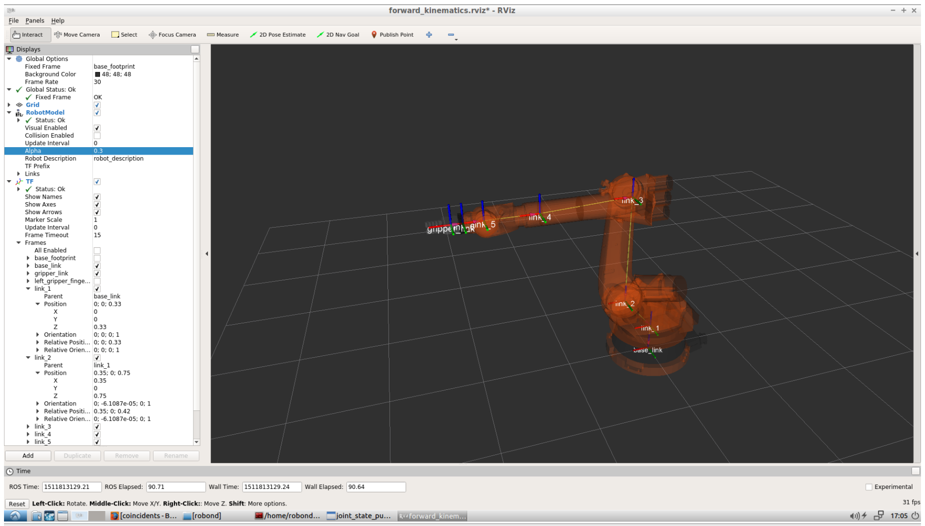
Project: Kinematics Pick & Place

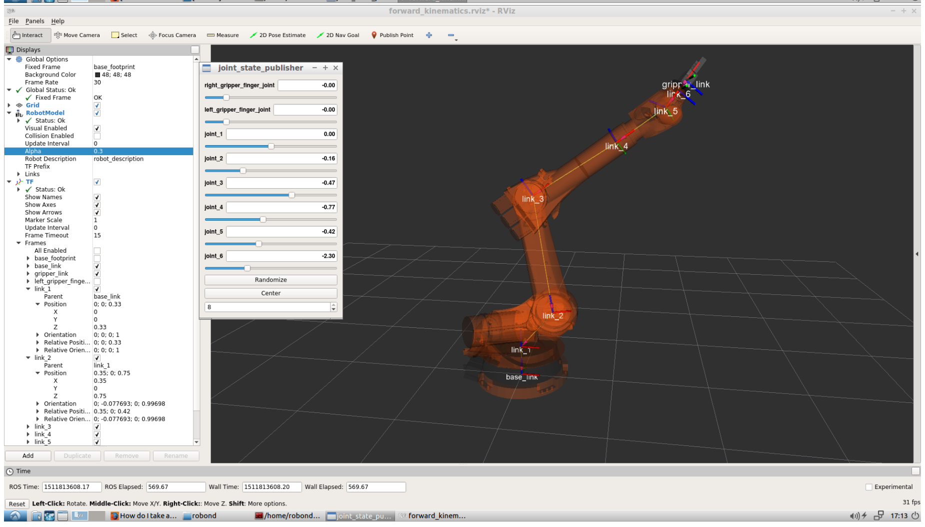
Kinematic Analysis

1. Run the forward__kinematics demo and evaluate the kr210.urdf.xacro file to perform kinematic analysis of Kuka KR210 robot and derive its DH parameters.



```
Global Options
  Fixed Frame            base_footprint
  Background Color       48; 48; 48
  Frame Rate             30
Global Status: Ok
  Fixed Frame            OK
Grid                     ✓
RobotModel               ✓
  Status: Ok
  Visual Enabled         ✓
  Collision Enabled
  Update Interval        0
  Alpha                  0.3
  Robot Description      robot_description
  TF Prefix
  Links
TF                       ✓
  Status: Ok
  Show Names             ✓
  Show Axes              ✓
  Show Arrows            ✓
  Marker Scale           1
  Update Interval        0
  Frame Timeout          15
  Frames
    All Enabled
    base_footprint
    base_link            ✓
    gripper_link         ✓
    left_gripper_finge...
    link_1               ✓
      Parent             base_link
      Position           0; 0; 0.33
        X                0
        Y                0
        Z                0.33
      Orientation        0; 0; 0; 1
      Relative Positi... 0; 0; 0.33
      Relative Orien...  0; 0; 0; 1
    link_2               ✓
      Parent             link_1
      Position           0.35; 0; 0.75
        X                0.35
        Y                0
        Z                0.75
      Orientation        0; -6.1087e-05; 0; 1
      Relative Positi... 0.35; 0; 0.42
      Relative Orien...  0; -6.1087e-05; 0; 1
    link_3               ✓
    link_4               ✓
    link_5               ✓

ROS Time: 1511813129.21   ROS Elapsed: 90.71   Wall Time: 1511813129.24   Wall Elapsed: 90.64
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click:: Move Z. Shift: More options.
```

forward_kinematics.rviz* - RViz
File Panels Help
Interact Move Camera Select Focus Camera Measure 2D Pose Estimate 2D Nav Goal Publish Point
Displays
Add Duplicate Remove Rename
Time
Reset
Experimental
31 fps
gripper_link
link_5
link_4
link_3
link_2
link_1
base_link
[coincidents - B... [robond] /home/robond... joint_state_pu... forward_kinem...
17:05

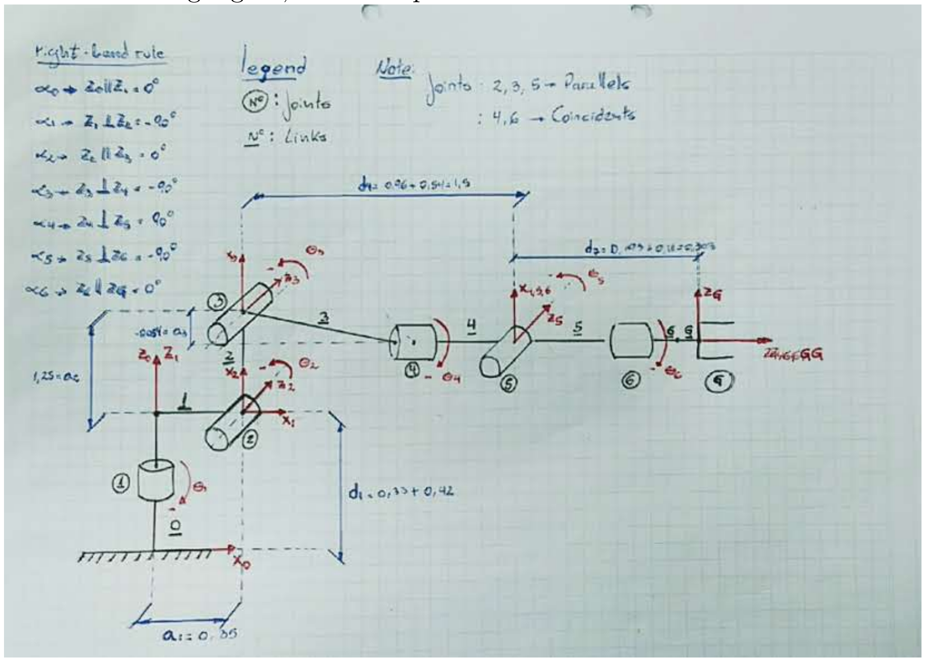


```
Global Options
  Fixed Frame            base_footprint
  Background Color       48; 48; 48
  Frame Rate             30
Global Status: Ok
  Fixed Frame            OK
Grid                     ✓
RobotModel               ✓
  Status: Ok
  Visual Enabled         ✓
  Collision Enabled
  Update Interval        0
  Alpha                  0.3
  Robot Description      robot_description
  TF Prefix
  Links
TF                       ✓
  Status: Ok
  Show Names             ✓
  Show Axes              ✓
  Show Arrows            ✓
  Marker Scale           1
  Update Interval        0
  Frame Timeout          15
  Frames
    All Enabled
    base_footprint
    base_link            ✓
    gripper_link         ✓
    left_gripper_finge...
    link_1               ✓
      Parent             base_link
      Position           0; 0; 0.33
        X                0
        Y                0
        Z                0.33
      Orientation        0; 0; 0; 1
      Relative Positi... 0; 0; 0.33
      Relative Orien...  0; 0; 0; 1
    link_2               ✓
      Parent             link_1
      Position           0.35; 0; 0.75
        X                0.35
        Y                0
        Z                0.75
      Orientation        0; -0.077693; 0; 0.99698
      Relative Positi... 0.35; 0; 0.42
      Relative Orien...  0; -0.077693; 0; 0.99698
    link_3               ✓
    link_4               ✓
    link_5               ✓

joint_state_publisher
  right_gripper_finger_joint   -0.00
  left_gripper_finger_joint    -0.00
  joint_1                       0.00
  joint_2                      -0.16
  joint_3                      -0.47
  joint_4                      -0.77
  joint_5                      -0.42
  joint_6                      -2.30
  Randomize
  Center
  8

ROS Time: 1511813608.17   ROS Elapsed: 569.67   Wall Time: 1511813608.20   Wall Elapsed: 569.67
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click:: Move Z. Shift: More options.
```

forward_kinematics.rviz* - RViz
File Panels Help
Interact Move Camera Select Focus Camera Measure 2D Pose Estimate 2D Nav Goal Publish Point
Displays
Add Duplicate Remove Rename
Time
Reset
Experimental
31 fps
gripper_link
link_6
link_5
link_4
link_3
link_2
link_1
base_link
How do I take a... robond /home/robond... joint_state_pu... forward_kinem...
17:13

From the forward__kinematic demonstration, the first step was to analyze the URDF.xacro file. Once it is known that the same information in the URDF file is available in RViz in the visualization tree within the TF parent, then the distances between the unions were started from there to create the DH parameter table. The demonstration was also used to analyze the signs of the rotation of each joint.

In the following figure, the DH parameters of the robot are schematized.


Right-hand rule
$\alpha_0 \rightarrow Z_0 \parallel Z_1 = 0°$
$\alpha_1 \rightarrow Z_1 \perp Z_2 = -90°$
$\alpha_2 \rightarrow Z_2 \parallel Z_3 = 0°$
$\alpha_3 \rightarrow Z_3 \perp Z_4 = -90°$
$\alpha_4 \rightarrow Z_4 \perp Z_5 = 90°$
$\alpha_5 \rightarrow Z_5 \perp Z_6 = -90°$
$\alpha_6 \rightarrow Z_6 \parallel Z_G = 0°$
Legend
Nº : Joints
Nº : Links
Note:
Joints : 2, 3, 5 → Parallels
: 4, 6 → Coincidents
$d_4 = 0.96 + 0.54 = 1.5$
$d_7 = 0.193 + 0.11 = 0.303$
$-0.054 = a_3$
$1.25 = a_2$
$d_1 = 0.33 + 0.42$
$a_1 = 0.35$
Z_0 Z_1 X_0 X_1 X_2 Z_2 X_3 Z_3 $X_{4,5,6}$ Z_5 Z_G $Z_{4,6,G}$
θ_1 θ_2 θ_3 θ_4 θ_5 θ_6

The resulting Table:

Joint	Links	alpha(i-1)	a(i-1)	d(i)	theta(i)
1	0->1	0	0	0.75	θ_1
2	1->2	$-\pi/2$	0.35	0	$-\pi/2 + \theta_2$
3	2->3	0	1.25	1.5	θ_3
4	3->4	$-\pi/2$	-0.54	0	θ_4
5	4->5	$\pi/2$	0	0	θ_5
6	5->6	$-\pi/2$	0	0	θ_6
7	6->EE	0	0	0.303	0

- alpha(i-1): twist angle, angle between axis Z(i-1) and Z(i) measured about axis X(i-1)
- a(i-1): link length, distance from axis Z(i-1) to Z(i) measured along axis X(i-1)
- d(i): link offset, distance from axis X(i-1) to X(i) measured along axis Z(i)
- theta(i): joint angle, angle between axis X(i-1) and X(i) measured about axis Z(i)

As it was explained in the lessons, the only thing that need attention is the offset of Theta 2 by -90 degrees.

---

2. Using the DH parameter table you derived earlier, create individual transformation matrices about each joint. In addition, also generate a generalized homogeneous transform between base_link and gripper_link using only end-effector(gripper) pose.

> The code related to solve the Inverse Kinematic problem is in `FK_class.py`.

Following the modified parameters of Denavit-Hartenberg, the general form of the homogeneous transformation ${}_iT^{i-1}$ is defined as:

$${}_iT^{i-1} = Rot_x(\alpha_{(i-1)})Trl_x(a_{i-1})Rot_z(\theta_{(i)})Trl_z(d_i)$$

Resulting:

$${}_iT^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{(i-1)} \\ \sin(\theta_i)\cos\left(\alpha_{(i-1)}\right) & \cos\left(\alpha_{(i-1)}\right)\cos(\theta_i) & -\sin\left(\alpha_{(i-1)}\right) & -d_i\sin\left(\alpha_{(i-1)}\right) \\ \sin\left(\alpha_{(i-1)}\right)\sin(\theta_i) & \sin\left(\alpha_{(i-1)}\right)\cos(\theta_i) & \cos\left(\alpha_{(i-1)}\right) & d_i\cos\left(\alpha_{(i-1)}\right) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

From the previous definition, it is possible to obtain the transformation matrices for the board, obtained:

$${}_0T^1 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0.75 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_1T^2 = \begin{bmatrix} \sin(\theta_2) & \cos(\theta_2) & 0 & 0.35 \\ 0 & 0 & 1 & 0 \\ \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_2T^3 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & 1.25 \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_3T^4 = \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & -0.054 \\ 0 & 0 & 1 & 1.5 \\ -\sin(\theta_4) & -\cos(\theta_4) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_4T^5 = \begin{bmatrix} \cos(\theta_5) & -\sin(\theta_5) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin(\theta_5) & \cos(\theta_5) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$ {}_5T^6 = \begin{bmatrix} \cos(\theta_5) & -\sin(\theta_5) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin(\theta_5) & \cos(\theta_5) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}_6T^{EE} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.303 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Finally, to obtain the transformation matrix from the base link to the final effector, it is necessary to post-multiply each matrix T in the following way:

$$ {}_0T^{EE} = {}_0T^1 \; {}_1T^2 \; {}_2T^3 \; {}_3T^4 \; {}_4T^5 \; {}_5T^6 \; {}_6T^{EE} $$

> I do not put the resulting matrix because it is a huge complex matrix, full of trigonometric operations, anything but intuitive.

---

3. Decouple Inverse Kinematics problem into Inverse Position Kinematics and inverse Orientation Kinematics

Solve the inverse kinematic model of an anthropomorphic robotic manipulator it is not as easy as the direct kinematic model. By definition, the dexterous workspace of a robotic manipulator is the volume of space that the robot end-effector can reach with an arbitrary orientation. That means that any point of this workspace have multiple solutions. There also are points that can be reached with different configurations of the joints with the same orientation.

One strategy to attack this problem is to divide it into two sub-problems: Inverse Position Kinematics and inverse Orientation Kinematics.

Inverse Position Kinematics

For solving the Position, it is needed to calculate the angles of the first three joints θ_1, θ_2, θ_3 which define the position of the wrist center (**wc**).

The **wc** is gotten as it was explained in the Inverse Kinematics lessons, implementing the following equation:

$$ {}^0\mathbf{r}_{WC/0} = {}^0\mathbf{r}_{EE/0} - d\ {}^0\mathbf{R}_6 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} - d\ {}^0\mathbf{R}_6 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} $$

Then, now with the **wc**, be can calculate the θ_1 as follow:

$\theta_1 = atan2(wc_y, wc_x)$

That give us at least two solution, the clockwise direction and the counterclockwise direction.

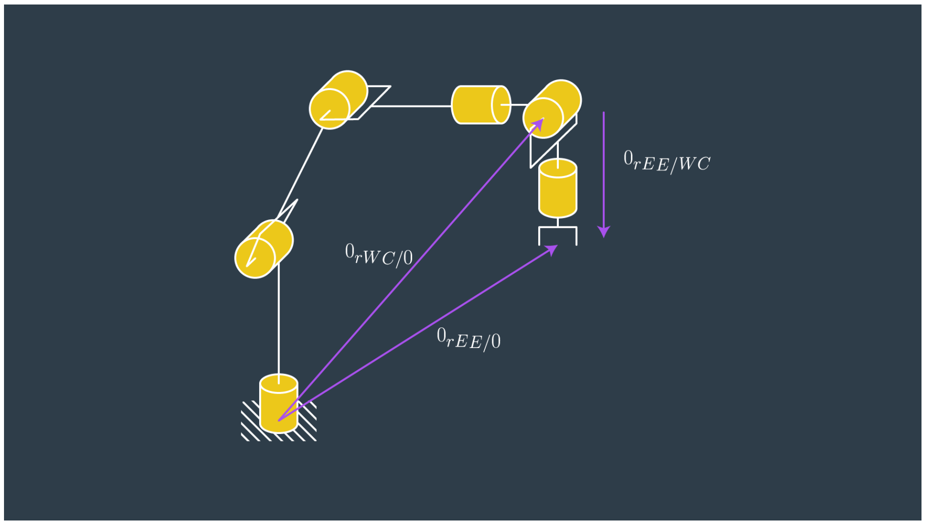


Figure 1: Schema for decouple Inverse Kinematics.

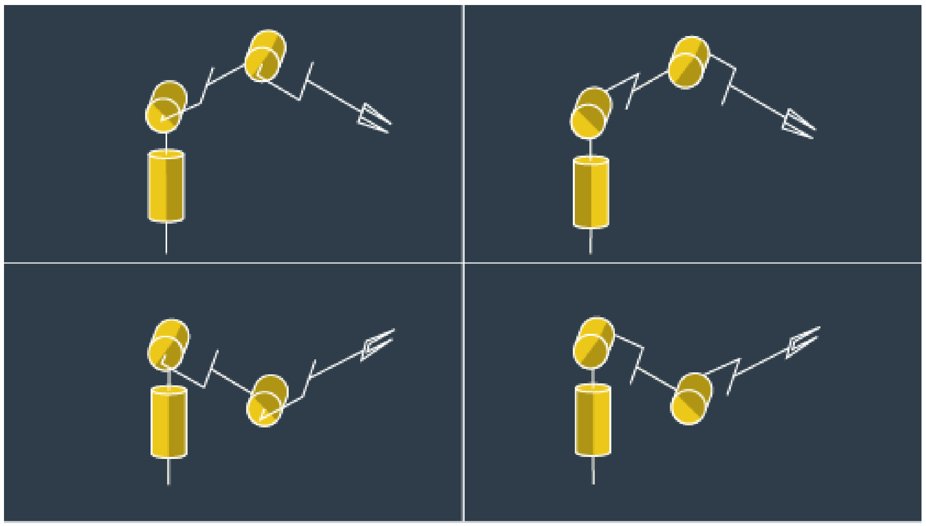


Figure 2: Multiple solutions IK.

Here it appears the first problem. By only know the **wc**, it is not possible to determine if the arm is working backwards, as show the columns to the picture above.

Later to calculate θ_2 and θ_3, we use the cosine law to solve triangle shown in the figure below.

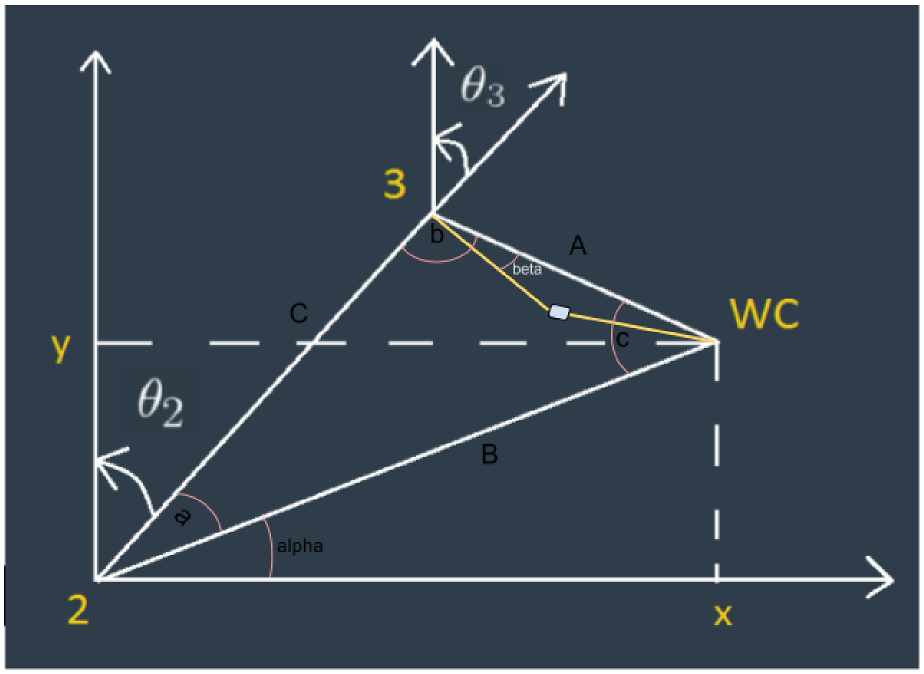


Figure 3: Diagram for cosine law.

The direct solution of the triangle only gives us a solution, because we really do not know a priori if the elbow is in the up or down position, as shown in the first picture with the multiple solutions.

Then:

$\theta_2 = \pi/2 - a - \alpha$

$\theta_2 = \pi/2 - b - \beta$

Inverse Orientation Kinematics

To obtain the angles of θ_4, θ_5 and θ_6 the procedure explained in the lesson of inverse kinematics was followed. Now that you have θ_1, θ_2 and θ_3 you calculate the one of ${}_0\mathbf{T}^3$ to extract the ${}_0\mathbf{R}^3$, then it transposes to finally get ${}_3\mathbf{R}^0$. As we also know, that the rotation matrix used early to define the orientation of the final effector is equal to ${}_0\mathbf{R}^E E$.

Then, working with the simbolic matrices:

${}_3\mathbf{R}^6 =_3 \mathbf{R}^0 \ {}_0\mathbf{R}^6$

Resulting:

$${}_3\mathbf{R}^6 = \begin{bmatrix} -s(\theta_4)s(\theta_6)+c(\theta_4)c(\theta_5)c(\theta_6) & -s(\theta_4)c(\theta_6)-s(\theta_6)c(\theta_4)c(\theta_5) & -s(\theta_5)c(\theta_4) \\ s(\theta_5)c(\theta_6) & -s(\theta_5)s(\theta_6) & c(\theta_5) \\ -s(\theta_4)c(\theta_5)c(\theta_6)-s(\theta_6)c(\theta_4) & s(\theta_4)s(\theta_6)c(\theta_5)-c(\theta_4)(\theta_6) & s(\theta_4)s(\theta_5) \end{bmatrix}$$

With $\mathbf{R}_{roll,pitch,yaw} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{21} & r_{22} & r_{23} \end{bmatrix}$ it can be defined the following equations:

$\theta_4 = atan2(r_{33}, -r_{31})$

$\theta_5 = atan2(\sqrt{1-{r_{23}}^2}, r_{23})$

$\theta_6 = atan2(-r_{22}, r_{21})$

> If $\theta_5 = 0$, a degree of freedom is lost, since it is a singularity of the robotic arm and that implies infinite solutions by combinations of θ_4 and θ_6.
>
> Another thing that must be taken into account is that θ_4 and θ_6 can take from -350ž to 350ž as show the URDF file, so there will always be more than one solution for inverse-orientation kinematics, not only by the sign of rotation but also the almost complete rotation that both joints can reach.

It is worth mentioning that to compensate for the differences in the difference in the orientation of the end effector, a rotation matrix of compensation for the discrepancy between DH parameters and Gazebo, R_{corr} was calculated, convincing a rotation in X of 180º and rotation in Y of -90º.

Project Implementation

For the implementation of the project, 2 main scripts were written:

`FK_class.py`: Here is a class that attempts to encapsulate all the variables and methods involved in the resolution of the direct kinematic model of the robot.

`IK_class.py`: Also a class that has all the variables and methods developed to solve the inverse kinematic model of the robot.

First, the code was tested and the necessary code was implemented in the 'IK_debug.py' provided. The debugger allowed working paying more attention to the problem of the multiple solutions of the inverse kinematics as the different test cases were selected. In addition, some test cases were added with the help of 'fordward_kinematic.launch'. Also, it was used the FK model to compare the resulting positions of the end-effector as it is recommended in comments of the script.

As a conclusion to this procedure, we recognized the four possible solutions of the * inverse position kinematics *, which given the geometry of KR210, was not trivial. One of the main factors was the non-alignment (parameter a1 DH) between joint 1 and joint 2, which clearly determines the difference between working forward or backward of the robot. The limits of the joints minimize the space accessible to the back of the robot. Anyway, that would not be a problem in the goal of the pick and place problem, because the robot always works in the front configuration. A similar criterion was taken on the configuration of the elbow, because there is no need to take an elbow configuration downwards.

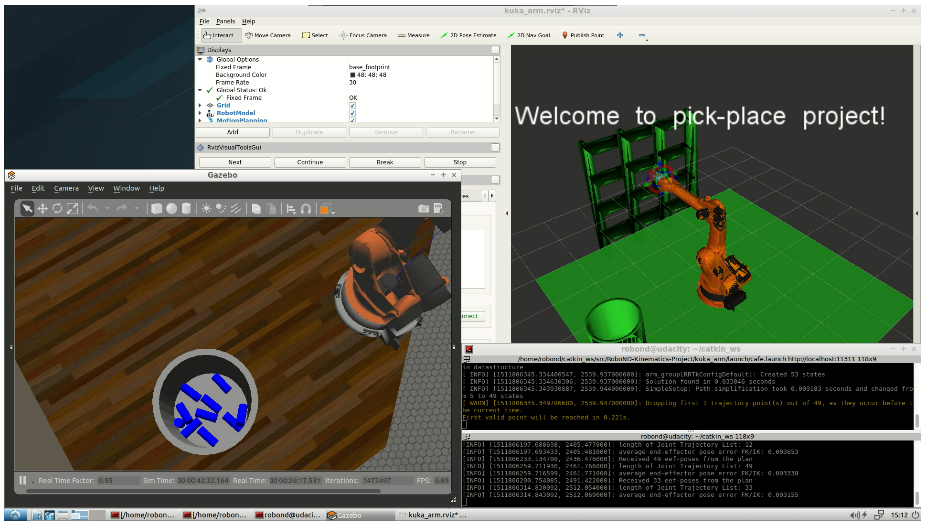
Global Conclusion

It was possible to carry out the complete kinematic analysis of the KR210 robot, calculating both the direct and the indirect kinetic model. The implementation could be validated, first by means of the debugging script and then by performing the full simulation using the `IK_server.py` developed. Although the solution developed is not perfect, the requested requirements were met and an error curve (below) recorded during the simulation is presented. On the other hand, a better understanding of the ROS framework and all the tools used here has been achieved.

Discuss

- It was decided to create the two classes with the aim of improving performance, but even so it seems to be slow.
- The reason could not be detected, but the suit solver had trouble calculating the trajectory of the center position on the shelf. I do not think the IK_solver has anything to do with the problem at all.
- An algorithm was not implemented for the multiple orientation solutions. But I think the best way to attack the problems is by taking into account the latest orientation configuration to get the minimum change of the wrist joints.
- In some cases, the trajectory calculated by MoveIt is long and illogical and cause problems because the criteria taken to chose the joint 1 angle.
- The major error is represented when the robot is close to a singularity, for example, when it is almost extended, so that behavior seems acceptable.

Example Video



```
/home/robond/catkin_ws/src/RoboND-Kinematics-Project/kuka_arm/launch/cafe.launch http://localhost:11311 118x9
in datastructure
[ INFO] [1511806345.334460547, 2539.937000000]: arm_group[RRTkConfigDefault]: Created 53 states
[ INFO] [1511806345.334630306, 2539.937000000]: Solution found in 0.033046 seconds
[ INFO] [1511806345.343930887, 2539.944000000]: SimpleSetup: Path simplification took 0.009183 seconds and changed fro
m 5 to 49 states
[ WARN] [1511806345.349786680, 2539.947000000]: Dropping first 1 trajectory point(s) out of 49, as they occur before t
he current time.
First valid point will be reached in 0.221s.
```

```
robond@udacity: ~/catkin_ws 118x9
[INFO] [1511806197.688698, 2405.477000]: length of Joint Trajectory List: 12
[INFO] [1511806197.693433, 2405.481000]: average end-effector pose error FK/IK: 0.003653
[INFO] [1511806233.134708, 2436.476000]: Received 49 eef-poses from the plan
[INFO] [1511806259.711930, 2461.766000]: length of Joint Trajectory List: 49
[INFO] [1511806259.716599, 2461.771000]: average end-effector pose error FK/IK: 0.003338
[INFO] [1511806290.754085, 2491.422000]: Received 33 eef-poses from the plan
[INFO] [1511806314.830092, 2512.054000]: length of Joint Trajectory List: 33
[INFO] [1511806314.843092, 2512.069000]: average end-effector pose error FK/IK: 0.003155
```

Figure 4: Results 8/10 + 2 (10/12)

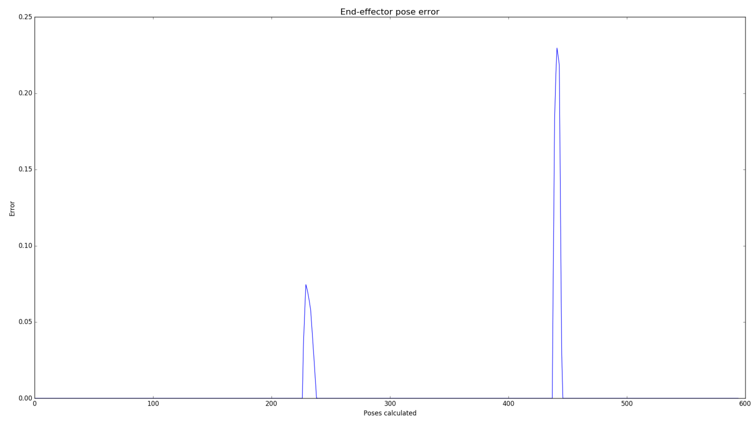


Figure 5: Error Curve

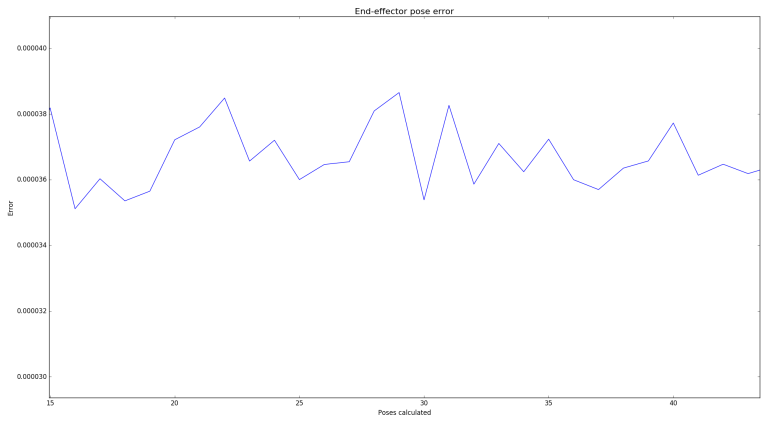


Figure 6: Zoom on flat zone of error curve

References

- John J. Craig, Introduction to Robotics: Mechanics and Control (3rd Edition) ISBN 978-0201543612