Partially Observable Markov Decision Process (POMDP)

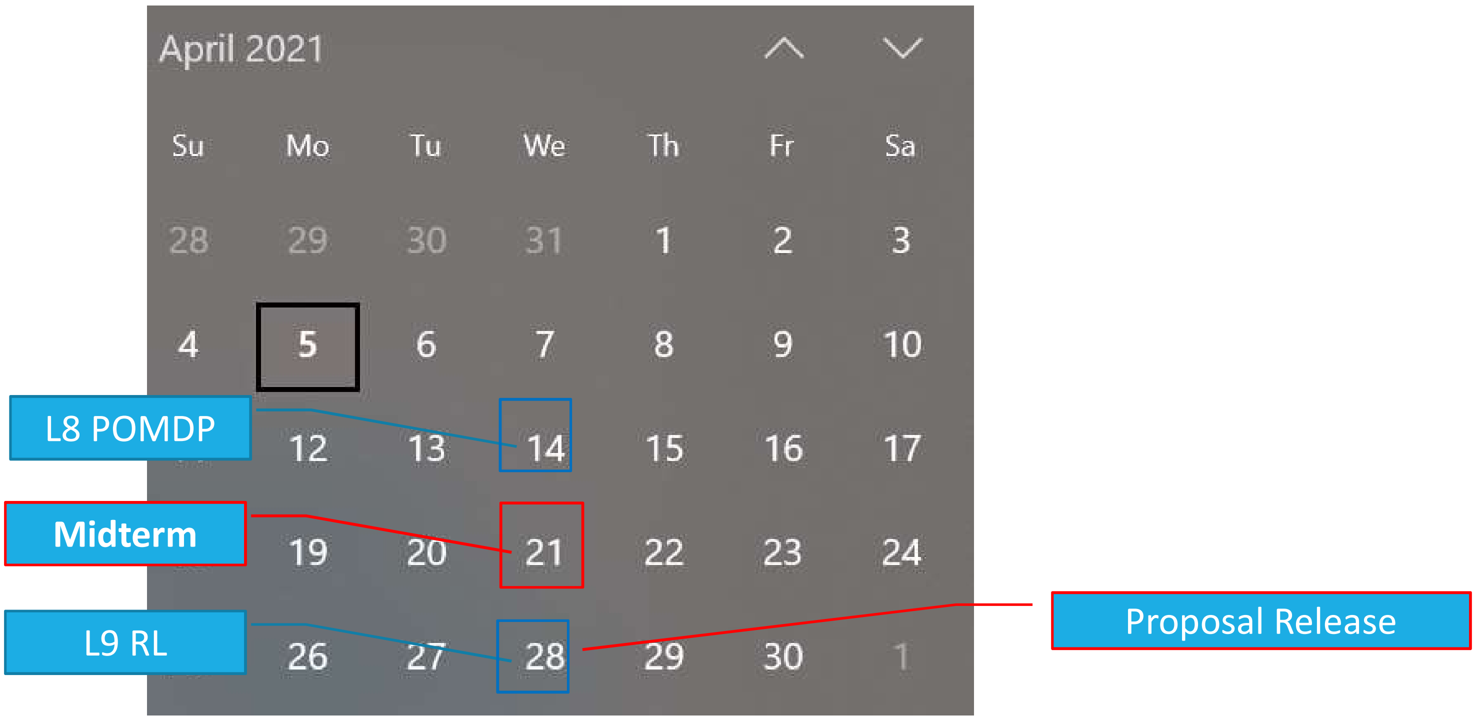
KUO-SHIH TSENG (曾國師)
DEPARTMENT OF MATHEMATICS
NATIONAL CENTRAL UNIVERSITY, TAIWAN

2021/04/14

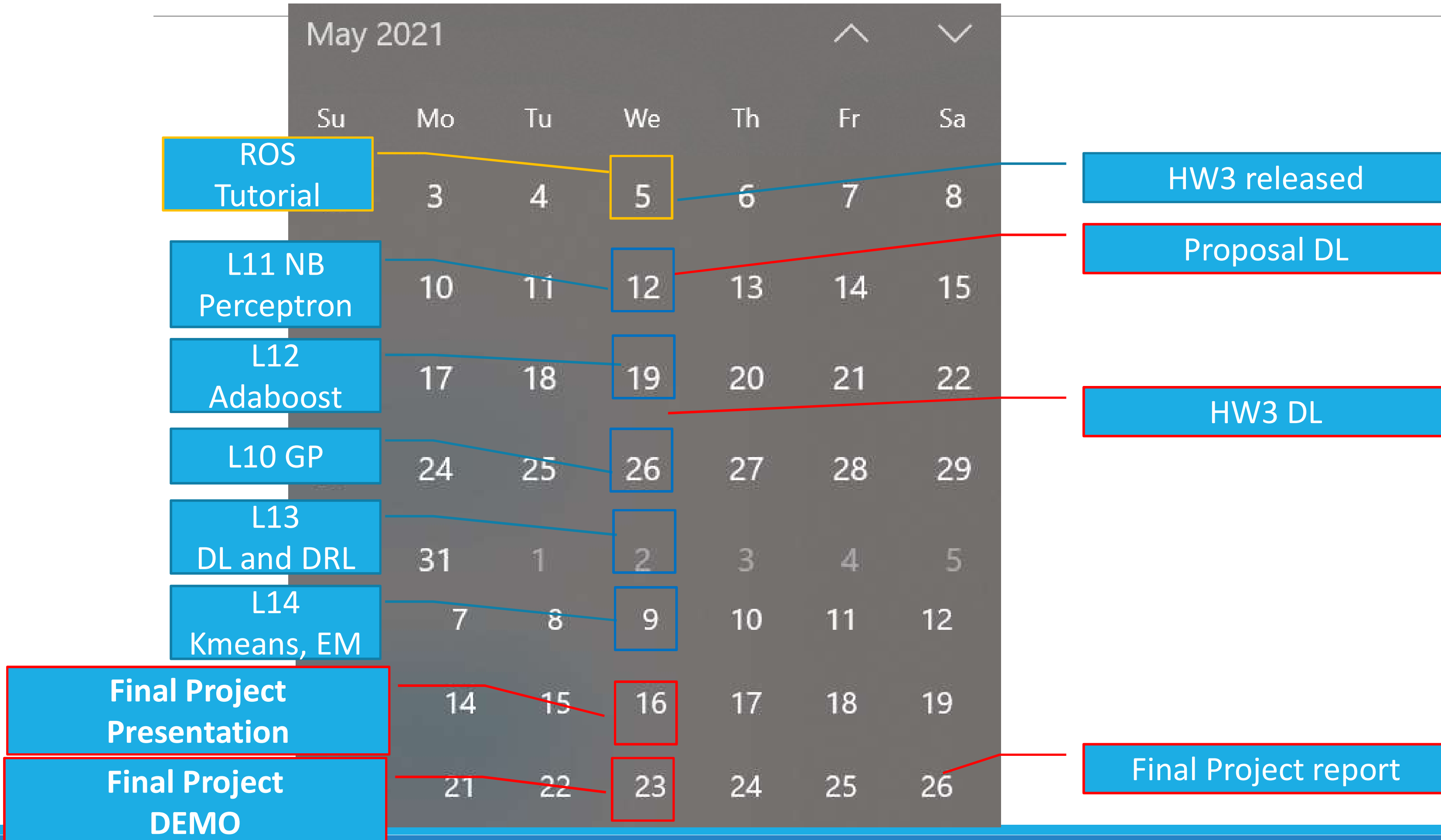
Course Announcement

- ***Midterm (04/21/2020), 3-5pm, in M430***
  - Given a real world problem.
  - Design a perception and decision-making system for this problem using MDP , MCTS and Bayesian approaches.
- You can take one A4-size cheating sheet.
- You cannot use any electrical devices (e.g., Notebook or mobilephone), which can access to internet.
- You don’t need calculators.
- You can find the **midterm_sample.pdf** on the eeclass.

Course Announcement



Course Announcement



Course Announcement

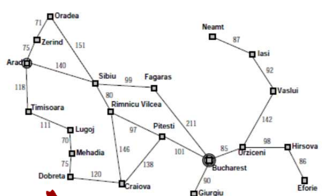
- In the final project, students need to propose interesting AI applications, formulate the problem, implement algorithms on a real robot (e.g., Minibot or Bebop).
  - A group should include **no more** than 2 students. If your project is large enough, you can have 3 group members (not recommended).
  - **Suggestion**: Find someone whose ability is similar to yours.
- The team members get the same credits (40%)!
  - Project Proposal (1-2 page) 10%
  - Project Presentation and Demonstration 10%
  - Project Report (4-8 pages) 20%
- You can find the sample files of final project on LMS.

Outline

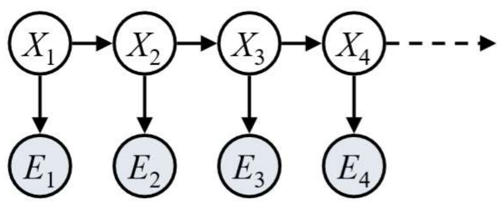
- Need
- Bayes filter
- POMDP

Outline

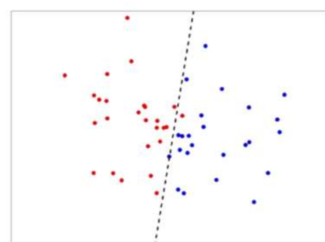
[Problem solving] → Search problems → Adversarial Search

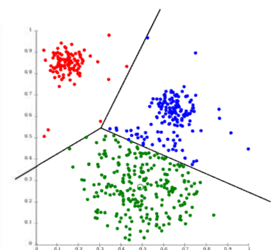


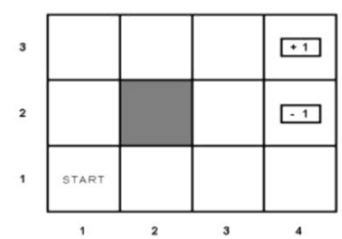
[Perception and Uncertainty] → Bayes Theorem → Bayes Filter and Smoothing



[Learning and Decision-making] → Supervised learning / Unsupervised learning / Reinforcement learning

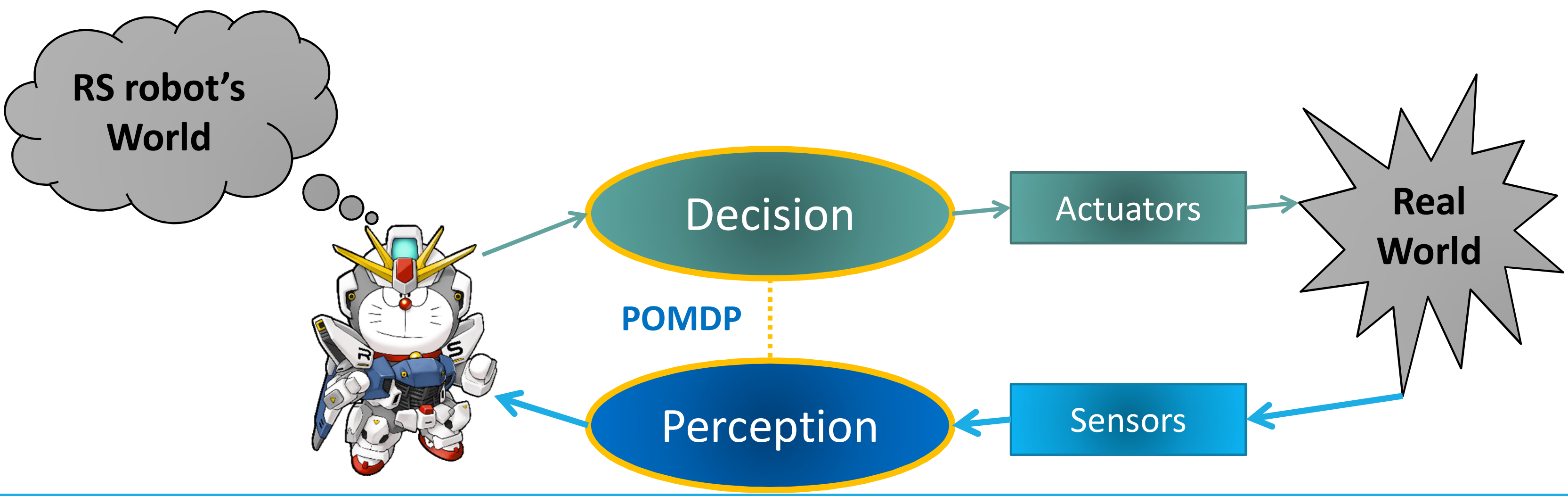






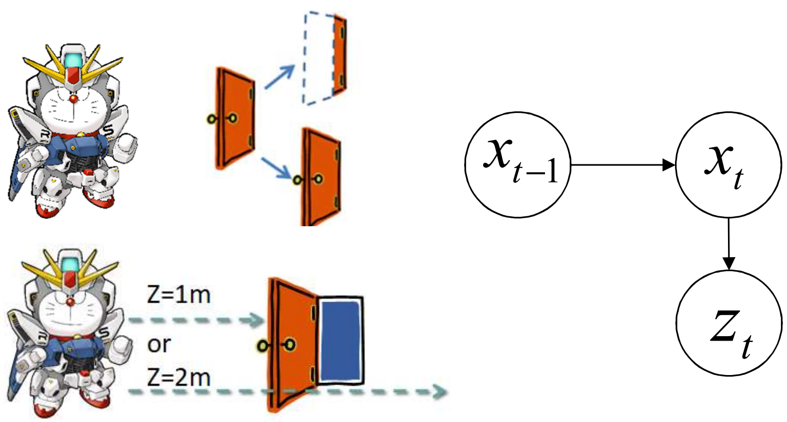
Need

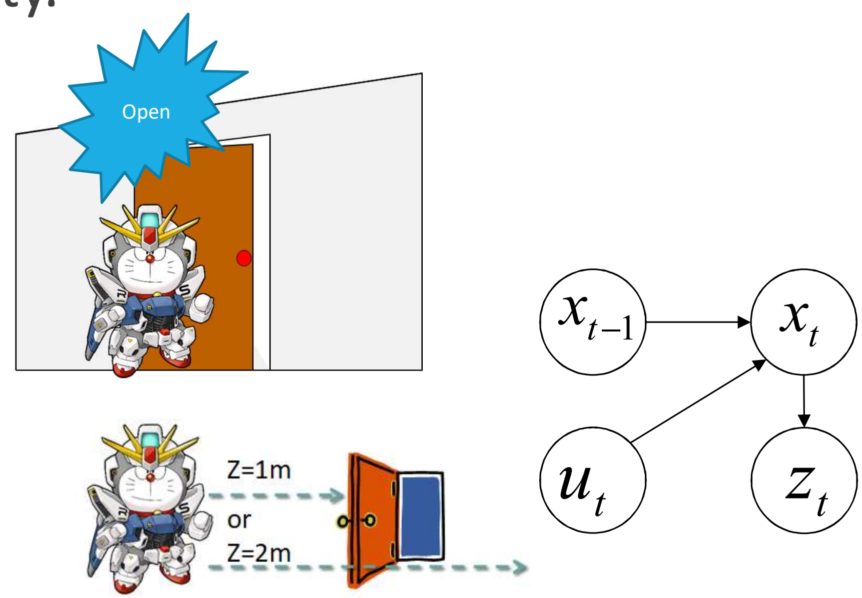
- Perception
- Decision making
- Feedback



Need

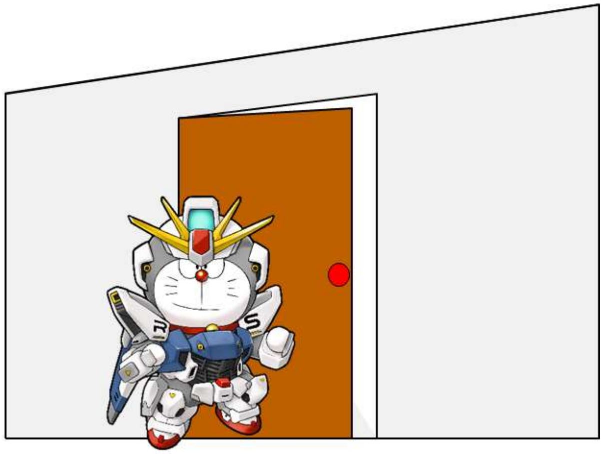
- The state of MDP is deterministic.
- If the state of MDP is probabilistic, it’s called **partially observable** Markov decision process (POMDP), which finds the optimal action sequence under the state uncertainty.





Need

- Example: z=2m, Open a door?



$$P(x_{t-1}=open)=P(x_{t-1}=close)=0.5$$

$$P(z_t=2\mid x_t=open)=0.6, P(z_t=1\mid x_t=open)=0.4$$

$$P(z_t=2\mid x_t=close)=0.2, P(z_t=1\mid x_t=close)=0.8$$

$$\boxed{\begin{aligned}
&P(x_t=open\mid u_t=push,x_{t-1}=open)=1\\
&P(x_t=close\mid u_t=push,x_{t-1}=open)=0\\
&P(x_t=open\mid u_t=push,x_{t-1}=close)=0.8\\
&P(x_t=close\mid u_t=push,x_{t-1}=close)=0.2
\end{aligned}}
\quad
\boxed{\begin{aligned}
&P(x_t=open\mid u_t=nothing,x_{t-1}=open)=1\\
&P(x_t=close\mid u_t=nothing,x_{t-1}=open)=0\\
&P(x_t=open\mid u_t=nothing,x_{t-1}=close)=0\\
&P(x_t=close\mid u_t=nothing,x_{t-1}=close)=1
\end{aligned}}$$

Transition matrix

$$Find: P(x_t=open\mid u_t=nothing, z_t=2)=?$$

$$P(x_{t+1}=open\mid u_{t+1}=push, z_{t+1}=2)=?$$

Bayes filter

- Prediction

$$\boxed{P(x_t \mid x_{t-1}, u_t, z_{t-1})} = \int \underbrace{P(x_t \mid x_{t-1}, u_t)}_{\text{Known (motion model)}} \boxed{P(x_{t-1} \mid z_{t-1})} dx_{t-1}$$

- Correction

$$\boxed{P(x_t \mid z_t)} = \eta \bullet \underbrace{P(z_t \mid x_t)}_{\text{Known (sensor model)}} \boxed{P(x_t \mid x_{t-1}, u_t, z_{t-1})}$$

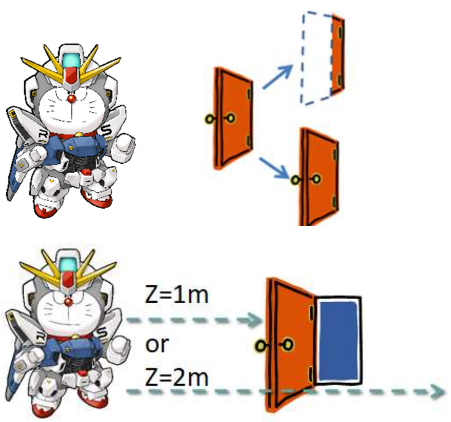
constant

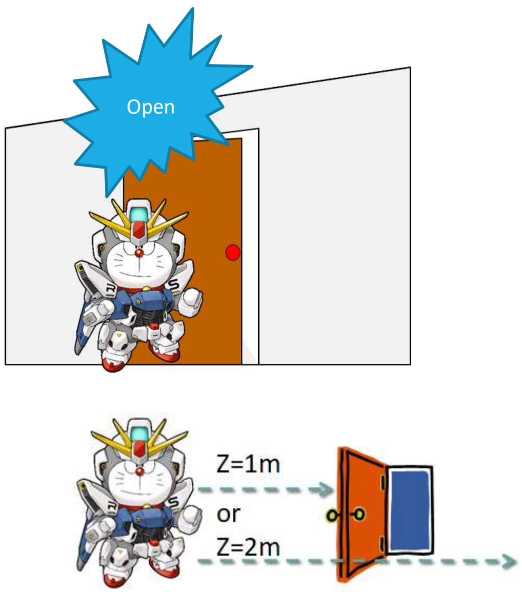
The robot action (u_t) and the door state (X_t) is relevant

Bayes filter

- EX:

$$P(x_2 \mid u_1 = nothing, z_1 = 2, u_2 = open, z_2 = 2) = ?$$





$$P(x_t = open \mid u_t = nothing, z_t = 2) = ?$$

Bayes filter

Given:

$$P(x_{t-1} = open) = P(x_{t-1} = close) = 0.5$$

$$P(z_t = 2 \mid x_t = open) = 0.6, P(z_t = 1 \mid x_t = open) = 0.4$$

$$P(z_t = 2 \mid x_t = close) = 0.2, P(z_t = 1 \mid x_t = close) = 0.8$$

$$P(x_t = open \mid u_t = push, x_{t-1} = open) = 1 \qquad P(x_t = open \mid u_t = nothing, x_{t-1} = open) = 1$$

$$P(x_t = close \mid u_t = push, x_{t-1} = open) = 0 \qquad P(x_t = close \mid u_t = nothing, x_{t-1} = open) = 0$$

$$P(x_t = open \mid u_t = push, x_{t-1} = close) = 0.8 \qquad P(x_t = open \mid u_t = nothing, x_{t-1} = close) = 0$$

$$P(x_t = close \mid u_t = push, x_{t-1} = close) = 0.2 \qquad P(x_t = close \mid u_t = nothing, x_{t-1} = close) = 1$$

$$P(x_t \mid x_{t-1}, u_t, z_{t-1}) = \sum_{x_{t-1}} P(x_t \mid x_{t-1}, u_t) P(x_{t-1} \mid z_{t-1})$$

$$\begin{cases} P(x_t = o \mid x_{t-1}, z_{t-1}) = \underset{1}{P(x_t = o \mid u_t = n, x_{t-1} = o)}\underset{0.5}{P(x_{t-1} = o)} + \underset{0}{P(x_t = o \mid u_t = n, x_{t-1} = c)}\underset{0.5}{P(x_{t-1} = c)} = 0.5 \\ P(x_t = c \mid x_{t-1}, z_{t-1}) = \underset{0}{P(x_t = c \mid u_t = n, x_{t-1} = o)}\underset{0.5}{P(x_{t-1} = o)} + \underset{1}{P(x_t = c \mid u_t = n, x_{t-1} = c)}\underset{0.5}{P(x_{t-1} = c)} = 0.5 \end{cases}$$

$$P(x_t = open \mid u_t = nothing, z_t = 2) = ?$$

Bayes filter

Given:

$$P(x_{t-1} = open) = P(x_{t-1} = close) = 0.5$$

$$P(z_t = 2 \mid x_t = open) = 0.6, P(z_t = 1 \mid x_t = open) = 0.4$$

$$P(z_t = 2 \mid x_t = close) = 0.2, P(z_t = 1 \mid x_t = close) = 0.8$$

$$P(x_t = open \mid u_t = push, x_{t-1} = open) = 1 \qquad P(x_t = open \mid u_t = nothing, x_{t-1} = open) = 1$$

$$P(x_t = close \mid u_t = push, x_{t-1} = open) = 0 \qquad P(x_t = close \mid u_t = nothing, x_{t-1} = open) = 0$$

$$P(x_t = open \mid u_t = push, x_{t-1} = close) = 0.8 \qquad P(x_t = open \mid u_t = nothing, x_{t-1} = close) = 0$$

$$P(x_t = close \mid u_t = push, x_{t-1} = close) = 0.2 \qquad P(x_t = close \mid u_t = nothing, x_{t-1} = close) = 1$$

$$P(x_t \mid z_t) = \eta \bullet P(z_t \mid x_t) P(x_t \mid x_{t-1}, z_{t-1})$$

$$\begin{cases} P(x_t = o \mid z_t) = \eta \bullet \underbrace{P(z_t \mid x_t = o)}_{0.6} \underbrace{P(x_t = o \mid x_{t-1}, u_t, z_{t-1})}_{0.5} = 0.75 \\ P(x_t = c \mid z_t) = \eta \bullet \underbrace{P(z_t \mid x_t = c)}_{0.2} \underbrace{P(x_t = c \mid x_{t-1}, u_t, z_{t-1})}_{0.5} = 0.25 \end{cases}$$

$$(0.4\eta = 1, \eta = 2.5)$$

$$P(x_{t+1}=open \mid u_{t+1}=push, z_{t+1}=2)=?$$

Bayes filter

Given:

$$P(x_{t-1}=open)=P(x_{t-1}=close)=0.5$$

$$P(z_t=2 \mid x_t=open)=0.6, P(z_t=1 \mid x_t=open)=0.4$$

$$P(z_t=2 \mid x_t=close)=0.2, P(z_t=1 \mid x_t=close)=0.8$$

$$P(x_t=open \mid u_t=push, x_{t-1}=open)=1 \qquad P(x_t=open \mid u_t=nothing, x_{t-1}=open)=1$$

$$P(x_t=close \mid u_t=push, x_{t-1}=open)=0 \qquad P(x_t=close \mid u_t=nothing, x_{t-1}=open)=0$$

$$P(x_t=open \mid u_t=push, x_{t-1}=close)=0.8 \qquad P(x_t=open \mid u_t=nothing, x_{t-1}=close)=0$$

$$P(x_t=close \mid u_t=push, x_{t-1}=close)=0.2 \qquad P(x_t=close \mid u_t=nothing, x_{t-1}=close)=1$$

$$P(x_t \mid x_{t-1}, u_t, z_{t-1})=\sum_{x_{t-1}} P(x_t \mid x_{t-1}, u_t) P(x_{t-1} \mid z_{t-1})$$

$$\begin{cases} P(x_t=o \mid x_{t-1}, z_{t-1})=\underbrace{P(x_t=o \mid u_t=n, x_{t-1}=o)}_{1}\underbrace{P(x_{t-1}=o)}_{0.75}+\underbrace{P(x_t=o \mid u_t=p, x_{t-1}=c)}_{0.8}\underbrace{P(x_{t-1}=c)}_{0.25}=0.95 \\ P(x_t=c \mid x_{t-1}, z_{t-1})=\underbrace{P(x_t=c \mid u_t=n, x_{t-1}=o)}_{0}\underbrace{P(x_{t-1}=o)}_{0.75}+\underbrace{P(x_t=c \mid u_t=p, x_{t-1}=c)}_{0.2}\underbrace{P(x_{t-1}=c)}_{0.25}=0.05 \end{cases}$$

$$P(x_{t+1}=open \mid u_{t+1}=push, z_{t+1}=2)=?$$

Bayes filter

Given:

$$P(x_{t-1}=open)=P(x_{t-1}=close)=0.5$$

$$P(z_t=2 \mid x_t=open)=0.6, P(z_t=1 \mid x_t=open)=0.4$$

$$P(z_t=2 \mid x_t=close)=0.2, P(z_t=1 \mid x_t=close)=0.8$$

$$P(x_t=open \mid u_t=push, x_{t-1}=open)=1 \qquad P(x_t=open \mid u_t=nothing, x_{t-1}=open)=1$$

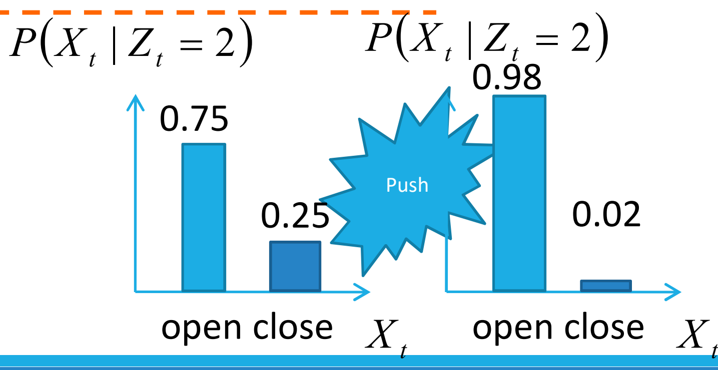
$$P(x_t=close \mid u_t=push, x_{t-1}=open)=0 \qquad P(x_t=close \mid u_t=nothing, x_{t-1}=open)=0$$

$$P(x_t=open \mid u_t=push, x_{t-1}=close)=0.8 \qquad P(x_t=open \mid u_t=nothing, x_{t-1}=close)=0$$

$$P(x_t=close \mid u_t=push, x_{t-1}=close)=0.2 \qquad P(x_t=close \mid u_t=nothing, x_{t-1}=close)=1$$

$$P(x_t \mid z_t)=\eta \bullet P(z_t \mid x_t)P(x_t \mid x_{t-1}, z_{t-1})$$

$$\begin{cases} P(x_t=o \mid z_t)=\eta \bullet \underset{0.6}{P(z_t \mid x_t=o)}\,\underset{0.95}{P(x_t=o \mid x_{t-1},u_t,z_{t-1})}=0.98 \\ P(x_t=c \mid z_t)=\eta \bullet \underset{0.2}{P(z_t \mid x_t=c)}\,\underset{0.05}{P(x_t=c \mid x_{t-1},u_t,z_{t-1})}=0.02 \end{cases}$$



MDP

- MDP is a model for finding sequential optimal decisions.
  - State: fully observable
  - State transition: stochastic **(Motion model)**

$[GIVEN]$

$S: state$

$A: action$

$\boxed{P(s'|s,a): Transition\ probability}$

$R(s,a): reward$

$\gamma: discount$

$[Find]$

$$\pi^* = \arg\max U^\pi(s)$$

$$U^\pi(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(S_t)\right]$$

$\pi: policy$

$\pi^*: optimal\ policy$

$U: utility$

POMDP

- POMDP is a model for finding sequential optimal decisions.
  - State: **stochastic** **(Sensor model)**
  - State transition: stochastic **(Motion model)**

$[GIVEN]$

$S : state$

$A : action$

$\boxed{P(s'|s,a) : Transition\ probability}$

$R(s,a) : reward$

$\gamma : discount$

$$\boxed{\begin{array}{l} z : a\ set\ of\ measurement \\ P(z|s) : sensor\ \mathrm{mod}\,el \end{array}}$$

$[Find]$

$$\pi^* = \arg\max U^{\pi}(s)$$

$$U^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(S_t)\right]$$

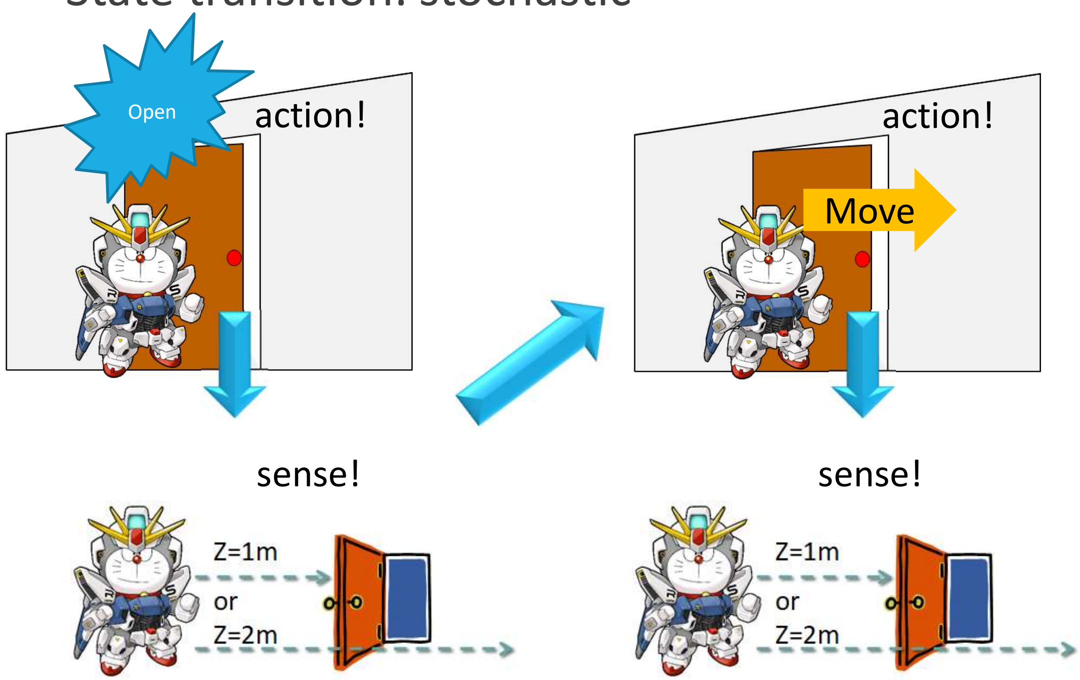
$\pi : policy$

$\pi^* : optimal\ policy$

$U : utility$

POMDP

- POMDP is a model for finding sequential optimal decisions.
  - State: **stochastic** (The robot has to ***sense*** the state via sensors)
  - State transition: stochastic



$$[Find]$$

$$\pi^* = \arg\max U^\pi(s)$$

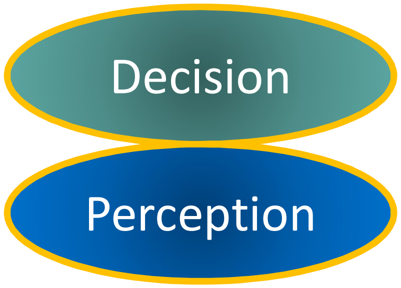
$$U^\pi(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(S_t)\right]$$

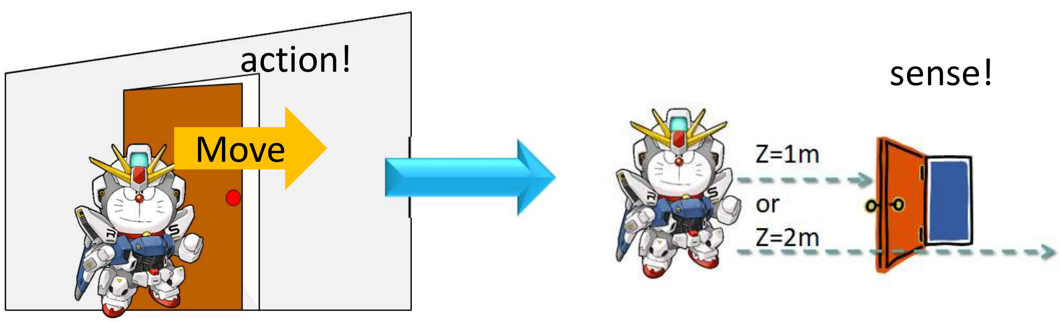
POMDP

- POMDP is a model for finding sequential optimal decisions.
  - State: **stochastic** (The robot has to ***sense*** the state via sensors)
  - State transition: stochastic

3 steps of POMDP :

1. Given current belief state b, execute the action $a = \pi^*(b)$.
2. Receive measurement z.
3. Set the current belief state to $Foward(b,a,z)$ and repeat.

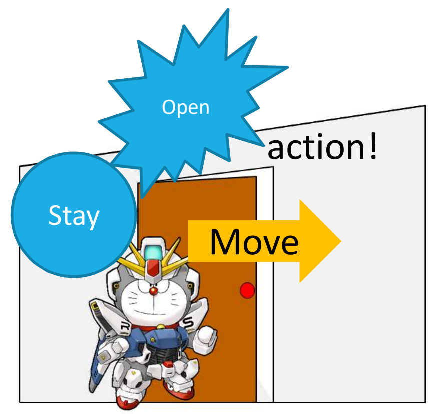




POMDP

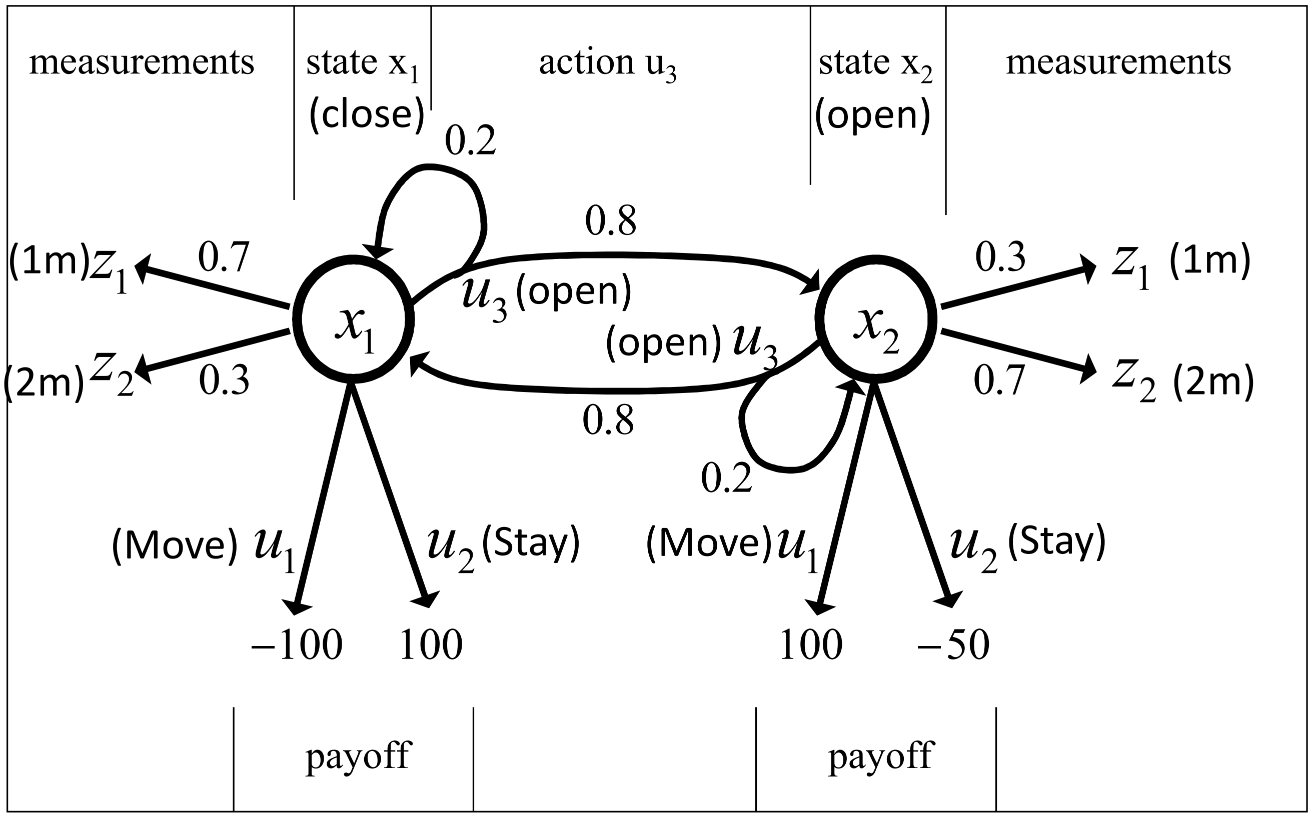
- Let's look at an example of POMDP

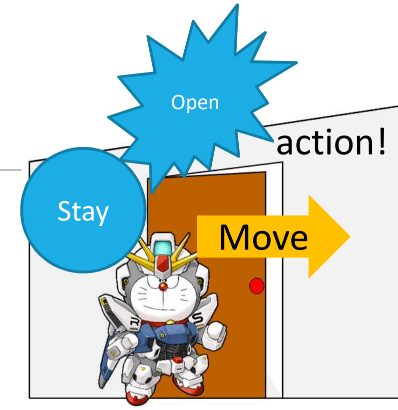
$$
\begin{aligned}
&[GIVEN] \\
&S \in \{Close, Open\} \\
&A \in \{Move, Stay, Open\} \\
&P(s'|s,a): Transition\ probability \\
&R(s,a): reward \\
&\gamma: discount \\
&z: a\ set\ of\ measurement \\
&P(z|s): sensor\ \mathrm{mod}\, el
\end{aligned}
$$



Let's check the state machine of this problem!

POMDP





Example from:
Probabilistic Robotics

POMDP

- The actions u_1 and u_2 are terminal actions. The action u_3 is a sensing action that potentially leads to a state transition.
- The horizon is finite and γ=1.

Motion Model:

$P(x_1 \mid x_1, u_3) = 0.2, P(x_2 \mid x_1, u_3) = 0.8$

$P(x_1 \mid x_2, u_3) = 0.8, P(x_2 \mid x_2, u_3) = 0.2$

Sensor Model:

$P(z_1 \mid x_1) = 0.7, P(z_2 \mid x_1) = 0.3$

$P(z_1 \mid x_2) = 0.3, P(z_2 \mid x_2) = 0.7$

Reward:

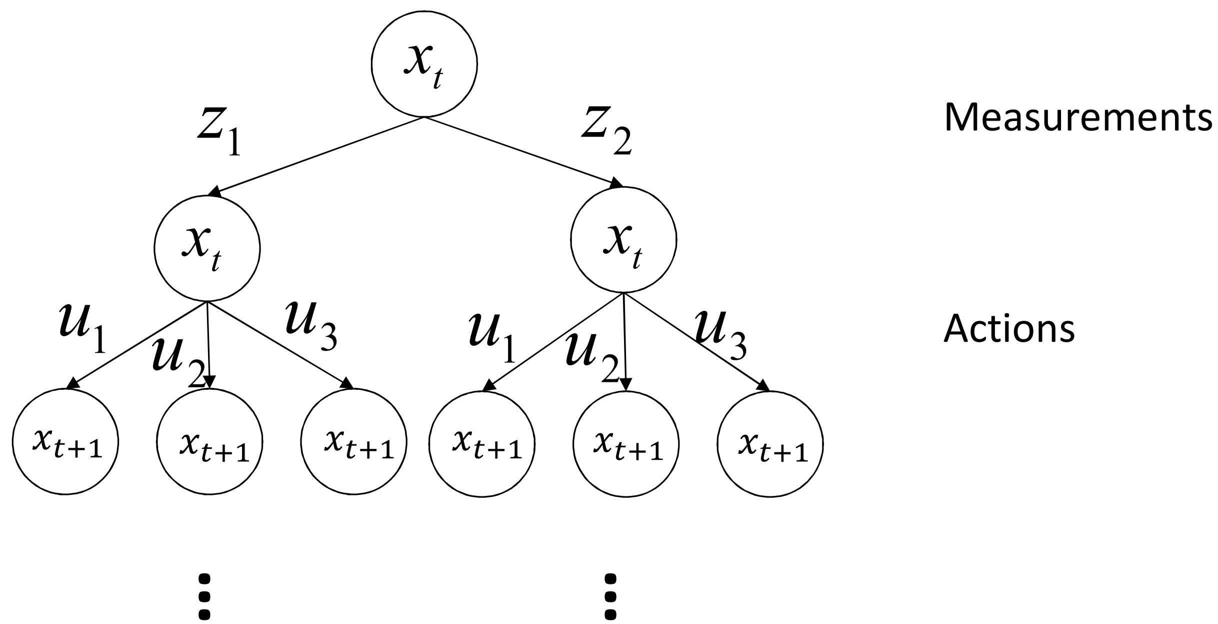
$r(x_1, u_1) = -100, r(x_2, u_1) = +100$

$r(x_1, u_2) = +100, r(x_2, u_2) = -50$

$r(x_1, u_3) = -1, r(x_2, u_3) = -1$

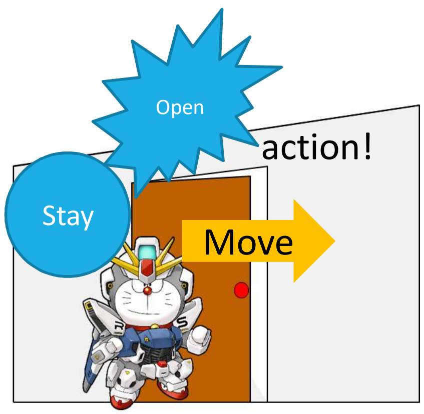
POMDP

- Search tree of POMDP

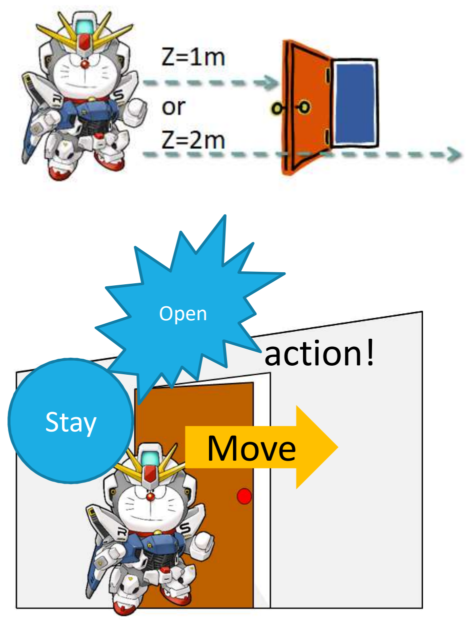


POMDP

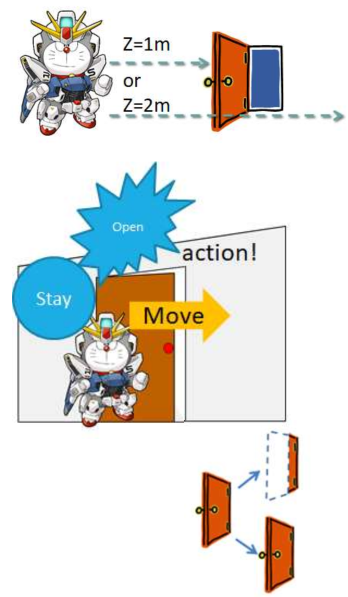
- **Action only**



- Sense
- Action



- Sense
- Action
- Transition



- Sense
- Action
- Transition
- H horizons

……

POMDP

- In MDPs, the payoff (or return) depended on the state of the system.
- In POMDPs, however, the true state is not exactly known.
- Therefore, we compute the **expected payoff** by **integrating/sum over all states**:

$$
\begin{aligned}
r(b,u) &= E_x\left[r(x,u)\right] \\
&= \sum_x r(x,u)P(x) \\
&= P(x_1)r(x_1,u) + P(x_2)r(x_2,u)
\end{aligned}
$$

POMDP

- If we are totally certain that we are in state x_1 and execute action u_1, we receive a reward of -100
- If, on the other hand, we definitely know that we are in x_2 and execute u_1, the reward is +100.
- In between it is the linear combination of the extreme values weighted by the probabilities

$$r(b,u_1) = -100P(x_1) + 100P(x_2)$$

$$= -100\,p_1 + 100(1 - p_1)$$

$$r(b,u_2) = +100\,p_1 - 50(1 - p_1)$$

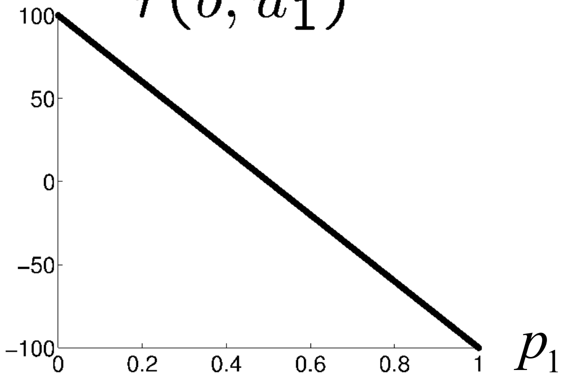
$$r(b,u_3) = -1$$

$$P(x_1) = p_1$$

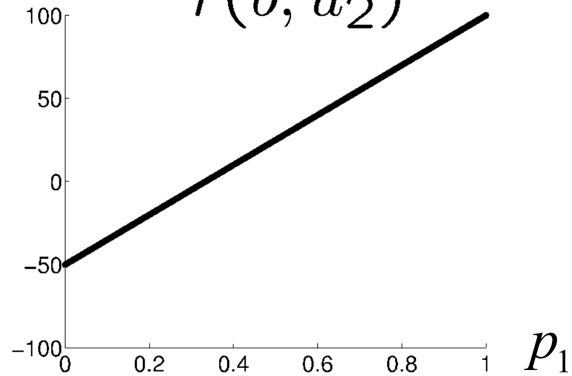
Let b be the belief of the agent about the state under consideration

POMDP

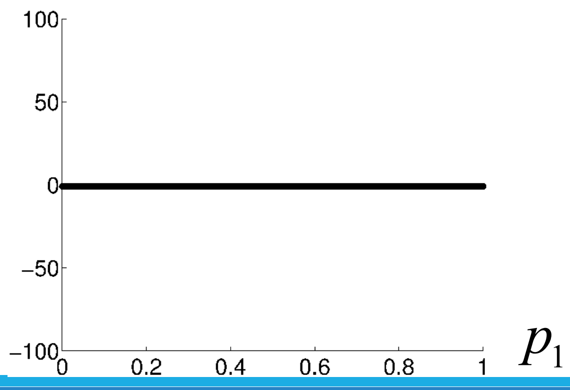
$r(b, u_1)$



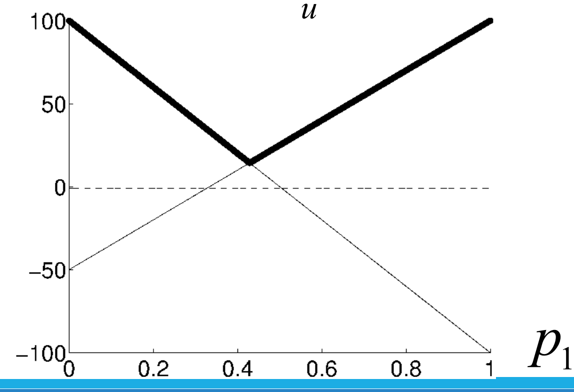
$r(b, u_2)$

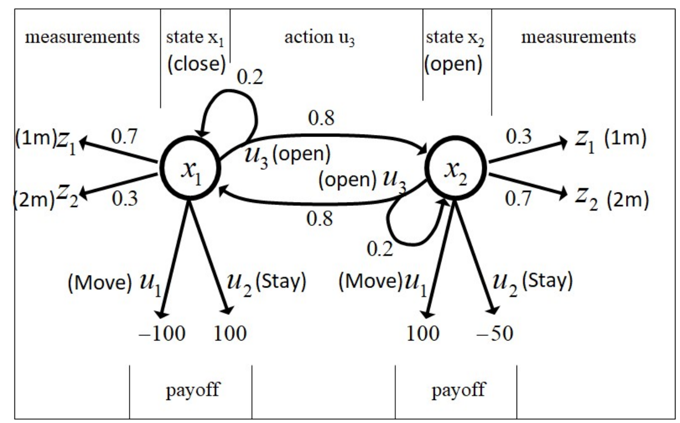


$r(b, u_3)$



$V_1(b) = \max_u r(b,u)$





POMDP

- The Resulting Policy for T=1
- Given we have a finite POMDP with T=1, we would use $V_1(b)$ to determine the optimal policy.
- In our example, the optimal policy for T=1 is

$$\pi_1(b) = \begin{cases} u_1 \text{ if } \mathrm{p}_1 \leq \frac{3}{7} \\ u_2 \text{ if } \mathrm{p}_1 > \frac{3}{7} \end{cases}$$

- This is the upper thick graph in the diagram.

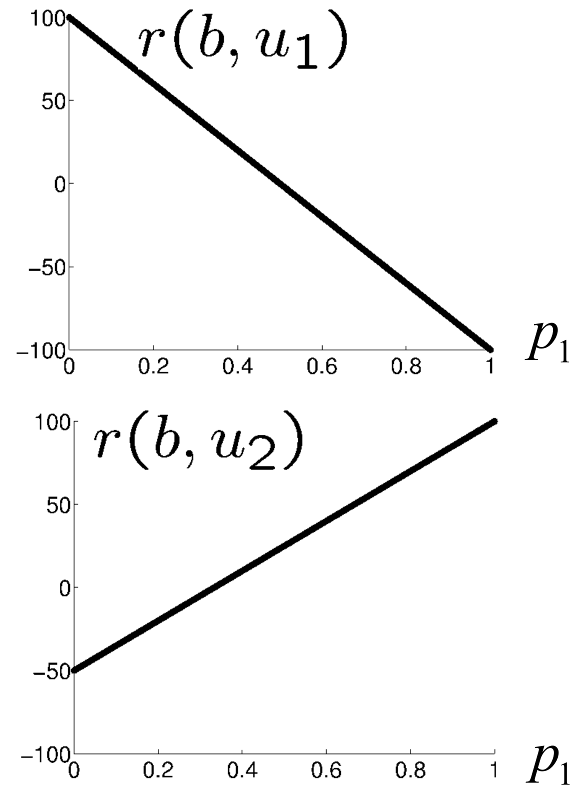
POMDP

- The resulting value function $V_1(b)$ is the maximum of the three functions at each point

$$V_1(b) = \max_u r(b,u)$$

$$= \max \begin{cases} -100\,p_1 + 100(1-p_1) & \leftarrow \text{U1} \\ 100\,p_1 - 50(1-p_1) & \leftarrow \text{U2} \\ -1 & \leftarrow \text{U3} \end{cases}$$

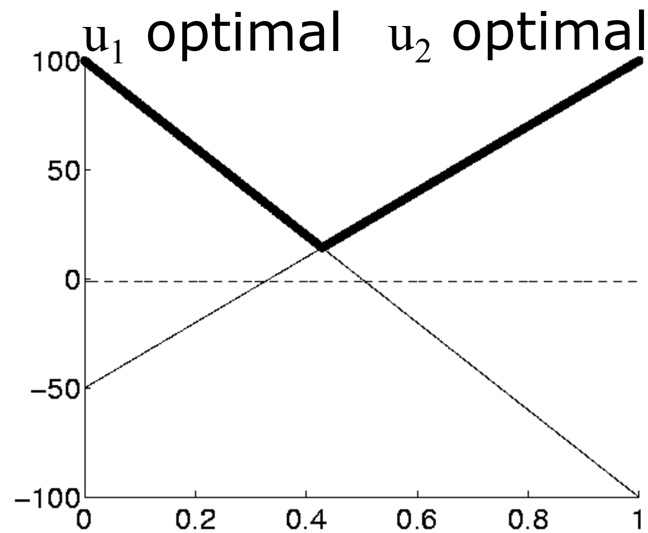
- It is piecewise linear and convex.



POMDP

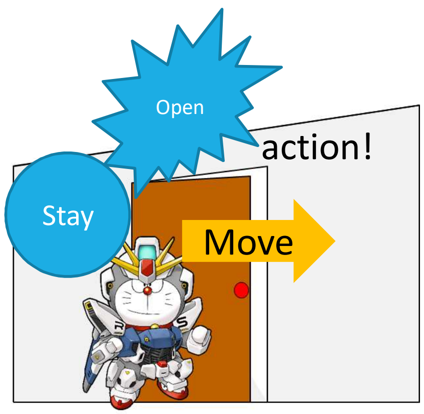
- Pruning:
- If we carefully consider $V_1(b)$, we see that only the first two components contribute.
- The third component can therefore safely be pruned away from $V_1(b)$.

$$V_1(b) = \max \begin{cases} -100\,p_1 + 100(1-p_1) \\ 100\,p_1 - 50(1-p_1) \end{cases}$$

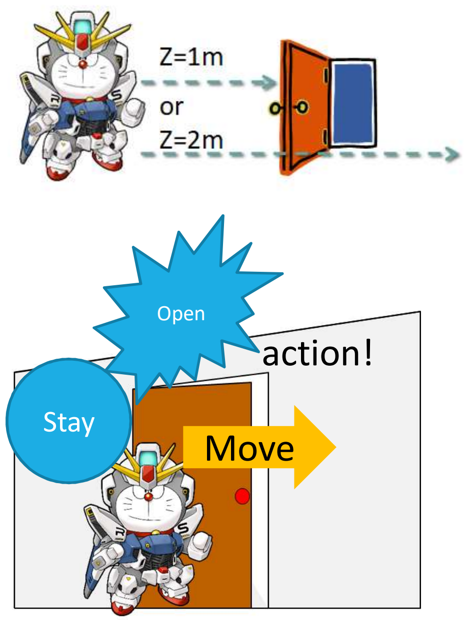


POMDP

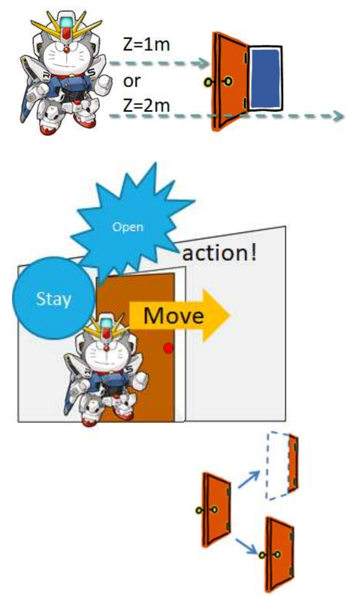
- Action only



- **Sense**
- **Action**



- Sense
- Action
- Transition

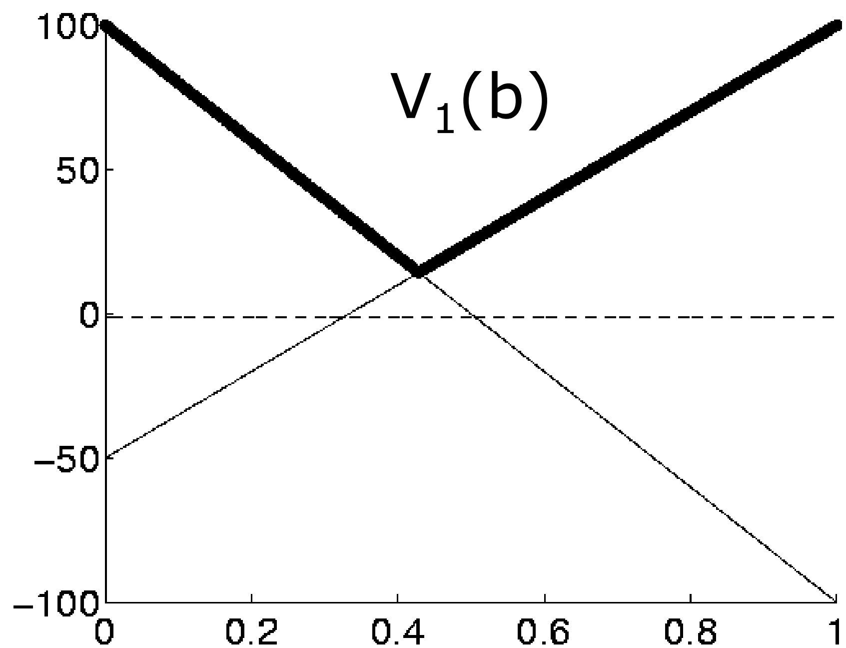


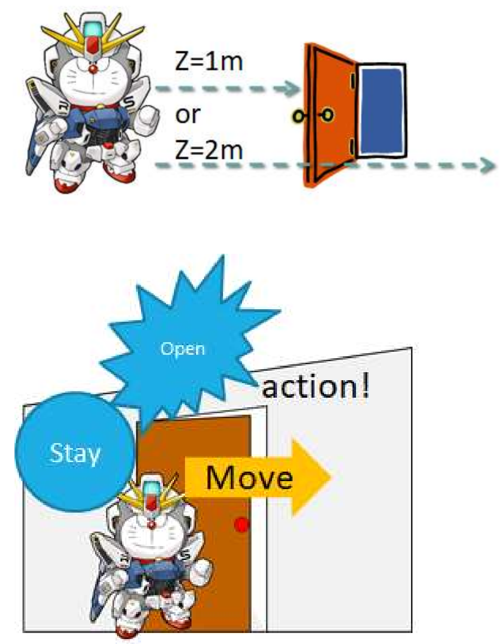
- Sense
- Action
- Transition
- H horizons

……

POMDP

- Assume the robot can make an observation before deciding on an action. ➔ Bayes theorem!





POMDP

- Assume the robot can make an observation before deciding on an action.
- Suppose the robot perceives z_1 for which $p(z_1 \mid x_1)=0.7$ and $p(z_1 \mid x_2)=0.3$.
- Given the observation z_1 we update the belief using Bayes rule.

$$p'_1 = \frac{0.7 p_1}{p(z_1)}$$

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

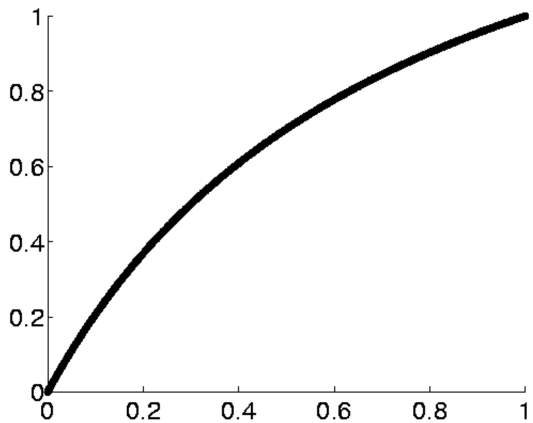
$$p'_2 = \frac{0.3(1-p_1)}{p(z_1)}$$

$$p(z_1) = 0.7 p_1 + 0.3(1-p_1) = 0.4 p_1 + 0.3$$

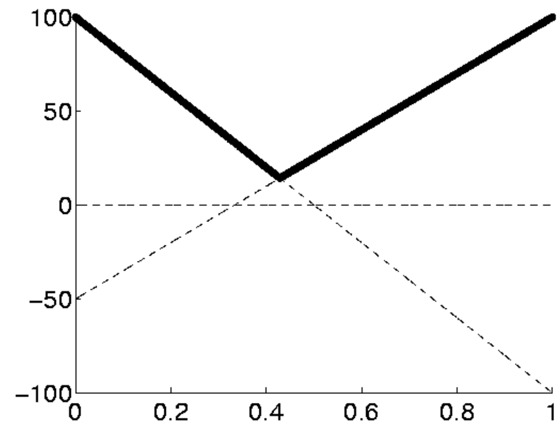
POMDP

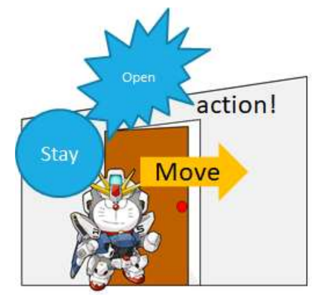
- Value function

$r(b|z_{1,}u_2)$



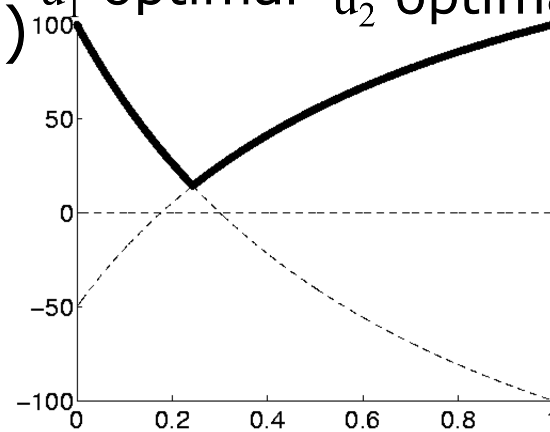
$V_1(b)$

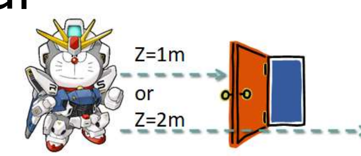


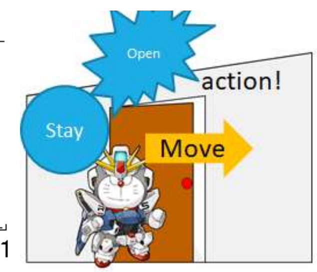


u_1 optimal u_2 optimal

$V_1(b|z_1)$







POMDP

- Assume the robot can make an observation before deciding on an action.
- **Suppose the robot perceives** z_1 for which $p(z_1 \mid x_1)=0.7$ and $p(z_1 \mid x_2)=0.3$.
- Given the observation z_1 we update the belief using Bayes rule.
- Thus $V_1(b \mid z_1)$ is given by

$$V_1(b \mid z_1) = \max \begin{cases} -100 \frac{0.7\, p_1}{P(z_1)} + 100 \frac{0.3(1-p_1)}{P(z_1)} \\ 100 \frac{0.7\, p_1}{P(z_1)} - 50 \frac{0.3(1-p_1)}{P(z_1)} \end{cases}$$

$$= \frac{1}{P(z_1)} \max \begin{cases} 70 p_1 + 30(1-p_1) \\ 70 p_1 - 15(1-p_1) \end{cases}$$

$V(S) \rightarrow$ MDP

$P(x \mid z)$

$V(b \mid z)$

POMDP

- Expected value after measuring:
- Since we do not know in advance what the *next* measurement will be, we have to compute the expected belief

$$\overline{V_1}(b) = E_z[V_1(b \mid z)] = \sum_{i=1}^{2} p(z_i) V_1(b \mid z_i)$$

$$= \sum_{i=1}^{2} p(z_i) V_1\left(\frac{p(z_i \mid x_1) p_1}{p(z_i)} \right)$$

$$= \sum_{i=1}^{2} V_1\left(p(z_i \mid x_1) p_1 \right)$$

POMDP

- Since we do not know in advance what the next measurement will be, we have to compute the expected belief

$$\bar{V}_1(b) = E_z[V_1(b \mid z)] = \sum_{i=1}^{2} p(z_i) V_1(b \mid z_i)$$

$$= \max \begin{Bmatrix} -70 p_1 + 30(1 - p_1) \\ 70 p_1 - 15(1 - p_1) \end{Bmatrix}$$

$$+ \max \begin{Bmatrix} -30 p_1 + 70(1 - p_1) \\ 30 p_1 - 35(1 - p_1) \end{Bmatrix}$$

POMDP

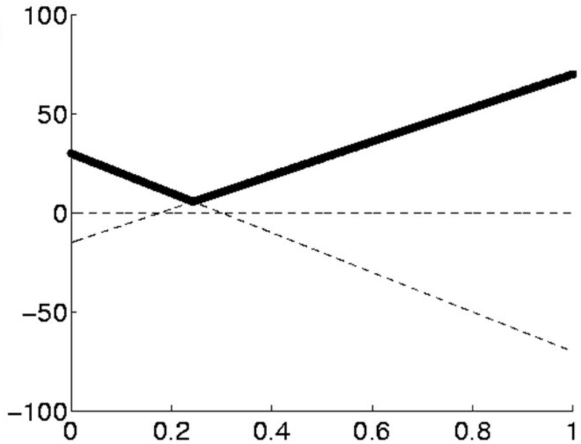
- The four possible combinations yield the following function which then can be simplified and pruned.

$$\overline{V}_1(b) = \max \begin{Bmatrix} -70p_1 + 30(1-p_1) - 30p_1 + 70(1-p_1) \\ -70p_1 + 30(1-p_1) + 30p_1 - 35(1-p_1) \\ 70p_1 - 15(1-p_1) - 30p_1 + 70(1-p_1) \\ 70p_1 - 15(1-p_1) + 30p_1 - 35(1-p_1) \end{Bmatrix}$$

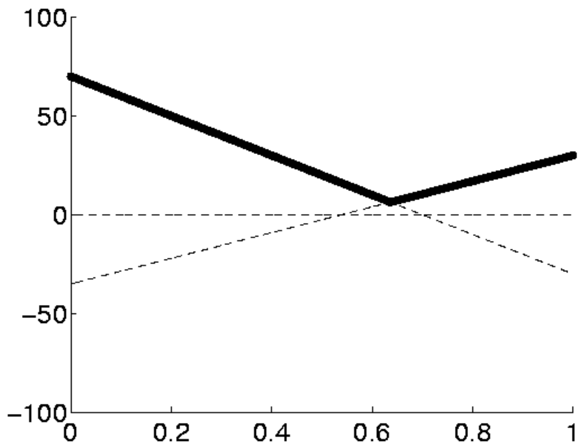
$$= \max \begin{Bmatrix} -100p_1 + 100(1-p_1) \\ +40p_1 + 55(1-p_1) \\ +100p_1 - 50(1-p_1) \end{Bmatrix}$$

POMDP

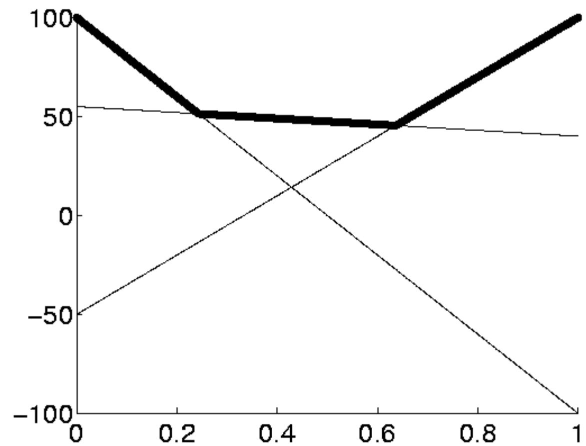
$p(z_1)V_1(b|z_1)$



$p(z_2)V_2(b|z_2)$

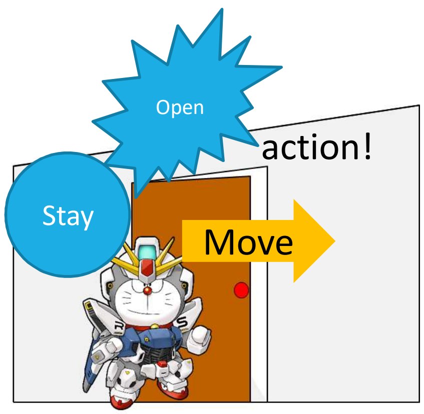


$\bar{V}_1(b)$

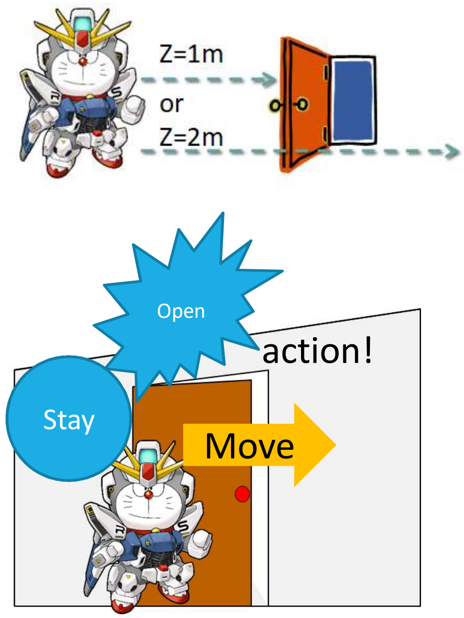


POMDP

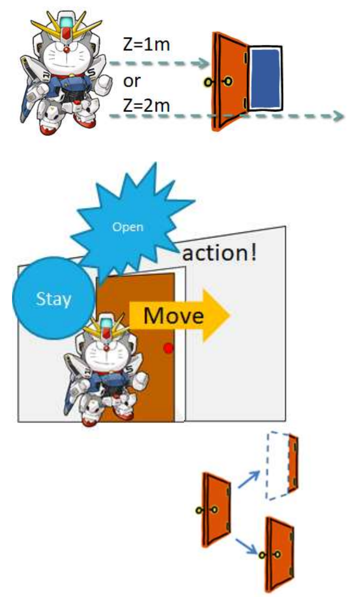
- Action only



- Sense
- Action



- **Sense**
- **Action**
- **Transition**



- Sense
- Action
- Transition
- H horizons

......

POMDP

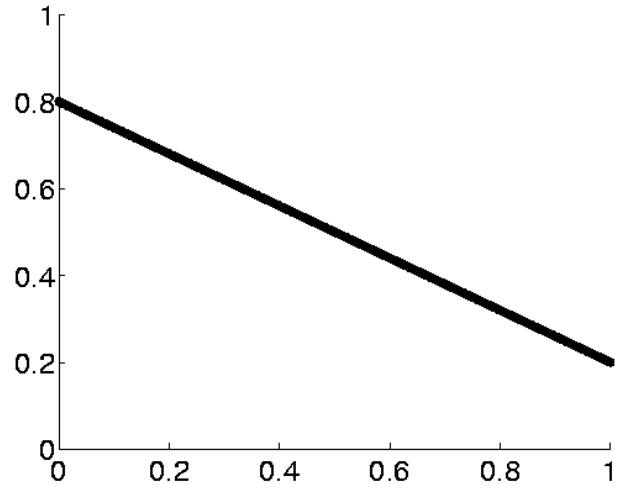
- State Transitions (Prediction):
- When the agent selects u_3 its state potentially changes.
- When computing the value function, we have to take these potential state changes into account.

$$p_1' = E_x\left[P(x_1 \mid x, u_3)\right]$$

$$= \sum_{i=1}^{2} P(x_1 \mid x_i, u_3) p_i$$

$$= 0.2 p_1 + 0.8(1 - p_1)$$

$$= 0.8 - 0.6 p_1$$



$$\begin{bmatrix} 0.2 & 0.8 \\ 0.8 & 0.2 \end{bmatrix} \begin{bmatrix} p_1 \\ 1 - p_1 \end{bmatrix}$$

POMDP

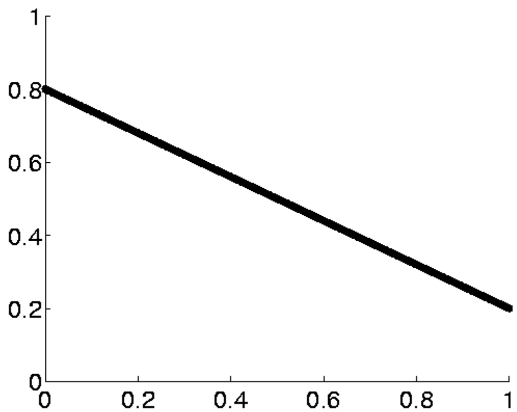
- Resulting Value Function after executing u_3
- Taking the state transitions into account, we finally obtain.

$$\bar{V}_1(b) = \max \begin{Bmatrix} -100\,p_1 + 100(1-p_1) \\ +40\,p_1 + 55(1-p_1) \\ +100\,p_1 - 50(1-p_1) \end{Bmatrix}$$

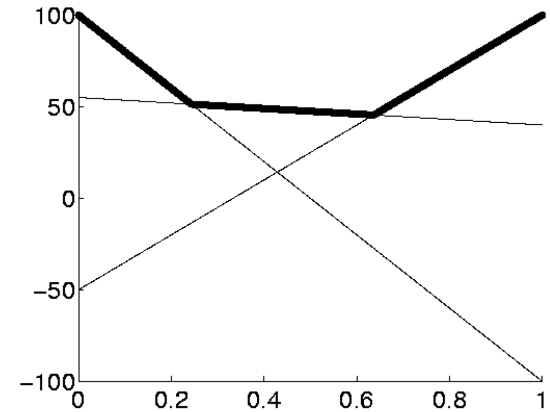
$$\bar{V}_1(b \mid u_3) = \max \begin{Bmatrix} 60\,p_1 - 60(1-p_1) \\ 52\,p_1 + 43(1-p_1) \\ -20\,p_1 + 70(1-p_1) \end{Bmatrix}$$

POMDP

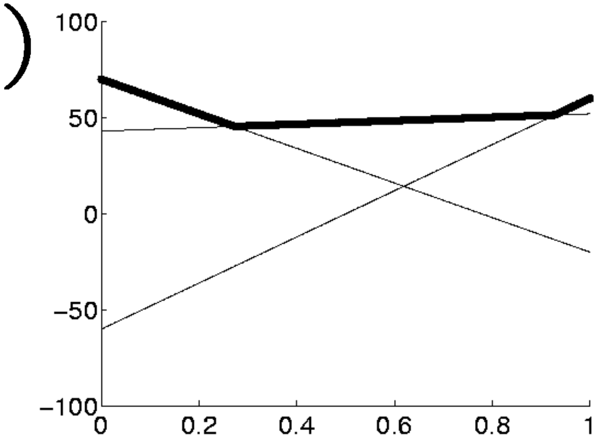
$$p_1^{'} = 0.8 - 0.6 p_1$$



$$\bar{V}_1(b)$$

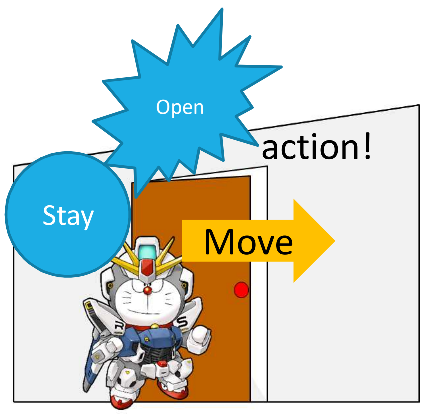


$$\bar{V}_1(b \mid u_3)$$

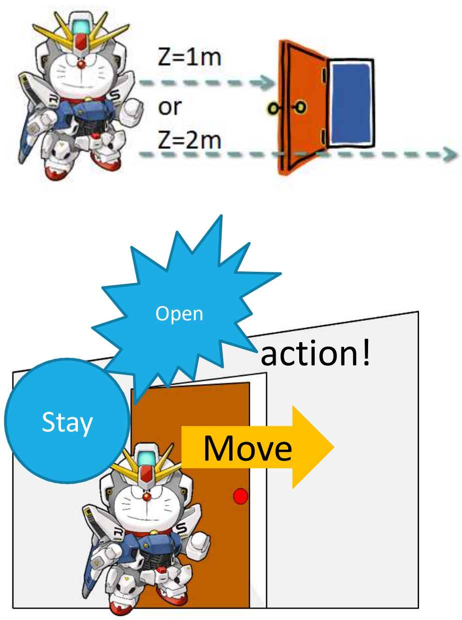


POMDP

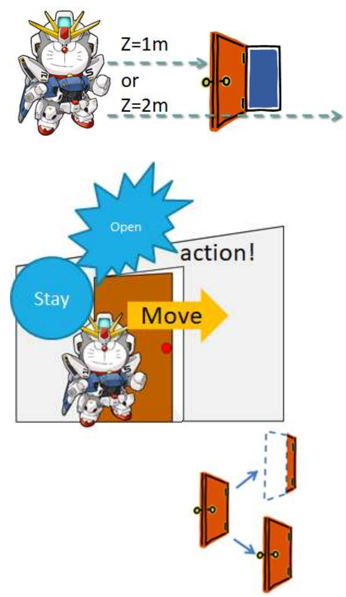
- Action only



- Sense
- Action



- Sense
- Action
- Transition



- **Sense**
- **Action**
- **Transition**
- **H horizons**

......

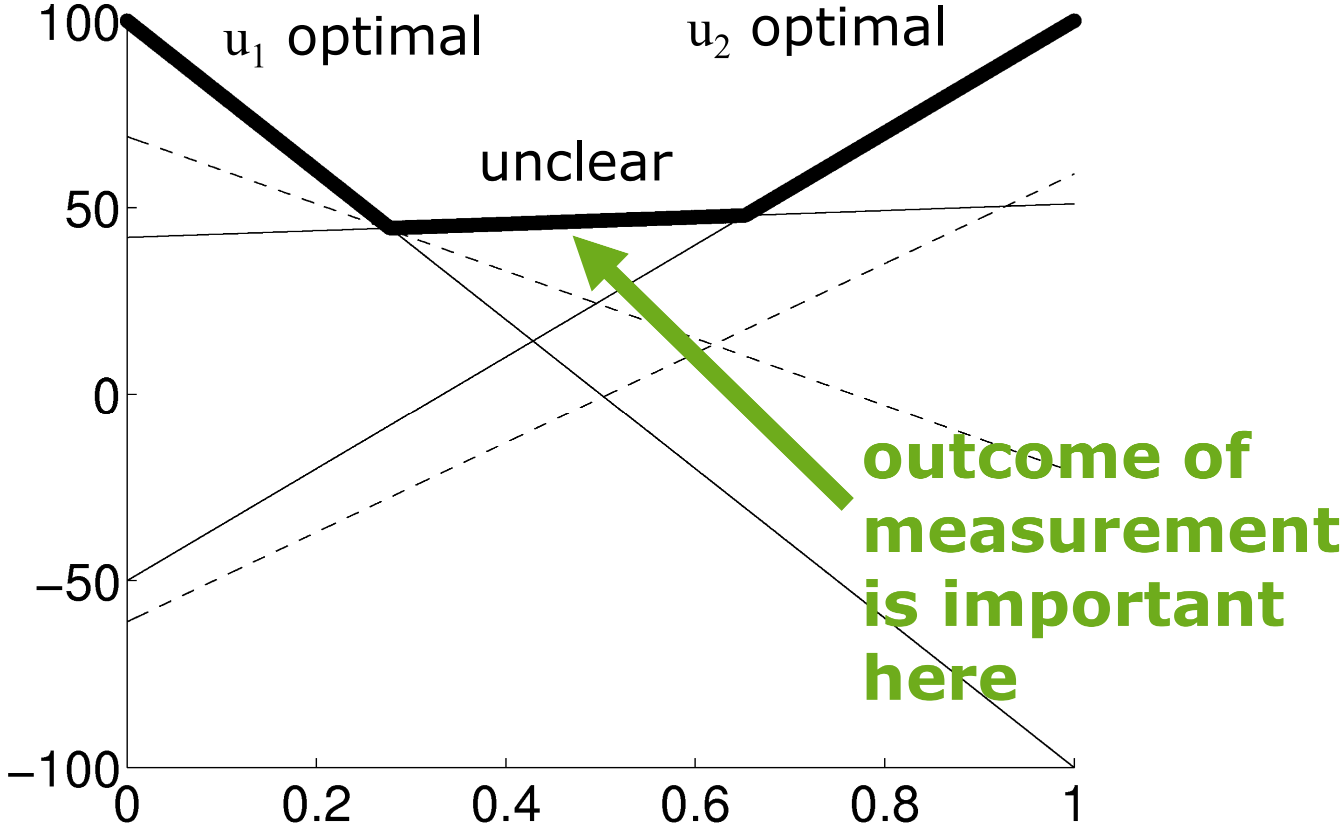
POMDP

- Value Function for T=2:
- Taking into account that the agent can either directly perform u_1 or u_2 or first u_3 and then u_1 or u_2, we obtain (after pruning)

$$\bar{V}_2(b) = \max \begin{Bmatrix} -100\,p_1 + 100(1-p_1) \\ 100\,p_1 - 50(1-p_1) \\ 51\,p_1 + 42(1-p_1) \end{Bmatrix}$$

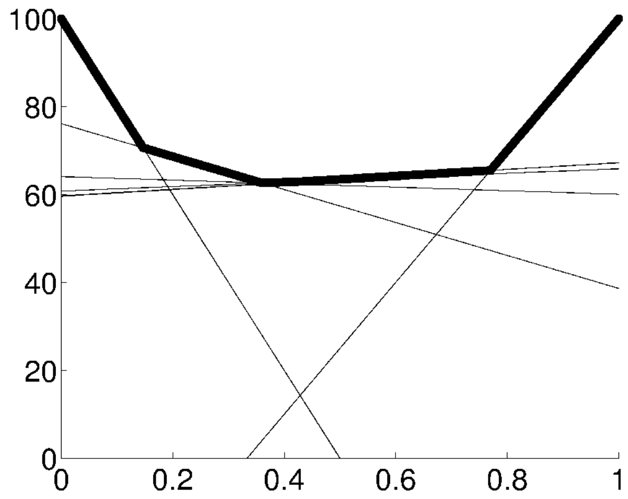
POMDP

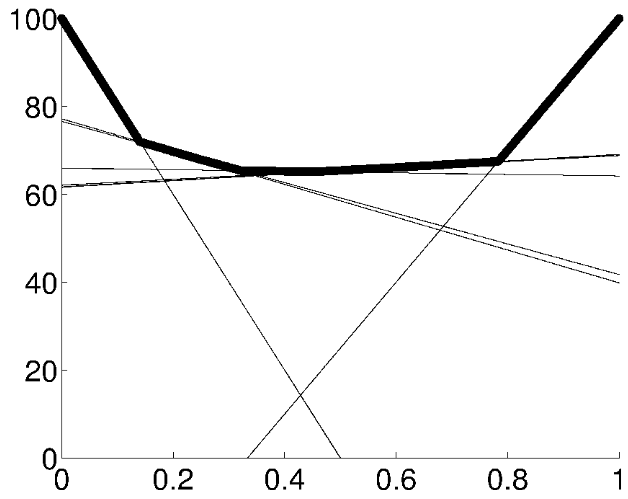
- Graphical Representation of $V_2(b)$



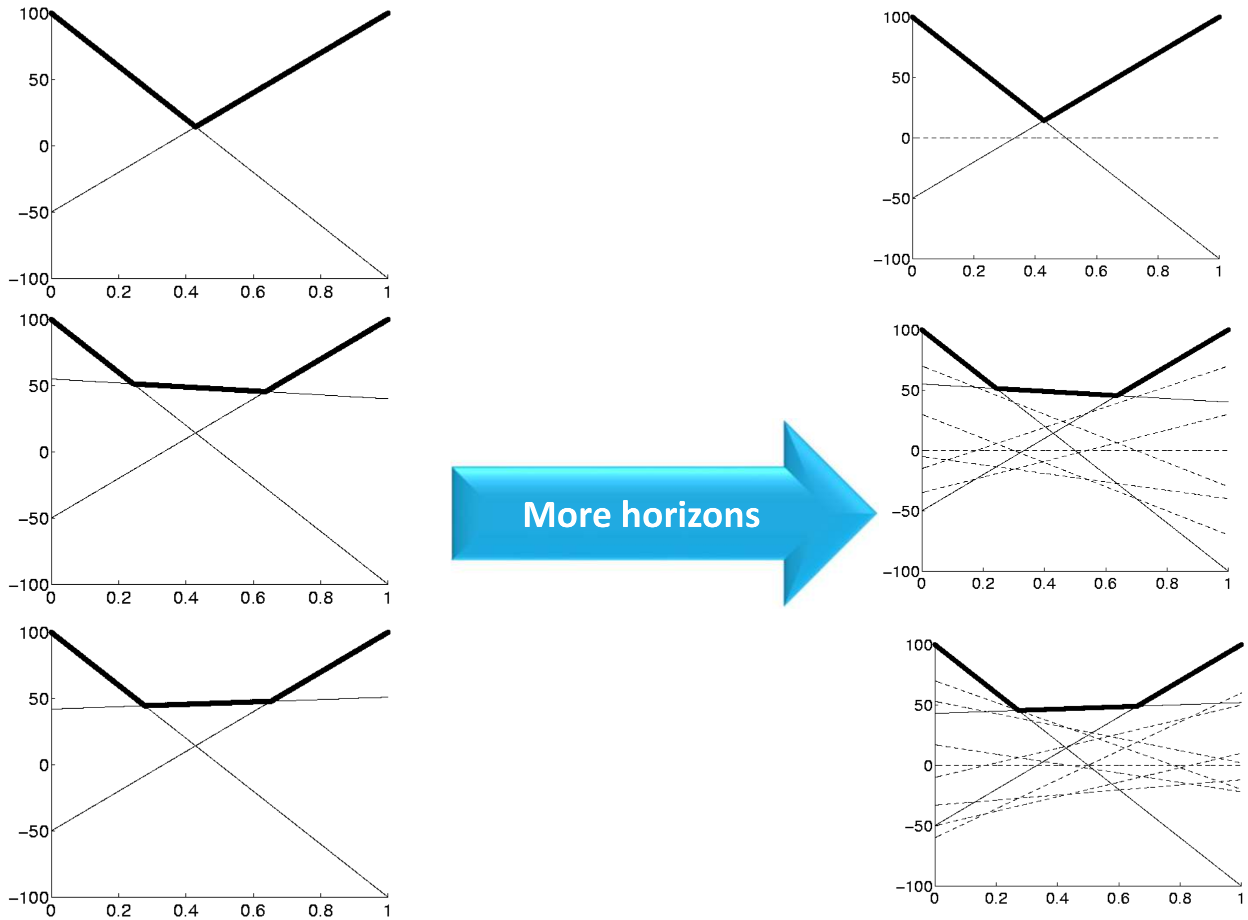
POMDP

- Deep Horizons and Pruning:
- We have now completed a full backup in belief space.
- This process can be applied recursively.
- The value functions for T=10 and T=20 are





POMDP



Algorithm POMDP(T):

$\Upsilon = (0, \ldots, 0)$
for $\tau = 1$ *to* T *do*
  $\Upsilon' = \emptyset$
  for all $(u'; v_1^k, \ldots, v_N^k)$ *in* Υ *do*
    for all control actions u *do*
      for all measurements z *do*
        for $j = 1$ *to* N *do*

$$v_{j,u,z}^k = \sum_{i=1}^{N} v_i^k \; p(z \mid x_i) \; p(x_i \mid u, x_j)$$

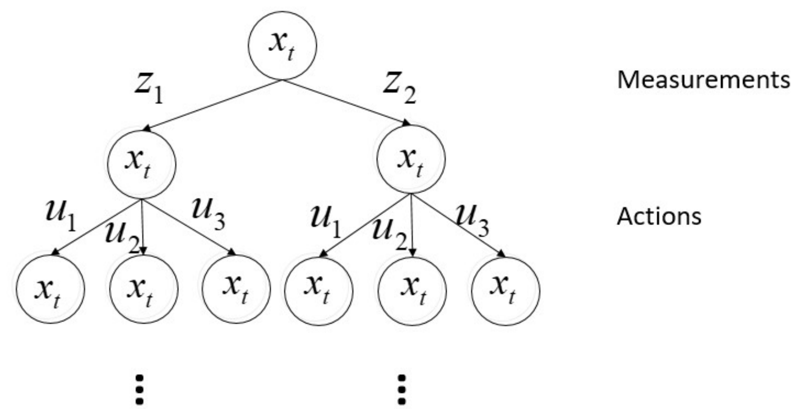
        endfor
      endfor
    endfor
  endfor
  for all control actions u *do*
    for all $k(1), \ldots, k(M) = (1, \ldots, 1)$ *to* $(|\Upsilon|, \ldots, |\Upsilon|)$ *do*
      for $i = 1$ *to* N *do*

$$v_i' = \gamma \left[r(x_i, u) + \sum_z v_{u,z,i}^{k(z)} \right]$$

      endfor
      add $(u; v_1', \ldots, v_N')$ *to* Υ'
    endfor
  endfor
  optional: prune Υ'
  $\Upsilon = \Upsilon'$
endfor
return Υ

$|A| = 3$
|Z|=2
d=11
$3^{O(2\text{^}\{d-1\})}$
=3^{1024} .....@@"

$|A|$: *actions*
$|Z|$: *observations*
d : *depth*
$|A|^{O(|Z|^{d-1})}$ paths



POMDP

- Why Pruning is Essential?
- Each **update introduces additional linear components** to V.
- Each **measurement squares the number of linear components**.
- Thus, an un-pruned value function for T=20 includes more than $10^{547,864}$ linear functions.
- At T=30 we have $10^{561,012,337}$ linear functions.
- The pruned value functions at T=20, in comparison, contains only 12 linear components.
- The combinatorial explosion of linear components in the value function are the major reason why **POMDPs are impractical for most applications**.

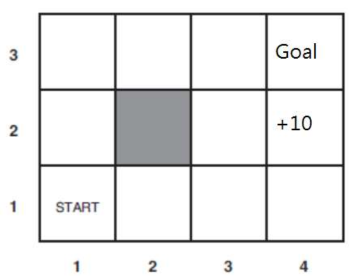
POMDP Summary

- POMDPs compute the optimal action in partially observable, stochastic domains.
- For finite horizon problems, the resulting value functions are piecewise linear and convex.
- In each iteration the number of linear constraints grows exponentially.
- POMDPs so far have only been applied successfully to very small state spaces with small numbers of possible observations and actions.

Conclusions

- LRTA*

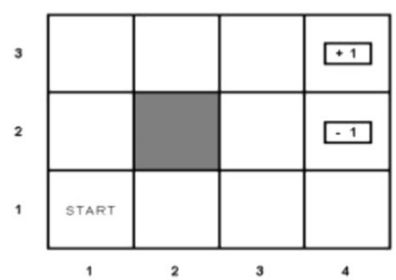
Deterministic action



$$s, a \rightarrow s'$$

MDP (RL)

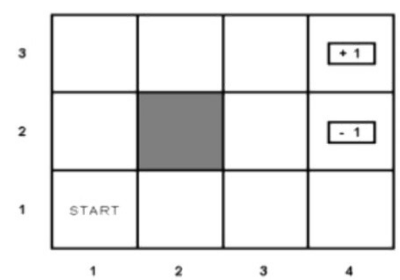
Probabilistic actions



$$P(s' | s, a)$$

POMDP

Probabilistic actions and states



$$P(s' | s, a), P(s)$$

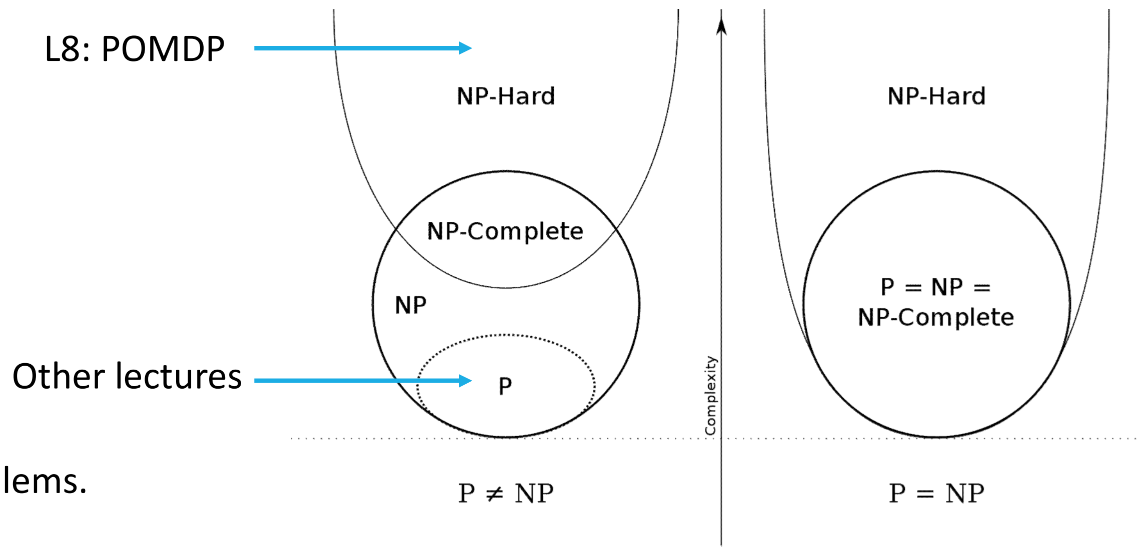
L2: Uninformed search
L3: Heuristic search (LRTA*)
L4: Adversarial search
L5: Bayes theorem
L6: Bayes theorem over time

L7: MDP
L9: Reinforcement learning
L10: GP and LWPR
L11: Naïve Bayes and Perceptron
L12: Adaboost
L13: Deep learning and DRL

L8: POMDP

Appendix – NP-hard problems

- NP-hardness (non-deterministic polynomial-time hardness)
- NP-complete: Class of decision problems which contains the hardest problems in NP. Each NP-complete problem has to be in NP.



In MAI, we will try to solve P problems.
In L8, we will face POMDP, one of NP-hard problems.

https://en.wikipedia.org/wiki/NP-hardness

Q&A