7630 – Autonomous Robotics
Robotic Behaviors and Simple Control

Cédric Pradalier

Today

Outline

Introduction

Objectives

- Examples of Behavior-Based systems
- Examples of Simple Control Laws
- Understanding of the advantages and limitations

Topics

- Braitenberg vehicles
- Point Stabilisation
- Pose Stabilisation
- Line Following

Outline

What is behavioral robotics?

Concept

Defining a behavior is a way to define how a robotic system will react to a specific set of stimuli, i.e. sensor readings or situations.

Questions

- What is the most appropriate way to define and describe such a behavior?
- How to optimise such a behavior?
- How to combine behaviors, i.e. how to build an architecture?

Advantage and inconvenients

Advantage

- No need for models (kinematic, dynamic)
- No need for complex localisation, mapping, environment modelling, control, ...
- Can be used to explain/interprete/model animal behavior.

Inconvenients

- No standard representation
- Not always possible to give performance guarantees
- Not necessarily very portable

JPL: Tooth and Rocky, '91

Video: `videos/vid1016-84-video6.avi`

Link with animal behaviors

Video: `videos/cockroaches.wmv`

Cockroach guide to survival

- Light goes on, the cockroach turns and runs
- When it gets to a wall, it follows it
- When it finds a hiding place, goes in and faces outward
- Wait until not scared, then comes out, even if the lights are turned back off earlier.

Braitenberg vehicles

History

Neuro-biologist Valentino Braitenberg, *Vehicles: Experiments into Synthetic Psychology* (1984). "How sentient creatures might have evolved from simpler organisms".

Concept

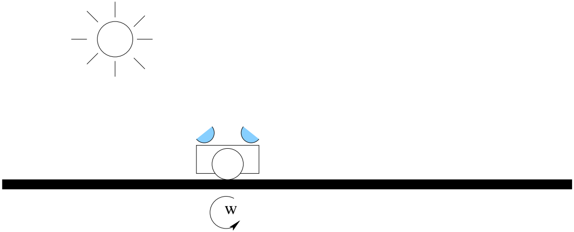
A robot is a system with a set of sensors with analog output and motors with analog input (velocity). A behaviour is defined by the connection of the sensor output to the motors, eventually through operators.

Example

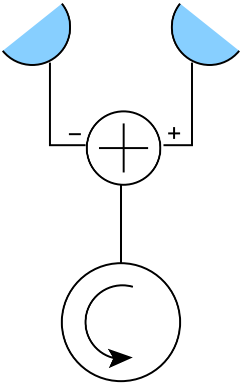
Setup

- Robot moving on a rail with a single motor
- Two light sensors looking left and right, analog output
- One light source, possibly moving

Objective: track the light source



Implementation



Application

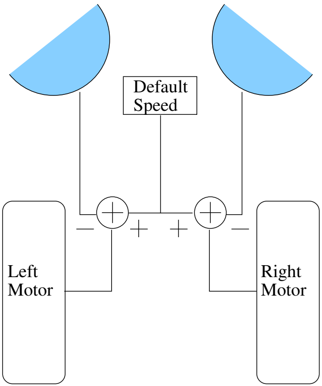
Setup

- Bidirectional robot (same as bubblebot)
- Some analog sensors (range or light)

Question

- Setup a system seeking for light
- Setup a system for obstacle avoidance and exploration
- Setup a system for wall following
- Setup a system for platooning with or without a light beacon on the front vehicle.

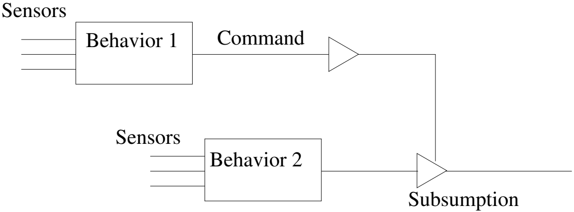
Light seeker



With some implicit scaling where appropriate.

Combining behaviours – Architectures

Example: Priority / Subsumption



Behavior 2 is active by default but will be replaced by Behavior 1 if the latter is active.

- Works when there is no sequentiality (but sequences can be encoded in the arbitration mechanism).
- Complexity increases when the number of possible behavior increases.

Summary

Advantages of Behavior-Based Robotics

- Very little computing power required
- Intuitive implementation of simple behaviors
- Emergent behaviors out of simple behaviors

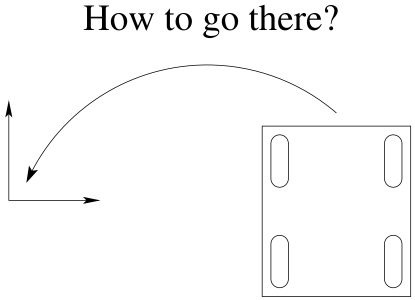
Challenges

- Often difficult to get performance guarantees
- Does not necessarily scale very well
- Hard to reuse

Ultimately, tools like classical control or sensor-based control are more often used outside of the AI community.

Outline

Point stabilisation



Objectives and constraints

- Desired position: (X^*, Y^*) or (X^*, Y^*, θ^*).
- Wheel constraints (kinematics) prevent decoupled control.

First approach with infinite curvature

Principle

1. Turn towards target:

$$\omega = k_\omega \left[\mathrm{atan2}(Y^* - Y, X^* - X) - \theta\right]$$

2. Go to the target:

$$v = \min(k_v \cdot \mathrm{hypot}(Y^* - Y, X^* - X), v_{\max})$$

3. Eventually align with desired angle:

$$\omega = k_\omega \left[\theta^* - \theta\right]$$

Step 2 together with step 1 (σ is another scaling gain):

$$v = \min(k_v \cdot \mathrm{hypot}(Y^* - Y, X^* - X), v_{\max}) \times e^{-\frac{\omega^2}{\sigma^2}}$$

First approach with infinite curvature

Remarks

- Works well with differential robots.
- Requires a state machine to decide when to align (step 3).

Questions

- Let's assume we work with a blimp. What is the consequence of wind? How to modify the control to deal with it?
- Draw the path of car trying to apply step 1 and 2 in a worst case situation.

Outline

Pose Stabilisation (External)

Source:

- Autonomous Mobile Robots, Siegwart et al.
- http://www.asl.ethz.ch/education/master/mobile_robotics/year2007/S_3b_-_Motion_Control_Wheel.pdf

Questions:

- Does this solve the problem of curvature constraints?
- What will be the effect of constant perturbation (e.g. wind)?

Outline

Paths and Trajectories

Definition: Path

Sequence of position/configuration that the robot must achieve, in general continuous but not necessarily indexed by time. Example, spline $X(u)$, with $u \in [0, 1]$

Definition: Trajectory

Path indexed by time. Example $X(t)$, $t \in [0, T]$. If X is a trajectory, $\dot{X}$ is the velocity vector of the system.

Trajectory following

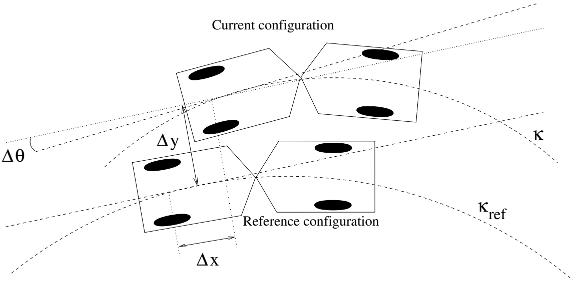
Context

- Desired trajectory $X^*(t)$. Here $X^*(t) = (x^*(t), y^*(t), \theta^*(t))$.
- Current state $X(t)$, here $X(t) = (x(t), y(t), \theta(t))$.

Constraints

- A good trajectory planner will guarantee that there exist a control function $u(t)$ which applied to the model of the vehicle, will make it realise $X^*(t)$.
- The vehicle has constraints: maximum velocity, maximum steering, maximum curvature, maximum acceleration...

Trajectory following



- Simplest solution (with $\Delta a = a^*(t) - a(t)$):

$$v = k_x \cdot \Delta x$$

$$\omega = k_y \Delta y + k_\theta \Delta\theta \ \ (+k_\kappa \Delta\kappa)$$

P. Ridley and P. Corke. Load haul dump vehicle kinematics and control. Journal of Dyn. Sys, Measurement and Control, 2003.

Trajectory/Path following: Samson et al.'91

For a path

- Reference point X^* is the closest point on the path
- $\Delta x = 0$
- $v = cste$

$$\omega = \omega_{ref} + k_y v_{ref} \frac{\sin \Delta\theta}{\Delta\theta} \Delta y + k_\theta \Delta\theta$$

For a trajectory

- Samson C. and Ait-Abderrahim K. Feedback control of a nonholonomic wheeled cart in cartesian space. ICRA'91.
- Stumm E. et al. Tensor Voting Based Navigation for Robotic Inspection of 3D Surfaces Using Lidar Point Clouds, IJRR'12, pp 13-14

Trajectory/Path following: Samson et al.'91

Questions

- What is the purpose of ω_{ref} or v_{ref}? Why are they called feed-forward terms?
- How would you tune such a controller?

Outline

Conclusion

Reactive Behaviour

- Generic term for any control with no memory, no planning, no environment model.
- Cheap and efficient, when possible.

Non-holonomic control

- Lot of ad-hoc tools (and only a subset presented).
- Point stabilisation and trajectory following.
- Control seen as a reactive behaviour