

RoboCode

Funktionalitäten des Roboters

Radarbewegungen



Christian Riess

Eva Eibenberger

**Pattern Recognition Lab (Computer Science Dep. 5)
Friedrich-Alexander-University Erlangen-Nuremberg**



Beispiel-Überlegungen

Überlegungen: Wie verhält sich der Roboter am besten wenn...

- ... er sich ganz allgemein fortbewegt?
 - Wie schaut eine gute Bewegungsbahn aus?
Stehen bleiben oder stetig bewegen?
 - Wie werden Radar und Kanone ausgerichtet?
- ... ein Gegner gefunden wurde?
 - Feuert er direkt eine Kugel ab?
 - In welche Richtung feuert er den Schuss am besten?
 - Mit welcher Schussenergie sollte er schießen?
 - Soll Radarbewegung eingeschränkt werden, um Gegner zu folgen?
 - Soll die Bewegungsrichtung des Roboters angepasst werden?
- ... er von einer gegnerischen Kugel getroffen wurde?
 - In welche Richtung flüchtet er am besten?
Einfach nach hinten? Senkrecht zur Kugelbahn?
- ... er einen Gegner mit einer Kugel trifft?
 - Soll die Radarbewegung eingeschränkt werden?
- ... es zu einer Kollision mit der Wand kommt?
 - In welche Richtung fährt er am besten weiter?
 - Kann die Kollision generell vermieden werden?

Ein paar Details auf den
folgenden Folien zu:



Beispiel-Strategie 2

- Bewegung des Radars und der Kanone
 - Szenario: Der Roboter bewegt sich und sucht gegnerischen Roboter
 - Fragen:
 - Wie sollte der Radar bewegt werden?
 - Wie und wann sollte die Kanone bewegt werden?
 - Hintergrund:
 - Drehen des Radar: schnell (45° pro Zeiteinheit)
 - Drehen der Kanone: langsam (20° pro Zeiteinheit)
 - Drehen des Körpers: extrem langsam
 - Idee:
 - Mit Radar stetig im Kreis scannen
 - Falls Gegner gesichtet wurde, auf kürzestem Weg Kanone auf Gegner richten und schießen



Beispiel-Strategie

■ Bewegung des Radars und der Kanone

- Szenario: Der Roboter bewegt sich und sucht gegnerischen Roboter
- Idee: Mit Radar stetig im Kreis scannen, bei Sichten des Gegners Kanone auf kürzesten Weg drehen

■ Vorbereitung:

- Sicherstellen, dass sich Radar, Kanone und Körper unabhängig voneinander drehen
- Radar sollte sich immer drehen

■ Fragen:

- Wann sollte die Kanone bewegt werden?
- In welche Richtung muss die Kanone gedreht werden?

```

...
public void run() {
    // unabh. Kanonen, Radar- und Körperbewegung
    setAdustGunForRobotTurn(true);
    setAdustRadarForRobotTurn(true);
    setAdustRadarForGunTurn(true);

    while(true) {
        turnRadarRight(360);
        turnRight(20);
        ahead(30);
    }
}
...
public void onScannedRobot(ScannedRobotEvent e) {
    fire(1);
}
...

```



Beispiel-Strategie

■ Bewegung des Radars und der Kanone

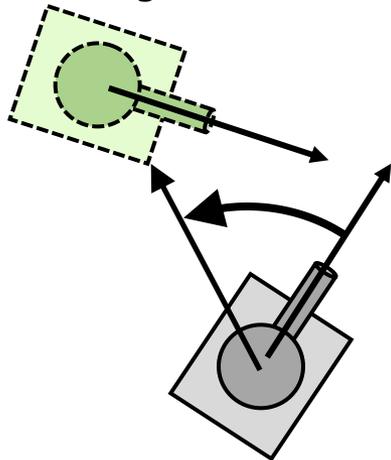
■ Idee:

- Um schneller einen Gegner zu finden: Rotieren des Scanners anstelle der Kanone
- Wenn Gegner gefunden wurde: Auf kürzestem Weg Kanone auf Gegner richten

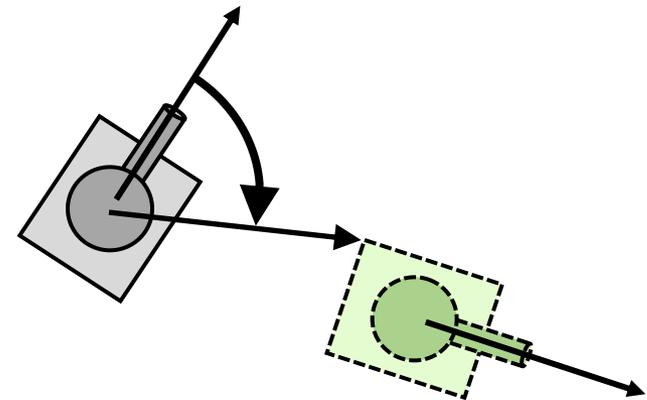
■ Überlegungen:

- In welche Methode gehört das Drehen des Scanners?
Wohin das Ausrichten der Kanone, wenn Gegner gefunden wurde?
- Wie soll sich die Kanone drehen?

Drehen nach **links**,
wenn Gegner links von Kanone



Drehen nach **rechts**,
wenn Gegner rechts von Kanone





Beispiel-Strategie

■ Bewegung des Radars und der Kanone

■ Implementierungshilfe:

■ Hilfreiche Informationen z.B.:

```
double hBody = getHeading();
double hGun = getGunHeading();
double bEnemy = e.getBearing();
```

■ Gesucht:

Winkel a zwischen Kanone und Gegner

■ Idee: Berechne als Zwischenergebnis den **Winkel d** zwischen Roboterausrichtung und Kanone

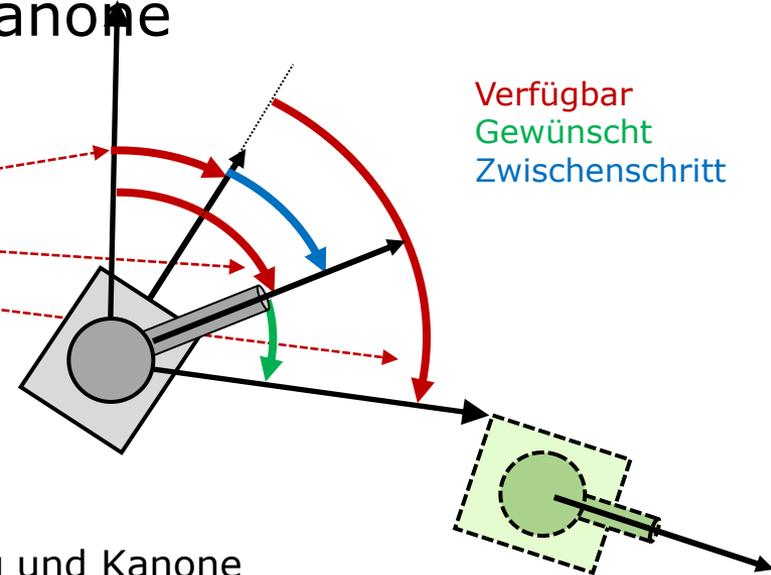
■ Daher:

1. Berechnung von d
2. Berechnung von a

■ Gut zu wissen:

Um einen Winkel nach links drehen ist gleichbedeutend mit einer negativen Drehung nach rechts:

```
turnGunLeft (winkel) = turnGunRight (-1*winkel)
```





Beispiel-Strategie

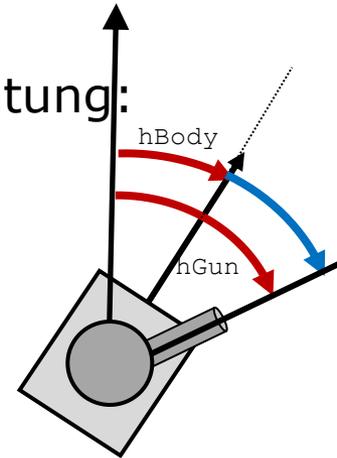
■ Bewegung des Radars und der Kanone

■ Berechnung des Winkels der Kanone zur Roboterorientierung:

■ Gegeben:

```
double hBody = getHeading();
double hGun = getGunHeading();
```

- Es müssen verschiedene Fälle unterschieden werden:
Überlege dir für jeden Fall, welchen Wertebereich der Winkel $d = hGun - hBody$ annimmt.



Beispielüberlegung:

Angenommen: Absolute Ausrichtung des Roboters ist
im 1. Quadranten: $0^\circ \leq hBody \leq 90^\circ$

Frage: Welchen Wertebereich hat der Winkel d ,
falls die Kanone in bestimmte Richtungen zeigt

Lösung: Überlegen und alle Szenarien durchspielen

Ergebnis:

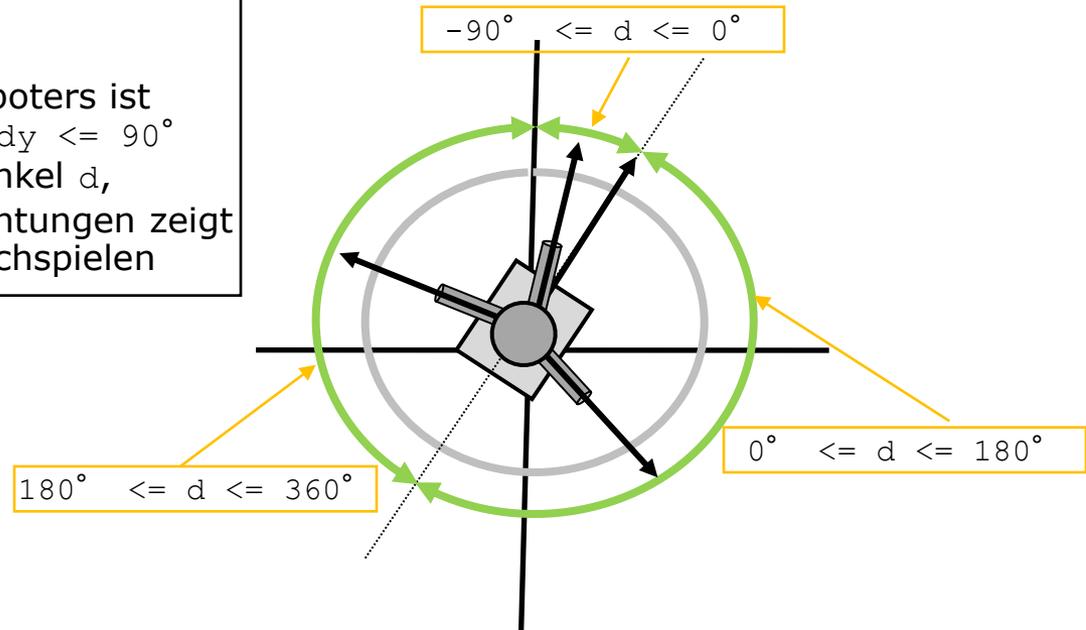
Kanone rechts von Roboter-Orientierung falls:

$$0^\circ \leq d \leq 180^\circ$$

Kanone links von Roboter-Orientierung falls:

$$-90^\circ \leq d \leq 0^\circ \quad \text{oder}$$

$$180^\circ \leq d \leq 360^\circ$$

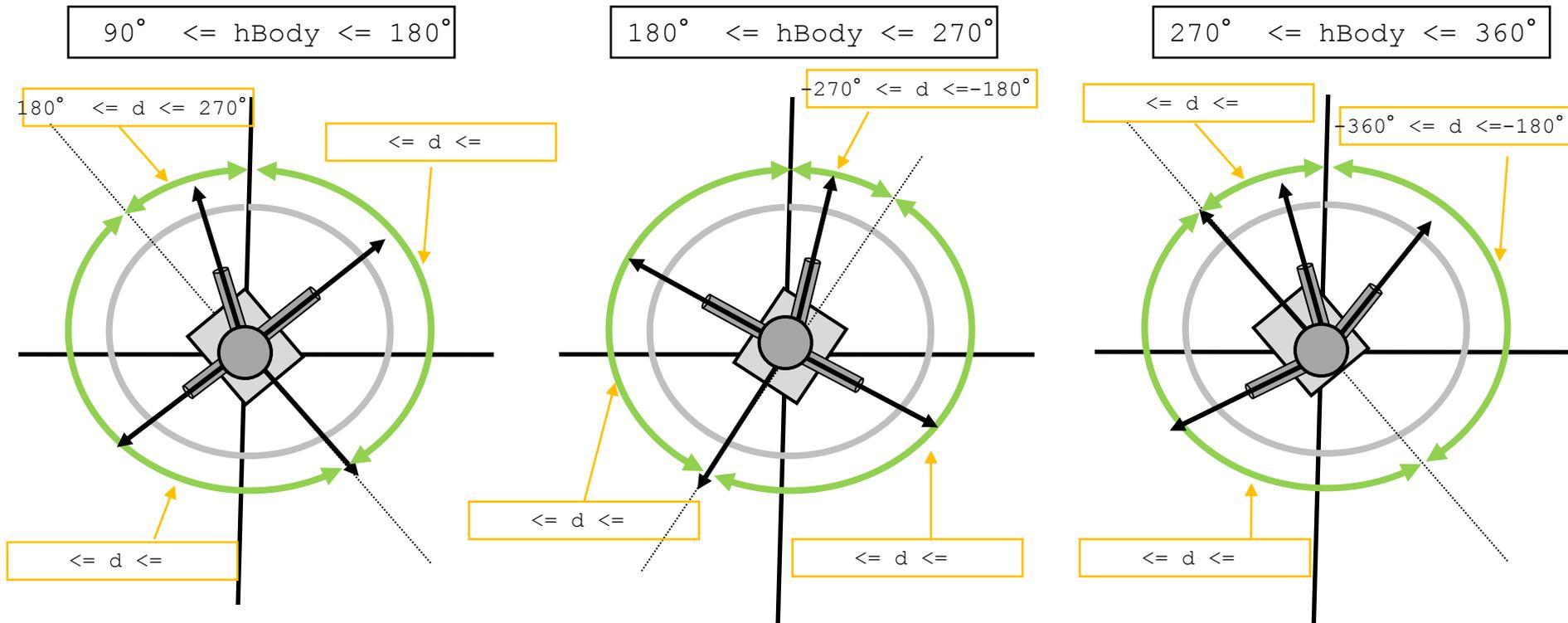




Beispiel-Strategie

■ Bewegung des Radars und der Kanone

- Gleiche Überlegungen für die Fälle: Absolute Ausrichtung des Roboters ist im 2., 3. oder 4. Quadranten





Beispiel-Strategie

■ Bewegung des Radars und der Kanone

- Kombiniert man alle 4 Fallunterscheidungen, ergibt sich:

Falls $hBody \leq 180^\circ$: Kanone **rechts** von Roboter-Orientierung falls:

$0^\circ \leq d \leq 180^\circ$	(1. Fall)
$0^\circ \leq d \leq 180^\circ$	(2. Fall)

Kanone **links** von Roboter-Orientierung falls:

$-90^\circ \leq d \leq 0^\circ$ oder $180^\circ \leq d \leq 360^\circ$	(1. Fall)
$-180^\circ \leq d \leq 0^\circ$ oder $180^\circ \leq d \leq 270^\circ$	(2. Fall)

Falls $hBody > 180^\circ$: Kanone **rechts** von Roboter-Orientierung falls:

$-270^\circ \leq d \leq -180^\circ$ oder $0^\circ \leq d \leq 180^\circ$	(3. Fall)
$-360^\circ \leq d \leq -180^\circ$ oder $0^\circ \leq d \leq 90^\circ$	(4. Fall)

Kanone **links** von Roboter-Orientierung falls:

$-180^\circ \leq d \leq 0^\circ$	(3. Fall)
$-180^\circ \leq d \leq 0^\circ$	(4. Fall)

- Daher: Zu Wissen, ob Kanone links/rechts ist, reicht aus:

Falls $hBody \leq 180^\circ$:

Kanone **rechts** von Roboter-Orientierung falls: $0^\circ \leq d \leq 180^\circ$
 Sonst ist sie **links** (Finales d : falls $d > 0^\circ$ muss $d = d - 360^\circ$ gerechnet werden).

Falls $hBody > 180^\circ$:

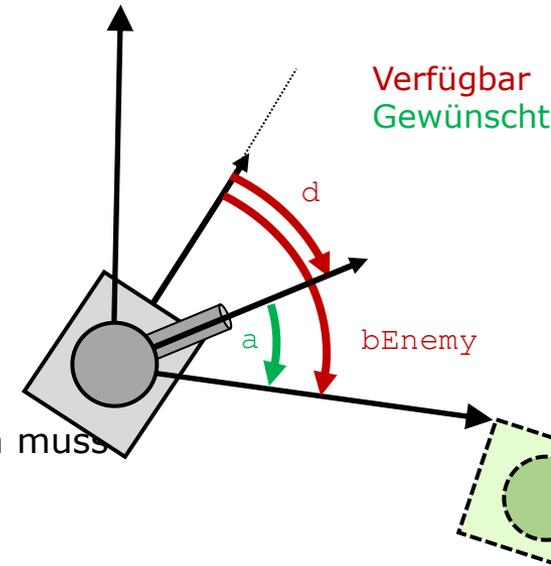
Kanone **links** von Roboter-Orientierung falls: $-180^\circ \leq d \leq 0^\circ$
 Sonst ist sie **rechts** (Finales d : falls $d < 0^\circ$ muss $d = d + 360^\circ$ gerechnet werden).



Beispiel-Strategie

■ Bewegung des Radars und der Kanone

- Gesucht: Winkel a zwischen Kanone und Gegner
- Gegeben:
 - Rel. Winkel d zw. Kanone und Roboter-Orientierung
 - Rel. Winkel b_{Enemy} zw. Gegner und Roboter-O.
- Berechne die Winkeldifferenz $a = b_{Enemy} - d$
 - Gibt genau an, um wie viel Grad die Kanone gedreht werden muss
 - Winkel d Werte zwischen -360° und $+360^\circ$ annehmen
- Kanone soll auf kürzesten Weg gedreht werden
 - Daher ist maximaler Drehwinkel in beide Richtungen 180°
 - Überlegungen:
 - `setTurnGunLeft(winkel) == setTurnGunRight(-winkel)`
 - Drehung um 200° nach rechts entspricht Drehung um 160° nach links



■ Daher:

```
double a = e.getBearing() - d;
if (a < -180) {
    a += 360;
} else if (a > 180) {
    a -= 360;
}
turnGunRight(a);
```