5. Objektorientierte Programmierung IAS

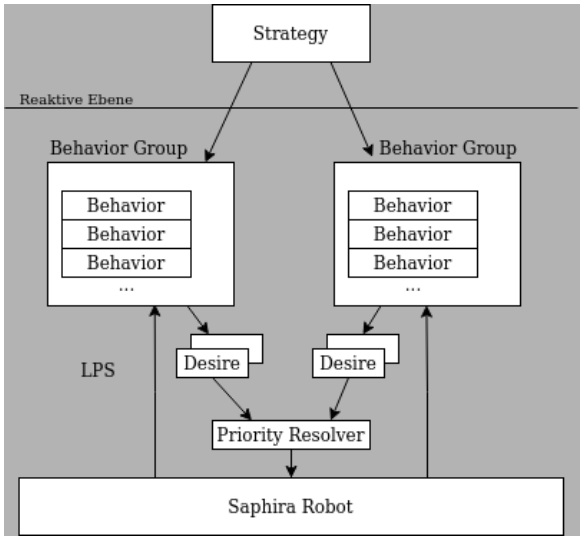
5.1 Programmierprinzip

Grundlage: THI RobCon mit Saphira-Architektur

Proaktive Ebene: Endlicher Automat

Reaktive Ebene: Verhaltensmuster mit Resolvierung

Basisklassen:



Basisklassen

Ausgewählte Methoden Basisklassen

Saphira Robot

Methode	Beschreibung
`Pose getPose()`	Positionskoordinaten Fahrzeug (x,y,th)
`double getTransVel()` `double getRotVel()`	aktuelle Geschwindigkeiten Fahrzeug
`int getSonarRange(int n)`	Entfernungsmessung Sonar n in mm bezogen auf Fahrzeugmittelpunkt, maximal 3 m
`int getRadius()`	Abfrage Roboterradius

Behaviour

Virtuelle Oberklasse Verhaltensmuster mit abstrakter Methode fire(), wird vom Resolver zykklisch alle 100ms aufgerufen

Erzeugen Instanzen von Unterklassen der Oberklasse Desire für den Priority Resolver als Aktionsvorschläge

Unterklassen von Desire

`DesTransVel, DesRotVel` Wünsche für Geschwindigkeiten

`DesCamPan, DesCamTilt` -=- Kamerabewegungen

`DesGrip, DesLift` -=- Manipulatorpositionen

Priority Resolver

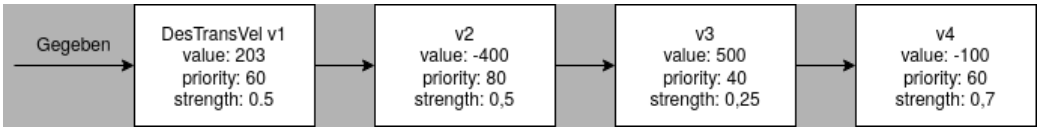
- statische Priorität (Priotiy): gnerelle Wichtigkeit der Verhaltensmuster
  - ^einmalig als Wert [0;100] festeglegt (0<100)
- dynamische priotität (Strength): Situationsabhängig, in jedem Zyklus neu als Wert [0;1.0] festgelegt

Algorithmus:

- Ordne Desires anch fallender Priorität der erzeugenden Verhaltensmuster
- Für alle Steuergrößen s:

```
{
    Desire r, c; r.value = 0; r.strength = 0;
    // Für alle Desire d, solange wie unsere resutlierende Strenght < 1
    ist
    {
        {// Für alle Deisre d1, ... dn mit di.priority == d.priority
        c.value =  Summe di.value * di.strength;
        c.strength = 1/n * Summe di.strength;
        }
    r.value += c.value; r.strength += c.strength;
    }
    s = r.value / r.strength;
}
```

Beispiel: Resolution TransVel



ResolutionTransVel

1 Schritt: Sortieren nach Priority

=> v2 , v1, v4, v3

2 Schritt:

c.value = -200 | => r.value = -200

c.strength = 0.5 | => r.strength = 0.5

3 Schritt:

c.value = 200 * 0.5 + (-100) *0,7 = 30* *c.strength = 1/2* (0,5 + 0,7) = 0,6 r.value = -200 + 30 = -170
r.strength = 0,5 + 0,6 = 1,1

4 Schritt:

s = 170/1,1 = 154

Strategy:

Oberklasse endlicher Automat mit abstrakter Methode plan() im 100 ms Zyklus aufgerufen.

Aktivierung / Deaktivierung Automatenknoten

5.2 Basisverhaltensmuster

Behaviour	Beschreibung
BehConstTransVel BehConstRotVel	Fahren mit konstanter Geschwindigkeit (ohne Kollisiionsvermeidung, ohne alles)
BehLimFor BehLimBack	Annähern an Hindernisse mit wechselnden Geschwindigkeiten
BehCamInit	Initialisierung Kamera
BehMove BehTurn	Fahre nach Odemtriedaten (untersch. Geschwindigkeit Räder)

Beispiel: Andocken

```
package ... dock;
    ... main (...){
       ...
       BehGroup dock = .. // Schlüssel interne verwaltung, 200 mm/s
       BehConstTransVel cv = new BehConstTransVel("cv", 200);
       // spätere Bereinigung durch Garbage
       BehLimFor lf = new BehLimFor("lf", 1000, 2000, 100);
       // stopDistance, slowDistance, slowSpeed
    }

    dock.add(cv, 50); // <- Statische Priorität
    dock.add(lf, 80);
    robot.add(dock);
    robot.run(); // <- start Echtzeitzyklus Resolver
```

5.3 Programmierung eigener Verhaltensmuster

Beispiel: Orthogonales Andocken an eine Wand

siehe Beiblatt

5.4 Sensordatenverarbeitung

5.4.1 Odometriedaten und Koordinaten

Standard koordinatensysteme

- ENN (East-North-Up) bei Landfahrzeugen
- NED (North-East-Down) bei Luftfahrzeugen
- 6 Basen (x,y,z) für Raumpunkt
  - (yaw, pitch, roll) für Orientierung (Euler-Winkel)
  - Drehung um jeweils z, y und x

Vereinfachung bei ENU in einer Ebene: (Diagramm/Notability)

Auslesen Roboterposition mit getPose() als Pose

- get(), set() für x,y,th - double x.findDistanceTo(Pose y) - Euklidischer Abstand zwischen x und y - double x.findAngleTo(Pose y) - Euklidscher Winkel zwischen x und y - double x.diffAngleTo(Pose y) - Differenzwinkel Orientierungen von x und y

5.4.2 Entfernungsdaten

Bisher: Abfrage der Sonare einzeln und direkt

Problem: - Wiederverwendung Programme auf unterschiedlichen Robotern - Entfernungsmessung nicht nur mit Sensoren

Abhilfe:

a) Konfiguration Multisensorsystem

b) Abfrage fusionierter Sensordaten in Regions of Interest

In THIRobCon: - robot.addLaser(); Einbindung Laserscanner in LPS - robot.addVCC4(); Einbindung Canon PTZ-Kamera - => später Persistentes Umfeldmodell mit Datenbank

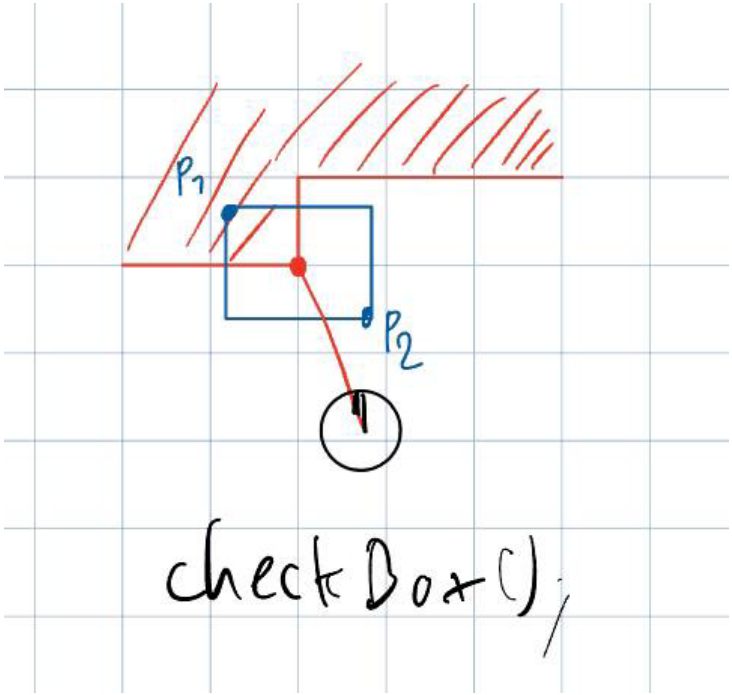
1. b1) Auswertung Kreissegment Methode: double checkPolar()

Beispiel: Orthogonales Andocken

```
// Mit Kegel
leftDist = (int) (checkPolar(2,10,relObstaclePose) - robot.getRadius());
rightDist = (int) (checkPolar(-10,-2,null) - robot.getRadius());
```

b2) Auswertung Rechteck

Methode: checkBox()



checkBox

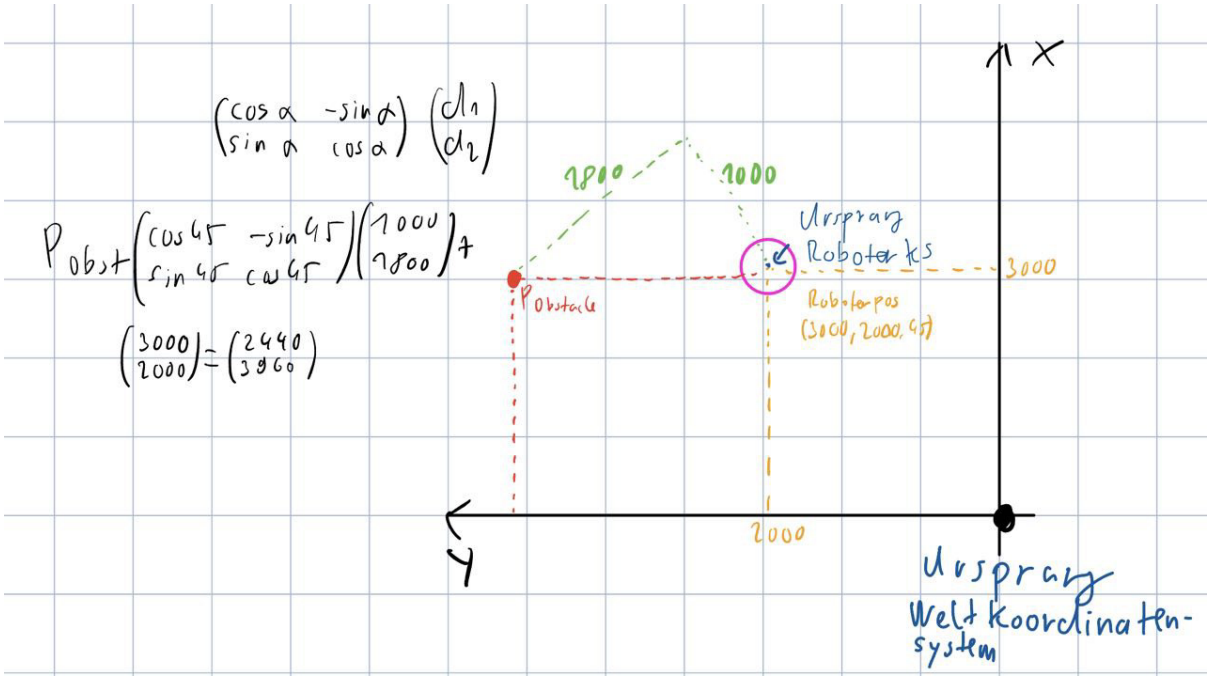
Beispiel: Orthogonales Andocken

```
// Mit Box                      P1.x  p1.y p2.x p2.y
leftDist = (int) (robot.checkBox(2500, 500, 100, 100,relObstaclePose) - robot.getRadius());
rightDist = (int) (robot.checkBox(2500, -500, 100, -100, null) - robot.getRadius());
```

Eintragung P in globale Karte erfordert Koordinatentransformation aus Roboterkoordinatensystem in basis Koordinatensystem

Beispiel: 2D-Koordinatentransformation

Drehmatrix und Verschiebungsvektor



Drehmatrix

5.4.2 Bilder

- Merkmalsteuerung über Farben und Kanten
- Farbbereiche durch Region Growing in Echtzeit -> ACTS
- Kantenerkennung durch Hough Transformation nicht in Echtzeit -> Halcon
- Trainieren Farbbrereiche offline zu Kanälen

Auswertung in THI RobCon:

int getNumBlock(int ch) Anzahl Blob im Kanal

Blob getBlob(int ch, int i) Auslesen Blob i zum Kanal ch in Schleife<

Auswertung eines Blob

int getxcg() (x center of gravity) Koordinaten Blobschwerpunkt

int getycg() bezogen auf Bildgröße

int getArea() Zahl Pixel im Blob

int getTop(), … , getRight() Ausdehnung

Positionierung Kamera:

- BehCam Init Startuausrichtung Kamera pan und tilt (yaw und pitch) - DesCamPan, DesCamTilt dynamisch in den Verhahltensmustern

Beispiel: Signal Dock

```
-> start -------> find ----------> dock ----->
     |            |                |
 BehCam Init    BehRotate         BehConstTransVel BehStop
 (success)      BehFindSignal     BehLiniFor?      (success)
              (success, error)    BehAlign
```