Probabilistic Robotics Hw2

Tahsincan Köse

20 November 2018

1

a

Initially let s,c,r be the abbreviations for sunny, cloudy and rainy respectively. Given Day-1 is s, it is pretty straightforward to compute probabilities:

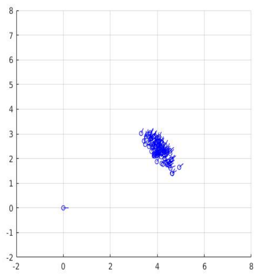
$$
\begin{aligned}
\text{P}(c_2|s_1) &= 0.2 \text{ from the table.}\\
\text{P}(c_3|s_1) &= \text{P}(c_3|c_2)\text{*P}(c_2|s_1) + \text{P}(c_3|r_2)\text{*P}(r_2|s_1) + \text{P}(c_3|s_2)\text{*P}(s_2|s_1)\\
&= (0.4 * 0.2) + 0.6 * 0 + (0.8 * 0.2)\\
&= 0.24\\
\text{P}(r_4|s_1) &= \text{P}(r_4|c_3)\text{*}[\text{P}(c_3|c_2)\text{*P}(c_2|s_1) + \text{P}(c_3|r_2)\text{*P}(r_2|s_1) + \text{P}(c_3|s_2)\text{*P}(s_2|s_1)]\\
&+ \text{P}(r_4|s_3)\text{*}[\text{P}(s_3|c_2)\text{*P}(c_2|s_1) + \text{P}(s_3|s_2)\text{*P}(s_2|s_1) + \text{P}(s_3|r_2)\text{*P}(r_2|s_1)]\\
&+ \text{P}(r_4|r_3)\text{*}[\text{P}(r_3|c_2)\text{*P}(c_2|s_1) + \text{P}(r_3|s_2)\text{*P}(s_2|s_1) + \text{P}(r_3|r_2)\text{*P}(r_2|s_1)]\\
&= 0.2 * [(0.4 * 0.2) + 0 + (0.8 * 0.2)] + 0 * [....] + 0.2 * [(0.2 * 0.2) + 0 + 0]\\
&= 0.056
\end{aligned}
$$

2

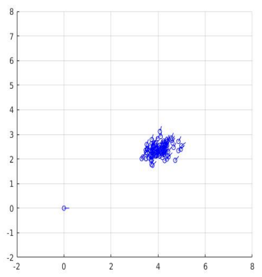
c

In Figure 1a, the noise parameters used are $a_1 = a_2 = a_3 = a_4 = a_5 = a_6 = 0.1$. Since all of the error parameters are designated as moderate values in this case, the resulting sample distribution have a relatively compact layout. This is simply due to the fact that overall added noise effectively creates a Gaussian distribution with a zero mean whose most samples correspond to the around-the-mean region.

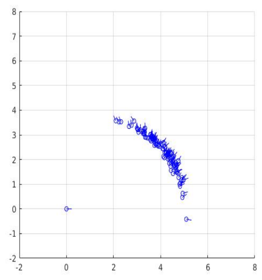
Figure 1b represents the case of small angular velocity and large translational velocity error. The noise parameters are $a_1 = a_2 = 0.2$ and $a_3 = a_4 = a_5 = a_6 = 0.05$. Because of the large translational error, the positions of samples are much more erroneous with respect to Figure 1a. However, due to small angular error the robot is located in the proximity of the goal each time. Contrarily, in Figure 1c the distribution of samples have a much thinner layout, because of the lower translational error. But in this case, due to larger angular error, the robot sometimes end up in radically wrong places. In essence, this implies the error in angular velocity may have relatively drastic effects on the motion model than the error in translational velocity. Error parameters are: $a_1 = a_2 = 0.05$, $a_3 = a_4 = 0.2$ and $a_5 = a_6 = 0.1$



(a) Moderate Error Parameters



(b) Small Angular Velocity Error and Large Translational Velocity Error



(c) Small Translational Velocity Error and Large Angular Velocity Error

Figure 1: Noisy Motion Models

These error parameters have a direct implication to the variances of sets that the noises are sampled. That is summation of a_1 and a_2, a_3 and a_4, a_5 and a_6

approximates respectively the first, second and third variance parameters.

3

5.1 asks the development of a motion model that computes the posterior over x' and $\dot{x}'$, from initial random variables x,$\dot{x}$ and $\ddot{x}$. In more compact form, it is required to have an implementation that outputs $P(x', \dot{x}'|x, \dot{x}, \ddot{x}_{actual})$. In order to compute this quantity, it is crucial to define these hypothetical entities such that:

$x' = x + \Delta t * \dot{x} + 1/2 * (\Delta t)^2 * \ddot{x}_{actual}$ (1)
$\dot{x}' = \dot{x} + \Delta t * \ddot{x}_{actual}$ (2)

If we were just sampling one of the possible outcomes from the PDF, it would be much more straightforward to compute $\ddot{x}_{actual}$ through the formula:

$\ddot{x}_{actual} = \ddot{x}_{commanded} + sample(\sigma)$

But since the question explicitly asks for the whole prob. distribution, there should be another random variable ϵ that represents the zero-mean Gaussian noise term with variance σ^2. Hence above formula becomes:

$\ddot{x}_{actual} = \ddot{x}_{commanded} + \epsilon$

which affects (1) and (2) in return, as follows:

$x' = x + \Delta t * \dot{x} + 1/2 * (\Delta t)^2 * \ddot{x}_{commanded} + 1/2 * (\Delta t)^2 * \epsilon$
$\dot{x}' = \dot{x} + \Delta t * \ddot{x}_{commanded} + \Delta t * \epsilon$ (2)

Rewriting them in matrix form would further facilitate seeing the linear Gaussian function property. Note that error parameter should be of the same dimension with the state.

$$\begin{bmatrix} x' \\ \dot{x}' \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 1/2 * (\Delta t)^2 \\ \Delta t \end{bmatrix} \ddot{x}_{commanded} + \begin{bmatrix} 1/2 * (\Delta t)^2 \\ \Delta t \end{bmatrix} \epsilon \ (3)$$

It is easily verifiable by the reader that above expression is of the form $x_t = A_t * x_{t-1} + B_t * u_t + \epsilon_t$ except the last term. It can be resolved with a simple substitution on ϵ parameter through introducing multivariate (2x1) Gaussian ϵ' which models the uncertainty introduced by the state transition. Such that:

$\epsilon' = B_{2x1} * \epsilon$, where $B_{2x1} = \begin{bmatrix} 1/2 * (\Delta t)^2 \\ \Delta t \end{bmatrix}$. Based on these information computing $R = B * B^T$ (covariance matrix of noise term) is trivial. Before that, the solution shall be written directly from the book. Since it is denoted that this model can be unrolled as a multivariate linear Gaussian probability distribution function, the exact form is:

$$p(x', \dot{x}' | x, \dot{x}, \ddot{x}_{actual}) = det(2\pi R)^{-1/2}$$
$$exp\{-\frac{1}{2}(\begin{bmatrix} x' \\ \dot{x}' \end{bmatrix} - A * \begin{bmatrix} x \\ \dot{x} \end{bmatrix} - B * \ddot{x}_{commanded})^T R^{-1}$$
$$(\begin{bmatrix} x' \\ \dot{x}' \end{bmatrix} - A * \begin{bmatrix} x \\ \dot{x} \end{bmatrix} - B * \ddot{x}_{commanded})\},$$
where $A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$, $B = \begin{bmatrix} 1/2 * (\Delta t)^2 \\ \Delta t \end{bmatrix}$ and
$R = \begin{bmatrix} 1/4(\Delta t)^4 & 1/2(\Delta t)^3 \\ 1/2(\Delta t)^3 & (\Delta t)^2 \end{bmatrix}$

Above pdf is the mathematic model which can be unrolled until its atomic sub-components to compute the probabilities of hypothesized states. In order to determine whether pose and velocity states are correlated, following formula should be used:

$corr(i, j) = \frac{cov(i, j)}{\sqrt{cov(i, i)}\sqrt{cov(j, j}}$. In this case, it is obvious that pose and velocity is correlated since Covariance matrice is not a Diagonal matrice, i.e. there are non-zero entries in the locations other than diagonal of the matrice R. Computing it:

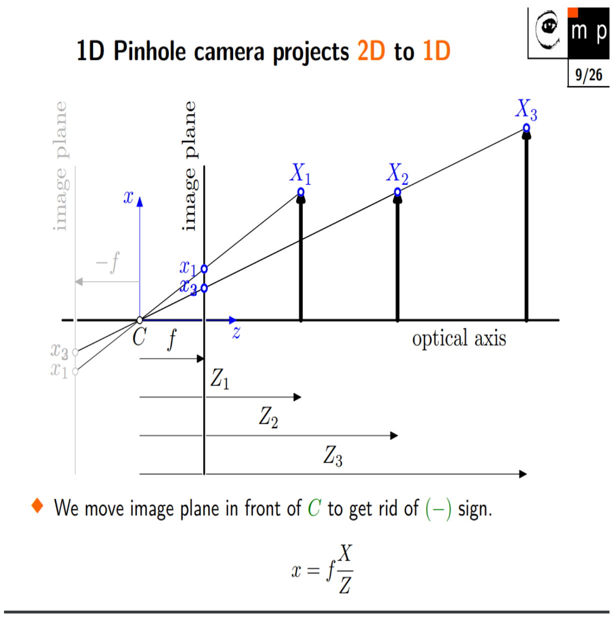
$corr(i, j = \frac{1/2(\Delta t)^3}{\sqrt{1/4(\Delta t)^4 (\Delta t)^2}} = 1$ denotes that pose and velocity is completely correlated.

4

a

As I understand, a camera that exists in 2D must project what it perceives into 1D. The figure below represents it graphically.

Figure 2 outlines an infinite image plane and the behavior of it in 2D space. In our case, the camera must have also a FOV limitation in which the image plane itself becomes constrained. Assume its max length as L. Consequently, deterministic parameters reveal out as f and L which can be implicitly built into λ as $atan2(F, L/2)$. The pinhole camera effectively represents the monochrome intensity of any location (x, y) in the map with a continuous pixel in its 1D image plane. Only assumption on camera model is that it should have a fixed orientation with respect to the robot for a simpler mathematical model.



1D Pinhole camera projects 2D to 1D

- We move image plane in front of C to get rid of $(-)$ sign.

$$x = f\frac{X}{Z}$$

Figure 2: 1D Pinhole

b

In the beginning, let z_t^* be the *true* pixel coordinate which is computed through pose (x, y, θ) and map m. Introducing a white Gaussian noise creates a normal distribution in the form of $N(z_t; z_t^*, \sigma^2)$ that adds required flexibility to the model so that it can sustain its robustness under uncertain conditions. It can be represented as a Gaussian function:

$$N(z_t; z_t^*, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(z_t - z_t^*)^2}{\sigma^2}} \quad (4)$$

c

Equation (4) can be utilized in closed-form solution. Moreover, in this case λ should be considered as well. λ is the angle of camera's field of view. Since this is a symmetric camera in terms of its focal point, $z_{max} = L/2$ and $z_{min} = -L/2$ due to its FOV angle λ. Hence, the equation becomes:

$$p(z|x, m, \lambda) = \begin{cases} \eta N(z; z^*, \sigma^2) & z_{min} \leq z \leq z_{max} \\ 0 & otherwise \end{cases}$$

, where η represents the normalizer whose divisor is integral of Gaussian function from z_{min} to z_{max}. This effectively prohibits any posterior probability of

”out-of-camera” pixel being 0.

d

In this part, the range between z_{min} and z_{max} must be discretized sufficiently in order to have a good approximation of the mathematical model with a computational approach, outlined below.

Algorithm 1 algorithm_camera_model(z,x,m)

```
$q = 1$
$z' = z_{min}$
**while** $z' < z_{max}$ **do**
    $q = q * p(z'|x, m, \lambda)$
    $z' = z' + 0.01$
```