

## Roboterkinematik

### Aufgabe 1

Realisieren Sie die folgenden beiden kinematischen Grundfertigkeiten des Roboters:

- `curveDrive(v, r,  $\Delta\theta$ )`
- `straightDrive(v, l)`

`curveDrive` bewegt den Roboter solange auf einem Kreisbogen mit Radius  $r$  und Geschwindigkeit  $v$ , bis sich seine Orientierung um  $\Delta\theta$  geändert hat. `straightDrive` bewegt den Roboter eine gerade Strecke der Länge  $l$  mit der Geschwindigkeit  $v$ . Definieren Sie eine Folge von Geschwindigkeitsbefehlen, die die gewünschte Bewegung umsetzt. Beachten Sie dabei, dass Geschwindigkeit und Rotationsgeschwindigkeit beim Roboter begrenzt sind.

Gehen Sie zunächst davon aus, dass der Roboter die gewünschten Geschwindigkeiten korrekt umsetzt (`motionNoise` auf 0 setzen) und realisieren Sie Kreis- und Rechteckfahrten und einen Fahrspurwechsel.

Gehen Sie nun davon aus, dass die gewünschten Geschwindigkeiten nicht korrekt umgesetzt werden (`motionNoise` werden auf die voreingestellten Werte gesetzt). Was beobachten Sie?

### Aufgabe 2

Die tatsächliche Position des Roboters kann mit `getTrueRobotPose()` abgefragt werden. Die Funktionen aus Aufgabe 1 sollen nun mit Hilfe dieser Funktion umgesetzt werden.

### Aufgabe 3

Realisieren Sie einen Linienverfolger `followLine(p1, p2)`, der eine Strecke von Punkt  $p1$  nach Punkt  $p2$  möglichst genau verfolgt. Realisieren Sie zunächst einen P-Regler, der den Abstand zur Linie auf 0 regelt. Was beobachten Sie? Verbessern Sie Ihre Regelung, indem Sie einen PD-Regler einsetzen.

### Aufgabe 4

Realisieren Sie eine Steuerung `goto(v, p, tol)`, die den Roboter auf den Punkt  $p$  mit der Geschwindigkeit  $v$  zusteuert, wobei der Punkt  $p$  lediglich mit einer gewissen Toleranz  $tol$  erreicht werden muss.

### Aufgabe 5

Realisieren Sie einen Linienverfolger `followPolyline(v, poly)`, der einen Polygonzug  $poly$  mit der Geschwindigkeit  $v$  abfährt.

Implementieren Sie zwei Verfahren:

- der Roboter stoppt bei jedem Eckpunkt des Polygonzugs, richtet sich aus und fährt erst dann auf den nächsten Eckpunkt zu
- die Geschwindigkeit  $v$  wird möglichst konstant gehalten. Sobald der Roboter einen Eckpunkt mit einer gewissen Toleranz erreicht hat, fährt er bereits auf den nächsten Eckpunkt zu.

Testen Sie Ihren Linienverfolger in typischen Szenarien.

