

Institute for Artificial Intelligence Faculty 03 Mathematics &

Computer Science

# Robot Programming with ROS

4. Motors and Kinematics

Arthur Niedźwiecki, Stefan Eirich 09<sup>th</sup> Nov. 2023





#### Robot Programming with ROS Arthur Niedźwiecki, 4. Motors and Kinematics Stefan Eirich

Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science

# Overview

- 1 What makes a robot? Links and joints
- 2 Actuators
- 3 Robot Arms
- 4 Gripper
- 5 Mobile Bases Wheeled locomotion
- 6 Robot Kinematics
  - Forward and Inverse Kinematics
- 7 Representation Of The Environment
- 8 Organizational



Robot Programming with ROS Arthur Niedźwiecki, 4 Motors and Kinematics Stefan Eirich

Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science

# Overview

- 1 What makes a robot? Links and joints
- 2 Actuators
- 8 Robot Arms
- 4 Gripper
- 5 Mobile Bases Wheeled locomotio
- 6 Robot Kinematics
  - Forward and Inverse Kinematics
- Representation Of The Environment
- 8 Organizational



## What makes a robot?

Robot Programming with ROS Arthur Niedźwiecki, 4. Motors and Kinematics Stefan Eirich

Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science

A Robot is an electro-mechanical device, composed of a collection of bodies (links), which are combined by joints. A robot is equipped with actuators (motors), that can move neighbouring links relative to each other by exerting forces.



# Robot Programming with ROSArthur Niedźwiecki,4. Motors and KinematicsStefan Eirich

09<sup>th</sup> Nov. 2023

Faculty 03 Mathematics & Computer Science

## What makes a robot?

A Robot is an electro-mechanical device, composed of a collection of bodies (links), which are combined by joints. A robot is equipped with actuators (motors), that can move neighbouring links relative to each other by exerting forces.

Robots are equipped with control programs that are designed to accomplish tasks by moving the body of the robot in a task-driven way.

The control software will be addressed in subsequent modules.



#### Robot Programming with ROS Arthur Niedźwiecki, 4. Motors and Kinematics Stefan Eirich

Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science

# **Basic Principles**

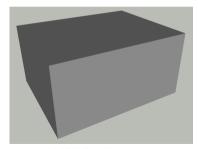
- Position the End effector (Move the end effector in the room)
- Touch objects (Manipulation)
- Exercise a force on the ground relative to itself (Move in the plane)

Universität Bremen	EASE	Artificial Intelligence	Robot Programming with ROS	Arthur Niedźwiecki,	Faculty 03
			4. Motors and Kinematics	Stefan Eirich 09 <sup>th</sup> Nov. 2023	Mathematics & Computer Science
-				05 1000. 2025	Computer Science

#### Links

#### Rigid body with visual and collision features and known inertia properties.

```
k name="${name}">
        <visual>
            <geometry>
4
                <box size="${base_length} ${base_width} ${
                     base_height}"/>
            </aeometry>
            <material name="LightGrev">
                <color rgba="0.7 0.7 0.7 1.0"/>
8
            </material>
9
        </visual>
        <collision>
            <geometry>
                <box size="${base length} ${base width} ${</pre>
                    base height}"/>
            </geometry>
14
        </collision>
        <inertial>
16
            <origin xyz="0 0 0.5" rpy="0 0 0"/>
            <mass value="100"/>
18
            <inertia ixx="100" ixy="0" ixz="0" iyy="100" iyz
                 ="0" izz="100"/>
19
        </inertial>
20
   </link>
```





Institute for Artificial Intelligence, University of Bremen CC BY-SA 4.0: https://creativecommons. org/licenses/by-sa/4.0/

6



Faculty 03 Mathematics & Computer Science

## Joints

- Two type of joints: **revolute** (rotary motion) and **prismatic** (linear motion)
- · Classical industrial application: high stiffness in links and joints
  - Advantage: position accuracy
  - Disadvantage: may lead to high forces exerted by robot  $\rightarrow$  danger!
  - Note: link deflection under load or joint play might reduce precision in the real-world

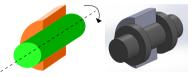




#### One translational degree of freedom



Jamesoniai Source: https://en.wikipedia.org/w/index.php?title= Prismatic\_joint&oldid=1003084413 CC BYSR4.0: https://creativecommons.org/licenses/by-sa/4.0/



#### One rotary degree of freedom

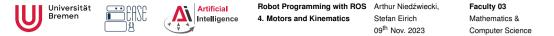


Source: https://en.wikipedia.org/w/index.php?title=Revolute\_ joint&oldid=1000911780 CC BY-SA 40: https://creativecommons.org/licenses/by-sa/4.0/



Public Domain Source: https://en.wikipedia.org/w/index.php?title=Revolute\_ joint&oldid=1000911780

CC BY-SA 4.0: https://creativecommons.org/licenses/by-sa/4.0/



## Joints

```
<joint name="${prefix}shoulder lift joint" type="continuous">
2
       <parent link="${prefix}base link"/>
3
       <child link = "${prefix}upper_arm_link"/>
4
       <origin xyz="0 0 0" rpy="0 0 0"/>
5
       <axis xyz="0 0 1"/>
6
       <limit lower="${shoulder lift lower limit}" upper="${shoulder lift upper limit}"</pre>
             effort="150.0" velocity="3.15"/>
7
       <dynamics damping="1.0" friction="100.0"/>
8
   </joint>
```





Robot Programming with ROS

4. Motors and Kinematics

Arthur Niedźwiecki, Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science

## Non-classical approach: soft robots

Impedance control





SOURCE: https://en.wikipedia.org/w/index.php?title=Justin\_(robot)&oldid=1004198463 CC BY-SA 4.0: https://creativecommons.org/licenses/by-sa/4.0/



# Summary

Robot Programming with ROS Arthur Niedźwiecki,

4. Motors and Kinematics

Stefan Eirich 09<sup>th</sup> Nov. 2023

- Basic principles of robotics
- Description of links and joints



# Overview

- 1 What makes a robot? Links and joints
- 2 Actuators
- 3 Robot Arms
- 4 Grippe
- 5 Mobile Bases Wheeled locomotio
- 6 Robot Kinematics
  - Forward and Inverse Kinematics
- Representation Of The Environment
- 8 Organizational

Robot Programming with ROS

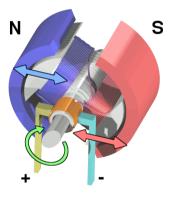
4. Motors and Kinematics



# Brushed DC Motor

Robot Programming with ROS Arthur Niedźwiecki, 4. Motors and Kinematics Stefan Eirich

Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science





Wapcapiel Source: https://en.wikipedia.org/w/index.php?title=Brushed\_DC\_ electric\_motorkoldid=1009901547 J CC BY-SA 4.0: https://creativecommons.org/licenses/by-sa/4.0/

Video: LearnEngineering2014SepDCMotorHow

cheap

- used in toys
- turns without motor controller
- commutation with brushes (may wear out)



Faculty 03 Mathematics & Computer Science

•

# **Brushless DC Motor**

- cheap as well
- used e.g. in CD-ROM drives
- electronic commutation necessary (with sensors or sensorless)
- coils can be inside or outside (better cooling when outside)





Koppene, Sedasuan Source: https://en.vikipedia.org/w/index.php?title=Brushless\_DC\_electric\_motor&oldid=1013753090 CC BY-SA 4.0: https://creativecommons.org/licenses/by-sa/4.0/

#### Video: LearnEngineering2014OctBrushlessDCMotor



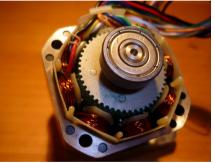
# Stepper Motor

- very standardized
- strong at low speeds
- used in printers
- moves repeatably
- feed-forward by nature
- "looses steps" when friction/inertia/... is too high ...
- ... or is moving at its resonance frequency

Robot Programming with ROSArthur Niedźwiecki,4. Motors and KinematicsStefan Eirich

09<sup>th</sup> Nov. 2023

Faculty 03 Mathematics & Computer Science





Dolly1010 Source: https://en.wikipedia.org/w/index.php?title= Stepper\_motorkoldid=1013297392 CC BYSA 4.0: https://creativecommons.org/licenses/by-sa/4.0/

Video: LearnEngineering2016OctHowDoesStepper



# Overview

- 1 What makes a robot? Links and joints
- 2 Actuators
- 3 Robot Arms
- 4 Grippei
- 5 Mobile Bases Wheeled locomotio
- 6 Robot Kinematics
  - Forward and Inverse Kinematics
- Representation Of The Environment
- 8 Organizational

Robot Programming with ROS

4. Motors and Kinematics



# Robot Arm

Robot Programming with ROSArthur Niedźwiecki,4. Motors and KinematicsStefan Eirich

09<sup>th</sup> Nov. 2023

Faculty 03 Mathematics & Computer Science

Sequence of links connected by joints and moved relative to each other by actuators.



Robot Programming with ROS Arthur Niedźwiecki,

4. Motors and Kinematics

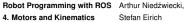
Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science

## Robot Arm

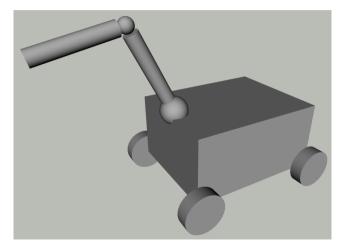
1	<pre><xacro:macro name="simple_arm" params="prefix parent *origin joint_limited:=true"></xacro:macro></pre>
2	
3	<joint name="shoulder_pan_joint" type="continuous"></joint>
4	<pre><pre>cparent link="\${parent}"/&gt;</pre></pre>
5	<pre><child link="\${prefix}base_link"></child></pre>
6	<xacro:insert_block name="origin"></xacro:insert_block>
7	<axis xyz="0 1 0"></axis>
8	<pre><limit 1.0"="" effort="150.0" friction="100.0" lower="\${shoulder_pan_lower_limit}" upper="\${shoulder_pan_upper_limit}" velocity="3.15&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;9&lt;/td&gt;&lt;td&gt;&lt;pre&gt;//&gt; &lt;dvnamics damping="></limit></pre>
10	</td
11	joint
12	<pre><link name="\${prefix}base link"/></pre>
13	<pre></pre>
14	<visual> <origin rpy="0 0 \${base_correction}" xyz="0 0 0"></origin></visual>
15	<pre><dought 0="" 1py="0" stass_contention;="" xyz="0"></dought></pre>
16	<pre><geometry> <sphere radius="\${base radius}"></sphere></geometry></pre>
17	<pre><pre>capital factors = \$10000 / 75</pre></pre>
18	<pre><rgeometry> <mathcare="lightgrev"></mathcare="lightgrev"></rgeometry></pre>
19	<pre></pre>
20	
21	
22	<collision></collision>
23	<pre><origin rpv="0 0 \${base correction}" xvz="0 0 0"></origin></pre>
24	<pre><geometry></geometry></pre>
25	<pre><sphere radius="\${base_radius}"></sphere></pre>
26	
27	collision
28	<pre><xacro:sphere inertial="" mass="\${base mass}" radius="\${base radius}"></xacro:sphere></pre>
29	<pre><origin rpy="0 0 0" xyz="0.0 0.0 0.0"></origin></pre>
30	
31	
32	
33	[]
34	
35	
36	



## **Robot Arm**



Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science





Institute for Artificial Intelligence, University of Bremen CC BY-SA 4.0: https://creativecommons.org/licenses/by-sa/4.0/

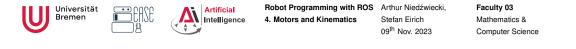


# Overview

- 1 What makes a robot? Links and joints
- 2 Actuators
- 8 Robot Arms
- 4 Gripper
- 5 Mobile Bases Wheeled locomotion
- 6 Robot Kinematics
  - Forward and Inverse Kinematics
- Representation Of The Environment
- 8 Organizational

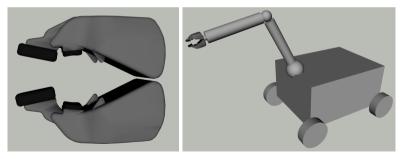
Robot Programming with ROS

4. Motors and Kinematics



#### Touch objects and interact with the environment.

Gripper





Institute for Artificial Intelligence, University of Bremen CC BY-SA 4.0: https://creativecommons.org/licenses/by-sa/4.0/



# **Overview**

- Links and joints

- 5 Mobile Bases Wheeled locomotion
- - Forward and Inverse Kinematics

Robot Programming with ROS Arthur Niedźwiecki. Stefan Firich 09<sup>th</sup> Nov. 2023

4. Motors and Kinematics



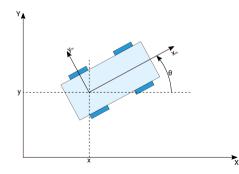
 Robot Programming with ROS
 Arthur Niedźwiecki,

 4. Motors and Kinematics
 Stefan Eirich

Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science

# Wheeled Locomotion

**Goal**: Bring the robot to a desired pose  $(x, y, \theta)$ : (position in **x**-axis, position in **y**-axis, **angle** with x-axis)  $\Rightarrow$  3 Degrees of Freedom (DOF)





Institute for Artificial Intelligence, University of Bremen CC BY-SA 4.0: https://creativecommons.org/licenses/by-sa/4.0/



Robot Programming with ROS Arthur Niedźwiecki,

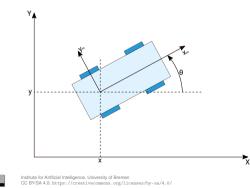
4. Motors and Kinematics

Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science

# Wheeled Locomotion

**Goal**: Bring the robot to a desired pose  $(x, y, \theta)$ : (position in **x**-axis, position in **y**-axis, **angle** with x-axis)

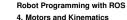
 $\Rightarrow$  3 Degrees of Freedom (DOF)



Wheeled motion typically has non-holonomic constraints

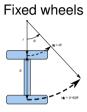


# Wheel Types



ROS Arthur Niedźwiecki,

Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science





NikNaks Source:

Source: https://en.wikipedia.org/w/index.php?title=Differential\_wheeled\_robot&oldid=1000825516 CC BY-SA 4.0: https://creativecommons.org/licenses/by-sa/4.0/









#### Video



Euchlasmus Source: https://en.wikipedia.org/w/index.php?title=Mecanum\_wheel&oldid=1012482690 CC BV-SA 4 0: https://creativecompons.org/licenses/by-sa/4.0/

obodoc9

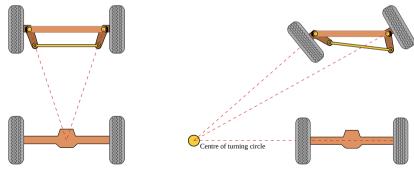
Source: https://en.wikipedia.org/w/index.php?title=Omni\_wheel&oldid=1013193962 CC BY-SA 4.0: https://creativecommons.org/licenses/by-sa/4.0/



Faculty 03 Mathematics & Computer Science

# Ackerman steering

- Car-like steering
- + Robust
- + Outer wheels moves on a circle of different radius than inner wheel
- But hard to control (parking!)





SOUTCE: https://en.wikipedia.org/w/index.php?title=Ackermann\_steering\_geometry&oldid=975976305 CC BY-SA 4.0: https://creativecommons.org/licenses/by-sa/4.0/



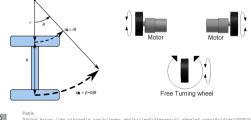
**Robot Programming with ROS** Arthur Niedźwiecki. 4. Motors and Kinematics Stefan Eirich

09<sup>th</sup> Nov. 2023

Faculty 03 Mathematics & Computer Science

# **Differential-Drive**

- + Turns on spot
- + Good choice for round robots
- + Parking is easier
- Cannot move sidewards





Source: https://en.wikipedia.org/w/index.php?title=Differential\_wheeled\_robot&oldid=1000825516 CC BY-SA 4.0: https://creativecommons.org/licenses/by-sa/4.0/

Source: https://en.wikipedia.org/w/index.php?title=Differential wheeled robotkoldid=1000825516



Faculty 03 Mathematics & Computer Science

# Turnable wheels

- + Omnidirectional (can drive forwards, sideways and turn)
- On change of direction, requires 'reconfiguration' of its wheels.
- $\rightarrow$  Controllers should not oscillate







Institute for Artificial Intelligence, University of Bremen CC BY-SA 4.0: https://creativecommons.org/licenses/by-sa/4.0/

#### PR2: Double wheel construction to reduce friction while turning the wheel



Robot Programming with ROSArthur Niedźwiecki,4. Motors and KinematicsStefan Eirich

Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science

# "Omniwheels"

- + Omnidirectional (can drive forwards, sideways and turn)
- Wheels have free rollers at 90°
- + Three wheels are enough
- Hard to make them run smooth





Rotacaster Source: https://en.wikipedia.org/w/index.php?title=Omni\_wheel&oldid=1013193962 CC BY-SA 4.0: https://creativecommons.org/licenses/by-sa/4.0/

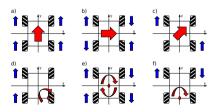


Faculty 03 Mathematics & Computer Science

# Mecanum-Wheels

- + Omnidirectional (can drive forwards, sideways and turn)
- Wheels have free rollers at 45°
- + No reconfiguration is involved
- Depending on wheels, requires flat ground







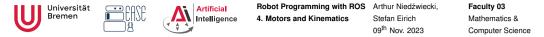
Institute for Artificial Intelligence, University of Bremen GC BY-SA 4.0: https://creativecommons.org/licenses/by-sa/4.0/



Source: https://en.wikipedia.org/w/index.php?title=Mecanum\_wheel&oldid=1012482690 CC BY-SA 4.0: https://creativecommons.org/licenses/by-sa/4.0/

#### $\textbf{Linearity} \Rightarrow$

A (linear) combination of cartesian movements can be achieved with the linear combination of the respective wheel velocities.



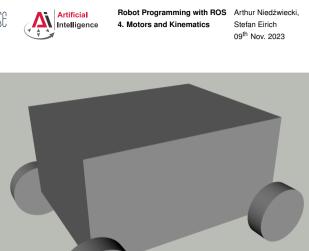
## Mobile Base

```
<xacro:simple robot base name="base link">
 2
        </xacro:simple robot base>
 3
4
        <xacro:omni wheel name="omni wheel fl" parent="base link">
 5
             <origin xyz="\{0.5 \times base \ length\} \{0.5 \times base \ width + 0.5 \times wheel \ thickness\}
                  ${-0.5*base height}" rpv="${pi/2} 0 0"/>
6
        </xacro:omni wheel>
 7
8
        <xacro:omni wheel name="omni wheel fr" parent="base link">
9
             <origin xvz="\{0.5 \times base length\} \{-0.5 \times base width - 0.5 \times wheel thickness\}
                  (-0.5 \text{ base height}) \text{ rpv} = (0.5 \text{ base height})^{-1}
10
        </xacro:omni wheel>
11
12
        <xacro:omni wheel name="omni wheel bl" parent="base link">
             <origin xvz="\{-0.5 \\ base length} \{0.5 \\ base width + 0.5 \\ wheel thickness}
13
                  \{-0.5 \text{ base height}\}" rpy="${pi/2} 0 0"/>
14
        </xacro:omni wheel>
15
16
        <xacro:omni wheel name="omni wheel br" parent="base link">
17
             <origin xvz="\{-0.5 \\ base length} \{-0.5 \\ base width -0.5 \\ wheel thickness}
```

```
29
```



## Mobile Base









# Summary

Robot Programming with ROS Arthur Niedźwiecki, 4. Motors and Kinematics Stefan Eirich

Stefan Eirich 09<sup>th</sup> Nov. 2023

- Actuation of robot arms
- Wheeled locomotion



# Overview

- 1 What makes a robot? Links and joints
- 2 Actuators
- 3 Robot Arms
- 4 Grippe
- 5 Mobile Bases Wheeled locomo
- 6 Robot Kinematics Forward and Inverse Kinematics
- Representation Of The Environment
- 8 Organizational

Robot Programming with ROS Arthur Niedźwiecki, 4. Motors and Kinematics Stefan Eirich

Stefan Eirich 09<sup>th</sup> Nov. 2023



## **Kinematics**

Robot Programming with ROS Arthur Niedźwiecki,

4. Motors and Kinematics

Stefan Eirich 09<sup>th</sup> Nov. 2023

- Joint Space
- Task Space (Workspace)
- Forward Kinematics
- Inverse Kinematics



# **Configuration Space**

Robot Programming with ROSArthur Niedźwiecki,4. Motors and KinematicsStefan Eirich

09<sup>th</sup> Nov. 2023

Faculty 03 Mathematics & Computer Science

#### Configuration

The **configuration** of a robot is a complete specification of the positions of every point of the robot.

"How can we represent the configuration?"



#### Robot Programming with ROS Arthur Niedźwiecki, 4. Motors and Kinematics Stefan Eirich

Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science

# **Configuration Space**

**Degrees of Freedom** 

The number of **degrees of freedom** is the minimum number of independent parameters needed to represent the configuration of the robot.

It is equivalently called the **mobility** of the robot.



 Robot Programming with ROS
 Arthur Niedźwiecki,

 4. Motors and Kinematics
 Stefan Eirich

 09<sup>th</sup> Nov. 2023
 100<sup>th</sup> Nov. 2023

Faculty 03 Mathematics & Computer Science

## Configuration Space

**Degrees of Freedom** 

The number of **degrees of freedom** is the minimum number of independent parameters needed to represent the configuration of the robot.

It is equivalently called the **mobility** of the robot.

**Configuration Space** 

If the robot has *n* degrees of freedom, the *n*-dimensional space containing all possible configurations of the robot is called the **configuration space**.

The robot's configuration is usually expressed in terms of joint variables, so the configuration space is also called the **joint space**.



 Robot Programming with ROS
 A

 4. Motors and Kinematics
 S

Arthur Niedźwiecki, Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science

### Configuration Space vs. Task Space





Institute for Artificial Intelligence, University of Bremen CC BY-SA 4.0: https://creativecommons.org/licenses/by-sa/4.0/

#### How can we get the configuration/joint trajectory to do this?

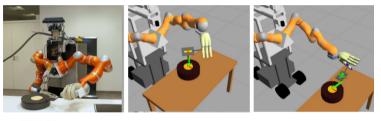


 Robot Programming with ROS
 A

 4. Motors and Kinematics
 S

Arthur Niedźwiecki, Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science

### Configuration Space vs. Task Space





Institute for Artificial Intelligence, University of Bremen CC BY-SA 4.0: https://creativecommons.org/licenses/by-sa/4.0/

• The robot interacts with **objects** and **obstacles** which are better defined by an external set of coordinates (*e.g.*, (**x**, **y**, **z**) coordinates in a **global frame**.)



Robot Programming with ROSAn4. Motors and KinematicsSt

Arthur Niedźwiecki, Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science

## Configuration Space vs. Task Space





Institute for Artificial Intelligence, University of Bremen CC BY-SA 4.0: https://creativecommons.org/licenses/by-sa/4.0/

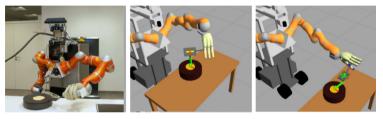
- The robot interacts with **objects** and **obstacles** which are better defined by an external set of coordinates (*e.g.*, (**x**, **y**, **z**) coordinates in a **global frame**.)
- The part interacting with environments such as obstacles and objects is called the end-effector.



Robot Programming with ROS Arthur Niedźwiecki. 4. Motors and Kinematics

Stefan Firich ngth Nov 2023 Faculty 03 Mathematics & Computer Science

### Configuration Space vs. Task Space





CC BY-SA 4.0: https://creativecommons.org/licenses/by-sa/4.0/

- The robot interacts with objects and obstacles which are better defined by their coordinates (e.g., (x, y, z) Euclidean 3D coordinates in a world frame.)
- The part interacting with environments such as obstacles and objects is called the end-effector
- It is useful to attach a coordinate frame to the end-effector and command the pose of the end-effector in the world/object frame.



Robot Programming with ROS

4. Motors and Kinematics

Arthur Niedźwiecki, Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science

### Task Space aka Operational Space

### **Operational Space**

Given a robot with a reference frame attached to its end-effector, the **operational space** is the set of all positions and/or orientations achievable by the end-effector frame.



Robot Programming with ROS

4. Motors and Kinematics

Arthur Niedźwiecki, Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science

### Required degrees of freedom

How many degrees of freedom do we need?



Robot Programming with ROS Arthu

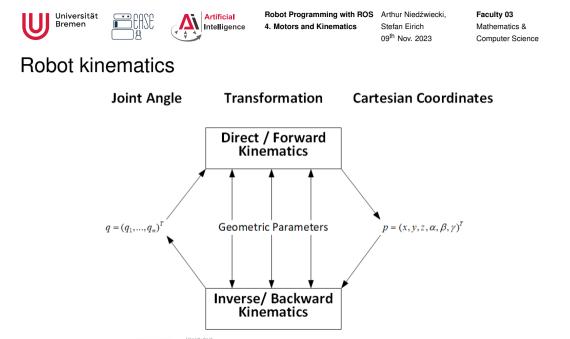
4. Motors and Kinematics

Arthur Niedźwiecki, Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science

### Required degrees of freedom

How many degrees of freedom do we need? (It depends on the task!)

- pointing a camera: 2 DOF
- placing an object (only position is important): 3 DOF
- placing an object (position and orientation): 6 DOF
- imitating a human arm: 7 DOF (from shoulder ball joint)





 Robot Programming with ROS
 Arthur Niedźwiecki,

 4. Motors and Kinematics
 Stefan Eirich

 09<sup>th</sup> Nov. 2023
 09<sup>th</sup> Nov. 2023

Faculty 03 Mathematics & Computer Science

## Forward Kinematics

### Forward Kinematics Problem

Given an open-chain robot arm with a prescribed task frame, the goal is to determine the task frame's position and orientation as a function of the joint values.



Given:  $(q_1, q_2, \dots, q_7)$ 





Robot Programming with ROS Arthur Niedźwiecki. 4. Motors and Kinematics Stefan Firich

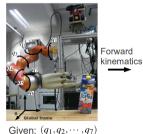
09<sup>th</sup> Nov. 2023

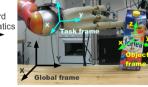
Faculty 03 Mathematics & Computer Science

# **Forward Kinematics**

### Forward Kinematics Problem

Given an open-chain robot arm with a prescribed task frame, the goal is to determine the task frame's position and orientation as a function of the joint values.





Calculate:  $(x, y, z, \alpha, \beta, \gamma)$ 

#### 40 For a given set of form angles there will usually exist a **unique** end-effector position



Robot Programming with ROSArthur Niedźwiecki,4. Motors and KinematicsStefan Eirich

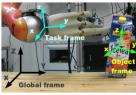
09<sup>th</sup> Nov. 2023

Faculty 03 Mathematics & Computer Science

## **Inverse Kinematics**

#### **Inverse Kinematics Problem**

Given a desired position and orientation of the end-effector frame, one seeks to determine the set of joint angles that achieves this desired end-effector configuration.



Given:  $(x, y, z, \alpha, \beta, \gamma)$ 





Robot Programming with ROS Arthur Niedźwiecki. 4. Motors and Kinematics Stefan Firich 09<sup>th</sup> Nov. 2023

Faculty 03 Mathematics & Computer Science

## **Inverse Kinematics**

### **Inverse Kinematics Problem**

Given a desired position and orientation of the end-effector frame, one seeks to determine the set of joint angles that achieves this desired end-effector configuration.



Given:  $(x, y, z, \alpha, \beta, \gamma)$ 



Calculate:  $(a_1, a_2, \cdots, a_7)$ 

For a particular end-effector position and orientation, there may exist **multiple** solutions, or even **none** at all. (*e.g., elbow-up* and *elbow-down* configurations)

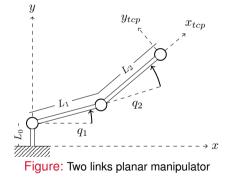


Robot Programming with ROSArthur Niedźwiecki,4. Motors and KinematicsStefan Eirich

09<sup>th</sup> Nov. 2023

Faculty 03 Mathematics & Computer Science

## Kinematics – An Example





Institute for Artificial Intelligence, University of Bremen CC BY-SA 4.0: https: //creativecommons.org/licenses/by-sa/4.0/

#### • Forward kinematics :

$$\begin{pmatrix} x_{tcp} \\ y_{tcp} \end{pmatrix} = \begin{pmatrix} L_1 \cos q_1 + L_2 \cos(q_1 + q_2) \\ L_1 \sin q_1 + L_2 \sin(q_1 + q_2) \end{pmatrix}$$



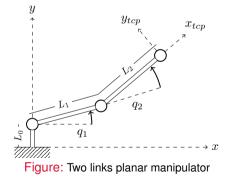
 Robot Programming with ROS
 Arthur Niedźwiecki,

 4. Motors and Kinematics
 Stefan Eirich

 09<sup>th</sup> Nov. 2023
 100<sup>th</sup> Nov. 2023

Faculty 03 Mathematics & Computer Science

### Kinematics – An Example





Institute for Artificial Intelligence, University of Bremen CC BY-SA 4.0: https: //creativecommons.org/licenses/by-sa/4.0/

#### Forward kinematics :

$$\begin{pmatrix} x_{tcp} \\ y_{tcp} \end{pmatrix} = \begin{pmatrix} L_1 \cos q_1 + L_2 \cos(q_1 + q_2) \\ L_1 \sin q_1 + L_2 \sin(q_1 + q_2) \end{pmatrix}$$

• Inverse kinematics :

$$\begin{split} \alpha &= \cos^{-1} \big( \frac{x_{tcp}^2 + y_{tcp}^2 + L_1^2 - L_2^2}{2L_1 \sqrt{x_{tcp}^2 + y_{tcp}^2}} \big), \, \beta = \cos^{-1} \big( \frac{L_1^2 + L_2^2 - x_{tcp}^2 - y_{tcp}^2}{2L_1 L_2} \big) \\ q_1 &= \tan^{-1} \frac{y_{tcp}}{x_{tcp}} - \alpha, \, q_2 = \pi - \beta \\ q_1 &= \tan^{-1} \frac{y_{tcp}}{x_{tcp}} + \alpha, \, q_2 = \pi + \beta \end{split}$$



# Overview

- 1 What makes a robot? Links and joints
- 2 Actuators
- 3 Robot Arms
- 4 Gripper
- 6 Mobile Bases Wheeled locomotion
- 6 Robot Kinematics
  - Forward and Inverse Kinematics
- 7 Representation Of The Environment
- 8 Organizational

09<sup>th</sup> Nov. 2023

Robot Programming with ROS

4. Motors and Kinematics

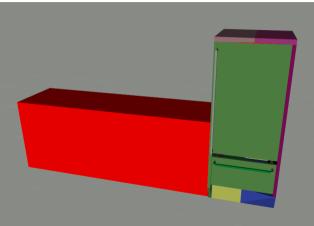
Mathematics & Computer Science



Robot Programming with ROS Arthur Niedźwiecki, 4. Motors and Kinematics Stefan Eirich

Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science

### The Representation Of The Environment





Institute for Artificial Intelligence, University of Bremen CC BY-SA 4.0: https://creativecommons.org/licenses/by-sa/4.0/



# Summary

Robot Programming with ROS Arthur Niedźwiecki, 4. Motors and Kinematics Stefan Eirich

Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science

- Configuration and task space
- Robot kinematics



# Overview

- 1 What makes a robot? Links and joints
- 2 Actuators
- 3 Robot Arms
- 4 Gripper
- 6 Mobile Bases Wheeled locomotio
- 6 Robot Kinematics
  - Forward and Inverse Kinematics
- 7 Representation Of The Environment
- 8 Organizational

Robot Programming with ROS Arthur Niedźwiecki, 4. Motors and Kinematics Stefan Eirich

Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science



Discord



Artificial Intelligence Robot Programming with ROS Arthur Niedźwiecki,

4. Motors and Kinematics

Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science



Follow this to our Discord server. Link open until 15.11.23 https://discord.gg/yzRurKba



Robot Programming with ROS Arthur Niedźwiecki,

4. Motors and Kinematics

Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science

## Assignment and dates

• Assignment 4:

 $\verb+https://github.com/artnie/rpwr-assignments$ 

- Grades: 8 points for this assignment
- Due: 15.11., 23:59 AM German time
- Next class: 16.11., 14:00



#### Robot Programming with ROS Arth 4. Motors and Kinematics Ste

Arthur Niedźwiecki, Stefan Eirich 09<sup>th</sup> Nov. 2023 Faculty 03 Mathematics & Computer Science

# Evaluation

#### Thanks for your attention!

Special thanks to the IAI team for the content of this lecture!



https://forms.gle/iZyKqLCxsrwBU3XZ6