Formale Modellierung
Vorlesung vom 14.05.12: Der sichere Roboter in Z

Till Mossakowski & Christoph Lüth

Universität Bremen

Sommersemester 2012

Heute im Programm

- Die Z Notation
- Modellierung des Sicheren Omnidirektionalen Roboters

Die Z Notation

- Basiert auf getypter Mengenlehre
- Entwickelt seit 1980 (Jean-Claude Abrial, Oxford PRG)
- Industriell genutzt (IBM, Altran Praxis (vorm. Praxis Critical Systems))
- LaTeX-Notation und Werkzeugunterstützung (Community Z Tools, HOL-Z)

Modell

- Bremszeit und Bremsstrecke (ab Bremszeitpunkt, $a_{brk} > 0$):

$$v(t) = v_0 - a_{brk}\,t \qquad s(t) = v_0 t - \frac{a_{brk}}{2} t^2 \qquad \Longrightarrow \qquad T = \frac{v_0}{a_{brk}} \qquad S = \frac{v_0^2}{2a_{brk}}$$

- Modellierung in Z: Berechung des Bremsweges. Verzögerung t_L und Bremsverzögerung a_{brk} fest:

$$a_{brk}, t_L : \mathbb{Z}$$

Damit Brems- und Anhalteweg als Funktion über v:

$$\begin{array}{l} brk : \mathbb{Z} \rightarrow \mathbb{Z} \\ stop : \mathbb{Z} \rightarrow \mathbb{Z} \\ \hline \forall\, v : \mathbb{Z} \bullet brk\ v = v * v \operatorname{div} 2 * a_{brk} \\ \forall\, v : \mathbb{Z} \bullet stop\ v = v * t_L + brk(v) \end{array}$$

Geometrie: Punkte

- Ein Punkt ist ein Schema mit Komponenten x und y:

$$\begin{array}{l} POINT \\ \hline x, y : \mathbb{Z} \\ \hline \end{array}$$

- Dazu einschlägige Operationen: Konstruktion aus Polarkoordinaten, Skalarmultiplikation:

$$\begin{array}{l} polar : \mathbb{Z} \times \mathbb{Z} \rightarrow POINT \\ \hline \forall\, d, \omega : \mathbb{Z} \bullet (polar(d, \omega)).x = d * cos\,\omega \\ \forall\, d, \omega : \mathbb{Z} \bullet (polar(d, \omega)).y = d * sin\,\omega \end{array}$$

$$\begin{array}{l} smult : \mathbb{Z} \times POINT \rightarrow POINT \\ \hline \forall\, d : \mathbb{Z};\ p : POINT \bullet (smult(d, p)).x = d * p.x \\ \forall\, d : \mathbb{Z};\ p : POINT \bullet (smult(d, p)).y = d * p.y \end{array}$$

Mehr Standardoperationen auf Punkten

- Addition und Subtraktion von Punkten:

$$\begin{array}{l} add : POINT \times POINT \rightarrow POINT \\ minus : POINT \times POINT \rightarrow POINT \\ \hline \forall\, p, q : POINT \bullet (add(p, q)).x = p.x + q.x \\ \forall\, p, q : POINT \bullet (add(p, q)).y = p.y + q.y \\ \forall\, p, q : POINT \bullet minus(p, q) = add(p, smult(0 - 1, q)) \end{array}$$

- Ganzzahlige Wurzel:

$$\begin{array}{l} sqrt : \mathbb{Z} \rightarrow \mathbb{Z} \\ \hline \forall\, i : \mathbb{Z} \bullet sqrt\ i * sqrt\ i \leq i \wedge i < (sqrt\ i + 1) * (sqrt\ i + 1) \end{array}$$

- Damit Betrag eines Punktes (als Vektor):

$$\begin{array}{l} len : POINT \rightarrow \mathbb{Z} \\ \hline \forall\, p : POINT \bullet len\ p = sqrt(p.x * p.x + p.y * p.y) \end{array}$$

Geometrie: Bewegung

- Bewegung: kombinierte Translation und Rotation

$$\begin{array}{l} movep : POINT \times \mathbb{Z} \times \mathbb{Z} \rightarrow POINT \\ \hline \forall\, p : POINT;\ d, \omega : \mathbb{Z} \bullet \\ \qquad movep(p, d, \omega) = add(p, polar(d, \omega)) \end{array}$$

- Bewegung eines Polygons:

$$\begin{array}{l} move : \operatorname{seq} POINT \times \mathbb{Z} \times \mathbb{Z} \rightarrow \operatorname{seq} POINT \\ \hline \forall\, p : \operatorname{seq} POINT;\ d, \omega : \mathbb{Z} \bullet \\ \qquad move(p, d, \omega) = (\lambda\, i : \operatorname{dom} p \bullet movep(p(i), d, \omega)) \end{array}$$

Geometrie: Konvexe Hülle

- Strecke zwischen zwei Punkten:

$$\begin{array}{l} seg : POINT \times POINT \rightarrow \mathbb{P}\, POINT \\ \hline \forall\, p, q : POINT \bullet seg(p, q) = \\ \quad \{r : POINT \mid \exists\, z : \mathbb{N} \bullet z \leq len(minus(p, q)) \wedge \\ \qquad\qquad r = add(smult(z, p), smult(len(minus(p, q)) - z, q))\} \end{array}$$

- Konvexität (Menge aller konvexen Punktmengen):

$$\begin{array}{l} conv : \mathbb{P}(\mathbb{P}\, POINT) \\ \hline conv = \\ \quad \{c : \mathbb{P}\, POINT \mid \forall\, p, q : c;\ r : POINT \bullet r \in seg(p, q) \Rightarrow r \in c\} \end{array}$$

- Konvexe Hülle: kleinste p umfassende konvexe Menge

$$\begin{array}{l} convhull : \mathbb{P}\, POINT \rightarrow \mathbb{P}\, POINT \\ \hline \forall\, p : \mathbb{P}\, POINT \bullet convhull\ p = \bigcap \{q : \mathbb{P}\, POINT \mid q \in conv \wedge p \subseteq q\} \end{array}$$

Der Roboter und seine Bremsfläche

- Der Roboter hat eine Kontour, Geschwindigkeit, Orientierung:

$$\begin{array}{|l} \textit{Robot} \\ \hline P : \operatorname{seq} POINT \\ v : \mathbb{Z} \\ \omega : \mathbb{Z} \\ \hline \# P > 2 \\ \operatorname{ran} P \in conv \end{array}$$

- Die beim Anhalten überstrichene Fläche:

$$\begin{array}{|l} area : \operatorname{seq} POINT \times \mathbb{Z} \times \mathbb{Z} \to \mathbb{P}\, POINT \\ \hline \forall\, v, \omega : \mathbb{Z};\ P : \operatorname{seq} POINT \bullet \\ \quad area(P, v, \omega) = convhull\,(\operatorname{ran} P \cup \operatorname{ran}(move(P, stop\; v, \omega))) \end{array}$$

Der sichere Roboter

- Menge von Hindernissen

$$\begin{array}{|l} obs : \mathbb{P}\, POINT \end{array}$$

- Der sichere Roboter

$$\begin{array}{|l} \textit{SafeRobot} \\ \hline Robot \\ \hline area(P, v, \omega) \cap obs = \emptyset \end{array}$$

- Sicherheitseigenschaft:

$$\begin{array}{|l} \textit{SafeMove} \\ \hline \Delta Robot \\ \hline SafeRobot \Rightarrow SafeRobot' \end{array}$$

Bewegung des Roboters

- Geschwindigkeit bleibt gleich:

$$\begin{array}{|l} \textit{RobotSameSpeed} \\ \hline \Delta Robot \\ \hline v' = v \\ \omega' = \omega \\ P' = move(P, v, \omega) \end{array}$$

- Sicher?

$$\begin{array}{|l} \textit{SafeSameSpeed} \\ \hline RobotSameSpeed \\ \hline SafeRobot \Rightarrow SafeRobot' \end{array}$$

Bewegung des Roboters

- Geschwindigkeit erhöht sich, Lenkwinkel bleibt nur in Bewegung:

$$\begin{array}{|l} \delta v : \mathbb{Z} \end{array}$$

$$\begin{array}{|l} \textit{RobotSpeedUp} \\ \hline \Delta Robot \\ \hline v' \leq v + \delta v \\ v \neq 0 \Rightarrow \omega' = \omega \\ P' = move(P, v, \omega) \end{array}$$

- Sicher?

$$\begin{array}{|l} \textit{SafeSpeedUp} \\ \hline RobotSpeedUp \\ \hline SafeRobot \Rightarrow SafeRobot' \end{array}$$

Zusammenfassung

- Z ist leichtgewichtige Notation
- Werkzeugunterstützung: LaTeX, Typcheck
- Nächste Woche: UML