Robotic Behaviors

Potential field techniques
- trajectory generation
- closed feedback-loop control

Design of variety of behaviors
- motivated by potential field based approach – steering behaviors

Closed feedback loop systems
- no memory
- behaviors no representation of the world
- world is implicitly encoded in the potential function

Motion planning (later)
- Representation of the environment
- Different choices
- Path planning algorithms

Potential Field Methods

• Idea robot is a particle
• Environment is represented as a potential field (locally)
• Advantage – capability to generate on-line collision avoidance

Compute force acting on a robot – incremental path planning

\[ F(q) = -\nabla U(q) \]

Example: Robot can translate freely, we can control independently
Environment represented by a potential function

\[ U(x, y) \]

Force is proportional to the gradient of the potential function

\[
\begin{bmatrix}
\dot{x} \\
\dot{y}
\end{bmatrix} = -\nabla U(x, y)
\]

Jana Kosecka, GMU
**Attractive potential field**
- Linear function of distance
  \[ U_a(q) = \xi \lVert q - q_{goal} \rVert \quad F_a(q) = -\nabla U_a(q) = -\xi \frac{(q - q_{goal})}{\lVert q - q_{goal} \rVert} \]
- Quadratic function of distance
  \[ U_a(q) = \xi \frac{1}{2} \lVert q - q_{goal} \rVert^2 \quad F_a(q) = -\nabla U_a(q) = -\xi (q - q_{goal}) \]

Combination of two - far away use linear, closer by use parabolic well

---

**Repulsive potential field**
\[ U_r(q) = \frac{1}{2} \nu \left( \frac{1}{\rho(q, q_{obst})} - \frac{1}{\rho_c} \right) \quad \text{if} \quad \rho(q, q_{obst}) \leq \rho_c \]
\[ \text{else} \quad U_r(q) = 0 \]

Minimal distance between the robot and the obstacle

**Resulting force**
\[ F(q) = -\nabla (U_a(q) + U_r(q)) \]

**Iterative gradient descent planning**
\[ q_{i+1} = q_i + \delta_i \frac{F(q)}{\lVert F(q) \rVert} \]

Issues - multiple obstacles - nonconvex obstacles - how to compute distance
Can be computed for polygonal and polyhedral obstacles

Issues - local minima
Heuristics for escaping the local minima
Can be used in local and global context
Numerical techniques
Random walk methods

---

Jana Kosecka, GMU
How to do the right thing?

- How to do the right thing?
- Go from A to B in the context of navigation
- Taking the environment into account
- Taking the uncertainty into account

- Using the model of the environment or not?

Perception, Model, Plan, Execute (Motion Controller)

Behavior Based Navigation

communicate data
discover new area
detect goal position
avoid obstacles
follow right / left wall

coordination / fusion
e.g. fusion via vector summation

sensors

actuators
Model Based Navigation

- no memory – no look-ahead reacts to the current environmental stimuli/ sensory information
- reactive behaviors:
  Feedback controllers are instances of reactive controllers
  (mappings between situations and actions
  mapping between state and control input)

Can we achieve bigger functionality if we combine them?

Simplest scenario one situation one action:
Motivation - biology, V. Braitenberg’s Vehicles
one can design simple continuous feedback strategies
or sets of if-then rule state rules.
Subsumption Architecture

Guidelines:
- Build the system from bottom up
- Components are task achieving behaviors
- Components are executed in parallel
- Components are organized in layers
- Lowest layers handle most basic tasks
- Higher levels exploit the lower levels
- Each component has its tight connection between Perception and action

Bottom up design process

Subsumption architecture

seek cans

wander

avoid

Upper levels -> explore, build maps, reason about world
In terms of identifiable objects, formulate and execute plans
Subsumption architecture

No model of the world
# of tight feedback loops - reactivity, robustness
inflexibility at run-time, needs expertise for the design

Model of the behavior is Augmented Finite State Machine
AFSM’s connected with communication wires
pass input and output messages

Results in fixed based priority arbitration

Coupling between the layers can be done through the world:
e.g. HERBERT soda can searching robot (Jon Connell, MIT)
1. If you see soda can grab it
2. If its heavy put it down
3. If its empty pick it up
... no notion of the model of the world or connections between
behaviors

Subsumption architecture claims:

World is is best model
“Intelligence” without representation

All the relevant parts of the systems interact with the each other
through sensing and the world

Many instances of robots build using this philosophy.

Strengths: reactivity, parallelism, incremental design (robustness)
Weakness: needed expertise at the design and inflexibility at
the run time
Issues of representation

- Before we discussed the model of the robot
  kinematics, dynamics characterize the motion of the robot
  and the way it interacts with the environment
  (lower part of the previous diagram)
  feedback-control - no memory, stimulus-action pairs
determined the next step

- Model of the environment where the robot resides
- Map of the environment (static/dynamic)
- Representation of the environment is the distinguishing feature
  of the robot architecture (we discuss different choices later)

Motivational examples

Motivation
Valentino Braitenberg: Vehicles
>> vehicles with different personalities
Walter Grey: Tortoise
analog implementation, one sensor per one effector
>> light seeking behavior
Behavior-Based Architecture

We had previously examples of behaviors

Feedback controllers, task-achieving behaviors
- design motivated by potential field based technique

How to composed them? Some examples of composition
- superposition (motivated by potential field techniques)

Behaviors

1. Behaviors are feedback controllers
2. Behaviors are executed in parallel
3. Achieve specific goals (avoid-obstacles, go-to-goal)
4. Can be combined to achieve more complex networks (make inputs of one behavior, outputs of another)
5. Behaviors can be designed to look-ahead, build and maintain representation of the world
Representation of behaviors

- Behavior is mapping from state to control command

Different representations
- Feedback controllers - gradient of some potential function
- Lookup table
- Stimulus/response diagrams
- Discrete and/or continuous representations
  (differential equations or if-then rules -> wall-following example)

Behavior Representations

Continuous representation
- Potential field techniques (attractive, repulsive potential fields)
- Schema (more general approach to vector fields design)
  (e.g. goto goal, follow corridor) - Arbib 81

issues with superposition - local minima, maxima, oscillatory behavior
Superposition of different behaviors

Behavior-based Architecture

Motivation
1. To keep all the advantages of the Reactive Control
2. Allow representation of the environment
3. Allow bigger flexibility and reconfiguration depending on the task

(... this is what subsumption architecture was lacking)
Behavior Assemblages

- Power of abstraction
- Modularity
- Reuse of elementary behaviors
- “reason” over them
- Coarser level of granularity - good for adaptation and learning

Abstraction’s in terms of FSM’s

Composition of the behaviors

Examples: FSM for navigation
Pole finding robot (AAA competition)

Wander and avoid behavior assemblage
Different behaviors/motor schemas

- Move-ahead
- Move-to-goal
- Avoid-obstacle
- Dodge – sidestep an approaching ballistic projectile
- Escape – move away from projected intercept point between robot and approaching predator
- Stay-on-path
- Noise
- Follow the leader
- Probe – move towards open areas
- Dock- approach an object from particular direction
- Avoid-past – move away from recently visited areas
- Move-up, down maintain altitude

Foraging

| Action          | Motor Schema                |
|-----------------|-----------------------------|
| Wander          | noise                       |
| Detect obst     | Avoid static obstacle       |
| Detect robot    | Avoid static obstacle       |
| Detect attractor | Move to goal                |
| Detect obst     | Avoid static obst.          |
| Detect robot    | Avoid static obst.          |
| Acquire         | noise                       |
| Detect homebase | Move to goal                |
| Detect obst     | Avoid static obstacle       |
| Detect robot    | Avoid static obstacle       |
| Deliver         | noise                       |
Example of trash collecting robot - each node is an assemblage of behaviors - more details on the transitions

**Behavior Composition**

Programming language for behavior composition

Elementary behaviors FSM’s

Composition operators:
1. Sequential  $B_1; B_2$
2. Conditional  $B_1 : B_2$
3. Parallel $B_1 || B_2$
4. Disabling $B_1 # B_2$
5. Iterative  $B_1:: B_2$, $B_1 :: B_2$

Examples of more complex tasks as networks of elementary behaviors (behaviors can communicate via shared memory)

Example: Classroom navigation, Clean Up, Foraging
Emergent Behaviors

Apparently new behaviors can “emerge” from

Interactions of rules
Interactions of behaviors
Interactions with the environment

• Since the behavior is just input output mapping externally observed
• Occasionally explicitly un-modeled interactions/behaviors can be observed
• Notion of emergent behavior – intuitive, not well defined except for some simple scenarios (wall following example, flocking, dispersing, foraging)

Emergent behaviors

Flocking example
1. Don’t run into another robot
2. Don’t get too far from other robots
3. Keep moving

Unexpected vs. emergent
• depends on the observer - subjective notion

Due to the un-modeled uncertainties, the behaviors are not exactly repeatable/predictable
Emergent behaviors can be achieved from parallel execution of many behaviors
For the purpose of analysis – undesirable phenomena
Streering behaviors

http://www.red3d.com/cwr/steer/

Behavior Composition

So far all the examples of behaviors were just reactive controllers represented in continuous or discrete manner

How to design more sophisticated behaviors?
E.g. to achieve map building of the environment

- Map building in the context of Behavior-Based Architecture
- What type of map representation would fit?
- Representation needs to satisfy the premises of the behavior-based architecture

Example: Toto the map building robot
M. Mataric, Learning a distributed map representation of the environment based on navigation behaviors, 1992
Example – Multi-robot Collision Avoidance

Air traffic Control – Collision Avoidance

• Distributed multi-agent motion planning approach
• Potential and Vortex field based motion planning
• Generation of the prototype maneuvers
• 2D planar conflict resolution
• 2-1/2D conflict resolution

Collision Avoidance – Vector field based approach

• Superposition of participating vector fields

Overtake maneuver
Collision Avoidance – Prototype Maneuvers

Roundabout maneuver

2-1/2 D avoidance maneuver, horizontal and vertical conflict resolution
Collision Avoidance – Visualization

Collision Avoidance – Visualization
Design and modeling of a hierarchical hybrid systems:
Safety guarantees, Safety vs performance tradeoff