Probabilistic Robotics*

Probabilistic Motion Models

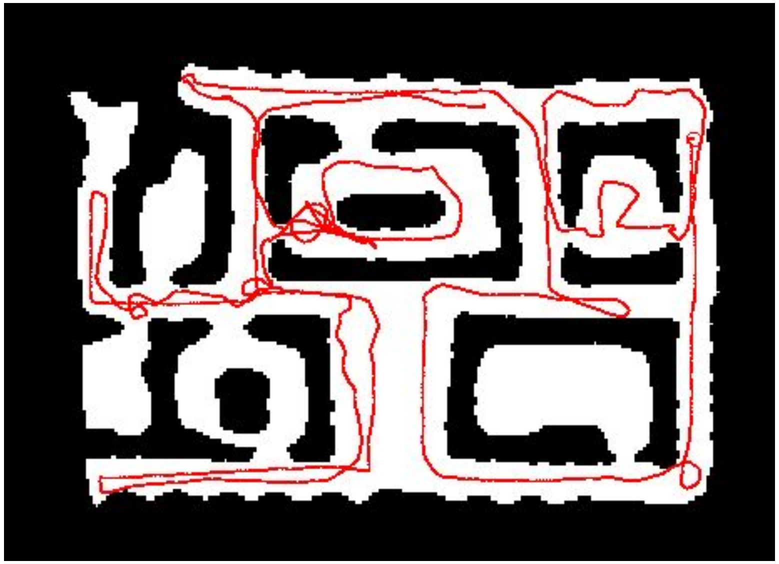
Mohan Sridharan
University of Birmingham, UK
m.sridharan@bham.ac.uk

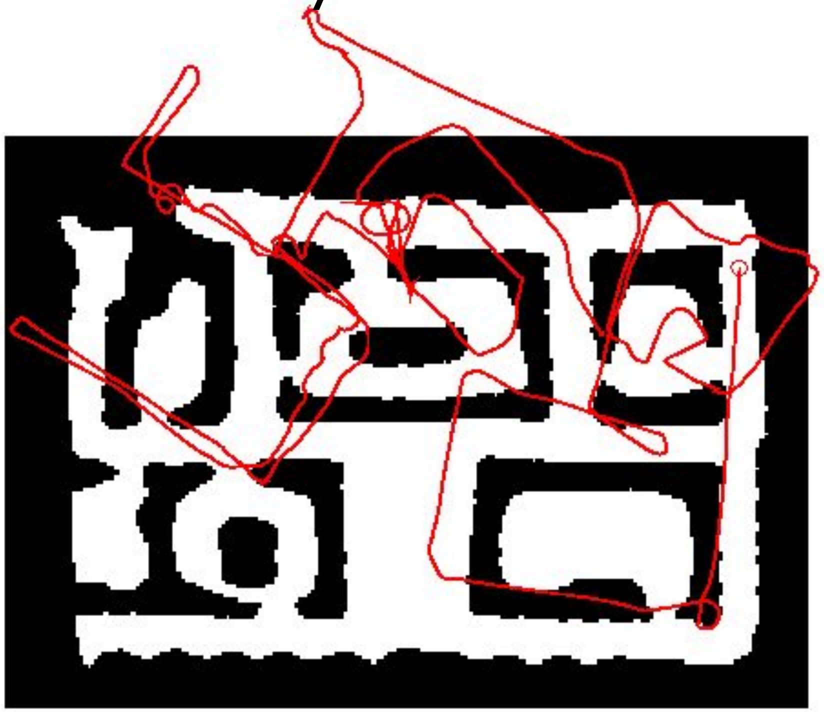
*Revised original slides that accompany the book by Thrun, Burgard and Fox.

Robot Motion

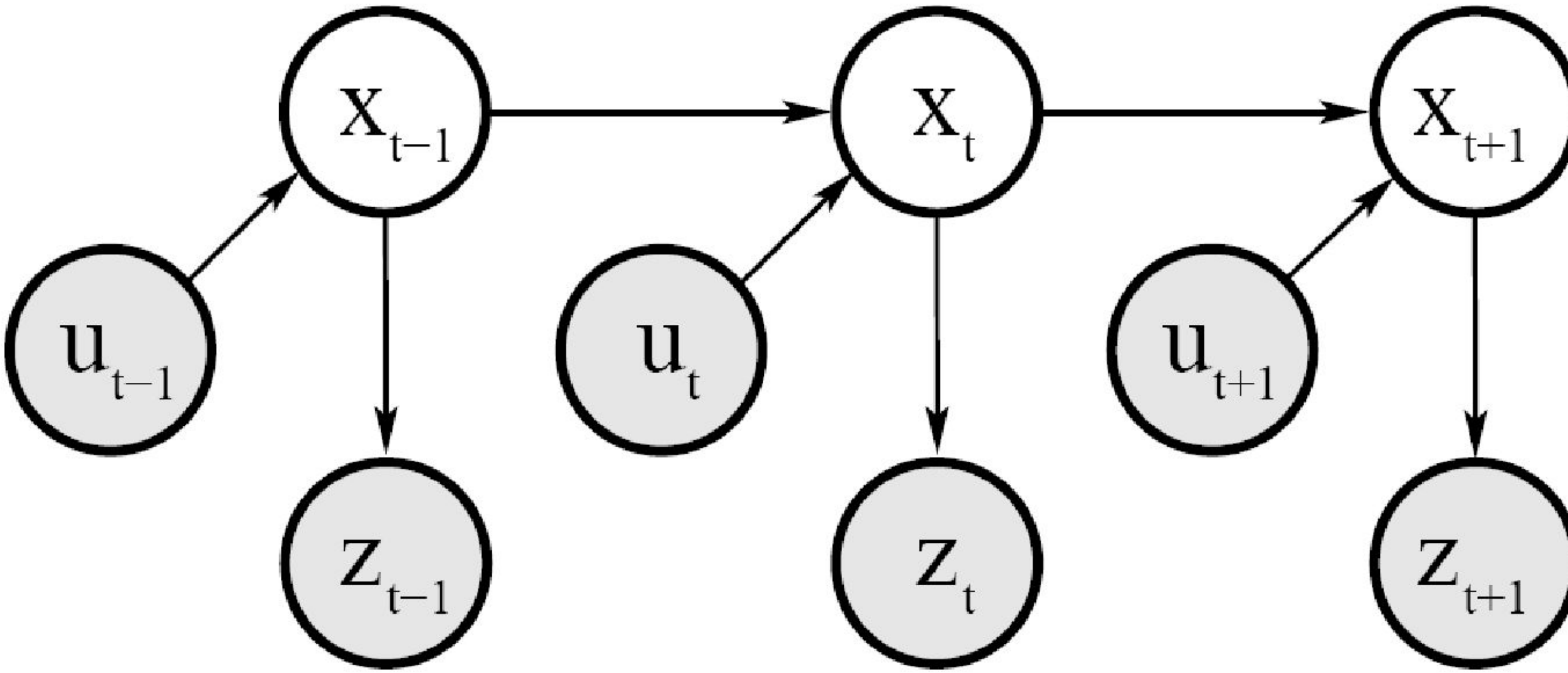
- Robot motion is inherently uncertain.

- How can we model this uncertainty?





Dynamic Bayesian Network for Controls, States, and Sensations



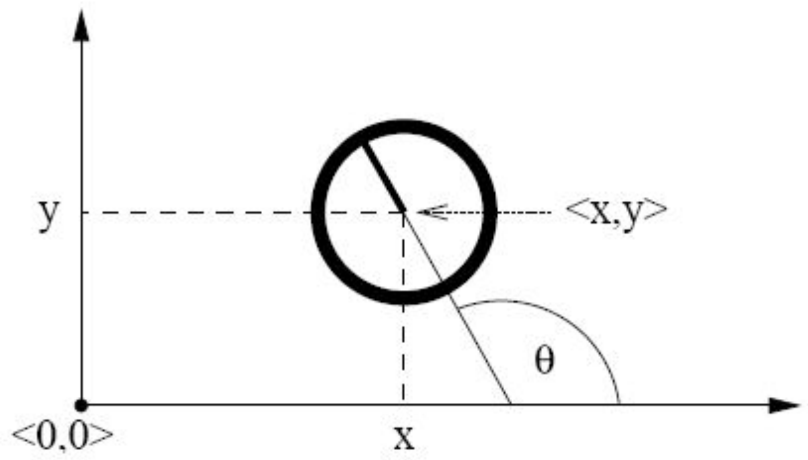
Probabilistic Motion Models

- To implement the Bayes Filter, we need the transition model $p(x \mid x', u)$.

- The term $p(x \mid x', u)$ is the posterior probability that action u carries the robot from x' to x.

- In this chapter we consider how $p(x \mid x', u)$ can be modeled based on the motion equations.

Coordinate Systems



- In general the configuration of a robot can be described by six parameters.
- Three-dimensional Cartesian coordinates plus three Euler angles pitch, roll, and tilt.
- We will consider robots operating on a planar surface.
- State space of such systems is 3D (x,y,θ).

Typical Motion Models

- Two types of motion models are typically considered:
  - **Odometry-based**
  - **Velocity-based** (**dead reckoning**)

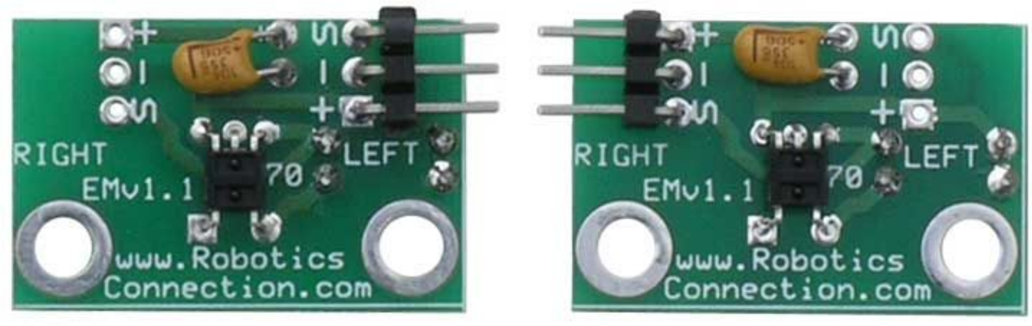
- Odometry-based models are used when systems are equipped with wheel encoders.

- 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 require +5V and GND to power them, and provide a 0 to 5V output. They provide +5V output when they "see" white, and a 0V output when they "see" black.



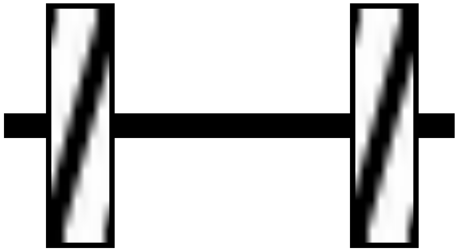
These disks are manufactured out of high quality laminated color plastic to offer a very crisp black to white transition. This enables a wheel encoder sensor to easily see the transitions.

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

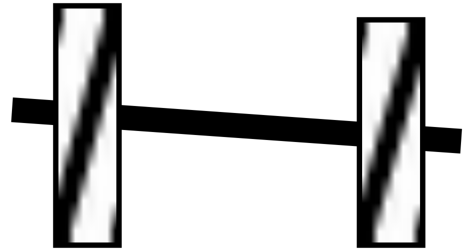
Dead Reckoning

- Term derived from “deduced reckoning.”
- Mathematical procedure for determining the present location of a vehicle.
- Calculate the current pose of the vehicle based on its velocities and the time elapsed.

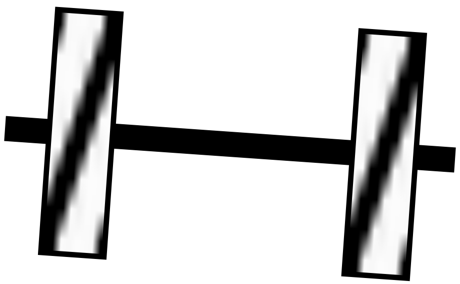
Reasons for Motion Errors



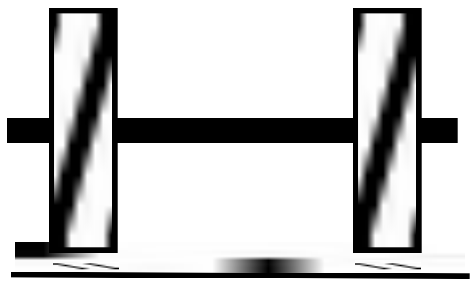
ideal case



different wheel
diameters



bump



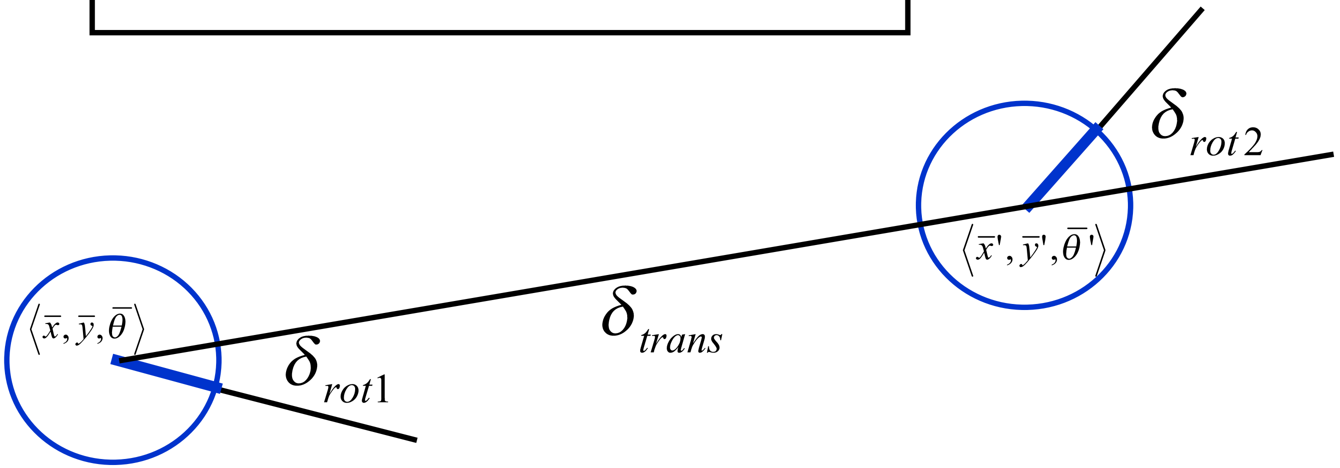
carpet

and many more …

Odometry Model

- Robot moves from $\bar{x}_{t-1} = \langle \bar{x}, \bar{y}, \bar{\theta} \rangle$ to $\bar{x}_t = \langle \bar{x}', \bar{y}', \bar{\theta}' \rangle$ (internal coordinates).
- Compute exact odometry parameters: $\langle \delta_{rot1}, \delta_{rot2}, \delta_{trans} \rangle$

$$\delta_{trans} = \sqrt{(\bar{x}' - \bar{x})^2 + (\bar{y}' - \bar{y})^2}$$
$$\delta_{rot1} = \operatorname{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$$
$$\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$$



The atan2 Function

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

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

Noise Model for Odometry

- Computed motion is given by the true motion corrupted with noise.

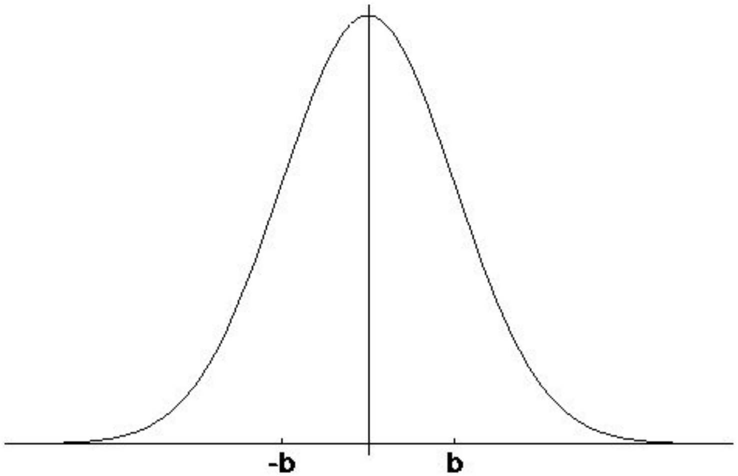
$$\hat{\delta}_{rot1} = \delta_{rot1} + \varepsilon_{\alpha_1 \delta^2_{rot1} + \alpha_2 \delta^2_{trans}}$$

$$\hat{\delta}_{trans} = \delta_{trans} + \varepsilon_{\alpha_3 \delta^2_{trans} + \alpha_4 \delta^2_{rot1} + \alpha_4 \delta^2_{rot2}}$$

$$\hat{\delta}_{rot2} = \delta_{rot2} + \varepsilon_{\alpha_1 \delta^2_{rot2} + \alpha_2 \delta^2_{trans}}$$

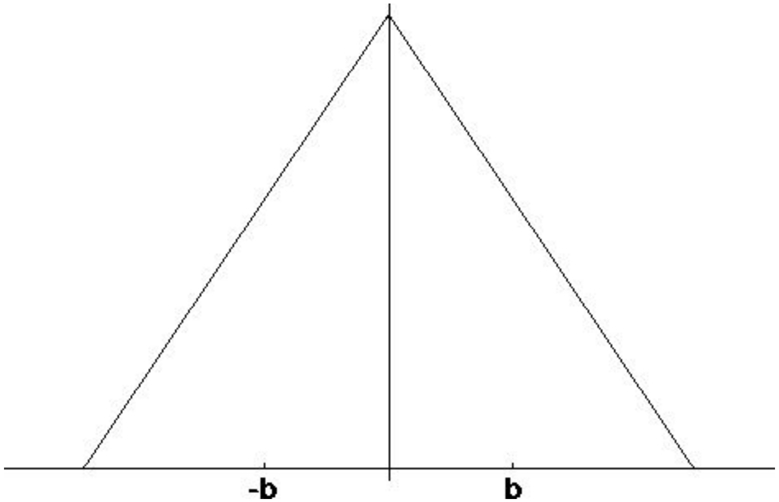
Typical Distributions for Probabilistic Motion Models

Normal distribution



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

Triangular distribution



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

Calculating the Probability of argument 'a'

- For a normal distribution:

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

Algorithm to compute $p(x_t | u_t, x_{t-1})$

- Compute odometry parameters from u_t in internal coordinates.
- Compute displacement based on desired transition from $x_{t-1} = \langle x \quad y \quad \theta \rangle$ to $x_t = \langle x' \quad y' \quad \theta' \rangle$
- Compute probability of desired state transition by matching measured odometry with desired displacement.

Calculating the Posterior Given x_t, x_{t-1}, and u_t

1. Algorithm **motion_model_odometry(x_t, u_t, x_{t-1})**
2. $\delta_{trans} = \sqrt{(\bar{x}' - \bar{x})^2 + (\bar{y}' - \bar{y})^2}$
3. $\delta_{rot1} = \operatorname{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$
4. $\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$
5. $\hat{\delta}_{trans} = \sqrt{(x - x')^2 + (y - y')^2}$
6. $\hat{\delta}_{rot1} = \operatorname{atan2}(y' - y, x' - x) - \theta$
7. $\hat{\delta}_{rot2} = \theta' - \theta - \hat{\delta}_{rot1}$
8. $p_1 = \operatorname{prob}(\delta_{trans} - \hat{\delta}_{trans}, \alpha_3 \hat{\delta}^2{}_{trans} + \alpha_4 \hat{\delta}^2{}_{rot1} + \alpha_4 \hat{\delta}^2{}_{rot2})$
9. $p_2 = \operatorname{prob}(\delta_{rot1} - \hat{\delta}_{rot1}, \alpha_1 \hat{\delta}^2{}_{rot1} + \alpha_2 \hat{\delta}^2{}_{trans})$
10. $p_3 = \operatorname{prob}(\delta_{rot2} - \hat{\delta}_{rot2}, \alpha_1 \hat{\delta}^2{}_{rot2} + \alpha_2 \hat{\delta}^2{}_{trans})$
11. return $p_1 \cdot p_2 \cdot p_3$

compute odometry (u)

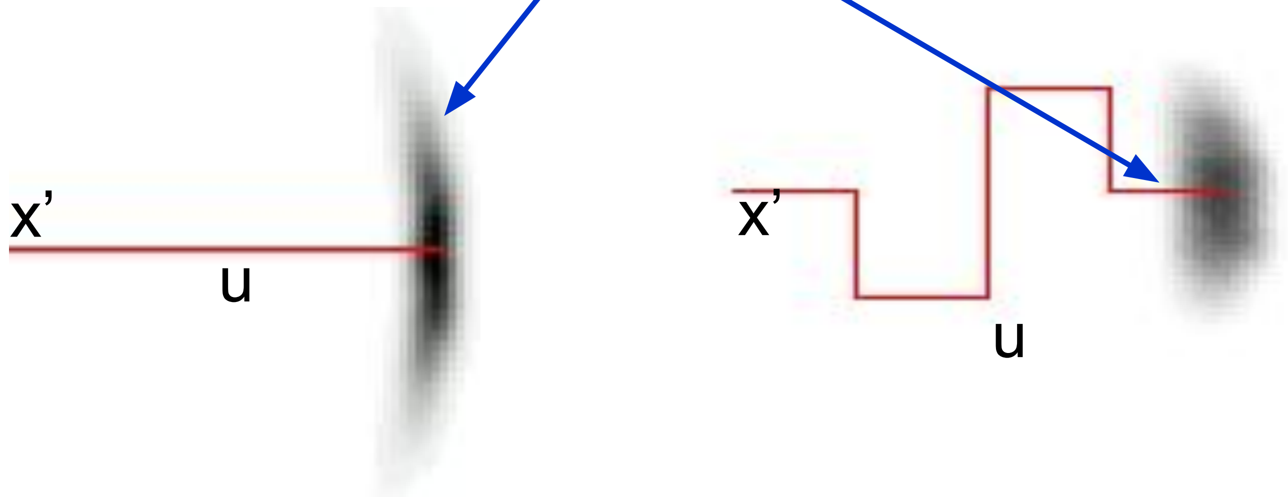
displacement based on state transition (x_t,x_{t-1})

probability of state transition

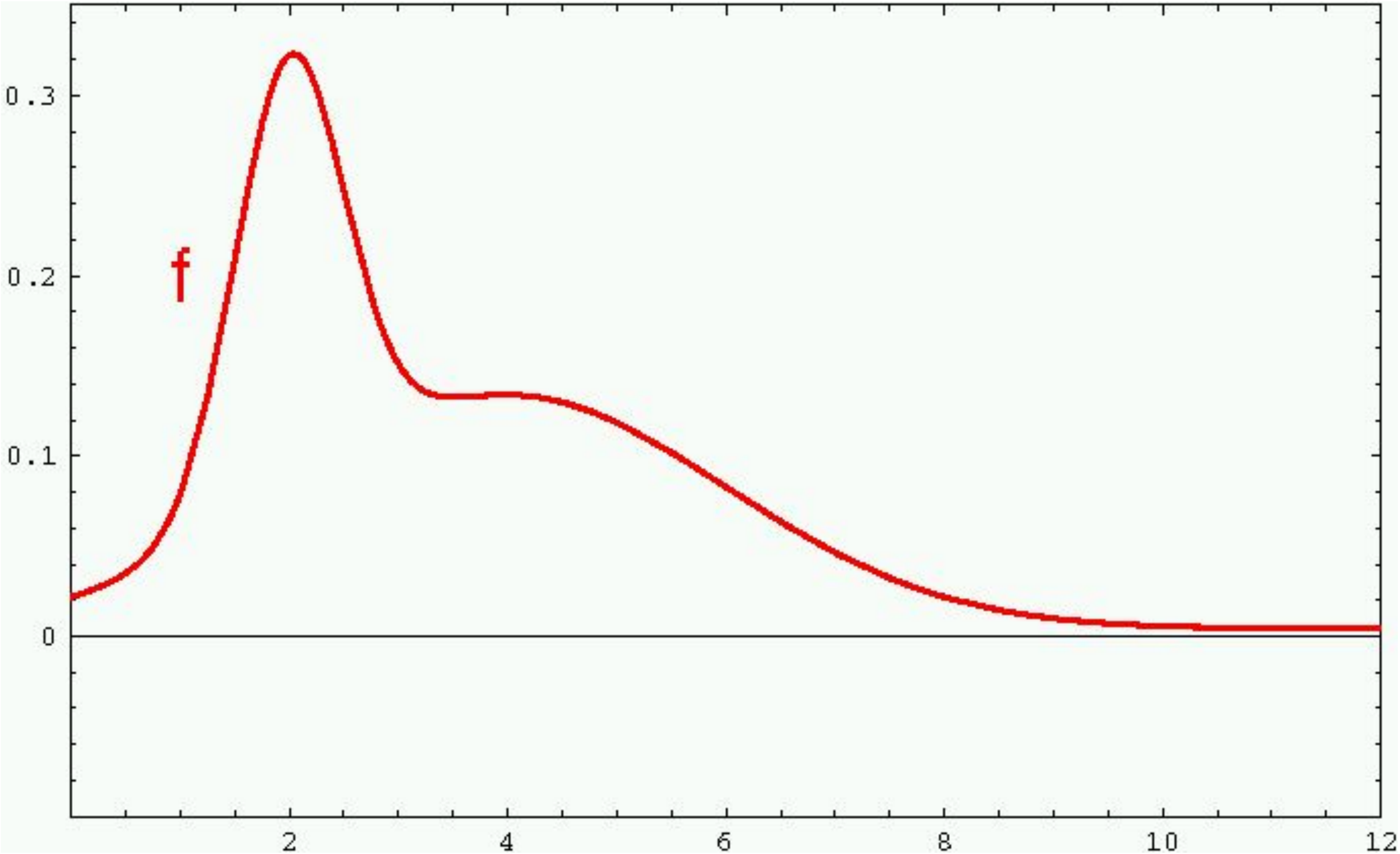
Application

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

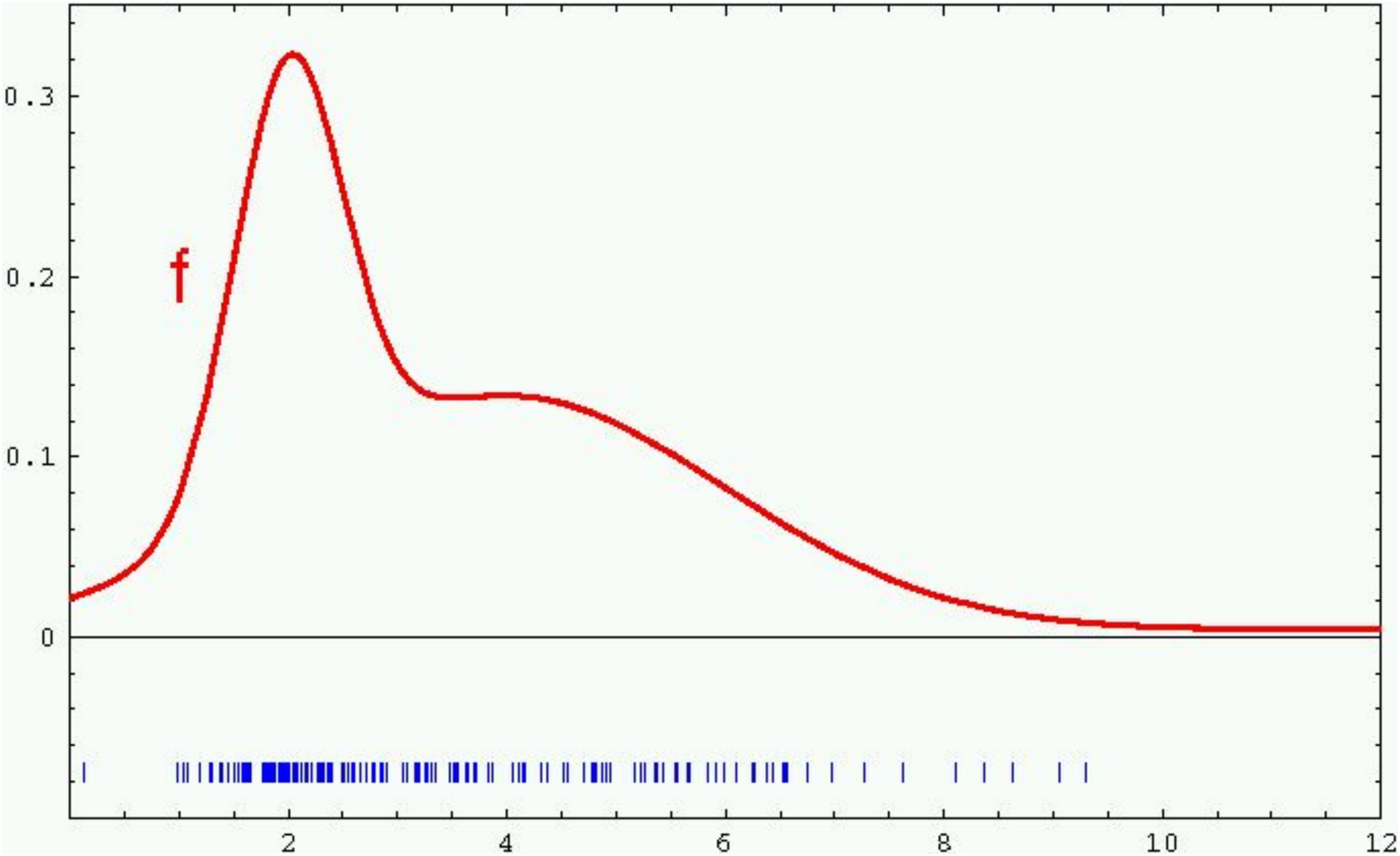
$$p(x_t|u_t,x_{t-1})$$



Sample-based Density Representation



Sample-based Density Representation



How to Sample from Normal or Triangular Distributions?

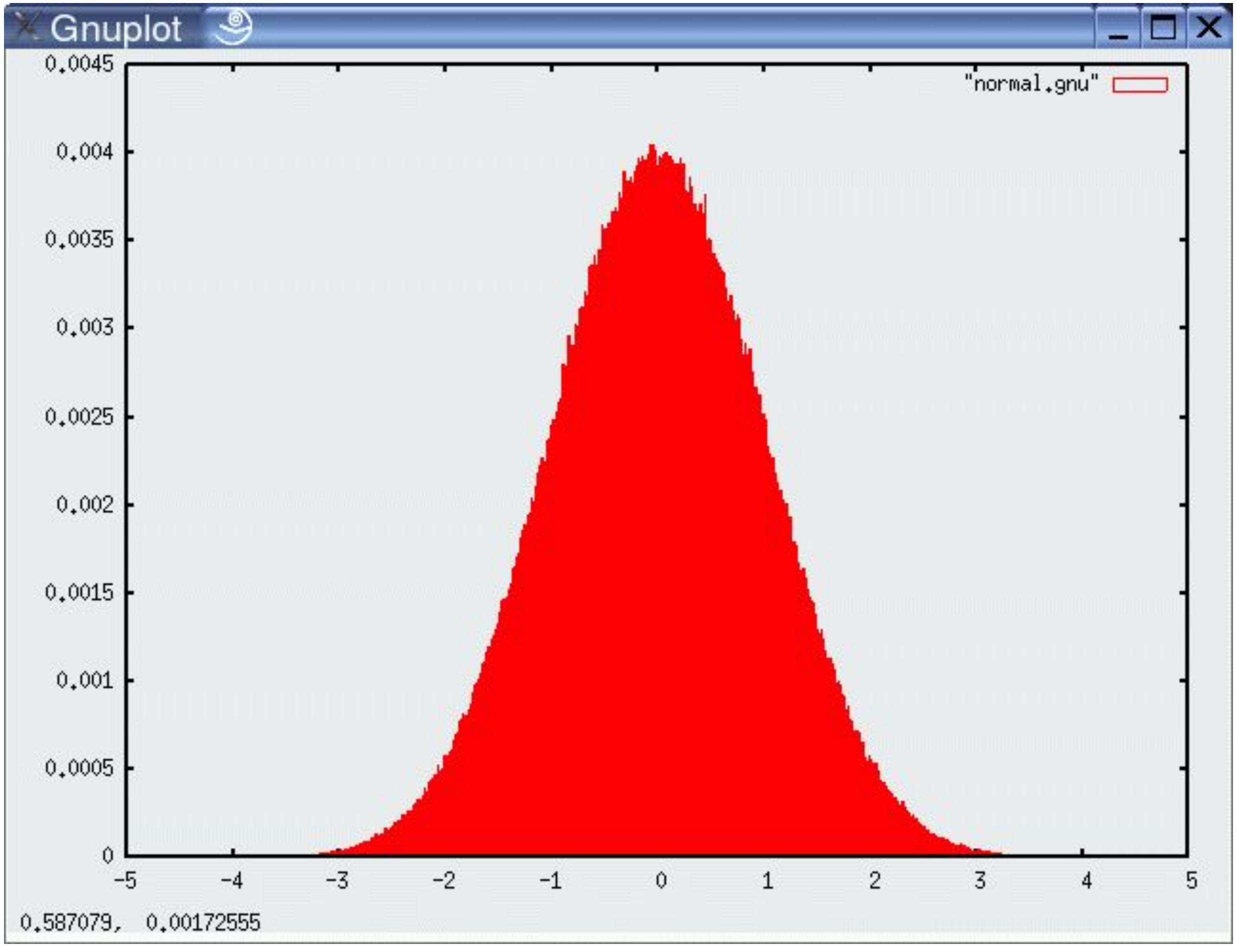
- Sampling from a normal distribution

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

- Sampling from a triangular distribution

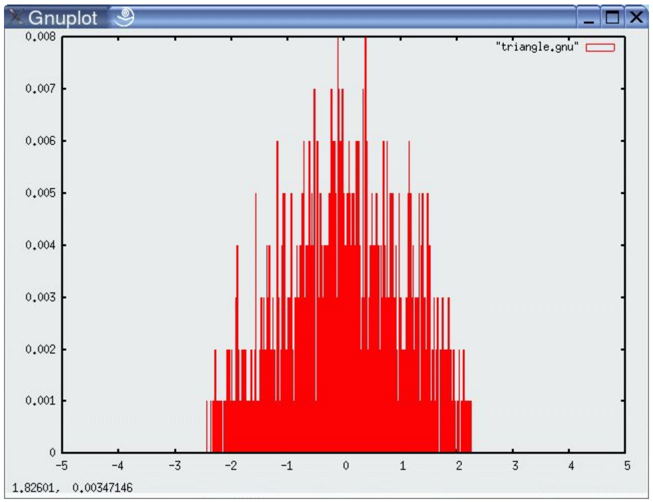
1. Algorithm **sample_triangular_distribution**(b^2):
2. return $\frac{\sqrt{6}}{2}\left[\mathbf{rand}(-b, b) + \mathbf{rand}(-b, b)\right]$

Normally Distributed Samples

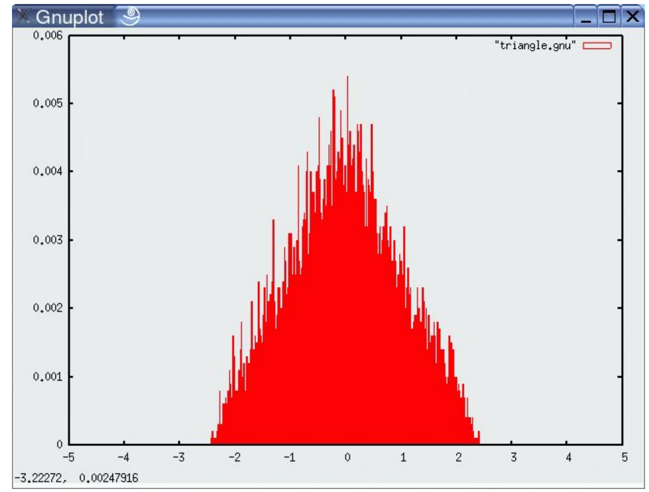


10^6
samples

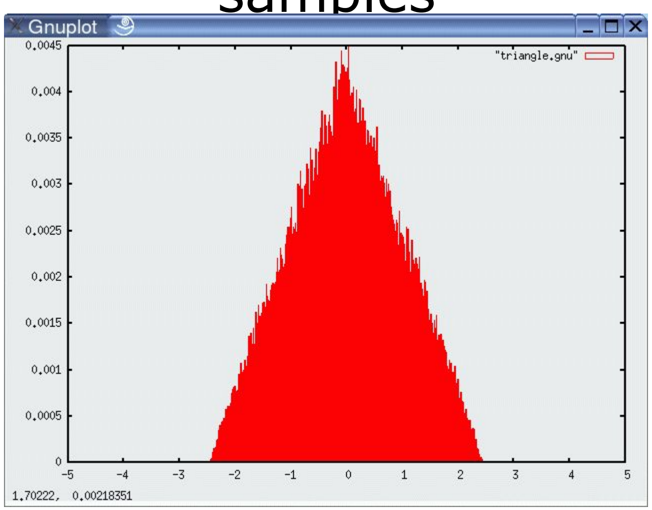
For Triangular Distribution



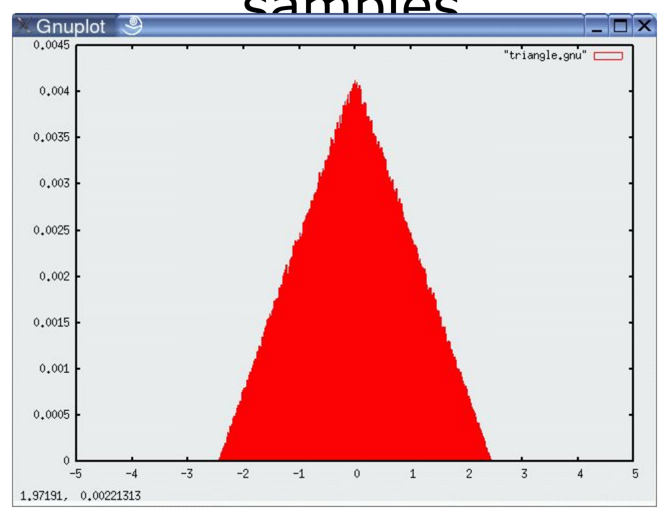
10^3 samples



10^4 samples



10^5 samples



10^6 samples

Rejection Sampling

- Sampling from arbitrary distributions:

Algorithm **sample_distribution**(f,b^2):

repeat

$x = \mathbf{rand}(-b, b)$

$y = \mathbf{rand}(0, \max\{f(x) \mid x \in (-b, b)\})$

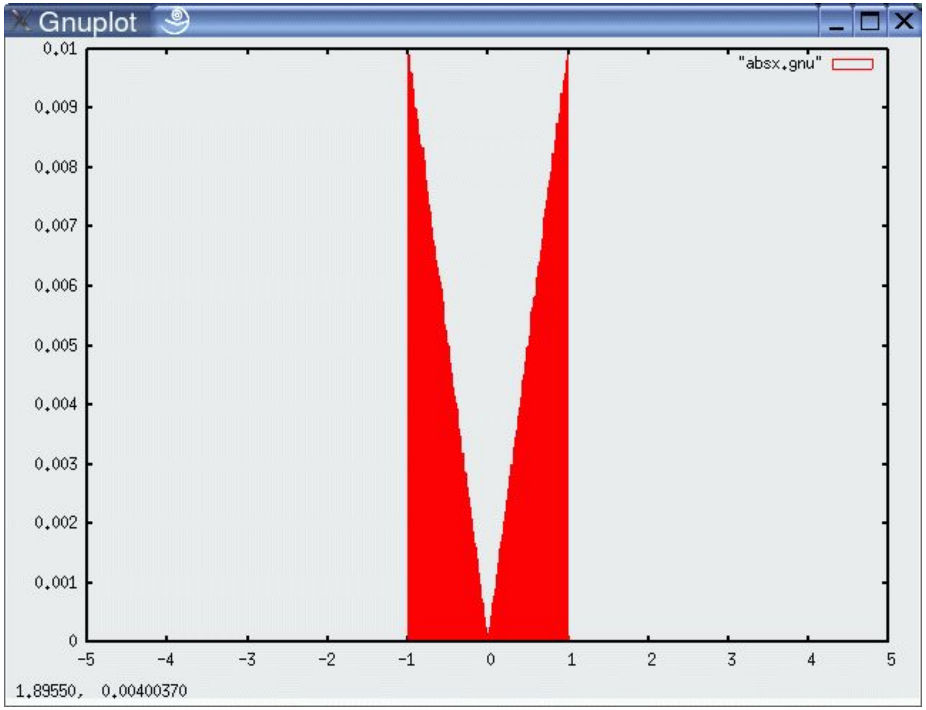
until $y \leq f(x)$

return x

Example

- Sampling from:

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



Algorithm to sample from $p(x_t | u_t, x_{t-1})$

- Compute odometry parameters $\langle \delta_{rot1}, \delta_{rot2}, \delta_{trans} \rangle$
- Compute noisy odometry parameters as odometry parameters + sample from noise distribution:

$$\left\langle \hat{\delta}_{rot1}, \hat{\delta}_{rot2}, \hat{\delta}_{trans} \right\rangle$$

- Compute new sample pose as previous sample pose + noisy displacement.

Sample Odometry Motion Model

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

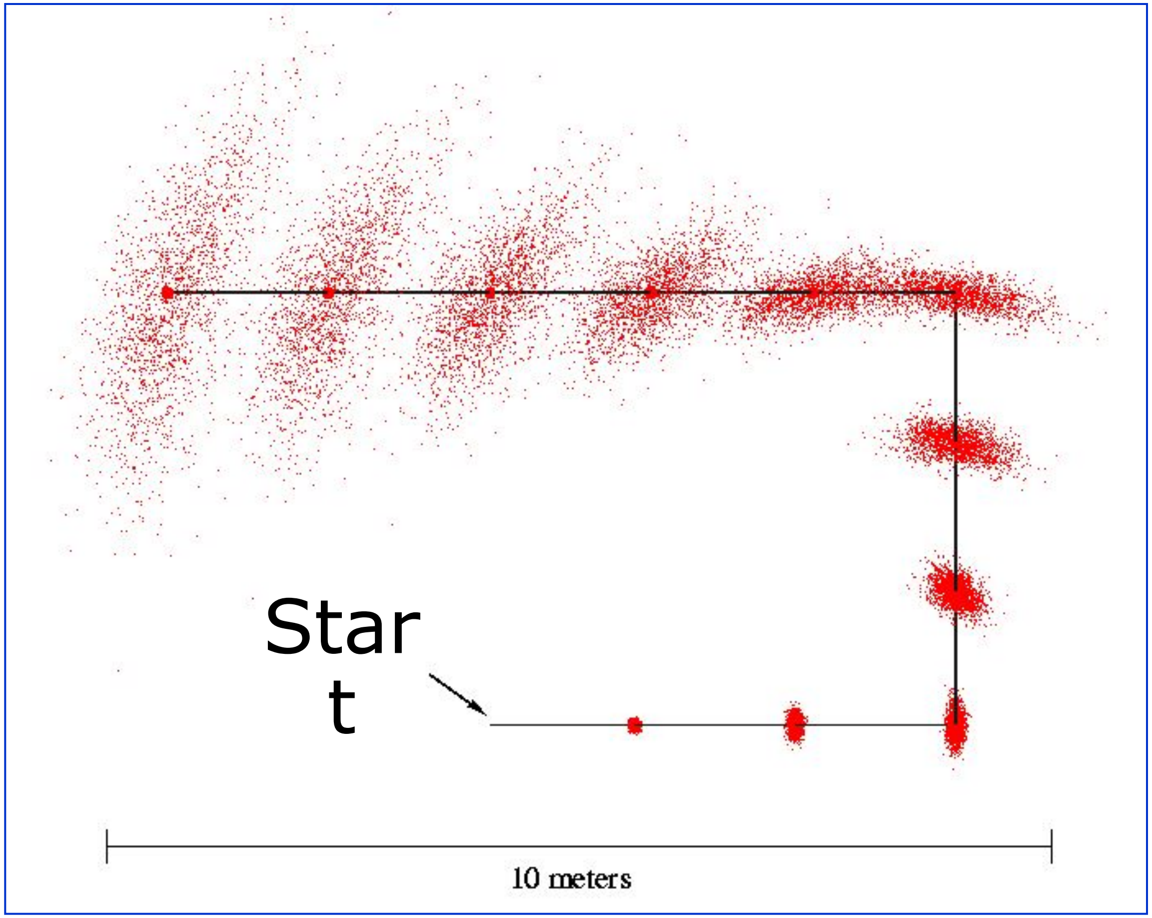
$$u_t = \langle \delta_{rot1}, \delta_{rot2}, \delta_{trans} \rangle, x_{t-1} = \langle x, y, \theta \rangle$$

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

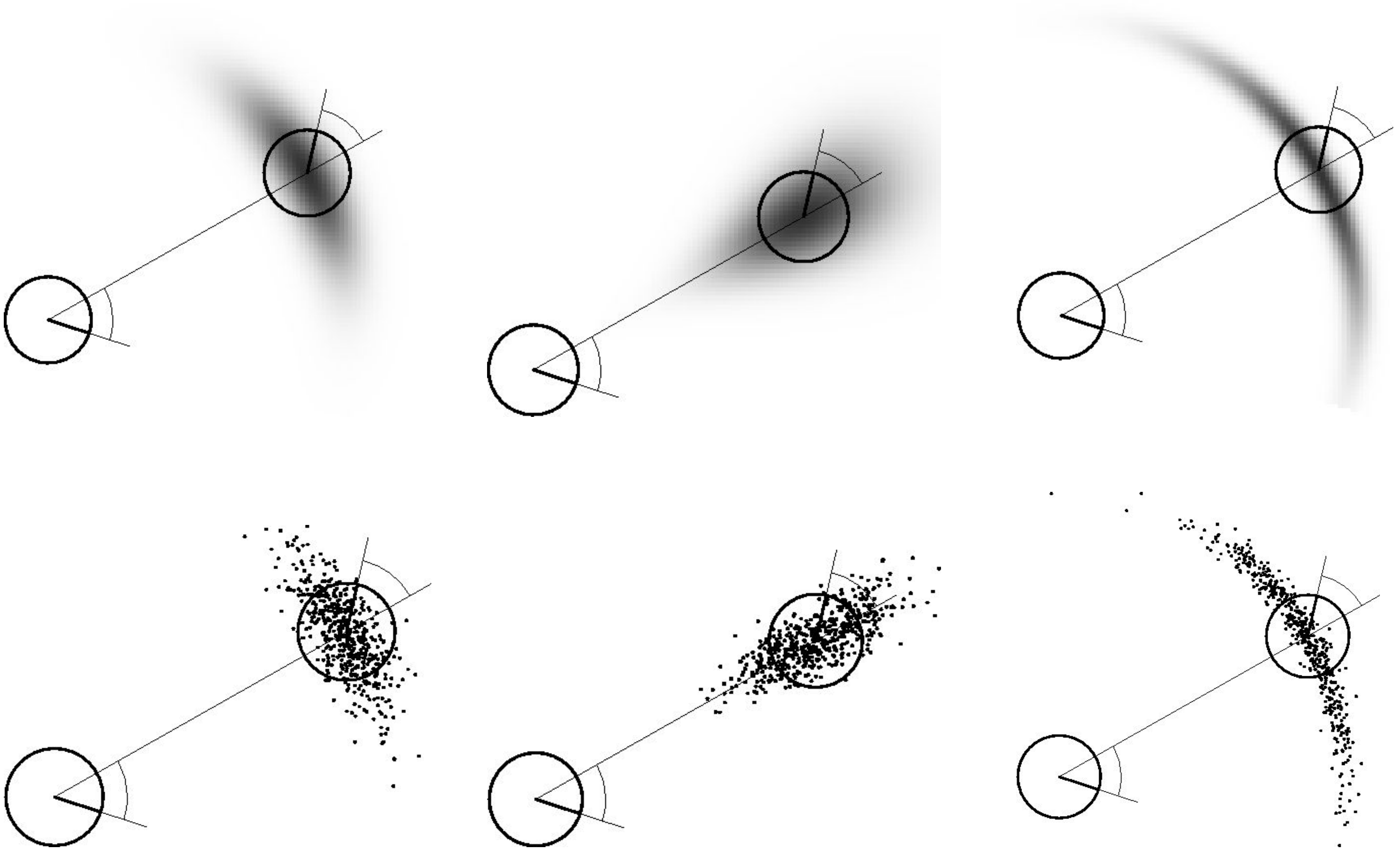
sample_normal_distribution

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 $x_t = \langle x', y', \theta' \rangle^T$

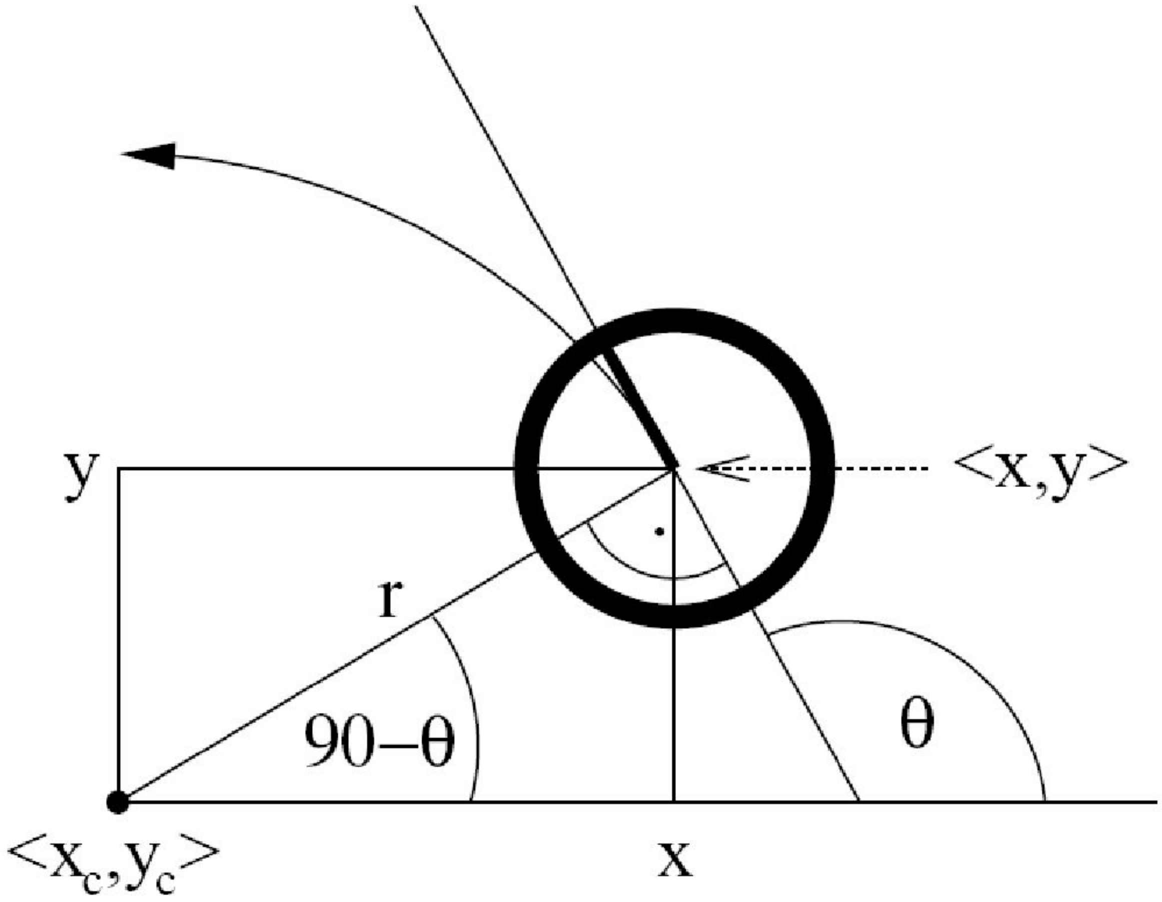
Sampling from Motion Model



Examples (Odometry-Based)



Velocity-Based Model



Exact Motion Model

- Control as translational and rotational velocities: $u_t = \begin{pmatrix} v_t \\ \omega_t \end{pmatrix}$
- Take as input the initial pose: $x_{t-1} = \langle x \quad y \quad \theta \rangle$
  control signal u_t and successor pose: $x_t = \langle x' \quad y' \quad \theta' \rangle$
- Compute probability: $p(x_t \mid u_t, x_{t-1})$
- Velocity measurements are true values + added noise.
- Derive in noise free case; motion on a circle with: $r = \dfrac{v}{\omega}$
- Velocities fixed in the time interval of one step.

Equation for the Velocity Model

- Derivation of exact motion (using basic trigonometry) in Section 5.3.3.
- Also see derivation for real motion.
- Compute probability of specific state transitions.
- Can also estimate current state given previous state and control signal.

Probability for Velocity Model: $p(x_t|x_{t-1}, u_t)$

$$\textbf{Algorithm motion\_model\_velocity}(x_t, u_t, x_{t-1})$$

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

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

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

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

$$\hat{\upsilon} = \frac{\Delta\theta}{\Delta t} r^*, \quad \hat{\omega} = \frac{\Delta\theta}{\Delta t}, \quad \hat{\gamma} = \frac{\theta' - \theta}{\Delta t} - \hat{\omega}$$

$$\text{return} \quad \text{prob}(\upsilon - \hat{\upsilon}, \alpha_1\upsilon^2 + \alpha_2\omega^2).\text{prob}(\omega - \hat{\omega}, \alpha_3\upsilon^2 + \alpha_4\omega^2).$$
$$\text{prob}(\hat{\gamma}, \alpha_5\upsilon^2 + \alpha_6\omega^2)$$

Sampling from Velocity Model to obtain $\mathbf{x}_t$

$$\textbf{Algorithm sample\_motion\_model\_velocity}(u_t, x_{t-1})$$

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

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

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

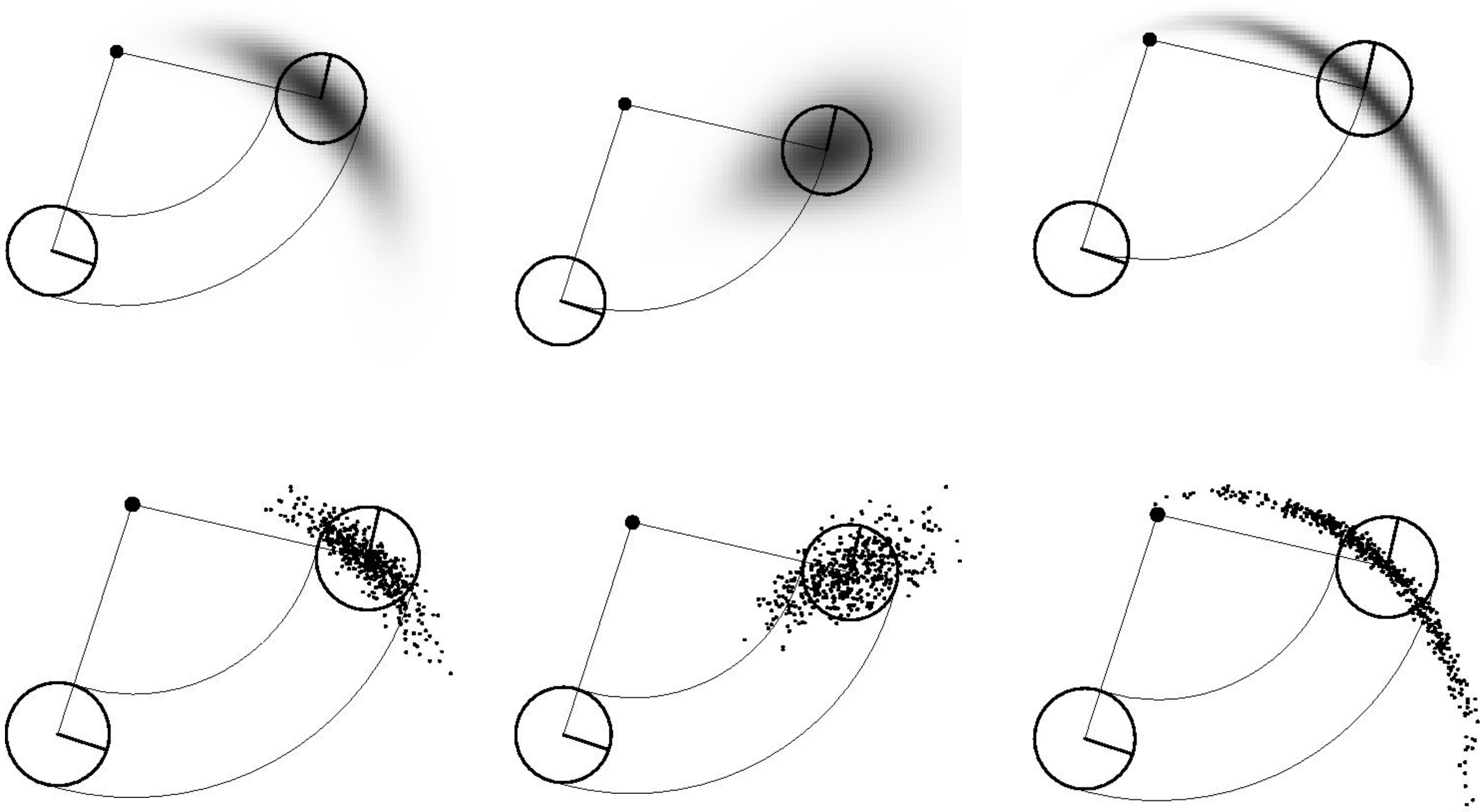
$$x' = x - \frac{\hat{\upsilon}}{\hat{\omega}} \sin\theta + \frac{\hat{\upsilon}}{\hat{\omega}} \sin(\theta + \hat{\omega}\Delta t)$$

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

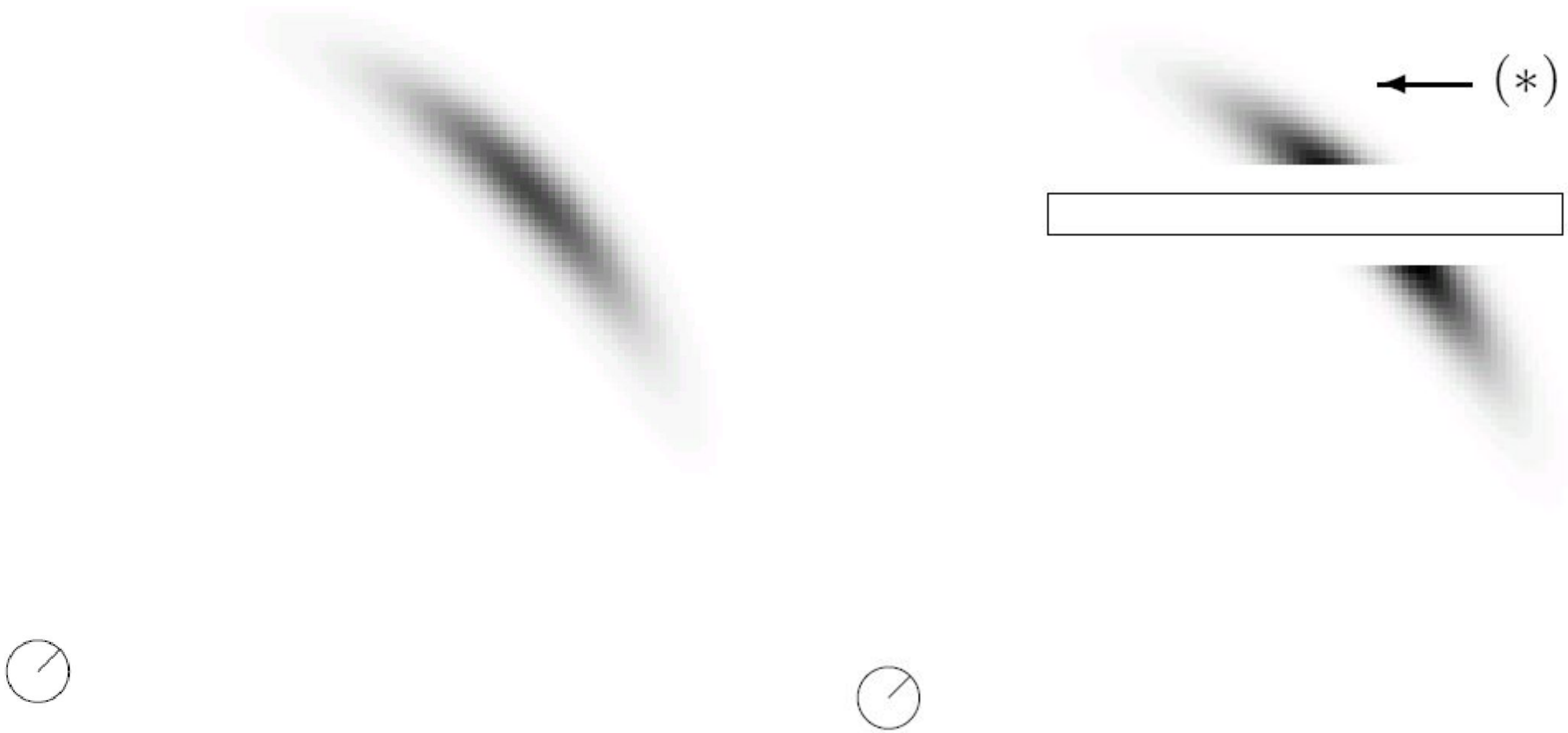
$$\theta' = \theta + \hat{\omega}\Delta t + \hat{\gamma}\Delta t$$

$$\text{return} \quad x_t = (x', y', \theta')^T$$

Examples (velocity based)



Map-Consistent Motion Model



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

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

Optional tasks…

- Derive motion model equations:
  - Without noise (exact motion).
  - With noise (real motion).
  - Section 5.3.3, PR.
  - Section 5.4.3, PR.

Summary

- Discussed motion models for odometry-based and velocity-based systems.
- Discussed ways to calculate posterior probability: $p(x | x', u)$.
- Described how to sample from $p(x | x', u)$.
- Typically calculations are done in fixed time intervals Δt.
- In practice, parameters of the models have to be learned.
- Discussed an extended motion model that takes the map into account.