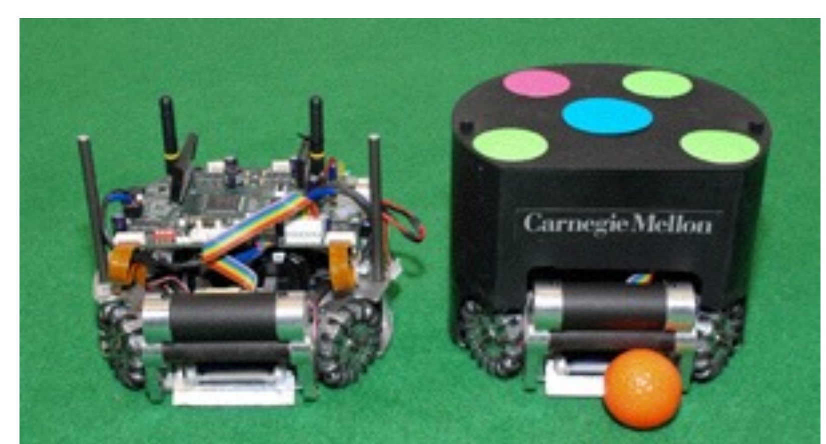
Radgetriebene Systeme

Mobilität, Räder

- Räder benötigen weniger Energie und erlauben eine schnellere Fortbewegung (auf entsprechendem Terrain)
- Benötigen Kinematische Gleichungen, d.h. Beschreibungen wie sich ein Roboter mit unterschiedlichen Rädern bewegen
- Annahme von optimalen Bedingungen (unrealistisch)
- Es gibt zwei Arten von Rädern
  - Konventionelle Räder (Reifen)
  - Spezialräder (Omni wheels)



https://www.youtube.com/watch?v=tZ0bq-jlg-o

https://www.youtube.com/watch?v=i2SYwUH6jfk

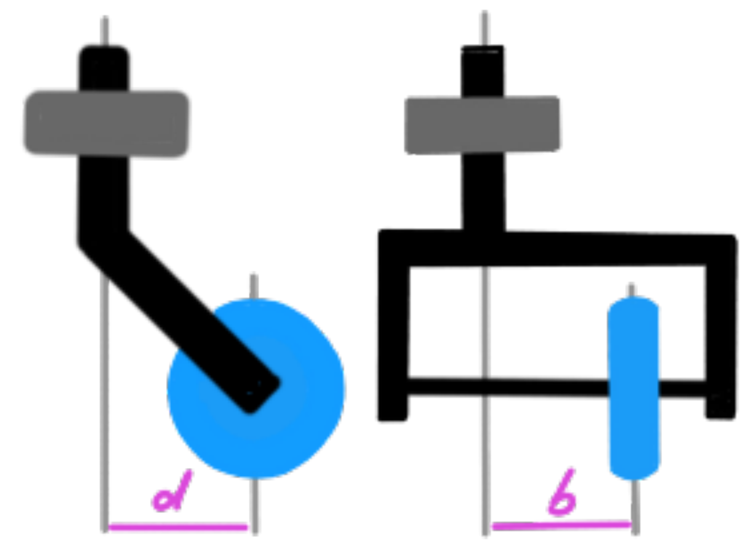
https://www.youtube.com/watch?v=pbe-YTotK3E

https://www.youtube.com/watch?v=seM8LSuDFog

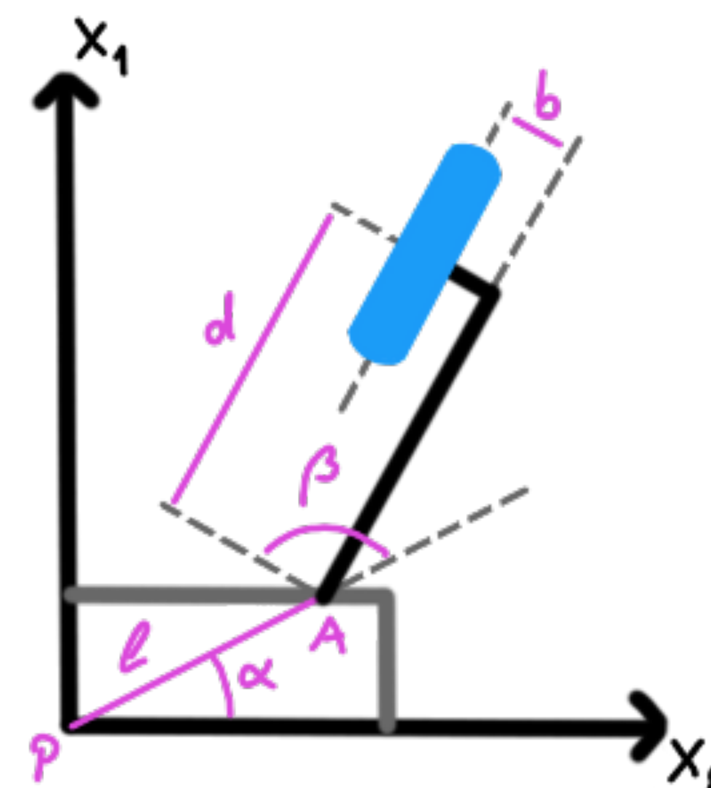
https://www.youtube.com/watch?v=TsaES--OTzM

Konventionelle Räder

- 3 Designeigenschaften, die man beachten muss
  - Offsets d und b
  - Beweglich oder fixiert
  - Passiv oder aktiv

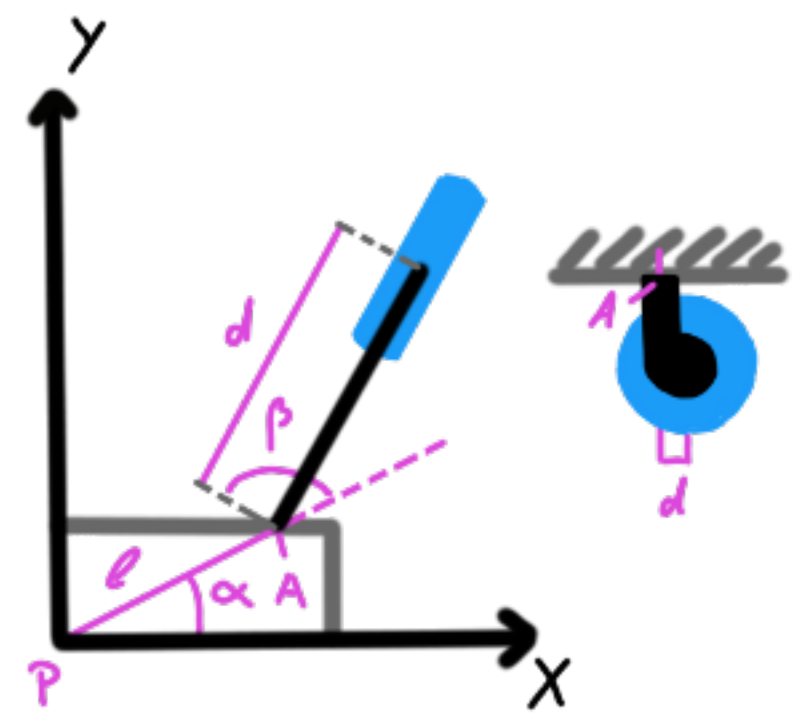


- Anmerkungen:
  - b wird meistens auf 0 gesetzt
  - Fixierte Räder limitieren die Bewegungsrichtung
  - d > 0 ist notwendig, damit sich das Rad und damit auch der Roboter drehen kann
    - Erlaubt einen holonomen Antrieb



Kinematische Gleichungen — Annahmen

- Perfekte Starrkörper
- Perfekte, nicht deformierbare Räder
- Ebener Untergrund
- Kontakt zwischen Rad und Boden ist ein Punkt
- Bewegung des Roboters entsteht, weil die Geschwindigkeit im Kontaktpunkt 0 ist



Repräsentation des Roboters: $\xi = (x\ y\ \theta)^T$

ξ — *Posture*

x,y — Globale Koordinaten

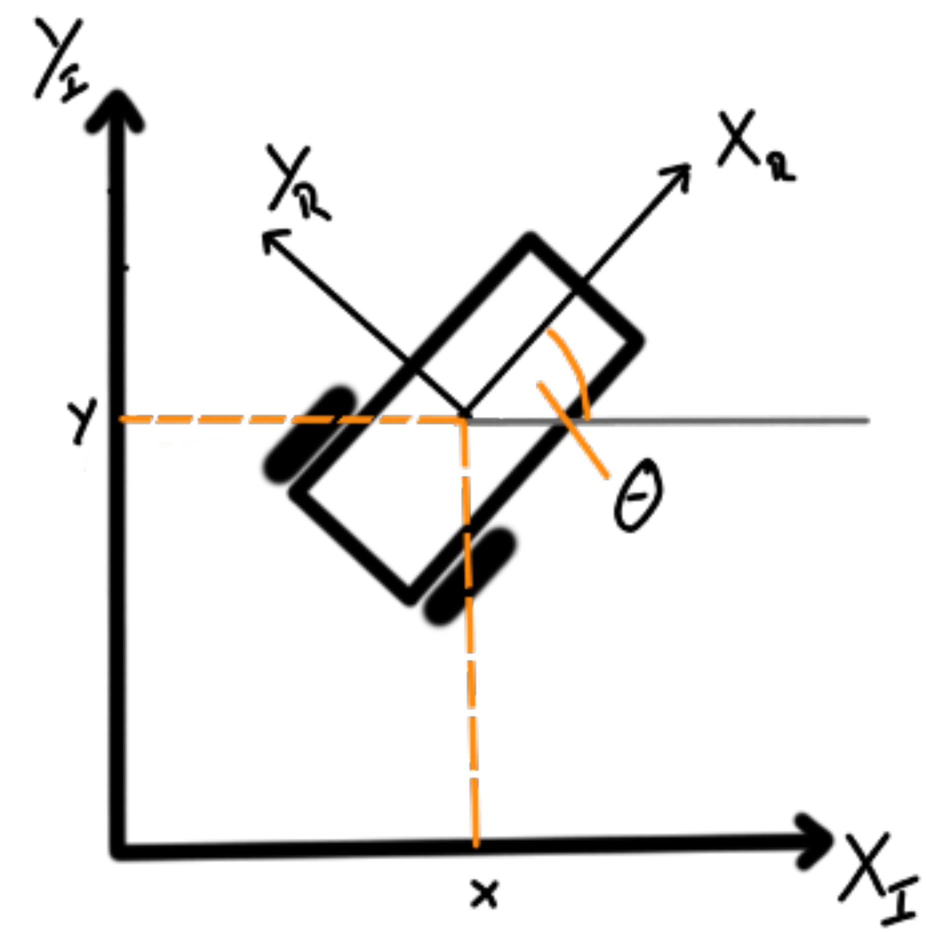
θ — Globale Orientierung

Kinematische Gleichungen

- Um die Bewegung eines Roboters zu beschreiben, muss man zunächst die Bewegung der einzelner Räder modellieren

- Zwei Referenzsysteme, ein lokales (*R* — Robot) und ein globales (*I* — Intertialsystem)

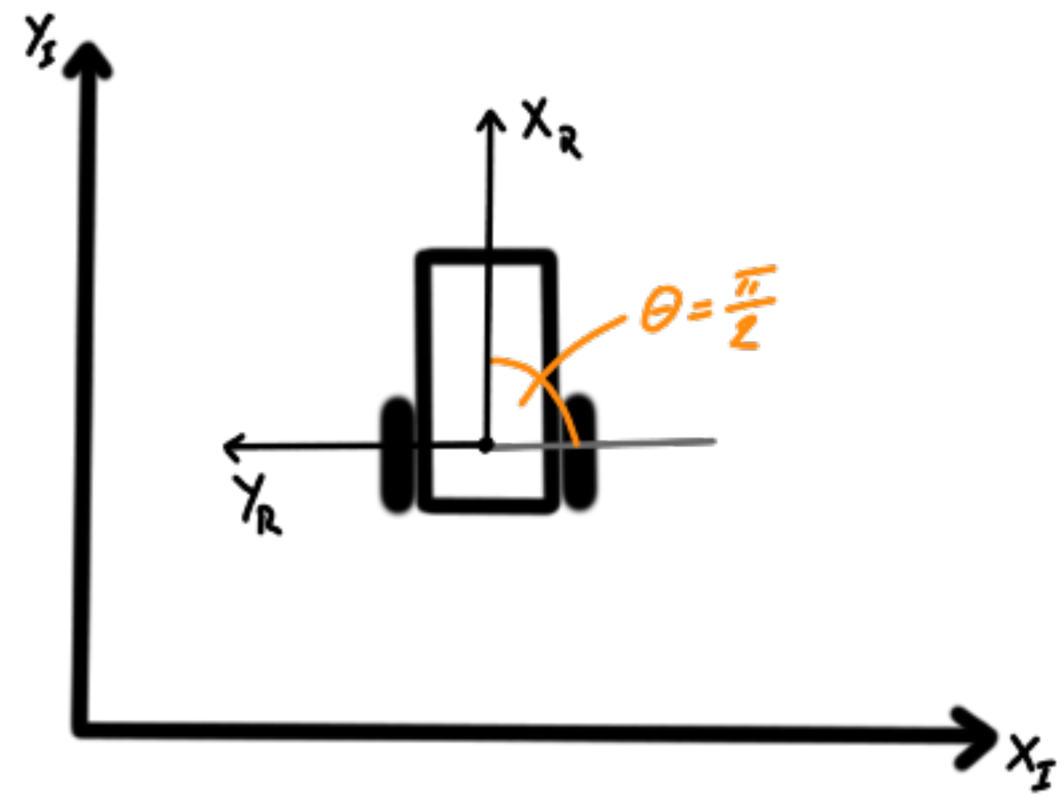
- Transformation der globalen Bewegung in das lokale Referenzsystem erfolgt mit der Rotationsmatrix



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

$$\dot{\xi}_R = R(\theta)\dot{\xi}_I$$

Beispiel — Transformation der Bewegung



$$R\left(\frac{\pi}{2}\right) = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

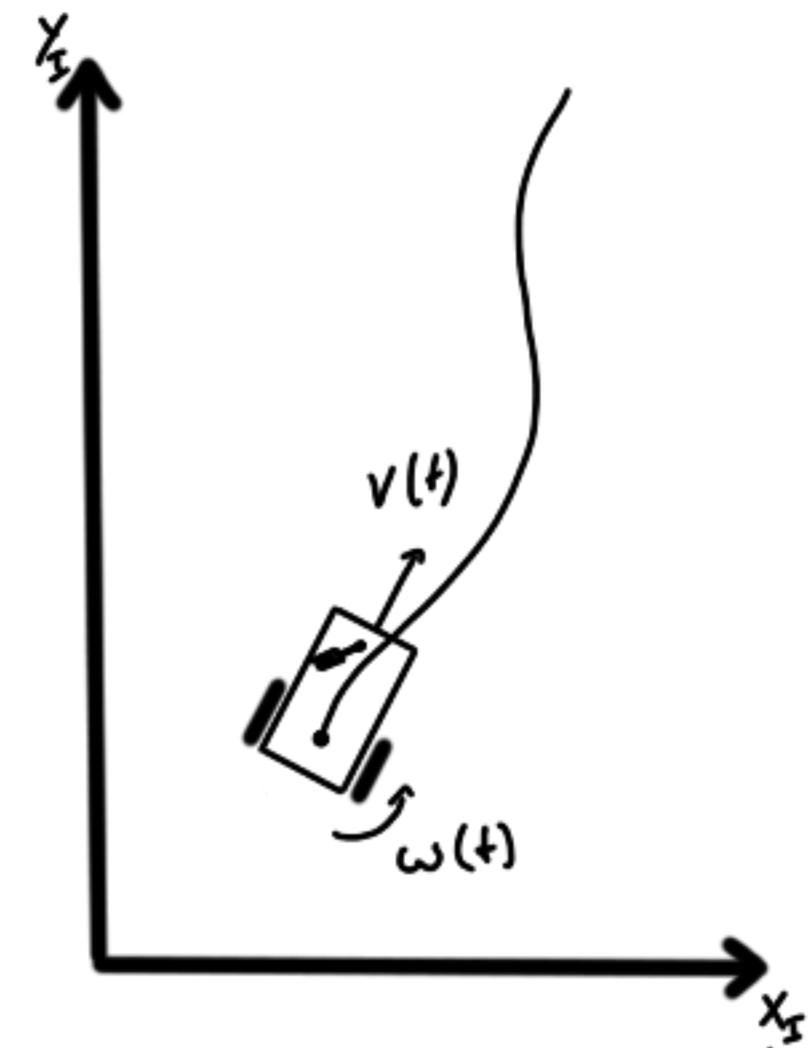
$$\dot{\xi}_R = R\left(\frac{\pi}{2}\right)\dot{\xi}_I = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{\theta}_I \end{pmatrix} \Rightarrow \begin{pmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{pmatrix} = \begin{pmatrix} \dot{y}_I \\ -\dot{x}_I \\ \dot{\theta}_I \end{pmatrix}$$

Bewegung erfolgt entlang $x_R = \dot{y}_I$ und $y_R = -\dot{x}_I$

Vorwärtskinematik für eine einfache (und weit verbreitete) Robotergeometrie

- Konfiguration:
  - 2 Räder mit Radius *r*
  - Referenzpunkt *P* ist in der Mitte der beiden Räder
  - Distanz der Räder zum Referenzpunkt *P* ist *l*
- Ein kinematisches Modell beschreibt die Bewegung des Roboters in globalen Koordinaten als Funktion von $\theta, r, l, \dot{\varphi}_r, \dot{\varphi}_l$, d.h. wir suchen *f*, so dass

$$\dot{\xi}_I = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = f(\theta, r, l, \dot{\varphi}_r, \dot{\varphi}_l)$$



Herleitung für einen Differential-Drive Robot

Startpunkt: $\dot{\xi}_R = R(\theta)\dot{\xi}_I$. Durch Invertierung erhalten wir $\dot{\xi}_I = R(\theta)^{-1}\dot{\xi}_R$

Ansatz: Berechnen, wieviel jedes Rad zu $\dot{\xi}_R$ beiträgt

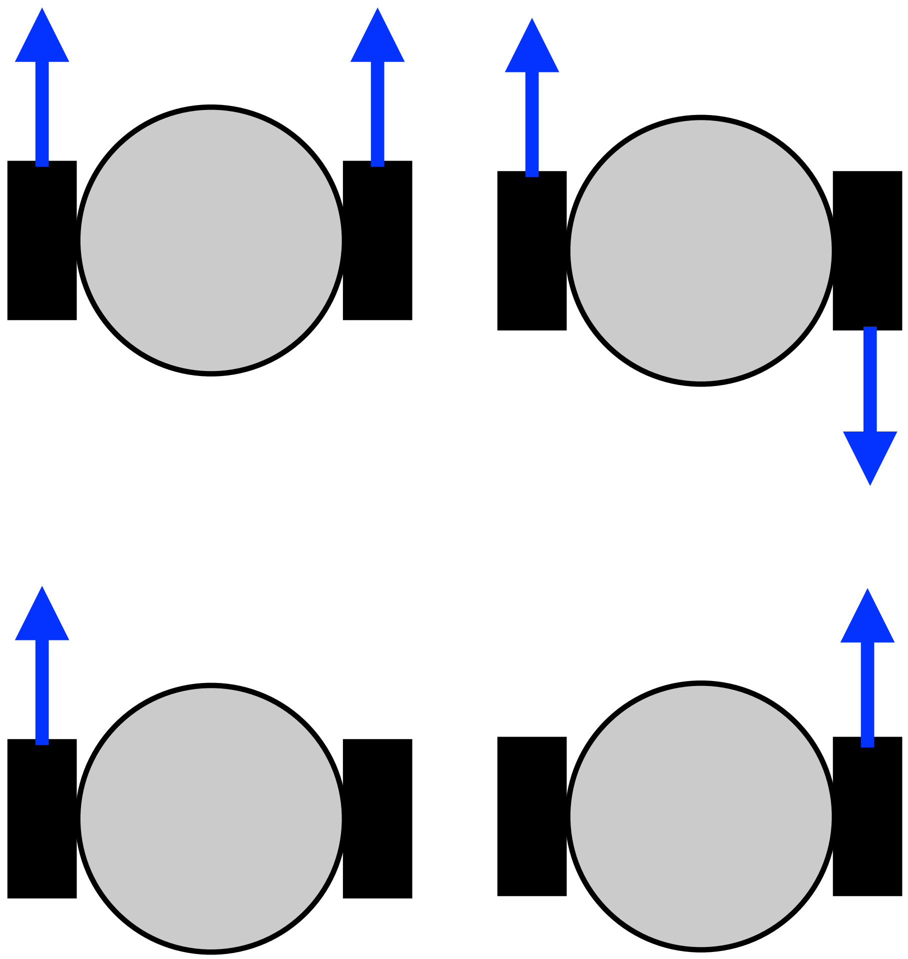
$$\dot{y}_R = 0 \quad \text{[per Definition]}$$

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

$$\dot{\theta}_R = \underbrace{\frac{r\dot{\varphi}_r}{2l}}_{(1)} - \underbrace{\frac{r\dot{\varphi}_l}{2l}}_{(2)}$$

(1) Rechtes Rad rotiert *P* im Uhrzeigersinn

(2) Linkes Rad rotiert *P* gegen den Uhrzeigersinn



Herleitung für Differential-Drive Robot

Lösung

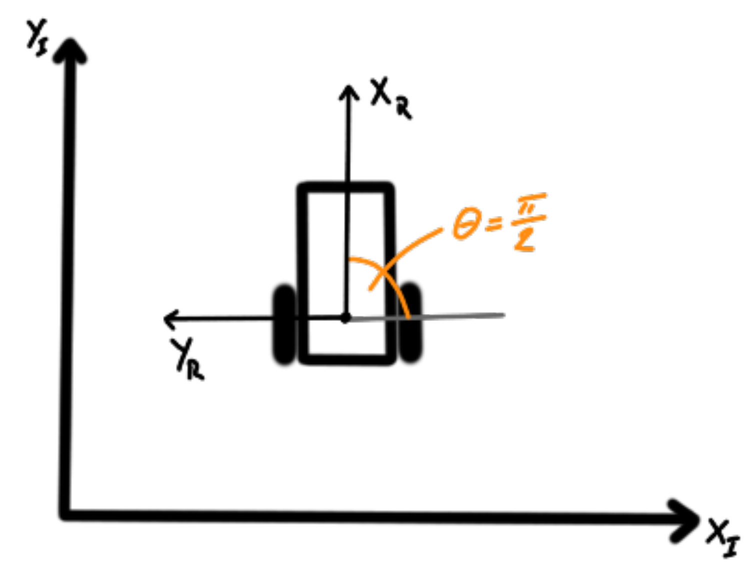
$$\dot{\xi}_I = R(\theta)^{-1} \begin{pmatrix} \frac{r\dot{\varphi}_r}{2} + \frac{r\dot{\varphi}_l}{2} \\ 0 \\ \frac{r\dot{\varphi}_r}{2l} - \frac{r\dot{\varphi}_l}{2l} \end{pmatrix}$$

$$R(\theta)^{-1} = R(\theta)^T = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Beispielrechnung:

$\dot{\varphi}_l = 4 \qquad \dot{\varphi}_r = 2$

$$\dot{\xi}_I = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 3 \\ 1 \end{pmatrix}$$

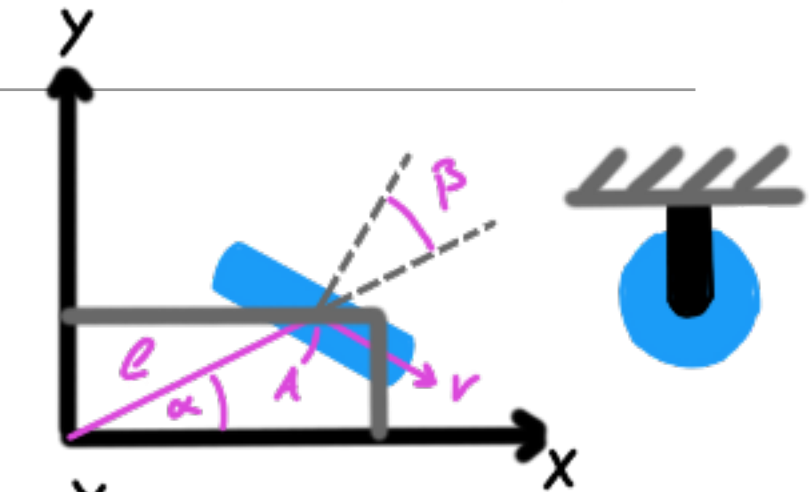


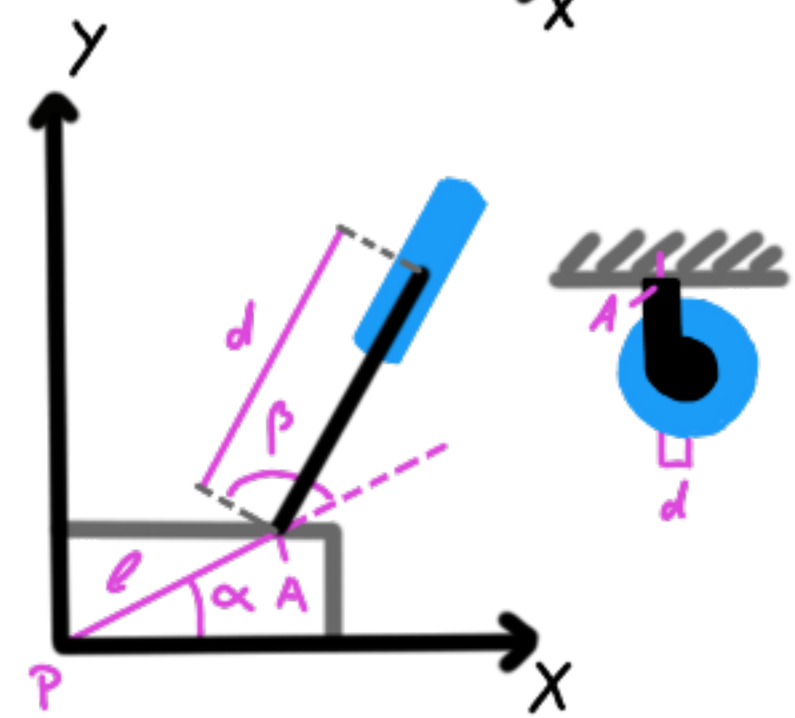
Kinematische Gleichung — Annahmen

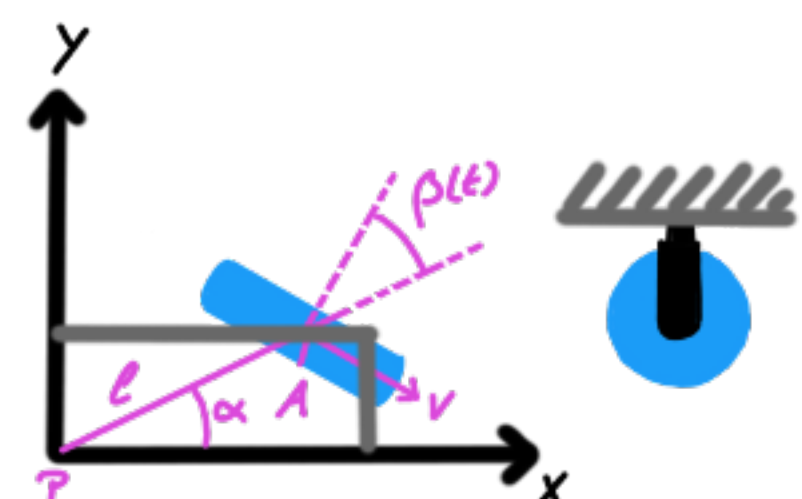
- Radebene bleibt immer senkrecht zum Boden
- Boden ist eben
- Kontakt zwischen Rad und Boden ist ein Punkt
- Keine Deformation der Räder und des Bodens
- Die Bewegung eines Rades wird vollständig durch die *pure rolling condition* und *nonslip condition* beschrieben

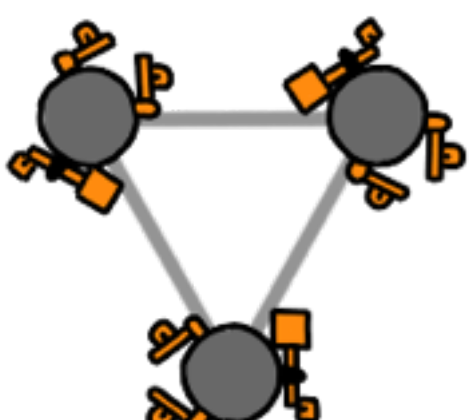
Unterschiedliche Radarten

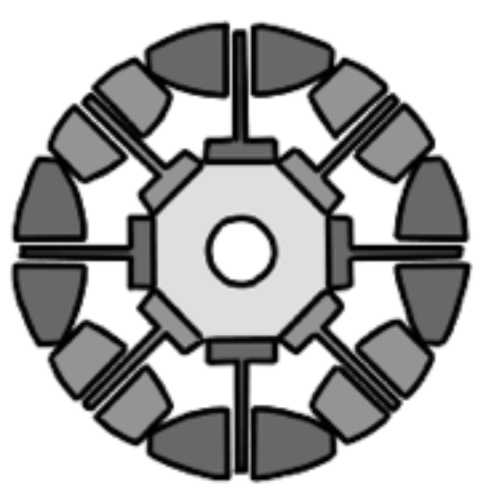
- Konventionelle Räder
  - Passiv mit fester Achse
  - Aktives Rad mit d = 0
  - Passives Lenkrad (*passive caster wheel*)
  - Aktives Lenkrad
- Spezialräder:
  - Kugelräder
  - Sweden wheel (Omni wheel)
  - Mecanum wheel

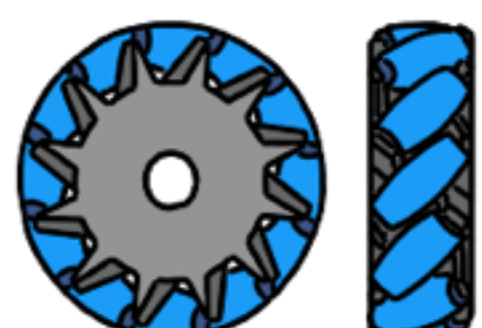






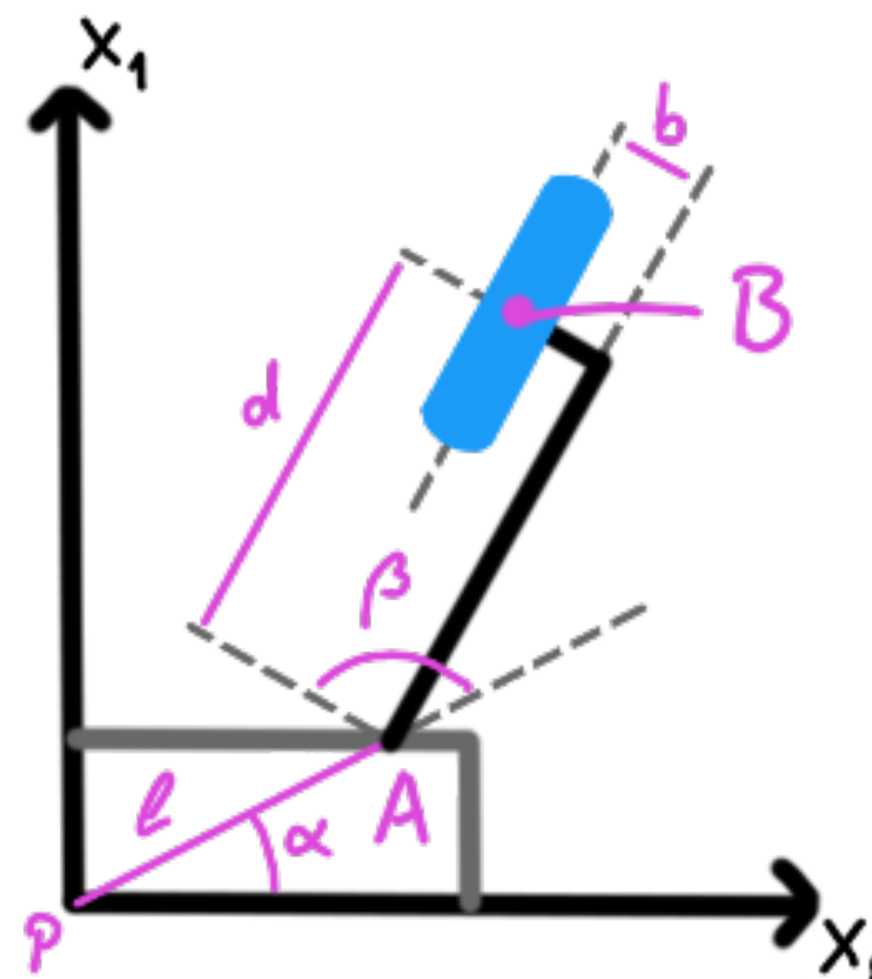






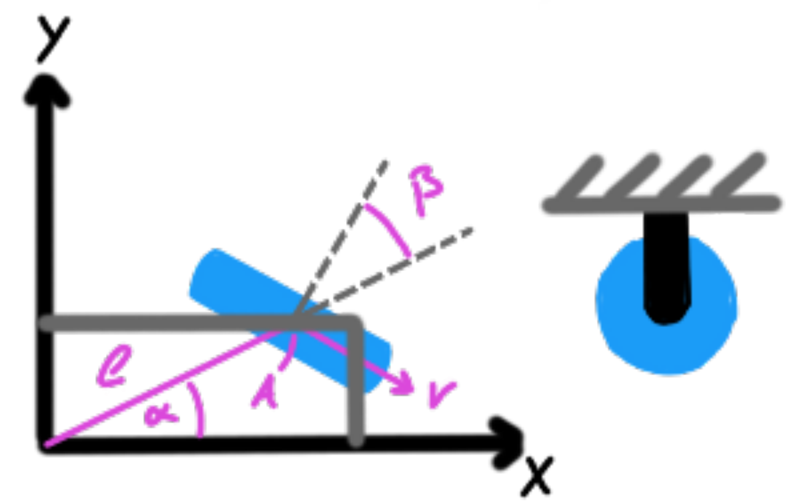
Kinematische Gleichung — Notation

- Nicht-zentriertes, bewegliches Rad (maximale Anzahl der Parameter)
- Das Zentrum des Rads (B) ist durch eine feste Stange mit der Karosserie des Roboters am Punkt A verbunden
- Rotation des Rads um *A* durch β
- Rad hat Radius *r* und Rotation φ
- 4 konstante Parameter α, l, r, d
- 2 variable Parameter $\varphi(t)$, $\beta(t)$



Fixiertes Standardrad (*Standard fixed wheel*)

- Keine Rotation um den Punkt A
- Bewegung entlang der Radebene muss der Rotation des Rades entsprechen



Pure rolling condition:

$$\underbrace{\begin{pmatrix}\sin(\alpha+\beta) & -\cos(\alpha+\beta) & -l\cos(\beta)\end{pmatrix}}_{\text{Bewegung entlang der Radebene}} R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

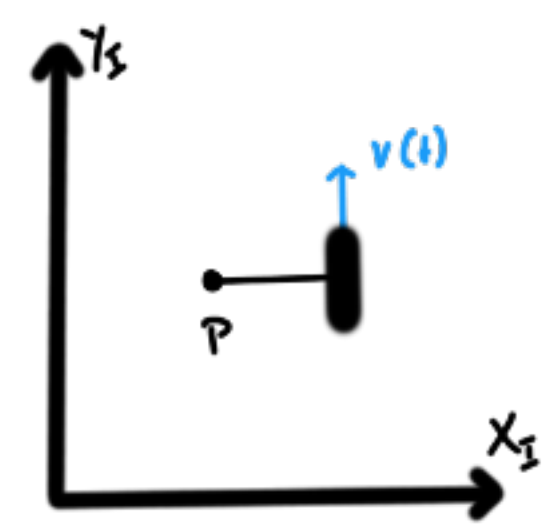
Nonslip condition:

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

Beispiel:

$$\alpha = \beta = 0 \Rightarrow (1\ 0\ 0)\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{pmatrix} = (1\ 0\ 0)\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{pmatrix} = 0$$

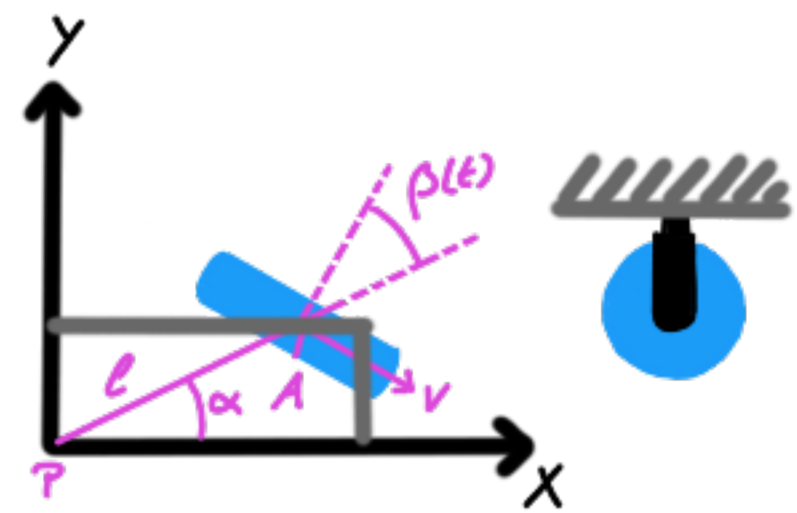
$$\Rightarrow \dot{x} = 0$$



Lenkbares Standardrad

- Zusätzlicher Freiheitsgrad β

- Einzige Änderung zum fixierten Rad: β ist nicht mehr fix, sondern eine Funktion der Zeit β(t)

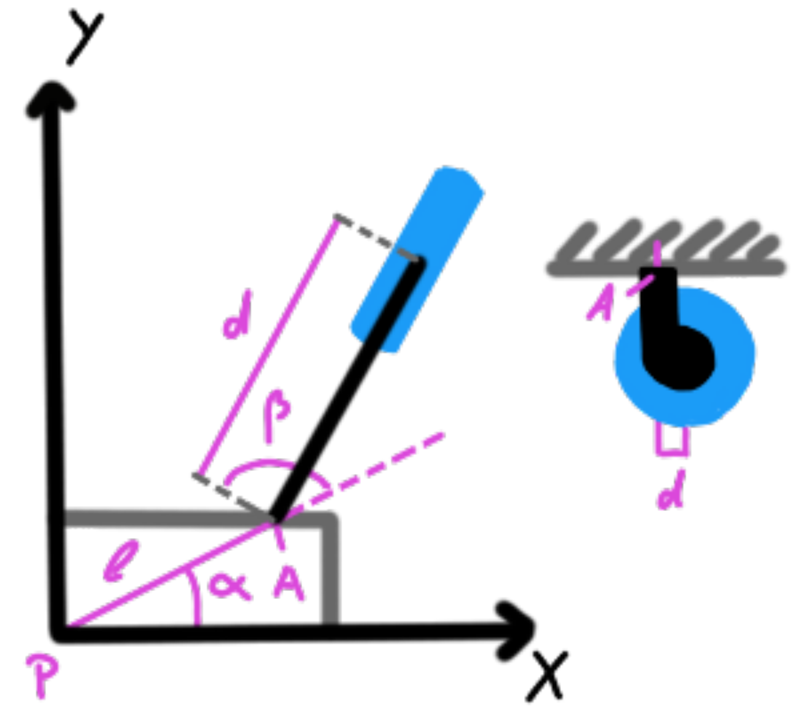


- Die Gleichungen beschreiben die instantane Änderung der Pose, daher sind die Gleichungen für die *pure rolling condition* und die *nonslip condition* identisch zum *fixed standard wheel*

- Änderungen von β werden durch die Integration über die Zeit berücksichtig

Steuerbares Rad (*castor wheel*)

Zusätzlicher Offset *d*



Pure rolling condition:

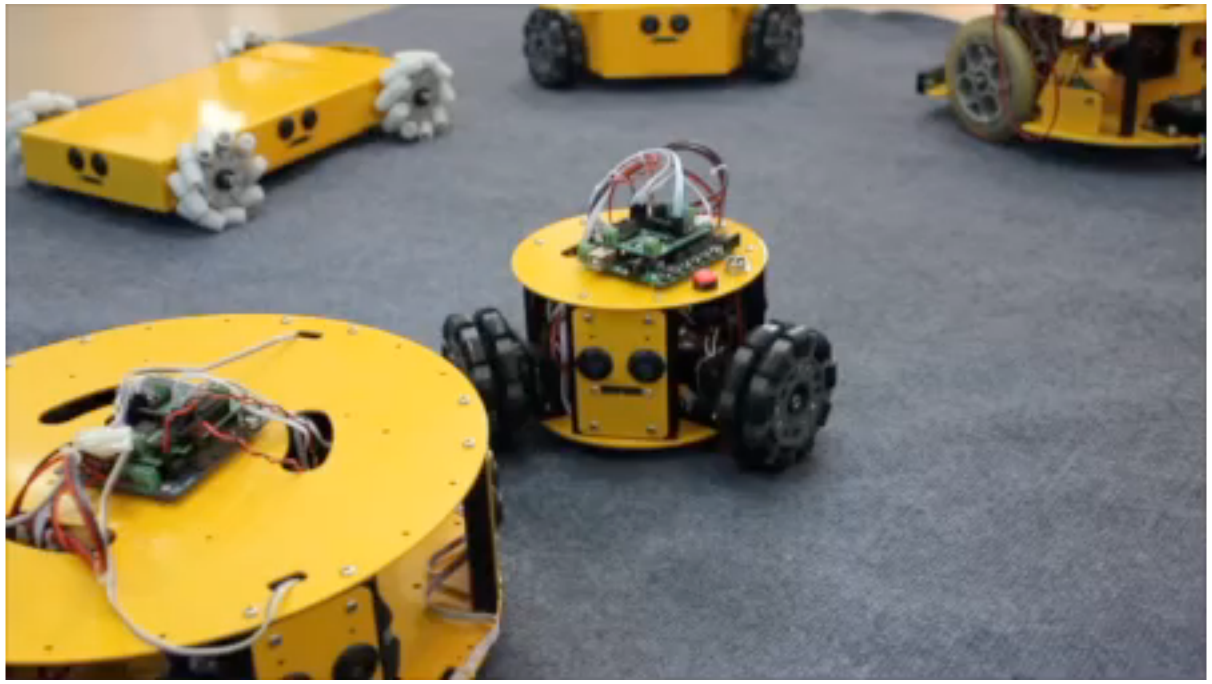
$$\left(\sin(\alpha+\beta) \quad -\cos(\alpha+\beta) \quad -l\cos(\beta)\right)R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

Nonslip condition:

$$\left(\cos(\alpha+\beta) \quad \sin(\alpha+\beta) \quad (d+l)\sin(\beta)\right)R(\theta)\dot{\xi}_I + d\dot{\beta} = 0$$

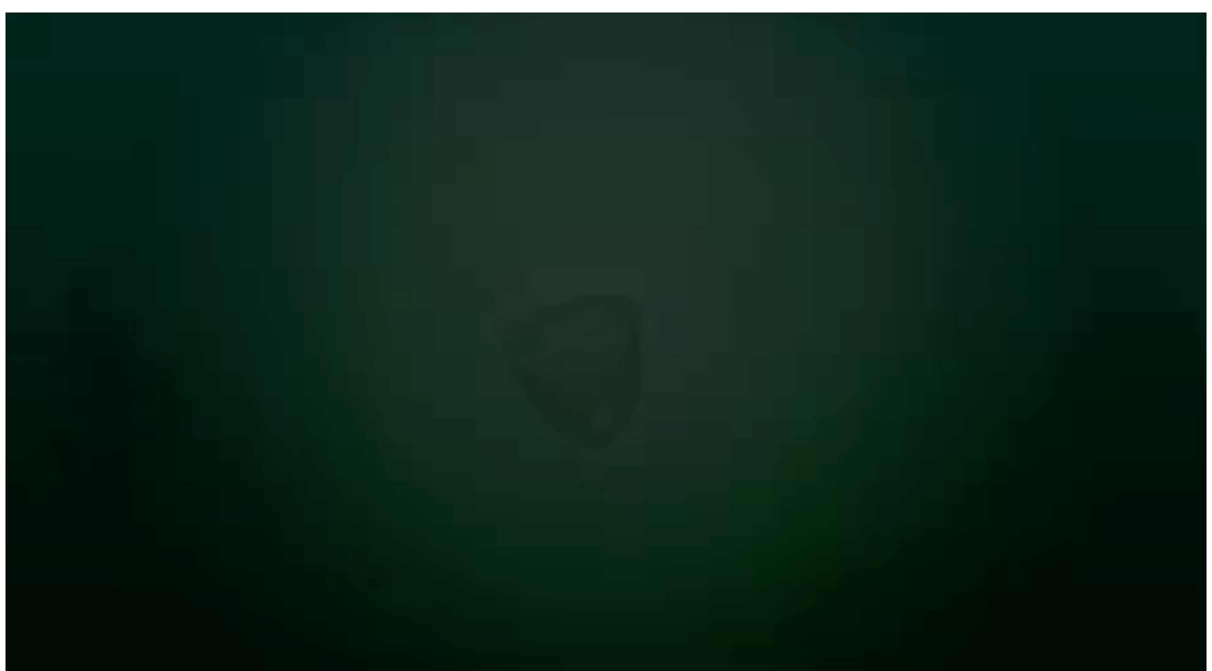
Aus den beiden Gleichungen folgt, dass jede beliebige Bewegung erzeugt werden kann: *omnidirectional* (Beispiel: Bürostuhl)

Sweden wheel versus Mecanum wheel



https://www.youtube.com/watch?v=I__IKY-Xrw4

https://www.youtube.com/watch?v=o-j9TReI1aQ



https://www.youtube.com/watch?v=IFKmGJVc9Q0

https://www.youtube.com/watch?v=IlmKcohyXG0

Sweden wheel (inkl. Mecanum wheel)

Zusätzlicher Freiheitsgrad γ beschreibt den Winkel der Räder auf dem Rad

Pure rolling condition:

$$\begin{pmatrix}\sin(\alpha+\beta+\gamma) & -\cos(\alpha+\beta+\gamma) & -l\cos(\beta+\gamma)\end{pmatrix} R(\theta)\dot{\xi}_I - r\dot{\varphi}\cos(\gamma) = 0$$

Nonslip condition:

$$\begin{pmatrix}\cos(\alpha+\beta+\gamma) & \sin(\alpha+\beta+\gamma) & l\sin(\beta+\gamma)\end{pmatrix} R(\theta)\dot{\xi}_I - r\dot{\varphi}\sin(\gamma) - r_{sw}\dot{\varphi}_{sw} = 0$$

Beispiele:

γ = 0 ⇒ Sweden wheel, γ = 45° ⇒ Mecanum wheel wheel

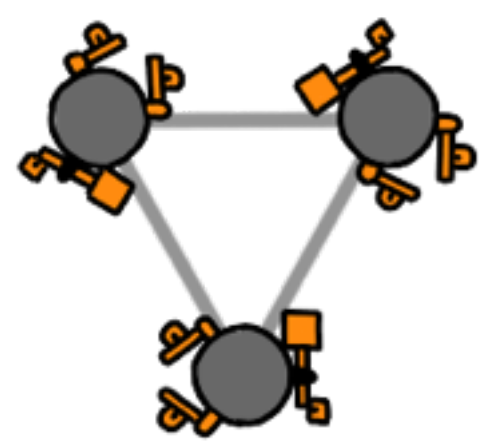
γ = 90° ⇒ kleine Rollen parallel zum Radachse. Aus der PRC folgt, dass keine Bewegung möglich ist

Kugelräder

Es gibt keine prinzipielle Rotationsachse

Die Bewegung wird vollständig durch die folgende Gleichung beschrieben

$$\left(\sin(\alpha+\beta) \quad -\cos(\alpha+\beta) \quad -l\cos(\beta)\right)R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$



Kinematische Gleichungen für radgetriebene Roboter

Der allgemeine Fall $N = N_f + N_s + N_c + N_{sw}$ Räder

Pure rolling condition:

$$J_1(\beta_s, \beta_c) R(\theta) \dot{\xi}_I - J_2 \dot{\varphi} = 0$$

$$J_1(\beta_s, \beta_c) = \begin{pmatrix} J_{1\mathrm{f}} \\ J_{1\mathrm{s}}(\beta_s) \\ J_{1\mathrm{c}}(\beta_c) \\ J_{1\mathrm{sw}} \end{pmatrix}$$

$$J_{1\mathrm{f}} = (N_\mathrm{f} \times 3)$$
$$J_{1\mathrm{s}} = (N_\mathrm{s} \times 3)$$
$$J_{1\mathrm{c}} = (N_\mathrm{c} \times 3)$$
$$J_{1\mathrm{sw}} = (N_\mathrm{sw} \times 3)$$

$J_2 = N \times N$ Radradii (für Sweden wheels mit cos γ multipliziert)

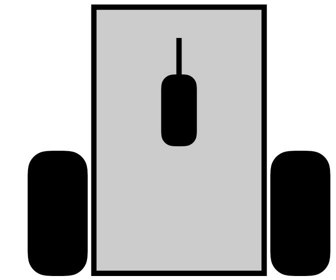
Nonslip condition:

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

$$C_1(\beta_s) = \begin{pmatrix} C_{1\mathrm{f}} \\ C_{1\mathrm{s}}(\beta_s) \end{pmatrix}$$

$$C_{1\mathrm{f}} = (N_f \times 3)$$
$$C_{1\mathrm{s}}(\beta_s) = (N_s \times 3)$$

Beispiel 1 — Differential Drive Robot



- Ausgehend von $J_1(\beta_s, \beta_c)R(\theta)\dot{\xi}_I - J_2\dot{\varphi} = 0$ berechnen wir die Bewegung des Roboters durch

$$\dot{\xi}_I = R(\theta)^{-1} J_1(\beta_s)^{-1} J_s \dot{\varphi}$$

- Castor wheel ist nicht angetrieben und kann ignoriert werden

- Antriebsräder sind fixiert, daher ist $J_1(\beta_s) = J_{1\mathrm{f}}$

- Bestimmen von α und β: $\alpha_l = -\frac{\pi}{2}, \quad \alpha_r = \frac{\pi}{2}, \quad \beta_l = \pi, \quad \beta_r = 0$

- Daraus folgt: $J_{1\mathrm{f}} = \begin{pmatrix} 1 & 0 & l \\ 1 & 0 & -l \end{pmatrix}$

- Pseudoinverse: $J_{1\mathrm{f}}^{-1} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 0 \\ \frac{1}{2l} & -\frac{1}{2l} \end{pmatrix}$

$$\boxed{\Rightarrow \dot{\xi}_I = R(\theta)^{-1} \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 0 \\ \frac{1}{2l} & -\frac{1}{2l} \end{pmatrix} J_2\dot{\varphi}}$$

Code-Example: Differential Drive Robot

```
R3(θ) =
[[cos(θ) -sin(θ) 0];
 [sin(θ)  cos(θ) 0];
 [0       0      1]];

R2(θ) =
[[cos(θ) -sin(θ)];
 [sin(θ)  cos(θ)];
];

R3i(θ) = inv(R3(θ));
```

```
J2 = eye(2)

J1(αl, βl, αr, βr, l) = [
    [sin(αl + βl)   -cos(αl + βl)   -l * cos(βl)];
    [sin(αr + βr)   -cos(αr + βr)   -l * cos(βr)]
    ]

j1 = J1(π/2, 0, -π/2, π, 1)

j1i = pinv(j1);
```

```
x = 0;
y = 0;
θ = 0;

vl = rand() * 0.05 + 0.05;
vr = rand() * 0.05 + 0.05;

N    = 2500
data = zeros(N, 5)
```

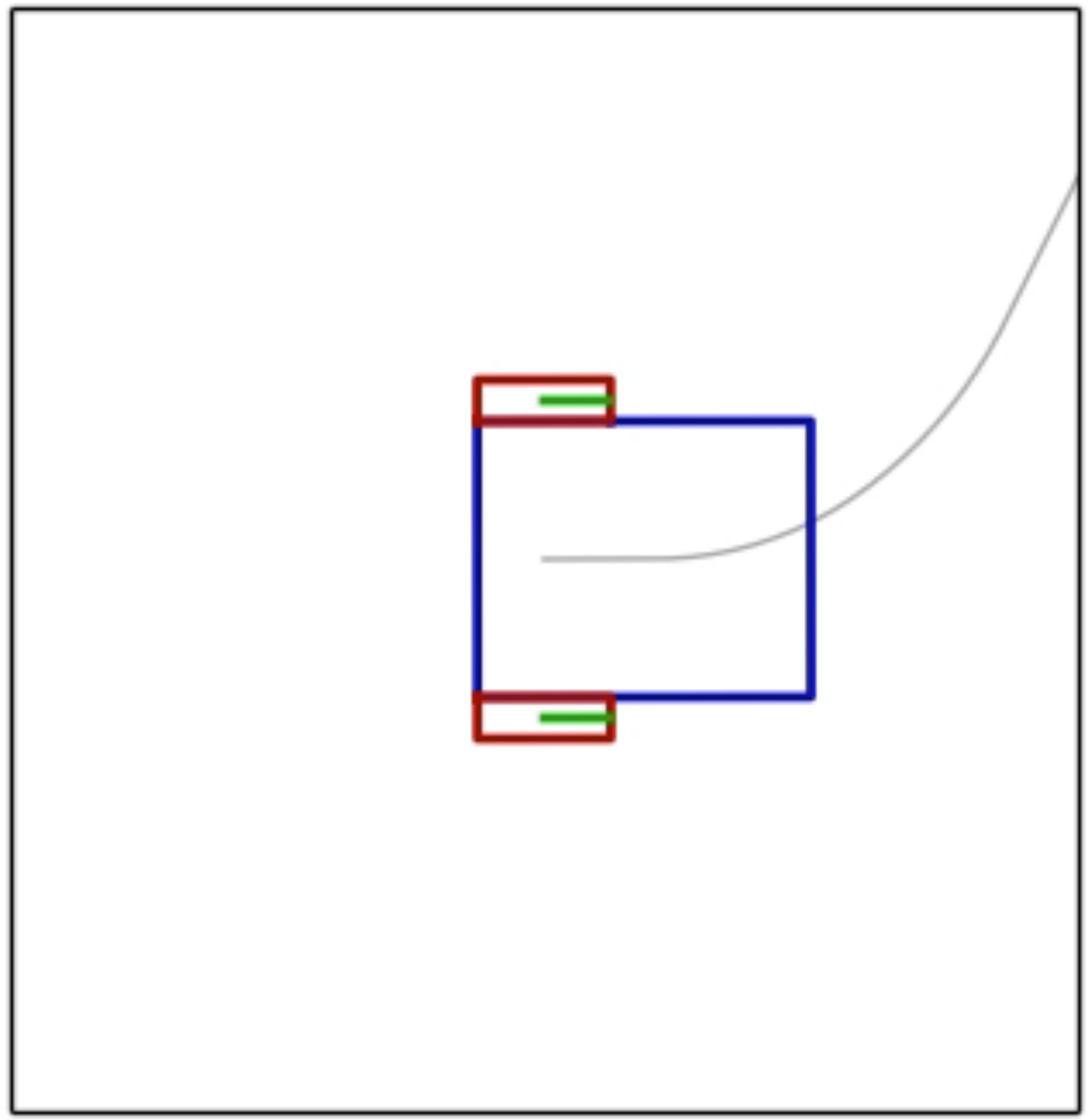
```
for i = 1:N
    if i % 10 == 0
        vl = vl + rand() * 0.2 - 0.05
        vr = vr + rand() * 0.2 - 0.05
    end
    vl = max(-0.1, min(0.3, vl))
    vr = max(-0.1, min(0.3, vr))

    r = R3i(θ) * j1i * J2 * [vl; vr];

    x = x + r[1]
    y = y - r[2]
    θ = θ + r[3]

    data[i, 1] = x
    data[i, 2] = y
    data[i, 3] = θ
    data[i, 4] = vl
    data[i, 5] = vr
end
```

```
for i=1:size(data)[1]
    fig = plotcar(data,i);
    savefig("$name/robot_$(i-1).png", bbox_inches="tight")
    close(fig)
end
```

Code-Example: Differential Drive Robot

```
car = [
    [-1.0  0.5];
    [-1.0  2.0];
    [1.0   2.0];
    [1.0  -0.5];
    [-1.0 -0.5];
    [-1.0  0.5];
]

wheel_left = [
    [-1.0 -0.5];
    [-1.3 -0.5]
    [-1.3  0.5];
    [-1.0  0.5];
    [-1.0 -0.5];
]

wheel_right = [
    [1.0 -0.5];
    [1.3 -0.5]
    [1.3  0.5];
    [1.0  0.5];
    [1.0 -0.5];
]

wheel_speed_left(v) = [
    [-1.15 0.0];
    [-1.15 5*v]
]

wheel_speed_right(v) = [
    [1.15 0.0];
    [1.15 5*v]
]
```

```
function plotcar(d, index)

    xo = d[index, 1]
    yo = d[index, 2]
    θ  = d[index, 3]
    vl = d[index, 4]
    vr = d[index, 5]
    p  = d[:,1:2]

    fig = figure(figsize=(4,4))
    ylim(yo-4, yo+4)
    xlim(xo-4, xo+4)
    xticks([])
    yticks([])

    plot(p[:,1], p[:,2], color = "#A0A0A0", lw=1.0)

    θc = θ - π/2

    cr=[R2(θc) * car[i,:]' for i = 1:size(car)[1]]
    x = [c[1] for c in cr]
    y = [c[2] for c in cr]
    plot(x + xo, y + yo, color = "#0000A0", lw= 2.0)

    wl=[R2(θc) * wheel_left[i,:]' for i = 1:size(wheel_left)[1]]
    x = [c[1] for c in wl]
    y = [c[2] for c in wl]
    plot(x + xo, y + yo, color = "#A00000", lw= 2.0)

    wr=[R2(θc) * wheel_right[i,:]' for i = 1:size(wheel_right)[1]]
    x = [c[1] for c in wr]
    y = [c[2] for c in wr]
    plot(x + xo, y + yo, color = "#A00000", lw= 2.0)

    vla = [R2(θc) * wheel_speed_left(vl)[i,:]'  for i = 1:size(wheel_speed_left(vl))[1]]
    x = [c[1] for c in vla]
    y = [c[2] for c in vla]
    plot(x + xo, y + yo, color = "#00A000", lw= 2.0)

    vra = [R2(θc) * wheel_speed_right(vr)[i,:]' for i = 1:size(wheel_speed_right(vr))[1]]
    x = [c[1] for c in vra]
    y = [c[2] for c in vra]
    plot(x + xo, y + yo, color = "#00A000", lw= 2.0)

#    savefig("differential_drive/robot_$(index-1).png", bbox_inches="tight")
    fig
end
```



Beispiel 2 — Sweden wheel Robot

Keine steuerbaren Räder, d.h. $\quad J_1(\beta_s, \beta_c) = J_{1sw}$

Aus $\quad J_1(\beta_s, \beta_c) R(\theta) \dot{\xi}_I - J_2 \dot{\varphi} = 0 \quad$ wird $\quad \dot{\xi}_I = R(\theta)^{-1} J_{1sw}^{-1} J_2 \dot{\varphi}$

Parameter sind $\alpha_1=\pi/3$, $\alpha_2=0$, $\alpha_3=-\pi/3$, $\gamma=0$, $\beta=0$ (Räder tangential zum Roboter)

$$J_{1sw} = \begin{pmatrix} \sin\left(\frac{\pi}{3}\right) & -\cos\left(\frac{\pi}{3}\right) & -l \\ 0 & -\cos(\pi) & -l \\ \sin\left(-\frac{\pi}{3}\right) & -\cos\left(-\frac{\pi}{3}\right) & -l \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} & -l \\ 0 & 1 & -l \\ -\frac{\sqrt{3}}{2} & -\frac{1}{2} & -l \end{pmatrix}$$

Beispiel

$$\dot{\varphi}_1 = 4, \quad \dot{\varphi}_2 = 1, \quad \dot{\varphi}_3 = 2$$

$$\dot{\xi}_I = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{R(\theta)^{-1}} \underbrace{\begin{pmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} & -l \\ 0 & 1 & -l \\ -\frac{\sqrt{3}}{2} & -\frac{1}{2} & -l \end{pmatrix}}_{J_{1sw}^{-1}} \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{J_2} \underbrace{\begin{pmatrix} 4 \\ 1 \\ 2 \end{pmatrix}}_{\dot{\varphi}} = \begin{pmatrix} \frac{2}{\sqrt{3}} \\ -\frac{4}{3} \\ -\frac{7}{3} \end{pmatrix}$$

Code-Example Omni-wheel

```
J2o = eye(3)

J1o(α1, β1, α2, β2, α3, β3, l) = [
    [sin(α1 + β1)   -cos(α1 + β1)   -l * cos(β1)];
    [sin(α2 + β2)   -cos(α2 + β2)   -l * cos(β2)];
    [sin(α3 + β3)   -cos(α3 + β3)   -l * cos(β3)];
]

j1o = J1o(π/3, 0, π, 0, -π/3, 0, 1)

j1oi = pinv(j1o);
```

```
x = 0
y = 0
θ = 0

v1 = rand() * 0.05 + 0.05;
v2 = rand() * 0.05 + 0.05;
v3 = rand() * 0.05 + 0.05;
```

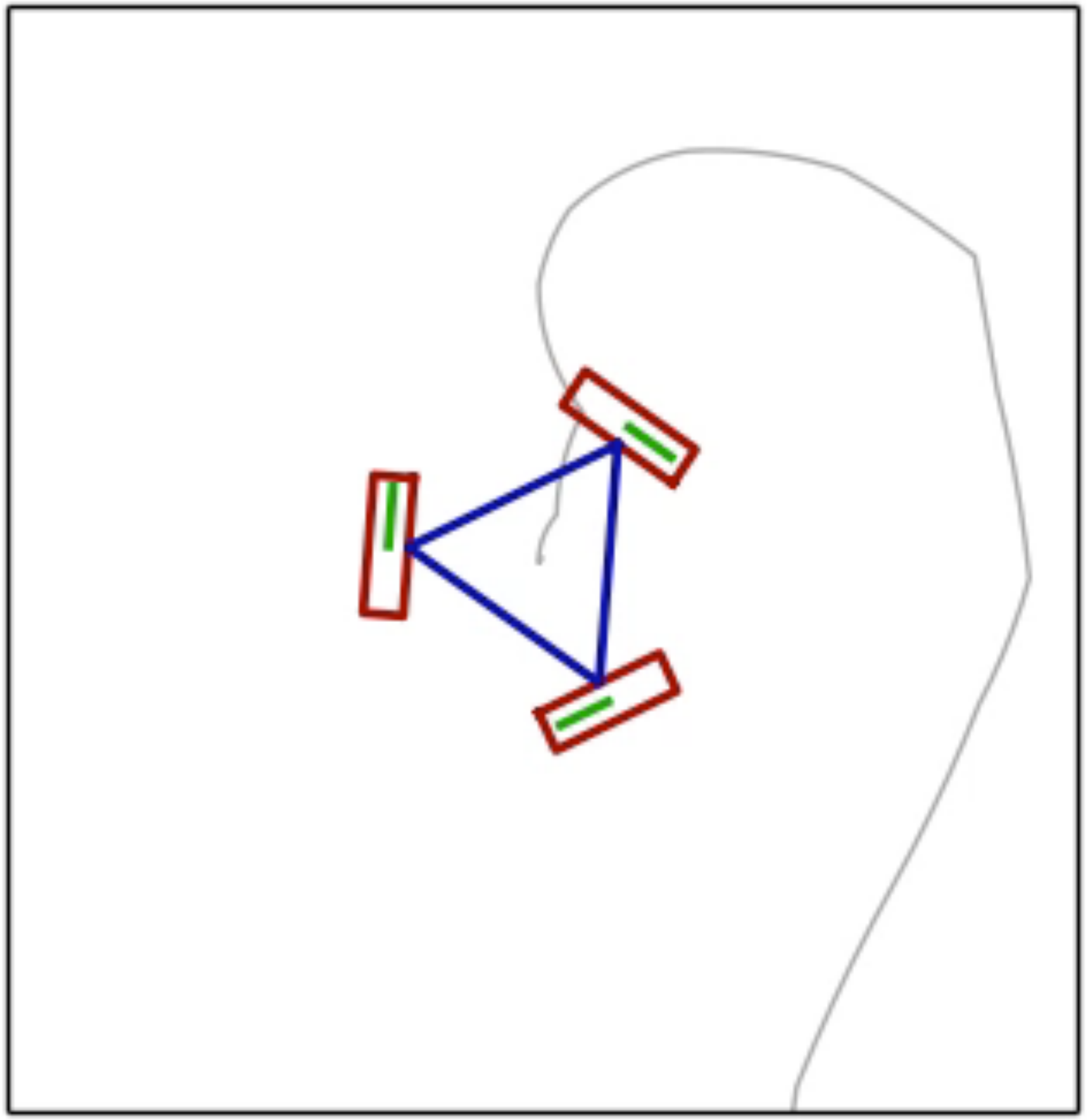
```
N = 2500
data = zeros(N, 6)

for i = 1:N
    if i % 10 == 0
        v1 = v1 + randn() * 0.05 - 0.015
        v2 = v2 + randn() * 0.05 - 0.015
        v3 = v3 + randn() * 0.05 - 0.015
    end

    v1 = max(-0.1, min(0.3, v1))
    v2 = max(-0.1, min(0.3, v2))
    v3 = max(-0.1, min(0.3, v3))

    r = R3i(θ) * j1oi * J2o * [v1; v2; v3];

    x = x + r[1]
    y = y - r[2]
    θ = θ + r[3]

    data[i, 1] = x
    data[i, 2] = y
    data[i, 3] = θ
    data[i, 4] = v1
    data[i, 5] = v2
    data[i, 6] = v3
end
```

```
for i=1:size(data)[1]
    fig = plotomni(data,i);
    savefig("$name/robot_$(i-1).png", bbox_inches="tight")
    close(fig)
end
```

Code-Example Omni-wheel

```
omni(θ) = [
    (R2(θ + π/3) * [1.0; 0.0])';
    (R2(θ + π)   * [1.0; 0.0])';
    (R2(θ - π/3) * [1.0; 0.0])';
    (R2(θ + π/3) * [1.0; 0.0])';
]

omni_wheel(θ) = [
    (R2(θ) * [1.0; -0.5])';
    (R2(θ) * [1.3; -0.5])';
    (R2(θ) * [1.3;  0.5])';
    (R2(θ) * [1.0;  0.5])';
    (R2(θ) * [1.0; -0.5])';
]

omni_wheel_speed(θ,v) = [
    (R2(θ) * [1.15; 0])';
    (R2(θ) * [1.15; v * 5])'
]
```

```
function plotomni(d, index)
    xo = d[index, 1]
    yo = d[index, 2]
    θ  = d[index, 3]
    v1 = d[index, 4]
    v2 = d[index, 5]
    v3 = d[index, 6]
    p  = d[:,1:2]

    fig = figure(figsize=(4,4))
    ylim(yo-4, yo+4)
    xlim(xo-4, xo+4)
    xticks([])
    yticks([])

    plot(p[:,1], p[:,2], color = "#A0A0A0", lw=1.0)

    w1 = omni_wheel(θ + π/3)
    w1 = reshape(w1, 5, 2)
    x = [w1[i,1] for i = 1:size(w1)[1]]
    y = [w1[i,2] for i = 1:size(w1)[1]]
    plot(x + xo, y + yo, color = "#A00000", lw= 2.0)

    w1v = omni_wheel_speed(θ + π/3, -v1)
    w1v = reshape(w1v, 2, 2)
    x = [w1v[i,1] for i = 1:size(w1v)[1]]
    y = [w1v[i,2] for i = 1:size(w1v)[1]]
    plot(x + xo, y + yo, color = "#00A000", lw= 2.0)
```



Zusammenfassung

- Kinematischen Gleichungen für jedes Rad
- Mit den Parametern d,l,α,β für jedes Rad, ergibt sich die kinematische Gleichung für den gesamten Roboter
- Anwendungen:
  - Simulation
  - Pfadplanung
  - Kalman-Filter