Tutorium Robotik und autonome Systeme

Einführung ROS und Turtlebot 3

Niklas Noack[1], Jan-Luca Regenhardt[1]

[1]Autonomous Mobile Robotics Lab
Hochschule für angewandte Wissenschaft und Kunst

12. November 2024

Wiederholung

Ergänzung
- rosrun -parameter
- roslaunch

Gazebo
- Turtlebot3
- Turtlebot3 steuern

RViz
- RViz Starten
- gmapping
- Navigation

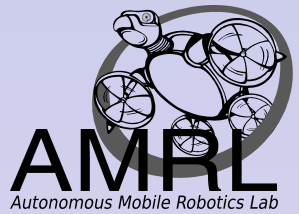
Catkin
- Arbeitsumgebung
- Pakete

Beispielprojekt
- Paket (Knoten) erstellen
- package.xml
- CMakesLists.txt
- Subscriber Node

Nützliche Links

Inhaltsverzeichnis

- Wiederholung
- Ergänzung
  - rosrun -parameter
  - roslaunch
- Gazebo
  - Turtlebot3
  - Turtlebot3 steuern
- RViz
  - RViz Starten
  - gmapping
  - Navigation
- Catkin
  - Arbeitsumgebung
  - Pakete
- Beispielprojekt
  - Paket (Knoten) erstellen
  - package.xml
  - CMakesLists.txt
  - Subscriber Node
- Nützliche Links



N. Noack,
J.-L. Regenhardt

Wiederholung
Ergänzung
rosrun -parameter
roslaunch
Gazebo
Turtlebot3
Turtlebot3 steuern
RViz
RViz Starten
gmapping
Navigation
Catkin
Arbeitsumgebung
Pakete
Beispielprojekt
Paket (Knoten) erstellen
package.xml
CMakesLists.txt
Subscriber Node
Nützliche Links

Wiederholung 1/2

- **Nachrichten** werden durch Netzwerke von Knoten ***Nodes*** ausgetauscht
- Ein ***Master*** koordiniert Kommunikation zwischen Knoten
- Asynchrone Nachrichten werden über ***Topics*** (***Publisher*** und ***Subscriber***) gesendet
- Synchrone Nachrichten werden durch ***Services*** (***Request*** und ***Response***) gesendet

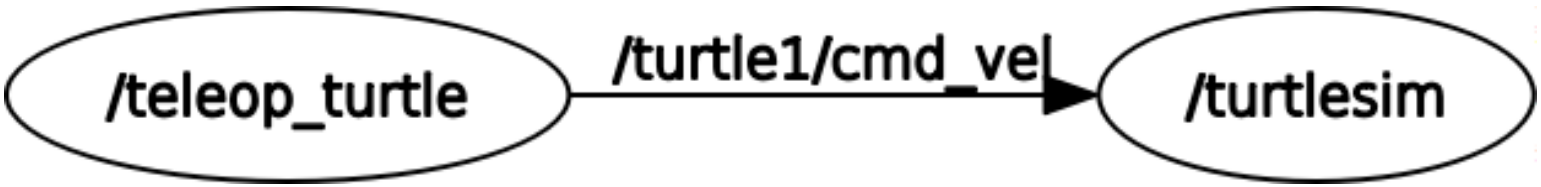


Abbildung: *Publisher*, *Topic*, *Subscriber*

Wiederholung
Ergänzung
rosrun -parameter
roslaunch
Gazebo
Turtlebot3
Turtlebot3 steuern
RViz
RViz Starten
gmapping
Navigation
Catkin
Arbeitsumgebung
Pakete
Beispielprojekt
Paket (Knoten) erstellen
package.xml
CMakesLists.txt
Subscriber Node
Nützliche Links

Wiederholung 2/2

Wichtige Befehle:

- Strg + Alt + T zum öffnen eines Terminals
- `roscore` zum starten des *Masters*
- `rosrun Paketname Knotenname` zum starten eines *nodes*
  Bsp.: `rosrun turtlesim draw_square`

Hilfe über Parameter --h; Bsp.: `rosnode --h` führt zu:

- `rosnode list` listet alle *nodes* auf
- `rosnode info` gibt Info über einen *node* aus

Befehle für Nachrichten (Topics) zwischen Nodes

- `rostopic list` listet topics auf
- `rostopic echo /topic_name` „lauscht“ einem topic
- `rostopic type /topic_name` gibt Datentyp des topics an

Autovervollständigung

- Autovervollständigung mit ↹ (Tabulator) Bsp.:
- `ros2`*x*↹ listet alle möglichen Befehle beginnend mit ros auf.
- `rosnode 2`*x*↹ listet alle möglichen Parameter auf.

rosrun Ergänzung

Parameteriesierung

```
rosrun turtlesim turtle_teleop_key _scale_linear:=0.5
```

Dabei ist:

- *turtlesim* der Paketname
- *turtle_teleop_key* der *node-/* Programmname
- *_scale_linear:=0.5* ein Parameter inkl. Wert

Beispiel

turtlesim mit rotem Hintergrund:

```
rosrun turtlesim turtlesim_node _background_b:=0
_background_g:=0 _background_r:=255
```

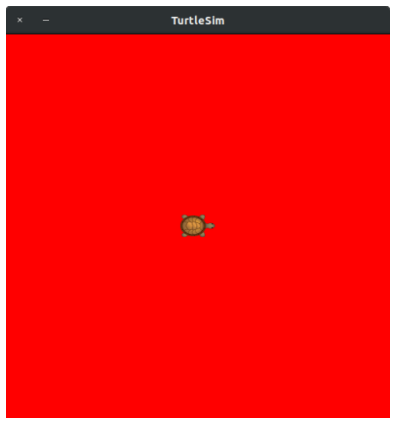


Abbildung: *turtlesim* roter Hintergrund

Wiederholung
Ergänzung
rosrun -parameter
roslaunch
Gazebo
Turtlebot3
Turtlebot3 steuern
RViz
RViz Starten
gmapping
Navigation
Catkin
Arbeitsumgebung
Pakete
Beispielprojekt
Paket (Knoten) erstellen
package.xml
CMakesLists.txt
Subscriber Node
Nützliche Links

roslaunch

- Erleichtert die Arbeit mit mehreren *nodes*
- Startet automatisch *roscore* und angegebene *nodes* definiert in ***.launch**-Datei
- Beinhaltet Paketnamen, *Nodes* und Parameter in **XML**-Format

```
<launch>
        <node
                name="turtlesim_node"
                pkg="turtlesim"
                type="turtlesim_node"
                output="screen">
                <param name="background_b" value="0" />
                <param name="background_g" value="0" />
                <param name="background_r" value="255"  />
        </node>
        <node
                name="turtlesim_draw"
                pkg="turtlesim"
                type="draw_square"
                output="screen">
        </node>
</launch>
```

- `roslaunch turtlesim turtlesim.launch`

Wiederholung
Ergänzung
rosrun -parameter
roslaunch
Gazebo
Turtlebot3
Turtlebot3 steuern
RViz
RViz Starten
gmapping
Navigation
Catkin
Arbeitsumgebung
Pakete
Beispielprojekt
Paket (Knoten) erstellen
package.xml
CMakesLists.txt
Subscriber Node
Nützliche Links

Gazebo Simulator

- 3D-Simulator für Roboter
- Physik-*Engine* im Unterbau
- anschauliche Graphik
- Graphische Bedienoberfläche
- Simuliert sämtliche Sensordaten(scan, imu, odom, ...)

Abbildung: Gazebo-Logo

Wiederholung
Ergänzung
rosrun -parameter
roslaunch
Gazebo
Turtlebot3
Turtlebot3 steuern
RViz
RViz Starten
gmapping
Navigation
Catkin
Arbeitsumgebung
Pakete
Beispielprojekt
Paket (Knoten) erstellen
package.xml
CMakesLists.txt
Subscriber Node
Nützliche Links

Simulation Turtlebot3 in Gazebo

Ausführen im Terminal

- `roslaunch turtlebot3_gazebo turtlebot3_house.launch`

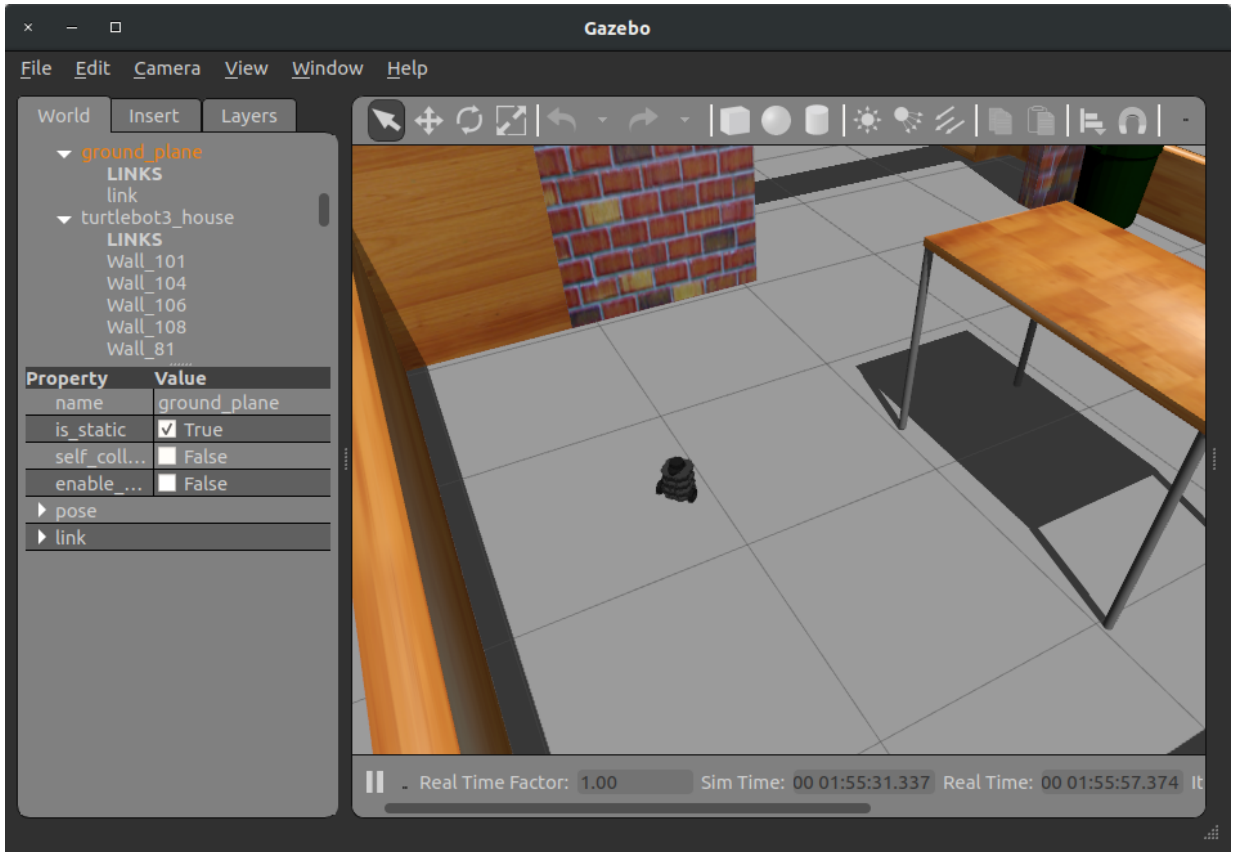


Abbildung: Gazebo-Simulator mit *Turtlebot*

AMRL
Autonomous Mobile Robotics Lab
Tutorium Robotik
N. Noack,
J.-L. Regenhardt

Wiederholung
Ergänzung
rosrun -parameter
roslaunch
Gazebo
Turtlebot3
Turtlebot3 steuern
RViz
RViz Starten
gmapping
Navigation
Catkin
Arbeitsumgebung
Pakete
Beispielprojekt
Paket (Knoten) erstellen
package.xml
CMakesLists.txt
Subscriber Node
Nützliche Links

Turtlebot3 in Gazebo steuern

Ausführen im Terminal 2

- `roslaunch turtlebot3_teleop turtlebot3_teleop_key`
- Mit den Tasten W,A,S und D Turtlebot3 steuern

Ausführen im Terminal 3

- `rostopic echo /cmd_vel`
- `rostopic echo /scan` - Unübersichtlich :(

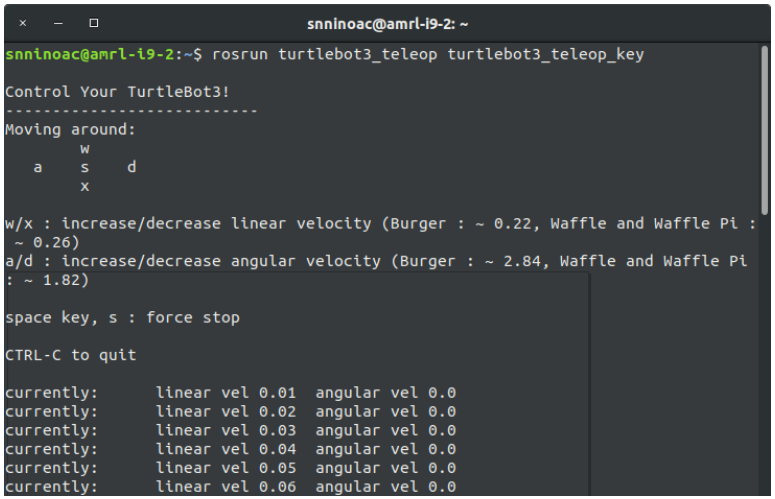


Abbildung: Teleop

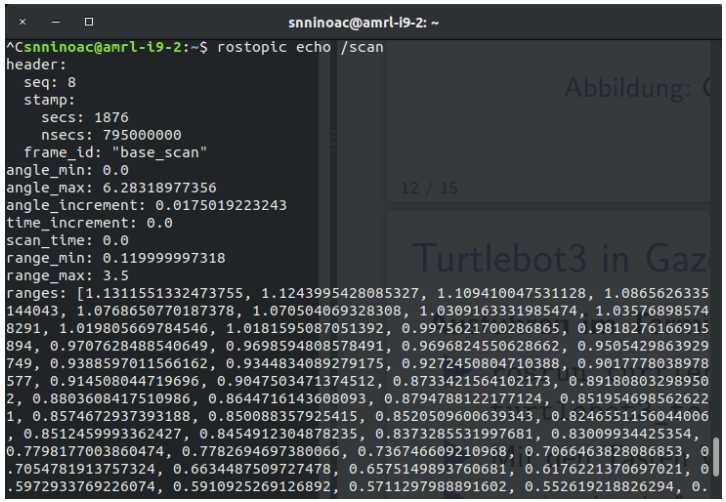


Abbildung: echo /scan

ROS Vizualisation

In **RViz** werden Nachrichten (Sensordaten) visualisiert.

- Karten
- *Laserscanner*
- Posen
- Kamerabilder
- Der Roboter selbst
- Pfade

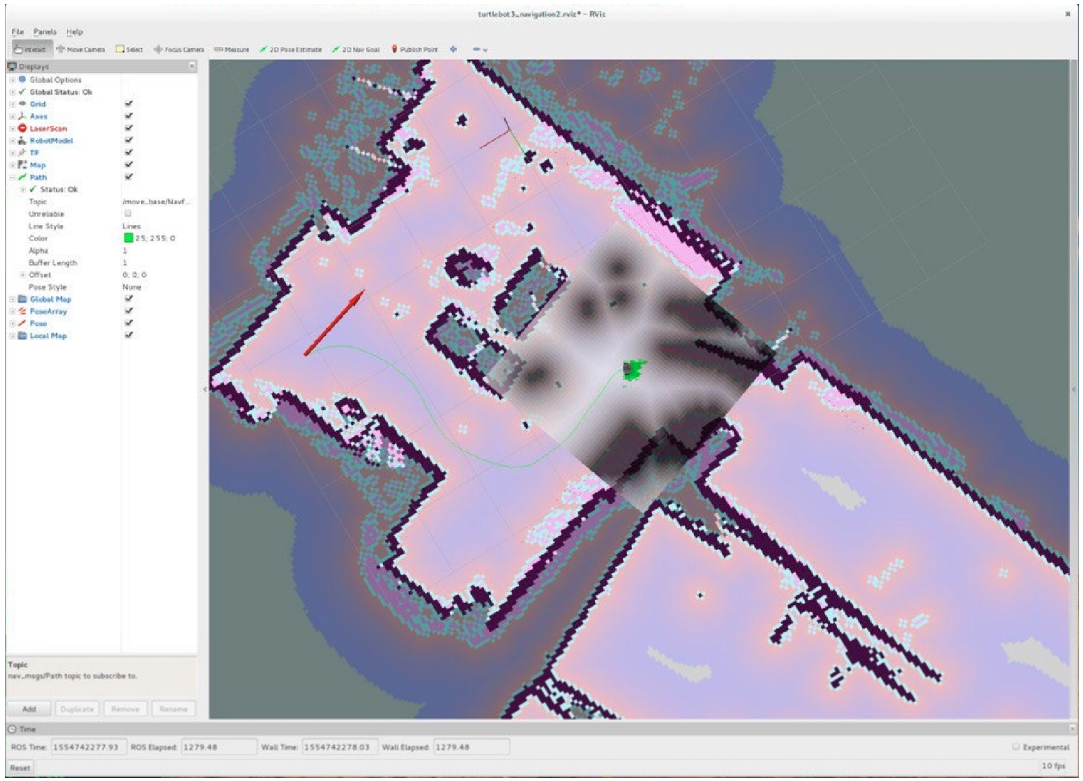


Abbildung: RViz mit *gmapping* und Pfadplanung

N. Noack,
J.-L. Regenhardt

Wiederholung
Ergänzung
rosrun -parameter
roslaunch
Gazebo
Turtlebot3
Turtlebot3 steuern
RViz
RViz Starten
gmapping
Navigation
Catkin
Arbeitsumgebung
Pakete
Beispielprojekt
Paket (Knoten) erstellen
package.xml
CMakesLists.txt
Subscriber Node
Nützliche Links

RViz starten

RViz über *.launch-Datei für turtlebot_gazebo über **Terminal 3** starten:

- `roslaunch turtlebot3_gazebo turtlebot3_gazebo_rviz.launch`
- Über **Terminal 2** Turtlebot steuern

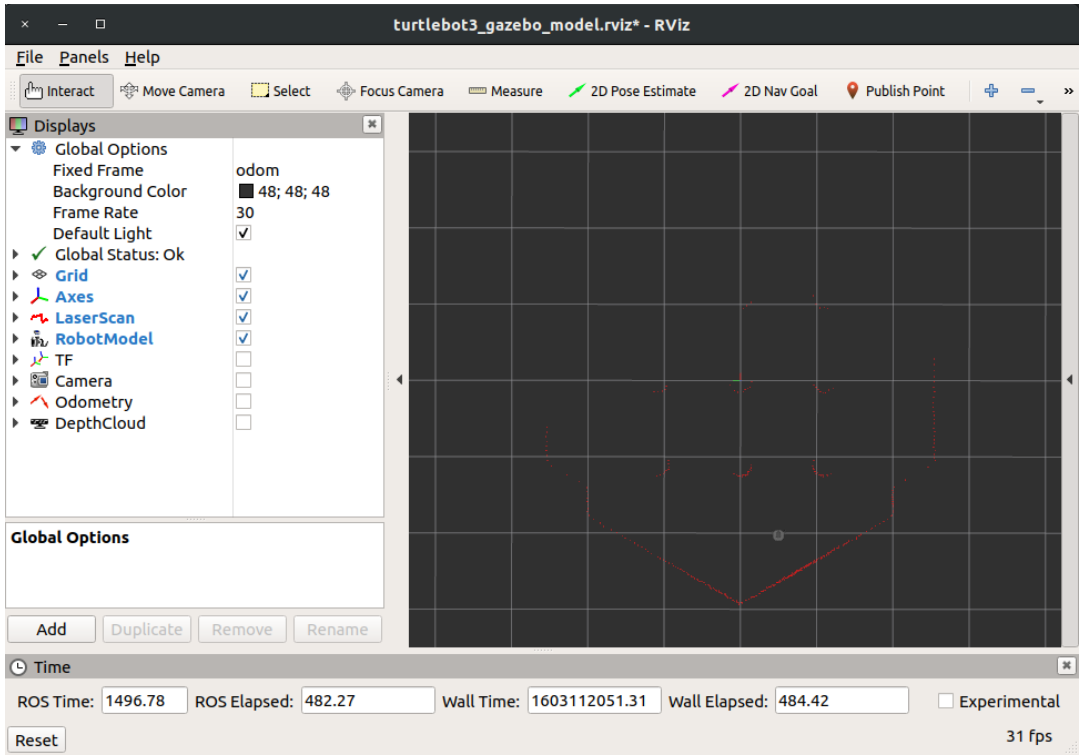


Abbildung: RViz - Gabezo

Wiederholung
Ergänzung
rosrun -parameter
roslaunch
Gazebo
Turtlebot3
Turtlebot3 steuern
RViz
RViz Starten
gmapping
Navigation
Catkin
Arbeitsumgebung
Pakete
Beispielprojekt
Paket (Knoten) erstellen
package.xml
CMakesLists.txt
Subscriber Node
Nützliche Links

Beispiele 1/2

SLAM - GMapping:

Gazebo, RViz sowie alle Terminals schließen.

- Strg+Alt+T um Terminal zu öffnen.
- `roslaunch turtlebot3_gazebo turtlebot3_house.launch`
- `roslaunch turtlebot3_slam turtlebot3_slam.launch slam_methods:=gmapping`
- `roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch`

Karte erstellen:

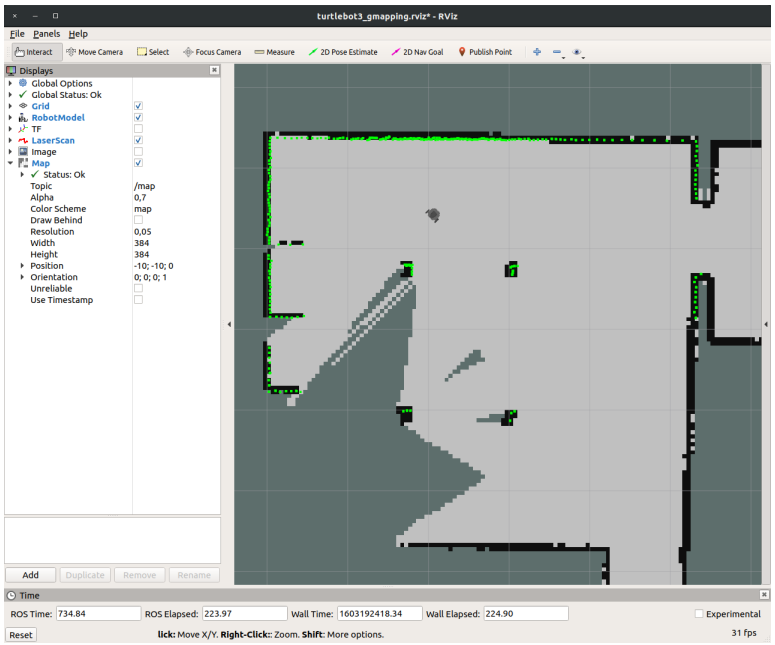


Abbildung: RViz - Gmapping

Karte speichern

- Karte im Heimverzeichnis speichern:
- `rosrun map_server map_saver -f /home/amrl/map`

Beispiele 2/2

Navigation:
Gazebo, RViz sowie alle Terminals schließen.

- Strg+Alt+T um Terminal zu öffnen.
- `roslaunch turtlebot3_gazebo turtlebot3_house.launch`
- `roslaunch turtlebot3_navigation turtlebot3_navigation.launch map_file:=/home/amrl/map.yaml`

In RViz

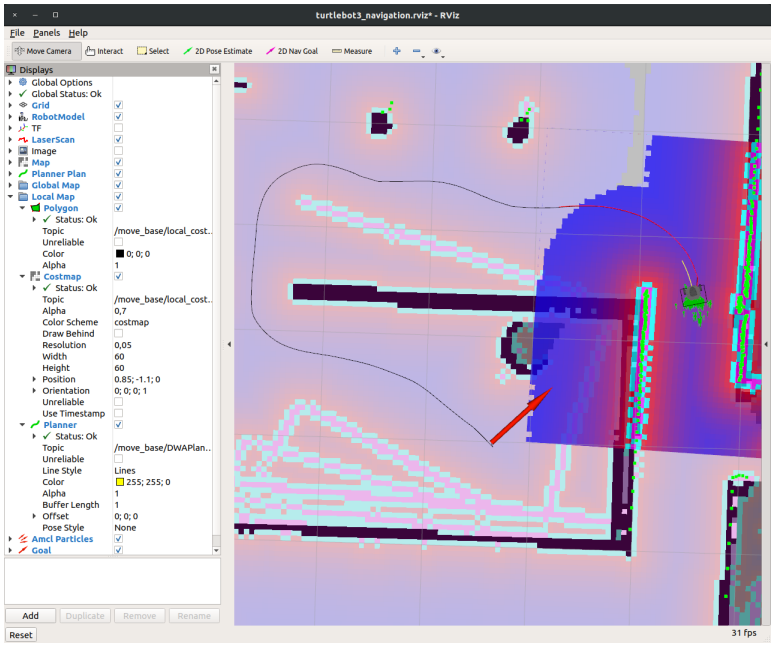


Abbildung: RViz - Navigation

- Initialpose bestimmen:
- Auf *2D Pose Estimate* klicken und auf der Karte die Pose bestimmen.
- Wenn die Punktwolke des *Laserscanners* mit der Karte deckungsgleich sind:
- Auf *2D Nav Goal* klicken und ein Ziel festlegen.

Arbeitsumgebung

Der *Catkin-Workspace* (*catkin_ws*) stellt den Arbeitsbereich dar. Dieser befindet sich im pers. Heimverzeichnis: */home/benutzer/* bzw. *~/*

Ordnerstruktur des Arbeitsverzeichnisses:

- Catkin *Workspace*
- Besteht aus:
  - *build*
  - *devel*
  - *src*

Hauptaugenmerk gilt dem src-Ordner, darin befinden sich Pakete zum kompilieren.

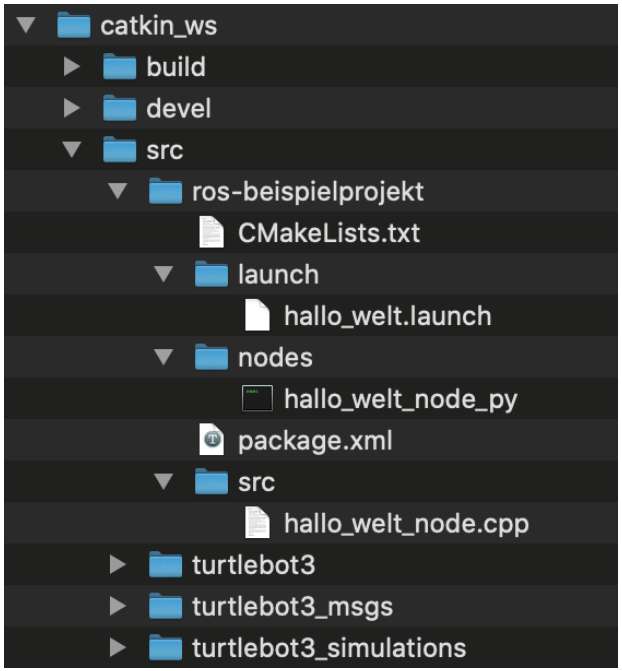


Abbildung: Catkin *Workspace*

Wiederholung
Ergänzung
rosrun -parameter
roslaunch
Gazebo
Turtlebot3
Turtlebot3 steuern
RViz
RViz Starten
gmapping
Navigation
Catkin
Arbeitsumgebung
Pakete
Beispielprojekt
Paket (Knoten) erstellen
package.xml
CMakesLists.txt
Subscriber Node
Nützliche Links

ROS-Paket

- Stellt kleinstmögliche Organisationseinheit dar
- vergleichbar mit Linux *Software*-Paketen
- Beinhaltet:
  - *Nodes*
  - Programme (,Skripte. etc.)
  - Bibliotheken
  - Konfigurationsdateien
  - u.v.m.

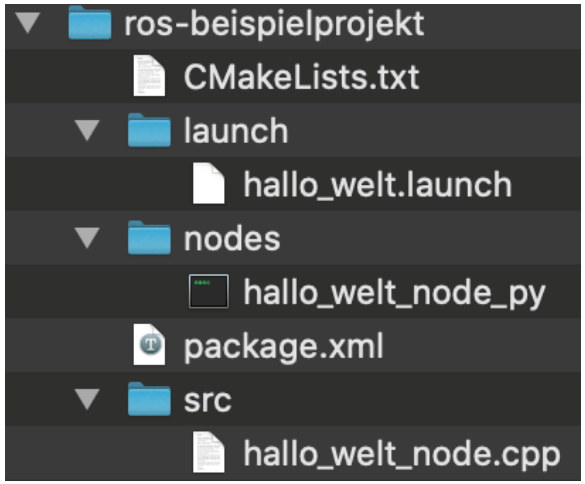


Abbildung: Paket: Beispielprojekt

Dateien und Ordner im Paketeverzeichniss (Auszug):

- CMakeLists.txt - Bauanleitung für *CMake*
- package.xml - *Software*abhängigkeiten, *Copyright*, Autor, Lizenzen, etc.
- *launch* beinhaltet *.launch-Dateien
- *nodes* beinhaltet *.py-Scripte mit Node-Funktion
- *src* beinhaltet *.cpp-Dateien und *.py-Module

Beispielprojekt

Idee: Fliehkraft des Turtlebots in der Kurve bestimmen.

$$\vec{F} = \frac{\vec{v}^2}{r} m \tag{1}$$

0. Vorüberlegungen

- translatorische Geschwindigkeit v_x in x-Richtung
- Radius r der Kurve
- *Topic /cmd_vel* liefert Geschwindigkeiten v_x und Winkelgeschwindigkeit der Gierachse ω.
- `rostopic info /cmd_vel`
- Nachricht hat Typ: *geometry_msgs/Twist*
- `rosmsg show geometry_msgs/Twist`

Wir programmieren einen *Subscriber* und abonnieren /cmd_vel!

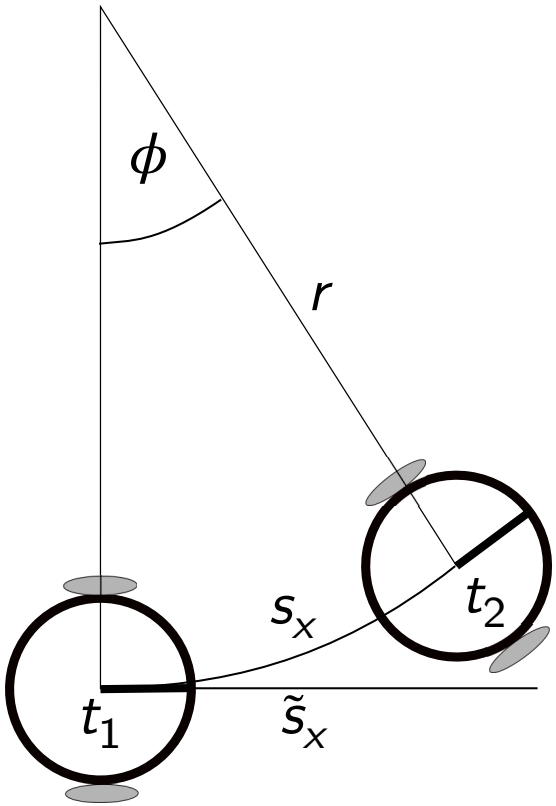


Abbildung: Turtlebot in der Kurve

$$\Delta t \to 0 \tag{2}$$

$$s_x = \Delta t v_x \tag{3}$$

$$\tilde{s}_x = s_x \tag{4}$$

$$\phi = \omega \Delta t \tag{5}$$

$$r = \frac{\tilde{s}_x}{sin\phi} \tag{6}$$

$$a = \frac{\vec{v_x}^2}{r} \tag{7}$$

Wiederholung
Ergänzung
rosrun -parameter
roslaunch
Gazebo
Turtlebot3
Turtlebot3 steuern
RViz
RViz Starten
gmapping
Navigation
Catkin
Arbeitsumgebung
Pakete
Beispielprojekt
Paket (Knoten) erstellen
package.xml
CMakesLists.txt
Subscriber Node
Nützliche Links

1. Projekt erstellen

Paket im Catkin-*Workspace* erstellen

- Strg+Alt+T zum öffnen eine Terminals
- `cd /home/benutzer/catkin_ws/src`
- `catkin_create_pkg --rosdistro melodic -s roscpp geometry_msgs std_msgs -l BSD -m maintainer_name -a author_name fliehkraft_node`
- `cd /home/benutzer/catkin_ws`
- `catkin_make`

Wiederholung
Ergänzung
rosrun -parameter
roslaunch
Gazebo
Turtlebot3
Turtlebot3 steuern
RViz
RViz Starten
gmapping
Navigation
Catkin
Arbeitsumgebung
Pakete
Beispielprojekt
Paket (Knoten) erstellen
package.xml
CMakesLists.txt
Subscriber Node
Nützliche Links

2. package.xml anpassen 1/2

```
gedit
~/catkin_ws/src/fliehkraft_node/package.xml
```

description tag

```
<name>fliehkraft_node</name>
<version>0.0.0</version>
<description>The fliehkraft_node package</description>
```

maintainer tags

```
<maintainer email="niklas.noack[at]hawk.de">
    Niklas
</maintainer>
```

2. package.xml anpassen 2/2

license tags

```
<license>BSD</license>
```

dependencies tag

```
<buildtool_depend>catkin</buildtool_depend>

<build_depend>roscpp</build_depend>
<build_depend>geometry_msgs</build_depend>
<build_depend>std_msgs</build_depend>

<build_export_depend>roscpp</build_export_depend>
<build_export_depend>geometry_msgs</
    build_export_depend>
<build_export_depend>std_msgs</build_export_depend>

<exec_depend>roscpp</exec_depend>
<exec_depend>geometry_msgs</exec_depend>
<exec_depend>std_msgs</exec_depend>
```


Wiederholung
Ergänzung
rosrun -parameter
roslaunch
Gazebo
Turtlebot3
Turtlebot3 steuern
RViz
RViz Starten
gmapping
Navigation
Catkin
Arbeitsumgebung
Pakete
Beispielprojekt
Paket (Knoten) erstellen
package.xml
CMakesLists.txt
Subscriber Node
Nützliche Links

3. CMakesLists.txt anpassen

```
gedit
~/catkin_ws/src/fliehkraft_node/CMakeLists.txt
```

Name des Nodes

```
project(fliehkraft_node)
```

Abhängigkeiten

```
find_package(catkin REQUIRED COMPONENTS
             roscpp
             std_msgs
             geometry_msgs)
```

Hinzufügen (bekanntmachen) von cpp/hpp Dateien

```
add_executable(fliehkraft_node src/fliehkraft_node.cpp
    )
target_link_libraries(${PROJECT_NAME} ${
    catkin_LIBRARIES})
```


Wiederholung
Ergänzung
rosrun -parameter
roslaunch
Gazebo
Turtlebot3
Turtlebot3 steuern
RViz
RViz Starten
gmapping
Navigation
Catkin
Arbeitsumgebung
Pakete
Beispielprojekt
Paket (Knoten) erstellen
package.xml
CMakesLists.txt
Subscriber Node
Nützliche Links

4. *Subscriber-Node* schreiben 1/3

Ordner und .cpp Datei erstellen

- `cd /home/benutzer/catkin_ws/src/fliehkraft_node`
- `mkdir src`
- `cd src`
- `touch fliehkraft_node.cpp`

fliehkraft_node.cpp öffnen

- `gedit fliehkraft_node.cpp`

```
//Wichtige Header einbinden, aus Vorüberlegungen
#include "ros/ros.h"
#include "geometry_msgs/Twist.h"
#include "math.h"
```

4. *Subscriber-Node* schreiben 2/3

```
//Beginn der Hauptmethode
int main(int argc, char **argv)
{

//Bekanntmachen des Nodes im ROS Master
//unter dem Namen fliehkraft
  ros::init(argc, argv, "fliehkraft");

//Ein Nodehandle Objekt anlegen.
  ros::NodeHandle n;

//abonniere cmd_vel, Puffergrösse 1000, Wenn Nachricht
    , rufe fliehkraftCallback auf
  ros::Subscriber sub = n.subscribe("cmd_vel",
        1000, fliehkraftCallback);

  ros::spin();

  return 0;
}
```


Wiederholung
Ergänzung
rosrun -parameter
roslaunch
Gazebo
Turtlebot3
Turtlebot3 steuern
RViz
RViz Starten
gmapping
Navigation
Catkin
Arbeitsumgebung
Pakete
Beispielprojekt
Paket (Knoten) erstellen
package.xml
CMakesLists.txt
Subscriber Node
Nützliche Links

4. *Subscriber-Node* schreiben 3/3

Die Funktion *fliehkraftCallback* wird bei jedem eintreffen einer Nachricht aufgerufen.

```
//Funktion mit der übergebenen Variable msg vom Typ
    geometry_msgs::Twist
void fliehkraftCallback(const geometry_msgs::Twist&
    msg)
{
static double t;
//berechne Zeitstempel t_1
t = ros::Time::now().sec+ros::Time::now().nsec*10e-10;

//Ausgabe einiger Werte in das Terminal
ROS_INFO("v_x:[%f],w_z:[%f],t:[%f]",msg.linear.x,msg.
    angular.z,t);

// Hier eure Berechnung!
}
```

Bei Problemen vergleicht eure Dateien mit der Lösung:
http://amrl.hawk.de:3373/ros_projekte_regenhardt/fliehkraft_node

Wiederholung
Ergänzung
rosrun -parameter
roslaunch
Gazebo
Turtlebot3
Turtlebot3 steuern
RViz
RViz Starten
gmapping
Navigation
Catkin
Arbeitsumgebung
Pakete
Beispielprojekt
Paket (Knoten) erstellen
package.xml
CMakesLists.txt
Subscriber Node
Nützliche Links

Kompilieren und Ausführen

0. Kompilieren

Strg+Alt+T

```
cd ~/catkin_ws
catkin_make
```

1. Roscore

Strg+Alt+T

```
roscore
```

2. cmd_vel publizieren

Strg+Alt+T

```
rosrun turtlebot3_teleop turtlebot3_teleop_key
```

3. fliehkraft_node starten

Strg+Alt+T

```
rosrun fliehkraft_node fliehkraft_node
```


Wiederholung
Ergänzung
rosrun -parameter
roslaunch
Gazebo
Turtlebot3
Turtlebot3 steuern
RViz
RViz Starten
gmapping
Navigation
Catkin
Arbeitsumgebung
Pakete
Beispielprojekt
Paket (Knoten) erstellen
package.xml
CMakesLists.txt
Subscriber Node
Nützliche Links

Nützliche Links 1

- **ROS-Framework**

  https://www.ros.org/

- **ROS-Tutorial**

  https://wiki.ros.org/ROS/Tutorials

- **ROS-Befehle**

  https://wiki.ros.org/ROS/CommandLineTools

- **TurtleBot3**

  https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/

- **Linux *Basics***

  https://cheatography.com/davechild/cheat-sheets/linux-command-line/

Wiederholung
Ergänzung
rosrun -parameter
roslaunch
Gazebo
Turtlebot3
Turtlebot3 steuern
RViz
RViz Starten
gmapping
Navigation
Catkin
Arbeitsumgebung
Pakete
Beispielprojekt
Paket (Knoten) erstellen
package.xml
CMakesLists.txt
Subscriber Node
Nützliche Links

Nützliche Links 2

- **roslaunch**

  https://wiki.ros.org/roslaunch

- **ROS-Packet erstellen**

  http://wiki.ros.org/ROS/Tutorials/CreatingPackage

- **Publisher/Subscriber schreiben (c++)**

  http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28c%2B%2B%29

Wiederholung
Ergänzung
rosrun -parameter
roslaunch
Gazebo
Turtlebot3
Turtlebot3 steuern
RViz
RViz Starten
gmapping
Navigation
Catkin
Arbeitsumgebung
Pakete
Beispielprojekt
Paket (Knoten) erstellen
package.xml
CMakesLists.txt
Subscriber Node
Nützliche Links

Nützliche Links 3

- **gazebo**

  `https://wiki.ros.org/gazebo`
- **RViz**

  `https://wiki.ros.org/rviz`
- **gmapping**

  `https://wiki.ros.org/gmapping`
- **Catkin**

  `https://wiki.ros.org/catkin/`

Niklas Noack: niklas.noack@hawk.de
Jan-Luca Regenhardt: jan-luca.regenhardt@stud.hawk.de

Wiederholung
Ergänzung
rosrun -parameter
roslaunch
Gazebo
Turtlebot3
Turtlebot3 steuern
RViz
RViz Starten
gmapping
Navigation
Catkin
Arbeitsumgebung
Pakete
Beispielprojekt
Paket (Knoten) erstellen
package.xml
CMakesLists.txt
Subscriber Node
Nützliche Links