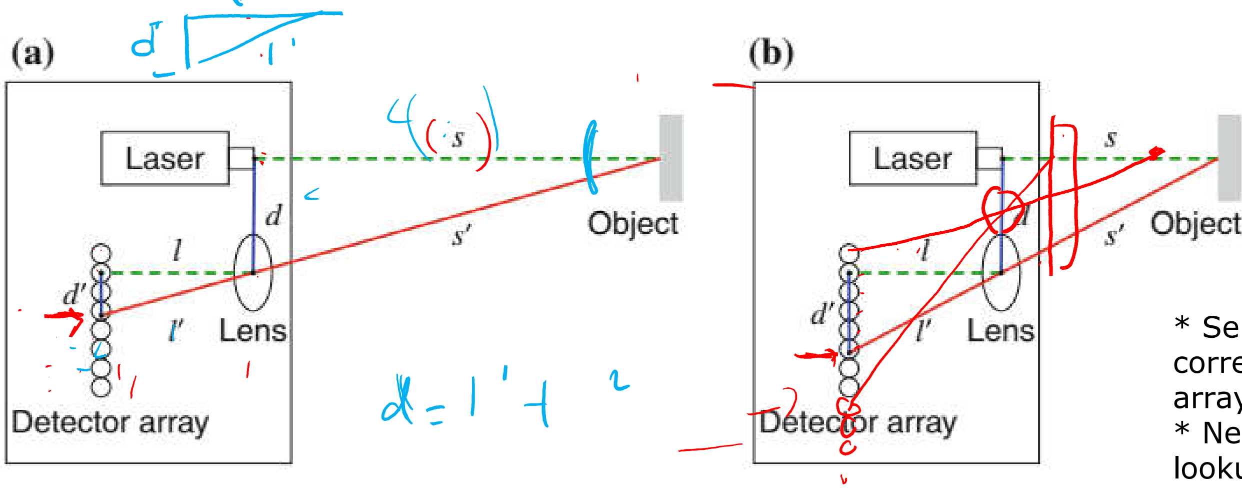
- Light reflected depends on object it hits
- Narrow light beam bounce off shiny surface in narrow beam
  - *Like laser (coherent) off a mirror*
- Reflections
  - *a) Specular reflection*
    - Angle of reflection angle same as angle of incidence
  - *b) Diffuse*
    - For rough surface, reflection scattered in all directions





Triangulating Sensors

- Simplified triangulating sensor on (a) far & (b) near objects
  - *Lens: **d** distance from laser, sensor array: **l** distance behind the lens*
  - *Assuming **diffuse**, some light collected by lens and focused onto sensors*
  - ***d** inversely proportional to **s***
  - *Triangles $\Delta ll'd'$ and $\Delta ss'd$ are similar, therefore* $\frac{s}{d} = \frac{l}{d'}$
  - ***l** & **d** are fixed, compute **s** by measuring **d'***



* Sensor calibrated by measuring ***s*** corresponding to each sensor in the array & stored in table in comp.
* Next time, easily get ***s*** by table lookup.

Triangulating Sensors

- Design parameters affecting performance
  - *Power of laser*
  - *Optical characteristics of lens*
  - *Number of sensors in array*
  - *Sensitivity of the sensors used*

- Trade-off
  - *Performance vs Cost (usual)*
  - *Range & minimum measurable distance of object (**main** trade-off)*
  - *To get very small **s**:*
    - detector array size **d'** becomes very large→ put limit on minimal distance
  - *To get shorter minimum distance*
    - Increase dist bet laser emitter & detector array → will reduce range

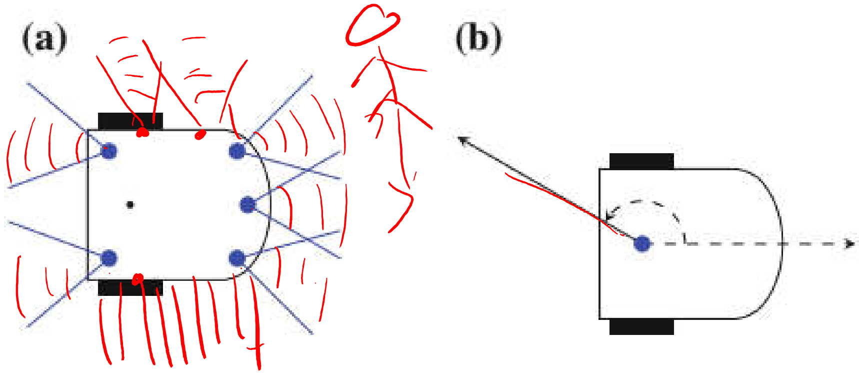
Laser Scanners

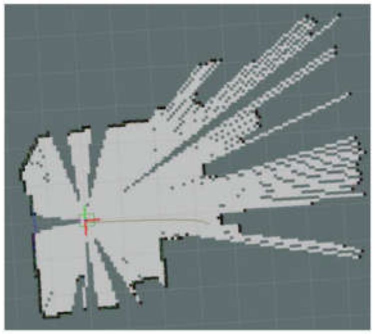
a) With ultrasound or proximity sensors

- *Small number placed around robot to detect objects around its area*
  - Robot approach/ avoid object but angle to object not measured accurately
- *With laser sensor*
  - Beam width so small, large number needed to detect objects at any angle

b) Laser scanner

- *Better design: Single laser sensor mounted on rotating shaft*
  - Full 360° can generate profile of objects in env





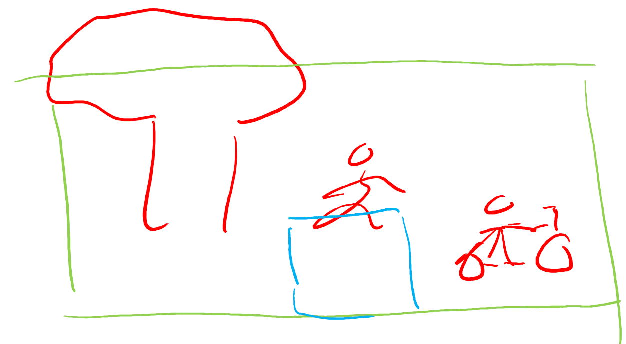
Map of environment from a laser scanner

Sensors

- Classification of sensors
- Distance sensors
- Cameras
- Other sensors
- Range, resolution, precision
- Nonlinearity

Cameras

- Digital cameras in robotics
  - *Widely used, more* ***detailed*** *info than just distance & angle to object(s)*
  - *Use* ***charge-coupled device***
    - Sense light waves
    - Return array of **picture elements** (or **pixels**)



- Characterized by
  - ***Number*** *of pixels captured/frame*
  - ***Content*** *of the pixels*
- Ex: A small camera in an educational robot
  - *Capture 192 rows of 256 pixels each* → *for total of* ***49,152*** *pixels*
    - Which is a very small picture
  - *Smartphone cameras record images in* ***millions*** *of pixels*

Cameras

- **Return** values for each pixel
  - *As **black and white** (1-bit per pixel)*
  - *Shades of **gray** (8-bits per pixel)*
  - ***Full color** red-green-blue (RGB) (3 x 8 = 24 bits per pixel)*
- Ex: A small 256 x 192 camera
  - *about 50 kbyte for single grayscale image*
  - *150 kbyte for a color image*
- Mobile robot (such as self-driving car)
  - *Several images/second*
    - Movies & TV approximately 24 images/sec
  - *Require large memory for storing & analyzing images*

Cameras

- Field of view: *Important characteristic*
  - *What portion of sphere surrounding camera is captured in image?*
  - *Lens with narrow FOV*
    - *Small area w/ high resolution & little distortion*
  - *Lens with wide FOV*
    - *Large area w/ lower resolution & more distortion*
- Omnidirectional camera
  - *Most extreme case of distortion*
  - *Capture almost entire sphere surrounding it*
- Cameras w/ wide FOV
  - *Used by robots to analyze environment*
    - *For navigation, detect objects*
    - *Interact w/ people or other robots using visual properties (like color)*

Camera's position

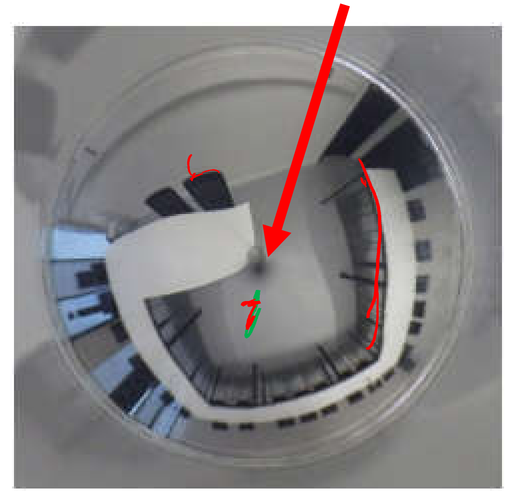


Image capture of Omnidirectional camera w/ 360 FOV

Cameras

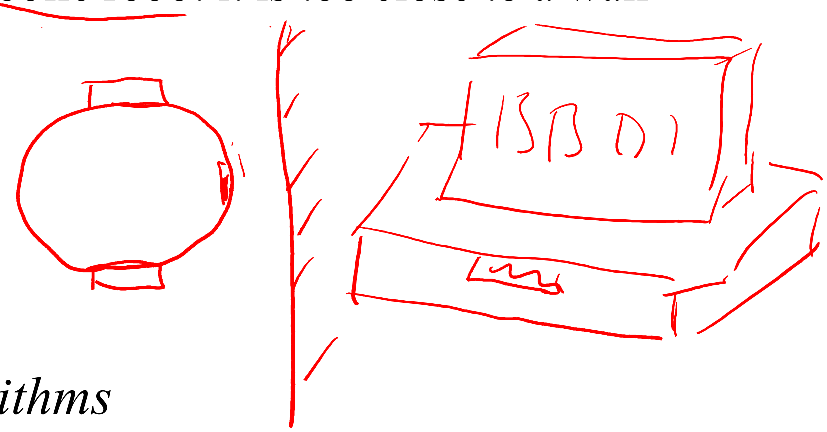
- Robotics is not interested in array of “raw” pixels
  - *But in identifying objects in the image*
- Human eye & brain perform recognition tasks instantly
  - *Driving*
    - Identify other vehicles, pedestrians, traffic lights, obstacles
    - Then take appropriate actions
- Image processing:
  - *Sophisticated algorithms*
  - *Significant processing power*
- Robots with cameras
  - *More complex & expensive than educational robots*

Sensors

- Classification of sensors
- Distance sensors
- Cameras
- Other sensors
- Range, resolution, precision
- Nonlinearity

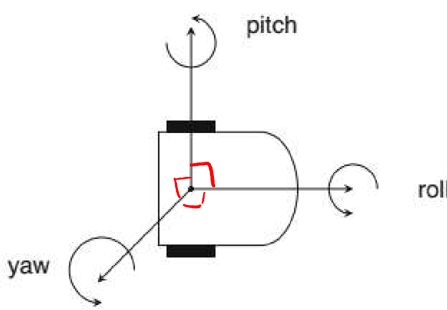
Other Sensors

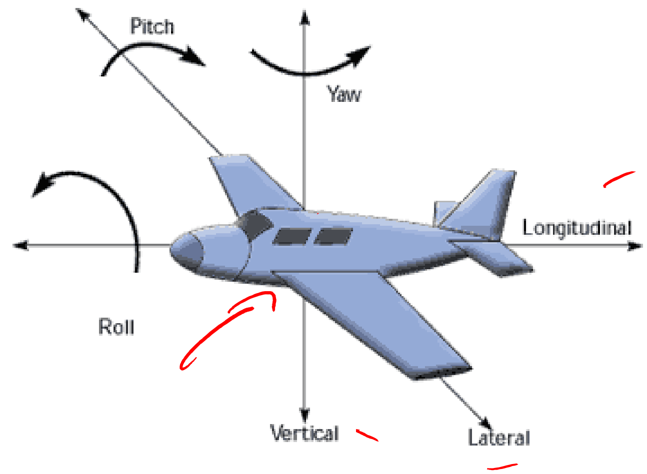
- Touch sensor
  - *Simplified distance sensor*
  - *Measure only two values: distance to object is "0" or " > 0"*
  - *Usually as safety method*
    - Used in bottom of small room heaters, will only run if heater detect/touch floor; heater will automatically turn off if it topples/fall over
    - Will apply emergency brake in a mobile robot it is too close to a wall
- Buttons & switches
  - *Direct user interaction with robot*
- Microphone
  - *Robot can sense sound*
  - *interpret voice commands via algorithms*

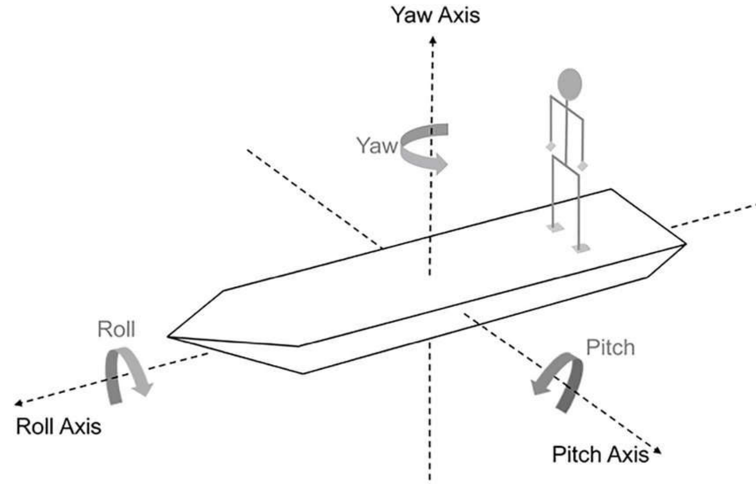


Other Sensors

- Accelerometer
  - *Measure acceleration* x/s^2
  - *Primarily the direction of gravitational force*
    - cause acceleration of 9.8 m/s^2 to earth's center
- Three-axis accelerometer
  - *Three perpendicular to each other*
  - *Allow measurement of robot's attitude:*
    - Pitch
    - Yaw
    - Roll







Sensors

- Classification of sensors
- Distance sensors
- Cameras
- Other sensors
- Range, resolution, precision
- Nonlinearity

Range

Extent of the set of values that sensor can measure

Example:

- Infrared sensor:
  - *Distance of 1cm ~ 30cm*
- Laser
  - *Larger range since focused power into narrow beam*
- Distance sensor
  - *Moving robot in a building: about 10m*
  - *Self-driving car: up to 100m*

Resolution

Smallest measurable change in sensor value

Example:

- Distance sensor
  - *One sensor returns distances in centimeters (1 cm , 2 cm, 3 cm, 4 cm…)*
  - *Better sensor is in hundredths of centimeters (4.00 cm, 4.01 cm, 4.02 cm…)*
- Self-driving car
  - *Resolution of centimeters is enough*
    - We will not park 1 cm away from another car, or even 0.1 cm away
- Surgical Robot
  - *Requires much higher resolution*
  - *Even a millimeter is critical when performing surgery*

Precision

Consistency of the measurement

- Same quantity measured repeatedly gives the same result?
  - *Very important*
  - *Inconsistent measurements* → *inconsistent decisions*

→ 36.5 / .4 ?

— 34.5 / 36.0 / 37

- Self-driving car
  - *Measure distance to nearest 10cm*
  - *But successive measurements*
    - Wide range i.e. 250 cm, 280 cm, 210 cm
  - *In maintaining fixed distance away from preceding vehicle*
    - Speed-up or slow-down for no good reason aside to maintain distance
    - Making for an uncomfortable ride
    - Waste in energy

Precision

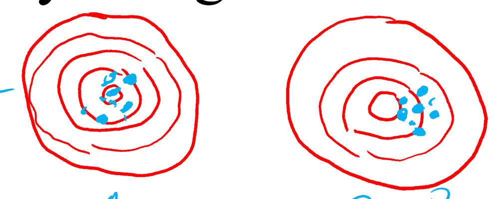
- High Resolution with Low Precision
  - *Usual configuration*
  - *~~Resolution~~ Precision cannot be trusted*
- Ex: A distance sensor
  - *Return value in millimeters*
  - *Precision not really high:* 4.5cm 4.0cm 4.9 → 5.0
    - Ex: 45 mm, 43 mm, 49 mm
  - Only values within the nearest centimeter or half-centimeter can be trusted

Accuracy

Closeness of measurement to real-world quantity being measured

- In robotics
  - Accuracy not as important as precision
    - Since sensors do not directly return physical quantity
  - Physical quantity computed from measured electronic value (dist or speed)
  - Calibrate sensor to get true physical quantity if consistent inaccuracy

10cm 1.48 9.95



- Ex:
  - *Distance sensor using light or sound compute distance from time of flight of signal as:* $$s = \frac{vt}{2}$$
  - *Since, sensor consistently returns value 5cm too large, we revise formula to* $$s = \frac{vt}{2} - 5$$

AD = 50cm SV = 55
60cm 105

Sensors

- Classification of sensors
- Distance sensors
- Cameras
- Other sensors
- Range, resolution, precision
- Nonlinearity

1 = 1°

Nonlinearity

0000 0000 → 0°

0001 → 1.~°

10 → 2.~°

- Sensors
  - *Return electronic quantities (**Volt** or **Amp**) proportional to what is measured*
    - Analog values **converted** to digital values
    - Ex: Proximity sensor return 8-bits of data (values of 0 ~ 255, range of 0 ~ 50)

128 64 32 16 8 4 2 1 - 255

1111 1111

360°

- Calibration
  - *An 8-bit sensor can't return angles in 0 ~ 360° range at 1° resolution*
  - *Computer must translate digital values to measurements of physical quantity*
    - Discovering mapping for this translation is called calibration
  - *Best case: Linear mapping and easy to compute*
  - *Other case: Nonlinear mapping, use table or nonlinear function*
  - *Table more efficient*
    - Entry lookup faster than compute function
    - But large memory requirement

$\frac{360}{255} = 1.5$

Linear Sensors

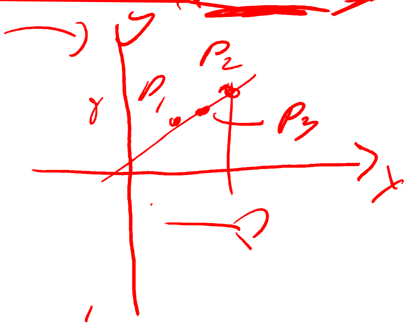
$y = mx + b$

- Linear mapping: $x = as + b$

  Where: $\boldsymbol{x}$ is value returned by sensor, $\boldsymbol{s}$ is distance of object from sensor, $\boldsymbol{a}$ is slope (constant) and $\boldsymbol{b}$ is intercept (slope) with sensor axis.

- Ex: Sensor return 100 if object is 2 cm away, and 0 when 30 cm.

$slope = \frac{\Delta y}{\Delta x} = \frac{y_{P_2} - y_{P_1}}{x_{P_2} - x_{P_1}}$



$P_2(30, 0)$

$P_1(2, 100)$

  – Computing slope & intercept:

$$a = slope = \frac{\Delta x}{\Delta s} = \frac{0-100}{30-2} = -3.57$$

  – When s=30, x =0, b=?

$$0 = (-3.57)(30) + b \rightarrow b = 107$$

  – Therefore

$$x = as + b \rightarrow$$

$$s = \frac{x-107}{-3.57} = \frac{107-x}{3.57}$$

$100 = s = ?$

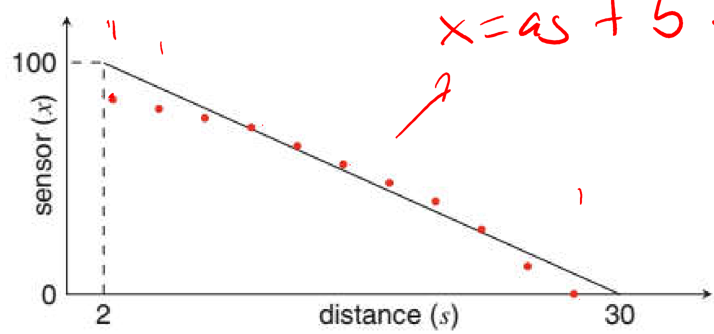
$$\boxed{s = -3.57x + 107}$$

Sensor value as linear function of distance

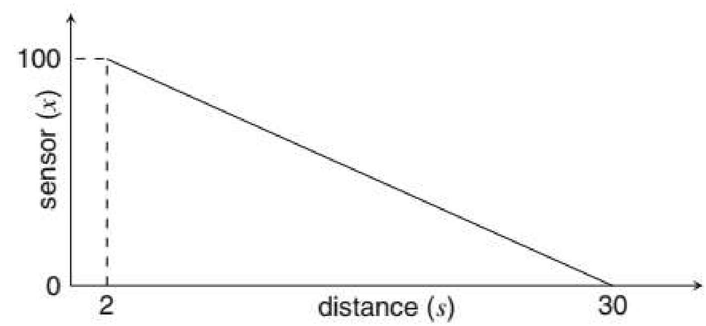
Mapping Linear Sensors

- Ex: Proximity sensor
  - *(a) Linearity plot of raw measure values, with its (b) Linear function*
- Function linear in the middle of range but nonlinear outside of it
  - *Impossible to use linear func of sensor raw values to get distance of object*
- Solution?
  - *Create table mapping sensor values to distances*

$x = as + b$



(a) Sample experimental values



(b) Linear function

Mapping Linear Sensors

- Table of real measurements with an educational robot
  - **(s)**: *Every 2 cms, for range 2 cm ~ 18 cm*
    - Object cannot be detected at 20 cm
  - **(x)**: *sensor value for each distance*
  - **(x_l)**: *value returned if linear fxn* $x = -2s + 48$
    - We can see actual sensor values (x) is not that different from linear values (x_l)
    - Therefore, sounds good to use linear function
  - *Best have entry for each possible returned sensor values*
    - But need lots of memory, esp if range of values is much larger, i.e.: 12 bits (0 ~ 4095)
  - *Possible solution is take nearest value* (**x**) *to one returned by sensor*
    - Ex: if measured value=27 → then mapping would give a distance (**s**) = 12

$s(cm)$	x	x_l
18	14	12
16	18	16
14	22	20
12	26	24
10	29	28
8	32	32
6	36	36
4	41	40
2	44	44

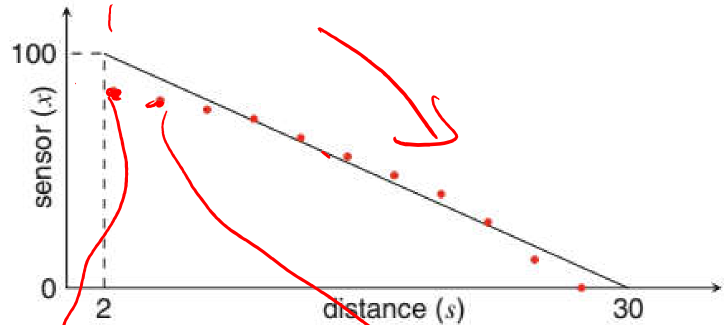
Mapping Linear Sensors

- Better solution is to interpolate
  - *Segments of curve are roughly linear (**a**)*
    - but slopes change according to curve
  - *Take relative distance on straight line between two points (**b**)*
  - *Given: $\mathbf{s_1}$ & $\mathbf{s_2}$ corresponding to sensor values $\mathbf{x_1}$ & $\mathbf{x_2}$.*
  - *For value $\mathbf{x_1} < \mathbf{x} < \mathbf{x_2}$, distance $\mathbf{s}$:*

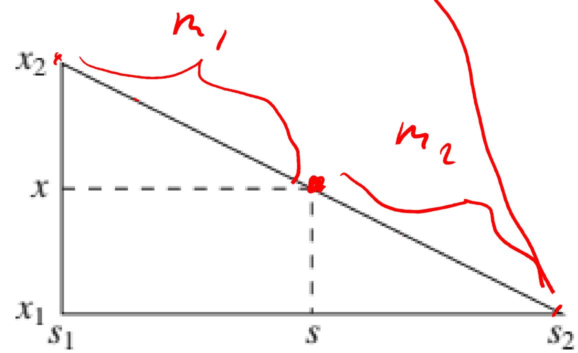
Sensor Value

$$s = s_1 + \frac{s_2 - s_1}{x_2 - x_1}\left(x - x_1\right)$$

$$\frac{s - s_1}{x - x_1} = m_1 = \frac{s_2 - s_1}{x_2 - x_1} \quad m_2$$



(a) Sample experimental values



(b) Linear function

Summary

- Choice of sensors is critical in designing a robot
  - *Designer must decide what needs to be measured*
    - i.e.: distance, attitude, velocity, etc.
  - *Then make trade-off's:*
    - Finer reso, higher prec & accu always better but comes at a price
  - *Educational robots*
    - Price is of great concern but don' t expect excellent performance
  - *Algorithmic principles the same even if using high-quality sensors or not*
- Computer must be able to calibrate sensors

$x = as + b$

  - *If sensor is linear, slope & intercept determine the linear function*
  - *If sensor is non-linear, a table or non-linear function must be used*

look up   interpolation

Thank you.