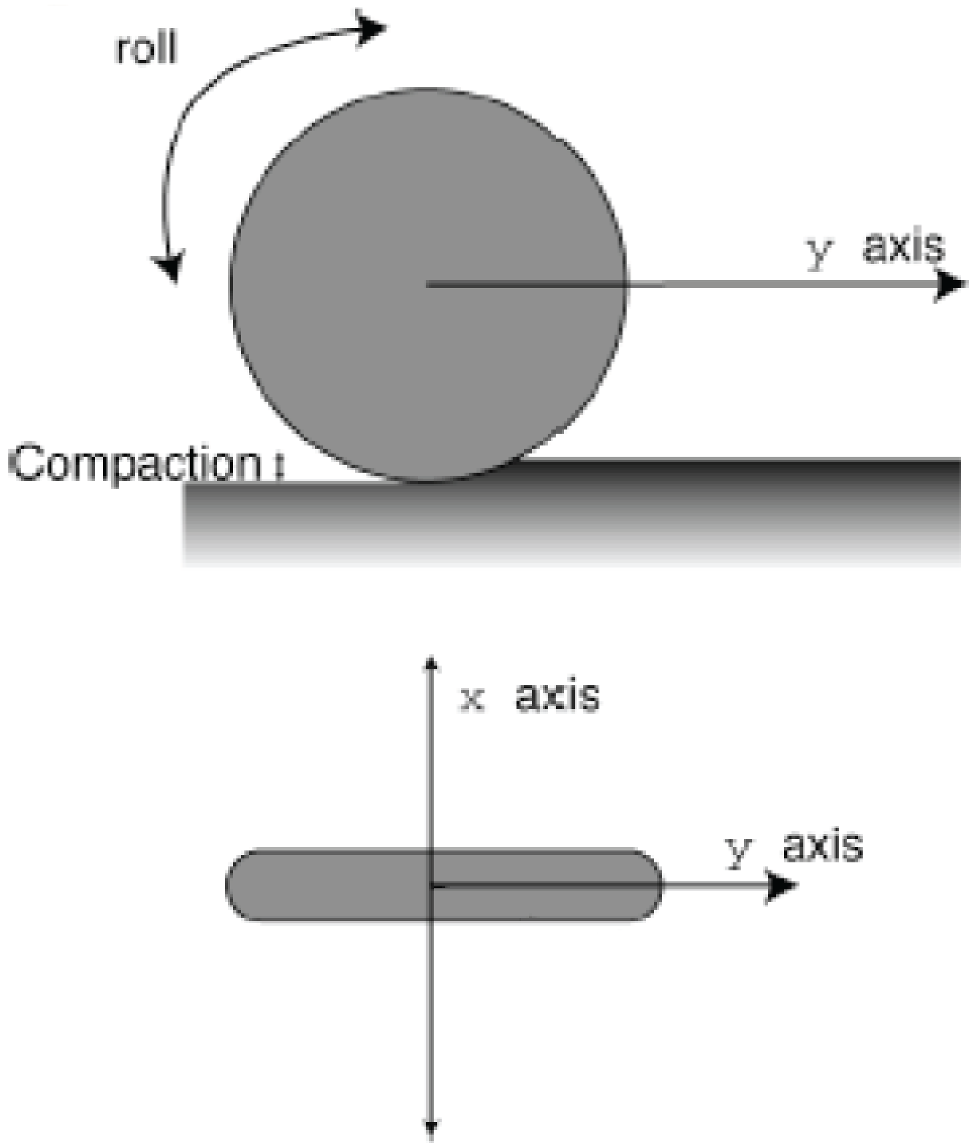
Kinematics of Wheeled Robots

- https://www.youtube.com/watch?v=giS41utjlbU

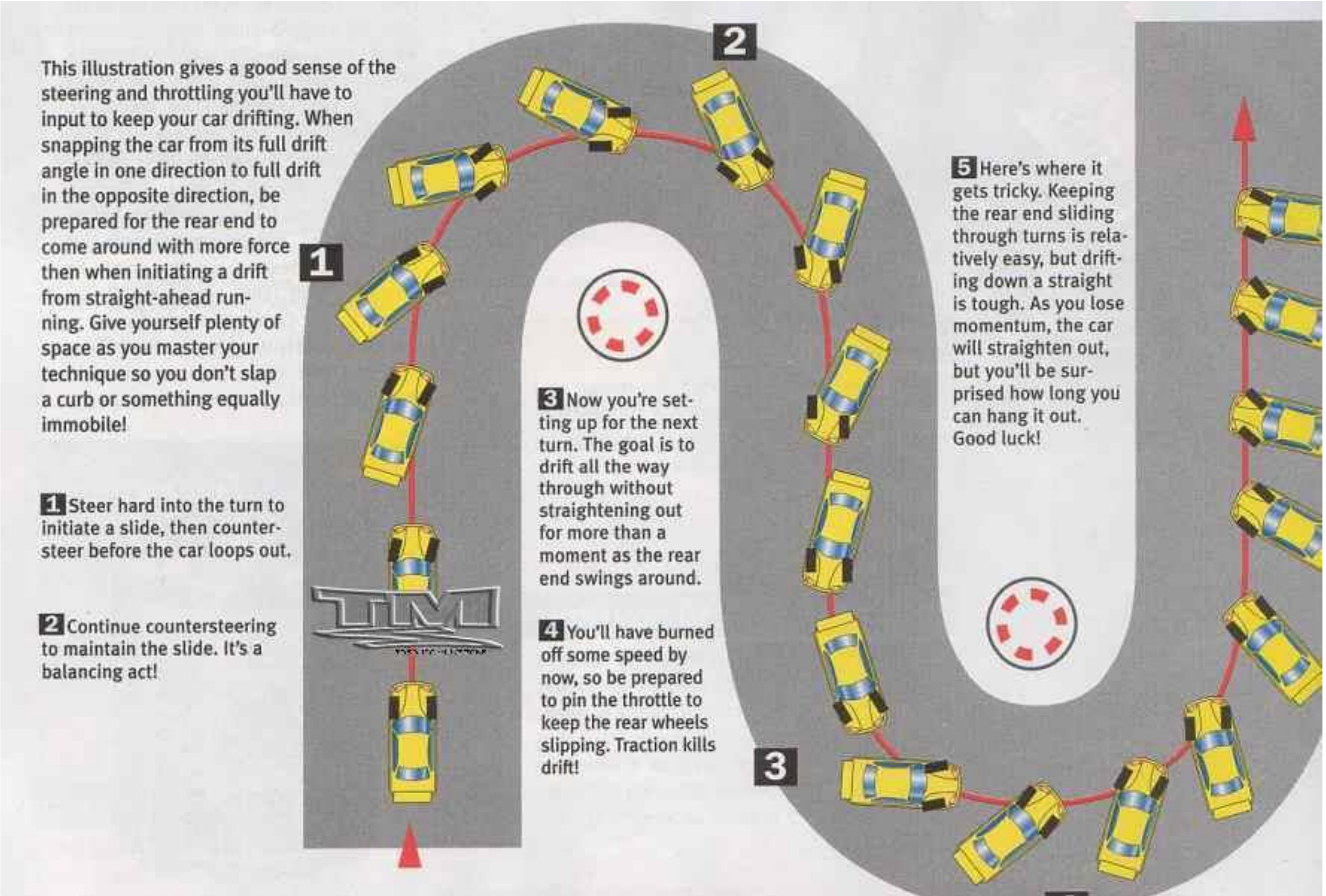
Wheeled Mobile Robots

- robot can have one or more wheels that can provide
  - steering (directional control)
  - power (exert a force against the ground)
- an ideal wheel is
  - perfectly round (perimeter $2\pi r$)
  - moves in the direction perpendicular to its axis

Wheel



Deviations from Ideal

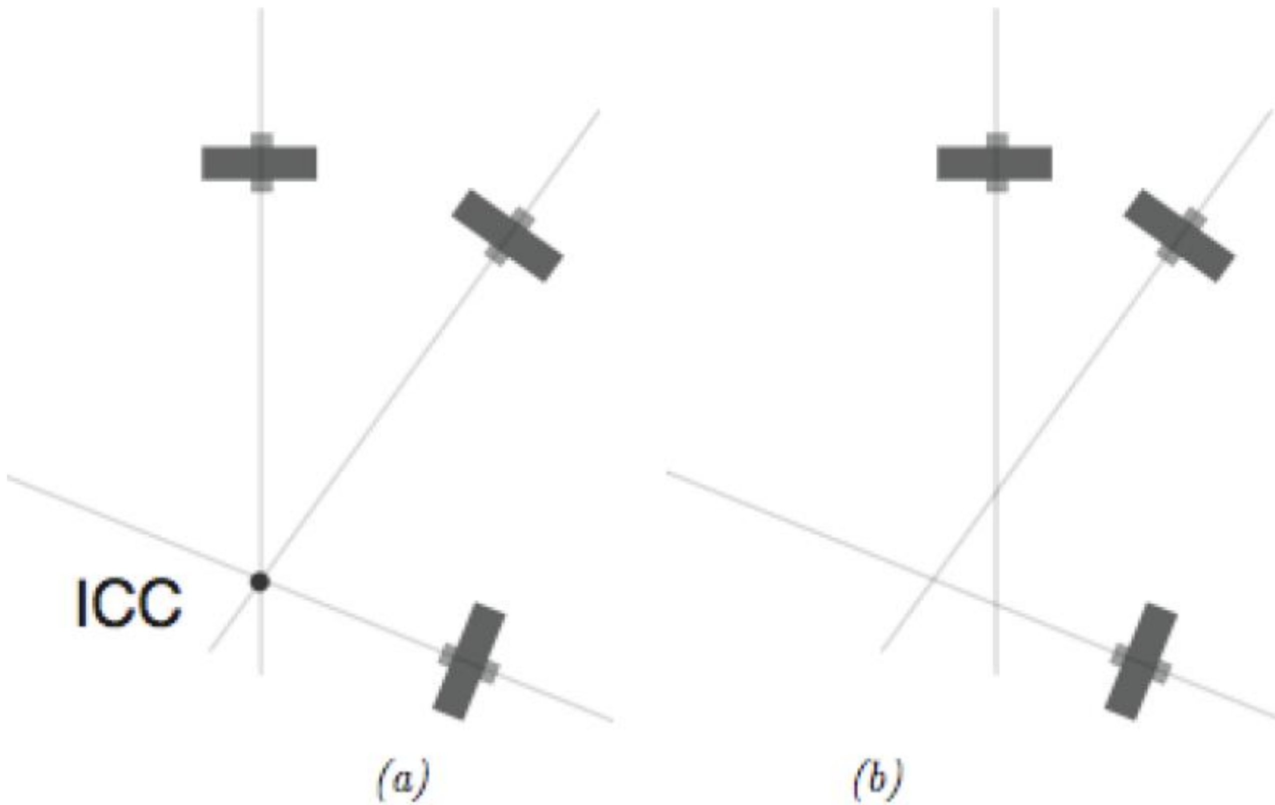


http://worlddrifting.com/wp-content/uploads/2010/02/drifting-techniques.jpg

Instantaneous Center of Curvature

- for smooth rolling motion, all wheels in ground contact must
  - follow a circular path about a common axis of revolution
    - each wheel must be pointing in its correct direction
  - revolve with an angular velocity consistent with the motion of the robot
    - each wheel must revolve at its correct speed

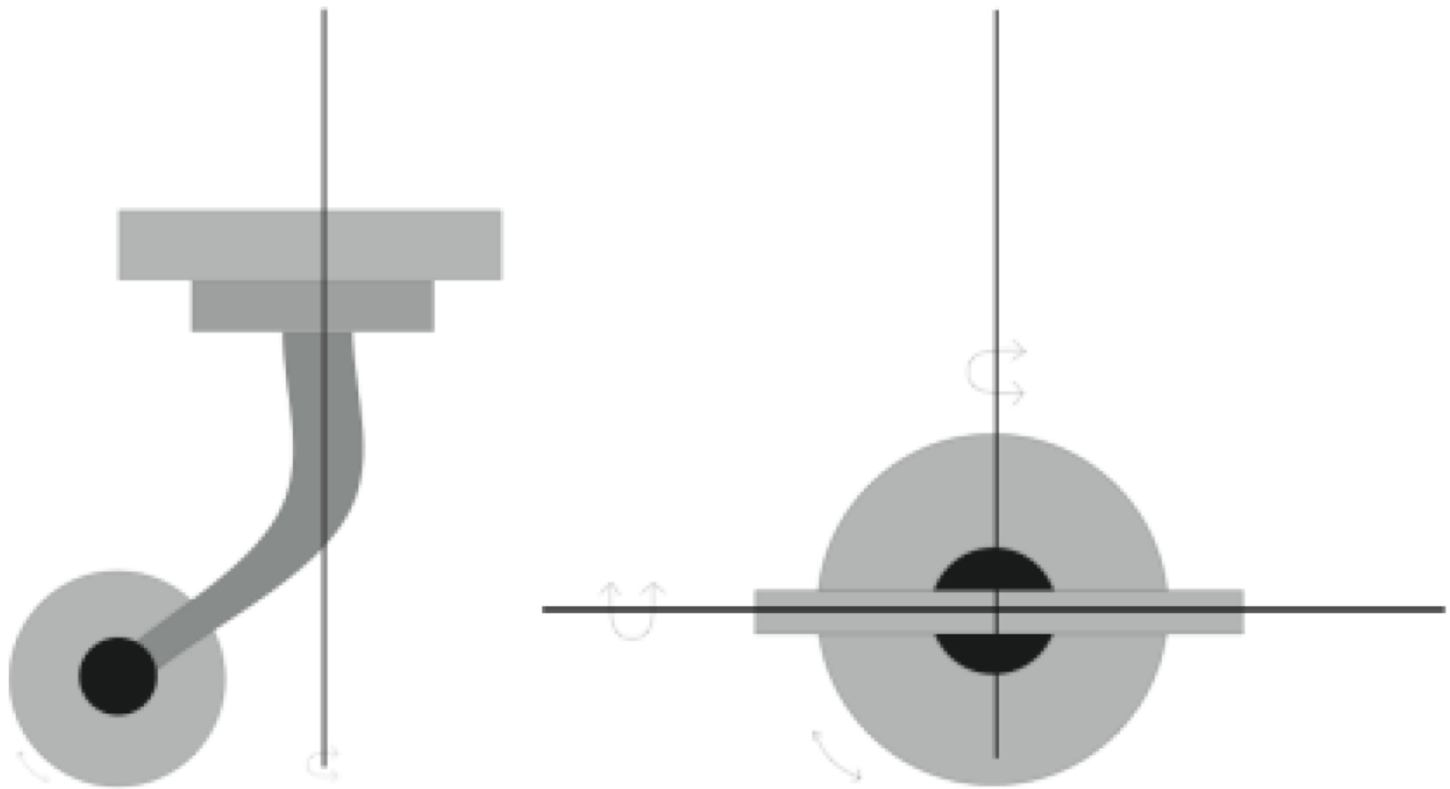
Instantaneous Center of Curvature



(a) 3 wheels with roll axes intersecting at a common point (the instantaneous center of curvature, ICC). (b) No ICC exists. A robot having wheels shown in (a) can exhibit smooth rolling motion, whereas a robot with wheel arrangement (b) cannot.

Castor Wheels

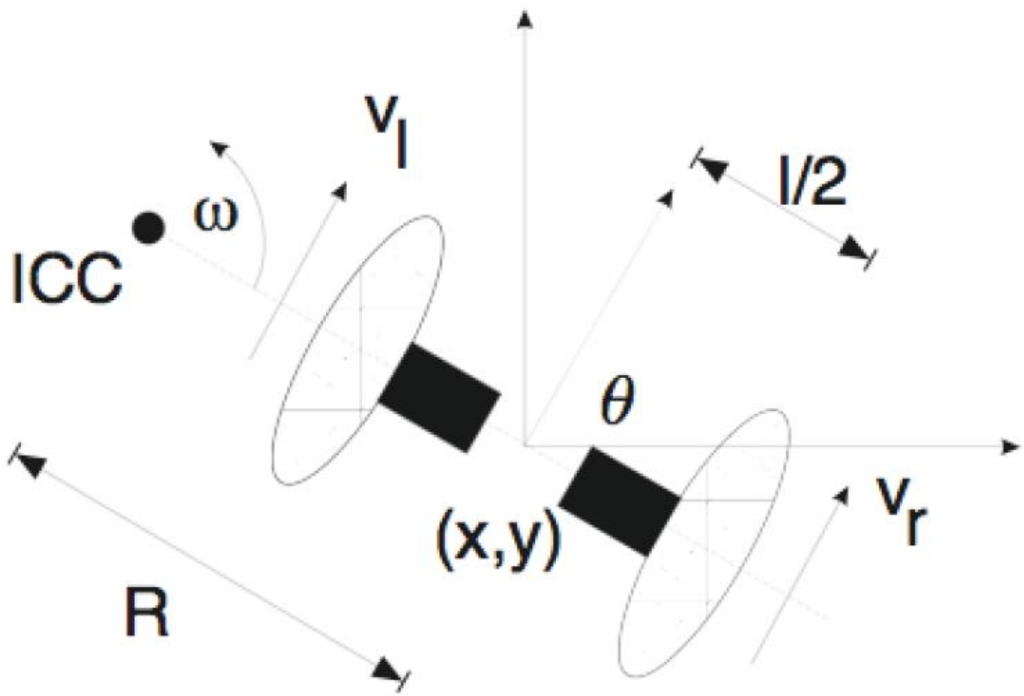
- provide support but not steering nor propulsion



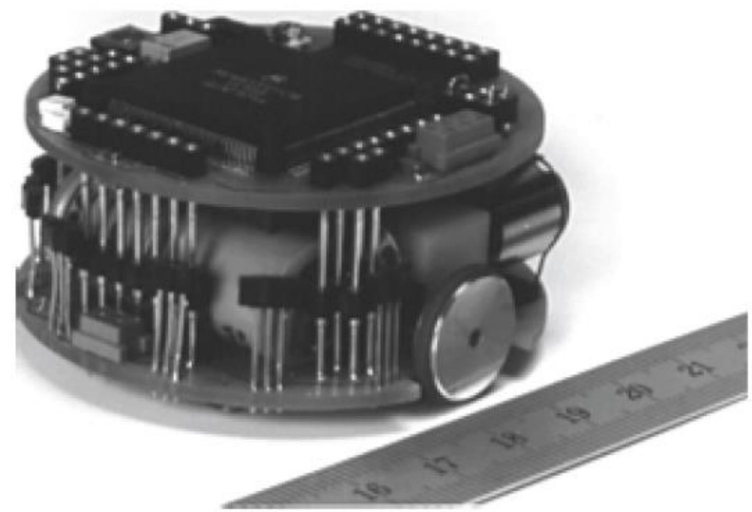
(a) Castor Wheel (b) Rollerball wheel

Differential Drive

- two independently driven wheels mounted on a common axis



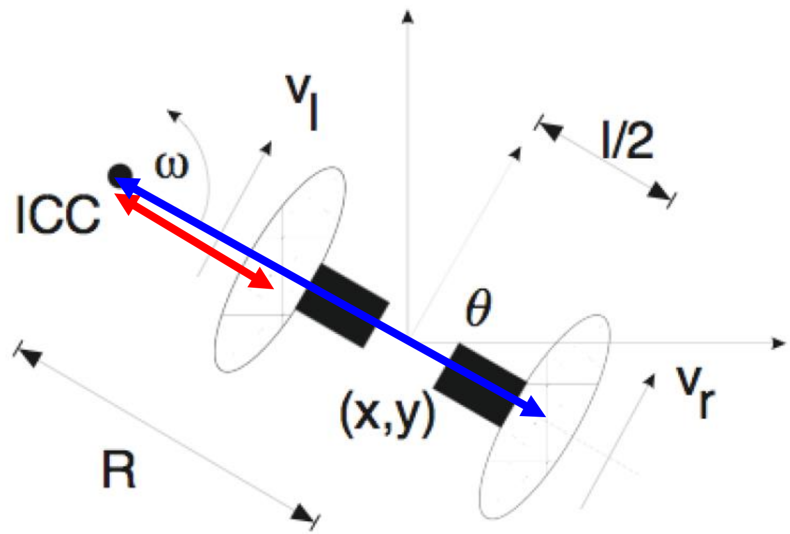
(a) Differential Drive



(b) Khepera Robot

Differential Drive

- angular velocity ω about the ICC defines the wheel ground velocities v_r and v_ℓ



(a) Differential Drive

$$v_r = \omega(\overbrace{R + \frac{\ell}{2}}^{\text{distance between ICC and right wheel}})$$

$$v_\ell = \omega(\underbrace{R - \frac{\ell}{2}}_{\text{distance between ICC and left wheel}})$$

https://opencurriculum.org/5481/circular-motion-linear-and-angular-speed/

Differential Drive

- given the wheel ground velocities it is easy to solve for the radius, R, and angular velocity ω

$$R = \frac{\ell}{2}\frac{(v_r + v_\ell)}{(v_r - v_\ell)}$$

$$\omega = \frac{(v_r - v_\ell)}{\ell}$$

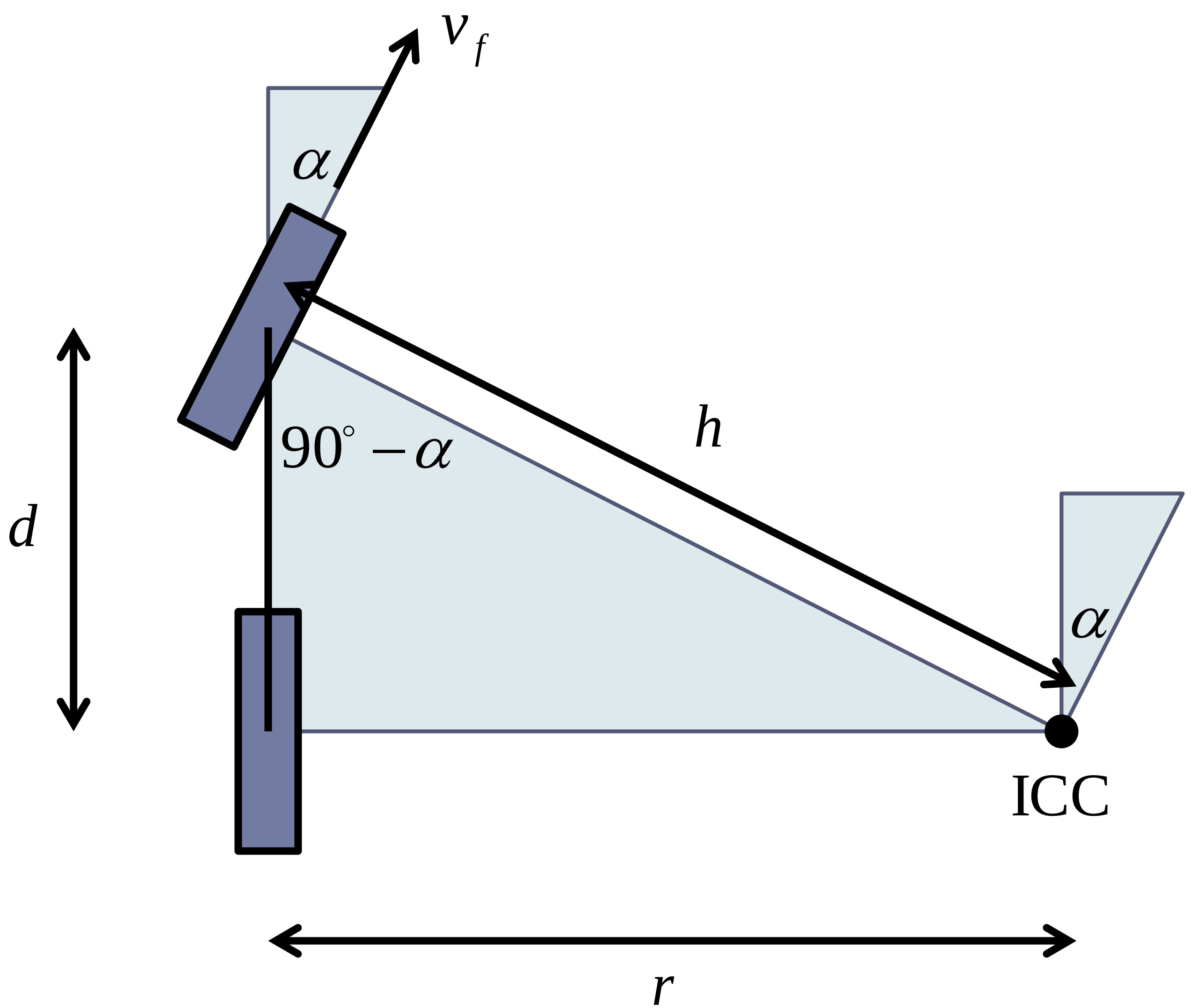
- interesting cases:
  - $v_\ell = v_r$
  - $v_\ell = -v_r$

Tracked Vehicles

- similar to differential drive but relies on ground slip or skid to change direction
  - kinematics poorly determined by motion of treads

http://en.wikipedia.org/wiki/File:Tucker-Kitten-Variants.jpg

Steered Wheels: Bicycle



Steered Wheels: Bicycle

- important to remember the assumptions in the kinematic model
  - smooth rolling motion in the plane
- does not capture all possible motions
  - http://www.youtube.com/watch?v=Cj6ho1-G6tw&NR=1#t=0m25s

Mecanum Wheel

- a normal wheel with rollers mounted on the circumference

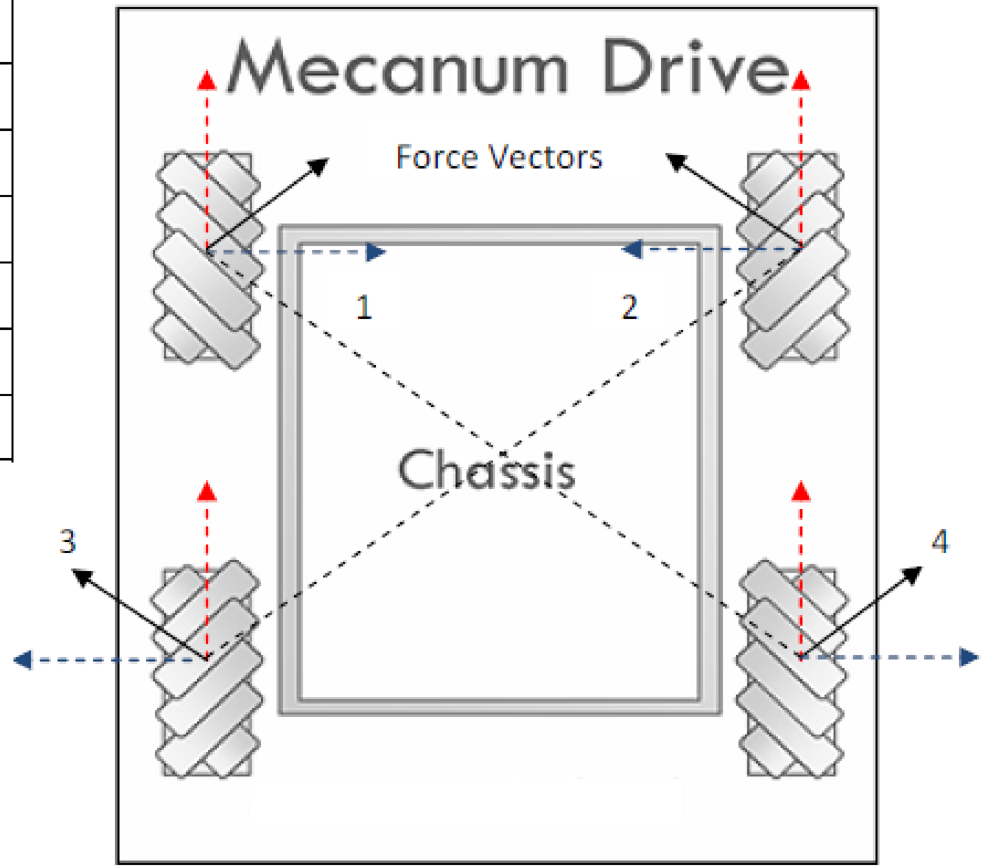
http://blog.makezine.com/archive/2010/04/3d-printable-mecanum-wheel.html

- https://www.youtube.com/watch?v=O7FbDy-gE70
- https://www.youtube.com/watch?v=mUoftURFsxM
- http://ftp.mi.fu-berlin.de/pub/Rojas/omniwheel/Diegel-Badve-Bright-Potgieter-Tlale.pdf

Mecanum Wheel

Direction of Movement	Wheel Actuation
Forward	All wheels forward same speed
Reverse	All wheels backward same speed
Right Shift	Wheels 1, 4 forward; 2, 3 backward
Left Shift	Wheels 2, 3 forward; 1, 4 backward
CW Turn	Wheels 1, 3 forward; 2, 4 backward
CCW Turn	Wheels 2, 4 forward; 1, 3 backward

To the right: This is a top view looking down on the drive platform. Wheels in Positions 1, 4 should make X- pattern with Wheels 2, 3. If not set up like shown, wheels will not operate correctly.



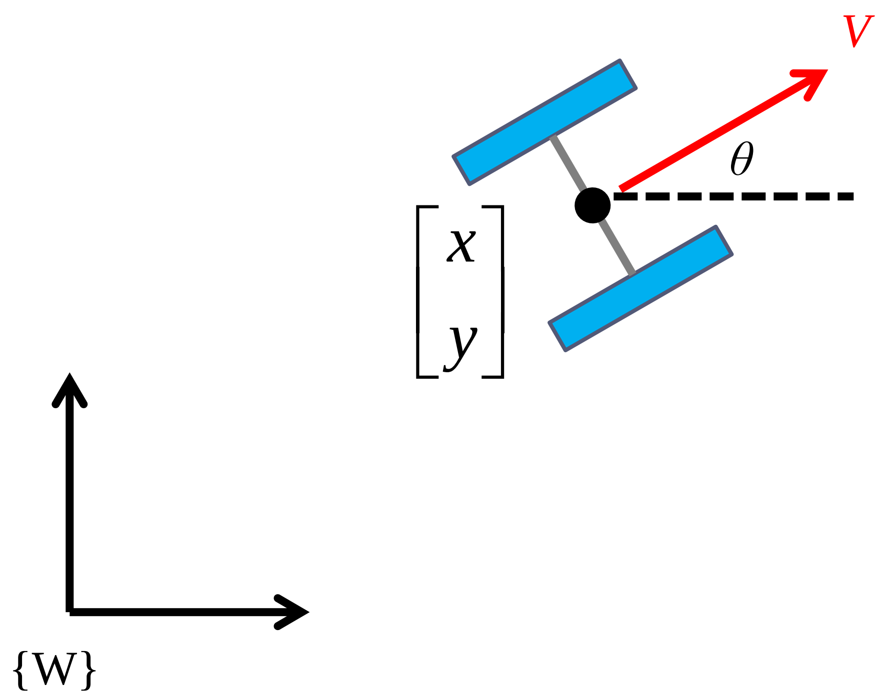
AndyMark Mecanum wheel specification sheet
http://d1pytrrjwm20z9.cloudfront.net/MecanumWheelSpecSheet.pdf

Forward Kinematics

- serial manipulators
  - given the joint variables, find the pose of the end-effector
- mobile robot
  - given the control variables as a *function of time*, find the pose of the robot
    - for the differential drive the control variables are often taken to be the ground velocities of the left and right wheels
      - it is important to note that the wheel velocities are needed as functions of time; a differential drive that moves forward and then turns right ends up in a very different position than one that turns right then moves forward!

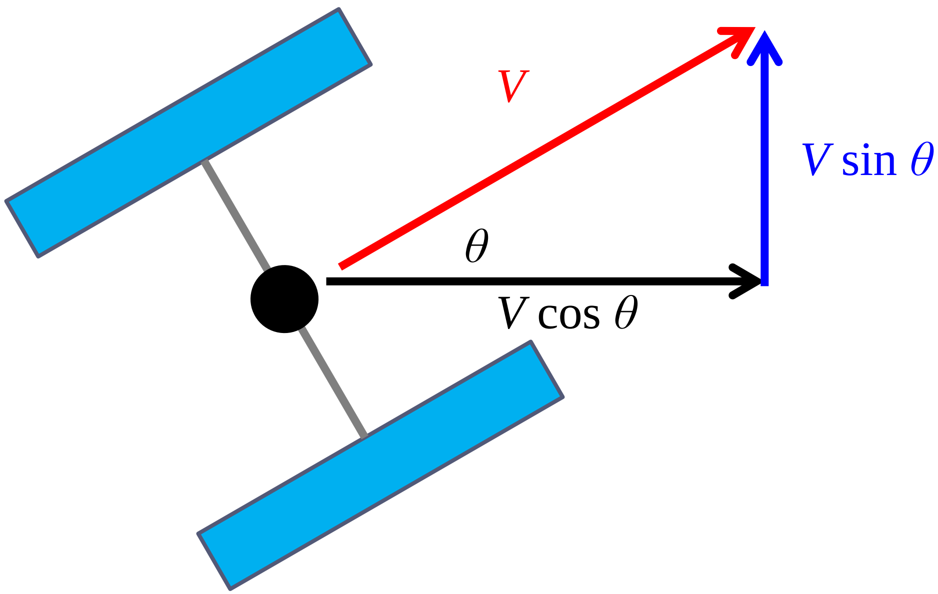
Forward Kinematics

- robot with pose $[x \;\; y \; \theta]^T$ moving with velocity V in a direction θ measured relative the x axis of $\{W\}$:



Forward Kinematics

- for a robot starting with pose $[x_0 \;\; y_0 \; \theta_0]^{\mathrm{T}}$ moving with velocity $V(t)$ in a direction $\theta(t)$:



$$x(t) = x_0 + \int_0^t V(t)\cos(\theta(t))\, dt$$

$$y(t) = y_0 + \int_0^t V(t)\sin(\theta(t))\, dt$$

$$\theta(t) = \theta_0 + \int_0^t \omega(t)\, dt$$

Forward Kinematics

- for differential drive:

$$x(t) = x_0 + \frac{1}{2}\int_0^t (v_r(t) + v_\ell(t))\cos(\theta(t))\,dt$$

$$y(t) = y_0 + \frac{1}{2}\int_0^t (v_r(t) + v_\ell(t))\sin(\theta(t))\,dt$$

$$\theta(t) = \theta_0 + \frac{1}{\ell}\int_0^t (v_r(t) - v_\ell(t))\,dt$$

Sensitivity to Wheel Velocity

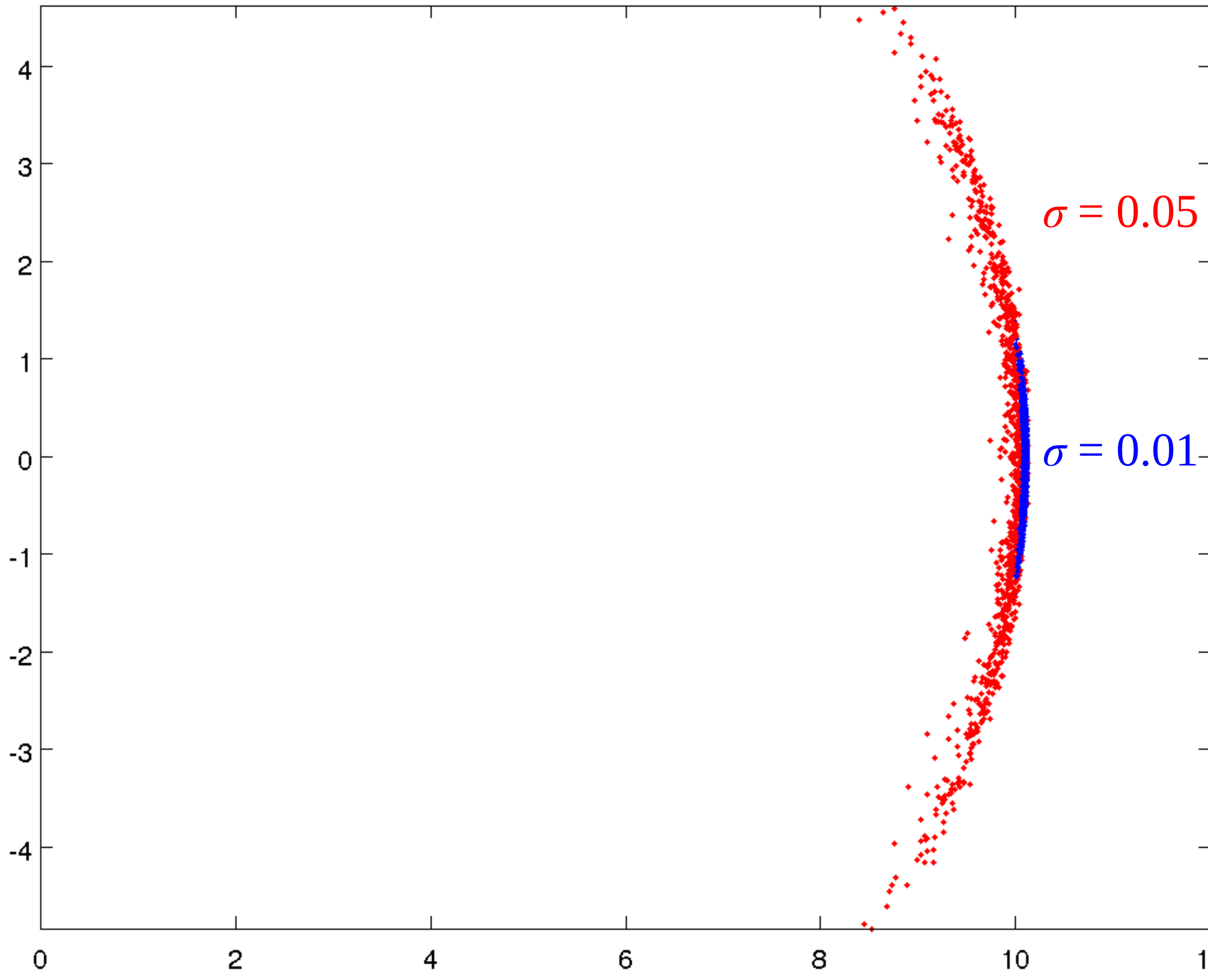
$$v_r(t) = 1 + \mathcal{N}(0, \sigma^2)$$

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

$$\theta(0) = 0$$

$$t = 0 \ldots 10$$

$$\ell = 0.2$$



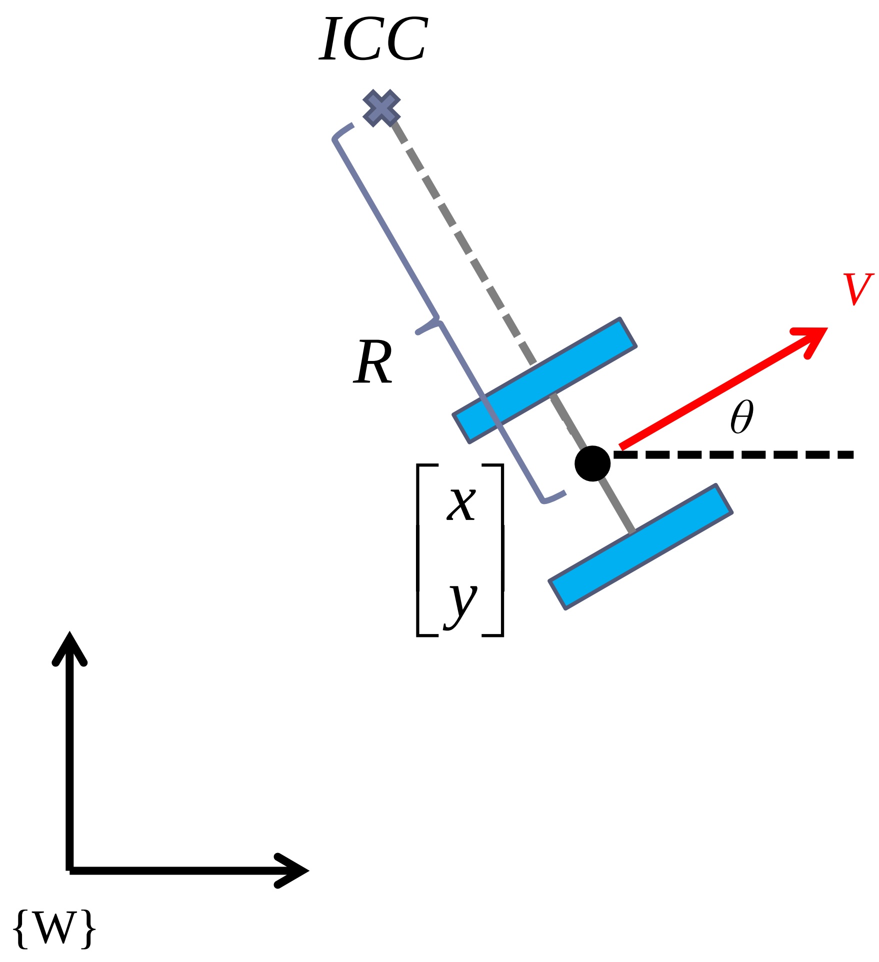
Sensitivity to Wheel Velocity

```
L = 0.2;
sigma = 0.05;
figure hold on
for i = 1:1000
    vR = 1 + normrnd(0, sigma);
    vL = 1 + normrnd(0, sigma);
    theta = 0;
    x = 0;
    y = 0;
    dt = 0.1;
    for t = 0.1:dt:10
        x = x + 0.5 * (vR + vL) * cos(theta) * dt;
        y = y + 0.5 * (vR + vL) * sin(theta) * dt;
        theta = theta + 1 / L * (vR - vL) * dt;
        vR = 1 + normrnd(0, sigma);
        vL = 1 + normrnd(0, sigma);
    end
    plot(x, y, 'b.');
end
```

Mobile Robot Forward Kinematics

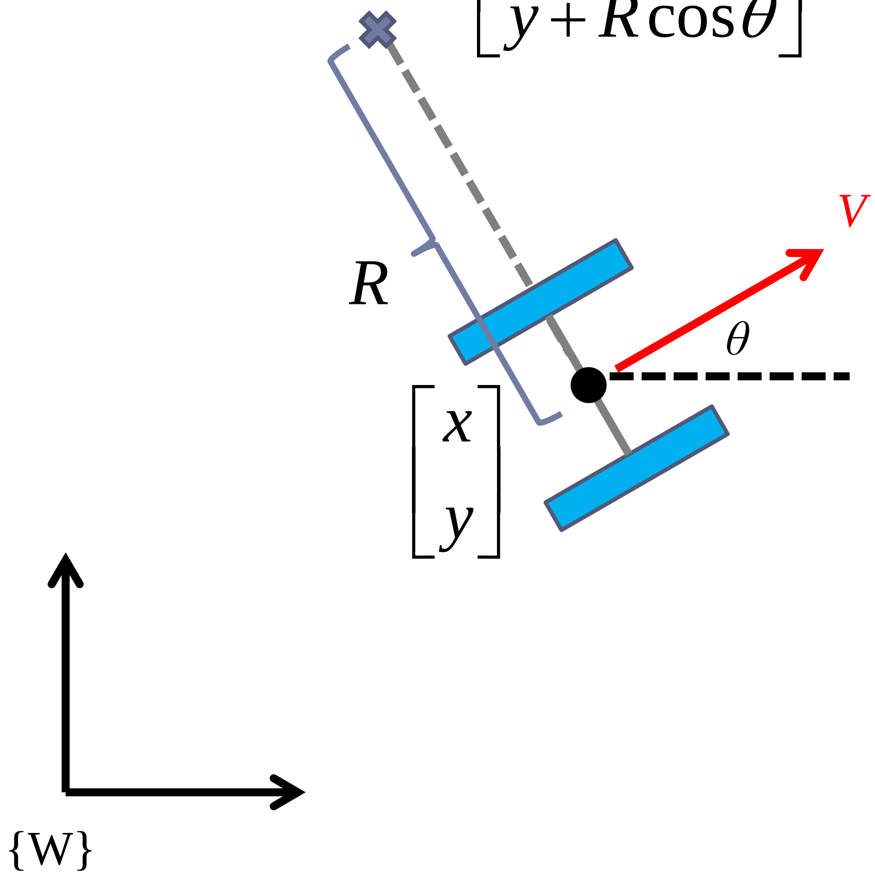
Forward Kinematics : Differential Drive

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



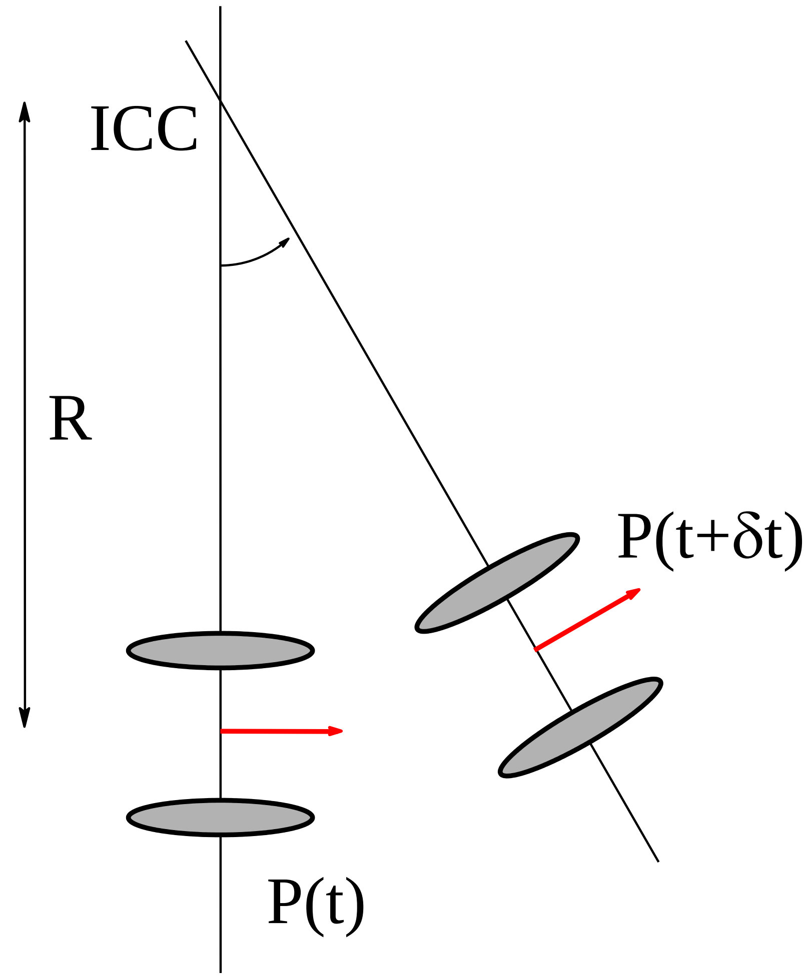
Forward Kinematics : Differential Drive

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



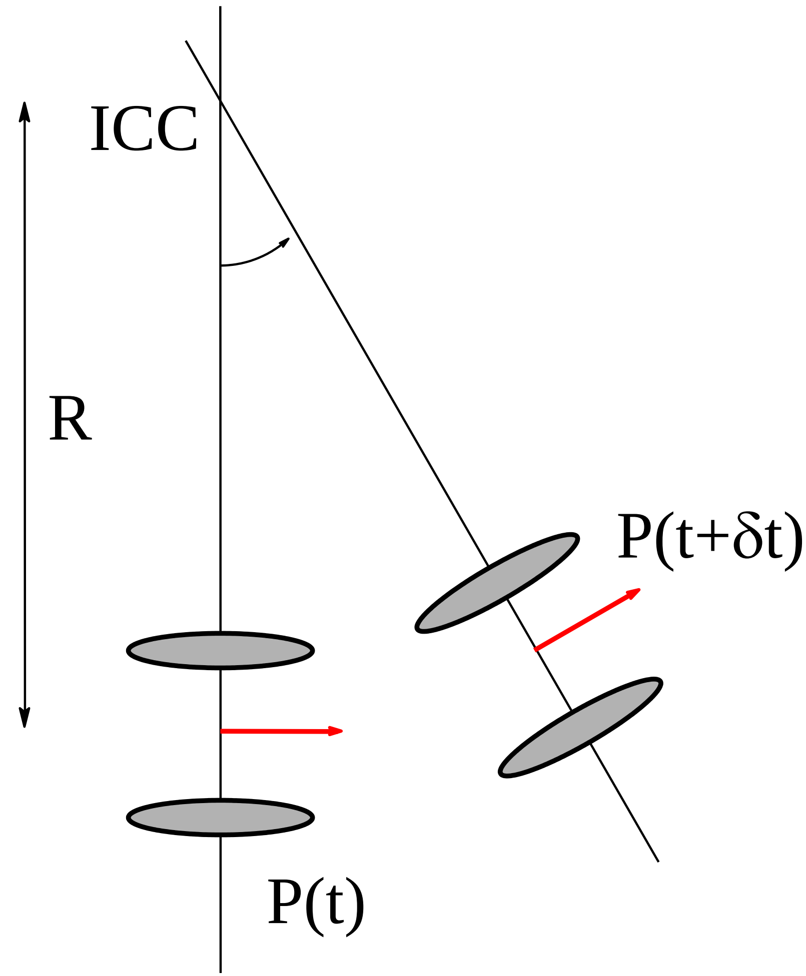
Forward Kinematics : Differential Drive

- **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

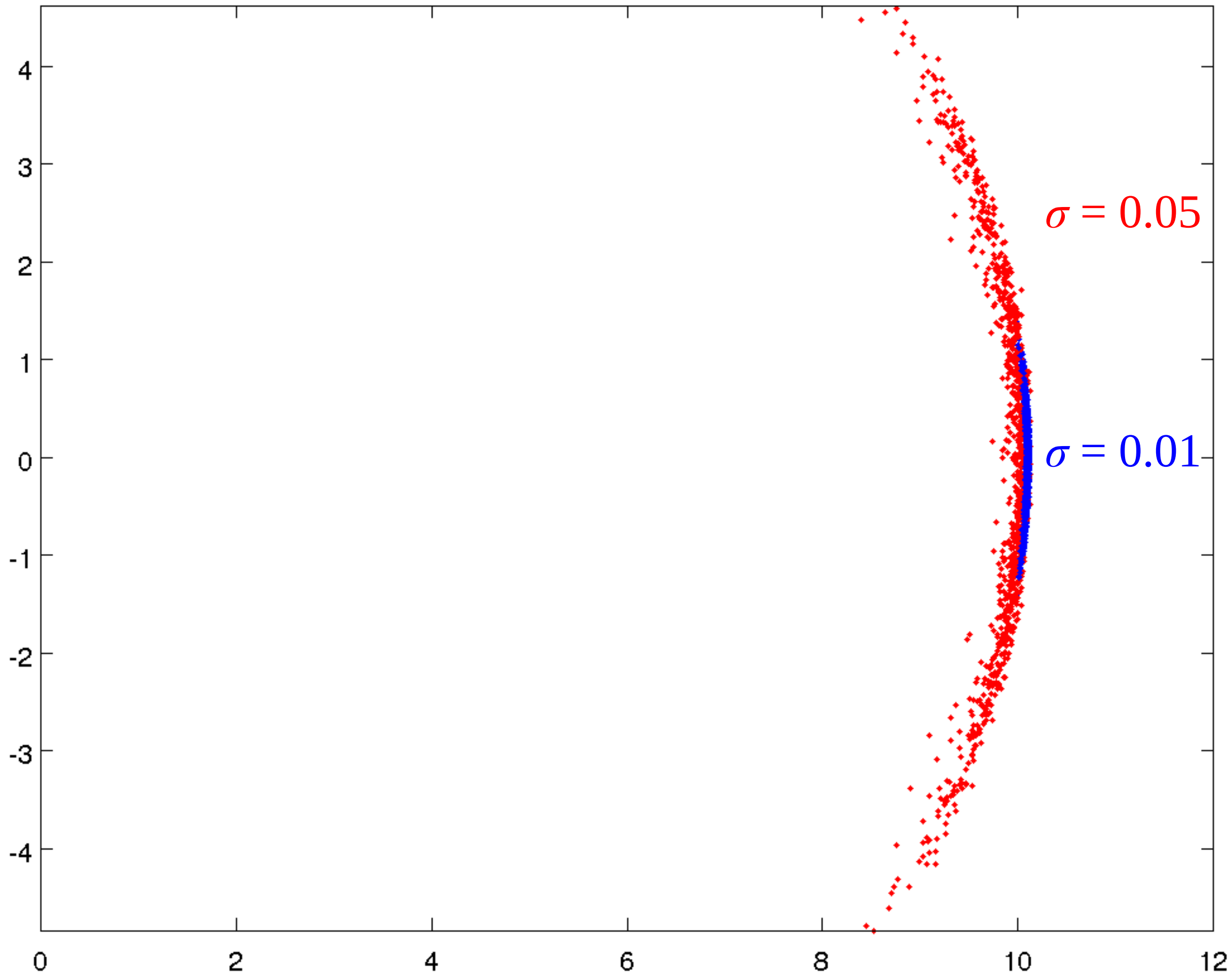
$$v_r(t) = 1 + \mathcal{N}(0, \sigma^2)$$

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

$$\theta(0) = 0$$

$$t = 0 \ldots 10$$

$$\ell = 0.2$$



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);
```