(9) Probability Theory

1. Uncertainty is Inherent in the World.

1. Four Factors

1. Environment is stochastic, unpredictable
2. Robot actions are stochastic
3. Sensor are limited and noisy
4. Models are inaccurate, incomplete.

2. Upsides and Downsides

Can accommodate inaccurate models / imperfect sensors

2. Axiom

1. $0 \leq Pr(A) \leq 1$
2. $Pr(A \vee B) = Pr(A) + Pr(B) - Pr(A \wedge B)$

3. Joint and Conditional Probability

$P(X=x \text{ and } Y=y) = P(x,y)$

If X, Y are independent then $P(x,y) = P(x) \cdot P(y)$

$P(x|y) = P(x,y)/P(y) \Rightarrow P(x|y) \cdot P(y) = P(x,y)$

4. Law of total Probability

Discrete $\quad P(x) = \sum_y P(x|y)\, P(y)$

Continous $\quad p(x) = \int p(x|y) \cdot p(y)\, dy$

5. Bayes Theorem

① State Space $S=\{F_1,\cdots,F_n\}$，已知 $P(F_j|E)$

$$P(F_j|E)=\frac{P(F_j\wedge E)}{P(E)}$$

$$=\frac{P(E|F_j)P(F_j)}{P(E)}$$

$$P(F_j|E)=\frac{P(E|F_j)P(F_j)}{\sum_i P(E|F_i)\cdot P(F_i)}$$

② $P(x,y)=P(x|y)P(y)=P(y|x)\cdot P(x)$

$$\Rightarrow P(x|y)=\frac{P(y|x)\cdot P(x)}{P(y)}=\frac{\text{Likelihood}\cdot\text{Prior}}{\text{Evidence}}$$

$$=\eta P(y|x)\cdot P(x)$$

$$\eta=P^{-1}(y)=\left(\sum_x P(y|x)\cdot P(x)\right)^{-1}$$

③ Multi-measurement in Bayes Theorem

$$P(x|y,z)=\frac{P(x,y,z)}{P(y,z)}=\frac{P(y|x,z)\cdot P(x,z)}{P(y,z)}$$

$$=\frac{P(y|x,z)\cdot P(x|z)\cdot P(z)}{P(y|z)\cdot P(z)}$$

$$=\frac{P(y|x,z)\cdot P(x|z)}{P(y|z)}$$

④ Markov Assumption : At time t, state only dependens on state at time $t-1$ and action u at time t.

At time t, measurement z, only connects with state at time t

$$P(x_t|x_{1:t-1},z_{1:t-1},u_{1:t})=P(x_t|x_{t-1},u_t)$$

$$P(z_t|x_{0:t},z_{1:t},u_{1:t})=P(z_t|x_t)$$

(1) Action Model

$P(x|u, x')$: Current State x' + Action u $\Downarrow$ Future State x — Using state transfer model to describe the influence of action to the state.

Continous : $P(x|u) = \int P(x|u, x')dx'$

Discrete : $P(x|u) = \sum P(x|u, x')P(x')$

Generally, robot do discrete actions

Example: $P(open|u)$: Compute the probablility of door is open after do action u.

$$P(open|u) = \sum P(open|u, x')P(x')$$

$$= P(open|u, open)\cdot P(open) + P(open|u, Closed)\cdot P(Closed)$$

All previous states

(2) Sensor Model

$$P(x|y, z) = \frac{P(y|x, z)\cdot P(x, z)}{P(y|z)}$$

1. Assume $z_n = y$, $(z_1, \cdots z_{n-1}) = z$

$$P(x|z_1, \cdots, z_n) = \frac{P(z_n|x_1, z_1, \cdots, z_n)\, P(x|z_1, \cdots, z_{n-1})}{P(z_n|z_1, \ldots, z_{n-1})}$$

2. Since Markov Assumption: Given x, z_n independent with $z_1 \cdots z_{n-1}$

$$P(x|z_1, \cdots, z_n) = \frac{P(z_n|x_1)\, P(x|z_1, \cdots, z_{n-1})}{P(z_n|z_1, \ldots, z_{n-1})}$$

3. Bayes Filter

$$P(x|z_1, \cdots, z_n) = \frac{P(z_n|x_1)\, P(x|z_1, \cdots, z_{n-1})}{P(z_n|z_1, \ldots, z_{n-1})}$$

Same Form

$$= \eta_n P(z_n|x)\cdot P(x|z_1 \cdots z_{n-1})$$

$$\vdots$$

$$= \eta_1 \cdots \eta_n \prod_{i=1\cdots n} P(z_i|x)\cdot P(x)$$

(3) Conclusion

(1) Add measurement (x, y) raise the probability of robot's state

(2) Do movement raise the blur of state

For Measurement

$$P(x|z_1,\cdots,z_n)=\eta_1\cdots\eta_n\prod_{i=1\cdots n}P(z_i|x)\cdot P(x)$$

For Movement

$$P(x|u)=\sum P(x|u,x')\cdot P(x')$$

It contains all possible conditions

$$Bel(x_t)=P(x_t|u_1,z_1,\cdots,u_t,z_t)$$

$$=\eta P(z_t|x_t,u_1,z_1,\cdots,u_t)\cdot P(x_t|u_1,z_1,\cdots,u_t) \quad \text{Bayes}$$

$$=\eta P(z_t|x_t)\cdot P(x_t|u_1,z_1,\cdots,u_t) \quad \text{Markov}$$

Total Prob

$$=\eta P(z_t|x_t)\cdot\int P(x_t|u_1,z_1,\cdots,u_t,x_{t-1})P(x_{t-1}|u_1,z_1,\cdots,u_t)dx_{t-1})$$

$$=\eta P(z_t|x_t)\cdot\int P(x_t|u_t,x_{t-1})P(x_{t-1}|u_1,z_1,\cdots,u_t)dx_{t-1}) \quad \text{Markov}$$

$$=\eta P(z_t|x_t)\int P(x_t|u_t,x_{t-1})P(x_{t-1}|u_1,z_1,\cdots,z_{t-1})dx_{t-1}) \quad \text{Markov}$$

$$=\eta P(z_t|x_t)\cdot\int P(x_t|u_t,x_{t-1})\cdot Bel(x_{t-1})dx_{t-1}$$

Algorithm **Bayes_filter**(*Bel(x),d*):

$\eta=0$

If *d* is a perceptual data item *z* then

For all *x* do

$$Bel'(x)=P(z|x)Bel(x)$$

$$\eta=\eta+Bel'(x)$$

For all *x* do

$$Bel'(x)=\eta^{-1}Bel'(x)$$

Update

Else if *d* is an action data item *u* then

For all *x* do

$$Bel'(x)=\int P(x|u,x')\,Bel(x')\,dx'$$

Prediction

Return *Bel'(x)*

Inside

1. Motion, Measure data from $t_0 \to t_t$ $d_t=\{u_1,z_1,\cdots,u_t,z_t\}$
2. Measurement Model $P(z|x)$
3. Motion Model $P(x|u,x')$
4. System state prior $P(x)$

Output: Belief 置信概率

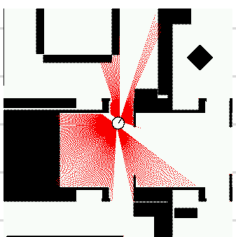
$$Bel(x_t)=P(x_t|u_1,z_1,\cdots u_t,z_t)$$

10. Sensor And Motion Model

1. Sensors for Mobile Robots

Contact Sensors: Bumpers
•Internal Sensors
–Accelerometers (spring-mounted masses)
–Gyroscopes (spinning mass, laser light)
–Compasses, inclinometers (earth magnetic field, gravity)
•Proximity Sensors
–Sonar (time of flight)
–Radar (phase and frequency)
–Laser range finders (triangulation, tof, phase)
–Infrared (intensity)
•Visual Sensors: Cameras
•Satellite-based sensors: GPS

2. Proximity Sensors



1. The central task is to determine $P(z|x)$:

The probability of a measurement z, given that the position x

3. Beam-based Sensor Model 光线传感器模型

1. Scan z consists of K measurements

$$z = \{z_1, z_2, \cdots, z_k\}$$

2. Individual measurements are independent given the robot position

$$P(z|x,m) = \prod_{k=1}^{K} P(z_k|x,m)$$

4. Typical Measurement Error of an Range Measurement

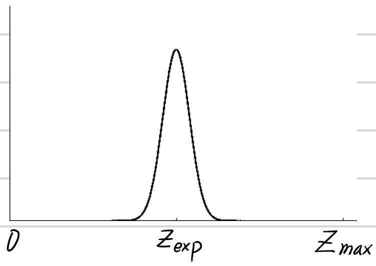
1. Beams reflected by obstacles
2. Beams reflected by persons/caused by crosstalk
3. Random measurement
4. Maximum range measurements

5. Proximity Measurement

1. Measurement can be caused by
   1. A known obstacle
   2. Cross-talk
   3. An unexpected obstacle (people, furniture)
   4. Missing all obstacles (total reflection, glass)
2. Noise is due to uncertainty
   1. In measuring distance to known obstacles.
   2. In position of known obstacles
   3. In position of additional obstacles
   4. Wheather obstacle is missed.

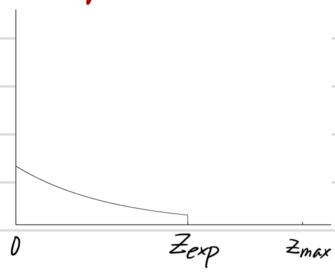
6. Beam-based Proximity Model

Measurement noise



$$P_{hit}(z|x,m) = \eta \frac{1}{\sqrt{2\pi\sigma}} \cdot e^{-\frac{(z-z_{exp})}{2b}}$$

Unexpected obstacles

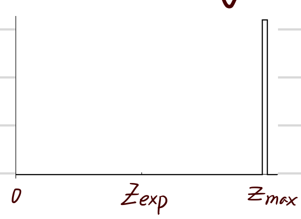


$$P_{unexp}(z|x,m) = \begin{cases} \eta \lambda e^{-\lambda z}, & z < z_{exp} \\ 0, & \text{otherwise} \end{cases}$$

Random Measurement

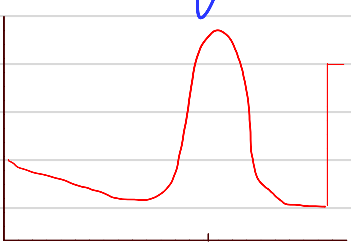
$$P_{rand}(z|x,m) = \eta \frac{1}{z_{max}}$$

Max Range



$$P_{max}(z|x,m) = \eta \frac{1}{z_{small}}$$

7. Resulting Mixture Density



$$p(z|x,m) = \begin{pmatrix} \alpha_{hit} \\ \alpha_{unexp} \\ \alpha_{max} \\ \alpha_{rand} \end{pmatrix}^T \cdot \begin{pmatrix} P_{hit}(z|x,m) \\ P_{unexp}(z|x,m) \\ P_{max}(z|x,m) \\ P_{rand}(z|x,m) \end{pmatrix}$$

Question? How to determine the model parameters?

8. Approximation 逼近 或者称之为优化

1. Maximize log likelihood of the data 最大化数据的对数可能性

   $P(z|z_{exp})$

2. Search space of n-1 parameters

   1. Hill climbing
   2. Gradient descent 梯度下降
   3. Genetic algorithm 遗传算法

   非确定性搜索算法

3. Deterministically compute the n-th parameter to satisfy normalization constrant

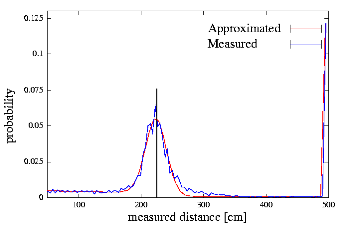
9. Discrete Model of Proximity Sensors 近似传感器的离散模型

1. Instead of dencities, consider discrete steps alone the sensor beam 之前为连续的密度模型(=考虑传感 Data Density)

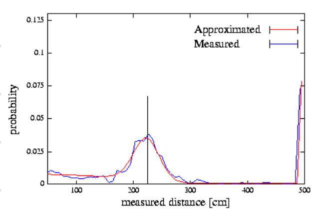
   现在开始思考离散步骤

2. Consider dependencies between different cases

$$P(d_i|l) = 1-(1-\sum_{j<i} P_u(d_j))c_d P_m(d_j|l))) \cdot (1-(1-\sum_{j<i} P(d_j))c_r)$$



Laser



Sonar

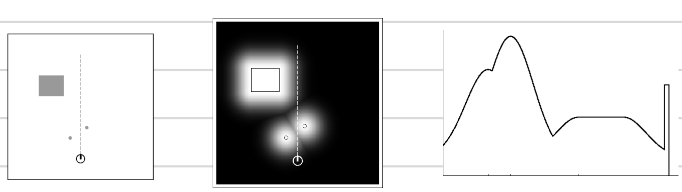
10. Scan-based Model

1. Beam-based model is
   1. Not smooth for small obstacles and at edge
   2. Not efficient (When doing approximation)

   Idea: Instead of following alone the beam, just check the end point.

2. Probability is a mixture of
   1. A Gaussian distribution with mean at distance to closest obstacle 以与最近障碍物的距离作为平均值的高斯分布
   2. A uniform distribution for random measurements. 随机测量的平均分布
   3. A small uniform distribution for max range measurements. 最大距离测量的平均分布

3. Again, independence between different components is assigned 以上三者都是相互独立的



12. Probabilistic Motion Models

1 Robob Motion

Robot motion is inherently uncertain, so, how to model this uncertainty

1. To implement the Bayes Filter. we need the transition model $P(x|x',u)$, which specifies a posterior probability, that action u carries the robot from $x' \Rightarrow x$.

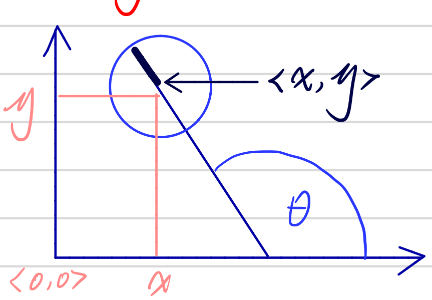
2. Now. I will specify, how $P(x|x',u)$ can be modeled based on the motion equations.

3. Coordinate Systems

   1. In general the configuration of a robot can be described by 6 parameters ——— 3-dimensional cartesian coordinate + 3 Euler angles (Pitch, Roll, Yaw)

   2. Simply, I'll do it on a planar surface.

      The state space of such system is 3-dimensional (x, y, θ)



4. Typical Motion Models

1. Odometry – based : Used when systems are equipped with wheel encoders

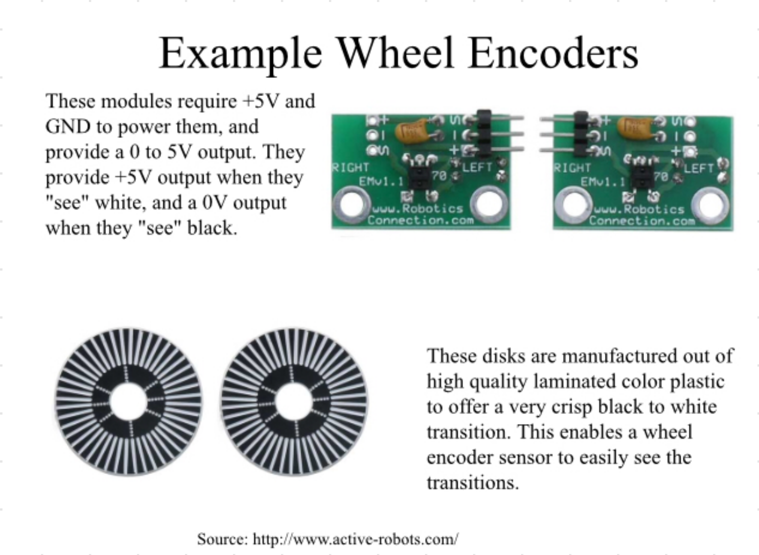
2. Velocity – based : Used when no wheel encoders are given.

dead reckoning

3. They calulated the new pose based on the velocities and time elapsed.

In ROS: Odometry.pose.pose. Point (float x;y;z)

Quaternion(float x,y,z,w)



4. Dead Reckoning

Compute the current pose, based on current velocity and time

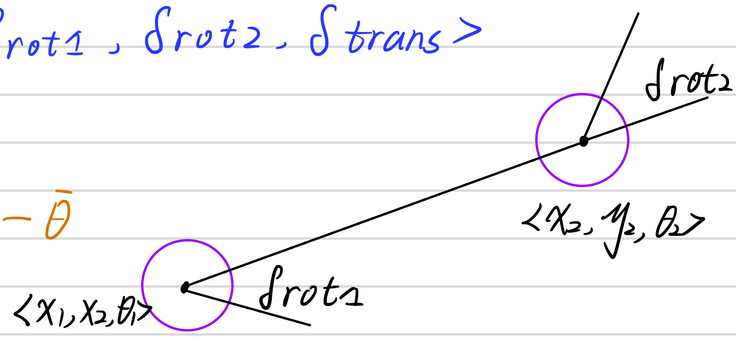
5. Reasons for Motion Errors

1. Different wheel diameters

2. Bump 3. carpet

6. Odometry Model

1. Robot moves from $\langle x_1, y_1, \theta_1 \rangle \Rightarrow \langle x_2, y_2, \theta_2 \rangle$
2. Odometry information $u = \langle \delta_{rot1}, \delta_{rot2}, \delta_{trans} \rangle$



$$\delta_{trans} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$\delta_{rot1} = atan2(y_2 - y_1, x_2 - x_1) - \bar{\theta}$$

$$\delta_{rot2} = \theta_2 - \theta_1 - \delta_{rot1}$$

7. Noise Model for Odometry

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

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

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

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

8. Typical Distribution

1. Normal Distribution
2. Triangular Distribution

9. Calculating The Probability (Zero-Centered)

1. Normal Distribution(a,b)

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

2. Triangular_distribution(a,b)

return $\max\left\{0, \frac{1}{\sqrt{6}\,b} - \frac{|a|}{6b^2}\right\}$

10. Calculating the Posterior, Given x, x', and u

在Odometry模型中,利用之前状态,当前状态,动作,以及选择一个合理的概率模型,对未来状态的值,进行估计.

Algorithm **motion_model_odometry(x,x',u)**

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

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

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

odometry values (u)

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

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

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

values of interest (x,x')

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

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

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

return $p1 \cdot p2 \cdot p3$

重复利用这个Sensor Model,进行短距离移动

11. Sample Odometry Motion Model

Sample Odometry Motion Model

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

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

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

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

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

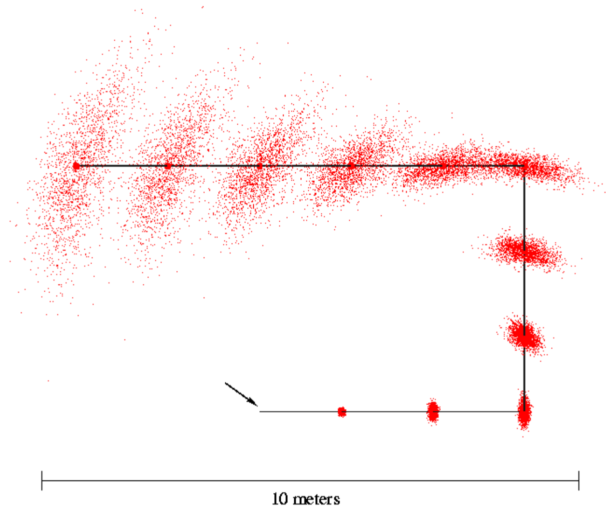
$x' = x + \hat{\delta}_{trans} \cos(\theta + \hat{\delta}_{rot1})$

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

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

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

sample_normal_distribution



这些红点就是机器人未来可能出现的位置

12. Posterier Probability for Velocity Model

Algorithm motion_model_velocity(x_t, u_t, x_{t-1}):

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

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

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

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

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

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

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

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

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

Sampling from Velocity Model

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

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

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

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

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

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

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

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

Examples (velocity based)