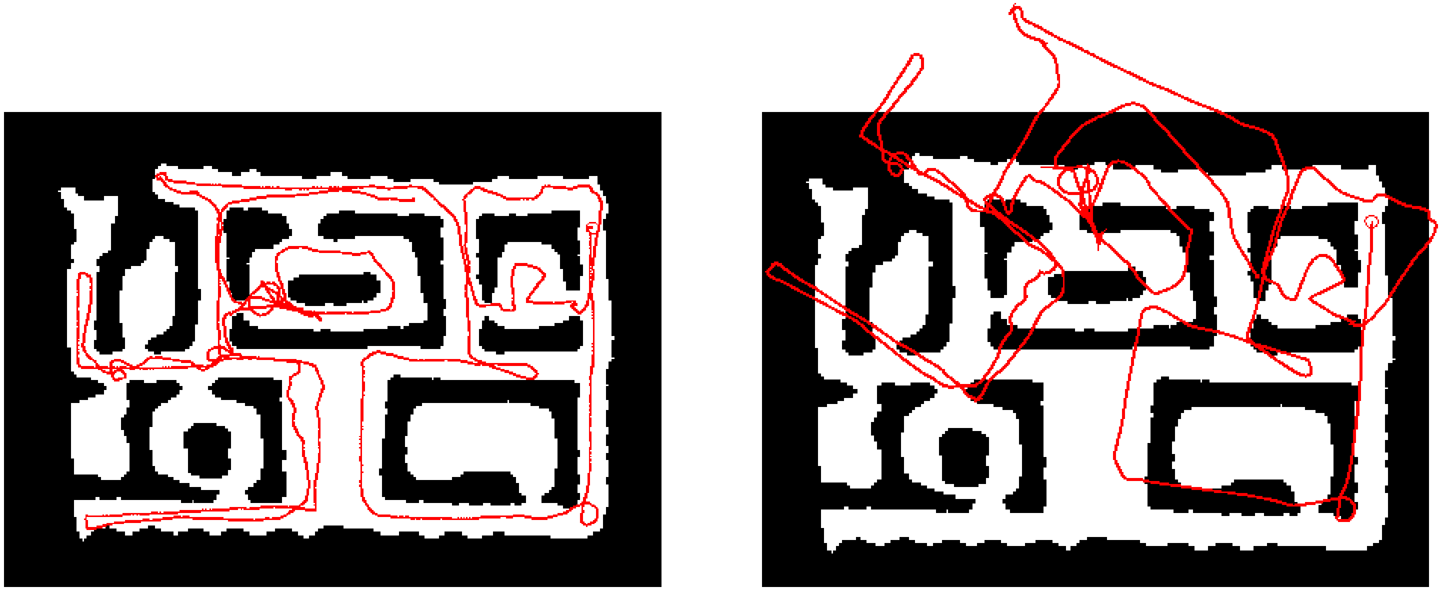
Introduction to Mobile Robotics

Probabilistic Motion Models

Daniel Büscher

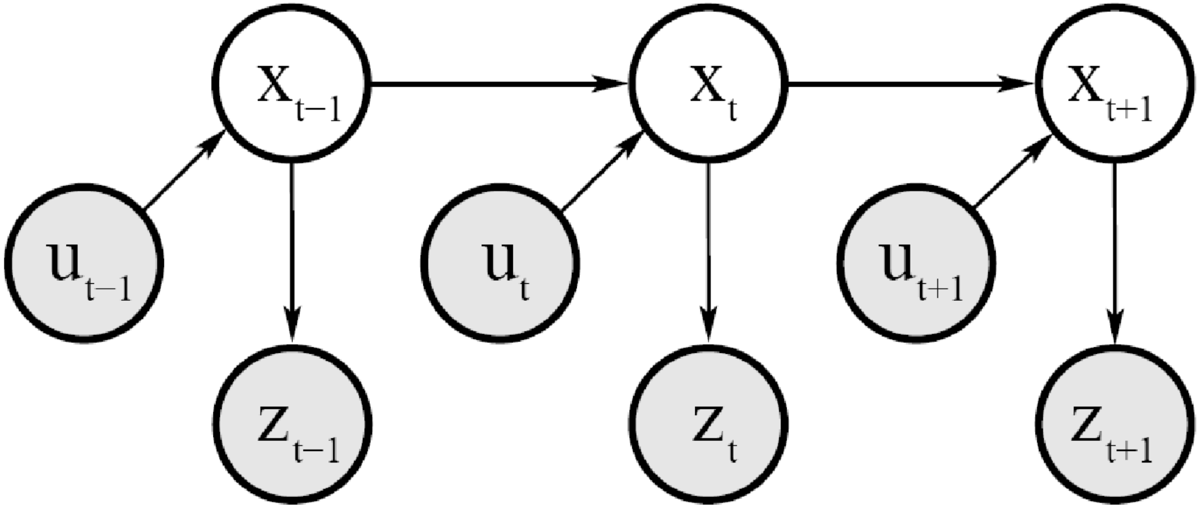
Robot Motion

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



Dynamic Bayesian Network

- Models dependencies of controls, states, and measurements

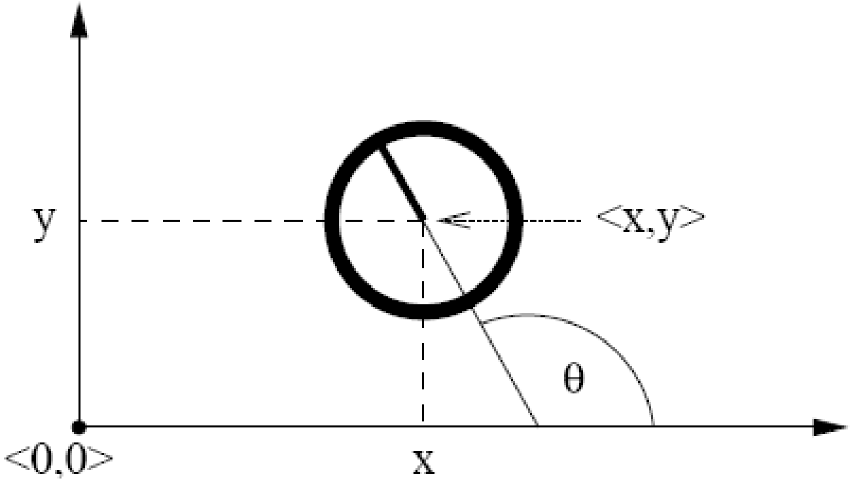


Probabilistic Motion Models

- To implement the Bayes Filter, we need the transition model $p(x_t \mid x_{t-1}, u_t)$
- It specifies a posterior probability that action u_t carries the robot from x_{t-1} to x_t
- In this section we will discuss how this “motion model” can be calculated using
  - the motion equations and
  - the uncertain outcome of the movements

Coordinate Systems

- The configuration of a wheeled robot in 3D can be described by six parameters:
  - three Cartesian coordinates x, y, z
  - three Euler angles for roll, pitch, and yaw
- For simplicity, we consider robots operating on a planar surface
- Reduced state space: three-dimensional (x, y, θ)



Typical Motion Models

- Odometry-based
  - Used if wheel encoders are available
  - Based on the measured wheel revolutions
  - Uncertainty from wheel slippage, …
- Velocity-based (“dead reckoning”)
  - Can be applied without wheel encoders
  - Typically based on a velocity control command
  - Additional uncertainty from actuation precision

Both calculate the new pose using the elapsed time

Content

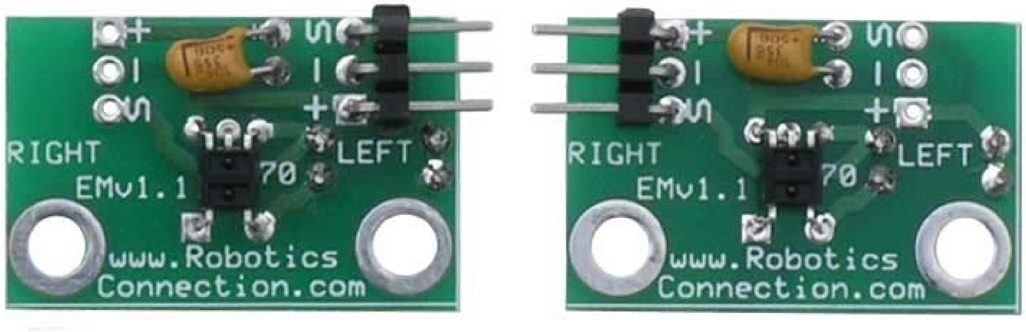
1. **Odometry model:** calculate posterior p(x’|x,u)
2. **Sampling:** draw an x’ according to the posterior
3. **Velocity-based model:** posterior and sampling
4. **Rejection sampling:** samples from arbitrary dist.
5. **Map-consistent motion:** considering obstacles

1. Odometry model

Example Wheel Encoders

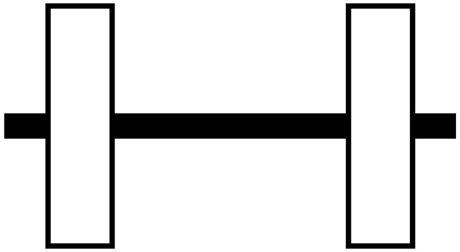
- Disks fixed to wheels
- Typically plastic with black/white transitions
- Enable wheel encoder sensors to easily detect transitions

Modules provide
- +5V output when they "see" white
- 0V output when they "see" black

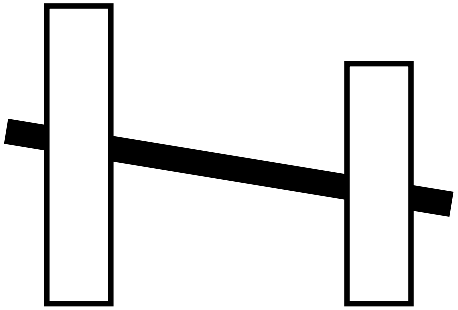


Source: www.active-robots.com

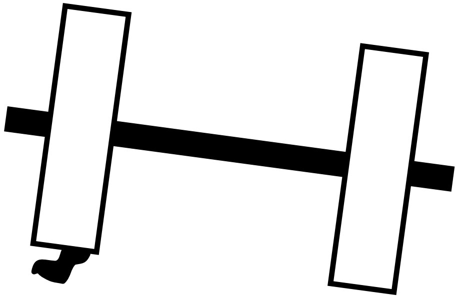
Typical Motion Errors



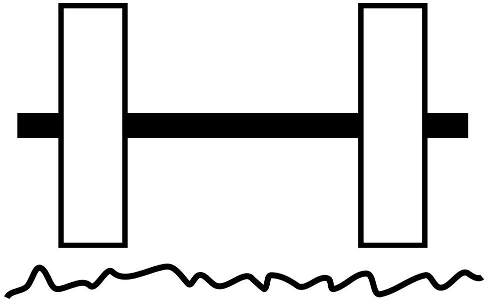
ideal case



different wheel diameters



bumps



carpets

and many more …

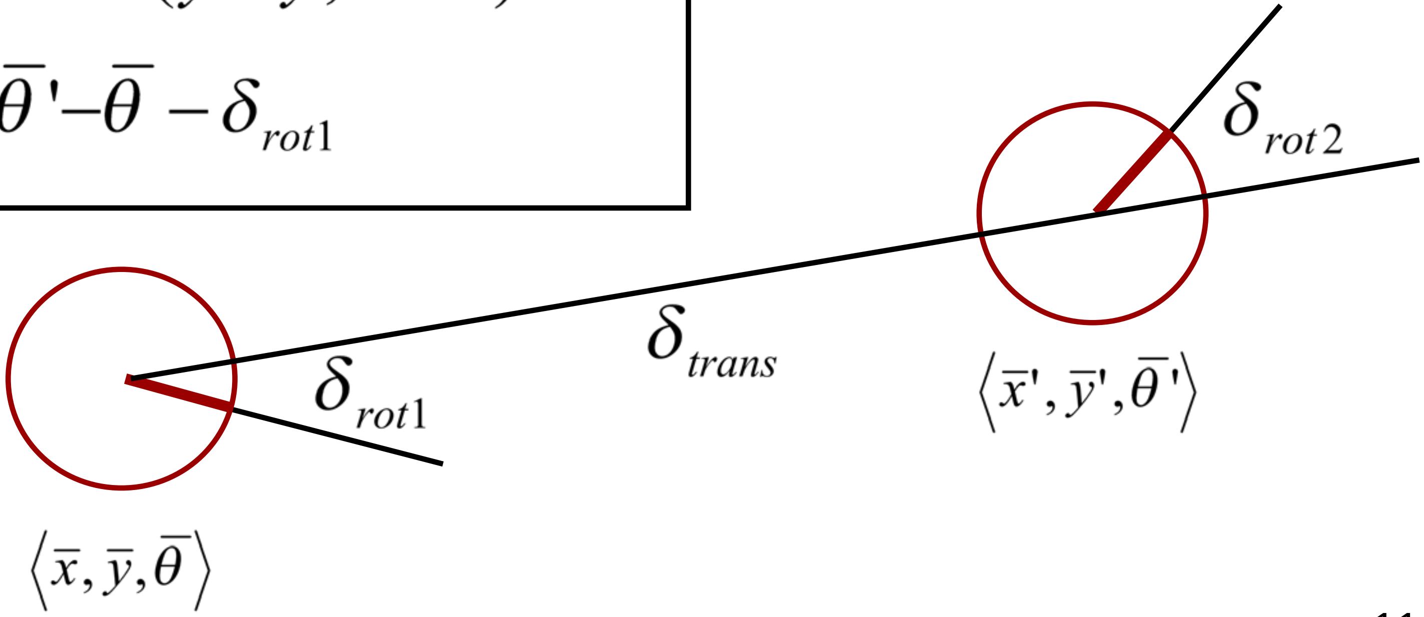
Odometry Model

- 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

- The tan function is periodic in (-π/2, π/2), hence its inverse (atan) covers only half a circle
- The atan2 function extends atan to the full circle
- Correctly copes with signs and zeros 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

- The measured motion is given by the true motion corrupted with noise
- Since we don't have access to the true motion, we can write

Hypothesis Measured Noise

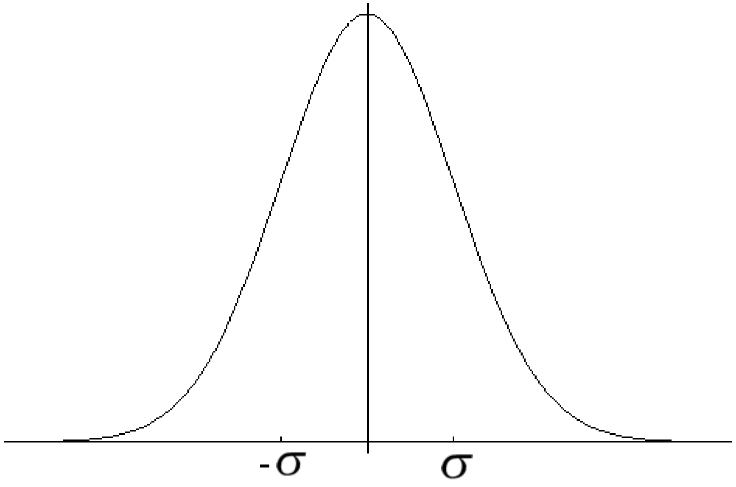
$$\hat{\delta}_{rot1} = \delta_{rot1} + \varepsilon_{\alpha_1 |\delta_{rot1}| + \alpha_2 |\delta_{trans}|}$$

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

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

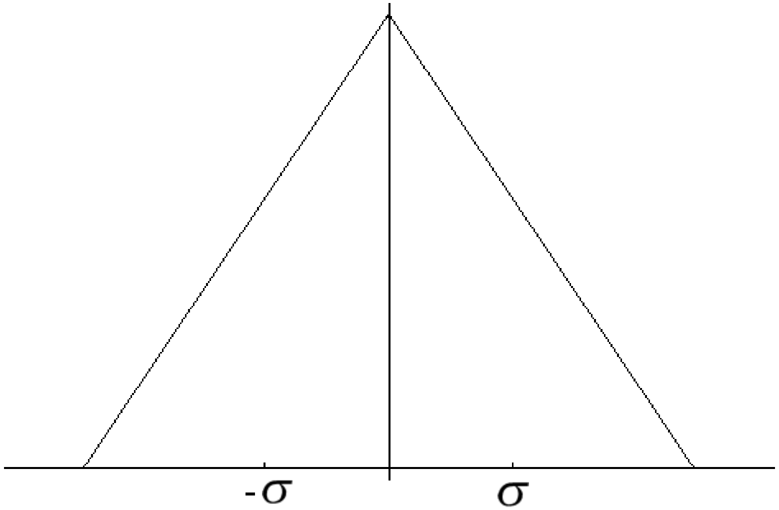
Typical Distributions for Probabilistic Motion Models

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 Densities (zero-centered)

- For a normal distribution

query point

```
1. Algorithm prob_normal_distribution($a$,$b$):
2. return $\frac{1}{\sqrt{2\pi\, b^2}} \exp\left\{-\frac{1}{2}\frac{a^2}{b^2}\right\}$
```

std. deviation

- For a triangular distribution

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

The Posterior p(x’|x, u)

```
Algorithm motion_model_odometry $(\boxed{x, x'}, \boxed{\bar{x}, \bar{x}'})$
$\delta_{trans} = \sqrt{(\bar{x}'-\bar{x})^2 + (\bar{y}'-\bar{y})^2}$
$\delta_{rot1} = \mathrm{atan2}(\bar{y}'-\bar{y}, \bar{x}'-\bar{x}) - \bar{\theta}$
$\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$
$\hat{\delta}_{trans} = \sqrt{(x'-x)^2 + (y'-y)^2}$
$\hat{\delta}_{rot1} = \mathrm{atan2}(y'-y, x'-x) - \theta$
$\hat{\delta}_{rot2} = \theta' - \theta - \hat{\delta}_{rot1}$
$p_1 = \mathrm{prob}(\delta_{\mathrm{rot1}} - \hat{\delta}_{\mathrm{rot1}}, \alpha_1 |\delta_{\mathrm{rot1}}| + \alpha_2 \delta_{\mathrm{trans}})$
$p_2 = \mathrm{prob}(\delta_{\mathrm{trans}} - \hat{\delta}_{\mathrm{trans}}, \alpha_3 \delta_{\mathrm{trans}} + \alpha_4 (|\delta_{\mathrm{rot1}}| + |\delta_{\mathrm{rot2}}|))$
$p_3 = \mathrm{prob}(\delta_{\mathrm{rot2}} - \hat{\delta}_{\mathrm{rot2}}, \alpha_1 |\delta_{\mathrm{rot2}}| + \alpha_2 \delta_{\mathrm{trans}})$
return $p_1 \cdot p_2 \cdot p_3$
```

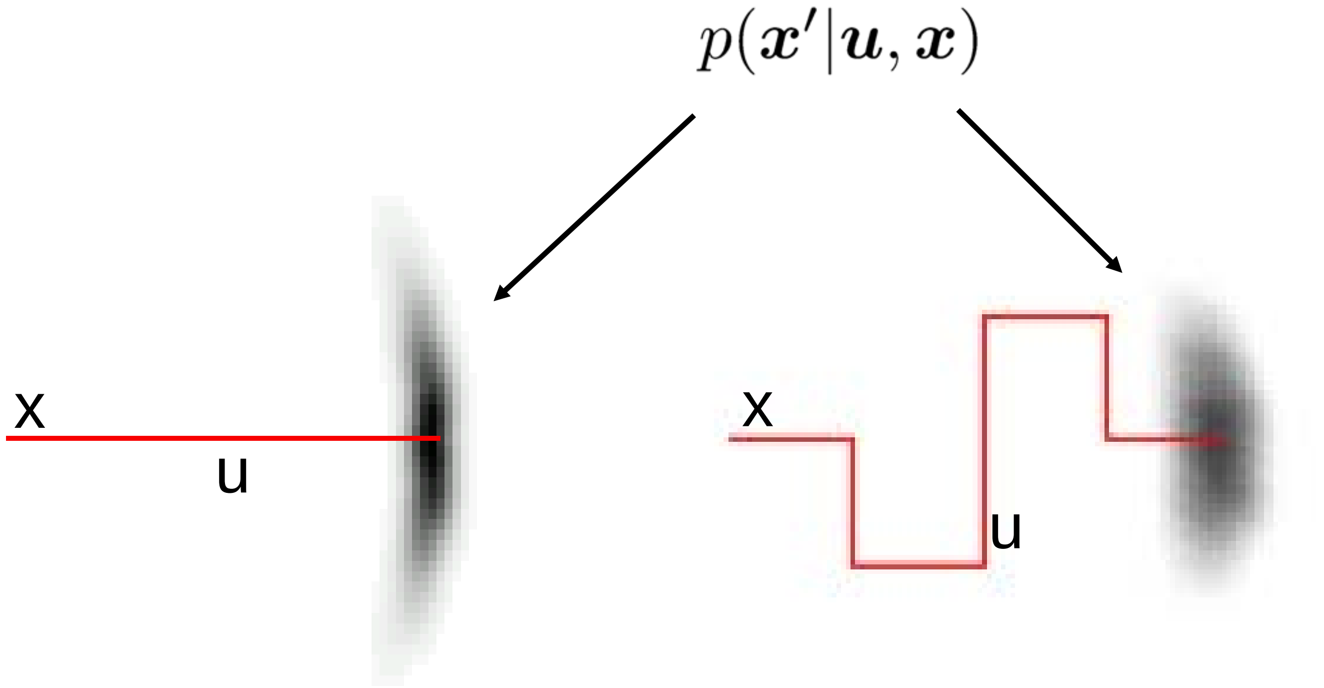
odometry (*u*)

values of interest / hypotheses (*x*,*x’*)

prop_normal_distribution

Typical outcome

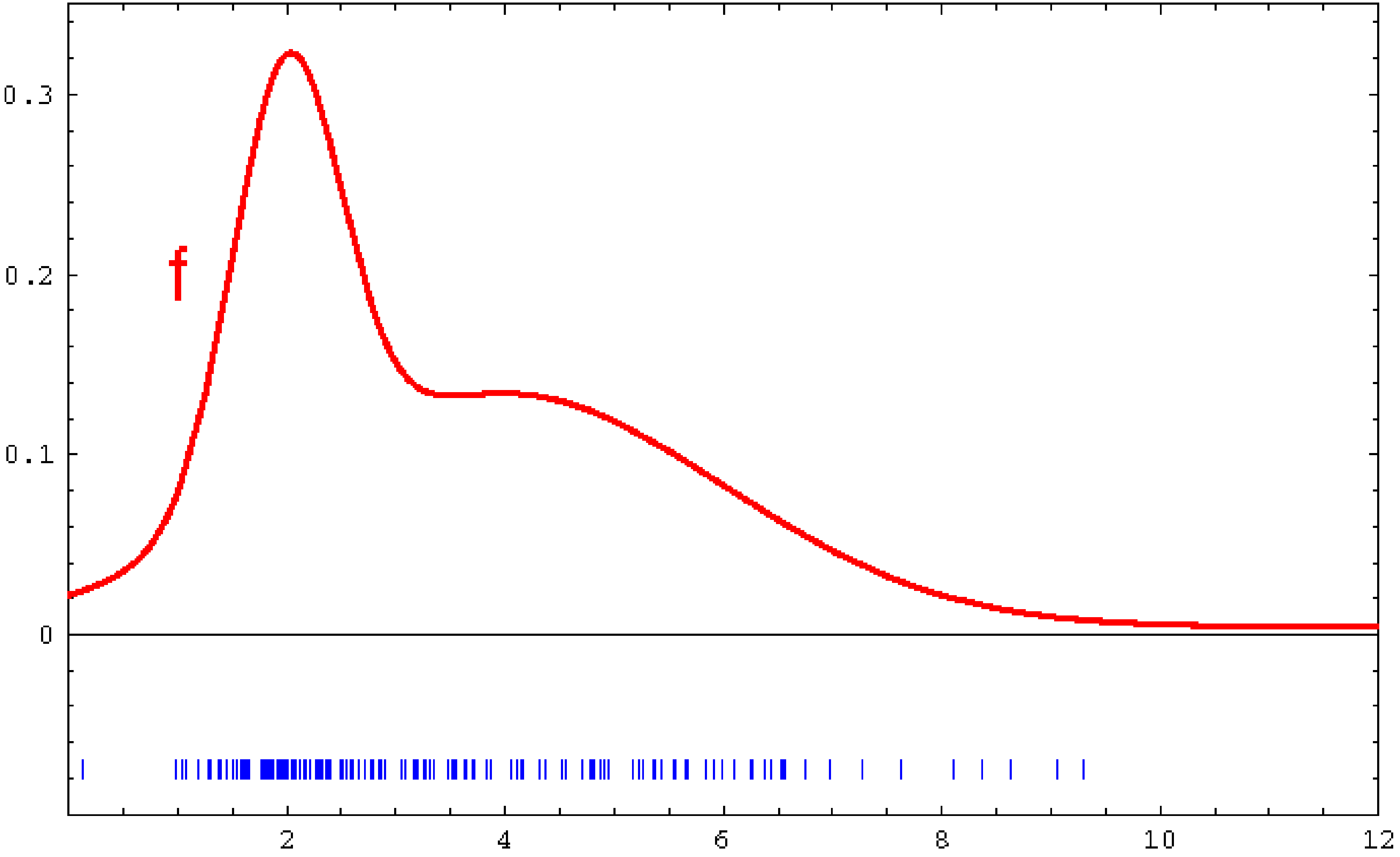
- Repeated application of the motion model for short movements:
- Banana-shaped distributions for the 2d-projection of the 3d posterior



$$p(\boldsymbol{x}' | \boldsymbol{u}, \boldsymbol{x})$$

2. Sampling

Sample-Based Density Representation

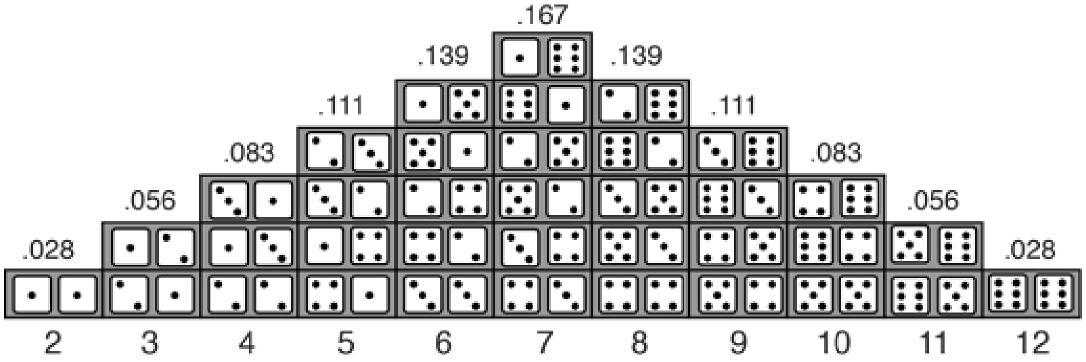


Sampling from a triangular dist.

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

Uniform distribution

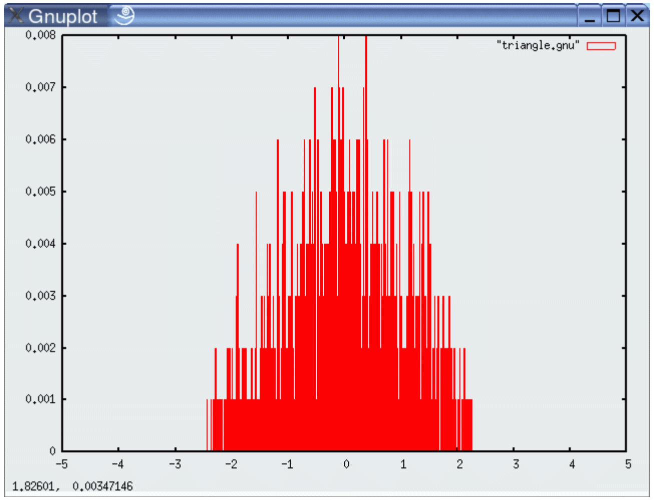
Common example: sum of two dice



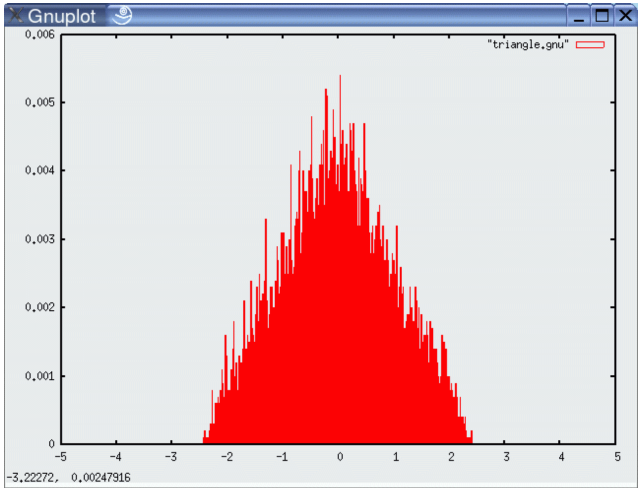
Total number of microstates: 36

http://hyperphysics.phy-astr.gsu.edu/hbase/Math/dice.html

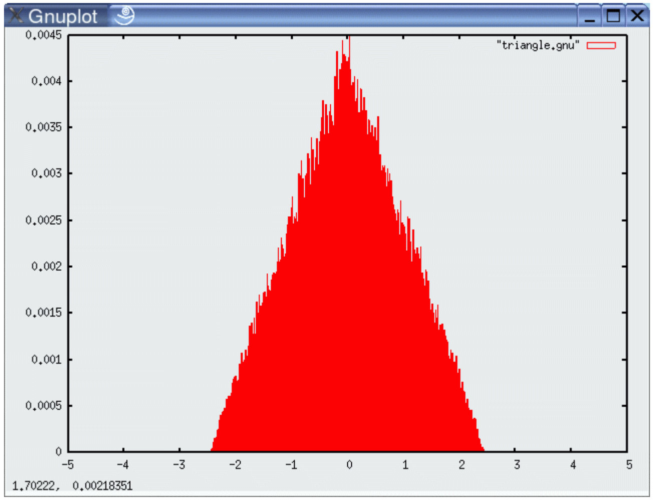
Triangular Distributed Samples



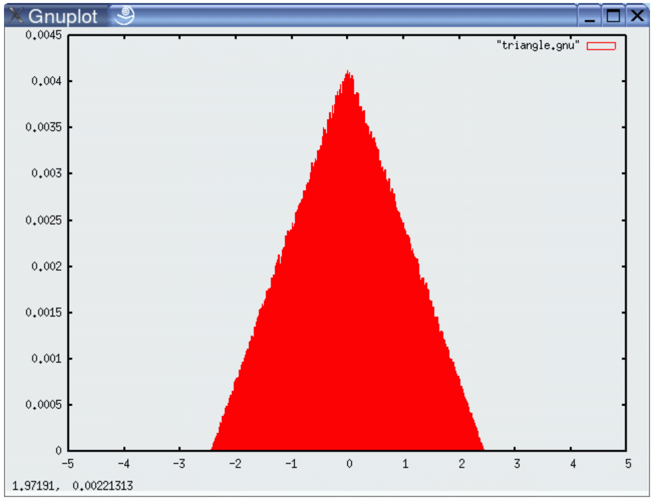
10^3 samples



10^4 samples



10^5 samples

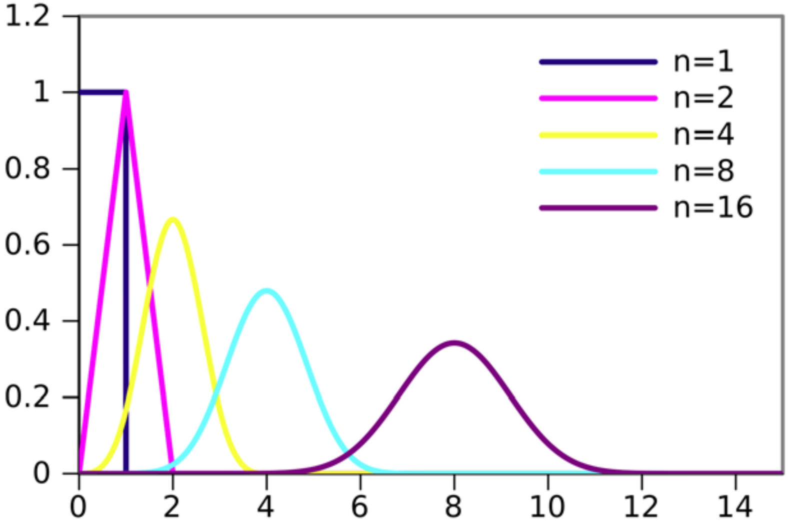


10^6 samples

Sampling from a normal dist.

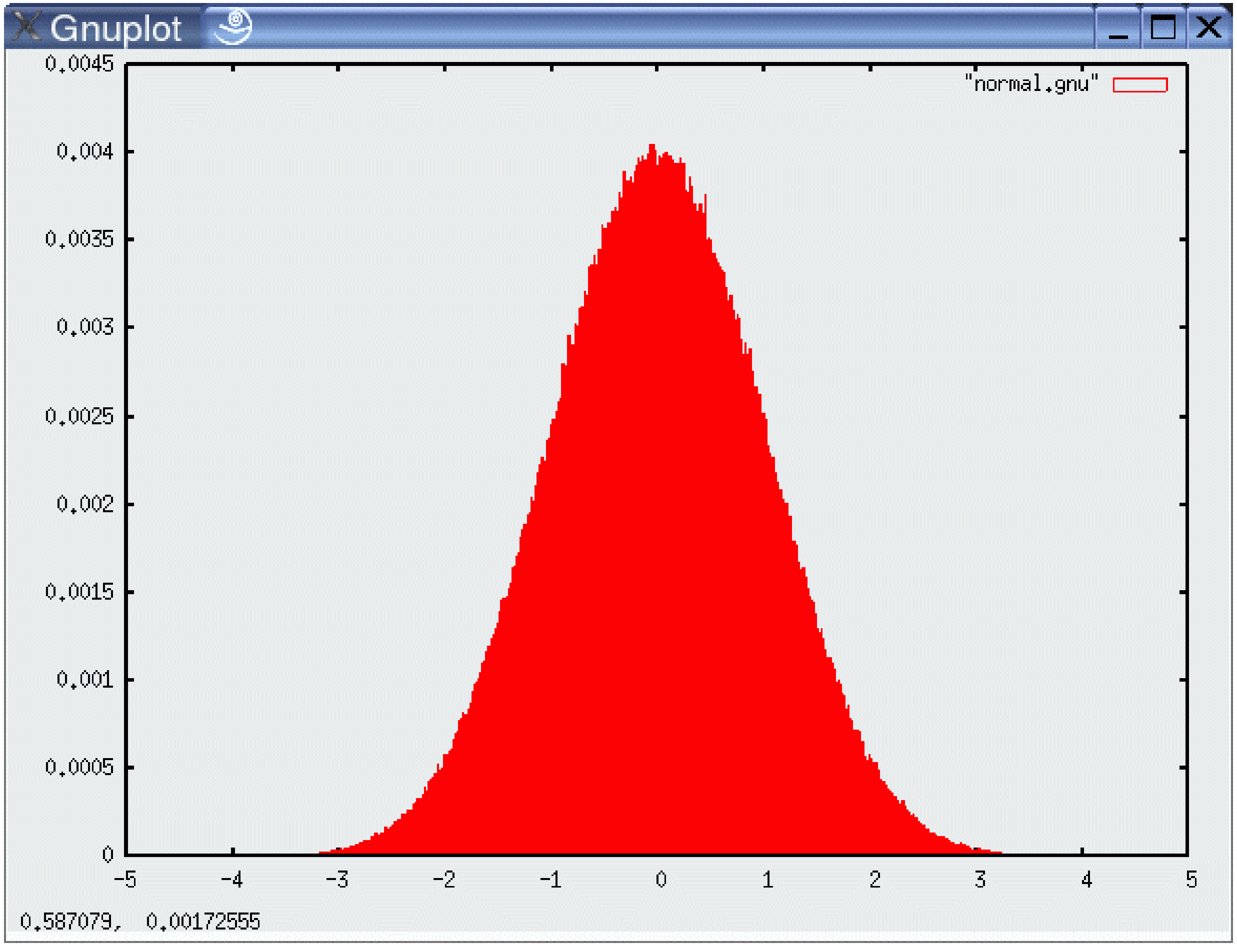
```
1. Algorithm **sample_normal_distribution**($b$):
2.     return $\frac{1}{2}\sum_{i=1}^{12} rand(-b, b)$
```

Corresponds to 12-step random walk / 12th order Irwin-Hall distribution:



Source: wikipedia

Normally Distributed Samples



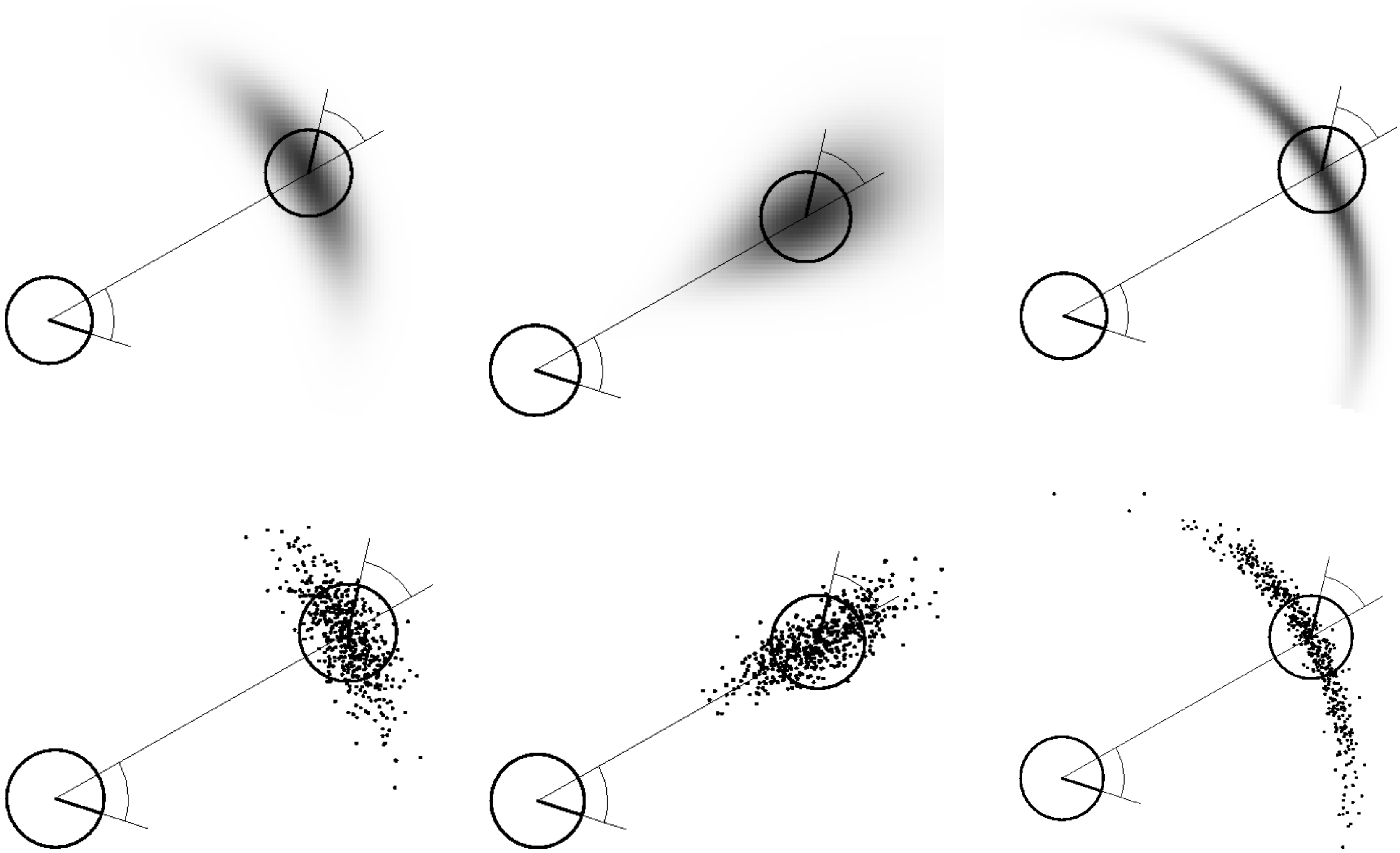
10^6
samples

Sample Odometry Motion Model

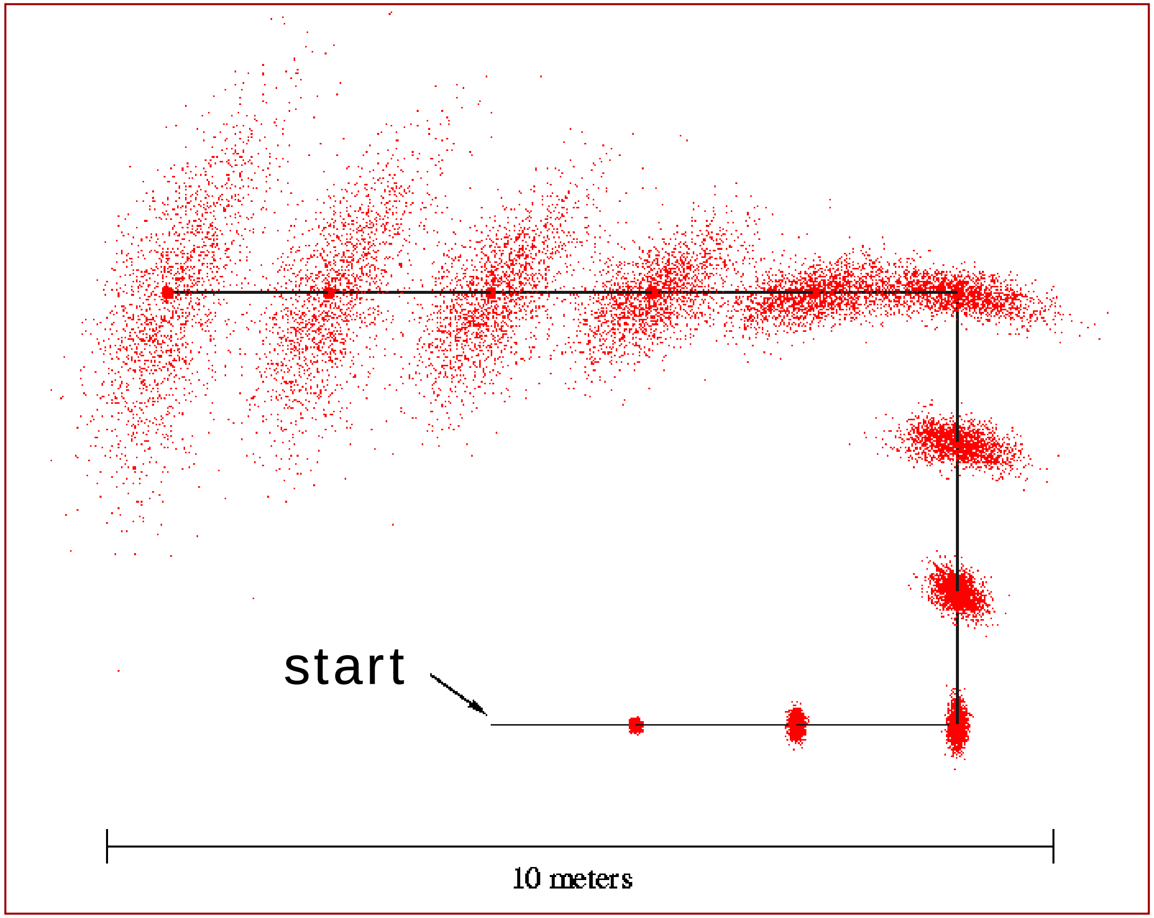
```
Algorithm **sample_motion_model**(u, x):
    $u = \langle \delta_{rot1}, \delta_{rot2}, \delta_{trans} \rangle, x = \langle x, y, \theta \rangle$

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

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

    return $\langle x', y', \theta' \rangle$
```

sample_normal_distribution

Examples for Odometry Model



Consecutive Samples



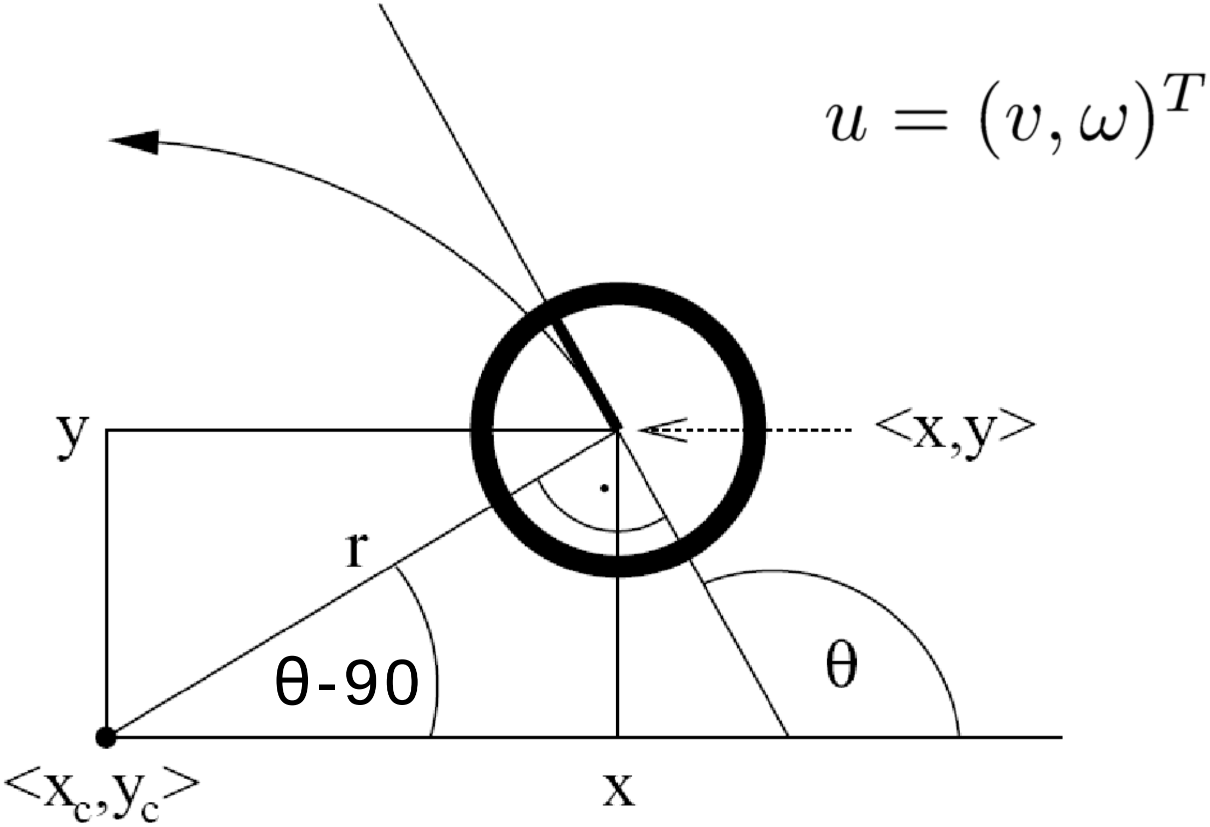
3. Velocity-based model

Dead Reckoning

- Procedure for determining the location of a vehicle
- Calculates the current pose based on its velocities and the elapsed time
- Historically used to estimate the position of ships
  - A "chip log" was thrown into the water
  - Attached to a rope with knots at known intervals
  - The number of knots that went overboard in a fixed time was used to determine the velocity

Source: Wikipedia

Velocity-Based Model



Noise Parameterization

- The hypothesis for the true motion is given by the measurement plus noise:

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

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

- What is a limitation of this parameterization?
- The $(\hat{v}, \hat{\omega})$-circle constrains the final orientation (2D manifold in a 3D space)

Noise Parameterization

- Add a parameter to account for an uncertainty in the final rotation:

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

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

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

Calculate Final Pose from the Velocities

Center of circle:

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

Distance to the ICC = radius of the circle $\lambda = v/\omega$

Note: center of the circle is orthogonal to the initial heading

Calculate Final Pose from the Velocities

Center of circle:

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

Final pose:

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

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

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

Calculate Velocities from Poses

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 μ to be determined.

The center of the circle lies on a ray halfway between x and x’ that is orthogonal to the line between x and x’

Calculate Velocities from Poses

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 to solve for μ:

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

Calculate Velocities from Poses

Parameters of the circle arc:

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

Allows for computing the velocities:

$$\begin{aligned} v &= \frac{\Delta\theta}{\Delta t} r^* \\ \omega &= \frac{\Delta\theta}{\Delta t} \end{aligned}$$

The Posterior p(x'|x, u)

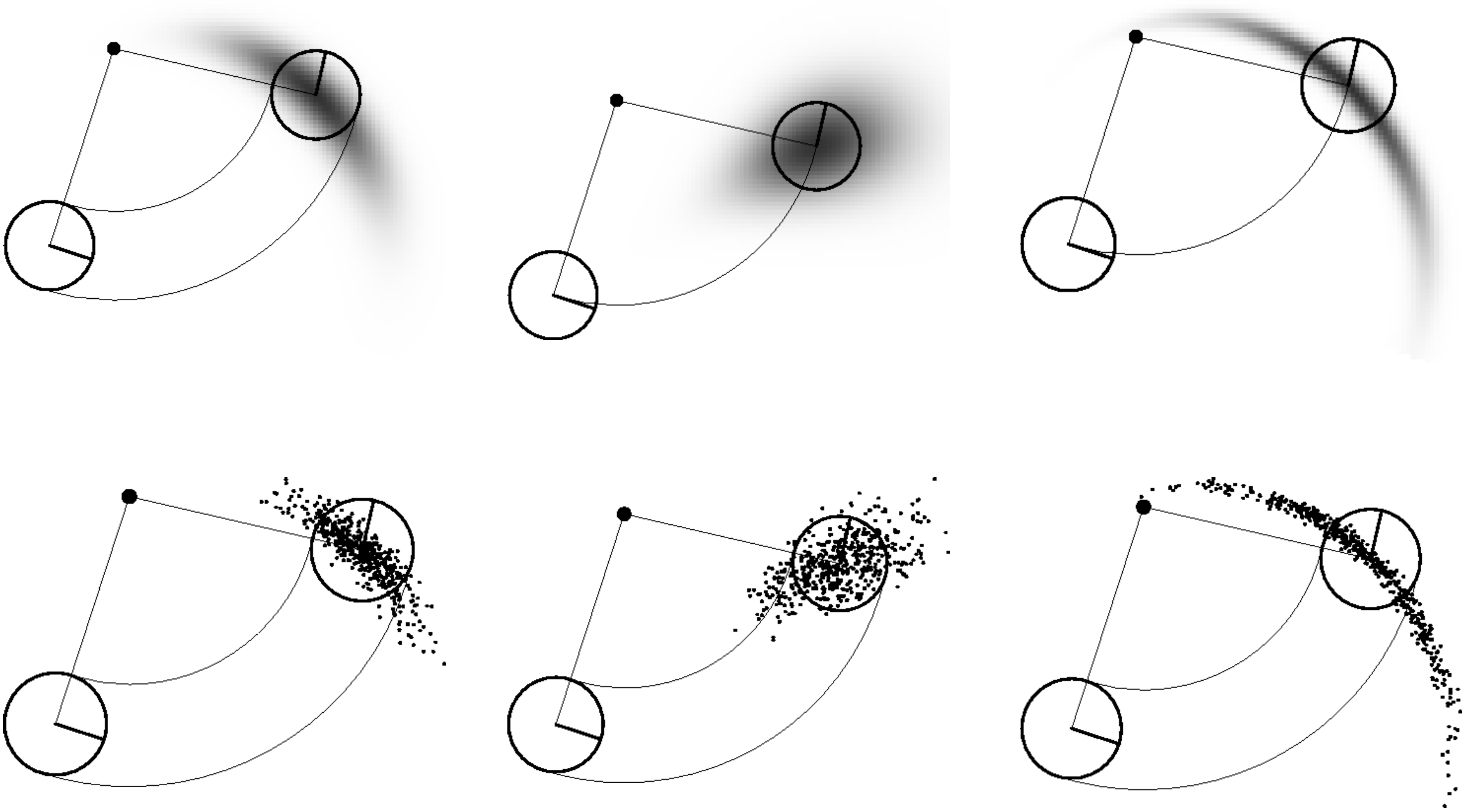
```
Algorithm motion_model_velocity($x_t, u_t, x_{t-1}$):
    $\mu = \frac{1}{2} \frac{(x-x')\cos\theta + (y-y')\sin\theta}{(y-y')\cos\theta - (x-x')\sin\theta}$
    $x^* = \frac{x+x'}{2} + \mu(y-y')$
    $y^* = \frac{y+y'}{2} + \mu(x'-x)$
    $r^* = \sqrt{(x-x^*)^2 + (y-y^*)^2}$
    $\Delta\theta = \mathrm{atan2}(y'-y^*, x'-x^*) - \mathrm{atan2}(y-y^*, x-x^*)$
    $\hat{v} = \frac{\Delta\theta}{\Delta t} r^*$
    $\hat{\omega} = \frac{\Delta\theta}{\Delta t}$
    $\hat{\gamma} = \frac{\theta'-\theta}{\Delta t} - \hat{\omega}$
    return $\mathbf{prob}(v-\hat{v}, \alpha_1 v^2 + \alpha_2 \omega^2) \cdot \mathbf{prob}(\omega-\hat{\omega}, \alpha_3 v^2 + \alpha_4 \omega^2)$
        $\cdot\, \mathbf{prob}(\hat{\gamma}, \alpha_5 v^2 + \alpha_6 \omega^2)$
```

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

Sampling from Velocity Model

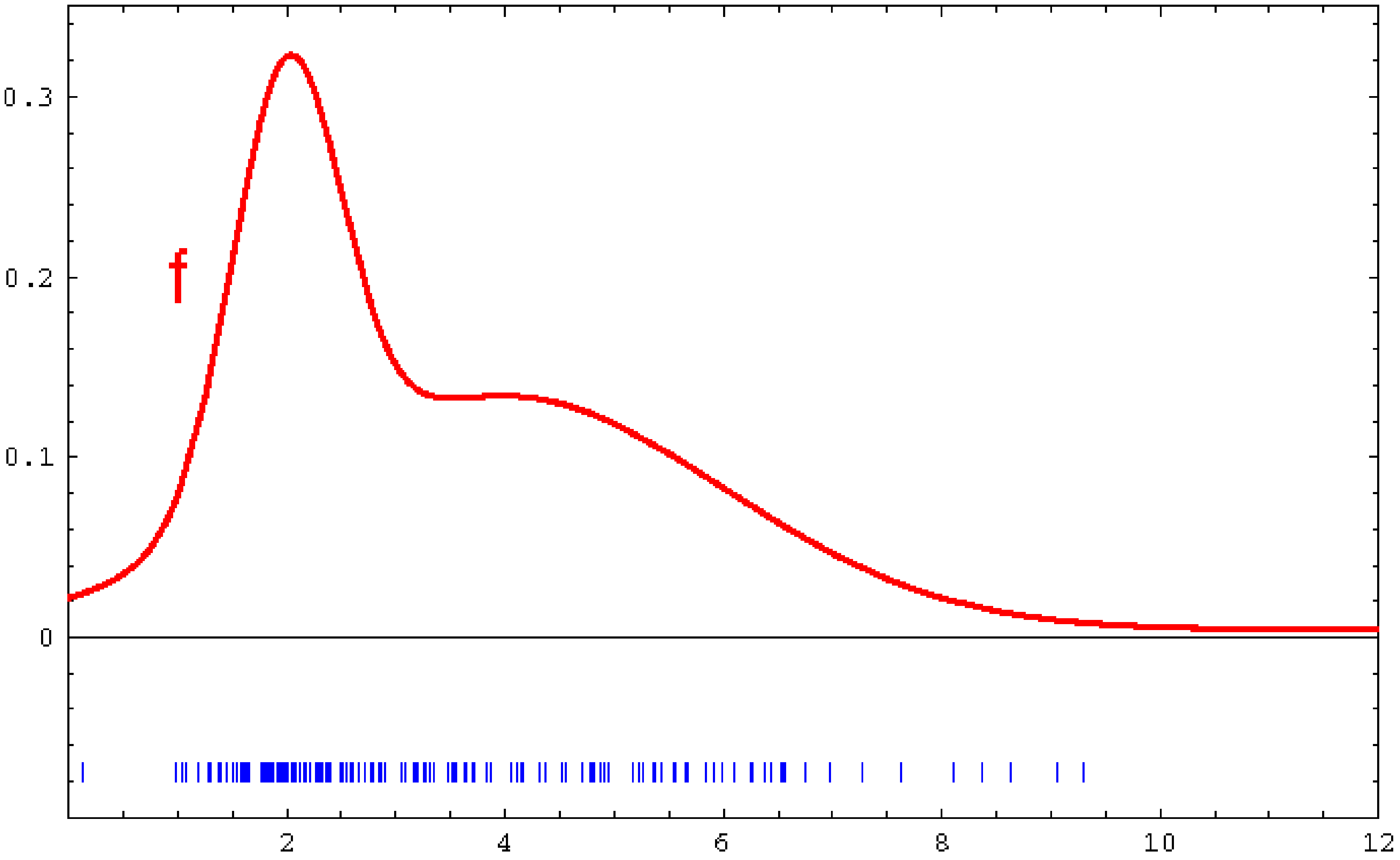
```
Algorithm sample_motion_model_velocity($u_t, x_{t-1}$):

    $\hat{v} = v + \textbf{sample}(\alpha_1 v^2 + \alpha_2 \omega^2)$
    $\hat{\omega} = \omega + \textbf{sample}(\alpha_3 v^2 + \alpha_4 \omega^2)$
    $\hat{\gamma} = \textbf{sample}(\alpha_5 v^2 + \alpha_6 \omega^2)$
    $x' = x - \frac{\hat{v}}{\hat{\omega}} \sin\theta + \frac{\hat{v}}{\hat{\omega}}\ sin(\theta + \hat{\omega}\Delta t)$
    $y' = y + \frac{\hat{v}}{\hat{\omega}} \cos\theta - \frac{\hat{v}}{\hat{\omega}}\ cos(\theta + \hat{\omega}\Delta t)$
    $\theta' = \theta + \hat{\omega}\Delta t + \hat{\gamma}\Delta t$
    return $x_t = (x', y', \theta')^T$
```

Examples for Velocity Model



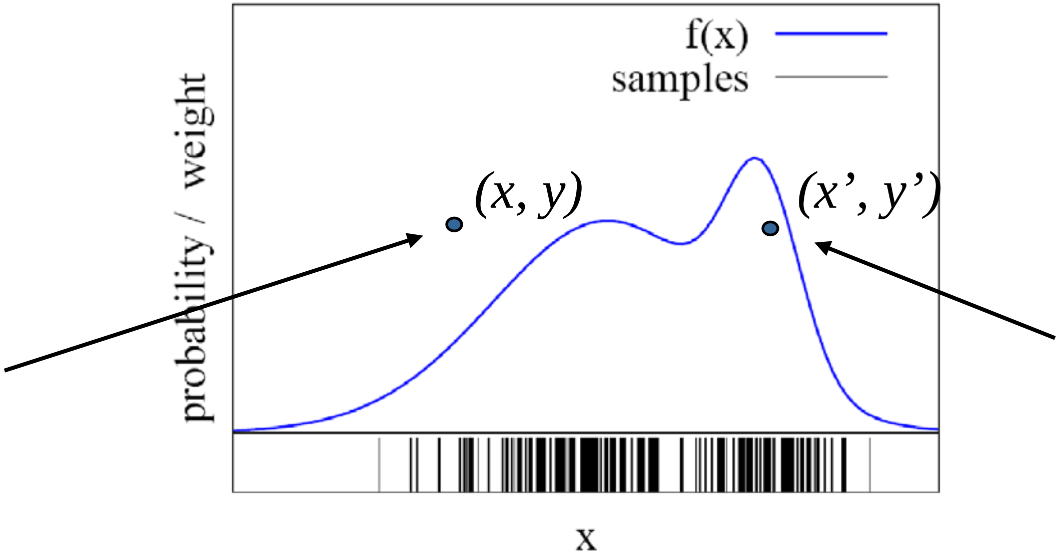
4. Rejection sampling

How to Sample from Arbitrary f?



Answer: Rejection Sampling

- First, sample from uniform distributions:
  - x in $[-b, b]$
  - y in $[0, \max f\,]$
- if $f(x) > y$ keep the sample x
  otherwise reject it



rejected

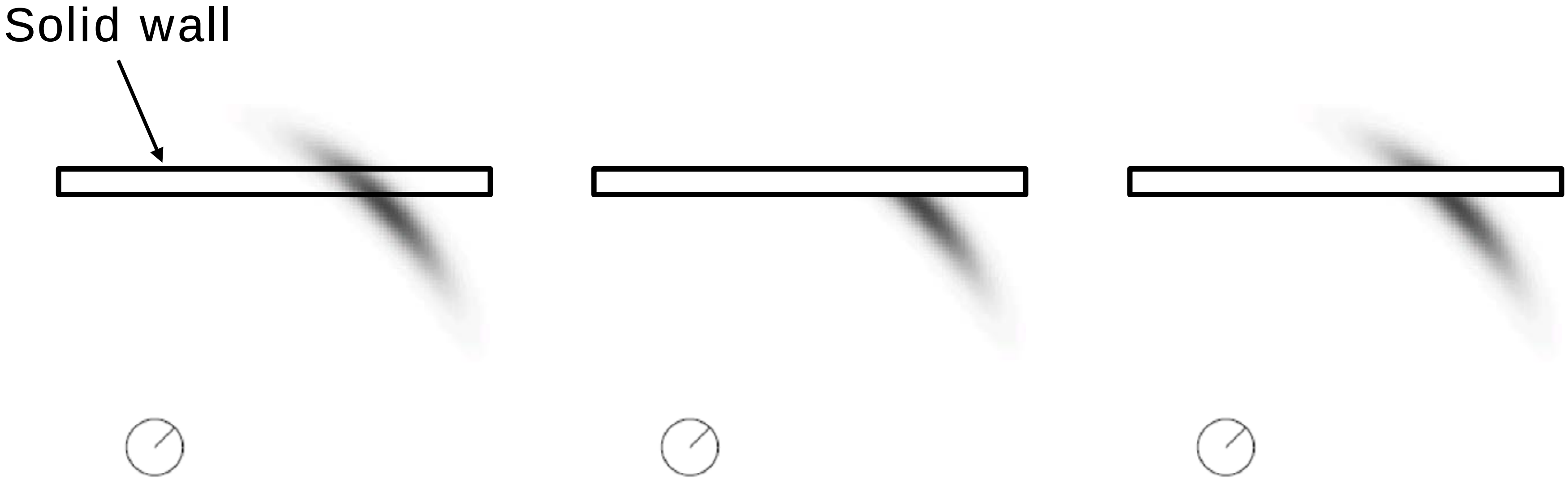
accepted

Rejection Sampling Algorithm

```
Algorithm **sample_distribution**( $f, b$ ):
repeat
    $x = \mathbf{rand}(-b, b)$
    $y = \mathbf{rand}(0, \max\{f(x) \mid x \in [-b, b]\})$
until $y \leq f(x)$
return $x$
```

5. Map-consistent motion

Map-Consistent Motion Model



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

Questions?