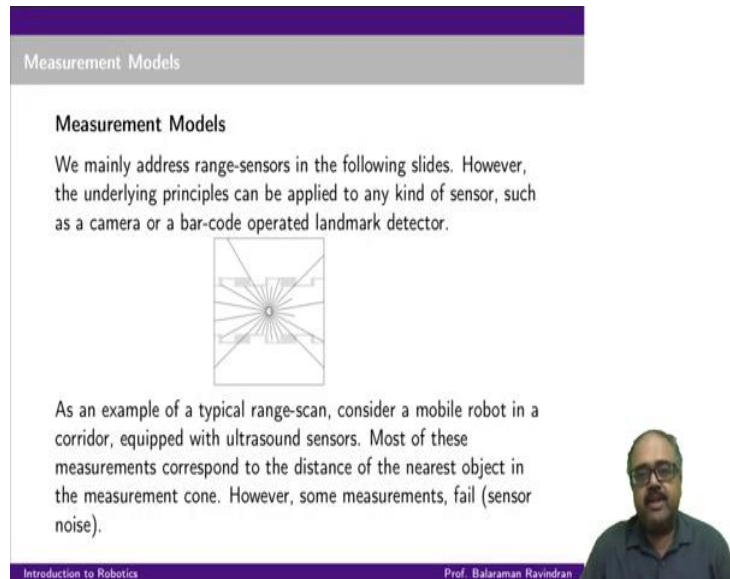


Introduction to Robotics
Professor Balaraman Ravindran
Department of Computer Science
Indian Institute of Technology, Madras
Lecture - 40
Range Finder Measurement Model

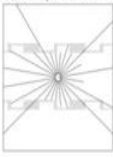
(Refer Slide Time: 00:14)



The slide is titled "Measurement Models" and contains the following text:

Measurement Models

We mainly address range-sensors in the following slides. However, the underlying principles can be applied to any kind of sensor, such as a camera or a bar-code operated landmark detector.



As an example of a typical range-scan, consider a mobile robot in a corridor, equipped with ultrasound sensors. Most of these measurements correspond to the distance of the nearest object in the measurement cone. However, some measurements fail (sensor noise).

The slide also features a video inset of Prof. Balaraman Ravindran in the bottom right corner. The slide footer includes "Introduction to Robotics" and "Prof. Balaraman Ravindran".

Welcome to the final lecture of week 11. And so, in this lecture, we will look at the last component that we need for making our estimation models work. So our state estimation models work, which is essentially the measurement model.

Remember, the measurement model tells you what is the probability of z given x_t . What is the probability of z_t given x_t . And if you are using a map, it is given x_t comma m . So the measurement model tells you what is the probability of z_t given x_t comma m .

And so, what I am going to do in the next few slides is basically look at one specific kind of sensor and develop a measurement model for that sensor alone. And so, we are going to look at what are called range sensors in the following slides. But whatever principles I am talking about now, so they basically can apply to other kinds of sensors as well, whether it is a camera sensor or a barcode-operated landmark detector, and so on, so forth.

So in fact, there is a funny story about which one of my students did this, when there was this job to build a robot for, to move around in office space in the company, while people were trying to come up with very complicated algorithms for localizing the robot, figuring out which room or which cubicle the robot was on. He came up with a very simple solution.

He printed unique barcodes for each location, and then he kept the robot with a barcode reader. So the robot just moves to a particular cubicle, reads the barcode and figures out exactly which cubicle it is in. It did not have to worry about all the cubicles looking similar, right. So sometimes, if you get the right engineering solution, problems become easier than trying to come up with something more sophisticated.

But anyway, so getting back to the main lecture here. So the idea here is I am going to look at a typical sensor. So it could be, for example, an ultrasound sensor. So here is a illustration of that. So there is a mobile robot in the corridor, and it is basically has a range finder, these multiple ultra sound detectors running off in different directions.

So typically, each of these ultra sound detectors is going to return the distance to the, the nearest object in the direction of the scan. So we can see that each of these rays here is one direction in which the ultrasound sensor is scanning. And most of the cases it is returning to you the distance of the nearest obstacle. In some cases, it fails. There is an obstacle here, it fails to detect. There are a couple of cases where it fails to detect objects.

In the other cases, it basically stop short. So even though there is no object here, it basically returns that there is no. So some kind of failure of the sensors could also happen. But typically, it tries to detect what is the distance to the nearest obstacle in the direction of the scan. And if there is no obstacle, it is supposed to give you back whatever is the maximum value for the sensor range.

In that case, you know that in that direction of scan, there are no obstacles. So that is basically what the rain sensor is going to give you and let us see how you can put together actual model for this rain sensor.

(Refer Slide Time: 03:42)



Measurement Models

Many sensors generate more than one numerical measurement value when queried. For example, range finders usually generate entire scans of ranges. We will denote these measurement values as:

$$z_t = \{z_t^1, \dots, z_t^K\}$$

We use z_t^k to refer to an individual measurement (e.g., one range value).


Further, we will **assume independence** between individual measurement likelihoods to approximate $p(z_t | x_t, m)$:

$$p(z_t | x_t, m) = \prod_{k=1}^K p(z_t^k | x_t, m)$$




Introduction to Robotics Prof. Balaraman Ravindran

Measurement Models

We mainly address range-sensors in the following slides. However, the underlying principles can be applied to any kind of sensor, such as a camera or a bar-code operated landmark detector.



As an example of a typical range-scan, consider a mobile robot in a corridor, equipped with ultrasound sensors. Most of these measurements correspond to the distance of the nearest object in the measurement cone. However, some measurements fail (sensor noise).



Introduction to Robotics Prof. Balaraman Ravindran

So just like we saw now, many sensors generate more than one numeric measurement value. So if I look at range finder as a sensor, it is going to give me the all the ultrasound readings that we saw. So we saw multiple ultra sound readings. So each one of these is going to return back a specific range value.

And so, when I say I am using a single sensor, which is rain sensor, it could still correspond to a vector of measurements. So we are going to assume that these are z_1 to z_K and at every time t , I am going to have all of these capital K measurements available to me. And so at every time step t , I am going to make this measurement.

And I am going to use $z_{t,k}$, small k , for a specific individual measurement. So it could be a specific value. So this could be k equal to 1, this could be k equal to 2, this could be k equal

to 3, and so on, so forth. So this was specific values I will be looking at each one of these measurements.

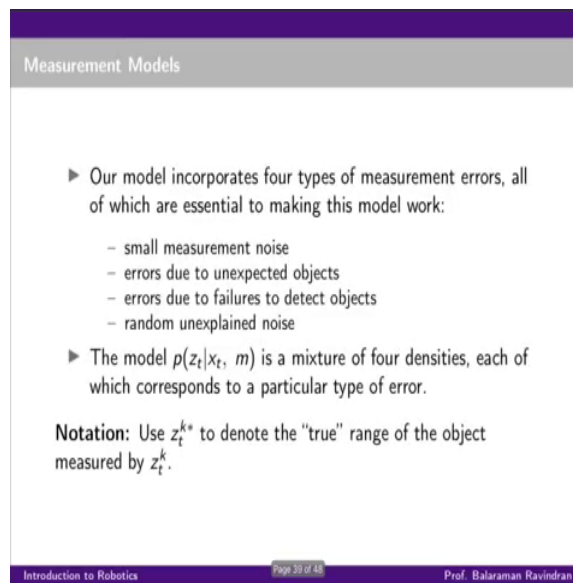
And the other thing that we are going to make life easier, so we are going to make an assumption that will make life easier is that I am going to assume that each of these different values that my range finder returns are independent. So probability z_t given x_t comma m is actually equal to the product over k of each of the individual sensors.

So probability that z_{t1} given x_t comma m times, probability of z_{t2} given x_t comma m , and so, on so forth. So that is basically what I am going to assume to make my life easier. You can see that almost surely this is not true, but we are just going to make that assumption. So if I, if the map had not been given to me, if I did not have a map, the probability that this being true is even lesser, because without knowing the map if I hit an obstacle in a particular range, then it is quite likely that I will see an obstacle in a slightly displaced.

If I see an obstacle in the direction θ is equal to 5, I am going to likely see something in that direction θ equal to 6. But given that I know the map, so when I say I know the map, that means, I know where the obstacle is exactly.

In such a case, when you know where the obstacle is, the probability of me getting a reading that obstacle is there and I am pointing at 5 degrees or 6 degrees is independent of whether I got the obstacle reading, when I pointed at 6 degrees. So because I know that there is an obstacle at that distance because of the map. If I did not know, then this would have been a harder independence to write. So this gives me a little bit more leeway here.

(Refer Slide Time: 06:25)



Measurement Models

- ▶ Our model incorporates four types of measurement errors, all of which are essential to making this model work:
 - small measurement noise
 - errors due to unexpected objects
 - errors due to failures to detect objects
 - random unexplained noise
- ▶ The model $p(z_t|x_t, m)$ is a mixture of four densities, each of which corresponds to a particular type of error.

Notation: Use z_t^{k*} to denote the “true” range of the object measured by z_t^k .

Introduction to Robotics Page 33 of 48 Prof. Balaraman Ravindran

So what we are going to do is, if you look, if you remember that we talked about multiple different kinds of errors that were happening, in some cases, there was an obstacle where it was missed. In other cases, there was no obstacle but it still returned an obstacle, and so on, so forth.

So I am going to break down the kinds of errors into four different quantities. So I am going to say there is a measurement noise, a small measurement noise. Basically, this is due to things like, you know, temperature variations, the sensor is getting a little bit heated or even atmospheric variations, and things like that. So that could cause a slight change in the reading. So that we call as a small measurement noise.

And then the second kind of errors are due to unexpected objects. The map says there is no object, but there might be an object, and therefore I am sensing something which I am, when I am expecting to not sense anything.

And the third kind of error is due to failure to detect an object, could be because suddenly the refractive index is very high or the object is too black and I am not able to actually get any reflection out of it, so whatever. And quite depending on the kind of sensor that you are using, there are many reasons why the sensor might fail to detect an object, so how to accommodate for that.

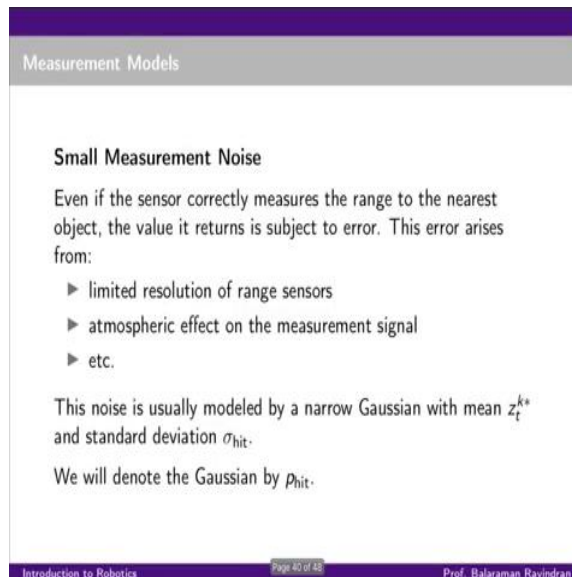
And finally, the last thing is, despite however clever I am, there is always a chance that something goes wrong, and how to accommodate for that. So I am going to say that the model that I have, the probability of z_t given x_t comma m is a mixture of all these four

sources of error. So the overall stochasticity in the measurement comes from all these four. And we will look at how to model each of these densities.

Before we go on, so just a small notation, when I say z_t^k , suppose there is an object in the direction that the k th sensor is looking at time t , then z_t^k is the true distance of the object. So z_t^k is the actual measurement. z_t^k is a measurement that is going to be influenced by all of these errors.

So z_t^k is a measurement that is influenced by all of these errors and z_t^k is the true value of the distance of the object. z_t^k is what z_t^k is trying to measure, but it is getting corrupted by all these noises. It is just a notational thing. We will see how we use it in the next slides.

(Refer Slide Time: 08:57)



Measurement Models

Small Measurement Noise

Even if the sensor correctly measures the range to the nearest object, the value it returns is subject to error. This error arises from:

- ▶ limited resolution of range sensors
- ▶ atmospheric effect on the measurement signal
- ▶ etc.

This noise is usually modeled by a narrow Gaussian with mean z_t^k and standard deviation σ_{hit} .

We will denote the Gaussian by ρ_{hit} .

Introduction to Robotics Page 42 of 88 Prof. Balaraman Ravindran

So the first thing we are going to look at is a small measurement noise. So the sensor roughly gets the range to the object, roughly gets the distance to the nearest object in the direction of the sensor. But then the sensor could have a limited resolution, so it could basically be rounding off because it does not have enough resolution.

Now, there could be things like, like I said earlier, the sensor could heat up or there could be some kind of an atmospheric effect that is affecting the signal, and so, on so forth. Multiple reasons. So what, how we model this is as a very narrow Gaussian, whose mean is z_t^k , I do not know this, but for the modelling purposes, I am assuming that z_t^k is the true distance and I have a standard deviation for this Gaussian, which we call a sigma hit.

Remember, this has to be unidimensional Gaussian, because measuring a single variable is a distance to the object. So it is like univariate Gaussian, not a multivariate Gaussian.

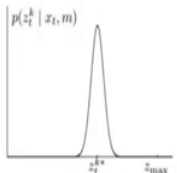
(Refer Slide Time: 09:57)


Measurement Models

In practice, 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:

$$p_{\text{hit}}(z_t^k | x_t, m) = \begin{cases} \eta \mathcal{N}(z_t^k; z_t^{k*}, \sigma_{\text{hit}}^2), & \text{if } 0 \leq z_t^k \leq z_{\max} \\ 0, & \text{otherwise} \end{cases}$$

Gaussian distribution p_{hit}





Introduction to Robotics
Page 41 of 48
Prof. Balaraman Ravindran

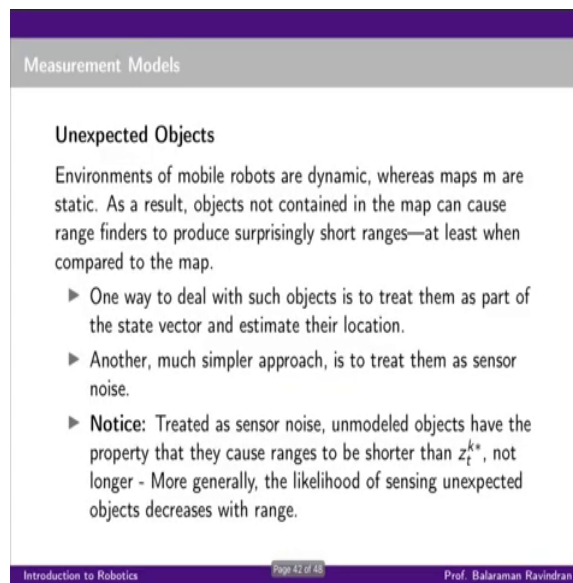
So I am going to model it like this. So z_t^{k*} is the actual distance and I am going to model it as a narrow Gaussian. And remember that, so we will denote this by p_{hit} . And remember that this measurement value cannot exceed z_{\max} , and it cannot go below 0 either. Because the distance is 0, I mean, so it cannot go below 0.

And z_{\max} is the maximum value that this sensor can measure. So it cannot go beyond 0 or z_{\max} and therefore, it is not really a Gaussian, because we are going to truncate it. So between zero and z_{\max} , if z_t^k actually the value that you are plugging in here, if it is between 0 and z_{\max} , then the probability is given by a Gaussian, which we call z_t^{k*} , so where z_t^{k*} is the mean and σ_{hit} is the variance of this Gaussian.

And we add a normalization parameter η , so that we compensate for the probability mass that was cut off beyond z_{\max} and beyond 0. Basically, we renormalize this by dividing by this, the area under 0 to z_{\max} . So that is basically what η is.

And if it is outside the 0 to z_{\max} range, the probability is 0. So it can never happen. So now we say this is the distribution p_{hit} . So this is the noise that comes from small measurement errors. If there was no error, I would have measured it as z_t^{k*} .

(Refer Slide Time: 11:28)



Measurement Models

Unexpected Objects

Environments of mobile robots are dynamic, whereas maps m are static. As a result, objects not contained in the map can cause range finders to produce surprisingly short ranges—at least when compared to the map.

- ▶ One way to deal with such objects is to treat them as part of the state vector and estimate their location.
- ▶ Another, much simpler approach, is to treat them as sensor noise.
- ▶ **Notice:** Treated as sensor noise, unmodeled objects have the property that they cause ranges to be shorter than z_t^k , not longer - More generally, the likelihood of sensing unexpected objects decreases with range.

Introduction to Robotics Page 44 of 48 Prof. Balaraman Ravindran

The second source of error is due to unexpected objects. So why is this happening? So typically, we are assuming that maps are static. I am assuming that there are objects at certain positions in the map and I am not updating the map on a very frequent basis.

But typical environments, which mobile robots are operating could be dynamic, I mean, there could be other robots moving around, there could be people moving around, or even things like paper flying around and stuff like that. And these are objects that are not contained in the map and, but can make the range finder give you a very short reading.

Suppose there is an object at this distance from the robot, but there is a paper flying somewhere in between. So when the robot is taking reading, it will hit the paper and it will come back, it will not get to the actual distance of the object. So it gives you a much shorter reading than the actual distance to the object z_t^k .

So one way to think about this, let me put all these moving objects into the map or putting, put all these moving objects into the state. So the map is there but I can put them into the state and I will estimate their location as well. But this is very hard. I mean, come on, paper flying around all those things it is hard to.

So what we will do is, instead of that, we will just treat it as a sensor noise. We will treat it sensor noise. So what do I mean by that? I will just say occasionally, my sensor might give you a wrong information. We will account for that as well. So one thing you should note here, when I start reading this as a sensor noise, so remember that I have I have a cone. So if

there is an object that is passing by very close to the robot, it is very likely to be sensed because it is going to block my sensor, quite likely.

But if it is flying away, flying farther away from the robot. This is the cone, if it is flying close, I am very likely to sense it. If there is a moving object that is farther away from me, I might not sense it, depending on the quality of the sensor, the resolution and things like that, I might actually miss this object and then most of my sense, my ultrasound emission might go past it and actually hit the true obstacle.

So what it really sums up is, so the closer the object, the moving object is to the robot or is to the sensor, the more likely that I will sense that object. Therefore, the likelihood of sensing these kinds of unexpected objects decreases with the range. The closer you are, the more likely that I will sense an unexpected object.


(Refer Slide Time: 14:09)


Measurement Models

Mathematically, the probability of range measurements in such situations is described by an exponential distribution. The parameter of this distribution, λ_{short} , is an intrinsic parameter of the measurement model.

$$p_{\text{short}}(z_t^k | x_t, m) = \begin{cases} \eta \lambda_{\text{short}} \exp(-\lambda_{\text{short}} z_t^k), & \text{if } 0 \leq z_t^k \leq z_{\text{max}} \\ 0, & \text{otherwise} \end{cases}$$

Exponential distribution p_{short}



Introduction to Robotics
Prof. Balaraman Ravindran


So what we do to accommodate for that is to treat the whole thing as an exponential distribution. So the closer I am to the robot, the closer I am to a distance of 0. There is a higher probability. So the farther I am, I have a lower probability.

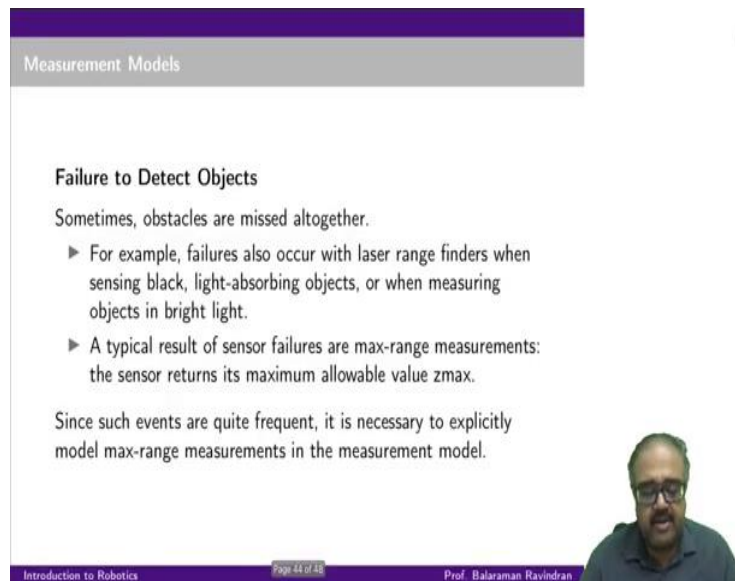
And notice that beyond z_t^k , I do not care. Because I would have hit the actual obstacle, so I do not care if there is an unexpected object after the obstacle. I am not going to see that. So only before the obstacle I am likely to see the unexpected object.

So I expect to see an unexpected object before the obstacle. Therefore, so this should, this is, this is again incorrect. So this will be between $0 \leq z_t^k < z_t^k$, not z_{max} . It should

be $z_t < k$. And so I am going to model this as an exponential distribution with the parameter λ short.

Again, I have a normalizing factor because after $z_t < k$, I set the probability to 0. So all this probability mass has to be redistributed before $z_t < k$ and we will take care of that. So this is the second model.

(Refer Slide Time: 15:25)



The slide is titled "Measurement Models" and features the NPTEL logo in the top right corner. The main content is under the heading "Failure to Detect Objects". It states: "Sometimes, obstacles are missed altogether." followed by two bullet points: "▶ For example, failures also occur with laser range finders when sensing black, light-absorbing objects, or when measuring objects in bright light." and "▶ A typical result of sensor failures are max-range measurements: the sensor returns its maximum allowable value z_{max} ." Below this, it says: "Since such events are quite frequent, it is necessary to explicitly model max-range measurements in the measurement model." A video inset in the bottom right shows Prof. Balaraman Ravindran speaking. The footer contains "Introduction to Robotics", "Page 44 of 48", and "Prof. Balaraman Ravindran".

And what is the third one? So the third one is when I fail to detect object altogether. So what happens when I fail to detect objects? That means that I have completely missed the object, so whatever reason. So it could be because the sensor failed, it just stuck somewhere. So my sensor keeps just returning the maximum value.

Or it could fail because as object was very good at absorbing the light that it was emitting, and therefore it just did not get any bounce back. and therefore it just assumed that there is nothing in that direction and that till the range of, the maximum range, the thing is free.

So at no, no conditions will I actually accept a reading that is greater than the maximum range. I will not accept a reading greater than z_{max} , because does not make sense. Because z_{max} is the maximum that the sensor could look at.

(Refer Slide Time: 16:15)

Measurement Models

We will model this case with a point-mass distribution centered at z_{\max} .

$$p_{\max}(z_t^k | x_t, m) = \mathbb{I}(z = z_{\max}) = \begin{cases} 1, & \text{if } z = z_{\max} \\ 0, & \text{otherwise} \end{cases}$$

point-mass distribution p_{\max}

Introduction to Robotics Page 45 of 46 Prof. Balaraman Ravindran

And so, what we will do is, we will just model it like a short noise. So it is just a point mass at z_{\max} . So point mass is z_{\max} . And this is essentially the probability that I completely miss the object at z_t^k . So I will just basically return the value z_{\max} .

So now, the thing is probability that that z_t^k is equal to z_{\max} , this 1, it is 0 otherwise. So for this component of it, just assigns some additional mass to z_{\max} .

(Refer Slide Time: 16:50)

Measurement Models

Random Measurements

Finally, range finders occasionally produce entirely unexplained measurements.

- ▶ sonars often generate phantom readings when they bounce off walls
- ▶ sensor readings may be subject to cross-talk with other different sensors.
- ▶ etc.

Introduction to Robotics Page 46 of 46 Prof. Balaraman Ravindran

Now we are coming to the last one, which is the random measurement. So it could basically miss the whole thing or it could generate some kind of a phantom reading when they bounce off walls. I mean, so there could be, I could detect something as being much closer than it is

actually is. There could be some kind of a cross talk, some interference with other sensors that makes me put it at some, some random location.

(Refer Slide Time: 17:13)

Measurement Models

To keep things simple, such measurements will be modeled using a uniform distribution spread over the entire sensor measurement range $[0; z_{\max}]$:

$$p_{\text{rand}}(z_t^k | x_t, m) = \begin{cases} \frac{1}{z_{\max}}, & \text{if } 0 \leq z_t^k \leq z_{\max} \\ 0, & \text{otherwise} \end{cases}$$


Uniform distribution p_{rand}

Introduction to Robotics Page 47 of 48 Prof. Balaraman Ravindran

So I am not going to overthink this. And this is going to say that, hey, look, after all of this careful consideration, there is a chance that I might mess, I could put the object anywhere between 0 to z_{\max} . I could put the object anywhere between 0 to z_{\max} . So I will say the probability is $1/z_{\max}$, this is basically uniform distribution.

So what are the things, we have four different things right. So we had a Gaussian, then we had exponential, we had a short noise, like a impulse function. And now, we have a uniform distribution. So there are four different probability distributions and the actual probability of z_t , given x_t comma m is actually a mix of all these four distributions as we will see here.

(Refer Slide Time: 17:59)



Measurement Models


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:

$$z_{hit} + z_{short} + z_{max} + z_{rand} = 1$$

```
1: Algorithm beam_range_finder_model( $z_t, x_t, m$ ):
2:    $q = 1$ 
3:   for  $k = 1$  to  $K$  do
4:     compute  $z_t^k$  for the measurement  $z_t^k$  using ray casting
5:      $p = z_{hit} \cdot p_{hit}(z_t^k | x_t, m) + z_{short} \cdot p_{short}(z_t^k | x_t, m)$ 
6:        $+ z_{max} \cdot p_{max}(z_t^k | x_t, m) + z_{rand} \cdot p_{rand}(z_t^k | x_t, m)$ 
7:      $q = q \cdot p$ 
8:   return  $q$ 
```

After iterating through all sensor measurements z_t^k in z_t , the algorithm returns the desired probability $p(z_t | x_t, m)$.

Introduction to Robotics Page 44 of 44 Prof. Balaraman Ravindran



So the four distributions, I basically look at it as a mixture. There is p hit, p short, p max, and p rand, right. So p max is when I actually miss the reading altogether. p short is when I hit on unexpected obstacles, p rand is I make a random error, and p hit is when I am actually measuring incorrectly, but I have a small measurement error.

And each of these is weighted by a corresponding z value here, z hit, z short, z max and z rand. And the condition is they are all positive and they sum to 1. So there p is a probability distribution. And so, this is for the individual ray z_t^k . That is the one ray, k . I will have to do this for all k . And so for the set of readings that I am going to get, since they are all independent, I am going to compute the probability for each measurement, each k , and then take the product.

So the, the final value q , that I return is a product of all the probabilities computed from each one of the 1 to k sensors that I have in my range sensor. So that is basically the probability of z_t given x_t comma m . So likewise, we do this kind of computation for various kinds of sensor models. And I mean, the book talks about a couple more. And if you want to get other flavors, you can read the book, so you can just look at the book.

But as far as I am concerned, I will be happy if you understand the range finder measurement model thoroughly and because others are all simple expansions of this. Thanks.