

(Probabilistic) Robotics

Applied artificial intelligence (EDAF70)

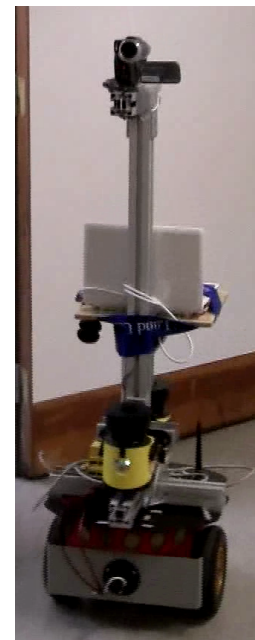
Lecture 13

2018-03-06

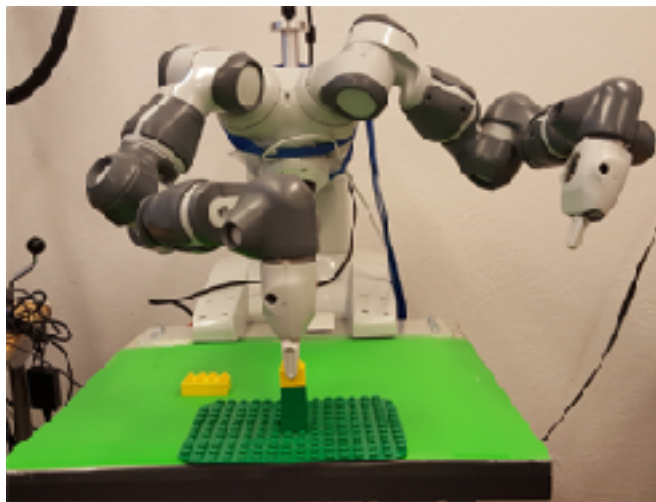
Elin A. Topp

Course book (chapters 15 and 25), images & movies from various sources, and original material
(Some images and all movies will be removed for the uploaded PDF)

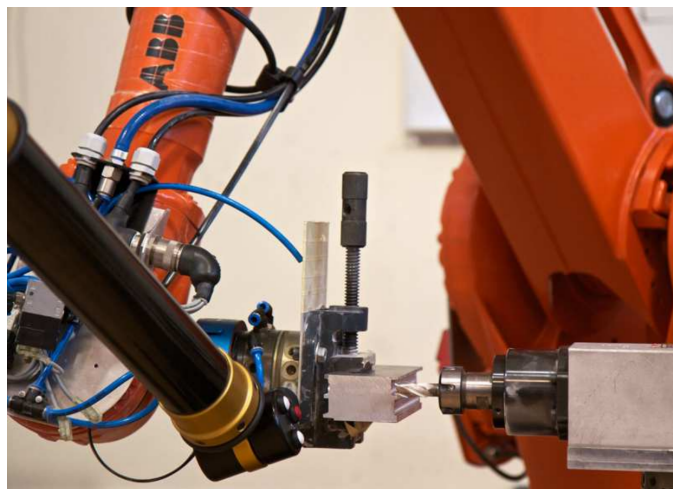
What is a “Robot”?



Honda Asimo



Keepon ?



Leonardo (MIT)



iCub (IIT)



Robots, and what they can do...

How far have we come?

Robots, and what they can do...

How far have we come?

Frida “feels” when work’s done... 2013 (Magnus Linderöth, LU)

YuMi wraps gifts... 2015 (Maj Stenmark & Andreas Stolt, LU & Cognibotics AB)

Anyone can program robots... right? 2017 (Maj Stenmark, Elin A. Topp, Cognibotics & LU)

YuMi has learned to feel when things click... 2018 (Martin Karlsson, LU)

Types of robots

Industrial robots vs. service robots vs. personal robots / robot toys

Static manipulators vs. mobile platforms (vs. mobile manipulators)

Mechanistic vs. humanoid / bio-inspired / creature-like

For all in common:

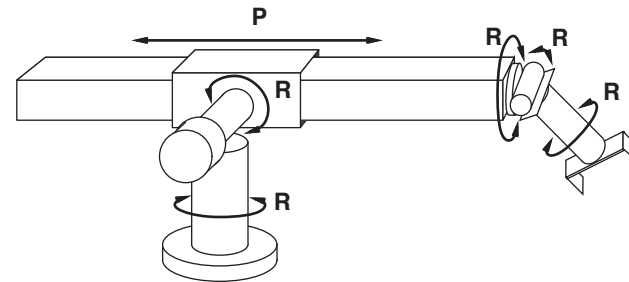
A robot is a (physical) agent in the (physical) world
(with all the consequences that might have... ;-)

Darpa Urban Challenge 2007
Sting racing crash

Impression from user study
(Maj Stenmark, Elin A. Topp)

Robot actuators - joints and wheels

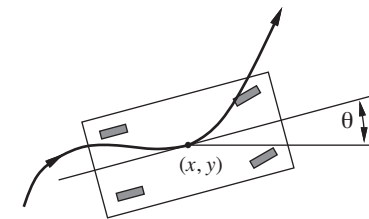
6 DOF (6 “joint”) arm:



2x7 DOF (“humanoid” torso “YuMi” / Frida):



2 (3 effective) DOF synchro drive (car):



2 (3 effective) DOF differential drive (Pioneer 3-DX):



3 DOF holonomic drive (“shopping cart”, DLR’s Justin):



Kinematics - controlling the DOFs

Direct (forward) kinematics (relatively simple):

Where do I get with a certain configuration of parts / wheel movement?

Inverse kinematics (less simple, but more interesting):

How do I have to control joints and wheels to reach a certain point?

Dynamics - controlling consequences of movement

Dynamics:

Make the robot move (and move stuff) without falling apart, or crashing into things

How much payload is possible?

How fast can I move without tipping over?

What is my braking distance?

How do I move smoothly? (ask the automatic control people ;-)



Weight: ca 1300 kg

Payload: ca 150 kg

Impression from user study
(Elin A. Topp)

Dynamics in practice

Dynamics also gets you into two problems: direct and inverse dynamics.

Direct dynamics:

Given masses, external forces, position, velocities and acceleration in the joints / wheels, what forces / moments are put to the depending joints and the tool centre point (TCP)? “Rather” simply solvable, at least more or less straight forward.

Inverse dynamics (again, more interesting than direct dynamics):

While solving the *inverse kinematics* problem is nasty, but still “only” a bunch of linear equations, solving the *inverse dynamics* problem leaves you with a bunch of more or less complex differential equations.

Supporting parts: Sensors

In a predictable world, we do not need perception, but good planning and programming

As the world is somewhat unpredictable, some perception is useful, i.e. robots / robot installations need sensors.

Passive / active sensors.

Range / colour / intensity / force / direction ...

Optical / sound / radar / smell / touch ...

Most common for mobile robots: position (encoders / GPS), range (ultrasound or laser range finder), image (colour/intensity), sound

Most common for manipulators: position (encoders), force / torque, images, (range - infrared, laser RF)

Sensors on a mobile robot



Microphones (sound)

Ultrasound (24 emitters / receivers) (range)

Camera (image - colour / intensity)

Laser range finder (SICK LMS 200) (range)

Infrared (range / interruption)

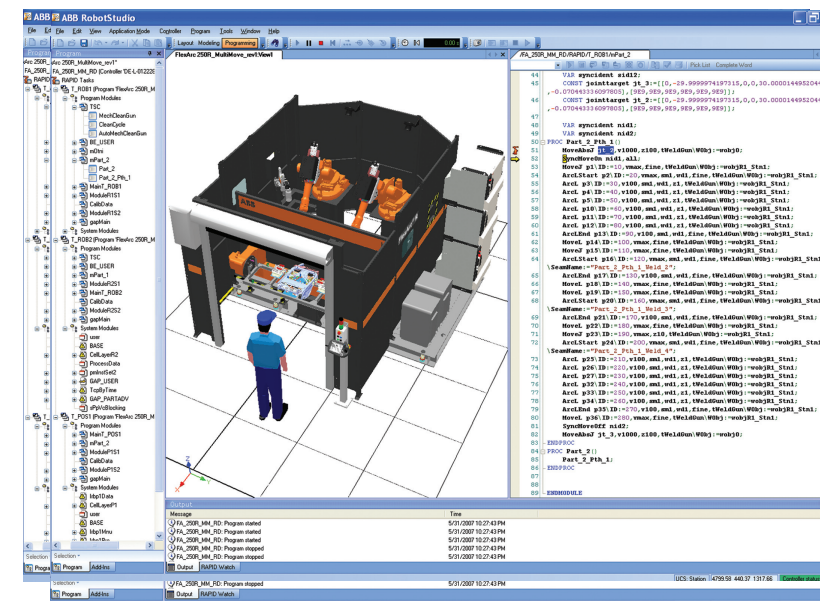
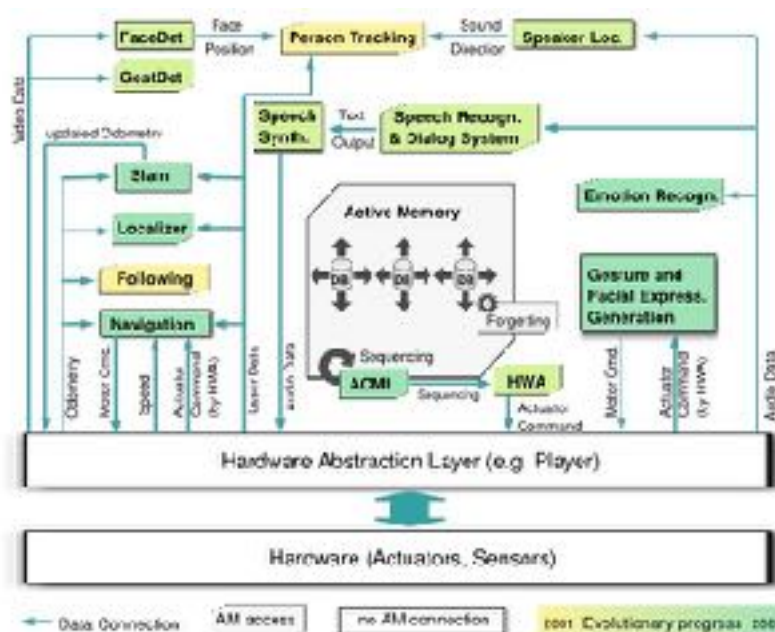
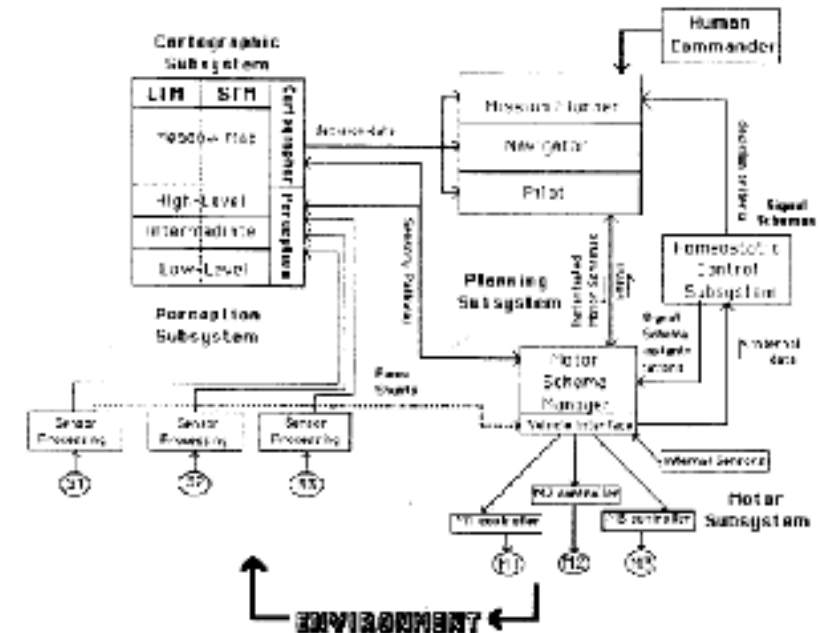
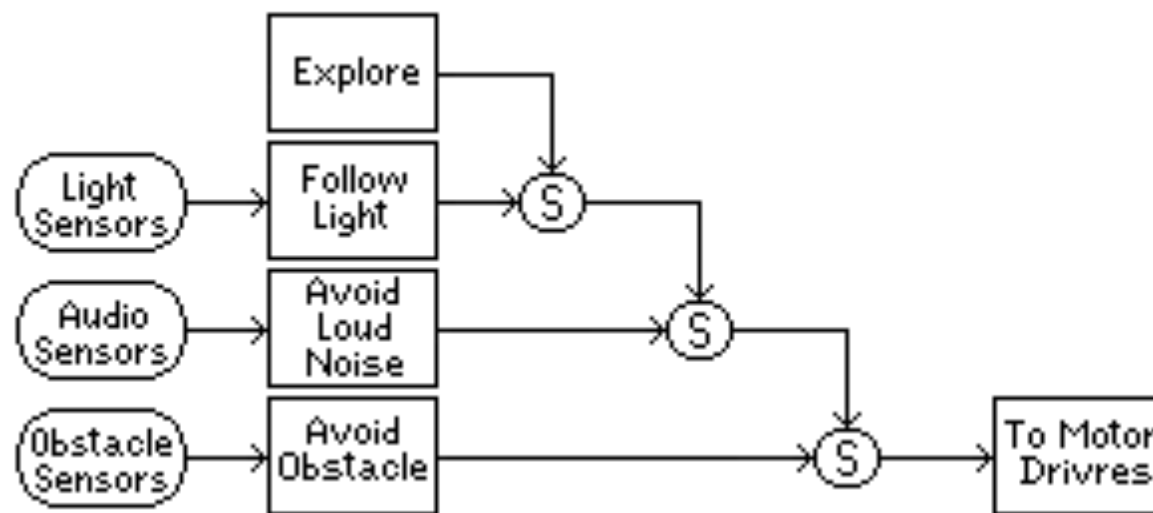
Bumpers (touch)

Wheel encoders (position / pose)

System integration

Make sensors, actuators and algorithms work together

Architectures, “operating systems”, controllers, programming tools ...



System integration - system is bigger than the sum of its components

Impression from user study
(Elin A. Topp, H. Huettenrauch, 2006)

Darpa Robotics Challenge 2015,
Robots falling over

Ethics detour

Robots can be the “cool” embodiment of artificially intelligent systems - but their reasoning mechanisms can only build upon a given baseline.

So far, systems will take instructions literally, and only reason within given limits.

AI-systems must be capable of explaining themselves, and we should not expect them to be more than they are!

Robot and Frank - what is stealing?

Some fundamental AI challenges in robotics - or should we say robotics challenges in AI?

We need to describe and control our robots' movement to make them do something

We need to either plan and program every little detail of their movement

OR

We need to equip them with capabilities for perception, understanding, insight, deliberation - but when we do that, we need to “plan and program” for that

How can a mobile platform know how to go to place X, when it does not know where X is - and what its own position is? (Describe the world)

How can a two-armed robot know how to glue two work-pieces together, if it does not know what the work-pieces are, where to find them, how to grasp them, how to direct them towards each other, or how to glue stuff together? (Describe tasks)

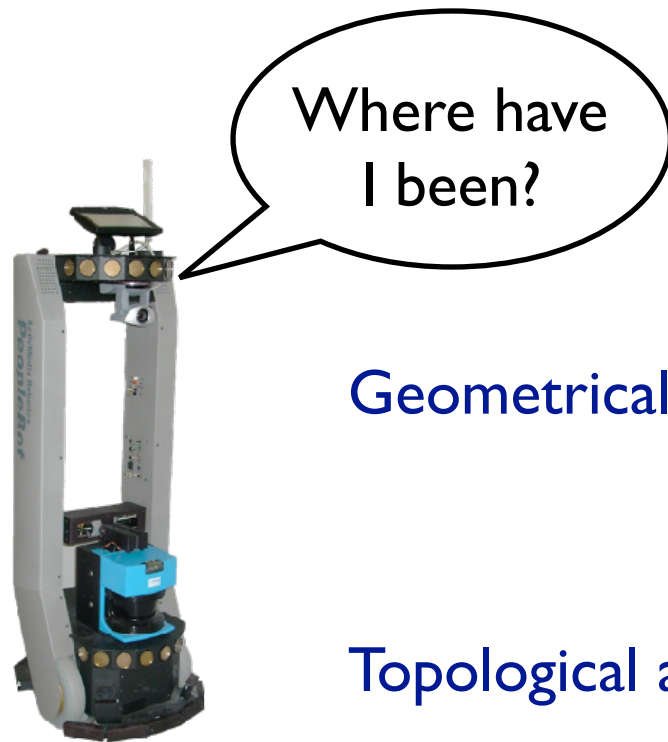
How can a robot understand that a correctly planned action did not result in the foreseen outcome? What should it do to resolve this situation? (Describe problems)

Outline

AI in Robotics - integrating the “brain” into the “body”

- Probabilistic methods for Mapping & Localisation
 - SJPDFs for person tracking
 - Identifying interaction patterns in Human Augmented Mapping with BNs
 - (Reinforcement) Learning in robotics
-
- Deliberation & High level decision making and planning
 - Knowledge representation, reasoning, and NLP to support HRI and high-level robot programming

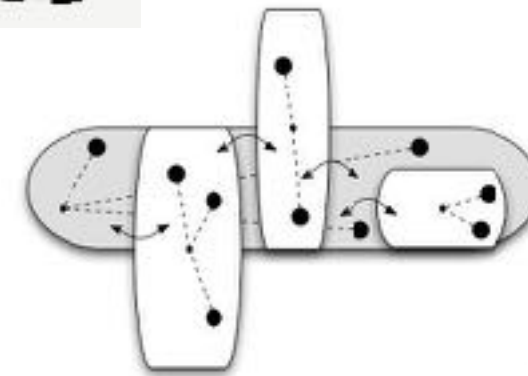
Mapping



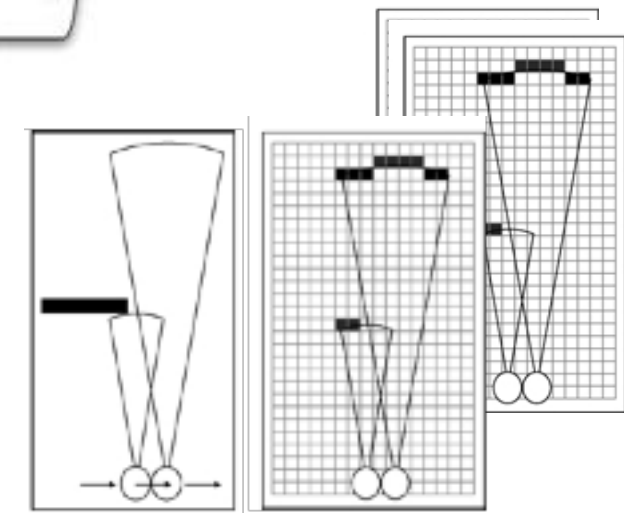
Geometrical approaches



Topological approaches



Occupancy grid approaches (e.g., Sebastian Thrun)



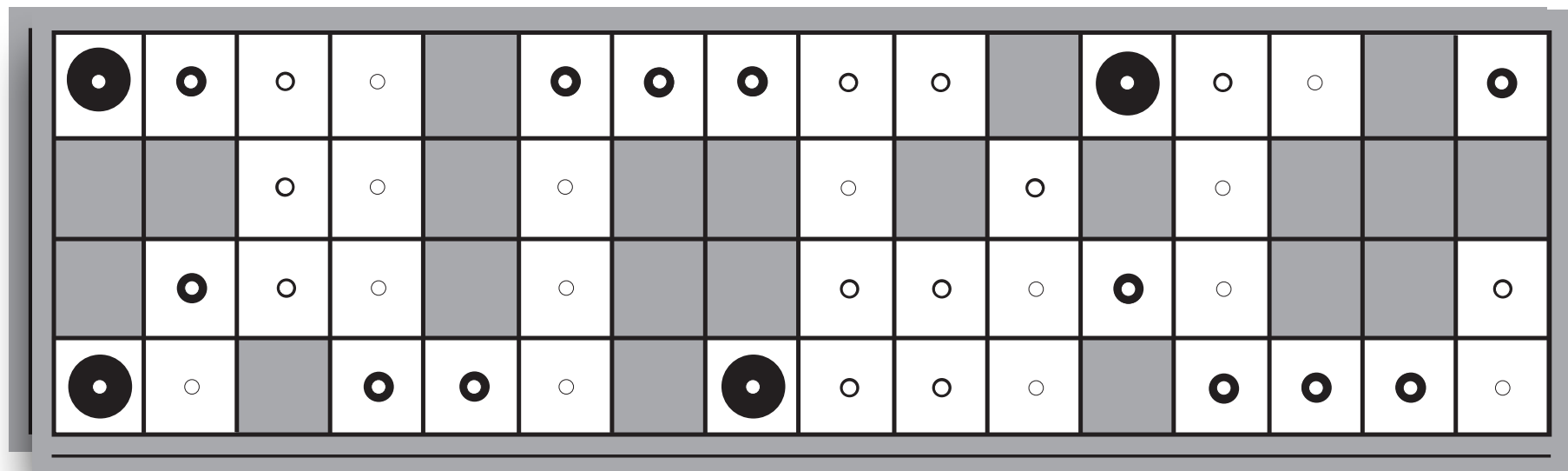
(Hybrid approaches)

Localisation

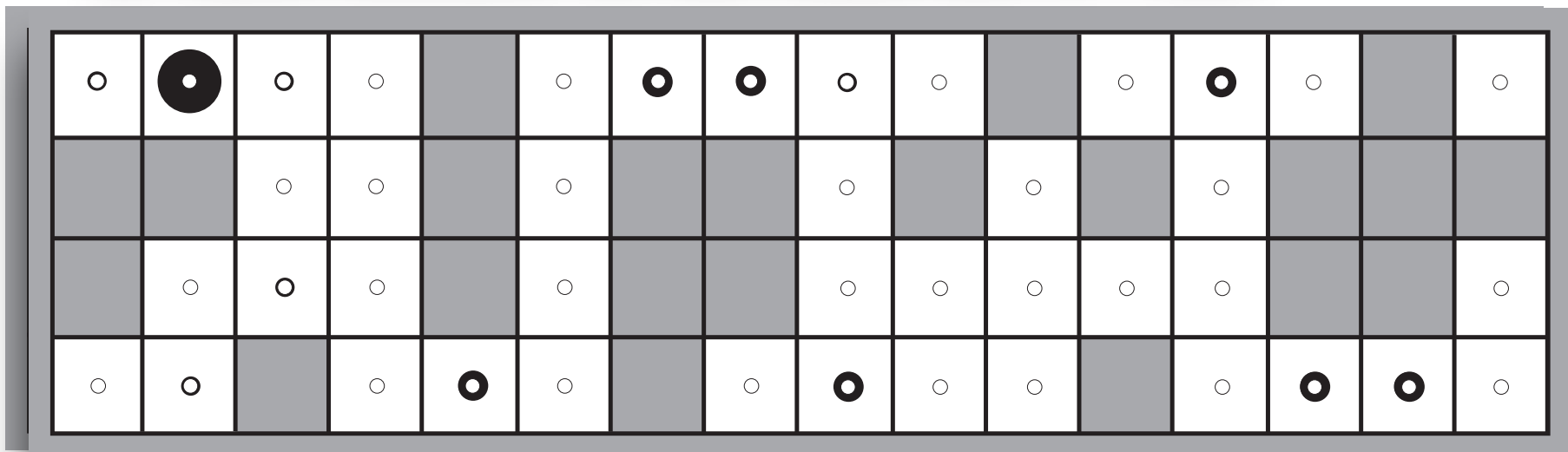


Where am I now?

HMM in a grid world



(a) Posterior distribution over robot location after $E_1 \equiv NSW$



(b) Posterior distribution over robot location after $E_1 \equiv NSW, E_2 \equiv NSS$

Data filters for state estimation

0. Represent state, identify system function
1. Estimate / predict state from model applying the function
2. Take a measurement
3. Update state according to model and observation (measurement)

Used for position tracking, detection of significant changes in a data stream, localisation ...

E.g., particle filters (Monte Carlo), Kalman filters

Particle filter

1. Represent possible positions by samples (uniform distribution) $\mathbf{x} = (x, y, \theta)$
2. Estimate movement / update samples according to assumed robot movement + noise
3. Take a measurement z
4. Assign weights to samples according to *posterior probabilities* $P(x_i | z)$
5. Resample (pick “good” samples, use those as new “seeds”, redistribute in position space and add some noise), continue at 2.

Localisation



E.g., Monte Carlo Localisation (D. Fox, S. Thrun, *et al.*)

Particle filter based localisation indoor

Kalman filter

Represent posteriors for transition and sensor models with a Gaussian.

$$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})(\mathbf{x}) = \alpha e^{-1/2((\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}))}$$

Assume *locally linear* dynamical system with

F, **H** system matrices, i.e., transition and sensor model;

x_t the state in time t, **z**_t measurement in time t (vectors);

Σ_x and **Σ**_z the covariances for the transitions and measurements, i.e., gaussian noise)

$$P(\mathbf{x}_{t+1} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{F}\mathbf{x}_t, \boldsymbol{\Sigma}_x)(\mathbf{x}_{t+1})$$

$$P(\mathbf{z}_t \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{H}\mathbf{x}_t, \boldsymbol{\Sigma}_z)(\mathbf{z}_t)$$

Then do prediction and update, just as with the HMM (well, almost ;-)

Kalman filter, updates

Update the overall mean *state estimate* μ_{t+1} (essentially \mathbf{x}_{t+1}) based on last estimate μ_t as

$$\mu_{t+1} = \mathbf{F}\mu_t + \mathbf{K}_{t+1}(\mathbf{z}_{t+1} - \mathbf{H}\mathbf{F}\mu_t)$$

with $\mathbf{F}\mu_t$ the state prediction and $\mathbf{H}\mathbf{F}\mu_t$ the prediction of the observation and

update the overall covariance as

$$\Sigma_{t+1} = (\mathbf{I} - \mathbf{K}_{t+1}\mathbf{H}) \mathbf{F} \Sigma_t \mathbf{F}^T + \Sigma_x$$

with $\mathbf{K}_{t+1} = (\mathbf{F} \Sigma_t \mathbf{F}^T + \Sigma_x) \mathbf{H}^T (\mathbf{H}(\mathbf{F} \Sigma_t \mathbf{F}^T + \Sigma_x) \mathbf{H}^T + \Sigma_z)^{-1}$ the *Kalman gain matrix* (essentially what we win in terms of certainty when doing the update)

Where am
I now?



GPS-free positioning at sea

Depth and magnetic field measurements in a PF + KF combination

Where am
I now?



GPS-free positioning at sea (2)

Example for discarding samples: When particles are “one the wrong side” of the bottom depth line, they are discarded

Where am
I now?



GPS-free positioning at sea (3)

Combining the different distributions (for depth and magnetic field measurements) into a joint PDF

Mapping & Localisation: Chicken & Egg?

Simultaneous localisation and mapping (SLAM)

While building the map, stay localised!

Use filters to “sort” landmarks:

Known? Update your pose estimation!

Unknown? Extend the map!

SLAM example

FastSLAM (D. Haehnel)