Lecture 5: Intelligent Control, Part 2

King's College London

23 October 2018
(version 1.0)

Outline

- Today
- Navi
- Cog
- Behaviour Based Systems
- Multi-Robot Systems

-
-
-
- Multi-Robot Systems

Outline

Assignment Project Exam Help

Today

Navi

https://eduassistpro.github.i

Cog

Behaviour Based Systems

Add WeChat edu_assist_pr

Multi-Robot Systems

What counts as navigation?

- Navigation is concerned with how a robot gets around the world.

-

- We assume that the robot:
  - Knows where it is.
  - Knows where it is going.

- We are concerned with getting the robot from one place to another.

-
  - Local navigation

- Global navigation is about deciding how to get from some start point to a goal point.
-
-
- series of steps — to get it from its current location to its goal.
  - The "plan" is typically a sequence of wayp

- We will look at some different methods that ar for different map representations.
  - Remember them?

- Local navigation is about obstacle avoidance.
-
  kind of information the robot has about the w
  - Depends on sensors

- One way to think about the difference betwe
  terms of the relationship between the robot
  the goal point.
- If there is a clear line of sight between the start point and the goal, then we only need to worry about obstacle avoidance.
  - Just avoiding some debris that isn't on the map

- However, if there is no line of sight from start to g have to find a path.
- Typically path segments will be between two points between which there is a line of sight.
  - Path segments connect waypoints

- Direct implementation of line-of-sight
- Connect up all the vertices in the map.
- Given the line segments, look for the shortest path from start to goal.
- Then translate the path into a series of waypoints (i.e., the end points of the line segments).

- Given the visibility graph on the previous slide, there is an obvious problem with using the lines as a guide for where the robot should go.
-
- objects.
  - Problems with collisions.

- Fix this by expanding objects by enough that t still clear them.
  - More than half the diameter of the robot.

Voronoi diagram

- A Voronoi diagram is a way to divide up a plane (a
- Given a set of points P, a Voronoi dia
  such that the points inside each polygon are closer to one member of P than any other.

- Can extend this to cases where P is a set of objects.
- Treat the line segments exactly like the edges in the visibility graph.

-
- lines of sight can.

- Voronoi diagrams also have a nice property in terms of
-
- waypoints.
  - Head to the next waypoint while maximi objects (obstacles).

- Voronoi diagrams work in 3D also.

- Voronoi diagrams were also famously used by John Snow to identify the source of the 1854 cholera epidemic in London.

- Exact cell decomposition

- Fixed cell decomposition

- Adaptive cell decomposition

- Given the maps, we still want to figure out a sequence of line

-
-
  - Explicit search of a connectivity graph
  - Wavefront planning

- These are really the same thing in different gui

- We need to identify which cells are next to which other cells.

- The question is how to figure out a path from the graph.
- When the graph is complex, we need to use search techniques.
-

- Standard approaches to search:
  - Breath first
  - Depth first
  - A*

- Plus there are robotics-specific approaches like D*.

- A general algorithm for search is:

```
agenda = initial node;
while agenda not empty do {
    state <- node from agenda;



        }
    add new nodes to agenda;
}
```


- Note that this doesn't generate a set of waypoints, it just looks for the goal state.
- In fact, it assumes that there are already set of possible waypoints.

- Let's think about how this search would work on the connectivity graph:

-

  `add new nodes to agenda;`

Depth-first search:

- Takes the first node on the agenda;
-
-

Breadth-first search:

- Takes the first node on the agenda;
- Adds new nodes to the back of the agenda.
- Explores all the nodes at one "level" before looking at the next level.

- A* search focuses the search by giving each node a pair of weights:

- We pick from the agenda by choosing the nod
  cost.
  (Choosing like this means we don't have to wo
  order we put nodes onto the agenda).

- In some domains we have to design clever functions to determine what "far" is
- In robotics we can just use Euclidean or Manhattan distance between points:

  - Manhattan distance

$$d^h_{s,g} = |(x_g - x_s$$

- Of course the distance to the goal may be an underestimate (may be no route through), but it turns out that this is a good thing for A*.

- Often in robotics we need to replan.
-

-
  - Usually have to replan from the robot to the change is near the robot.
  - That is where the robot senses failure

- In all these approaches we have to extract the waypoints after we find the goal.
-
  to the node.
  - When we get to the goal, we have the plan.
- Then we build a waypoint from each grid cell.

  - Typically the centre of gravity of the cell.

KING'S College LONDON

- Also known as Grassfire, wildfire or NF1.
- Essentially breadth-first search in a convenient form for application to grid-based maps:

n and

- Then read the sequence of cells to traverse by f
  labels down from the start, choosing the low
  label at each step.

- Works especially well with occupancy grids, where the obstacles are already factored into the map.

- Here's an example:

- Bug algorithms assume localization but no map.
-
- Leave the obstacle at the point closest to the go
- Circle the obstacle to be sure that you know wh
is.

- The second bug algorithm improves on the performance of

- Follow the obstacle always on the left or right s
- Leave the obstacle if you cross the direct (line o connection between start and goal.

- Works even on very complex obstacles

Potential field

- Robot is treated as a point under the influence of an artificial potential field.
- The goal attracts it and obstacles repel it.
-

- Generated robot movement is similar to a ball rolling down the hill
- Lots of possibilities to get stuck in local minima.

- The idea is that potential energy is stored in the
- The robot wants to minimise its potential en
- So it moves down the potential energy gradient.
- Goals "attract" potential energy.
- Obstacles "repel" potential energy.

- Approach that uses sensor readings to tell the robot how to avoid obstacles.
- Representing the area around the robot as a grid, compute the probability that any square has an obstacle.
- Provides a local map to decide how the robot should move.

- The local map is reduced to a 1 DOF histogram:

- Then compute the steering angle for the best
- Best selected using function G wh

$$G = a.\ \texttt{target-direction} + b.\ \texttt{wheel-orientation} \\ + c.\ \texttt{previous-direction}$$

- An issue with VFH is that it does not account for how the robot can really move.

- The best gap could be one that the robot has to stop and do some complex maneuver to go through.

- VFH+ considers motion on trajectories.
- Any turn that has a trajectory that intersects an obstacle is blocked

- VFH in action.

-
-
-
- Dynamics of the robot not really considered

Curvature velocity methods

- Go further than VFH+ in modelling the motion of the robot.

- Transform obstacles into the velocity space of the robot.
- Apply acceleration constraints to determine possible velocities.

Today

Navi

Cog

Behaviour Based Systems

Multi-Robot Systems

-
-
-
- Knowledge Representation

Environment

Robot environments are characterized by various properties:

- accessible vs inaccessible
  - In an accessible environment, a robot has access to all the necessary information required to make an informed decision about its actions.

-

-
  - In an episodic environment, activity pro repeated episodes.

- static vs dynamic
  - In a static environment, things change only due to actions effected by the robot.
- discrete vs continuous
  - In a discrete enviroment, sensor readings and actions have a distinct, separable values.

KING'S College LONDON

- A robot's state refers to knowledge about itself and its environment.
  - Kinematics is the study of the correspondence between actuator mechanisms and the resulting motion, which can be either:

    (e.g., the result of linear motion)
    - Did I extend my arm as far as I think i did?
    (e.g., the result of rotary motion)

- A robot's environment is full of information. I… determine what is relevant to represent, given the robot's abilities and task.
  - What properties can be sensed?
  - How can the sensed information be stored in a useable and useful way?

KING'S College LONDON

Like with humans, a robot's memory is divided into 2

-
- Long Term Memory

- STM is transitory.
-
-
-
  - avoid-past: avoid recently visited places to encourage exploration of novel areas
  - wall-memory: store past sensor readin of wall detection

- LTM is persistent.
- Metric maps use absolute measurements and coordinate
-
- For example:
  - Markov models are graph representati
    augmented with probabilities for each a
    each sensed state.

- State — A robot's state can (which comprises knowledge about the robot and its environment) can be totally observable, partially observable or unobservable. States can be

- the navigable surroundings in a robot's envi structure. Spatial information is typically topological map.

- Objects — Objects are categories and/or i detectable things in the robot's environment.

KING'S College LONDON

- Actions — Actions that a robot can perform are part of its knowledge representation. This includes the expected outcomes of specific actions on the robot and on its

-

  include information about perceptions (h
  behaviours (how to act).

- Intentional — A robot's intentions are com
  its plans to achieve those goals and its intende
  make up the plans.

- Maps
  - Euclidean map — Represents each point in space according to

  - Cognitive map — Represents behavio
    previous experience and use for action. U
    forage and home (animal navigation).
    collections of vectors.

KING'S College LONDON

- Graphs —
  - Represent the relationship between states (nodes in the graph) and actions (links between nodes). The links indicate how to move from one state to another, based on an action.

-
  - Associate probabilities with states and
  - In a graph, actions are given probabilities experimentally, by observation) indi an action taken in one state will lead to anot thus possible to have a single action taken in one state to lead to multiple possible subsequent states, each with a different probability.

Today

Navi

Cog

Behaviour-Based Systems

Multi-Robot Systems

KING'S College LONDON

- Control models
-
-
- Behaviour coordination
- Emergent behaviour

Control models

There is distinction between:

- Classic model-based control
- Focusses on symbolic representations
-

-
- Focuses on numeric representations
- Based on 'nouvelle AI' – e.g., [Brooks, 1986]

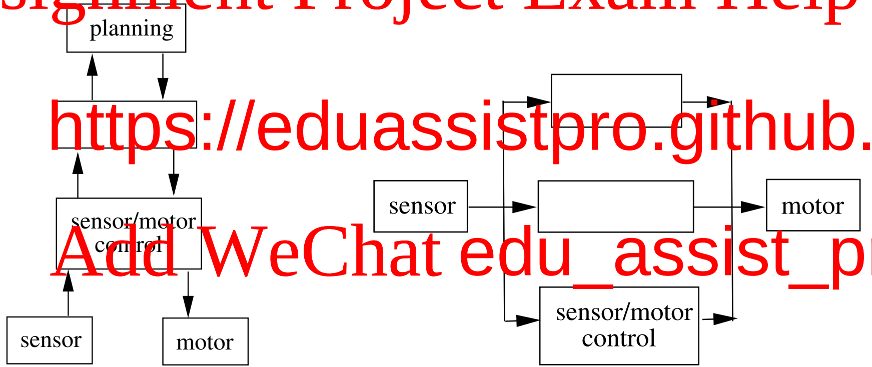
- There are also hybrid models that combine aspects of both model-based and behaviour-based.

Control models, p2

- Deliberative models: Sense, Plan, Act
  - Provide a functional decomposition

- Reactive models
  - Provide a task-oriented decompositi
  - Systems consist of concurrently execu specific tasks, such as: “Avoid obstacle”, “

- Two orthogonal control flows:



vertical horizontal

- They are characterized by behavioural decomposition, rather than a functional or a task-oriented decomposition.

- Systems consist of sequential modules achieving independent

-

  given perceptual stimulus.
  - The basis for these systems is in biological s result biology is an inspiration for the desi behaviour-based models.

  - The field of Artificial Life or ALife focuses o of computational models of natural phenomena, including behaviour of individual and groups of animal(s).

KING'S College LONDON

- A behaviour is anything observable that the system/robot does
- How do we distinguish internal behaviours (components of a
-
- organized in layers (e.g., subsumption arc
- Behaviour-based models actually consi programmed in behaviours, which have hi rules, extended in time (as opposed to rules, which are typically short-term), and capable of using and maintaining sophisticated knowledge representations.

-
- We make a comparison here.

A behaviour . . .

- is based on dynamic processes

- exploits emergence
  - which are side-effects from combining p
  - often use properties of the environment

- tends to be reactive.

An action . . .

- is discrete in time,
  - with well defined start and end points, and

-
  - related) steps, and
  - because it also avoids conflicts.

- tends to be deliberative.

- Actions are building blocks for behaviours.

Distinguishing behaviour-based systems

Characteristics:

- Achieve specific tasks/goals (e.g., "Avoid Others", "Find Friend")
-
-
-
- Can take inputs from other behaviours and s
  other behaviours (e.g., can be connected in b
  networks)

- Typically higher-level than actions (e.g., a behaviour might be "Go Home", whereas an action would be "Turn left 45 degrees")
- Typically closed loop, but extended in time

-
-
- Ability to use a uniform structure and repres throughout

Challenges:

- How can a representation be effectively distributed over the behaviour structure? The time scale must be similar to that of

- Some components may be reactive.
- Not every component is involved with repre computation.

- Some systems use a simple representation
- As long as the basis is in behaviours and not rules, then the system is a behaviour-based system.

- Behaviours can be expressed using various representations.
-
-
  - Functional notation
  - FSA (Finite State Automata) diagram
  - Subsumption Architecture
  - Stimulus Response (SR) formalism

- Strengths and weaknesses of various behavioural encodings:

Strengths:

-
-
-

- Support for modularity

Weaknesses:

- Niche targetability
- Hardware retargetability
- Combination pitfalls (local minima, oscillations)

- Mathematical model:
  - represented as triples (S, R, β)

    S = stimulus

    R = range of response

    β = behavioural mapping between S and R

-
-

```
coordinate-behaviours [
   move-to-classroom ( detect-c
   avoid-objects ( detect-objec
   dodge-students ( detect-stud
   stay-to-right-on-path ( detect-path ),
   defer-to-elders ( detect-elders )
] = motor-response
```


KING'S College LONDON

- Finite State Automata can be used to show sequences of behaviour transitions, where states represent behaviours.
-
- tain (maint) the goals.
  - (ach in-classroom)
  - (maint avoid-objects)

- Once defined, tasks can be compiled into circ reactive.

- For example:


```
(maint dodge-
(maint stay-t
(maint defer-
```

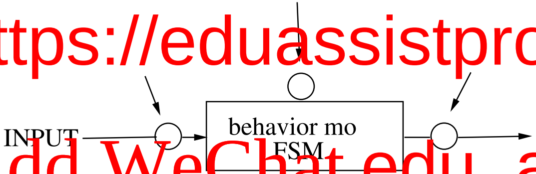
- Subsumption Architecture [Brooks, 1986]

- Behaviour-based elements
- Layered approach based on levels of compe
- Uses an augmented Finite State Machine (
  behaviour

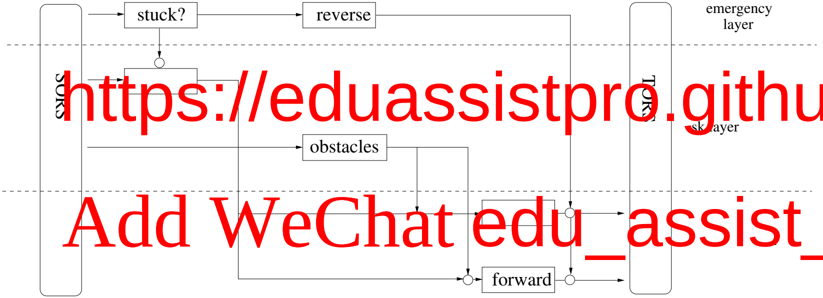
KING'S College LONDON

Subsumption architecture, p2

- Each comoponent:



- Combines FSMs in a sort of layered, circuit diagram:



- The Stimulus-Response formalism is based on the premise that a behavioural response in physical space has a strength and an orientation.
-

  - R = response
  - β = behavioural mapping categories

  discrete, or continuous

- Mapping can either be:
  - Discrete
  - Continuous

Discrete Mapping

- Expressed as a finite set of situation-response pairs/mappings
- Mappings often include rule-based (IF-THEN) formulae.
- Examples:

- Instead of discretizing the input and output mathematical function describes the inp
- Can be simple, time-varying, harmonic.
- Examples:
  - Potential fields
- However, here are problems with local minima, maxima, oscillatory behaviour.

-
-
- cooperative, or a combination of the two.

-
  - state-based: discrete event systems
  - function-based: spreading of activati

Cooperative coordination:

-
  - fuzzy (formalized voting)
  - superposition (linear combinations)
  - schemas, dynamical systems

Assignment Project Exam Help

- https://eduassistpro.github.i

Add WeChat edu_assist_pr

- **Emergence** is an important but not well-understood

-

  Coded behaviour is in the programming scheme.

  Observed behaviour is in the eyes of the obser

  There is no one-to-one mapping between t

-

  the behaviour is not explicitly specified a

- Unexpected vs Emergent
  - Some researchers say the above is not enough for behaviour to

    “unexpected” is highly subjective; it dep
    observer was expecting. Naïve observe
    Informed observers are rarely surprise

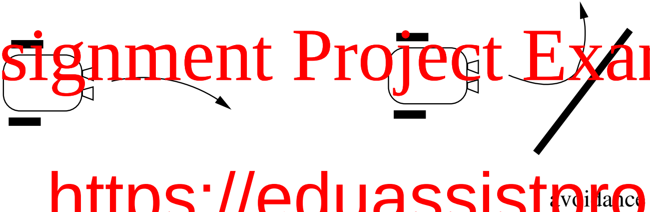
  - Once a behaviour is observed, it is no longer
    Is new behaviour then “predictable”?

Wall following behaviour can be implemented with these rules:

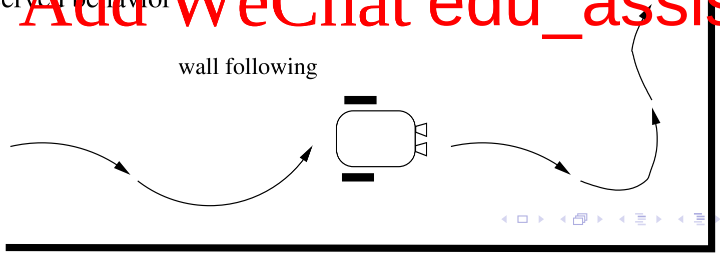
-
-
-

- Over time, in an environment with walls, this wall-following.

coded behavior



observed behavior



- Is this emergent behaviour?
-
  - The concepts of "wall" and "following" are not stored in the robot's controller.
  - The system is just a collection of simple rule

- Today
- Navi
- Cog
- Behaviour Based Systems
- Multi-Robot Systems

-
-
-
- Planning
- Coordination

-

  robots and employ biologically inspire

-
  - a human-robot team can exchange sear appropriately

-
-
  "map" of the area/resources/features

-
-
  - moving in formation where the environ to maintaining formation

- Strategies:
  - Formation: structuring relative positions of multiple robots
  - Caging or Object Closure:
    - instead of single-robot idea of caging, where a multi-fingered

      sensing and is based around vector fields
    - comparing experimental data gat
      only (omnidirectional) cameras to
      overhead camera

  - Coverage:
    - surveilling pursuer, while continuing to keep an eye on an evader examining every point on the boundary of a 2D space
    - swarm-based approaches

- Challenges:
  - Maintaining communication (wifi/radio connectivity) during exploration
  - Information fusion fusing data from multiple robots to give position estimates of objects of interest

  - deciding how to search a space with a group will find an evading target to result in "guar
  - analyzing range of auction bidding rules f robots bid for targets; robots have to visit al "win", and the team is judged by combinations of the distances travelled by all robots visiting their targets (e.g., traveling salesman problem)
  - implementing foraging using robot swarms

- Multiple robots jointly estimate each other's positions
- A group of robots can localize faster (than they can on their own) if they share information.
-
  - another, it updates its location with information that takes into account the belief the other robot has o
  - showing that it is possible to use wireless et particular the strength of wireless ether basis for localization
  - planning the trajectories of multiple robots in order to improve their localization performance

- Multi-robot Simultaneous Localization And Mapping (SLAM)
  - additional information from several robots can speed up

  - applying particle filters to multi-robot S
  - providing a strategy for accurate contro
    (i.e., keeping on the desired path while loc
  - considering performance bounds

- Aspects of planning in multi-robot systems:
  - path and motion planning
  - task planning
- Challenges in multi-robot planning:

  - task planning for multi-robot systems u
- Strategies for multi-robot path planning:
  - multi-robot path planning using proba
  - planning in terms of roles, allowing robot through "exchange" as situations arise
  - shared memory task scheduling for a heterogenous multi-robot team making use of a "shared global unit" to reduce communication overhead

Multi-robot **Coordination**

- Standard tasks:
  - Traffic control
  - Box-pushing

- is an intersection of several robots' plans—they aren't working together, but they are in the same space at th

  - Using parallel stochastic hill-climbin team of robots in a pursuer-evader task
  - Dynamically coordinating robots using task assignment and integrates data on obstacles
  - Coordinating movement of objects by homogeneous teams of robots

Task Allocation: dividing responsibilities in a group of

-
  - abilities of individual robots are conditioned on changing locations
  - different robots have different capabili

- Approaches:
  - providing a local approach to task allocation in a multi-robot team; robots only use information from immediate neighbors
  - performing concurrent mapping and localization and showing improvements with multi-robot data collection
  - distributed
  - learning using Alliance which aims to ha teams of robots in a variety of scenarios
  - learning specialities of robots using rein using a blackboard to distribute knowledge sharing
  - Contrasting task allocation in robot swarms using random assignment with more measured approaches that use different amounts of bandwidth and time to run

Many apply market-based approaches to multi-robot coordination and task allocation, such as **auctions**

-
-

ensure that robots are not idle when several ro
the same unexplored area.

- tasks are offered "for sale" to robot team me
- robots indicate how much they are willin
- tasks are allocated based on bids for them

- Approaches:

  movement of robots to certain points wit
  - focusing on bid evaluation in market-ba mechanism

References

- Some content from:

  http://www.cs.cmu.edu/~moti

- Many slides thanks to Prof Simon Parsons.