Robotics

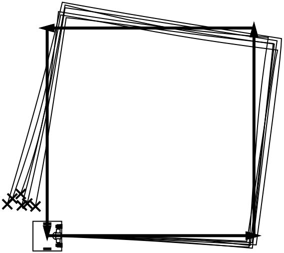
Lecture 3: Sensors

See course website

http://www.doc.ic.ac.uk/~ajd/Robotics/ for up to date information.

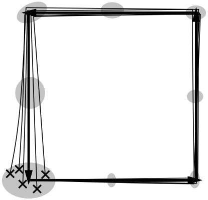
Andrew Davison
Department of Computing
Imperial College London

Review: Locomotion Practical from Lecture 2



- Gradual 'motion drift' from perfect square
- Causes: initial alignment errors; wheel slip; miscalibration; unequal left/right motors; others . . . ?
- Careful calibration of distance and angle (probably in position control, by adjusting demand in degrees, and also possibly maximum speed and gains) improves matters but we will never achieve a perfect result every time in this experiment.

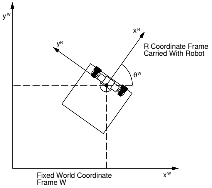
A Well-Calibrated Robot



- After careful calibration the robot should *on average* return to the desired location, but scatter will remain due to uncontrollable factors (variable wheel slip, rough surface, air currents!?. . .)
- *Systematic error* removed; what remains are *zero mean errors*.
- The errors occur *incrementally*: every small additional movement or rotation induces a little more potential error.
- The size of the distribution of the errors in the world frame will grow as the robot moves further around the square.
- We can model the zero mean errors probabilistically: in many cases a Gaussian (normal) distribution is suitable.

What does this mean for estimating motion?

- Perfect motion integration from odometry is not possible:



Recall from the last lecture: state update equations:

- During a straight-line period of motion of distance D:

$$\begin{pmatrix} x_{new} \\ y_{new} \\ \theta_{new} \end{pmatrix} = \begin{pmatrix} x + D\cos\theta \\ y + D\sin\theta \\ \theta \end{pmatrix}$$

- During a pure rotation of angle angle α:

$$\begin{pmatrix} x_{new} \\ y_{new} \\ \theta_{new} \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta + \alpha \end{pmatrix}$$

Uncertainty in Motion

- A better model acknowledges that this 'ideal' trajectory is affected by uncertain perturbations ('motion noise'). For example, we could use this simple model:
- During a straight-line period of motion of distance D:

$$\begin{pmatrix} x_{new} \\ y_{new} \\ \theta_{new} \end{pmatrix} = \begin{pmatrix} x + (D+e)\cos\theta \\ y + (D+e)\sin\theta \\ \theta + f \end{pmatrix}$$

- During a pure rotation of angle α:

$$\begin{pmatrix} x_{new} \\ y_{new} \\ \theta_{new} \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta + \alpha + g \end{pmatrix}$$

- Here e, f and g are 'uncertainty' terms, with zero mean and a Gaussian distribution, which model how actual motion might deviate from the ideal trajectory.
- Adding these terms won't help us to move a robot more accurately when it is guided with only odometry; but are important later when we probabilistically combine odometry with other sensing.

Sensors: Proprioceptive and Outward-Looking

- Sensors are either *proprioceptive* (literally self-sensing) or *exteroceptive* (outward-looking).
- Proprioceptive sensors (such as motor encoders or internal force sensors) will improve a robot's sense of its own internal state and motion.
- But without outward-looking sensors a mobile robot is moving blindly. It needs them to, for example:
  - Localise without drift with respect to a map.
  - Recognise places and objects it has seen before.
  - Map out free space and avoid obstacles.
  - Interact with objects and people.
  - In general, be *aware* of its environment.

Sensor Measurements: Proprioceptive

- Sensors gather numerical readings or *measurements*. In the case of proprioceptive sensors, the value of the measurement $\mathbf{z}_p$ will depend on (be a function of) just the state of the robot $\mathbf{x}$:

$$\mathbf{z}_p = \mathbf{z}_p(\mathbf{x}) \ .$$

- More generally, a proprioceptive measurement might depend not just on the current state but also previous states in the robot's history or the current rate of change of state. e.g. wheel odometry will report a reading depending on the difference between the current and previous state. A gyro which is part of an Inertial Measurement Unit (IMU) will report a reading depending on the current *rate* of rotation.

Sensor Measurements: Outward-Looking

- A measurement from an outward-looking sensor will depend both on the state of the robot $\mathbf{x}$ and the state of the world around it $\mathbf{y}$:

$$\mathbf{z}_o = \mathbf{z}_o(\mathbf{x}, \mathbf{y}) \ .$$

- The state of the world might be parameterised in many ways; e.g. a list of the geometric coordinates of walls or landmarks; and may either be uncertain or perfectly known.

Single and Multiple Value Sensors

- Touch, light and sonar sensors each return a *single value* within a given range.
- Sensors such as a camera or laser range-finder return an *array* of values. This can be achieved by scanning a single sensing element (as in a laser range-finder) or by having an array of sensing elements (such as the pixels of a camera's CCD chip).

Touch Sensors

- Binary on/off state — no processing required.
- Switch open — no current flows.
- Switch closed — current flows (hit).

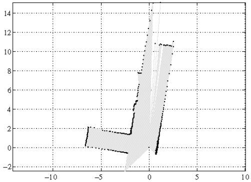
Light Sensors

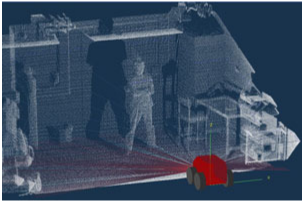
- Detect intensity of passive light incident from a single forward direction, with some range of angular sensitivity.
- Multiple sensors pointing in different directions can guide steering behaviours.
- The Lego sensors also have a mode where they emit their own light, which will reflect off close targets and can be used for following a line on the floor or quite effective short-range obstacle avoidance.

Sonar (Ultrasonic) Sensors

- Measures depth (distance) by emitting an ultrasonic pulse and timing the interval until echo returns. Sonar beam typically has an angular width of 10 to 20 degrees.
- Fairly accurate depth measurement (centimetre) in one direction but can give noisy measurements in the presence of complicated shapes. Maximum range a few metres.
- Robots sometimes have a ring of sonar sensors for obstable detection.
- Especially important underwater where it is the only serious option beyond very short ranges.

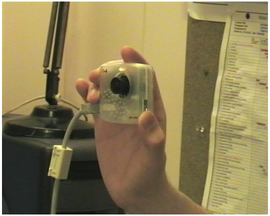
External Sensing: Laser Range-Finder





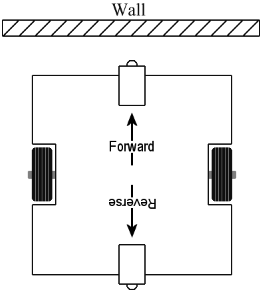
- Like a sonar, measures depth using an active signal. Commercial Ladar sensors return an array of depth measurements from a scanning beam.
- Very accurate measurement depth (sub-millimetre) and works on most types of surface.
- Normally scans in a 2D plane but 3D versions are also available
- Rather bulky (and expensive) for small robots

External Sensing: Vision



- The generalisation of a light sensor. A camera measures passive light intensity in many directions simultaneously by directing incident light onto a light sensitive chip.
- Returns a large, rectangular array of measurements.
- A single camera measures light intensity, rather than any direct information about geometry. 3D information processing and matching with data from either multiple cameras a single moving one.
- Vast research area: object recognition, location recognition, tracking, 3D reconstruction, etc.
- Huge motivation from biology.
- Highly attractive for general purpose, low-cost robotics.

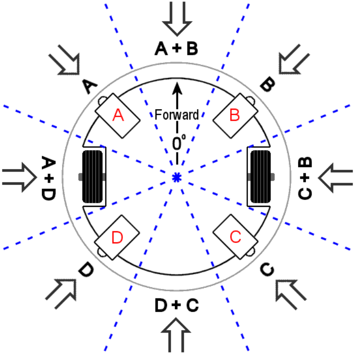
Touch Sensors for Bump Detection



- Sensor detects when an obstacle has been hit (last line of defence).
- Demands immediate reaction — evasive manoeuvre, or stop forward motion at least.

Multiple Touch Sensors — Where was I hit?

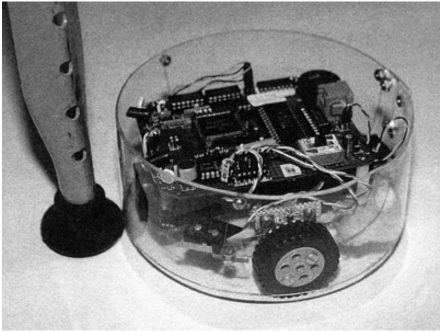
Touch sensors mounted inside ‘floating skirt’ around circular robot:



A	B	C	D	Sector	Centre
1	1	0	0	337.5° to 22.5°	0°
0	1	0	0	25.5° to 67.5°	45°
0	1	1	0	67.5° to 112.5°	90°
0	0	1	0	112.5° to 157.5°	135°
0	0	1	1	157.5° to 202.5°	180°
0	0	0	1	202.5° to 247.5°	225°
1	0	0	1	247.5° to 292.5°	270°
1	0	0	0	292.5° to 337.5°	315°

- Four sensors give the ability to measure eight bump directions.

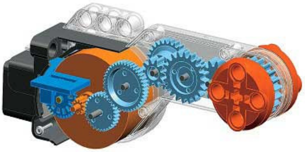
Strategies After a Collision

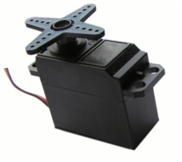


- Try to move around object: reverse, and try to go around (e.g. turn to a fixed angle from hit and proceed).
- Random bounce: rotate through random angle and head off straight again until next collision.

Servoing

- Servoing is a robot control technique where control parameters (such as the desired speed of a motor) are coupled directly to a sensor reading and updated regularly in a *negative feedback loop*. It is also sometimes known as *closed loop control*.
- Servoing needs high frequency update of the sensor/control cycle or motion may oscillate.
- The way that a motor controls its motion using encoder feedback is one example of servoing and negative feedback. This concept can also be used with outward-looking sensors.



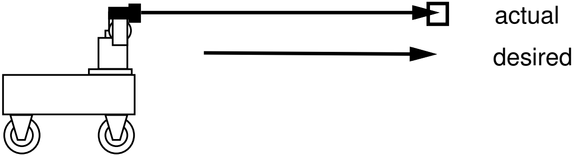


Proportional Control using an External Sensor: Servoing

- In servoing, a control demand is set which over time aims to bring the current value of a sensor reading into agreement with a desired value.
- Proportional control: set demand proportional to negative error (difference between desired sensor value and actual sensor value): e.g. set velocity proportional to error:

$$v = -k_p(z_{desired} - z_{actual}) \ ,$$

  where k_p is the proportional gain constant. (Note that since this is a different control loop, k_p will not have the same value as in last week's motor tuning but will need to be individually adjusted through trial and error.)



- Proprotional control is a special case of more general *PID Control* (Proportional, Integral, Differential).

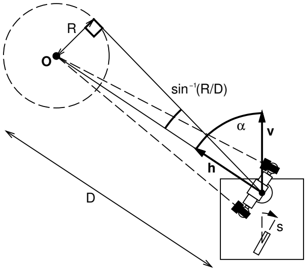
Visual Servoing to Control Steering

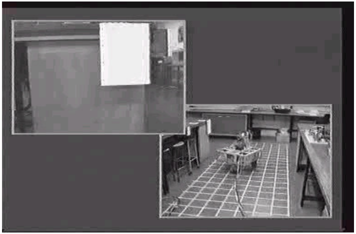
- For a robot with a tricycle or car-type wheel configuration.
- Simple steering law which will guide robot to collide with target:

$$s = k_p \alpha$$

- Steering law which will guide robot to avoid obstace at a safe radius: subtract offset:

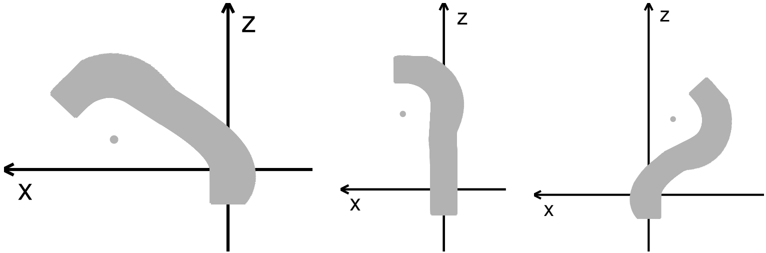
$$s = k_p(\alpha - \sin^{-1} \frac{R}{D})$$



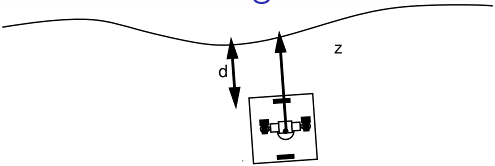


- https://youtu.be/Ce2FrSBKwB8

Visual Servoing Trajectories



Wall Following with Sonar



- Use sideways-looking sonar to measure distance z to wall.
- Use velocity control and a loop at for instance 20Hz.
- With the goal of maintaining a desired distance d, set difference between left and right wheel velocities proportional to difference between z and d:

$$v_R - v_L = K_p(z - d)$$

Symmetric behaviour can therefore be achieved using a constant offset v_C:

$$v_R = v_C + \frac{1}{2}K_p(z - d)$$

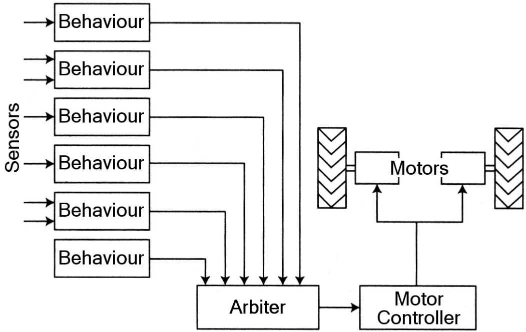
$$v_L = v_C - \frac{1}{2}K_p(z - d)$$

Combining: Sensing/Action Loops

- No modelling and planning! Consider each local ‘servo’-like sensing-action loop as a **behaviour**.

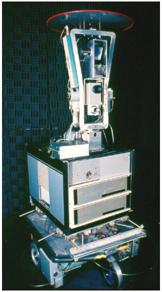
Sense $\rightarrow$ Act

- The challenge is in combining many behaviours into useful overall activity: see TR Programs, Subsumption, Braitenberg vehicles.



Combining Sensors: World Model Approach

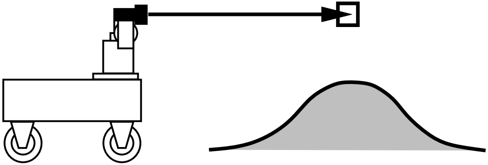
- Capture data; store and manipulate it using symbolic representations.
- *Plan* a sequence of actions to achieve a given goal
- Execute plan.
- If the world changes during execution, stop and re-plan.
- Powerful, but computationally expensive and complicated!



Shakey: one of the first mobile robots.

- Probabilistic state inference and planning is the modern version of this able to cope with uncertainty in sensors.

Probabilistic Sensor Modelling



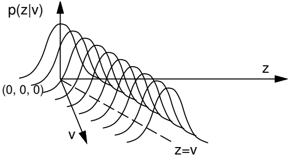
- Like robot motion, robot sensing is also fundamentally *uncertain*. Real sensors do not report the exact truth of the quantities they are measuring but a perturbed version.
- Having characterized (modelled; calibrated?) a sensor and understood the uncertainty in its measurements we can build a probabilistic measurement model for how it works. This will be a probability distribution (specifically a *likelihood function*) of the form:

$$p(\mathbf{z}_o|\mathbf{x}, \mathbf{y}) \ .$$

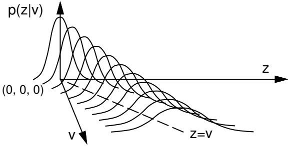
  Such a distribution will often have a Gaussian shape.

Likelihood Functions

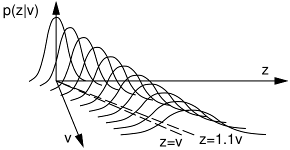
- A likelihood function fully describes a sensor's performance.
- $p(z|v)$ is a function of both measurement variables z and ground truth v and can be plotted as a probability surface. e.g. for a depth sensor:



Constant Uncertainty



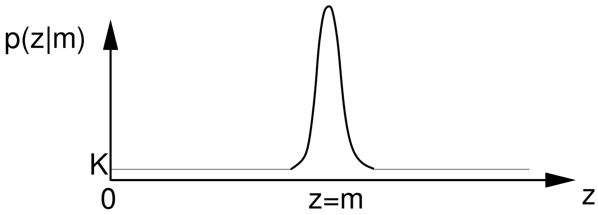
Growing



Systematic Error (Biased)

Robust Likelihood for Sonar Measurements

- We will return to this in later lectures on probabilistic robotics, but a suitable likelihood function for a sonar sensor is as follows: it says 'what is the probability of obtaining sensor measurement z given that the ground truth value I expect is m?
- This distribution has a narrow Gaussian band around the expected value, plus a constant additive band representing a fixed percentage of 'garbage' measurements.



$$p(z|m) \propto e^{\frac{-(z-m)^2}{2\sigma_s^2}} + K$$

This week's practical: Simple Sensor Control Loops and Wall Following

- Example wall follower: `https://www.youtube.com/watch?v=BU9k5Z0CKjs`. We will try to do even better with smooth proportional gain!