

# Arbeit zum *Praktikum Mess- und Regelungstechnik* Sommersemester 2022

Simon Klüpfel, Lukas Zeller  
Robotik und Telematik  
Universität Würzburg  
Am Hubland, D-97074 Würzburg  
lukas.zeller@stud.uni-wuerzburg.de  
simon.kluepfel@stud.uni-wuerzburg.de

Würzburg, 24.08.2022

## 1 Einleitungzum Volksbot-Roboter

In dieser Arbeit befassen wir uns mit dem Robot Operating System, kurz *ROS*, und der Verwendung dessen auf einem einfachen Roboter, dem *Volksbot*.

Der verwendete Roboter ist der *RT3-2* mit zwei passiven Rädern und der *RT-3* mit einem passiven Rad. Entwickelt wurden diese vom *Fraunhofer Institut IAIS* aus Sankt Augustin.

Die technischen Daten sind wie folgt:

|                          |                           |
|--------------------------|---------------------------|
| Abmessungen              | 580x520x315mm (L x B x H) |
| Gewicht                  | 17kg                      |
| Raddurchmesser           | 260x85mm (aktive Räder)   |
|                          | 200mm (passive Räder)     |
| Maximale Geschwindigkeit | 2,2 $\frac{m}{s}$         |
| Maximale Zuladung        | 25kg                      |

Auszug aus <https://www.volksbot.de/rt3-de.php>

Auf dem Roboter ist ein Laserscanner sowie Hardware zur Verbindung mit dem verwendeten Laptop installiert. Außerdem lässt sich der Roboter über einen Joystick, ähnlich eines Gamecontrollers, manuell steuern. Wir verwenden sowohl vorgegebene ROS-Nodes, die uns von Institut für Robotik und Telematik zur Verfügung gestellt wurden, als auch angepasste Nodes die wir selbst erstellt beziehungsweise geändert haben.

## 2 Kartierung mit GMapping

Wir starten zuerst die Node zur Kommunikation mit dem Roboter:

```
roslaunch volksbot messtechnikpraktikum.launch
```

Um den Roboter zu steuern mit dem Controller benötigen wir die zugehörige Node:

```
roslaunch volksbot localjoystick.launch
```

Nun erstellen wir eine *rosvbag*, die alle Topics des Roboters aufzeichnet:

```
rosvbag record -a
```

Um nun eine Karte erstellen zu können, müssen wir die aufgenommene *bag* abspielen. Wichtig hier ist es den Parameter

```
rosvparam set use_sim_time true
```

zu setzen um die Zeit aus dem *clock*-Topic zu nutzen anstelle der globalen Systemzeit. Nun kann man aus den Laserscanner-Daten, also hauptsächlich dem *LMS*-Topic, die Map erstellen. Hierzu starten wir das GMapping-Tool via

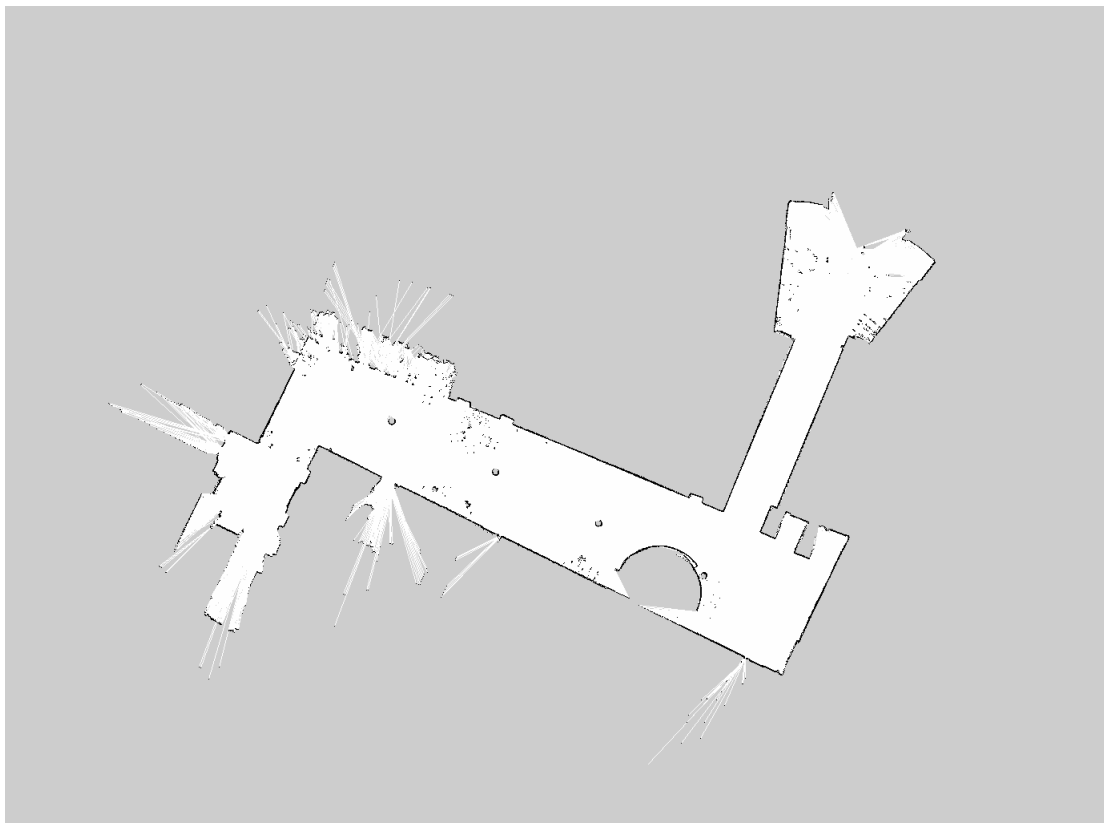
```
roslaunch volksbot messtechnikgmapping.launch
```

Die Bag kann dann abgespielt werden mit

```
rosvbag play filename.bag --clock
```

Der *clock*-Befehl am Ende ist wichtig um die Aufnahmen der *rosvbag* mit einem Zeitstempel zu versehen damit der Roboter sie nicht verwirft. Startet man nun RViz kann man über das *map*-Topic den Aufbau der Map betrachten. Um diese zu speichern startet man eine *ROS-node* die das *map*-Topic in eine Datei speichert.

Abbildung 1: Aufgezeichnete Karte des Informatik-Untergeschosses



### **3 AMCL-Lokalisierung**

todo: vergleich AMCL-Odometrie

### **4 Pfadverfolgung**

Pfade vergleichen, was gut, was schlecht, eventuell Code-Kommentare

### **5 Auswertung**

todo: Testen Sie den Regler mit dem aufgenommenen Pfad aus dem Informatikgebäude. Vergleichen Sie das Verhalten, wenn einmal die Odometrie und einmal die von AMCL bestimmten Positionen als Messwerte für die Regelung verwendet werden.

also: plots, kommentare, was ist besser, was schlechter?