UNIVERSITÀ di VERONA

Dipartimento di INFORMATICA

ALTAIR robotics lab

Robotica 2019/2020: Cinematica diretta in ROS

Diego Dall’Alba

Altair robotics lab
Department of computer science – University of Verona, Italy

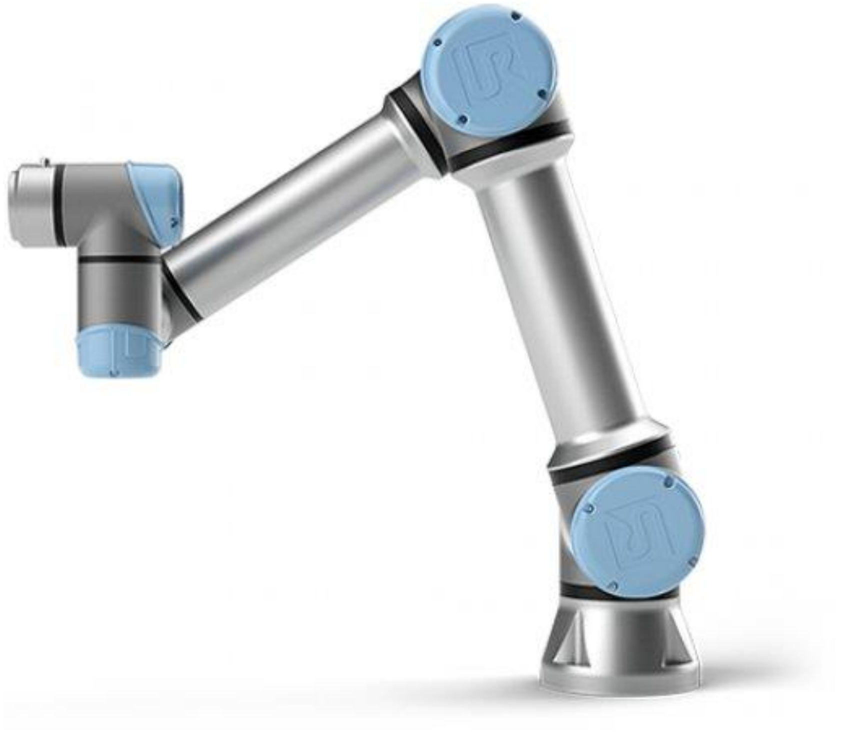
Summary

- UR5 Example again
- Kinematic modelling in ROS
- Tools
- UR5 Ros model

Universal Robot UR5 Robotic Manipulator

- Industrial manipulator
- Suitable for collaborative applications
- First (successful) commercial system providing cost effective solution
- Used in many industrial and research sectors
- See some example:

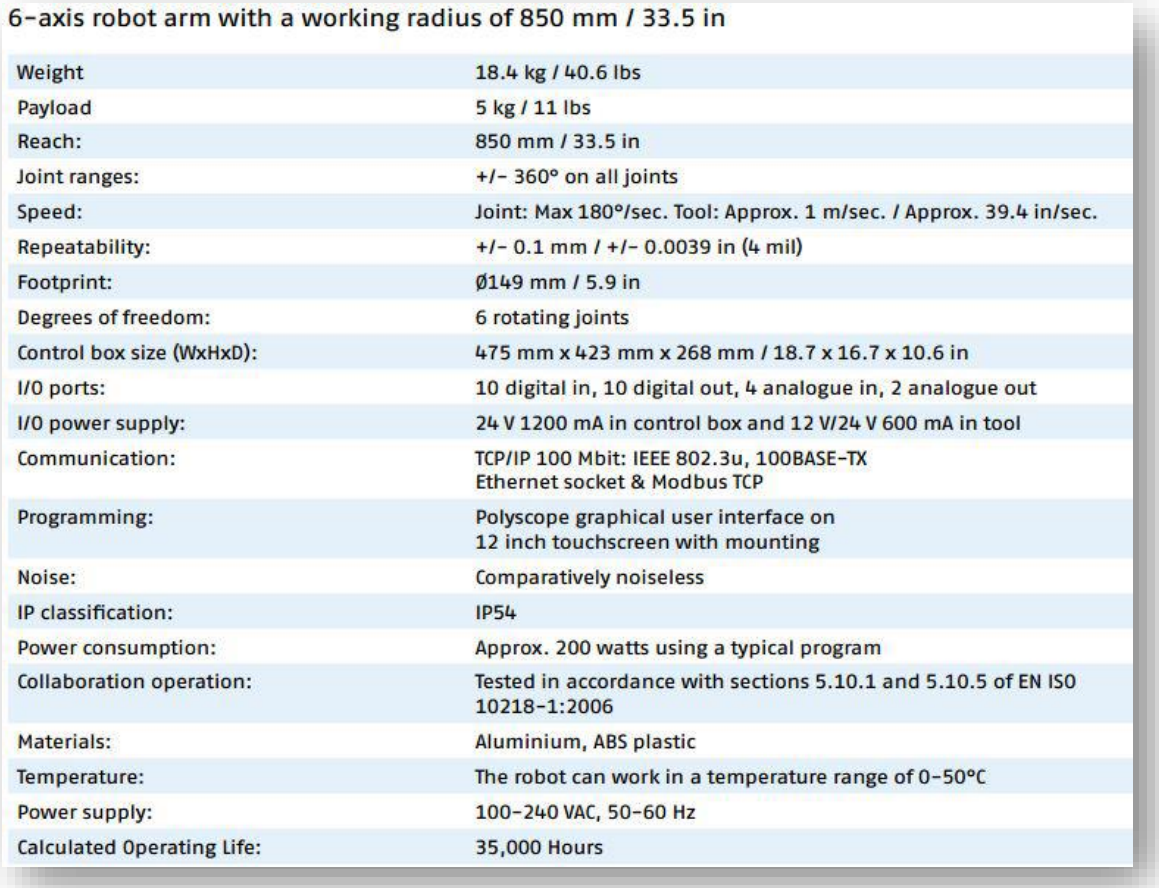
https://www.youtube.com/watch?v=pIcxOGo7ieU&t=134s



Universal Robot UR5 Specifications

- For more information please check the datasheet:

https://www.universal-robots.com/media/50588/ur5_en.pdf

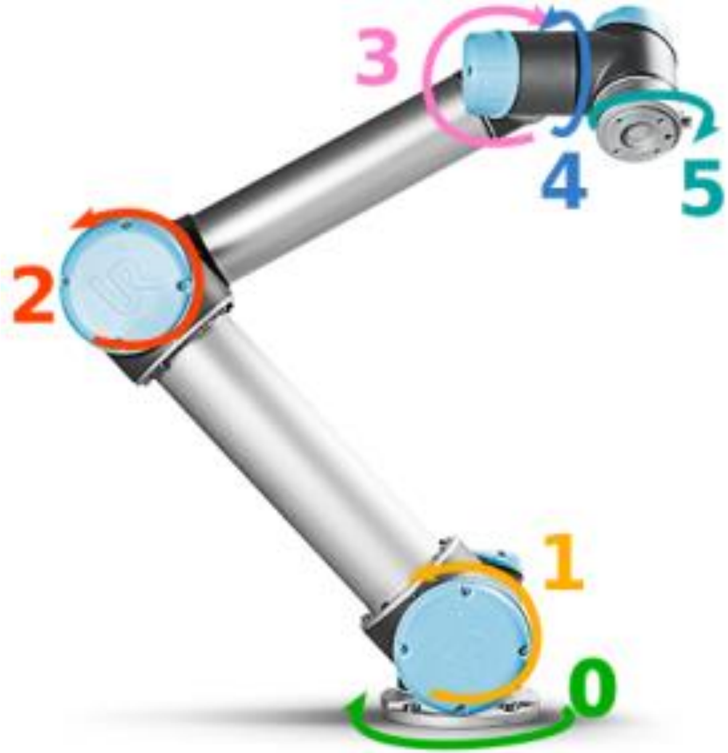


6-axis robot arm with a working radius of 850 mm / 33.5 in

Weight	18.4 kg / 40.6 lbs
Payload	5 kg / 11 lbs
Reach:	850 mm / 33.5 in
Joint ranges:	+/- 360° on all joints
Speed:	Joint: Max 180°/sec. Tool: Approx. 1 m/sec. / Approx. 39.4 in/sec.
Repeatability:	+/- 0.1 mm / +/- 0.0039 in (4 mil)
Footprint:	Ø149 mm / 5.9 in
Degrees of freedom:	6 rotating joints
Control box size (WxHxD):	475 mm x 423 mm x 268 mm / 18.7 x 16.7 x 10.6 in
I/O ports:	10 digital in, 10 digital out, 4 analogue in, 2 analogue out
I/O power supply:	24 V 1200 mA in control box and 12 V/24 V 600 mA in tool
Communication:	TCP/IP 100 Mbit: IEEE 802.3u, 100BASE-TX Ethernet socket & Modbus TCP
Programming:	Polyscope graphical user interface on 12 inch touchscreen with mounting
Noise:	Comparatively noiseless
IP classification:	IP54
Power consumption:	Approx. 200 watts using a typical program
Collaboration operation:	Tested in accordance with sections 5.10.1 and 5.10.5 of EN ISO 10218-1:2006
Materials:	Aluminium, ABS plastic
Temperature:	The robot can work in a temperature range of 0-50°C
Power supply:	100-240 VAC, 50-60 Hz
Calculated Operating Life:	35,000 Hours

UR5 Joints description

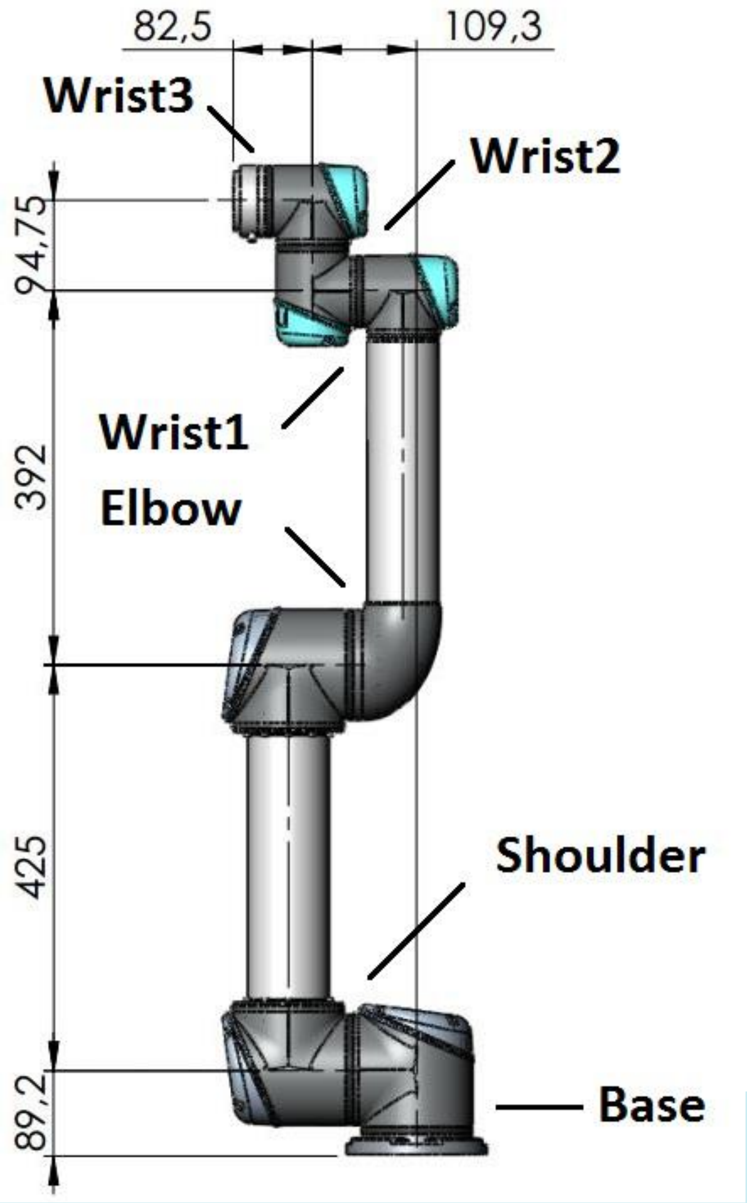
- 6R
- Non-spherical wrist
- Joint index is starting from 0 in this image, ~~but in our convention is starting from 1~~



UR5 Mechanical specs and DH parameters

i	α_i	a_i	d_i	θ_i
1	$-\pi/2$	0	$d_1 = 89.2$	$\theta_1 = 0$
2	0	$a_2 = -425$	0	$\theta_2 = \pi/2$
3	0	$a_3 = -392$	0	$\theta_3 = 0$
4	$\pi/2$	0	$d_4 = 109.3$	$\theta_4 = -\pi/2$
5	$-\pi/2$	0	$d_5 = 94.75$	$\theta_5 = 0$
6	0	0	$d_6 = 82.5$	$\theta_6 = 0$

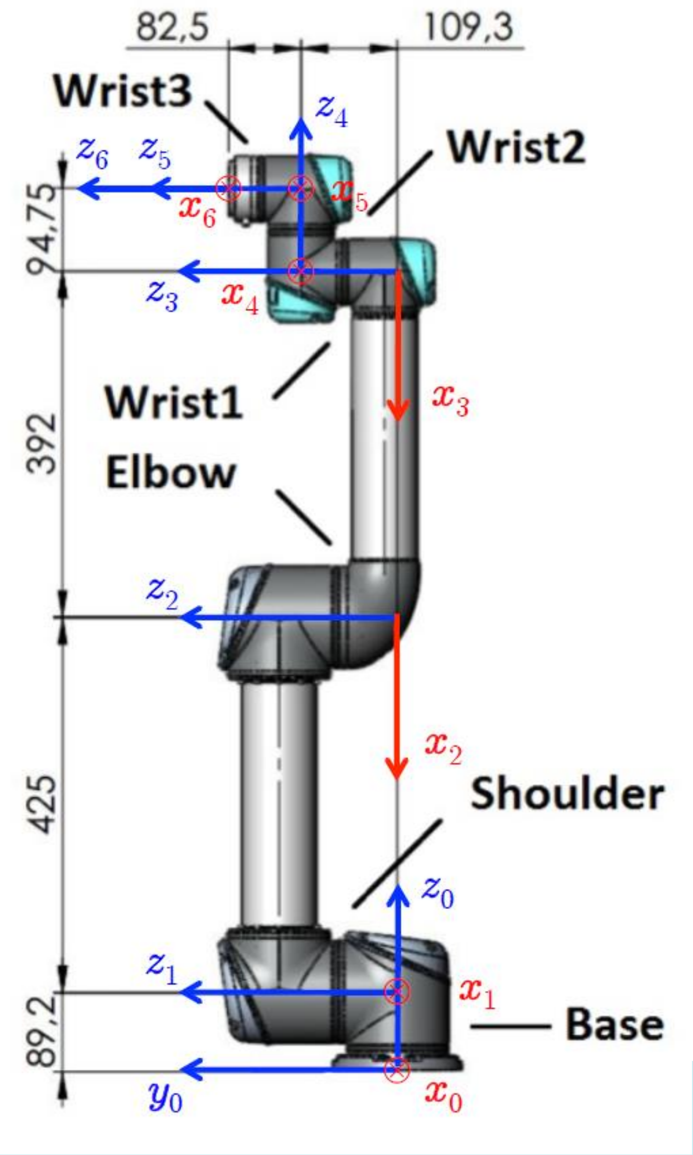
DH parameters (in mm or rad), with the value of θ in the shown configuration.



UR5 Mechanical specs and DH parameters

i	α_i	a_i	d_i	θ_i
1	$-\pi/2$	0	$d_1 = 89.2$	$\theta_1 = 0$
2	0	$a_2 = -425$	0	$\theta_2 = \pi/2$
3	0	$a_3 = -392$	0	$\theta_3 = 0$
4	$\pi/2$	0	$d_4 = 109.3$	$\theta_4 = -\pi/2$
5	$-\pi/2$	0	$d_5 = 94.75$	$\theta_5 = 0$
6	0	0	$d_6 = 82.5$	$\theta_6 = 0$

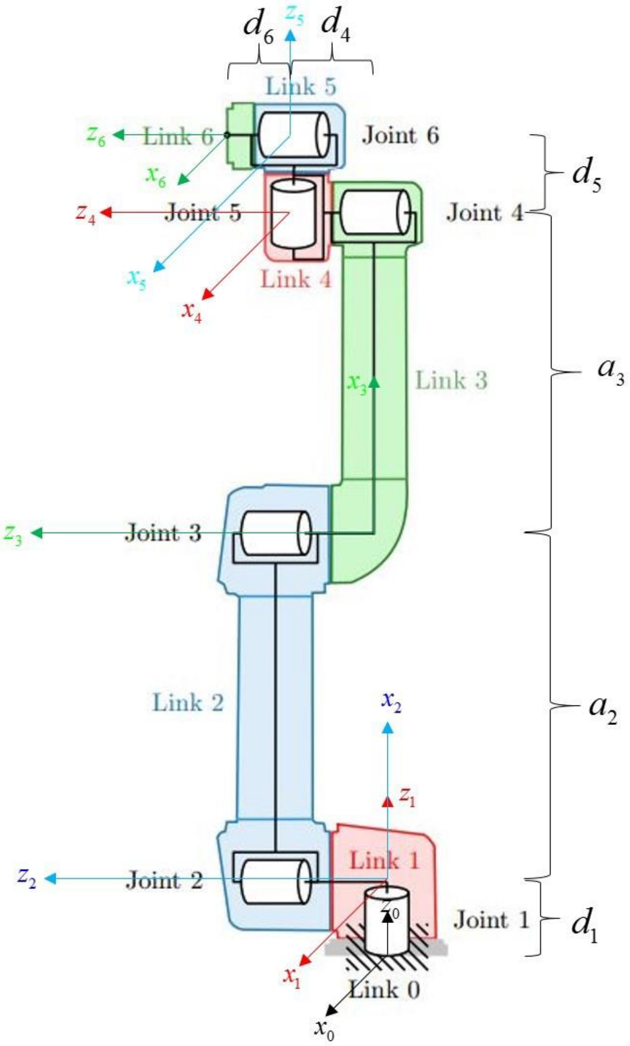
DH parameters (in mm or rad), with the value of θ in the shown configuration.



Better UR5 DH parameters ☺

i	a_i	α_i	d_i	θ_i
0	0	0	-	-
1	0	$\frac{\pi}{2}$	0.08916	θ_1
2	0.425	0	0	θ_2
3	0.39225	0	0	θ_3
4	0	$\frac{\pi}{2}$	0.10915	θ_4
5	0	$-\frac{\pi}{2}$	0.09456	θ_5
6	-	-	0.0823	θ_6

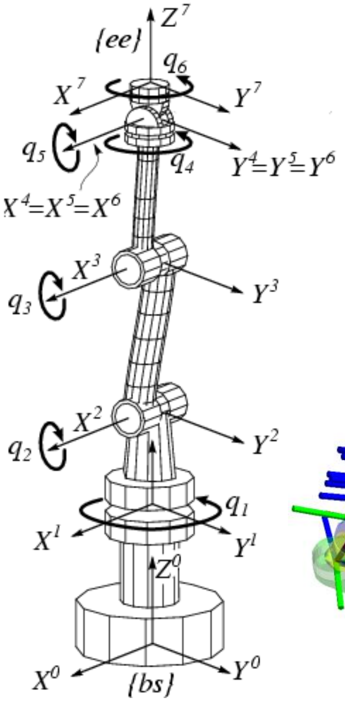
DH parameters (in meter or rad), for more details please check: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7844896

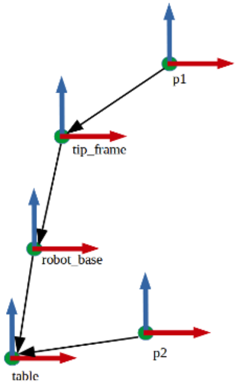


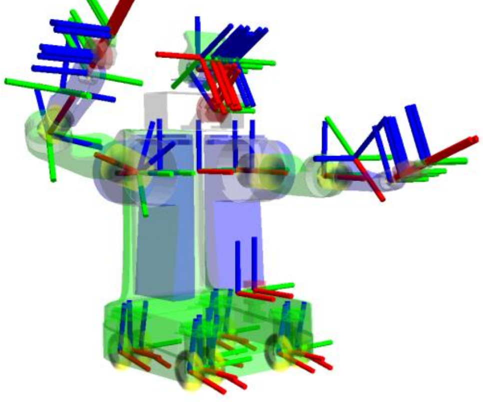
Kinematic modeling in ROS

Ros provide powerful tools for kinematic modelling of a robotic manipulator, for example:

- **Tf:** representing and manipulating frame transformation (over time)
- **URDF+XACRO**: XML based robotic description
- **RViz:** ROS 3D visualizer
- **KDL:** Kinematics and Dynamics Library (from Orocos project)



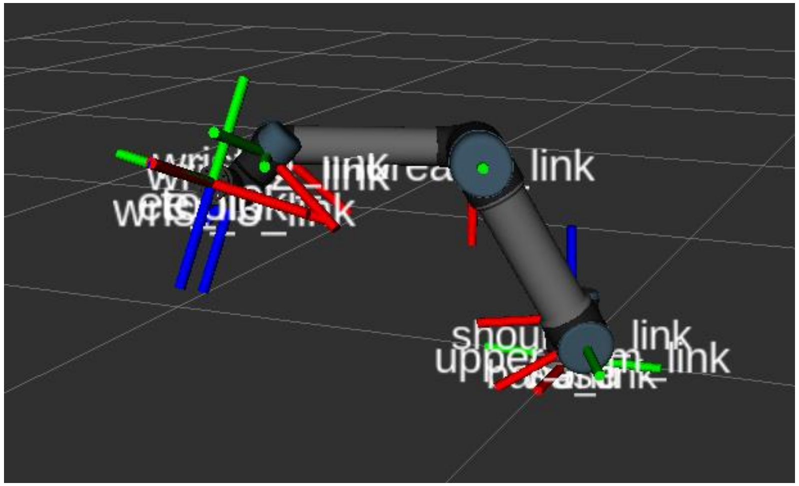




UR5 in ROS

RUN this example:

```
roslaunch ur_description view_ur5.launch
```



Preliminary steps:

In the source space of your local catkin workspace please clone this git repository:

```
git clone -b $ROS_DISTRO-devel https://github.com/rosindustrial/universal_robot.git
```

Check for required dependencies and install them:

```
rosdep update
rosdep install --rosdistro $ROS_DISTRO --ignore-src --from-paths src
```

Compile:

```
catkin_make
```

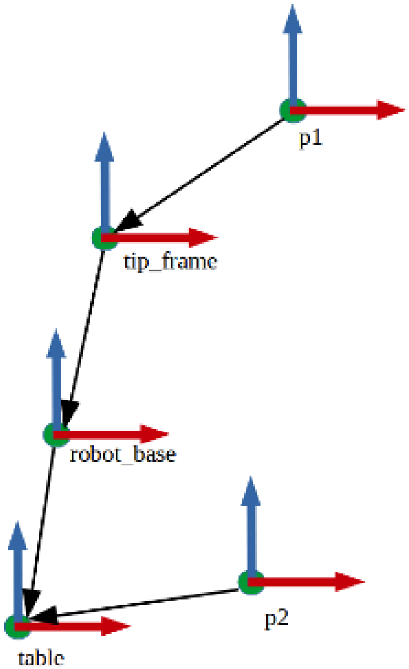
UR5 Exercises

1. Run the previous example
2. Using the provided slide commands move the robotic manipulator (please keep attention to direction of the rotation)
3. Disable the robotic visual model form Rviz interface and check visualized Tf (RGB axis)
4. Find Tf non respecting DH conventions, you could hide some of the visualized Tf to ease this step

URDF+Xacro

Unified Robot Description Format (**URDF**) is an XML format for representing a robot model.
It enable to describe kinematic, visual and dynamic properties of a manipulator.
http://wiki.ros.org/urdf

Xacro is an XML macro language: enable construction of shorter and more readable XML files by using macros that expand to larger XML expressions.
http://wiki.ros.org/xacro

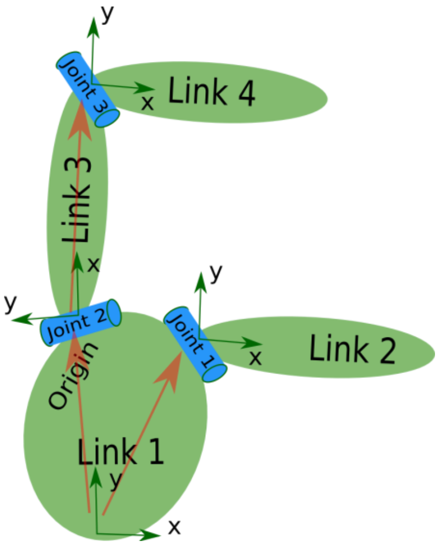


ROS provides parsing tools for reading and checking URDF files:
http://wiki.ros.org/urdf/Tutorials

URDF simple example

Please follow this tutorial:

http://wiki.ros.org/urdf/Tutorials/Create%20your%20own%20urdf%20file



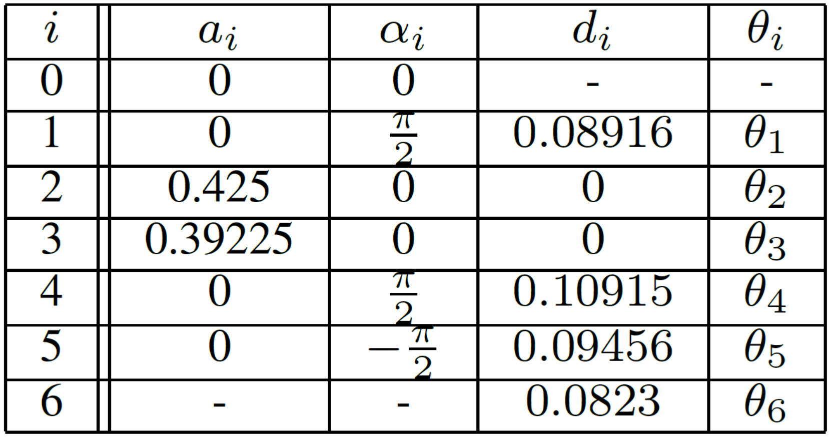
```
<robot name="test_robot">
  <link name="link1" />
  <link name="link2" />
  <link name="link3" />
  <link name="link4" />

  <joint name="joint1" type="continuous">
    <parent link="link1"/>
    <child link="link2"/>
  </joint>
  <joint name="joint2" type="continuous">
    <parent link="link1"/>
    <child link="link3"/>
  </joint>
  <joint name="joint3" type="continuous">
    <parent link="link3"/>
    <child link="link4"/>
  </joint>
</robot>
```

URDF UR5

You could check the following file for a complete description:

$HOME/catkin_ws/src/universal_robot/ur_description/urdf/ur5_joint_limited_robot.urdf.xacro



i	a_i	α_i	d_i	θ_i
0	0	0	-	-
1	0	$\frac{\pi}{2}$	0.08916	θ_1
2	0.425	0	0	θ_2
3	0.39225	0	0	θ_3
4	0	$\frac{\pi}{2}$	0.10915	θ_4
5	0	$-\frac{\pi}{2}$	0.09456	θ_5
6	-	-	0.0823	θ_6

```
<!-- ********** KINEMATIC PROPERTIES (JOINTS)
********** -->
<joint name="world_joint" type="fixed">
<parent link="world"/>
<child link="base_link"/>
<origin rpy="0.0 0.0 0.0" xyz="0.0 0.0 0.0"/>
</joint>
<joint name="joint1" type="continuous">
<parent link="base_link"/>
<child link="link1"/>
<origin rpy="0.0 0.0 0.0" xyz="0.0 0.0 0.089159"/>
<axis xyz="0 0 1"/>
</joint>
<joint name="joint2" type="continuous">
<parent link="link1"/>
<child link="link2"/>
<origin rpy="“???" xyz="“???"/>
<axis xyz="0 1 0"/>
</joint>
```