DDD426 – Robotics and Autonomous Systems
Lecture 3: Kinematics and control

Patric Jensfelt

Kungliga Tekniska Högskolan
patric@kth.se

April 3,2008

Course admin

- Can everyone come in to the lab?
- Please direct questions about the lab to Mattias
- Next lab session next Thursday 15-17
- Look at the reports from previous years online

Motion control

- Motion control requires:
  - Kinematic/dynamic model of the robot
  - Model of the ground/wheel interaction (very complicated)
  - Definition of required motion
  - Design of control law
  - Verification and tuning

Mobile robot kinematic

- Model of mechanical behavior of robot for design and control
- Models for manipulators similar to mobile robots
- Some important differences
  - Cannot measure robot position directly (encoders give manipulator end effector position)
  - Position must be integrated over time
- "Next generation" manipulators also quite challenging
  - Light weight and flexible materials, cannot get position as easily

Kinematics model

- We want to:
  - Get the robot speed

$$\dot{\xi} = [\dot{x} \quad \dot{y} \quad \dot{\theta}]^T$$

as a function of wheel speeds $\dot{\varphi}_i$, steering angles β_i, steering speeds $\dot{\beta}_i$ and the geometric parameters of the robot

  - Forward kinematics:

$$\dot{\xi} = [\dot{x} \quad \dot{y} \quad \dot{\theta}]^T = f(\dot{\varphi}_1, \ldots, \dot{\varphi}_n, \beta_1, \ldots, \beta_m, \dot{\beta}_1, \ldots, \dot{\beta}_m)$$

  - Inverse kinematics:

$$[\dot{\varphi}_1, \ldots, \dot{\varphi}_n, \beta_1, \ldots, \beta_m, \dot{\beta}_1, \ldots, \dot{\beta}_m]^T = f(\dot{x} \quad \dot{y} \quad \dot{\theta})$$

Why use velocities?

- Why not give $[x \quad y \quad \theta]^T = f(\varphi_1, \ldots, \varphi_n, \beta_1, \ldots, \beta_m)$?
- Consider simple differential drive robot and the difference between first turning the left wheel an angle φ and then the right wheel φ compared to turning both wheels at the same time this much

Coordinate systems

- An good choice of coordinate system can save a lot of time
- Common choices are inertial frame (I) and the robot frame (R).
- A point can be described by

$$\bar{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

- For structures you need the orientation as well (α, ϕ, θ)
- Need to be able to transform between coordinate systems

Transformations

- Translation is simple, just add the vectors

$$\bar{x}_2 = \bar{x}_0 + \bar{x}_1$$

- Rotation can be modeled with a rotation matrix

$$\bar{x}_1 = \mathbf{R}\bar{x}_0$$

- The rotation matrix for a rotation α around the X axis is given by

$$\mathbf{R}_X(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & cos\alpha \end{bmatrix}$$

- Applying several rotations results in matrix multiplications

$$\mathbf{R}_{XY} = \mathbf{R}_Y \mathbf{R}_X$$

Transformation matrix

- Can combine translation and rotation into one matrix

$$\mathbf{T} = \begin{bmatrix} \mathbf{R}_{3x3} & | & \bar{x}_{3x1} \\ -- & & -- \\ \mathbf{0}_{1x3} & | & 1 \end{bmatrix} = \begin{bmatrix} rotation & | & translation \\ -- & & -- \\ 0 & | & 1 \end{bmatrix}$$

- Can be made even more general by incorporating scale and perspective transformation.
- Extend the position vector with a "1"

$$\bar{x} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Transformation matrix cont'd

- Pure rotation around X

$$\mathbf{T}_X(\alpha) = \mathbf{R}(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Pure translation

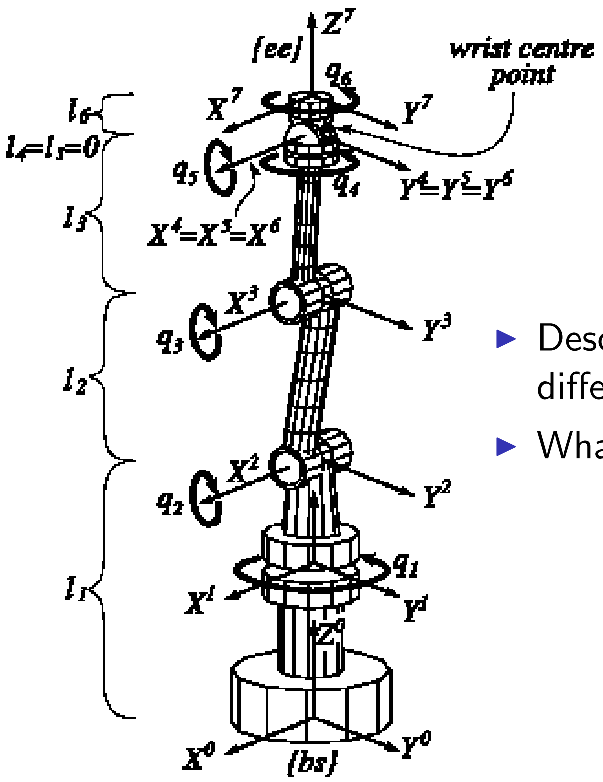
$$\mathbf{T}_{trans} = \mathbf{Tr} = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Multiple matrices to build compound transformations
- Heavily used in computer graphics

Reference frames

- Many reference frames for articulated systems
- Estimation and control depends critically on choice of reference frame.
- What is simple in one can be very difficult in another.

Manipulator kinematics



- Description of motion across different coordinate systems
- What is the overall motion?

Forward and inverse manipulator kinematics

- Forward kinematic typically straight forward to get for stiff manipulator
- Inverse kinematics can sometimes only be solved numerically
- Important to think about multiple solutions
- How to get smooth motion from one point to another?
- Have to take care of "shoulder flips" for example
- A small motion in 3D might sometimes require huge motions for the joints

Example: moving human arm

- Move you fist to the shoulder with your elbow pointing back.
- Try to move the fist back 2dm more.
- Can you do this without a large change of the other joints?
- Need to deal with this for robot manipulators as well. Must plan ahead

Example: opening door

- Base + arm has 9 DOFs. Can use the redundant DOFs to stay away from singularities.

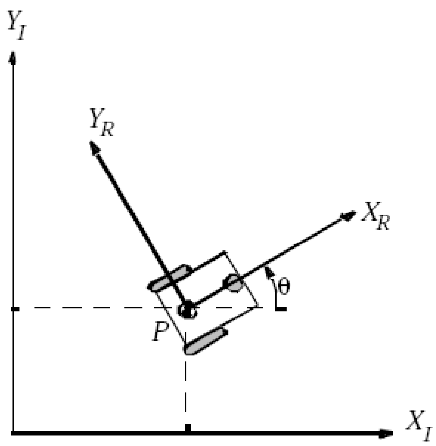
- Videos of door opening (grabbing and opening)

Robot pose in 2D

- We assume here that we deal with a rigid body robot moving on a horizontal plane.
- Gives 3 degrees of freedom (DOF)

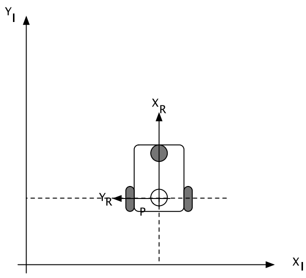
$$(\underbrace{x \quad y}_{positions} \quad \underbrace{\theta}_{orientation})$$

- Robot pose (position + orientation)
- $\bar{\xi}_I = (x, y, \theta)$



Motion between frames

- Can relate the speed in inertial frame to that in the robot frame with $\dot{\xi}_R = \mathbf{R}(\theta)\dot{\xi}_I$
- or $\dot{\xi}_I = \mathbf{R}(\theta)^{-1}\dot{\xi}_R$
- Example:



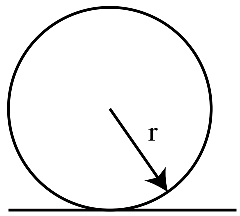
$$\mathbf{R}_Z(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}(\frac{\pi}{2}) = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

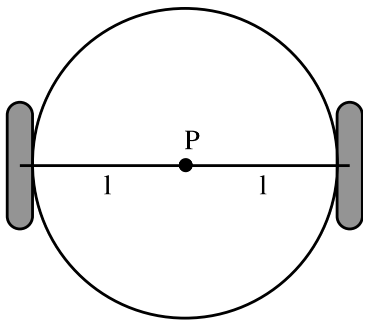
$$\dot{\xi}_I = \mathbf{R}(\frac{\pi}{2})\dot{\xi}_I = \begin{bmatrix} \dot{y} \\ -\dot{x} \\ \dot{\theta} \end{bmatrix}_R$$

Forward kinematics - differential drive

- Differential drive robot
- Wheel radius r (assumed to be the same for both wheels)
- Distance between the wheels $B = 2l$
- Wheels rotate with $\dot{\varphi}_R$ and $\dot{\varphi}_L$
- Seeking motion in inertial frame



$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(l, r, \theta, \dot{\varphi}_R, \dot{\varphi}_L)$$



Forward kinematics - differential drive

- $\dot{\xi}_I = \mathbf{R}(\theta)\dot{\xi}_R$
- Translation speed of each wheel given by $r\dot{\varphi}$
- Translation speed of the robot

$$\dot{x}_R = r\frac{\dot{\varphi}_R + \dot{\varphi}_L}{2}$$

- Rotation speed at P for right wheel $\omega = \frac{r\dot{\varphi}_R}{2l}$
- and for left wheel $\omega = \frac{r\dot{\varphi}_L}{2l}$
- Rotation speed is then

$$\dot{\theta} = \frac{r}{2l}(\dot{\varphi}_R - \dot{\varphi}_L)$$

- Full model

$$\dot{\xi}_I = \mathbf{R}^{-1}\frac{r}{2}\begin{bmatrix}\dot{\varphi}_R + \dot{\varphi}_L & 0 & \frac{\dot{\varphi}_R - \dot{\varphi}_L}{l}\end{bmatrix}$$

Odometry

- By attaching encoders on the wheels their rotation can be measured
- The so called odometry is given by integrating encoder measurements over time
- Can integrate the kinematic model
- The better the kinematic model, the better the odometry

Encoder resolution

- Say you have a wheel with radius 100mm
- What encoder resolution would you need if you place the encoder on the wheel axis and want to be able to detect a motion as small as 1mm?
- What if you put the encoder on the motor axis on the other end of a gear box?

Control with (v, ω) vs $(\dot{\varphi}_L, \dot{\varphi}_R)$

- Often have abstraction layer that allows you to control the motion of the robot with translation and rotation speed instead of individual wheel speeds
- Translation speed v (same as $\dot{X}_R$)
- Rotation speed $\omega = \dot{\theta}$

$$v = \frac{r}{2}(\dot{\varphi}_R + \dot{\varphi}_L)$$

$$\omega = \frac{r}{2l}(\dot{\varphi}_R - \dot{\varphi}_L)$$

- Express $\dot{\varphi}_L$ and $\dot{\varphi}_R$ in v, ω

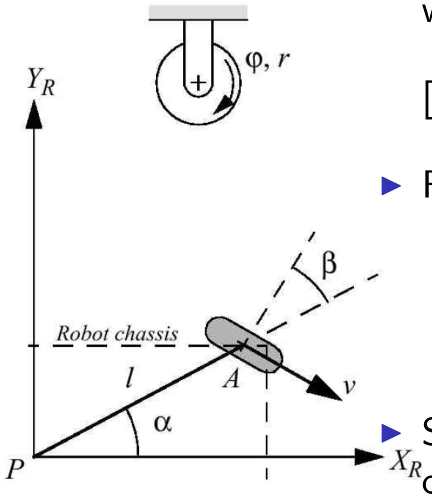
Driving on arcs

- Control with (v, ω) results in motion on arcs
- What is the radius of the motion as a function of (v, ω)?

Motion constraints

- Mobile robots typically cannot move freely, there are constraints
- Deriving a motion model for the robot is a bottom-up process
- Start with constrains on the wheels
- Assumptions:
  - Plane of wheel is always vertical
  - Single point contact with surface
  - Motion is purely by rolling (no slippage)
  - Rotation of wheel is around the vertical axis

Fixed wheel constraint



- Speed of wheel $v = r\dot{\varphi}$
- Rolling constraint (speed along the wheel direction should be $r\dot{\varphi}$):

$$\begin{bmatrix} \sin(\alpha+\beta) & -\cos(\alpha+\beta) & (-l)\cos\beta \end{bmatrix} \mathbf{R}(\theta)\dot{\xi}_I = r\dot{\varphi}$$

- Remember:

$$\cos(\alpha+\beta-\pi/2) = sin(\alpha+\beta)$$

$$\sin(\alpha+\beta-\pi/2) = -cos(\alpha+\beta)$$

- Sliding constraint (no motion orthogonal to wheel direction):

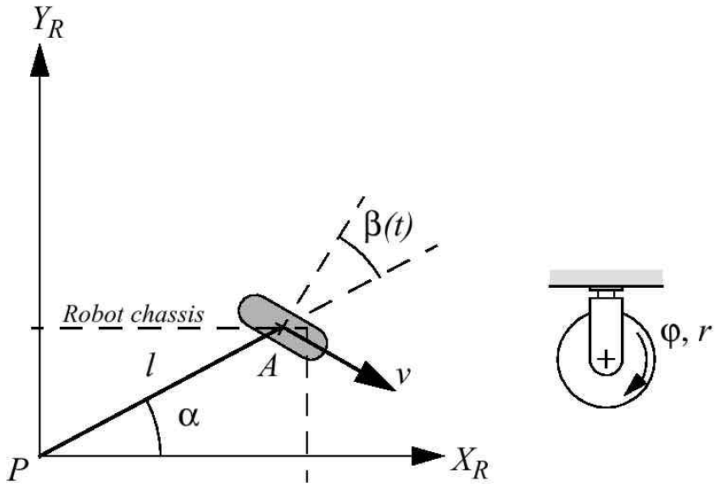
$$\begin{bmatrix} \cos(\alpha+\beta) & \sin(\alpha+\beta) & l\sin\beta \end{bmatrix} \mathbf{R}(\theta)\dot{\xi}_I = 0$$

Example

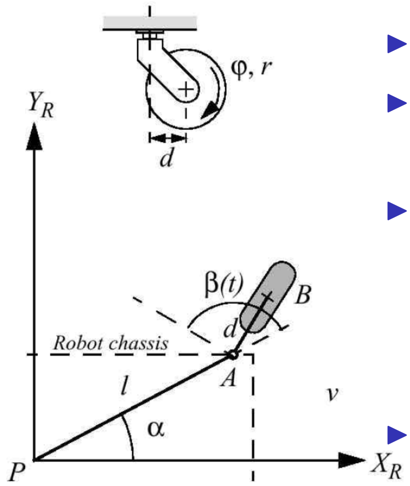
- Say that $\alpha = \beta = 0$
- What sliding constraint would you expect on $\dot{x}, \dot{y}, \dot{\theta}$)?
- Calculate the sliding constraint. Is it the same?

Steered wheel constraint

- Same as for fixed standard wheel
- Only difference is that β is a function of time



Castor wheel constraint



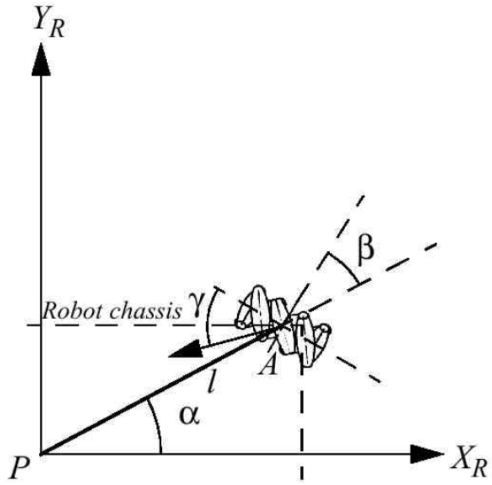
- Speed of wheel $v = r\dot{\varphi}$ (as before)
- Rolling constraint same as for fixed wheel
- Constraint around wheel attachment point:

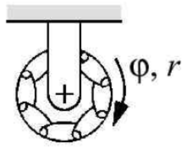
$$\begin{bmatrix}\cos(\alpha+\beta) & \sin(\alpha+\beta) & (d+l)\sin\beta\end{bmatrix}\mathbf{R}(\theta)\dot{\xi}_I + d\dot{\beta} = 0$$

- Can set $\dot{\beta}$ to satisfy arbitrary motion

Swedish wheel constraint

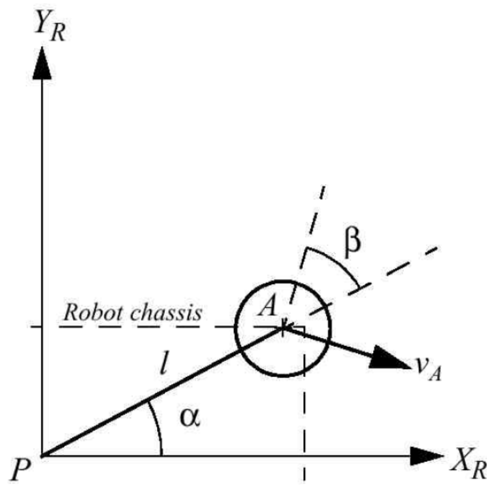
- No kinematic constraints!

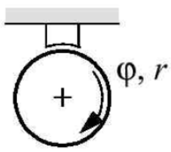




Spherical wheel constraint

- No kinematic constraints!





Robot kinematic constraints

- Combining the wheel constraints gives the overall constraints for the vehicle
- Only fixed (f) and steerable (s) standard wheels impose constraints
- Assume $N = N_f + N_s$ wheels
- β_f orientation of the fixed wheels
- $\beta_s(t)$ is the steering angle of the steerable wheels
- Total motion of wheels

$$\varphi(t) = \begin{bmatrix} \varphi_f(t) \\ \varphi_s(t) \end{bmatrix}$$

Collecting constraints

- Collecting the rolling constraints:

$$\underbrace{\begin{bmatrix} J_{1f} \\ J_{1s}(\beta_s) \end{bmatrix}}_{J_1(\beta_s)} R(\theta)\dot{\xi}_I = J_2\dot{\varphi}$$

where J_2 is a diagonal matrix with wheel radii r_i

- Collecting the sliding constraints:

$$\underbrace{\begin{bmatrix} C_{1f} \\ C_{1s}(\beta_s) \end{bmatrix}}_{C_1(\beta_s)} R(\theta)\dot{\xi}_I = 0$$

Example: Differential drive constraints

- Combine rolling and sliding constraints

$$\begin{bmatrix} J_1(\beta_s) \\ C_1(\beta_s) \end{bmatrix} R(\theta)\dot{\xi}_I = \begin{bmatrix} J_2\dot{\varphi} \\ 0 \end{bmatrix}$$

- Assume robot is facing along X
  - $\alpha_R = -\frac{\pi}{2}$ and $\beta_R = \pi$
  - $\alpha_L = \frac{\pi}{2}$ and $\beta_L = 0$
- Insert α and β in J_1 and C_1 and remember

$$J_2 = \begin{bmatrix} r_R & 0 \\ 0 & r_L \end{bmatrix}$$

Example cont'd

- Wheel axes are parallel which means that the two sliding constraints are identical (can skip one)

$$\begin{bmatrix} \begin{bmatrix} 1 & 0 & l \\ 1 & 0 & -l \end{bmatrix} \\ \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \end{bmatrix} R(\theta)\dot{xi}_I = \begin{bmatrix} J_2\dot{\varphi} \\ 0 \end{bmatrix}$$

- Inverting results in:

$$\dot{\xi}_I = R(\theta)^{-1} \begin{bmatrix} 1 & 0 & l \\ 1 & 0 & -l \\ 0 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} J_2\dot{\varphi} \\ 0 \end{bmatrix}$$

$$= \frac{1}{2} R(\theta)^{-1} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 2 \\ 1/l & -1/l & 0 \end{bmatrix} \begin{bmatrix} r_R\dot{\varphi}_R \\ r_L\dot{\varphi}_L \\ 0 \end{bmatrix} = R(\theta)^{-1} \underbrace{\begin{bmatrix} \frac{1}{2}(r_R\dot{\varphi}_R + r_L\dot{\varphi}_L) \\ 0 \\ \frac{1}{2l}(r_R\dot{\varphi}_R - r_L\dot{\varphi}_L) \end{bmatrix}}_{\dot{\xi}_R}$$

- Same as with "manual" derivation from before

Degree of mobility

- Sliding constraints

$$C_1(\beta_s)R(\theta)\dot{\xi}_I = 0$$

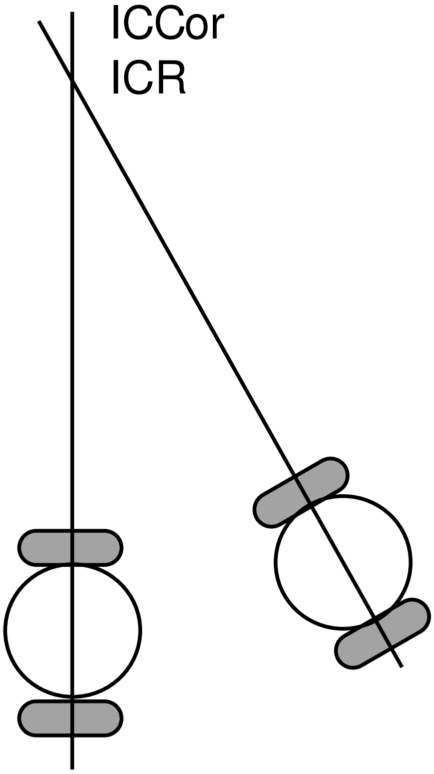
- Motion of the robot must belong to the null space of C_1, i.e.

$$C_1(\beta)m = 0, \quad m \in null(C_1)$$

- Degree of mobility, δ_m, is defined as

$$\delta_m = dim(null(C_1)) = 3 - rank(C_1)$$

Instantaneous Center of Rotation (ICR)



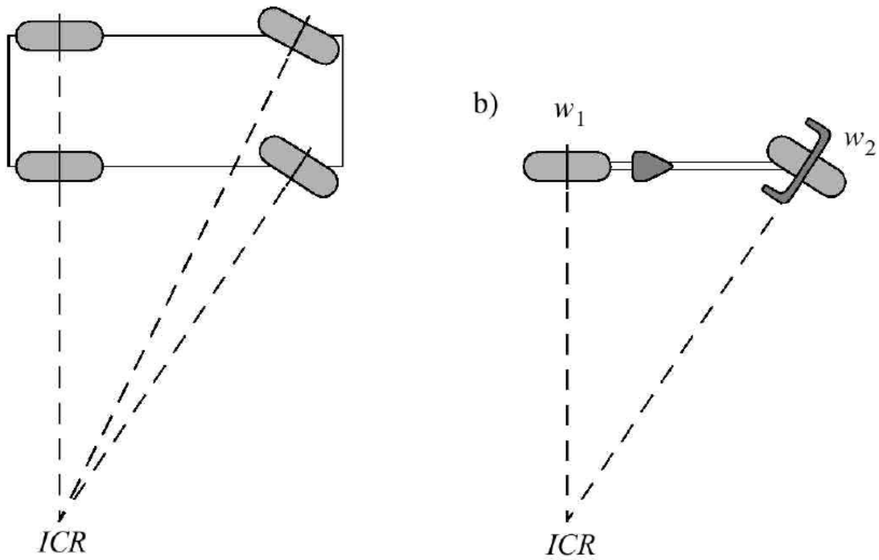
- Instantaneous Center of Rotation (ICR)
- also known as Instantaneous Center of Curvature
- Defines the point around which the platform is rotating
- Radius is given by (fir diff. drive)

$$R = \frac{v}{\omega} = l\frac{\dot{\varphi}_R + \dot{\varphi}_L}{\dot{\varphi}_R - \dot{\varphi}_L}$$

- Straight line motion $\dot{\varphi}_R = \dot{\varphi}_L$ gives $R = \infty$.

Instantaneous Center of Rotation

- Sliding constraints can be illustrated graphically with ICR



Degree of steerability and maneuverability

- Degree of steerability, δ_s, is defined as

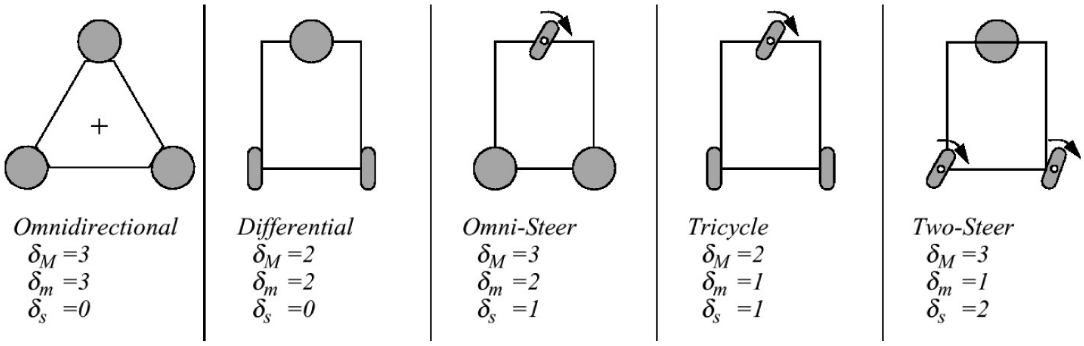
$$\delta_s = rank[C_{1s}(\beta_s)]$$

- Changing orientation of steerable wheels can lead to additional degrees of maneuverability
- Finally, degree of maneuverability

$$\delta_M = \delta_m + \delta_s$$

- $\delta_M = 2 \Rightarrow$ ICR constrained to a line
- $\delta_M = 3 \Rightarrow$ ICR anywhere in the plane

Example: Maneuverability



Holonomy

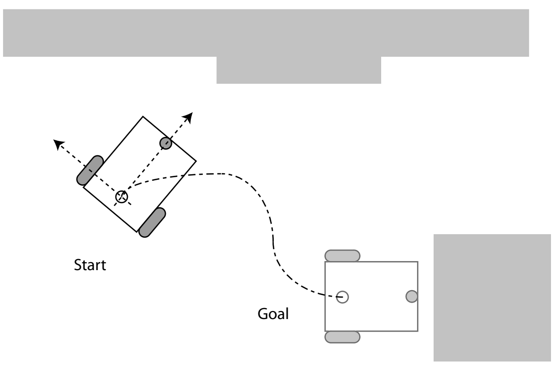
- Degrees of freedom (DOF)
  The robot's ability to achieve various poses
- Differential degrees of freedom (DDOF)
  The robot's ability to achieve various paths
- $DDOF \leq \delta_m \leq DOF$
- Holonomic robot
  - A holonomic kinematic constraint can be written as a function of position variables only
  - A non-holonomic constraint requires a differential relationship (derivative of position variables).
  - Fixed and steered wheels impose non-holonomic constraints
  - Holonomic iff $DDOF = DOF$

Beyond basic kinematics

- Strong assumptions in analysis, no sliding
- Some platforms use skid steering, i.e. steer by sliding
- Need to have friction model
- For higher speeds the dynamics must be taken into account

Path/trajectory considerations

- The constraints only define what can be achieved
- but not how?
- Trajectory planning covered partly in last lecture
- Trajectory control/tracking: given a trajectory specification, how can the robot move to follow it?
- A trajectory is a path with time specified



Open loop trajectory control

- Trajectory often divided into segments of clearly defined shape, such as lines and arcs
- Problems with getting smooth trajectories
- No adaptation to dynamic changes of the environment
- Better to use closed loop control and sensor feedback