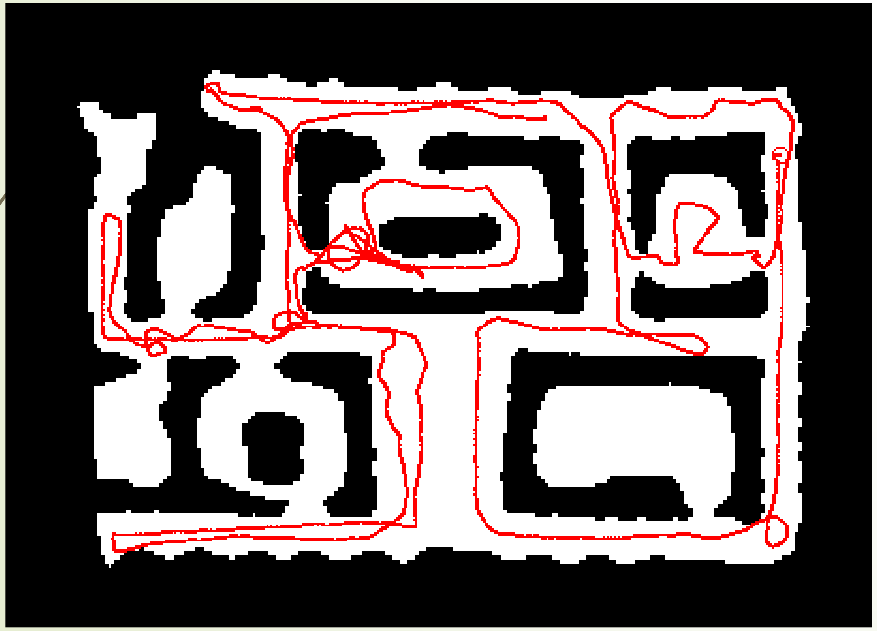
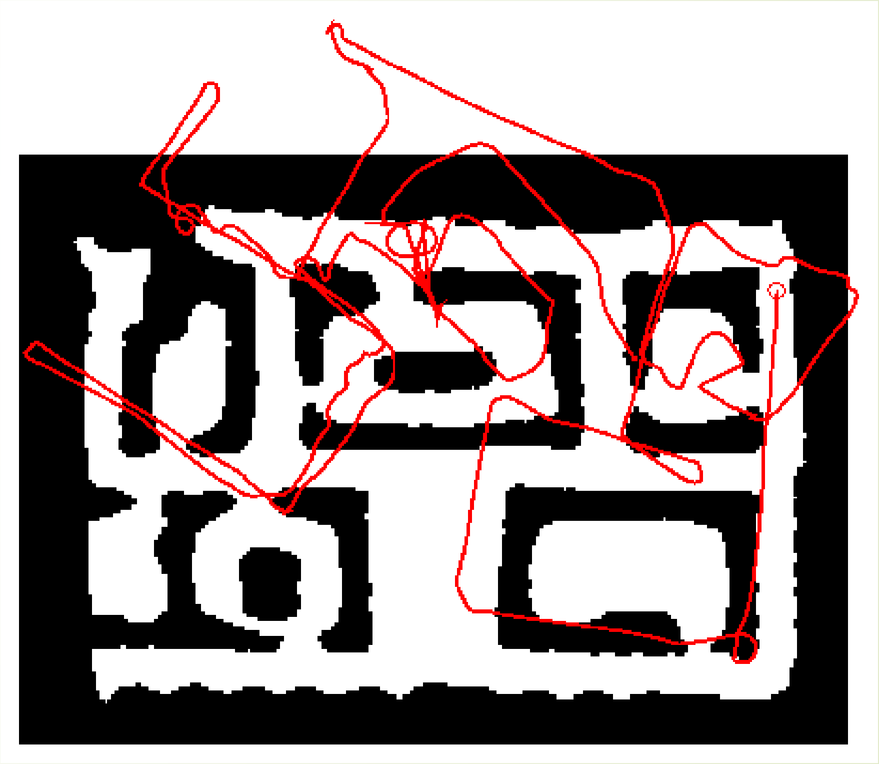
MCE 412 - Autonomous Robotics

Probabilistic Motion Models

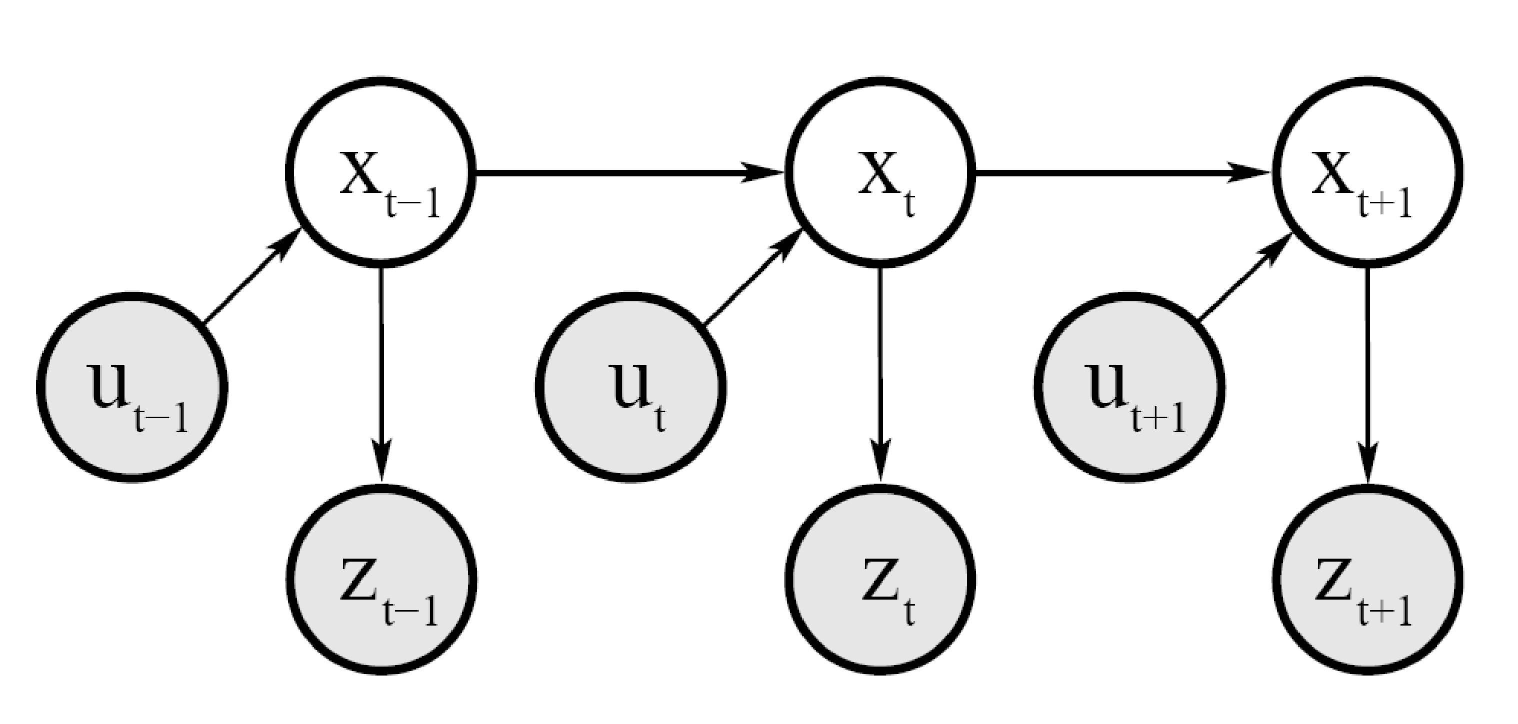
Motivation

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





Dynamic Bayesian Network for Controls, States, and Sensations

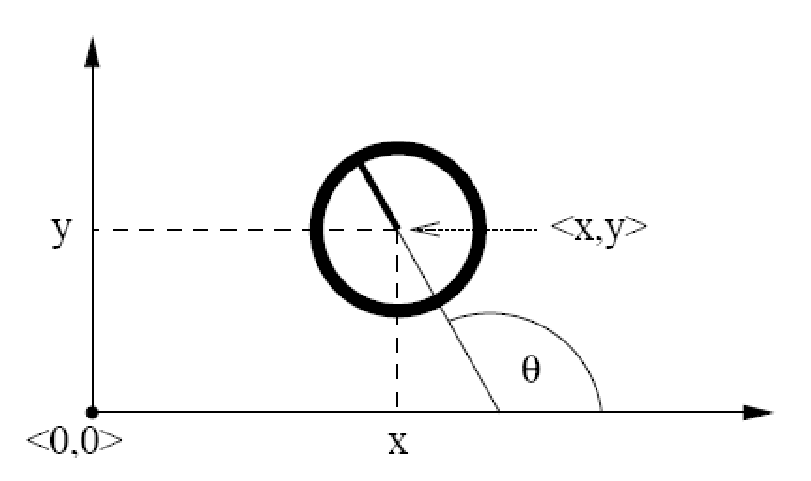


Probabilistic Motion Models

- To implement the Bayes Filter, we need the transition model $p(x_t|x_{t-1}, u_t)$.
- The term $p(x_t|x_{t-1}, u_t)$ specifies a posterior probability, that action u_t carriers the robot from x_{t-1} to x_t.
- In this section we will discuss, how $p(x_t|x_{t-1}, u_t)$ can be modeled based on the motion equations and the uncertain outcome of the movements.

Coordinate Systems

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

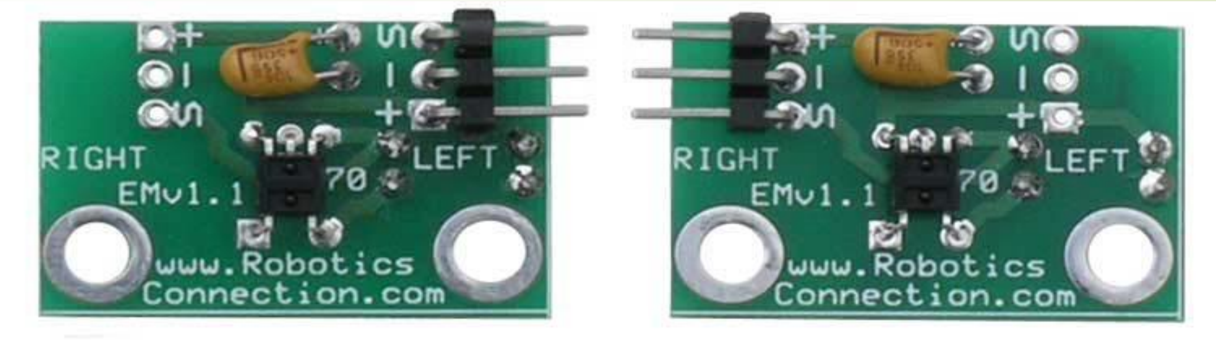


Typical Motion Models

- 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 0V output when they 'see' black.



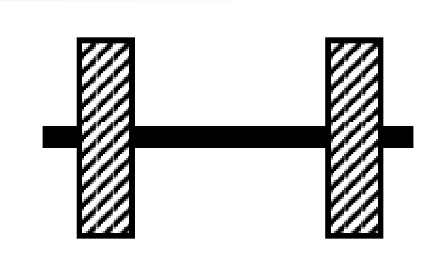
- These disks are manufactured out of high quality laminated color plastic to offer a very crisp black to white transition. This enables a wheel encoder sensor to easily see the transitions.

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

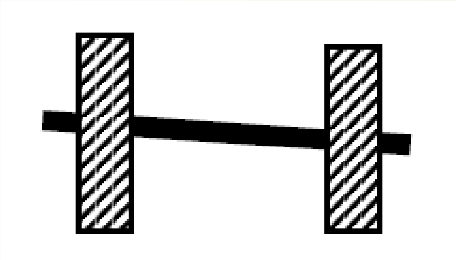
Dead Reckoning

- 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.

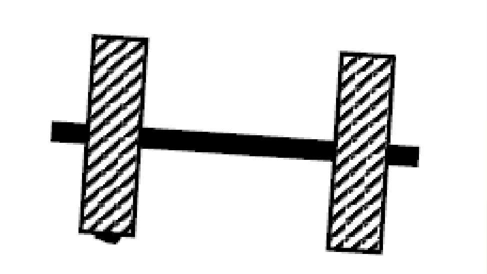
Reasons for Motion Errors of Wheeled Robots



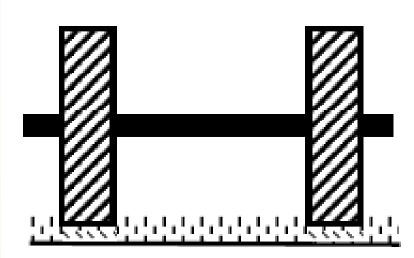
ideal case



Different wheel diameters



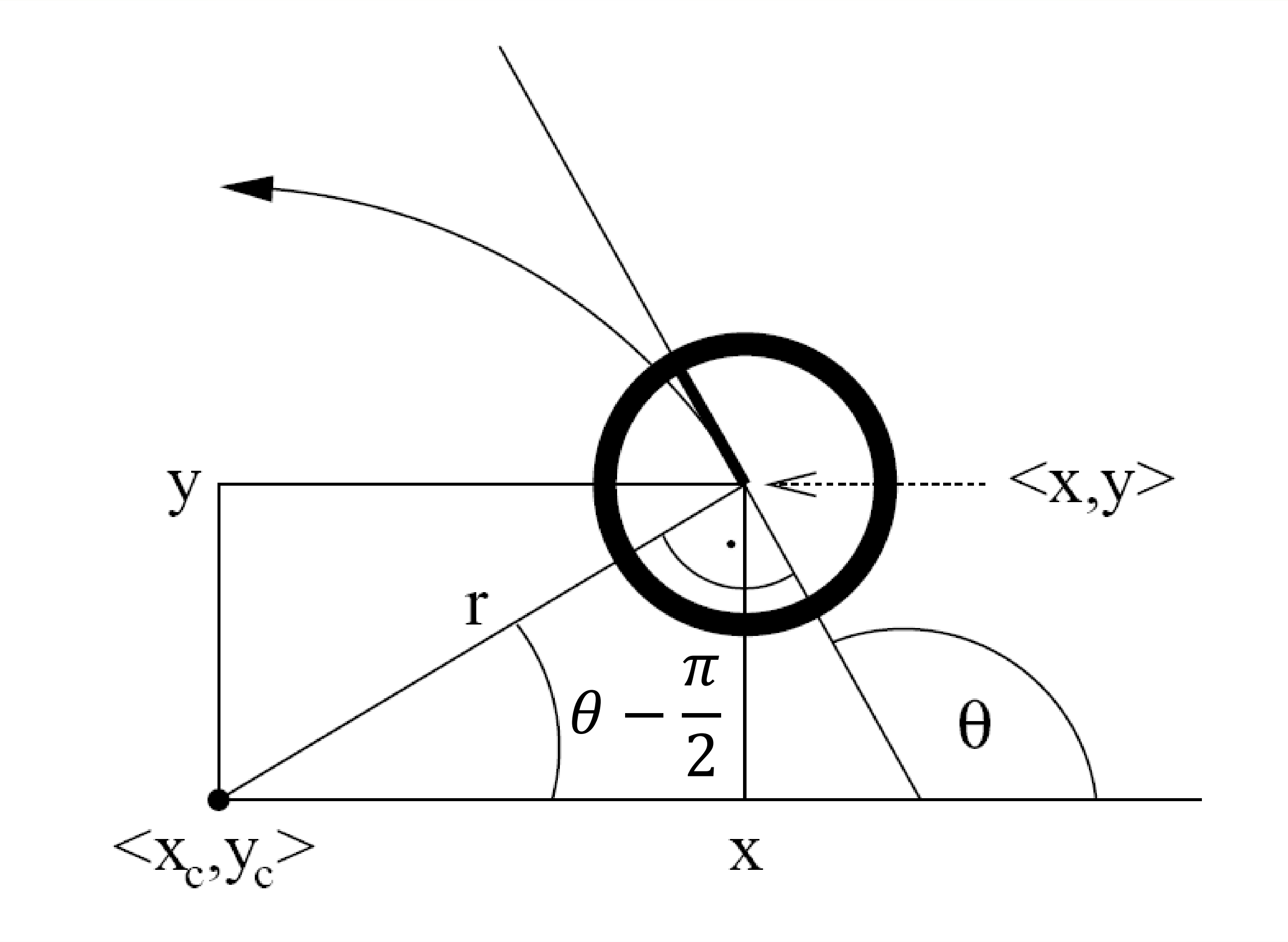
bump



carpet

and many more …

Velocity Based Model



Noise Model for the Velocity Based Model

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

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

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

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

Noise Model for the Velocity Based Model

- The $(\hat{v}, \hat{w})$-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 Third 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

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

- Initial pose: $x_{t-1} = (x, y, \theta)^T$
- Final pose: $x_t = (x', y', \theta')^T$
- Center of circle:

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

Some constant (the center of the circle 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

- Initial pose: $x_{t-1} = (x, y, \theta)^T$
- Final pose: $x_t = (x', y', \theta')^T$
- Center of circle:

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

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

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

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

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

Radius of the circle

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

Change in heading direction

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

Equation for the Velocity Model

- The parameters of the circle:

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

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

- Allow for computing the velocities as

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

Posterior Probability for Velocity Model

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

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

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

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

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

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

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

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

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

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

Due to the independence of noises

Sampling from Velocity Model

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

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

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

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

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

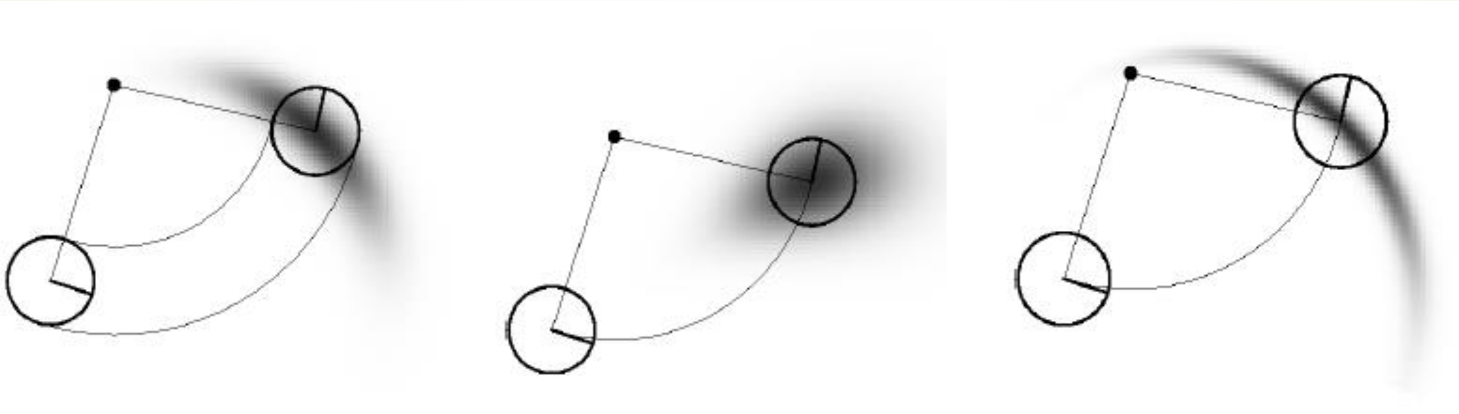
$$y' = y + \frac{\hat{v}}{\hat{\omega}} \cos\theta - \frac{\hat{v}}{\hat{\omega}}\, cos(\theta + \hat{\omega}\Delta t)$$

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

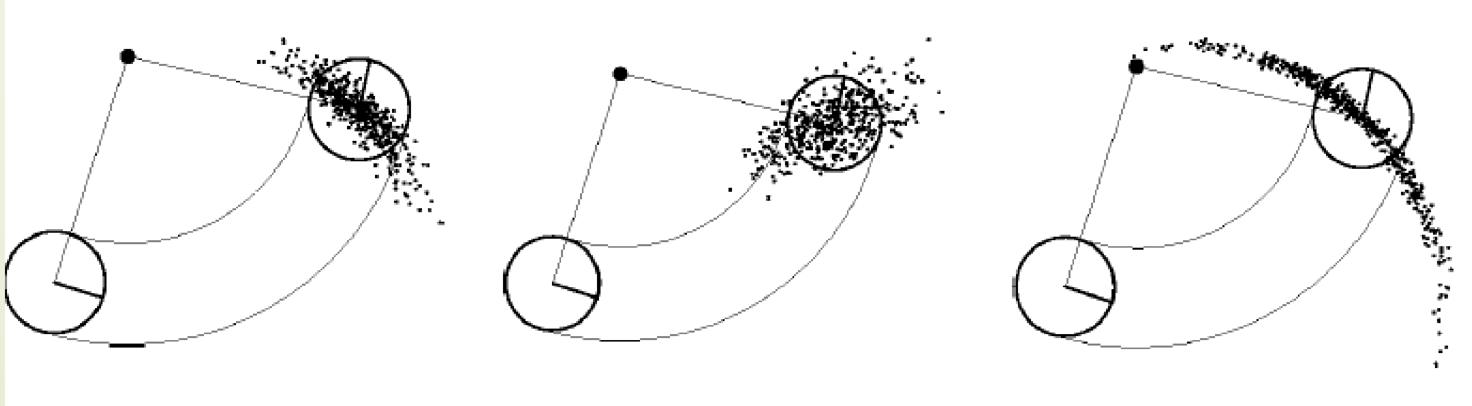
return $x_t = (x', y', \theta')^T$

Examples (velocity based)

- The velocity motion model for different noise parameter settings



- Sampling from the velocity motion model (each diagram shows 500 samples)



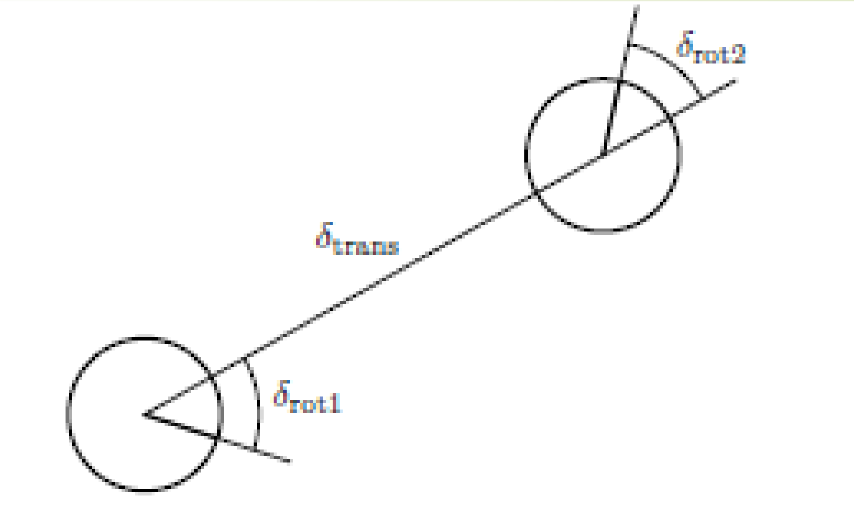
Odometry Motion Model

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

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

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

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



The atan2 Function

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

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

Noise Model for Odometry

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

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

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

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

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 $\frac{\sqrt{6}}{2}\left[\mathbf{rand}(-b, b) + \mathbf{rand}(-b, b)\right]$

Calculating the Posterior Given x,x' and Odometry

hypotheses odometry

Algorithm **motion_model_odometry**$(\boxed{x, x'}\,\boxed{\bar{x}, \bar{x}'})$

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

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

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

odometry params (**u**)

$$\hat{\delta}_{trans} = \sqrt{(x'-x)^2 + (y'-y)^2}$$

$$\hat{\delta}_{rot1} = \text{atan2}(y'-y, x'-x) - \theta$$

$$\hat{\delta}_{rot2} = \theta' - \theta - \hat{\delta}_{rot1}$$

values of interest (**x**,**x'**)

$$p_1 = \text{prob}(\delta_{\text{rot1}} - \hat{\delta}_{\text{rot1}}, \alpha_1 |\delta_{\text{rot1}}| + \alpha_2 \delta_{\text{trans}})$$

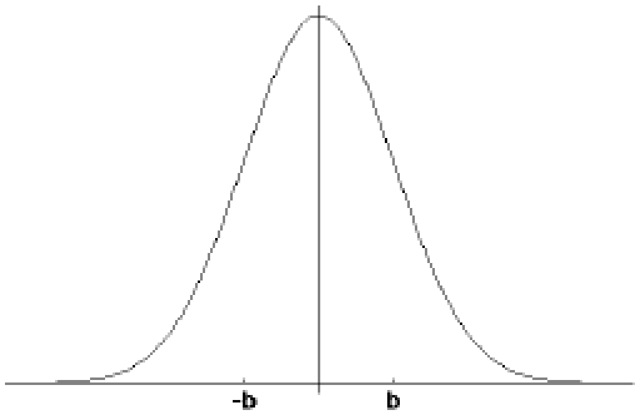
$$p_2 = \text{prob}(\delta_{\text{trans}} - \hat{\delta}_{\text{trans}}, \alpha_3 \delta_{\text{trans}} + \alpha_4 (|\delta_{\text{rot1}}| + |\delta_{\text{rot2}}|))$$

$$p_3 = \text{prob}(\delta_{\text{rot2}} - \hat{\delta}_{\text{rot2}}, \alpha_1 |\delta_{\text{rot2}}| + \alpha_2 \delta_{\text{trans}})$$

return $p_1 \cdot p_2 \cdot p_3$

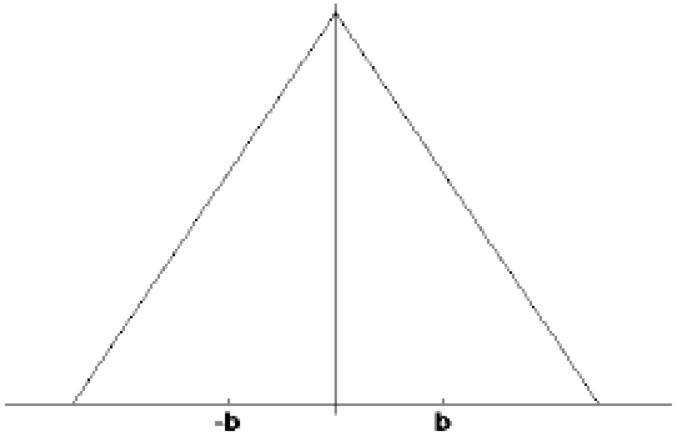
Typical Distributions for Probabilistic Motion Models

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 (zero-centered)

- For a normal distribution

query point

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

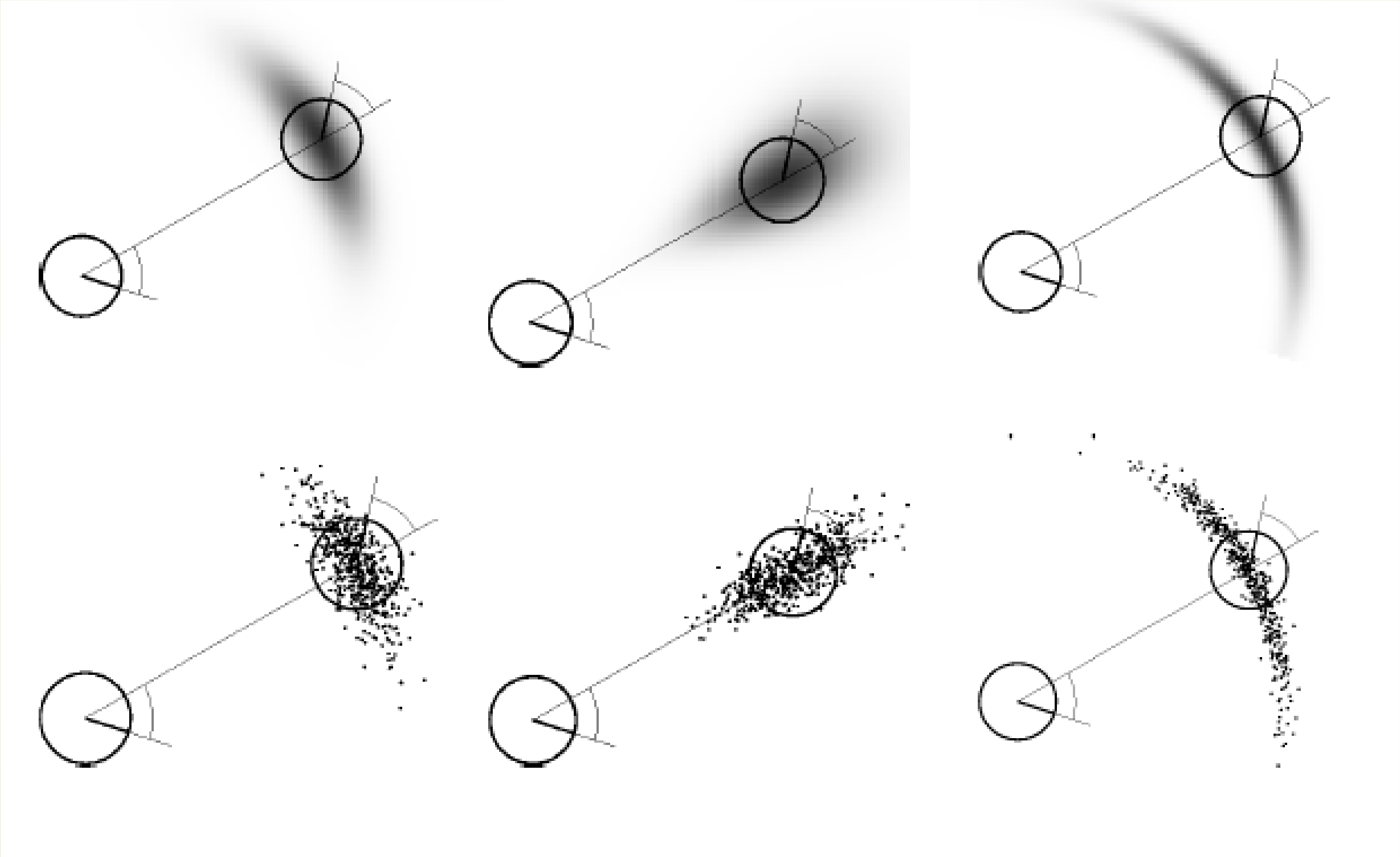
std. deviation

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

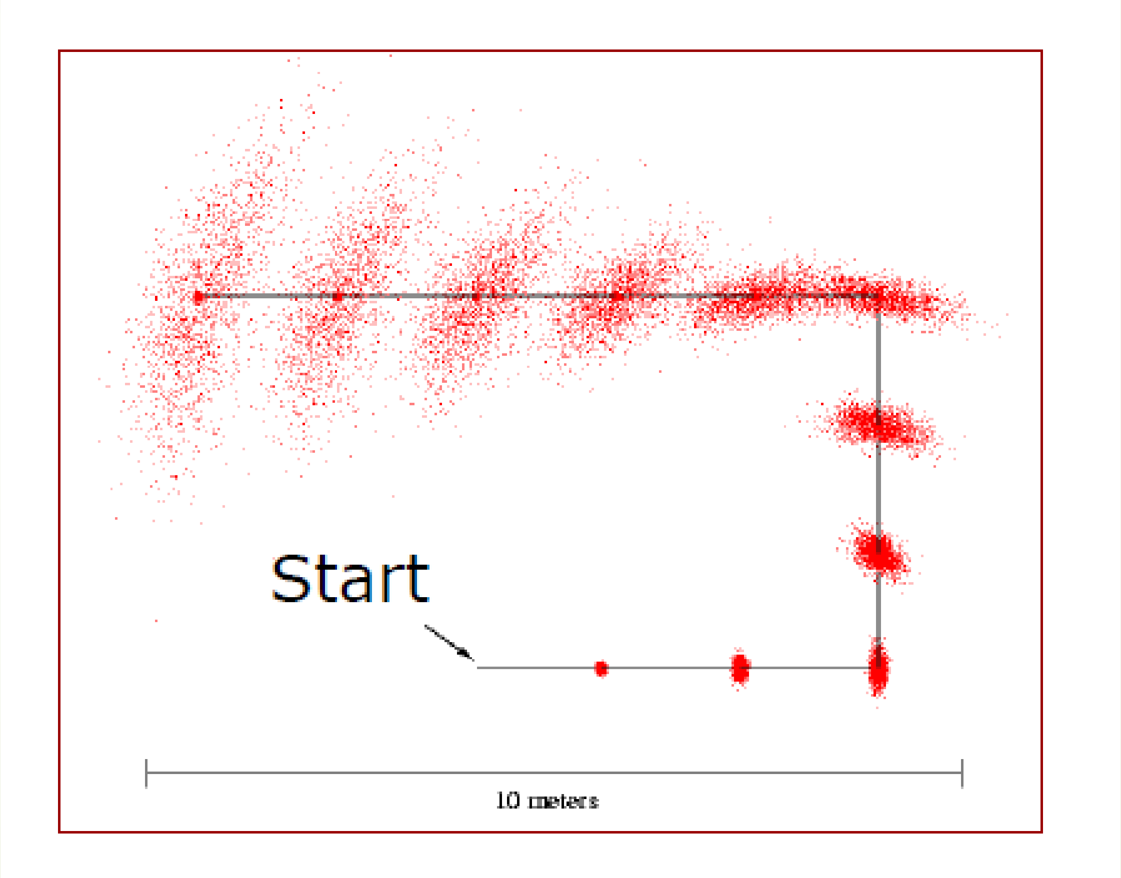
- For a triangular distribution

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

Examples (Odometry-based)



Sampling approximation of the position belief for a non-sensing robot



Summary

- We discussed motion models for odometry-based and velocity based systems
- We discussed ways to calculate the posterior probability $p(x_t|x_{t-1}, u_t)$
- We also described how to sample from $p(x_t|x_{t-1}, u_t)$
- Typically the calculations are done in fixed time intervals Δt
- In practice, the parameters of the models have to be learned.

References

- Probabilistic Robotics, Sebastian Thrun, Wolfram Burgard and Dieter Fox, Chapter 5