

#### 

# Robotics

Robot Localization – Wheels Odometry

Matteo Matteucci matteo.matteucci@polimi.it

Artificial Intelligence and Robotics Lab - Politecnico di Milano



# Where Am I?

To perform their tasks autonomous robots and unmanned vehicles need

- To know where they are (e.g., Global Positioning System)
- To know the environment map (e.g., Geographical Institutes Maps)

These are not always possible or reliable

- GNSS are not always reliable/available
- Not all places have been mapped
- Environment changes dynamically
- Maps need to be updated



How would you program this robot?



#### **A Simplified Sense-Plan-Act Architecture**





#### Localization with Known Map





#### Mapping with Known Poses





#### **Simultaneous Localization and Mapping**





### **Dynamic Bayesian Networks and (Full) SLAM**



Smoothing :  $p(\Gamma_{1:t}, l_1, ..., l_N | Z_{1:t}, U_{1:t})$ 





POLITECNICO MILANO 1863

#### Localization with Known Map





#### **Wheeled Mobile Robots**

A robot capable of locomotion on a surface **solely through the actuation of wheel assemblies** mounted on the robot and in contact with the surface. A wheel assembly is a device which provides or allows motion between its mount and surface on which it is intended to have **a single point of rolling contact.** 

(Muir and Newman, 1986)



**Robot Mobile** 



#### Unmanned vehicle



# **Wheels Types**





# Mobile Robots Types/Kinematics (some)

#### Two wheels (differential drive)

- Simple model
- Suffers terrain irregularities
- Cannot translate laterally

#### Tracks

- Suited for outdoor terrains
- Not accurate movements (with rotations)
- Complex model
- Cannot translate laterally

#### Omnidirectional (synchro drive)

- Can exploit all degrees of freedom (3DoF)
- Complex model
- Complex structure





# **Differential Drive (MRT – Politecnico di Milano)**





POLITECNICO MILANO 1863

# **Differential Drive (MRT – Politecnico di Milano)**





#### **Omnidirectional (Swedish wheels)**





POLITECNICO MILANO 1863

# **Omnidirectional (Syncro drive)**





POLITECNICO MILANO 1863

#### Some Definitions ...

Locomotion: the process of causing an autonomous robot to move

• To produce motion, forces must be applied to the vehicle

<u>Dynamics</u>: the study of motion in which forces are modeled

Includes the energies and speeds associated with these motions

Kinematics: study of motion without considering forces that affect if

- Deals with the geometric relationships that govern the system
- Deals with the relationship between control parameters and the behavior of a system in state space



## **Kinematics**



**Direct kinematics** 

• Given control parameters, e.g., wheels and velocities, and a time of movement t, find the pose (x, y,  $\theta$ ) reached by the robot

Inverse kinematics

• Given the final pose (x, y,  $\theta$ ) find control parameters to move there in a given time t



#### Wheeled robot assumptions

- 1. Robot made only by rigid parts
- 2. Each wheel may have a 1 link for steering
- 3. Steering axes are orthogonal to soil
- 4. Pure rolling of the wheel about its axis (x axis)
- 5. No translation of the wheel





Wheel parameters: r = radius v = linear velocity $\omega = angular velocity$ 



For a robot to move on the plane (3DoF), without slipage, wheels axis have to intersect in a single point named Instantaneous Center of Curvature (ICC) or Instantaneous Center of Rotation (ICR)





#### **Representing a Pose**





# **Differential Drive Kinematics (1)**

Construction

- 2 wheels on the same axis
- 2 independent motors (one for wheel)
- 3rd passive supporting wheel

Variables independently controlled

- V<sub>R</sub>: velocity of the right wheel
- V<sub>L</sub>: velocity of the left wheel

Pose representation in base reference:

Control input are:

- v: linear velocity of the robot
- $\omega$ : angular velocity of the robot  $\sim$

 $P = (x_b, y_b, \theta) = (x, y, \theta)$ 

Н

Linearly related to  $V_R$  and  $V_L$  ...  $\$ 



#### **Differential Drive Kinematics (2)**

Right and left wheels follow a circular path with  $\omega$  angular velocity and different curvature radius

$$ω$$
 (R + L/2) = V<sub>R</sub>  
 $ω$  (R - L/2) = V<sub>L</sub>

Given  $V_R$  and  $V_L$  you can find  $\omega$  solving for R and equating

$$\omega = (V_{R} - V_{L})/L$$

Similarly you can find R solving for  $\omega$  and equating





# **Differential Drive ICC**



POLITECNICO MILANO 1863

#### **Differential Drive Equations (Remember!)**





# **Differential drive odometry**

# Being known

$$\omega = (V_{R} - V_{L}) / L$$
  

$$R = L/2 (V_{R} + V_{L}) / (V_{R} - V_{L})$$
  

$$V = \omega R = (V_{R} + V_{L}) / 2$$

Compute the velocity in the base frame

$$V_{x} = V(t) \cos (\theta(t))$$
$$V_{y} = V(t) \sin (\theta(t))$$

Integrate position in base frame

$$x(t) = \int V(t) \cos(\theta(t)) dt$$

$$y(t) = \int V(t) \sin(\theta(t)) dt$$

$$\theta(t) = \int \omega(t) dt$$

$$Can integrate at discrete time$$

$$y(t) = \frac{1}{2} \int_{0}^{t} (V_{R}(t') + V_{L}(t')) \cdot \sin(\theta(t')) \cdot dt'$$

$$\theta(t) = \frac{1}{L} \int_{0}^{t} (V_{R}(t') - V_{L}(t')) \cdot dt'$$
mattee matteucci@polimi it

ICR

**≰ Y**₁

У



V<sub>R</sub>

Χ<sub>1</sub>

Х

 $x(t) = \frac{1}{2} \int_{0}^{1} (V_{R}(t') + V_{L}(t')) \cdot \cos(\theta(t')) \cdot dt'$ 

# **Odometry Integration (1)**

Assume constant linear velocity  $v_k$  and angular velocity  $\omega_k$  in  $[t_k, t_{k+1}]$ can use Euler integration to compute robot odometry

$$x(t) = \frac{1}{2} \int_{0}^{t} (V_{R}(t') + V_{L}(t')) \cdot \cos(\theta(t')) \cdot dt'$$

$$x_{k+1} = x_{k} + v_{k} T_{S} \cos \theta_{k}$$

$$y_{k+1} = y_{k} + v_{k} T_{S} \sin \theta_{k}$$

$$\theta_{k+1} = \theta_{k} + \omega_{k} T_{S}$$

$$\theta(t) = \frac{1}{L} \int_{0}^{t} (V_{R}(t') - V_{L}(t')) \cdot dt'$$

$$T_{S} = t_{k+1} - t_{k}$$
Exact



# **Odometry Integration (2)**

Assume constant linear velocity  $v_k$  and angular velocity  $\omega_k$  in  $[t_k, t_{k+1}]$  can use 2nd order Runge-Kutta integration to compute robot odometry



Better

approximation

# **Odometry Integration (3)**

Assume constant linear velocity  $v_k$  and angular velocity  $\omega_k$  in  $[t_k, t_{k+1}]$  can use exact integration to compute the robot odometry

$$x(t) = \frac{1}{2} \int_{0}^{t} (V_{R}(t') + V_{L}(t')) \cdot \cos(\theta(t')) \cdot dt'$$

$$x_{k+1} = x_{k} + \frac{v_{k}}{\omega_{k}} (\sin \theta_{k+1} - \sin \theta_{k})$$

$$y(t) = \frac{1}{2} \int_{0}^{t} (V_{R}(t') + V_{L}(t')) \cdot \sin(\theta(t')) \cdot dt'$$

$$y_{k+1} = y_{k} - \frac{v_{k}}{\omega_{k}} (\cos \theta_{k+1} - \cos \theta_{k})$$

$$\theta_{k+1} = \theta_{k} + \omega_{k} T_{S}$$
Need to use Rugge
$$T_{S} = t_{k+1} - t_{k}$$
Need to use Rugge



Exact

# **Odometry Comparison**





# **Tips and Tricks**

Proprioceptive measurements are used to compute linear  $v_k$  and angular velocity  $\omega_k$ 

$$v_k T_S = \Delta s, \qquad \omega_k T_S = \Delta \theta, \qquad \frac{v_k}{\omega_k} = \frac{\Delta s}{\Delta \theta}$$

with  $\Delta s$  is the traveled distance and  $\Delta \theta$  is the orientation change



# Synchronous drive

Complex mechanical robot design

- (At least) 3 wheels actuated and steered
- A motor to roll all the wheels, a second motor to rotate them
- Wheels point in the same direction
- It is possible to control directly θ with an additional actuator

Robot control variables

- Linear velocity v(t)
- Angular velocity  $\omega(t)$



Its ICC is always at the infinite and the robot is **non holonomic** (it can only translate freely)



### Synchronous drive kinematics

Robot control for the synchronous drive

- Direct control of v(t) and  $\omega$ (t)
- Steering changes the direction of ICC

Particular cases:

- $v(t)=0, \omega(t) = \omega$  for dt  $\rightarrow$  robot rotates in place
- $v(t)=v, \omega(t) = 0$  for dt  $\rightarrow$  robot moves linearly



Compute the velocity in the base frame

 $x(t) = \int_{\Omega} v(t') \cos[\theta(t')]dt'$  $V_x = V(t) \cos(\theta(t))$ Calls odometry  $y(t) = \int v(t') \sin[\theta(t')]dt'$ also for diff drive!  $V_v = V(t) \sin(\theta(t))$ Integrate position in base frame to get  $\theta(t) = \int \omega(t') \, dt'$ the robot odometry (traversed path) ...



# **Omnidirectional (Syncro drive)**





POLITECNICO MILANO 1863

# **Omnidirectional Robot**

Simple mechanical robot design

- (At least) 3 Swedish wheels actuated
- One independent motor per wheel
- Wheels point in different direction
- It is possible to control directly x, y,  $\theta$

Robot control variables

- Linear velocity v(t) (each component)
- Angular velocity  $\omega(t)$









# **Omnidirectional (Swedish wheels)**





POLITECNICO MILANO 1863

# **Tricycle kinematics**

The Tricycle is the typical kinematics of AGV

- One actuated and steerable wheel
- 2 additional passive wheels
- Cannot control θ independently unless α(t) can be up to 90 degrees
- ICC must lie on the line that passes through the fixed wheels

Robot control variables

- Steering direction α(t)
- Angular velocity of steering wheel  $\omega_s(t)$

Particular cases:

- $\alpha(t)=0$ ,  $\omega_s(t) = \omega \rightarrow \text{moves straight}$
- $\alpha(t)=90, \omega_s(t) = \omega \rightarrow rotates in place$







# **Tricycle kinematics**



# **Tricycle kinematics**



In the base frame

$$\dot{x}(t) = V_{s}(t) \cdot \cos \alpha(t) \cdot \cos \theta(t) = V(t) \cdot \cos \theta(t)$$
$$\dot{y}(t) = V_{s}(t) \cdot \cos \alpha(t) \cdot \sin \theta(t) = V(t) \cdot \sin \theta(t)$$
$$\dot{\theta} = \frac{V_{s}(t)}{d} \cdot \sin \alpha(t) = \omega(t)$$





## **Ackerman steering**



# Ackerman steering (bicycle approximation)

Most diffused kinematics on the planet

- Four wheels turning
- Wheels have limited turning angles
- No in-place rotation
- **Bicycle** approximation

$$R = \frac{d}{\tan \alpha}$$
$$\frac{\omega d}{\sin \alpha} = V_F$$

Referred to the center of real wheels

$$\omega R = V \quad \Longrightarrow \quad \omega = V \cdot \frac{\tan \alpha}{d}$$





# **Vehicles with tracks**

Vehicles with track have a kinematics similar to the differential drive

- Speed control of each track
- Use the height of the track as wheel diameter



Often named Skid Steering

Need proper calibration and slippage modeling ...



# **Skid Steering (approximate) Kinematics**

Let' assume:

 $v_{x}$ 

- Mass in the center of the vehicle
- All wheels in contact with the ground
- Wheels on the same side have same speed

 $\omega_l = \omega_1 = \omega_2 \qquad \qquad \omega_r = \omega_3 = \omega_4$ 

While moving we have multiple ICR and all share  $\omega_z$ 

$$y_{G} = \frac{x}{\omega_{Z}}$$

$$y_{l} = \frac{v_{x} - \omega_{l}r}{\omega_{Z}}$$

$$y_{r} = \frac{v_{x} - \omega_{r}r}{\omega_{Z}}$$

$$y_{r} = x_{r} = -\frac{v_{y}}{\omega_{Z}}$$

$$\int_{\omega} = \frac{1}{y_{l} - y_{r}} \begin{bmatrix} -y_{r} & y_{l} \\ x_{G} & -x_{G} \\ -1 & 1 \end{bmatrix}$$

Ar.

# **Skid Steering (approximate) Kinematics**

Let' assume:

- Mass in the center of the vehicle
- All wheels in contact with the ground
- Wheels on the same side have same speed

 $\omega_l = \omega_1 = \omega_2 \qquad \qquad \omega_r = \omega_3 = \omega_4$ 

Assume the robot is symmetric







# **Skid Steering (approximate) Kinematics**

Let' assume:

- Mass in the center of the vehicle
- All wheels in contact with the ground
- Wheels on the same side have same speed

 $\omega_l = \omega_1 = \omega_2 \qquad \qquad \omega_r = \omega_3 = \omega_4$ 

We can get the istantaneous radius of curvature







### **Kinematics**



**Direct kinematics** 

• Given control parameters, e.g., wheels and velocities, and a time of movement t, find the pose (x, y,  $\theta$ ) reached by the robot

Inverse kinematics

• Given the final pose (x, y,  $\theta$ ) find control parameters to move there in a given time t



#### **Inverse kinematics**

Given a desired position or velocity, what can we do to achieve it?

Finding "some" solution is not hard, but finding the "best" solution can be very difficult:

- Shortest time
- Most energy efficient
- Smoothest velocity profiles

Moreover if we have non holonomic constrains and only two control variables; we cannot directly reach any of the 3DoF final positions ...





#### **Differential drive inverse kinematics**

Decompose the problem and control only few DoF at the time

1. Turn so that the wheels are parallel to the line between the original and final position of robot origin

 $-V_{L}(t) = V_{R}(t) = V_{max}$ 

2. Drive straight until the robot's origin coincides with destination

 $V_{L}(t) = V_{R}(t) = V_{max}$ 

3. Rotate again in to achieve the desired final orientation

$$-V_{L}(t) = V_{R}(t) = V_{max}$$





# Synchro drive inverse kinematics

Decompose the problem and control only a few degrees of freedom at a time

1. Turn so that the wheels are parallel to the line between the original and final position of robot origin

 $\omega(t) = \omega_{\max}$ 

2. Drive straight until the robot's origin coincides with destination

 $v(t) = v_{max}$ 

3. Rotate again in to achieve the desired final orientation

$$\omega(t) = \omega_{\max}$$



