

Welding Simulation with Kuka LBR iiwa 14 R820

Varun Asthana

University of Maryland, College Park, USA

December, 2019

Contents

1	Introduction	3
2	Motivation	3
3	Kuka LBR iiwa 14 R820 Specification	3
3.1	Joint Configuration	4
3.2	Joint Constraints	4
3.3	Kinematic Lengths	4
4	Assumptions	5
5	Modeling	6
5.1	Forward Kinematics	6
5.2	Inverse Kinematics	8
5.2.1	Inverse Position	9
5.2.2	Inverse Orientation	10
6	Validation in MATLAB	11
7	Simulation in V-REP	12
7.1	Path Planning for Welding along the Edges	12
8	Conclusion	13

List of Figures

1	Basic data of Kuka LBR iiwa14 R820	3
2	Main assembly and robot axes	4
3	Joint Constraints	4
4	Kinematic Lengths	5
5	Positioning of assumed weld torch with respect to axis 7	6
6	Frame assignment (dimensions in mm)	6
7	Model deformation due to a change in orientation of 2 nd joint x-axis "x ₁ "	7
8	Joints of Kuka LBR iiwa 14 R820	8
9	Inverse position geometrical representation	9
10	Object position with respect to robot in world frame	11
11	Defining R_d^b for simulation	12
12	MATLAB output example	13
13	Multiple solutions for inverse kinematics	13
14	Path planning in V-Rep	14
15	Welding simulation in V-Rep	14

1 Introduction

With the improving technologies and the advent of automation, manufacturing industries are not far behind to catch-up with these changes. With a focus on manufacturing industries dealing with metallic parts, one core operation is fusing of different parts to form an assembly through a process of welding. Welding technology has existed since 1881 and has been deployed manually for a very long time. Over the centuries, the technology has undergone tremendous changes in terms of quality by improving upon various process parameters. While the focus of many researchers have been on such improvements, many others have also focused on quantitative aspect of the technology, i.e. to improve the output rate. One of the methodology is to automate the process of welding, instead of relying on human worker. Since the machines can do a repetitive task with same efficiency and consistency, welding automation is best suited on mass manufacturing production line.

This report presents a detailed study of the kinematics of Kuka LBR iiwa 14 R820 robotic arm. This robot is suited for small to medium sized components. The report spans over the topics of- robot specification and technical details, assumptions considered during the study of kinematics, mathematical model development for the forward kinematics and the inverse kinematics, validation of the derived models in MATLAB (2019b), and a welding simulation by path planning in V-Rep ProEDU (v3.6.2).

2 Motivation

Before joining the graduate program in robotics, I was employed with an automobile manufacturing firm in India, for 5 years. My work required me to set-up robotic production line for motorbikes. In my various work projects, I have utilized the robotic arms of Kuka, Panasonic, Abb and many others. It was astonishing to see these machines work continuously with the same performance throughout the day. This application of robotic arms is so powerful in terms of quality and quantity, that at times based on the available robotic arm and its workspace, RnD team had to modify the product design in order to avoid any manual welding operation.

Hence, previously being an end user of these various robotic arms for weld applications, this project offered me an opportunity to get first-hand experience with the kinematics of a of a robotic arm- Kuka LBR iiwa 14 R820.

3 Kuka LBR iiwa 14 R820 Specification

This section highlights some required technical and geometrical specifications of the robotic arm used in this project. These specification will then be used repeatedly thereafter in the report to demonstrate and validate the approach used to formulate a mathematical model to solve the forward and inverse kinematics. Some basic details of the robot is shown in fig(1). For the brevity of the space, from now we will refer the Kuka LBR iiwa 14 R820 with a shorthand of Kuka14.

	LBR iiwa 14 R820
Number of axes	7
Number of controlled axes	7
Volume of working envelope	1.8 m ³
Pose repeatability (ISO 9283)	± 0.15 mm
Weight	approx. 29.9 kg
Rated payload	14 kg
Maximum reach	820 mm

Figure 1: Basic data of Kuka LBR iiwa14 R820

3.1 Joint Configuration

Kuka14 is classified as a lightweight robot and is a jointed arm with 7 axis. As shown in fig(2) it has 7 revolute joints with no prismatic joints. The joint configuration can be described as RRRRRRR type with an in-line wrist.

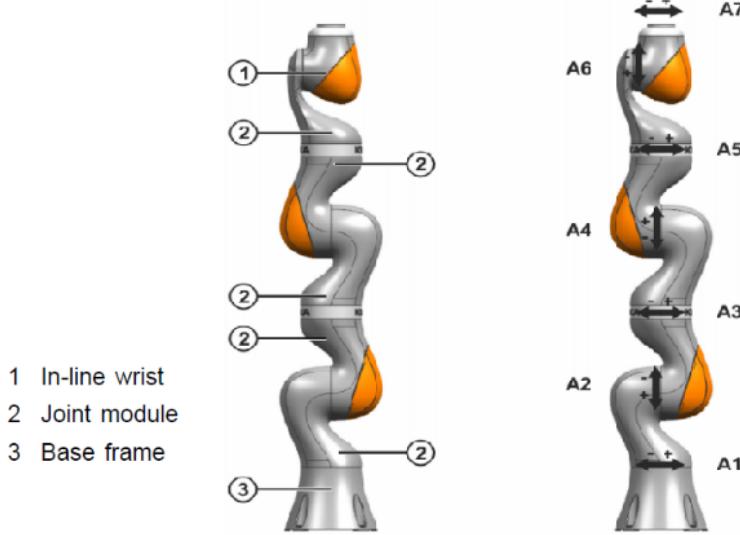


Figure 2: Main assembly and robot axes

Every axis contains multiple sensors that provide signals for robot control (e.g. position control and impedance control) and that are also used as a protective function for the robot. In this report we will only concentrate on kinematics to reach a desired end-effector position and orientation with an assumption that the robot operates in ideal condition and within functional constraints as per the official manual[1]. For a comprehensive list of all assumptions refer section (4).

3.2 Joint Constraints

The 7 revolute joints in the Kuka14 are not free to move 360° . There are angular constraints imposed on each joint by the manufacturer. Fig(3) shows the allowable range of each joint with respect to zeroth pose condition of the robot as shown in fig(2).

Motion range	
A1	$\pm 170^\circ$
A2	$\pm 120^\circ$
A3	$\pm 170^\circ$
A4	$\pm 120^\circ$
A5	$\pm 170^\circ$
A6	$\pm 120^\circ$
A7	$\pm 175^\circ$

Figure 3: Joint Constraints

3.3 Kinematic Lengths

The official manual of Kuka14 provides some dimensional details of the joint positions relative to the robot's base. It should be noted in fig(4) that not all kinematic lengths are provided. Height of every alternate joint from the base is only provided.

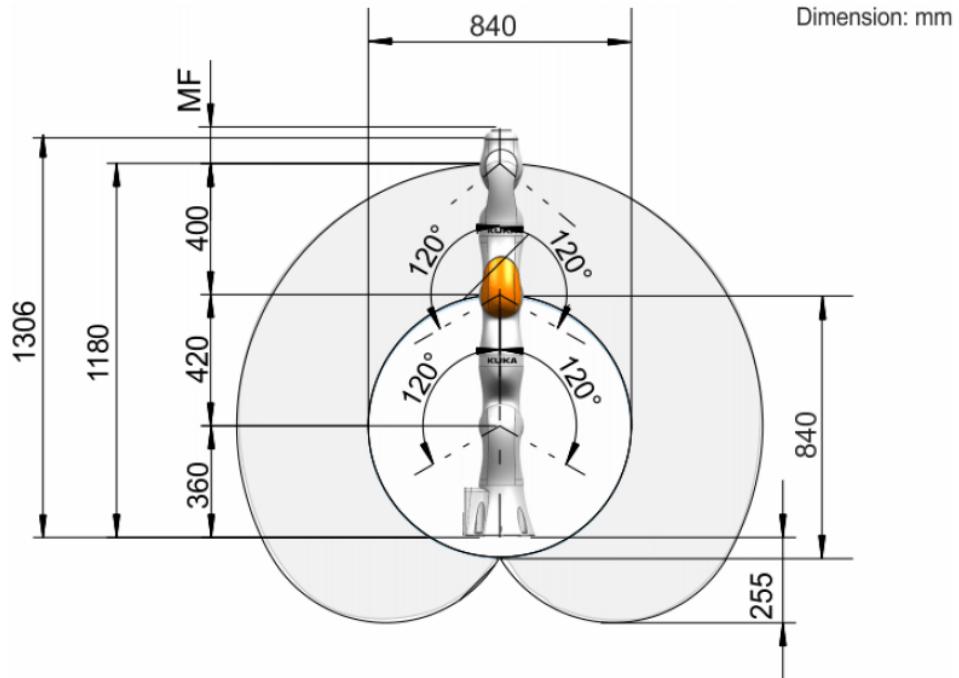


Figure 4: Kinematic Lengths

4 Assumptions

In order to formulate a mathematical model for forward kinematics and inverse kinematics of the KUKA14 certain assumptions have been considered under which the robot will operate. The list of assumptions is as below:

1. All bodies are assumed to be perfectly rigid
2. No slip between mating parts
3. No backlash in the gears of the motors
4. Required power, torque and weld wire feed is always available to the system
5. A finite length straight cylindrical weld torch is mounted at the last joint of the robot arm as shown in fig(5)
6. Robotic arm has a fixed base
7. No physical obstruction in the workspace of the robot
8. All the revolute joints are at zero degrees in zeroth condition. Robot pose shown in fig(4) is considered as the zeroth condition
9. For developing inverse kinematics model, 6-DOFs are only considered, instead of 7-DOFs, by locking the 3rd joint, to avoid infinite number of solutions
10. Robot's base is placed at the origin of the world frame and is oriented in such a way that the axis directions of the world frame and the base frame coincide with each other.
11. Desired position and orientation of the end-effector lies within the robot's workspace



Figure 5: Positioning of assumed weld torch with respect to axis 7

5 Modeling

In consideration of the previously highlighted assumptions and the robot's specifications, a mathematical model is worked out for the forward and the inverse kinematics of the Kuka14 robotic arm. Forward kinematics is solved for all 7-DOFs of the robotic arm, while the inverse kinematics is solved for 6-DOFs by locking the 3rd joint to avoid infinite solutions. Frame assignments are done by following the popular DenavitHartenberg (DH) convention.

5.1 Forward Kinematics

Forward kinematics is used to determine the end-effector's position and the orientation for a given set of joint parameters. For Kuka14 since all the 7 joints are revolute joints, let us denote the joint parameters by θ_i for $i \in [1,7]$, where θ_i is the angular position of the respective joint measured in degrees.

Forward kinematics is solved by formulating the transformation matrices (T_n^b) for the end-effector with respect to the world frame (base frame). As stated in the section (3.3), the kinematics lengths for all the joints are not available, but only for the alternate joints, hence joint frames are assigned and positioned in such a manner that they are at the known heights from the base of the robot. Fig(6) shows the frames assigned to the Kuka14 to formulate the DH table. An end-effector frame $x_n - z_n$ is also shown at a finite distance (d_n) from the frame $x_6 - z_6$. In this report, the frame notation of n and 7 are used interchangeably for the end-effector frame.

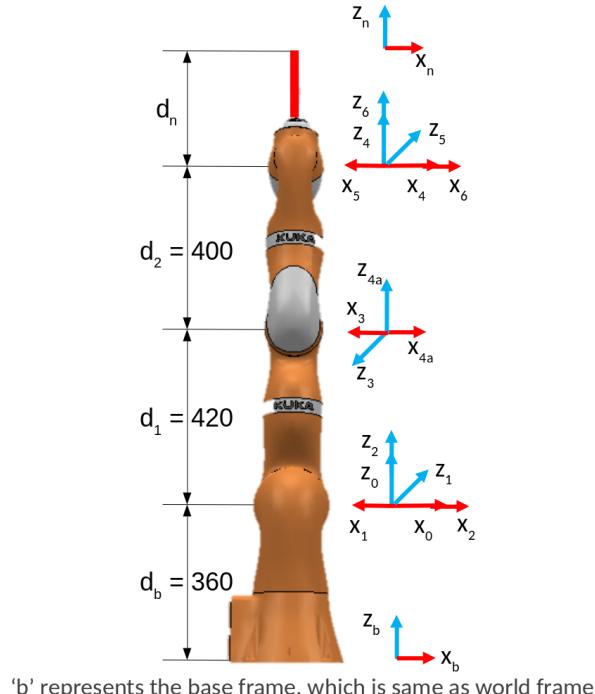


Figure 6: Frame assignment (dimensions in mm)

Important aspects of the frame assignment

1. It can be noticed that the positive x -axis of the frames toggles by 180° between left and right direction, which can complicate the DH table. Such frame assignment is used to maintain coherence with the local frames of each joint pre-defined in the V-Rep model of the Kuka14. An attempt to correct it to simplify the DH table was made, but it affected the geometry of the model, as shown in fig(7).

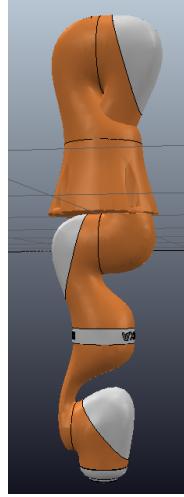


Figure 7: Model deformation due to a change in orientation of 2^{nd} joint x -axis "x₁"

2. An additional frame $x_{4a} - z_{4a}$ has been inserted at 4^{th} joint. It was required as without it the translation of distance d_2 was not feasible as per the DH conventions. Another approach could have been to place the $x_4 - z_4$ frame at the local origin of the frame $x_3 - z_3$, but this has not been done intentionally to simplify the inverse kinematics by introducing a wrist center.

By using the frame diagram as shown in fig(6), the below DH table is formulated with θ and α measured in degrees.

Frame	θ_{i-1}	d_{i-1}	a_i	α_i
b-0	0	d_b	0	0
0-1	$180^\circ + \theta_1$	0	0	90°
1-2	$180^\circ + \theta_2$	0	0	90°
2-3	$180^\circ + \theta_3$	d_1	0	-90°
3-4a	$180^\circ + \theta_4$	0	0	-90°
4a-4	0	d_2	0	0
4-5	$180^\circ + \theta_5$	0	0	90°
5-6	$180^\circ + \theta_6$	0	0	90°
6-n	θ_7	d_n	0	0

Table 1: DH Table

Using the above DH table, all A and T matrices are calculated from eq(1) and eq(2).

$$A_i = \begin{bmatrix} \cos(\theta_{i-1}) & -\sin(\theta_{i-1})\cos(\alpha_i) & \sin(\theta_{i-1})\sin(\alpha_i) & \cos(\theta_{i-1})a_i \\ \sin(\theta_{i-1}) & \cos(\theta_{i-1})\cos(\alpha_i) & -\cos(\theta_{i-1})\sin(\alpha_i) & \sin(\theta_{i-1})a_i \\ 0 & \sin(\alpha_i) & -\cos(\alpha_i) & d_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$T_i^b = A_0 \times A_1 \times A_2 \dots \times A_i \quad \text{where } i = 0, 1, 2, 3, 4, 5, 6, 7 \quad (2)$$

Expressions for the calculated transformation matrices are very large, and hence they are not included here.

5.2 Inverse Kinematics

Inverse kinematics is used to find the joint parameters for a given desired position and an orientation of the end-effector with respect to the world frame. After formulating the model for the forward kinematics, the next step is to solve the inverse kinematics of Kuka14 robotic arm. Fig(8) shows the joint layer of the Kuka14 model from V-Rep. It can be seen that the z-axis of top 3 revolute joints intersect at a common point. This point is referred to as *wrist center*.

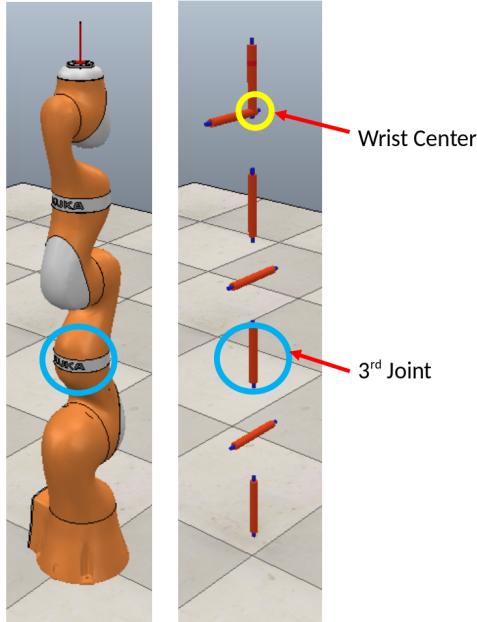


Figure 8: Joints of Kuka LBR iiwa 14 R820

The presence of the wrist center allows the inverse kinematics problem to be decoupled into 2 problems of-

1. Inverse Position depending on bottom 4 joints (joints 1 to 4), and
2. Inverse Orientation depending on top 3 joints (joints 5 to 7)

With respect to the base frame, let the desired position of the end-effector be denoted by O_d^b and the desired orientation by R_d^b . Also the distance of the end-effector frame from the frame $x_6 - z_6$ is known (d_n). Hence we can determine the position of the wrist center (O_c^b) by traversing a distance of d_n units in the direction of the negative z_n axis. This can be represented mathematically by eq(3).

$$O_c^b = O_d^b - (d_n \times R_d^b) \quad (3)$$

O_c^b can be decomposed into its components and represented as (x_c^b, y_c^b, z_c^b)

5.2.1 Inverse Position

In our context, inverse position refers to the problem of calculating the 4 joint parameters that will position the wrist center at O_c^b . Since 4 unknowns (4 joint parameters) are required to be solved with 3 known values i.e. x_c^b , y_c^b and z_c^b , hence this forms an under-constraint system with infinite solutions. Because of this reason it was stated previously that inverse kinematics will be solved by locking the 3rd joint (refer fig(8)).

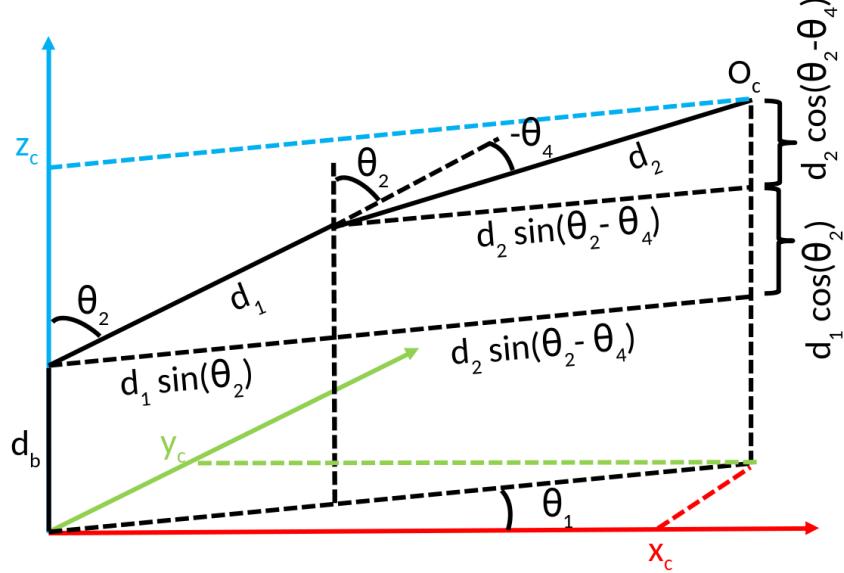


Figure 9: Inverse position geometrical representation

Inverse position for joint parameters θ_1 , θ_2 and θ_4 can be solved geometrically from the fig(9). We get,

$$x_c^b = (d_1 \sin(\theta_2) + d_2 \sin(\theta_2 - \theta_4)) \times \cos(\theta_1) \quad (4)$$

$$y_c^b = (d_1 \sin(\theta_2) + d_2 \sin(\theta_2 - \theta_4)) \times \sin(\theta_1) \quad (5)$$

$$z_c^b = d_b + d_1 \cos(\theta_2) + d_2 \cos(\theta_2 - \theta_4) \quad (6)$$

Let $z_{cc}^b = z_c^b - d_b$. We can write eq(6) as

$$z_{cc}^b = d_1 \cos(\theta_2) + d_2 \cos(\theta_2 - \theta_4) \quad (7)$$

Solving for θ_1

By dividing eq(5) with eq(4), we get-

$$\tan(\theta_1) = \frac{y_c^b}{x_c^b} \quad \text{or} \quad \theta_1 = \text{atan2}\left(\frac{y_c^b}{x_c^b}\right) \quad (8)$$

Solving for θ_4

By squaring and adding eq(4) and eq(5), we get

$$x_c^{b^2} + y_c^{b^2} = (d_1 \sin(\theta_2) + d_2 \sin(\theta_2 - \theta_4))^2 \quad (9)$$

By squaring and adding eq(7) and eq(9), we get

$$x_c^{b^2} + y_c^{b^2} + z_{cc}^{b^2} = d_1^2 + d_2^2 + 2d_1 d_2 \cos(\theta_4) \quad (10)$$

$$\cos(\theta_4) = \frac{x_c^{b^2} + y_c^{b^2} + z_{cc}^{b^2} - d_1^2 - d_2^2}{2d_1 d_2} \quad \text{or} \quad \theta_4 = \text{acos}\left(\frac{x_c^{b^2} + y_c^{b^2} + z_{cc}^{b^2} - d_1^2 - d_2^2}{2d_1 d_2}\right) \quad (11)$$

Solving for θ_2

By expanding the term $\cos(\theta_2 - \theta_4)$, we can rewrite the eq(7) as

$$\begin{aligned}\Rightarrow z_{cc}^b &= d_1 \cos(\theta_2) + d_2 \cos(\theta_2) \cos(\theta_4) - d_2 \sin(\theta_2) \sin(\theta_4) \\ \Rightarrow z_{cc}^b - d_1 \cos(\theta_2) - d_2 \cos(\theta_2) \cos(\theta_4) &= -d_2 \sin(\theta_2) \sin(\theta_4)\end{aligned}$$

Squaring on both sides,

$$\begin{aligned}\Rightarrow (z_{cc}^b - d_1 \cos(\theta_2) - d_2 \cos(\theta_2) \cos(\theta_4))^2 &= (-d_2 \sin(\theta_2) \sin(\theta_4))^2 \\ \Rightarrow (z_{cc}^b - d_1 \cos(\theta_2) - d_2 \cos(\theta_2) \cos(\theta_4))^2 &= d_2^2 (1 - \cos^2(\theta_2)) (1 - \cos^2(\theta_4)) \\ \Rightarrow \cos^2(\theta_2) (d_1^2 + d_2^2 + 2d_1 d_2 \cos(\theta_4)) - 2z_{cc}^b (d_1 + d_2 \cos(\theta_4)) \cos(\theta_2) + [z_{cc}^b]^2 - d_2^2 (1 - \cos^2(\theta_4)) &= 0\end{aligned}$$

Using eq(10), we get

$$\Rightarrow \cos^2(\theta_2) (x_c^b)^2 + (y_c^b)^2 + (z_{cc}^b)^2 - \frac{z_{cc}^b}{d_1} (x_c^b)^2 + (y_c^b)^2 + (z_{cc}^b)^2 + d_1^2 - d_2^2 \cos(\theta_2) + [z_{cc}^b]^2 - d_2^2 (1 - \cos^2(\theta_4)) = 0 \quad (12)$$

The above equation is a quadratic equation in $\cos(\theta_2)$ of the form $Ax^2 + Bx + C = 0$. It can be solved by using the quadratic formula to find the solutions for $\cos(\theta_2)$.

Note : Eq(8) gives a unique solution for θ_1 , while the eq(11) and the eq(12) are in \cos^{-1} and hence have multiple solutions. These solutions occur in pairs of θ_2 and θ_4 .

5.2.2 Inverse Orientation

For a given position and an orientation of the end-effector, we have solved the inverse position problem. Therefore, θ_1 , θ_2 and θ_4 are now known parameters (also we have locked the 3rd joint, hence $\theta_3 = 0$). Now we need to solve for θ_5 , θ_6 and θ_7 . These are calculated by comparing the orientation matrix for the end-effector obtained from the mathematical model of forward kinematics, with the already known orientation matrix R_d^b .

$$R_d^b = R_7^b \quad (13)$$

Transformation matrix is a combination of the rotation matrix and the translation matrix between 2 frames into considerations. It is of the form,

$$T_i^j = \begin{bmatrix} R_i^j & O_i^j \\ \vec{0} & 1 \end{bmatrix} \quad \text{where } T_i^j = A_j \times A_{j+1} \times A_{j+2} \dots A_i \quad \text{or} \quad T_i^j = A_j \times A_{j-1} \times A_{j-2} \dots A_i \quad (14)$$

Hence, by extracting the rotation matrices from the respective transformation matrices obtained by solving the forward kinematics, the orientation of the end-effector with respect to base frame can be represented as

$$\Rightarrow R_7^b = R_4^b R_7^4$$

using eq(13), we can write

$$\Rightarrow R_d^b = R_4^b R_7^4$$

Pre-multiplying by R_4^b ⁻¹, we get

$$\Rightarrow R_4^b R_d^b = R_7^4$$

For a rotation matrix, $R^{-1} = R^T$, hence we can write

$$\Rightarrow R_4^b T R_d^b = R_7^4 \quad (15)$$

Let us denote the L.H.S of eq(15) as R_{wrist} , and can be computed with known parameters. Also, R_7^4 can be expressed in terms of θ_5 , θ_6 and θ_7 by using eq(23).

$$R_7^4 = \begin{bmatrix} -\sin(\theta_5)\sin(\theta_7) + \cos(\theta_5)\cos(\theta_6)\cos(\theta_7) & -\sin(\theta_5)\cos(\theta_7) - \cos(\theta_5)\cos(\theta_6)\sin(\theta_7) & \cos(\theta_5)\sin(\theta_6) \\ \cos(\theta_6)\sin(\theta_5)\cos(\theta_7) + \cos(\theta_5)\sin(\theta_7) & \cos(\theta_5)\cos(\theta_7) - \cos(\theta_6)\sin(\theta_5)\sin(\theta_7) & \sin(\theta_5)\sin(\theta_6) \\ -\sin(\theta_6)\cos(\theta_7) & \sin(\theta_6)\sin(\theta_7) & \cos(\theta_6) \end{bmatrix} \quad (16)$$

By equating the eq(16) to R_{wrist} , we can write the equations-

$$\tan(\theta_5) = \frac{(R_{wrist})_{23}}{(R_{wrist})_{13}} \quad \text{or} \quad \theta_5 = \text{atan2}\left(\frac{(R_{wrist})_{23}}{(R_{wrist})_{13}}\right) \quad (17)$$

$$\cos(\theta_6) = (R_{wrist})_{33} \quad \text{or} \quad \theta_6 = \text{acos}((R_{wrist})_{33}) \quad (18)$$

$$\tan(\theta_7) = \frac{(R_{wrist})_{32}}{-(R_{wrist})_{31}} \quad \text{or} \quad \theta_7 = \text{atan2}\left(\frac{(R_{wrist})_{32}}{-(R_{wrist})_{31}}\right) \quad (19)$$

Note :

Eq(18) gives two solutions for θ_6 , one positive and one negative. Based on this, the value of $\sin(\theta_6)$ will be either positive or negative respectively. Thus this affects the R_7^4 matrix. Hence for

$$\theta_6 = -\text{acos}((R_{wrist})_{33}) \quad (20)$$

another set of solutions exist for θ_5 and θ_7 . They are given by-

$$\tan(\theta_5) = \frac{-(R_{wrist})_{23}}{-(R_{wrist})_{13}} \quad \text{or} \quad \theta_5 = \text{atan2}\left(\frac{-(R_{wrist})_{23}}{-(R_{wrist})_{13}}\right) \quad (21)$$

$$\tan(\theta_7) = \frac{-(R_{wrist})_{32}}{(R_{wrist})_{31}} \quad \text{or} \quad \theta_7 = \text{atan2}\left(\frac{-(R_{wrist})_{32}}{(R_{wrist})_{31}}\right) \quad (22)$$

Note :

All the possible pairs of solutions obtained by solving the equations for $\theta_1 - \theta_7$ are not practically feasible due to the joint constraints stated in section (3.2).

6 Validation in MATLAB

To validate the mathematical model for joint parameters obtained by solving the inverse kinematics, a MATLAB script for the equations eq(4)- eq(22) was written.

To validate the model, end-effector parameters as well as the parameters for the object to be welded were to be defined. To do so the below considerations were made-

1. Kuka14 is at the origin of the world frame
2. A cuboid shaped object (100mm x 600mm x 20mm) is to be welded and is placed parallel to the ground (within the robot's workspace)
3. Center of the cuboid lies at (0.55, 0, 0.5) in the world frame (as shown in fig(10)). Unit of measurement in V-Rep is meters.

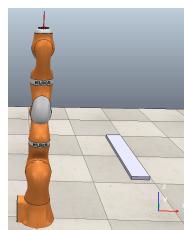


Figure 10: Object position with respect to robot in world frame

4. Welding to be done at the perimeter of the top surface of the object
5. Welding is to be done at 45° from the negative x-axis of the world frame, and will be directed from top to bottom (as shown in fig(11)) End-effector parameters were defined,

$$R_d^b = R_x(180^\circ)R_y(45^\circ)$$

$$R_d^b = \begin{bmatrix} 0.7071 & 0 & 0.7071 \\ 0 & -1 & 0 \\ 0.7071 & 0 & -0.7071 \end{bmatrix} \quad (23)$$

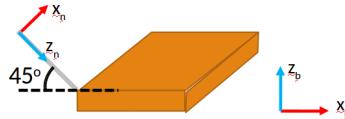


Figure 11: Defining R_d^b for simulation

6. End-effector frame distance from wrist center is taken as $d_n = 200$ mm

With these consideration, the MATLAB script is run with various different values of x_d , y_d and z_d (on the vertices of the top surface of the object). The script performs the below tasks-

1. Calculates all possible solutions for θ_1 - θ_4 by using the mathematical model for inverse position
2. Validate the correct combination of the pairs for inverse position parameters by using the mathematical model for forward kinematics (to get the same set of x_d , y_d and z_d)
3. All correct combinations for θ_1 - θ_4 are checked against joint constraints
4. Calculates all possible solution for θ_5 - θ_7 by using the mathematical model for inverse orientation. Pair combinations for inverse orientation is not checked as it has been pre-defined in the mathematical model.
5. All correct combinations for θ_5 - θ_7 are checked against joint constraints
6. Filtered set of solutions are displayed as output

MATLAB script is provided separately as an external attachment to this report. A snippet of the output of the script is shown in fig(12).

7 Simulation in V-REP

Results obtained from MATLAB were simulated in V-Rep to provide a visual verification methodology. The joint parameters calculated in MATLAB were used as input for the V-Rep script to verify if the end-effector is reaching the desired position in the desired orientation or not.

Fig(13) shows the snippets of the verification of multiple results obtained against a single input.

7.1 Path Planning for Welding along the Edges

Path planning is done by defining multiple number of O_d^b along the edges of the top surface of the object. Path is traversed in a anti-clock wise direction. The points used on the objects includes all the vertices of the cuboid and 3 additional points along the length of the object, as shown in fig(14). From the multiple solutions obtained from the MATLAB script, an optimum solution for each O_d^b was selected manually to avoid unnecessary angular motions of the joints. Fig(15) shows the trajectory followed by the robot. A V-Rep scene file is provided separately with the above discussed configurations that will simulate the welding operation.

```

NOTE:
Inverse Kinematics have been solved by locking the 3rd Joint. As the end effector is symmetric
Hence any value for theta_7 will satisfy the problem, but is still calculated by utilizing
Inverse Kinematics concepts and producing 2 values for theta_7

"x=Equal 1-k sol m" means the Case x has been satisfied for the axis constraints
from axis-1 till axis-k.
m=1 means t6 has been considered same as calculated value of acos and m=2 means
t6 is considered as -acos

output_order =
    'theta_1  theta_2  theta_3  theta_4  theta_5  theta_6  theta_7'

2_Equal 1-4
2_Equal 1-7 sol1

theta =
146.7484 -22.3713      0   81.6125 -144.4373  41.7843  127.1192

2_Equal 1-7 sol2

theta =
146.7484 -22.3713      0   81.6125   35.4902 -41.7843 -52.8084

8_Equal 1-4
8_Equal 1-7 sol1

theta =
146.7484 -101.5707      0  -81.6125 -154.0493 117.5044 167.7604

8_Equal 1-7 sol2

theta =
146.7484 -101.5707      0  -81.6125   25.8783 -117.5044 -12.1672

```

Figure 12: MATLAB output example

8 Conclusion

Study of the kinematics of the Kuka LBR iiwa 14 R820 robotic arm was successfully completed. The mathematical models derived for the forward kinematics and the inverse kinematics, under pre-defined assumptions, were verified in two different soft-tools, namely MATLAB (2019b) and V-Rep ProEdu (v 3.6.2).

The MATLAB script generated was made versatile in nature to solve inverse kinematics and provide all feasible set of solutions for the joint parameters by taking all the joint constraints into consideration. These set of solutions were duly verified, within the script, by injecting the same parameters in the mathematical model of the forward kinematics. A V-Rep script was also generated to verify the models visually in an simulated environment.

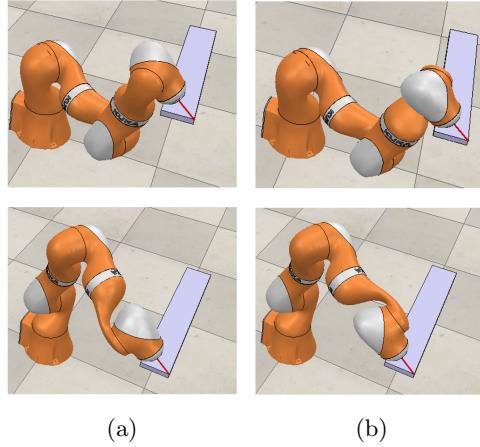


Figure 13: Multiple solutions for inverse kinematics

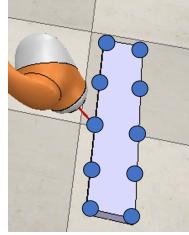


Figure 14: Path planning in V-Rep

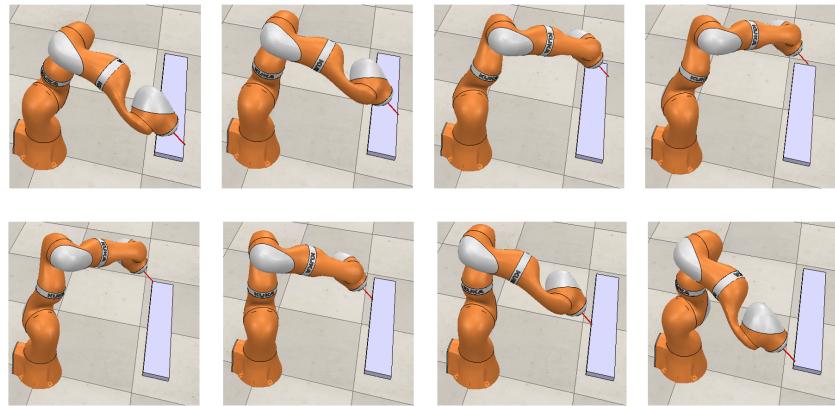


Figure 15: Welding simulation in V-Rep

Welding simulation was completed by doing a path planning around the perimeter of a cuboid shaped object. Path planning was achieved by defining multiple points along the edges of the object.

References

- [1] Kuka official manual for LBR iiwa14 R820. <https://www.kuka.com/en-us/products/robotics-systems/industrial-robots/lbr-iiwa7>
- [2] Mark W. Spong. S Hutchinson and M. Vidyasagar. Robot Modeling and Control, 2006.