Normal Distribution

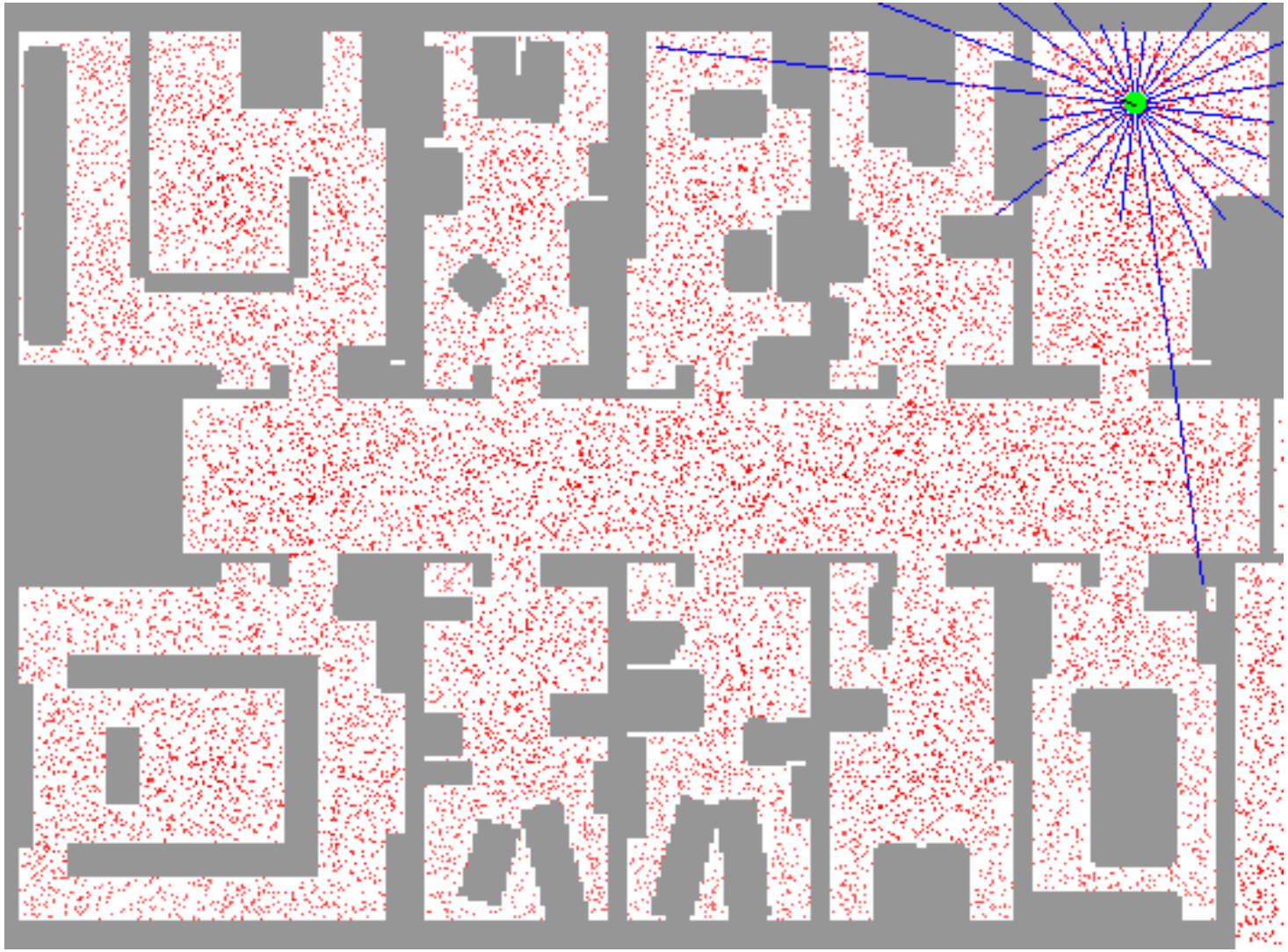
Robot Localization and Mapping
16-833

Michael Kaess

September 16, 2024

Slides courtesy of Ryan Eustice

Monte Carlo Localization



Monte Carlo Localization

To appear in *Proc. of the Sixteenth National Conference on Artificial Intelligence (AAAI-99), Orlando, Florida, 1999*

Monte Carlo Localization: Efficient Position Estimation for Mobile Robots

Dieter Fox, Wolfram Burgard†, Frank Dellaert, Sebastian Thrun

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

†Computer Science Department III
University of Bonn
Bonn, Germany

Abstract

This paper presents a new algorithm for mobile robot localization, called Monte Carlo Localization (MCL). MCL is a version of Markov localization, a family of probabilistic approaches that have recently been applied with great practical success. However, previous approaches were either computationally cumbersome (such as grid-based approaches that represent the state space by high-resolution 3D grids), or had to resort to extremely coarse-grained resolutions. Our approach is computationally efficient while retaining the ability to represent (almost) arbitrary distributions. MCL applies sampling-based methods for approximating probability distributions, in a way that places computation "where needed." The number of samples is adapted on-line, thereby invoking large sample sets only when necessary. Empirical results illustrate that MCL yields improved accuracy while requiring an order of magnitude less computation when compared to previous approaches. It is also much easier to implement.

Introduction

Throughout the last decade, sensor-based localization has been recognized as a key problem in mobile robotics (Cox 1991; Borenstein, Everett, & Feng 1996). Localization is a

While the majority of early work focused on the tracking problem, recently several researchers have developed what is now a highly successful family of approaches capable of solving both localization problems: *Markov localization* (Nourbakhsh, Powers, & Birchfield 1995; Simmons & Koenig 1995; Kaelbling, Cassandra, & Kurien 1996; Burgard *et al.* 1996). The central idea of Markov localization is to represent the robot's belief by a probability distribution over possible positions, and use Bayes rule and convolution to update the belief whenever the robot senses or moves. The idea of probabilistic state estimation goes back to Kalman filters (Gelb 1974; Smith, Self, & Cheeseman 1990), which use multivariate Gaussians to represent the robot's belief. Because of the restrictive nature of Gaussians (they can basically represent one hypothesis only annotated by its uncertainty) Kalman-filters usually are only applied to position tracking. Markov localization employs discrete, but *multi-modal* representations for representing the robot's belief, hence can solve the global localization problem. Because of the real-valued and multi-dimensional nature of kinematic state spaces these approaches can only *approximate* the belief, and accurate approximation usually requires prohibitive amounts of computation and memory.

In particular, *grid-based* methods have been devel-

AAAI Classic Paper Award 2017

IEEE ICRA Milestone Award 2020

Gaussian (Covariance Form)

- In this class, we'll (mostly) focus on Gaussians (normal distributions)
  - For both observations and our beliefs

$$p(\mathbf{x}) = \frac{1}{\sqrt{|2\pi\Sigma|}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu)}$$

- Characterized by mean & covariance

$$\mu_{\mathbf{x}} = E[\mathbf{x}]$$

$$\Sigma_{\mathbf{xx}} = E[(\mathbf{x} - E[\mathbf{x}])(\mathbf{x} - E[\mathbf{x}])^\top]$$

Gaussian (Covariance Form)

$$p(\mathbf{x}) = \frac{1}{\sqrt{|2\pi\Sigma|}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu)}$$

- All that gunk out front is just for normalization. Your mental model:

$$p(\mathbf{x}) = \alpha e^{-\frac{1}{2}(\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu)}$$

Gaussian (Information Form)

- An alternative parameterization of the Gaussian distribution

$$
\begin{aligned}
p(\boldsymbol{\xi}_t) &= \mathcal{N}(\boldsymbol{\xi}_t; \boldsymbol{\mu}_t, \Sigma_t) \\
&= \frac{1}{\sqrt{|2\pi\Sigma_t|}} \exp\left\{-\tfrac{1}{2}(\boldsymbol{\xi}_t - \boldsymbol{\mu}_t)^\top \Sigma_t^{-1}(\boldsymbol{\xi}_t - \boldsymbol{\mu}_t)\right\} \\
&= \frac{1}{\sqrt{|2\pi\Sigma_t|}} \exp\left\{-\tfrac{1}{2}(\boldsymbol{\xi}_t^\top \Sigma_t^{-1}\boldsymbol{\xi}_t - 2\boldsymbol{\mu}_t^\top \Sigma_t^{-1}\boldsymbol{\xi}_t + \boldsymbol{\mu}_t^\top \Sigma_t^{-1}\boldsymbol{\mu}_t)\right\} \\
&= \frac{e^{-\frac{1}{2}\boldsymbol{\mu}_t^\top \Sigma_t^{-1}\boldsymbol{\mu}_t}}{\sqrt{|2\pi\Sigma_t|}} \exp\left\{-\tfrac{1}{2}\boldsymbol{\xi}_t^\top \Sigma_t^{-1}\boldsymbol{\xi}_t + \boldsymbol{\mu}_t^\top \Sigma_t^{-1}\boldsymbol{\xi}_t\right\} \\
&= \frac{e^{-\frac{1}{2}\boldsymbol{\eta}_t^\top \Lambda_t^{-1}\boldsymbol{\eta}_t}}{\sqrt{|2\pi\Lambda_t^{-1}|}} \exp\left\{-\tfrac{1}{2}\boldsymbol{\xi}_t^\top \Lambda_t\boldsymbol{\xi}_t + \boldsymbol{\eta}_t^\top \boldsymbol{\xi}_t\right\} \\
&= \mathcal{N}^{-1}(\boldsymbol{\xi}_t; \boldsymbol{\eta}_t, \Lambda_t)
\end{aligned}
$$

Gaussian (Information Form)

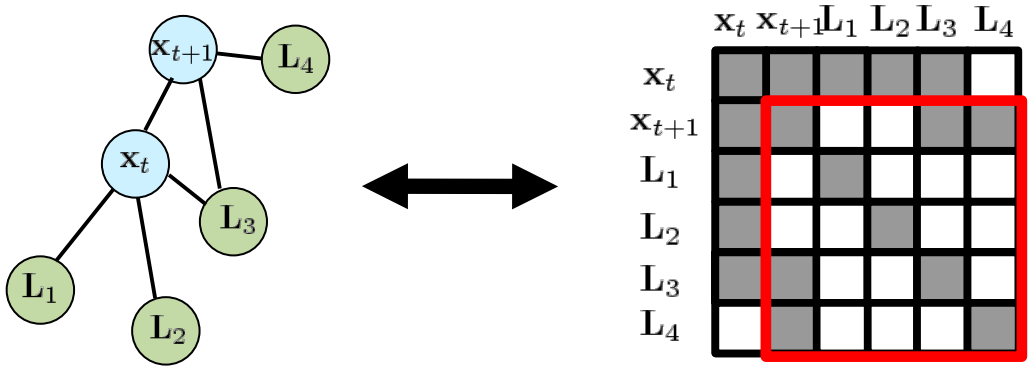
- Information matrix and vector

$$p(\boldsymbol{\xi}_t) = \mathcal{N}(\boldsymbol{\xi}_t; \boldsymbol{\mu}_t, \Sigma_t) = \mathcal{N}^{-1}(\boldsymbol{\xi}_t; \boldsymbol{\eta}_t, \Lambda_t)$$

$$\Lambda_t = \Sigma_t^{-1}$$

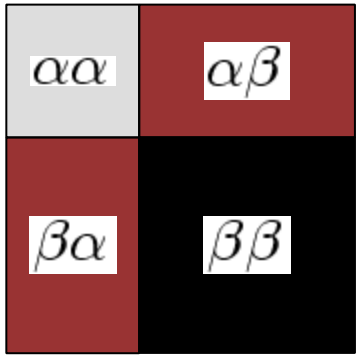
$$\boldsymbol{\eta}_t = \Lambda_t \boldsymbol{\mu}_t$$

- Encodes a graphical model
  - Markov Random Field (a.k.a. Markov Network)



Sparsity => Missing Edges => Available Conditional Independence

Gaussian Covariance & Information Parameterizations:



	Covariance Form	Information Form
Marginalization $p(\boldsymbol{\alpha}) = \int p(\boldsymbol{\alpha}, \boldsymbol{\beta}) d\boldsymbol{\beta}$	$\boldsymbol{\mu} = \boldsymbol{\mu}_\alpha$ $\Sigma = \Sigma_{\alpha\alpha}$ (sub-block)	$\boldsymbol{\eta} = \boldsymbol{\eta}_\alpha - \Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}\boldsymbol{\eta}_\beta$ $\Lambda = \Lambda_{\alpha\alpha} - \Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}\Lambda_{\beta\alpha}$ (Schur complement)
Conditioning $p(\boldsymbol{\alpha}\|\boldsymbol{\beta}) = \frac{p(\boldsymbol{\alpha}, \boldsymbol{\beta})}{p(\boldsymbol{\beta})}$	$\boldsymbol{\mu}' = \boldsymbol{\mu}_\alpha + \Sigma_{\alpha\beta}\Sigma_{\beta\beta}^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_\beta)$ $\Sigma' = \Sigma_{\alpha\alpha} - \Sigma_{\alpha\beta}\Sigma_{\beta\beta}^{-1}\Sigma_{\beta\alpha}$ (Schur complement)	$\boldsymbol{\eta}' = \boldsymbol{\eta}_\alpha - \Lambda_{\alpha\beta}\boldsymbol{\beta}$ $\Lambda' = \Lambda_{\alpha\alpha}$ (sub-block)

Visualizing Gaussians

- Recall our pdf:

$$p(\mathbf{x}) = \alpha e^{-\frac{1}{2}(\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu)}$$

- Find contours of constant probability

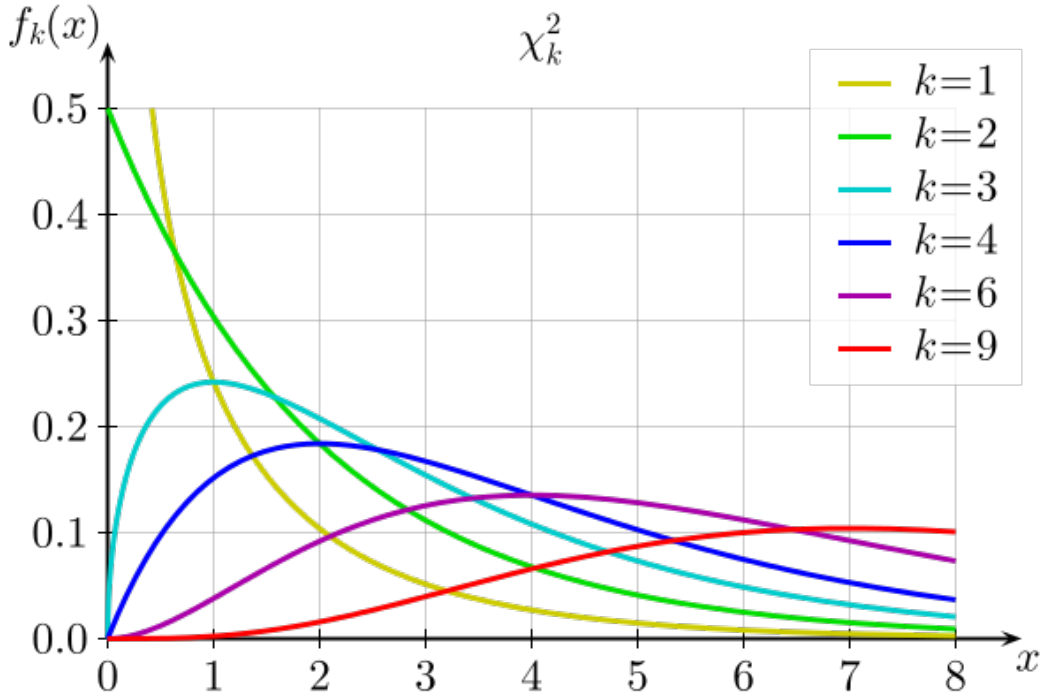
$$q^2 = (x-\mu)^\top \Sigma^{-1}(x-\mu) \qquad q = 1,2,3,\ldots$$

  - $q^2 \sim \chi_k^2$ is chi-squared distributed with k degrees of freedom
  - q (i.e., its square root) is known as the "Mahalanobis distance"

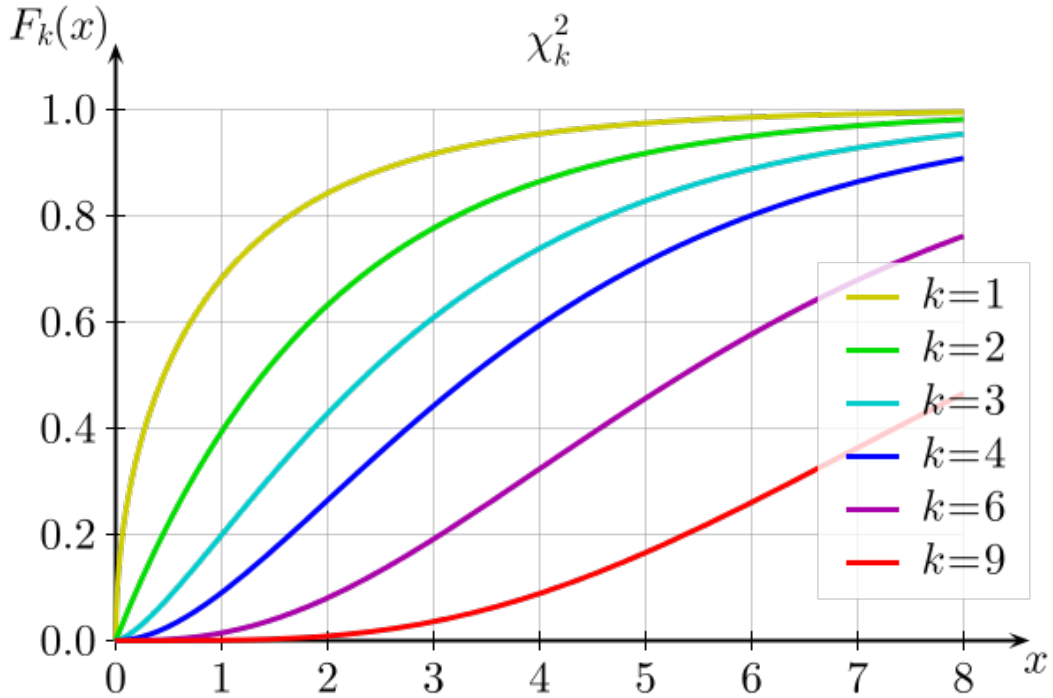
- Expand these terms, we end up with quadratic curve
  - An ellipse

Chi-Square Distribution

- Distribution of sum of squares of independent standard normal random variables



Probability density function [Wikipedia]



Cumulative distribution function

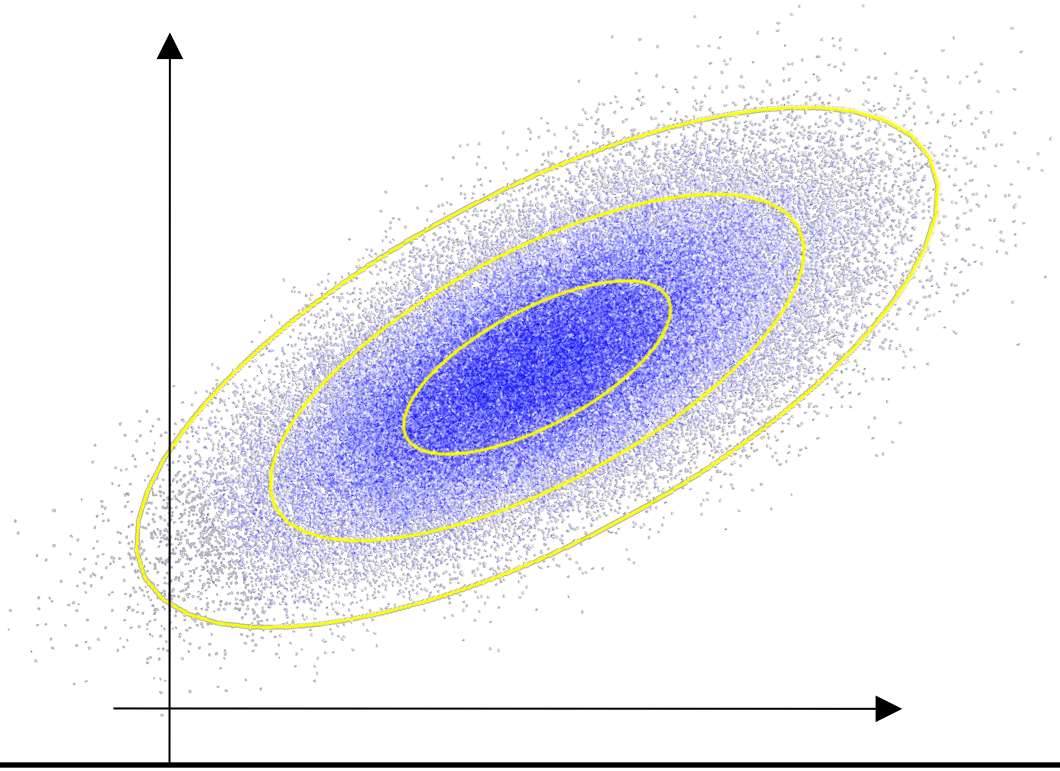
Visualizing Gaussians

- Number of particles within each ellipse can be computed based on properties of Gaussian distributions
  - Actually related to

χ_k^2 k

Sigma	1D	2D
1	0.6827	0.3935
2	0.9545	0.8647
3	0.9973	0.9889

n

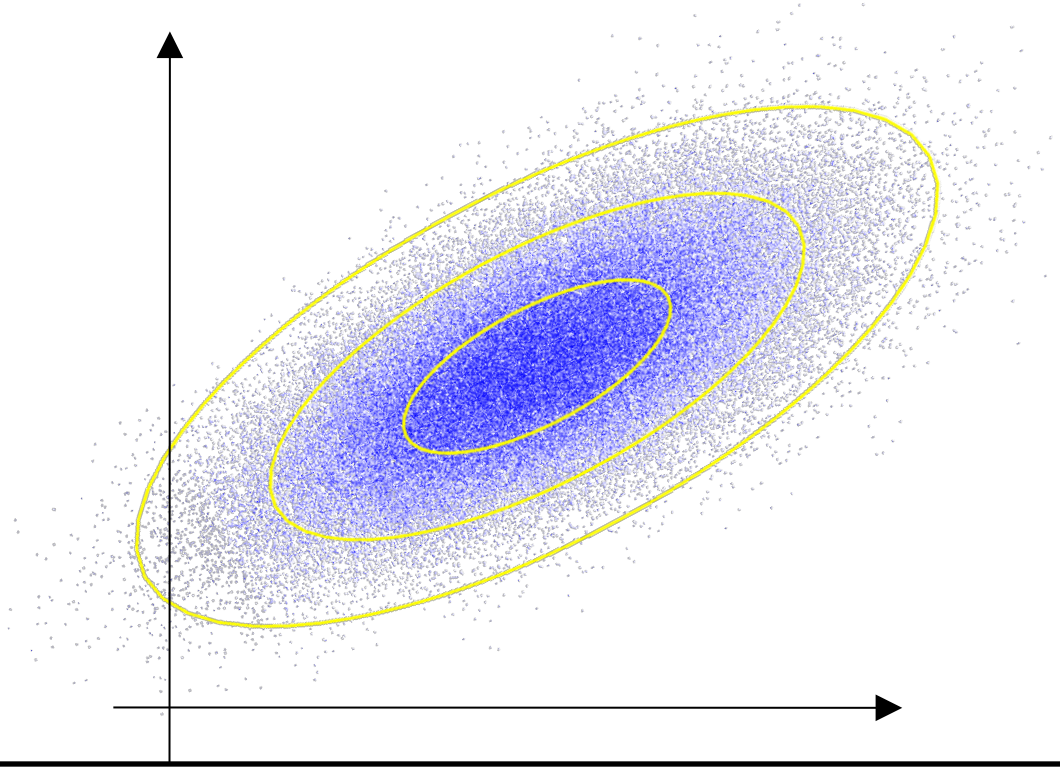


Visualizing Gaussians

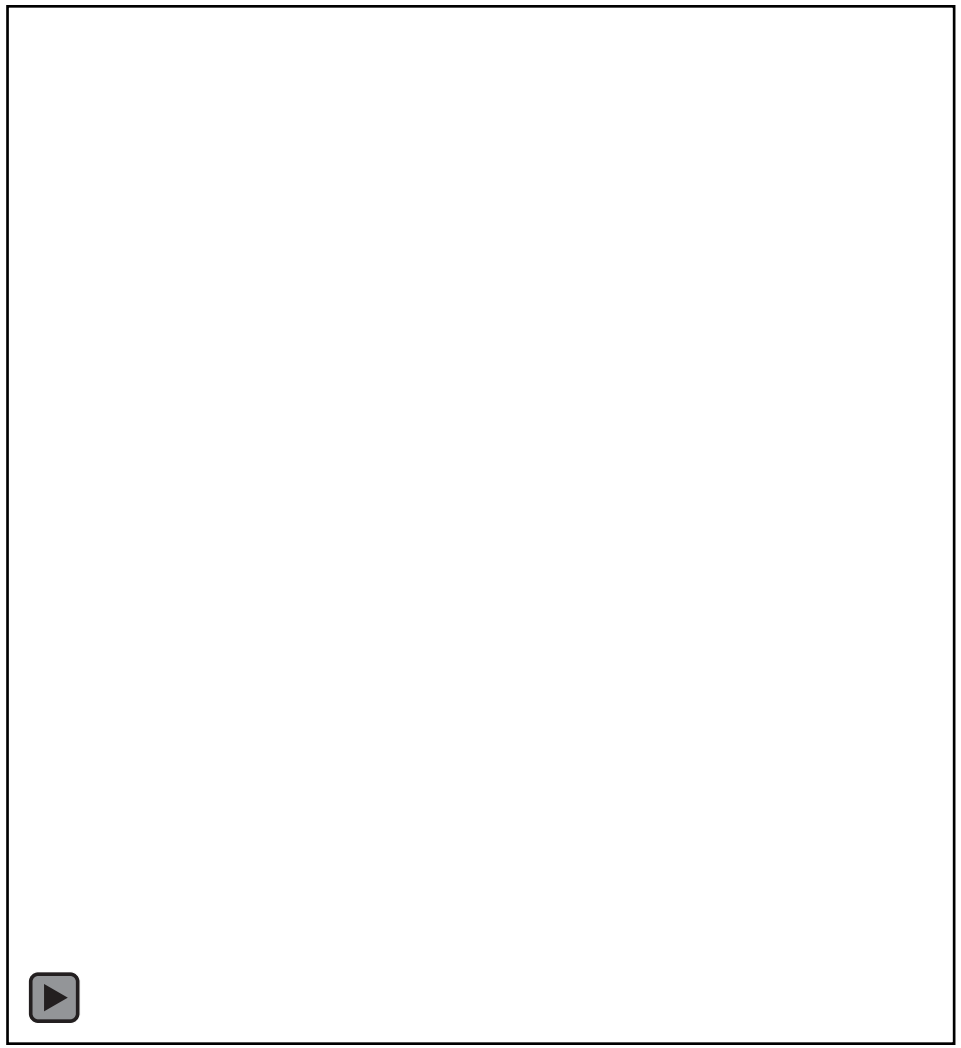
- Number of particles within each ellipse can be computed based on properties of Gaussian distributions
  - Actually related to

χ_k^2

n \ k	Sigma	1D	2D
	1	chi2cdf(1,1)	chi2cdf(1,2)
	2	chi2cdf(4,1)	chi2cdf(4,2)
	3	chi2cdf(9,1)	chi2cdf(9,2)



2-DOF Gaussian Correlation



2-DOF Gaussian Correlation

rho = 0

Sigma =

1 0

0 4

V =

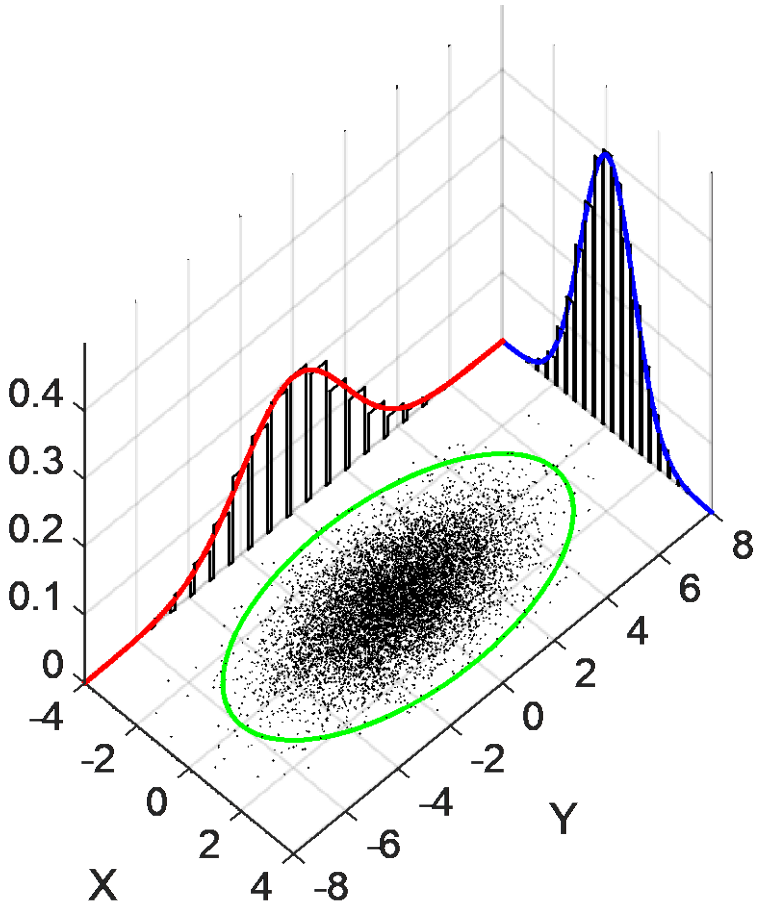
1 0

0 1

D =

1 0

0 4



2-DOF Gaussian Correlation

rho = -0.5000

Sigma =

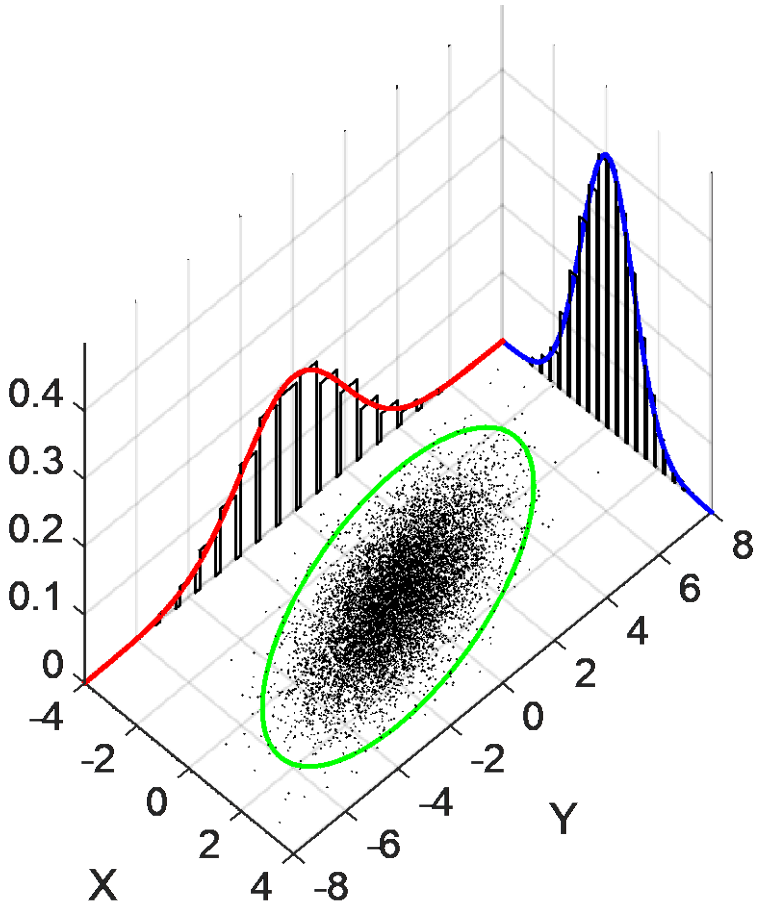
1	-1
-1	4

V =

-0.9571	-0.2898
-0.2898	0.9571

D =

0.6972	0
0	4.3028



2-DOF Gaussian Correlation

rho = 0.5000

Sigma =

1 1

1 4

V =

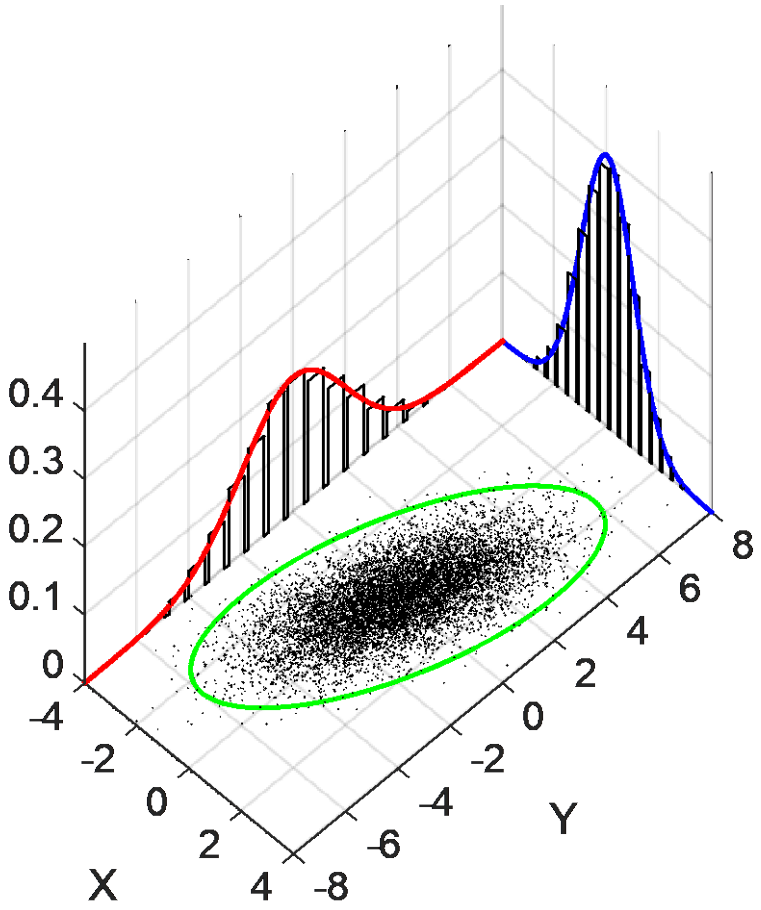
-0.9571 0.2898

0.2898 0.9571

D =

0.6972 0

0 4.3028



2-DOF Gaussian Correlation

rho = 1

Sigma =

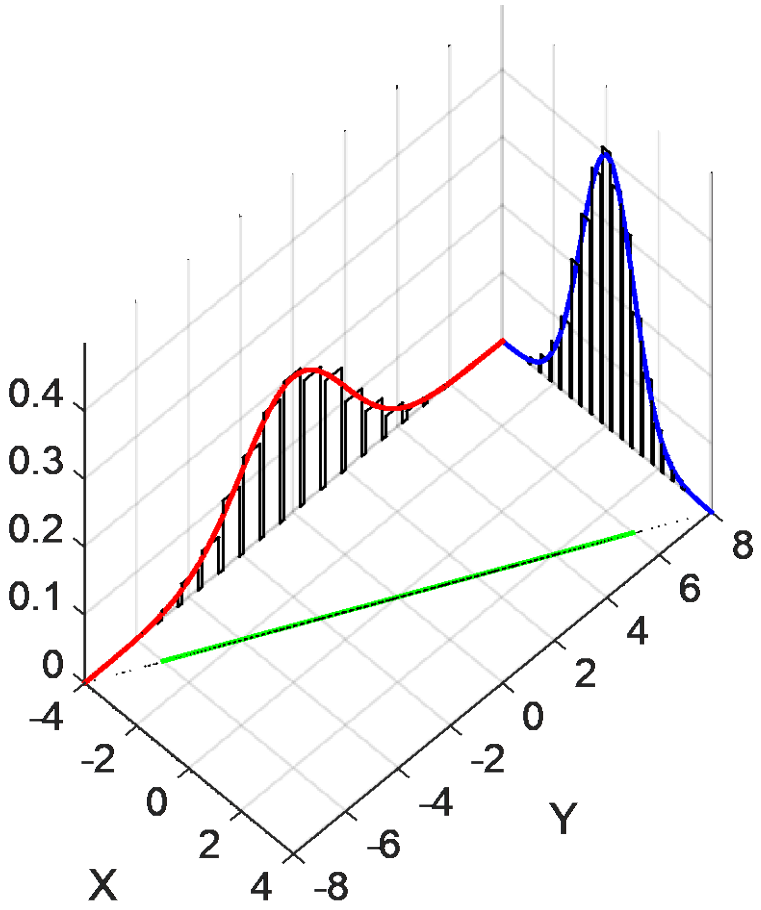
1	2
2	4

V =

-0.8944	0.4472
0.4472	0.8944

D =

0	0
0	5



Why use Gaussians?

- Convenience
  - Compact representation
  - Linear operations on Gaussians produce new Gaussians

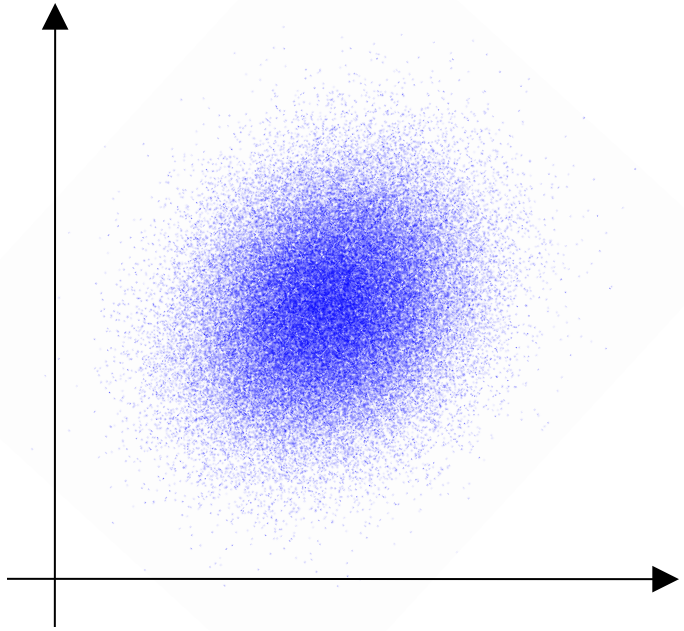
- Central Limit Theorem: Distribution of the sum (or average) of N independent and identically distributed (IID) random variables approaches a normal distribution.
  - Only minor restrictions on the distribution of the individual random variables

Implications of CLT

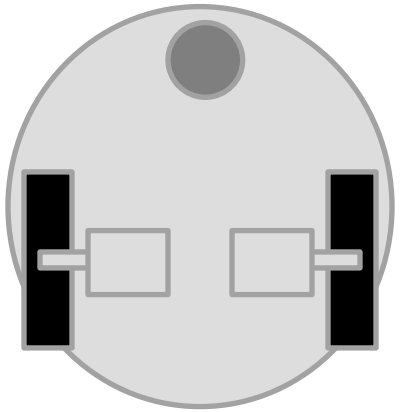
- We often estimate the state of something using many observations
  - Measuring the gravity on the moon by dropping a weight and timing the result
- Even if the distribution of each observation is non-Gaussian, their average will tend towards one.

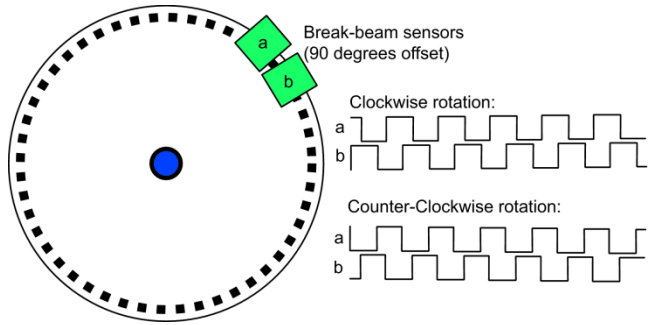
Estimating Uncertainty

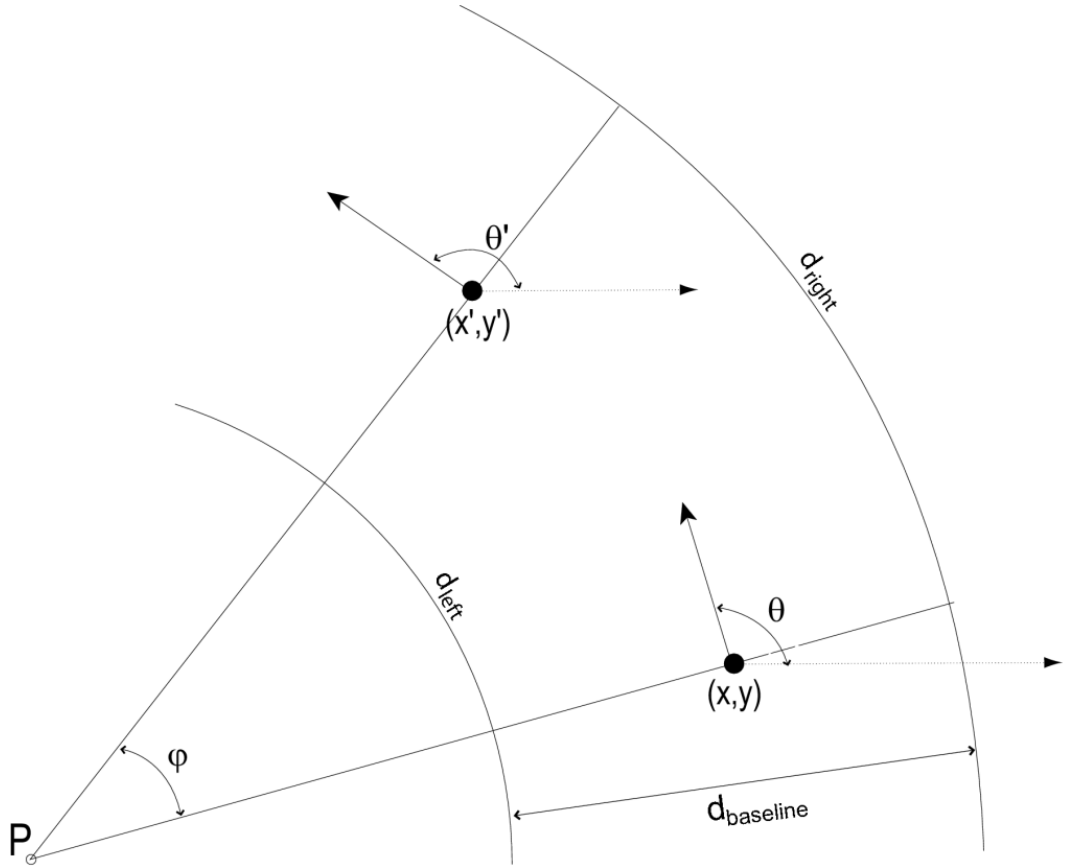
- Where do uncertainty estimates come from?
  - Empirically measure uncertainty
  - Manufacturer data sheets
  - Educated guesses
    - Validate with χ^2 error



Odometry Example





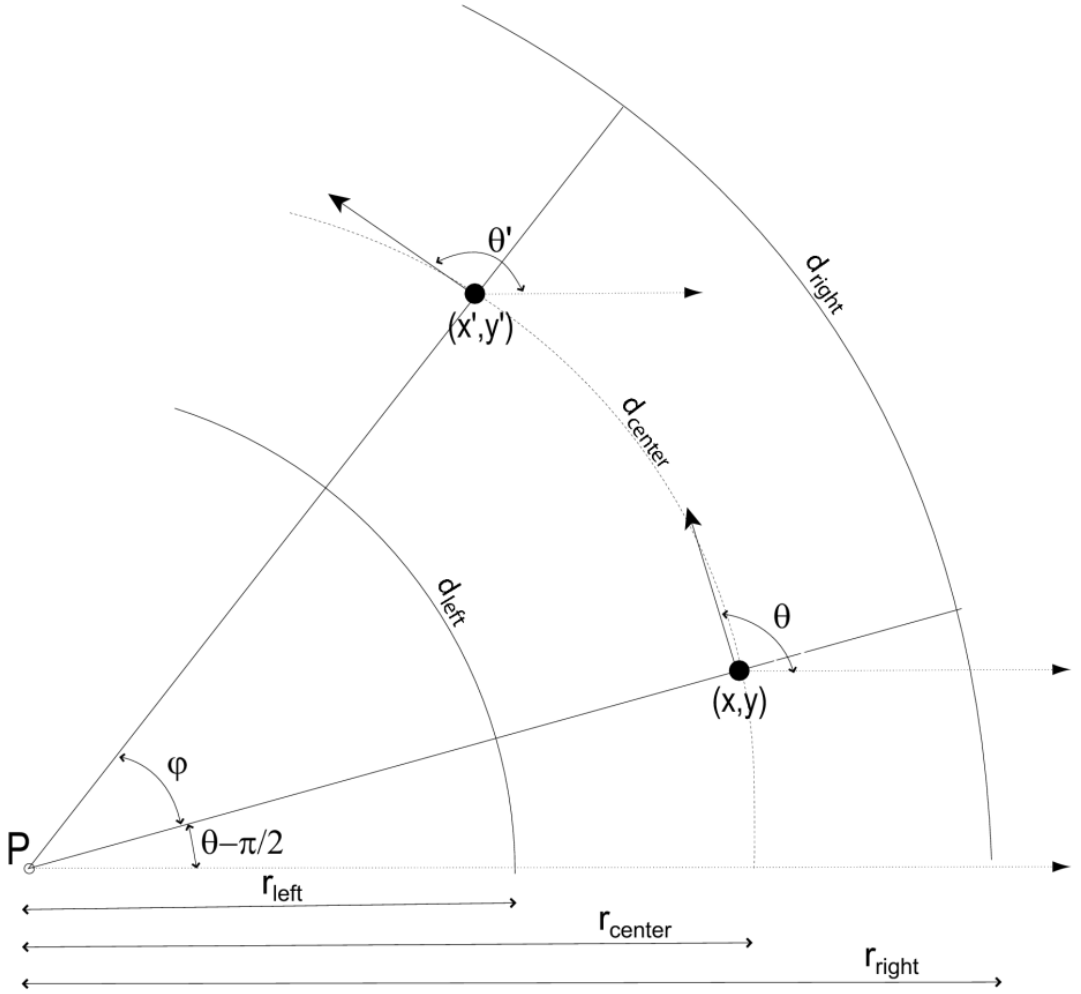


Odometry Example

- How to convert left/right ticks to a change in position?

$$\Delta x = \frac{d_R + d_L}{2}$$

$$\Delta \theta = \frac{d_R - d_L}{d_B}$$



Odometry Example

- Sensors observe:
  - Counts on left and right wheels

- No “noise” in those counts, however, there’s slippage. Model distance as:

$$d_R = \alpha c_R + w_1$$
$$d_L = \alpha c_L + w_2$$

- Noise w_1, w_2 are iid Gaussian:

$$w_1, w_2 \sim N(0, \sigma^2)$$

Odometry Example

- What is the uncertainty of $\Delta x,\ \Delta\theta$?
  - First, what's the uncertainty of d_R, d_L

$$\underbrace{\begin{bmatrix} d_R \\ d_L \end{bmatrix}}_{\boldsymbol{d}} = \underbrace{\begin{bmatrix} \alpha & 0 & 1 & 0 \\ 0 & \alpha & 0 & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} c_R \\ c_L \\ w_1 \\ w_2 \end{bmatrix}}_{\mathbf{w}}$$

$$d_R = \alpha c_R + w_1$$
$$d_L = \alpha c_L + w_2$$

$$\Delta x = \frac{d_R + d_L}{2}$$
$$\Delta\theta = \frac{d_R - d_L}{d_B}$$

$$\Sigma_d = A \Sigma_w A^T$$

But what's Σ_w ???

Odometry Example

$$\begin{bmatrix} d_R \\ d_L \end{bmatrix} = \begin{bmatrix} \alpha & 0 & 1 & 0 \\ 0 & \alpha & 0 & 1 \end{bmatrix} \begin{bmatrix} c_R \\ c_L \\ w_1 \\ w_2 \end{bmatrix}$$

$$\Sigma_d = A \Sigma_w A^T$$

But what's Σ_w ???

Remember, we said c_R, c_L were "error-free", and $w_1, w_2 \sim N(0, \sigma^2)$ (iid)

$$\Sigma_w = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2 & 0 \\ 0 & 0 & 0 & \sigma^2 \end{bmatrix}$$

Odometry Example

- We are half-way there now!

$$\Sigma_d = \begin{bmatrix} \alpha & 0 & 1 & 0 \\ 0 & \alpha & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2 & 0 \\ 0 & 0 & 0 & \sigma^2 \end{bmatrix} \begin{bmatrix} \alpha & 0 & 1 & 0 \\ 0 & \alpha & 0 & 1 \end{bmatrix}^T$$

$$= \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$$

- Does this make intuitive sense?
  - Answer is 2x2?
  - No alphas?

Odometry Example

- Where are we going again?
  - Trying to compute uncertainty of odometry measurements $\Delta x,\ \Delta\theta$
  - We know these in terms of : d_R, d_L

$$d_R = \alpha c_R + w_1$$
$$d_L = \alpha c_L + w_2$$

$$\Delta x = \frac{d_R + d_L}{2}$$
$$\Delta\theta = \frac{d_R - d_L}{d_B}$$

- We've gone from Σ_w to Σ_d
- Now, we need to go from Σ_d to Σ_x

Odometry Example

- Write $\boldsymbol{x}$ in terms of $\boldsymbol{d}$

$$\underbrace{\begin{bmatrix} \Delta x \\ \Delta\theta \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} 1/2 & 1/2 \\ 1/d_B & -1/d_B \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} d_R \\ d_L \end{bmatrix}}_{\mathbf{d}}$$

$$d_R = \alpha c_R + w_1$$
$$d_L = \alpha c_L + w_2$$

$$\Delta x = \frac{d_R + d_L}{2}$$
$$\Delta\theta = \frac{d_R - d_L}{d_B}$$

$$\Sigma_x = B\Sigma_d B^T$$

Odometry Example

- We're done!

$$\begin{aligned}\Sigma_x &= \begin{bmatrix} 1/2 & 1/2 \\ 1/d_B & -1/d_B \end{bmatrix} \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \begin{bmatrix} 1/2 & 1/2 \\ 1/d_B & -1/d_B \end{bmatrix}^\top \\ &= \begin{bmatrix} \sigma^2/2 & 0 \\ 0 & 2\sigma^2/d_B^2 \end{bmatrix}\end{aligned}$$

- Cross-correlations happen to cancel out
  - This does *not* happen in general!

Could do this all in one step

$$\underbrace{\begin{bmatrix} d_R \\ d_L \end{bmatrix}}_{\mathbf{d}} = \underbrace{\begin{bmatrix} \alpha & 0 & 1 & 0 \\ 0 & \alpha & 0 & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} c_R \\ c_L \\ w_1 \\ w_2 \end{bmatrix}}_{\mathbf{w}}$$

$$\underbrace{\begin{bmatrix} \Delta x \\ \Delta \theta \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} 1/2 & 1/2 \\ 1/d_B & -1/d_B \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} d_R \\ d_L \end{bmatrix}}_{\mathbf{d}}$$

$$\mathbf{x} = BA\mathbf{w}$$

$$\Sigma_x = BA\Sigma_w(BA)^\top = BA\Sigma_w A^\top B^\top$$

Sampling from Gaussians

- Sample from Gaussian y where $y \sim N(\mu_y, \sigma_y^2)$
  - Generate Gaussian noise w with $w \sim N(0, 1)$
  - return

$$y = \sigma_y w + \mu_y$$

- Sample from Gaussian $y \sim N(\mu_y, \Sigma_y)$
  - Factor $\Sigma_y = LL^T$
    - If PD, Cholesky gives a unique lower triangular L
    - If PSD, Eigendecomposition gives a (non-unique) factorization $L=VD^{½}$
  - Generate Gaussian noise w with $w \sim N(0, I)$
  - return

$$y = Lw + \mu_y$$

Summary

- Simple parametric model: Normal density
- Covariance vs. information form
- Visualizing normal densities

- Next: Kalman Filter
  Probabilistic Robotics book 3.2