Robotics, Vison and Control [Chap. 1]

Fundamental Algorithms in MATLAB

Introduction

The study of mobile robots is an intrinsically interdisciplinary research area

- **Mechanical engineering**
  : vehicle design and in particular locomotive mechanisms
- **Computer science**
  : representations and sensing and planning algorithms
- **Electrical engineering**
  : system integration, sensors, and communications
- **Cognitive psychology, perception, and neuroscience**
  : insights into how biological organisms solve similar problems
- **Mechatronics**
  : the combination of mechanical engineering with computer science, computer engineering, and/or electrical engineering

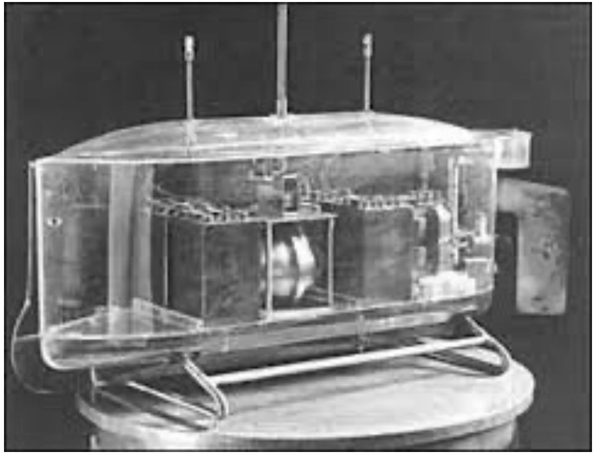
Introduction

Mobile robot well suited for tasks that exhibit one or more of the following characteristics:

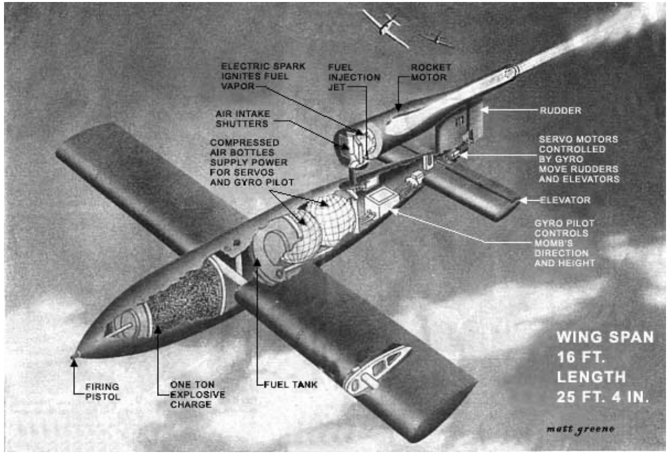
- An environment that is inhospitable, so that sending a human being is either very costly or very dangerous
- An environment that is remote, so that sending a human operator is too difficult or takes too long. Extreme instances are domains that are completely inaccessible to humans, such as microscopic environments
- A task with a very demanding duty cycle or a very high fatigue factor
- A task that is highly disagreeable to a human

경희대학교 Kyung Hee University

Early Autonomous Robots

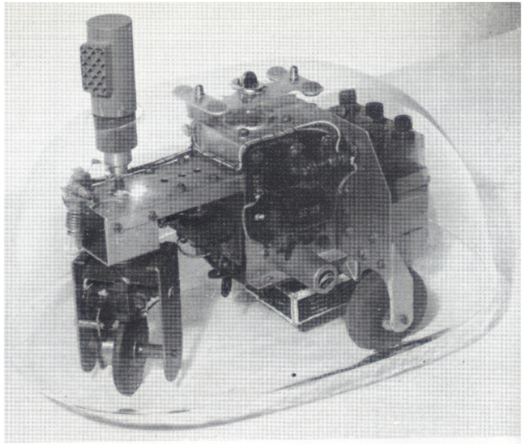


- Tesla's robot

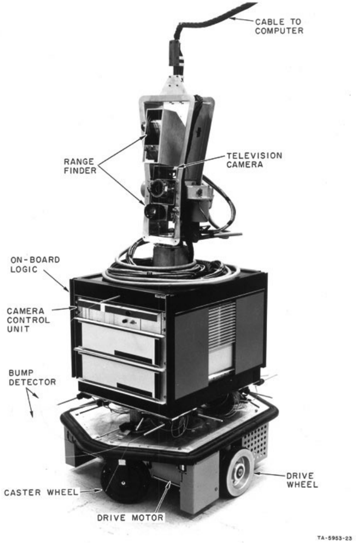


- V1 flying bomb

Early Autonomous Robots

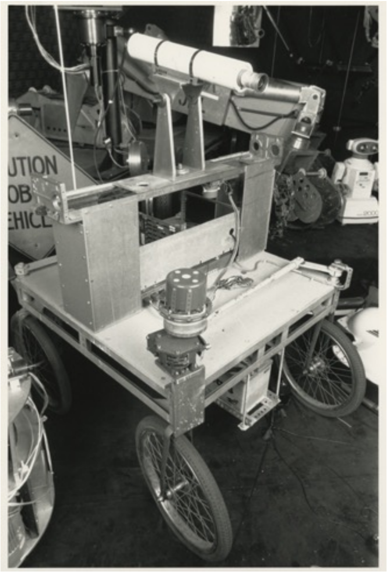


- Walter's robot



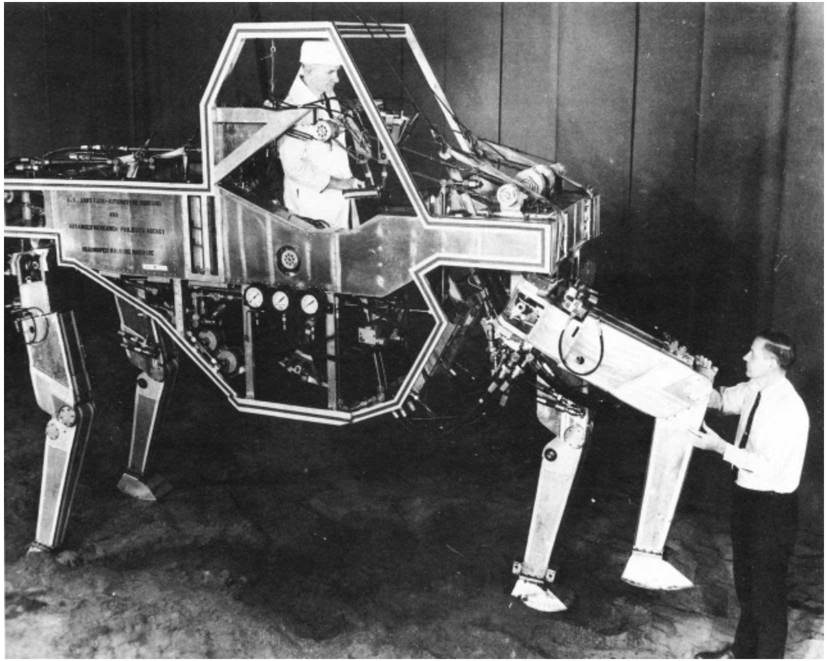
- Shakey

Early Autonomous Robots



- CMU Rover
- Stanford Cart

Early Autonomous Robots



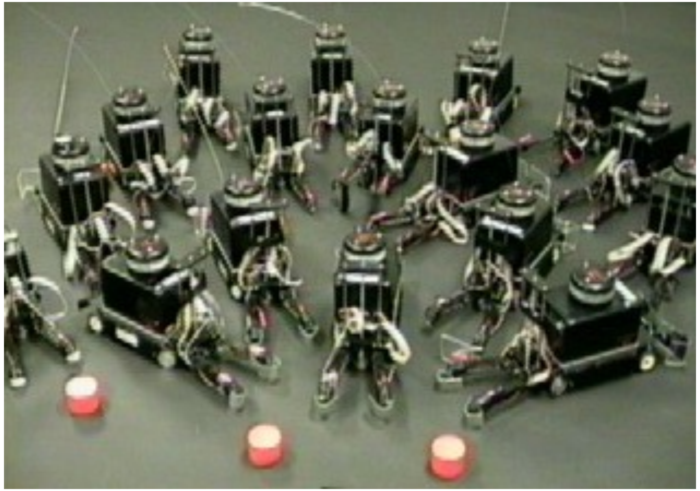
- The GE Quadruped

Biological Inspiration

- Inspirations from biological systems may influence
  - physical design
  - sensory mechanisms
  - underlying model of computational control

- Simple biological systems are capable of
  - locomotion
  - sensing
  - reasoning

Operational Modes



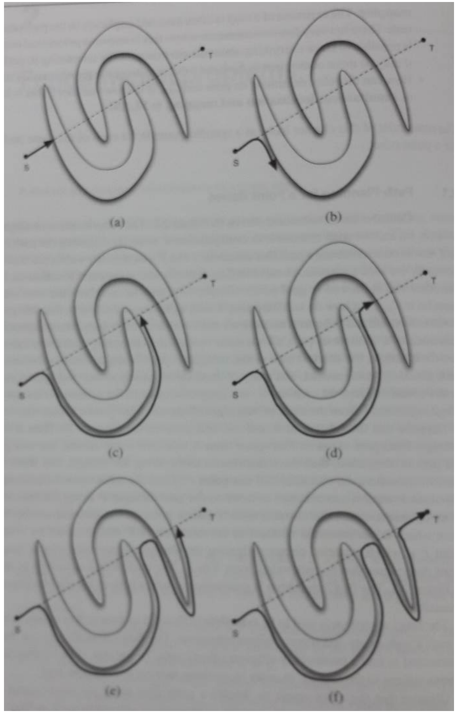
- Swarm of robots

Fundamental problems

Introduction

- **Path planning**
  : Is it possible to get the robot from one configuration to another while remaining within C_{free}?
- **Localization**
  : How can the robot determine its state if it has local measurements of C_{free}?
- **Sensing**
  : How can the robot determine which parts of its environment are occupied?
- **Mapping**
  : How can the robot determine C_{free}, assuming it always knows where it is?
- **SLAM**
  : How can the robot determine its pose and C_{free} if it knows neither?

Path Planning



```
Visualize a direct path ST from the start S to the goal T.
while the goal T is not archived, do:
    begin
        move towards the goal along the path ST,
        if the path is obstructed then
            begin
                mark the current location as P circumnavigate
                the object until the robot either:
                (a) hits the line ST at a point closer to T than P
                    and can move towards T, in which case the robot
                    follows ST;
                (b) returns to where P in which case T is unreachable.
            end
    end
end
```

- Bug2 algorithm

- Bug algorithm example

Localization

- When the robot moves from (x, y) to some point (x+△x, y+△y) how do we know where the robot actually is?

- After making N motions from some start position (x_0, y_0), where is the robot relative to $(x_N, y_N) = (x_0, y_0) + \sum(\Delta x_i, \triangle y_i)$?

$$\forall_i E[\varepsilon_i^x] = E[\varepsilon_i^y] = 0$$
$$\forall_i E[(\varepsilon_i^x - E[\varepsilon_i^x])(\varepsilon_i^x - E[\varepsilon_i^x])] = \sigma^2$$
$$\forall_i E[(\varepsilon_i^y - E[\varepsilon_i^y])(\varepsilon_i^y - E[\varepsilon_i^y])] = \sigma^2$$
$$\forall_{i \neq j} E[(\varepsilon_i^x - E[\varepsilon_i^x])(\varepsilon_i^y - E[\varepsilon_i^y])] = 0$$

$$\begin{aligned} E[(x_N, y_N)] &= E[(x_0, y_0) + \sum(\Delta x_i + \varepsilon_i^x, \Delta y_i + \varepsilon_i^y)] \\ &= (x_0, y_0) + (E[\sum \Delta x_i + \varepsilon_i^x], E[\sum \Delta x_i + \varepsilon_i^x]) \\ &= (x_0, y_0) + (\sum \Delta x_i + \sum E[\varepsilon_i^x], \sum \Delta x_i + \sum E[\varepsilon_i^x]) \\ &= (x_0, y_0) + (\sum \Delta x_i, \sum \Delta x_i) \end{aligned}$$

$$\Sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{bmatrix}$$

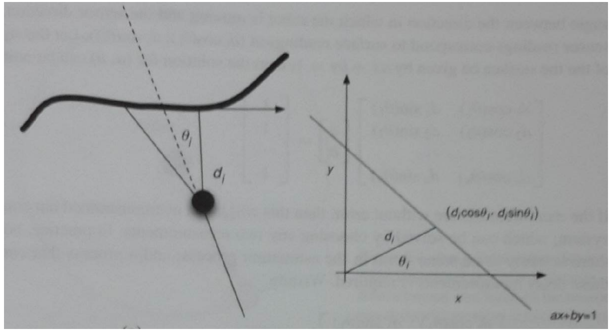
$$\begin{aligned} \sigma_{xx} &= E[(x - E[x])^2] \\ &= E[(x_0 + \sum(\Delta x_i + \varepsilon_i^x) - E[x_0 + \sum(\Delta x_i + \varepsilon_i^x)])^2] \\ &= E[(x_0 + \sum \Delta x_i + \sum \varepsilon_i^x - E[x_0] - \sum E[\Delta x_i] - \sum E[\varepsilon_i^x])^2] \\ &= E[(\sum \varepsilon_i^x - \sum E[\varepsilon_i^x])^2] \\ &= E[\sum(\varepsilon_i^x - E[\varepsilon_i^x])^2] \\ &= \sum E[(\varepsilon_i^x - E[\varepsilon_i^x])^2] + 2\sum_{i<j} \sum E[(\varepsilon_i^x - E[\varepsilon_i^x])(\varepsilon_i^x - E[\varepsilon_i^x])] \end{aligned}$$

$$\sigma_{xy} = \sigma_{yx} = 0$$

$$\Sigma = N \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$$

Sensing

- In order to deal with the realities of navigation and localization, a robot requires some mechanism to sense its environment.



- Laser sensor
- Infrared sensor

$$\begin{bmatrix} d_1\cos(\theta_1) & d_1\sin(\theta_1) \\ d_2\cos(\theta_2) & d_1\sin(\theta_2) \\ \cdots & \cdots \\ d_n\cos(\theta_n) & d_1\sin(\theta_n) \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \cdots \\ 1 \end{bmatrix}$$

$$\mathrm{A} = \begin{bmatrix} d_1\cos(\theta_1) & d_1\sin(\theta_1) \\ d_2\cos(\theta_2) & d_1\sin(\theta_2) \\ \cdots & \cdots \\ d_n\cos(\theta_n) & d_1\sin(\theta_n) \end{bmatrix}$$

$$\mathrm{X} = \begin{bmatrix} a \\ b \end{bmatrix}$$

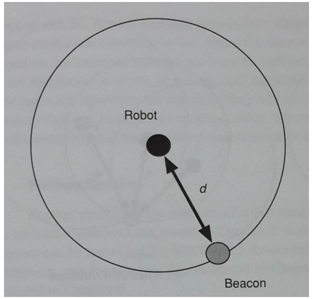
$$\mathrm{B} = \begin{bmatrix} 1 \\ 1 \\ \cdots \\ 1 \end{bmatrix}$$

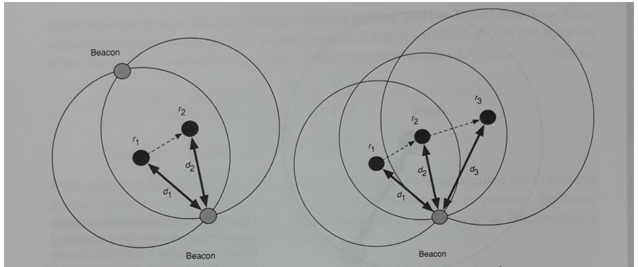
$$\mathrm{AX} = \mathrm{B}$$

경희대학교 Kyung Hee University

Mapping and SLAM

- How can a robot construct a map of the locations of beacons?





(a) After one motion (b) After two motions

- In order to deal with both localization and mapping information contained in the sensor, readings must also be used to continually refine the map and the estimate of the robot's position.

- Visible beacons with known location

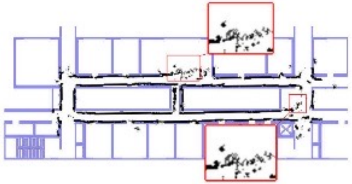
(x_i^b, y_i^b)

- Measured distance

d_i

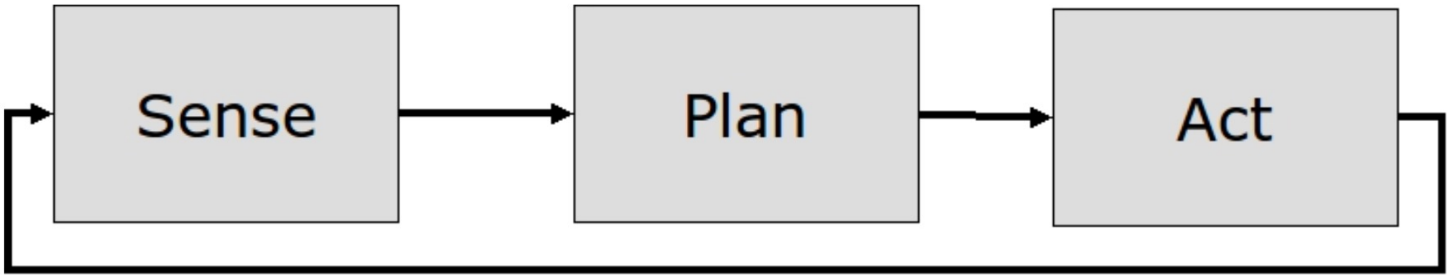
- ERR

$\sum((x_i^b - x)^2 + (y_i^b - y)^2 - d_i)^2$



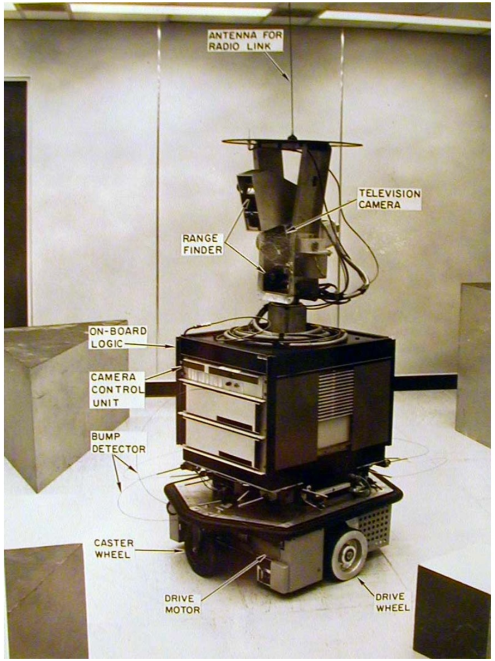
경희대학교 Kyung Hee University

Trends in Robotics Research: Classical Paradigm



- Focus on automated reasoning and knowledge representation
- Perfect world model
- Closed world assumption: “what is not currently known to be true, is false”
- STRIPS (Stanford Research Institute Problem Solver)

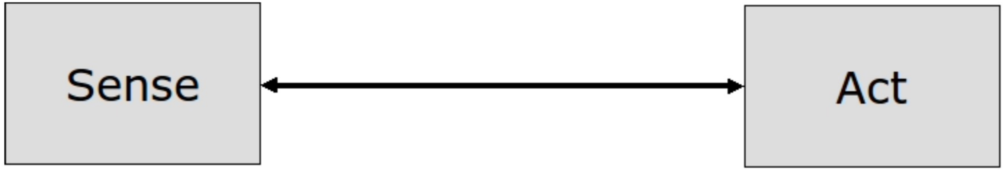
Shakey (Stanford Research Institute, 1966)



http://en.wikipedia.org/wiki/Shakey_the_robot

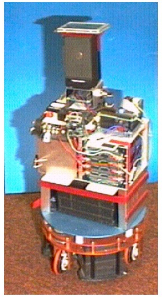
- First mobile robot to reason about its own actions
- Programs for “seeing,” “reasoning,” and “acting”
- Triangulating range-finder for sensing obstacles
- Wireless radio and video camera
- Used STRIPS to perform “block-worlds” tasks
- Conducted in Robot Hall of Fame in 2004

Trends in Robotics Research: Reactive Paradigm



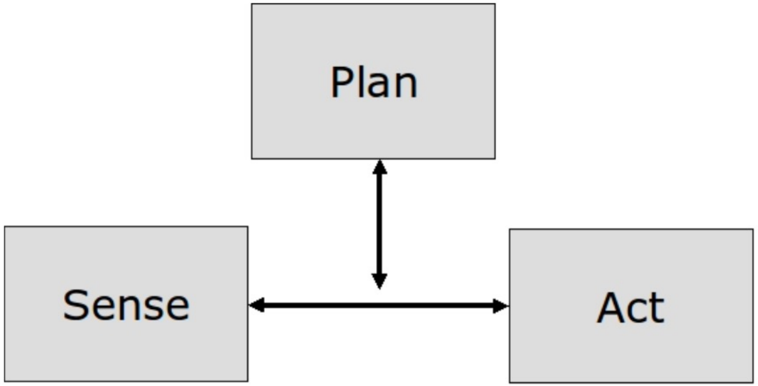
- No models: The world is its own, best model
- Many successes, but also limitations
- Inspired by biological systems

Genghis by Rodney A. Brooks



Polly by Ian Horswill

Trends in Robotics Research: Hybrid Paradigm = Planning + Reactive



- Combines advantages of previous paradigms
- World model used for planning
- Closed loop, reactive control

Trends in Robotics Research

Classical Paradigm (mid 1970s)

- exact models
- no sensing necessary

Reactive Paradigm (mid 1980s)

- no models
- relies heavily on good sensing

Hybrid Paradigm (since 1990s)

- model-based at higher levels
- reactive at lower levels

Probabilistic Paradigm (since mid 1990s)

- seamless integration of models and sensing
- inaccurate models, inaccurate sensors