

Evaluation der Nao Plattform für den Einsatz maschineller Lernverfahren

Medieninformatik
HdM Stuttgart - SS2011

Übersicht

- Evaluation der Plattform
 - Vereinfachen der Schnittstelle
 - Lernversuche
 - Ausblick
 - Fazit
-
- Folien 8 – 17:
Fragen / Programmcode / Diskussion

Evaluation der Plattform

- Hardware

- Motoren: Kraft, Hitze, Auslenkungen
- Sensoren: genauer als die Mechanik
- Schnittstellen: WLAN, LAN, Bluetooth

- Software

- Choregraphie
- NAOqi SDK

- Schnittstelle zum Roboter

- Liste der Proxies: <http://Nao-IP:9559>

Vereinfachen der Schnittstelle

- Kapselung der Proxies
 - Objektorientierte Lösung
 - Erweiterbarkeit
- Steuerungsmethoden
 - Manuelle Steuerung des Roboters
 - Steuerung durch Lern-Algorithmus
 - Direkte Kommandos
 - Schrittweise Kommandos
 - Auswirkungen der Mechanik

Lernversuche

- PyBrain

- Experiment
- Environment: Sensoren, Aktoren
- Task: Bewertungsfunktion
- Agent
- Learner: Lern-Algorithmus
- KNN: Strategierepräsentation
- Komplexe Verschachtelung über sehr viele Klassen bei erweiterten / neueren Lern-Algorithmen
- PGPE: Policy-Gradient Parameter Exploration
- Simulation von Versuchen zur Vorbereitung

Ausblick

- Fähigkeiten der Roboterhand
 - Versuche zum Beispiel in Richtung RoboCup orientieren
 - Weitere Versuche:
 - Neuer vereinfachter Stand-Alone Code für PGPE
 - Überarbeiten und Vervollständigen der Steuerungsmodule
 - Versuchsreihen zu maschinellem Lernen (Vertiefen)
 - Kommunikation & Interaktion zwischen zwei Robotern
- Begrenzte Möglichkeiten der Nao Plattform berücksichtigen

Fazit

- Breite \leftrightarrow Tiefe
 - Lernversuche
 - Viel „Trial & Error“ mit PyBrain
 - „Proof-Of-Concept“ (bisher ohne Belege)
 - konkrete Ergebnisse & Auswertung nötig!
 - Einfache Verwendung der Plattform
- Nao ist geeignet für Einstieg in die Robotik

Fragen?

Programmcode 1/8

- Startmodul
 - nao_control_prototype.py
 - Controller- / Nao-Objekt erzeugen
 - Konfiguration
 - Verbindung aufbauen

Programmmcode 2/8

- Eingabe
 - padInput.py
 - Eigener Thread
 - Lesen vom Eingabegerät
 - getAll()
 - Deadzones

Programmcode 3/8

- Mapping der Eingabe auf Funktionen
 - mappedInput.py
 - Abfrage-Intervall
 - Knöpfe und Achsen
 - Funktionen des Steuerungsmoduls

Programmcode 4/8

- Steuerungsmodul
 - naoControl.py
 - Funktionen zur Steuerung
 - Blocking Calls \leftrightarrow Non-Blocking Calls
 - Sensoren vs. Interne „aktuelle“ Werte
 - Feedback \rightarrow Erkennen von erfolglosen Aktionen
 - Greifen

Programmcode 5/8

- Kameras
 - naoCamera.py
 - Konfiguration
 - switchCam()
 - getPILImage() / getCVImage()

Programmcode 6/8

- Sensoren:

- naoSensors.py

- Gelenke: Sensoren / letztes Kommando

- Ultraschall

- Drucksensoren

- Touchsensoren

Programmcode 6/8

- Einbindung von Lernverfahren (I/II)
 - grabbingTestNew.py
 - Startet im aktuellen Code erst auf Knopfdruck
 - Referenz auf Nao-Objekt
 - Environment, Task, Agent, KNN, Experiment
 - Batch, Episodes, Steps

Programmcode 7/8

- Einbindung von Lernverfahren (II/II)
 - Starten von Lern-Versuchen
 - doEpisodes()
 - Verwenden des „Gelernten“
 - Unklare Dokumentation
 - neuer PGPE-Code von Frank Sehnke
 - Visualisierung von Aktionen
 - Lernen, Testen, Visualisierung der Bewertung

Diskussion?