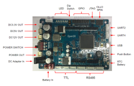
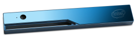
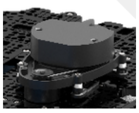
Project Ballbot

Markus Lamprecht
Florian Müller
Michael Suffel

TU Darmstadt, RTM

Inhaltsverzeichnis

1 Item - List

Item	#	W.[g]	Weblink	Picture
OpenCR Board (Controlling the motors, IMU)	1	60	github_wiki	
UpBoard (Main PC)	1	96	127€	
Intel RealSense R200	1	9.4	datasheet, 84.15€	
Laser Distance Sensor	1	124	specs, 100€	
Battery: LI-PO 11.1 1800mAh LB-12 19	1	132	44.90€	
Turtlebot3 Layers()	4			
XM430-W350-R Dynamixel (Motors)	3	72	robotis,250€	
Ball(alum., dia.: 140mm, material thickness 2.5mm)	1	400	ball-tech gmbh,40€.	
Omni wheels(dia: 60mm, thickness:25mm)	3	51.46	10.38€	
Kreisring (PLA, 3D printeted)	1	28		
Halterung (PLA, 3D printeted)	3	18		
Mitnehmer (PLA, 3D printeted)	3	8		
Plain washer (Beilagscheibe),(PLA, 3D printeted)	3	0.45		
M3 (Mutter-Halterung-Kreisring-Layer)	9			
M2.5 (Kreisring-Layer)	2			
M3x8mm Halterung	6		Zylinderkopf (Imbus)	
M3x22mm Layer	3	1.34	Zylinderkopf (Imbus)	
M2.5x22 (Motoren-Halterung)	12		Sechskant	
M2.5x38 (Motoren-Rad)	3		Zylinderkopf (Imbus)	
M2.5x24 (Layer)	2		Zylinderkopf (Imbus)	
M2x6mm (Mitnehmer-Motor)	12		Zylinderkopf (Imbus)	

Total Cost: 1176€ + Cost of opencr board and all plastic (incl. tb3 structure) and scrwes
TODO:

1. Abmessungen von einer struckture layer
2. upboard1-link noch eintragen

2 Simulation

2.1 Launch

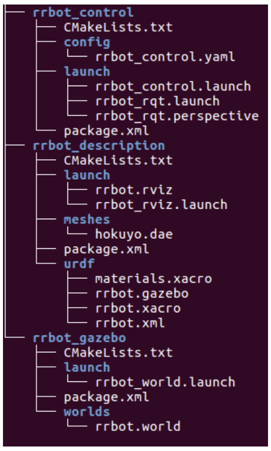
These files are executed one after another:

1. bb_simulation: ballbot.launch
2. bb_description: bb_description.launch
3. bb_description -> urdf: bb.xacro
4. bb_description -> urdf: bb.urdf.xacro
5. bb_description -> urdf: common_properties.xacro
6. bb_description -> urdf: bb.gazebo.xacro

2.2 Simulation design

Ballbot SDF Reference: Ballbotmodel

We use not the sdf but the xacro description as in this example here.



```
├── rrbot_control
│   ├── CMakeLists.txt
│   ├── config
│   │   └── rrbot_control.yaml
│   ├── launch
│   │   ├── rrbot_control.launch
│   │   ├── rrbot_rqt.launch
│   │   └── rrbot_rqt.perspective
│   └── package.xml
├── rrbot_description
│   ├── CMakeLists.txt
│   ├── launch
│   │   ├── rrbot.rviz
│   │   └── rrbot_rviz.launch
│   ├── meshes
│   │   └── hokuyo.dae
│   ├── package.xml
│   └── urdf
│       ├── materials.xacro
│       ├── rrbot.gazebo
│       ├── rrbot.xacro
│       └── rrbot.xml
└── rrbot_gazebo
    ├── CMakeLists.txt
    ├── launch
    │   └── rrbot_world.launch
    ├── package.xml
    └── worlds
        └── rrbot.world
```

Gazebo uses different physics engines:

- Open Dynamics Engine (ODE) (Default)
- Bullet
- Dynamic Animation and Robotics Toolkit (DART)
- Simbody

which all have different friction etc. models.

Files:

- bb.urdf.xacro: Link's: Visual description of the Robot and its collision model(STL file). Pose Mass and Inertias. Joint's: Pose,axis,effort and velocity limits, friction.

- common_properties.xacro: Macros for color definition.
- bb.gazebo.xacro: gazebo references dynamics of the links: friction parameters (mu1,mu2),

Gazebo Parameter's List:

name(xacro)	description	value	sdf group
mu1	is the Coulomb friction coefficient for the first friction direction	1.0	ode
mu2	is the friction coefficient for the second friction direction (perpendicular to the first friction direction)	2.0	ode
kp	spring constant equivalents of a contact as a function of SurfaceParams::cfm and SurfaceParams::erp		ode
kd	spring damping constant equivalents of a contact as a function of SurfaceParams::cfm and SurfaceParams::erp.		ode
cfm	Constraint Force Mixing parameter.		ode
erp	Error Reduction Parameter.		ode
min_depth	Minimum depth before ERP takes effect.		ode
max_Vel	Maximum interpenetration error correction velocity. If set to 0, two objects interpenetrating each other will not be pushed apart.		ode
slip1	Artificial contact slip in the primary friction direction		ode
slip2	Artificial contact slip in the secondary friction dirction.		ode

See: ODESurfaceParams

2.3 Gazebo Parameters

2.4 Control

sobald diff drive plugin angeschaltet drehen sich die raeder viel zu schnell ....
Diff Drive in ballbot.launch an oder ausschalten.
in bb.gazebo.xacro transmission und controller festlegen.
zudem yaml file(currently I use: effort_controllers/JointVelocityController)
Effort Joint Interface as Hardware Interface is used.
Do this example first: `http://gazebosim.org/tutorials/?tut=ros_control`

2.4.1 Plugins

- gazebo-ros-control
- diff drive

2.4.2 Launch

```
roslaunch rrbot_control rrbot_control.launch
```

These files are executed one after another:

1. load config
2. controller_spawner

2.5 Sensors

2.5.1 IMU

We want to simulate the IMU of the opencr board. STRG+T to see imu topic values! Imu of opencr board simulated

Simulate like this: rviz rviz dann als fixed frame nimm: imu_link. Und add topic imu und waehle als topic ballbot/sensor/imu

The simulated IMU outputs values like: orientation (x,y,z,w), angluar velocity(x,y,z), linear velocity(x,y,z), linear acceleration(x,y,z).

The opencr real IMU gives values like: orientation(x,y,z,w), angular velocity(x,y,z), linear acceleration(x,y,z) see http://turtlebot3.readthedocs.io/en/latest/appendix_opencr.html

3 Model

3.1 Composition

The Ballbot consists of three parts, which are depicted in Figure 3.1.

- Body with motors
- 3 omni-directional wheels
- Ball

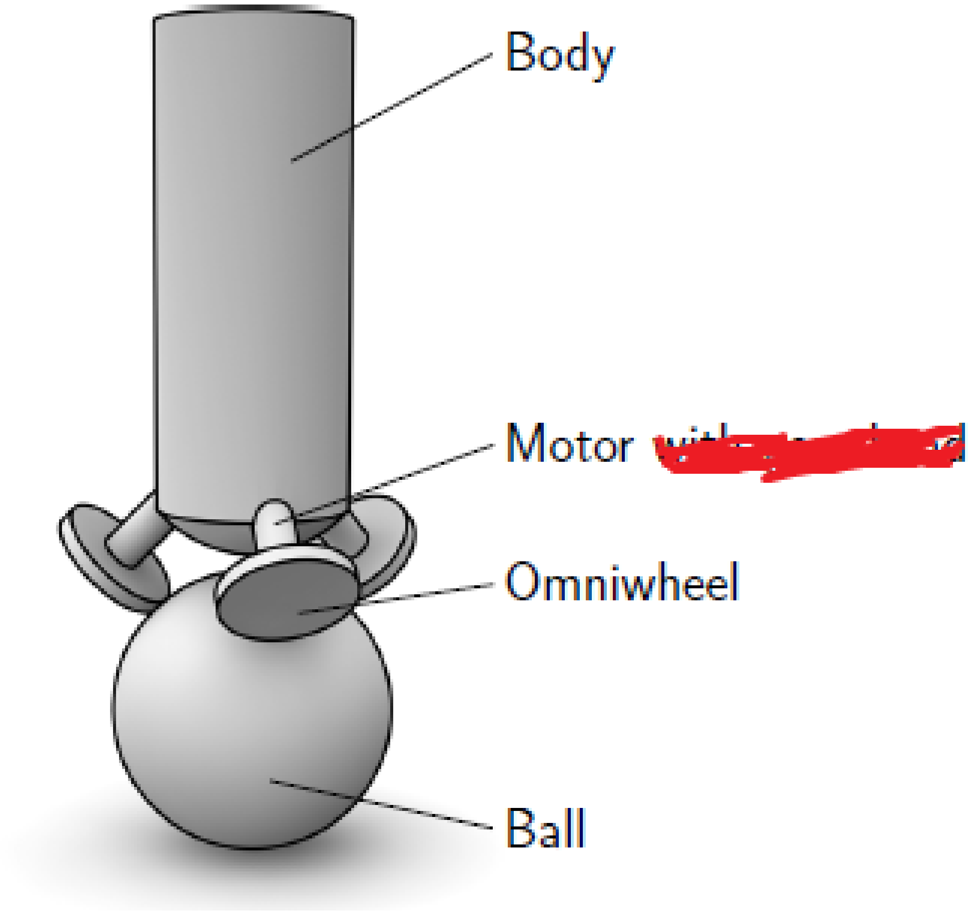


Abbildung 3.1: Parts for the 3D-Model

name	Mass [kg]	inertia	pic	Translation from basis point to name
Upstructure (all except wheels)	1.557	i_{xx} =?? i_{yy} =?? i_{zz} =?? i_{xy} =?? i_{zy} =?? i_{xz} =??		x= y= z=

3.2 Assumptions

To reduce the complexity of the system, the following assumptions are made:

- No slip between the contact points between the ball/ground and wheels/ball
- No friction; except the friction, which occurs at the rotation of the ball around the z-axis
- No deformation
- Fast motor dynamics; The controlling of the motor is much faster than the controller of the Ballbot
- Ball moves only horizontal

3.3 Dynamic