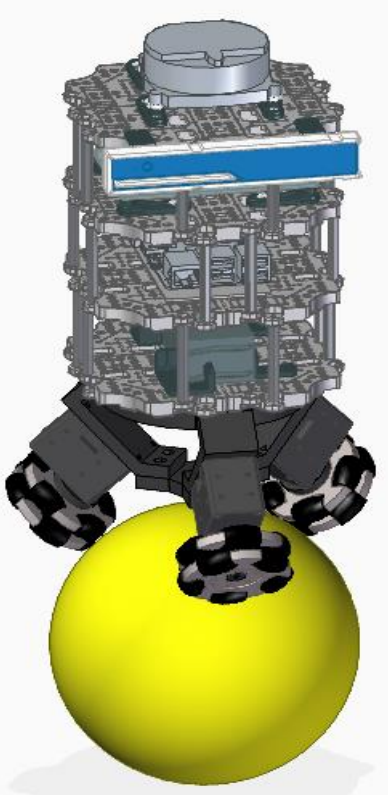
Aufbau und Regelung eines Ballbots

TECHNISCHE UNIVERSITÄT DARMSTADT

Florian Müller, Markus Lamprecht, Michael Suffel
Projektseminar
Betreuer: Dr.-Ing. Eric Lenz



Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT

Abb. 1.1: Humanoider Ballbot [1]

Abb. 1.2: Humanoider Roboter [2]

Inhaltsverzeichnis

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Konstruktion
Konzeptionierung

TECHNISCHE UNIVERSITÄT DARMSTADT

Unter Zuhilfenahme bestehender Ballbot-Konzepte

Abb. 1.3: Ballbot REZERO [3]

Unter Zuhilfenahme eines modelbasierten Konzeptes

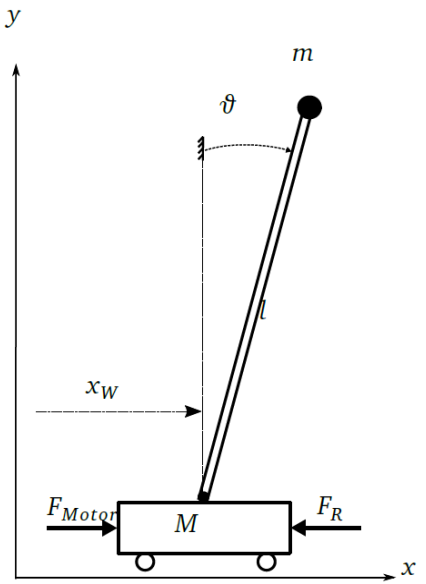


Abb. 1.4: Inverses Pendel auf einem Schlitten [4]

1. Konstruktion
Analyse bestehender Ballbots

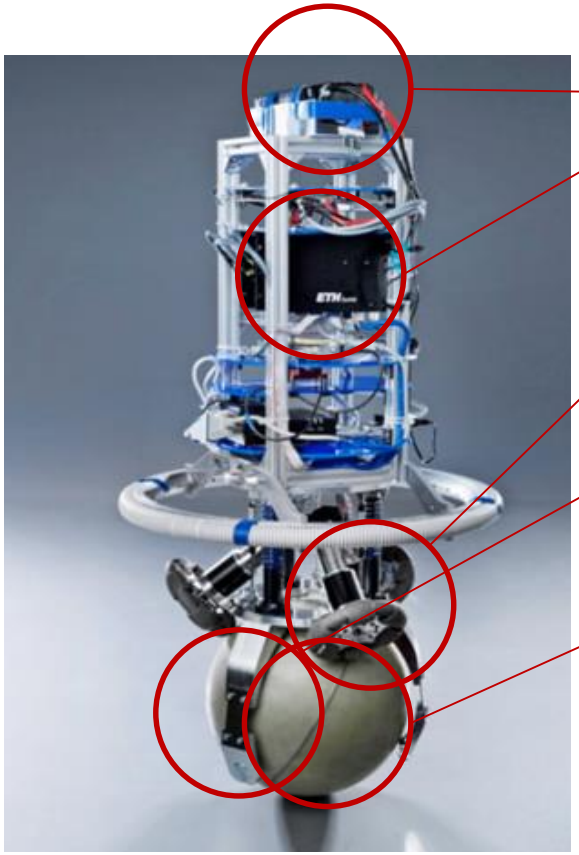


Abb. 1.5: Ballbot REZERO [3]

- Verlagerung des Schwerpunktes nach oben
- Sehr präzise gefertigte omnidirektionale Räder
- Ball-Arrester
- Harte Kugel mit sehr hohem Reibkoeffizienten

1. Konstruktion
Oberbau – 1. Prototyp

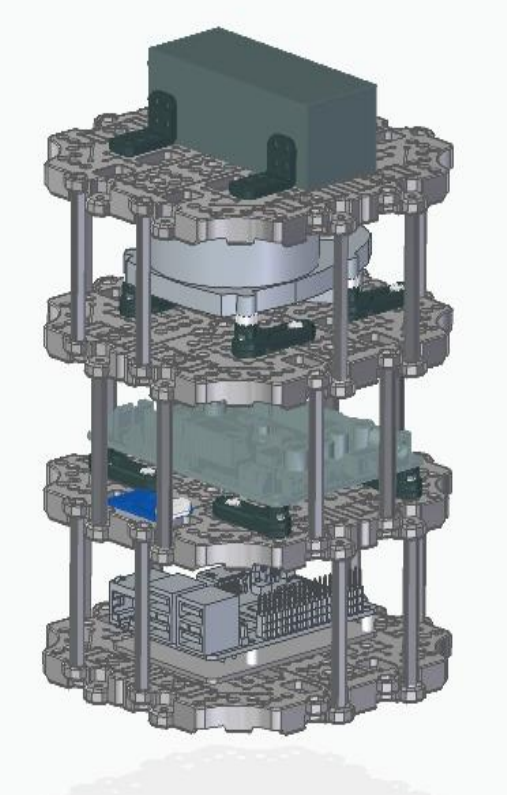


Abb. 1.6: Ballbot Oberbau

Vorteile:

- Hoher Schwerpunkt
- Stabiles Gerüst

Nachteile:

- Ungeeignete Platzierung der Sensoren
- Schlechtes Regelverhalten aufgrund von Schlupf zwischen Omniwheels und Ball

1. Konstruktion
Oberbau - Analyse modelbasiertes Konzept

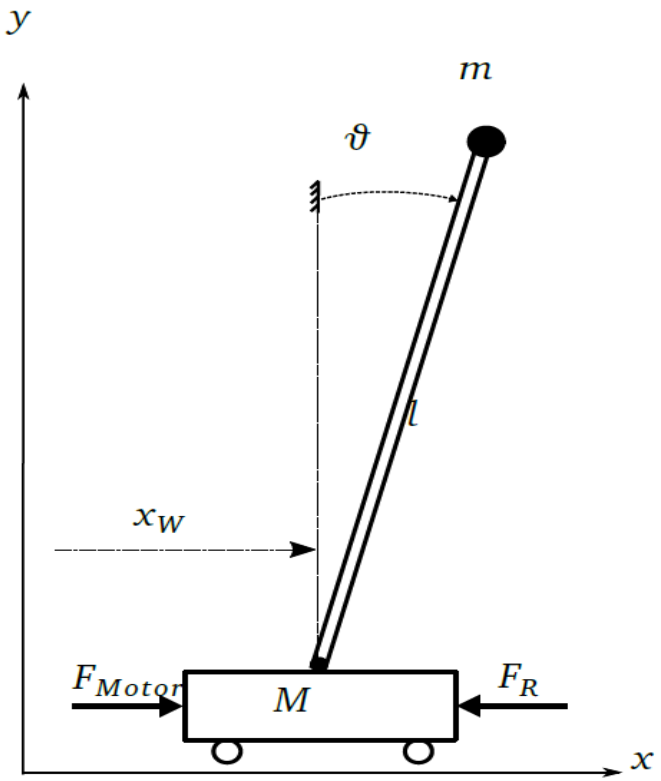


Abb. 1.7: Inverses Pendel auf einem Schlitten [4]

Dynamische Gleichungen:

$$\ddot{\vartheta} = \frac{3}{4 \cdot l} \cdot [-\cos\vartheta \cdot \ddot{x}_w + g \cdot \sin\vartheta]$$

$$\ddot{x}_w = \frac{F_{motor} - F_{Reib}}{M+m} + \frac{m \cdot l}{M+m} \cdot \dot{\vartheta}^2 \cdot \sin\vartheta - \ddot{\vartheta} \cdot \cos\vartheta$$

Erkenntnisse:

- $l\downarrow \rightarrow \ddot{\vartheta}\uparrow$ und $\ddot{x}_w\downarrow$
- $l\uparrow \rightarrow \ddot{\vartheta}\downarrow$ und $\ddot{x}_w\uparrow$

1. Konstruktion
Oberbau - 2. Prototyp

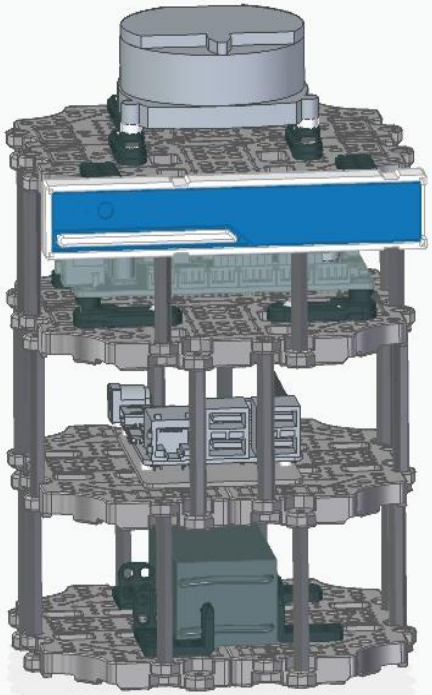


Abb. 1.9: Oberbau mit Hardware

Eigenschaften:

- Verbessertes Regelungsverhalten
- Stabiles Gerüst
- Bessere Platzierung der Sensoren

1. Konstruktion
Unterbau - Layout

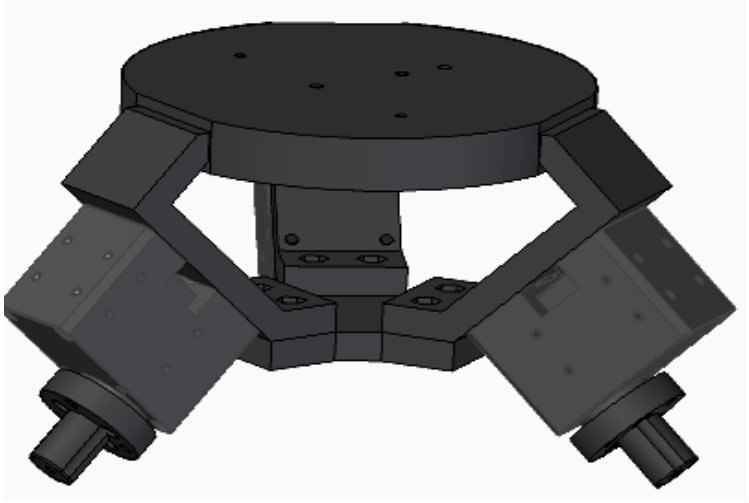


Abb. 1.10: Unterbau - Gesamt

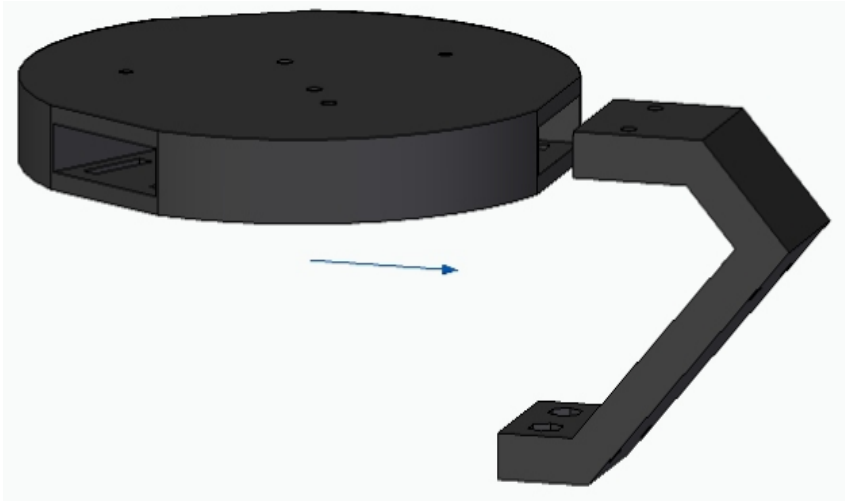
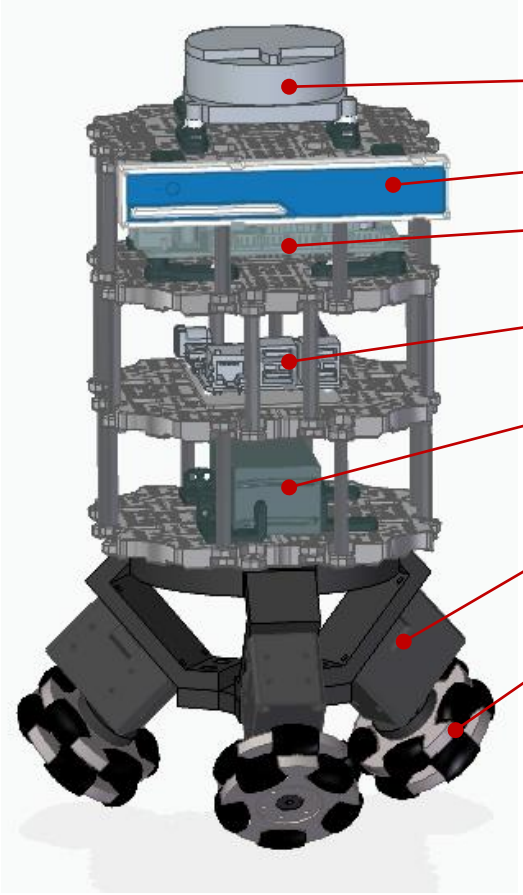


Abb. 1.11: Teleskopsystem

1. Konstruktion
Ergebnis



Vergebaute Hardware:

- Lidar: LDS-01 (Lokalisierung)
- Kamera: Intel Realsense R200 (Lokalisierung)
- OpenCR-Board: IMU MPU-9250 (Stabilisierung)
- Up-Board (ROS)
- Batterie: LI-PO 11.1V 1800 mAh
- Servomotoren: Dynamixel XM430-W350T
- Omniwheels

Abb. 1.12: Hardwarekomponenten Ballbot

2. Modellbildung und Regelung
Einführung

- Aufteilung in eine yz-, xz- und xy-Ebene
- System bestehend aus Kugel, virtuellem Rad und Körper je Ebene

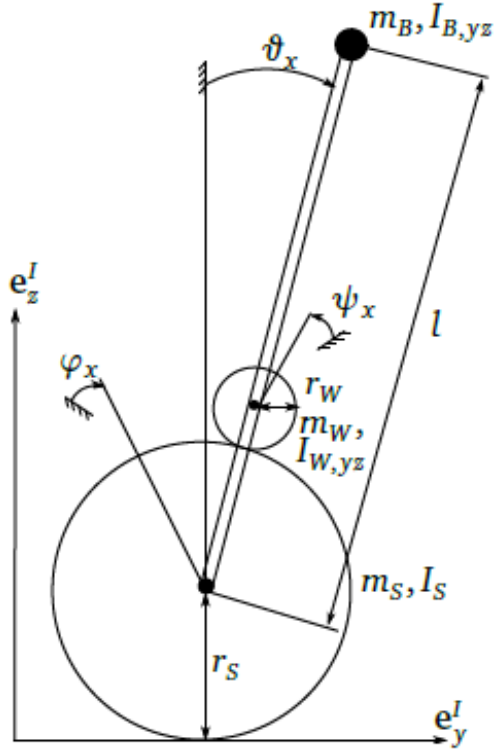


Abb. 2.1: Darstellung yz-Ebene [5]

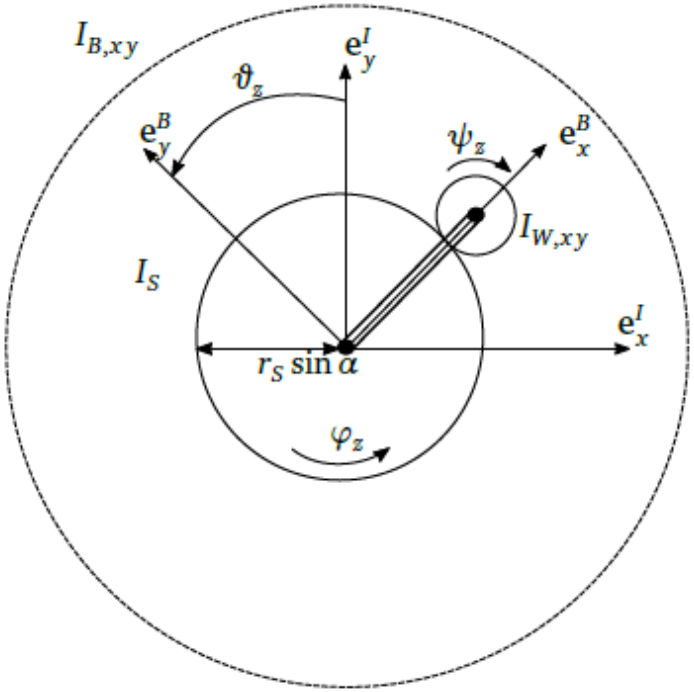


Abb. 2.2: Darstellung xy-Ebene [6]

2. Modellbildung und Regelung
Bewegungsgleichungen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Festlegen minimale Koordinate (yz-Ebene):

$$\boldsymbol{q}_{yz} = \begin{bmatrix} \varphi_x \\ \vartheta_x \end{bmatrix}$$

- Nichtlineare Bewegungsgleichungen in Matrixform (yz-Ebene):

$$\boldsymbol{M}_x(\boldsymbol{q}, \dot{\boldsymbol{q}}) \cdot \ddot{\boldsymbol{q}} + \boldsymbol{C}_x(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{G}_x(\boldsymbol{q}) = \boldsymbol{f}_{NP,yz}$$

- Mit Zustandsvektor $\boldsymbol{x} = [\varphi_x \quad \vartheta_x \quad \dot{\varphi}_x \quad \dot{\vartheta}_x]$ in folgende Form:

$$\dot{\boldsymbol{x}}_{nl} = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \ddot{\boldsymbol{q}} \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \boldsymbol{M}_x^{-1} \cdot \left(\boldsymbol{f}_{NP,yz} - (\boldsymbol{C}_x + \boldsymbol{G}_x) \right) \end{bmatrix}$$

2. Modellbildung und Regelung
Zustandsraummodell und Reglerentwurf

TECHNISCHE UNIVERSITÄT DARMSTADT

- Linearisierung mit Taylor-Reihenentwicklung um $\boldsymbol{x}_0 = [0 \quad 0 \quad 0 \quad 0]$

- Gleichgewichtslage Roboter von Interesse → $\boldsymbol{x}^* = [\vartheta_x \quad \dot{\vartheta}_x]$

$$\boldsymbol{A}^* = \begin{bmatrix} 0 & 1 \\ 17{,}11 & 0 \end{bmatrix} \quad \boldsymbol{B}^* = \begin{bmatrix} 0 \\ -10{,}69 \end{bmatrix} \quad \boldsymbol{C}^* = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \boldsymbol{D}^* = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- Minimierung folgendes Gütemaß:

$$J = \int_0^\infty \left(\boldsymbol{x}^T(t) \cdot \boldsymbol{Q} \cdot \boldsymbol{x}(t) + \boldsymbol{u}^T(t) \cdot \boldsymbol{R} \cdot \boldsymbol{u}(t) \right) dt \qquad (2.6)$$

mit: Trajektorienverlauf: $\boldsymbol{Q} = \begin{bmatrix} 100 & 0 \\ 0 & 50 \end{bmatrix}$

Stellgrößenverlauf: $\boldsymbol{R} = 200$

2. Modellbildung und Regelung
Matlab/SIMULINK

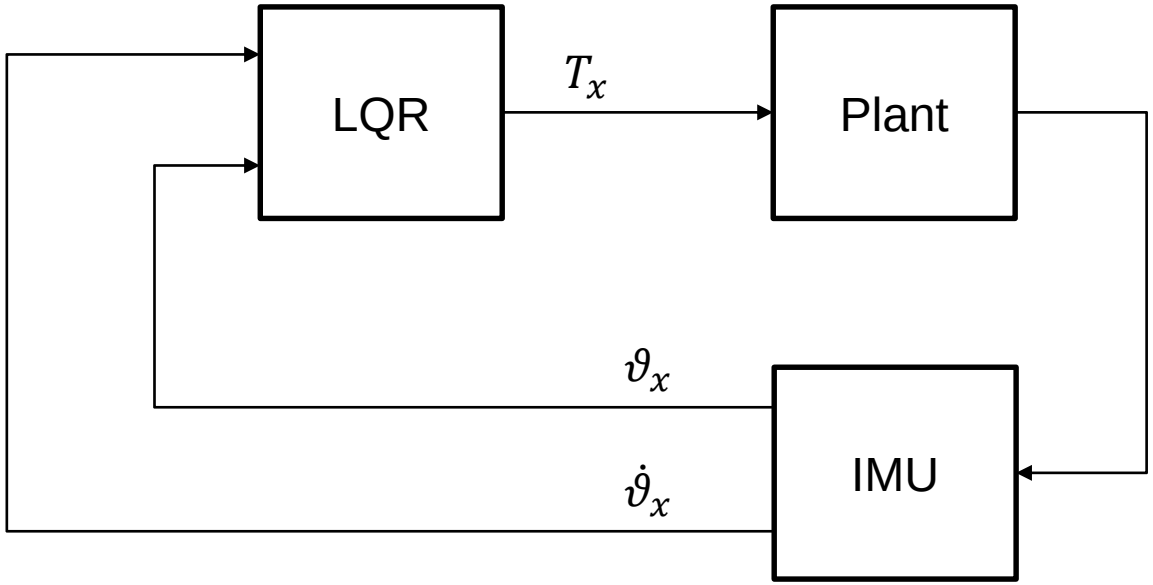


Abb. 1.12: Regelungsprinzip

2. Modellbildung und Regelung
Matlab/SIMULINK

TECHNISCHE UNIVERSITÄT DARMSTADT

Anpassung an reale Gegebenheiten:

- Verrauschte Sensorsignale: weißes Rauschen (Datenblatt)

Auslesen/Verarbeiten Sensorsignale:

- aktualisierte Sensordaten (IMU) nur zu bestimmten Zeitabständen
- additiv aus: - Aktualisierungsfrequenz IMU (200Hz)
  - Auslesen/Verarbeiten Sensordaten(<<1ms)
  - „Schreiben" Drehmomente (3ms)
- Abtastfrequenz $f_{Abt} = 125\ Hz$
- Simulink-Modell: Abtast-Halteglied und Totzeitglied

3. Implementierung
Programmaufbau

TECHNISCHE UNIVERSITÄT DARMSTADT

Entwicklungsumgebung:

ArduinoIDE, Unterstützung OpenCR-Board ab Version 1.6.4

Grundstruktur Arduino-Programm:

- setup()-Methode
- loop()-Methode

Regelungsprogramm

- Interrupt-Routine (HardwareTimer)
- readIMU(): Einlesen Sensordaten
- computeController(): Berechnung Regelgesetz
- computeTorque(): Umrechung/Ausgabe auf Stromwerte

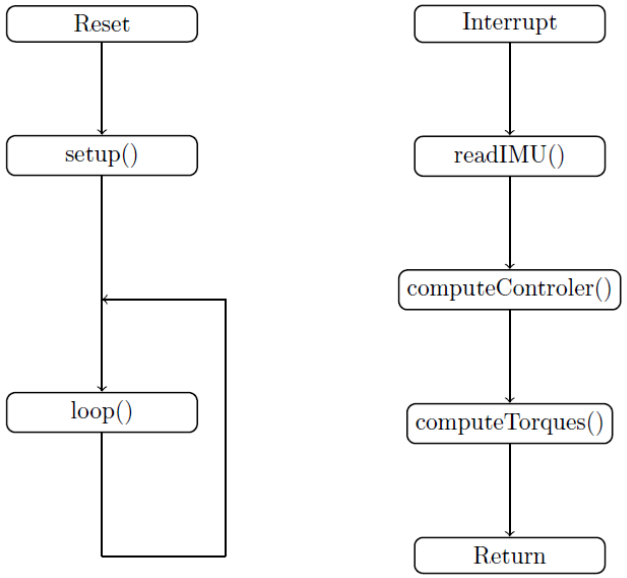


Abb. 3.1: Hauptprogramm und Serviceroutine

3. Implementierung
Anpassungen

TECHNISCHE UNIVERSITÄT DARMSTADT

Filterung Messrauschen:

Letzten drei Sensorwerte gespeichert, aufsummiert und Mittelwert gebildet

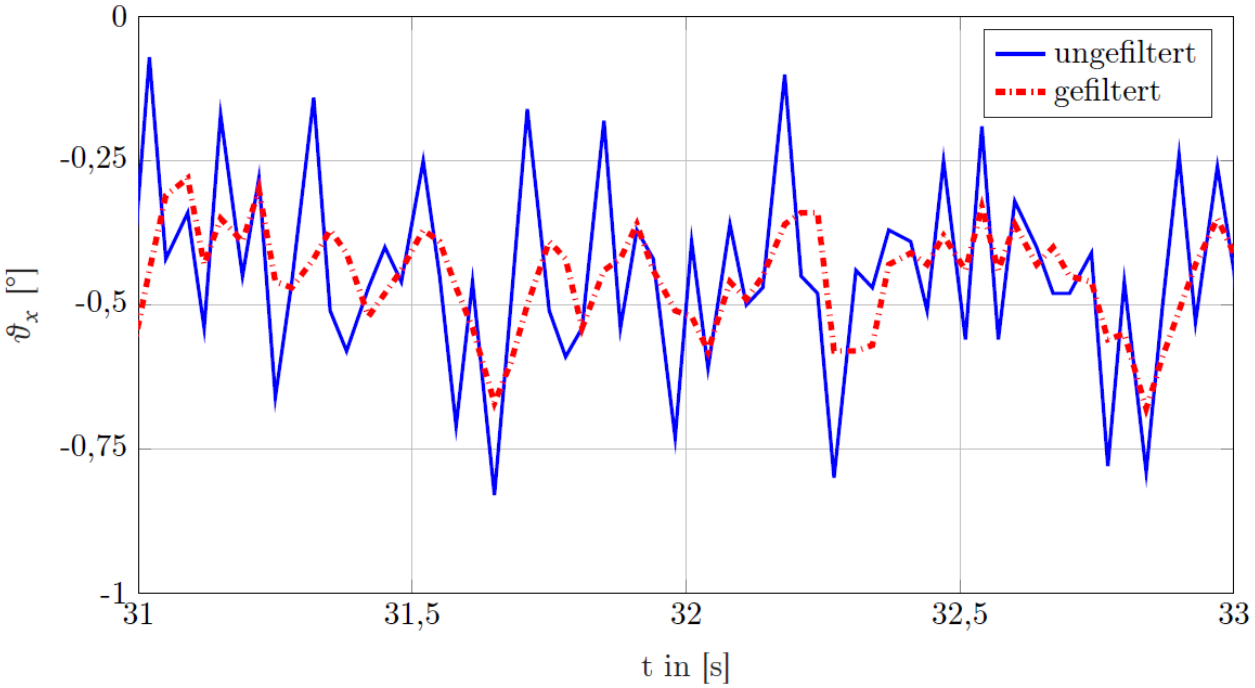


Abb. 3.2: Vergleich ungefilterte(blau) und gefilterte Messwinkel

3. Implementierung
Berücksichtigung Motoreigenschaften

TECHNISCHE UNIVERSITÄT DARMSTADT

Bestimmung der jeweiligen Drehmoment-Offsets:

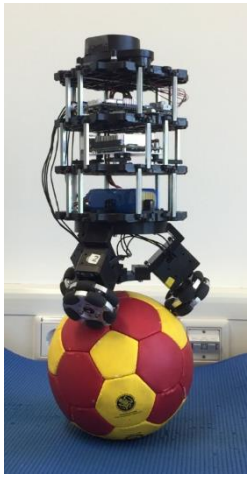


Abb. 3.3: Hardwarekomponenten Ballbot

Bestimmung der jeweiligen Drehmoment-Konstanten:

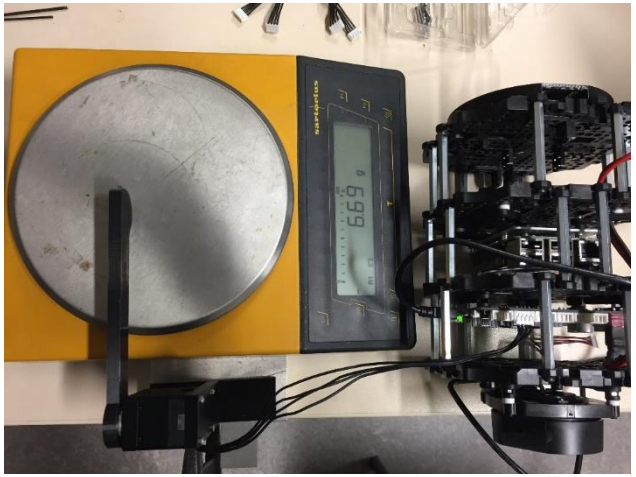


Abb. 3.4: Hardwarekomponenten Ballbot

<u>Motor 1:</u>
- 125,53 Units/Nm
- Offset: 71 mNm

<u>Motor 2:</u>
- 132,82 Units/Nm
- Offset: 90 mNm

<u>Motor 3:</u>
- 207,1 Units/Nm
- Offset: 53 mNm

4. Simulation
Übersicht - Simulatoren

TECHNISCHE UNIVERSITÄT DARMSTADT

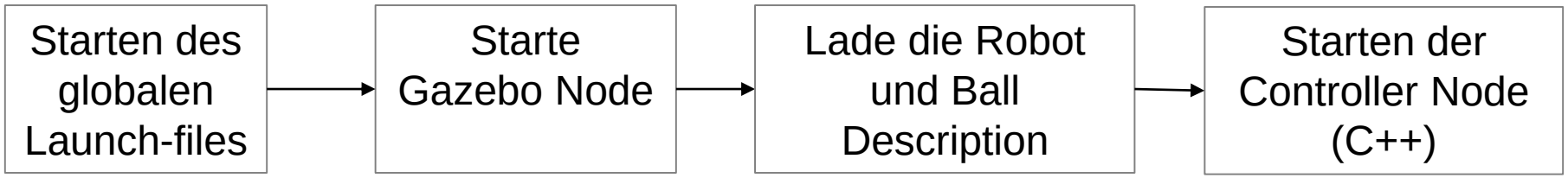
Vorteile:

- Standardsimulator von ROS
- Open-Source

Vorteile:

- Geringere CPU-Auslastung
- Benutzerfreundliches Interface

4. Simulation
Ablauf




```
<!-- macro for wheels and their joints: -->
<xacro:macro name="wheel_link_joint_macro" params="pos *joint_origin">
   <joint name="wheel${pos}_joint" type="continuous">
      <parent link="body_link" />
      <child link="wheel${pos}_link" />
      <xacro:insert_block name="joint_origin" />
      <axis xyz="1 0 0" />
      <limit effort="${XM430_W350_R_max_effort}" velocity="${XM430_W350_R_max_velocity}" />
      <dynamics friction="${wheel_joint_friction}" damping="${wheel_joint_damping}" />
   </joint>

   <link name="wheel${pos}_link">
      <visual>
         <geometry>
            <mesh filename="package://ballbot_description/wheel.stl" scale="0.001 0.001 0.001" />
         </geometry>
      </visual>
      <inertial>
         <mass value="${mass_wheel}" />
         <inertia ixx="2.75e-5" ixy="0.0" ixz="0.0" iyy="1.718e-5" iyz="0.0" izz="1.718e-5" />
      </inertial>
   </link>
</xacro:macro>
```

Abb. 4.1: Ballbot Description macro

4. Simulation
Ergebnisse

TECHNISCHE UNIVERSITÄT DARMSTADT

Simulations-Ergebnisse:

- Regelung stabilisiert Ballbot
- Bestätigung der gewünschten Balleigenschaften (hoher Reibwert, große Steifigkeit)
- Simulation der Sensoren durch Einbindung von Plugins
- Drehzahl-Begrenzung der Motoren verhindert stabiles Regelverhalten

4. Simulation
Auswirkung der Drehzahlbegrenzung

Abb. 4.2: Simulation mit begrenzter Drehzahl der Motoren

Abb. 4.2: Simulation mit begrenzter Drehzahl der Motoren

4. Simulation
Herausforderungen

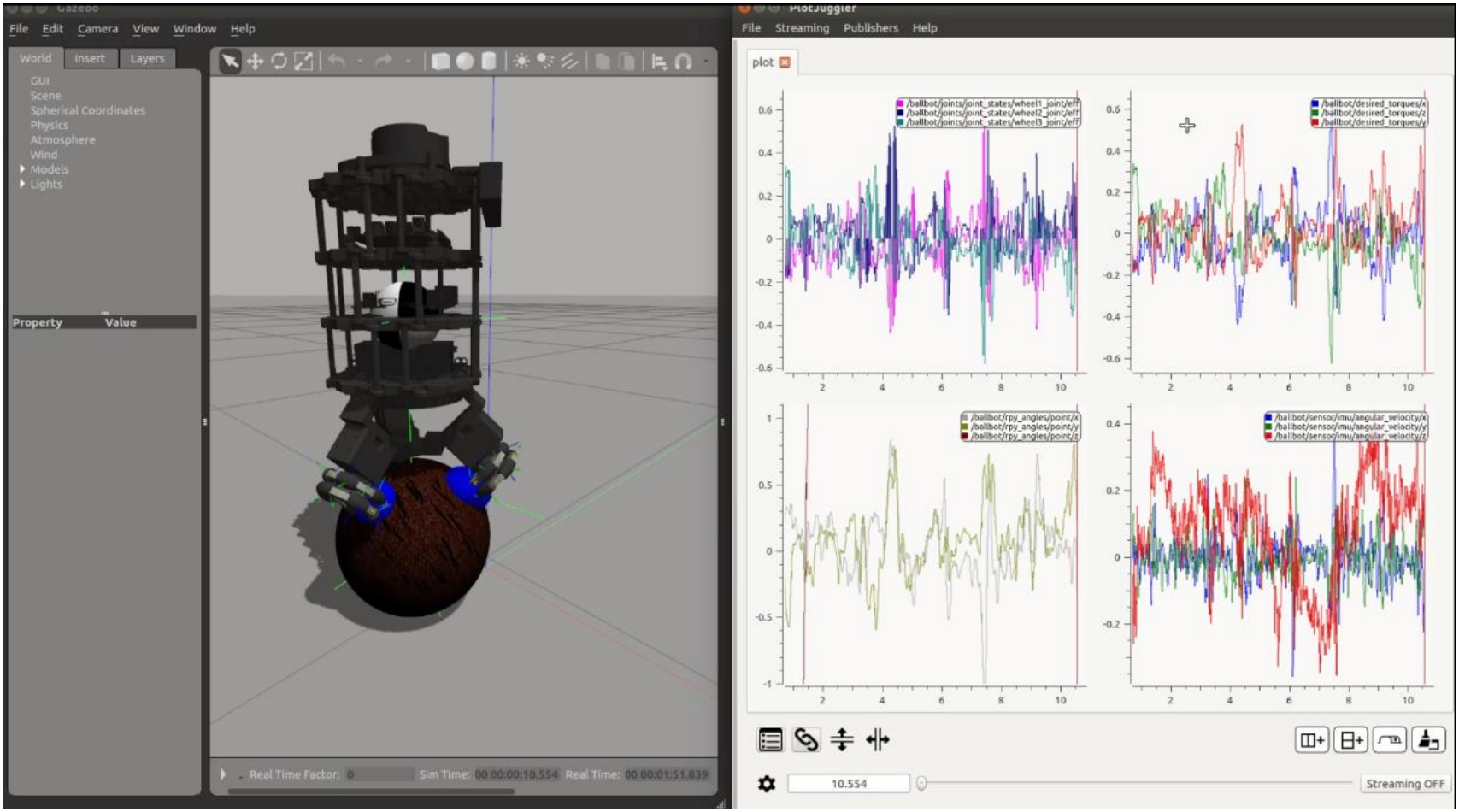


Abb. 4.4: Gazebo in Verbindung mit PlotJuggler

4. Simulation
Gazebo und RVIZ

TECHNISCHE UNIVERSITÄT DARMSTADT

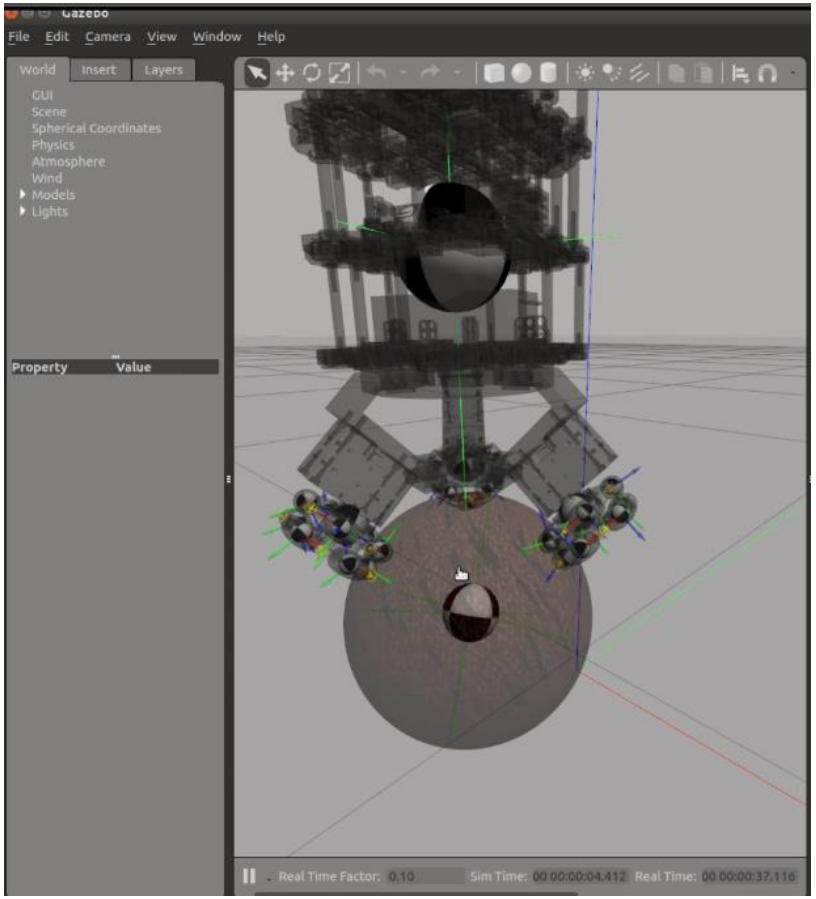


Abb. 4.5: Gazebo Links und Joints

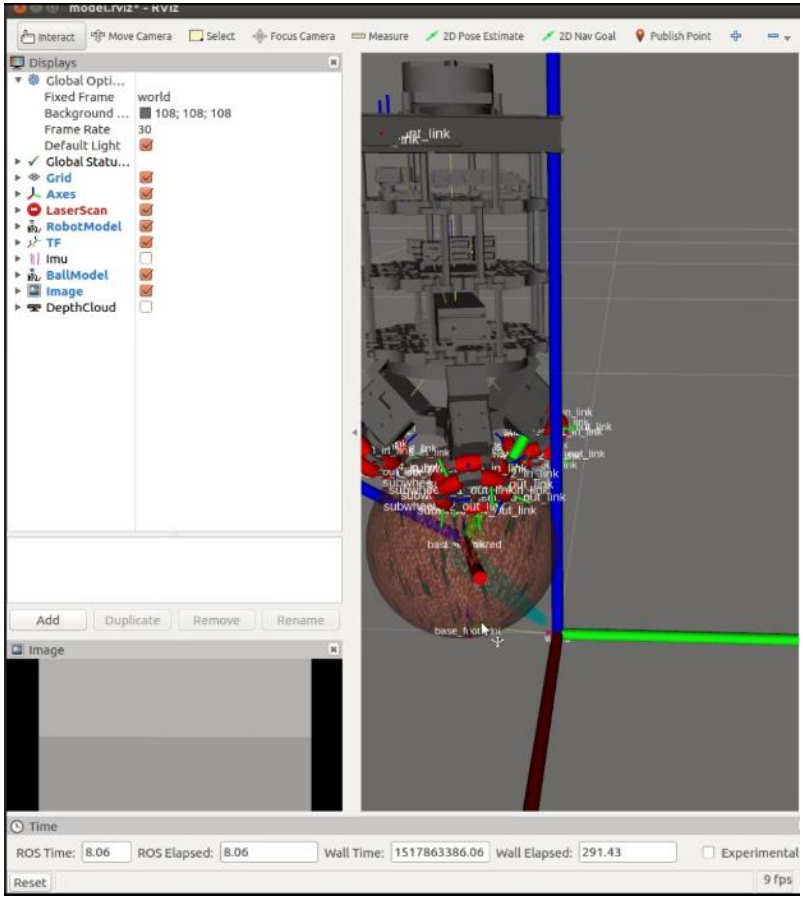


Abb. 4.6: RVIZ

5. Zusammenfassung und Ausblick

TECHNISCHE UNIVERSITÄT DARMSTADT

Was wir erreicht haben:

- Stabile mechanische Konstruktion
- Regelung und Simulation mit Simulink
- 3D Simulation mit Gazebo
- Stabiles Regelverhalten

Weiterführende Arbeiten:

- Ball: höhere Steifigkeit sowie größerer Reibkoeffizient
- Motoren: Motoren mit größerer maximalen Drehzahl
- IMU: Leistungsfähigere Hardware einbauen (Rauschminderung)
- Ball-Arrester: Stabilisierung des Gesamtsystems
- Modellbildung: 3D-Regelung (Berücksichtigt Verkopplungen)

Ab hier zusatzfolien

1. Konstruktion
Ergebnis

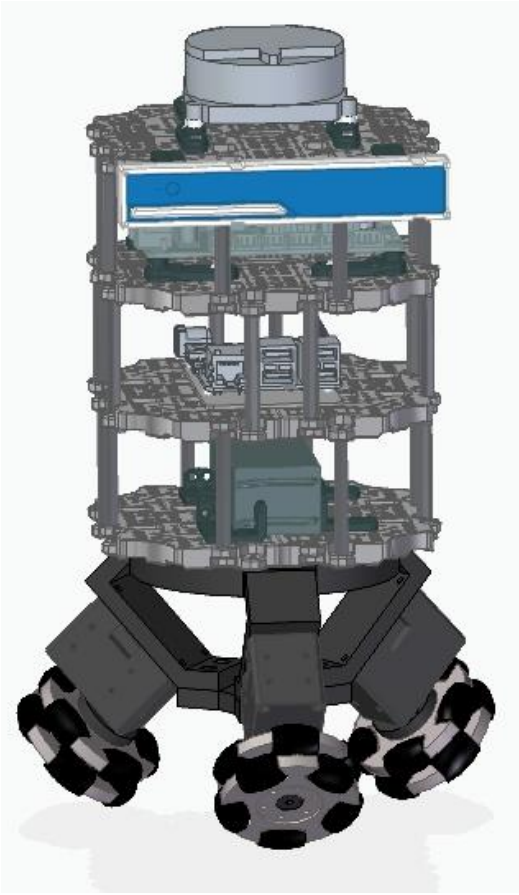


Abb. 1.8: Mechanischer Endaufbau

Vorteile:

- Stabiles Regelverhalten
- Schlupf konnte reduziert werden
- Stabiler Aufbau
- Optimale Sensorplatzierung

1. Konstruktion
Oberbau - Hardware

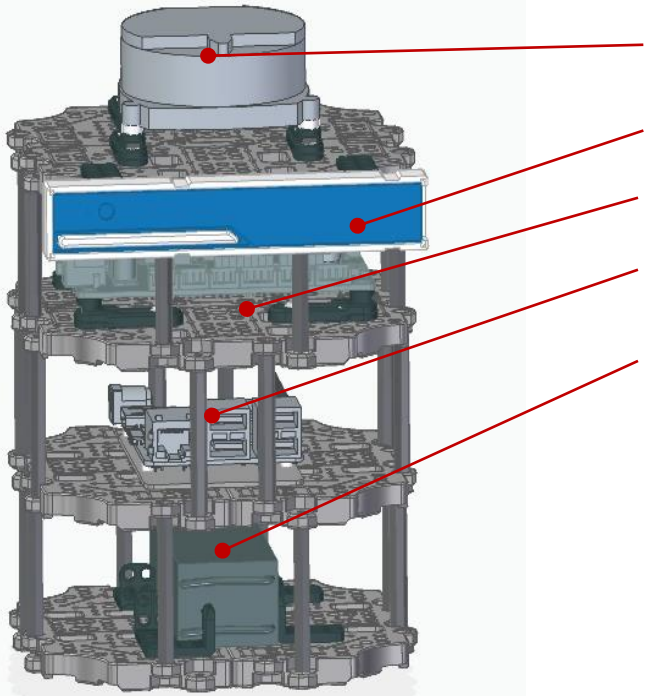


Abb. 1.9: Oberbau mit Hardware

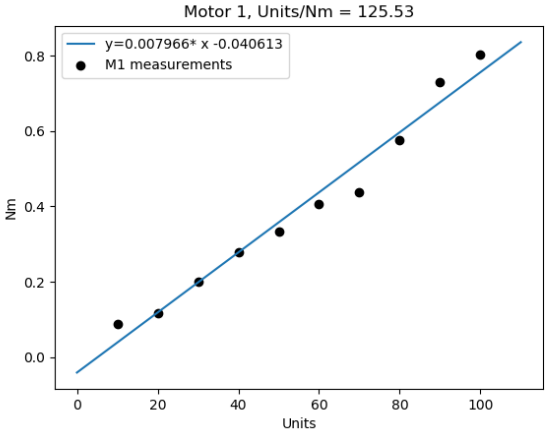
Vergebaute Hardware:

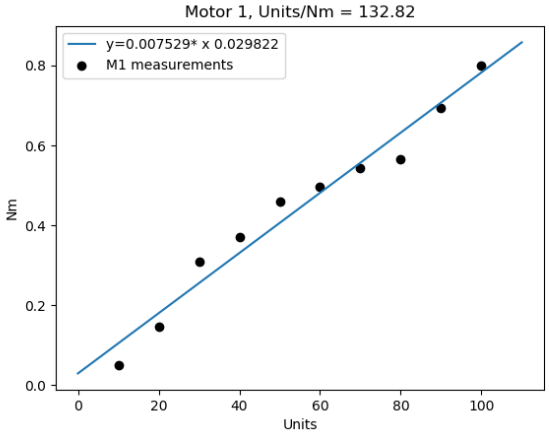
- Lidar: LDS-01 (Lokalisierung)
- Kamera: Intel Realsense R200 (Lokalisierung)
- OpenCR-Board: IMU MPU-9250 (Stabilisierung)
- Up-Board (ROS)
- Batterie: LI-PO 11.1V 1800mAh

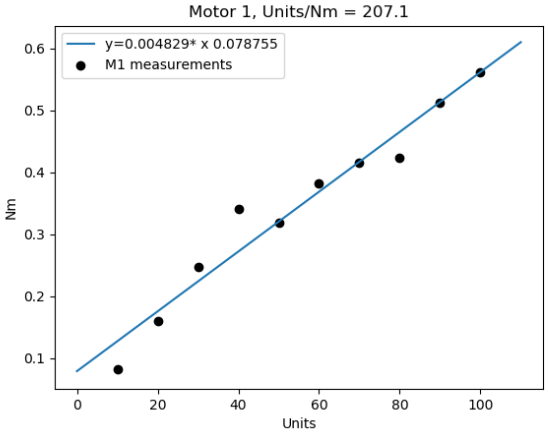
Eigenschaften:

- Schwerpunkt: 236 mm
- Durchmesser: 140 mm
- Höhe Körper: 336 mm
- Gewicht: 1731 g

1. Konstruktion
Berücksichtigung der jeweiligen Motoreigenschaften:





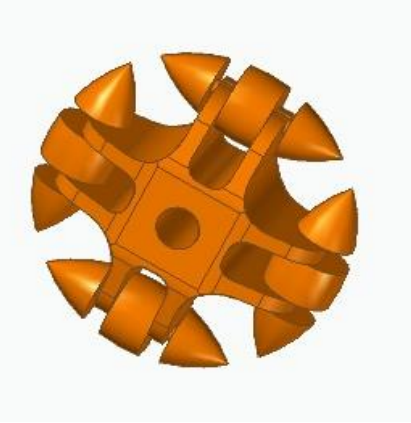


<u>Motor 1:</u>
- 125,53 Units/Nm
- Offset:

<u>Motor 2:</u>
- 132,82 Units/Nm
- Offset:

<u>Motor 3:</u>
- 207,1 Units/Nm
- Offset:

1. Konstruktion
Räder



1. Konstruktion
Anbindung der Motoren:

<u>Servomotoren Dynamixel XM430 W350T:</u>

- Drehmoment: 3,8NM bei 11.1V
- No Load Speed: 43rpm
- Kommunikation: RS485-Protokoll
- Positions-Regelung
- Drehzahl-Regelung
- Drehmoment-Regelung

<u>Nachteile:</u>

- Schmales Drehzahlband
- Ungenaue Drehmomentregelung für einmalige Drehmomentbefehle

1. Konstruktion
Design Unterbau

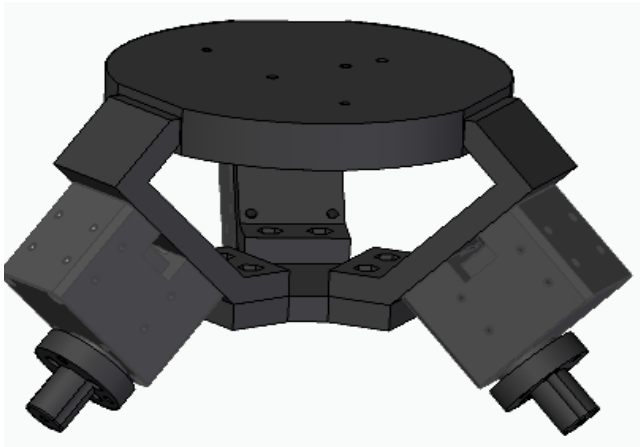


Abb. 1.11: Unterbau Design

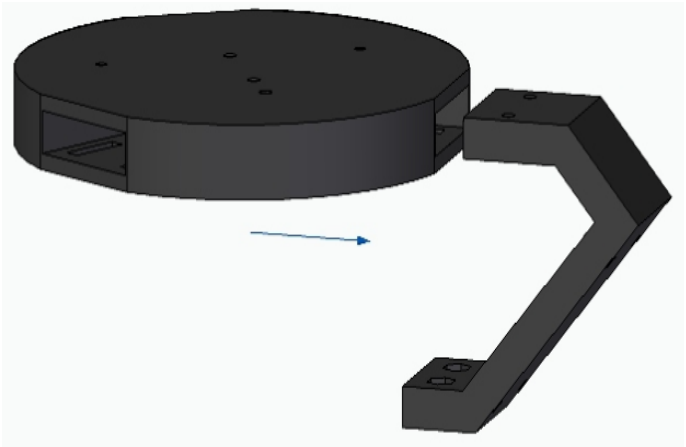


Abb. 1.12: Teleskopsystem

Eigenschaften:

- Stabile und zugleich flexible Konstruktion
- Durchmesser: 140 mm
- Auslegbar für Bälle mit Durchmessern zwischen 150 mm – 190 mm

4. Implementierung
Anpassungen

TECHNISCHE UNIVERSITÄT DARMSTADT

Drehmoment-Offset:

Grund: Reibung zwischen Räder/Ball bei Modellbildung vernachlässigt

1. Konstruktion
Unterbau - Layout

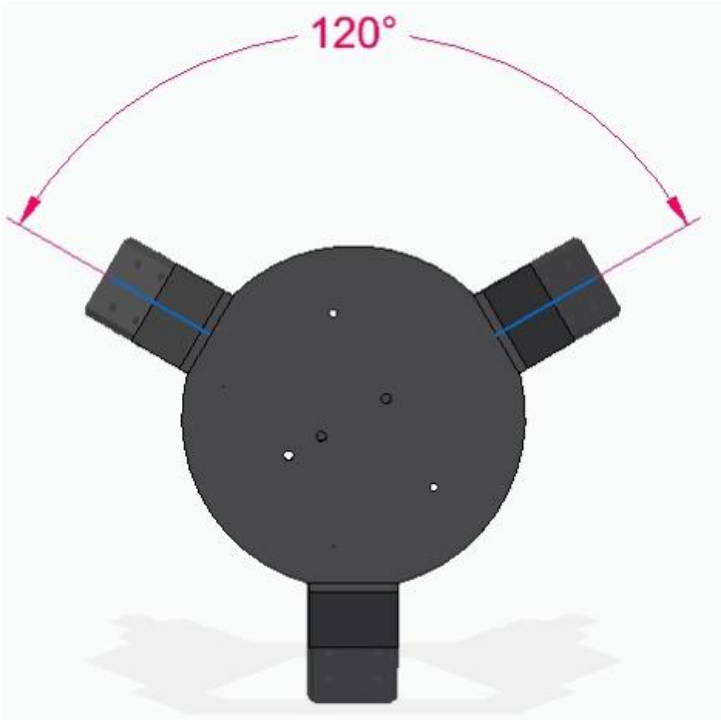


Abb. 1.10: Draufsicht Unterbau

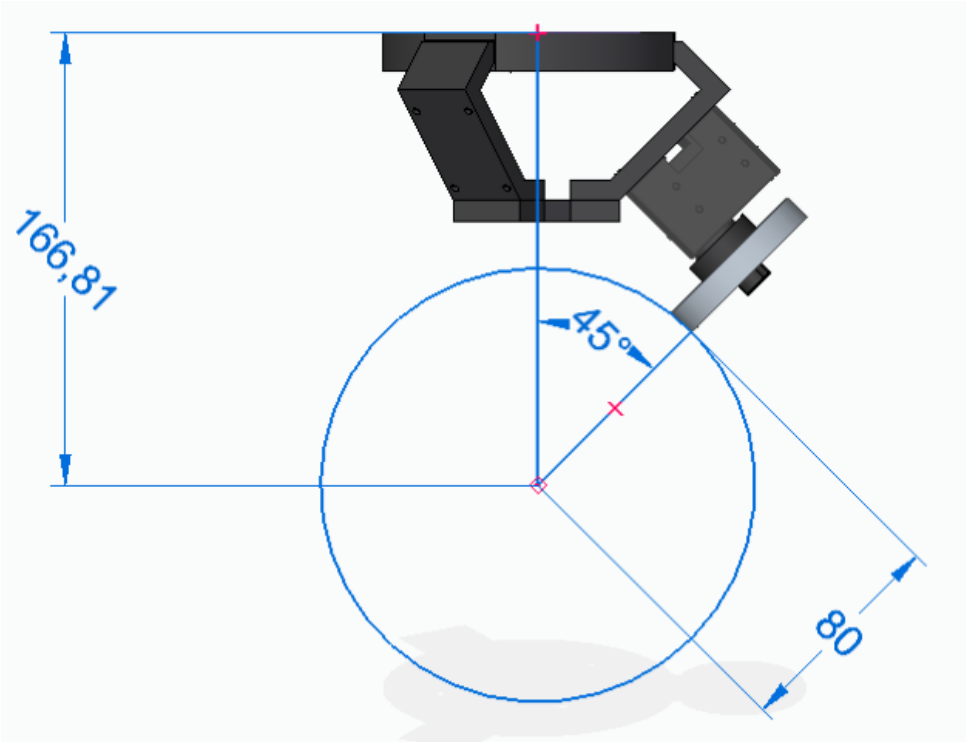


Abb. 1.11: Seitenansicht Unterbau

4. Simulation
Auswirkung der Drehzahlbegrenzung