Introduction to Mobile Robotics

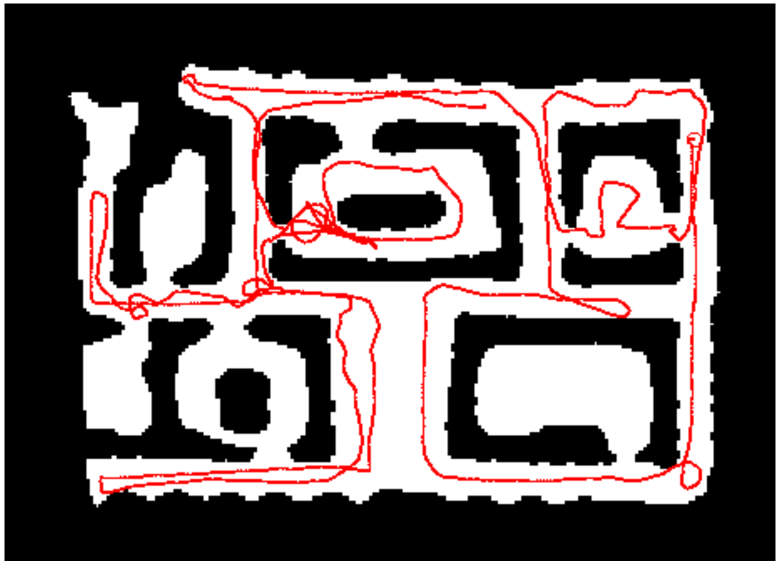
Probabilistic Motion Models

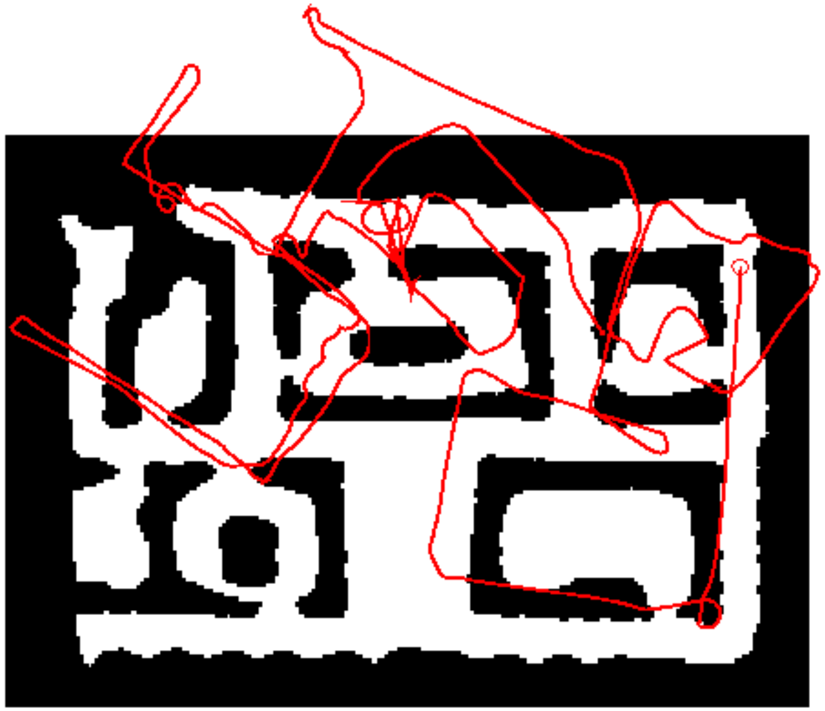
Wolfram Burgard, Maren Bennewitz,
Diego Tipaldi, Luciano Spinello

UNI FREIBURG

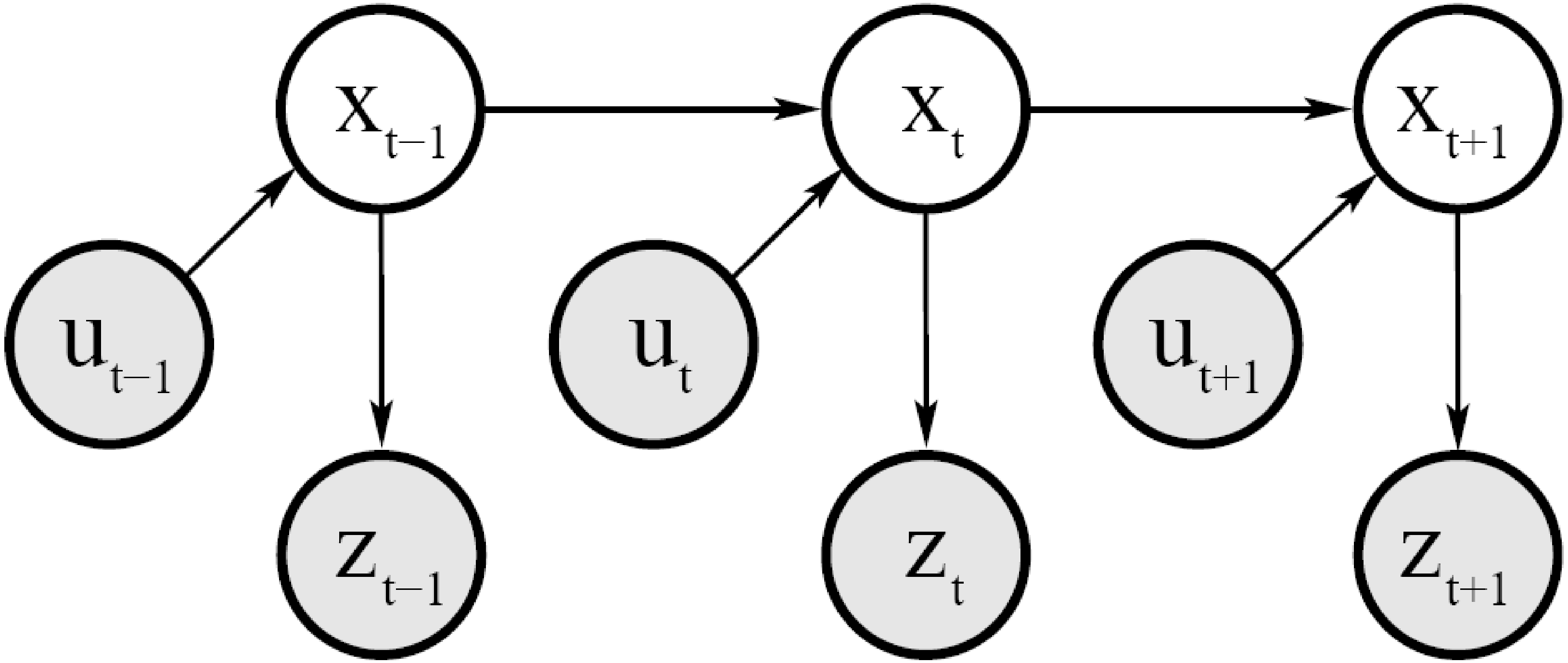
Robot Motion

- Robot motion is inherently uncertain.
- How can we model this uncertainty?





Dynamic Bayesian Network for Controls, States, and Sensations

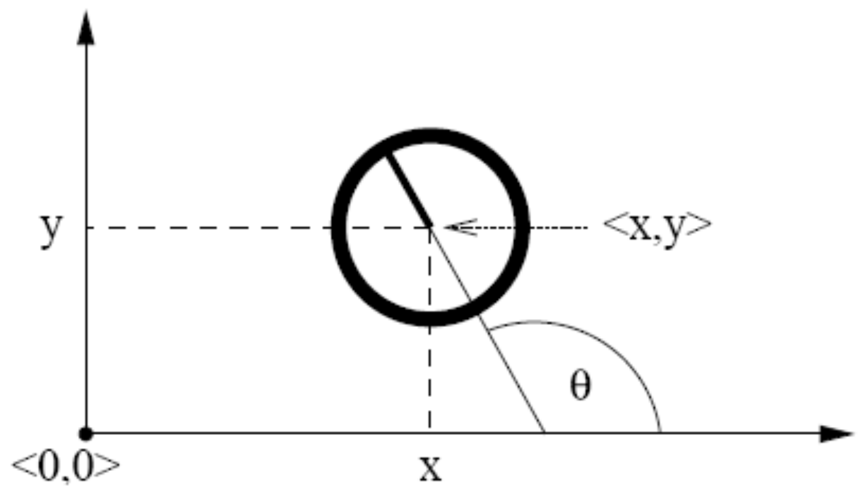


Probabilistic Motion Models

- To implement the Bayes Filter, we need the transition model $p(x_t \mid x_{t-1}, u_t)$
- The term $p(x_t \mid x_{t-1}, u_t)$ specifies a posterior probability, that action u carries the robot from x_{t-1} to x_t.
- In this section we will specify, how $p(x_t \mid x_{t-1}, u_t)$ can be modeled based on the motion equations.

Coordinate Systems

- The configuration of a typical wheeled robot in 3D can be described by six parameters.
- Three-dimensional Cartesian coordinates plus three Euler angles roll, pitch, and yaw.
- Throughout this section, we consider robots operating on a planar surface.
- The state space of such systems is three-dimensional (x,y,θ).

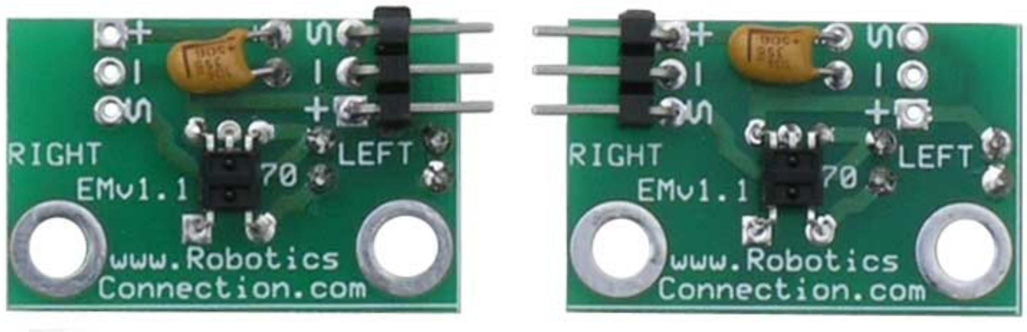


Typical Motion Models

- In practice, one often finds two types of motion models:
  - **Odometry-based**
  - **Velocity-based** (**dead reckoning**)
- Odometry-based models are used when systems are equipped with wheel encoders.
- Velocity-based models have to be applied when no wheel encoders are given.
- They calculate the new pose based on the velocities and the time elapsed.

Example Wheel Encoders

These modules provide +5V output when they "see" white, and a 0V output when they "see" black.



These disks are manufactured out of high quality laminated color plastic to offer a very crisp black to white transition. This enables a wheel encoder sensor to easily see the transitions.

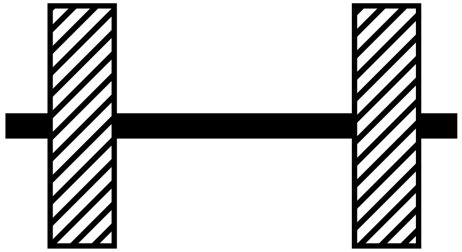
Source: http://www.active-robots.com/

Dead Reckoning

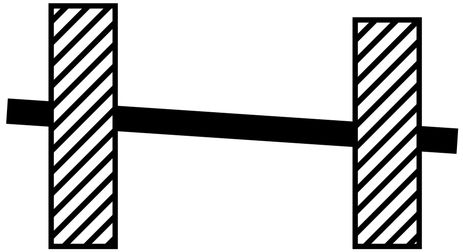
- Derived from "deduced reckoning."
- Mathematical procedure for determining the present location of a vehicle.
- Achieved by calculating the current pose of the vehicle based on its velocities and the time elapsed.
- Historically used to log the position of ships.

[Image source: Wikipedia, LoKiLeCh]

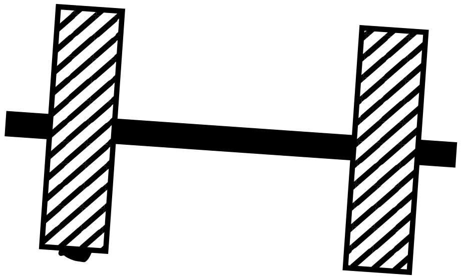
Reasons for Motion Errors



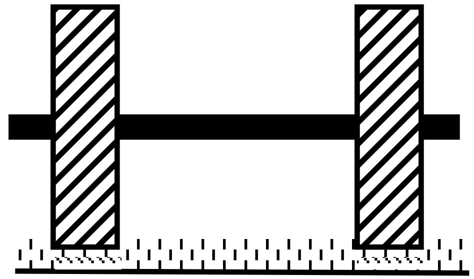
ideal case



different wheel diameters



bump



carpet

and many more …

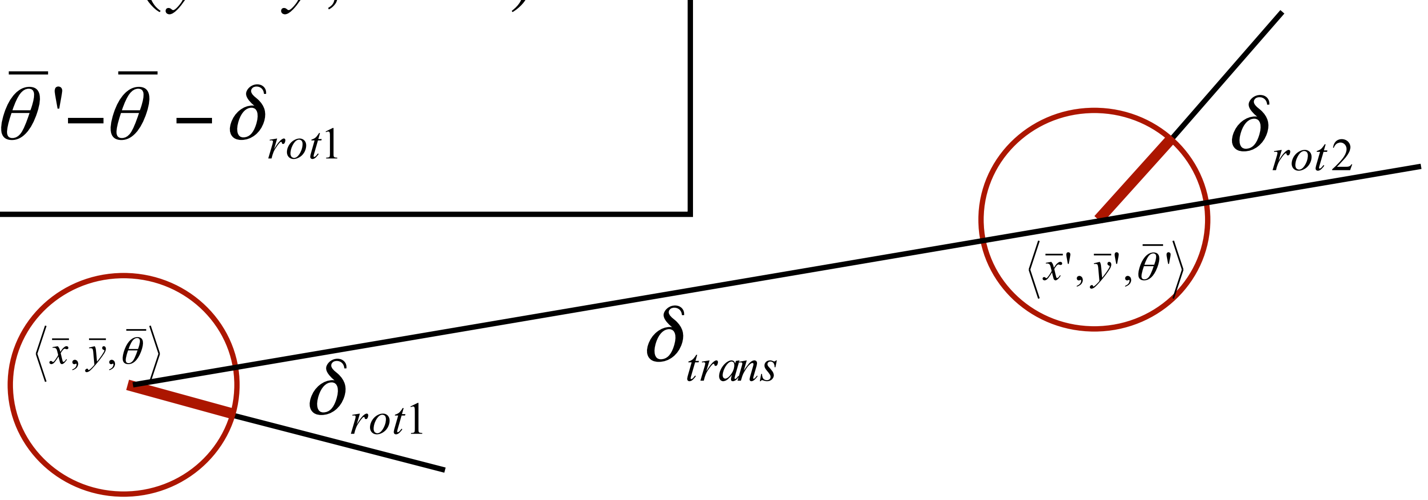
Odometry Model

- Robot moves from $\langle \bar{x}, \bar{y}, \bar{\theta} \rangle$ to $\langle \bar{x}', \bar{y}', \bar{\theta}' \rangle$.
- Odometry information $u = \langle \delta_{rot1}, \delta_{rot2}, \delta_{trans} \rangle$.

$$\delta_{trans} = \sqrt{(\bar{x}' - \bar{x})^2 + (\bar{y}' - \bar{y})^2}$$

$$\delta_{rot1} = \operatorname{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$$

$$\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$$



The atan2 Function

- Extends the inverse tangent and correctly copes with the signs of x and y.

$$\text{atan2}(y, x) = \begin{cases} \text{atan}(y/x) & \text{if } x > 0 \\ \text{sign}(y)\,(\pi - \text{atan}(|y/x|)) & \text{if } x < 0 \\ 0 & \text{if } x = y = 0 \\ \text{sign}(y)\,\pi/2 & \text{if } x = 0, y \neq 0 \end{cases}$$

Noise Model for Odometry

- The measured motion is given by the true motion corrupted with noise.

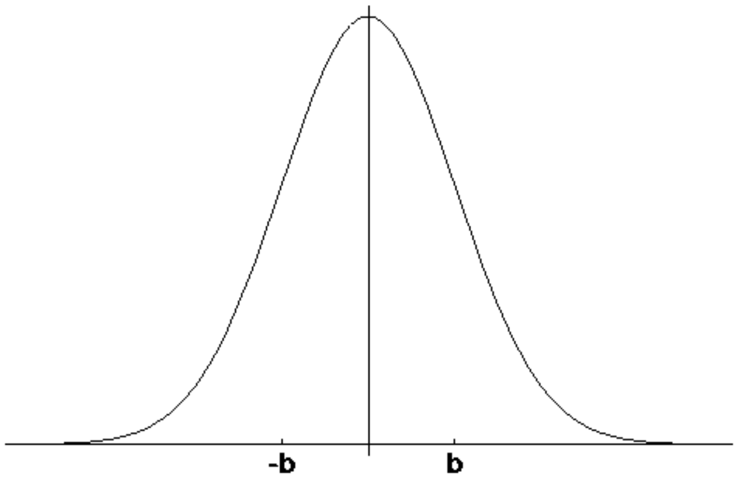
$$\hat{\delta}_{rot1} = \delta_{rot1} + \varepsilon_{\alpha_1 |\delta_{rot1}| + \alpha_2 |\delta_{trans}|}$$

$$\hat{\delta}_{trans} = \delta_{trans} + \varepsilon_{\alpha_3 |\delta_{trans}| + \alpha_4 |\delta_{rot1} + \delta_{rot2}|}$$

$$\hat{\delta}_{rot2} = \delta_{rot2} + \varepsilon_{\alpha_1 |\delta_{rot2}| + \alpha_2 |\delta_{trans}|}$$

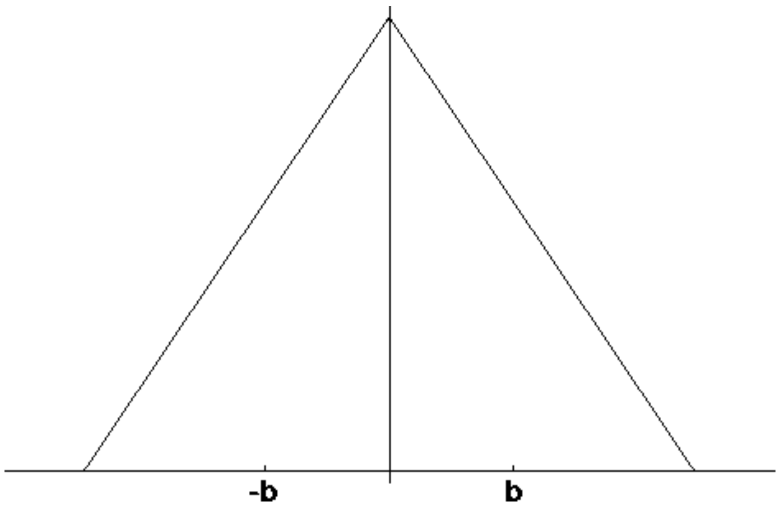
Typical Distributions for Probabilistic Motion Models

Normal distribution



$$\varepsilon_{\sigma^2}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{x^2}{\sigma^2}}$$

Triangular distribution



$$\varepsilon_{\sigma^2}(x) = \begin{cases} 0 \text{ if } |x| > \sqrt{6\sigma^2} \\ \dfrac{\sqrt{6\sigma^2} - |x|}{6\sigma^2} \end{cases}$$

Calculating the Probability Density (zero-centered)

- For a normal distribution

query point

1. Algorithm **prob_normal_distribution**(a,b):

std. deviation

2. return $\frac{1}{\sqrt{2\pi\, b^2}} \exp\left\{-\frac{1}{2}\frac{a^2}{b^2}\right\}$

- For a triangular distribution

1. Algorithm **prob_triangular_distribution**(a,b):
2. return $\max\left\{0, \frac{1}{\sqrt{6}\, b} - \frac{|a|}{6\, b^2}\right\}$

Calculating the Posterior Given x, x', and Odometry

odometry

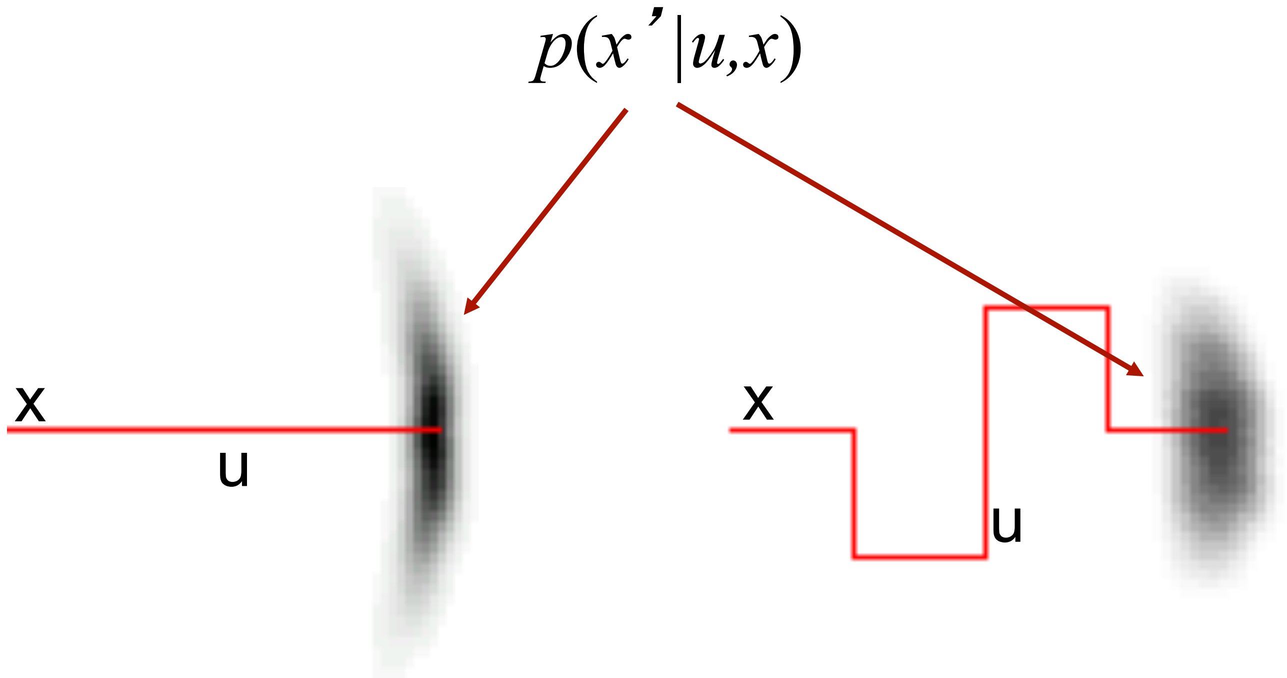
1. Algorithm **motion_model_odometry**$(x, x', \bar{x}, \bar{x}')$
2. $\delta_{trans} = \sqrt{(\bar{x}'-\bar{x})^2 + (\bar{y}'-\bar{y})^2}$
3. $\delta_{rot1} = \text{atan2}(\bar{y}'-\bar{y}, \bar{x}'-\bar{x}) - \bar{\theta}$
4. $\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$
5. $\hat{\delta}_{trans} = \sqrt{(x'-x)^2 + (y'-y)^2}$
6. $\hat{\delta}_{rot1} = \text{atan2}(y'-y, x'-x) - \theta$
7. $\hat{\delta}_{rot2} = \theta' - \theta - \hat{\delta}_{rot1}$
8. $p_1 = \text{prob}(\delta_{\text{rot1}} - \hat{\delta}_{\text{rot1}}, \alpha_1 |\delta_{\text{rot1}}| + \alpha_2 \delta_{\text{trans}})$
9. $p_2 = \text{prob}(\delta_{\text{trans}} - \hat{\delta}_{\text{trans}}, \alpha_3 \delta_{\text{trans}} + \alpha_4 (|\delta_{\text{rot1}}| + |\delta_{\text{rot2}}|))$
10. $p_3 = \text{prob}(\delta_{\text{rot2}} - \hat{\delta}_{\text{rot2}}, \alpha_1 |\delta_{\text{rot2}}| + \alpha_2 \delta_{\text{trans}})$
11. return $p_1 \cdot p_2 \cdot p_3$

odometry params (u) (lines 2–4)

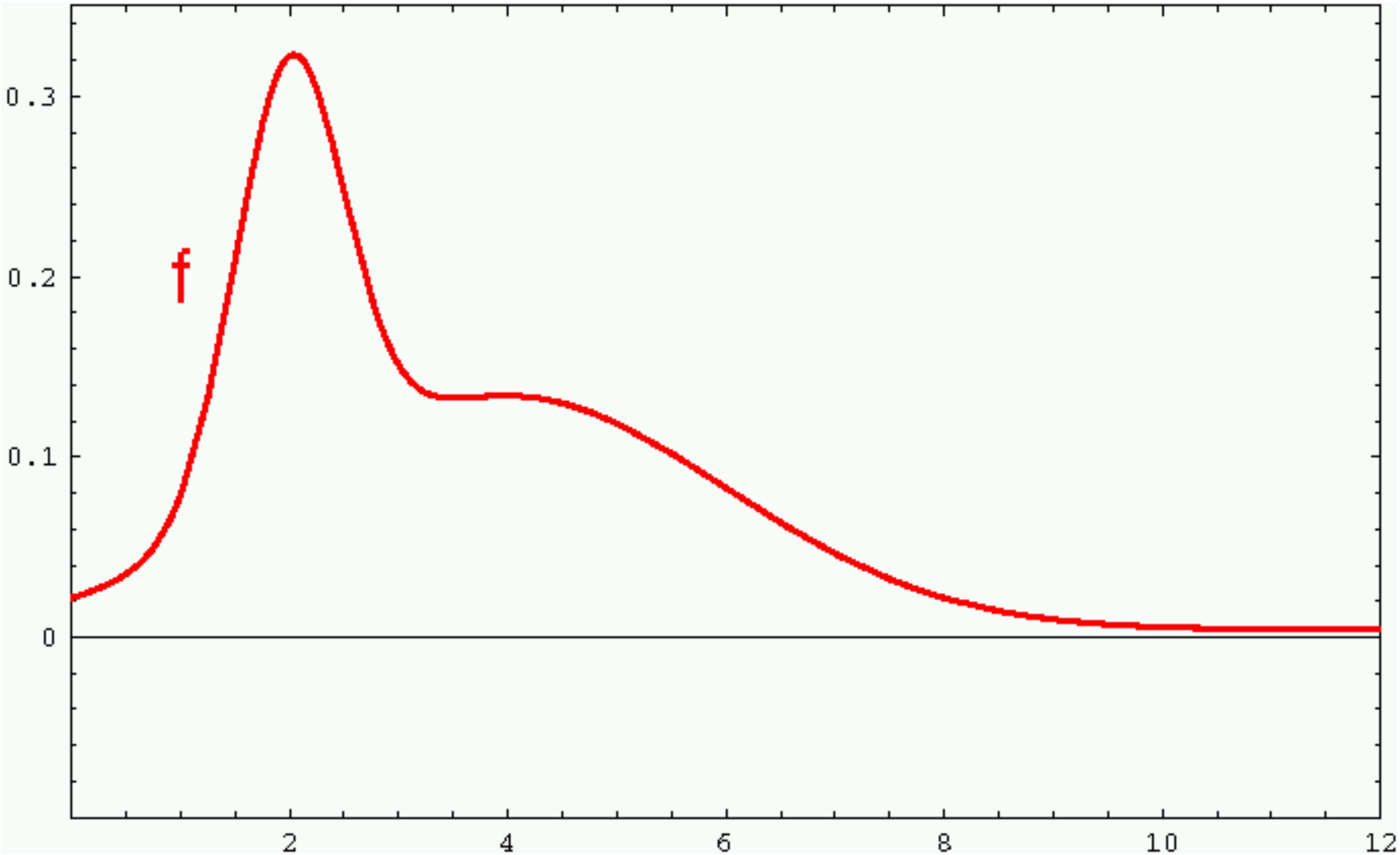
values of interest (x,x') (lines 5–7)

Application

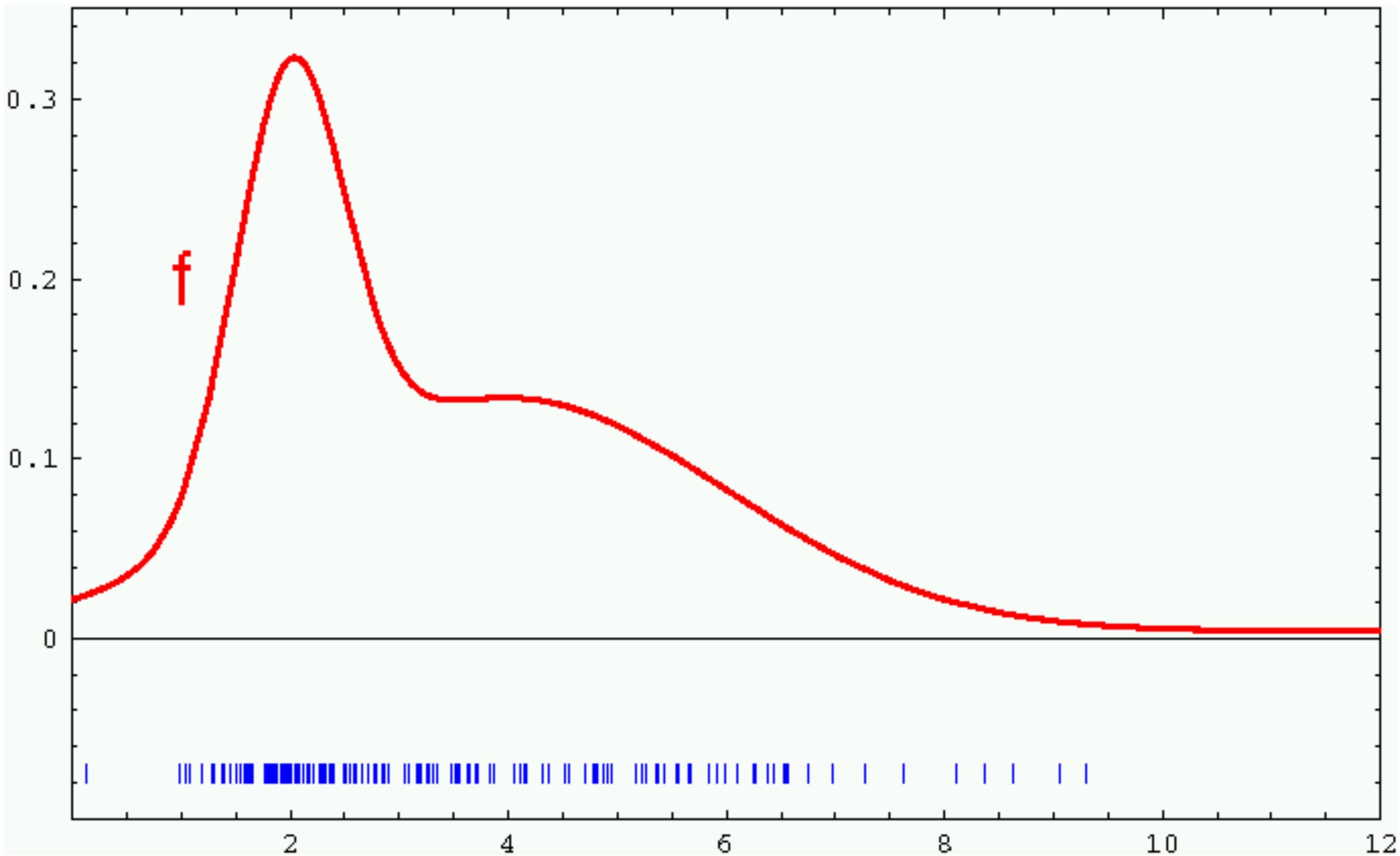
- Repeated application of the sensor model for short movements.
- Typical banana-shaped distributions obtained for the 2d-projection of the 3d posterior.



Sample-Based Density Representation



Sample-Based Density Representation

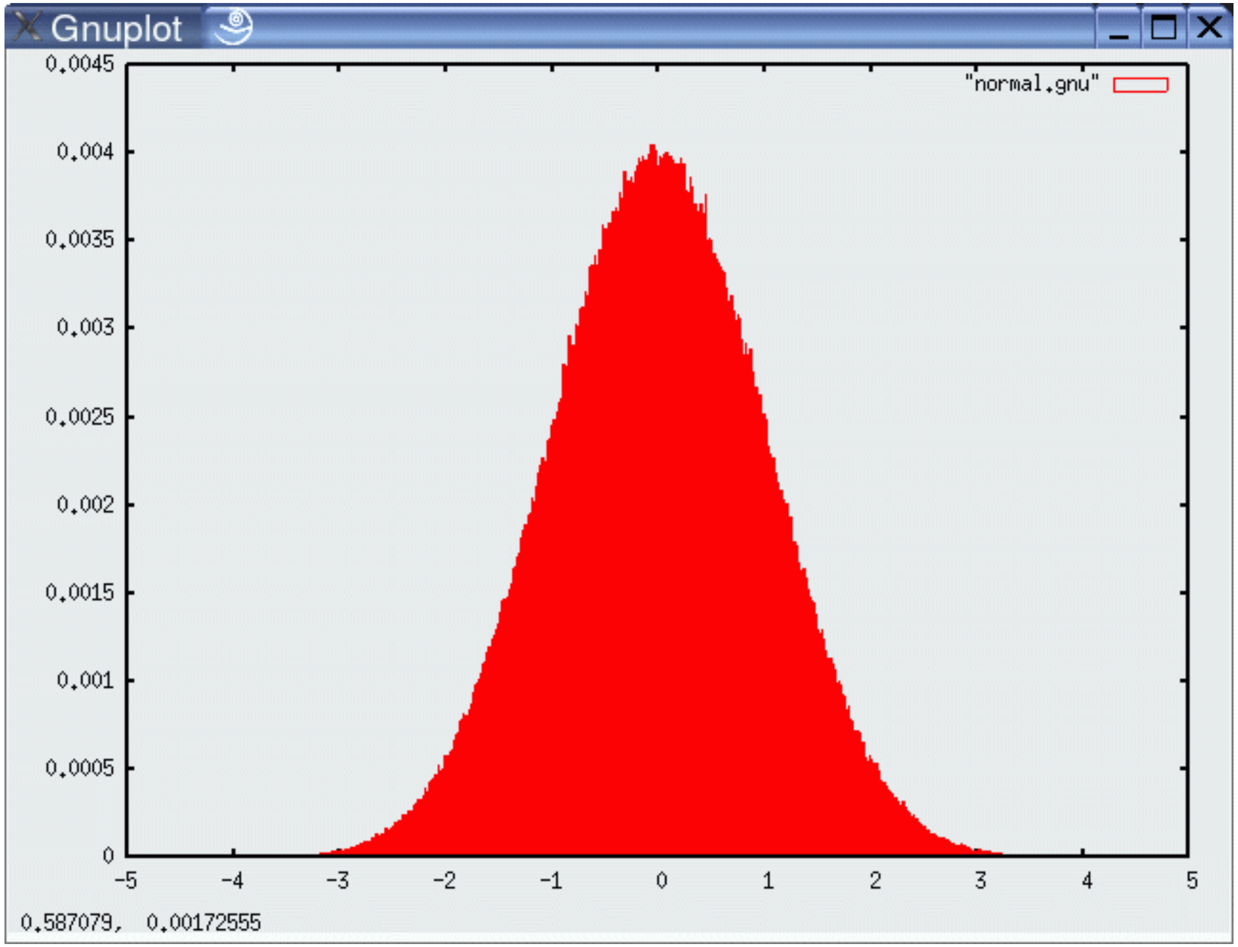


How to Sample from Normal Distributions?

- Sampling from a normal distribution

1. Algorithm **sample_normal_distribution**(b):
2. return $\frac{1}{2} \sum_{i=1}^{12} rand(-b, b)$

Normally Distributed Samples



10^6 samples

How to Sample from Normal or Triangular Distributions?

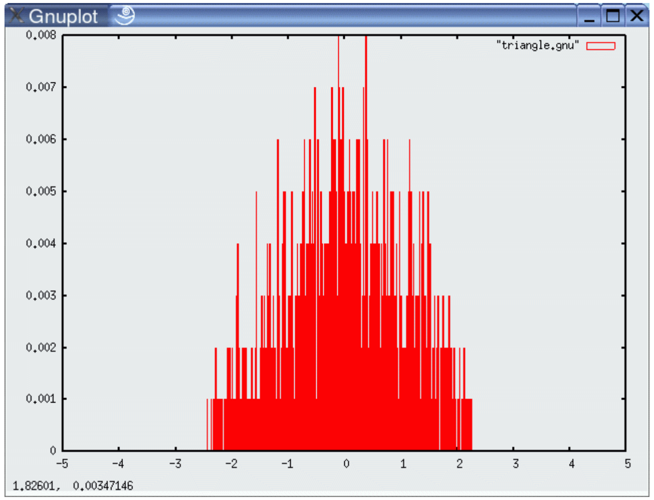
- Sampling from a normal distribution

1. Algorithm **sample_normal_distribution**(b):
2. return $\frac{1}{2}\sum_{i=1}^{12} rand(-b, b)$

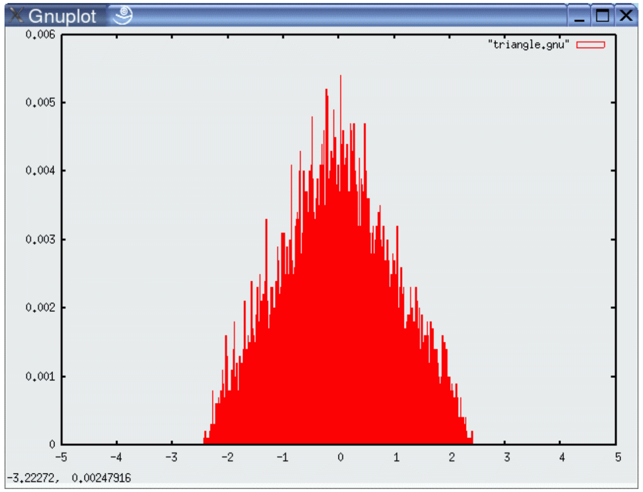
- Sampling from a triangular distribution

1. Algorithm **sample_triangular_distribution**(b):
2. return $\frac{\sqrt{6}}{2}\left[\mathbf{rand}(-b, b) + \mathbf{rand}(-b, b)\right]$

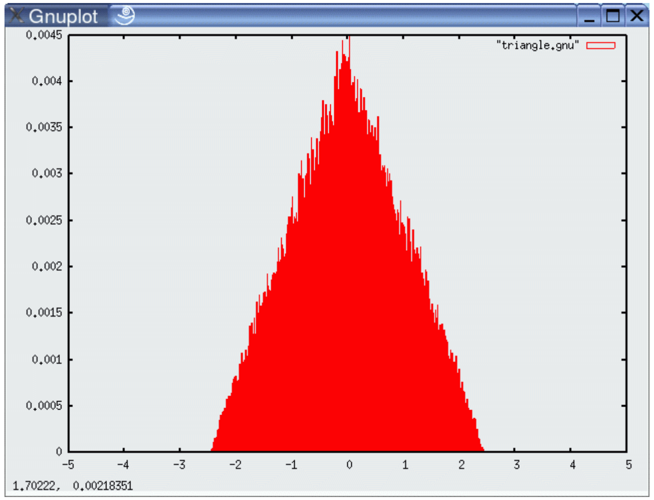
For Triangular Distribution



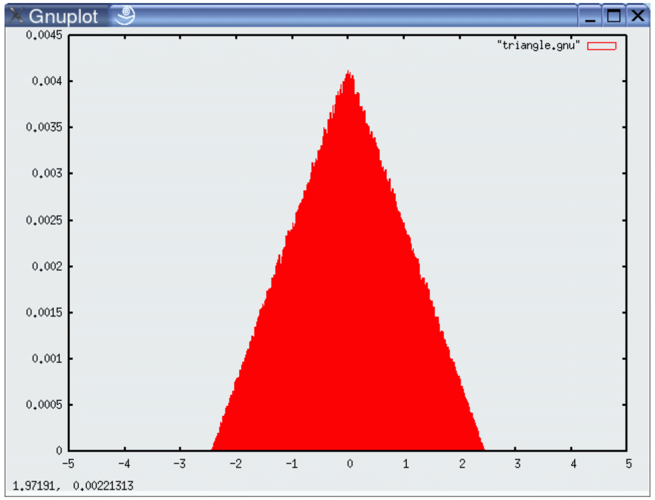
10^3 samples



10^4 samples

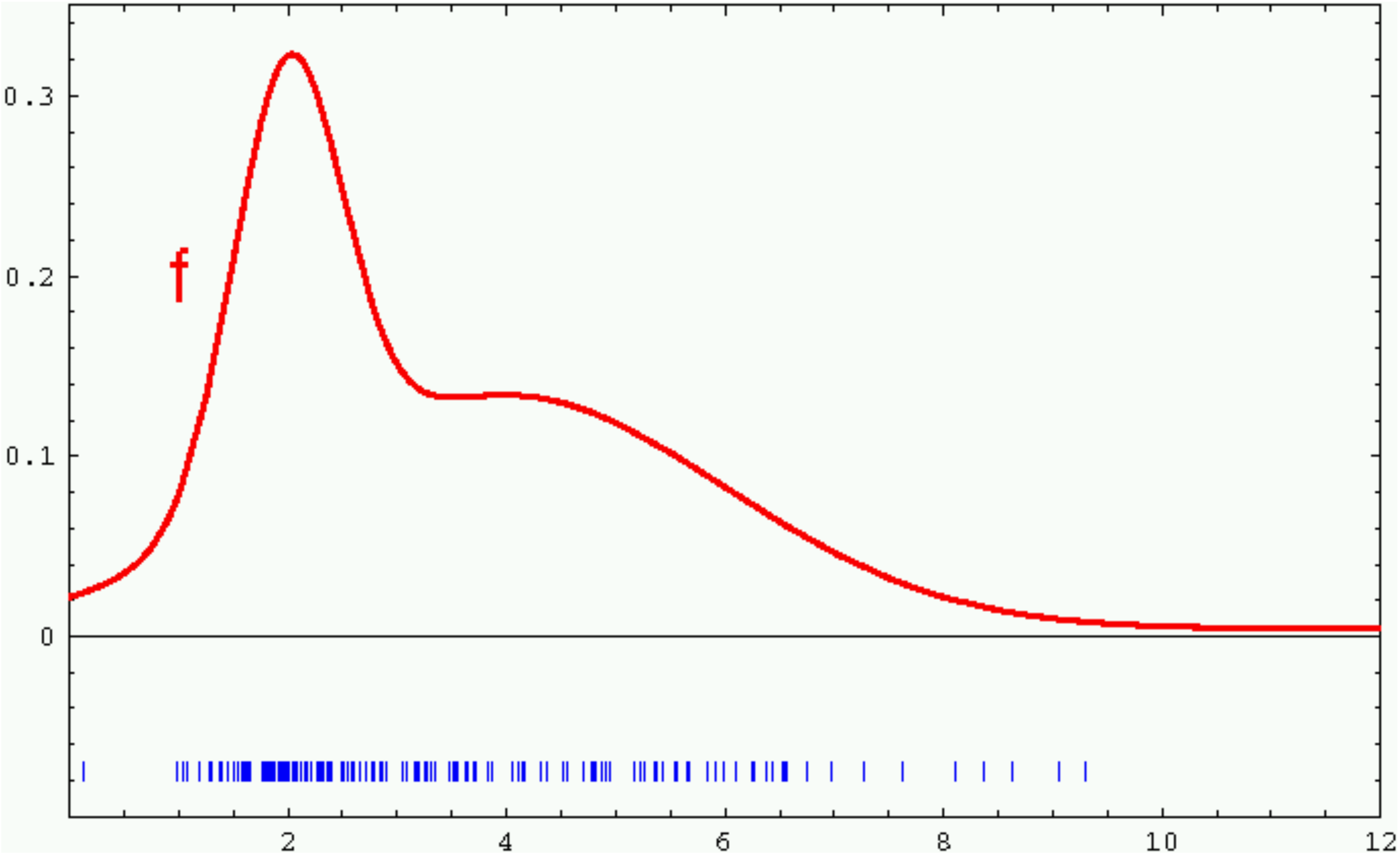


10^5 samples



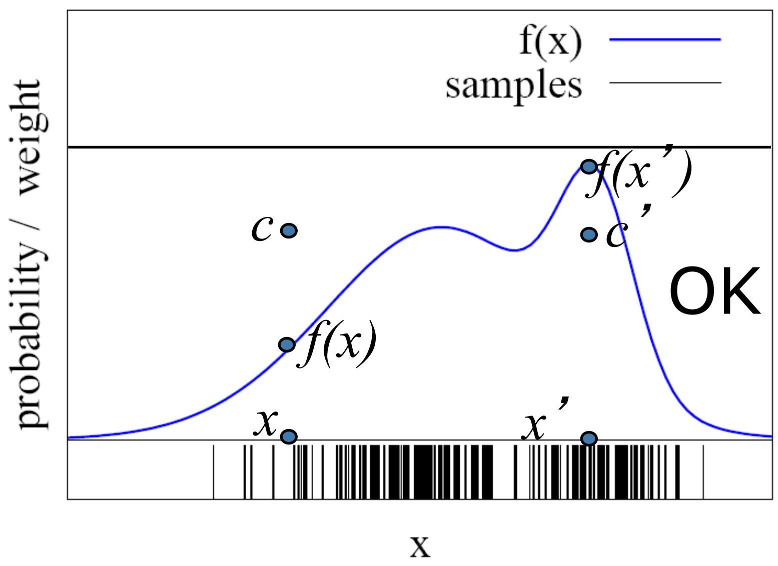
10^6 samples

How to Obtain Samples from Arbitrary Functions?



Rejection Sampling

- Sampling from arbitrary distributions
- Sample x from a uniform distribution from $[-b,b]$
- Sample c from $[0, max\ f]$
- if $f(x) > c$ keep the sample
  otherwise reject the sample



Rejection Sampling

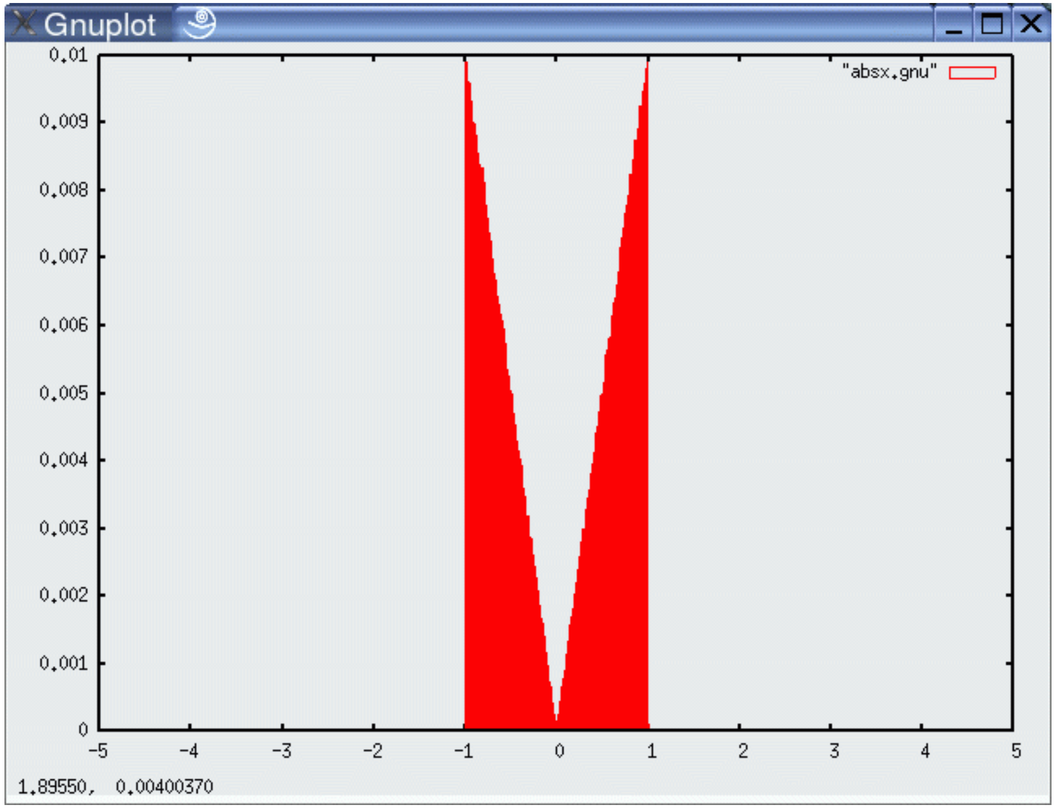
- Sampling from arbitrary distributions

1. Algorithm **sample_distribution**(f,b):
2. repeat
3. $x = \textbf{rand}(-b, b)$
4. $y = \textbf{rand}(0, \max\{f(x) \mid x \in [-b, b]\})$
5. until $(y \le f(x))$
6. return x

Example

- Sampling from

$$f(x) = \begin{cases} \text{abs}(x) & x \in [-1;1] \\ 0 & \text{otherwise} \end{cases}$$



Sample Odometry Motion Model

1. Algorithm **sample_motion_model**(u, x):

   $u = \langle \delta_{rot1}, \delta_{rot2}, \delta_{trans} \rangle, x = \langle x, y, \theta \rangle$

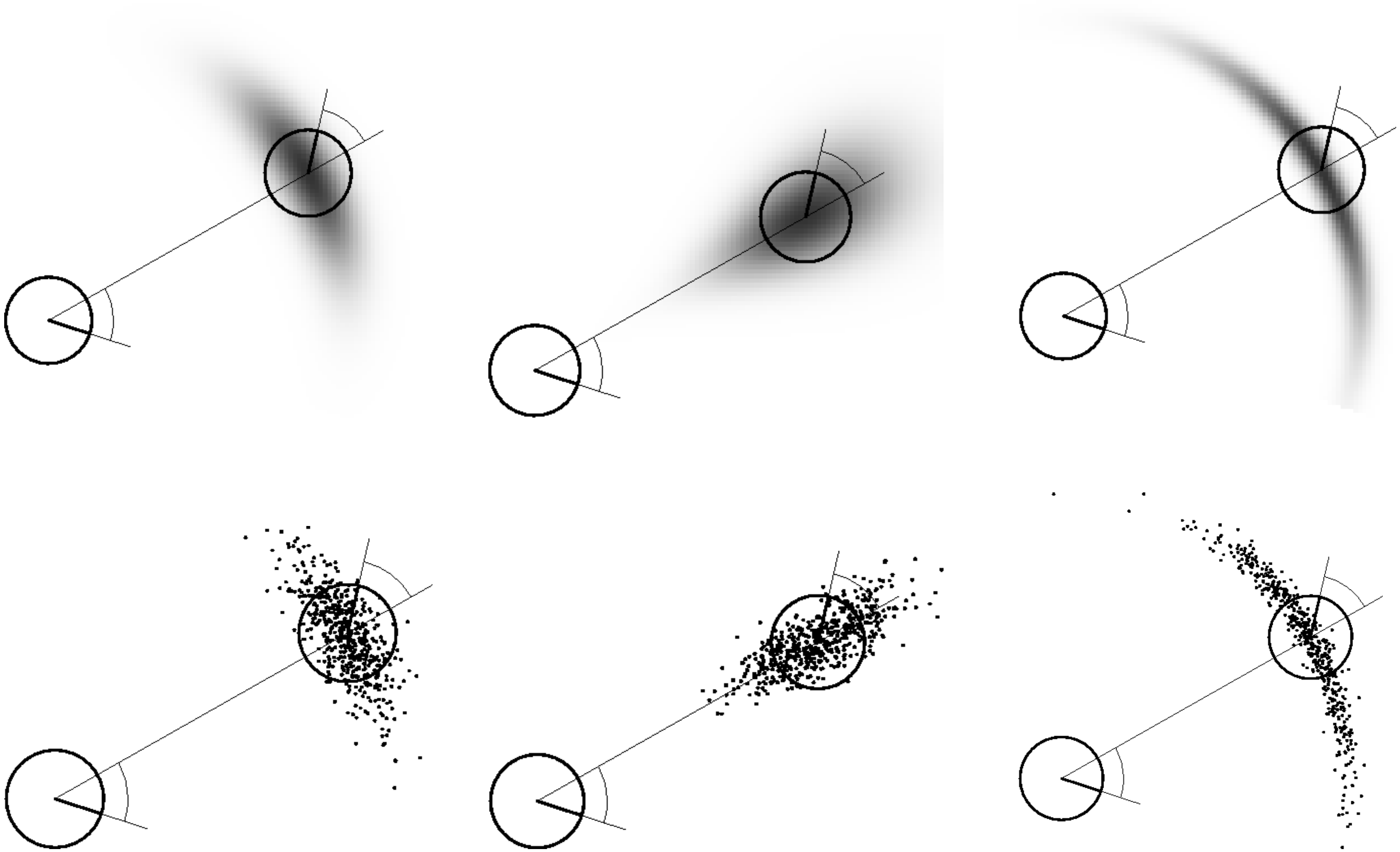
1. $\hat{\delta}_{rot1} = \delta_{rot1} + \text{sample}(\alpha_1 |\delta_{rot1}| + \alpha_2 \delta_{trans})$
2. $\hat{\delta}_{trans} = \delta_{trans} + \text{sample}(\alpha_3 \delta_{trans} + \alpha_4 (|\delta_{rot1}| + |\delta_{rot2}|))$
3. $\hat{\delta}_{rot2} = \delta_{rot2} + \text{sample}(\alpha_1 |\delta_{rot2}| + \alpha_2 \delta_{trans})$

4. $x' = x + \hat{\delta}_{trans} \cos(\theta + \hat{\delta}_{rot1})$
5. $y' = y + \hat{\delta}_{trans} \sin(\theta + \hat{\delta}_{rot1})$
6. $\theta' = \theta + \hat{\delta}_{rot1} + \hat{\delta}_{rot2}$

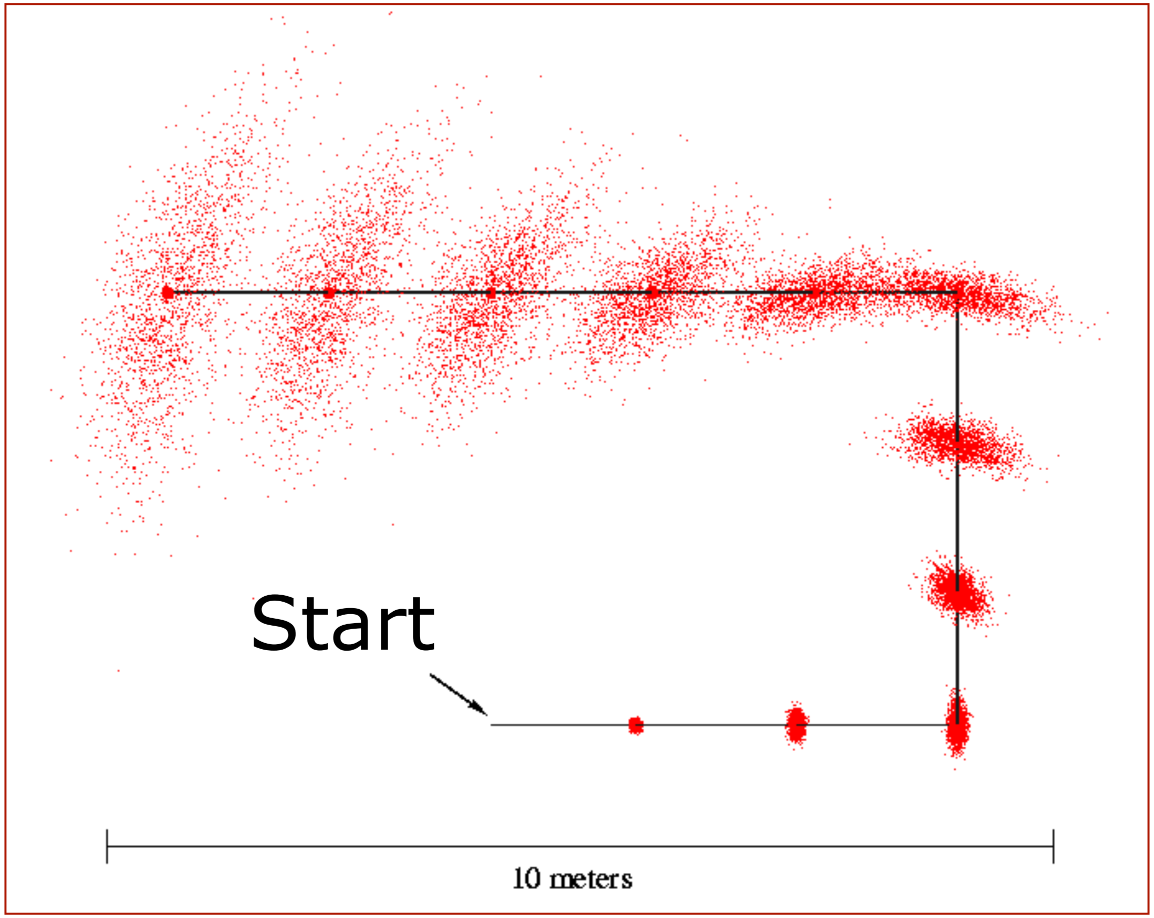
7. Return $\langle x', y', \theta' \rangle$

sample_normal_distribution

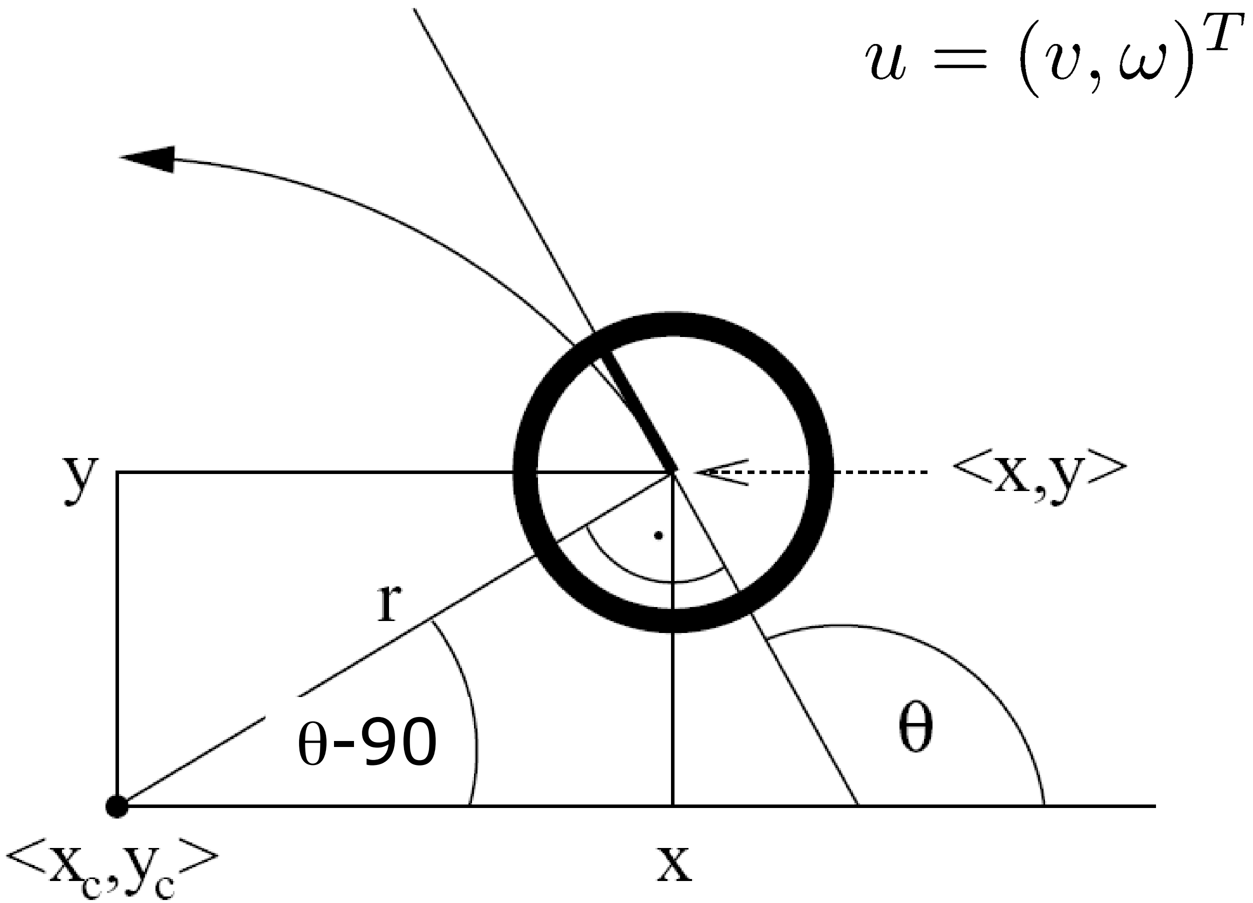
Examples (Odometry-Based)



Sampling from Our Motion Model



Velocity-Based Model



Noise Model for the Velocity-Based Model

- The measured motion is given by the true motion corrupted with noise.

$$\hat{v} = v + \varepsilon_{\alpha_1 |v| + \alpha_2 |\omega|}$$

$$\hat{\omega} = \omega + \varepsilon_{\alpha_3 |v| + \alpha_4 |\omega|}$$

- Question: What is the disadvantage of this noise model?

Noise Model for the Velocity-Based Model

- The $(\hat{v}, \hat{\omega})$-circle constrains the final orientation (2D manifold in a 3D space)
- Better approach:

$$\hat{v} = v + \varepsilon_{\alpha_1 |v| + \alpha_2 |\omega|}$$

$$\hat{\omega} = \omega + \varepsilon_{\alpha_3 |v| + \alpha_4 |\omega|}$$

$$\hat{\gamma} = \varepsilon_{\alpha_5 |v| + \alpha_6 |\omega|}$$

Term to account for the final rotation

Motion Including 3rd Parameter

$$x' = x - \frac{\hat{v}}{\hat{\omega}} \sin\theta + \frac{\hat{v}}{\hat{\omega}} \, sin(\theta + \hat{\omega}\Delta t)$$

$$y' = y + \frac{\hat{v}}{\hat{\omega}} \cos\theta - \frac{\hat{v}}{\hat{\omega}} \, cos(\theta + \hat{\omega}\Delta t)$$

$$\theta' = \theta + \hat{\omega}\Delta t + \hat{\gamma}\Delta t$$

↑

Term to account for the final rotation

Equation for the Velocity Model

$$x_{t-1} = (x, y, \theta)^T$$
$$x_t = (x,' y', \theta')^T$$

Center of circle:

$$\begin{pmatrix} x^* \\ y^* \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} -\lambda \sin\theta \\ \lambda \cos\theta \end{pmatrix}$$

some constant (distance to ICC)

(center of circle is orthogonal to the initial heading)

Equation for the Velocity Model

$$x_{t-1} = (x, y, \theta)^T$$
$$x_t = (x,' y', \theta')^T$$

some constant

Center of circle:

$$\begin{pmatrix} x^* \\ y^* \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} -\lambda \sin\theta \\ \lambda \cos\theta \end{pmatrix} = \begin{pmatrix} \frac{x+x'}{2} + \mu(y - y') \\ \frac{y+y'}{2} + \mu(x' - x) \end{pmatrix}$$

some constant (circle's center lies on a ray half way between x and x' and is orthogonal to the line between x and x')

Equation for the Velocity Model

$$x_{t-1} = (x, y, \theta)^T$$
$$x_t = (x,' y', \theta')^T$$

some constant

Center of circle:

$$\begin{pmatrix} x^* \\ y^* \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} -\lambda \sin\theta \\ \lambda \cos\theta \end{pmatrix} = \begin{pmatrix} \frac{x+x'}{2} + \mu(y-y') \\ \frac{y+y'}{2} + \mu(x'-x) \end{pmatrix}$$

Allows us to solve the equations to:

$$\mu = \frac{1}{2} \frac{(x-x')\cos\theta + (y-y')\sin\theta}{(y-y')\cos\theta - (x-x')\sin\theta}$$

Equation for the Velocity Model

$$x_{t-1} = (x, y, \theta)^T$$
$$x_t = (x', y', \theta')^T$$

$$\begin{pmatrix} x^* \\ y^* \end{pmatrix} = \begin{pmatrix} \frac{x+x'}{2} + \mu(y - y') \\ \frac{y+y'}{2} + \mu(x' - x) \end{pmatrix} \qquad \mu = \frac{1}{2} \frac{(x - x')\cos\theta + (y - y')\sin\theta}{(y - y')\cos\theta - (x - x')\sin\theta}$$

and

$$r^* = \sqrt{(x' - x)^2 + (y' - y)^2}$$
$$\Delta\theta = \operatorname{atan2}(y' - y^*, x' - x^*) - \operatorname{atan2}(y - y^*, x - x^*)$$

Equation for the Velocity Model

- The parameters of the circle:

$$r^* = \sqrt{(x' - x)^2 + (y' - y)^2}$$
$$\Delta\theta = \mathrm{atan2}(y' - y^*, x' - x^*) - \mathrm{atan2}(y - y^*, x - x^*)$$

- allow for computing the velocities as

$$v = \frac{\Delta\theta}{\Delta t} r^*$$
$$\omega = \frac{\Delta\theta}{\Delta t}$$

Posterior Probability for Velocity Model

Algorithm motion_model_velocity(x_t, u_t, x_{t-1})**:** $p(x_t \mid x_{t-1}, u_t)$

$$\mu = \frac{1}{2} \frac{(x - x')\cos\theta + (y - y')\sin\theta}{(y - y')\cos\theta - (x - x')\sin\theta}$$

$$x^* = \frac{x + x'}{2} + \mu(y - y')$$

$$y^* = \frac{y + y'}{2} + \mu(x' - x)$$

$$r^* = \sqrt{(x - x^*)^2 + (y - y^*)^2}$$

$$\Delta\theta = \text{atan2}(y' - y^*, x' - x^*) - \text{atan2}(y - y^*, x - x^*)$$

$$\hat{v} = \frac{\Delta\theta}{\Delta t}\, r^*$$

$$\hat{\omega} = \frac{\Delta\theta}{\Delta t}$$

$$\hat{\gamma} = \frac{\theta' - \theta}{\Delta t} - \hat{\omega}$$

return $\mathbf{prob}(v - \hat{v}, \alpha_1 v^2 + \alpha_2 \omega^2) \cdot \mathbf{prob}(\omega - \hat{\omega}, \alpha_3 v^2 + \alpha_4 \omega^2)$
$\cdot\, \mathbf{prob}(\hat{\gamma}, \alpha_5 v^2 + \alpha_6 \omega^2)$

Sampling from Velocity Model

Algorithm sample_motion_model_velocity(u_t, x_{t-1})**:**

$\hat{v} = v + \textbf{sample}(\alpha_1 v^2 + \alpha_2 \omega^2)$

$\hat{\omega} = \omega + \textbf{sample}(\alpha_3 v^2 + \alpha_4 \omega^2)$

$\hat{\gamma} = \textbf{sample}(\alpha_5 v^2 + \alpha_6 \omega^2)$

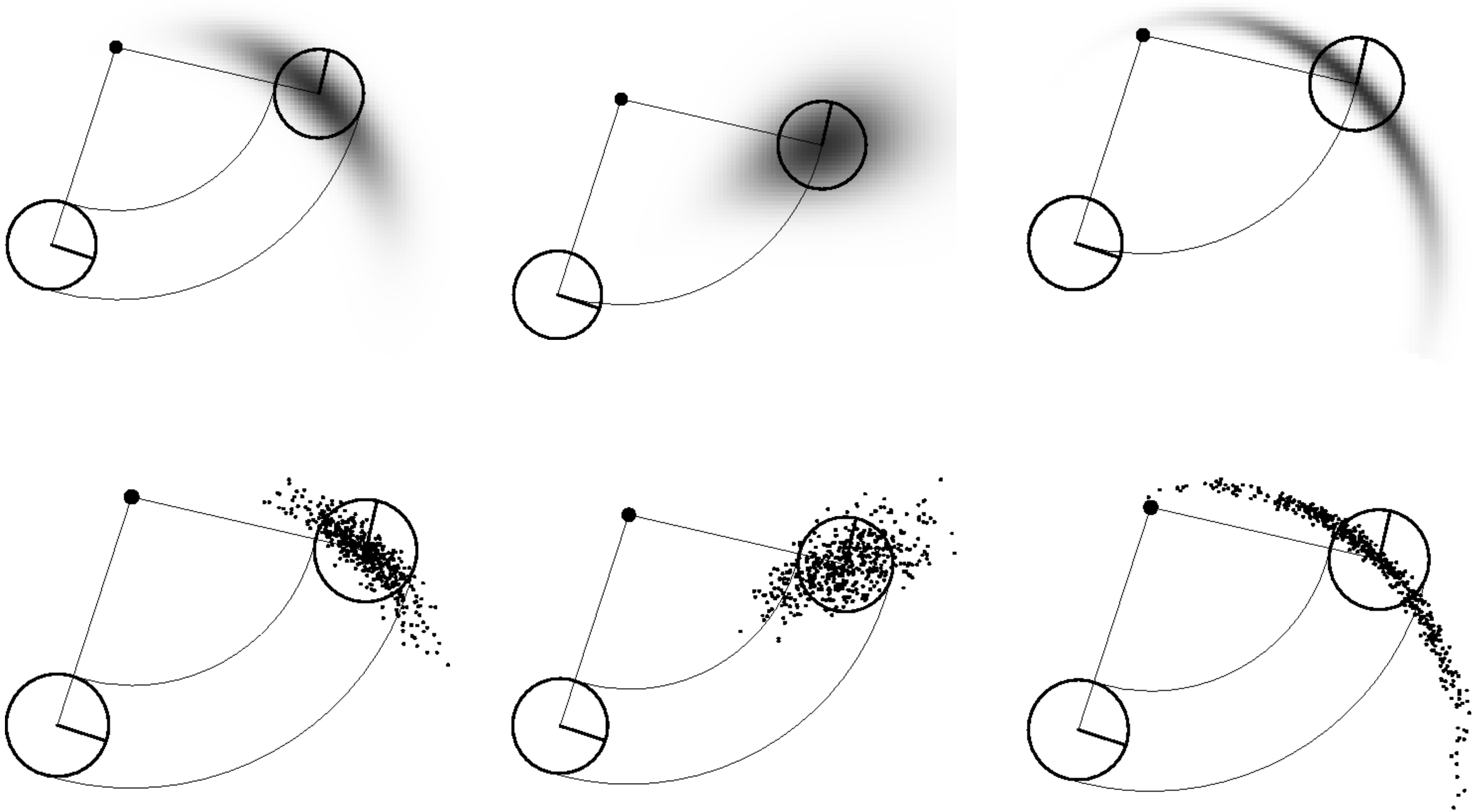
$x' = x - \frac{\hat{v}}{\hat{\omega}} \sin\theta + \frac{\hat{v}}{\hat{\omega}}\ sin(\theta + \hat{\omega}\Delta t)$

$y' = y + \frac{\hat{v}}{\hat{\omega}} \cos\theta - \frac{\hat{v}}{\hat{\omega}}\ cos(\theta + \hat{\omega}\Delta t)$

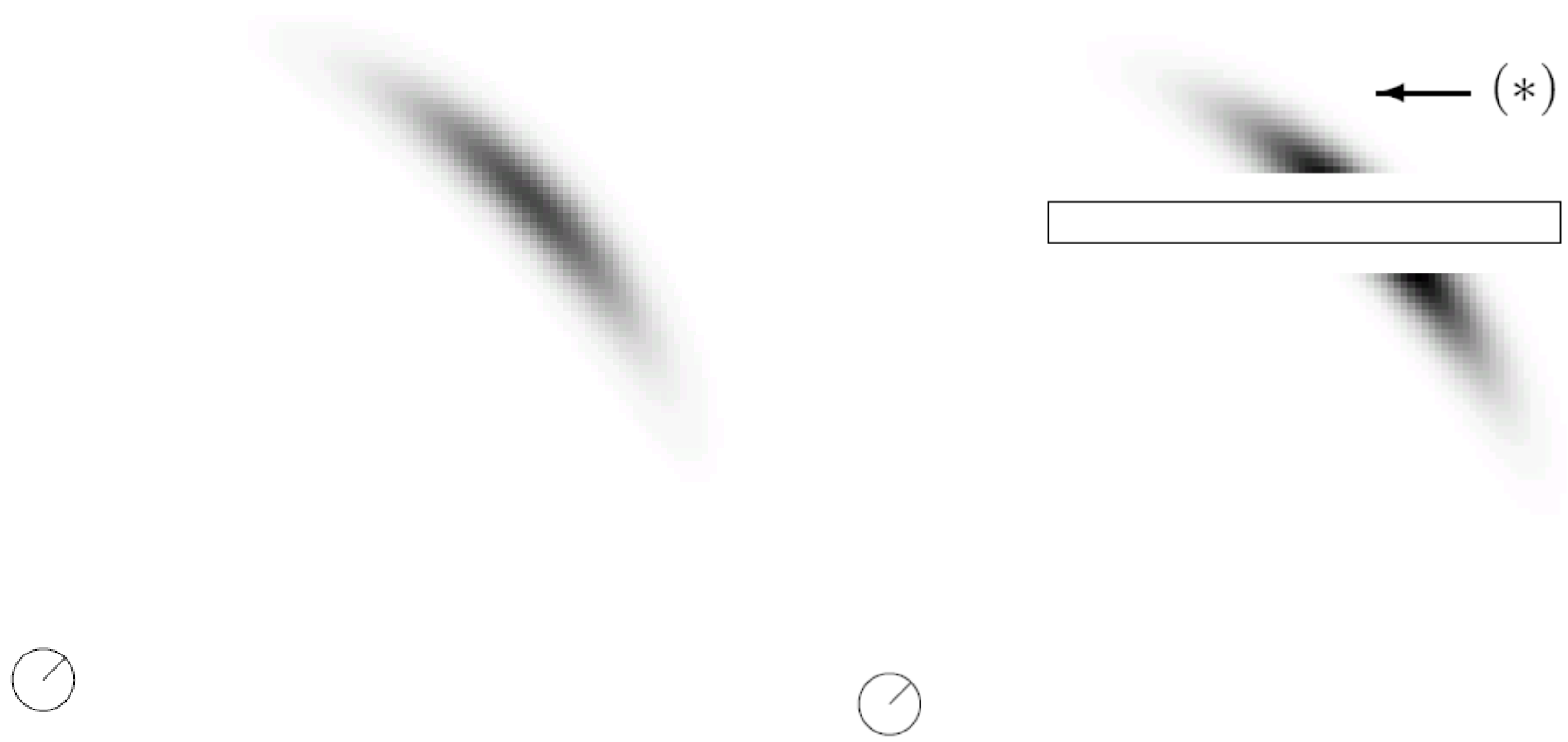
$\theta' = \theta + \hat{\omega}\Delta t + \hat{\gamma}\Delta t$

return $x_t = (x', y', \theta')^T$

Examples (Velocity-Based)



Map-Consistent Motion Model



$$p(x' \mid u, x) \neq p(x' \mid u, x, m)$$

Approximation: $p(x' \mid u, x, m) = \eta \; p(x' \mid m) \; p(x' \mid u, x)$

Summary

- We discussed motion models for odometry-based and velocity-based systems
- We discussed ways to calculate the posterior probability $p(x' \mid x, u)$.
- We also described how to sample from $p(x' \mid x, u)$.
- Typically the calculations are done in fixed time intervals Δt.
- In practice, the parameters of the models have to be learned.
- We also discussed how to improve this motion model to take the map into account.