ENGR 3421: Robotics I

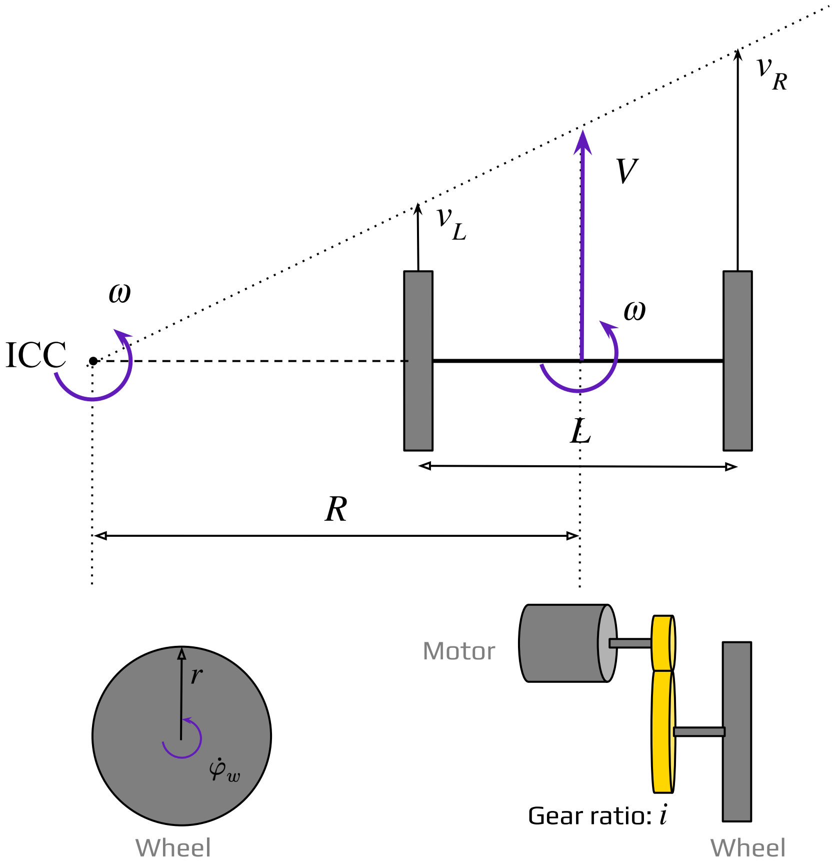
Kinematics of Differential Drive

10/22/2024

Outline

- Motion: From Motor to Robot
- Forward Kinematics (w.r.t. different frames)

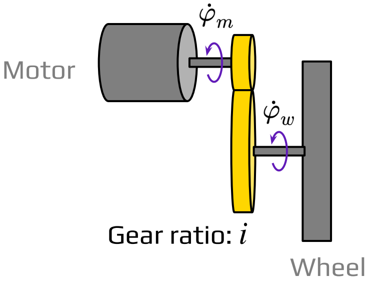
Motion: From Motor to Robot

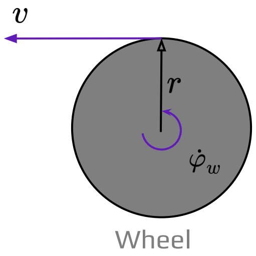


- ICC: Instantaneous Center of Curvature
- R: radius of curvature
- L: wheel separation distance
- V: robot linear velocity
- ω: robot angular velocity
- r: radius of wheel
- i: gear ratio
- $\dot{\varphi}_w$: angular velocity of wheel
- $\dot{\varphi}_m$: angular velocity of motor
- v_L: linear velocity of left wheel
- v_R: linear velocity of right wheel

Speed Computation: Motor to Wheel

1. Time "Counts Per Second"
2. Revolutions Per Second = Counts Per Second / Counts Per Revolution
3. Shaft Speed = Revolutions Per Second / Gear Ratio = Wheel Angular Speed
4. Wheel Linear Speed = Wheel Angular Speed * Wheel Radius





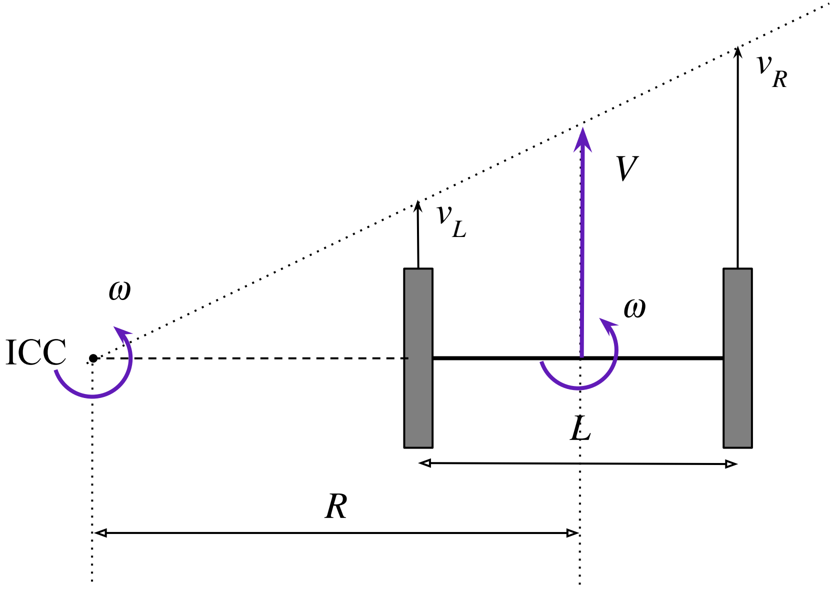
$$\dot{\varphi}_w = \frac{\dot{\varphi}_m}{i}$$

Motor velocity to wheel velocity

$$v = \dot{\varphi}_w r$$

Wheel angular velocity to linear velocity

Motion: From Wheel to Robot



$$\omega\left(R-\frac{L}{2}\right)=v_L$$
$$\omega\left(R+\frac{L}{2}\right)=v_R$$

Rotation about ICC must be same for both wheels.

$$v_L = V - \frac{\omega L}{2}$$

Linear velocity of left wheel

$$v_R = V + \frac{\omega L}{2}$$

Linear velocity of right wheel

$$V = \frac{v_L + v_R}{2}$$

Linear velocity of robot

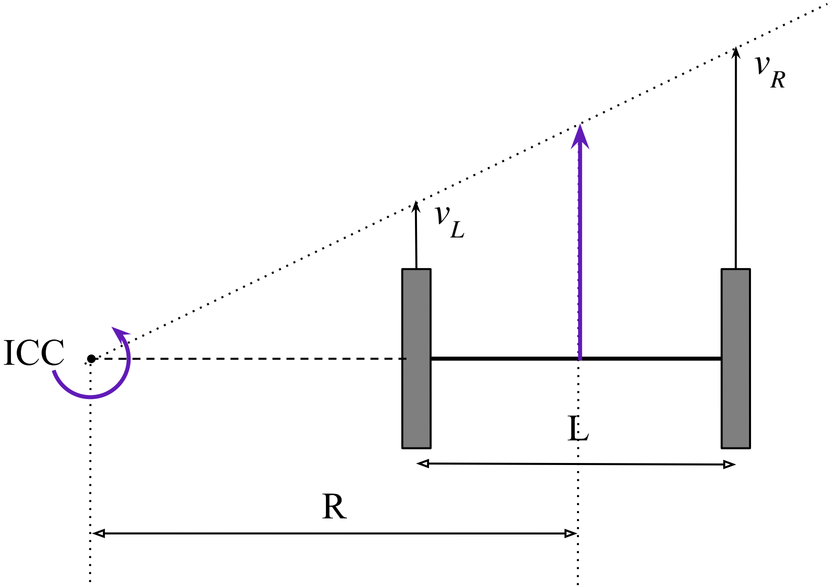
$$\omega = \frac{v_R - v_L}{L}$$

Angular velocity of robot

$$R = \frac{L}{2}\frac{v_L + v_R}{v_L - v_R}$$

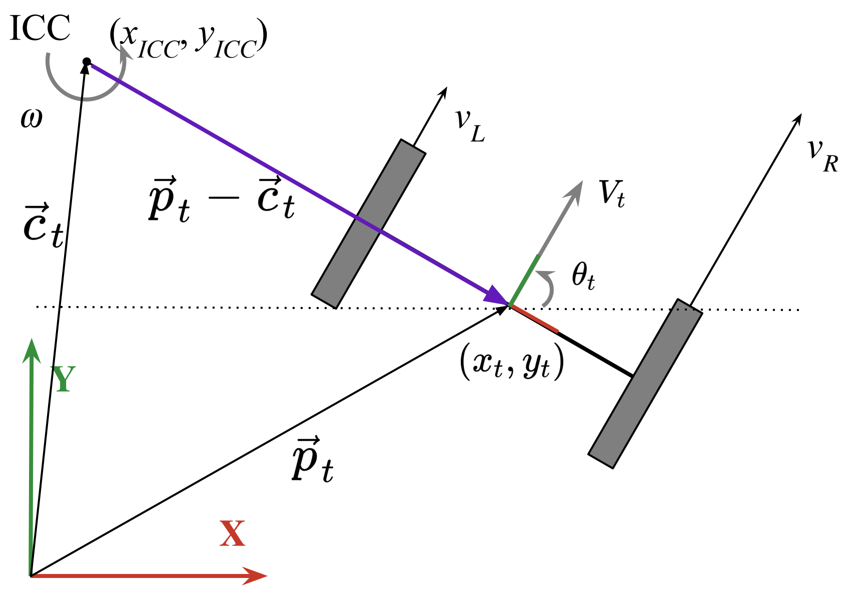
Rotation radius.

Motion: Special Cases



- If $v_L = v_R$, then linear motion in a straight line. R becomes infinite, no rotation ω=0.
- If $v_L = -v_R$, then rotation about the midpoint of the wheel axis, R = 0.
- If $v_L = 0$, then rotation about the left wheel, R = L/2. Rotation about the right wheel if $v_R = 0$.

Forward Kinematics (Continuous)



$$\vec{p} = [x,\ y]^T$$

$$\vec{c} = [x - R\sin(\theta),\ y + R\cos(\theta)]^T$$

$$\vec{p} - \vec{c} = [x - x_{ICC},\ y - y_{ICC}]^T$$

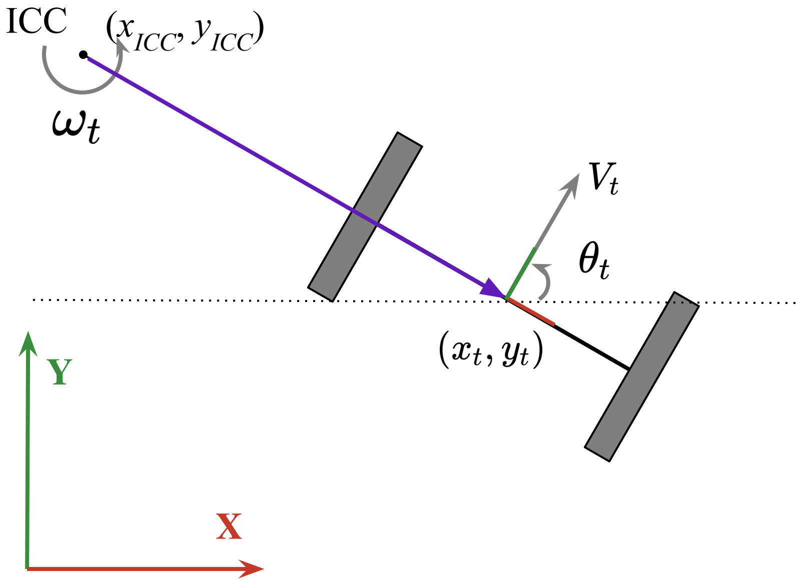
$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega\delta t) & -\sin(\omega\delta t) & 0 \\ \sin(\omega\delta t) & \cos(\omega\delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - x_{ICC} \\ y - y_{ICC} \\ \theta \end{bmatrix} + \begin{bmatrix} x_{ICC} \\ y_{ICC} \\ \omega\delta t \end{bmatrix}$$

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

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

$$\theta(t) = \int_0^t \omega(t)dt$$

Forward Kinematics (Discrete)

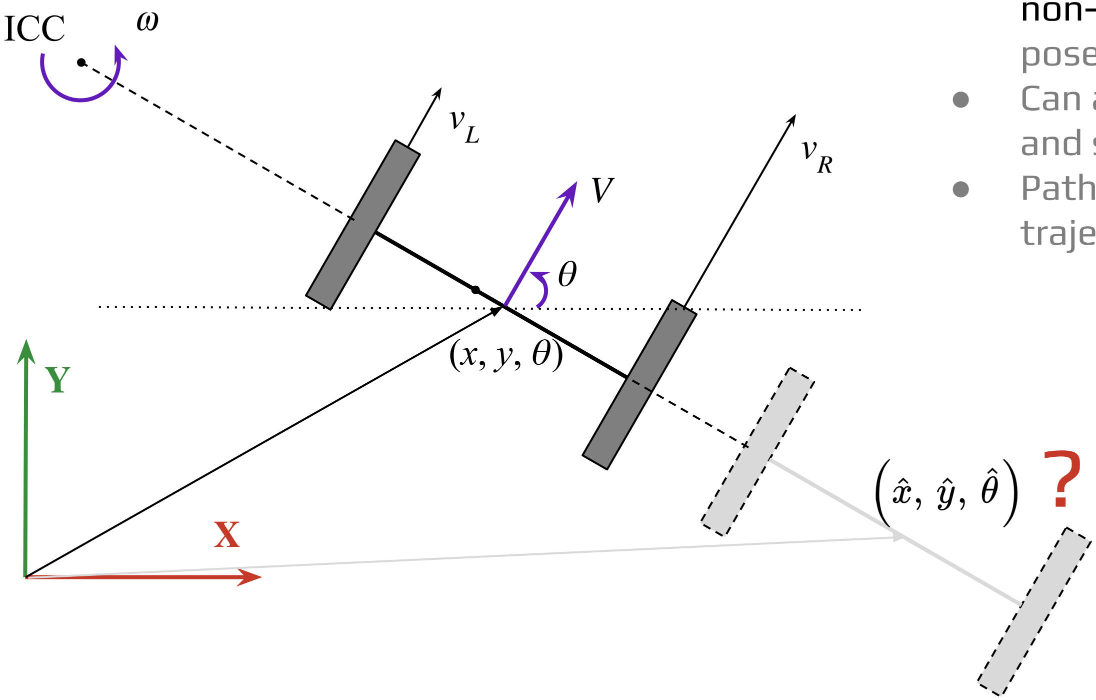


$$x_{t+1} = x_t + \Delta x$$
$$= x_t + V_t \cos\theta_t \cdot \Delta t$$

$$y_{t+1} = y_t + \Delta y$$
$$= y_t + V_t sin\theta_t \cdot \Delta t$$

$$\theta_{t+1} = \theta_t + \Delta\theta$$
$$= \theta_t + \omega_t \cdot \Delta t$$

Inverse Kinematics



- Given a target $(\hat{x},\, \hat{y},\, \hat{\theta})$, What is $V(t)$ and $\omega(t)$?
- Two-wheeled differential drive vehicle imposes non-holonomic constraints on establishing its pose (think about lateral translation).
- Can achieve the goal by moving in straight line and spinning in place.
- Path planning algorithms may find smoother trajectories.

```
from motor_driver import MotorDriver
class DualMotorDriver:
    def __init__(self, lmotor_ids, rmotor_ids):
        self.left_motor = MotorDriver(*lmotor_ids)
        self.right_motor = MotorDriver(*rmotor_ids)
    def forward(self, duty):
        self.left_motor.forward(duty)
        self.right_motor.forward(duty)
    def backward(self, duty):
        self.left_motor.backward(duty)
        self.right_motor.backward(duty)
    def spin_left(self, duty):
        self.left_motor.backward(duty)
        self.right_motor.forward(duty)
    def spin_right(self, duty):
        self.left_motor.forward(duty)
        self.right_motor.backward(duty)
    def stop(self):
        self.left_motor.stop()
        self.right_motor.stop()
```

```
# Test
if __name__ == '__main__':
    from time import sleep
    dmd = DualMotorDriver((11, 12,
13), (18, 19, 20))
    dmd.forward(40000)
    sleep(2)
    dmd.stop()
    sleep(0.25)
    dmd.backward(40000)
    sleep(2)
    dmd.stop()
    sleep(0.25)
    dmd.spin_left(40000)
    sleep(2)
    dmd.stop()
    sleep(0.25)
    dmd.spin_right(40000)
    sleep(2)
    dmd.stop()
    sleep(0.25)
```

```python
from dual_motor_driver import DualMotorDriver
from time import sleep

# SETUP
bot = DualMotorDriver((11, 12, 13), (18, 19, 20))

# LOOP
for i in [0, 10000, 25000, 50000]:
    bot.forward(i)
    sleep(1)
for i in reversed([0, 10000, 25000, 50000]):
    bot.forward(i)
    sleep(1)
bot.stop()
```