5 Markov Localization using Sampling

In this Section we develop the concept of Markov Localization, extending the ideas of Sections 2 and 3 to take into account sensor readings from the environment.

5.1 Markov Localization

Previously we developed an uncertainty model for robot motion. Now we discuss how to extend this to several motion steps, where at each step we also use sensor information to localize the robot's position. In general, under uncertain motion the error of the estimated robot position grows. Using sensor information and a map, it is possible to shrink that error estimate.

Notation: let *0,1,2,…* be a sequence of time steps. $\mathbf{X}_{n|m}$ is the estimated position of the robot at time *n*, given sensor information up to and including time *m*. Using this notation, the estimated position of the robot from dead-reckoning is $\mathbf{X}_{n|0}$, that is, no sensor information is utilized.

In Markov Localization, progression from $\mathbf{X}_{n|n}$ to $\mathbf{X}_{n+1|n+1}$ proceeds in two steps:

$$
\begin{aligned}
&\mathbf{X}_{n+1|n} \leftarrow \mathbf{X}_{n|n} \qquad \textbf{[Prediction]} \qquad\qquad \textbf{(5-1)}\\
&\mathbf{X}_{n+1|n+1} \leftarrow \mathbf{X}_{n+1|n} \quad \textbf{[Update]}
\end{aligned}
$$

The robot prediction/update process is *Markovian* if all information is encoded by the current pose of the robot. This means, for example, that the effect of sensor readings in the Update step is dependent only on the current position of the robot, not the past history of sensor readings.

In terms of PDFs, the Prediction step is based on a motion model $p(x \mid x', a)$, where *x* is the robot pose produced by an action *a* from the robot at *x'*.

$$p_{n+1|n}(x) = \int p(x \mid x', a)\, p_{n|n}(x')\, dx \qquad \textbf{[Prediction]} \qquad \textbf{(5-2)}$$

This Equation is exactly Equation 2-11, the addition of random variables for robot pose and robot motion.

The Update step incorporates the sensor model, using Bayes' rule to generate the posterior probability of a pose, given the sensor reading *s*. At each pose *x*, the probability of being at that pose is modulated by the probability of finding the sensor reading *s* from that pose:

$$p_{n+1|n+1}(x) = \alpha p(s \mid x)\, p_{n+1|n}(x) \qquad \textbf{[Update]} \qquad \textbf{(5-3)}$$

The factor α is a normalization factor, to make the posterior distribution integrate to 1.

5.2 Update using Samples

For a sampled distribution, the Update step of Equation 5-3 is readily implemented, by re-calculating the probability of each sample. The sensor model is $p(s \mid [x, y, \theta])$, the probability of getting sensor reading *s* from the pose $[x, y, \theta]$. For each sample $s_i = \langle [x_i, y_i, \theta_i], p_i \rangle$, the probability is updated by:

$$p_i' = p(s \mid [x, y, \theta])\, p_i \quad \textbf{[Update]} \qquad \textbf{(5-4)}$$

The updated sample set will now have non-uniform probabilities, and the probabilities will not be normalized (sum to 1). Normalization is easy to accomplish: just sum up the probabilities of the revised sample set, and divide each probability by the sum.

Why should we want a *uniform* sample set? Because it will concentrate samples where there is the most probability mass. A uniform sample set *Q*' can be constructed from a sample set *Q* by the following process:

Do N times:	**(5-5)**
Pick a sample from *Q* based on its likelihood, and add it to *Q'*, changing its probability to 1/N.	

To pick a sample from *Q* based on its likelihood, we use the technique described in Section 3.5, to transform from a uniform random value on [0,1] to a random value on a distribution represented by its *cumulative* distribution. First, we form the cumulative distribution *CQ* of *Q*:

$$\begin{aligned} &\langle \mathbf{x}_0, p_0 \rangle \\ &\langle \mathbf{x}_1, p_0 + p_1 \rangle \\ &\langle \mathbf{x}_2, p_0 + p_1 + p_2 \rangle \\ &\ldots \end{aligned} \qquad \textbf{(5-6)}$$

This is a cumulative distribution table. Given a value *v* on [0,1], we can use binary search to find the value of **x** for which the cumulative distribution is equal to *v*. Note that it is not possible to use interpolation on this table, since the sample poses are not ordered in any fashion.

Because the resampling is done by picking from the original distribution based on likelihood, this technique is called *importance resampling*.

5.3 Poisson Process

In formulating the sonar sensor model, we will need the concept of a *Poisson process*. Suppose we are looking at echoes from a number of targets in front of a sensor. Assume that echoes coming back from the targets have the following characteristics:

1. For any interval, the probability of getting an echo from some target is the same as that for any other equal, non-overlapping interval.
2. The mean number of echoes from an interval of size x is λx.

A process with these characteristics is called a Poisson process. The probability that there will be no hits from an interval of size *x* is given by:

$$P(NHI > x) = e^{-\lambda x} \qquad \textbf{(5-7)}$$

We are interested in the PDF for the random variable: there is *some* echo in the interval *x*. From the above formula, we can write the cumulative density function *F*(*x*) as:

$$F(x) = 1 - e^{-\lambda x} \qquad \textbf{(5-8)}$$

The PDF is the derivative of the CDF:

$$f(x) = \lambda e^{-\lambda x} \qquad \textbf{(5-9)}$$

Equations 5-7 and 5-9 will be key ones in formulating the sonar sensor model. Typically, λ will be a random target detection rate, in target hits per meter, e.g., 5% chance of a hit in 1 meter.

5.4 Sonar Sensor Model

The sonar sensor model is surprisingly complex. It must take into account the distance to the nearest target, the angle of the target to the sonar, and the noise characteristics of the sonar and its environment. Figure 5-1 shows the general setup of the modeling process.

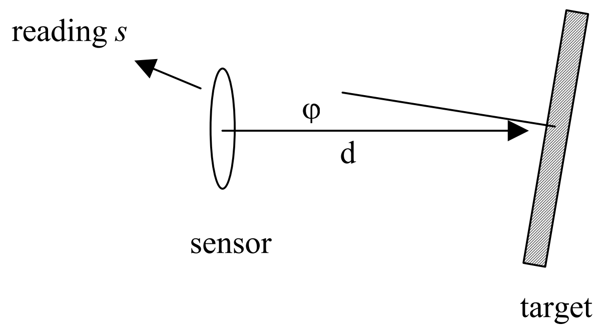

Figure 5-1 Sensor model geometry. The sensor is located a distance *d* from the nearest target wall. The wall is at an angle φ to the perpendicular. The sensor returns a reading *s*.

The sensor model is $p(s \mid d, \varphi)$. We could construct it experimentally, by putting a sonar sensor into an environment with a target at a distance *d* and angle ϕ, and then estimating the probability model by observing the frequency of responses *s*. However, this would take a lot of sensor readings, and in general it would be biased by the characteristics of the environment. Instead, we will use a theoretic model of the sonar sensor and the environment, and calculate the sensor response.

We will determine the sensor model in three steps:

1. On-axis sensor model (single reflection from target)
2. Off-axis sensor model (multi-reflections)
3. Gain reduction

5.4.1 On-Axis Sensor Model

In this case, we assume the target is perpendicular to the sensor, i.e., ϕ=0. In Figure 5-2, the general form of the PDF is diagrammed. There is an initial region where random targets are detected, either from misfirings of the sonar sensor electronics, or from unmodeled dynamic objects or other unknown obstacles in the environment. Then, the true sensor return occurs with a maximum at *s*=*d*, in the form of a normal distribution. Finally, there is some chance that no sensor reading is returned, that is, neither a random nor the true target is detected. This gives a probability blip at the maximum sensor range *m*.

Mathematically, the initial random target detection, up to the target distance *d*, is modeled as a Poisson process with rate λ_F, the *false positive* rate. The true target reflection is assumed to be randomly distributed around its distance *d*. Further, the chance of being detected decreases with increasing distance, so the normal is attenuated by a (linear) factor $\beta(d)$. After *d* and before *m*, the return is just from the true

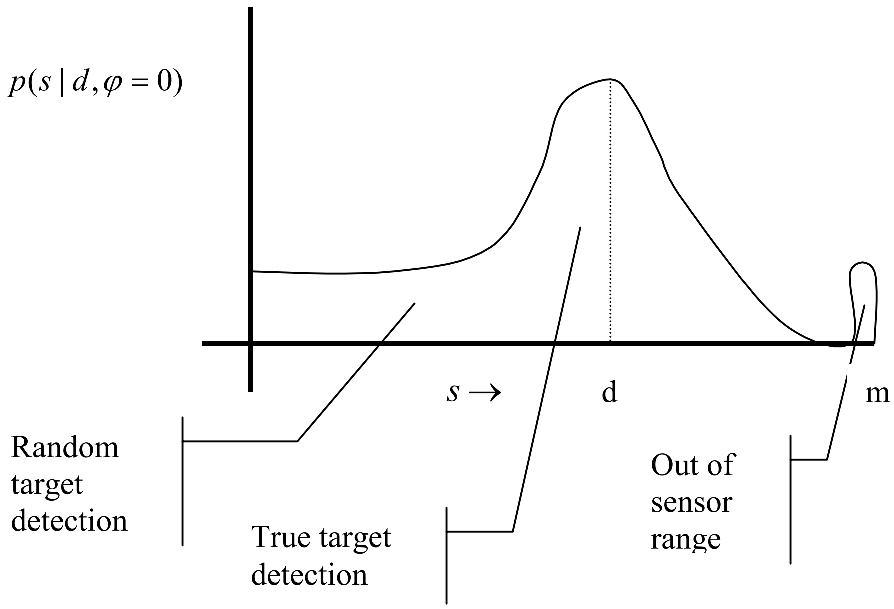

Figure 5-2 General form of the sensor model for on-axis target returns. Target is at distance *d*, perpendicular to the sensor normal.

target. In all cases, the true target return is further attenuated by the probability that a false target has already been detected. The total return is thus:

$$
\begin{aligned}
s < d: &\quad \lambda_F e^{-\lambda_F s} + \beta(d) N_{d,\sigma^2_{sonar}}(s) P(NHI > s) \\
d \le s < m: &\quad \beta(d) N_{d,\sigma^2_{sonar}}(s) P_F(NHI > d) \\
s = m: &\quad 1 - \int_0^d [s < d] ds - \int_d^m [d \le s < m] ds
\end{aligned}
\tag{5-10}
$$

5.4.2 Off-Axis Sensor Model

In this case, we assume the target is at a large angle to the sensor, i.e., |ϕ|>15. In Figure 5-3, the general form of the PDF is diagrammed. There is an initial region where random targets are detected, either from misfirings of the sonar sensor electronics, or from unmodeled dynamic objects or other unknown obstacles in the environment. Then, the true sensor return occurs starting at *s*=*d*. Instead of a normal distribution for a direct reflection, we have to model a *multi-reflection*, that is, the echo bouncing off several surfaces before returning. Finally, there is some chance that no sensor reading is returned, that is, neither a random nor the true target is detected. This gives a probability blip at the maximum sensor range *m*.

Mathematically, the initial random target detection, up to the target distance *d*, is modeled as a Poisson process with rate λ_F, the *false positive* rate. The multi-reflection return is also modeled as a Poisson process, with a different rate λ_R that will generally be greater than λ_F. In this case, we get a total PDF as follows:

$$
\begin{array}{ll}
s < d: & \lambda_F e^{-\lambda_F s} \\
d \le s < m: & \lambda_R e^{-\lambda_R s} \cdot P_F(NHI > d) \\
s = m: & 1 - \int_0^d [s < d] ds - \int_d^m [d \le s < m] ds
\end{array}
\quad \textbf{(5-11)}
$$

5.4.3 Mixtures and Gain

When the target is at an intermediate angle between 0 and 15 degrees, the PDF for the sensor model can be computed by interpolating appropriately between the on-axis and off-axis model for $0 \le x < m$. As usual, the final probability for x=m is computed by subtracting the PDF up to m from 1.

In practical applications, the theoretic sensor model must be modified because it will generally give results that are too certain. There are several reasons for this; the biggest ones are:

1. Sensor readings are not completely independent of each other, given robot pose.
2. The map can have errors.

To moderate the effects of the sensor model, a *uniform distribution* can be mixed in. The uniform distribution is a no-information sensor model, which does not change the prior.

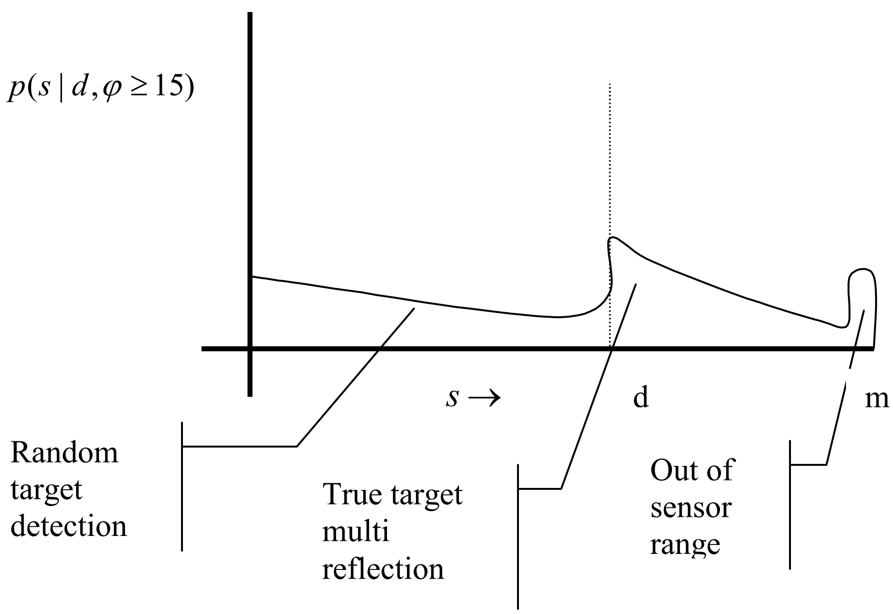

Figure 5-3 General form of the sensor model for off-axis target returns. Target is at distance *d*, perpendicular to the sensor normal.