Lecture 9: *Sensor Models and Localization and Mapping*

Scribe: *Jingru Qin, Subhrajeet Ray, Anson Wong, Utkarsh Patel*

Contents

1 Measurement Models

Formally, the measurement model is defined as a conditional probability distribution, $p(z_t|x_t, m)$ where x_t is the robot pose, z_t is the measurement at time t, and m is the map of the environment. A map of the environment is a list of objects in the environment and their locations.

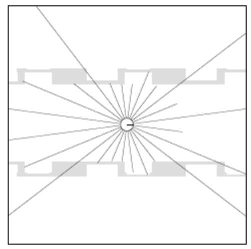

Figure 1: Ultrasound scan of a robot in its environment

Many sensors generate more than one numerical measurement value when queried.For example, cameras generate entire arrays of values (brightness, saturation, color); similarly, range finders usually generate entire scans of ranges. We will denote the number of such measurement values within a measurement z_t by K, hence write:

$$z_t = \left\{z_t^1, z_t^2, ..., z_t^K\right\} \tag{1}$$

We will use z_t^k to refer to an individual measurement(e.g., one range value). We will approximate $p(z_t|x_t, m)$ by the product of the individual measurement likelihoods

$$p(z_t|x_t, m) = \prod_k p(z_t^k|x_t, m) \tag{2}$$

This amounts to an assumption that the noise in each individual measurement value are independent of each other.

1.1 Maps

To express the process of generating measurements, we need to specify the environment in which a measurement is generated. A map of the environment is a list of objects in the environment and their locations. It is a representation of how the environment looks like.A map m is a list of objects in the environment along with their properties:

$$m = \{m_1, m_2, ..., m_N\} \tag{3}$$

Here N is the total number of objects in the environment, and each m_n with $1 < n < N$ specifies a property.

1.2 Sources of Error

The measurement model incorporates four types of measurement errors, measurement noise, errors due to unexpected objects, errors due to failures to detect objects, and random unexplained noise. The desired model $p(z_t|x_t, m)$ is therefore a consolidation of four densities, each of which corresponds to a particular type of error.

1.2.1 Local Measurement Noise

The measurement noise arises from the limited resolution of range sensors, atmospheric effect on the measurement signal, and so on. This noise is usually modeled by a narrow Gaussian with mean z_k^{t*} and standard deviation σ_{hit}. We will denote the Gaussian by p_{hit}. Figure 2 illustrates this density p_{hit}, for a specific value of z_k^{t*} .

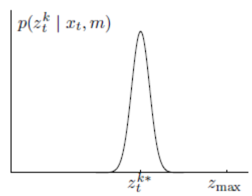

Figure 2: Measurement Noise

The values measured by the range sensor are limited to the interval [0, z_{max}], where z_{max} denotes the maximum sensor range. Thus, the measurement probability is given by

$$p_{hit}(z_k^t|x_t, m) = \left\{ \begin{array}{ll} \eta N(z_t^k; z_t^{k*}, \sigma_{hit}^2) & if 0 <= z_t^k <= z_{max} \\ 0 & otherwise \end{array} \right\} \quad (4)$$

where z_k^t is calculated from x_t and m via ray tracing, and $N(z_k^t; z_k^{t*}, \sigma_{hit}^2)$ denotes the univariate normal distribution with mean z_k^t and variance σ_{hit}^2, and η represents the normalizer.

1.2.2 Unexpected Objects

Generally, the likelihood of sensing unexpected objects decreases with range. To see, imagine there are two people that independently and with the same, fixed likelihood show up in the perceptual field of a proximity sensor. One person's range is z_1, and the second person's range is z_2. Let us further assume that $z_1 < z_2$, without loss of generality. Then we are more likely to measure z_1 than z_2. Whenever the first person is present, our sensor measures z_1. However, for it to measure z_2, the second person must be present and the first must be absent.

Mathematically, the probability of range measurements in such situations is described by an exponential distribution.

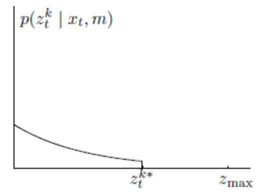

Figure 3: Unexpected Object

The parameter of this distribution, p_{short}, is an intrinsic parameter of the measurement model. According to the definition of an exponential distribution we obtain the following equation for $pshort(z_k^t|x_t, m)$:

$$p_{short}(z_k^t|x_t, m) = \left\{ \begin{array}{ll} \eta\lambda_{short}e^{-\lambda_{short}z_t^k} & if 0 <= z_t^k <= z_t^{k*} \\ 0 & otherwise \end{array} \right\} \tag{5}$$

As in the previous case, we need a normalizer η since our exponential is limited to the interval $[0, z_t^{k*}]$.

1.2.3 Failures

A typical result of sensor failures are max-range measurements: the sensor returns its maximum allowable value z_{max}. Since failures are quite frequent, it is necessary to explicitly model max-range measurements in the measurement model. We will model this case with a point-mass distribution centered at z_{max}

$$p_{max}(z_k^t|x_t, m) = \left\{ \begin{array}{ll} 1 & if z = z_{max} \\ 0 & otherwise \end{array} \right\} \tag{6}$$

Here p_{max} does not possess a probability density function because p_{max} is a discrete distribution.

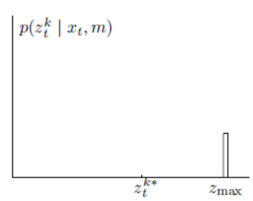

Figure 4: Failures

1.2.4 Random Measurements

Finally, range finders occasionally produce entirely unexplained measurements. For example, sonars often generate phantom readings when they bounce off walls, or when they are subject to cross-talk between different sensors. To keep things simple, such measurements will be modeled using a uniform distribution spread over the entire sensor measurement range $[0, z_{max}]$:

$$p_{rand}(z_k^t|x_t, m) = \left\{ \begin{array}{ll} 1/z_{max} & if 0 < z_k^t < z_{max} \\ 0 & otherwise \end{array} \right\} \tag{7}$$

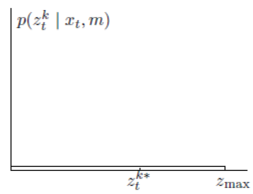

Figure 5: Random Measurements

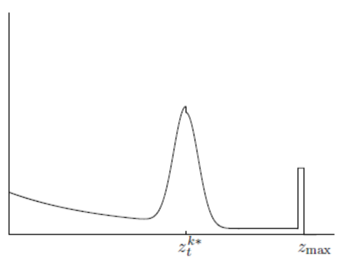

Figure 6: Pseudo Density of Mixture Distribution $p(z_k^t|x_t, m)$

These four different distributions are now mixed by a weighted average, defined by the parameters $z_{hit}, z_{short}, z_{max}, and z_{rand} with z_{hit} + z_{short} + z_{max} + z_{rand} = 1$.

$$p(z_k^t|x_t, m) = \begin{pmatrix} z_{hit} \\ z_{short} \\ z_{max} \\ z_{rand} \end{pmatrix}^T \cdot \begin{pmatrix} p_{hit}(z_k^t|x_t, m) \\ p_{short}(z_k^t|x_t, m) \\ p_{max}(z_k^t|x_t, m) \\ p_{rand}(z_k^t|x_t, m) \end{pmatrix} \tag{8}$$

2 Feature-based Sensor Models

An alternative approach to raw sensor measurements is to extract features. The advantages of this approach is that it efficiently reduces the computation complexity by extract a small number of features from high-dimensional sensor measurements. There are a variety of sensors that can extract a wide range of features. For example, we can extract features such as edges, lines, corners in Computer Vision.

2.1 Landmark Measurements

In Robotics, it is common to navigate by identifying physical objects in the real world and we call these objects *landmarks*. The most common sensor model for processing landmarks is to measure the range and the bearing of the landmark relative to robot's local coordinate frame.

We can denote the feature extractor as a function f. The features extracted from a range measurement are given by $f(z_t)$ where r is the range, θ is the bearing, and s is the signature. The feature vector is as follows:

$$f(z_t) = \left\{f_t^1, f_t^2, ...\right\} = \left\{ \begin{array}{c} r_t^1 \\ \phi_t^1 \\ s_t^1 \end{array}, \begin{array}{c} r_t^2 \\ \phi_t^2 \\ s_t^2 \end{array}, ... \right\} \tag{9}$$

Assuming conditional independence between each feature and independence between the noise in each individual measurement value $(r_t^i, \phi_t^i, s_t^i)^T$ we can calculate the probability of features given state x_t and map m:

$$p(f(z_t)|x_t, m) = \prod_i p(r_t^i, \phi_t^i, s_t^i | x_t, m) \tag{10}$$

where a map m contains a list of N objects in the environment and each object is marked with a Cartesian location on the map:

$$m = \{m_1, m_2, ..., m_N\} \tag{11}$$

Assuming that the sensor can measure the range and the bearing of the landmark relative to the robot's local coordinate frame. We can devise a feature-based landmark measurement model where the i-th feature at time t corresponds to the j-th landmark in the map and the robot pose is given by $x_t = (x\, y\, \theta)^T$:

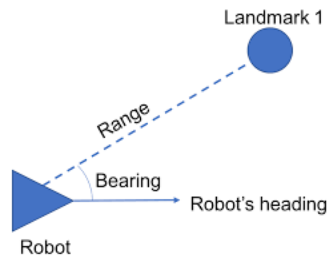

Figure 7: Landmark Measurement

$$\begin{pmatrix} r_t^i \\ \phi_t^i \\ s_t^i \end{pmatrix} = \begin{pmatrix} \sqrt{(m_(j,x) - x)^2 - (m_(j,y) - y)^2} \\ atan2^1(m_(j,y) - y, m_(j,x) - x) - \theta \\ s_j \end{pmatrix} + \begin{pmatrix} \varepsilon_(\sigma_r^2) \\ \varepsilon_(\sigma_\phi^2) \\ \varepsilon_(\sigma_s^2) \end{pmatrix} \tag{12}$$

Here $\varepsilon_(\sigma_r^2)$, $\varepsilon_(\sigma_\phi^2)$, and $\varepsilon_(\sigma_s^2)$ are zero-mean Gaussian error variables with variances σ_r^2, σ_ϕ^2 , and σ_s^2, respectively.

2.2 Practical Considerations

As noted above, the main motivation for using features instead of the full measurement vector is reduction in computation. However, a lot of information is lost by using features instead of full measurement. This lost information makes certain problems more difficult, such as the data association problem of determining whether or not the robot just re- visited a previously explored location.

[1] atan2 allows calculating the arctangent of all four quadrants whereas arctan only allows calculating from quadrants 1 and 4

3 Localization

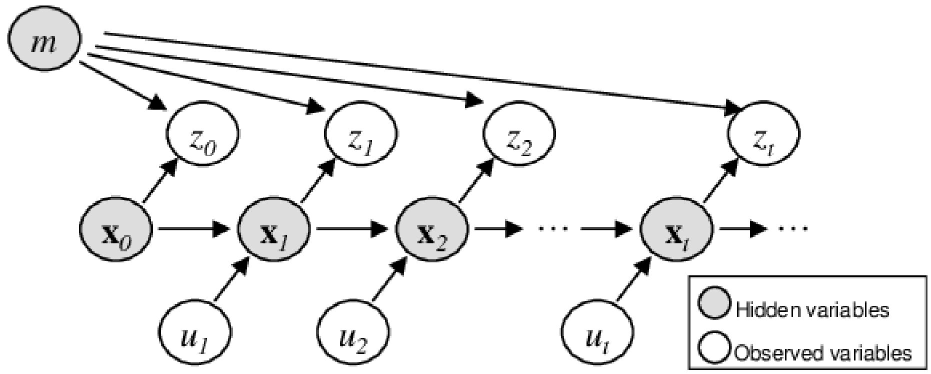

Figure 8: Robot Localization. Mobile robot localization can be represented by this dynamic Bayes network, which characterizes the evolution of controls, states and measurements. The values of the unshaded nodes are given: the map m, the measurements z, and the controls u. The goal of localization is to infer the robot pose variable x.

3.1 Extended Kalman Filter Localization

Extended Kalman Filter(EKF) localization represents beliefs $bel(x_t)$ by the mean μ_t and the covariance Σ_t.

Algorithm Extended_Kalman_filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):

$\bar{\mu}_t = g(u_t, \mu_{t-1})$

$\bar{\Sigma}_t = G_t \, \Sigma_{t-1} \, G_t^T + R_t$

$K_t = \bar{\Sigma}_t \, H_t^T (H_t \, \bar{\Sigma}_t \, H_t^T + Q_t)^{-1}$

$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$

$\Sigma_t = (I - K_t \, H_t) \, \bar{\Sigma}_t$

return μ_t, Σ_t

Figure 9: The extended Kalman filter algorithm.

For detail of basic EKF algorithm, please refer to the scribing for Lecture 7. The focus here is EKF localization algorithm.

Following the above environment where the robot is equipped with sensors for measuring range and bearing, we assume that a feature-based map m is given and all features in the map can be uniquely identifiable. A map is the representation of how world looks like. In robotics, there are three major map models: 1. Grid Based: A occupancy grid of free and occupied space-pixels. 2. Feature Based: A list of nearby features in the environment. e.g. list of nearby landmarks' location and bearings. 3. Topological: A list of nodes and their interconnections.

The EKF localization algorithm is illustrated in Figure 10. It requires as its input a Gaussian estimate of the robot pose at time $t-1$, with mean μ_{t-1} and covariance Σ_{t-1}. Further, it requires a control u_t, a map m, and a set of features $z_t = z_t^1, z_t^2, ...$ measured at time t, along with the correspondence variables $ct = c_t^1, c_t^2, ...$ Here, c_t^i can be thought of as a mapping from feature to object id.

3.1.1 Algorithm intuition

This section explains the EKF localization algorithm intuitively. For detail mathematical derivation of the algorithm, please refer to the book *Probabilistic Robotics* section 7.4.3 (page 205-208).

We use Kalman Filter for localization when the state transition function is linear. However, when the function is non-linear, in the form of $g(u_t, x_{t-1})$, Kalman Filter is no longer valid.

Recall that EKF linearization approximates the function g through a Taylor expansion:

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + G_t(x_{t-1} - \mu_{t-1})$$

G_t is the Jacobian matrix which represents the derivative of the function g with respect to x_{t-1} evaluated at u_t (the control input) and μ_{t-1} (the mean estimate).

Line 6 performs mean estimate update according to the state transition function (motion model) $x_t = g(\mu_t, x_{t-1})$. Substituting x by the mean μ, we get

$$\mu_t = g(u_t, \mu_{t-1}) + G_t(x_{t-1} - \mu_{t-1})$$
$$= g(u_t, \mu_{t-1}) + G_t(\mu_{t-1} - \mu_{t-1}) = g(u_t, \mu_{t-1})$$

In line 5, M_t represents how much uncertainty we have due to the dynamics of the system (i.e robot motion error). The standard deviation of this error is proportional to the commanded velocity. The parameters α_1 to α_4 are robot-specific error parameters. They model the accuracy of the robot.

We cannot directly add M_t (uncertainty of motion) to $\bar{\Sigma}_t$ (uncertainty of state). First, we need to find how does the uncertainty in motion propagate to the uncertainty in state. To this end, we need a Jacobian V_t, which is the derivative of the state transition function g with respect to control input u. More details on how the Jacobian helps to propagate the uncertainty (noise) between two spaces can be found here.

Line 7 performs covariance estimate update (similar to the one in EKF) according to noise from state transition and also motion noise. The term $V_t M_t V_t^T$ can be intuitively thought of as the mapping of motion noise from the control space to state space.

In line 8, Q_t is the covariance matrix of the measurement noise. It represents how much uncertainty we have from sensors when we measure range, orientation and signature. σ_r, σ_ϕ, and σ_s are the variance of range, orientation and signature.

From line 9 to 18, we loop through each measurement, and run regular EKF correction steps.

Line 10 is the equation for the correspondence problem. Given a feature, we need to find which object in the map it corresponds to. As stated above, we assume that the correspondence for all objects is known (i.e. all features in the map are uniquely identifiable).

In line 12, we compute the expected measurement $\bar{z}_t{}^i$. The first row of $\bar{z}_t{}^i$ is expected distance of the object from robot, and the second row is the expected orientation of the object. Same as EKF, we later use the expected measurement in line 16 to update our mean estimate. The difference between the expected measurement and the actual measurement is inversely proportional to kalman gain.

The algorithm outputs a new, revised estimate μ_t, Σ_t along with the likelihood of the feature observation p_{zt}. p_{zt} is not essential for EKF update, but is useful for outlier rejection or in the case of unknown correspondences.

Algorithm EKF_localization_known_correspondences($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, c_t, m$):

$$\theta = \mu_{t-1,\theta}$$

$$G_t = \begin{pmatrix} 1 & 0 & -\frac{v_t}{\omega_t}\cos\theta + \frac{v_t}{\omega_t}\cos(\theta + \omega_t\Delta t) \\ 0 & 1 & -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta + \omega_t\Delta t) \\ 0 & 0 & 1 \end{pmatrix}$$

$$V_t = \begin{pmatrix} \frac{-\sin\theta + \sin(\theta+\omega_t\Delta t)}{\omega_t} & \frac{v_t(\sin\theta - \sin(\theta+\omega_t\Delta t))}{\omega_t^2} + \frac{v_t\cos(\theta+\omega_t\Delta t)\Delta t}{\omega_t} \\ \frac{\cos\theta - \cos(\theta+\omega_t\Delta t)}{\omega_t} & -\frac{v_t(\cos\theta - \cos(\theta+\omega_t\Delta t))}{\omega_t^2} + \frac{v_t\sin(\theta+\omega_t\Delta t)\Delta t}{\omega_t} \\ 0 & \Delta t \end{pmatrix}$$

$$M_t = \begin{pmatrix} \alpha_1 v_t^2 + \alpha_2\omega_t^2 & 0 \\ 0 & \alpha_3 v_t^2 + \alpha_4\omega_t^2 \end{pmatrix}$$

$$\bar{\mu}_t = \mu_{t-1} + \begin{pmatrix} -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta + \omega_t\Delta t) \\ \frac{v_t}{\omega_t}\cos\theta - \frac{v_t}{\omega_t}\cos(\theta + \omega_t\Delta t) \\ \omega_t\Delta t \end{pmatrix}$$

$$\bar{\Sigma}_t = G_t\ \Sigma_{t-1}\ G_t{}^T + V_t\ M_t\ V_t{}^T$$

$$Q_t = \begin{pmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_\phi^2 & 0 \\ 0 & 0 & \sigma_s^2 \end{pmatrix}$$

for all observed features $z_t^i = (r_t^i\ \ \phi_t^i\ \ s_t^i)^T$ *do*

$$j = c_t^i$$

$$q = (m_{j,x} - \bar{\mu}_{t,x})^2 + (m_{j,y} - \bar{\mu}_{t,y})^2$$

$$\hat{z}_t^i = \begin{pmatrix} \sqrt{q} \\ \operatorname{atan2}(m_{j,y} - \bar{\mu}_{t,y}, m_{j,x} - \bar{\mu}_{t,x}) - \bar{\mu}_{t,\theta} \\ m_{j,s} \end{pmatrix}$$

$$H_t^i = \begin{pmatrix} -\frac{m_{j,x} - \bar{\mu}_{t,x}}{\sqrt{q}} & -\frac{m_{j,y} - \bar{\mu}_{t,y}}{\sqrt{q}} & 0 \\ \frac{m_{j,y} - \bar{\mu}_{t,y}}{q} & -\frac{m_{j,x} - \bar{\mu}_{t,x}}{q} & -1 \\ 0 & 0 & 0 \end{pmatrix}$$

$$S_t^i = H_t^i\ \bar{\Sigma}_t\ [H_t^i]^T + Q_t$$

$$K_t^i = \bar{\Sigma}_t\ [H_t^i]^T [S_t^i]^{-1}$$

$$\bar{\mu}_t = \bar{\mu}_t + K_t^i(z_t^i - \hat{z}_t^i)$$

$$\bar{\Sigma}_t = (I - K_t^i\ H_t^i)\ \bar{\Sigma}_t$$

endfor

$$\mu_t = \bar{\mu}_t$$

$$\Sigma_t = \bar{\Sigma}_t$$

$$p_{z_t} = \prod_i \det\left(2\pi S_t^i\right)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\ (z_t^i - \hat{z}_t^i)^T [S_t^i]^{-1} (z_t^i - \hat{z}_t^i)\right\}$$

return μ_t, Σ_t, p_{z_t}

Figure 10: The extended Kalman filter localization algorithm.