Mathematics For Intelligent Systems 5

Dr.Don.S and Dr. Sunder Ram K

ML Experts

- Grady Jensen – linear regression and linear classification https://argmax.ai/ml-course/
- Nando de Freitas - Deep understanding of ML
- Kilian Weinberger – Deep understanding of ML
- Steve Brunton – Control theory

<table>
<tr><td colspan="3">Internal</td><td>External</td><td>Total</td></tr>
<tr><td>Components</td><td colspan="2">Weightage</td><td rowspan="4">Project Based
3 evaluations
Weightage: 30%
(7.5,7.5,15)

1st evaluation nov 30
7.5
2nd evaluation Dec 20th
7.5
3rd evaluation Jan 10th
15 : report 5 +demo 10

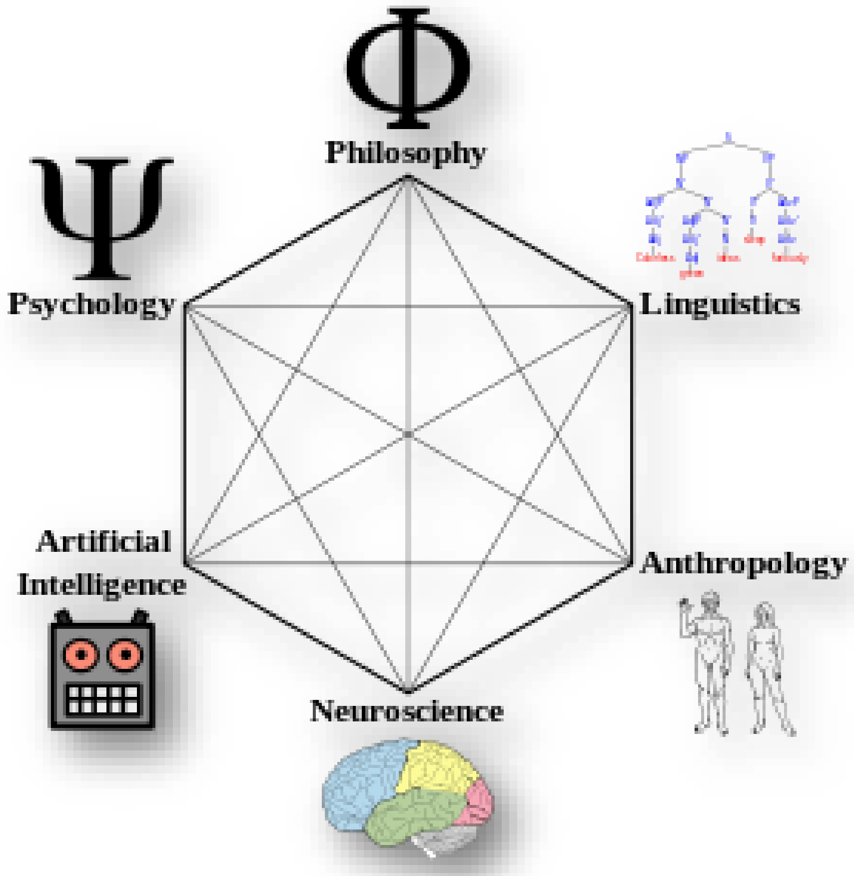
Negative mark for late submission
Max team size: 2</td><td rowspan="4">Internal + External=100</td></tr>
<tr><td>Assignments(2)
(3th week of Nov ,
4rd week of Dec)
Lab weekly evaluation - Summarizing research article, implementation, quiz based on research articles</td><td>30%</td><td rowspan="3">70%</td></tr>
<tr><td>Mid-Term</td><td>20%</td></tr>
<tr><td>Quiz (2)
(1st week of Dec , -unit 2
1st week of Jan) –unit 1 and 3</td><td>20%</td></tr>
</table>

Course Outcome

- CO1: Understand and implement basic concepts and techniques of probabilistic graphical models needed for causal reasoning in AI
- CO2: Apply the concepts of linear algebra, optimization and probability theory for controlling real-world systems
- CO3: Identify the connection between the concepts of linear algebra, differential equation and probability theory
- CO4: Understand and implement latest data-driven modelling of linear and non-linear dynamical systems through modern matrix/tensor decomposition techniques

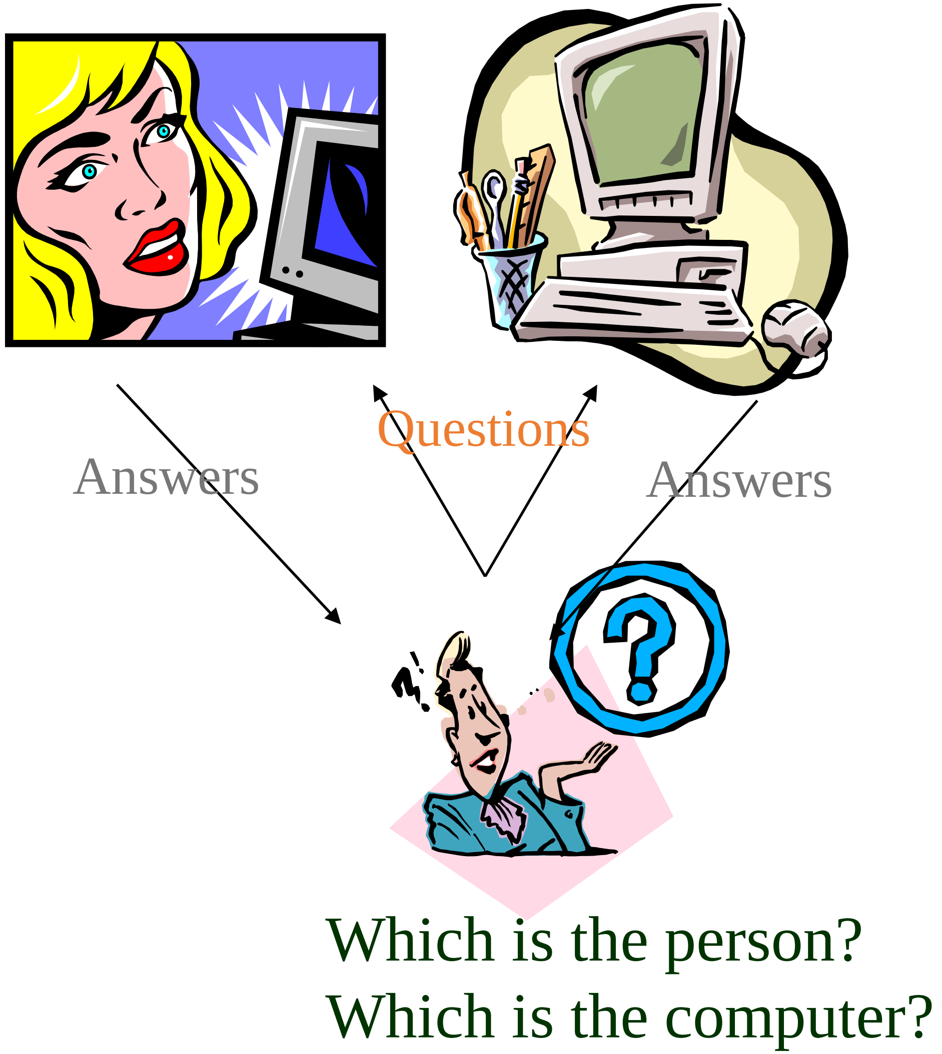
Motivation

Cognitive Science – scientific study of the Human brain,**Understanding Intelligence**



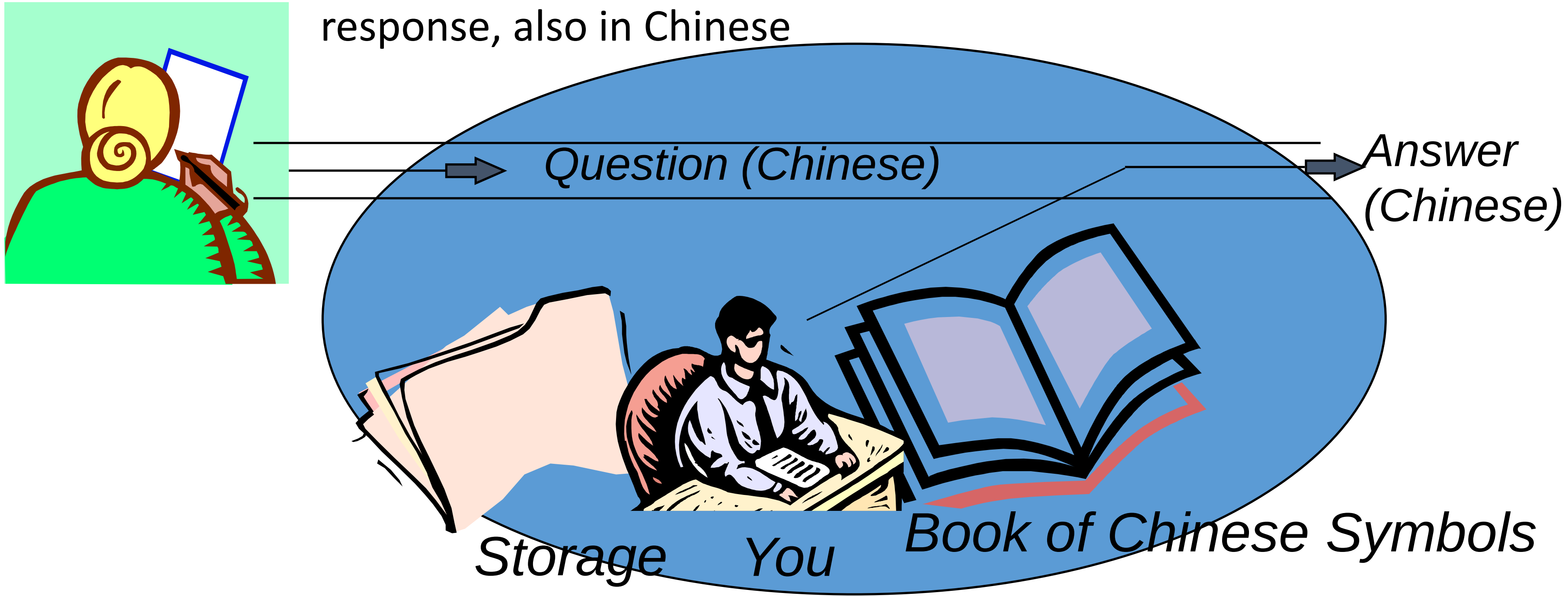
Testing “Intelligence” with the Turing Test

- 1950 – Alan Turing devised a test for intelligence called the Imitation Game
  - Ask questions of two entities, receive answers from both
  - If you can’t tell which of the entities is human and which is a computer program, then you are fooled and we should therefore consider the computer to be intelligent



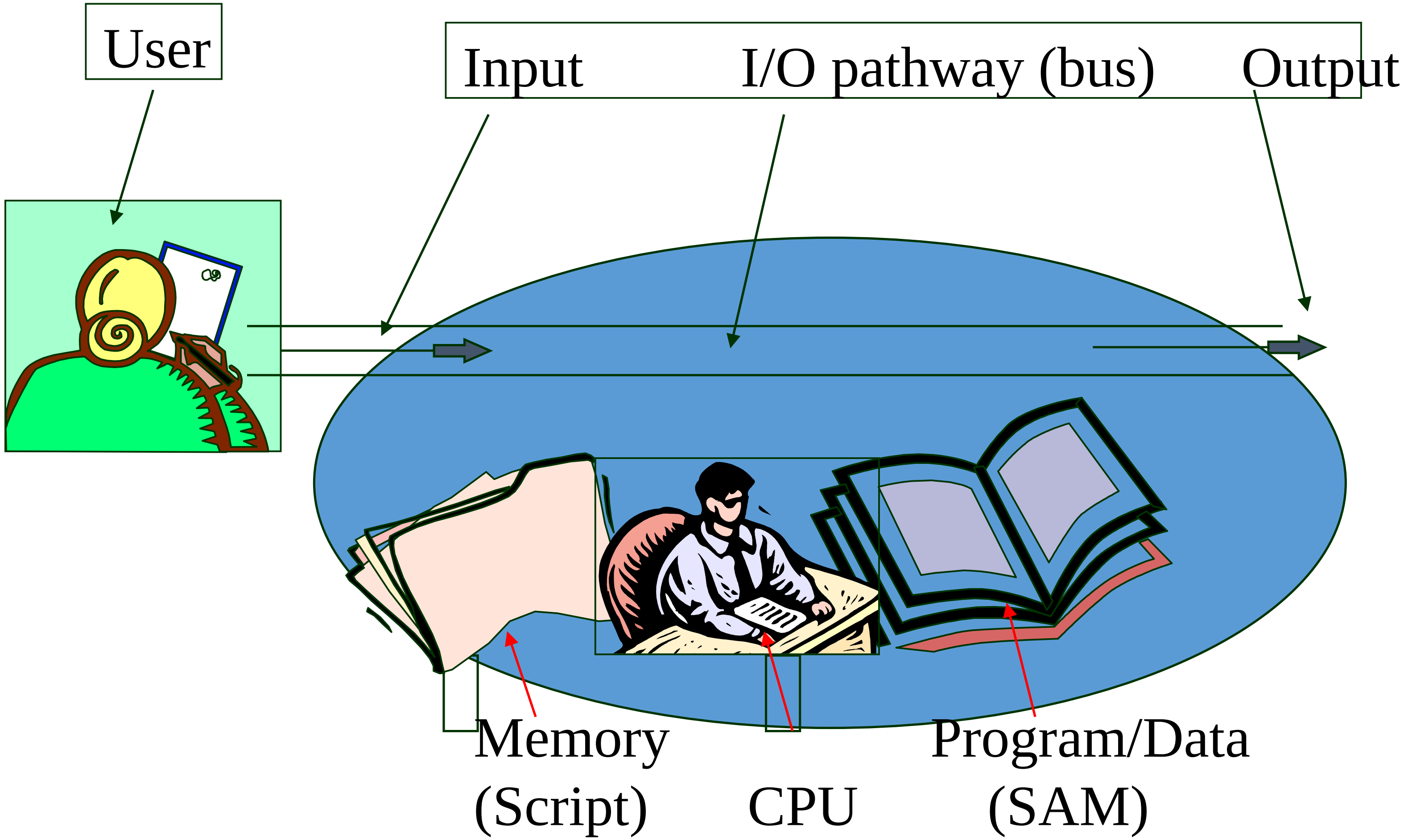
The Chinese Room Problem

- From John Searle, Philosopher, in an attempt to demonstrate that computers cannot be intelligent
  - The room consists of you, a book, a storage area (optional), and a mechanism for moving information to and from the room to the outside
    - a Chinese speaking individual provides a question for you in writing
    - you are able to find a matching set of symbols in the book (and storage) and write a response, also in Chinese



Chinese Room: An Analogy for a Computer

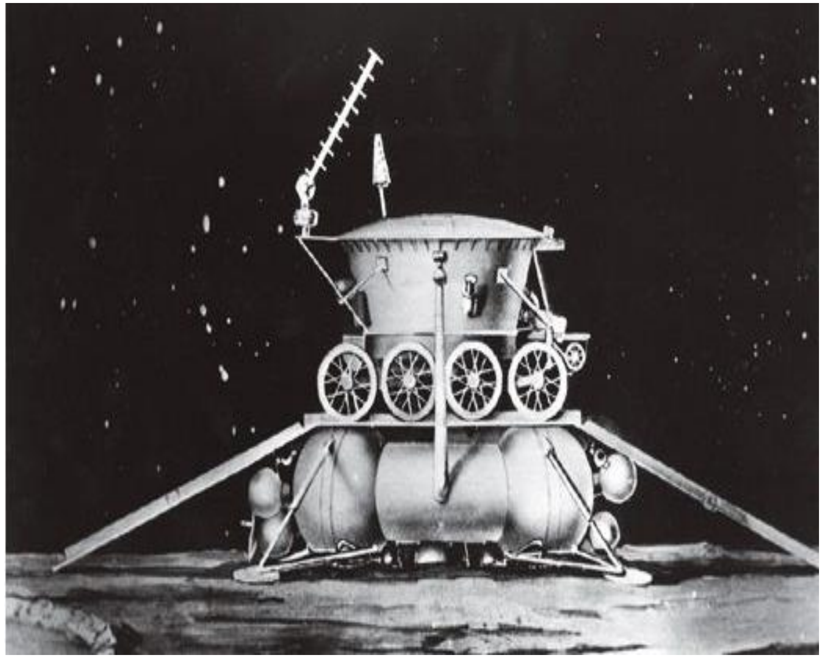
Note: Searle’s original Chinese Room actually was based on a Script that was implemented in Chinese, our version is just a variation on the same theme



- You were able to solve the problem of communicating with the person/user and thus you/the room passes the Turing Test
- But did you understand the Chinese messages being communicated?
  - since you do not speak Chinese, you did not understand the symbols in the question, the answer, or the storage
  - can we say that you actually *used* any intelligence?
- By analogy, since you did not understand the symbols that you interacted with, neither does the computer understand the symbols that it interacts with (input, output, program code, data)
- Searle concludes that the computer is not intelligent, it has no "semantics," but instead is merely a symbol manipulating device
  - the computer operates solely on syntax, not semantics

What is Intelligent?

- "Intelligence denotes the **ability of an individual to adapt his thinking to new demands**; it is the common mental adaptability to new tasks and conditions of life" (William Stern, 1912)

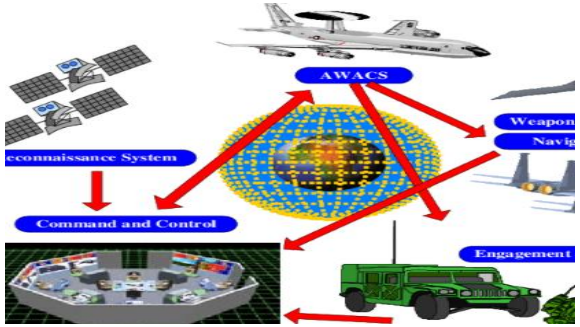


Intelligence must be able to perform

- perceive, reason and infer, solve problems, learn and adapt, apply common sense, apply analogy, recall, apply intuition, reach emotional states, achieve self-awareness

Application of Intelligent Systems

Industrial Automation



Military Applications

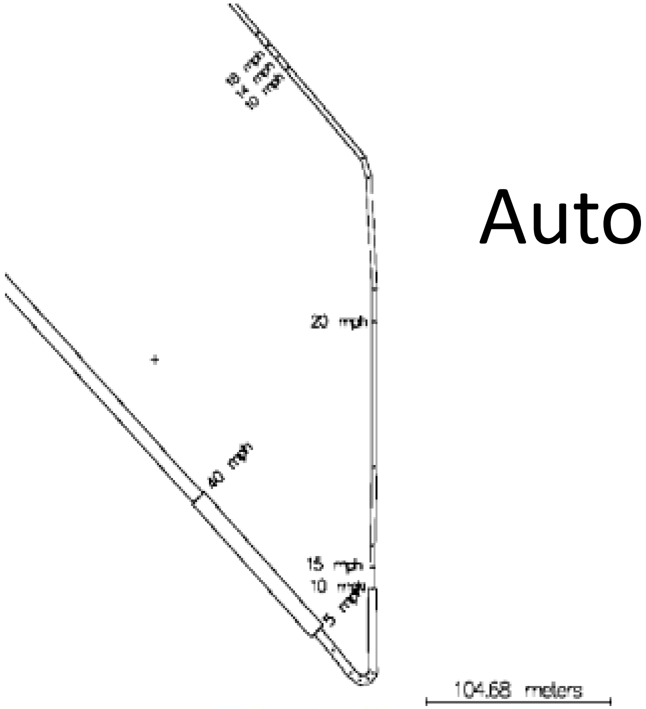
Clinical Applications

Challenges

- Uncertainty
- Dynamic World
- Time consuming computation
- Mapping
- ….

Why we study MIS5

(a)

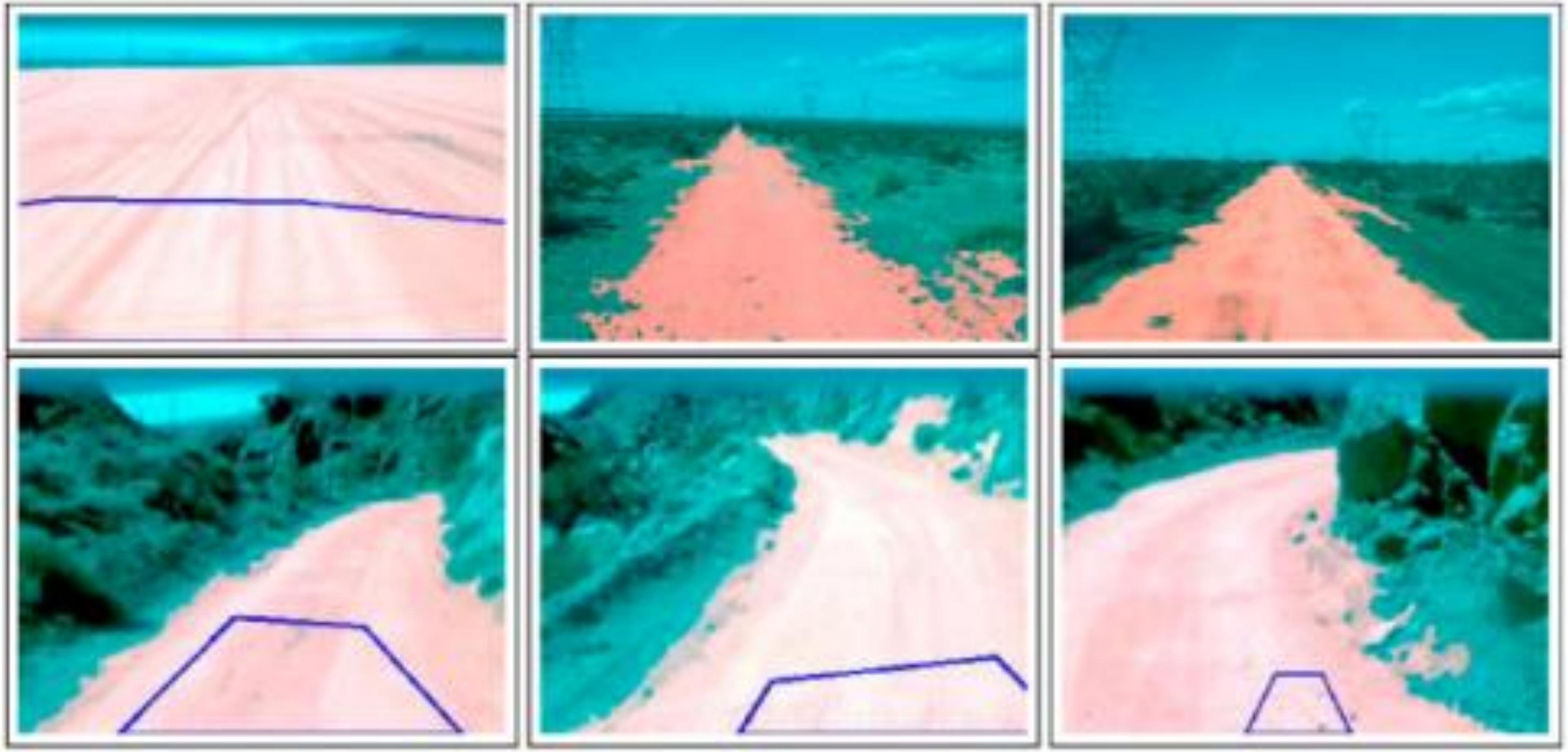


Autonomous Driving

(a)

(b)

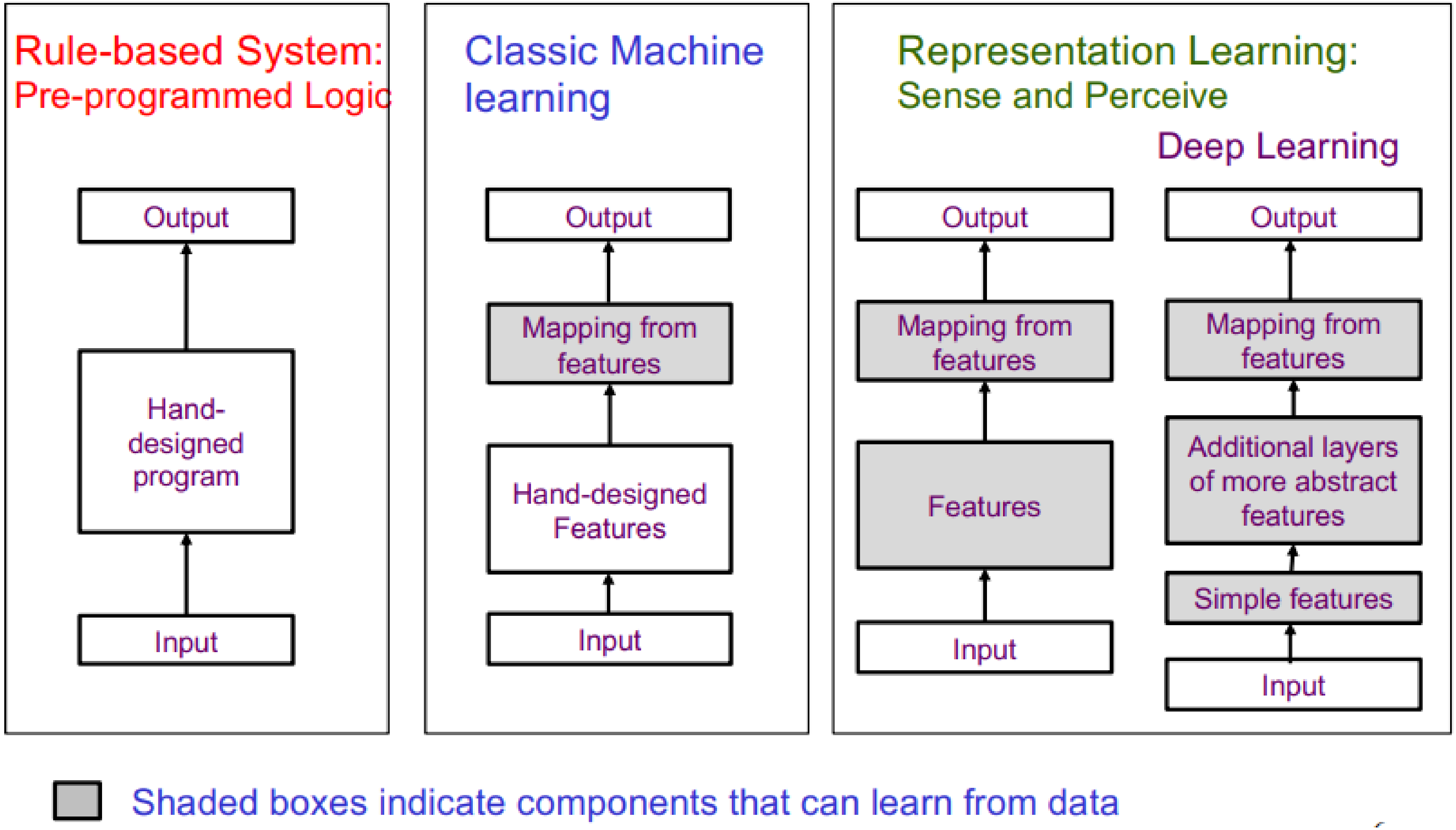
(c)



- Module 1 Dr. Don.S
  - Data Driven Dynamical Systems: Motivation and Challenges, Dynamic Mode decomposition, Sparse identification of Non-linear Dynamics.
- Module 2 Dr. Sunder Ram K
  - Probability theory, Bayesian Networks (BNs), Representation Learning in Bayesian Networks, Markov Random Fields- MRF, Inference, Message Passing, Learning in Markov Networks, Numerical Optimization, MRFs and BNs Monte Carlo Method.
- Module 3 Dr.Don.S
  - Linear Control Theory: Closed loop Feedback Control, LTI, Controllability and Observability, Optimal Full State Control, Optimal Full-State Estimation, The Kalman Filter.

Traditional Computer System Vs Machine Learning Vs Artificial Intelligence

Current AI Models



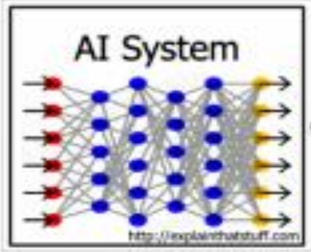
AI vs Human Intelligence

- If you are driving a car and see a soccer ball roll into the street,
- Your immediate and natural reaction is to stop the car since we can assume a child is running after the ball and isn't far behind.

Role of Probabilistic Systems

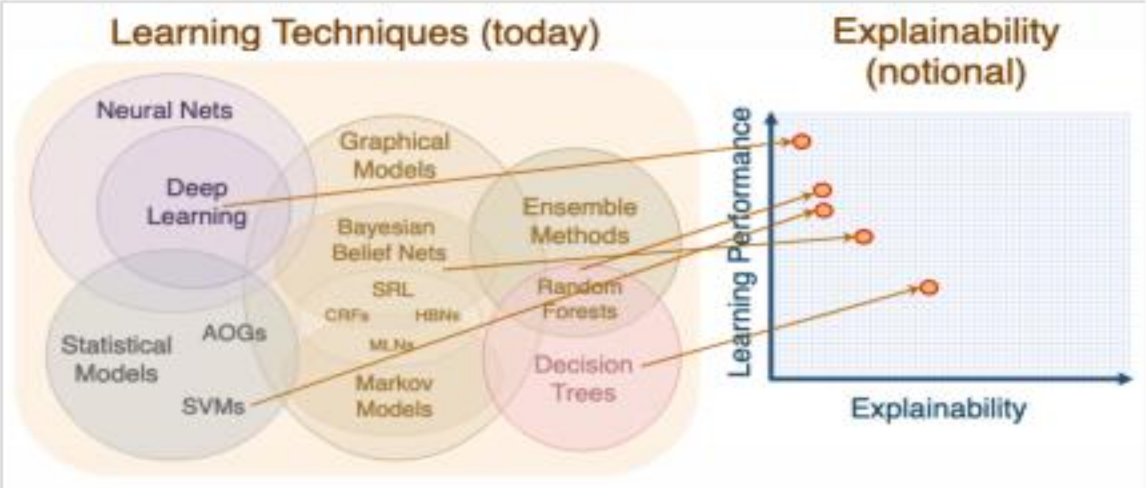
- Driver reaches the decision to stop the car based on **experience of natural data and assumptions about human behavior**.
  - But, a traditional computer likely wouldn’t reach the same conclusion in real-time, because today’s **systems are not programmed to mine noisy data efficiently** and to make decisions based on environmental awareness.
  - You would want a **probabilistic system** calling the shots-one that could quickly assess the situation and act (stop the car) immediately.

PGMs in Explainable AI

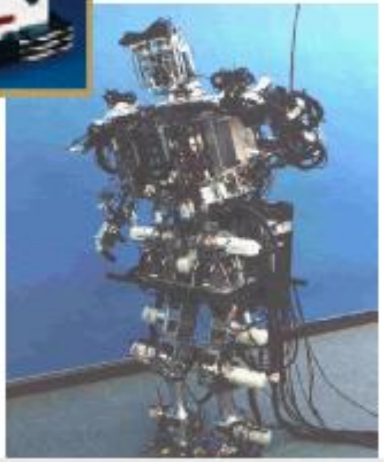


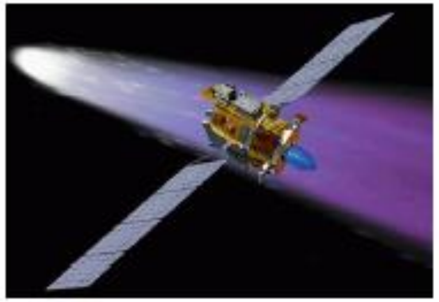
Why did you do that?
Why not something else?
When do you succeed?
When do you fail?
When can I trust you?
How do I correct an error?

Anecdote: Medical AI
Decisions can be worse with AI
e.g., Patient discharge to a nursing home



Robotics Today

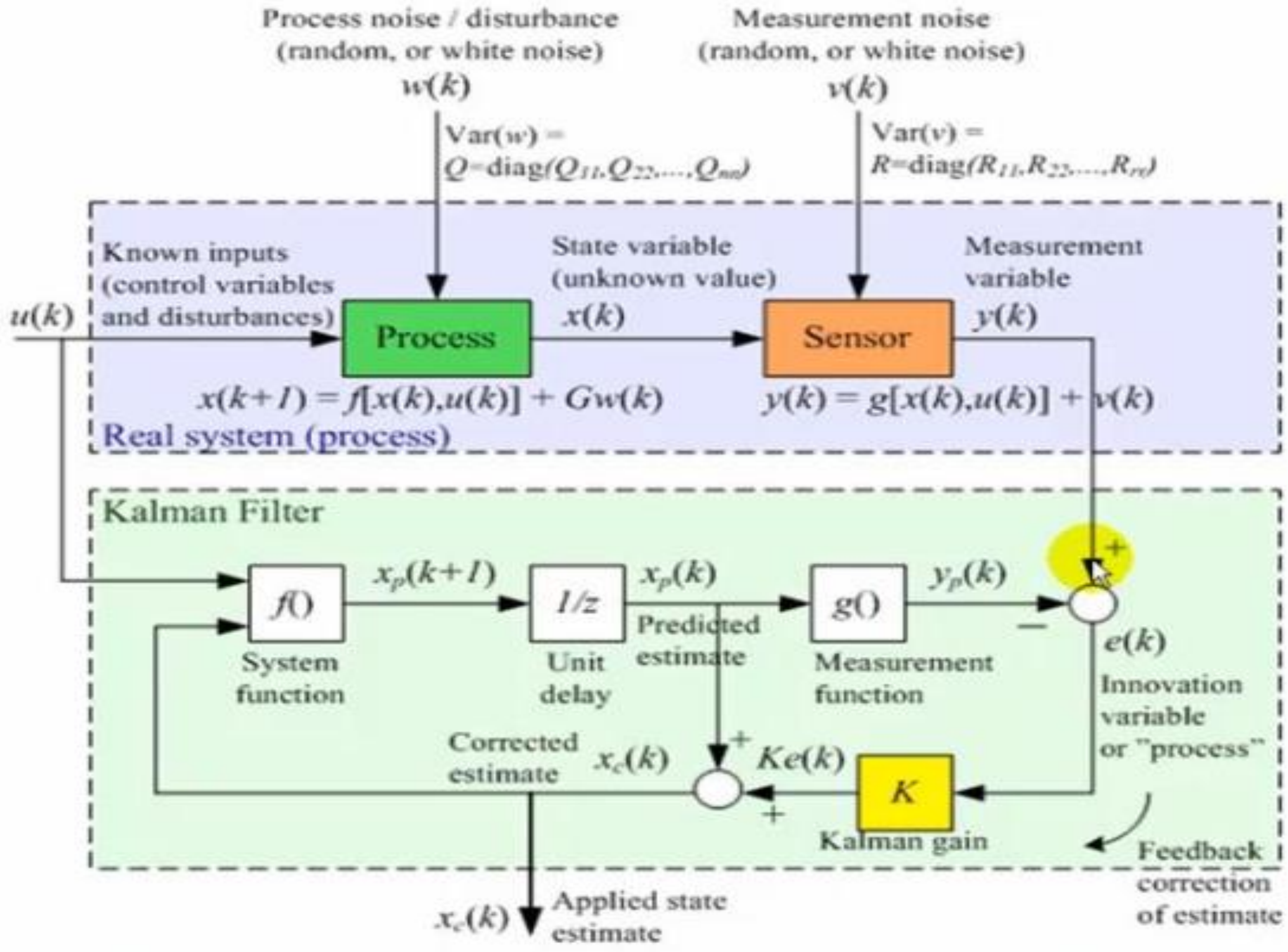




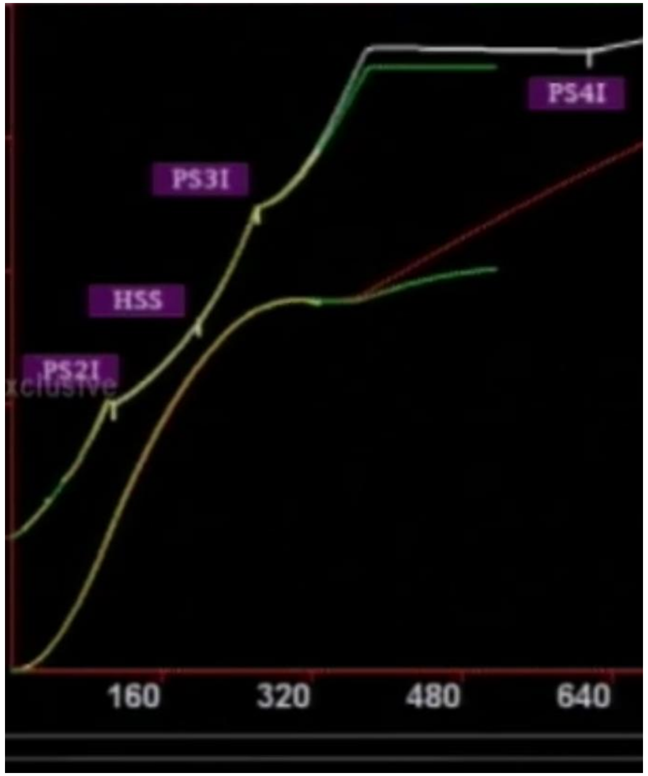
State space model -Robot

$$x_t = A_{t-1}x_{t-1} + B_{t-1}u_{t-1}$$

$$\begin{bmatrix} x_t \\ y_t \\ \gamma_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \gamma_{t-1} \end{bmatrix} + \begin{bmatrix} \cos\gamma_{t-1} * dt & 0 \\ \sin\gamma_{t-1} * dt & 0 \\ 0 & dt \end{bmatrix} \begin{bmatrix} v_{t-1} \\ w_{t-1} \end{bmatrix} + \begin{bmatrix} noise_{t-1} \\ noise_{t-1} \\ noise_{t-1} \end{bmatrix}$$



Closed loop system

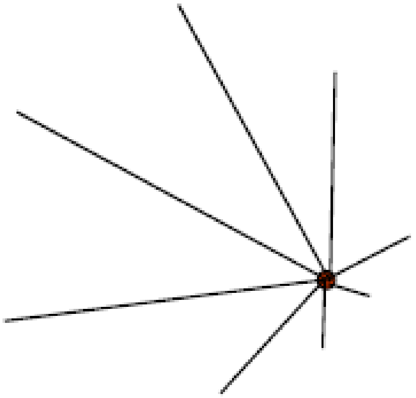


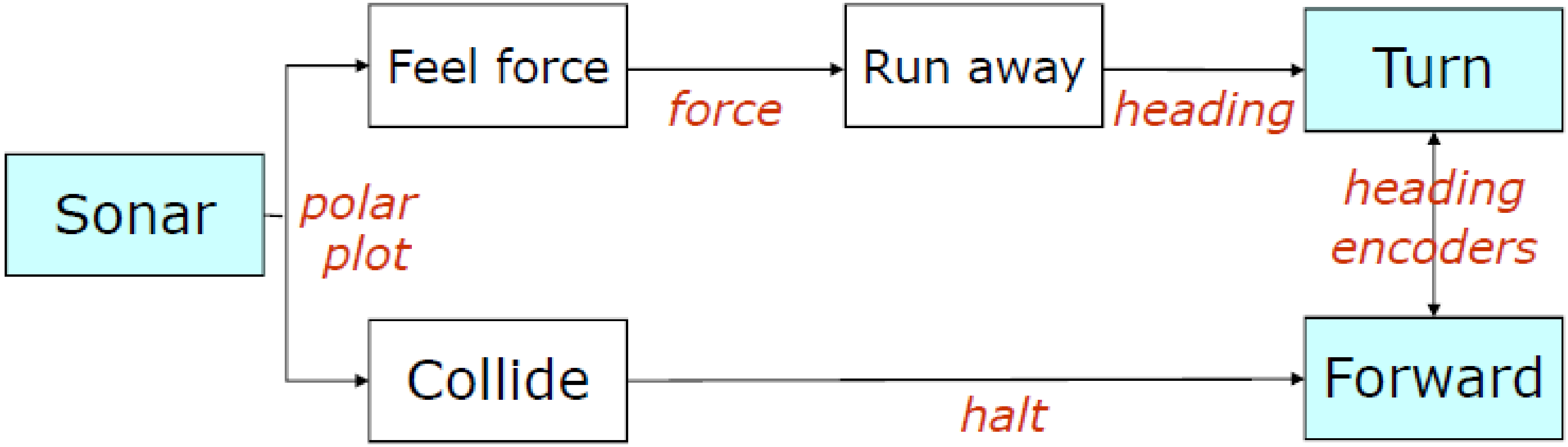
Actual vs predicted

Designing the architecture

Level 0: Avoid

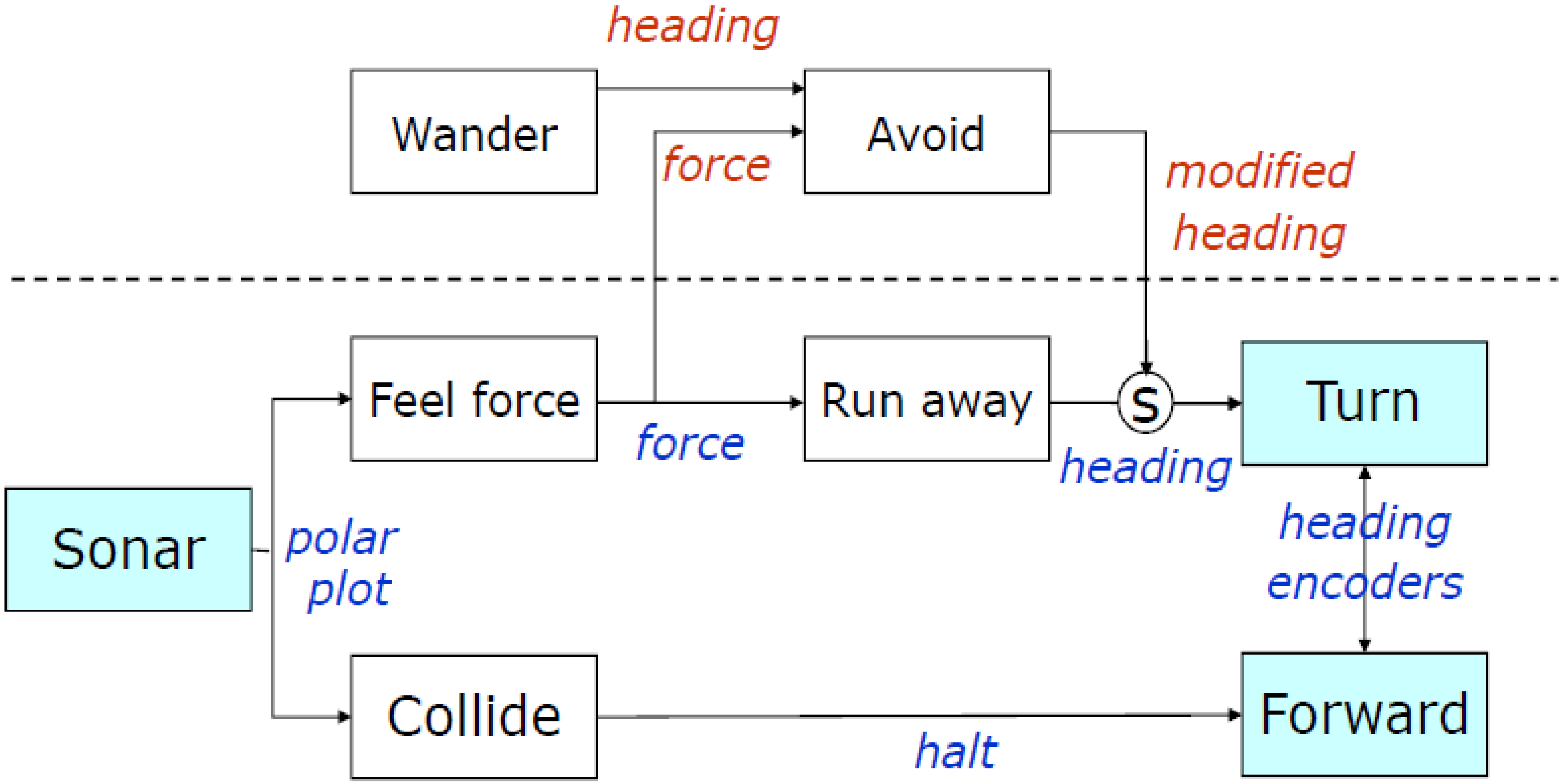
Polar plot of sonars



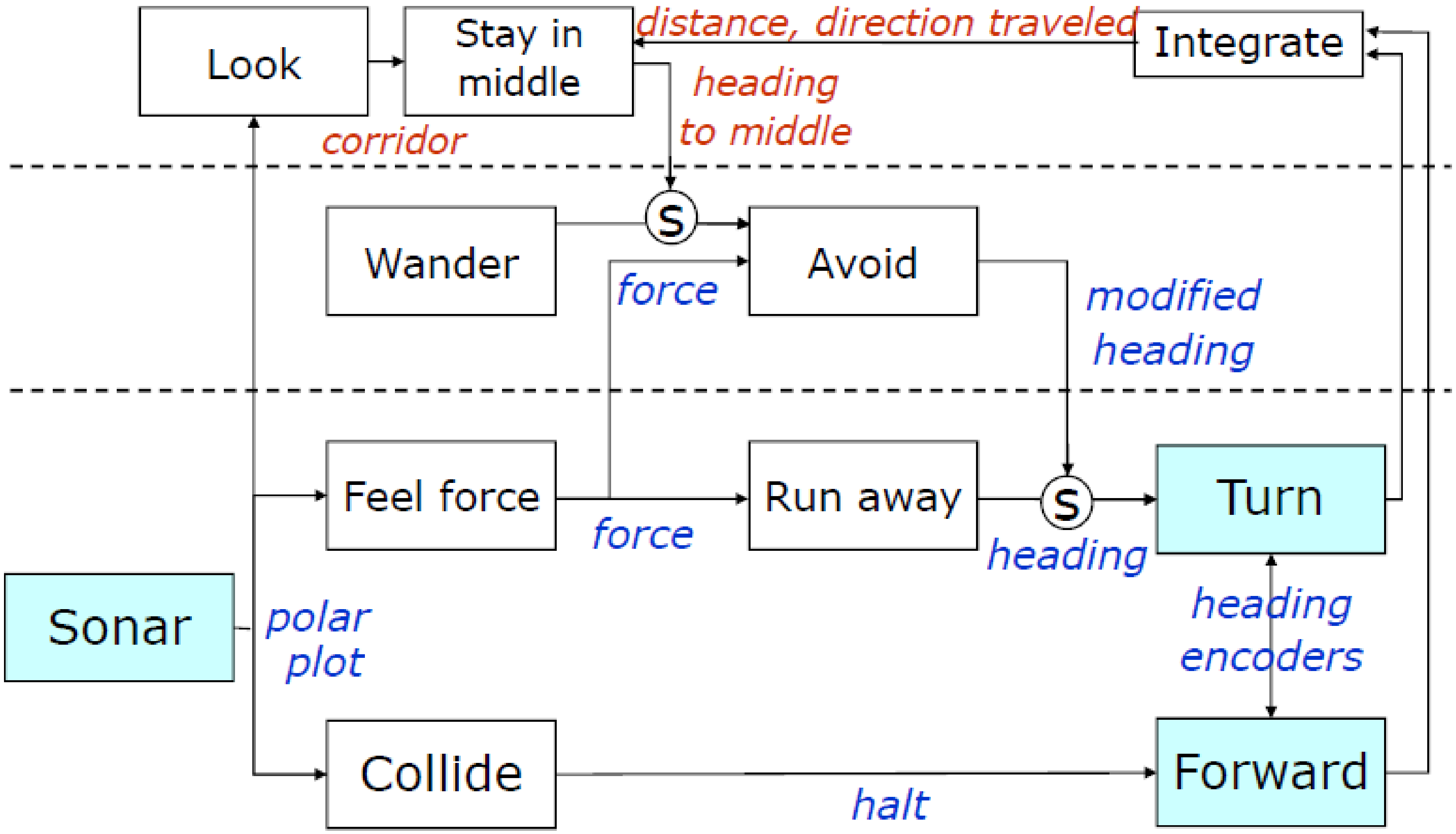


Simple mechanism on how robot will react

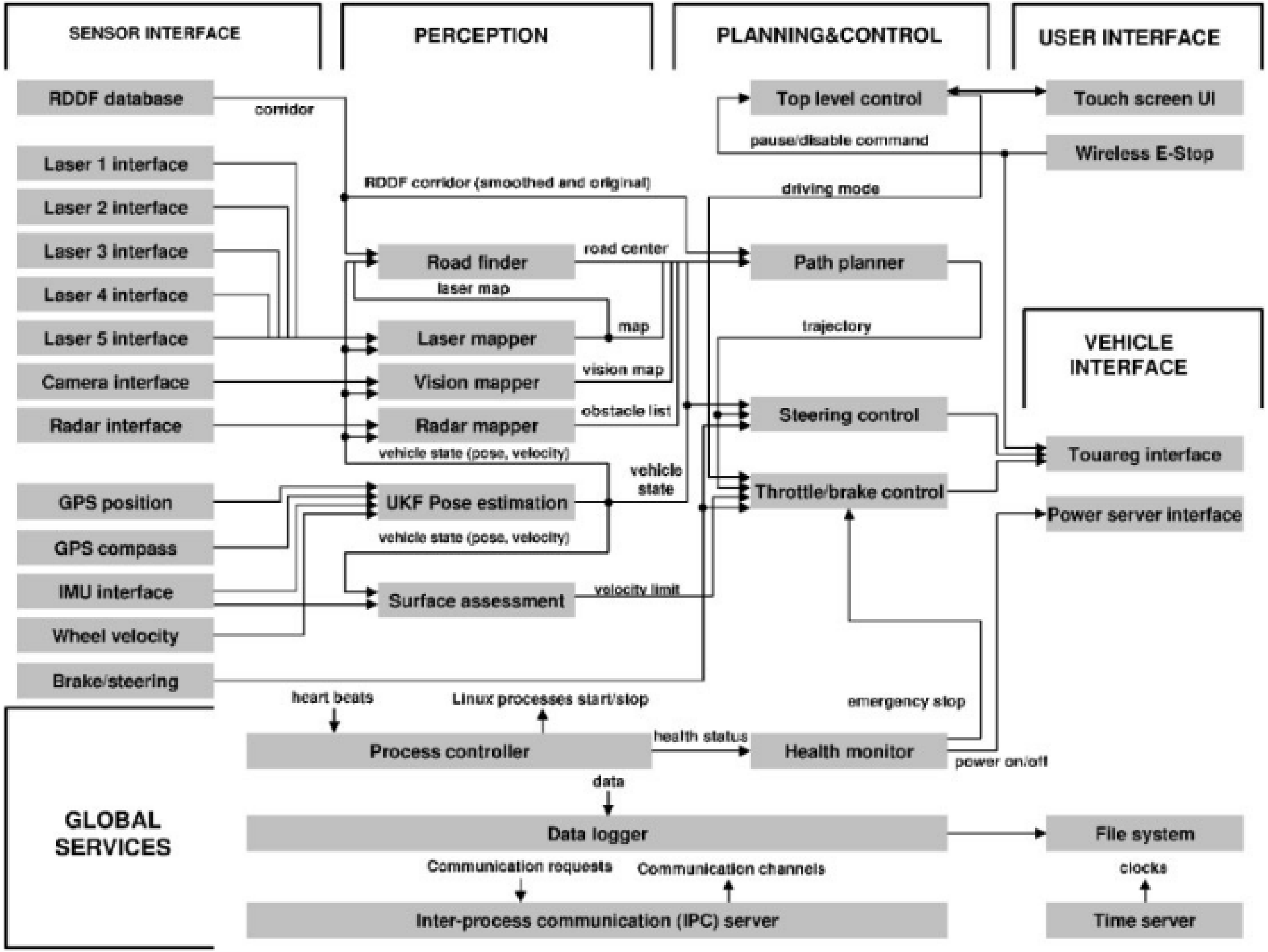
Level 1: Wander

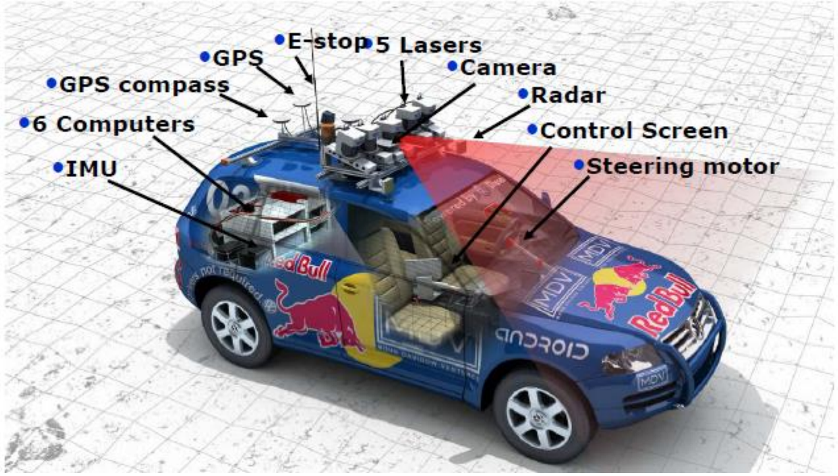


Level 2: Follow Corridor



Flowchart of Stanley software system





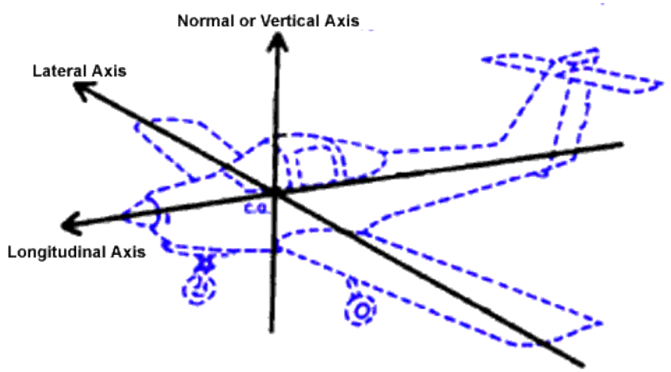
Fixed-Wing UAV

$$\text{Velocity vector, } \mathbf{v} = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \text{forward velocity} \\ \text{sideway velocity} \\ \text{vertical velocity} \\ \text{roll rate} \\ \text{pitch rate} \\ \text{yaw rate} \end{bmatrix}$$

$$\begin{bmatrix} X \\ Y \\ Z \\ L \\ M \\ N \end{bmatrix} = \begin{bmatrix} \text{forward force} \\ \text{sideway force} \\ \text{vertical force} \\ \text{rollmoment} \\ \text{pitchmoment} \\ \text{yawmoment} \end{bmatrix}$$

flat earth. non-rotation mass. aircraft is rigid body. aircraft is symmetric. constant wind. no rotating earth

6DOF



The Axes of an Airplane

- Longitudinal stability derivatives

Stability Derivative, X_u = -6.68
Angle of Attack Derivative, X_w = 4.1754
Elevator Deflection, $X_{\delta e}$ = -0.649
Thrust Deflection ,$X_{\delta T}$= 0
Compressibility Effect Derivative ,
M_u = -0.01376
Dimensional Pitching Moment ,
Derivative, M_w = 0.05852

Pitching moment (Elevator Deflection) ,$M_{\delta e}$ = -1.1526
Dimensionless Pitching Moment Derivative, M_q= -0.1179
Pitching moment (Thrust Deflection) ,$M_{\delta T}$= 0
Pitch Rate Derivative X_q = -1.16
Stability Derivative, Z_u = -0.6276
Angle of Attack Derivative, Z_w =-3.0503
Elevator Deflection ,$Z_{\delta e}$ = 26.0063
Thrust Deflection, $Z_{\delta T}$= 0
Pitch Rate Derivative, Z_q= 9.67

Lateral stability derivatives

Roll Rate ,Y_p = -0.05579
Aileron Deflection Derivative ,$Y_{\delta a}$ = 0
Yaw Rate Derivative, Y_r = 0
Sideslip Derivative Y_β = -4.5129
Rolling Moment , L_p = -0.3295
Rolling Moment L_r ,= 0.0205
Rolling Moment,$L_{\delta a}$ = 3.6299
Roll Acceleration ,L_β = 3.7096
Yawing Moment, $N_{\delta a}$ = 3.0316
Yawing Moment , N_p = 0.02025
Yawing Moment ,N_r = -0.10266
Yaw Acceleration ,N_β = 0.79937

$$\begin{bmatrix} \Delta u' \\ \Delta w' \\ \Delta q' \\ \Delta \theta' \end{bmatrix} = \begin{bmatrix} X_u & Xw & Xq + w_0 & -g\cos\theta_0 \\ Z_u & Zw & Zq + w_0 & -g\sin\theta_0 \\ M_u & Mw & Mq & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta w \\ \Delta q \\ \Delta \theta \end{bmatrix} + \begin{bmatrix} X\delta e & X\delta T \\ Z\delta e & Z\delta T \\ M\delta e & M\delta T \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \delta_e \\ \Delta \delta T \end{bmatrix}$$

$$\begin{bmatrix} \Delta\beta \\ \Delta p \\ \Delta\phi \\ \Delta r \end{bmatrix} = \begin{bmatrix} -0.2051 & -0.05579 & -21.9543 & 32.174 \\ -0.1686 & -0.3295 & 0.0205 & 0 \\ 0.03633 & 0.02025 & -0.10266 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\beta \\ \Delta p \\ \Delta\phi \\ \Delta r \end{bmatrix} + \begin{bmatrix} 0 \\ 3.6299 \\ 3.0316 \\ 0 \end{bmatrix} [\Delta\delta a]$$

$$\begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \text{forward velocity} \\ \text{sideway velocity} \\ \text{vertical velocity} \\ \text{roll rate} \\ \text{pitch rate} \\ \text{yaw rate} \end{bmatrix}$$

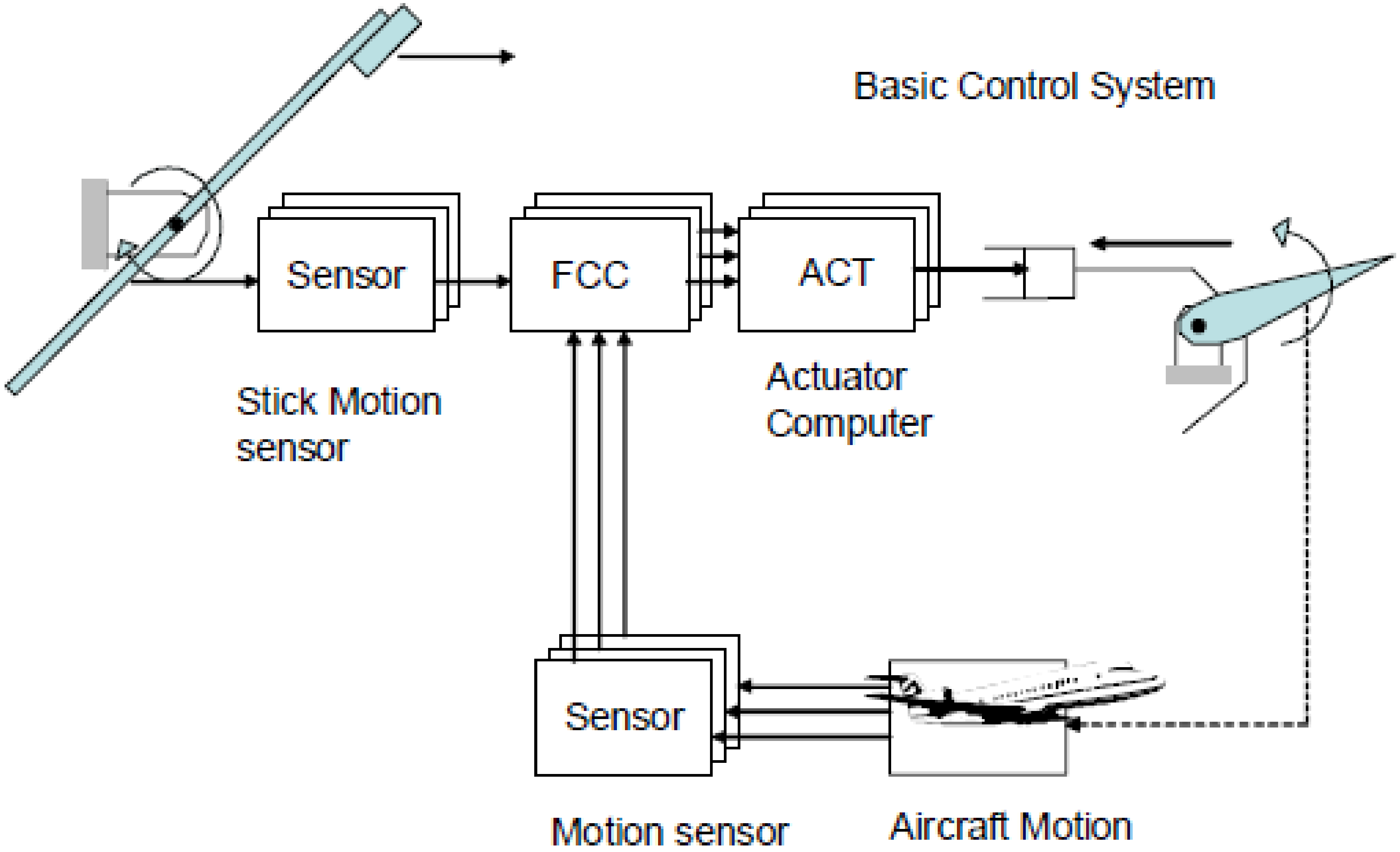
C_F \ ξ	Longitudinal				Lateral - Directional				
	u	α	q	δ_e	β	p	r	δ_a	δ_r
C_L	C_{Lu}	$C_{L\alpha}$	C_{Lq}	$C_{L\delta e}$					
C_D	C_{Du}	$C_{D\alpha}$	0	$C_{D\delta e}$					
C_M	C_{Mu}	$C_{M\alpha}$	C_{Mq}	$C_{M\delta e}$					
C_Y					$C_{Y\beta}$	C_{Yp}	C_{Yr}	$C_{Y\delta a}$	$C_{Y\delta r}$
C_l					$C_{l\beta}$	C_{lp}	C_{lr}	$C_{l\delta a}$	$C_{l\delta r}$
C_N					$C_{N\beta}$	C_{Np}	C_{Nr}	$C_{N\delta a}$	$C_{N\delta r}$

Control Derivative Coefficient

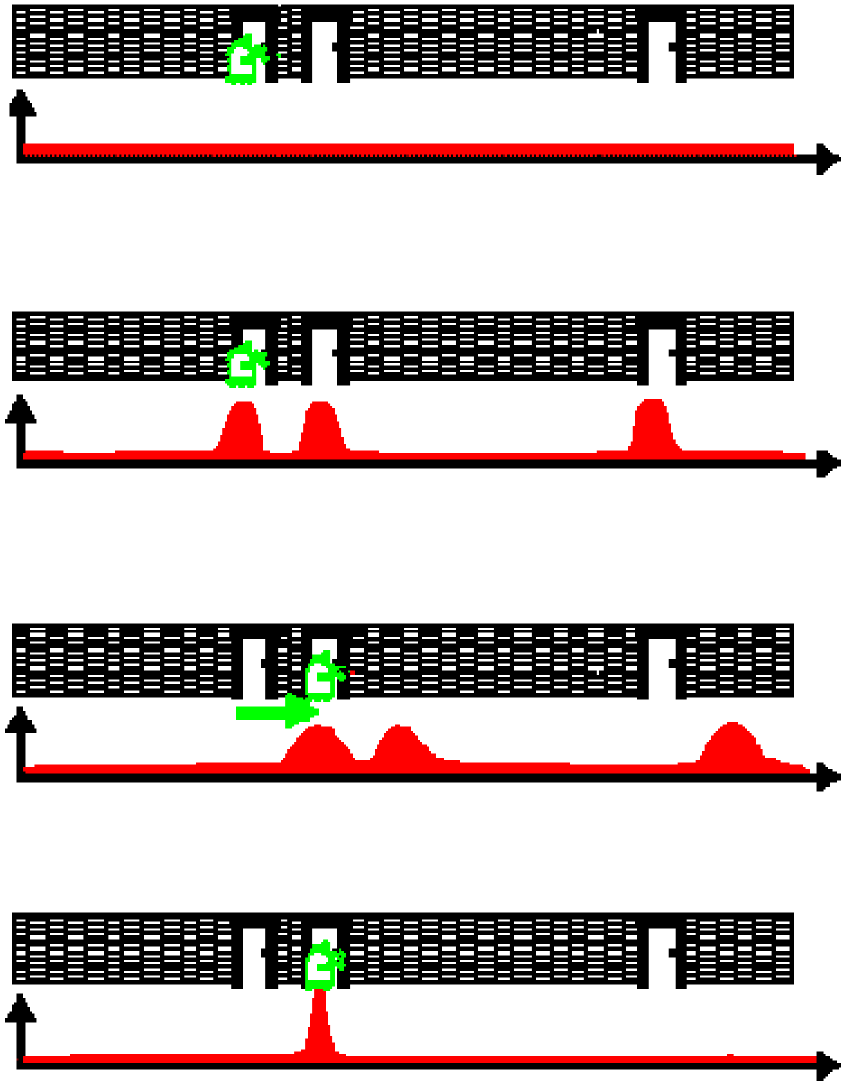
Static Stability Derivatives Coefficient For speed deviation

Dynamic Stability Derivative Coefficient

Static Stability Derivative Coefficient for angular deviation


Basic Control System
Sensor
FCC
ACT
Stick Motion sensor
Actuator Computer
Sensor
Motion sensor
Aircraft Motion

Probabilistic Robotics



Bayes Formula

$$P(x,y) = P(x \mid y)P(y) = P(y \mid x)P(x)$$

$$\Rightarrow$$

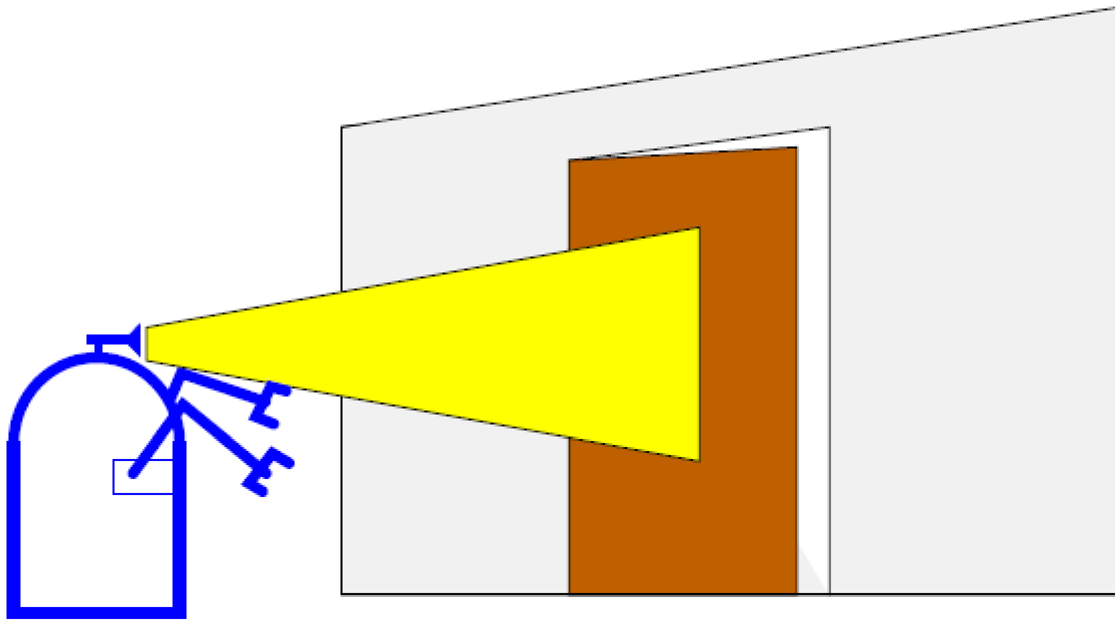
$$\boxed{P(x \mid y) = \frac{P(y \mid x)\ P(x)}{P(y)} = \frac{\text{likelihood} \cdot \text{prior}}{\text{evidence}}}$$

Normalization

$$P(x \mid y) = \frac{P(y \mid x)\ P(x)}{P(y)} = \eta\ P(y \mid x)\, P(x)$$

Example of state Estimation

- Suppose a robot obtains measurement z
- What is $P(open|z)$?



Causal vs Diagnostic Reasoning

- $P(open|z)$ is diagnostic.
- $P(z|open)$ is causal.
- Often causal knowledge is easier to obtain.

  •count frequencies!

- Bayes rule allows us to use causal knowledge:

$$P(open \mid z) = \frac{P(z \mid open)P(open)}{P(z)}$$

Example

- $P(z|open) = 0.6$ $P(z|\neg open) = 0.3$
- $P(open) = P(\neg open) = 0.5$

$$P(open \mid z) = \frac{P(z \mid open)P(open)}{P(z \mid open)\,p(open) + P(z \mid \neg open)\,p(\neg open)}$$

$$P(open \mid z) = \frac{0.6 \cdot 0.5}{0.6 \cdot 0.5 + 0.3 \cdot 0.5} = \frac{2}{3} = 0.67$$

- **z raises the probability that the door is open.**

Combining Evidence

- Suppose our robot obtains another observation z_2.
- How can we integrate this new information?
- More generally, how can we estimate $P(x| z_1 \ldots z_n)$?

Recursive Bayesian Updating

$$P(x \mid z_1, \ldots, z_n) = \frac{P(z_n \mid x, z_1, \ldots, z_{n-1})\, P(x \mid z_1, \ldots, z_{n-1})}{P(z_n \mid z_1, \ldots, z_{n-1})}$$

- **Markov assumption: z_n is independent of $z_1, \ldots, z_{n-1}$ if we know x.**

$$P(x \mid z_1, \ldots, z_n) = \frac{P(z_n \mid x)\, P(x \mid z_1, \ldots, z_{n-1})}{P(z_n \mid z_1, \ldots, z_{n-1})}$$

$$= \eta\, P(z_n \mid x)\, P(x \mid z_1, \ldots, z_{n-1})$$

$$= \eta_{1\ldots n} \left[\prod_{i=1\ldots n} P(z_i \mid x) \right] P(x)$$

Example – Second Measurement

- $P(z_2|open) = 0.5$ $\quad$ $P(z_2|\neg open) = 0.6$
- $P(open|z_1)=2/3$

$$P(open \mid z_2, z_1) = \frac{P(z_2 \mid open)\, P(open \mid z_1)}{P(z_2 \mid open)\, P(open \mid z_1) + P(z_2 \mid \neg open)\, P(\neg open \mid z_1)}$$

$$= \frac{\frac{1}{2} \cdot \frac{2}{3}}{\frac{1}{2} \cdot \frac{2}{3} + \frac{3}{5} \cdot \frac{1}{3}} = \frac{5}{8} = 0.625$$

- **z_2 lowers the probability that the door is open.**

Actions

- Often the world is **dynamic** since
  - **actions carried out by the robot**,
  - **actions carried out by other agents**,
  - or just the **time** passing by

  change the world.

- How can we **incorporate** such **actions**?

Typical Actions

- The robot **turns its wheels** to move
- The robot **uses its manipulator** to grasp an object
- Plants grow over **time**…

- Actions are **never carried out with absolute certainty**.
- In contrast to measurements, **actions generally increase the uncertainty**.

Modeling Actions

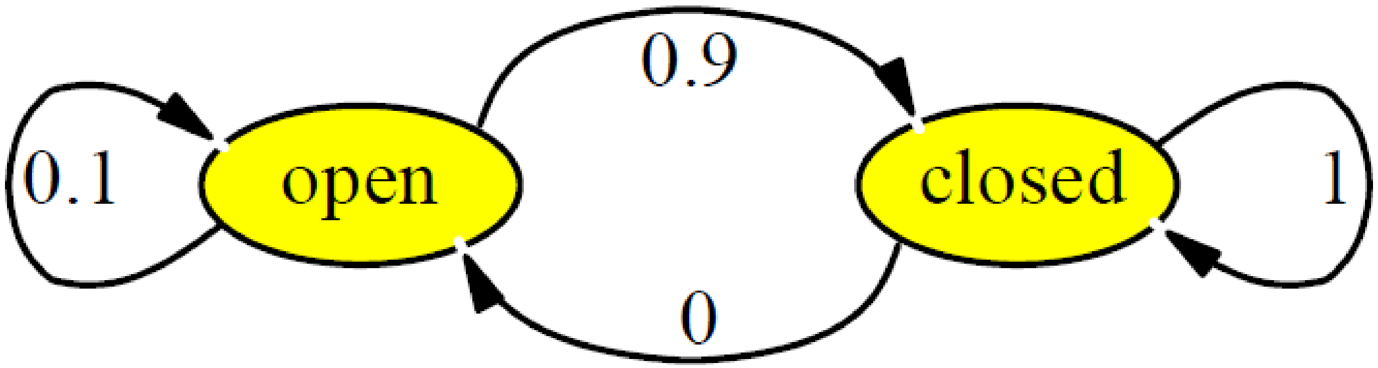
- To incorporate the outcome of an action u into the current "belief", we use the conditional pdf

$$P(x|u,x')$$

- This term specifies the pdf that **executing u changes the state from x' to x.**

State Transitions

$P(x|u,x')$ for u = "close door":



If the door is open, the action "close door" succeeds in 90% of all cases.

Integrating the Outcome of Action

Continuous case:

$$P(x \mid u) = \int P(x \mid u, x')P(x')dx'$$

Discrete case:

$$P(x \mid u) = \sum P(x \mid u, x')P(x')$$

Example : The Resulting Belief

$$
\begin{aligned}
P(closed \mid u) &= \sum P(closed \mid u, x')P(x') \\
&= P(closed \mid u, open)P(open) \\
&\quad + P(closed \mid u, closed)P(closed) \\
&= \frac{9}{10} * \frac{5}{8} + \frac{1}{1} * \frac{3}{8} = \frac{15}{16}
\end{aligned}
$$

$$
\begin{aligned}
P(open \mid u) &= \sum P(open \mid u, x')P(x') \\
&= P(open \mid u, open)P(open) \\
&\quad + P(open \mid u, closed)P(closed) \\
&= \frac{1}{10} * \frac{5}{8} + \frac{0}{1} * \frac{3}{8} = \frac{1}{16} \\
&= 1 - P(closed \mid u)
\end{aligned}
$$

Bayes Filter-Framework

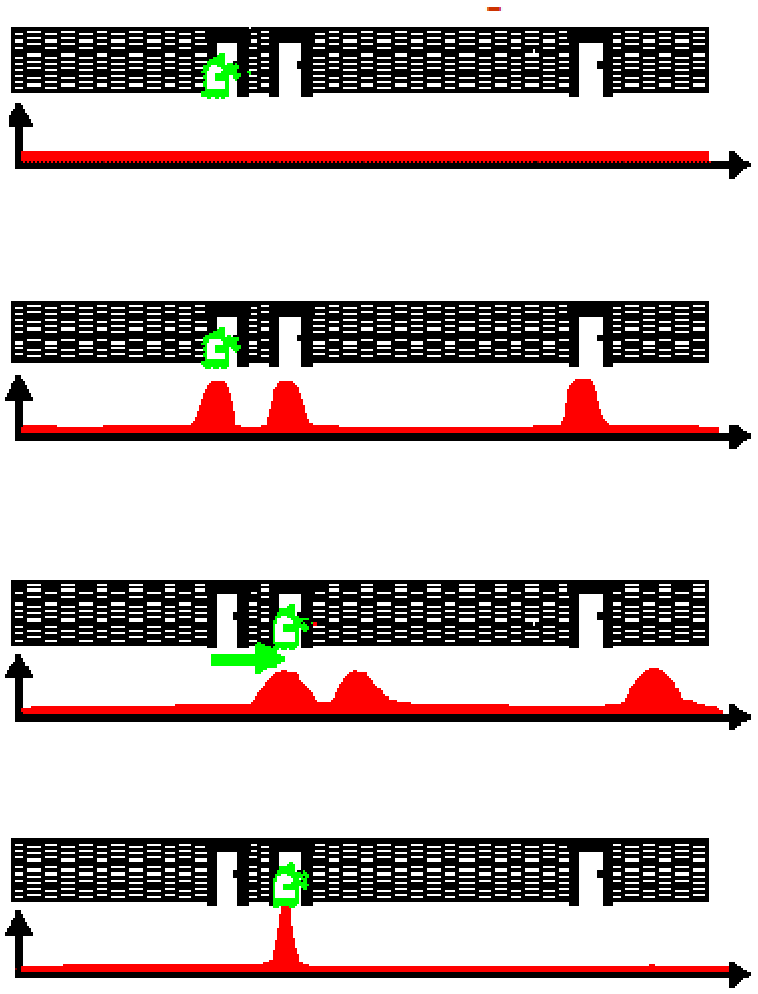
- **Given:**
  - Stream of observations *z* and action data *u:*

$$d_t = \{u_1, z_1 \ldots, u_t, z_t\}$$

  - Sensor model *P(z|x).*
  - Action model *P(x|u,x').*
  - Prior probability of the system state *P(x).*
- **Wanted:**
  - Estimate of the state *X* of a dynamical system.
  - The posterior of the state is also called **Belief**:

$$Bel(x_t) = P(x_t \mid u_1, z_1 \ldots, u_t, z_t)$$

Bayes Filter Example



Bayes Filter Cont..

- z = observation
- u = action
- x = state

$$\boxed{Bel(x_t)} = P(x_t \mid u_1, z_1 \ldots, u_t, z_t)$$

•Bayes $\quad = \eta \ P(z_t \mid x_t, u_1, z_1, \ldots, u_t) \ P(x_t \mid u_1, z_1, \ldots, u_t)$

•Markov $\quad = \eta \ P(z_t \mid x_t) \ P(x_t \mid u_1, z_1, \ldots, u_t)$

•Total prob. $\quad = \eta \ P(z_t \mid x_t) \int P(x_t \mid u_1, z_1, \ldots, u_t, x_{t-1}) \ P(x_{t-1} \mid u_1, z_1, \ldots, u_t) \ dx_{t-1}$

•Markov $\quad = \eta \ P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \ P(x_{t-1} \mid u_1, z_1, \ldots, u_t) \ dx_{t-1}$

•Markov $\quad = \eta P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) P(x_{t-1} \mid u_1, z_1, \ldots, z_{t-1}) \ dx_{t-1}$

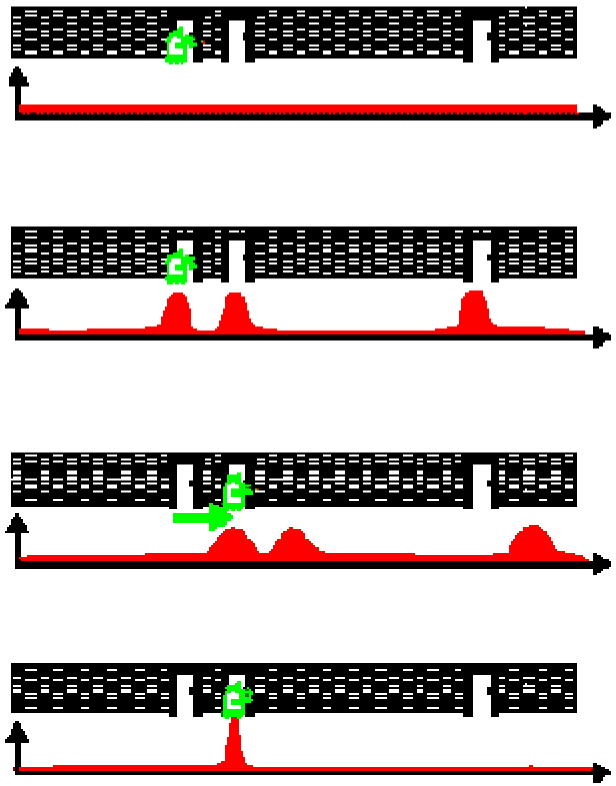
$$\boxed{= \eta \ P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \ Bel(x_{t-1}) \ dx_{t-1}}$$

Bayes Filters are Familiar

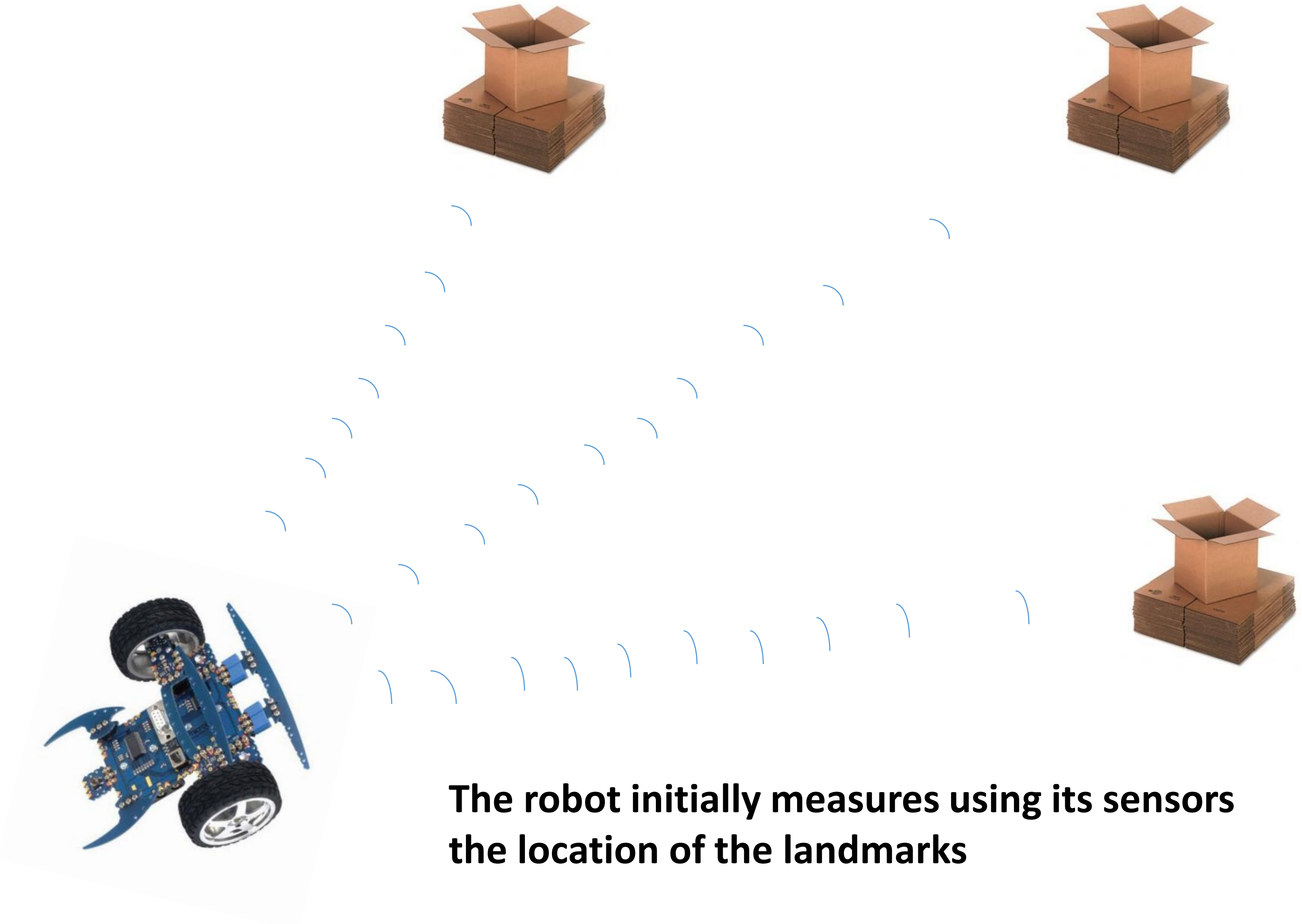
$$Bel(x_t) = \eta\ P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1})\ Bel(x_{t-1})\ dx_{t-1}$$

- Kalman filters
- Particle filters
- Hidden Markov models
- Dynamic Bayesian networks
- Partially Observable Markov Decision Processes (POMDPs)

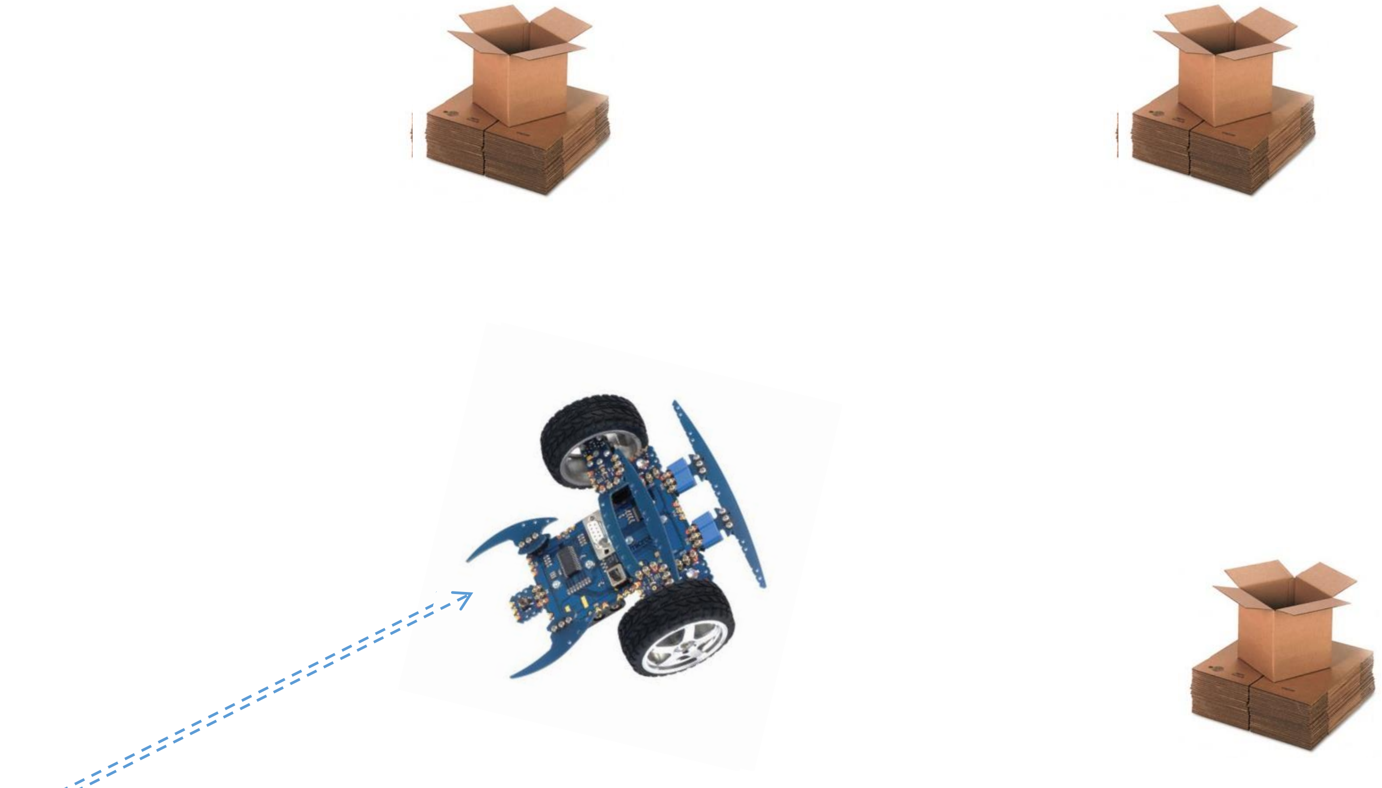
Bayes Filter Locaization



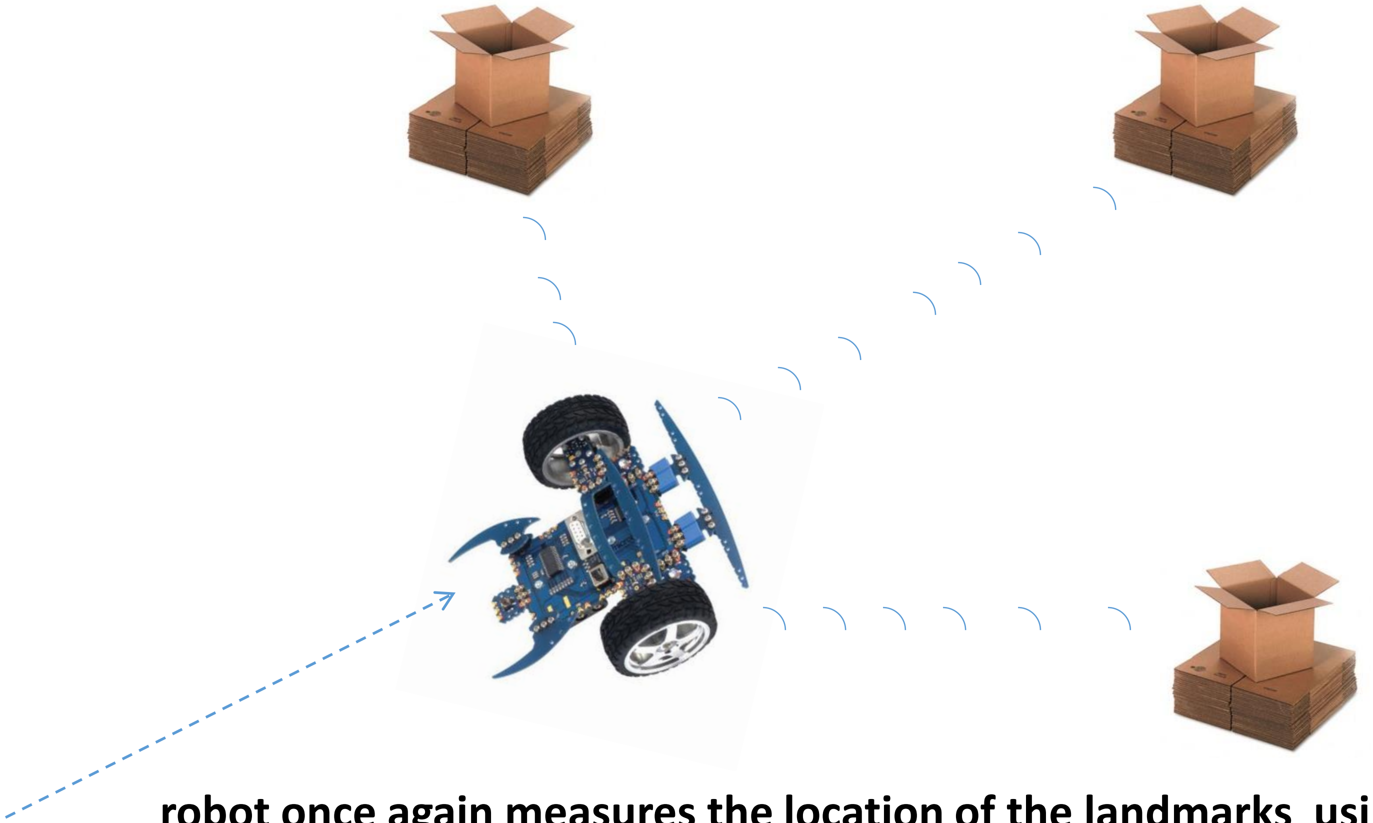
$$Bel(x_t) = \eta \; P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \; Bel(x_{t-1}) \; dx_{t-1}$$



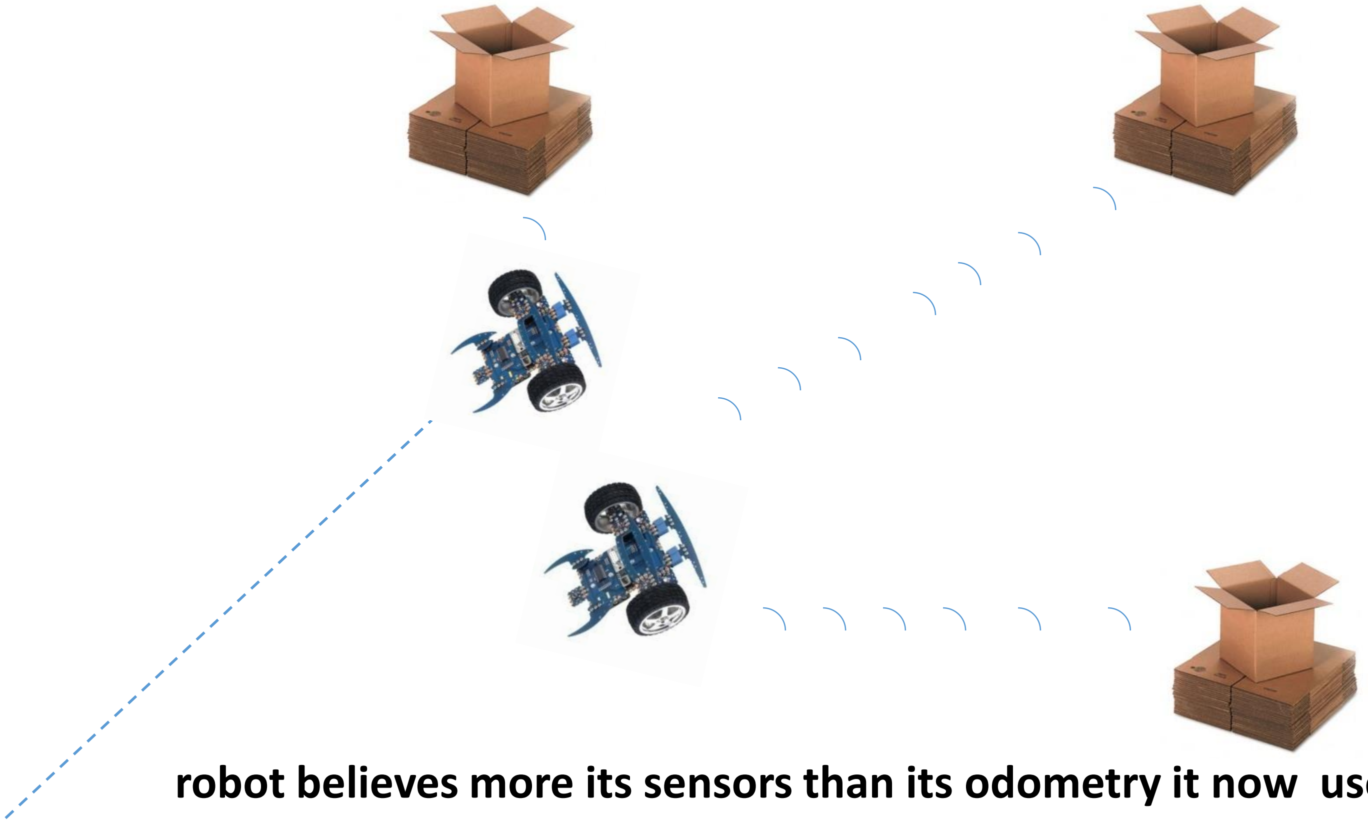
The robot initially measures using its sensors the location of the landmarks



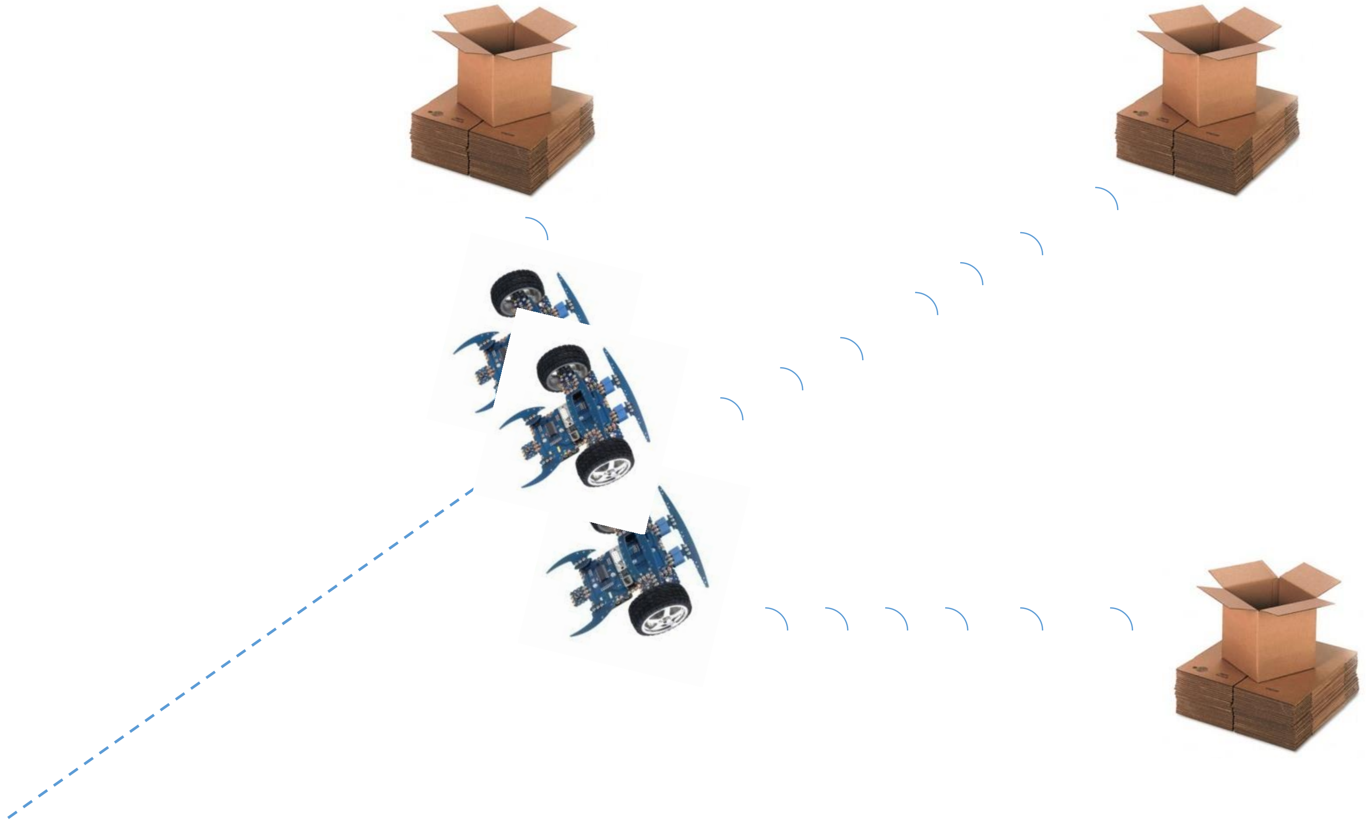
The robot moves so it now thinks it is here. The distance moved is given by the robots odometry.



robot once again measures the location of the landmarks using its sensors but finds out they don’t match where robot thinks they should be. Robot is not where it thinks it is.



robot believes more its sensors than its odometry it now uses the information gained about where the landmarks actually are to determine where it is



Probabilistic Sensor Models and Motion Model

Bayes Filters

Sensing: p(z | x)

Motion step/Prediction step
Measurement step

•z = observation
•u = action
•x = state

Action

$$Bel(x_t) = \eta \, P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \, Bel(x_{t-1}) \, dx_{t-1}$$

S xt-1	S xt	..	..	..	...	..	S xtm

Range sensors

- Contact sensors: Bumpers, Whiskers
- Internal sensors
  - Accelerometers (spring-mounted masses)
  - Gyroscopes (spinning mass, laser light)
  - Compasses, inclinometers (earth magnetic field, gravity)
- Proximity sensors
  - Sonar (time of flight)
  - Radar (phase and frequency)
  - Laser range-finders (triangulation, tof, phase)
  - Infrared (intensity)
  - Stereo
- Visual sensors: Cameras, Stereo
- Satellite-based sensors: GPS

Ultrasound Sensor

- Emit an ultrasound signal
- Wait until they receive the ecl
- Time of flight sensor

Polaroyd 6500

Time of Flight sensor

$$d = v \times t / 2$$

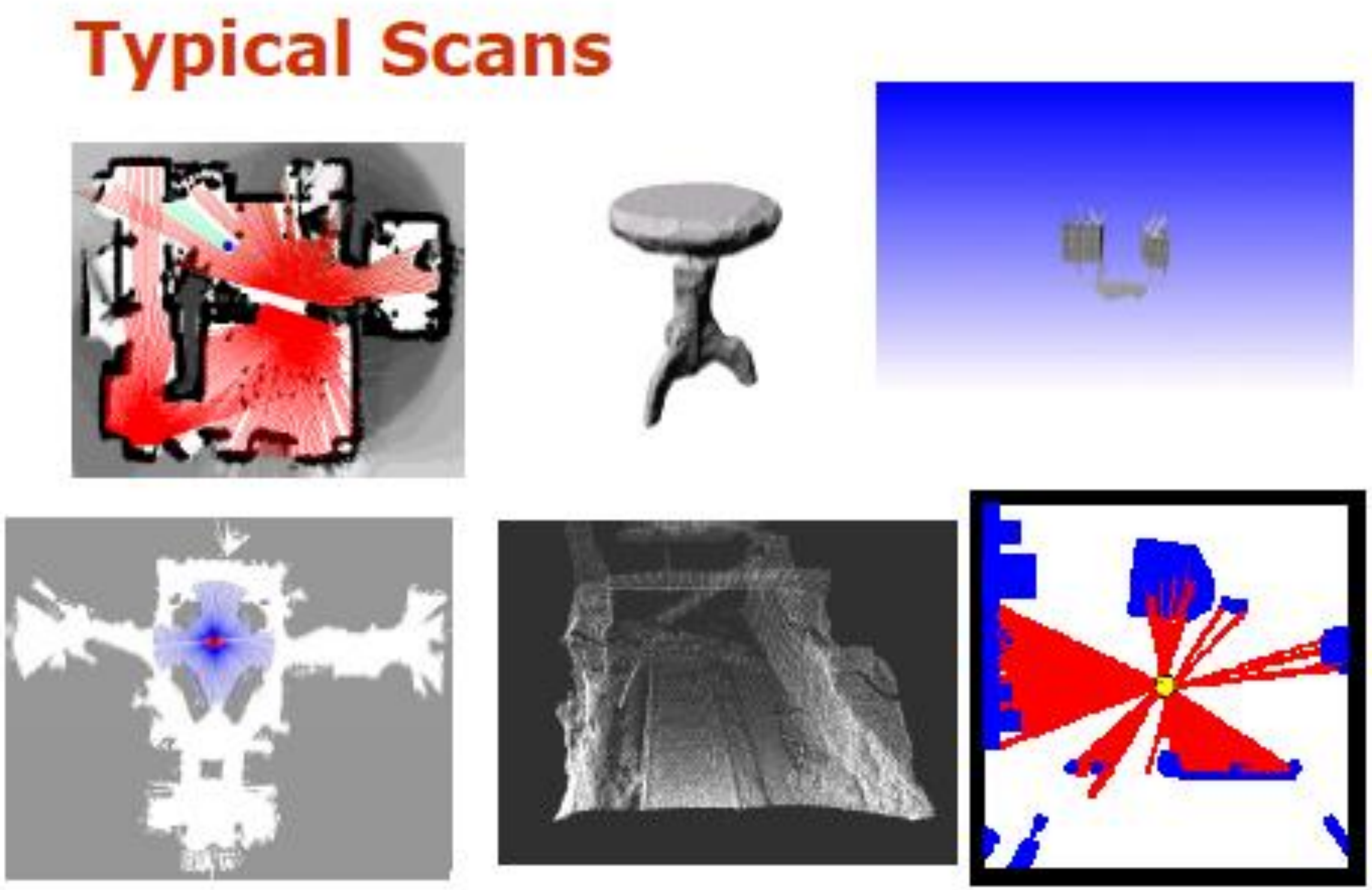
v: speed of the signal

t: time elapsed between broadcast of signal and reception of the echo.

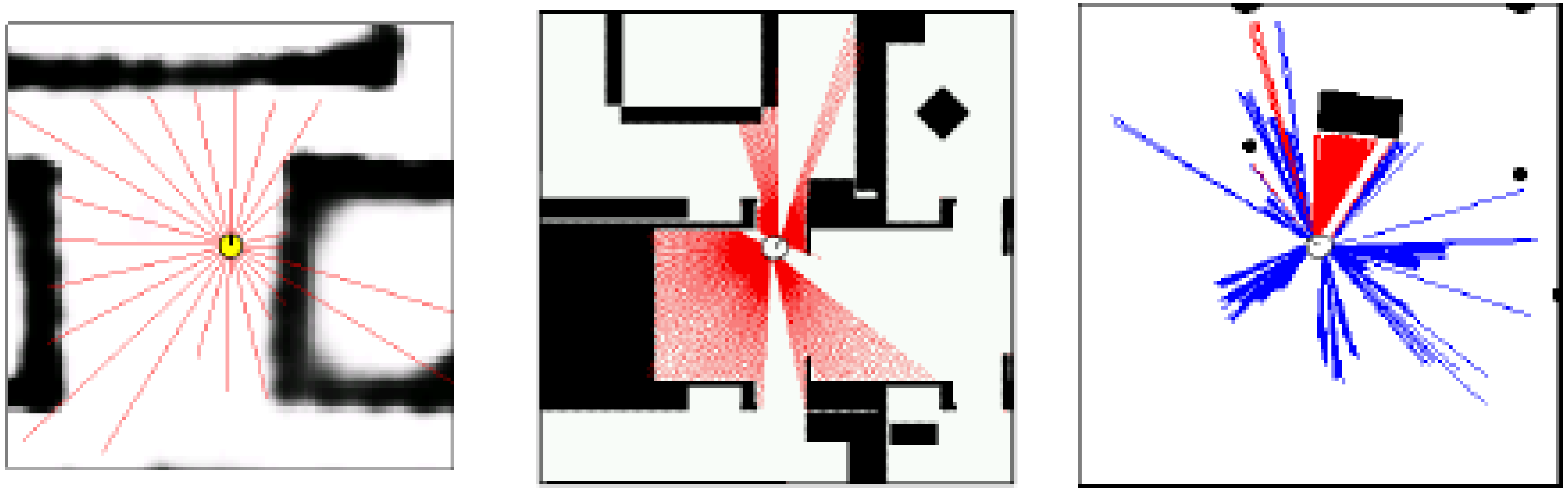
Laser Range Scanner

- High precision
- Wide field of view
- Approved security for collision detection

Typical Scans



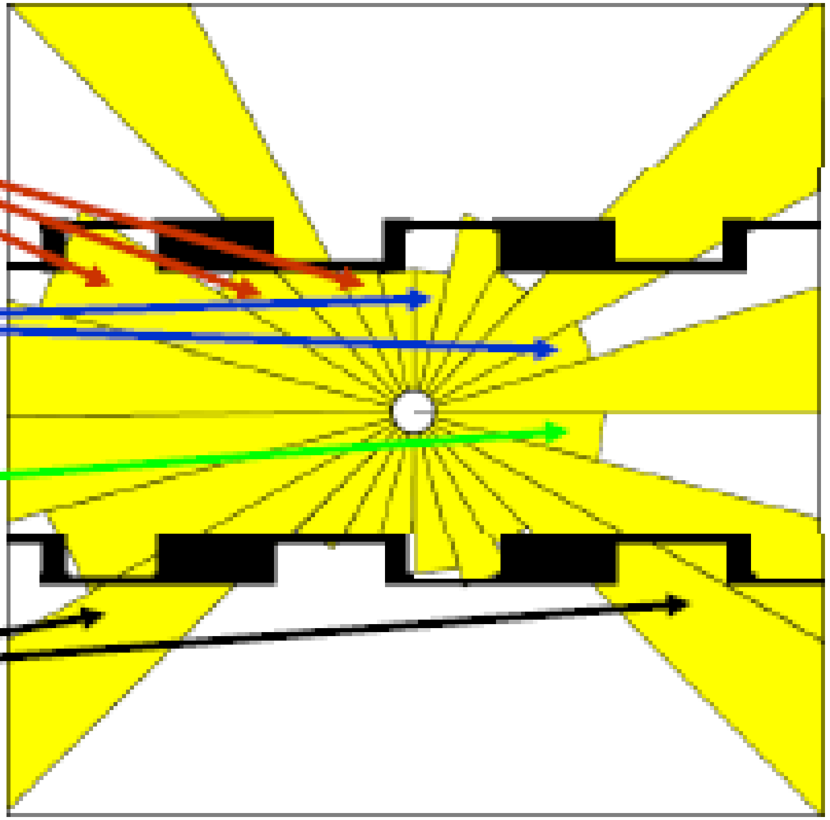
Proximity Sensors



- The central task is to determine $P(z|x)$, i.e., the probability of a measurement z given that the robot is at position x.
- **Question**: Where do the probabilities come from?

Typical Measurement Errors of an Range Measurements

1. Beams reflected by obstacles
2. Beams reflected by persons / caused by crosstalk
3. Random measurements
4. Maximum range measurements

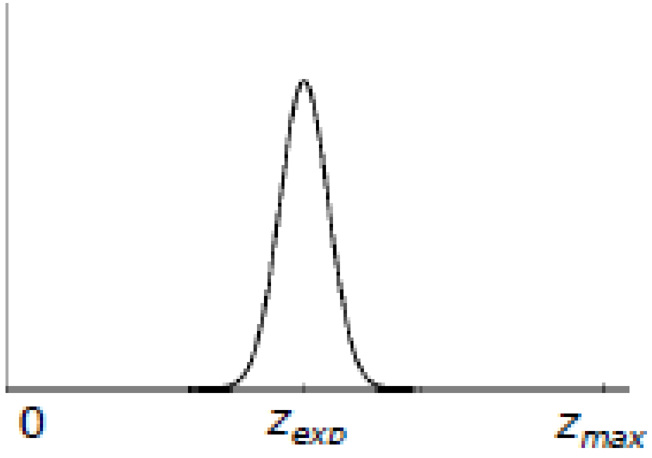


Proximity Measurement

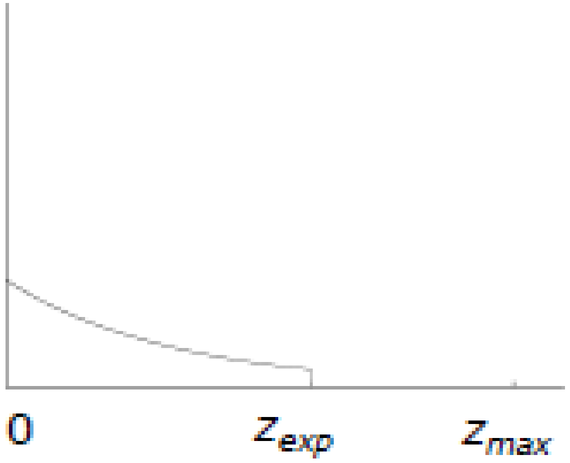
- Measurement can be caused by ...
  - a known obstacle.
  - cross-talk.
  - an unexpected obstacle (people, furniture, ...).
  - missing all obstacles (total reflection, glass, ...).
- Noise is due to uncertainty ...
  - in measuring distance to known obstacle.
  - in position of known obstacles.
  - in position of additional obstacles.
  - whether obstacle is missed.

Beam based Proximity Model

Measurement noise



Unexpected obstacles

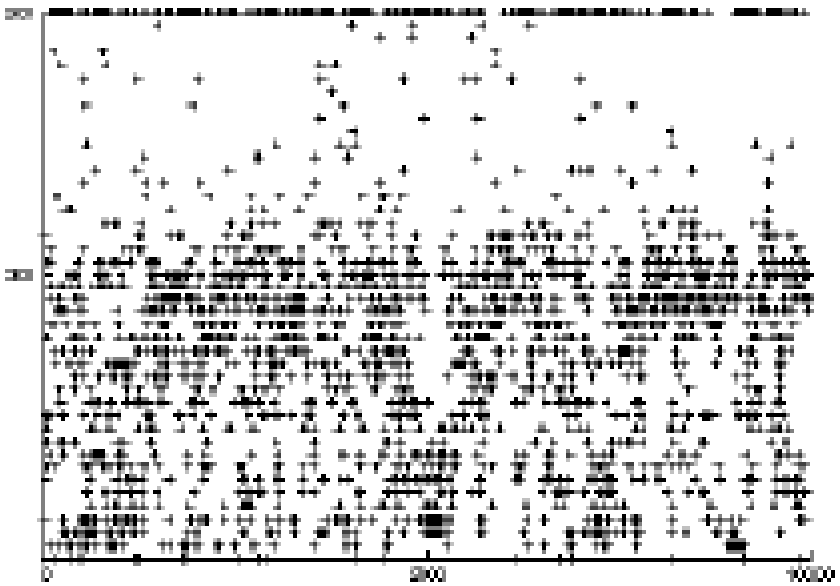


$$P_{hit}(z \mid x, m) = \eta \frac{1}{\sqrt{2\pi b}} e^{-\frac{1}{2}\frac{(z-z_{exp})^2}{b}}$$

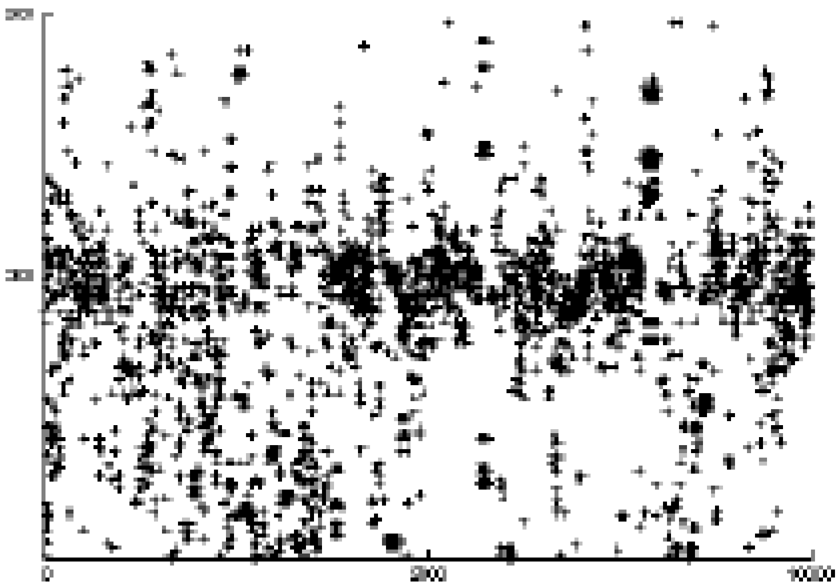
$$P_{unexp}(z \mid x, m) = \begin{cases} \eta \, \lambda \, e^{-\lambda z} & z < z_{exp} \\ 0 & otherwise \end{cases}$$

Raw Sensor Data

Measured distances for expected distance of 300 cm.



Sonar



Laser

Probabilistic Motion Model

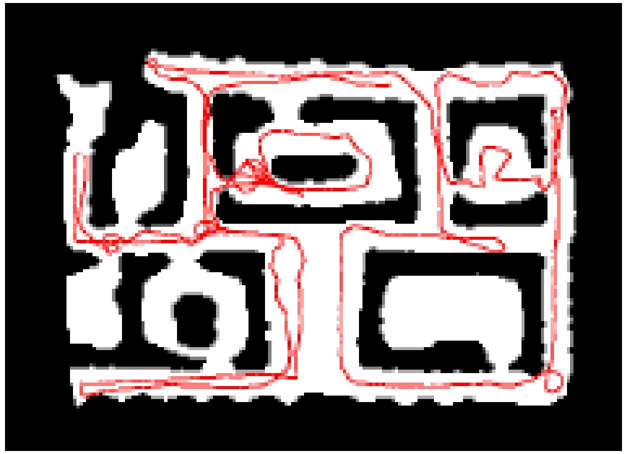
Action: p(x' | u, x)

$$Bel(x_t) = \eta \; P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \, Bel(x_{t-1}) \, dx_{t-1}$$

Robot Motion

Robot motion is inherently uncertain.

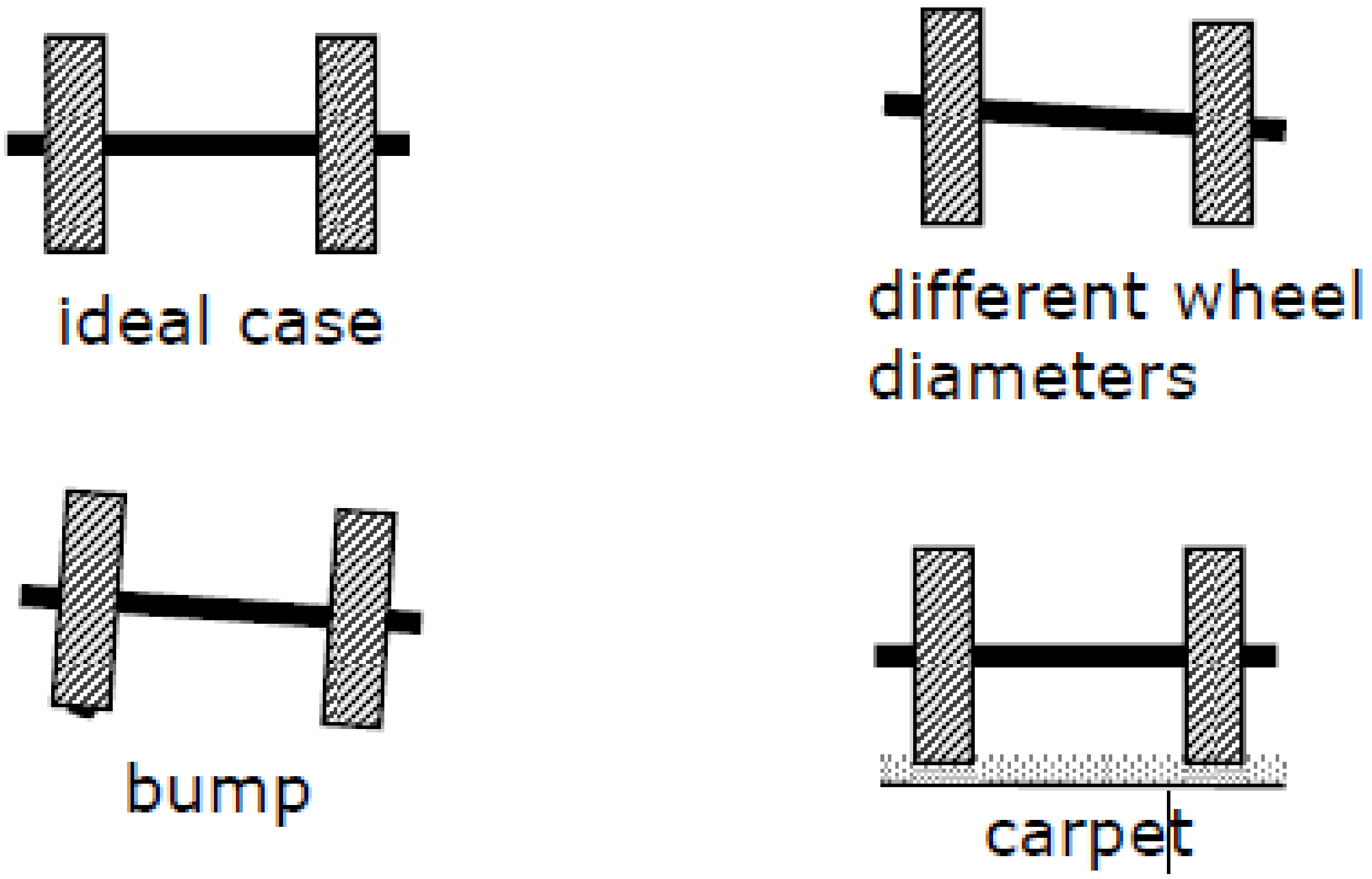
How can we model this uncertainty?



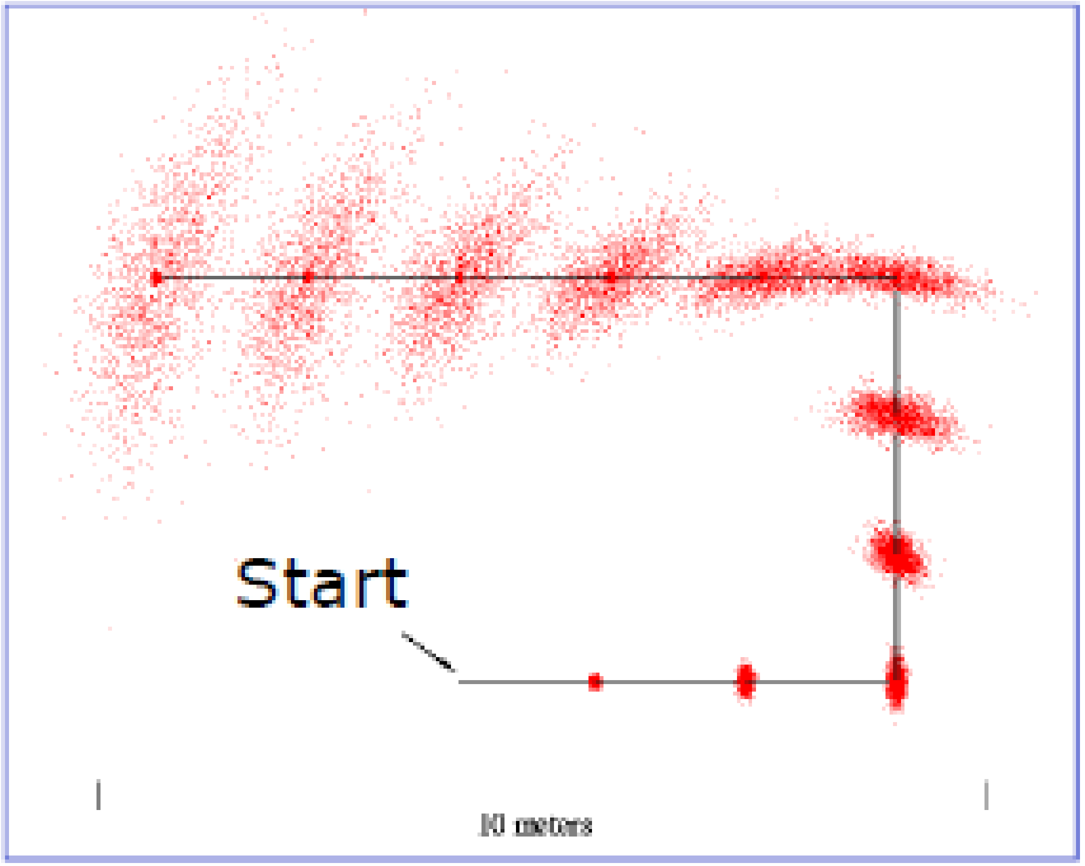
Motion Model

- To implement the Bayes Filter, we need the transition model $p(x \mid x', u)$.
- The term $p(x \mid x', u)$ specifies a posterior probability, that action u carries the robot from x' to x.
- In this section we will specify, how $p(x \mid x', u)$ can be modeled based on the motion equations.

Reason for motion error



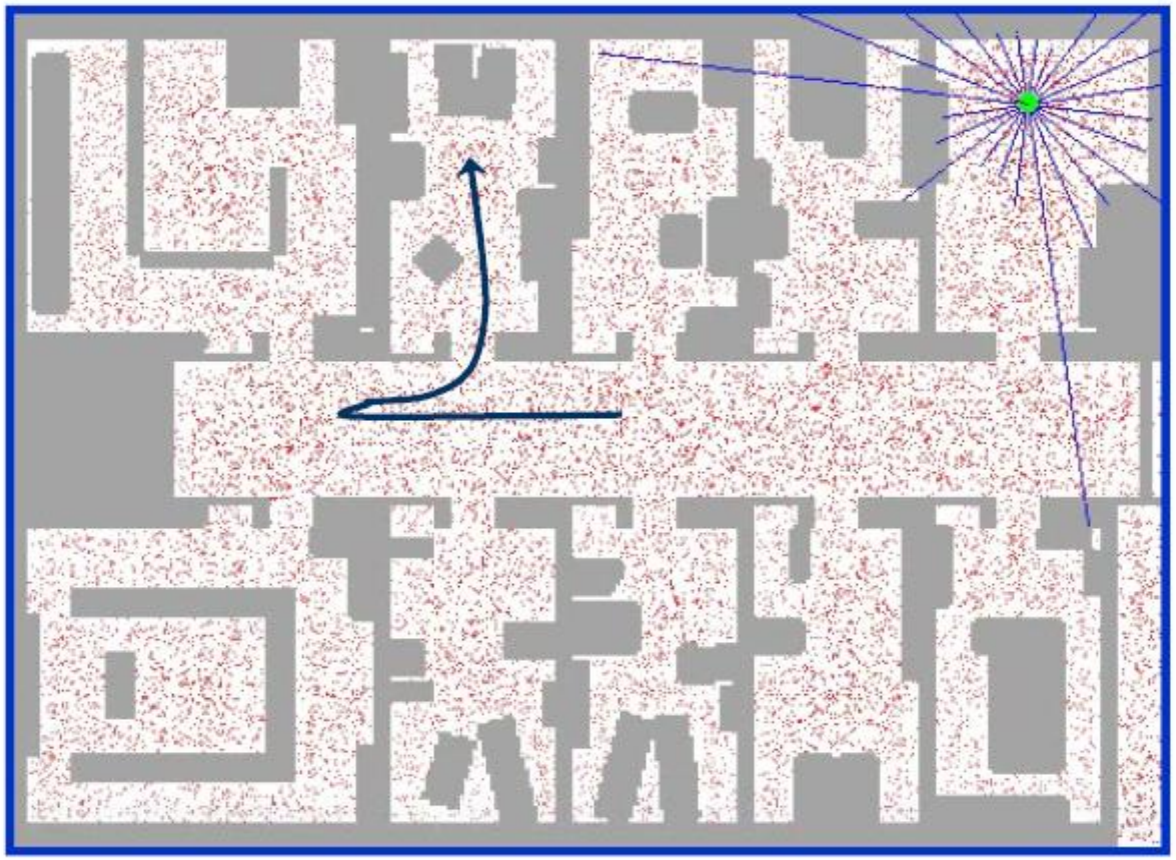
Sampling from our motion model



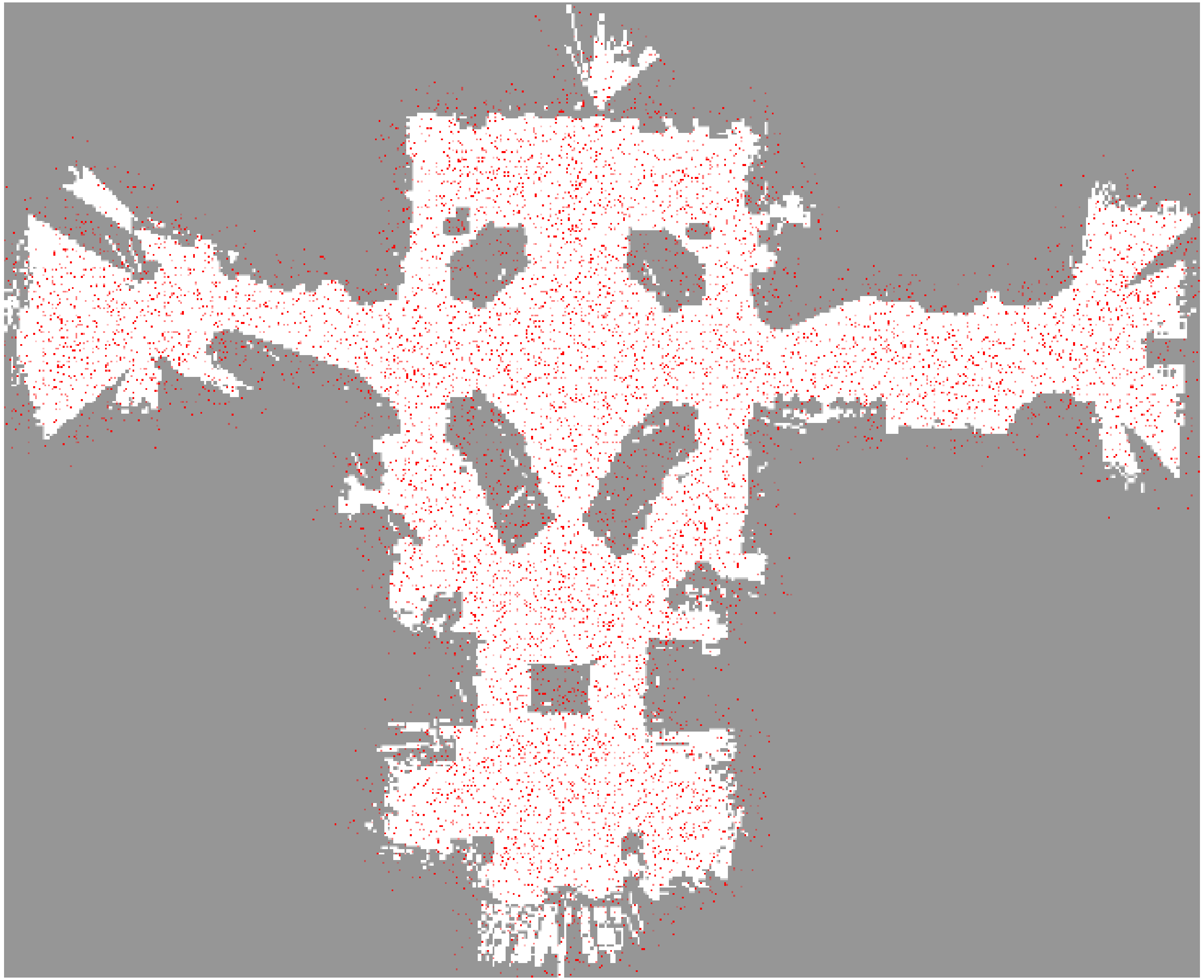
Monte Carlo Localization

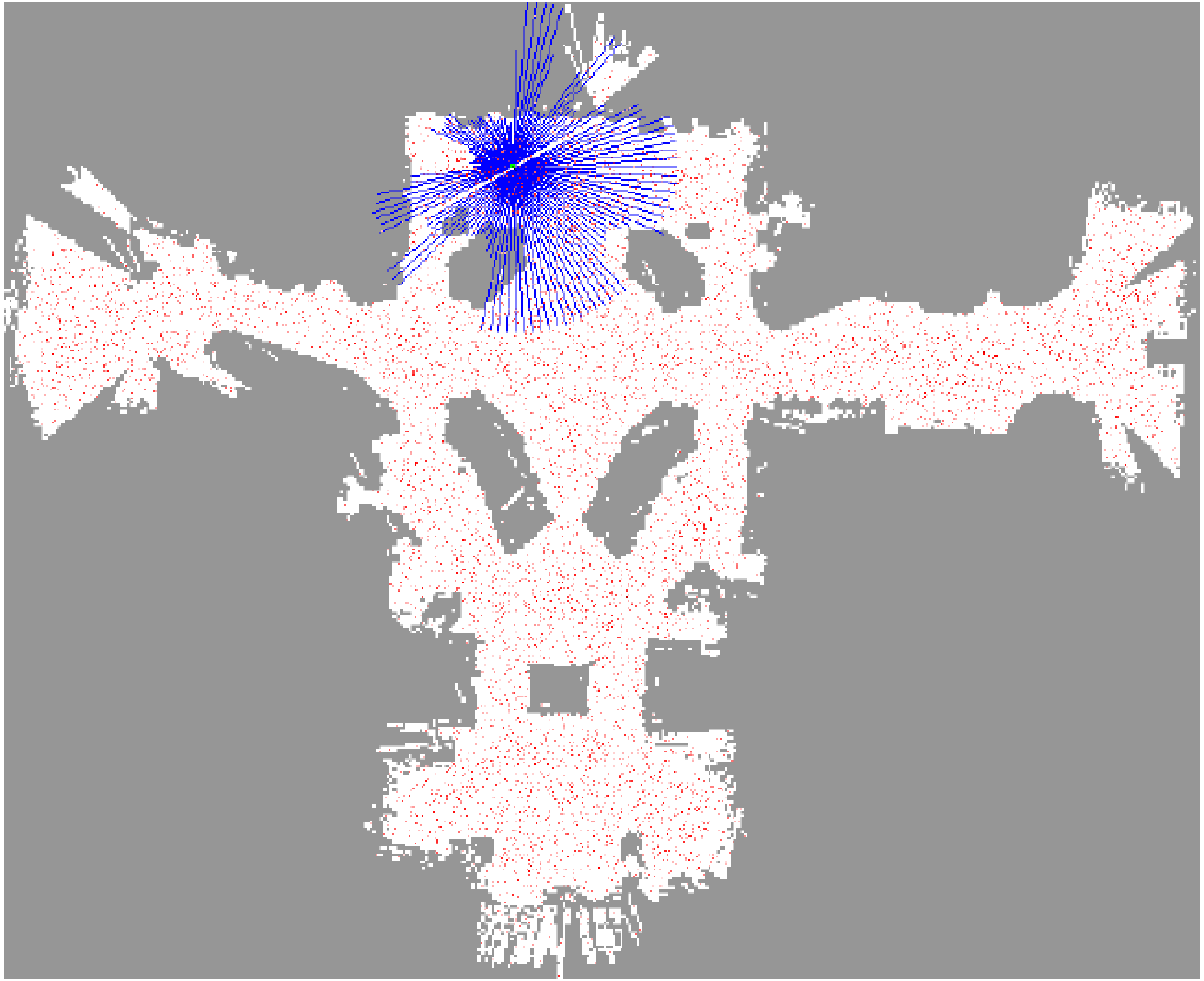
- Particle Filter
  - Recall: Discrete filter
    - Discretize the continuous state space
    - High memory complexity
    - Fixed resolution (does not adapt to the belief)
  - Particle filters are a way to **efficiently** represent **non-Gaussian distribution** **Used for localization**
  - Basic principle were the robot might be-together we approximate the belief
    - Set of state hypotheses ("particles")
    - Survival-of-the-fittest

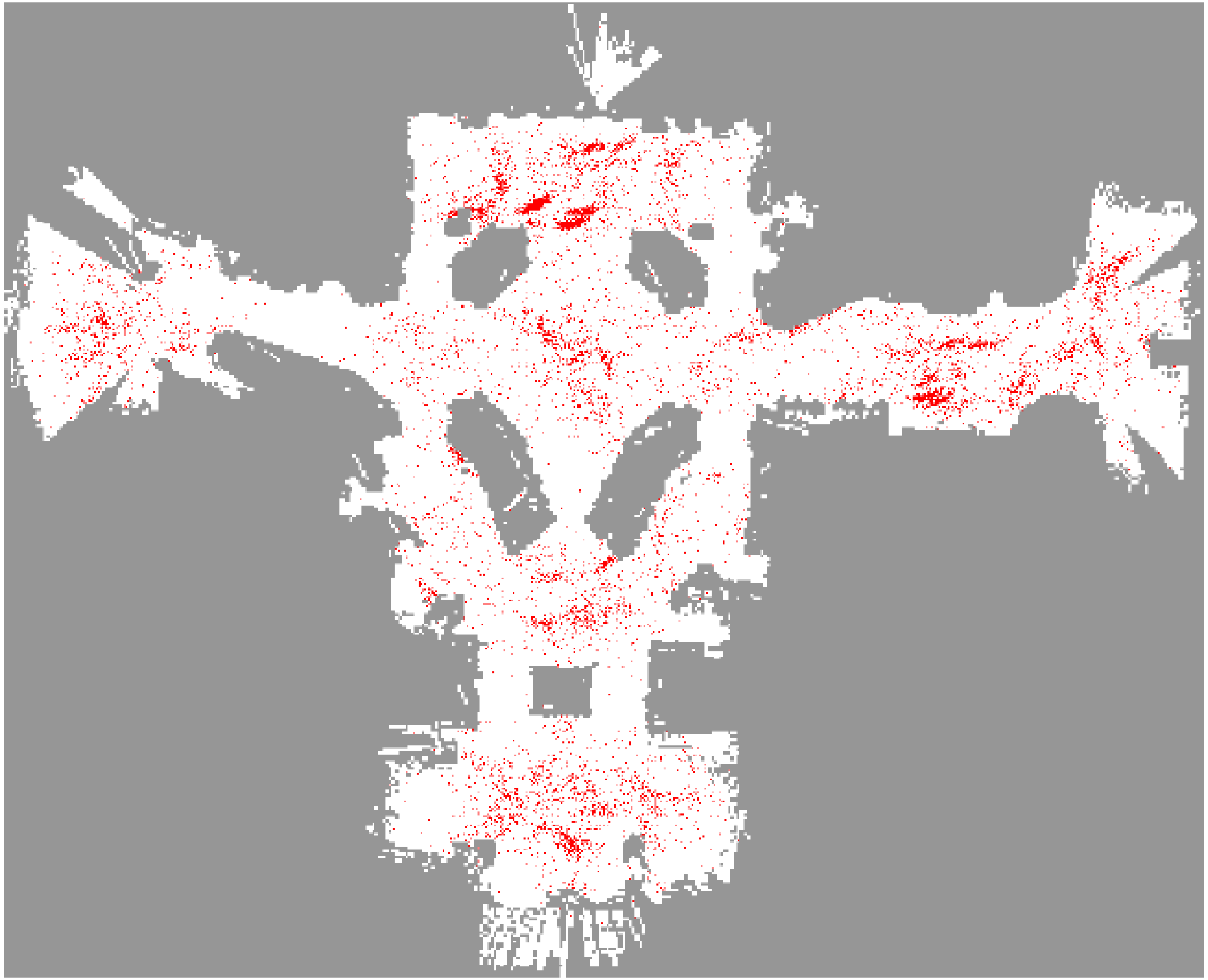
Sample based Localization

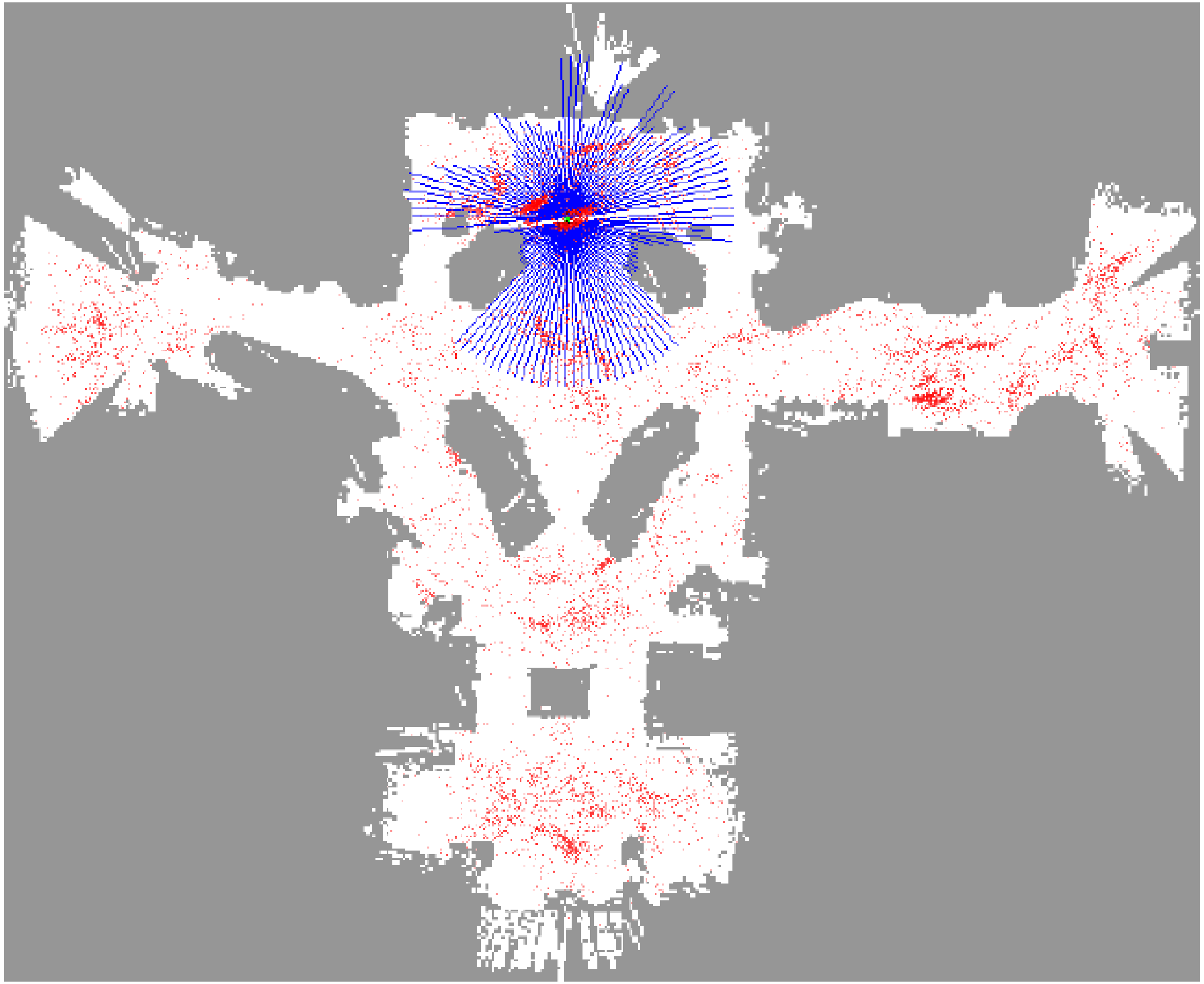


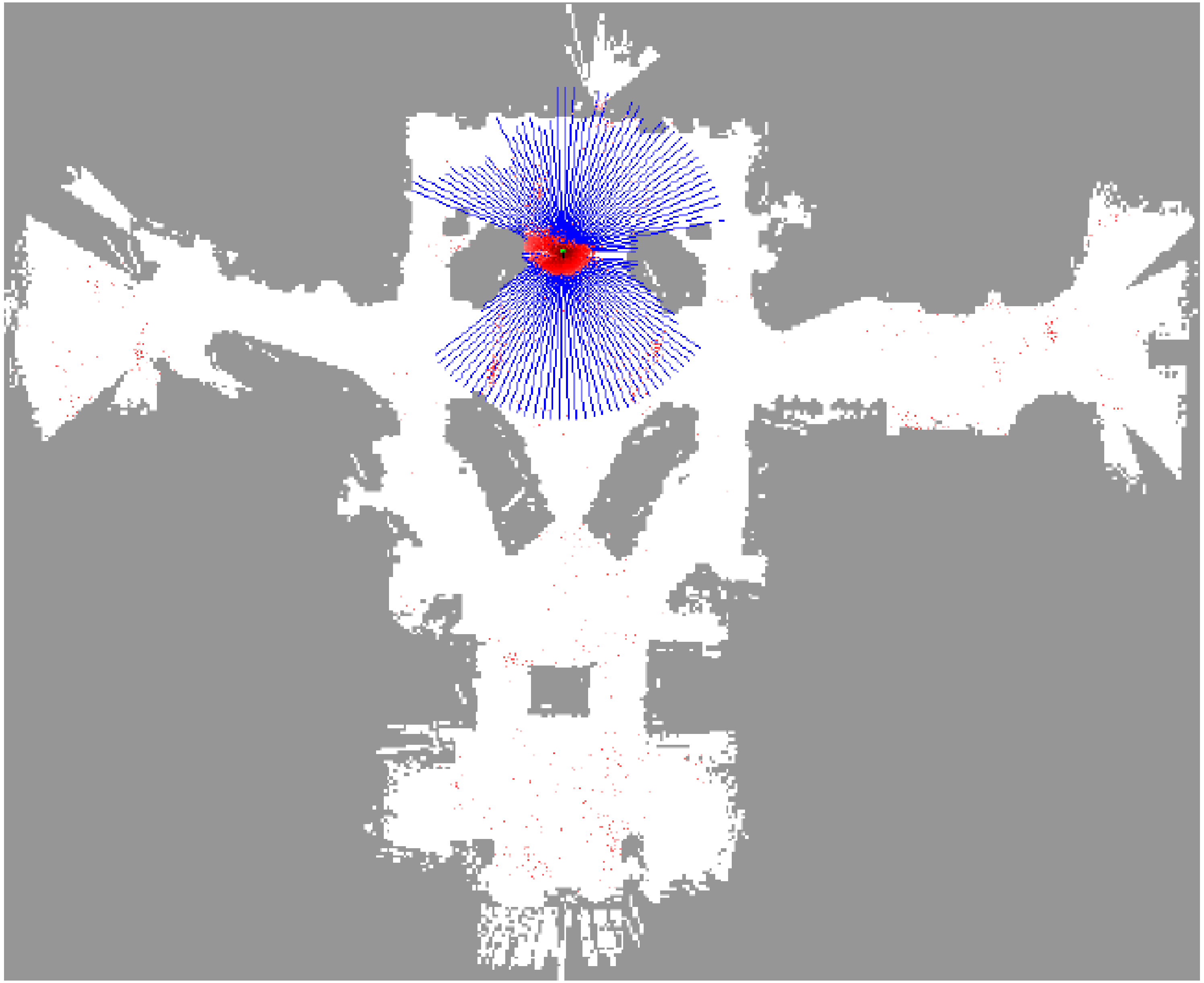
- Particle filter two parts –take a measurement
  - each particle compute how likely are these measurements given the particles are the correct one.
  - Particle with consistent are having high likelihood
- particle survive randomly

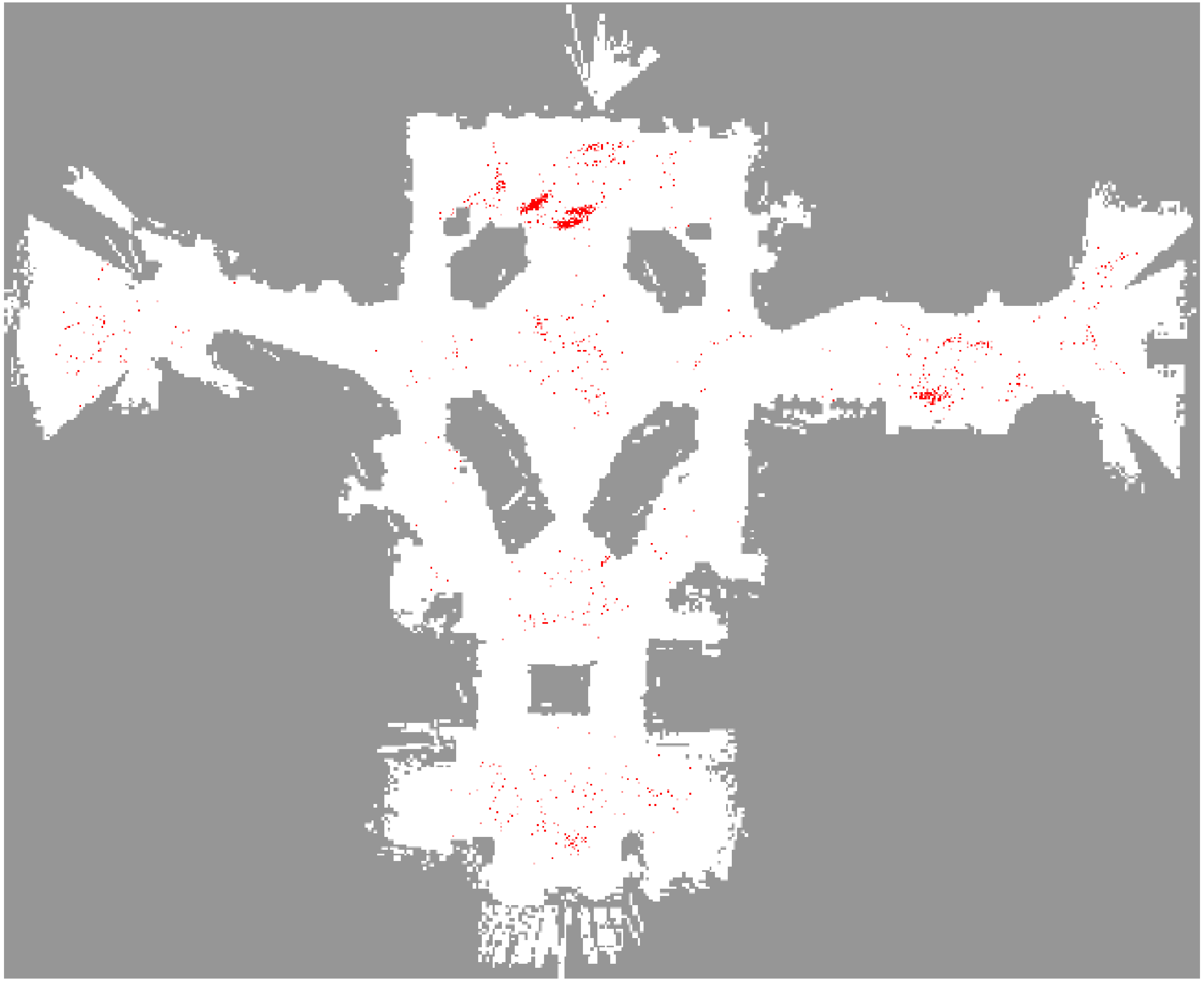


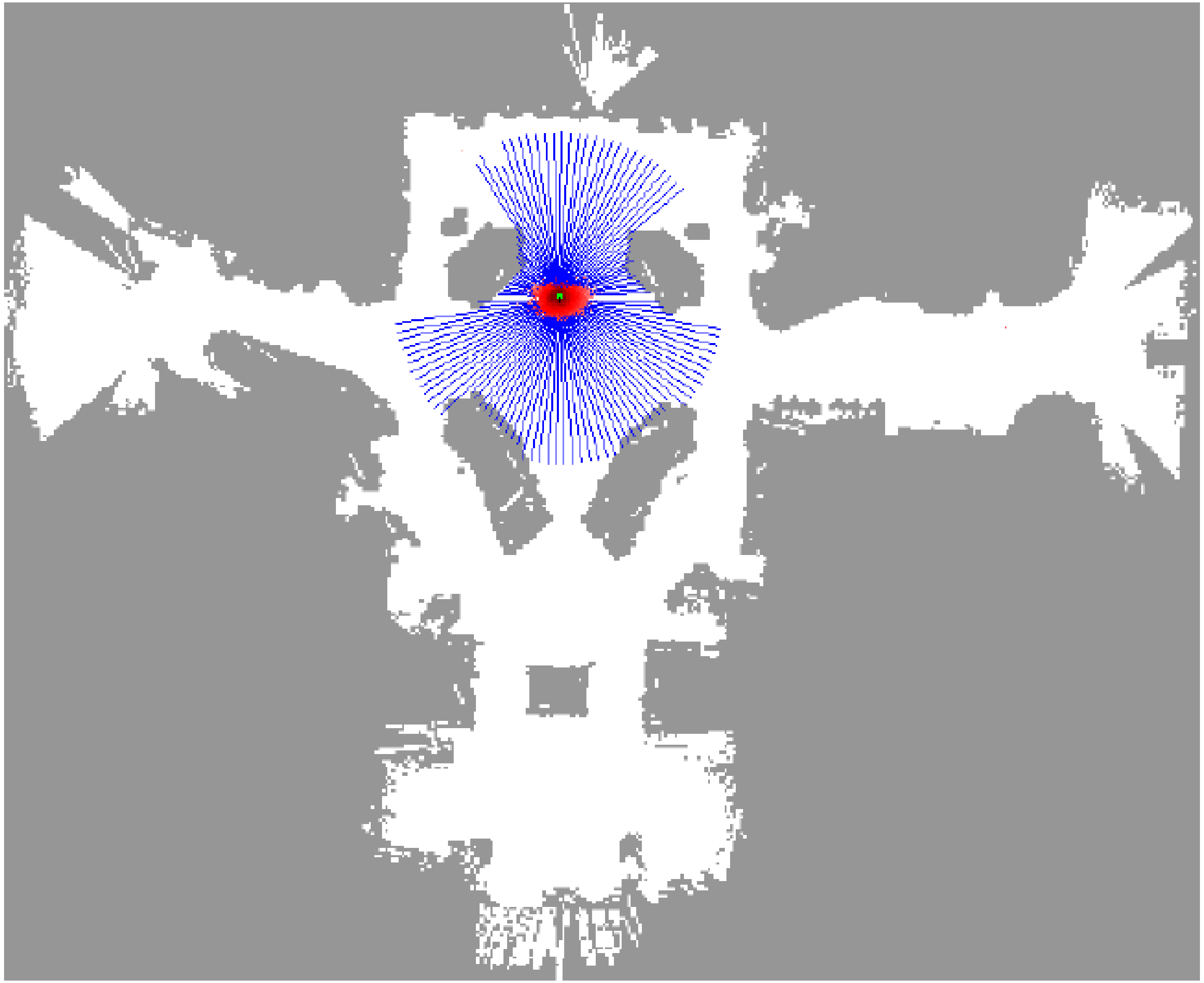












Mathematical Description

- Set of weighted samples

$$S = \left\{ \left\langle s^{[i]}, w^{[i]} \right\rangle \mid i = 1, \ldots, N \right\}$$

State hypothesis Importance weight Survival weight

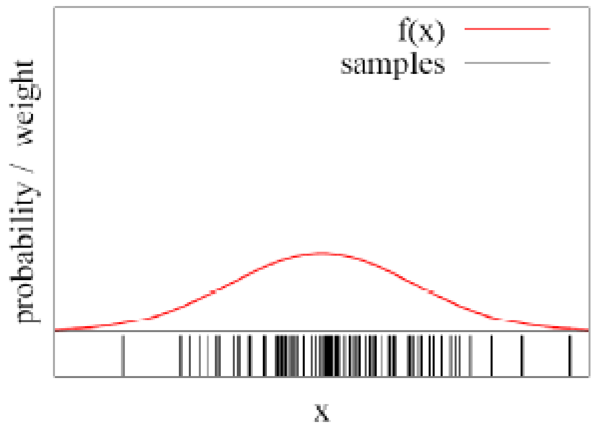
- The samples represent the posterior

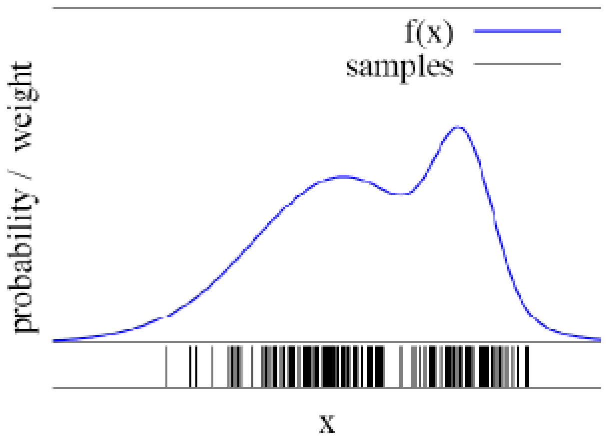
$$p(x) = \sum_{i=1}^{N} w_i \cdot \delta_{s^{[i]}}(x)$$

P(x) distribution of every particle

Function approximation

- Particle sets can be used to approximate functions

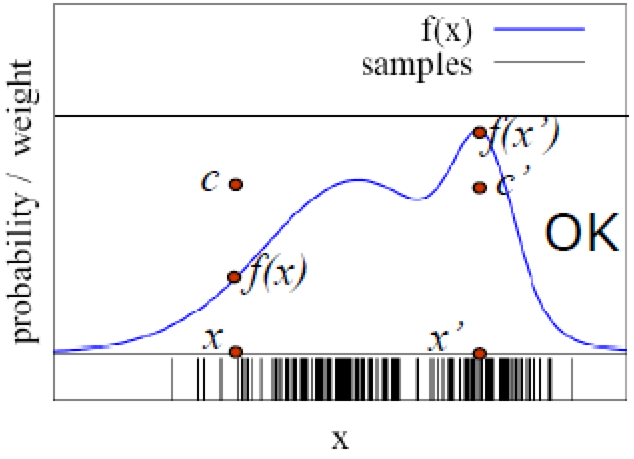




- The more particles fall into an interval, the higher the probability of that interval
- How to draw samples form a function/distribution?

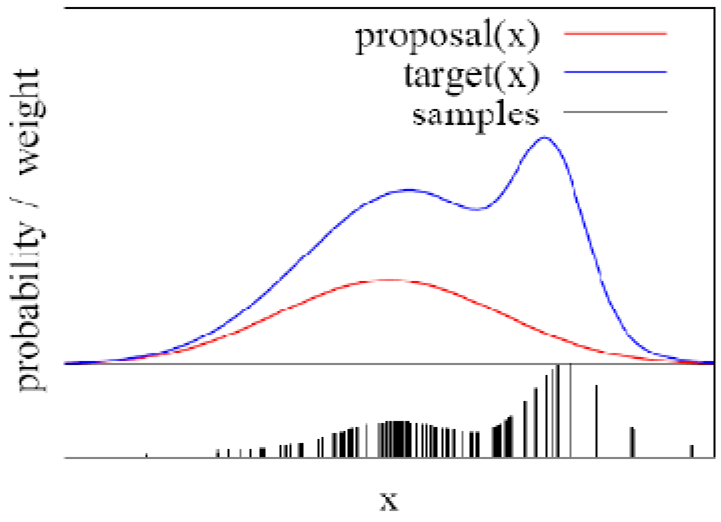
Rejection Sampling

- Let us assume that $f(x)<1$ for all x
- Sample x from a uniform distribution
- Sample c from [0,1]
- if $f(x) > c$ keep the sample
  otherwise reject the sampe



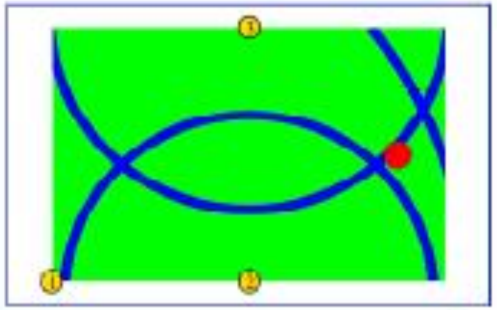
Importance Sampling Principle

- We can even use a different distribution g to generate samples from f
- By introducing an importance weight w, we can account for the "differences between g and f"
- $w = f / g$
- f is often called target
- g is often called proposal
- Pre-condition: $f(x)>0 \rightarrow g(x)>0$

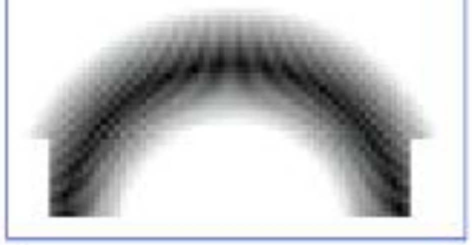


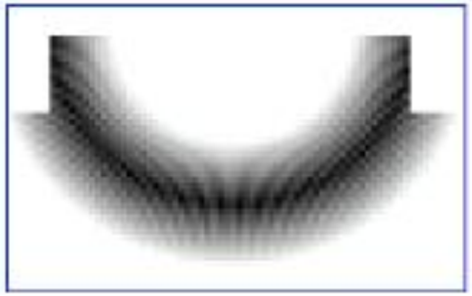
Importance Sampling with Resampling
Landmark Detection Example

Distributions

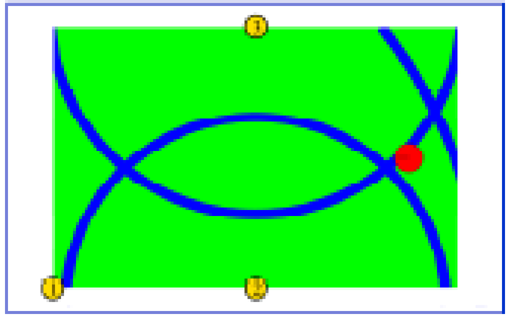


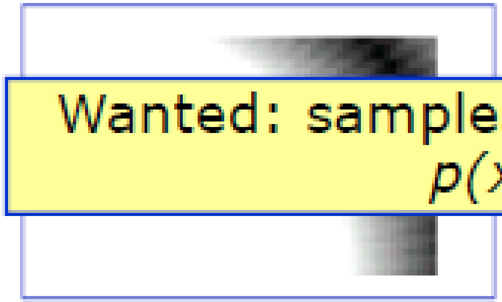
landmark in fixed range

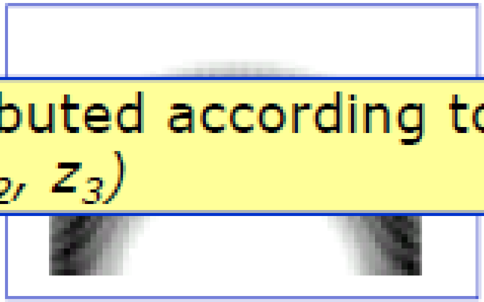




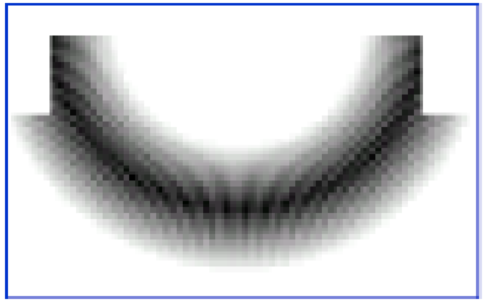
Distributions

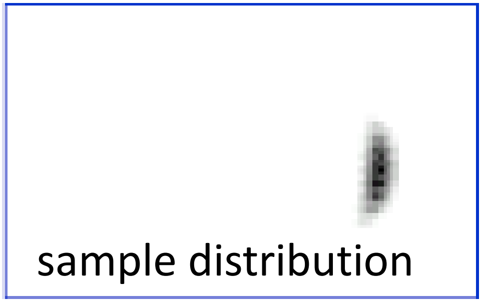




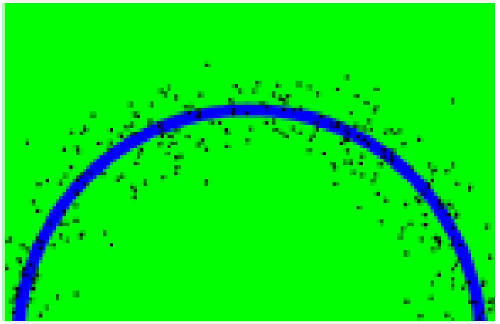


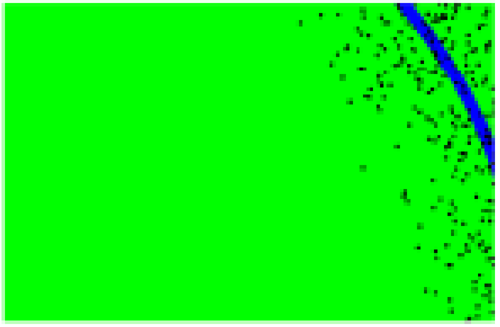
Wanted: samples distributed according to $p(x|\ z_1,\ z_2,\ z_3)$

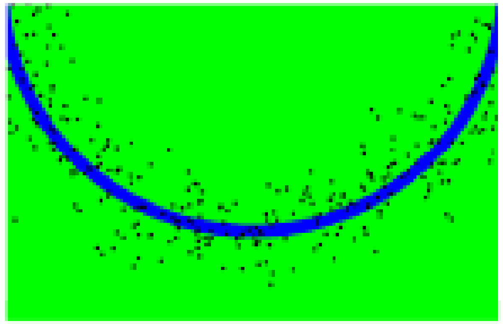




We can draw samples from $p(x|z_l)$ by adding noise to the detection parameters.

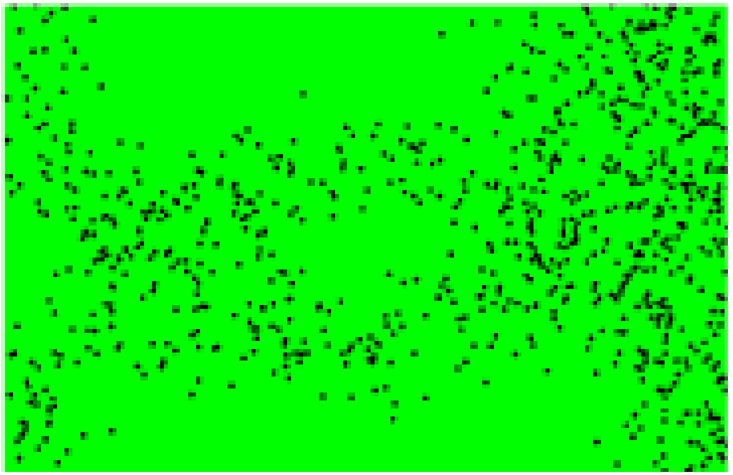




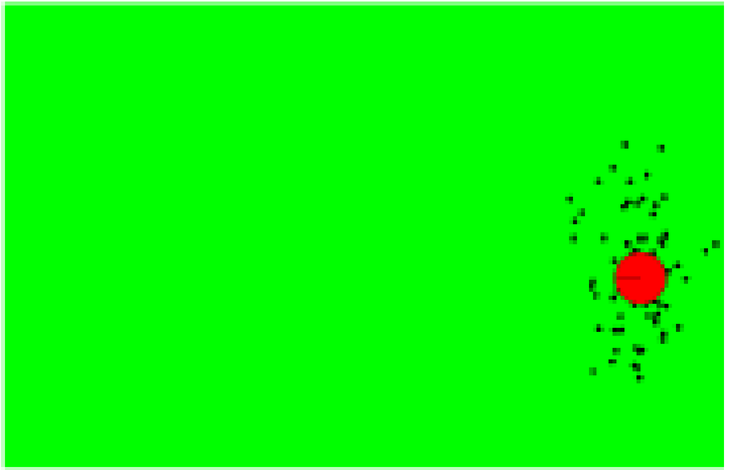


Question-how to resample it?

Importance Sampling with Resampling



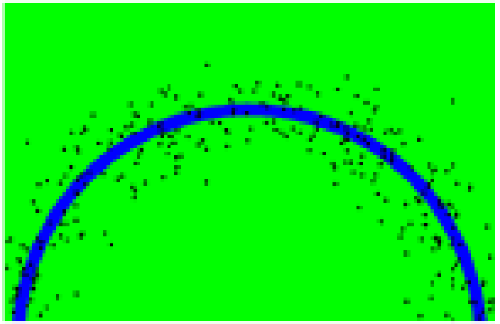
Weighted samples

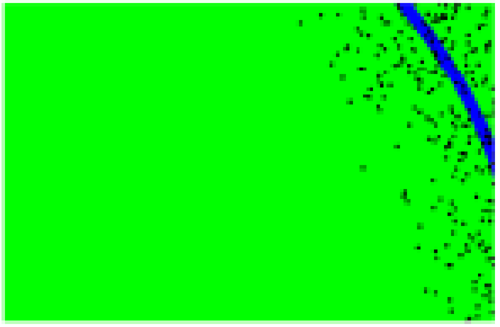


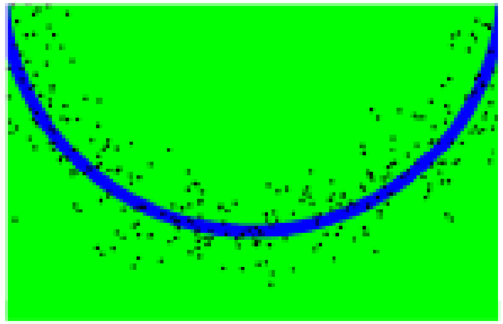
After resampling

Process flow

We can draw samples from $p(x|z_l)$ by adding noise to the detection parameters.





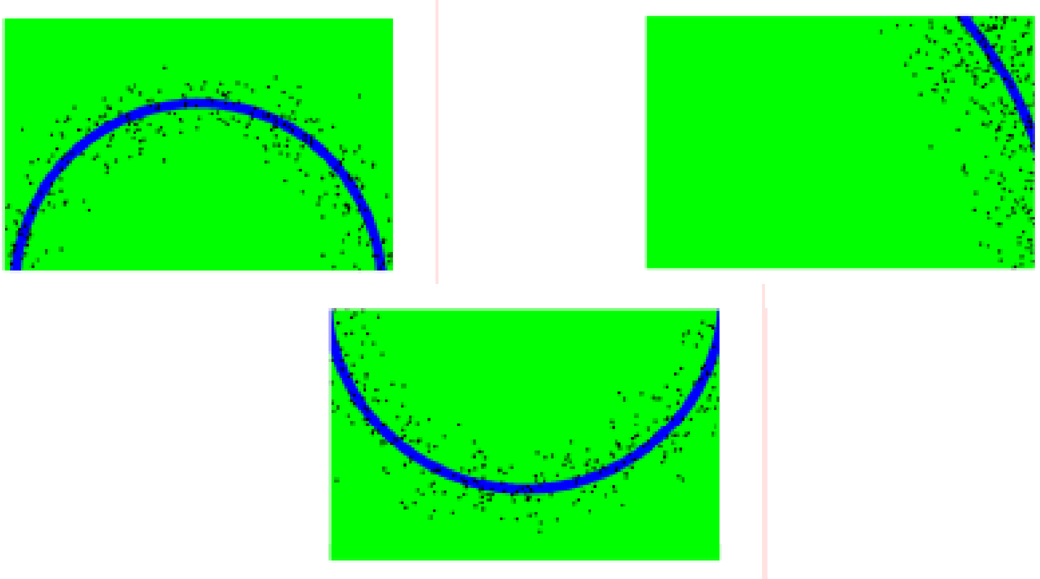


Importance Sampling

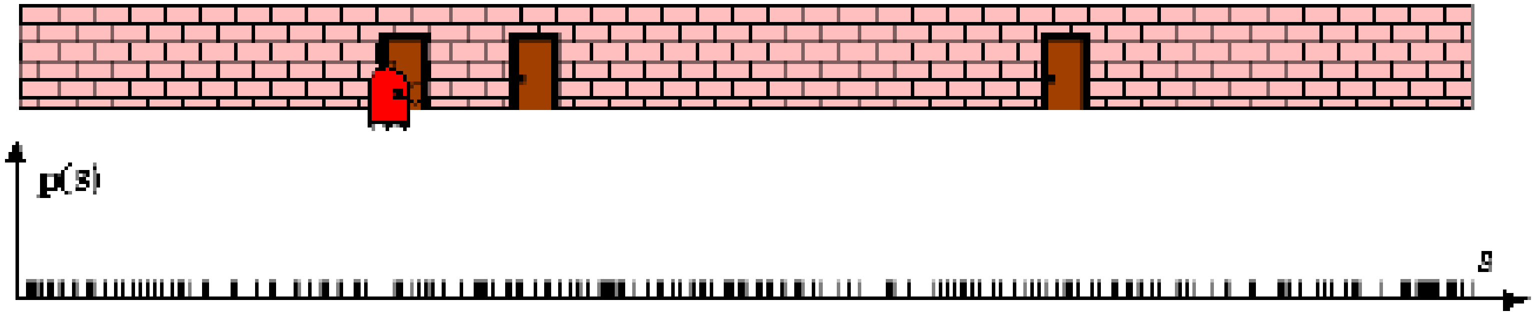
$$\text{Target distribution f} : p(x \mid z_1, z_2, \ldots, z_n) = \frac{\prod_k p(z_k \mid x) \quad p(x)}{p(z_1, z_2, \ldots, z_n)}$$

$$\text{Sampling distribution g} : p(x \mid z_l) = \frac{p(z_l \mid x) p(x)}{p(z_l)}$$

$$\text{Importance weights w} : \frac{f}{g} = \frac{p(x \mid z_1, z_2, \ldots, z_n)}{p(x \mid z_l)} = \frac{p(z_l) \prod_{k \neq l} p(z_k \mid x)}{p(z_1, z_2, \ldots, z_n)}$$



Particle Filter

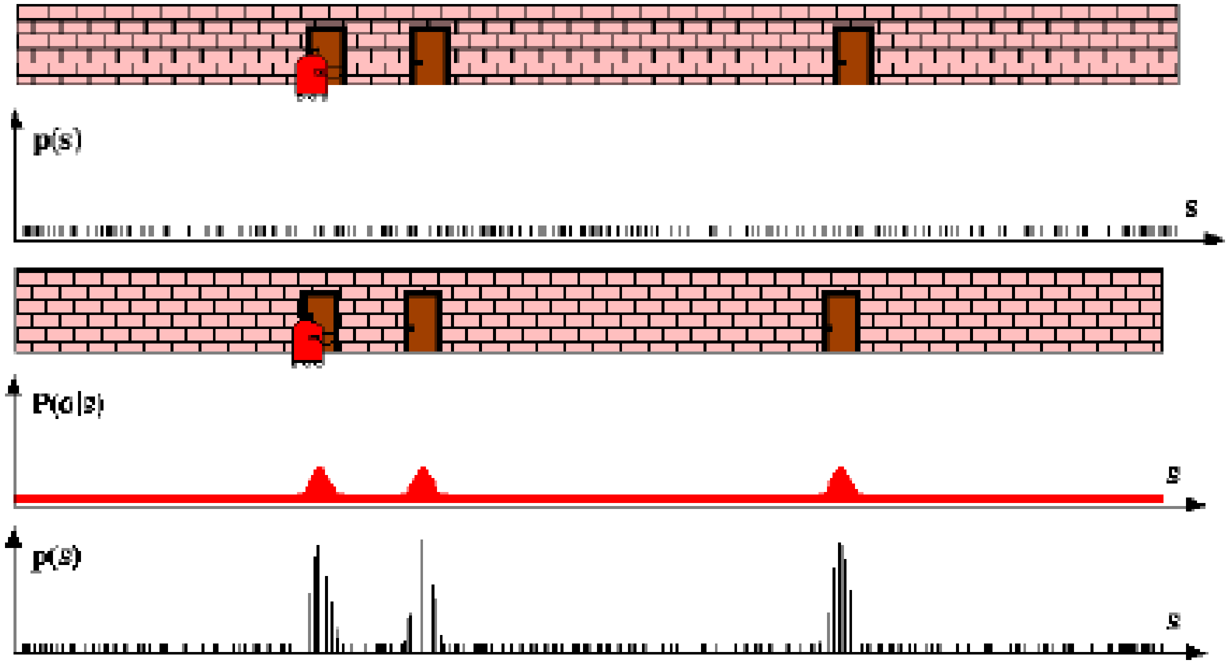


Random guess

Sensor Information : importance Sampling

$$Bel(x) \leftarrow \alpha\, p(z \mid x)\, Bel^{-}(x)$$

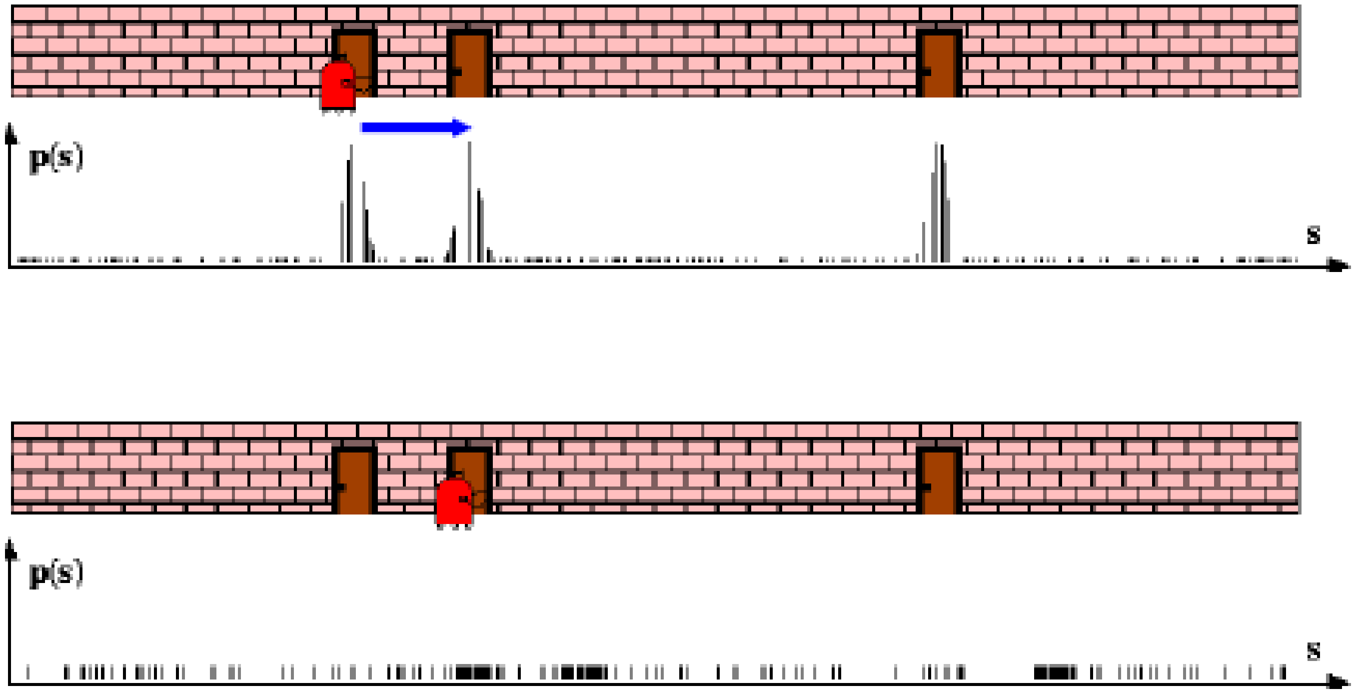
$$w \leftarrow \frac{\alpha\, p(z \mid x)\, Bel^{-}(x)}{Bel\;\;(x)} = \alpha\, p(z \mid x)$$



measurement weights

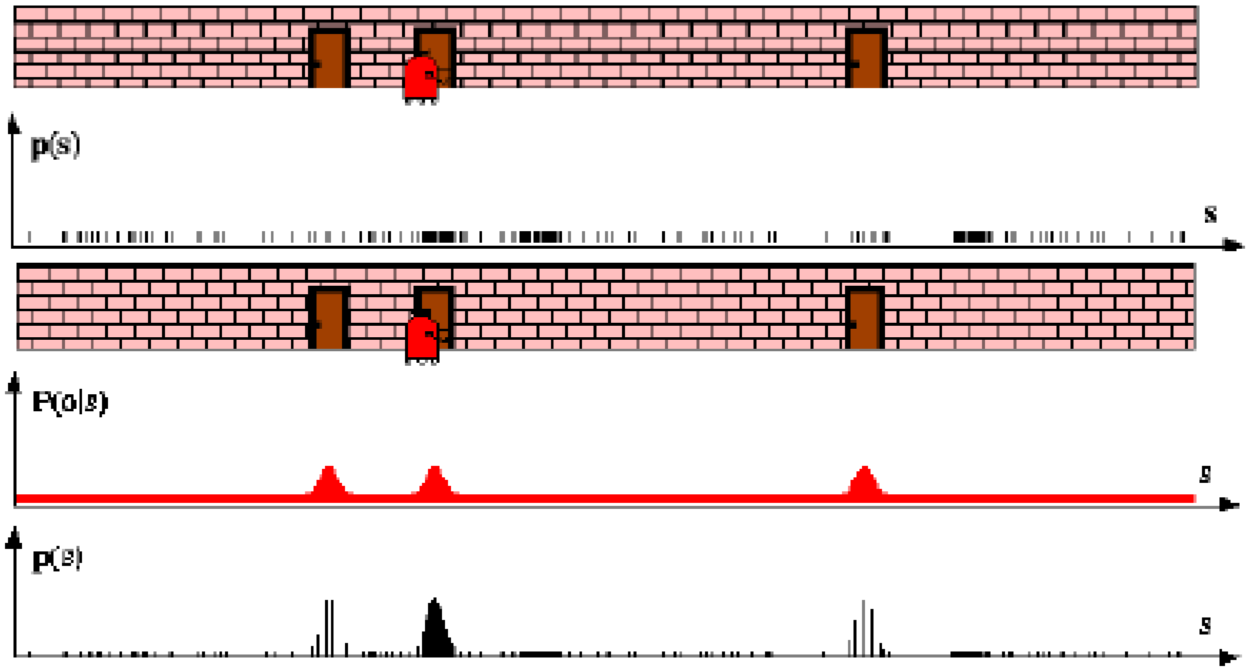
Robot Motion

$$\boxed{Bel^-(x) \leftarrow \int p(x \mid u, x')\, Bel(x')\, \mathrm{d}x'}$$



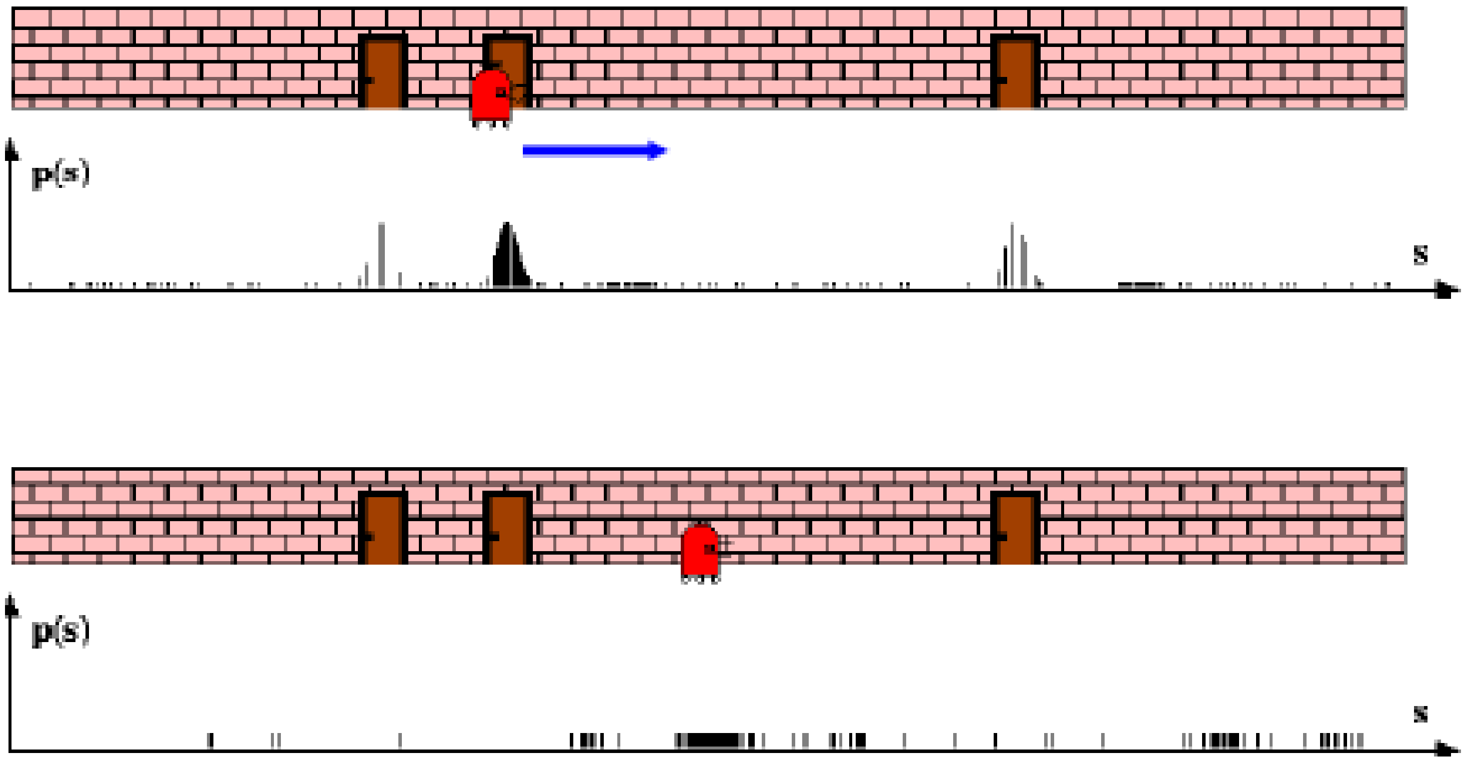
Second iter- Sensor Information: Importance Sampling

$$
\begin{aligned}
Bel(x) &\leftarrow \alpha\, p(z \mid x)\, Bel^{-}(x) \\
w &\leftarrow \frac{\alpha\, p(z \mid x)\, Bel^{-}(x)}{Bel\ (x)} = \alpha\, p(z \mid x)
\end{aligned}
$$



Robot Motion

$$Bel^{-}(x) \leftarrow \int p(x \mid u, x')\, Bel(x')\; \mathrm{d}x'$$



Particle Filter Algorithm

- Sample the next generation for particles using the proposal distribution
- Compute the importance weights :

  weight = target distribution / proposal distribution
- Resampling: "Replace unlikely samples by more likely ones"

Particle Filter Algorithm

$$Bel(x_t) = \eta\, p(z_t \mid x_t) \int p(x_t \mid x_{t-1}, u_{t-1})\, Bel(x_{t-1})\, dx_{t-1}$$

draw x^i_{t-1} from $Bel(x_{t-1})$

draw x^i_t from $p(x_t \mid x^i_{t-1}, u_{t-1})$

Importance factor for x^i_t:

$$\begin{aligned} w^i_t &= \frac{\text{target distribution}}{\text{proposal distribution}} \\ &= \frac{\eta\ p(z_t \mid x_t)\ p(x_t \mid x_{t-1}, u_{t-1})\ Bel\ (x_{t-1})}{p(x_t \mid x_{t-1}, u_{t-1})\ Bel\ (x_{t-1})} \\ &\propto p(z_t \mid x_t) \end{aligned}$$

Particle filter algorithm

1. Algorithm **particle_filter**(S_{t-1}, u_{t-1} z_t):
2. $S_t = \emptyset, \quad \eta = 0$
3. **For** $i = 1 \ldots n$ ***Generate new samples***
4. Sample index $j(i)$ from the discrete distribution given by w_{t-1}
5. Sample x_t^i from $p(x_t \mid x_{t-1}, u_{t-1})$ using $x_{t-1}^{j(i)}$ and u_{t-1}
6. $w_t^i = p(z_t \mid x_t^i)$ ***Compute importance weight***
7. $\eta = \eta + w_t^i$ ***Update normalization factor***
8. $S_t = S_t \cup \{< x_t^i, w_t^i >\}$ ***Insert***
9. **For** $i = 1 \ldots n$
10. $w_t^i = w_t^i / \eta$ ***Normalize weights***

Resampling

- **Given**: Set S of weighted samples.

- **Wanted** : Random sample, where the probability of drawing x_i is given by w_i.

- Typically done n times with replacement to generate new sample set S'.

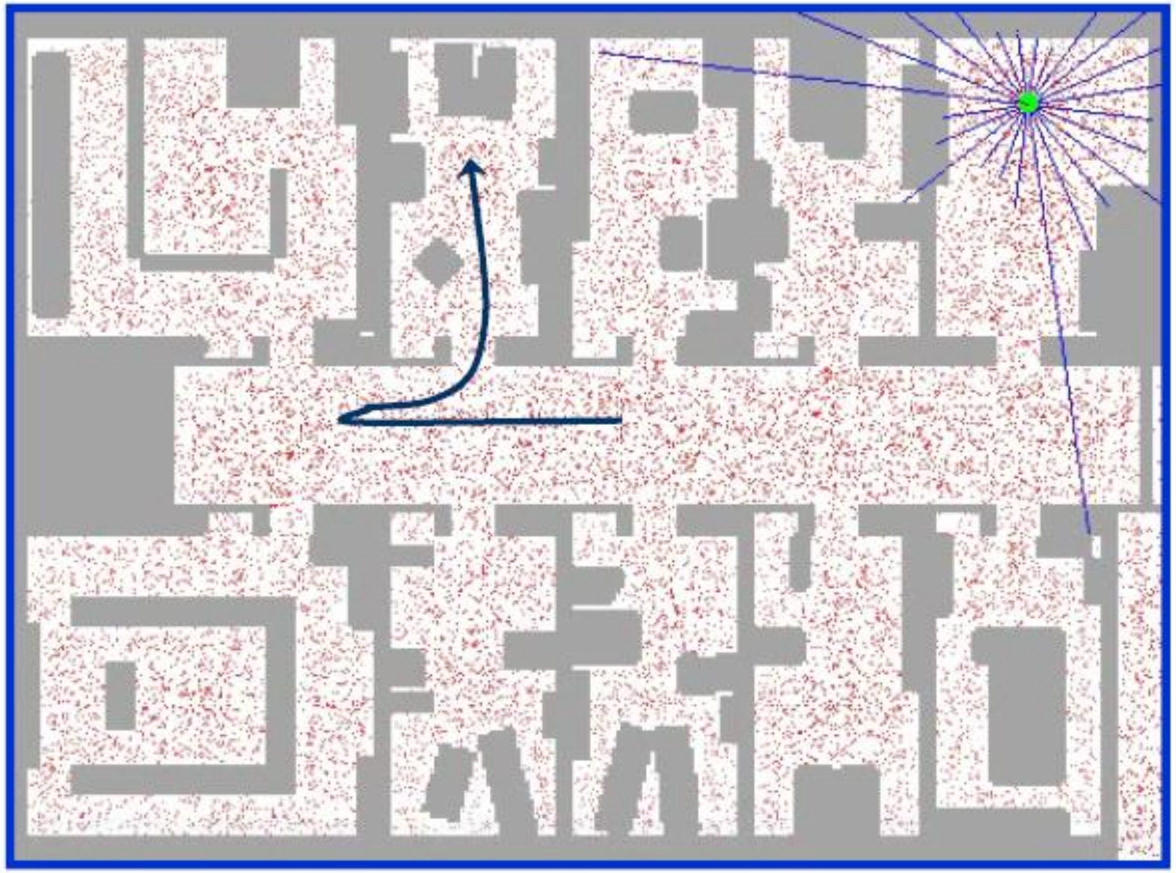
Resampling

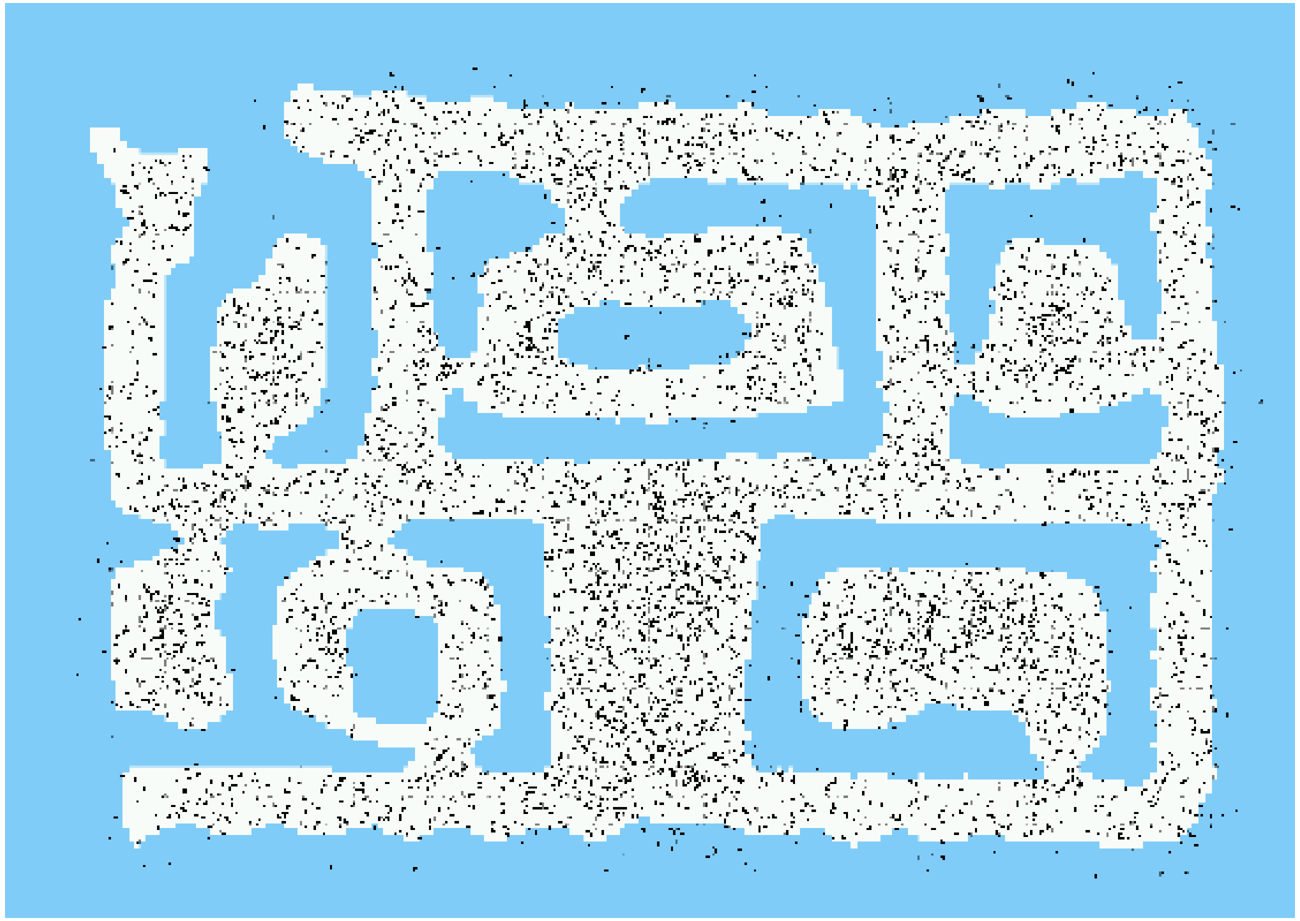
1. Algorithm **systematic_resampling**(S,n):

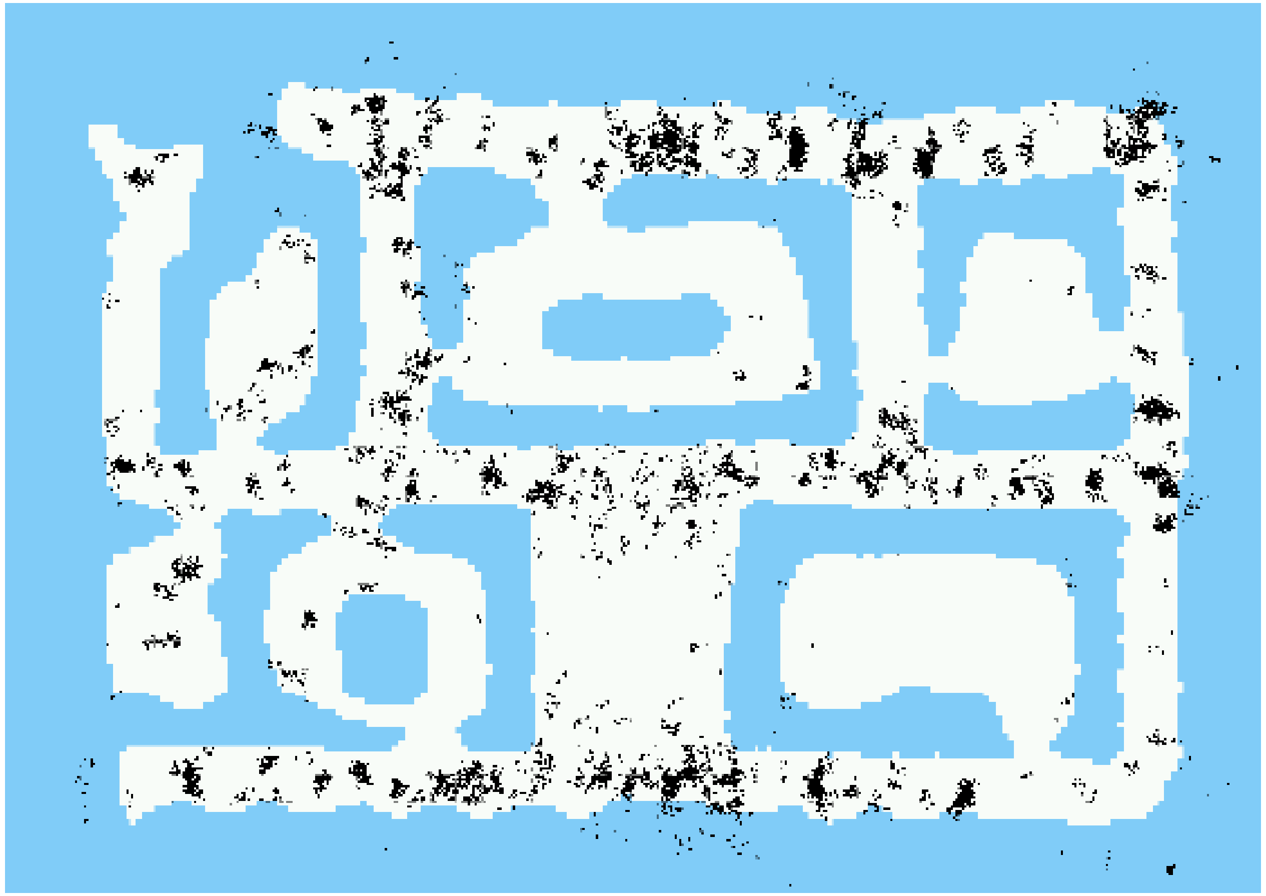
2. $S' = \emptyset, c_1 = w^1$
3. **For** $i = 2 \ldots n$ ***Generate cdf***
4. $\quad c_i = c_{i-1} + w^i$
5. $u_1 \sim U]0, n^{-1}], i = 1$ ***Initialize threshold***

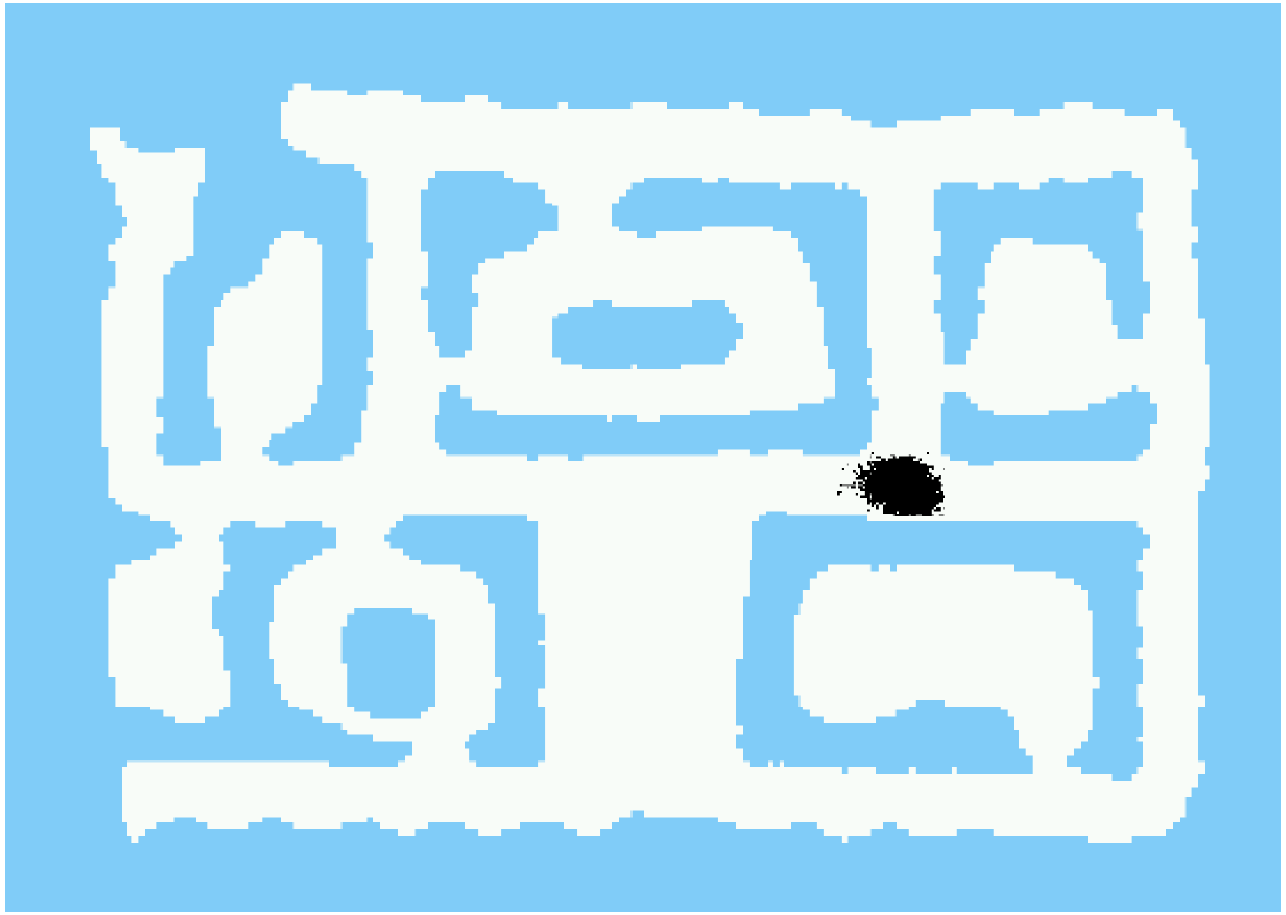
6. **For** $j = 1 \ldots n$ ***Draw samples …***
7. $\quad$ **While** ($u_j > c_i$) ***Skip until next threshold reached***
8. $\quad\quad i = i + 1$
9. $\quad S' = S' \cup \{< x^i, n^{-1} >\}$ ***Insert***
10. $\quad u_{j+1} = u_j + n^{-1}$ ***Increment threshold***

11. **Return** S'









Estimated Path