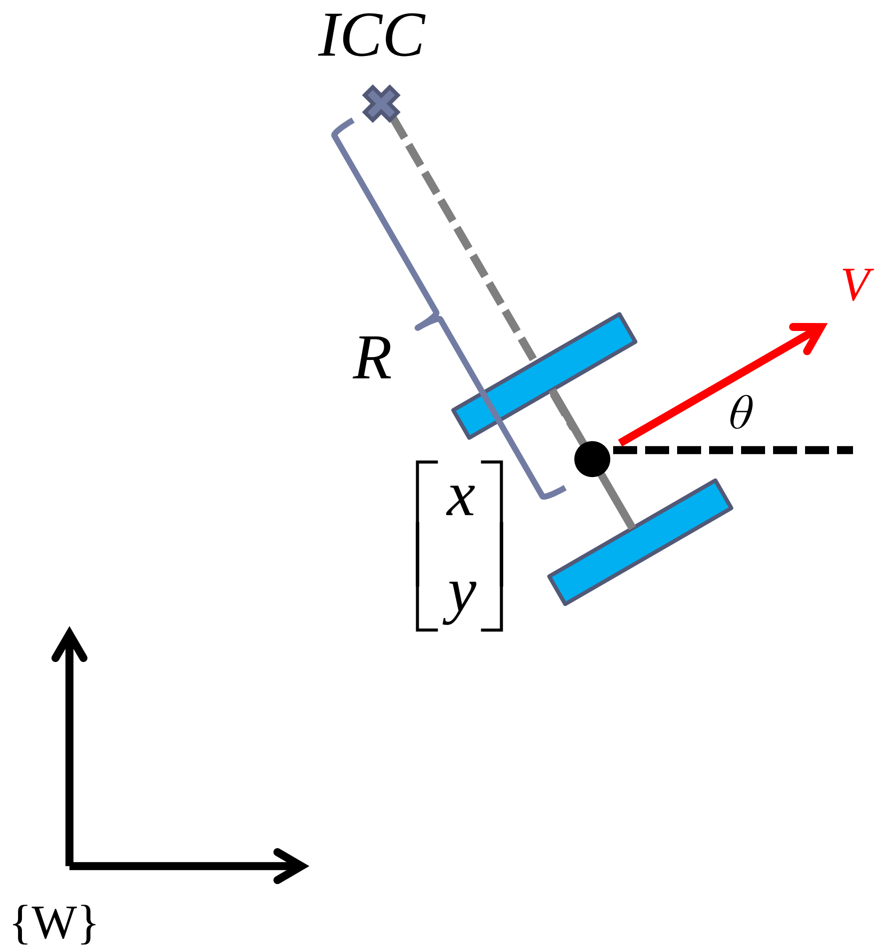
Mobile Robot Forward Kinematics

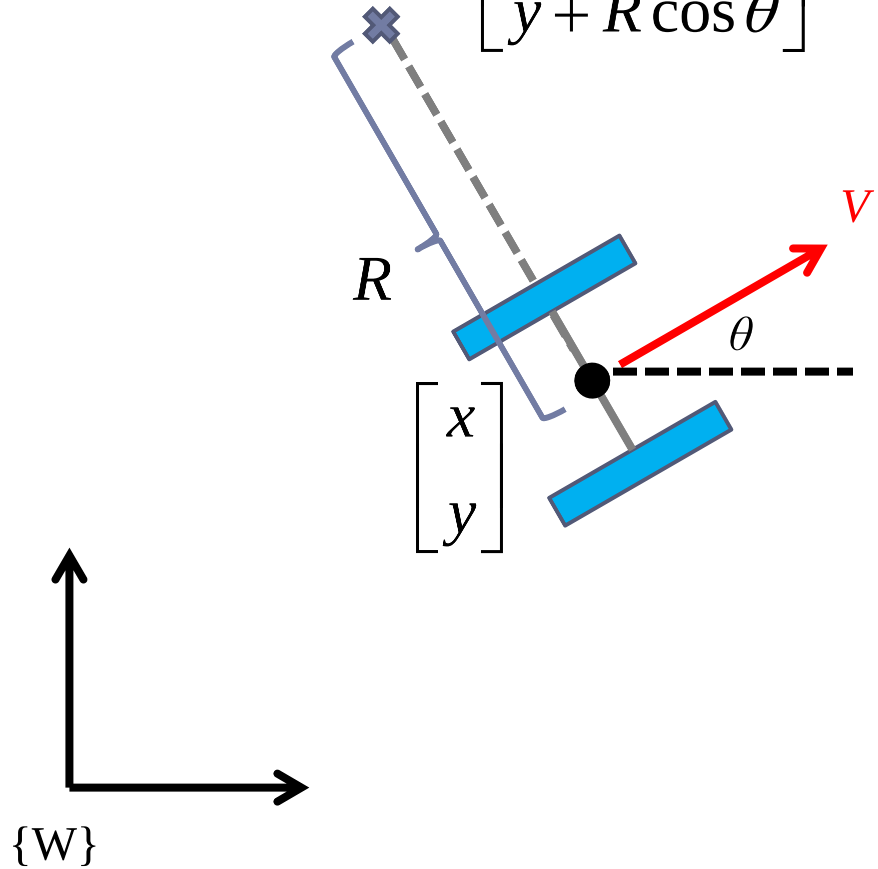
Forward Kinematics : Differential Drive

- what is the position of the ICC in {W}?



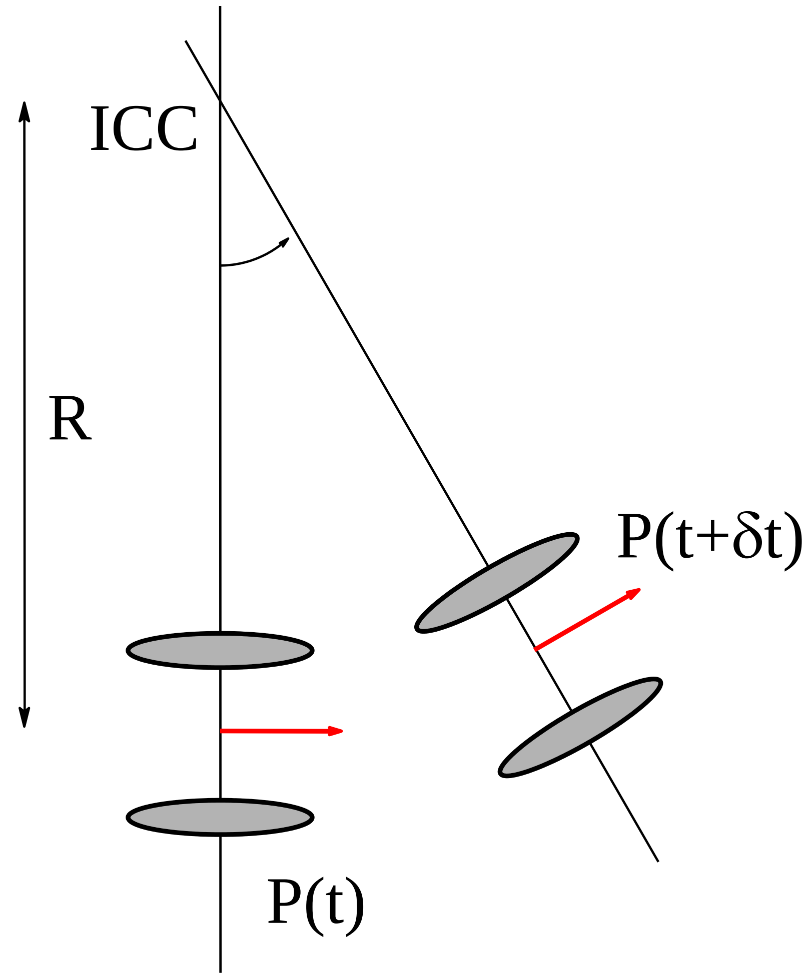
Forward Kinematics : Differential Drive

$$ICC = \begin{bmatrix} x - R\sin\theta \\ y + R\cos\theta \end{bmatrix}$$



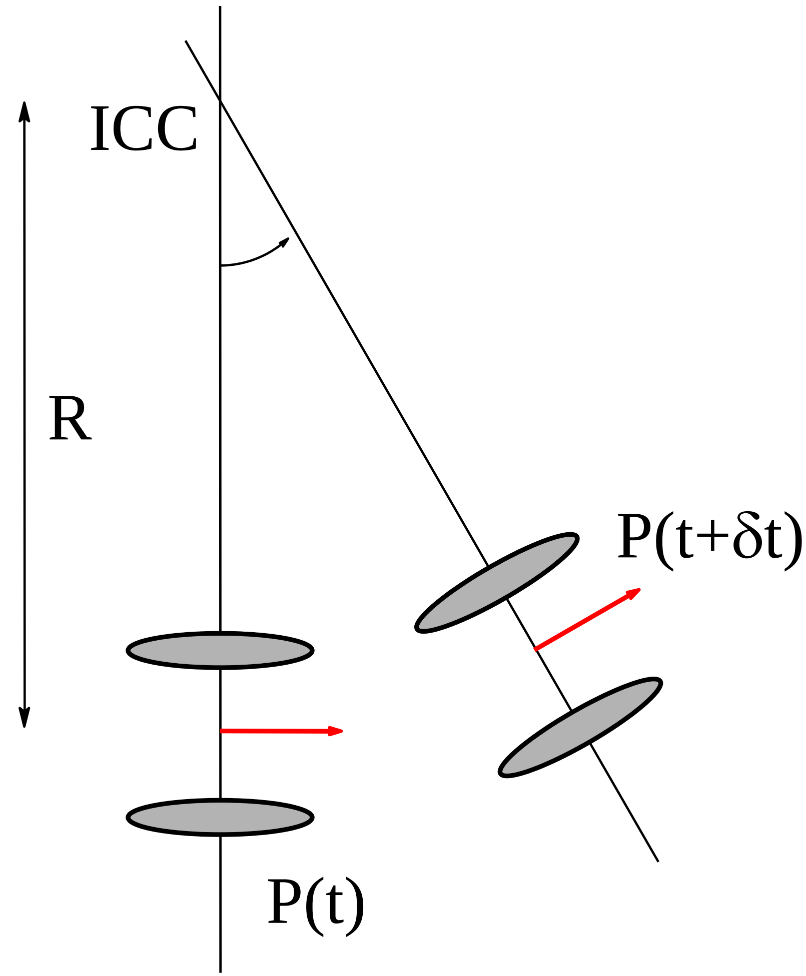
Forward Kinematics : Differential Drive

- assuming smooth rolling motion at each point in time the differential drive is moving in a circular path centered on the ICC
  - thus, for a small interval of time δt the change in pose can be computed as a rotation about the ICC



Forward Kinematics : Differential Drive

- **computing the rotation about the ICC**
  1. translate so that the ICC moves to the origin of {W}
  2. rotate about the origin of {W}
  3. translate back to the original ICC



Forward Kinematics : Differential Drive

- computing the rotation about the ICC
  1. translate so that the ICC moves to the origin of {W}
  2. rotate about the origin of {W}
  3. translate back to the original ICC

$$ICC = \begin{bmatrix} x - R\sin\theta \\ y + R\cos\theta \end{bmatrix} = \begin{bmatrix} ICC_x \\ ICC_y \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} ICC_x \\ ICC_y \end{bmatrix} = \begin{bmatrix} x - ICC_x \\ y - ICC_y \end{bmatrix}$$

Forward Kinematics : Differential Drive

- computing the rotation about the ICC
  1. translate so that the ICC moves to the origin of {W}
  2. rotate about the origin of {W}
  3. translate back to the original ICC
- how much rotation over the time interval?
  - angular velocity * elapsed time = $\omega\delta t$

$$\begin{bmatrix} \cos(\omega\delta t) & -\sin(\omega\delta t) \\ \sin(\omega\delta t) & \cos(\omega\delta t) \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \end{bmatrix}$$

Forward Kinematics : Differential Drive

- computing the rotation about the ICC
  1. translate so that the ICC moves to the origin of {W}
  2. rotate about the origin of {W}
  3. translate back to the original ICC

$$\begin{bmatrix} x(t+\delta t) \\ y(t+\delta t) \end{bmatrix} = \begin{bmatrix} \cos(\omega\delta t) & -\sin(\omega\delta t) \\ \sin(\omega\delta t) & \cos(\omega\delta t) \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \end{bmatrix}$$

Forward Kinematics : Differential Drive

- what about the orientation $\theta(t+\delta t)$?
  - just add the rotation for the time interval
- new pose

$$\begin{bmatrix} x(t+\delta t) \\ y(t+\delta t) \end{bmatrix} = \begin{bmatrix} \cos(\omega\delta t) & -\sin(\omega\delta t) \\ \sin(\omega\delta t) & \cos(\omega\delta t) \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \end{bmatrix}$$

$$\theta(t+\delta t) = \theta + \omega\delta t$$

- which can be written as

$$\begin{bmatrix} x(t+\delta t) \\ y(t+\delta t) \\ \theta(t+\delta t) \end{bmatrix} = \begin{bmatrix} \cos(\omega\delta t) & -\sin(\omega\delta t) & 0 \\ \sin(\omega\delta t) & \cos(\omega\delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \\ \omega\delta t \end{bmatrix}$$

Forward Kinematics: Differential Drive

- the previous equation is valid if $v_L \neq v_R$
  - i.e., if the differential drive is not travelling in a straight line
- if $v_L = v_R = v$ then

$$\begin{bmatrix} x(t+\delta t) \\ y(t+\delta t) \\ \theta(t+\delta t) \end{bmatrix} = \begin{bmatrix} x + v\delta t \cos\theta \\ y + v\delta t \sin\theta \\ \theta \end{bmatrix}$$

Sensitivity to Wheel Velocity

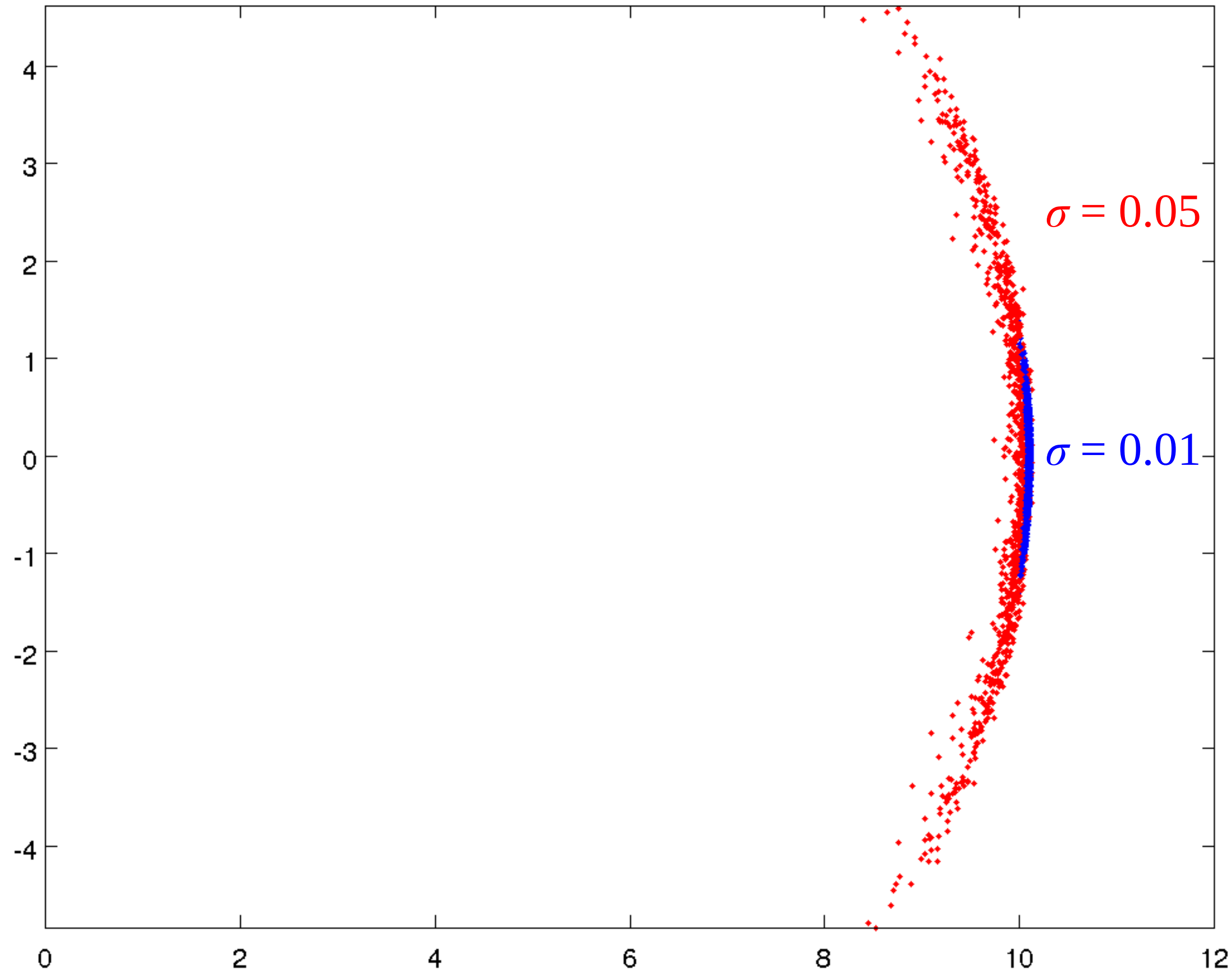
$$v_r(t) = 1 + \mathcal{N}(0, \sigma^2)$$

$$v_\ell(t) = 1 + \mathcal{N}(0, \sigma^2)$$

$$\theta(0) = 0$$

$$t = 0 \ldots 10$$

$$\ell = 0.2$$



Sensitivity to Wheel Velocity

- given the forward kinematics of the differential drive it is easy to write a simulation of the motion
  - we need a way to draw random numbers from a normal distribution
  - in Matlab
    - **`randn(n)`** returns an n-by-n matrix containing pseudorandom values drawn from the standard normal distribution
    - see **`mvnrnd`** for random values from a multivariate normal distribution

Sensitivity to Wheel Velocity

```
POSE = [];          % final pose of robot after each trial
sigma = 0.01;       % noise standard deviation
L = 0.2;            % distance between wheels
dt = 0.1;           % time step
TRIALS = 1000;      % number of trials

for trial = 1:TRIALS

    -run each trial-
    see next slide

end
```

Sensitivity to Wheel Velocity

```
vr = 1;                   % initial right-wheel velocity
vl = 1;                   % initial left-wheel velocity
pose = [0; 0; 0];         % initial pose of robot

for t = 0:dt:10
```

-move the robot one time step -
see next slide

```
end
POSE = [POSE pose];   % record final pose after trial t
```

Sensitivity to Wheel Velocity

```
theta = pose(3);
if vr == vl
   pose = pose + [vr * cos(theta) * dt;
                  vr * sin(theta) * dt;
                  0];
else
   omega = (vr – vl) / L;
   R = (L / 2) * (vr + vl) / (vr – vl);
   ICC = pose + [-R * sin(theta);
                  R * cos(theta);
                  0];
   pose = rz(omega * dt) * (pose – ICC) + ICC +
          [0; 0; omega * dt];
end
vr = 1 + sigma * randn(1);
vl = 1 + sigma * randn(1);
```