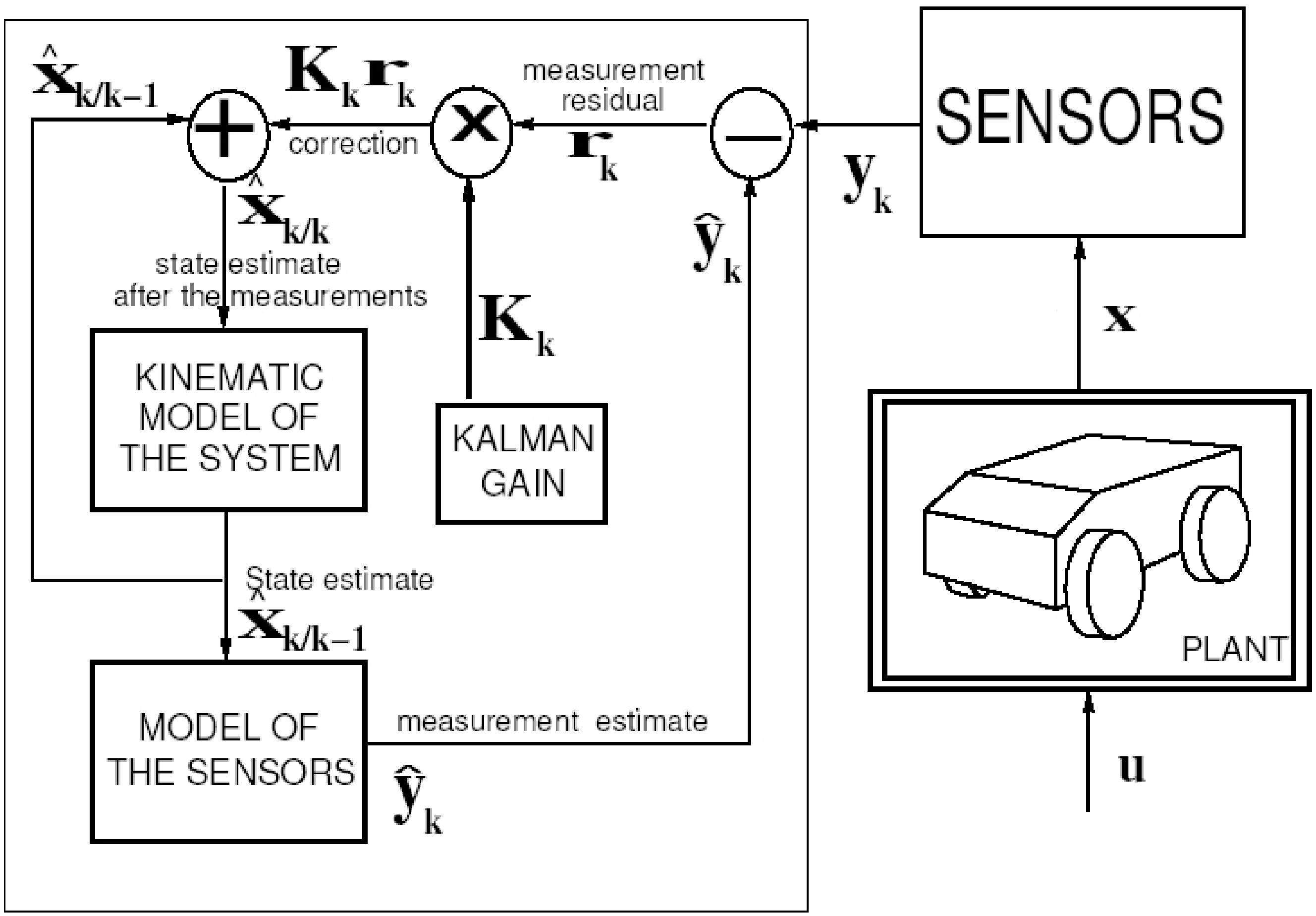
# What we know…
# What we don’t know…

- We know what the control inputs of our process are
  - We know what we’ve told the system to do and have a model for what the expected output should be if everything works right
- We don’t know what the noise in the system truly is
  - We can only estimate what the noise might be and try to put some sort of upper bound on it
- When estimating the state of a system, we try to find a set of values that comes as close to the truth as possible
  - There will always be some mismatch between our estimate of the system and the true state of the system itself. We just try to figure out how much mismatch there is and try to get the best estimate possible

# Kalman Filter Block Diagram



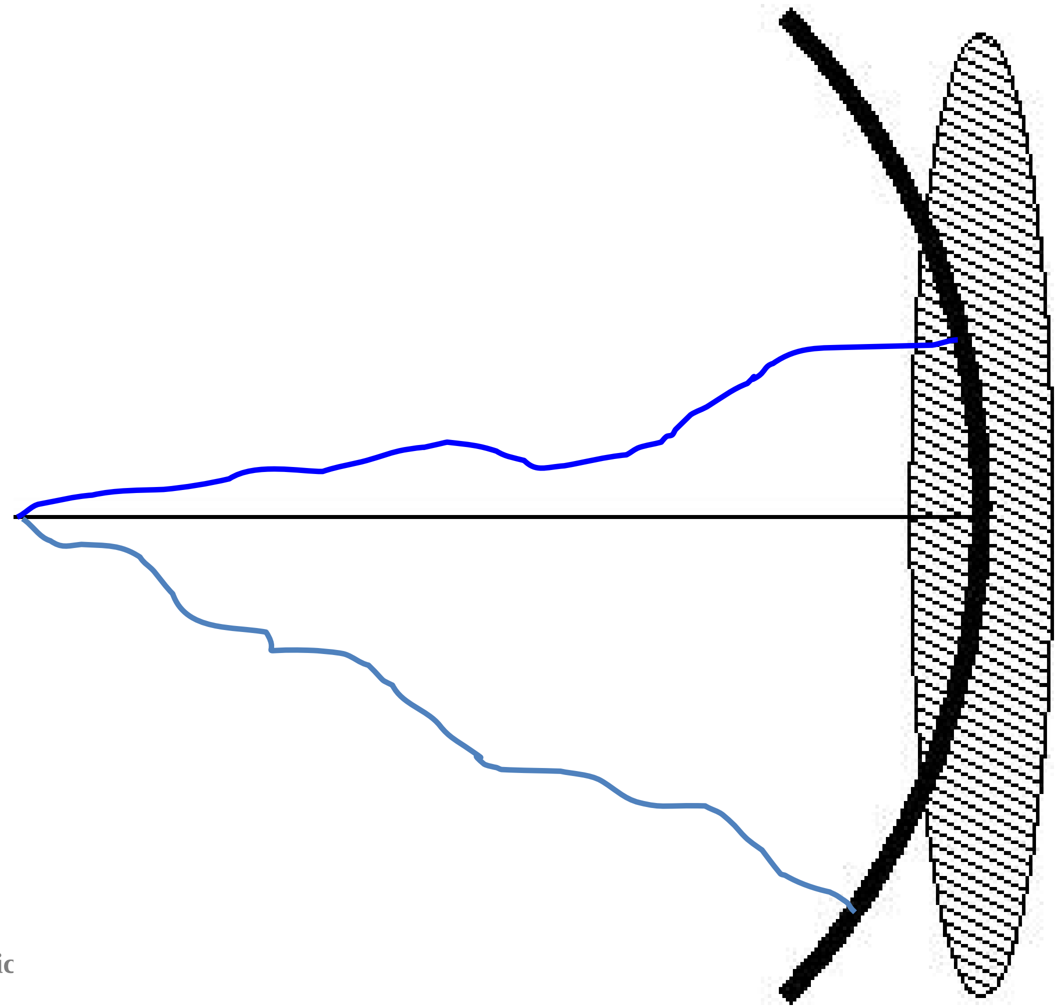
# Using the Kalman Filter for Mobile Robots

- Sensor modeling
  - The odometry estimate is not a reflection of the robot's control system is rather treated as a sensor
  - Instead of directly measuring the error in the state vector (such as when doing tracking), the error in the state must be estimated
  - This is referred to as the Indirect Kalman Filter
- State vector for robot moving in 2D
  - The state vector is 3x1: $[x, y, \theta]$
  - The covariance matrix is 3x3
- Problem: Mobile robot dynamics are NOT linear

# Problems with the Linear Model Assumption

- Many systems of interest are highly non-linear, such as mobile robots
- In order to model such systems, a linear process model must be generated out of the non-linear system dynamics
- The Extended Kalman filter is a method by which the state propagation equations and the sensor models can be linearized about the current state estimate
- Linearization will increase the state error residual because it is not the best estimate

# Approximating Robot Motion Uncertainty with a Gaussian?



# Process Modeling

## Kinematics Modeling

$$x(k+1) = x(k) + r\frac{W_R(k+1) + W_L(k+1)}{2}\cos(\theta(k))$$

$$y(k+1) = y(k) + r\frac{W_R(k+1) + W_L(k+1)}{2}\sin(\theta(k))$$

$$\theta(k+1) = \theta(k) + r\frac{W_R(k+1) - W_L(k+1)}{B}$$

$$x_{k+1} = f(x_k, u_{k+1})$$

## Linearization

$$\nabla_x f = A = \begin{bmatrix} 1 & 0 & -r\frac{W_R(k+1)+W_L(k+1)}{2}\sin(\theta(k)) \\ 0 & 1 & r\frac{W_R(k+1)+W_L(k+1)}{2}\cos(\theta(k)) \\ 0 & 0 & 1 \end{bmatrix}$$

$$\nabla_u f = W = \begin{bmatrix} \frac{r}{2}\cos(\theta(k)) & \frac{r}{2}\cos(\theta(k)) \\ \frac{r}{2}\sin(\theta(k)) & \frac{r}{2}\sin(\theta(k)) \\ \frac{r}{B} & -\frac{r}{B} \end{bmatrix}$$

$$Q_t = \begin{pmatrix} \varepsilon\omega_R^2 & 0 \\ 0 & \varepsilon\omega_L^2 \end{pmatrix}$$ : Process noise

# Sensor Model

Camera or GPS 라면...

$$z_k = h(x_k, v_k)$$

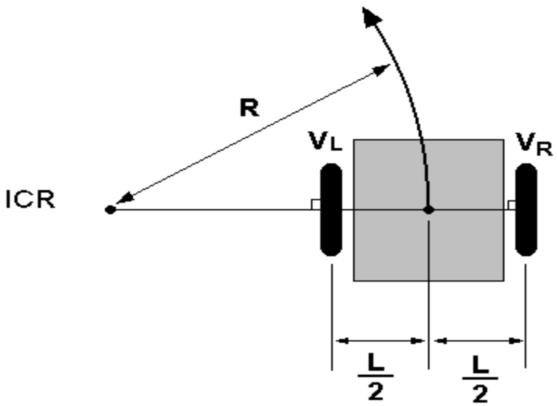
$$R_t = \begin{pmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{pmatrix}$$

: Measurement noise

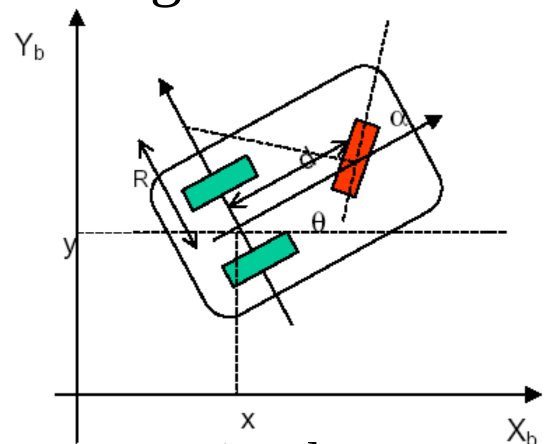
?

# Mobile Robot Locomotion

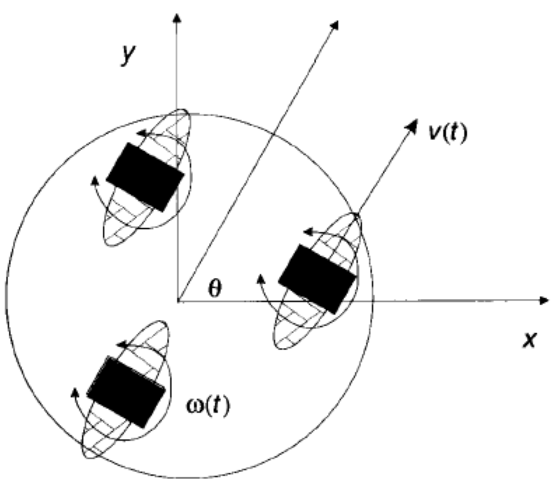
Locomotion: the process of causing a robot to move



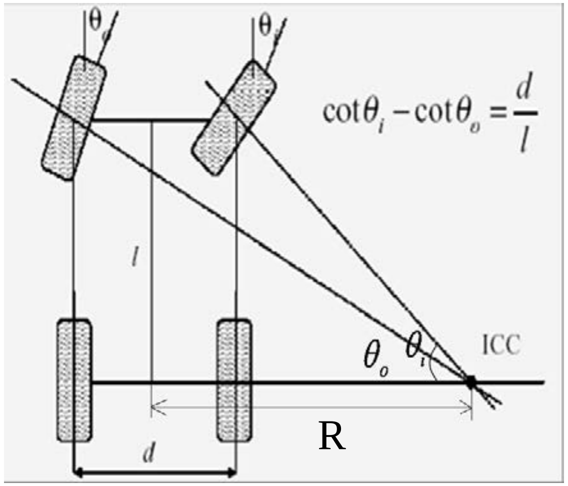

- Differential Drive



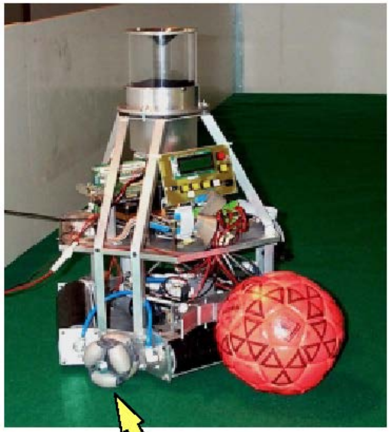

- Tricycle




- Synchronous Drive




- Ackerman Steering



Swedish Wheel

- Omni-directional

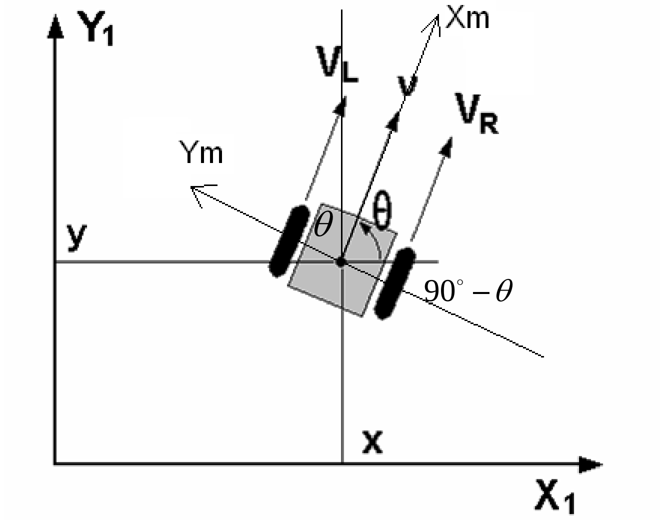
# Differential Drive

Property: At each time instant, the left and right wheels must follow a trajectory that moves around the ICC at the same angular rate ω, i.e.,

$$\boxed{\omega\left(R+\frac{L}{2}\right)=V_R \qquad \omega\left(R-\frac{L}{2}\right)=V_L}$$

$$V_L = r\ \omega_L \qquad V_R = r\ \omega_R$$

$$\omega = \frac{V_R - V_L}{L} \qquad v = \frac{V_R + V_L}{2}$$




- Kinematic equation

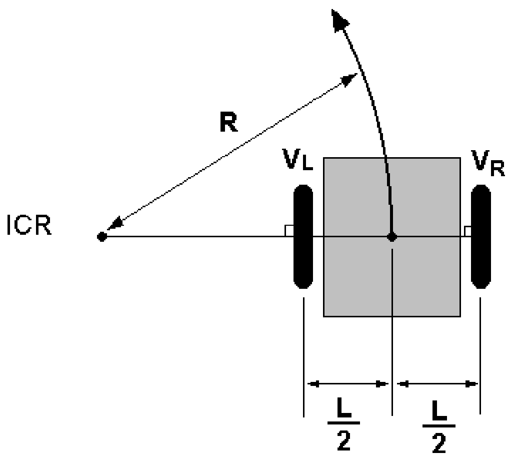
$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}$$

- Nonholonomic Constraint

$$\begin{bmatrix} \sin\theta & -\cos\theta \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \dot{x}\sin\theta - \dot{y}\cos\theta = 0$$

# Differential Drive

- Basic Motion Control




$$(V_R - V_L)/\ L \ = \ V_R\ /(R + \frac{L}{2})$$

$$R = \frac{L}{2}\,\frac{V_R + V_L}{V_R - V_L}$$

R : Radius of rotation

- Straight motion

  R = Infinity → $V_R = V_L$

- Rotational motion

  R = 0 → $V_R = -V_L$

# Tricycle

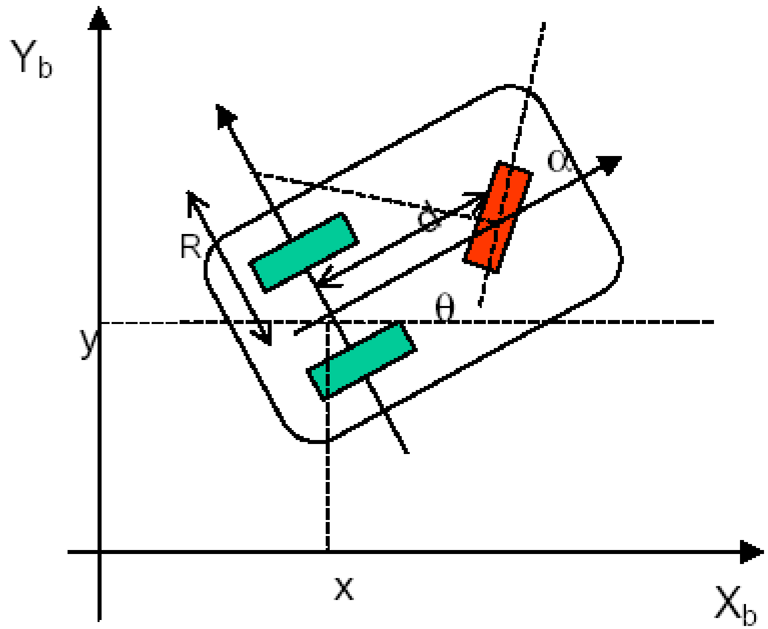
- Steering and power are provided through the front wheel
- control variables:
  - angular velocity of steering wheel $w_s(t)$
  - steering direction $\alpha(t)$

r = steering wheel radius

$$v_s(t) = w_s(t)\, r$$ linear velocity of steering wheel

$$R(t) = d\, tg\left(\pi/2 - \alpha(t)\right)$$

$$w(t) = \frac{w_s(t)\, r}{\sqrt{d^2 + R(t)^2}}$$ angular velocity of the moving fram relative to the base frame




$$w(t) = \frac{v_s(t)}{d} \sin\alpha(t)$$

*d*: distance from the front wheel to the rear axle

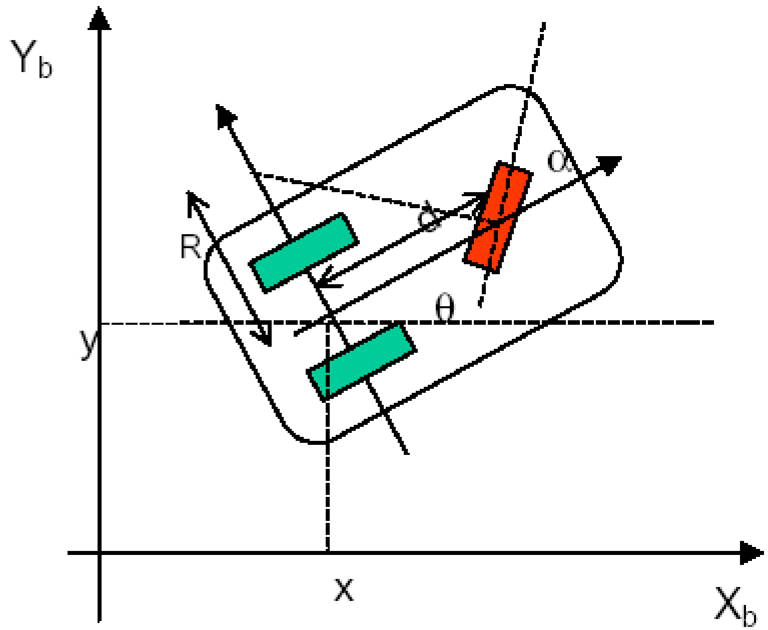
# Tricycle

**Kinematics model in the world frame**
**---Posture kinematics model**

$$\dot{x}(t) = v_s(t)\cos\alpha(t)\cos\theta(t)$$
$$\dot{y}(t) = v_s(t)\cos\alpha(t)\sin\theta(t)$$
$$\dot{\theta}(t) = \frac{v_s(t)}{d}\sin\alpha(t)$$

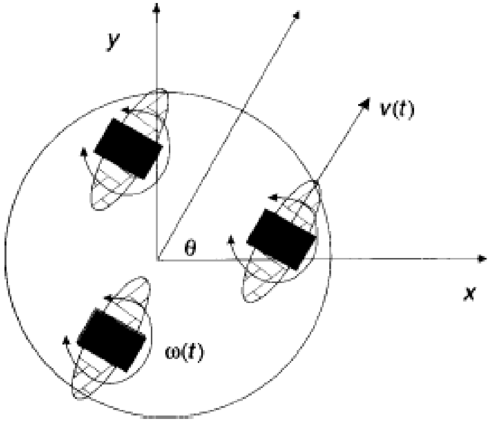
$$\begin{bmatrix}\dot{x}(t)\\ \dot{y}(t)\\ \dot{\theta}(t)\end{bmatrix} = \begin{bmatrix}\cos\theta(t) & 0\\ \sin\theta(t) & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}v(t)\\ w(t)\end{bmatrix}$$

$$v(t) = v_s(t)\cos\alpha(t)$$
$$w(t) = \frac{v_s(t)}{d}\sin\alpha(t)$$



# Synchronous Drive

- All the wheels turn in unison
  - All wheels point in the same direction and turn at the same rate
  - Two independent motors, one rolls all wheels forwar' one rotate them for turnin
- Control variables (independent)
  - v(t), ω(t)


$$x(t) = \int_0^t v(\sigma)\cos(\theta(\sigma))d\sigma$$

$$y(t) = \int_0^t v(\sigma)\sin(\theta(\sigma))d\sigma$$

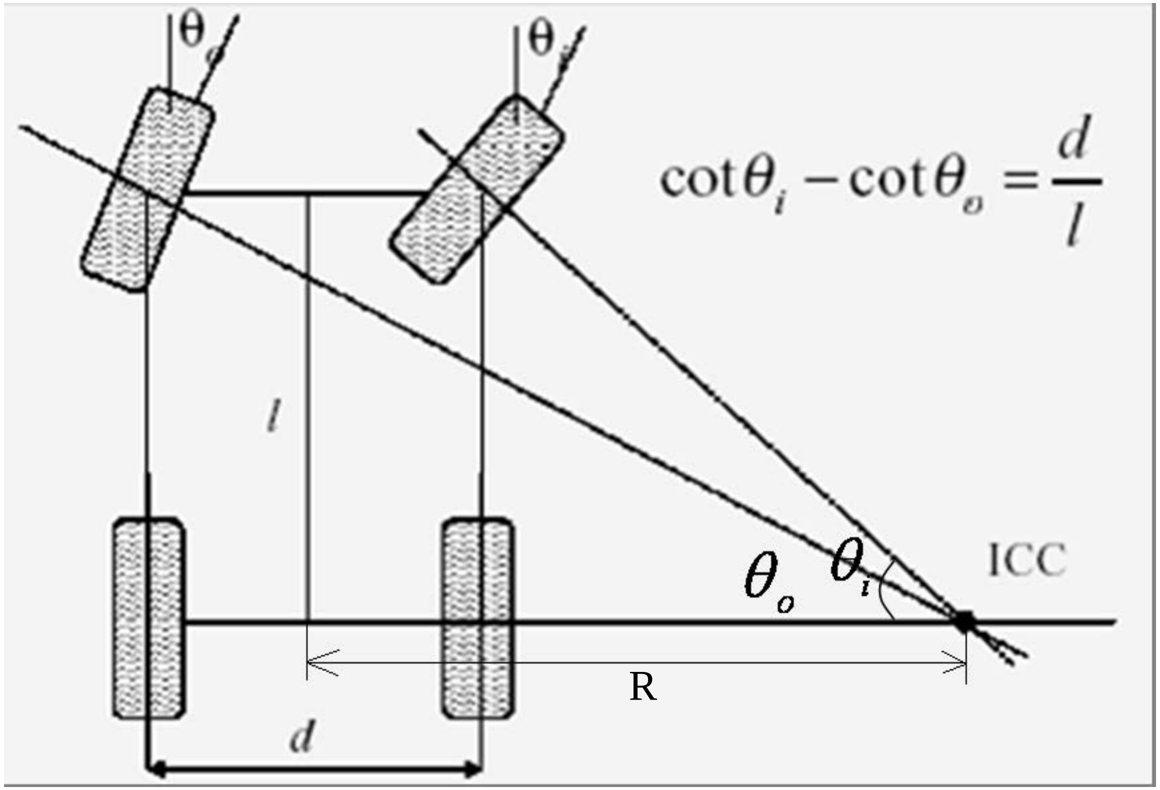
$$\theta(t) = \int_0^t w(\sigma)d\sigma$$

# Ackerman Steering (Car Drive)

- The Ackerman Steering equation:
  - :

$$\cot\theta_i - \cot\theta_o = \frac{d}{l}$$

$$\cot\theta = \frac{\cos\theta}{\sin\theta}$$

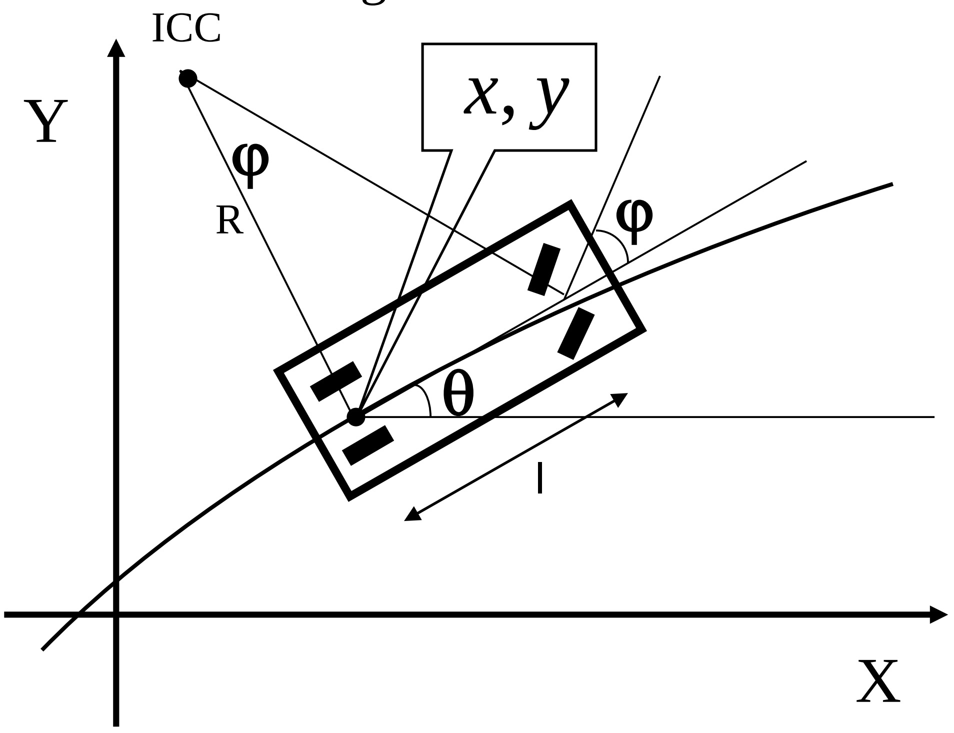



$$\begin{aligned}&\cot\theta_i - \cot\theta_o \\ &= \frac{R + d/2}{l} - \frac{R - d/2}{l} \\ &= \frac{d}{l}\end{aligned}$$

# Car-like Robot

Driving type: Rear wheel drive, front wheel steering

$$\dot{\theta} \cdot R = u_1 \Rightarrow \dot{\theta} \frac{l}{\tan \varphi} = u_1$$




Rear wheel drive car model:

$$\dot{x} = u_1 \cos \theta$$

$$\dot{y} = u_1 \sin \theta$$

$$\dot{\theta} = \frac{u_1}{l} \tan \varphi$$

$$\dot{\varphi} = u_2$$

non-holonomic constraint:

$$\dot{x} \sin \theta - \dot{y} \cos \theta = 0$$

$u_1$ : forward velocity of the rear wheels

$u_2$ : angular velocity of the steering wheels

$l$ : length between the front and rear wheels

# Extended Kalman Filter for SLAM

- State vector
  - Expanded to contain entries for all landmarks positions:
  - State vector can be grown as new landmarks are discovered
  - Covariance matrix is also expanded

$$X = \begin{bmatrix} X_R^T & X_{L_1}^T & \ldots & X_{L_n}^T \end{bmatrix}^T$$

$$\begin{bmatrix} P_{RR} & P_{RL_1} & \cdots & P_{RL_N} \\ P_{L_1R} & P_{L_1L_1} & \cdots & P_{L_1L_N} \\ \vdots & \vdots & \ddots & \vdots \\ P_{L_NR} & P_{L_NL_1} & \cdots & P_{L_NL_N} \end{bmatrix}$$

# Extended Kalman Filter for SLAM

- Kinematic equations for landmark propagation

$$\hat{x}_{t+1} = \hat{x}_t + (V_t + w_{V_t})\delta t \cos \hat{\phi}_t$$
$$\hat{y}_{t+1} = \hat{y}_t + (V_t + w_{V_t})\delta t \sin \hat{\phi}_t$$
$$\hat{\phi}_{t+1} = \hat{\phi}_t + (\omega_t + w_{\omega_t})\delta t$$
$$\hat{x}_{L_i t+1} = \hat{x}_{L_i t}$$
$$\hat{y}_{L_i t+1} = \hat{y}_{L_i t}$$
$$\hat{\phi}_{L_i t+1} = \hat{\phi}_{L_i t}$$

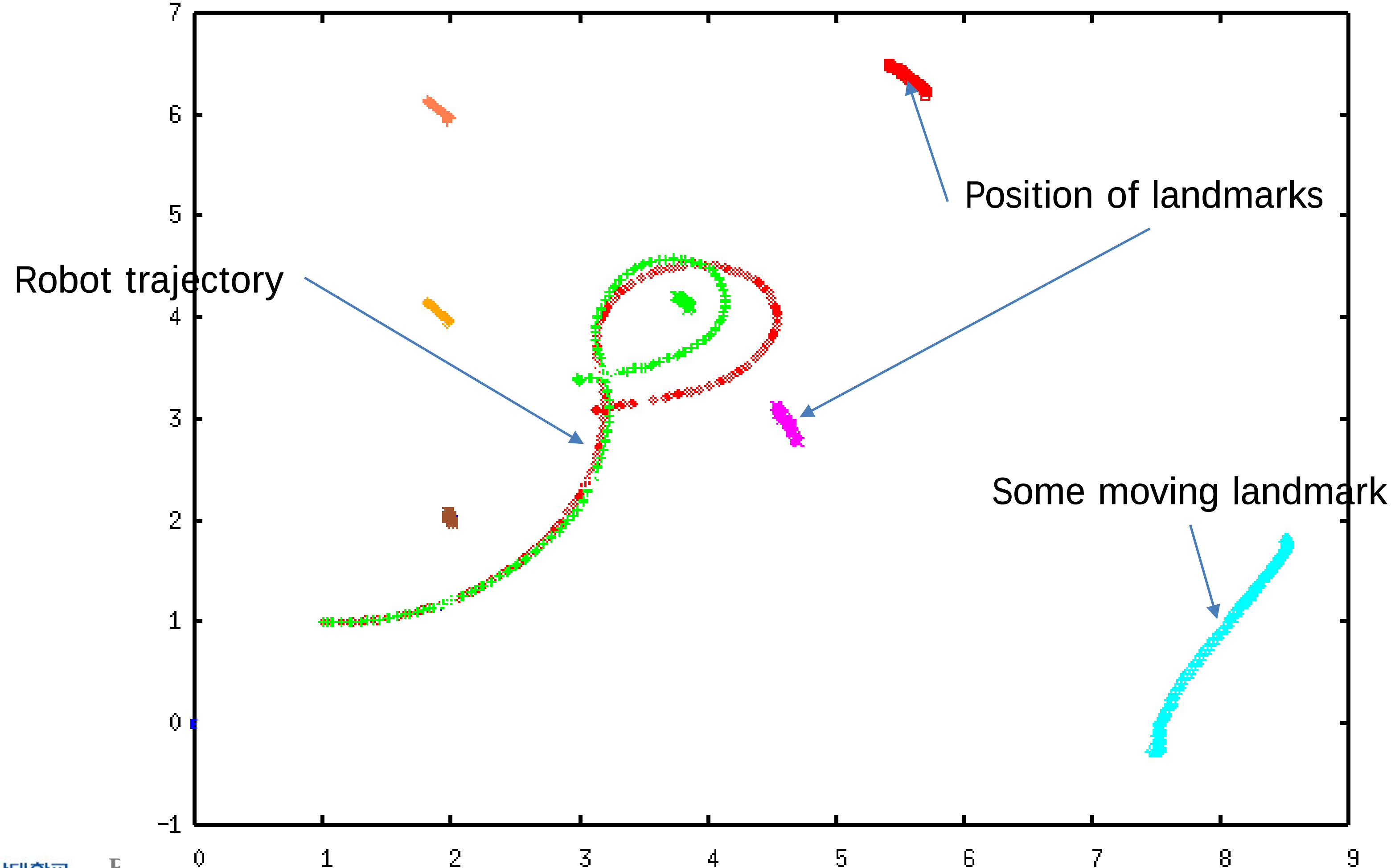
# Extended Kalman Filter for SLAM

- Sensor equations for update:

$$\tilde{X} = \begin{bmatrix} \tilde{X}_R^T & \tilde{X}_{L_1}^T & \cdots & \tilde{X}_{L_i}^T & \cdots & \tilde{X}_{L_n}^T \end{bmatrix}$$
$$H = \begin{bmatrix} H_R & 0 & \cdots & 0 & H_{L_i} & 0 & \cdots & 0 \end{bmatrix}$$

- Very powerful because covariance update records *shared information* between landmarks and robot positions

# EKF for SLAM



Position of landmarks
Robot trajectory
Some moving landmark
7
6
5
4
3
2
1
0
−1
0
1
2
3
4
5
6
7
8
9