

Exercise 4: Robot Control Architecture

Controller

"bang-bang" control: eine Funktion implementieren mit Fallunterscheidungen (if-then-else) auf den Sensorwerten, und in jedem dieser Fälle eine Aktion den Roboteraktoren mitteilt

proportional control: eine Funktion implementieren, die eine lineare Abbildung der Sensorwerte auf die Aktorenwerte durchführt und letztere dann den Roboteraktoren mitteilt

Nachdem unser Team erst verspätet Coppelia auf Barnabas' Linuxrechner zum Laufen bringen konnte, haben wir uns nun auf die Implementation der bang-bang Controller konzentriert.

Umsetzung:

Neben den konkreten Controller-Klassen PushController, DoorController und WallController haben wir auch die abstrakte Klasse AbstractController implementiert, welche gemeinsame Parameter und Methoden abstrahiert.

AbstractController:

Enthält die Attribute:

```
protected final double maxVel = 120.0 * Math.PI / 180.0; // 4/3 of a full wheel turn  
protected final double noDetectionDistance = 0.05;
```

Diese Werte sind direkt vom basicReactive Controller uebernommen. Dieser Controller uebernimmt die Einhaltung des standardisierten Ablaufs, zuerst setup und anschliessend act solange die Verbindung erhalten ist. Er enthaelt ausserdem Methoden die in allen Controllern verwendet werden koennen um Beispielsweise die Sensorwerte auszuwerten. Manche dieser Methoden koennten evtl. in eine eigene Klasse ausgelagert werden, wurden aber hier der Einfachheit halber alle im Controller implementiert.

PushController:

Ein zylindrischer Gegenstand soll erkannt und an die Wand geschoben werden. Der Zylinder muss am Beginn vor dem Epuck stehen.

DoorController:

Laesst den Epuck die Tuer suchen und darauf zu fahren. Je nach dem wie der Controller instanziiert wurde bleibt der Epuck vor der Tuere stehen oder faehrt stur weiter.

WallController

Der Epuck faehrt gerade aus bis er eine Wand findet. Hat er diese gefunden richtet er sich so aus das die Wand auf seiner linken Seite ist und beginnt ihr zu folgen.

Die verwendeten Werte wurden entweder vom Beispiel Code uebernommen oder durch trial and error her gefuehrt

Klassen Diagramm:

