

Introduction to **Mobile Robots**

Dr. Mario Gongora

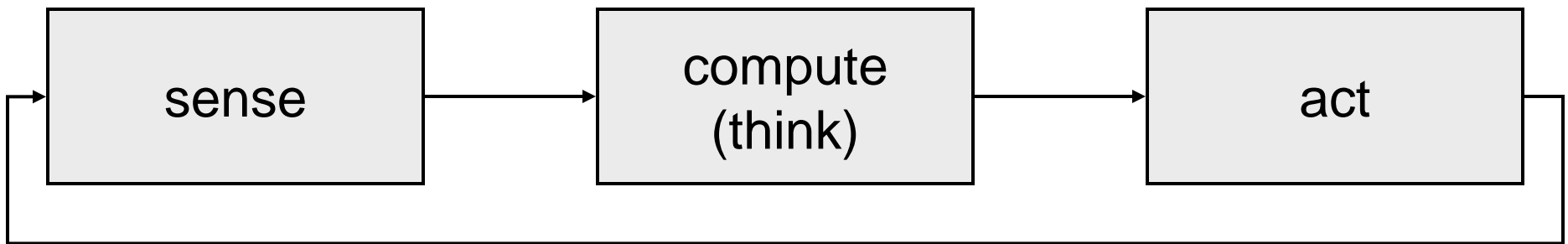
mgongora@dmu.ac.uk

<http://www.cse.dmu.ac.uk/~mgongora/>

Control Models

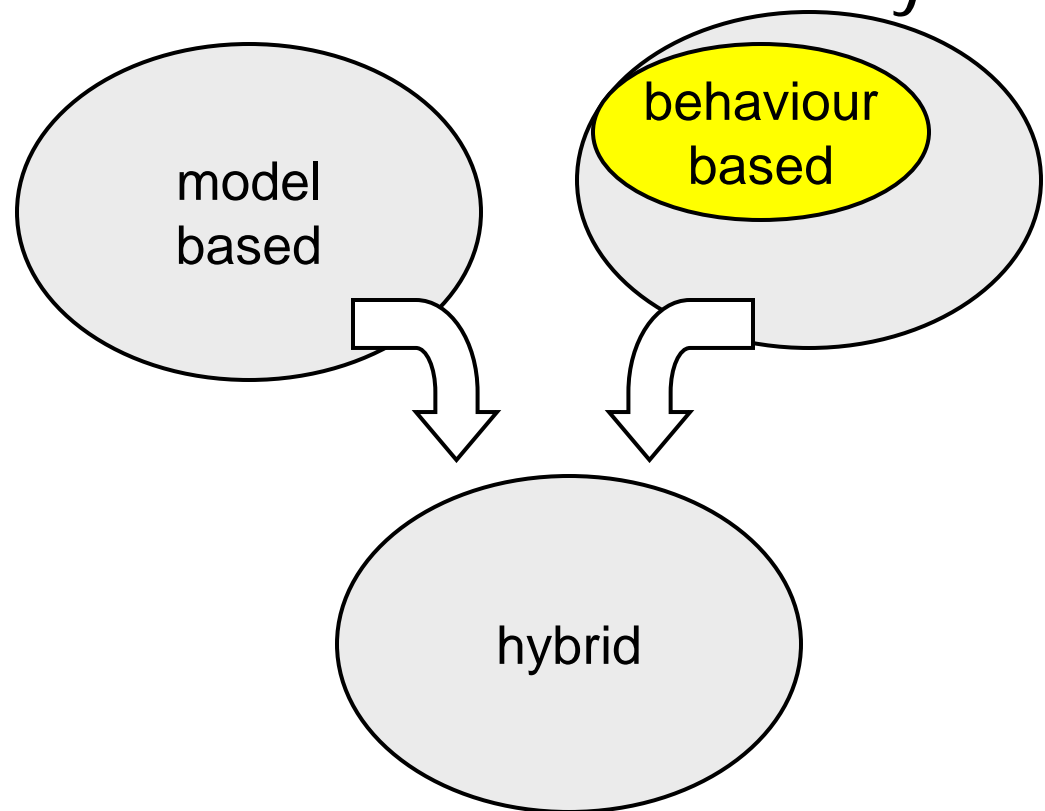
The Control Cycle

- A fundamental methodology derived in the early days of robotics from engineering principles is the *sense-think-act* cycle
 - the principle is to continuously attempt to minimise the error between the actual state and the desired state
 - based on control theory



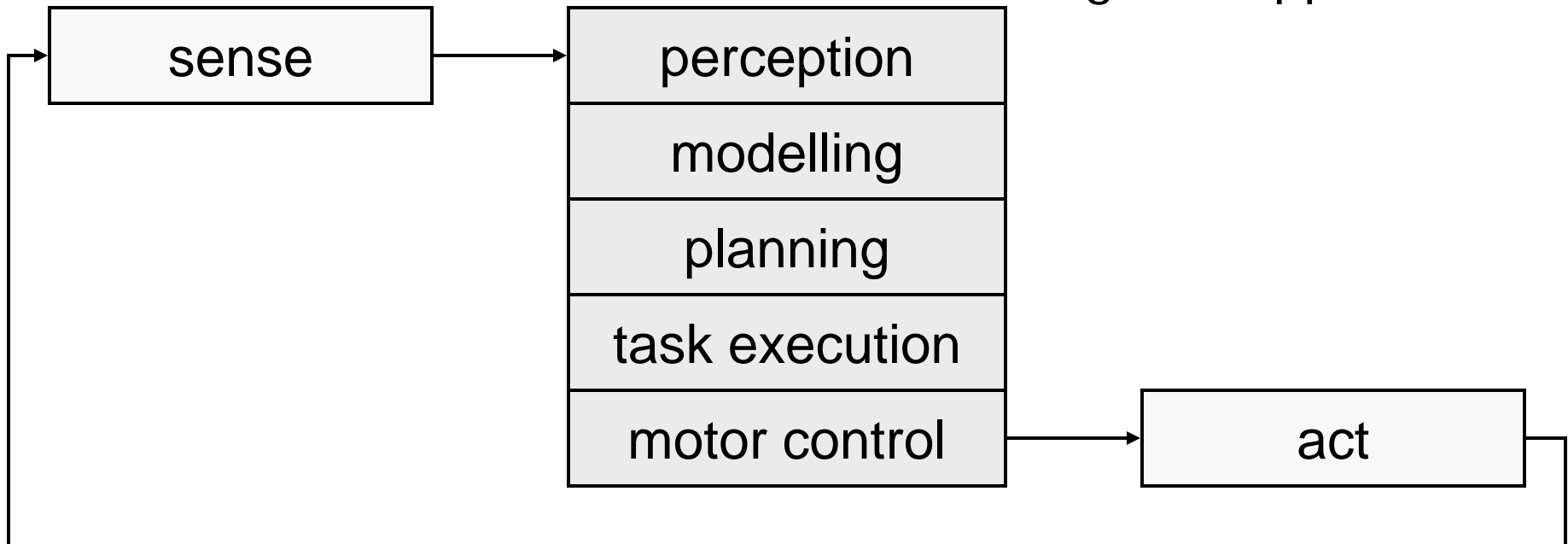
Control Architectures

- A variety of different approaches have been tried for implementing the *sense-think-act* control cycle
- These approaches can be categorised as
 - model-based
 - reactive
 - hybrid



Model Based

- A symbolic internal 'world-model' is maintained
 - the sub-tasks are *decomposed* into **functional layers**
 - similar to 'classical' artificial intelligence approach





Problems with Models

- An **adequate, accurate** and **up-to-date** model must be maintained at all times
 - this is very difficult in practice!
 - what if sensors detect an object that hasn't been defined
- A model-based system is **extremely brittle**
 - if one of the functional layers fails (e.g. hardware problems, software bugs), then the whole system fails
- Significant **processing power** is required
 - maintaining the model takes time, so slow responses!?
- Despite much effort, little progress was made!

Reactive Robotics

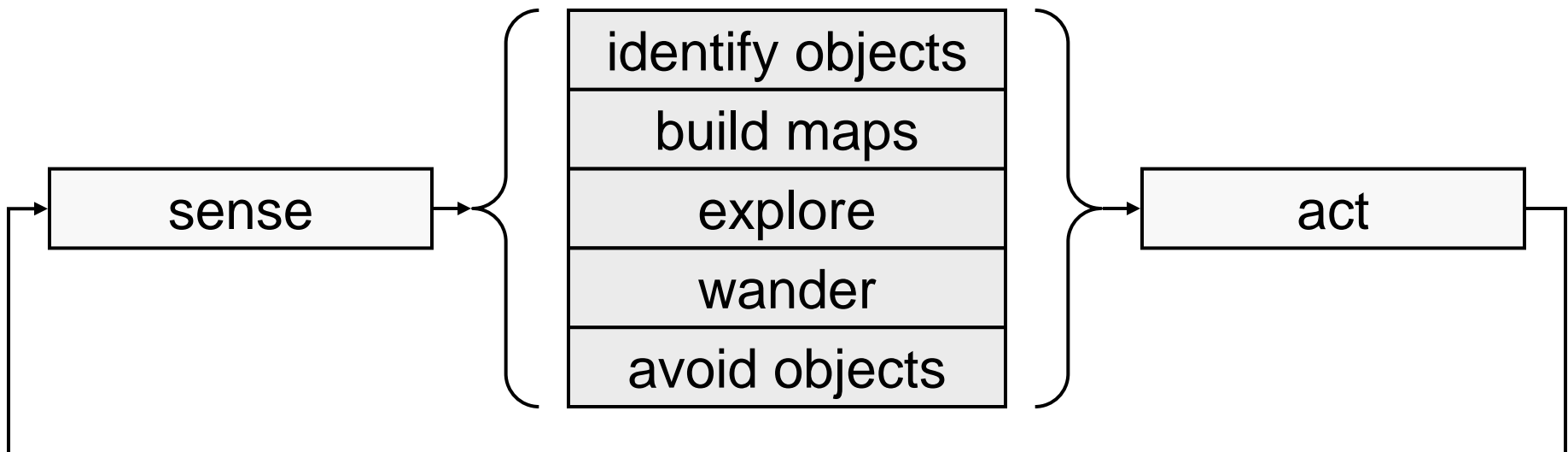


Reactive Controllers

- In order to try to overcome the shortcomings of model-based robots, modern approaches have centred predominantly on simple *reactive* systems with minimal amounts of computation
 - 'model-free systems'
- More correctly, the models are *simple* and *implicit*
 - the systems do not use symbolic models but, for example, a rule-set which tells a robot how to react to a corner when following a wall may be considered to be a simple, implicit model fragment
 - it implicitly encodes assumptions about the environment

Behaviour Based

- The control system is broken down into horizontal modules, or *behaviours*, that run in parallel
 - each behaviour has direct access to sensor readings and can control the robot's motors directly





Behaviour Advantages

- It supports **multiple goals** and is more efficient
 - there is no functional hierarchy between layers
 - one layer does not call another layer
 - each layer can work on different goals in parallel
 - communication between layers is achieved via message passing which need not be synchronised
- The system is easier to **design, debug** and **extend**
 - each module can be designed and tested individually
- The system is **robust**
 - if one module fails, e.g. *wander*, then other layers, e.g. *avoid obstacles*, still function and behave correctly



Behaviour Limitations

- It is extremely difficult to implement **plans**
 - in pure form a behaviour-based robot has no memory (not even an internal state memory) and so is unable to follow an externally specified sequences of actions
- It can be very hard to predict how a large number of multiple behaviours may interact
 - *emergent behaviour* is the term given to unexpected behaviour that comes about through these interactions
 - sometimes it is useful, sometimes it is not!
- The robot can get trapped in a *limit cycle*
 - trapped in a dead-end, repeatedly turning left then right

Other Approaches

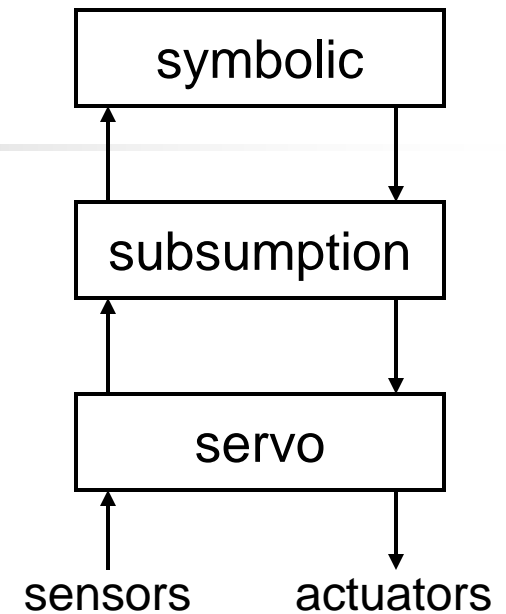


Other Reactive Approaches

- Two other reactive approaches that are popular
- **Potential field methods**
 - a potential field is a concept from physics
 - e.g. the *gravitational field*
 - you do not need to be told which way to fall
 - planets do not need to plan how to move around the sun
 - obstacles exert hypothetical repulsive forces on the robot
- **Motor schema navigation**
 - multiple, concurrent schema generate separate behaviours which are summed to produce output
 - schema are dynamically created/destroyed as needed

Hybrid Approaches

- The *SSS* three-layer architecture
 - the *servo-subsumption-symbolic* architecture combines Brooks' architecture with a lower-level servo control level and a higher-level symbolic system [Connell]
 - It provides a neat way to combine the advantages of each architecture:
 - Quick response of servo
 - Robust response of reactive (subsumption)
 - Planning capabilities of model based (symbolic)





Learning Approaches

- Traditional learning techniques
 - rather than attempt to predefine and predict a symbolic model of the 'real-world', the robot learns how to operate and how to behave
- Evolutionary algorithms
 - using genetic algorithms to find good network weights
 - significant problems with evolving real solutions in reasonable time on current mobile robot hardware



Introduction to Sensors



Robot Sensors

- Sensors are devices that can sense and **measure** physical properties of the environment,
 - e.g. temperature, luminance, weight, size, etc.
- They deliver *low-level* information about the environment the robot is working in
- This information is
 - *noisy (imprecise)* and even *contradictory and ambiguous*
- Sensors and Detectors are similar but **NOT** equal



Sensor Characteristics

- All sensors are characterised by a number of properties that describe their capabilities
 - Sensitivity: $(\text{change of output}) \div (\text{change of input})$
 - Linearity: constancy of $(\text{output} \div \text{input})$
 - measurement range: difference between min. and max.
 - response time: time required for a change in input to cause a change in the output
 - Accuracy: difference between measured & actual
 - Repeatability: difference between repeated measures
 - Resolution: smallest observable increment
 - Bandwidth: range of frequencies it can respond to



Sensor Types

- It is impossible for us not to view the world in a human-centric manner, but this is not absolute:
 - vision is our main sense, as the human brain is dominated by a large **visual cortex**
 - the dog brain is dominated by an **olfactory cortex**, with 125 to 220 million smell-sensitive receptors in the olfactory bulb
 - cockroaches have **30,000** wind-sensitive hairs on legs
- There are many different types of sensor
 - many sense phenomena that humans cannot detect
 - e.g. magnetism, infra-red, ultra-violet, ultrasound, phase of light (e.g. polarised sun-glasses), etc.



Active vs. Passive

- Sensors can be broadly classified as either
 - *active*
 - radiating some form of energy into the environment
 - e.g.
 - radar (radio direction and ranging)
 - sonar (sound navigation and ranging)
 - lidar (light direction and ranging)
 - *passive*
 - relying on energy emitted by various objects in environment
- It is tempting to imagine that passive sensors do not interfere with the environment
 - *ANY sensor may affect the environment*



Internal State vs. External State

- The variable that a Sensors reads can provide information either
 - *About the internal state of the Robot*
 - Internal variables, e.g. battery level
 - Self measurements, e.g. odometry
 - Time
 - *About the external environment*
 - External variables, e.g. outside temperature
 - Position relative to external objects, e.g. obstacles
 - Communications



Non-Visual Sensors



Proximity Sensors

- Tactile sensors allow obstacle *detection* but proximity sensors are needed for true obstacle *avoidance*
- Several technologies can detect the presence of particular fields without mechanical contact
 - magnetic reed switches
 - two thin magnetic strips of opposite polarity not quite touching
 - an external magnetic field closes the strip & makes contact
 - Hall effect sensors
 - small voltage generated across a conductor carrying current
 - inductive sensors, capacitive sensors
 - inductive sensors can detect presence of metallic objects
 - capacitive sensors can detect metallic or dielectric materials



Infrared Sensors

- Infrared sensors are probably the simplest type of non-contact sensor
 - widely used in mobile robotics to avoid obstacles
- They work by
 - emitting infrared light, (to differentiate emitted IR from ambient IR - e.g. lights, sun, etc.- , a code is added)
 - detecting any reflections off nearby surfaces
- In certain environments, with **careful calibration**, IR sensors can be used for object distance
 - requires uniform surface colours and structures



Sonar Sensors

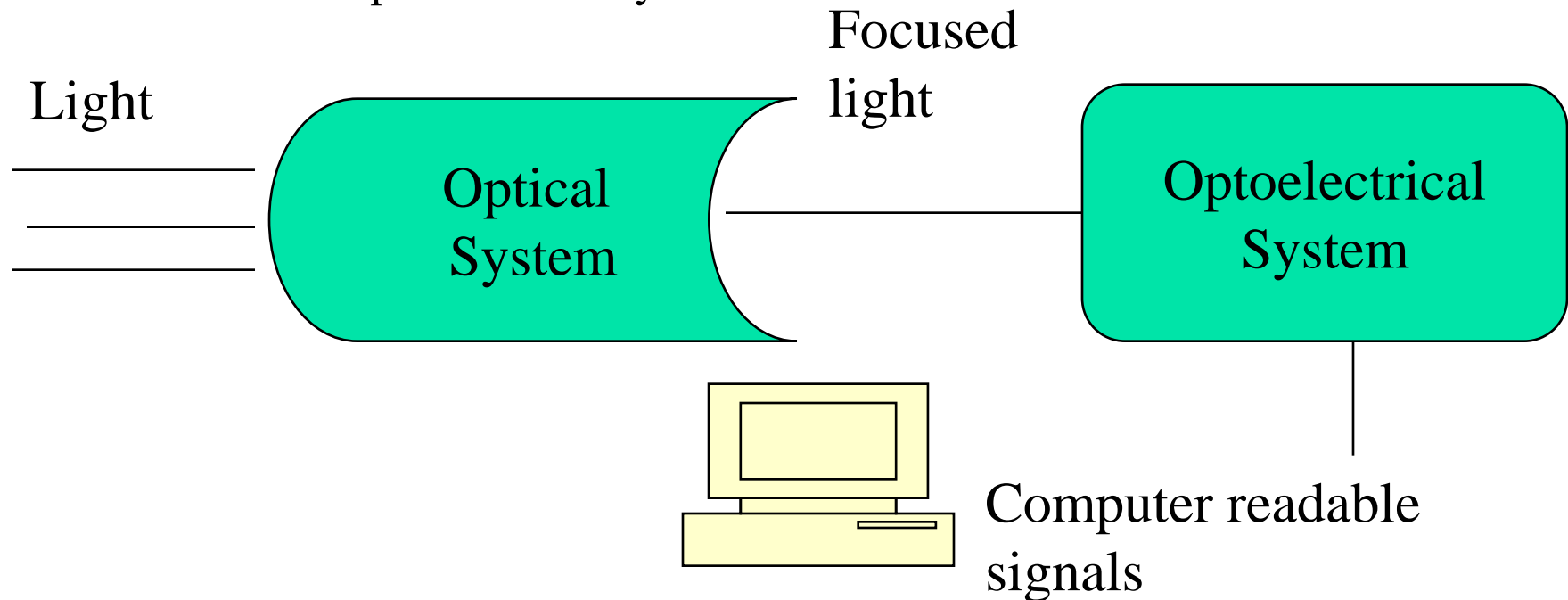
- The fundamental principle of robot sonar sensors is the same as that used by bats
 - emit a chirp (e.g. 1.2 milliseconds)
 - a short powerful pulse of a range of frequencies of sound
 - its reflection off nearby surfaces is detected
- As the speed of sound in air is known ($\approx 330 \text{ m}\cdot\text{s}^{-1}$) the distance to the object can be computed from the elapsed time between chirp and echo
 - minimum distance = $165 t_{chirp}$ (e.g. 21 cm at 1.2 ms)
 - maximum distance = $165 t_{wait}$ (e.g. 165 m at 1 s)
- Usually referred to as *ultrasonic sensors*



Visual Sensors

Visual Sensors

- Visual sensors are based on light. They consist of two main components:
 - An optical system
 - An optoelectrical system



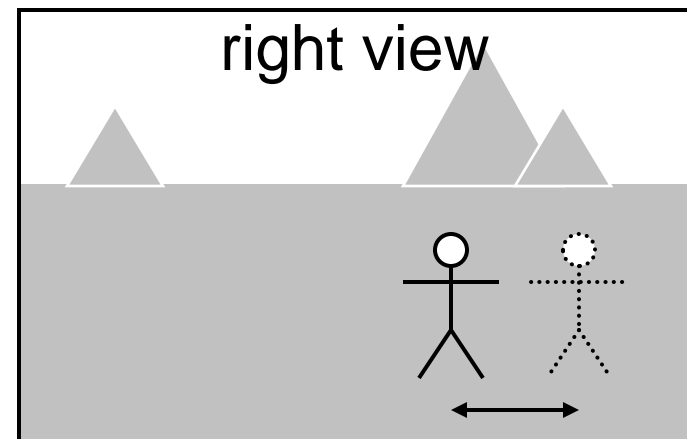
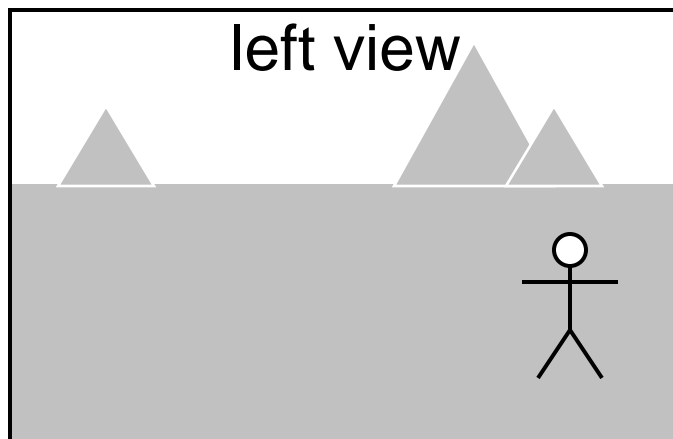


Visual Sensors

- A digital image is a collection of samples representing the visual content of the environment.
- Each individual sample is represented in the form of a pixel or picture element. Each pixel is a point on rectangular lattice with finite resolution encoding some form of intensity.
- For the image to look unbroken (continuous) a minimum number of samples should be taken. The minimum sampling frequency is known as the Nyquist frequency:
 - The signal should be sampled ‘at a rate at least twice as great as the highest frequency in the signal’

Stereoscopic Vision

- Viewing the world with two cameras (eyes) allows a three-dimensional representation to be formed
 - unfortunately the signal is complex and noisy
- Each camera receives a slightly different view
 - the distance between corresponding points in an image is known as the *stereo disparity*



disparity



Object Recognition

- Much research has gone into the field of object recognition (it seems so easy for us humans!), but the problem has not been solved yet
 - it seems humans use a combination of techniques
- Edge detection
 - fairly simple filter operations can detect clean edges
 - e.g. the discrete Laplace filter
 - reliable detection of all edges is much more difficult
- Area based techniques
 - connected regions of similar colour, texture or brightness probably belong to the same object



Actuators

Locomotion



Mobility Considerations

- Issues impact selection of drive
 - *manoeuvrability* ability to alter direction/speed
 - *controllability* practical and not too complex
 - *traction* sufficient to minimise slippage
 - *climbing ability* traversal of minor discontinuities, slope rate, surface type, terrain

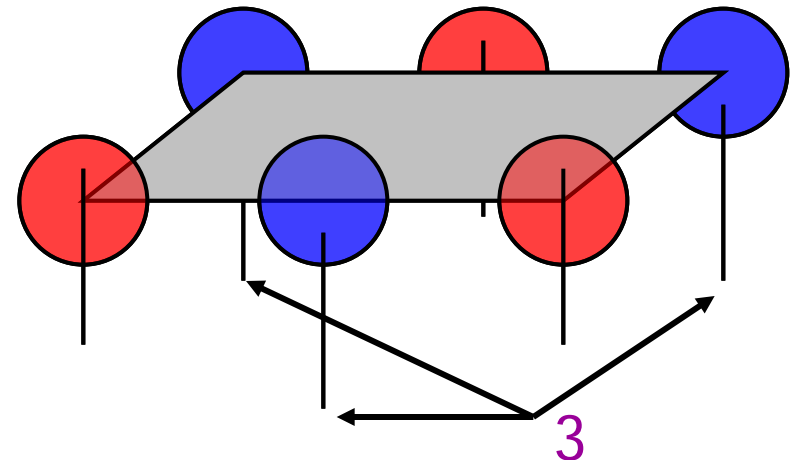


Mobility Considerations

- Issues that impact selection of drive (cont...)
 - *stability* mustn't fall over!
 - *efficiency* power consumption reasonable
 - *maintenance* easy to maintain, reliable
 - *environmental impact* does not do damage
 - *navigation* accuracy of dead-reckoning

Legs

- Two legs seems the most familiar configuration
 - But balance is an incredibly difficult problem
 - Need very fast and supple joints to move and balance
 - Standing still is an active task (uses energy!!)
- Four legs are much easier to balance and move ... are they?
- Six legs are much easier to balance and move
 - Stable when not moving
 - Can work with simple cams and rigid legs
 - Processing is much simpler!





Wheels

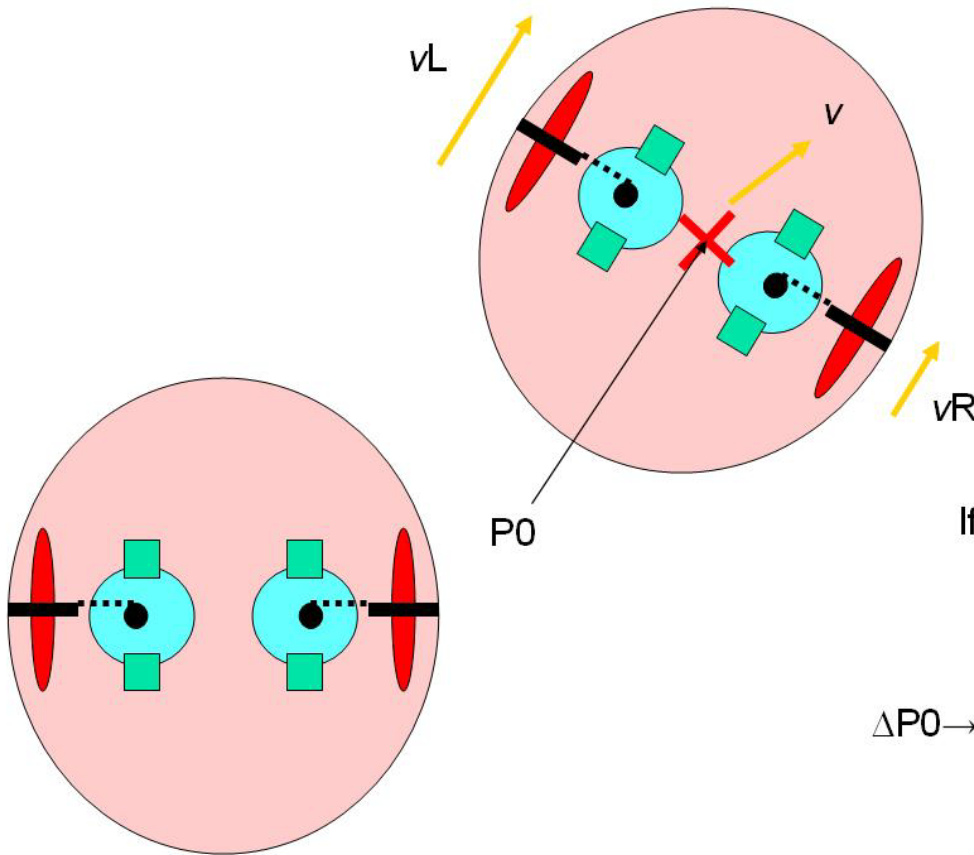
- Again, any number of wheels is possible
 - there are many different configurations that are useful
- Two individually driven wheels on either side
 - usually with one or more *idler wheels* for balance
 - independently driven wheels allows zero *turning radius*
 - one wheel drives forwards, one wheel drives backwards
- Rear wheel drive, with front wheel steering
 - the vehicle will have a non-zero turning radius
 - for two front wheels, turning geometry is complex
 - the rear wheels need a *differential* to prevent slippage
- 4WD is possible, but it is even more complex



Exotic Wheels and Tracks

- Tracks can be used in the same way as 2 wheels
 - good for rough terrain (as compared to wheels)
 - tracks must slip to enable turns (*skid steering*)
- In *synchro drive* three or more wheels are coupled
 - drive in same direction at same rate
 - pivot in unison about their respective steering axes
 - allows body of robot to remain in the same orientation
- Tri-star wheels are composed of 3 *sub-wheels*
 - entire wheel assembly rolls over a large obstacle
- There are many other exotic wheel configurations
 - multiple-degrees-of-freedom (MDOF), e.g. Kilough

Wheels and Tracks



If $v_L = -v_R$:

$$\Delta P_0 = \{$$

$$|\Delta P_0| = 0;$$

$$\Delta P_{0 \rightarrow} = |v_L * \Delta t| / R\omega;$$

$$\}$$

If $v_L \neq v_R$ and Δt is small:

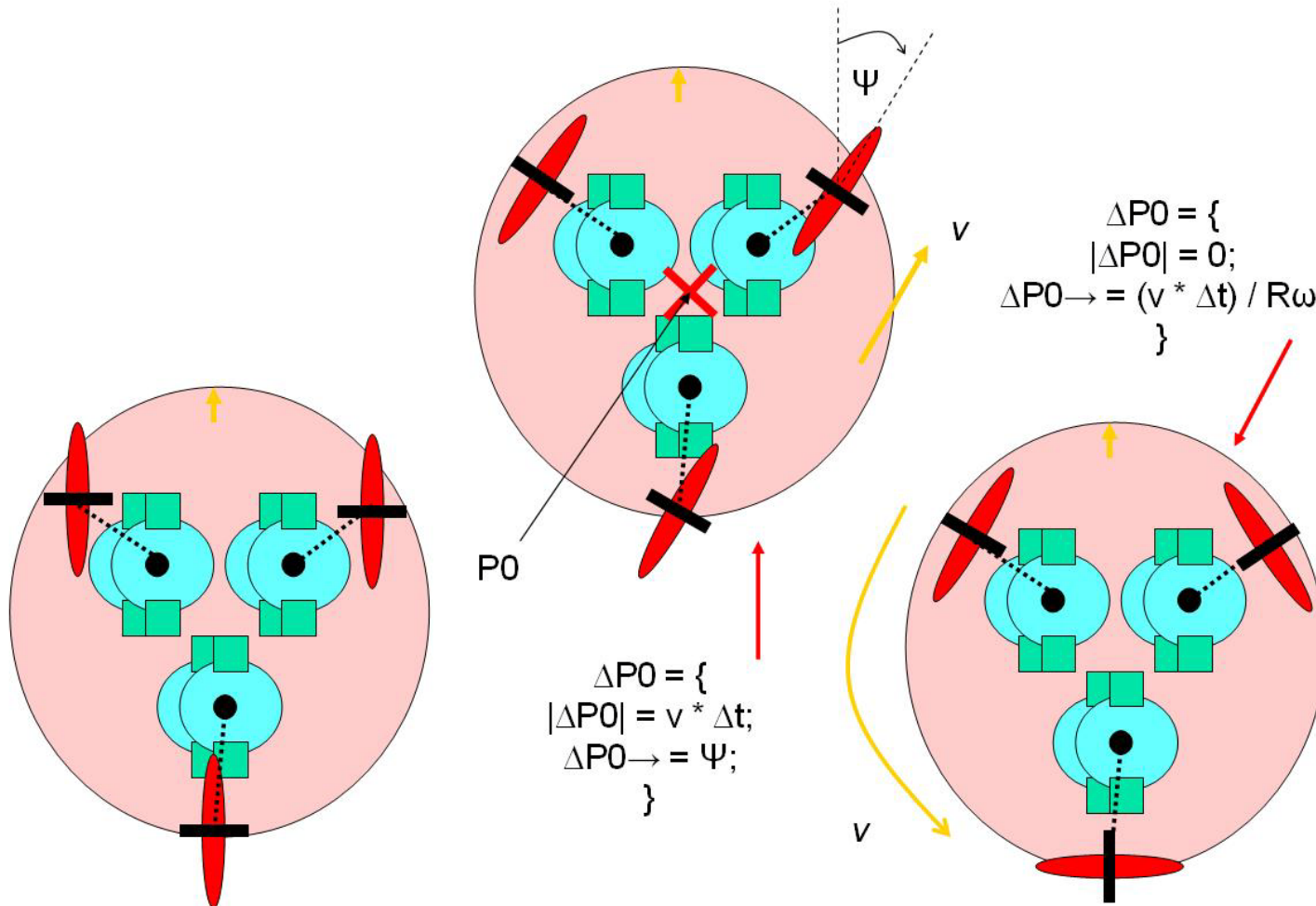
$$\Delta P_0 = \{$$

$$|\Delta P_0| = |v_L + v_R| / 2 * \Delta t;$$

$$\Delta P_{0 \rightarrow} = (|v_L - v_R| / 2 * \Delta t) / R\omega;$$

$$\}$$

Exotic Wheels and Tracks





Actuators

Manipulation



Actuator Types

- Electric
 - DC motor is most common type used in mobile robots
 - stepper motors turn a certain amount with a voltage pulse
- Pneumatic
 - operate by pumping compressed air through chambers: slow.
- Hydraulic
 - pump pressurised oil: usually too heavy, dirty and expensive to be used on mobile robots
- Shape memory alloys (SMA's)
 - metallic alloys that deform under heat and then return to their previous shape: *used for artificial muscles*



Arms and Grippers

- *Degrees-of-freedom*
 - independently controllable components of motion
- Arms
 - convenient method to allow full movement in 3D
 - more often used in fixed robots due to power & weight
 - even more difficult to control!
 - due to extra degrees of freedom
- Grippers
 - may be very simple (two rigid arms) to pick up objects
 - may be complex device with *fingers* on end of an arm
 - probably need feedback to control grip force
 - e.g. picking up an egg!



Measuring Motion



Odometry

- If wheels are being used, then distance traveled can be calculated by measuring number of turns
 - *dead-reckoning* or *odometry* is the name given to the direct measure of distance (for navigation)
- Motor speed and timing are very inaccurate
 - measuring the number of wheel rotations is better
 - *shaft encoders*, or *rotation sensors*, measure this
 - Can be used in Mobile Robots as well as Manipulators



Odometry

- There are many different types and technologies of shaft encoder
 - Optical encoders (relative)
 - Potentiometers (absolute for each turn)
- In practice, slippage means that dead-reckoning is unreliable over all but very short distances
- And is a relative measurement at the end, so...