Robotik

Informatik
2. Klassen
Kantonsschule am Burggraben

Ivo Blöchliger

Robotik

- Was ist ein Roboter?
- Mathe, Software, Physikalische Realität
- Herausforderungen
- Python-Repe
- Simulationsmodus

Was ist ein Roboter?

Was ist ein Roboter? (*Wikipedia*)

- **Definition nach VDI-Richtlinie 2860**

  „**Industrieroboter** sind universell einsetzbare Bewegungsautomaten mit **mehreren Achsen**, deren Bewegungen hinsichtlich Bewegungsfolge und Wegen bzw. Winkeln frei (d. h. ohne mechanischen bzw. menschlichen Eingriff) **programmierbar** und gegebenenfalls sensorgeführt sind. Sie sind mit Greifern, Werkzeugen oder anderen Fertigungsmitteln ausrüstbar und können Handhabungs- und/oder **Fertigungsaufgaben** ausführen.“

Was ist ein Roboter? (*Wikipedia*)

- Definition nach Robotic Industries Association

  A robot is a **reprogrammable**, **multifunctional manipulator** designed to move material, parts, tools or specialized devices through **variable programmed** motions for the performance of a **variety of tasks**

Verschiedene Levels

- **Manual Manipulator**
- **Fixed Sequence Robot**
- **Variable Sequence Robot**
- **Playback Robot**
- **Numerical Control Robot**
- **Intelligent Robot**
  Diese höchste Roboterklasse ist für Geräte gedacht, die über verschiedene **Sensoren** verfügen und damit in der Lage sind, den **Programmablauf** selbsttätig den Veränderungen des Werkstücks und der Umwelt **anzupassen**.

Roboter oder nicht?

Aktoren	
Anzahl Achsen	
Sensoren	
Umprogrammierbar?	
“Intelligent”?	
Kategorie?	

Roboter oder nicht?

Aktoren	
Anzahl Achsen	
Sensoren	
Umprogrammierbar?	
“Intelligent”?	
Kategorie?	

Roboter oder nicht?

Aktoren	
Anzahl Achsen	
Sensoren	
Umprogrammierbar?	
“Intelligent”?	
Kategorie?	

Roboter oder nicht?

Aktoren	
Anzahl Achsen	
Sensoren	
Umprogrammierbar?	
“Intelligent”?	
Kategorie?	

Roboter oder nicht?

Aktoren	
Anzahl Achsen	
Sensoren	
Umprogrammierbar?	
“Intelligent”?	
Kategorie?	

Roboter oder nicht?

Aktoren	
Anzahl Achsen	
Sensoren	
Umprogrammierbar?	
“Intelligent”?	
Kategorie?	

Roboter oder nicht?

Aktoren	
Anzahl Achsen	
Sensoren	
Umprogrammierbar?	
“Intelligent”?	
Kategorie?	

Roboter oder nicht?

Aktoren	
Anzahl Achsen	
Sensoren	
Umprogrammierbar?	
“Intelligent”?	
Kategorie?	

Roboter oder nicht?

Aktoren	
Anzahl Achsen	
Sensoren	
Umprogrammierbar?	
“Intelligent”?	
Kategorie?	

Roboter oder nicht?

Aktoren	
Anzahl Achsen	
Sensoren	
Umprogrammierbar?	
“Intelligent”?	
Kategorie?	

Roboter oder nicht?

Aktoren	
Anzahl Achsen	
Sensoren	
Umprogrammierbar?	
“Intelligent”?	
Kategorie?	

Roboter oder nicht?

Aktoren	
Anzahl Achsen	
Sensoren	
Umprogrammierbar?	
“Intelligent”?	
Kategorie?	

Mathematische Welt

- Alles ist exakt

  $2 + 2 = 4$

- Vieles ist beweisbar
  - Es gibt prinzipiell unbeweisbare wahre Aussagen in der Mathematik. (Unvollständigkeitssatz von Gödel).
- Das ist “einfach”, im Sinne von “keine Überraschungen”

Software

- Ziemlich deterministisch
  - Gleicher Input, gleicher Output
- Unvorhergesehene Dinge
  - Datei nicht gefunden
  - Netzwerkprobleme
- "%.20f" % 0.2 liefert '0.20000000000000001110'

Physikalische Realität

- 1m ist nur ungefähr 1m
- Geradeaus ist nur ungefähr geradeaus
- GPS-Position ist nur auf 5m bis 50m genau
- Jegliche Messdaten sind ungenau und fehlerbehaftet
- Alle Sensoren und Aktoren sind Unikate und leicht verschieden

Physikalische Realität

- Befehl an Roboter: **Fahr 1m geradeaus**
  - Roboter: fährt 95cm leichte Rechtskurve
- Befehl an Roboter: **Halt**
  - Steht nach 0.5s still, hat sich dabei noch um 5° nach links gedreht.
- Über dem gleichen Untergrund meldet der Helligkeitssensor fluktuierende Werte zwischen 354 und 431.

Umgang mit dieser Ungenauigkeit

- Grosse Beschleunigungen vermeiden

  F = m*a

- Beschleunigung: Änderung der Geschwindigkeit pro Zeit.
- Unsinnige Sensordaten verwerfen, andere mitteln
- Positionsfehler über Sensordaten korrigieren
  - z.B. einer Linie entlang fahren, sich einer Wand nähern.

Unser Roboter

- 2 unabhängige Räder
  - Ermöglicht Drehen auf Platz
- 1 Heber
- 1 Druckknopf
- 1 Ultraschall-Distanzsensor
- 1 (2) Helligkeits/Farb-Sensoren

Repetition Python

```
if Bedingung:
  machwas()
else:                    # Optional
  machwasanderes()


while Bedingung:         # So lange als
  wiederholwas()
```

Repetition Python

Bedingung → Resultat True/False

- Vergleiche: **<** **<=** **==** **>=** **>** **!=**
- Verknüpfungen: **not** **and** **or**

and: Beide True → True

or: Mindestens eines True → True

Grundstruktur Robotik

```
# Initialisierung (Motoren, Sensoren)


while True:
  # Sensoren auslesen
  # Je nach Sensordaten und internem Zustand
    # Aktoren steuern
    # Interner Zustand neu definieren
```

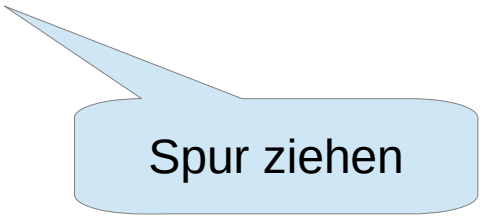
Initialisierung ohne Sensoren

```
from simrobot import *

RobotContext.enableTrace(True)

robot = LegoRobot()

# Raeder hinzufuegen
gear = Gear()
robot.addPart(gear)
```



Roboter bewegen (Gear -Objekt)

- Blockierende Methoden (Angabe in Millisekunden)
  - Fährt entsprechende Zeit, hält dann an
  - Programm ist so lange blockiert.
- Nicht-blockierende Methoden
  - Motoren drehen von diesem Zeitpunkt an
  - Programm läuft weiter, kann andere Dinge tun
  - Am Schluss `gear.stop()`

Roboter bewegen

```
gear.forward(1000)   # 1000 ms vorwärts, dann stoppen,
                     # Programm blockiert so lange.
Tools.delay(1000)    # 1000 ms warten
gear.left()          # links drehen, nicht blockierend,
                     # Programm läuf weiter, wie Motoren
Tools.delay(500)     # 500 Millisekunden warten.
gear.stop()          # Motoren stoppen
```

Statt diese 3 Zeilen: gears.left(500)

Aufgaben

- Einstiegsaufgaben
  - Bewegen
  - Helligkeitssensor auslesen und darauf reagieren
- Challenges
  - Einer Linie entlang fahren
    - Wie würden Sie einer Linie folgen, wenn Sie nur einen münzengrosses Feld auf dem Boden vor Ihren Füssen sehen würden?
  - Sich einer Wand nähern.

Spielwiese

- **Start auf gelb**
- **Richtung unten**
- Dreiecke sind Mauern für Distanzsensor
- Verläufe für Helligkeits-sensor zur Orientierung

Challenge

- Startposition und -orientierung werden zufällig gewählt
  - Sonst wäre das Programm deterministisch, unrealistisch!
- Farbverlauf anstatt weiss/schwarz
  - Sensorsichtfeld ist einige mm gross
    → schwarz/weisse Kante ist grau.

Challenge

- Zip-Archiv robochallenge.zip vom Wiki herunterladen
  - Rechts-Klick → Link speichern
- Zip-Archiv **vollständig entpacken.**
  - Nur ein Doppelklick zeigt nur was drin ist, entpackt aber nix.
- TigerJython starten, Datei challenge.py öffnen
- Testen, Verstehen
- Challenge 1: Robo stoppt auf rotem Punkt unten rechts
- Challenge 2: Robot stoppt auf rotem Punkt oben