1 Berechnung der ct-bot Bewegung (ct-sim)

Die Berechnung der Roboterbewegung basiert auf Abbildung 1.1. Winkel werden im folgenden im Bogenmaß angegeben. Der bot beginnt seine Bewegung bei Position P_0 in Richung v_0. Dabei legt das linke bzw. rechte Rad jeweils die Distanz s_l bzw. s_r zurück. Die neue Position ergibt sich effektiv als Bewegung entlang $\hat{v}$ um $\hat{s}$.

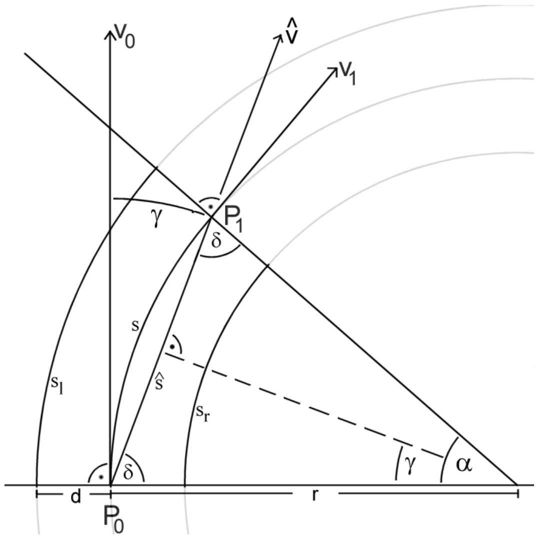

Abbildung 1.1: Bewegung des Roboters von P_0 nach P_1 entlang einer Kreisförmigen Bahn

Zuerst wird der Winkel α berechnet. Für die Längen der Kreissegmente s_l und s_r gilt:

$$s_l = \alpha * (r + d) \quad (1.1)$$

$$s_r = \alpha * (r - d) \quad (1.2)$$

Durch lösen dieses Gleichungssystems erhält man:

$$\alpha = \frac{s_l - s_r}{2d} \quad (1.3)$$

$$r = d\frac{s_l + s_r}{s_l - s_r} \tag{1.4}$$

Damit können wir bereits die Richtung v_1 des bot bestimmen. Diese ergibt sich durch Drehung des vektors v_0 um α (Siehe 1.1). Für die Positionsbestimmung benötigt man noch $\hat{v}$ und $\hat{s}$. $\hat{v}$ ergibt sich durch Drehung von v_0 um γ. Es gilt offensichtlich:

$$\gamma = \frac{\pi}{2} - \delta \tag{1.5}$$

$$2 * \delta + \alpha = \pi \tag{1.6}$$

daraus ergibt sich:

$$\gamma = \frac{\alpha}{2} = \frac{s_l - s_r}{4d} \tag{1.7}$$

Nun fehlt nur noch die zurückgelegte Distanz $\hat{s}$. Man sieht dass

$$\frac{\frac{\hat{s}}{2}}{r} = \sin\gamma \tag{1.8}$$

$$\Rightarrow \hat{s} = 2r\sin\gamma \tag{1.9}$$

Durch einsetzen von Gleichung 1.4 erhält man:

$$\hat{s} = 2d\frac{s_l + s_r}{s_l - s_r}\sin\gamma \tag{1.10}$$

Aus Gleichung 1.7 ergibt sich:

$$s_l - s_r = 4d\gamma \tag{1.11}$$

Setzen wir dies in die vorhergehende Gleichung ein erhält man:

$$\hat{s} = \frac{s_l + s_r}{2}\frac{\sin\gamma}{\gamma} \tag{1.12}$$

Offensichtlich ist $\hat{s}$ für $\gamma = 0$ (dies entspricht einer Bewegung geradeaus) nicht wohldefiniert. Aus diesem Grund muss dieser Fall gesondert behandelt werden. Da

$$\lim_{x\to 0}\frac{\sin x}{x} = 0 \tag{1.13}$$

erhält man als Grenzwert

$$\lim_{\gamma \to 0} \hat{s} = \frac{s_l + s_r}{2} \tag{1.14}$$

Für $\gamma = 0$ ist $s_l = s_r$ und folglich

$$\hat{s} = s_l = s_r \tag{1.15}$$

1.1 Notizen zur Drehung der Richtungsvektoren

Die Drehung eines Vektors v um einen Winkel α läßt sich einfach berechnen durch Multiplikation mit der entsprechenden Rotationsmatrix R:

$$R = \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix} \tag{1.16}$$

$$v' = R \cdot v \tag{1.17}$$

2 Der Maussensor (ct-sim)

Abbildung 2.1 illustriert einen Schritt der Roboterbewegung (vergleiche Abb. 1.1). M_0 stellt dabei die Position des Maussensors vor der Bewegung, M_1 die Position nach der Bewegung dar. Der Maussensor registriert bei diesem Schritt eine Bewegung in Richtung v_m um die Distanz l_m. Dabei ist v_m die Tangente in M_0 und somit orthogonal zur Strecke t.

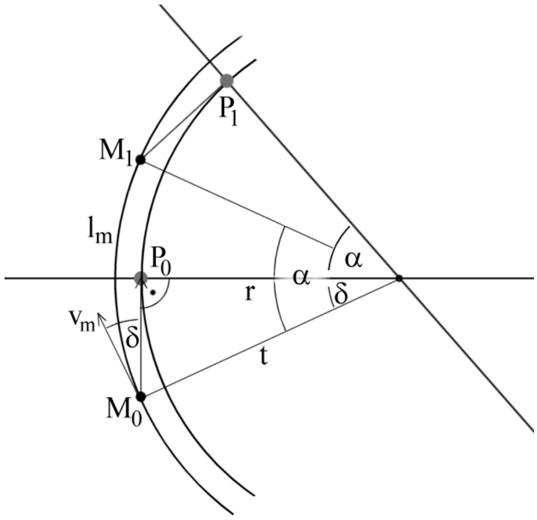

Abbildung 2.1: Bewegung des Roboters von P_0 nach P_1 entlang einer Kreisförmigen Bahn

P_0 und M_0 sind gegeben, die Werte α und r wurden bereits für die Bot-Bewegung berechnet. Da nur die relative Ausrichtung von v_m zu $\vec{M_0P_0}$ benötigt wird, berechnen wir den Winkel δ:

$$\delta = \arctan \frac{r}{|\vec{M_0P_0}|} \tag{2.1}$$

Die zurückgelegte Strecke l_m ergibt sich wieder als Bogenlänge, also:

$$l_m = \alpha * t \tag{2.2}$$

Die Länge von t ergibt sich wiederum als

$$t = \sqrt{|\vec{M_0P_0}|^2 + r^2} \tag{2.3}$$

Nun zerlegen wir noch die Daten in X und Y Komponente der Sensorwerte:

$$X = l_m \sin \delta = \alpha t * \sin(\arctan \frac{r}{|\vec{M_0P_0}|}) \tag{2.4}$$

$$= \alpha t \frac{|\vec{M_0P_0}|}{t} = \alpha |\vec{M_0P_0}| \tag{2.5}$$

$$Y = l_m \cos \delta = \alpha t * \cos(\arctan \frac{r}{|\vec{M_0P_0}|}) \tag{2.6}$$

$$= \alpha t \frac{r}{t} = \alpha r \tag{2.7}$$

Für die Y-Richtung muss noch der Sonderfall einer Geradeausbewegung bedacht werden. In diesem Fall geht α gegen 0 und r gegen unendlich. Wir wissen jedoch bereits aus der Bewegungsberechnung, dass der Grenzwert in diesem Fall der zurückgelegten Distanz eines beliebigen Rades entspricht. Also gilt für $\alpha = 0$

$$Y = s_l = s_r \tag{2.8}$$

2.1 Berechnen der Bot-Bewegung aus den Maussensordaten

Sieht man sich die Berechnung der Maussensordaten an erkennt man leicht, wie man daraus die zugrundeliegende Bewegung erkennt. Gegeben seien die Maussensordaten X und Y. Dann gilt:

$$\alpha = \frac{X}{|\vec{M_0P_0}|} \tag{2.9}$$

Ist $\alpha = 0$ haben wir eine Bewegung geradeaus um die Distanz Y. Andernfalls gilt:

$$r = \frac{Y}{\alpha} \tag{2.10}$$

Indem man diese Werte in den Gleichungen 1.7ff. einsetzt erhält man die Positionsveränderung des Bot.