

Ansprechpartner:



Philipp Karg, M. Sc.

IRS, Raum 206

Tel.: 0721/608-42708

philipp.karg@kit.edu

Beginn: ab Oktober/November

Dauer: 3-6 Monate

experimentell anwendungsorientiert theorieorientiert

Ihre Interessen:

ROS/Gazebo

Optimierung

C++/Python/Matlab

Robotik

Reglerentwurf

Modellbildung/Identifikation

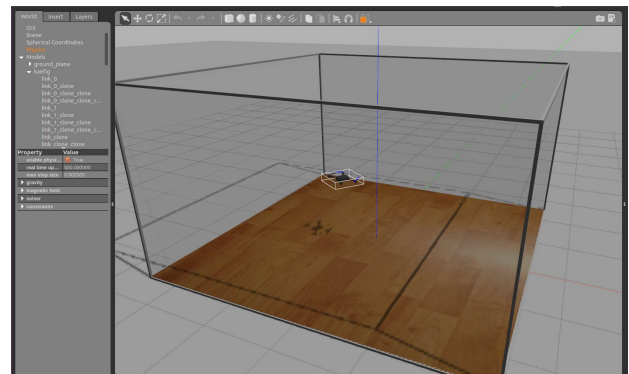


Bachelorarbeit

Simulationsumgebung für die optimale Regelung von Robotersystemen

Motivation:

Moderne Simulationsumgebungen bieten die Möglichkeit verschiedenste Robotersysteme (z.B. mobile Roboterplattformen, Drohnen) in komplexen 3D-Umgebungen zu simulieren und deren Verhalten möglichst realitätsnah abzubilden. Hierbei kann bereits frühzeitig in der Entwicklung die Software für das reale System getestet und evaluiert werden (Software-in-the-loop). Eines der am weitesten verbreiteten Simulationsframeworks für robotische Systeme ist Gazebo, welches in das Robot Operating System (ROS) integriert ist. Die Open-Source-Software bietet eine Anbindung an die in der Regelungstechnik übliche Matlab-/Simulink-Umgebung, womit die Nutzung der Potentiale von Gazebo/ROS für den praktischen Entwurf regelungstechnischer Systeme ermöglicht wird. Beispielsweise kann das Verhalten von in Matlab entworfenen Regelalgorithmen zur optimalen Folge einer Referenztrajektorie durch das Robotersystem praxisnah simuliert und visualisiert werden. Letztlich ist damit auch eine einfache Übertragung auf das reale System gegeben.



Aufgabenstellung:

Ziel der Arbeit ist die Konzeptionierung und Implementierung einer Simulationsumgebung für die optimale Regelung eines Robotersystems. Das Konzept soll dabei so ausgelegt werden, dass unterschiedliche Simulationsszenarien und auch Robotersysteme berücksichtigt werden können. Die Simulation selbst soll in Gazebo realisiert werden, wobei die Anpassung von bereits frei verfügbaren Simulationsumgebungen für mobile Roboterplattformen oder Quadrocopter im Fokus steht. Die Realisierung der Schnittstellen in ROS soll schließlich die Möglichkeit zur Ansteuerung und Auswertung mittels Matlab garantieren. Tests der Simulationsumgebung – von der Ansteuerung bis zur Auswertung der Simulationsdaten in Matlab – anhand eines aussagekräftigen Beispiels runden die Arbeit ab.

