
Kapitel 3

Hindernisvermeidung

3.1.	Hindernisklassifikation	104
3.2.	Hinderniskarten	104
3.2.1.	Robotzentrierte Karten	104
3.2.2.	Sektorkarte	105
3.2.3.	Dynamische Hindernisse	105
3.2.4.	Umrechnung Sektor- --> Rasterkarte	106
3.2.5.	Rasterkarte mit Hinderniswahrscheinlichkeiten (probability map)	107
3.3.	Umfahren von Hindernissen	108
3.3.1.	WSA-Algorithmus (wandering standpoint algorithm)	109
3.3.2.	Potentialfeldmethode	110
3.3.3.	Polarhistogramm	111
3.3.4.	Basisfähigkeit "Kurvenfahrt"	112
3.3.5.	Basisfähigkeit "Fahrt einer S-Kurve"	113
3.3.6.	Basisfähigkeit "Türdurchfahrt"	114
3.3.7.	Andockmanöver	115
3.3.8.	Basisfähigkeit "Wandfolgen"	115

3.1. Hindernisklassifikation

Klassifikation der Umgebung in

- > befahrbar (= Freiraum)
- > Hindernis (nicht befahrbar - schlecht befahrbar)
- > unbekanntes Gelände

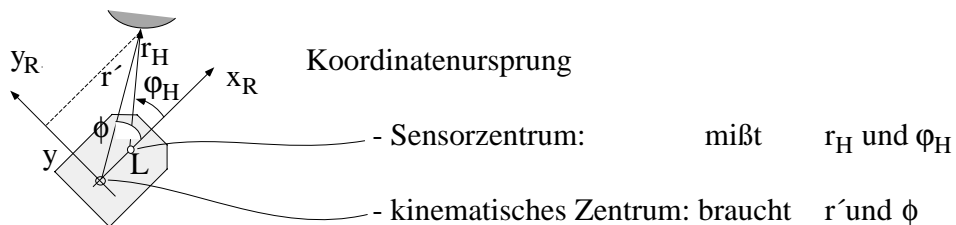
Hilfsmittel der Klassifikation

- vorgegebene Karten (zuverlässig? alle Hindernisse enthalten? i. allg. nein!)
- Bumper und/oder Tasthaare: erkennen unmittelbaren Kontakt mit Hindernis
- Sonarsensoren (Raumkegel erfasst, nur bedingt zuverlässig, kurzreichweitig)
- Laserradar (Linie von Licht, zuverlässig, langreichweitig)
- Kamera (großer Raumkegel, langreichweitig, Hindernisdetektion?)

3.2. Hinderniskarten

3.2.1. Robotzentrierte Karten

Hindernisse in Relation zum momentanen Standort des Robots



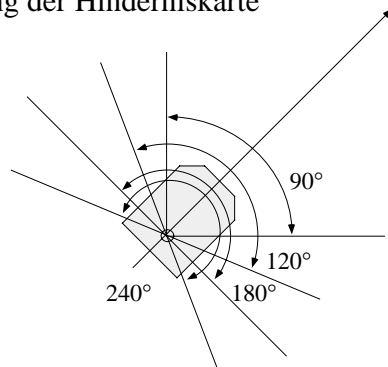
$$\sin \varphi_H = y / r_H \quad \sin \phi = y / r' \implies r_H \cdot \sin \varphi_H = r' \cdot \sin \phi$$

$$r_H \cdot \cos \varphi_H + L = r' \cdot \cos \phi \implies \begin{cases} \tan \phi = \frac{r_H \cdot \sin \varphi_H}{r_H \cdot \cos \varphi_H + L} \\ r' = \frac{r_H \cdot \sin \varphi_H}{\sin \phi} \end{cases}$$

Umrechnung der vom Roboter gemessenen Laser-Radarwerte in Werte bezogen auf das kinematische Zentrum.

3.2.2. Sektorkarte

robaufteilung der Hinderniskarte



in N Sektoren auf

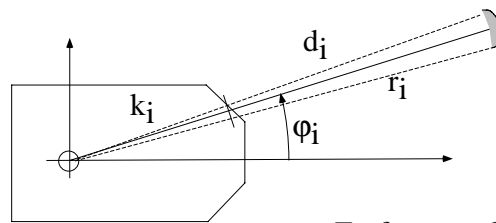
240 --> 180 --> 120 --> 90°

ausreichend ein Blickwinkel von 120°

Winkelauflösung darf gering sein,
da Roboter nur ausweichen soll

$\Delta\phi = 5^\circ \implies 36$ Sektoren in 180°

Darstellung im Polardiagramm



zu jedem ϕ_i ist im Robot k_i bekannt

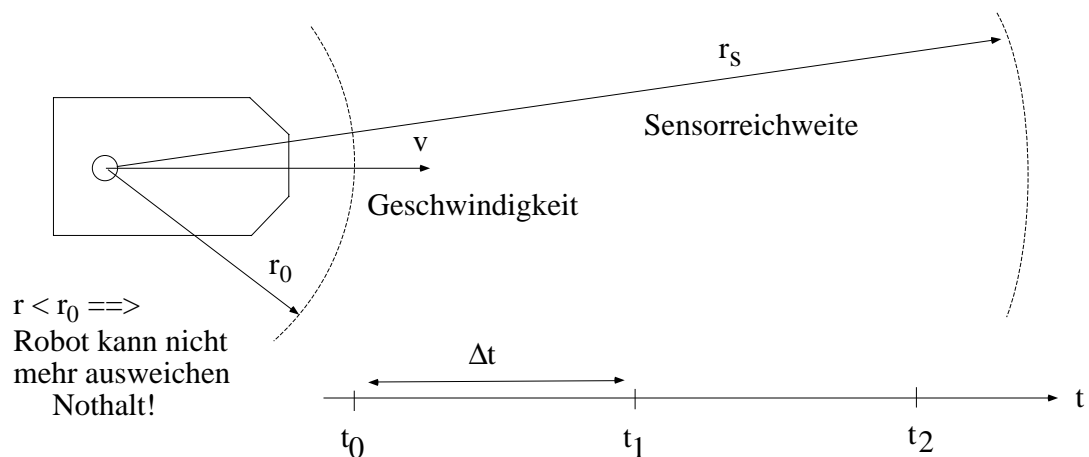
$r_i \rightarrow d_i + k_i$

gerechnet wird mit $d_i(\phi_i)$

Entfernung darf ungenau sein: $r_i \pm k_i/10$

3.2.3. Dynamische Hindernisse

Wie oft muss eine neue Karte aufgenommen werden?



$r < r_0 \implies$
Robot kann nicht
mehr ausweichen
Nothalt!

um Zeitpunkt t_1 Hindernis in Entfernung r_s (wird gerade nicht erfasst)

um Zeitpunkt $t_1 + \Delta t$ ist es bei r_0 ; $\implies \Delta s = (r_s - r_0) \implies \Delta t < \frac{\Delta s}{v}$

beispiel: $v = 1 \text{ m/s}$; $r_s = 4\text{m}$; $r_0 = 15 \text{ m}$ $\implies \Delta s = 2,5 \text{ m}$ $\implies \Delta t < 2,5 \text{ s}$

Sei Δt die Zeit zwischen zwei Sensoraufnahmen und Δs die Differenz $r_s - r_0$ in dynamisches Hindernis darf dann maximal die Geschwindigkeit

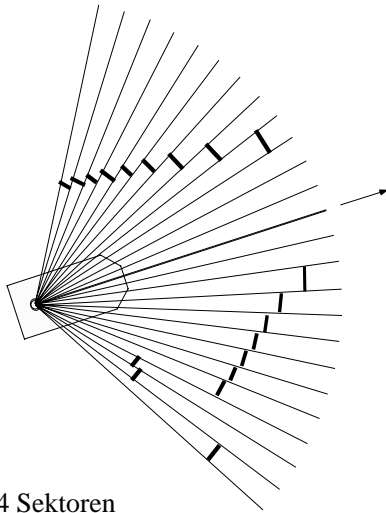
$$v_{\text{dyn}} \leq \frac{\Delta s}{\Delta t} - v \quad \text{haben}$$

Beispiel: $v = 1 \text{ m/s}$; $\Delta s = 2,5 \text{ m}$; $\Delta t = 0,5 \text{ s} \implies v_{\text{dyn}} \leq 5 \text{ m/s}$

Nur dann hat der AMR eine Chance, dem dynamischen Hindernis auszuweichen.

3.2.4. Umrechnung Sektor- --> Rasterkarte

Sektorkarte (roboterfest)

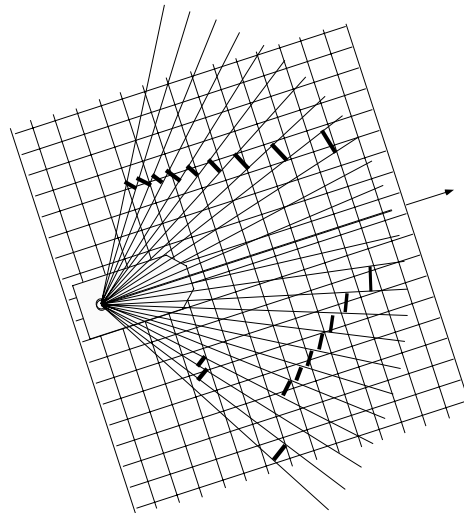


24 Sektoren

5° pro Sektor

120° Schwinkel

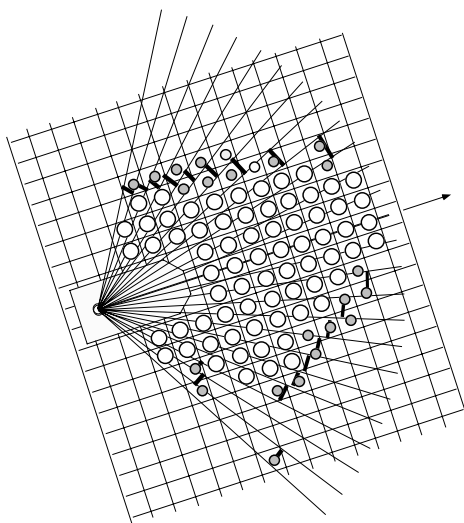
Rasterkarte (roboterfest)



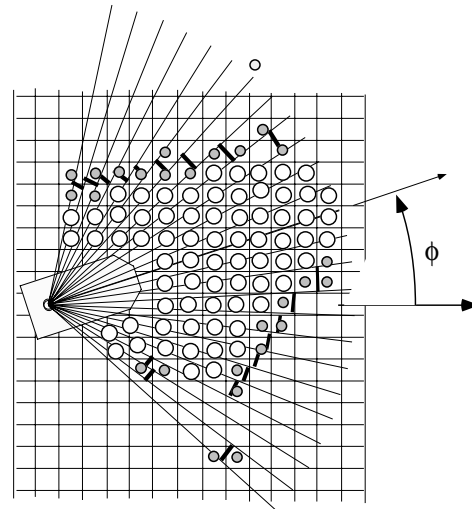
Rastermaß:

Auflösung der Hindernissensorik

Roboterfeste vs. raumfeste Rasterkarte

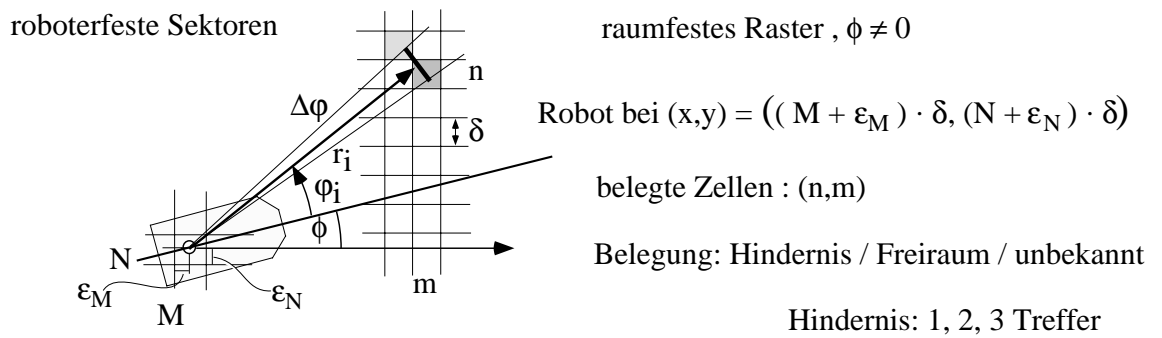


robotzentriertes Raster



raumfestes Raster

Orientierung des Robots ϕ



suche Zellen (n,m) mit Hindernismarkierung im Sektor i

$$(N - \epsilon_N + n) \cdot \delta \leq r_i \cdot \sin(\varphi_i + \phi) \leq (N - \epsilon_N + n + 1) \cdot \delta$$

Treffer:

$$(N - \epsilon_N + n) \cdot \delta \leq r_i \cdot \sin(\varphi_i + \phi \pm \Delta\phi/2) \leq (N - \epsilon_N + n + 1) \cdot \delta$$

(n,m) mit erfüllter Ungleichung für ϕ , und/oder $\phi + \Delta\phi/2$ und/oder $\phi - \Delta\phi/2$

$$(M - \epsilon_M + m) \cdot \delta \leq r_i \cdot \sin(\varphi_i + \phi) \leq (M - \epsilon_M + m + 1) \cdot \delta$$

$$(M - \epsilon_M + m) \cdot \delta \leq r_i \cdot \sin(\varphi_i + \phi \pm \Delta\phi/2) \leq (M - \epsilon_M + m + 1) \cdot \delta$$

3.2.5. Rasterkarte mit Hinderniswahrscheinlichkeiten (probability map)

Aufsammeln der Hindernistreffer : Hindernis: +1, +2, +3; Freiraum: -3
 über mehrere Sensoraufnahmen Unwissen: 0
 von verschiedenen Positionen aus

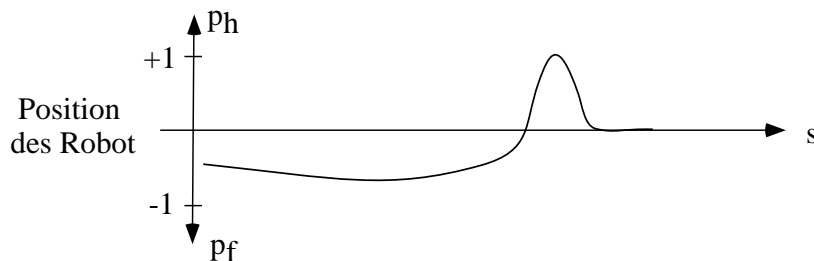
Interpretation als Wahrscheinlichkeiten

$$p_h(n,m) = \frac{\# \text{ Treffer in } (n,m)}{3 \cdot \# \text{ Sensoraufnahmen}}$$

$$p_f(n,m) = \frac{\# \text{ Treffer in } (n,m)}{3 \cdot \# \text{ Sensoraufnahmen}}$$

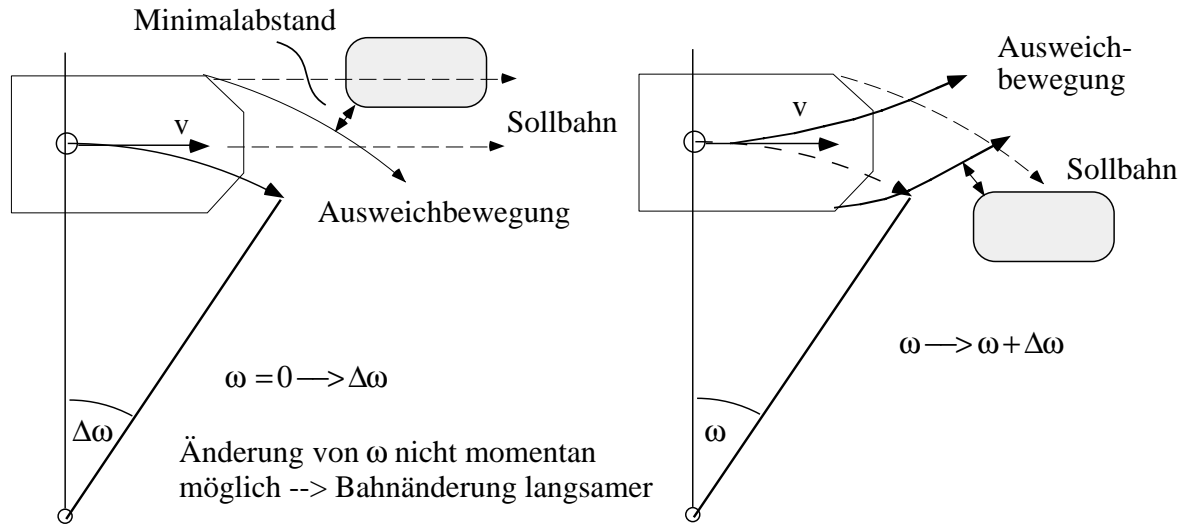
Hinderniswahrscheinlichkeit

Freiraumwahrscheinlichkeit

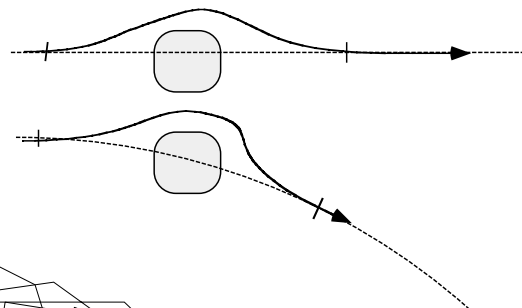


3.3. Umfahren von Hindernissen

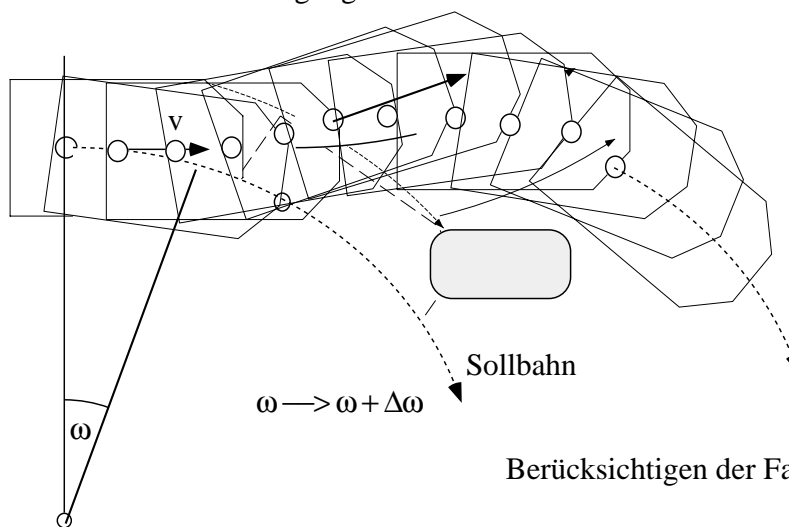
Grundfertigkeit des AMR: Hindernissen ausweichen können, d. h. bei gegebenem v ein ω so erzeugen, dass die Bahn vom Hindernis wegführt.



Ausweichbewegungen



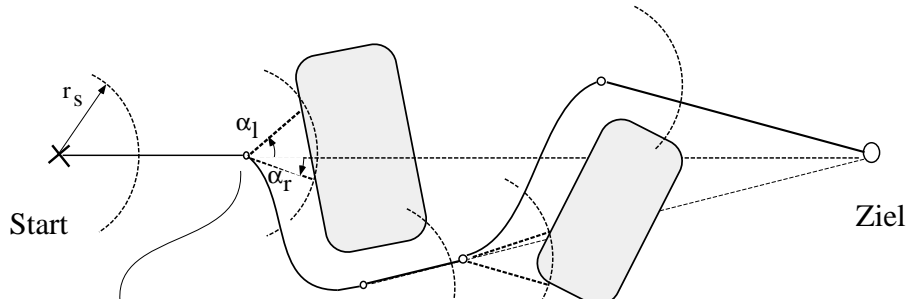
Ausweichbewegung



Berücksichtigen der Fahrzeugabmessungen

3.3.1. WSA-Algorithmus (wandering standpoint algorithm)

gegeben: Roboter mit Hindernissensor begrenzter Reichweite r_s
 Richtung zu einem Ziel (Kirchturm in der Ferne)



$\alpha_l > \alpha_r \implies$ rechts abbiegen

Mindestabstand

zum Hindernis halten Weg zum Ziel frei

\implies draufzufahren

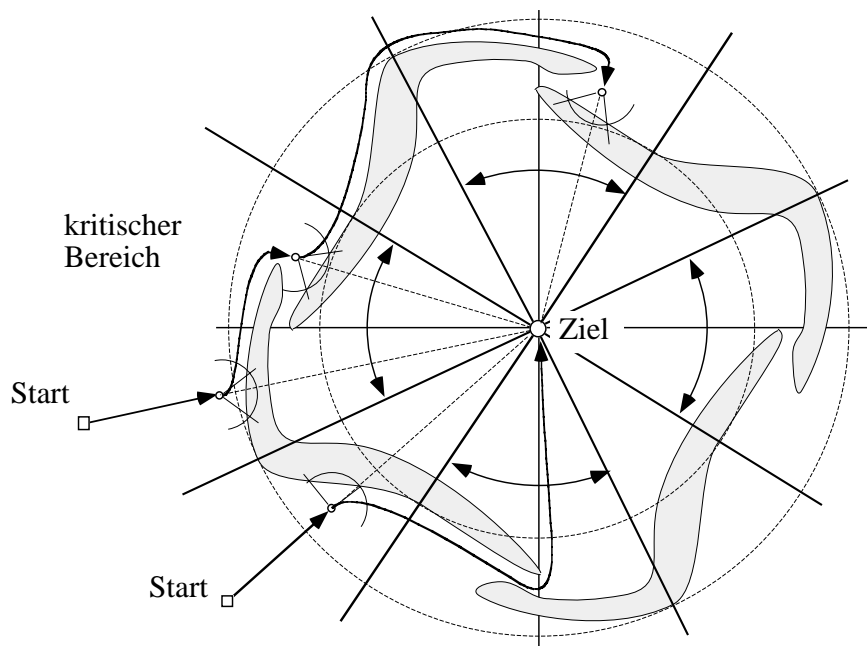
$\alpha_r > \alpha_l \implies$ links abbiegen

Entscheidung lokal ok; global ??

bewegte Hindernisse können den Roboter abdrängen

WSA-Algorithmus - pathologischer Fall

Die Vorschrift, zum kleineren Winkel hin abzubiegen und auf das Ziel hinzufahren, wenn kein Hindernis da ist, kann zu einem endlosen Weg führen. Der AMR muß erkennen können, dass er sich um 360° gedreht hat, dann sollte er sein Verhalten ändern: hat er sich nach einem Umlauf dem Ziel nicht genähert, soll er nun zum größeren Winkel hin abbiegen; hat er sich genähert, so fährt er auf einer Spirale auf das Ziel zu.

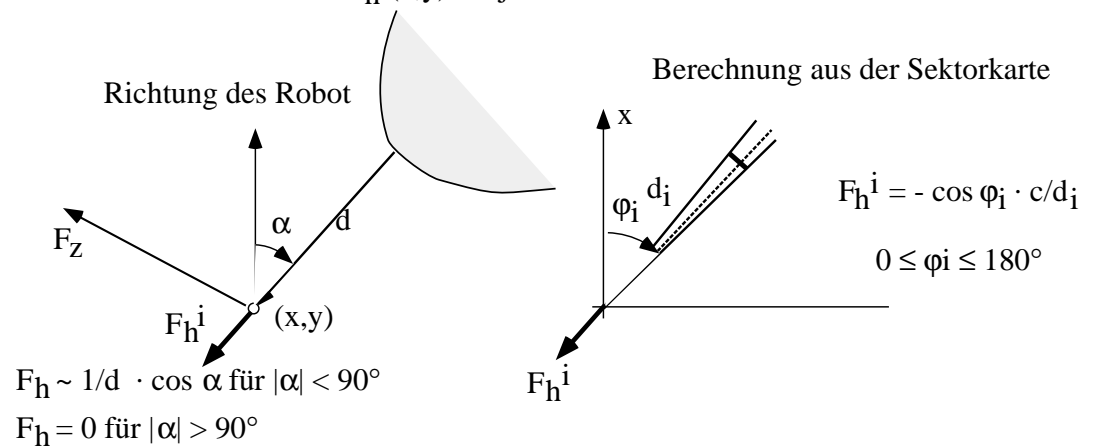


3.3.2. Potentialfeldmethode

gegeben: Robot mit Hindernisrasterkarte;
 Hindernisse als Wahrscheinlichkeiten modelliert
 Ziel (Richtung oder Position)

Potentialfeld : Zielkraft $F_Z(x,y)$ (Rinne in Richtung Ziel) für jeden Punkt (x,y)

+ Hinderniskraft $F_h^i(x,y)$ für jedes Hindernis



esamtkraft: $F_g(x,y) = F_Z(x,y) + \sum_{i=1}^n F_h^i(x,y)$ Vektorfeld $(f_x(x,y), f_y(x,y))$

Potentialfeld: $F_g(x,y) = -\text{grad } \phi(x,y)$ $\phi(x,y)$ skalares Feld (Gebirge)

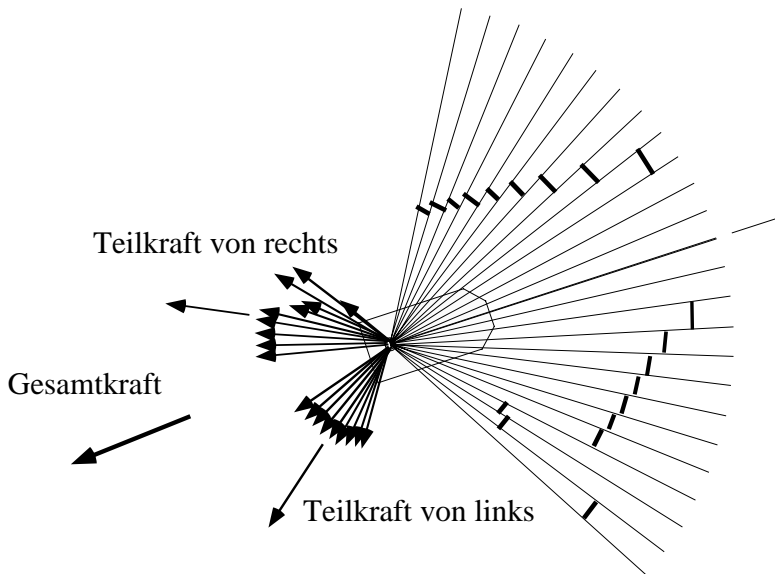
$\text{grad} = (\partial\phi/\partial x, \partial\phi/\partial y)$ (Richtung des steilsten Anstiegs)

Der Robot fährt in Richtung der Gesamtkraft (wie Wasser im Gebirge).

Probleme:

- lokale Minima - Zielkraft muß genügend stark sein (Gelände fällt hinreichend ab)
- enge Durchfahrten - Zielkraft genügend stark (Wasser fließt zwischen den Bergen durch)
- Regelung des Robot ohne Überschwingen

Der Beitrag jedes Sektors der Sektorkarte zu der Hinderniskraft ist umgekehrt proportional zur Entfernung des Hindernisses und proportional zum Cosinus des Winkels.

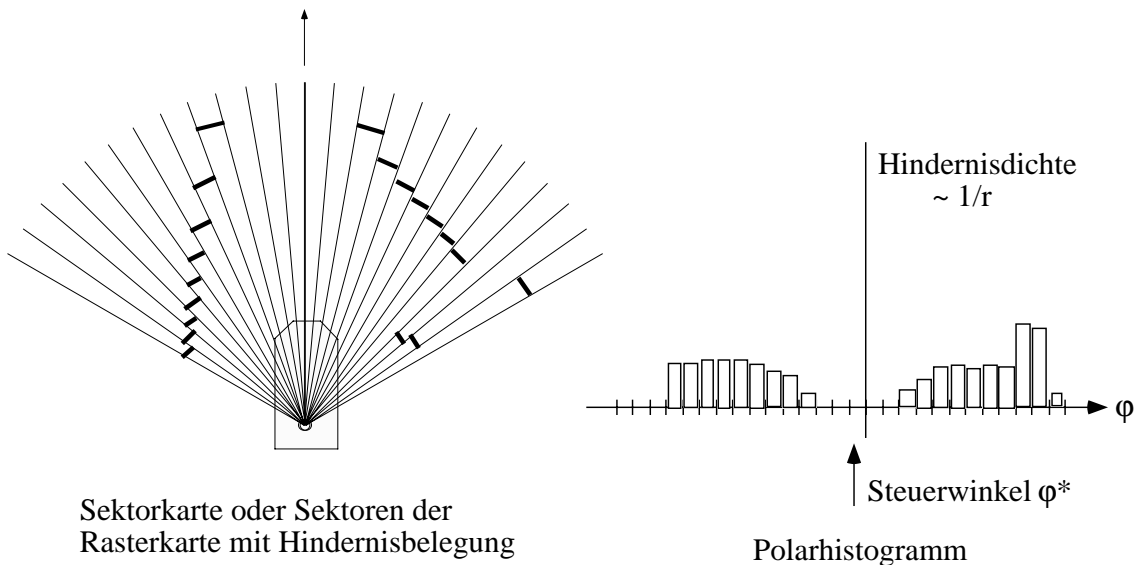


Berechnung der Hinderniskraft aus der Sektorkarte

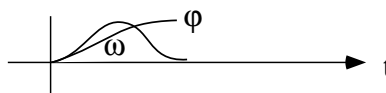
In Fahrtrichtung ist ein freier Korridor, rechts und links Hindernisse. Das nutzt aus das

3.3.3. Polarhistogramm

Es berechnet die Hindernisdichte und sucht in dem Histogramm freie Durchfahrten.



Regelungsaufgabe: wähle $\omega(t)$ so, daß $\varphi(t) = \int_0^t \omega(\tau) d\tau$ den Steuerwinkel bildet



3.3.4. Basisfähigkeit "Kurvenfahrt"

Es soll eine Kurve gefahren werden, bei der der AMR um einen Winkel $\Delta\phi$ schwenkt.

gegeben : Form von $\omega(t)$

$$\omega(t) = \omega_0 \cdot \left(1 - \cos \frac{2\pi}{T} \cdot t\right)$$

$$\Delta\phi = \int_0^T \omega(\tau) d\tau$$

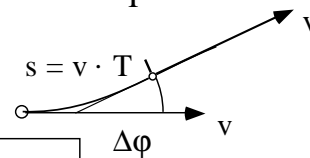
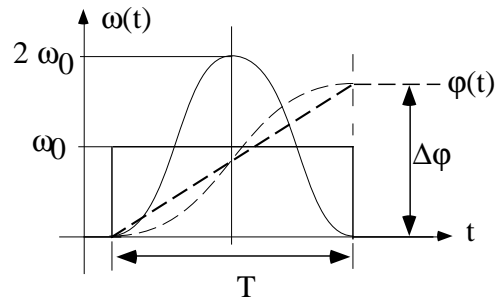
Begrenzung durch Lenkbeschleunigung

$$d\omega/dt = \omega_0 \cdot \frac{2\pi}{T} \cdot \sin \frac{2\pi}{T} \cdot t \quad \text{maximal } \omega' = \omega_0 \cdot \frac{2\pi}{T}$$

$$\Delta\phi = \int_0^T \omega_0 \cdot \left(1 - \cos \frac{2\pi}{T} \cdot t\right) dt$$

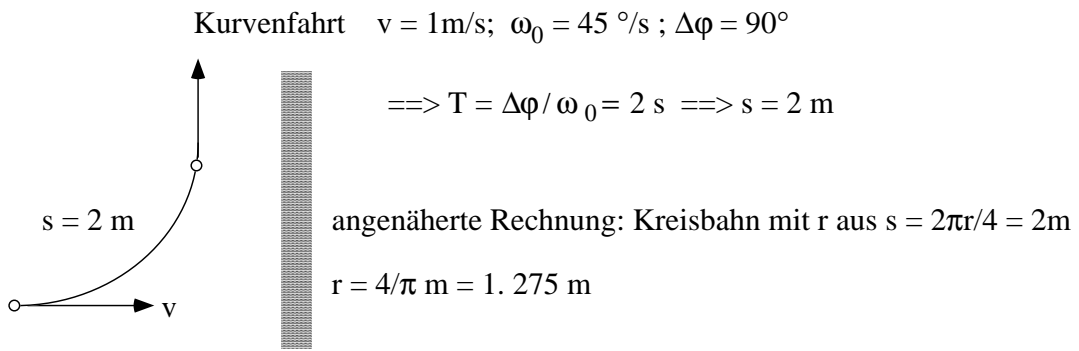
$$= \omega_0 \cdot T \implies \omega_0^2 = \Delta\phi \cdot \omega' / 2\pi;$$

$$T^2 = 2\pi \Delta\phi / \omega'$$

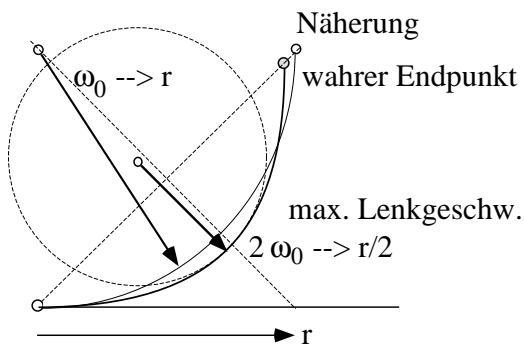


Bei gegebener maximaler Lenkbeschleunigung ω' und zu fahrendem Winkel $\Delta\phi$ erhält man so ω_0 und die Zeit T , die mit $\omega(t)$ zu lenken ist. Dann ist der AMR um die Kurve gefahren. Ebenfalls erhält man die gefahrene Strecke $s = v \cdot T$. Allerdings nicht den Endpunkt der Strecke nach T . Für Winkel $\Delta\phi < 90^\circ$ ist eine brauchbare erste Näherung ein Kreis mit $r = s / \Delta\phi$, der durch Start- und Endpunkt der Kurvenfahrt geht.

Als Beispiel soll eine 90° -Kurve gefahren werden. Sie dient zum Ausweichen an einem Hindernis. Bei gegebener Geschwindigkeit v und Lenkgeschwindigkeit ω_0 werden 90° in $T = \Delta\phi / \omega_0$ erreicht und der AMR ist eine Strecke $s = v \cdot T$ gefahren.



genauere Kurvenform



bei Einleiten der Ausweichbewegung

fährt der AMR noch annähernd

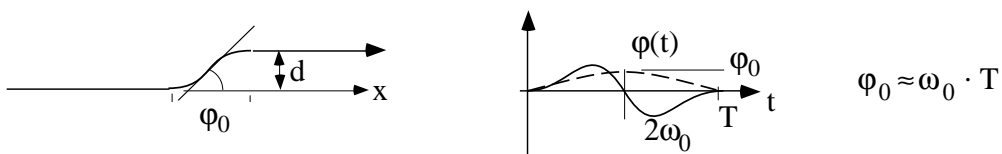
$r = 2/\pi \cdot v \cdot 90^\circ / \omega_0\text{ m}$

auf das Hindernis zu

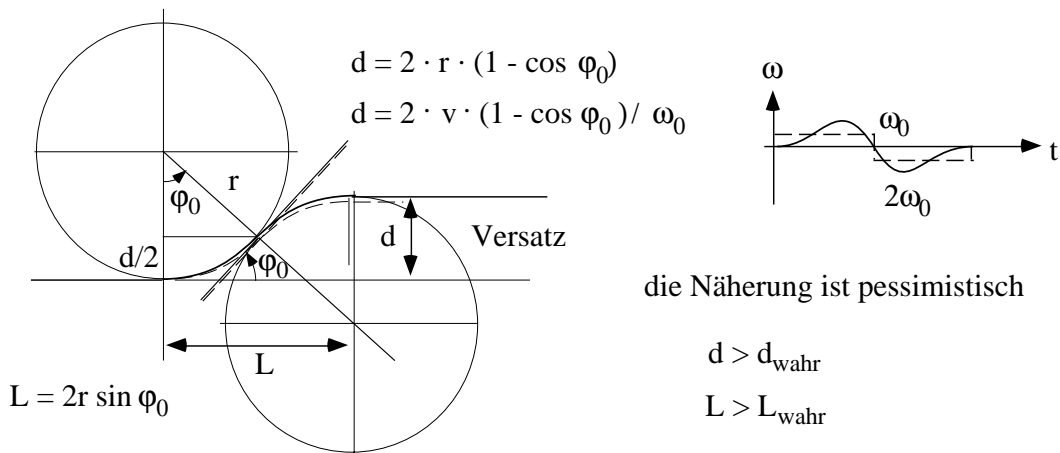
Die Kurve wird durch einen Kreis angenähert, dessen Viertelumfang gerade s ist:

$r = 2 \cdot s / \pi$

3.3.5. Basisfähigkeit "Fahrt einer S-Kurve"

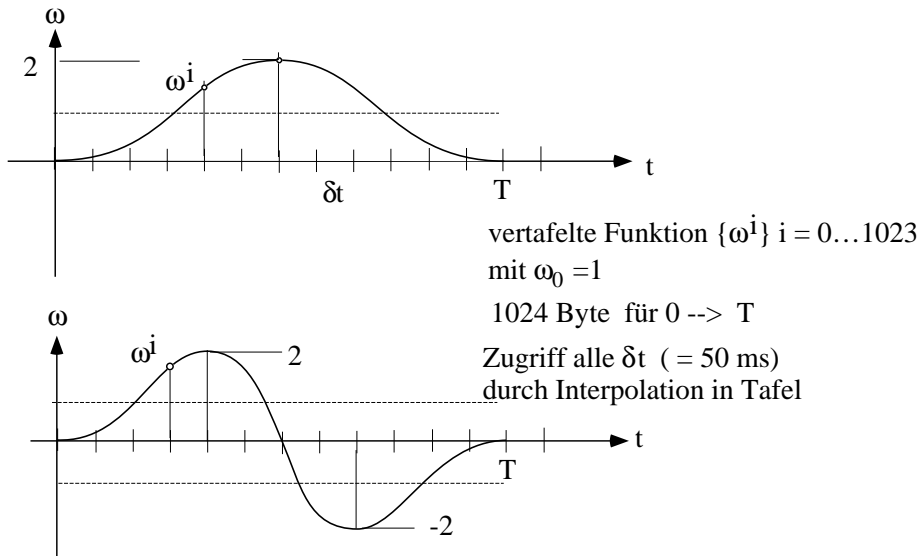


Die S-Kurve wird durch den Versatz d und den Winkel ϕ_0 im Wendepunkt gekennzeichnet.



Die Lenkbewegung $\omega(t)$ ist antisymmetrisch um die Stelle, an der $\varphi(t) = \varphi_0$ ist. Zur Bestimmung von d wird eine Näherung mit zwei Kreisen betrachtet. Sie haben den Radius $r = v / \omega_0$. Die Lenkfunktion $\omega(t)$ mit Amplitude $2\omega_0$ erzeugt eine Bahn sehr ähnlich der Näherung.

Die Funktion $\omega(t)$ für die Kurvenfahrt oder die Fahrt einer S-Kurve wird vertafelt. Die Tabellenwerte werden mit ω_0 multipliziert und so an den Piloten gegeben, dass für die Zeit T die Tabelle durchfahren wird. Wenn der Pilot alle δt einen neuen Wert braucht, dann muss in der Tabelle an den Stellen $\Delta x = 1024 \cdot \delta t / T$ zugegriffen werden.



3.3.6. Basisfähigkeit "Türdurchfahrt"

Der AMR lernt diese Fähigkeit durch Vormachen: Er wird einige wenige Male von Hand durch eine Tür geschickt, merkt sich dabei die jeweils gesehenen Sensoreindrücke zusammen mit den kommandierten Aktionen, baut sie in ein neuronales Netz ein und ist dann in der Lage, selbst durch eine Tür zu fahren.

Lernen: Aufbau eines "growing neural gas nets"

- taught trajectories
- replayed trajectories

Knoten: Sensorsituation, Position, Orientierung

und Aktion (v, ω)

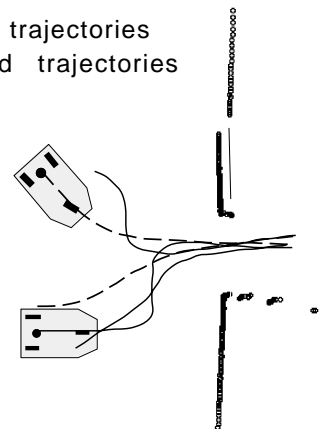
Kanten: verbinden aufeinanderfolgende

Schritte einer Fahrt

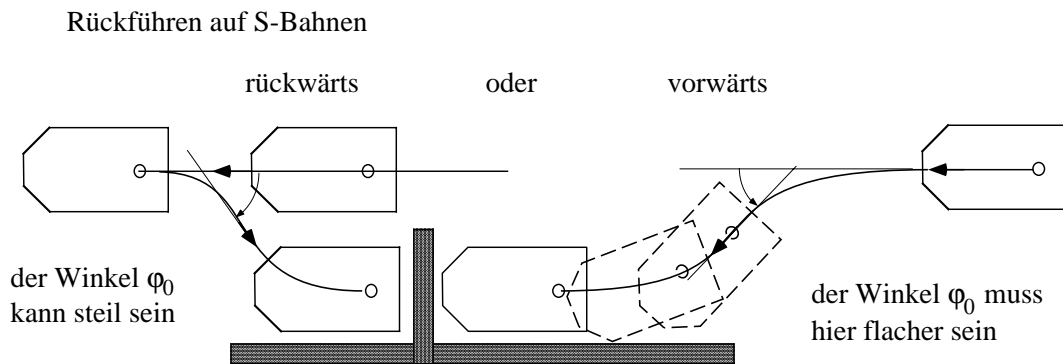
Abspielen: finden der ähnlichsten Knoten

zu gegebener Position und Sensorsituation

und Interpolation der Aktionen zur auszuführenden Aktion



3.3.7. Andockmanöver



3.3.8. Basisfähigkeit "Wandfolgen"

In den folgenden Bildern wird gezeigt, wie ein AMR einer Wand folgt, wobei nur eine Sektorkarte benutzt wird. Der AMR sucht eine freie Passage mit einem Mindestabstand zur rechten Wand.

