Introduction to Robotics

Felipe P. Vista IV

Class Admin Matters

Grading

➢ Attendance 5%

Name (Original Name)	User Email	Join Time	Leave Time	Duration (Minutes)
[redacted]		4/12/2021 9:12	4/12/2021 10:14	62
		4/12/2021 9:12	4/12/2021 9:14	3
		4/12/2021 9:12	4/12/2021 9:14	3
		4/12/2021 9:12	4/12/2021 9:14	3
		4/12/2021 9:12	4/12/2021 9:14	3
		4/12/2021 9:12	4/12/2021 9:14	3
		4/12/2021 9:13	4/12/2021 9:13	1
		4/12/2021 9:13	4/12/2021 9:14	2
		4/12/2021 9:14	4/12/2021 9:14	1
		4/12/2021 9:14	4/12/2021 9:14	1
		4/12/2021 9:14	4/12/2021 10:14	60

Bad ZOOM User Name (Absent)

➢ Iphone → Not your name

➢ SiAko 202100001 → Wrong order

➢ SiAko → Name only

➢ 202100001 → ID Num only

ZOOM User Name (Present)

➢ University ID Num_Name

➢ 202100001 SiAko → GOOD (Present)

Name (Original Name)	User Email	Total Duration (Minutes)
[redacted]		62
		63
		62
		62
		63
		62
		63

Student Responsibilities

- Download/Install ZOOM app for online lecture
  - *Zoom profile must be your **OASIS ID+name** similar to OASIS*
  - *Ex.: **202061234 YourName***
  - *If you are asked, but no reply, then you'll be out of zoom & mark absent*
- Regularly login, check OLD IEILMS for updates, notifications
  - *https://ieilmsold.jbnu.ac.kr*
  - *Presentations & lecture videos will be uploaded after class*
- Regularly check Kakao Group Chat for class
  - *Everybody must have a Kakao talk account*
  - *Search & add account "**botjok**", introduce yourself and name of class ("Robotics"), then you will be added to the group chat*

Intro To Robotics

LOCALIZATION

Intro

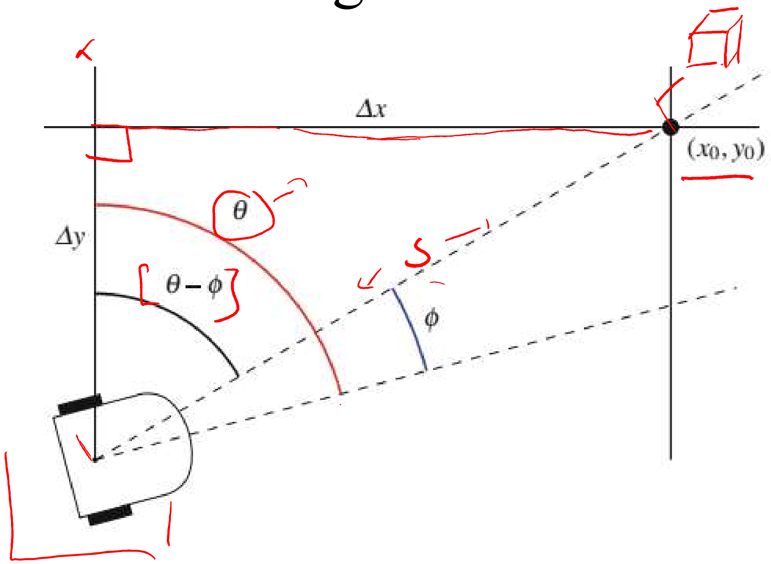
- Navigation by odometry
  - *Prone to errors; Only estimate real pose of robot*
  - *Especially heading : farther robot moves → larger error in estimation*
- Odometry
  - *Like human walking w/ closed eyes (Counting steps until goal reached)*
  - *Farther walked → more uncertain about location*
    - Open eyes once in a while to reduce uncertainties
- For a robot
  - *Count steps?? Open eyes once in a while??*
  - *Short distance → Odometry enough*
  - *Longer distance → Determine position relative to external reference called landmark → This process is* ***localization***

Localization

- Landmarks
- Determining Position from Objects whose Position is Known
- Global Positioning System
- Probabilistic Localization
- Uncertainty in Motion

Landmarks

- Landmarks
  - *Like lines on the ground, doors in a corridor*
  - *Can be identified by the robot & used for localization*
- Determining Position from an Angle & a Distance



Geometry of a robot relative to an object

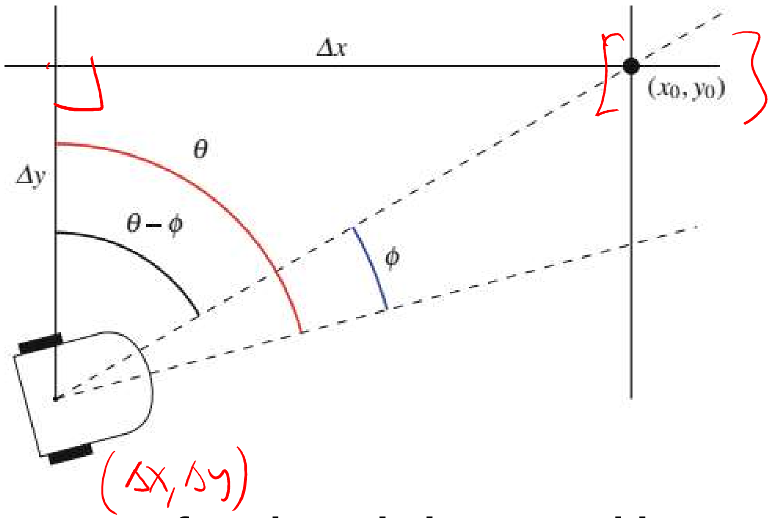
- (x_0, y_0) - *known coordinates of object at origin of coordinate system*
- θ - *azimuth, angle bet* North & forward *direction of the robot (compass)*
- ϕ - *angle bet* forward *dir of robot & object*
- s - *distance to the object (laser scanner)*
- $(\Delta x, \Delta y)$ - *relative coordinates*

The relative coordinates can be derived by:

? ? ? ? ?

Landmarks

- Landmarks
  - *Like lines on the ground, doors in a corridor*
  - *Can be identified by the robot & used for localization*
- Determining Position from an Angle & a Distance



Geometry of a robot relative to an object

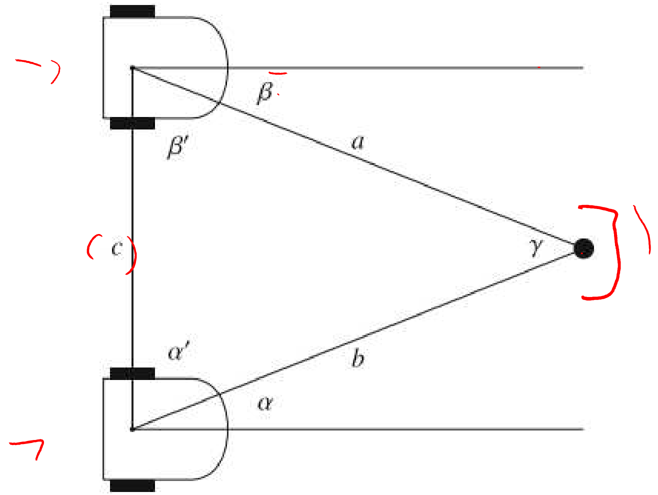
- (x_0, y_0) - *known coordinates of object at origin of coordinate system*
- θ - *azimuth, angle bet* North & forward *direction of the robot (compass)*
- ϕ - *angle bet* forward *dir of robot & object*
- s - *distance to the object (laser scanner)*
- $(\Delta x, \Delta y)$ - *relative coordinates*

The relative coordinates can be derived by:

$$\Delta x = s\sin(\theta - \phi), \quad \Delta y = s\cos(\theta - \phi)$$

Det Position by Triangulation

- Triangulation
  - *Determine coordinates when difficult/impossible to measure distances*
  - *Used before lasers, since impossible to accurately measure long distances*
- Principle of triangulation
  - Given : *Two angles, length included side* → Derive : *length of the other sides*
  - Hence : *Relative position of distance object can be calculated*

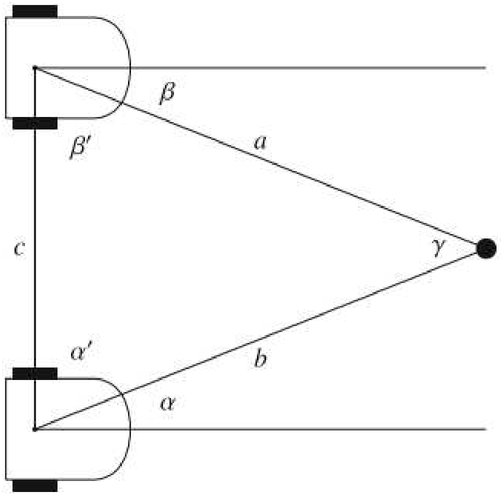


- ***α***, ***β*** - angles to object
- ***c*** - distance separating two positions
- ***a***, ***b*** - lengths

In surveying :
? ? ? ? ?

Det Position by Triangulation

- Triangulation
  - *Determine coordinates when difficult/impossible to measure distances*
  - *Used before lasers, since impossible to accurately measure long distances*
- Principle of triangulation
  - Given : *Two angles, length included side* → Derive : *length of the other sides*
  - Hence : *Relative position of distance object can be calculated*



- ***α***, ***β*** - angles to object
- ***c*** - distance separating two positions
- ***a***, ***b*** - lengths

In surveying :
- *If two coordinates known → distance between can be computed → determine coordinates of object.*

Det Position by Triangulation

Lengths ***a***, ***b*** computed using **law of sines**:

$$\frac{a}{\sin\alpha'} = \frac{b}{\sin\beta'} = \frac{c}{\sin\gamma}$$

where $\alpha' = 90° - \alpha, \beta' = 90° - \beta$ are interior angles.

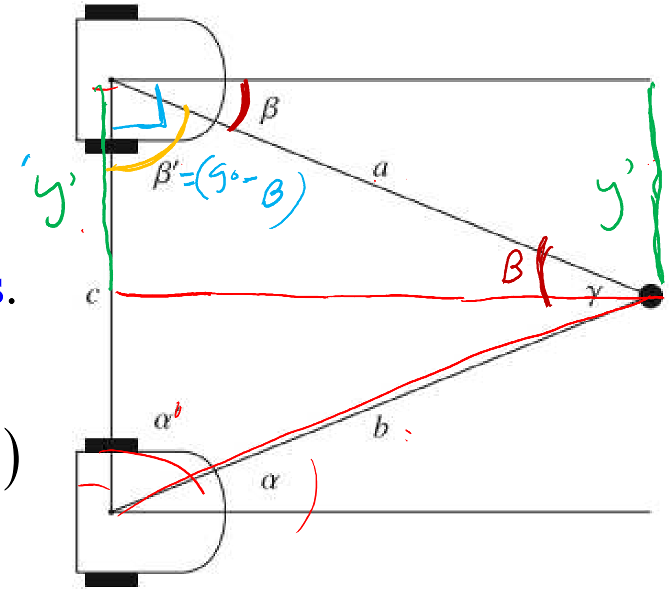
To use the law, we need ***c*** (measured), and ***γ*** is:

$$\gamma = 180° - \alpha' - \beta' = 180° - (90° - \alpha) - (90° - \beta)$$
$$= \alpha + \beta$$

Using law of sines:

$$b = \frac{c\sin\beta'}{\sin\gamma} = \frac{c\sin(90° - \beta)}{\sin(\alpha+\beta)} = \frac{c\cos\beta}{\sin(\alpha+\beta)}$$

Similar computation will be used for ***a***.



$$a = \frac{c\sin\alpha'}{\sin\gamma} = \frac{c\sin(90°-\alpha)}{\sin(\alpha+\beta)}$$

Localization

- Landmarks
- Determining Position from Objects whose Position is Known
- Global Positioning System
- Probabilistic Localization
- Uncertainty in Motion

Global Positioning System

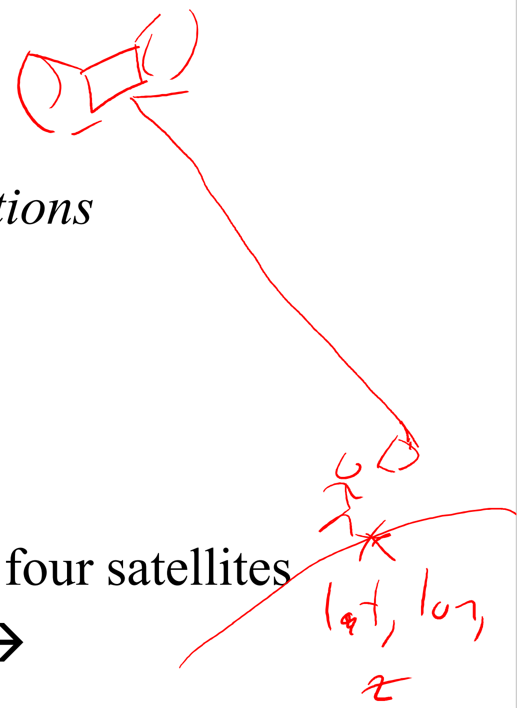
- Global Positioning System (GPS)
  - *Determination of location easier and more accurate*
  - *Each satellite know its*

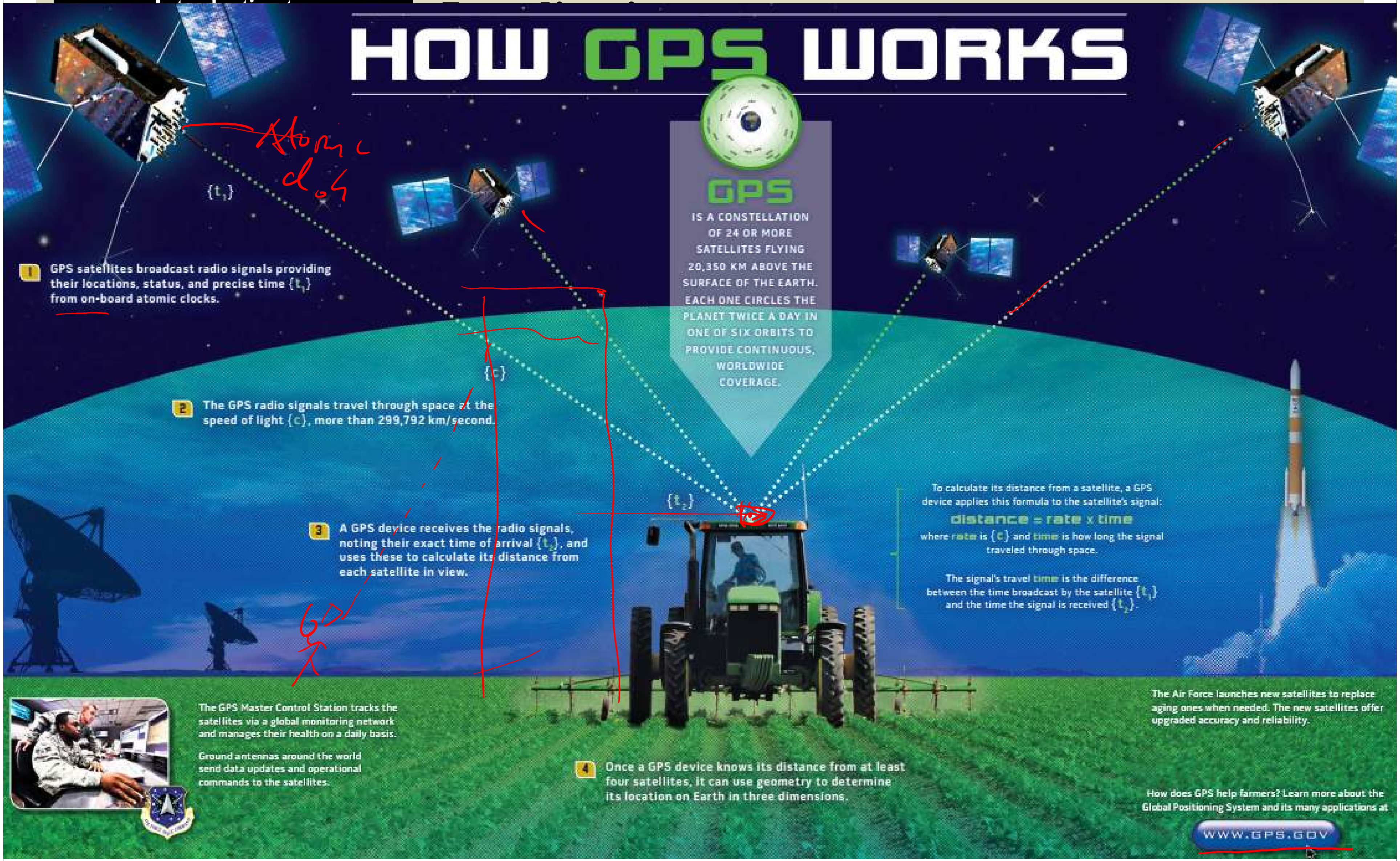
    * Precise position : *Ground stations w/ complex calculations*

    * Local time : *Highly accurate atomic clock*
- GPS receiver
  - *Able to receive data from* four *satellites*

    * 24 - 32 satellites

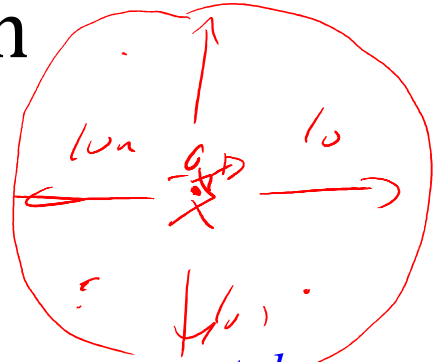
    * so that always line-of-sight bet any location & at least four satellites
  - *(Time signal travelled from satellite) * (Speed of light)* → *Distance from satellites*
  - *Computation using distance & satellite location* → *3D position of receiver (latitude, longitude, elevation)*





HOW GPS WORKS
GPS IS A CONSTELLATION OF 24 OR MORE SATELLITES FLYING 20,350 KM ABOVE THE SURFACE OF THE EARTH. EACH ONE CIRCLES THE PLANET TWICE A DAY IN ONE OF SIX ORBITS TO PROVIDE CONTINUOUS, WORLDWIDE COVERAGE.
{t1}
1 GPS satellites broadcast radio signals providing their locations, status, and precise time {t1} from on-board atomic clocks.
{c}
2 The GPS radio signals travel through space at the speed of light {c}, more than 299,792 km/second.
{t2}
3 A GPS device receives the radio signals, noting their exact time of arrival {t2}, and uses these to calculate its distance from each satellite in view.
To calculate its distance from a satellite, a GPS device applies this formula to the satellite's signal:
distance = rate x time
where rate is {c} and time is how long the signal traveled through space.
The signal's travel time is the difference between the time broadcast by the satellite {t1} and the time the signal is received {t2}.
The GPS Master Control Station tracks the satellites via a global monitoring network and manages their health on a daily basis.
Ground antennas around the world send data updates and operational commands to the satellites.
4 Once a GPS device knows its distance from at least four satellites, it can use geometry to determine its location on Earth in three dimensions.
The Air Force launches new satellites to replace aging ones when needed. The new satellites offer upgraded accuracy and reliability.
How does GPS help farmers? Learn more about the Global Positioning System and its many applications at
WWW.GPS.GOV
Atomic clock

Global Positioning System



- Advantage
  - *Accurate & available anywhere*
  - *No additional equipment needed*
  - *Small & cheap electrical component only → found in every smartphones*
- Main problems with GPS navigation
  - *Position error roughly 10m* :
    - Good for navigating car to choose correct road at intersection
    - Not sufficient enough for higher accuracy tasks (i.e. parking)
  - *Not strong enough for indoor navigation*
    - Subject to interference in dense urban environments
- Global Navigation Satellite Systems (GNSS)
  - *GPS (USA), BeiDou (China), GLONASS (Russia), Galileo (EU)*

civilian = 1-5m
military = 0.1m

Global Positioning System

GPS utilizes both of Einstein's Special & General Relativity:

General Relativity

- *Deals with space-time continuum*
- *Clocks on satellite run **faster** (by 45.9 µsecs/day) since force of Earth's gravity is smaller at distant satellite that in is for us on the surface*

Special Relativity

- *Deals only with inertial frames*
- *Clocks on satellites run **slower** (by 7.2 µsec/day) since satellites move fast relative to Earth.*

*** Their effects don't cancel out, correction factor used when broadcasting the time signals.*

[1] Ashby, N., "Relativity in the Global Positioning System", Living Reviews in Relativity, 6(1), 2003.

Localization

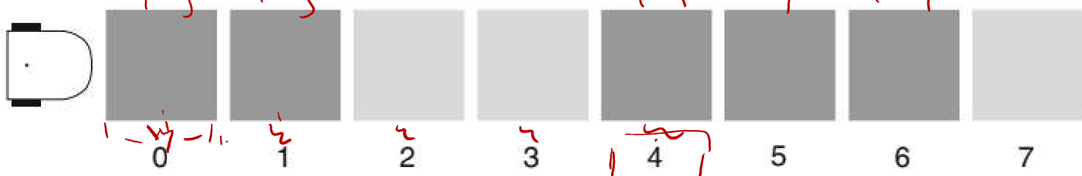
- Landmarks
- Determining Position from Objects whose Position is Known
- Global Positioning System
- Probabilistic Localization
- Uncertainty in Motion

Probalistic Localization

- **Ex. : Robot enter a specific door (Pos 4)**
  - *Navigating w/in a known environment for w/c it has a map*
  - *Floor map show wall with doors (dark gray) & areas w/o a door (light gray)*
  - *Robot measure intensity using ground sensors*
- How can robot know where it is? By odometry…
  - *Determine current pos given known starting position*
  - *If at left end of wall* → ***? ? ? ? ?***
  - *If at Pos 3* → ***? ? ? ? ?***

Probalistic Localization

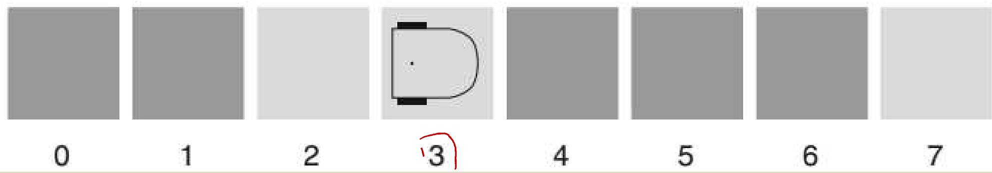
- **Ex. : Robot enter a specific door (Pos 4)**
  - *Navigating w/in a known environment for w/c it has a map*
  - *Floor map show wall with doors (dark gray) & areas w/o a door (light gray)*
  - *Robot measure intensity using ground sensors*
- How can robot know where it is? By odometry…
  - *Determine current pos given known starting position*
  - *If at left end of wall → knows it must move (5 x width) of door*



$s = v t$

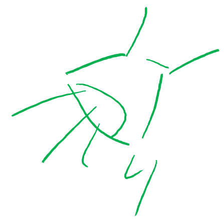
$s = \frac{dist}{t}$

  - *If at Pos 3 → target door is next one to the right of Pos 3*



Sensing Increases Certainty

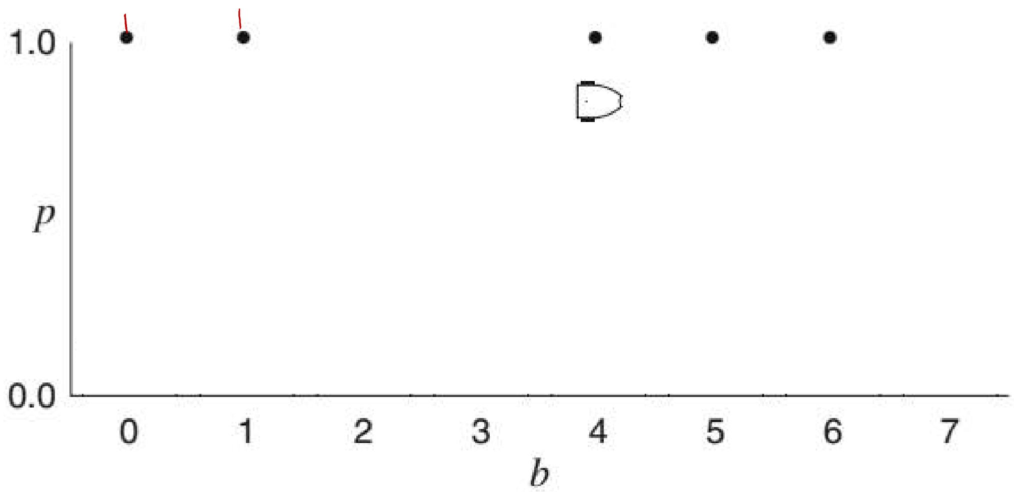
- Due to errors in Odometry
  - *Quite possible robot will be lost*
  - *Hence, utilize a one-dimensional version of Markov localization algo*
- Markov localization algorithm
  - *Take into account uncertainty in sensors and in the robot's motion*
  - *Return most probable locations of the robot*
- Recall the robot
  - *In known environment of walls and doors*
  - *Has no information as to its location*



Sensing Increases Certainty

- Robot assigns probability to positions where it might be located

1. **Initially**, no idea where it is
   - *? ? ? ? ?*

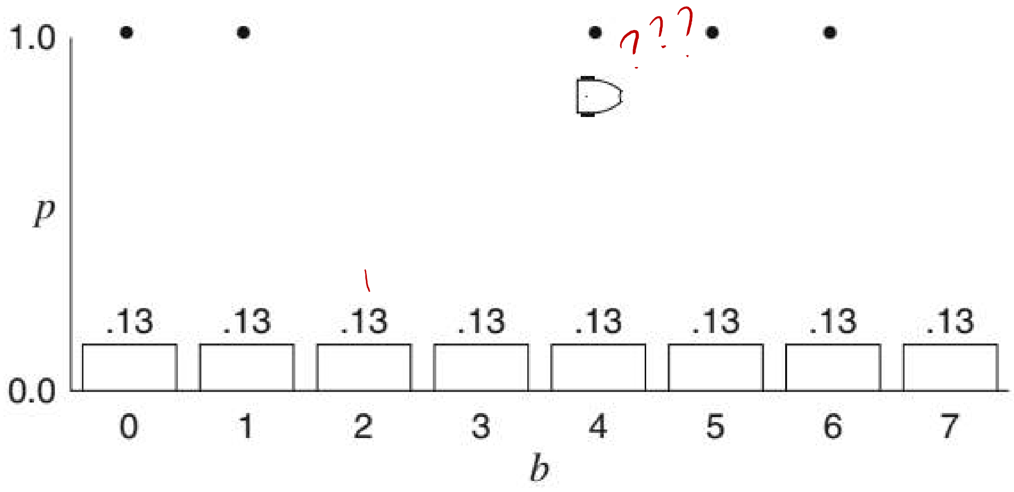


- **Dots** - *position of doors*
- **Small robo icon** - *actual position of the robot (facing right)*

Sensing Increases Certainty

- Robot assigns probability to positions where it might be located

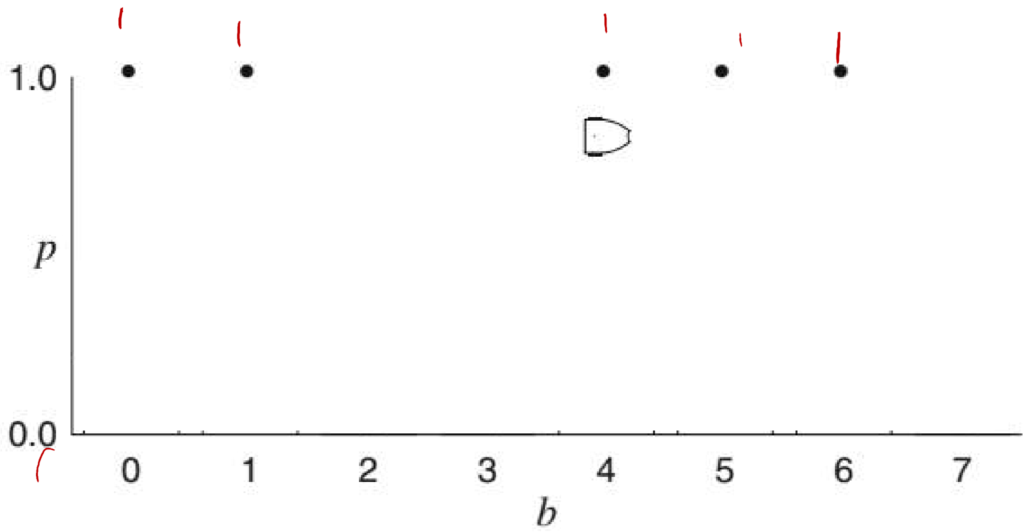
1. **Initially**, no idea where it is
   – *Hence, each pos probability* $b[i] = 1.0/\ 8 = 0.125 \approx 0.13$,
   where ***b*** *is* **belief array**.



- **Dots** - *position of doors*
- **Small robo icon** - *actual position of the robot (facing right)*

Sensing Increases Certainty

2. **Suppose now**, robot detect a gray area → uncertainty is reduced
   - *Since it knows it must be in front of one of the five doors*
   - ***? ? ? ? ?***

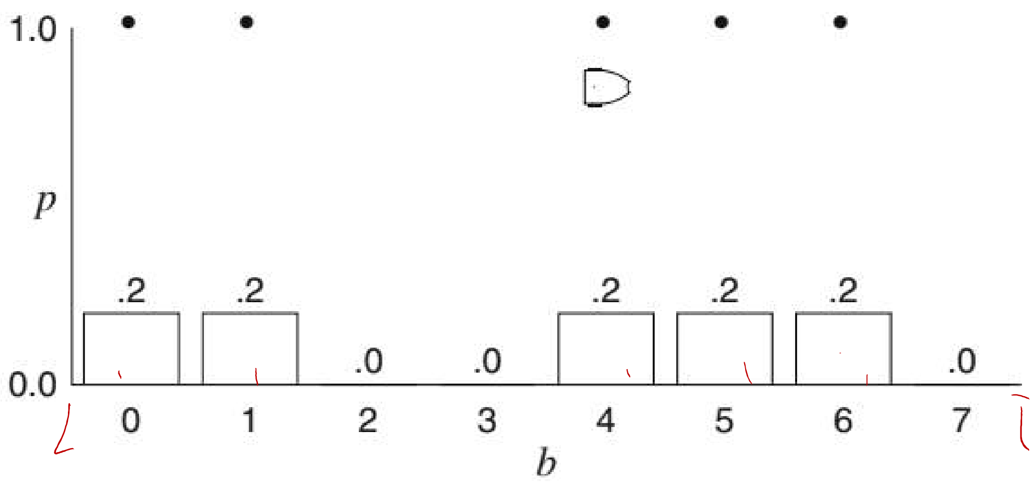


- **Dots** - *position of doors*
- **Small robo icon** - *actual position of the robot (facing right)*

Sensing Increases Certainty

2. **Suppose now**, robot detect a gray area → uncertainty is reduced
   - *Since it knows it must be in front of one of the five doors*
   - *Hence,* **belief array** *shows* **0.2** *(each of the doors)* & **0.0** *(for the walls)*

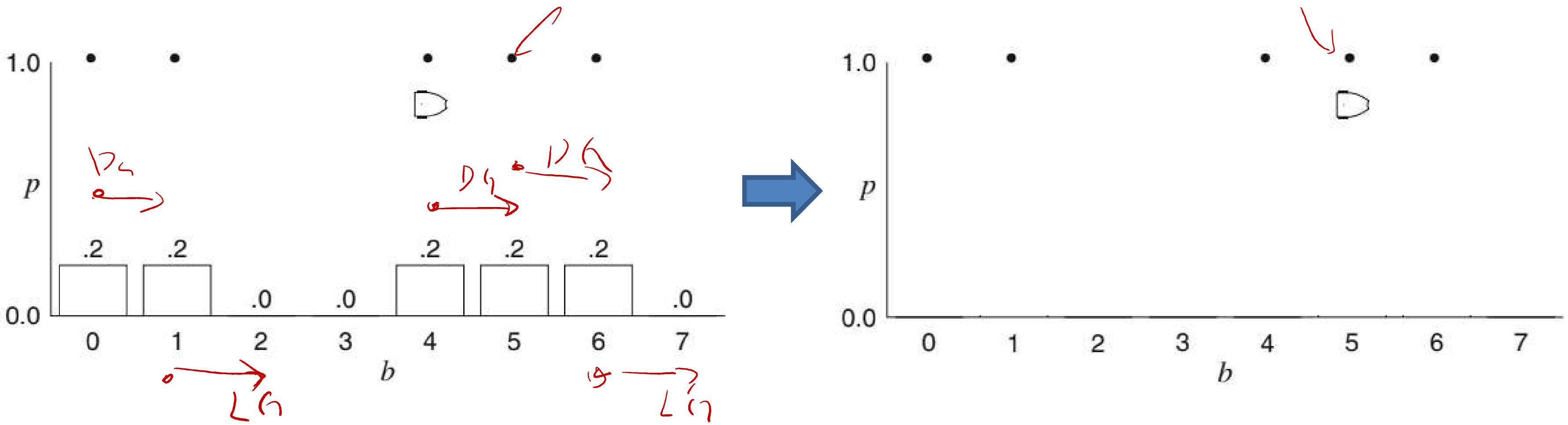
1/5 = 0.2



- **Dots** - *position of doors*
- **Small robo icon** - *actual position of the robot (facing right)*

Sensing Increases Certainty

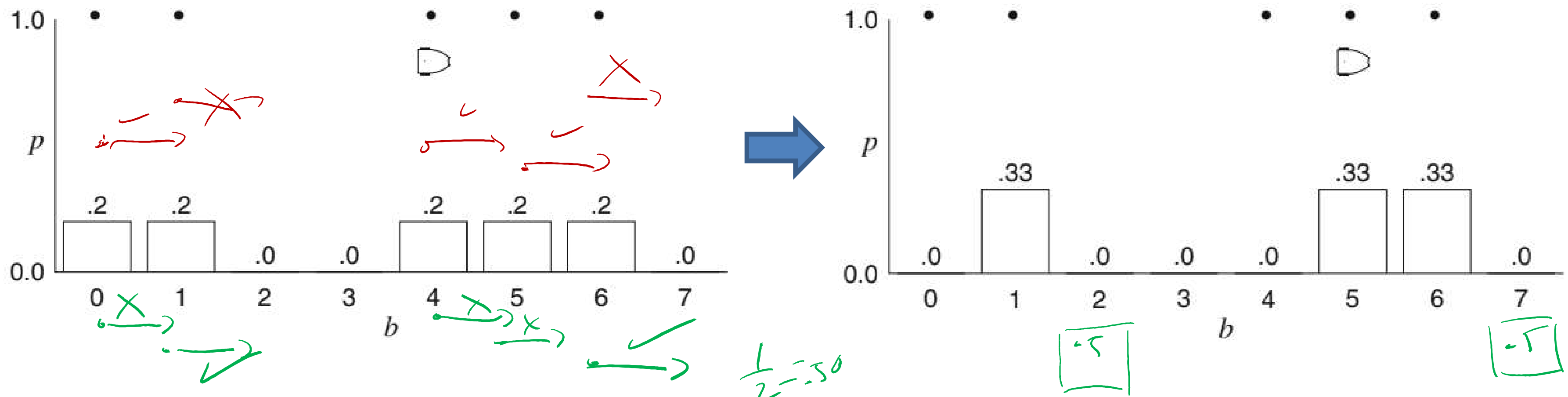
3. **Robot move forward** → again senses dark gray area.
   - *Now only* ***? ? ? ? ?***
     ? ? ? ? ?
   - *Not the following since* ***? ? ? ? ?***
     ? ? ? ? ?
   - *? ? ? ? ?*



Sensing Increases Certainty

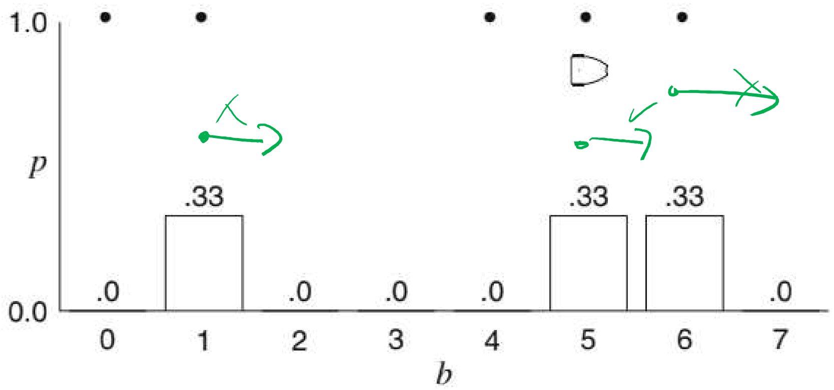
3. **Robot move forward** → again senses dark gray area.
   - *Now only* three *possibilities (Prev → Curr)* : (1, 5, 6)
     Pos 0 → Pos 1, Pos 4 → Pos 5, Pos 5 → Pos 6
   - *Not the following since its new current is not dark gray area, (Pos 2, Pos 7):*
     Pos 1 → Pos 2, Pos 6 → Pos 7
   - *Hence, probability is now* **0.33** *for each of the three positions (1, 5, 6).*

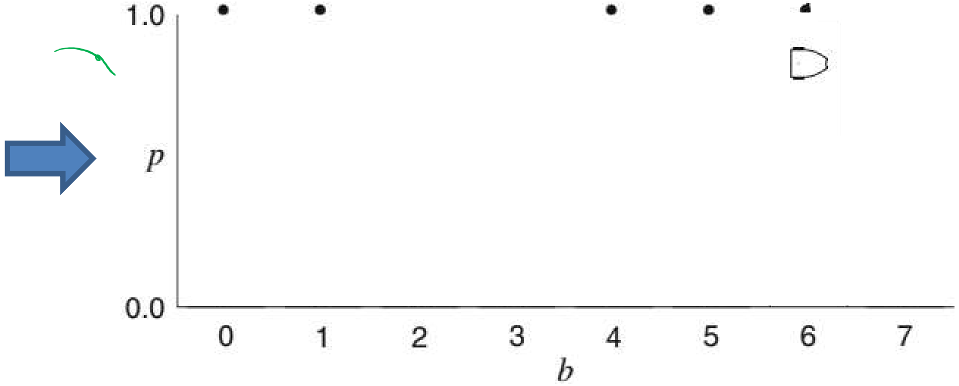
1/3 = 0.33



Sensing Increases Certainty

4. **Robot move forward again** → again senses dark gray area (door).
   - ? ? ? ? ?
     ? ? ? ? ?
   - *Not the following since* ? ? ? ? ?
     ? ? ? ? ?
   - *Hence, probability* ? ? ? ? ?





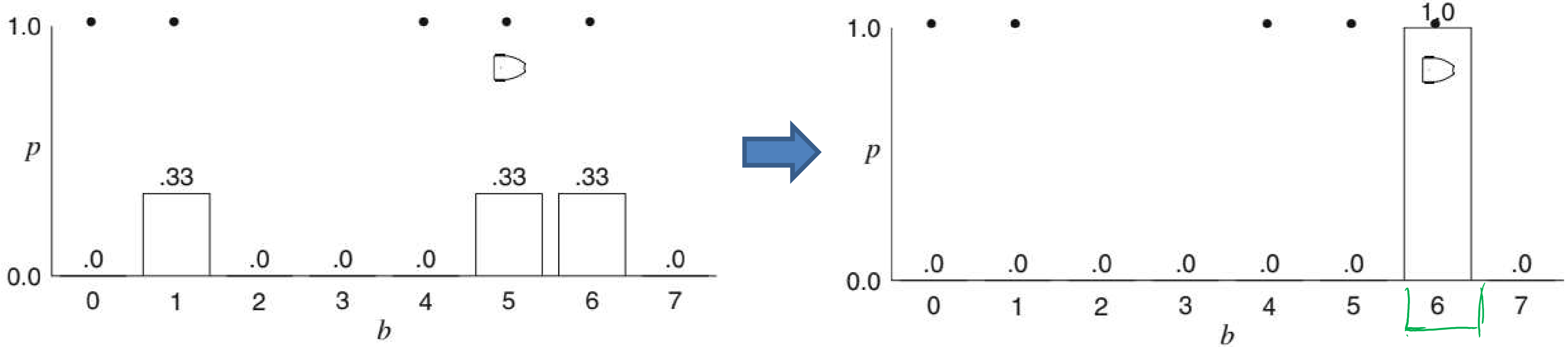
Sensing Increases Certainty

4. Robot **move forward again**→ again senses dark gray area (door).

– *Now only one possibility (Prev → Curr) :*
Pos 5 → Pos 6
1.0 (100%) DG → Door

– *Not the following since its new current is not dark gray area : (Pos 2, Pos 7)*
Pos 1 → Pos 2, Pos 6 → Pos 7

– *Hence, probability is now* **1.0** *for Pos 6.*

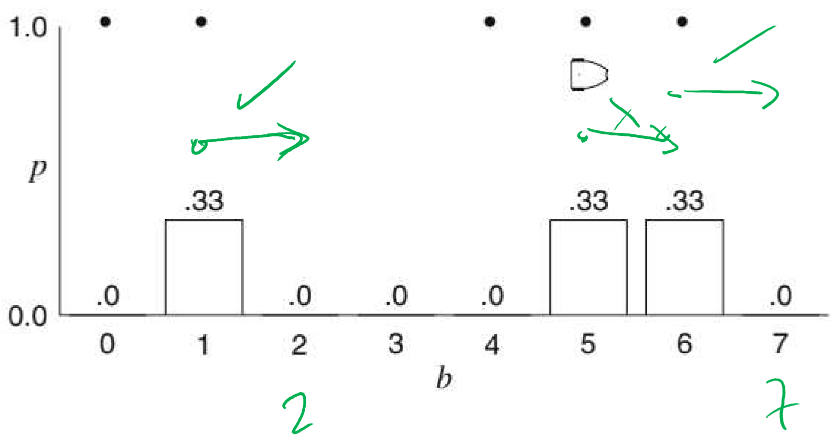


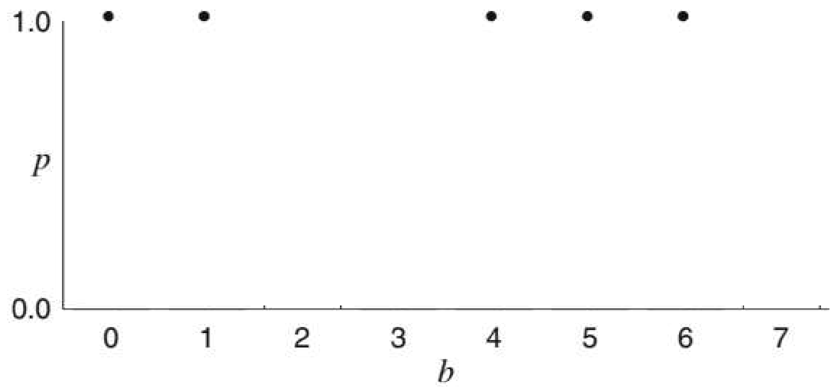
Sensing Increases Certainty

4. Robot move forward → If dark gray area (door) not detected.

light gray 1.0 (100%)

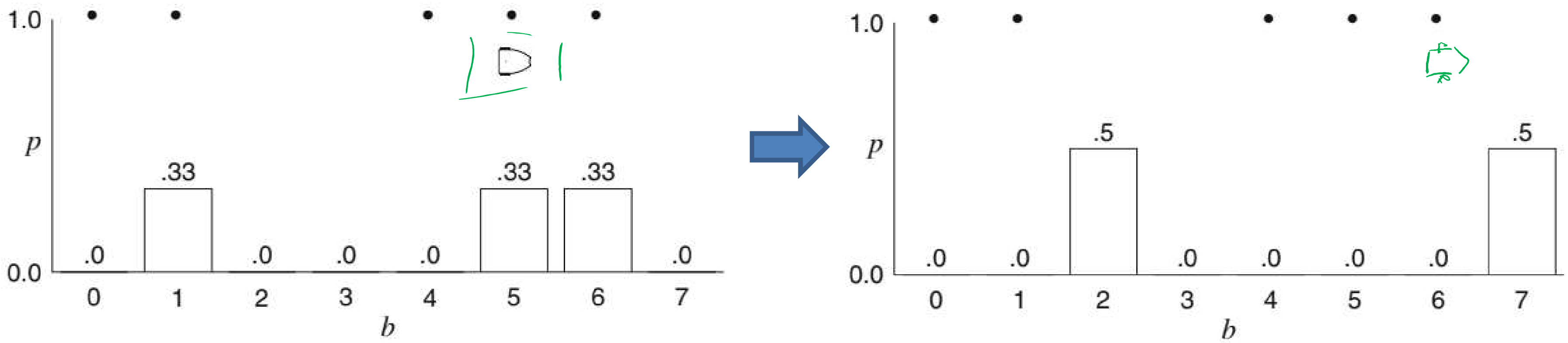
- *? ? ? ? ?*
  ? ? ? ? ?
- *? ? ? ? ?*
  ? ? ? ? ?
- *Hence, probability* ***? ? ? ? ?***





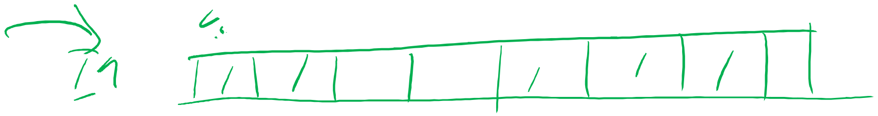
Sensing Increases Certainty

4. **Robot move forward → If dark gray area (door) not detected.**
   - *Now only one possibility (Prev → Curr) :*
     Pos 1 → Pos 2, Pos 6 → 7
   - *Not the following since its new current is dark gray area, (Pos 6) :*
     Pos 6 → Pos 7
   - *Hence, probability is now* **0.5** *for Pos 2 & Pos 7.*



Sensing Increases Certainty

- The robot
  - *Maintains a belief array*



  - *Integrates new data when it detects presence of a door (dark gray area)*
  - *As time goes on → uncertainty decreases*
  - *Knows with greater certainty where it is actually located*
- From given example
  - *Robot eventually knows its position* [0] → [7]
    - In front of Door 6 (Pos 6) [6]
  - *Or, it has reduced its uncertainty to one of the two positions*
    - Pos 2 or Pos 7 [2], [7]

Uncertainty in Sensing

- Sensor values
  - *Light intensity reflected by gray colors of the doors & walls*
- Difference between door(dark gray) & wall(light gray) not great?
  - *May sometimes detect "dark gray door as light gray wall"*
  - *May sometimes detect "light gray wall as dark gray door"*
- This wrong identification due to
  - *Changes in ambient lighting*
  - *Error in sensors themselves*
- Hence, the robot
  - *Cannot distinguish between two with complete certainty*

Uncertainty in Sensing

- Model uncertainties by assigning probabilities to the detection
- Probabilities of detection
  - 100/9 → door
  - *Senses dark gray* → (*correctly as door, 0.9*) & (*mistakenly as wall, 0.1*)
  - 100/1.6 → wall
  - *Senses light gray* → (*correctly as wall, 0.9*) & (*mistakenly as door, 0.1*)

Position door?	0 •	1 •	2	3	4 •	5 •	6 •	7
Initial	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
Sensor	0.11	0.11	0.01	0.01	0.11	0.11	0.11	0.01
Norm	0.19	0.19	0.02	0.02	0.19	0.19	0.19	0.02
Right	0.02	0.19	0.19	0.02	0.02	0.19	0.19	0.19
Sensor	0.02	0.17	0.02	0.00	0.02	0.17	0.17	0.02
Norm	0.03	0.29	0.03	0.00	0.03	0.29	0.29	0.03
Right	0.03	0.03	0.29	0.03	0.00	0.03	0.29	0.29
Sensor	0.03	0.03	0.03	0.00	0.00	0.03	0.26	0.03
Norm	0.07	0.07	0.07	0.01	0.01	0.07	0.63	0.07

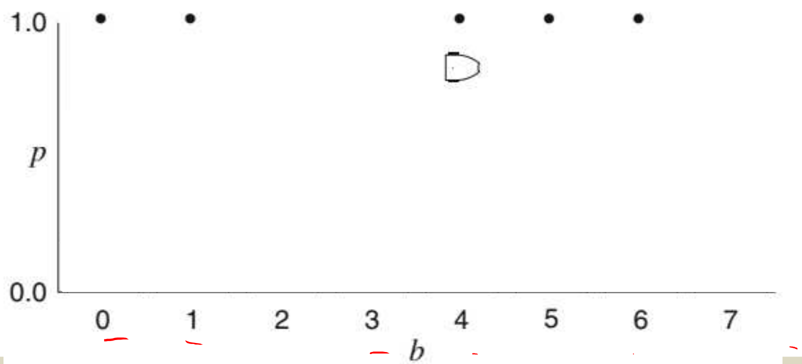
Localization w/ uncertainty in sensing

- **Sensor** – *multiplied by sensor uncertainty*
- **Norm** – *normalized*
- **Right** – *moved right one position*

Uncertainty in Sensing

Initially, $b[i] = 1.0/\ 8 = 0.125 \approx 0.13$

$$0.13 + 0.13 + 0.13 + 0.13 + 0.13 + 0.13 + 0.13 + 0.13 = 1.0$$



Uncertainty in Sensing

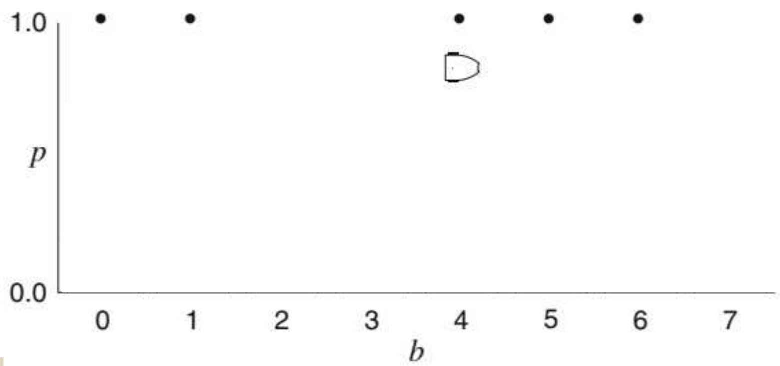
Initially, $b[i] = 1.0/\ 8 = 0.125 \approx 0.13$

$$0.13 + 0.13 + 0.13 + 0.13 + 0.13 + 0.13 + 0.13 + 0.13 = 1.0$$

Sensor: dark gray (door) at a door : $0.125 * 0.9 = 0.1125 \approx 0.11$

: light gray (wall) at a door : $0.125 * 0.1 = 0.0125 \approx 0.01$

$$0.11 + 0.11 + 0.01 + 0.01 + 0.11 + 0.11 + 0.11 + 0.01 = 0.60$$



Uncertainty in Sensing

Initially, $b[i] = 1.0/\ 8 = 0.125 \approx 0.13$

$$0.13+0.13+0.13+0.13+0.13+0.13+0.13+0.13=1.0$$

Sensor: dark gray (door) at a door : $0.125*0.9 = \underline{0.1125 \approx 0.11}$

: light gray (wall) at a door : $0.125*0.1 = 0.0125 \approx 0.01$

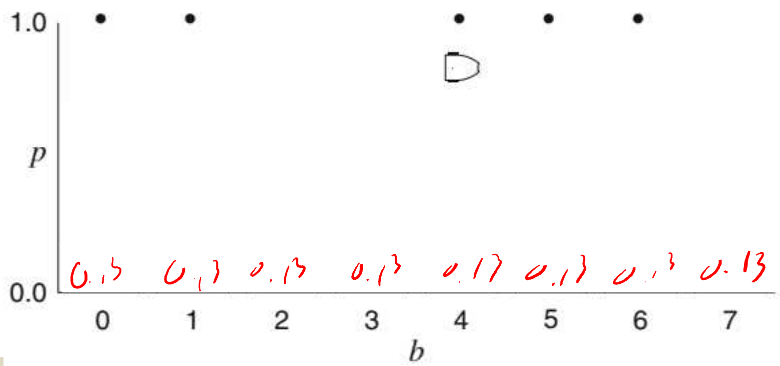
P_0 P_1 P_2 P_3 P_4 P_5 P_6 P_7

$$0.11+0.11+0.01+0.01+0.11+0.11+0.11+0.01=\underline{0.60}$$

Norm: To normalize probabilities:

$0.1125/0.6 \approx 0.19$ and $0.0125/0.6 \approx 0.02$

$$0.19+0.19+0.02+0.02+0.19+0.19+0.19+0.02 \approx \boxed{1.0}$$



Uncertainty in Sensing

Initially, $b[i] = 1.0/\ 8 = 0.125 \approx 0.13$

$$0.13 + 0.13 + 0.13 + 0.13 + 0.13 + 0.13 + 0.13 + 0.13 = 1.0$$

Sensor: dark gray (door) at a door : $0.125 * 0.9 = 0.1125 \approx 0.11$

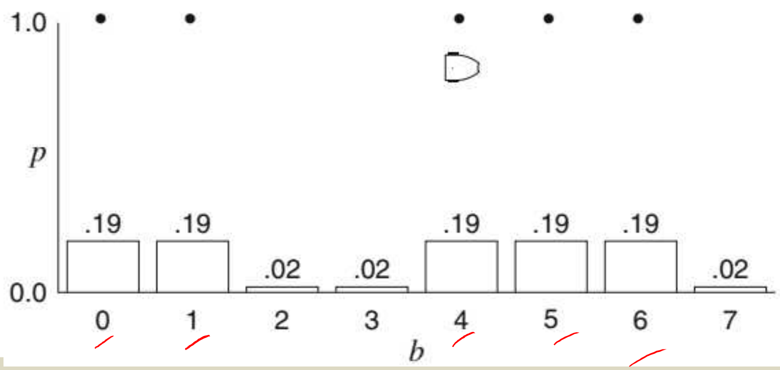
: light gray (wall) at a door : $0.125 * 0.1 = 0.0125 \approx 0.01$

$$0.11 + 0.11 + 0.01 + 0.01 + 0.11 + 0.11 + 0.11 + 0.01 = 0.60$$

Norm: To normalize probabilities:

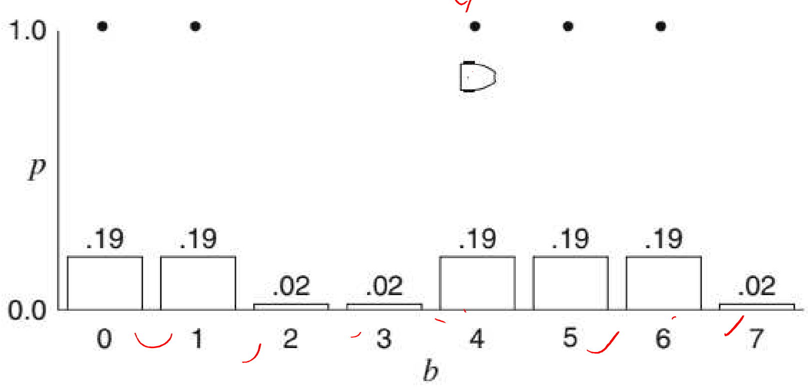
$0.1125 / 0.6 \approx 0.19$ and $0.0125 / 0.6 \approx 0.02$

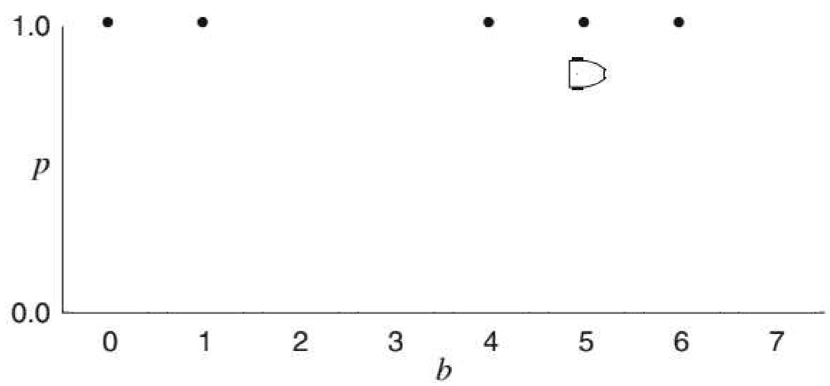
$$0.19 + 0.19 + 0.02 + 0.02 + 0.19 + 0.19 + 0.19 + 0.02 \approx 1.0$$



Uncertainty in Sensing

Right : Robot moves one position to the right →
belief array also move one position to the right

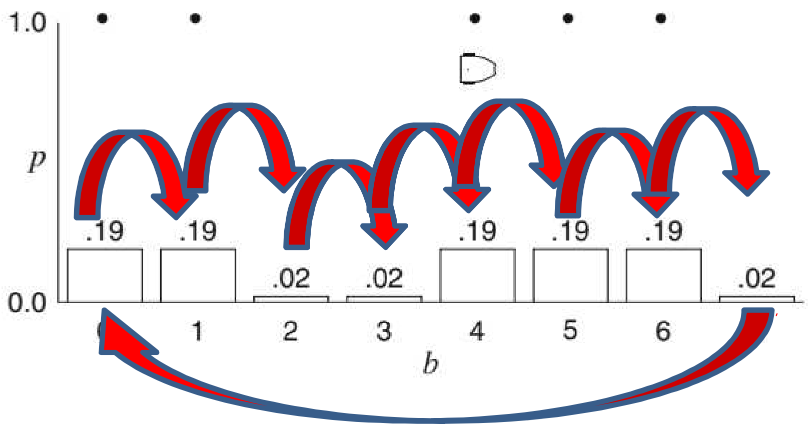


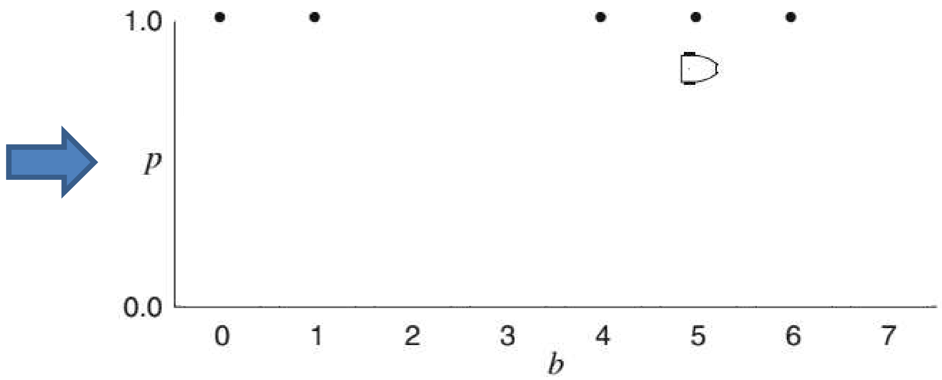


$$0.19 + 0.19 + 0.02 + 0.02 + 0.19 + 0.19 + 0.19 + 0.02 \approx 1.0$$

Uncertainty in Sensing

Right : Robot moves one position to the right → belief array also move one position to the right

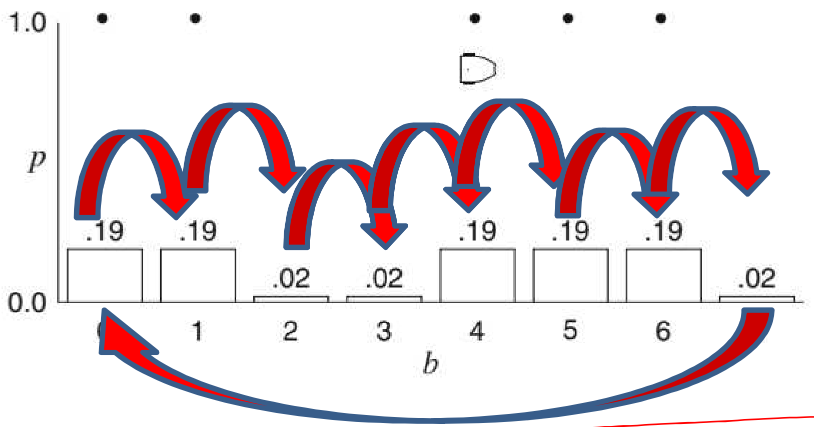


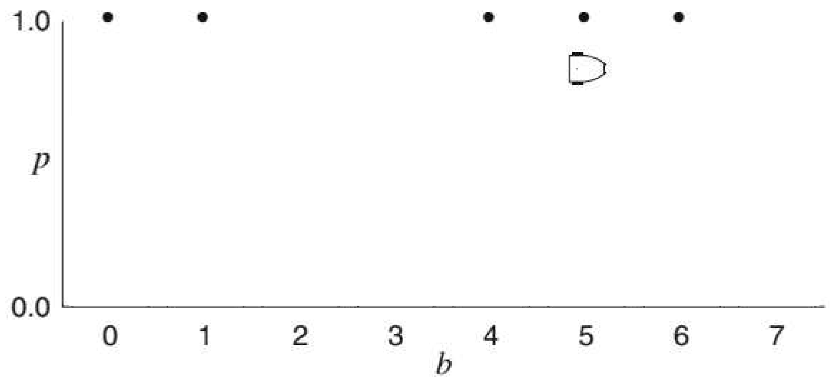


$$0.19 + 0.19 + 0.02 + 0.02 + 0.19 + 0.19 + 0.19 + 0.02 \approx 1.0$$

Uncertainty in Sensing

Right : Robot moves one position to the right → belief array also move one position to the right



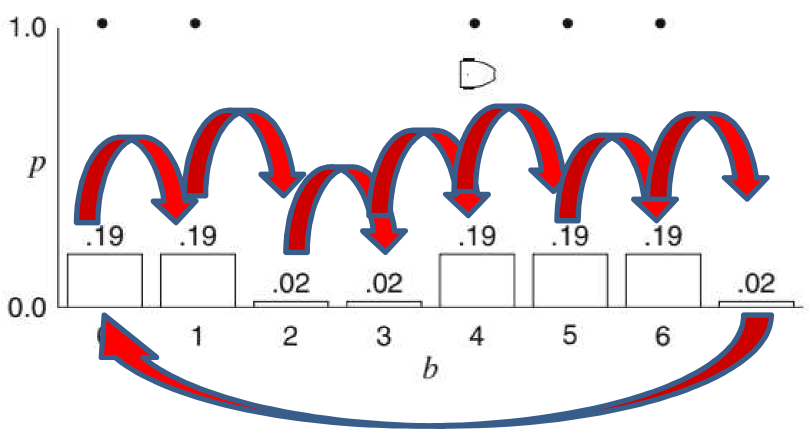


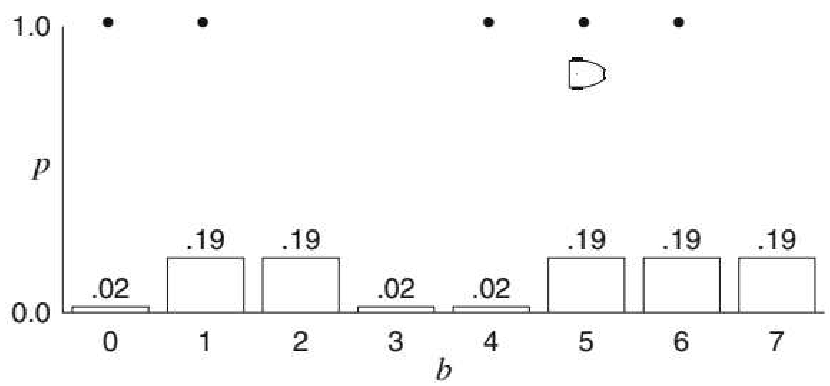
$$0.19+0.19+0.02+0.02+0.19+0.19+0.19+0.02 \approx 1.0$$

$$0.02+.19+0.19+0.02+0.02+0.19+0.19+0.19 \approx 1.0$$

Uncertainty in Sensing

Right : Robot moves one position to the right → belief array also move one position to the right





$$0.19 + 0.19 + 0.02 + 0.02 + 0.19 + 0.19 + 0.19 + 0.02 \approx 1.0$$

$$0.02 + .19 + 0.19 + 0.02 + 0.02 + 0.19 + 0.19 + 0.19 \approx 1.0$$

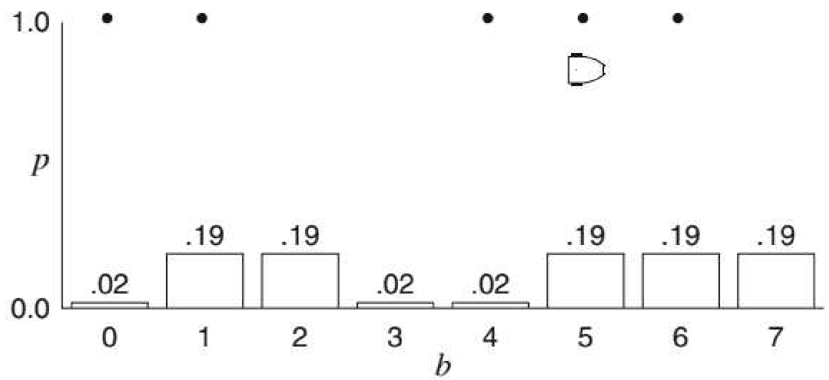
Uncertainty in Sensing

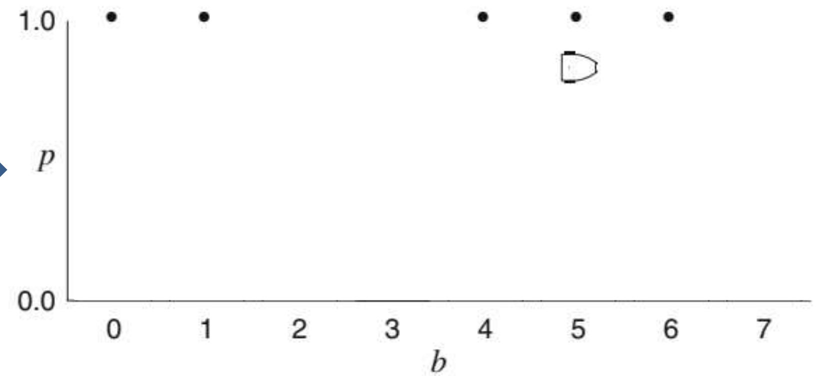
Robot senses gray again → Probability of being at Pos(1, 5, or 6) should increase.

Sensor: dg at door : $0.19 * 0.9 = 0.171 \approx 0.17$, lg at wall :$0.02 * 0.9 = 0.018 \approx 0.02$

: lg at door : $0.19 * 0.1 = 0.019 \approx 0.02$, dg at wall:$0.02 * 0.1 = 0.002 \approx 0.00$

$$0.02 + 0.17 + 0.02 + 0.00 + 0.02 + 0.17 + 0.17 + 0.02 = 0.58$$





Uncertainty in Sensing

Robot senses gray again → Probability of being at Pos(1, 5, or 6) should increase.

Sensor: dg at door : $0.19 * 0.9 = \overset{0.167}{\cancel{0.171}} \approx \overset{.17}{0.17}$, lg at wall :$0.02 * 0.9 = 0.018 \approx 0.02$

: lg at door : $0.19 * 0.1 = 0.019 \approx 0.02$, dg at wall:$0.02 * 0.1 = 0.002 \approx 0.00$

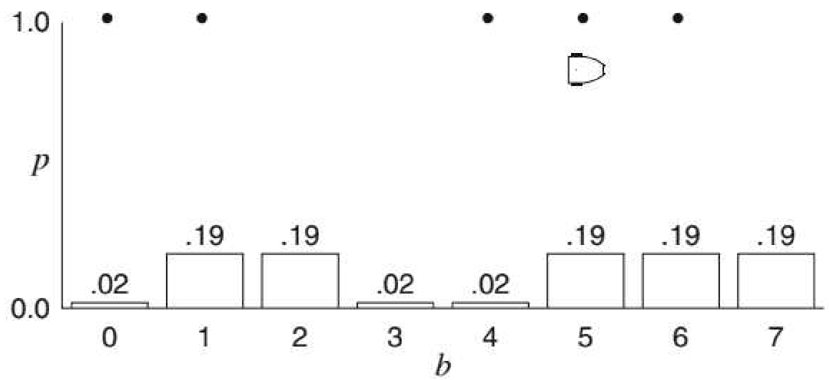
$$0.02 + 0.17 + 0.02 + 0.00 + 0.02 + 0.17 + 0.17 + 0.02 = 0.58$$

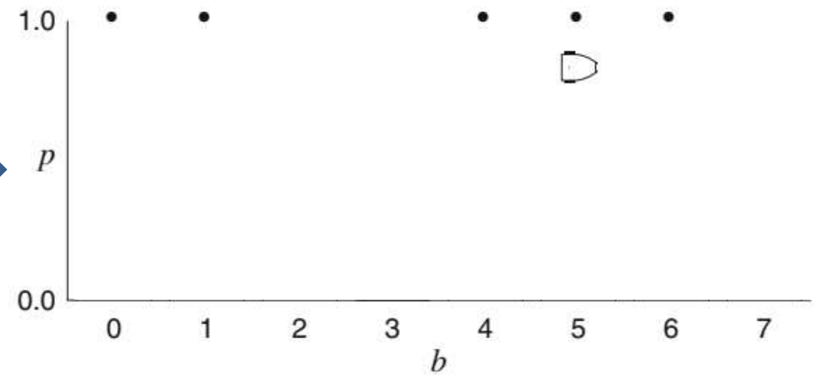
Normal: To normalize probabilities:

.62

$$0.1688 / 0.583 \approx 0.29, \quad 0.0188 / 0.583 \approx 0.03 \quad \text{and} \quad 0.002 / 0.583 \approx 0.00$$

$$0.03 + 0.29 + 0.03 + 0.00 + 0.03 + 0.29 + 0.29 + 0.03 \approx 1.0$$





Uncertainty in Sensing

Robot senses gray again → Probability of being at Pos(1, 5, or 6) should increase.

Sensor: dg at door : $0.19 * 0.9 = 0.171 \approx 0.17$, lg at wall :$0.02 * 0.9 = 0.018 \approx 0.02$

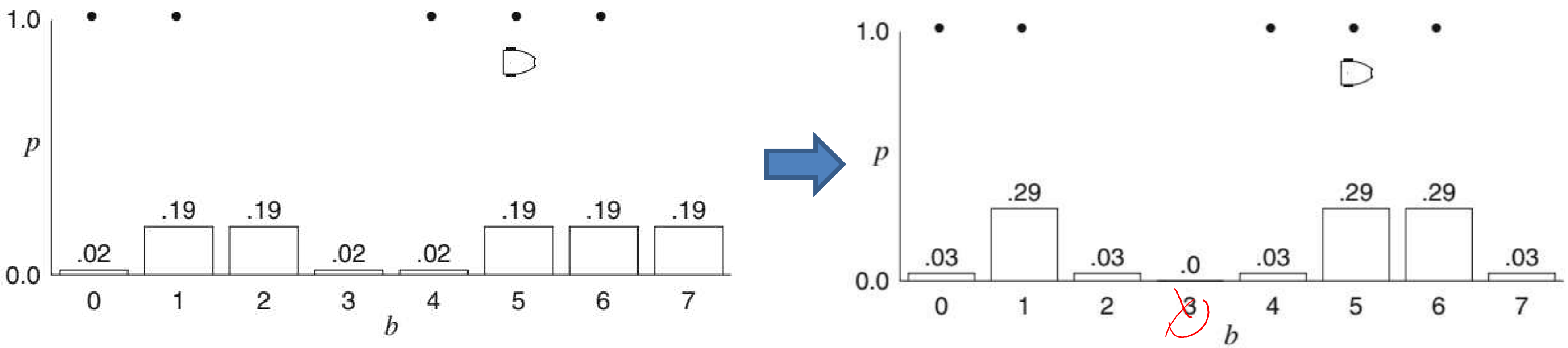
: lg at door : $0.19 * 0.1 = 0.019 \approx 0.02$, dg at wall:$0.02 * 0.1 = 0.002 \approx 0.00$

$$0.02 + 0.17 + 0.02 + 0.00 + 0.02 + 0.17 + 0.17 + 0.02 = 0.58$$

Normal: To normalize probabilities:

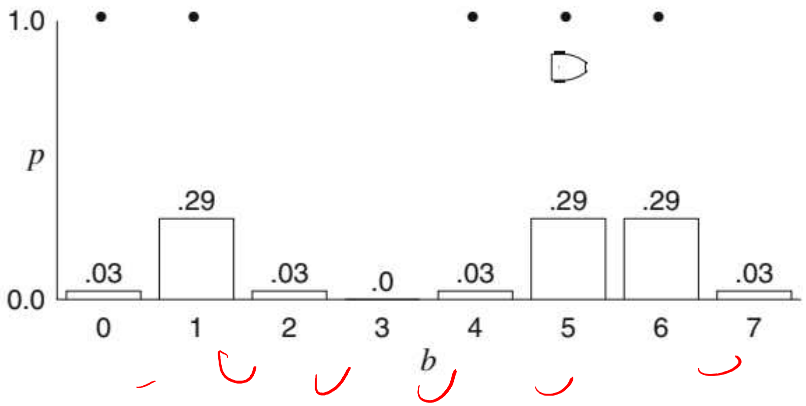
$$0.1688 / 0.583 \approx 0.29, \ \ 0.0188 / 0.583 \approx 0.03 \ \text{ and } \ 0.002 / 0.583 \approx 0.00$$

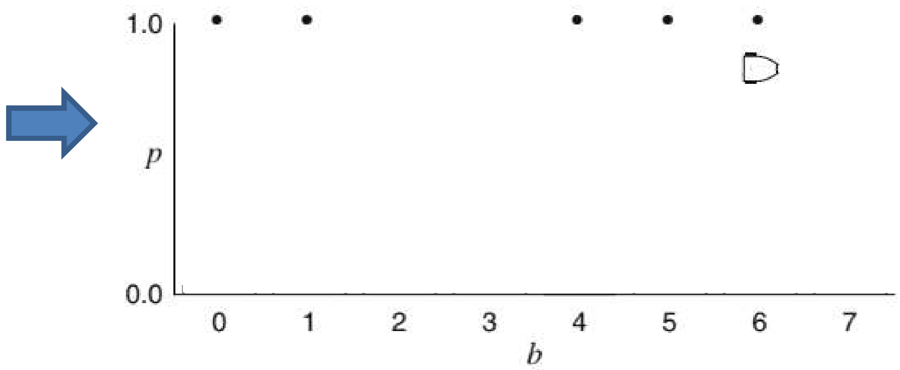
$$0.03 + 0.29 + 0.03 + 0.00 + 0.03 + 0.29 + 0.29 + 0.03 \approx 1.0$$



Uncertainty in Sensing

Right again : Robot moves one position to the right again → belief array also move one position to the right again

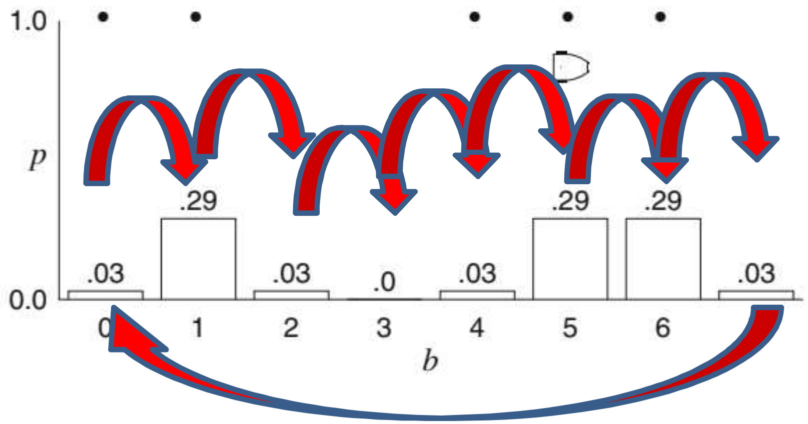


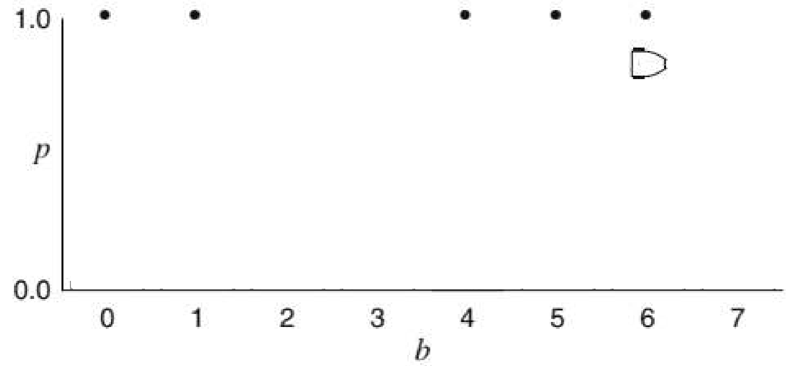


$$0.03 + 0.29 + 0.03 + 0.00 + 0.03 + 0.29 + 0.29 + 0.03 \approx 1.0$$

Uncertainty in Sensing

Right again : Robot moves one position to the right again → belief array also move one position to the right again

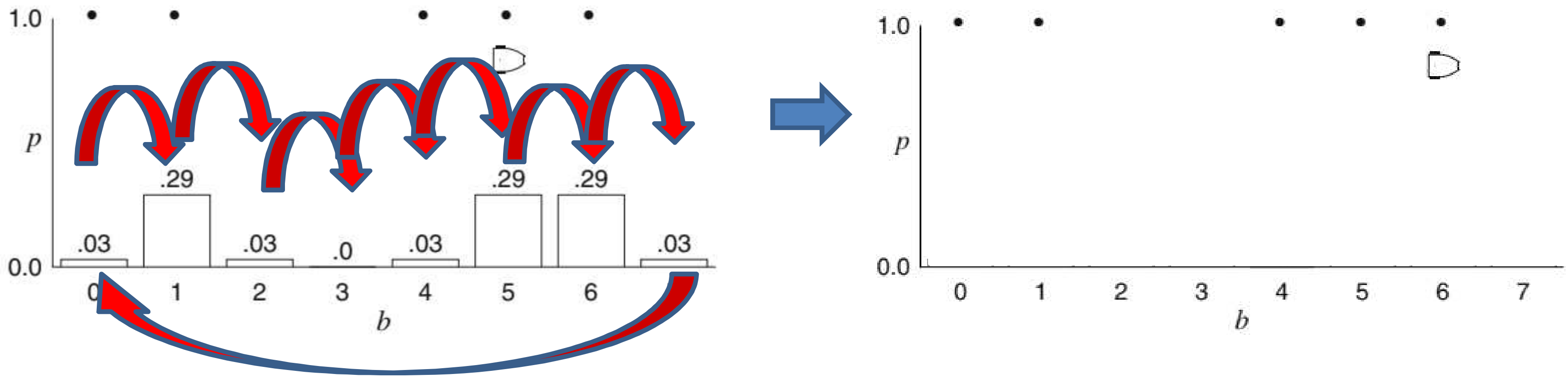




$$0.03 + 0.29 + 0.03 + 0.00 + 0.03 + 0.29 + 0.29 + 0.03 \approx 1.0$$

Uncertainty in Sensing

Right again **:** Robot moves one position to the right again → belief array also move one position to the right again

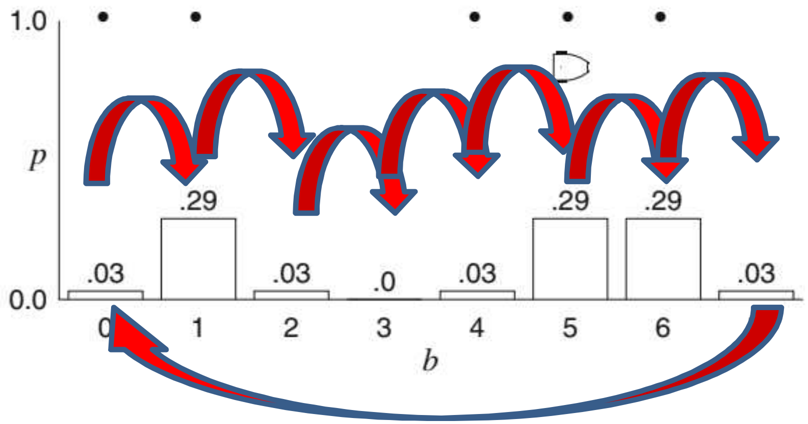


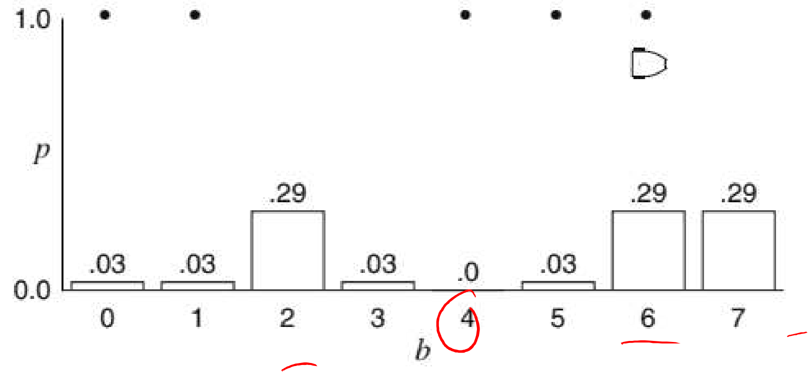
$$0.03+0.29+0.03+0.00+0.03+0.29+0.29+0.03 \approx 1.0$$

$$0.03+0.03+0.29+0.03+0.00+0.03+0.29+0.29 \approx 1.0$$

Uncertainty in Sensing

Right again : Robot moves one position to the right again → belief array also move one position to the right again





$$0.03+0.29+0.03+0.00+0.03+0.29+0.29+0.03 \approx 1.0$$

$$0.03+0.03+0.29+0.03+0.00+0.03+0.29+0.29 \approx 1.0$$

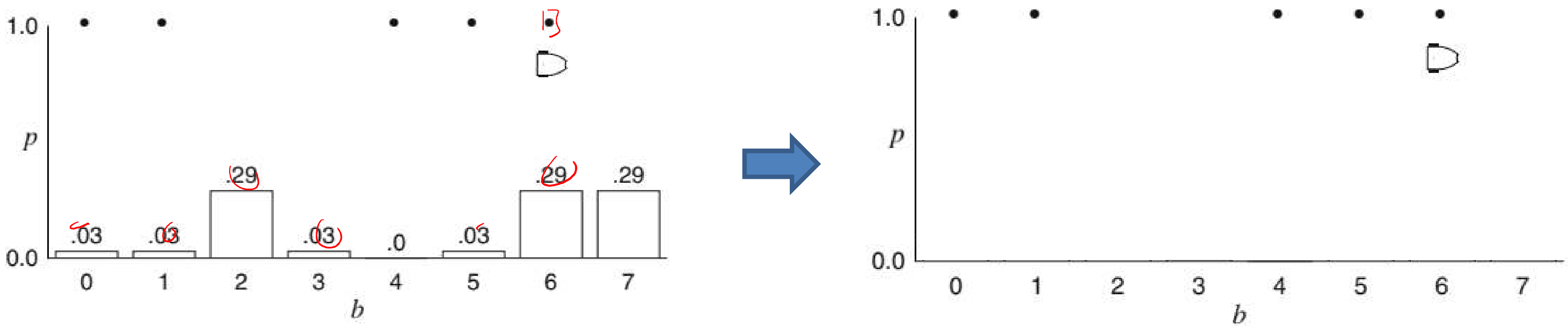
Uncertainty in Sensing

Robot senses gray again → Robot is almost certainly at Pos(6)

Sensor: dg at door : $0.29 * 0.9 = 0.261 \approx 0.26$, lg at wall : $0.03 * 0.9 = 0.029 \approx 0.03$

: lg at door : $0.29 * 0.1 = 0.029 \approx 0.03$, dg at wall: $0.03 * 0.1 = 0.003 \approx 0.00$

$$0.03 + 0.03 + 0.03 + 0.00 + 0.00 + 0.03 + 0.26 + 0.03 = 0.41$$



Uncertainty in Sensing

Robot senses gray again → Robot is almost certainly at Pos(6)

Sensor: dg at door : $0.29 * 0.9 = 0.261 \approx 0.26$, lg at wall : $0.03 * 0.9 = 0.029 \approx 0.03$

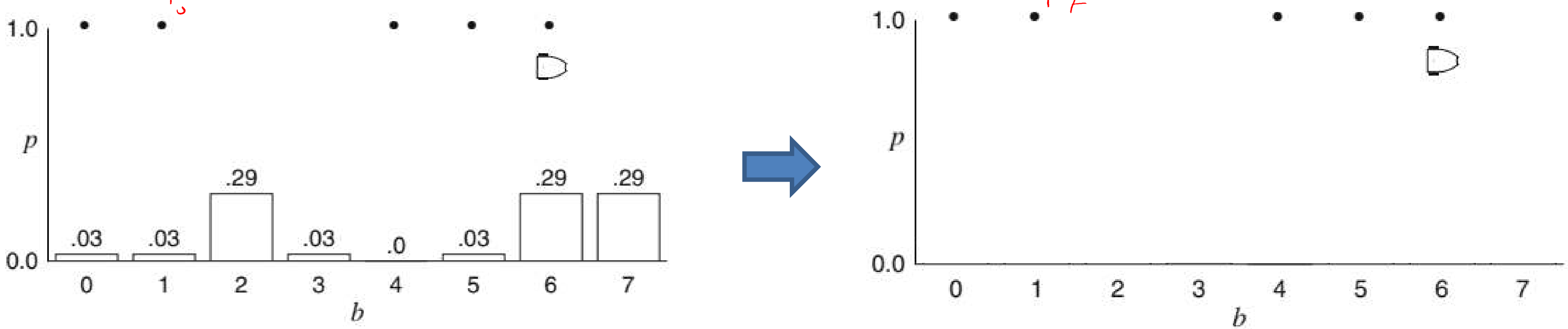
: lg at door : $0.29 * 0.1 = 0.029 \approx 0.03$, dg at wall: $0.03 * 0.1 = 0.003 \approx 0.00$

$$\{0.03 + 0.03 + 0.03 + 0.00 + 0.00 + 0.03 + 0.26 + 0.03\} = \underline{0.41}$$

Normal: To normalize probabilities:

$0.261 / 0.41 \approx 0.63$, $0.029 / 0.41 \approx 0.07$ and $0.003 / 0.41 \approx 0.01$

$$0.07 + 0.07 + 0.07 + 0.00 + 0.00 + 0.07 + 0.63 + 0.07 \approx 1.0$$



Uncertainty in Sensing

belief

Robot senses gray again → Robot is almost certainly at Pos(6)

Sensor: dg at door : $0.29 * 0.9 = 0.261 \approx 0.26$, lg at wall : $0.03 * 0.9 = 0.029 \approx 0.03$

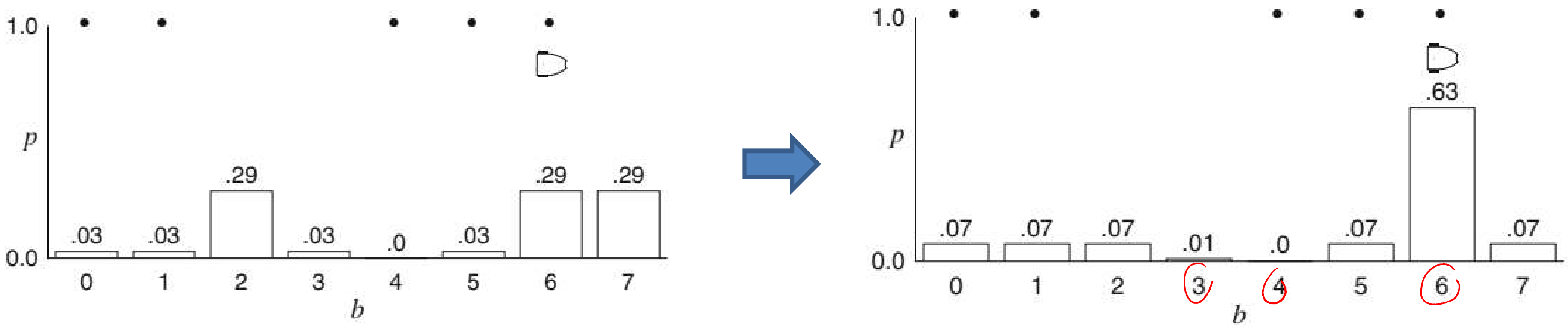
: lg at door : $0.29 * 0.1 = 0.029 \approx 0.03$, dg at wall: $0.03 * 0.1 = 0.003 \approx 0.00$

$$0.03 + 0.03 + 0.03 + 0.00 + 0.00 + 0.03 + 0.26 + 0.03 = 0.41$$

Normal: To normalize probabilities:

$$0.261 / 0.41 \approx 0.63,\ \ 0.029 / 0.41 \approx 0.07 \text{ and } \ 0.003 / 0.41 \approx 0.01$$

$$0.07 + 0.07 + 0.07 + 0.00 + 0.00 + 0.07 + 0.63 + 0.07 \approx 1.0$$



Localization

- Landmarks
- Determining Position from Objects whose Position is Known
- Global Positioning System
- Probabilistic Localization
- Uncertainty in Motion

Uncertainty in Motion

- Robots also subject to uncertainty in motion
  - *Tell robot move one position to the right*
    - Might move two pos to right, or
    - Move very little & stay current pos
- Modified algorithm
  - b' : belief array
  - b'_i : new value of b',
  - p_i : probability of detecting door

$$b'_i = p_i b_i$$

$$p_i = 0.9 \text{ for } i = 0,1,4,5,6$$

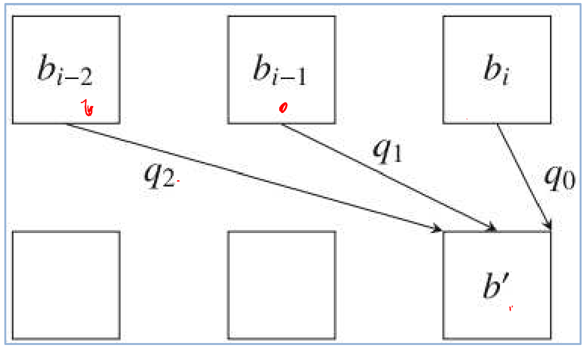
$$p_i = 0.1 \text{ for } i = 2,3,7$$

Uncertainty in Motion

- If motion is certain : move one position to right
- If motion is uncertain :
  - *Following computation take into account probabilities* $\boldsymbol{q_j}$ *that robot actually moves* $\boldsymbol{j}$ *= 0, 1, 2 positions:*

$$b'_i = p_i\left(b_{i-2}q_2 + b_{i-1}q_1 + b_i q_0\right)$$

as shown in the given diagram:



- Highly likely : robot moves correctly
  - *Reasonable values :* $\boldsymbol{q_1} = 0.8$ *and* $\boldsymbol{q_0} = \boldsymbol{q_2} = 0.1$

Uncertainty in Motion

- With values for uncertainty of motion & previous values of p_i
  - *Belief array after three moves is given in the table*

Position door?	0 •	1 •	2	3	4 •	5 •	6 •	7
Initial	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
Sensor	0.11	0.11	0.01	0.01	0.11	0.11	0.11	0.01
Norm	0.19	0.19	0.02	0.02	0.19	0.19	0.19	0.02
Right	0.05	0.19	0.17	0.04	0.04	0.17	0.19	0.17
Sensor	0.05	0.17	0.02	0.00	0.03	0.15	0.17	0.02
Norm	0.08	0.27	0.03	0.01	0.06	0.25	0.28	0.03
Right	0.06	0.12	0.23	0.05	0.01	0.07	0.23	0.25
Sensor	0.05	0.10	0.02	0.01	0.01	0.06	0.21	0.02
Norm	0.11	0.21	0.05	0.01	0.02	0.13	0.43	0.05

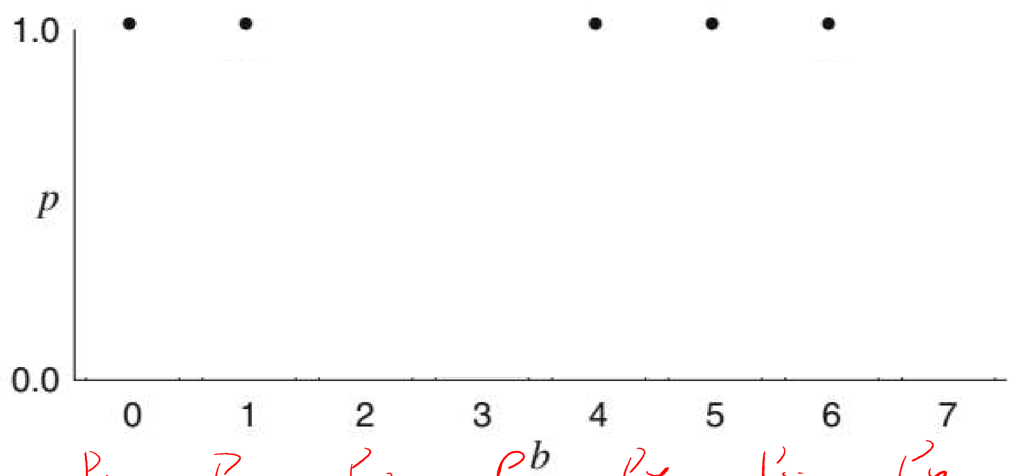
- **Sensor** – *multiplied by sensor uncertainty*
- **Norm** – *normalized*
- **Right** – *moved right one position*

$$p_i = 0.9 \text{ for } i = 0,1,4,5,6 \qquad b_i' = p_i\left(b_{i-2}q_2 + b_{i-1}q_1 + b_i q_0\right)$$

$$p_i = 0.1 \text{ for } i = 2,3,7 \qquad q_1 = 0.8,\ q_0 = q_2 = 0.1$$

Uncertainty in Motion

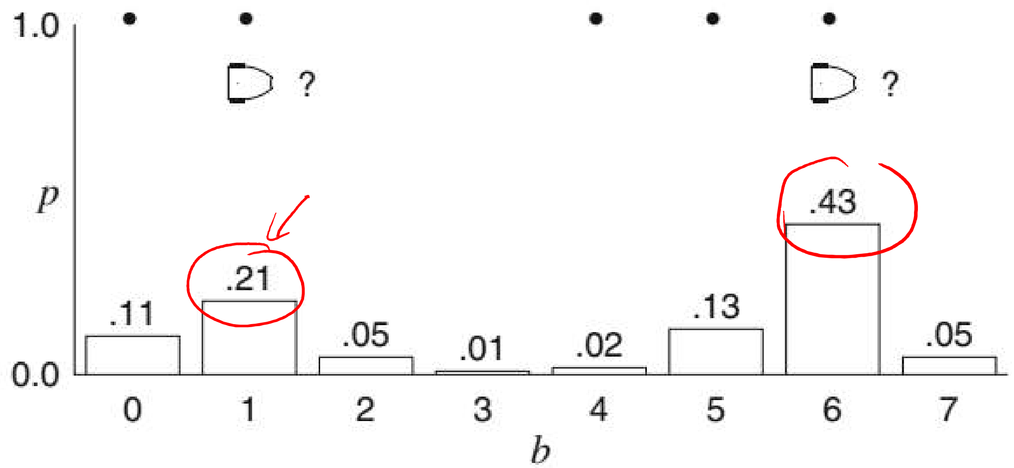
- Robot likely at **? ? ? ? ?**
  - *? ? ? ? ?*
  - *? ? ? ? ?*



	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
Right	0.06	0.12	0.23	0.05	0.01	0.07	0.23	0.25
Sensor	0.05	0.10	0.02	0.01	0.01	0.06	0.21	0.02
Norm	0.11	0.21	0.05	0.01	0.02	0.13	0.43	0.05

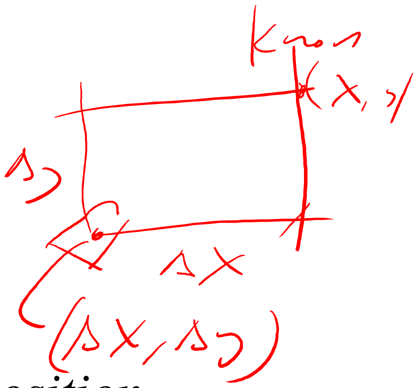
Uncertainty in Motion

- Robot likely at Pos(6)
  - *But less certain since probability only 0.43, instead of 0.63*
  - *There is non-negligible probability of 0.21 that robot is at Pos(1)*



Right	0.06	0.12	0.23	0.05	0.01	0.07	0.23	0.25
Sensor	0.05	0.10	0.02	0.01	0.01	0.06	0.21	0.02
Norm	0.11	0.21	0.05	0.01	0.02	0.13	0.43	0.05

Summary



- Odometry
  - *Provide estimation of robot position*
- Use surveying techniques
  - *Computing position relative to an object with known position*
- GPS
  - *Give excellent location data but not accurate enough & limited indoors*
- Probabilistic localization
  - *Estimate position with high probability*
  - *If multiple known objects can be sensed + map of environment*
- Probability reduced
  - *If lots of uncertainty in sensors or motion of robot*

Thank you.