



#### N. Noack, J.-L. Regenhardt

#### Wiederholung

#### rgänzung

rosrun -parame

#### Gazebo

Turtlebot3 Turtlebot3 steuern

#### RViz

gmapping Navigation

## Navigation

Arbeitsumgebung

#### Beispielprojekt

package.xml CMakesLists.txt

#### Nützliche Lin

# Tutorium Robotik und autonome Systeme Einführung ROS und Turtlebot 3

Niklas Noack<sup>1</sup>, Jan-Luca Regenhardt<sup>1</sup>

<sup>1</sup>Autonomous Mobile Robotics Lab Hochschule für angewandte Wissenschaft und Kunst

12. November 2024

## **Inhaltsverzeichnis**

Wiederholung Ergänzung

> rosrun -parameter roslaunch

Gazebo

Turtlebot3 Turtlebot3 steuern

RVi<sub>z</sub>

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 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

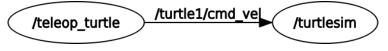


Abbildung: Publisher, Topic, Subscriber



#### N. Noack, J.-L. Regenhardt

#### Wiederholung

Ergänzung

oslaunch

Gazebo

Turtlebot3 Turtlebot3 steuern

RViz

mapping Vavigation

atkin

rbeitsumgebun<sub>l</sub> akete

Beispielprojekt

package.xml CMakesLists.t

Subscriber Node

# Wiederholung 2/2

## Wichtige Befehle:

- ightharpoonup 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 \( \square{\square} \) (Tabulator) Bsp.:
- ros2x≒ listet alle möglichen Befehle beginnend mit ros auf.
  - rosnode  $2x \leftrightarrows$  listet alle möglichen Parameter auf.



N. Noack, J.-L. Regenhardt

#### Wiederholung

rgänzung

osrun -parame oslaunch

azebo

Furtlebot3 steuen

RViz

mapping

Vavigation

#### Catkin

arbeitsumgebun Pakete

# Beispielprojekt Paket (Knoten) ersteller

MakesLists.txt

Subscriber Node

Nutziiche Lini

## 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



N. Noack, J.-L. Regenhardt

**Viederholung** 

rgänzung

rosrun -parameter

Gazebo

urtlebot3

0\/;~

Viz

gmapping Navigation

Catkin

Pakete

Beispielpr

package.xml CMakesLists.txt

Subscriber Node

## 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



N. Noack, J.-L. Regenhardt

**Viederholung** 

gänzung

roslaunch

iazebo

urtlebot3 urtlebot3 steue

RViz

gmapping

Navigation

Latkın

akete

Beispielp

package.xml

.MakesLists.txt Subscriber Node

## Gazebo Simulator

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





#### N. Noack, J.-L. Regenhardt

#### **Viederholung**

Ergänzung rosrun -paramet

#### Gazebo

Furtlebot3 Furtlebot3 steuern

#### RViz

gmapping Navigation

#### atkin

rbeitsumgebui akete

## Beispiel

Paket (Knoten) erstelle package.xml CMakesLists.txt

## Simulation Turtlebot3 in Gazebo

#### Ausführen im Terminal

roslaunch turtlebot3\_gazebo turtlebot3\_house.launch

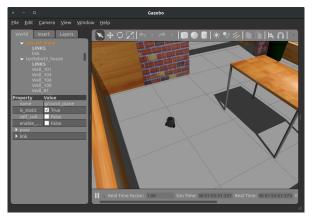


Abbildung: Gazebo-Simulator mit *Turtlebot* 



#### N. Noack, J.-L. Regenhardt

#### **Wiederholung**

#### gänzung

srun -paramete slaunch

#### azebo

## Turtlebot3

Furtlebot3 steuern

#### RVI

gmapping Navigation

#### Catkin

rbeitsumgebu akete

## Beispiel

'aket (Knoten) erstell ackage.xml 'MakesLists.txt

Subscriber Node

Nutzliche Link

## 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



N. Noack, J.-L. Regenhardt

#### Viederholung

## gänzung

osrun -parame oslaunch

#### azebo

tlebot3

#### Turtlebot3 steuern

#### RViz

mapping lavigation

## lavigation

rrbeitsumgebur Pakete

## Beispielp

package.xml CMakesLists.txt

## ROS Vizualisation

In RViz werden Nachrichten (Sensordaten) visualisiert.

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

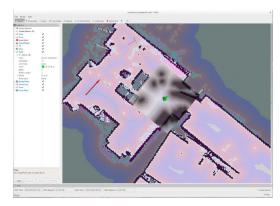


Abbildung: RViz mit *gmapping* und Pfadplanung



N. Noack, J.-L. Regenhardt

**Viederholung** 

rgänzung

rosrun -paramete roslaunch

izebo

urtlebot3 steuern

RViz

mapping Vavigation

atkin

beitsumgebun ikete

Beispielprojekt

Paket (Knoten) erstelle package.xml CMakesLists.txt

## 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

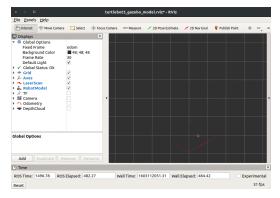


Abbildung: RViz - Gabezo



#### N. Noack, J.-L. Regenhardt

#### Wiederholung

## Ergänzung

rosrun -parame roslaunch

#### azebo

rtlebot3 steuer

#### RViz

#### RViz Starten

Vavigation

#### Catkin

akete

#### Beispiel

package.xml

Subscriber Node



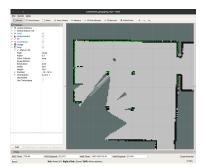
## Beispiele 1/2

#### 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:



## Karte speichern

- Karte im Heimverzeichnis speichern:
- rosrun map\_server map\_saver
  -f /home/amrl/map



N. Noack, J.-L. Regenhardt

**Viederholung** 

ganzung

roslaunch

Turtlebot3

2\/iz

Viz Starte

gmapping

Vavigation

atkin

akete

Beispielprojekt

package.xml CMakesLists.txt

Subscriber Node

Nützliche Link

Abbildung: RViz - Gmapping

# 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

# En plan gib Filmanian Chara C

Abbildung: RViz - Navigation

#### In RViz

- 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.



N. Noack, J.-L. Regenhardt

#### vviedernoiung

#### rganzung

rostun -parame roslaunch

#### Sazebo

Turtlebot3 steuern

#### TVIZ

RViz Starter

#### Navigation

#### Catkin

Arbeitsumgebung Pakete

## Beispielprojekt

ackage.xml MakesLists.txt

Mütaliche Lin

# 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:
  - **b** build
  - **devel**
  - src

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

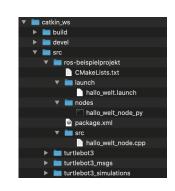


Abbildung: Catkin Workspace



N. Noack, J.-L. Regenhardt

Wiederholung

Ergänzung

roslaunch

Turtlebot3

Turtlebot3 steuern

RViz

gmapping Navigation

Catkin

Arbeitsumgebung Pakete

Pakete

Paket (Knoten) erstellen

MakesLists.txt ubscriber Node

Subscriber Node

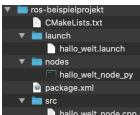


## **ROS-Paket**

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

Dateien und Ordner im Paketeverzeichniss (Auszug):

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



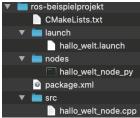


Abbildung: Paket: Beispielprojekt

Tutorium Robotik N. Noack, J.-L. Regenhardt

Pakete

# Beispielprojekt

Idee: Fliehkraft des Turtlebots in der Kurve bestimmen.

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

## 0. Vorüberlegungen

- $\triangleright$  translatorische Geschwindigkeit  $v_x$  in x-Richtung
- Radius r der Kurve
- Topic /cmd\_vel liefert Geschwindigkeiten  $v_x$  und Winkelgeschwindigkeit der Gierachse  $\omega$ .
- rostopic info /cmd\_vel
- ▶ Nachricht hat Typ: geometry\_msgs/Twist
- rosmsg show geometry\_msgs/Twist

Wir programmieren einen *Subscriber* und abonnieren /cmd\_vel!



#### N. Noack, J.-L. Regenhardt

Wiederholung

rganzung

rostun -parame roslaunch

Gazebo

Turtlebot3 steuern

Viz

iz Starten

lavigation

Catkin

Pakete

Paket (Knoten) erstellen

package.xml

Subscriber Node

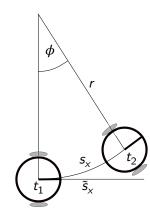
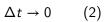


Abbildung: Turtlebot in der Kurve



$$s_{x} = \Delta t v_{x}$$
 (3)

$$\tilde{s}_x = s_x$$
 (4)

$$\phi = \omega \Delta t \qquad (5)$$

$$r = \frac{\tilde{s}_{x}}{\sin\!\phi} \qquad (6)$$

$$a = \frac{\vec{v_x}^2}{r} \qquad (7)$$



#### N. Noack, J.-L. Regenhardt

#### Viederholung

rgänzung

#### `azaba

Furtlebot3

## RViz

gmapping

#### Catkin

Arbeitsumgebung Pakete

## Paket (Knoten) erstellen

package.xml CMakesLists.txt

# 1. Projekt erstellen

## Paket im Catkin-Workspace erstellen

- ► Strg+Alt+T zum öffnen eine Terminals
- ▶ cd /home/ber ter/catkin\_ws/src
- catkin\_create\_pkg --rosdistro me ic -s
  roscpp geometry\_msgs std\_msgs -1 BSD -m
  maintainer\_name -a author\_name
  fliehkraft\_node
- cd /home/benutzer/catkin\_ws
- catkin\_make





#### N. Noack, J.-L. Regenhardt

#### Wiederholung

#### rgänzung

roslaunch

#### Gazebo

Furtlebot3 steuern

#### RViz

Viz Starter

gmapping Vavigation

#### Navigation

#### Latkin

Pakete

#### Beispielprojekt

Paket (Knoten) erstellen

CMakesLists.t:

# 2. package.xml anpassen 1/2



#### N. Noack, J.-L. Regenhardt

Viederholung

#### gänzung

osrun -parameter

#### azebo

urtiebot3 urtlebot3 steuern

#### RViz

gmapping

## Vavigation

## atkin

Pakete

## Beispielp

Paket (Knoten) erstelle

## CMakesList

Subscriber Node

Nutziiche Link

## gedit

~/catkin\_ws/src/fliehkraft\_node/package.xml

## description tag

```
<name>fliehkraft_node </name>
<version > 0.0.0 </version >
<doccupation > The fliehkraft node node
```

<description>The fliehkraft\_node package</description>

## maintainer tags

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

# 2. package.xml anpassen 2/2

<exec\_depend>roscpp</exec\_depend> <exec\_depend>geometry\_msgs</exec\_depend> <exec\_depend>std\_msgs</exec\_depend>

## 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>
<build_export_depend></build_export_depend></build_export_depend>
```

```
AMPL
Autonomous Mobile Robotiss Lab
Tutorium Robotik
```

#### N. Noack, J.-L. Regenhardt

#### /iederholung

#### änzung

rosrun -paramet

#### azebo

tlebot3

#### /:\_

/iz Starten

gmapping

#### ivavigation

#### Catkin

Pakete

#### Beispielp

Paket (Knoten) erstel

#### package.xml

Subscriber Node

# 3. CMakesLists.txt anpassen



~/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(${PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJE($\sqrt{PROJe($\sqrt{PROJE($\sqrt{PROJe($\sqrt{PROJe($\sqrt{PROJe($\sqrt{PROJe($\sqrt{PROJe($\sqrt{PROJe($\sqrt{PROJe($\sqrt{PROJe($\sqrt{PROJe($\sqrt{PROJe($\sqrt{PROJe($\sqrt{PROJe($\sqrt{PROJe($\sqrt{PROJe($\sqrt{PROJe($\sqrt{PROJe($\sqrt{PROJe($\sqrt{PROJe($\sqrt{PROJe($\sqrt{PROJe($\sqrt{PROJe($\sqrt{PROJe($\sqrt{PROJe(\sqrt{PROJe(\sqrt{PROJe(\sqrt{PROJe(\sqrt{PROJe(\sqrt{PROJe(\sqrt{PROJe(\sqrt{PROJe(\sqrt{PROJe(\sqrt{PROJe(\s)}})}})} \end{PROJe(\sqn
```



#### N. Noack, J.-L. Regenhardt

/iederholung

gänzung

nelaunch

azebo

rtlebot3

/;--

/iz

gmapping Navigation

atkin

atkin

Pakete

Beispielpro

Paket (Knoten) erst package.xml

CMakesLists.txt Subscriber Node

# 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

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



#### N. Noack, J.-L. Regenhardt

#### Wiederholung

#### gänzung

rosrun -para roslaunch

#### Gazebo

Turtlebot3 steuern

#### RViz

RViz Starte mapping

Navigation

#### Catkin

Pakete

### Beispiel

aket (Knoten) erstel ackage.xml

Subscriber Node

# 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:
```



#### N. Noack, J.-L. Regenhardt

#### Wiederholung

#### rgänzung

rosrun -parar roslaunch

#### azebo

rtlebot3 steuern

#### RViz

gmapping

#### atkin

Pakete

#### Beispielpi

package.xml

Subscriber Node

# 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
woid 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



#### N. Noack, J.-L. Regenhardt

#### Viederholung

#### gänzung

roslaunch

#### azebo

Turtlebot3 steuern

#### VIZ

gmapping Navigation

#### ivavigation

#### Jatkiii

Pakete

## Beispiel

aket (Knoten) erstelle ackage.xml

#### Subscriber Node

# Kompilieren und Ausführen

## 0. Kompilieren

 $\mathsf{Strg} {+} \mathsf{AIt} {+} \mathsf{T}$ 

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



N. Noack, J.-L. Regenhardt

#### Wiederholung

#### gänzung

rosrun -parame

#### azebo

urtiebot3 'urtlebot3 steuern

#### RViz

Viz Starten

Navigation

#### Catkin

Pakete

#### Beispielp

Paket (Knoten) ersteller nackage.xml

Subscriber Node

## 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/



N. Noack, J.-L. Regenhardt

/iederholung

rgänzung

rosrun -param roslaunch

Gazebo

Turtlebot3 steuern

RViz

kviz Starte gmapping

Navigation

Catkin

Pakete

Beispielpr

package.xml CMakesLists.txt

Subscriber Node

## Nützliche Links 2

AMPLE Autonomous Mobile Robotics Lab
Tutorium Robotik

N. Noack, J.-L. Regenhardt

**Viederholung** 

rganzung

rosrun -parai roslaunch

azebo

urtlebot3 urtlebot3 steuern

RViz

RViz Sta

gmapping Navigation

Catkin

Arbeitsumgebu

Beispielprojekt

aket (Knoten) erstelle

Subscriber Node

Nützliche Links

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

## 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



#### N. Noack, J.-L. Regenhardt

Viederholung

rgänzung

rosrun -parame roslaunch

iazebo

urtlebot3 urtlebot3 steuern

RViz

mapping

Vavigation

atkin

Pakete

Beispiel

Paket (Knoten) ersteller

Subscriber Node