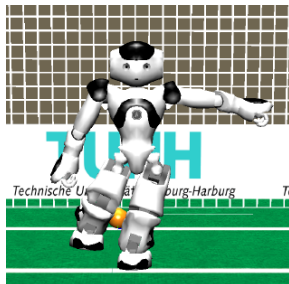


How NAO learns to kick the ball ...

Jannick, André, Daniel, Florian

11.07.2012



Agenda

- 1 Einleitung
- 2 Arbeitsschritte
 - Phase 1
 - Phase 2
 - Phase 3
- 3 Ergebnis und Ausblick
- 4 Präsentation

Einleitung

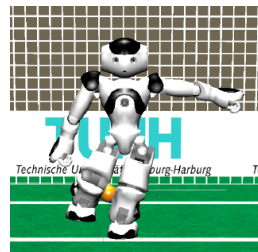
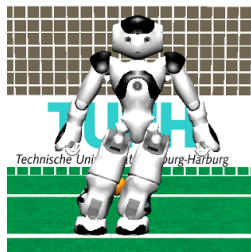
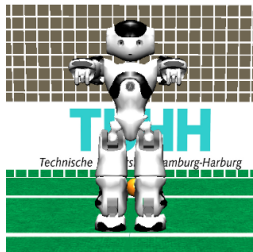
- Arbeit mit Robotiksystemen
- Implementieren einer Schussbewegung für NAO

Arbeitsschritte

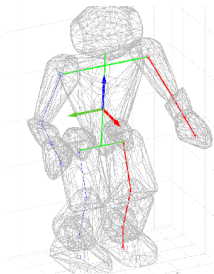
- 1 Statisches Gleichgewicht auf einem Bein (Simulator)
- 2 Vollführen einer Schussbewegung (Simulator)
- 3 “Übertragen“ auf den NAO (Simulator, NAO)

Phase 1 - Schwerpunkt

- Einarbeiten in die API und Nutzung des Simulators Webots (naoController.m)
- Verschieben des Schwerpunktes über linkes Bein (ForwardKinematics, InverseKinematics)
- Anheben des rechten Fußes



Schwerpunktverschiebung



Pseudo-Code:

*KinematicMatrix XFoot KinematicMatrix
Com2Torso*

Torso2XFoot = XFoot⁻¹

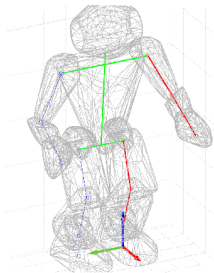
Com2XFoot = Torso2XFoot · Com2Torso

set y and x coordinates

XFoot2Com = Com2XFoot⁻¹

XFoot = Com2Torso · XFoot2Com

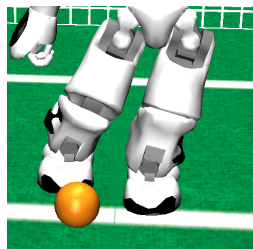
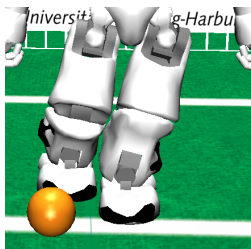
get angles via InverseKinematics



Phase 2 - Schuss

- Ausgangslage:
 - Ball liegt vor der Fußspitze des NAOs
- Die Frage nach dem "Wie" der Schussbewegung
- Ziele:
 - Ball auf mittlerer Höhe treffen
 - Nicht umfallen
 - Einen großen Impuls übertragen
 - Keine innere Kollision
 - Einstellen des Schusswinkels

Einstellen des Schusswinkels und der Fußposition



Phase 3 - Schuss optimieren und übertragen auf den NAO

- Übertragen des naoController (Matlab) in C++
- Test auf dem Roboter
- Idee: Genauere Schusswinkeleinstellung (Interpolation der Fußform), aber Praxistest zeigt, dass noch weitere Parameter wichtig sind.

Ergebnis und Ausblick

- TuhhSDK und vorhandene Projektstruktur erleichtern Einarbeitung
- Implementierung der Schussbewegung als einzelnes Modul
- Eingeschränkter Schusswinkel
- Nächster nötiger Arbeitsschritt: genauere Schusswinkelbestimmung (und erkennen, annähern und positionieren vor dem Ball)

Präsentation

