



Autonomous Robotics

Spring 2026

Abhishek Gupta, Siddhartha Srinivasa

TAs: Helen Wang, Sidharth Talia, Rohan Baijal, Christopher Tan



Recap

Fundamental Problem: State is hidden

But all decision making depends on knowing state

Solution: Estimate **belief** over state

$$bel(x_t) = P(x_t | z_{1:t}, u_{1:t})$$

Belief is a probability of each possible state given history

Also called Posterior / Information state / State of knowledge

Represent belief? Parametric (Gaussian), Non-parametric (Histogram)

Bayes filter in a nutshell

Key Idea: Apply Markov to get a recursive update!

Step 0. Start with the belief at time step $t-1$

$$bel(x_{t-1})$$

Step 1: Prediction - push belief through dynamics given **action**

$$\overline{bel}(x_t) = \sum P(x_t | u_t, x_{t-1}) bel(x_{t-1})$$

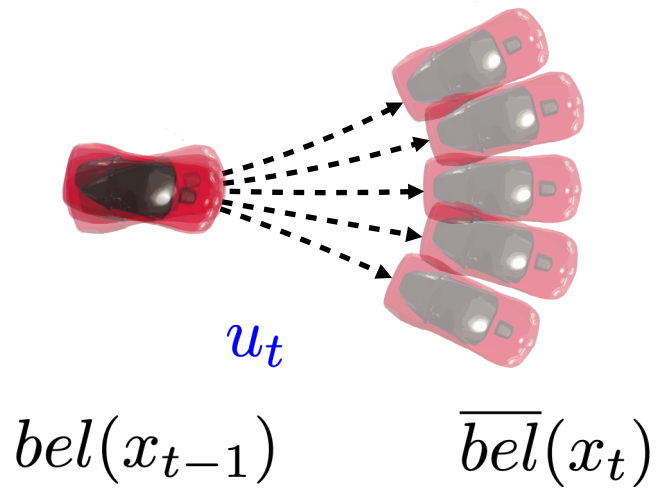
Step 2: Correction - apply Bayes rule given **measurement**

$$bel(x_t) = \eta P(z_t | x_t) \overline{bel}(x_t)$$

Bayes filter in a nutshell

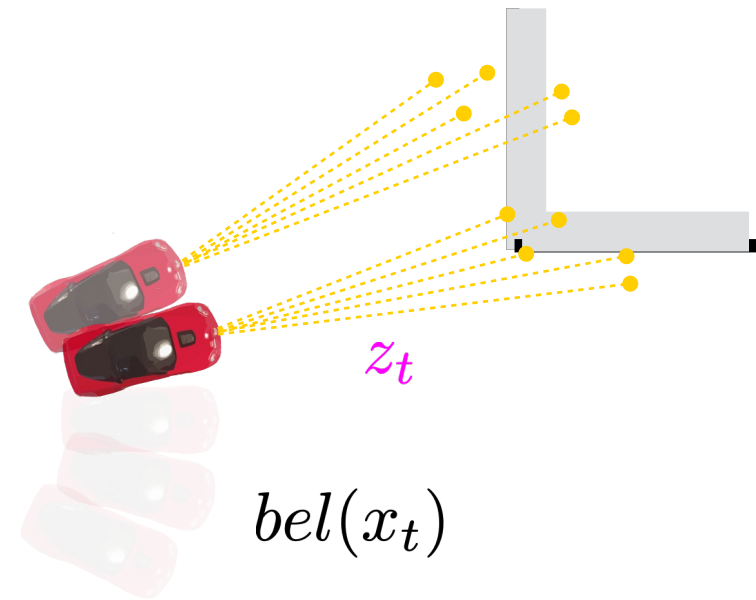
Step 1: Prediction - push belief through dynamics given **action**

$$\overline{bel}(x_t) = \sum P(x_t | u_t, x_{t-1}) bel(x_{t-1})$$



Step 2: Correction - apply Bayes rule given **measurement**

$$bel(x_t) = \eta P(z_t | x_t) \overline{bel}(x_t)$$

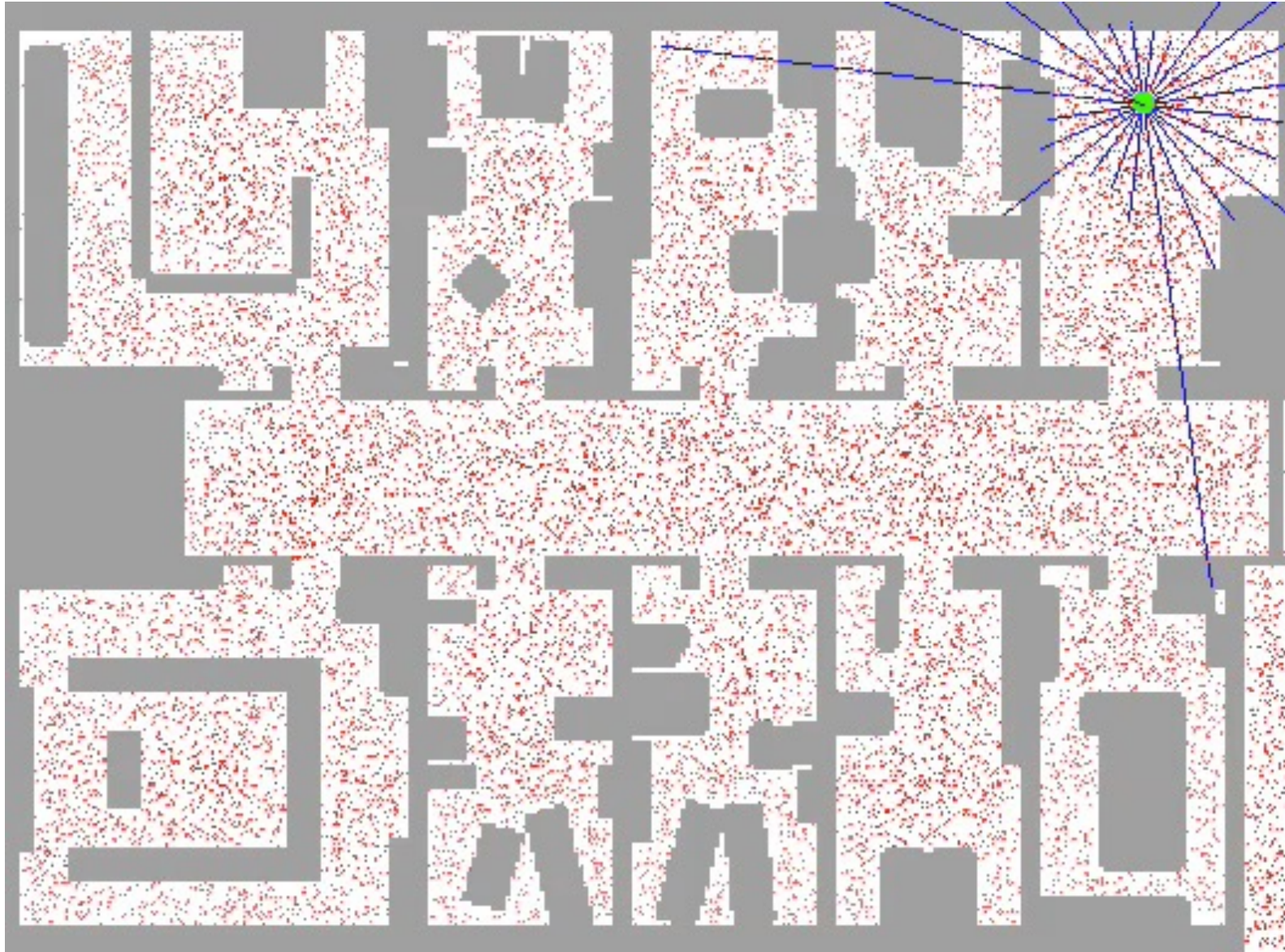


How does this connect back to our racecar?



Where am I in the world?

What does localization look like?



Lecture Outline

Recap



Motion Models



Observation Models

So what do we need to define to instantiate this?

Key Idea: Apply Markov to get a recursive update!

Step 0. Start with the belief at time step $t-1$

$$bel(x_{t-1})$$

Step 1: Prediction - push belief through dynamics given **action**

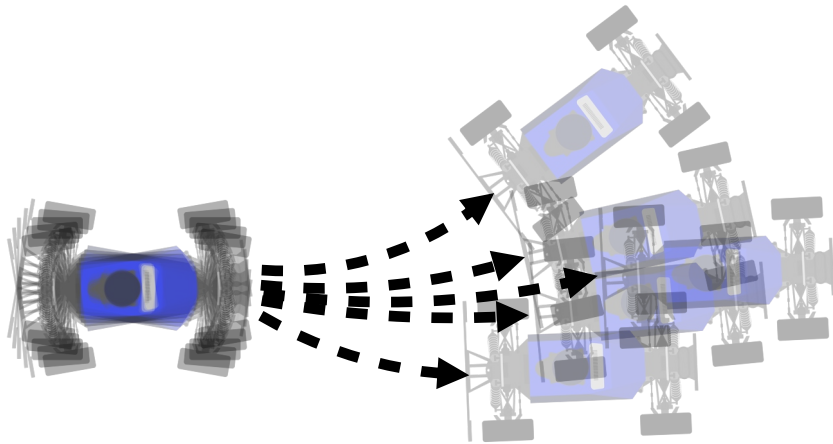
$$\bar{bel}(x_t) = \sum P(x_t | u_t, x_{t-1}) bel(x_{t-1})$$

Step 2: Correction - apply Bayes rule given **measurement**

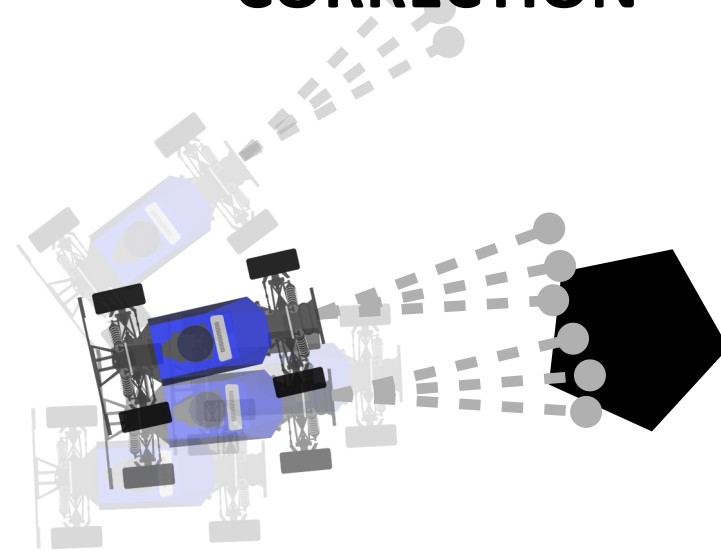
$$bel(x_t) = \eta P(z_t | x_t) \bar{bel}(x_t)$$

Let's ground this in the context of the car

PREDICTION



CORRECTION



PREDICTION

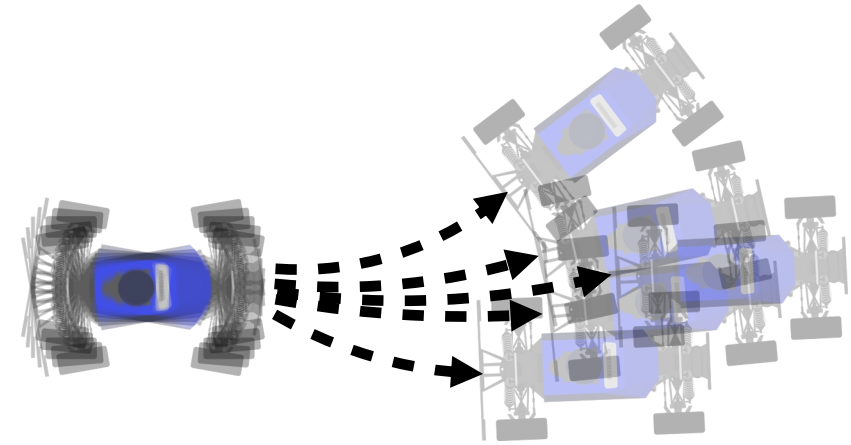
$$P(x_t | u_t, x_{t-1})$$

CORRECTION

$$P(z_t | x_t)$$

Motion Model

How do we know this?
→ it's just physics!

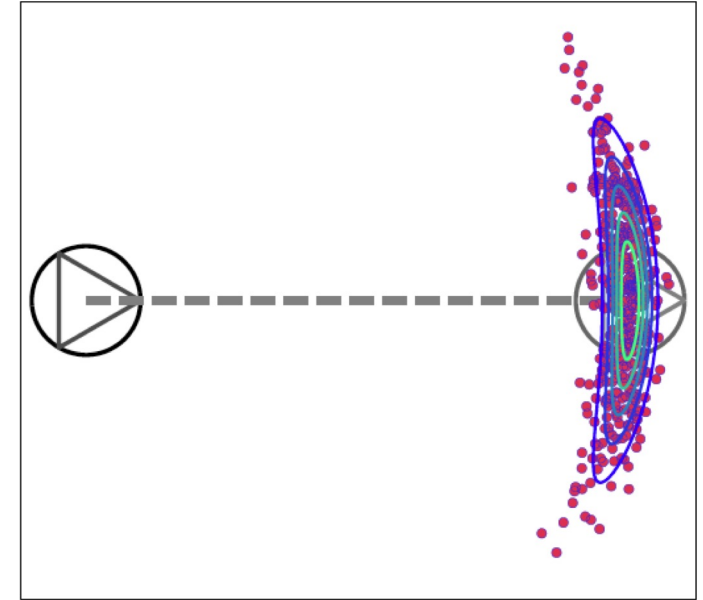


$$P(x_t | u_t, x_{t-1})$$

A Spectrum of Motion Models



VS



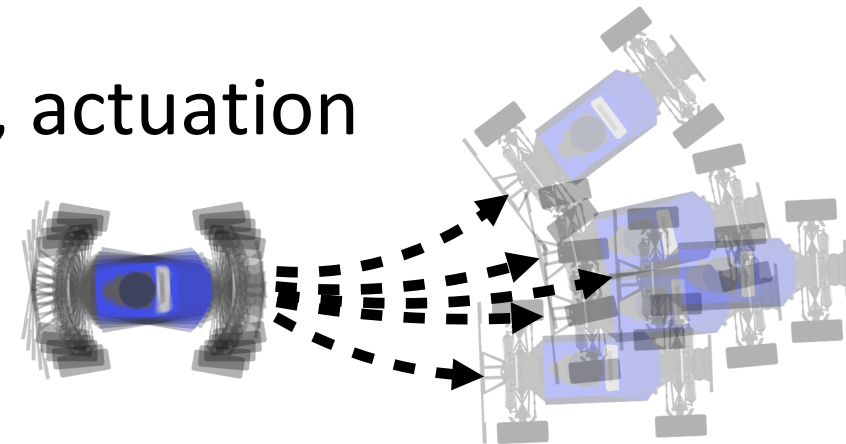
Highest-fidelity models
capturing everything
we know

([Red Bull F1 Simulator](#))

Simple model
with lots of noise

Why is the motion model probabilistic?

- If we know how to write out equations of motion, shouldn't we be able to predict exactly where an object ends up?
- “All models are wrong, but some are useful” — George Box
 - Examples: ideal gas law, Coulomb friction
- Stochasticity is a catch-all for model error, actuation error, ...



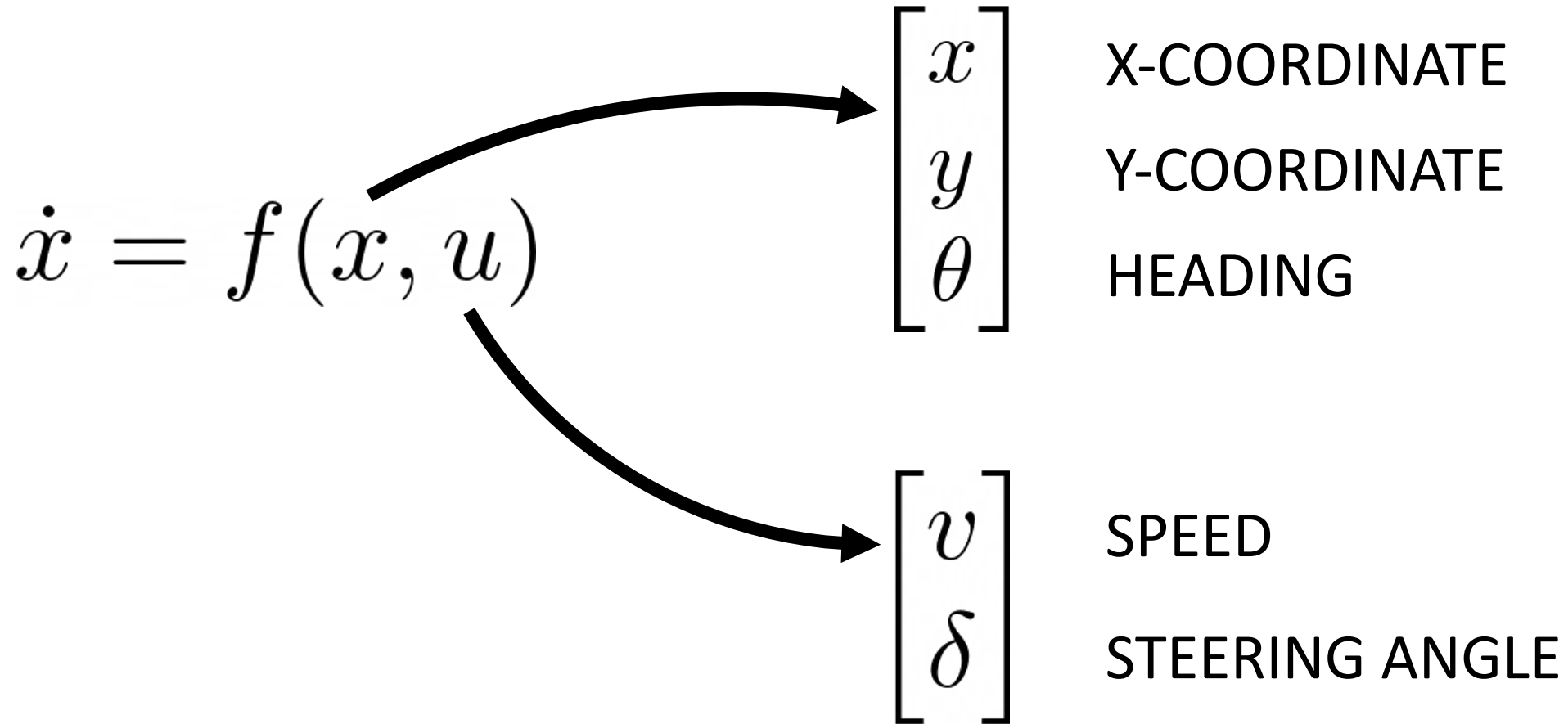
What defines a good motion model?

- In theory: try to accurately model the uncertainty (e.g., actuation errors)
- In practice...
 - We need just enough stochasticity to **explain any measurements** we'll see
(Bayes filter uses measurements to hone in on the right state)
 - We need a model that can deal with **unknown unknowns**
(No matter the model, we need to overestimate uncertainty)
 - We would like a model that is **computationally cheap**
(Bayes filter repeatedly invokes this model to predict state after actions)
- Key idea: simple model + stochasticity

What motion model should I use for MuSHR?

- A **kinematic model** governs how wheel speeds map to robot velocities
- A **dynamic model** governs how wheel torques map to robot accelerations
- For MuSHR, we'll ignore dynamics and focus on kinematics (assuming the wheel actuators can set speed directly)
- Other assumptions: wheels roll on hard, flat, horizontal ground without slipping

Kinematic Car Model

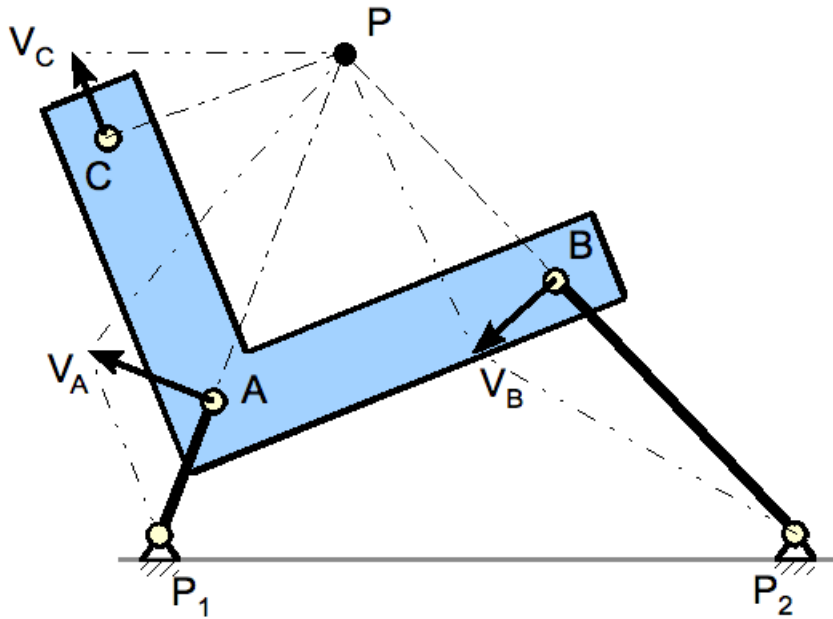


Kinematic Car Model

$$\dot{x} = f(x, u) \quad \xrightarrow{\text{INTEGRATE}} \quad \begin{bmatrix} x_{t-1} + \Delta x \\ y_{t-1} + \Delta y \\ \theta_{t-1} + \Delta \theta \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}$$

$$\xrightarrow{\text{ADD NOISE}} \quad P(x_t | u_t, x_{t-1})$$

Definition: Instant Center of Rotation (CoR)

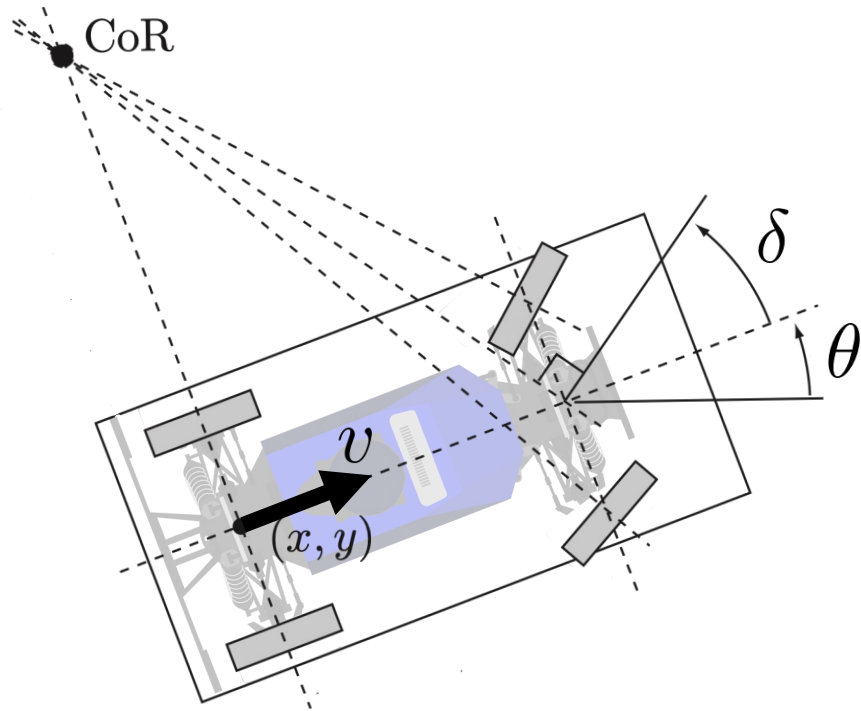


A planar **rigid body** undergoing a **rigid transformation** can be viewed as undergoing a **pure rotation** about an instant center of rotation.

rigid body: a non-deformable object

rigid transformation: a combined rotation and translation

Equations of Motion

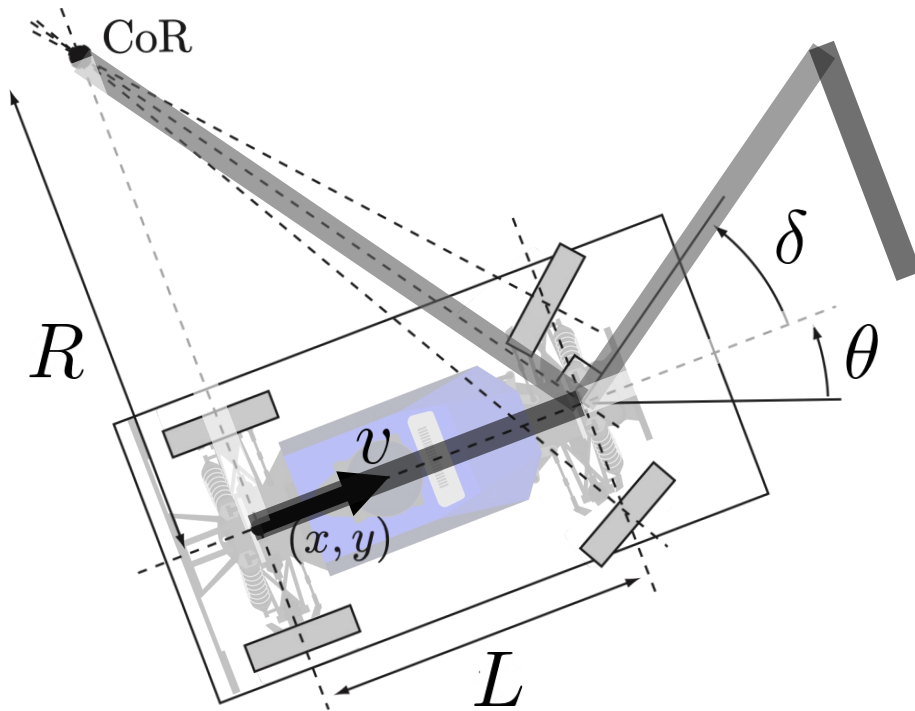


$$\dot{x} = v \cos \theta$$

$$\dot{y} = v \sin \theta$$

$$\dot{\theta} = \mathbf{?}$$

Equations of Motion



$$\dot{x} = v \cos \theta$$

$$\dot{y} = v \sin \theta$$

$$\dot{\theta} = \omega = \frac{v}{R} = \frac{v \tan \delta}{L}$$

$$\tan \delta = \frac{L}{R} \rightarrow R = \frac{L}{\tan \delta}$$

Kinematic Car Model

$$\dot{x} = f(x, u) \quad \xrightarrow{\text{INTEGRATE}} \quad \begin{bmatrix} x_{t-1} + \Delta x \\ y_{t-1} + \Delta y \\ \theta_{t-1} + \Delta \theta \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}$$

Integrate the Kinematics Numerically

$$\begin{aligned}\dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= \frac{v}{L} \tan \delta\end{aligned}$$

Assume that steering angle is **piecewise constant** between t and t'

Integrate the Kinematics Numerically

$$\begin{aligned}\Delta x &= \int_t^{t'} v \cos \theta(t) dt = \int_t^{t'} \frac{v \cos \theta}{\dot{\theta}} \frac{d\theta}{dt} dt = \frac{v}{\dot{\theta}} \int_{\theta}^{\theta'} \cos \theta d\theta \\ &= \frac{L}{\tan \delta} (\sin \theta' - \sin \theta)\end{aligned}$$

$$\Delta y = \frac{L}{\tan \delta} (\cos \theta - \cos \theta')$$

$$\Delta \theta = \int_t^{t'} \dot{\theta} dt = \frac{v}{L} \tan \delta \Delta t$$

$$\begin{aligned}\dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= \frac{v}{L} \tan \delta\end{aligned}$$

Assume that steering angle is **piecewise constant** between t and t'

Kinematic Car Update

$$\theta_t = \theta_{t-1} + \Delta\theta = \theta_{t-1} + \frac{v}{L} \tan \delta \Delta t$$

$$x_t = x_{t-1} + \Delta x = x_{t-1} + \frac{L}{\tan \delta} (\sin \theta_t - \sin \theta_{t-1})$$

$$y_t = y_{t-1} + \Delta y = y_{t-1} + \frac{L}{\tan \delta} (\cos \theta_{t-1} - \cos \theta_t)$$

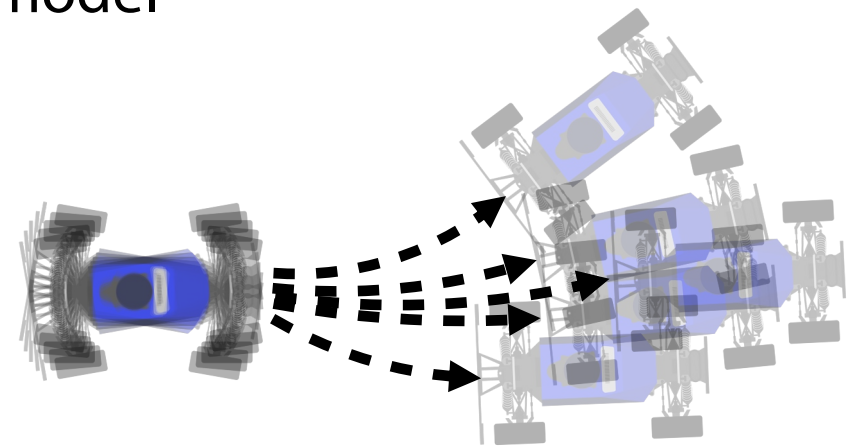
Kinematic Car Model

$$\dot{x} = f(x, u) \xrightarrow{\text{INTEGRATE}} \begin{bmatrix} x_{t-1} + \Delta x \\ y_{t-1} + \Delta y \\ \theta_{t-1} + \Delta \theta \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}$$

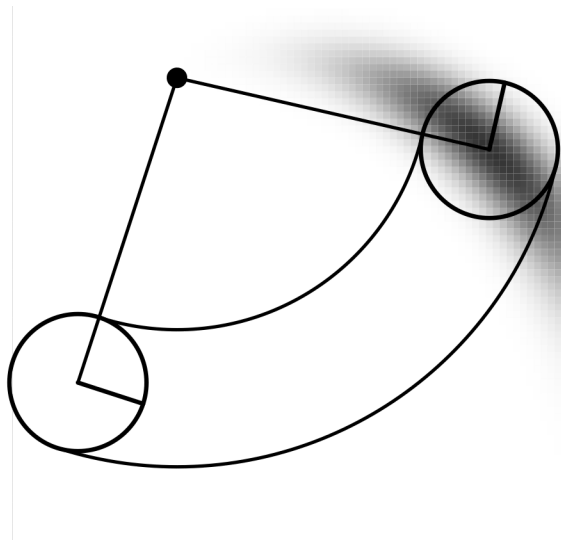
$$\xrightarrow{\text{ADD NOISE}} P(x_t | u_t, x_{t-1})$$

Why is the kinematic car model probabilistic?

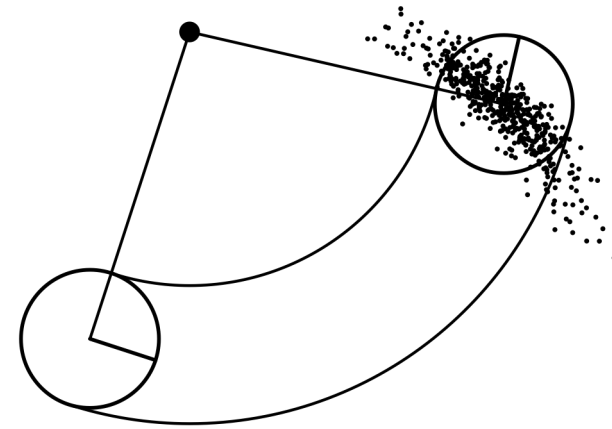
- Control signal error: voltage discretization, communication lag
- Unmodeled physics parameters: friction of carpet, tire pressure
- Incorrect physics: ignoring tire deformation, ignoring wheel slippage
- Our probabilistic motion model
 - Add noise to control before propagating through model
 - Add noise to state after propagating through model



Motion Model Summary



MOTION MODEL
PROB. DENSITY FUNCTION



MOTION MODEL
SAMPLES

- Write down the deterministic equations of motion (kinematic car model)
- Introduce stochasticity to account against various factors

Lecture Outline

Recap

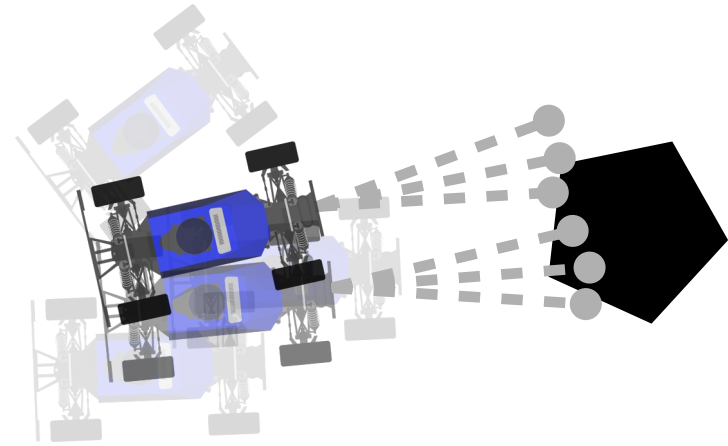


Motion Models



Observation Models

Sensor Model



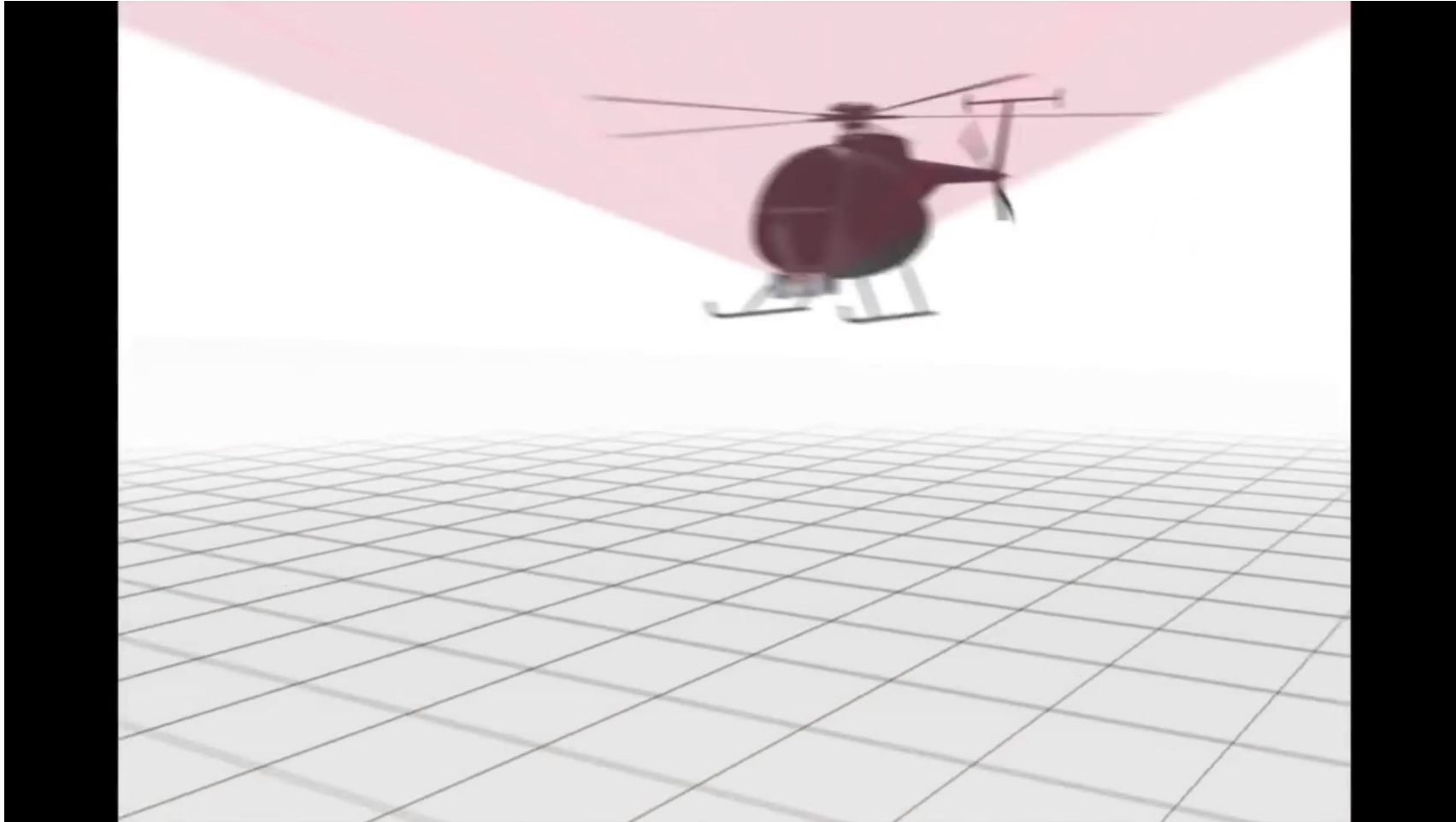
$$P(z_t | x_t)$$

How Does LIDAR Work?



[HTTPS://YOUTU.BE/NZKVF1CXE8S](https://youtu.be/NZKVF1CXE8S)

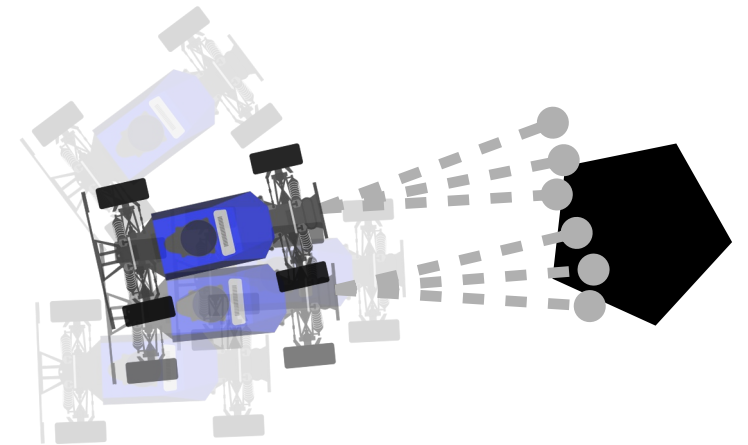
LIDAR in the Real World



[HTTPS://YOUTU.BE/I8YV5D8CPOC](https://youtu.be/I8YV5D8CPOC)

Why is the sensor model probabilistic?

- Incomplete/incorrect map: pedestrians, objects moving around
- Unmodeled physics: lasers go through glass
- Sensing assumptions: light interference from other sensors, multiple laser returns (bouncing off multiple objects)



What defines a good sensor model?

- Overconfidence can be catastrophic for Bayes filter
- LIDAR is very precise, but has distinct modes of failure
 - Anticipate specific types of failures, and add stochasticity accordingly

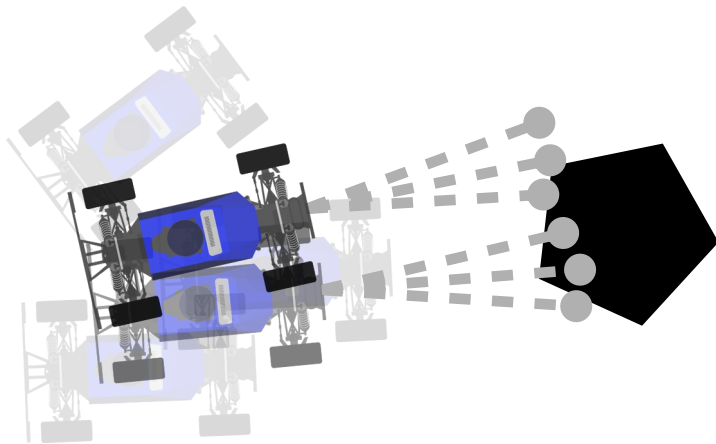
What sensor model should I use for MuSHR?

$$P(z_t | x_t) \rightarrow P(z_t | x_t, m)$$

LASER SCAN

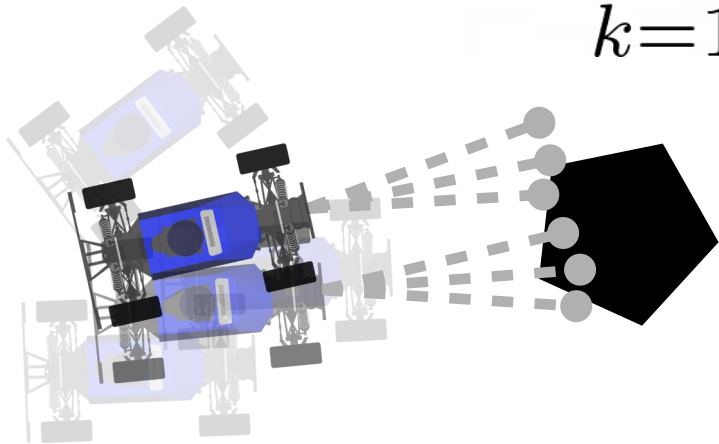
STATE

MAP



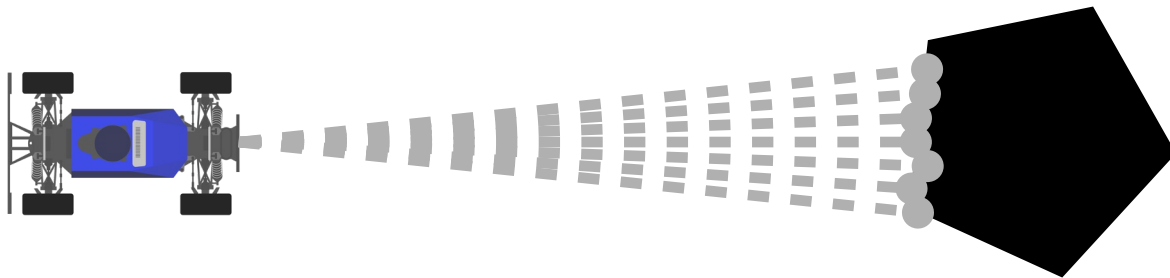
Assumption: Conditional Independence

$$\begin{aligned} P(z_t | x_t, m) &= P(z_t^1, z_t^2, \dots, z_t^K | x_t, m) \\ &= \prod_{k=1}^K P(z_t^k | x_t, m) \end{aligned}$$



Assumption: Conditional Independence

$$\begin{aligned} P(z_t | x_t, m) &= P(z_t^1, z_t^2, \dots, z_t^K | x_t, m) \\ &= \prod_{k=1}^K P(z_t^k | x_t, m) \end{aligned}$$



Single Beam Sensor Model

$$P(z_t^k | x_t, m)$$

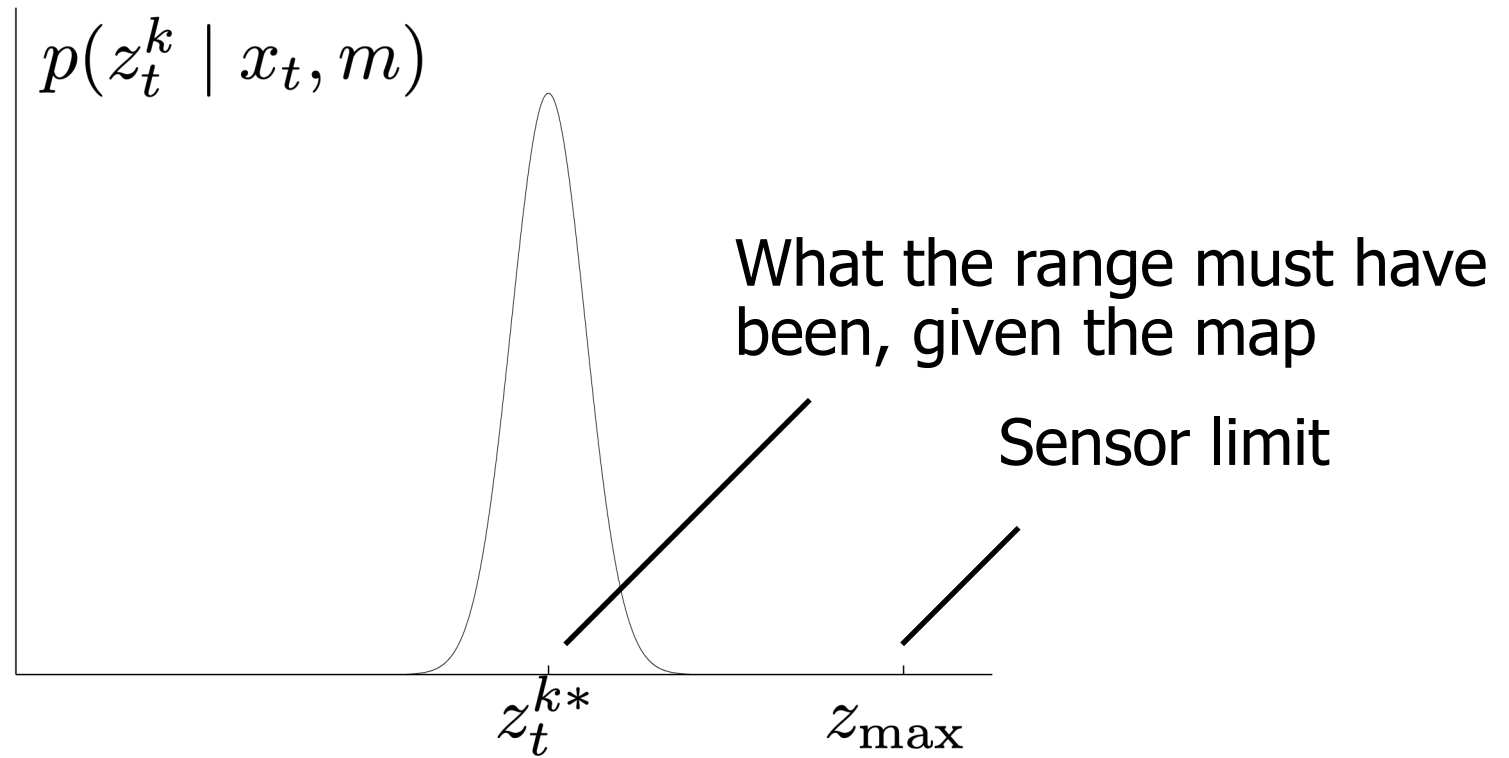
DISTANCE



Typical Sources of Stochasticity

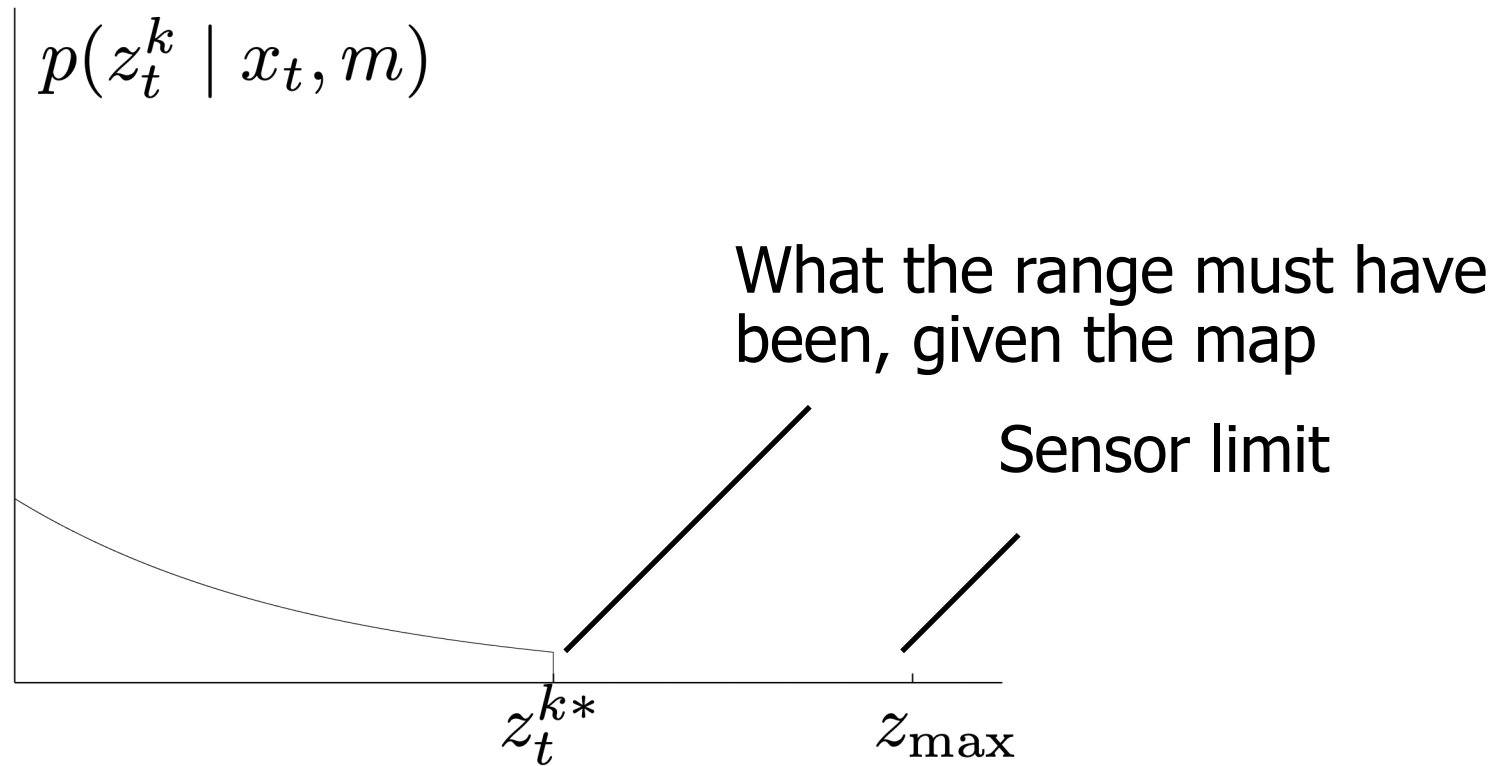
1. Correct range (distance) with local measurement noise
2. Unexpected objects
3. Sensor failures
4. Random measurements

Factor 1: Local Measurement Noise



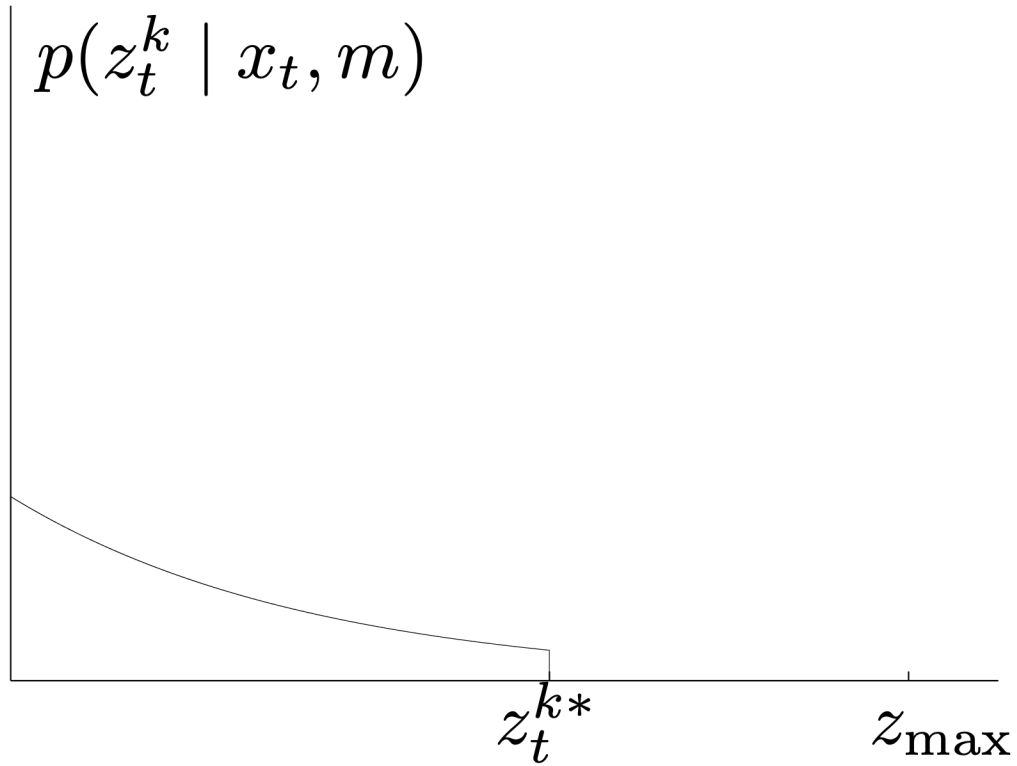
$$p_{\text{hit}}(z_t^k | x_t, m) = \begin{cases} \eta \mathcal{N}(z_t^k; z_t^{k*}, \sigma_{\text{hit}}^2) & \text{if } 0 \leq z_t^k \leq z_{\max} \\ 0 & \text{otherwise} \end{cases}$$

Factor 2: Unexpected Objects



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

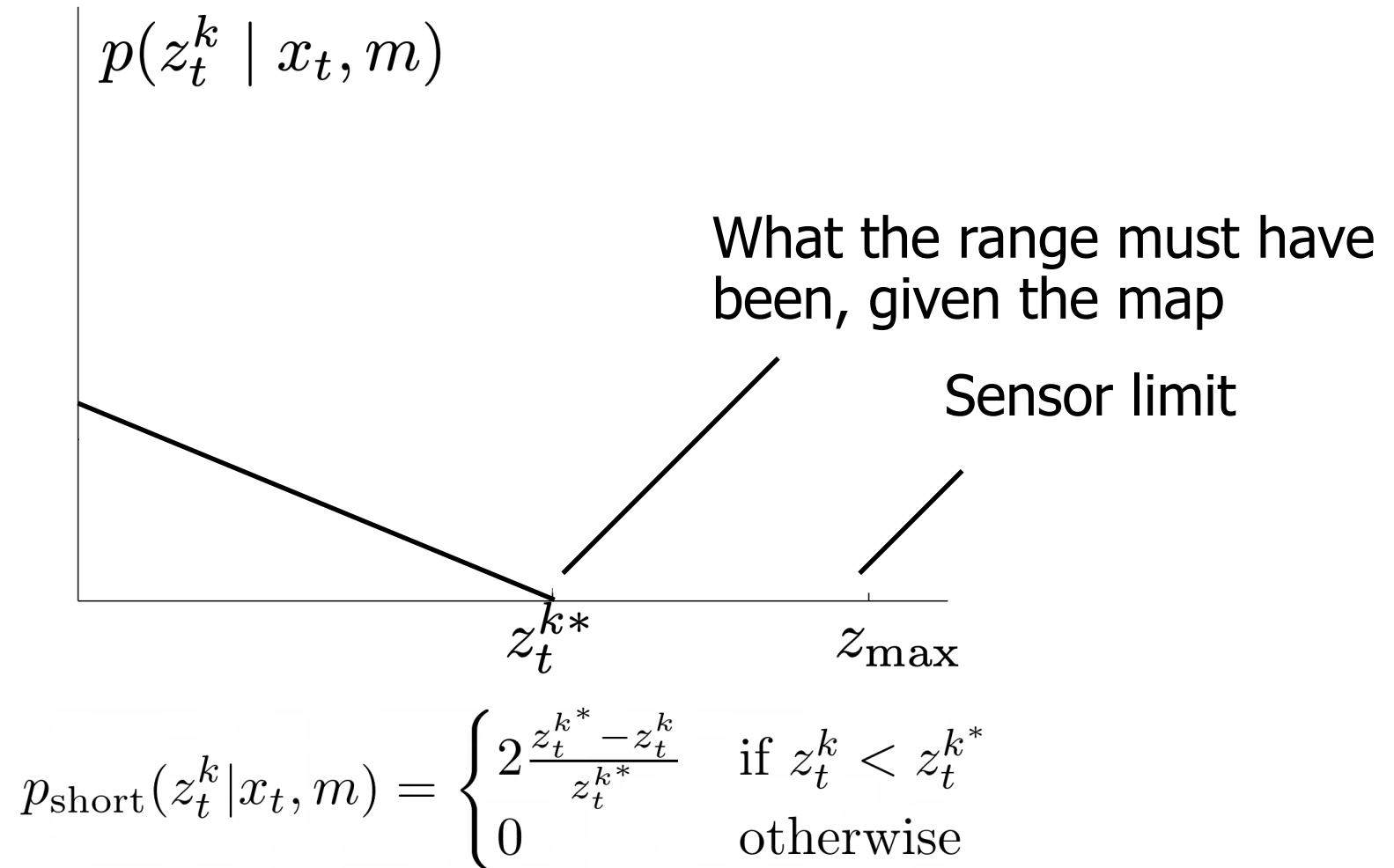
Factor 2: Unexpected Objects



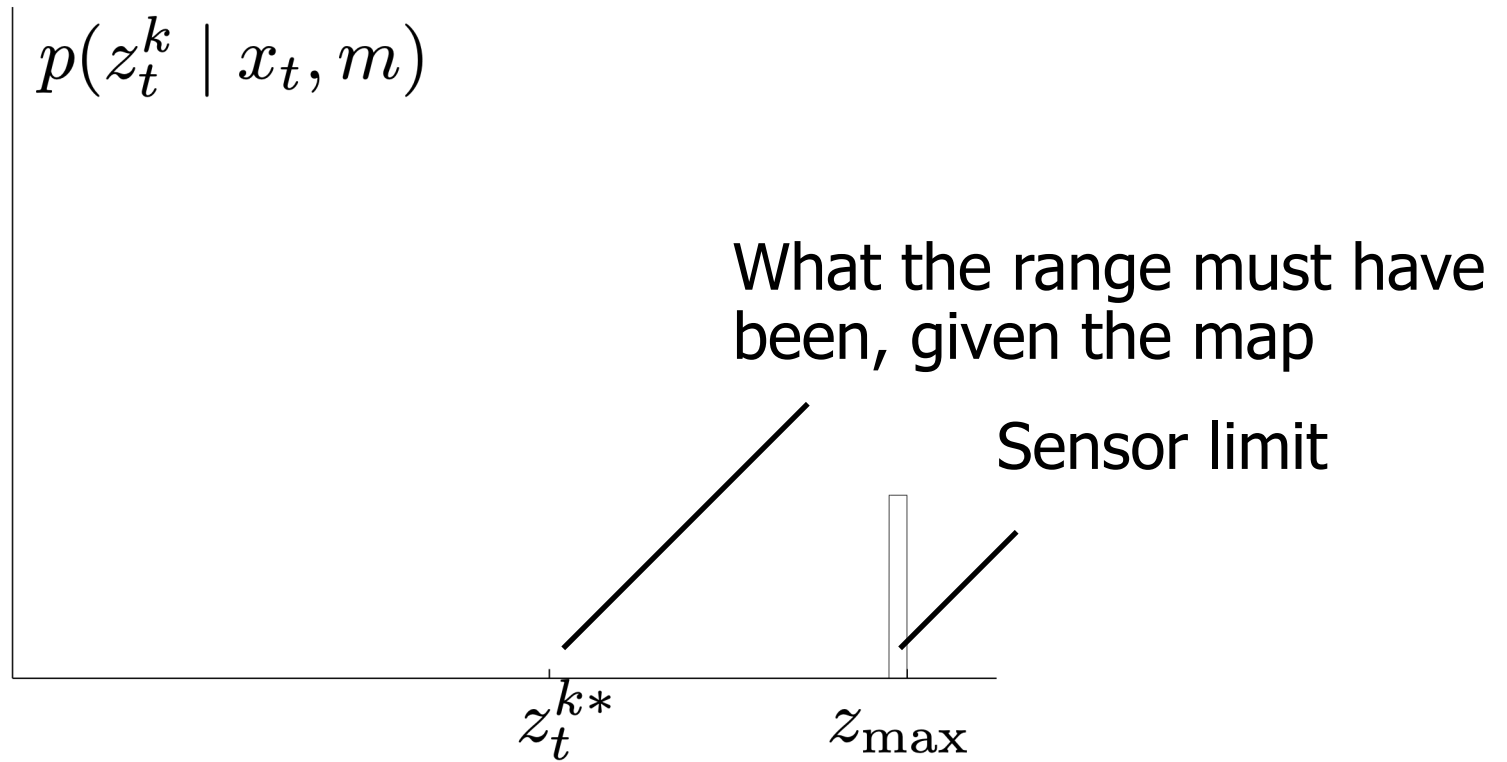
1								128
0	1							64
0	0	1						32
0	0	0	1					16
0	0	0	0	1				8
0	0	0	0	0	1			4
0	0	0	0	0	0	1		2
0	0	0	0	0	0	0	1	1

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

Factor 2: Unexpected Objects (Project)

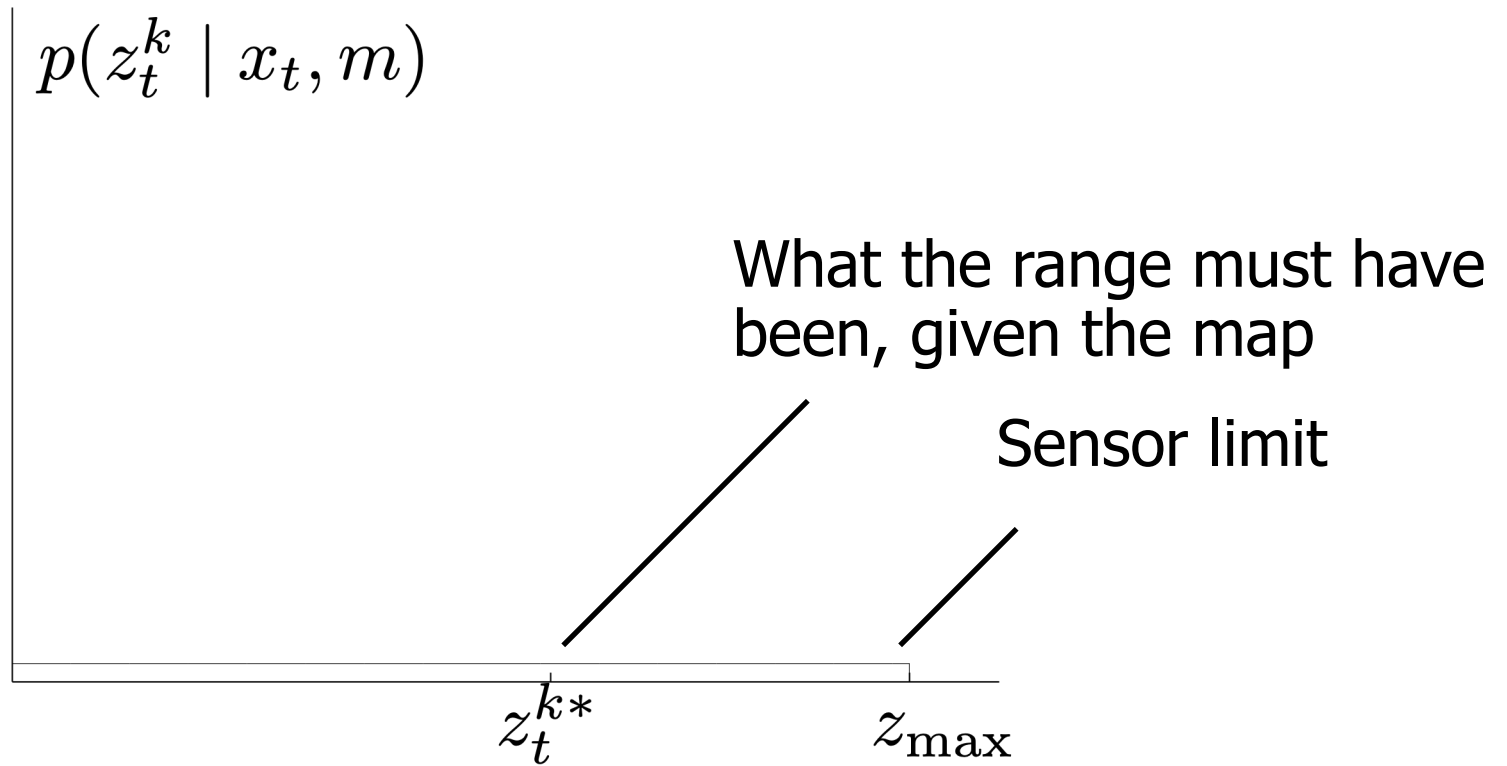


Factor 3: Sensor Failures



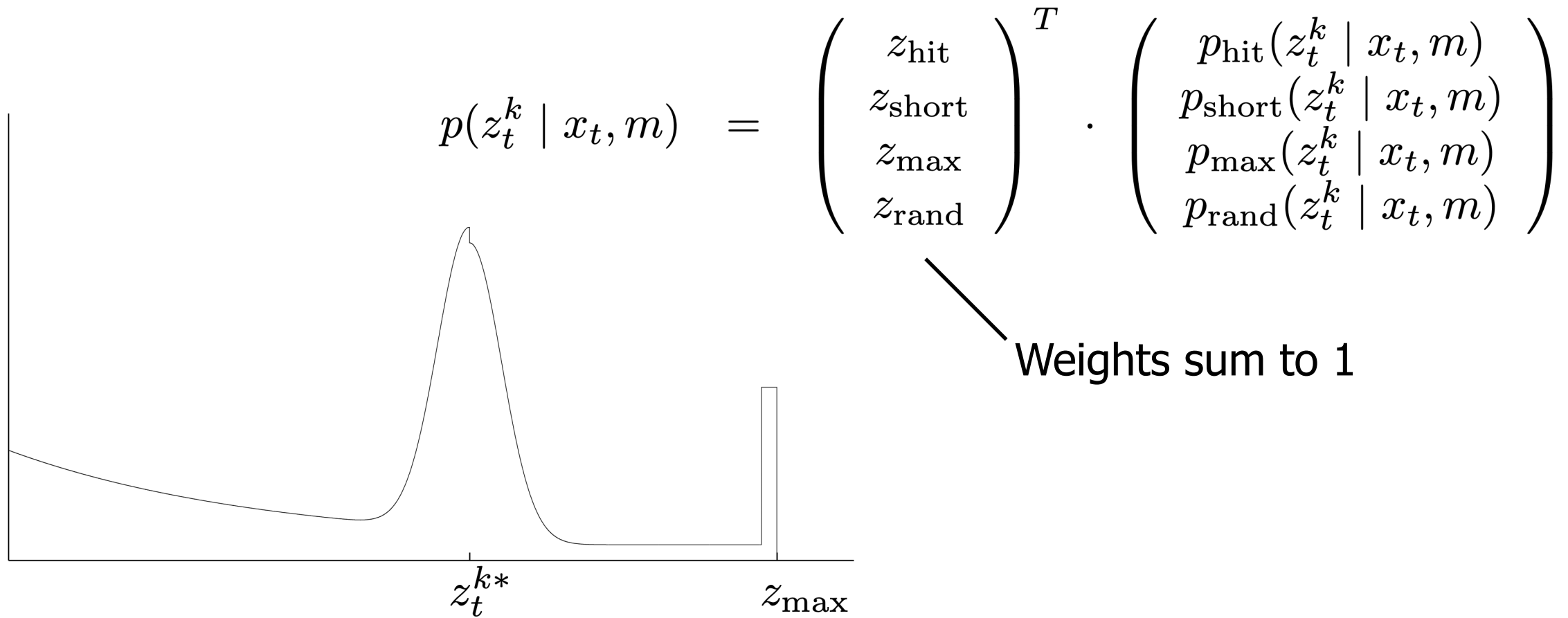
$$p_{\max}(z_t^k | x_t, m) = I(z = z_{\max}) = \begin{cases} 1 & \text{if } z = z_{\max} \\ 0 & \text{otherwise} \end{cases}$$

Factor 4: Random Measurements



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

Putting It All Together



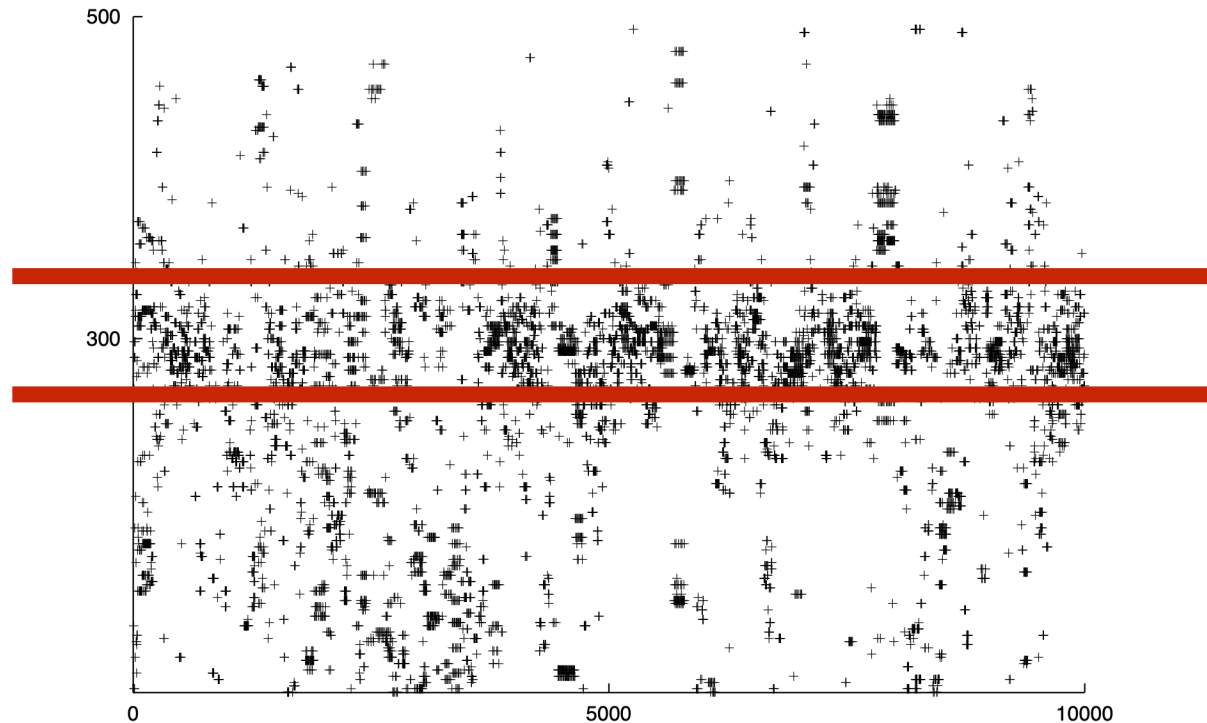
LIDAR Model Algorithm

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

1. Use robot **state** to compute the sensor's pose on the **map**
2. Ray-cast from the sensor to compute a simulated laser scan
3. For each beam, compare ray-casted distance to **real laser scan distance**
4. Multiply all probabilities to compute the likelihood of that real laser scan

Tuning Single Beam Parameters

- Offline: collect lots of data and optimize parameters

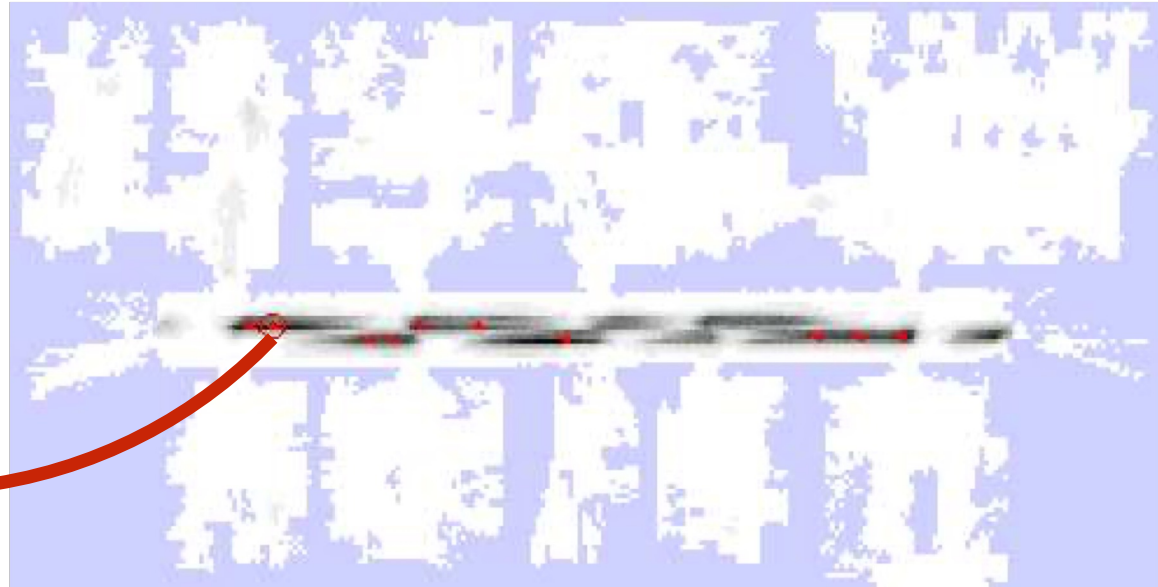


Tuning Single Beam Parameters

- Online: simulate a scan and plot the likelihood from different positions



Actual scan



Likelihood at various locations

Dealing with Overconfidence

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

- Subsample laser scans: convert 180 beams to 18 beams
- Force the single beam model to be less confident

$$P(z_t^k | x_t, m) \rightarrow P(z_t^k | x_t, m)^\alpha, \alpha < 1$$

MuSHR Localization Project

- Implement kinematic car motion model
- Implement different factors of single-beam sensor model
- Combine motion and sensor model with the Particle Filter algorithm

Lecture Outline

Recap



Motion Models



Observation Models