# Probabilistic Robotics, PRR06, Fall 2017

Exam, Wednesday October 26th, 09:00 - 12:00, room A1.02

Arnoud Visser Emiel Hoogeboom

November 15, 2017

## Question 1

A company likes to use automatic driving vehicles to guarantee that the rental cars are at the place where it will be the most probable that they will be hired.

Their business case starts with a rental point at Schiphol, combined with at the surrounding suburbs Hoofddorp, Badhoevedorp, Amstelveen and Aalsmeer. Based on the number of inhabitants, the company has made about the probability $p_k$ that a vehicle is used for a transition to Schiphol (resp. $p_k = 0.76, 0.12, 0.89, 0.31$), as illustrated in Fig. 1. This are the probabilities that an inhabitant of this town reserves the vehicle for that a certain hour $t$ in the morning under the condition that a vehicle is available. For the afternoon the company expects other transition probabilities.

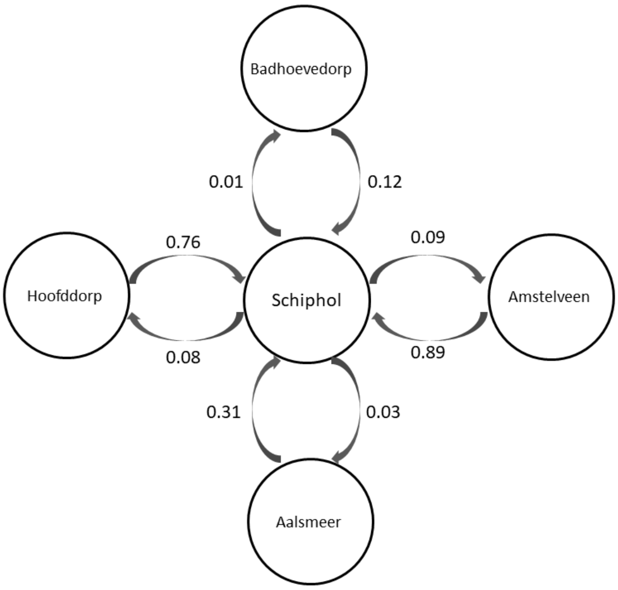


Figure 1: The probabilities of a transition to and from Schiphol from a number of surrounding towns (in the morning).

**(a)** What is the chance that there is still a vehicle available after 6 hours in Badhoevedorp, assuming that there was one available in this town initially? For the moment you may ignore the $1\%$ chance that a vehicle is driven back to Badhoevedorp.

Answer $46.4\% = (0.88)^6$

Now assume that you decide to maintain your belief of the location of the vehicle in a 2D-grid. This is a simple $3 \times 3$ grid, with Schiphol in the center and the four towns at Manhattan locations west, north, east and south of Schiphol. And while in question **(a)** you assumed the vehicle was initially in Badhoevedorp (prior $p_a$), you can now indicate that the vehicle is initially in one of the four towns (prior $p_b$), with an uniform distribution of the chance over the four towns.

$$p_{\mathbf{a}} = \begin{bmatrix} 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix} \quad p_{\mathbf{b}} = \begin{bmatrix} 0.0 & 0.25 & 0.0 \\ 0.25 & 0.0 & 0.25 \\ 0.0 & 0.25 & 0.0 \end{bmatrix}$$

**(b)** Calculate the chance that there is a vehicle at the central location Schiphol after a time-step of one hour.

Answer: $52\% <= 0.25 * (0.89 + 0.76 + 0.31 + 0.12) = 2.08 * 0.25 = 0.52$

For the autonomous vehicle this transition table describes how the state $x$ evolves from t to t-1 without any control $u_t$, because the customers are here modelled as a non-deterministic force of nature. Yet, the autonomous vehicle can be given controls (at an high level) $u_t$, for instance by giving it the command to drive autonomously back from Schiphol to Amstelveen (which should increase the chance on another customer with a successful reservation). If such a command is given in the second hour (so the prior distribution is the situation after question **b**), the probability $p_{(1,2)}$ of a vehicle at the eastern grid-position $(1, 2)$ (equivalent with topical location Amstelveen) consist of three terms:

1. A customer that has reserved a vehicle and drives it to Schiphol
2. A customer who returns with a morning flight and drives a vehicle back to Amstelveen
3. A self-driving vehicle which executes the command $u_t$ and drives autonomously back to Amstelveen

It is now your task to come up with a probabilistic formulation of the motion model.

**(c)** Formulate an equation for the posterior probability for a vehicle which takes those three terms into account. Take into account that the probability of a success for a command is not 100%, there is a probability $p_f$ that the vehicle is not able to reach the destination Amstelveen.

Answer:
If $x_{A_t}$ indicates that the vehicle is at Amstelveen at time $t$, and $x_{S_{t-1}}$ indicates that the vehicle was at Schiphol at time $t-1$, then

$$x_{A_t} = (1 - P(x_{S_t}|x_{A_{t-1}}))P(x_{A_{t-1}}) + P(x_{A_t}|x_{S_{t-1}})P(x_{S_{t-1}}) + P(x_{A_t}|x_{S_{t-1}}, u_t)(1-p_f)P(x_{S_{t-1}})$$

Now, assume that the companies vehicle has a crude compass on board, which gives an indication of the quadrant the vehicle can be found. This measurements can be matched with the following map:

$$m = \begin{bmatrix} 'NW' & 'N' & 'NE' \\ 'W' & 'C' & 'E' \\ 'SW' & 'S' & 'SE' \end{bmatrix} \quad p(z = 'E'|m, x) = \begin{bmatrix} 0.0 & 0.0 & 0.333 \\ 0.0 & 0.0 & 0.333 \\ 0.0 & 0.0 & 0.333 \end{bmatrix}$$

**(d)** Do you now have enough information to implement line 4 of Table 8.1? If the **measurement_model** returns the value $q$, indicated how this $q$ is calculated.

Answer:
The grid localization algorithm of Table 8.1 already loops over all grid cells and takes the normalization $\eta$ in account. The **measurement_model** only has to multiply the $p(z = 'E'|m, x)$ with the probability $p(x)$, as already indicated line 4 of the book correction on the 2nd edition[1]. So $p(z = 'E'|m, x)$ contains all information to return $q$:

$$q = p(z = 'E'|x = mean(\mathbf{x}_k), m)$$

**(e)** On line 4 of Table 8.1 you see the symbol $\eta$. What does this symbol mean? What does that mean for your implementation?

Answer:
$\eta$ indicates the normalizer constant, already introduced in Table 2.1, which ensures that the belief is a probability, i.e. guarantees that the sum of all beliefs for each grid cell is 1.0 ($\sum_k p_{k,t} = 1$).

[1] href=http://probabilistic-robotics.informatik.uni-freiburg.de/corrections1/pg238.pdf

# Question 2

Suppose that a robot is equipped with a sensor measuring range and bearing to a landmark, including the landmarks identity (the identity sensor is noise free). We want to perform global localization with EKFs. When seeing a single landmark, the posterior is usually poorly approximated by a Gaussian.

**(a)** Explain the shape what shape of posterior you could expect from a single range and bearing measurement. Sketch your shape.

Answer:
There is both uncertainty in the range and the bearing. The uncertainty in the range is a band, the uncertainty in the bearing is an arc, together you expect a banana shaped distribution:

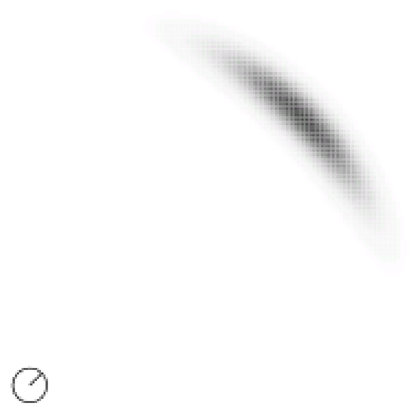

Figure 2: The true uncertainty distribution: a banana shape.

Now imagine the situation that you have two measurements from two locations $l_1$ and $l_2$ at two timesteps; both observations of the same landmark with a known identity. Yet, now you have only range measurements $r_1$ and $r_2$, no estimate of the bearing to this landmark. You can model this with two circles with radii $r_1$ and $r_2$ and centers $l_1$ and $l_2$, which intersect at two locations. The location of those intersection are known. If $d = ||l2 - l1||$ is the distance between the locations, then is $a = \frac{r_1^2 - r_2^2 + d^2}{2d}$ the distance between the two intersections. So for each observation there are two possible bearings. The uncertainty on this bearing estimate is approximated with a Gaussian, with 10 times as high uncertainty in the bearing ($\nu_t = 10 \times \nu_r$) than the uncertainty in the range $\nu_r$, by defining the covariance matrix C with the formula:

$$C = \Phi \begin{bmatrix} \nu_r & 0 \\ 0 & 10 \times \nu_r \end{bmatrix} \Phi^T \quad \text{with} \quad \Phi = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

with $\theta$ the possible bearing of the observation (see Fig. 3 left). The two overlapping Gaussians at the intersection are multiplied, which results in another Gaussian (see Fig. 3 right).

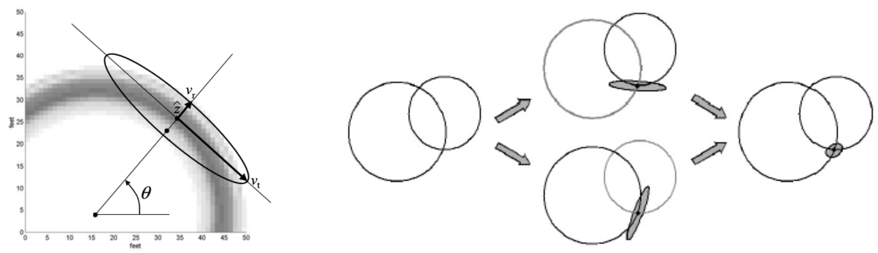

Figure 3: Gaussian approximation of the uncertainty of a range measurement with bearing $\theta$, which can be combined for two measurements (Courtesy [3] and [4]).

**(b)** Is the resulting Gaussian as approximation of the combination of the two observations a good approximation? Make a more detailed sketch (more detailed compared to Fig. 3 right) the two original distributions and the resulting distribution.

Answer:

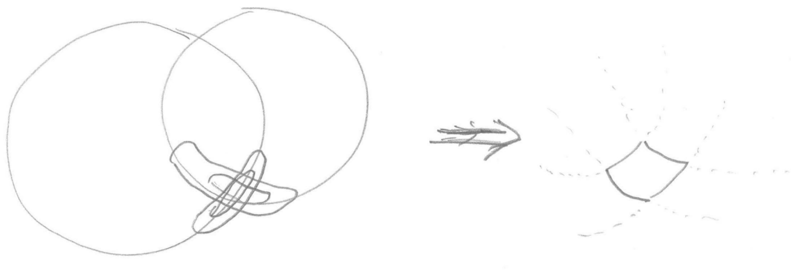

Figure 4: Two orthogonal banana shape distributions combine to diamond shape with slightly curved edges.

When two elongated, slightly curved distributions are combined, the result is concentrated peak at the crossing, which can be quite well approximated with the elliptical (nearly round) shape of a Gaussian distribution.

Combining two observations to two hypothesis is a good approach, although in practice one receives multiple observations when driving along a landmark. One could spawn for each observation a full circle of hypotheses of the bearing, and rely on later observations to separate the more probable observations from the less probable observations (See Fig. 5).

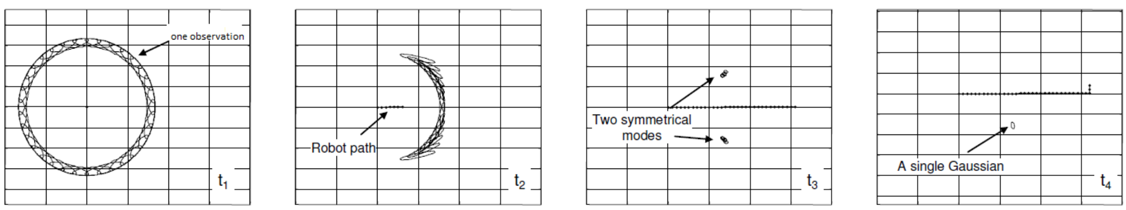


Figure 5: Combination of multiple observations at multiple timesteps $t_1 \cdots t_4$ (Courtesy [1]).

To separate the more probable observations from the less probable observations one need to weight the probability of each hypothesis. This can be done iteratively by updating each of weights $w_t^k$ of each hypothesis $k$ with the formula:

$$w_t^k \propto w_{t-1}^k \mathcal{N}(z_t; h_t^k, \sigma_t^{k^2})$$

where the mean $h_t^k = h(x_t^{[i]}, \hat{m}_t^k)$ is computed using the sensor model $h(\cdot)$, based on hypothesis $i$ of the robot location $x_t^{[i]}$ and the hypothesis on the landmark position $\hat{m}_t^k$.

The variance $\sigma_t^{k^2}$ includes the sensor noise $\sigma_s$ and the projection of the beacon uncertainty:

$$\sigma_t^{k^2} = H\Sigma_t^k H^T + \sigma_s^2$$

**(c)** What is actually the mathematical definition of $h(\cdot)$ (equivalent with Eq. 7.12 from the book) for the range-only observation $z_t^l = (r_t^l, s_t^l)^T$ of landmark with signature $s_t^l$ equal to the identy of the landmark $c_t^l$?

Answer:

$$h(\cdot) = z_t^l = \begin{bmatrix} r_t^l \\ s_t^l \end{bmatrix} = \begin{bmatrix} \sqrt{(\hat{m}_x^k - x^{[i]})^2 + (\hat{m}_y^k - y^{[i]})^2} \\ \hat{m}_s^k \end{bmatrix} + \mathcal{N}(0, Q_t)$$

With $k$ the identity $l$ of the signature $\hat{m}_s^k = c_t^l$ of the landmark.

**(d)** Also give the corresponding Jacobian $H_t^l$ of $h(\cdot)$.

Answer:

$$H_t^l = \frac{\partial h(x_t^{[i]}, l, \hat{m}_t)}{\partial x_t^{[i]}} = \begin{bmatrix} \frac{\partial r_t^l}{\partial x^{[i]}} & \frac{\partial r_t^l}{\partial y^{[i]}} & \frac{\partial r_t^l}{\partial \theta^{[i]}} \\ \frac{\partial s_t^l}{\partial x^{[i]}} & \frac{\partial s_t^l}{\partial y^{[i]}} & \frac{\partial s_t^l}{\partial \theta^{[i]}} \end{bmatrix} = \begin{bmatrix} -\frac{(\hat{m}_x^k - x^{[i]})}{\sqrt{q}} & -\frac{(\hat{m}_y^k - y^{[i]})}{\sqrt{q}} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

With $q = (\hat{m}_x^k - x^{[i]})^2 + (\hat{m}_y^k - y^{[i]})^2$

**(e)** Explain for $t_4$ in Fig. 5 why only a single Gaussian is left.

Answer:
Although Fig. 5 is quite small, it can be seen that the path for timestep $t_2$ and $t_3$ is still straight to the right. With range-only observations both symmatrical nodes in $t_3$ are equivalent probable (because they are always as far from the straight path). At $t_4$ the robot takes a sharp turn to the left which results that two remaining hypotheses $k$ result in different prediction of the range $h_4^k$ and corresponding probability $\mathcal{N}(z_4; h_4^k, \sigma_t^{k^2})$ of the measured range $z_4$ for both $k$s, which results in different values for $w_4^k$ (one big and one small).

# Question 3

For this question you have to rely on a distance-only sensor. You try to locate your friend using her cellphone signals. Your friend is either at home or at Science Park. Suppose that on the map of Amsterdam, the Science Park is located at $m_0 = (10, 8)^T$, and your friend's home is situated at $m_1 = (6, 3)^T$. You have access to the data received by two cell towers. You have access to the data received by two cell towers, which are located at the positions $x_0 = (12, 4)^T$ and $x_1 = (5, 7)^T$, respectively. The distance between your friend's phone and the towers can be computed from the intensities of your friend's cell phone signals. The distance measurements are distributed by white Gaussian noise with variances $\sigma_0^2 = 1$ for tower 0 and $\sigma_1^2 = 1.5$ for tower 1. You receive the distance measurements $d_0 = 3.9$ and $d_1 = 4.5$ from the two towers.

**(a)** Make a drawing of the situation.

Answer:
As can be seen, $m_0$ is located in both variance bands, while $m_1$ is outside the band from tower $x_0$.

**(b)** At which of the two places is your friend more likely to be? Explain your calculations.

Answer:
When your friend is at $m_0$ (i.e $at\_university$), you would expect the distance $\delta_{0,0} = \sqrt{(m_{0,x} - x_{0,x})^2 + (m_{0,y} - x_{0,y})^2} = \sqrt{(10-12)^2 + (8-4)^2} = 4.47$ to tower $x_0$ and $\delta_{0,1} = \sqrt{(m_{0,x} - x_{1,x})^2 + (m_{0,y} - x_{1,y})^2} = \sqrt{(10-5)^2 + (8-7)^2} = 5.09$ to tower $x_1$. When your friend is at $m_1$, you would expect expect the distance $\delta_{1,0} = \sqrt{(m_{1,x} - x_{0,x})^2 + (m_{1,y} - x_{0,y})^2} = \sqrt{(6-12)^2 + (3-4)^2} = 6.08$ to tower $x_0$ and $\delta_{1,1} = \sqrt{(m_{1,x} - x_{1,x})^2 + (m_{1,y} - x_{1,y})^2} = \sqrt{(6-5)^2 + (3-7)^2} = 4.12$ to tower $x_1$. These distances $\delta_{0,0} = 4.47$ and $\delta_{1,0} = 6.08$ should be compared with $d_0 = 3.9$ and the distances $\delta_{0,1} = 5.09$ and $\delta_{1,1} = 4.12$ should be compared with $d_1 = 4.5$. The measurements are modelled with a Gaussian and the measurements are not correlated (so can be multiplied).

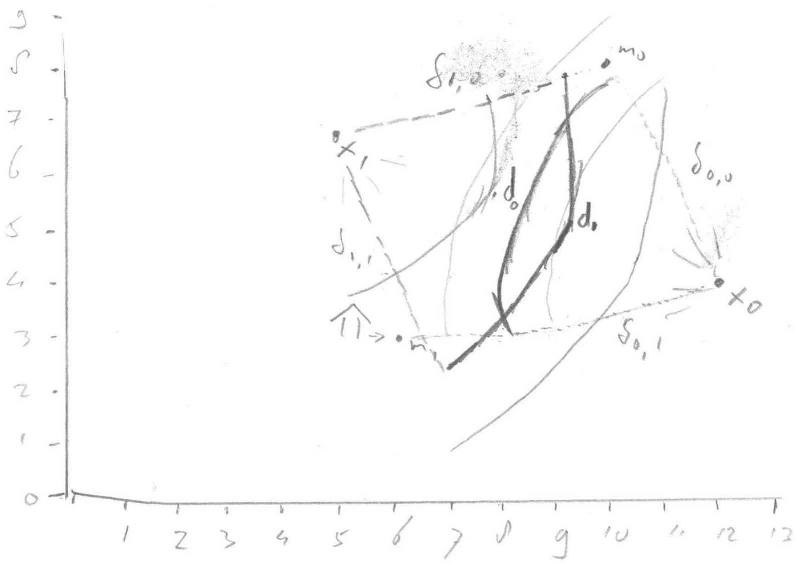


Figure 6: The two possible locations $m_0$ and $m_1$ and the two landmarks $x_0$ and $x_1$, including the band of the distance $d_0$ and $d_1$ measured (with of band equavalent with $\sigma_1$ and $\sigma_2$).

So if $z$ is the combined observation $(d_0, d_1)^T$. Then $p(z|at\_university) \approx \mathcal{N}(\delta_{0,0} - d_0, \sigma_0^2) \cdot \mathcal{N}(\delta_{0,1} - d_1, \sigma_1^2)$ and $p(z|at\_home) \approx \mathcal{N}(\delta_{1,0} - d_0, \sigma_0^2) \cdot \mathcal{N}(\delta_{1,1} - d_1, \sigma_1^2)$. One should compare $p(at\_university) \approx \epsilon^{-\frac{1}{2\sigma_0}(\delta_{0,0}-d_0)^2} \cdot \epsilon^{-\frac{1}{2\sigma_1}(\delta_{0,1}-d_1)^2} \approx \epsilon^{-(0.57)^2/2} \cdot \epsilon^{-(0.59)^2/3}$ versus $p(at\_home) \approx \epsilon^{-\frac{1}{2\sigma_0}(\delta_{1,0}-d_0)^2} \cdot \epsilon^{-\frac{1}{2\sigma_1}(\delta_{1,1}-d_1)^2} \approx \epsilon^{-(2.18)^2/2} \cdot \epsilon^{-(0.38)^2/3}$. The smallest term is $\mathcal{N}(\delta_{1,0} - d_0, \sigma_0^2) \approx \epsilon^{-(2.18)^2/2} = 0.09$, all other terms are a factor 10 larger. So $p(z|at\_university) > p(z|at\_home)$.

**(c)** Now, suppose you have prior knowledge about your friend's habits which suggest that your friend is currently is at home with probability $P(at\_home) = 0.7$, at the university with $P(at\_university) = 0.3$ and at any other place with $P(other) = 0$. Use this prior knowledge to recalculate the posterior of b).

Answer:
So with the prior $P(at\_home)$, the posterior is $P(at\_home|z) \simeq P(z|at\_home)P(at\_home)$. With $P(z|at\_home) \simeq 0.093 \cdot 0.953$, $P(at\_home|z) \simeq 0.093 \cdot 0.953 \cdot 0.7 = 0.062$, while $P(at\_home|z) \simeq 0.845 \cdot 0.890 \cdot 0.3 = 0.226$, so your friend is still at the university ($m_0$). That is not strange, because the prior makes it a factor 2 more probable, while the term $\mathcal{N}(\delta_{1,0} - d_0, \sigma_0^2)$ was a factor 10 smaller than the other terms.

## Question 4

Consider a robot that resides in a circular world consisting of ten different locations that are numbered counterclockwise. The robot is unable to sense the number of its present location directly. However, places 0, 3, and 6 contain a distinct landmark, whereas all other places do not. All three of these landmarks look alike. The likelihood that the robot observes the landmark given it is in one of these places is 0.8. For all other places, the likelihood of observing the landmark is 0.4. For each place on the circle compute the probability that the robot is in that place given that the following sequence of actions is carried out deterministically and the following sequence of observations is obtained:

- the robot detects a landmark,
- moves 4 grid cells counterclockwise and detects a landmark,
- moves 4 grid cells counterclockwise and finally perceives no landmark.

Answer:
Lets first draw a sketch of the situation.

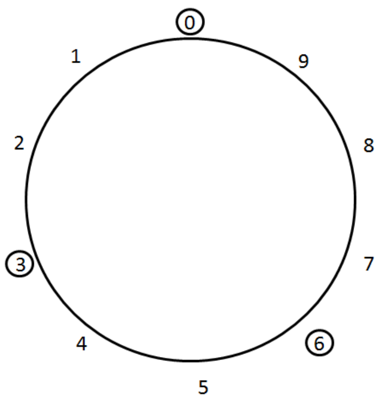


Figure 7: A circular world with three distinct landmarks.

Yet, it is easier to keep track of the beliefs in an array, as long as we remember that is a circular array. In the situation above are three measurement updates and two control updates (see Table 2.1). In principle normalization is possible after each timestep, but in this case normalization is only applied after the last observation.

| location | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| landmark | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| prior $\overline{bel}(x_0)$ | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| $p(z_0 \vert x_0)$ | 0.8 | 0.4 | 0.4 | 0.8 | 0.4 | 0.4 | 0.8 | 0.4 | 0.4 | 0.4 |
| $bel(x_0) = \eta p(z_0 \vert x_0)\overline{bel}(x_0)$ | 0.08 | 0.04 | 0.04 | 0.08 | 0.04 | 0.04 | 0.08 | 0.04 | 0.04 | 0.04 |
| $\overline{bel}(x_1) = \int p(x_1 \vert u_1, x_0) bel(x_0)$ | 0.08 | 0.04 | 0.04 | 0.04 | 0.08 | 0.04 | 0.04 | 0.08 | 0.04 | 0.04 |
| $p(z_1 \vert x_1)$ | 0.8 | 0.4 | 0.4 | 0.8 | 0.4 | 0.4 | 0.8 | 0.4 | 0.4 | 0.4 |
| $bel(x_1) = \eta p(z_1 \vert x_1)\overline{bel}(x_1)$ | 0.064 | 0.016 | 0.016 | 0.032 | 0.032 | 0.016 | 0.032 | 0.032 | 0.016 | 0.016 |
| $\overline{bel}(x_2) = \int p(x_2 \vert u_2, x_1) bel(x_1)$ | 0.032 | 0.032 | 0.016 | 0.016 | 0.064 | 0.016 | 0.016 | 0.032 | 0.032 | 0.016 |
| $p(z_2 \vert x_2)$ | 0.2 | 0.6 | 0.6 | 0.2 | 0.6 | 0.6 | 0.2 | 0.6 | 0.6 | 0.6 |
| $bel(x_2) = \eta p(z_2 \vert x_2)\overline{bel}(x_2)$ | 0.0064 | 0.0192 | 0.0096 | 0.0032 | 0.0384 | 0.0096 | 0.0032 | 0.0192 | 0.0192 | 0.0096 |

The sum of all $p(z_2|x_2)\overline{bel}(x_2)$ is 0.1376, so the normalizer $\eta = 1/0.1376 = 7.27$. So, the probability, in percentage will be

| location | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| | 4.7% | 14.0% | 7.0% | 2.3% | 27.9% | 7.0% | 2.3% | 14.0% | 14.0% | 7.0% |

## Question 5

Compare the sparseness in GraphSLAM with the sparseness in SEIFs: what are the advantages and disadvantages of each? Provide conditions under which either would be clearly preferable. The more concise your reasoning the better.

Answer:

Both GraphSLAM and SEIF create information matrices $\Omega$ to store the correlatons between the landmarks and the robot path. The information matrix contain many elements where no or nearly no information is available, which makes it a sparse matrix. Yet, GraphSLAM and SEIF make different use of the sparseness. GraphSLAM uses an *factorization trick* to aggregate a number of soft constraints related to features in a smaller set of stronger constraints on the robot path. SEIF does precisely the opposite, and aggregates all information on the robot path in the latest robot location and constraints between landmarks. To keep this computational feasable, the update is not performed for all features but a limited set of active features.

The obvious criterion to select GraphSLAM or SEIF is if one needs an full SLAM or online SLAM. Both algorithm are based on information matrices, which mean that they are a good choice when information has to be combined (local submaps and multi-robot applications). SEIF has the benefit to add new information locally, which is a benefit if there are many possible correlations (a forest). GraphSLAM integrates the information in the path, so benefits if the map can be reduced to a skeleton (streets, corridors).

## Question 6

Imagine you are modelling the location of a robot with linear Kalman filters. Due to a circuit error, sometimes the measurement variance is a few orders of magnitude larger than expected. This happens with a small failure rate $p_f$.

**(a)** Explain what happens to your estimates.

Answer:
Your estimates will erratically move/bounce over the field, because the gaussian measurements are taken too precise. Gaussian estimates are very sensitive to outliers.

**(b)** Improve the filter, such that it fixes the above mentioned problem.

Answer:
Use validation gate. Reject all measurements for which the innovation factor exceeds a threshold.

**(c)** Consider that $p_f$ goes to 1, suggest an alternative improvement to the filter.

Answer:
Increase variance of measurement model (i.e. Q) to reflect the few order larger variance. The Kalman filter automatically will rely more on the motion model.

## Success!

## Acknowledgements

The first question is inspired by a Quiz from Sebastian Thrun's course "Artificial Intelligence for Robotics". The second question is original, although inspired by a question from the "Probabilistic Robotics" book [5] The third question is based on an assignment from the Albert-Ludwigs-Universität Freiburg, written by Wolfram Burgard. The fourth question is from the "Principles of Robot Motion" textbook [2]. The fifth question is directly from the "Probabilistic Robotics" book [5]. The last question is again original.

## References

[1] J. L. Blanco, J. A. Fernandez-Madrigal and J. Gonzalez, "Efficient probabilistic Range-Only SLAM", in "2008 IEEE/RSJ International Conference on Intelligent Robots and Systems", pp. 1017–1022, Sept 2008, ISSN 2153-0858, doi:10.1109/IROS.2008.4650650.

[2] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*, MIT Press, Pittsburgh, PA, June 2005.

[3] G. Kantor and S. Singh, "Preliminary results in range-only localization and mapping", in "Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)", volume 2, pp. 1818–1823 vol.2, 2002, doi:10.1109/ROBOT.2002.1014805.

[4] S. Singh, G. Kantor and D. Strelow, *Recent Results in Extensions to Simultaneous Localization and Mapping*, pp. 210–221, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, ISBN 978-3-540-36268-5, doi:10.1007/3-540-36268-1_18.

[5] S. Thrun, W. Burgard and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, The MIT Press, September 2005, ISBN 0-262-20162-3.