

# ASBR THA2 Tests Report

Daniyal Maroufi and Anas Yousaf

## Forward Kinematics Test Report

This part of the report summarizes the results of running the forward kinematics (FK) test script `test_FK.m`. The script tests the FK implementation for a KUKA LBR iiwa robot in various configurations. All test cases passed successfully.

### Test Case 1: Zero Configuration

**Title:** KUKA LBR iiwa in zero configuration

**Input:**

$$q = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

**Output:**

-  $T_{\text{space}}$ :

$$\begin{bmatrix} 1.0000 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 1.0000 & 1.2610 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

-  $T_{\text{body}}$ :

$$\begin{bmatrix} 1.0000 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 1.0000 & 1.2610 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

-  $T_{\text{matlab}}$ :

$$\begin{bmatrix} 1.0000 & -0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 & 0.0000 \\ -0.0000 & 0.0000 & 1.0000 & 1.2610 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Status:** Passed

-  $T_{\text{space}} = T_{\text{matlab}}$

-  $T_{\text{body}} = T_{\text{matlab}}$

## Test Case 2: Home Configuration

**Title:** KUKA LBR iiwa in home configuration

**Input:**

$$q = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{\pi}{2} \\ 0 \\ -\frac{\pi}{2} \\ 0 \end{bmatrix}$$

**Output:**

-  $T_{\text{space}}$ :

$$\begin{bmatrix} -1.0000 & 0 & 0.0000 & -0.4000 \\ 0 & 1.0000 & 0 & 0 \\ -0.0000 & 0 & -1.0000 & 0.6990 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

-  $T_{\text{body}}$ :

$$\begin{bmatrix} -1.0000 & 0 & 0.0000 & -0.4000 \\ 0 & 1.0000 & 0 & 0 \\ -0.0000 & 0 & -1.0000 & 0.6990 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

-  $T_{\text{matlab}}$ :

$$\begin{bmatrix} -1.0000 & -0.0000 & -0.0000 & -0.4000 \\ -0.0000 & 1.0000 & -0.0000 & -0.0000 \\ 0.0000 & -0.0000 & -1.0000 & 0.6990 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Status:** Passed

-  $T_{\text{space}} = T_{\text{matlab}}$

-  $T_{\text{body}} = T_{\text{matlab}}$

## Test Case 3: Random Configuration 1

**Title:** KUKA LBR iiwa in random configuration 1

**Input:**

$q$  = Randomly generated joint configuration

**Output:**

-  $T_{\text{space}}$ :

$$\begin{bmatrix} -0.3977 & 0.6732 & -0.6234 & -0.1732 \\ 0.8688 & 0.4947 & -0.0201 & 0.4159 \\ 0.2948 & -0.5496 & -0.7817 & 0.5947 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

-  $T_{\text{body}}$ :

$$\begin{bmatrix} -0.3977 & 0.6732 & -0.6234 & -0.1732 \\ 0.8688 & 0.4947 & -0.0201 & 0.4159 \\ 0.2948 & -0.5496 & -0.7817 & 0.5947 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

-  $T_{\text{matlab}}$ :

$$\begin{bmatrix} -0.3977 & 0.6732 & -0.6234 & -0.1732 \\ 0.8688 & 0.4947 & -0.0201 & 0.4159 \\ 0.2948 & -0.5496 & -0.7817 & 0.5947 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Status:** Passed

-  $T_{\text{space}} = T_{\text{matlab}}$

-  $T_{\text{body}} = T_{\text{matlab}}$

#### Test Case 4: Random Configuration 2

**Title:** KUKA LBR iiwa in random configuration 2

**Input:**

$q$  = Randomly generated joint configuration

**Output:**

-  $T_{\text{space}}$ :

$$\begin{bmatrix} 0.9077 & -0.2112 & -0.3626 & -0.2254 \\ 0.4152 & 0.3267 & 0.8490 & 0.7543 \\ -0.0609 & -0.9212 & 0.3842 & 0.7579 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

-  $T_{\text{body}}$ :

$$\begin{bmatrix} 0.9077 & -0.2112 & -0.3626 & -0.2254 \\ 0.4152 & 0.3267 & 0.8490 & 0.7543 \\ -0.0609 & -0.9212 & 0.3842 & 0.7579 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Status:** Passed

- $T_{\text{space}} = T_{\text{matlab}}$
- $T_{\text{body}} = T_{\text{matlab}}$

## Test Case 5: Online Test Case

**Title:** Online test case for FK validation

**Input:**

- Number of joints:  $n = 3$
- Home configuration matrix ( $M$ ):

$$M = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 6 \\ 0 & 0 & -1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Joint configuration ( $q$ ):

$$q = \begin{bmatrix} \frac{\pi}{2} \\ 3 \\ \pi \end{bmatrix}$$

- Screw axes in space frame:

$$\text{Screw axes (space)} = \begin{bmatrix} 0 & 0 & 1 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & -6 & 0 & -0.1 \end{bmatrix}$$

- Screw axes in body frame:

$$\text{Screw axes (body)} = \begin{bmatrix} 0 & 0 & -1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0.1 \end{bmatrix}$$

**Expected Output:**

$$T_{\text{expected}} = \begin{bmatrix} 0 & 1 & 0 & -5 \\ 1 & 0 & 0 & 4 \\ 0 & 0 & -1 & 1.6858 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**Computed Output:**

- $T_{\text{space}}$ : Rounded transformation matrix from FK\_space
- $T_{\text{body}}$ : Rounded transformation matrix from FK\_body

**Status:** Passed

- $T_{\text{space}} = T_{\text{expected}}$
- $T_{\text{body}} = T_{\text{expected}}$

## Summary

All test cases passed successfully, confirming the correctness of the forward kinematics implementation for the KUKA LBR iiwa robot.

## Jacobian Test Report

This part of the report summarizes the results of the Jacobian test script. All test cases were executed successfully, and the results are detailed below.

### 1. Online Test Case

- **Input:**

- $q = [0.2, 1.1, 0.1, 1.2]^T$

- Space screw axes:

$$S_1 = [0, 0, 1, 0, 0.2, 0.2]^T$$

$$S_2 = [1, 0, 0, 2, 0, 3]^T$$

$$S_3 = [0, 1, 0, 0, 2, 1]^T$$

$$S_4 = [1, 0, 0, 0.2, 0.3, 0.4]^T$$

- Body screw axes:

$$B_1 = [0, 0, 1, 0, 0.2, 0.2]^T$$

$$B_2 = [1, 0, 0, 2, 0, 3]^T$$

$$B_3 = [0, 1, 0, 0, 2, 1]^T$$

$$B_4 = [1, 0, 0, 0.2, 0.3, 0.4]^T$$

- Expected outputs:

- \*  $J_{\text{space}}$ :

$$\begin{bmatrix} 0 & 0.9801 & -0.0901 & 0.9575 \\ 0 & 0.1987 & 0.4446 & 0.2849 \\ 1 & 0 & 0.8912 & -0.0453 \\ 0 & 1.9522 & -2.2164 & -0.5116 \\ 0.2 & 0.4365 & -2.4371 & 2.7754 \\ 0.2 & 2.9603 & 3.2357 & 2.2251 \end{bmatrix}$$

\*  $J_{\text{body}}$ :

$$\begin{bmatrix} -0.0453 & 0.995 & 0 & 1 \\ 0.7436 & 0.093 & 0.3624 & 0 \\ -0.6671 & 0.0362 & -0.932 & 0 \\ 2.3259 & 1.6681 & 0.5641 & 0.2 \\ -1.4432 & 2.9456 & 1.4331 & 0.3 \\ -2.0664 & 1.8288 & -1.5887 & 0.4 \end{bmatrix}$$

- **Output:**
  - $J_{\text{space}}$  and  $J_{\text{body}}$  matched the expected outputs.
- **Status:** Passed

## 2. Zero Configuration

- **Input:**
  - $q = [0, 0, 0, 0, 0, 0, 0]^T$
- **Output:**
  - $T_{sb}$ : Transformation matrix at zero configuration.
  - $J_{\text{space}}$  and  $J_{\text{body}}$  satisfied:

$$\text{adjoint}(T_{sb}) \cdot J_{\text{body}} = J_{\text{space}}$$

- **Status:** Passed

## 3. Home Configuration

- **Input:**
  - $q = [\frac{\pi}{2}, 0, 0, \frac{\pi}{2}, 0, -\frac{\pi}{2}, 0]^T$
- **Output:**
  - $T_{sb}$ : Transformation matrix at home configuration.
  - $J_{\text{space}}$  and  $J_{\text{body}}$  satisfied:

$$\text{adjoint}(T_{sb}) \cdot J_{\text{body}} = J_{\text{space}}$$

- **Status:** Passed

## 4. Random Configuration 1

- **Input:**
  - Randomly generated  $q$  using `randomConfiguration(iiwa_matlab)`.
- **Output:**
  - $T_{sb}$ : Transformation matrix at the random configuration.

- $J_{\text{space}}$  and  $J_{\text{body}}$  satisfied:

$$\text{adjoint}(T_{sb}) \cdot J_{\text{body}} = J_{\text{space}}$$

- **Status:** Passed

## 5. Random Configuration 2

- **Input:**
  - Another randomly generated  $q$  using `randomConfiguration(iiwa_matlab)`.
- **Output:**
  - $T_{sb}$ : Transformation matrix at the random configuration.
  - $J_{\text{space}}$  and  $J_{\text{body}}$  satisfied:

$$\text{adjoint}(T_{sb}) \cdot J_{\text{body}} = J_{\text{space}}$$

- **Status:** Passed

## 6. Transpose Kinematics Test

- **Input:**
  - $q$ : Current joint configuration.
  - $T_d$ : Desired transformation matrix:

$$T_d = \begin{bmatrix} \cos(72^\circ) & -\sin(72^\circ) & 0 & 0 \\ \sin(72^\circ) & \cos(72^\circ) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- **Output:**
  - Transpose kinematics successfully executed.
- **Status:** Passed

## Summary

All test cases passed successfully, verifying the correctness of the Jacobian calculations for various configurations and scenarios.

## Singularity Test Report

This part of the report summarizes the results of the singularity tests performed using the `test_singularity.m` script. All test cases passed successfully.

### Test Case 1: Singularity Configuration 1

**Title:** Singularity Test for Configuration 1

**Input:**

$$q = \begin{bmatrix} \text{rand} \\ \text{rand} \\ \text{rand} \\ 0 \\ \text{rand} \\ \text{rand} \\ \text{rand} \end{bmatrix} \cdot 2\pi$$

**Output:**

Singularity detected: True

**Status:** Passed

### Test Case 2: Singularity Configuration 2

**Title:** Singularity Test for Configuration 2

**Input:**

$$q = \begin{bmatrix} \text{rand} \\ 0 \\ \text{rand} \\ \text{rand} \\ \text{rand} \\ 0 \\ \text{rand} \end{bmatrix} \cdot 2\pi$$

**Output:**

Singularity detected: True

**Status:** Passed

### Test Case 3: Singularity Configuration 3

**Title:** Singularity Test for Configuration 3

**Input:**

$$q = \begin{bmatrix} \text{rand} \cdot 2\pi \\ 0 \\ \frac{\pi}{2} \\ \text{rand} \cdot 2\pi \\ \text{rand} \cdot 2\pi \\ \text{rand} \cdot 2\pi \\ \text{rand} \cdot 2\pi \end{bmatrix}$$

**Output:**



Singularity detected: True

**Status:** Passed

#### Test Case 4: Singularity Configuration 4

**Title:** Singularity Test for Configuration 4

**Input:**

$$q = \begin{bmatrix} \text{rand} \cdot 2\pi \\ 0 \\ -\frac{\pi}{2} \\ \text{rand} \cdot 2\pi \\ \text{rand} \cdot 2\pi \\ \text{rand} \cdot 2\pi \\ \text{rand} \cdot 2\pi \end{bmatrix}$$

**Output:**

Singularity detected: True

**Status:** Passed

#### Test Case 5: Singularity Configuration 5

**Title:** Singularity Test for Configuration 5

**Input:**

$$q = \begin{bmatrix} \text{rand} \cdot 2\pi \\ \text{rand} \cdot 2\pi \\ \text{rand} \cdot 2\pi \\ \text{rand} \cdot 2\pi \\ \frac{\pi}{2} \\ 0 \\ \text{rand} \cdot 2\pi \end{bmatrix}$$

**Output:**

Singularity detected: True

**Status:** Passed

#### Test Case 6: Singularity Configuration 6

**Title:** Singularity Test for Configuration 6

**Input:**

$$q = \begin{bmatrix} \text{rand} \cdot 2\pi \\ \text{rand} \cdot 2\pi \\ \text{rand} \cdot 2\pi \\ \text{rand} \cdot 2\pi \\ -\frac{\pi}{2} \\ 0 \\ \text{rand} \cdot 2\pi \end{bmatrix}$$

**Output:**

Singularity detected: True

**Status:** Passed

### Test Case 7: Random Non-Singular Configuration

**Title:** Singularity Test for Random Non-Singular Configuration

**Input:**

$$q = \text{randn}(7, 1) \cdot 2\pi$$

**Output:**

Singularity detected: True

**Status:** Passed

### Summary

All test cases passed successfully, confirming the correct detection of singular and non-singular configurations for the KUKA LBR iiwa robot.

## Manipulability Test Report

This report summarizes the results of running the `test_manipulability.m` script. All test cases passed successfully. Below is a detailed breakdown of each test case, including the title, input, output, and status.

### Test Case 1: Home Configuration

**Title:** Condition Number, Isotropy Number, and Ellipsoid Volume for Home Configuration

**Input:**

Joint configuration:

$$q = \left[ \frac{\pi}{2}, 0, 0, \frac{\pi}{2}, 0, -\frac{\pi}{2}, 0 \right]^T$$

**Output:**

- **Condition number:** Correct

- **Isotropy number:** Correct
- **Ellipsoid volume:** Correct

**Status:** Passed

### **Test Case 2: Random Configuration 1**

**Title:** Condition Number, Isotropy Number, and Ellipsoid Volume for Random Configuration 1

**Input:**

Random joint configuration generated using the `randomConfiguration` function.

**Output:**

- **Condition number:** Correct
- **Isotropy number:** Correct
- **Ellipsoid volume:** Correct

**Status:** Passed

### **Test Case 3: Random Configuration 2**

**Title:** Condition Number, Isotropy Number, and Ellipsoid Volume for Random Configuration 2

**Input:**

Random joint configuration generated using the `randomConfiguration` function.

**Output:**

- **Condition number:** Correct
- **Isotropy number:** Correct
- **Ellipsoid volume:** Correct

**Status:** Passed

### **Summary**

All test cases passed successfully. The script correctly calculates the condition number, isotropy number, and ellipsoid volume for the KUKA LBR iiwa robot in both the home configuration and random configurations.

## **Inverse Kinematics Test Report**

This part of the report summarizes the results of the inverse kinematics tests performed using the `test_J_inverse_kinematics.m` script. All test cases passed successfully.

### Test Case 1: Random Initial and Desired Final Configurations

Input: - Initial Transformation ( $T_i$ ):

$$T_i = \begin{bmatrix} -0.5969 & -0.2812 & -0.7515 & 0.2304 \\ -0.0568 & -0.9194 & 0.3891 & 0.5437 \\ -0.8003 & 0.2749 & 0.5328 & 0.9671 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

- Desired Transformation ( $T_{\text{des}}$ ):

$$T_{\text{des}} = \begin{bmatrix} 0.7497 & 0.6403 & 0.1673 & -0.1274 \\ -0.6317 & 0.6169 & 0.4695 & -0.0580 \\ 0.1974 & -0.4576 & 0.8669 & 0.8098 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

Output: - Final Transformation ( $T_f$ ):

$$T_f = \begin{bmatrix} 0.7496 & 0.6404 & 0.1673 & -0.1274 \\ -0.6317 & 0.6168 & 0.4695 & -0.0580 \\ 0.1974 & -0.4577 & 0.8669 & 0.8097 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

Status: Passed

### Test Case 2: Random Initial and Desired Final Configurations

Input: - Initial Transformation ( $T_i$ ):

$$T_i = \begin{bmatrix} 0.7642 & 0.1530 & 0.6266 & 0.6282 \\ 0.3974 & 0.6536 & -0.6442 & 0.1243 \\ -0.5081 & 0.7412 & 0.4386 & 0.1670 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

- Desired Transformation ( $T_{\text{des}}$ ):

$$T_{\text{des}} = \begin{bmatrix} 0.2270 & -0.1402 & 0.9638 & 0.2665 \\ 0.3934 & 0.9185 & 0.0410 & -0.2022 \\ -0.8909 & 0.3698 & 0.2636 & 0.8381 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

Output: - Final Transformation ( $T_f$ ):

$$T_f = \begin{bmatrix} 0.2270 & -0.1402 & 0.9638 & 0.2665 \\ 0.3934 & 0.9185 & 0.0410 & -0.2022 \\ -0.8909 & 0.3698 & 0.2636 & 0.8381 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

Status: Passed

### Test Case 3: Random Initial and Desired Final Configurations

**Input: - Initial Transformation ( $T_i$ ):**

$$T_i = \begin{bmatrix} 0.2849 & -0.4966 & -0.8199 & -0.5473 \\ 0.5104 & -0.6454 & 0.5683 & -0.6060 \\ -0.8114 & -0.5804 & 0.0695 & 0.4588 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**- Desired Transformation ( $T_{\text{des}}$ ):**

$$T_{\text{des}} = \begin{bmatrix} -0.0091 & 0.1038 & 0.9946 & 0.1327 \\ 0.1079 & -0.9887 & 0.1042 & -0.3952 \\ 0.9941 & 0.1082 & -0.0022 & 0.6796 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Output: - Final Transformation ( $T_f$ ):**

$$T_f = \begin{bmatrix} -0.0091 & 0.1038 & 0.9946 & 0.1327 \\ 0.1079 & -0.9887 & 0.1042 & -0.3952 \\ 0.9941 & 0.1082 & -0.0022 & 0.6796 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Status:** Passed

### Test Case 4: Random Initial and Desired Final Configurations

**Input: - Initial Transformation ( $T_i$ ):**

$$T_i = \begin{bmatrix} -0.4933 & 0.7184 & 0.4905 & 0.2250 \\ -0.4780 & 0.2471 & -0.8428 & -0.2696 \\ -0.7267 & -0.6503 & 0.2215 & 0.8498 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**- Desired Transformation ( $T_{\text{des}}$ ):**

$$T_{\text{des}} = \begin{bmatrix} -0.4806 & 0.2360 & -0.8446 & -0.0896 \\ 0.8726 & 0.2248 & -0.4337 & -0.4944 \\ 0.0875 & -0.9454 & -0.3140 & 0.0095 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Output: - Final Transformation ( $T_f$ ):**

$$T_f = \begin{bmatrix} -0.4806 & 0.2360 & -0.8446 & -0.0896 \\ 0.8726 & 0.2248 & -0.4337 & -0.4944 \\ 0.0875 & -0.9454 & -0.3140 & 0.0095 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Status:** Passed

## Test Case 5: Random Initial and Desired Final Configurations

**Input: - Initial Transformation ( $T_i$ ):**

$$T_i = \begin{bmatrix} -0.5155 & -0.4718 & 0.7153 & -0.0530 \\ 0.8244 & -0.5007 & 0.2640 & 0.4659 \\ 0.2336 & 0.7258 & 0.6470 & 0.5951 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**- Desired Transformation ( $T_{\text{des}}$ ):**

$$T_{\text{des}} = \begin{bmatrix} 0.6645 & -0.6579 & -0.3544 & 0.3106 \\ -0.7455 & -0.6161 & -0.2543 & 0.2148 \\ -0.0510 & 0.4332 & -0.8998 & 0.4173 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Output: - Final Transformation ( $T_f$ ):**

$$T_f = \begin{bmatrix} 0.6645 & -0.6579 & -0.3544 & 0.3106 \\ -0.7455 & -0.6160 & -0.2543 & 0.2148 \\ -0.0510 & 0.4332 & -0.8998 & 0.4173 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Status:** Passed

## Summary

All test cases passed successfully, confirming the correctness of the inverse kinematics implementation for the given scenarios.

## Jacobian Tranpose Kinematics Test Report

This part of the report summarizes the results of the test cases executed using the `test_J_tranpose_kinematics.m` script. The purpose of these tests is to validate the implementation of the Jacobian Transpose Kinematics algorithm for the KUKA iiwa robot. All 5 test cases failed due to the final transformation matrices exceeding the specified tolerance threshold.

### Test Case 1

**Title:** Random Initial ( $T_i$ ) and Desired Final ( $T_{\text{des}}$ ) Configurations for Test 1

**Input: - Initial Transformation ( $T_i$ ):**

$$T_i = \begin{bmatrix} -0.1504 & -0.9492 & -0.2765 & -0.2091 \\ -0.9446 & 0.0554 & 0.3235 & 0.6553 \\ -0.2917 & 0.3098 & -0.9049 & 0.7099 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

- **Desired Final Transformation ( $T_{\text{des}}$ ):**

$$T_{\text{des}} = \begin{bmatrix} 0.3824 & 0.7999 & 0.4626 & 0.0850 \\ -0.5165 & -0.2301 & 0.8248 & 0.2239 \\ 0.7662 & -0.5543 & 0.3251 & 0.8460 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Output: - Final Transformation from Transpose Kinematics ( $T_f$ ):**

$$T_f = \begin{bmatrix} 0.3827 & 0.8002 & 0.4617 & 0.0890 \\ -0.5145 & -0.2305 & 0.8259 & 0.2282 \\ 0.7674 & -0.5536 & 0.3235 & 0.8552 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

- **Status:** Failed

## Test Case 2

**Title:** Random Initial ( $T_i$ ) and Desired Final ( $T_{\text{des}}$ ) Configurations for Test 2

**Input: - Initial Transformation ( $T_i$ ):**

$$T_i = \begin{bmatrix} -0.5353 & -0.8281 & 0.1663 & 0.6065 \\ -0.3875 & 0.4157 & 0.8229 & -0.4939 \\ -0.7505 & 0.3761 & -0.5434 & 0.3573 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

- **Desired Final Transformation ( $T_{\text{des}}$ ):**

$$T_{\text{des}} = \begin{bmatrix} -0.6990 & 0.1292 & -0.7033 & -0.3133 \\ -0.4887 & -0.8044 & 0.3379 & -0.2749 \\ -0.5221 & 0.5799 & 0.6255 & 0.9191 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Output: - Final Transformation from Transpose Kinematics ( $T_f$ ):**

$$T_f = \begin{bmatrix} -0.6986 & 0.1288 & -0.7038 & -0.3176 \\ -0.4887 & -0.8043 & 0.3379 & -0.2779 \\ -0.5225 & 0.5800 & 0.6249 & 0.9241 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

- **Status:** Failed

## Test Case 3

**Title:** Random Initial ( $T_i$ ) and Desired Final ( $T_{\text{des}}$ ) Configurations for Test 3

**Input: - Initial Transformation ( $T_i$ ):**

$$T_i = \begin{bmatrix} 0.3604 & -0.7432 & 0.5638 & -0.3436 \\ -0.0136 & 0.6001 & 0.7998 & 0.2173 \\ -0.9327 & -0.2959 & 0.2062 & 0.8587 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**- Desired Final Transformation ( $T_{\text{des}}$ ):**

$$T_{\text{des}} = \begin{bmatrix} 0.8733 & 0.0827 & 0.4801 & 0.6557 \\ -0.0819 & -0.9465 & 0.3120 & 0.1566 \\ 0.4803 & -0.3118 & -0.8198 & -0.0413 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Output: - Final Transformation from Transpose Kinematics ( $T_f$ ):**

$$T_f = \begin{bmatrix} 0.8659 & 0.0841 & 0.4932 & 0.6194 \\ -0.0849 & -0.9468 & 0.3104 & 0.1543 \\ 0.4930 & -0.3106 & -0.8127 & -0.0403 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**- Status:** Failed

#### Test Case 4

**Title:** Random Initial ( $T_i$ ) and Desired Final ( $T_{\text{des}}$ ) Configurations for Test 4

**Input: - Initial Transformation ( $T_i$ ):**

$$T_i = \begin{bmatrix} 0.9890 & -0.1194 & -0.0875 & -0.3084 \\ -0.0780 & -0.9227 & 0.3775 & -0.5331 \\ -0.1258 & -0.3665 & -0.9219 & 0.5055 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**- Desired Final Transformation ( $T_{\text{des}}$ ):**

$$T_{\text{des}} = \begin{bmatrix} -0.5802 & -0.6263 & -0.5207 & -0.2429 \\ -0.1115 & -0.5722 & 0.8125 & -0.1555 \\ -0.8068 & 0.5295 & 0.2621 & 0.8459 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Output: - Final Transformation from Transpose Kinematics ( $T_f$ ):**

$$T_f = \begin{bmatrix} -0.5801 & -0.6265 & -0.5205 & -0.2432 \\ -0.1114 & -0.5721 & 0.8126 & -0.1568 \\ -0.8069 & 0.5294 & 0.2621 & 0.8469 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**- Status:** Failed



## Test Case 5

**Title:** Random Initial ( $T_i$ ) and Desired Final ( $T_{\text{des}}$ ) Configurations for Test 5

**Input: - Initial Transformation ( $T_i$ ):**

$$T_i = \begin{bmatrix} -0.8115 & -0.5027 & -0.2979 & -0.6841 \\ 0.3778 & -0.0624 & -0.9238 & 0.1550 \\ 0.4458 & -0.8622 & 0.2405 & 0.0565 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**- Desired Final Transformation ( $T_{\text{des}}$ ):**

$$T_{\text{des}} = \begin{bmatrix} -0.4534 & 0.1308 & 0.8817 & -0.0312 \\ -0.8669 & -0.2948 & -0.4020 & 0.1496 \\ 0.2074 & -0.9466 & 0.2470 & 1.1411 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Output: - Final Transformation from Transpose Kinematics ( $T_f$ ):**

$$T_f = \begin{bmatrix} -0.4535 & 0.1320 & 0.8814 & -0.0287 \\ -0.8670 & -0.2946 & -0.4019 & 0.1413 \\ 0.2066 & -0.9465 & 0.2481 & 1.1135 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**- Status:** Failed

## Summary

The Jacobian Transpose Kinematics algorithm failed to converge to the desired final transformation ( $T_{\text{des}}$ ) in all test cases, and the maximum number of iterations was reached without robot convergence. The Jacobian Transpose method is not guaranteed to converge in all scenarios, especially for configurations requiring precise control or when the desired transformation is far from the initial configuration.

# Redundancy Resolution Test Report

This report summarizes the results of the redundancy resolution tests performed using the script `test_redundancy_resolution.m`. All test cases passed successfully.

## Test Case 1

**Title:** Redundancy Resolution Test 1

**Input:**

- Initial Transformation ( $T_i$ ):

$$T_i = \begin{bmatrix} 0.4754 & 0.2994 & 0.8273 & 0.2569 \\ -0.0423 & -0.9314 & 0.3614 & 0.7231 \\ 0.8788 & -0.2068 & -0.4301 & 0.6734 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

- Desired Final Transformation ( $T_{des}$ ):

$$T_{des} = \begin{bmatrix} -0.9966 & 0.0809 & -0.0127 & 0.4173 \\ 0.0018 & 0.1763 & 0.9843 & 0.4231 \\ 0.0818 & 0.9810 & -0.1759 & 0.2798 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Output:**

- Final Transformation ( $T_f$ ):

$$T_f = \begin{bmatrix} -0.9966 & 0.0809 & -0.0127 & 0.4173 \\ 0.0018 & 0.1763 & 0.9843 & 0.4231 \\ 0.0818 & 0.9810 & -0.1759 & 0.2798 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Status:** Passed

## Test Case 2

**Title:** Redundancy Resolution Test 2

**Input:**

- Initial Transformation ( $T_i$ ):

$$T_i = \begin{bmatrix} 0.3515 & -0.9359 & 0.0233 & 0.7305 \\ 0.3228 & 0.0978 & -0.9414 & -0.3704 \\ 0.8788 & 0.3384 & 0.3365 & 0.2668 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

- Desired Final Transformation ( $T_{des}$ ):

$$T_{des} = \begin{bmatrix} -0.6949 & 0.7007 & 0.1615 & -0.3468 \\ -0.3921 & -0.1809 & -0.9020 & -0.2863 \\ -0.6028 & -0.6901 & 0.4004 & 1.0058 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Output:**

- Final Transformation ( $T_f$ ):

$$T_f = \begin{bmatrix} -0.6949 & 0.7008 & 0.1615 & -0.3468 \\ -0.3921 & -0.1809 & -0.9020 & -0.2863 \\ -0.6029 & -0.6901 & 0.4004 & 1.0058 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Status:** Passed

### Test Case 3

**Title:** Redundancy Resolution Test 3

**Input:**

- Initial Transformation ( $T_i$ ):

$$T_i = \begin{bmatrix} -0.8840 & -0.3632 & -0.2943 & -0.5127 \\ 0.0735 & 0.5137 & -0.8548 & -0.7152 \\ 0.4617 & -0.7773 & -0.4274 & 0.4523 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

- Desired Final Transformation ( $T_{des}$ ):

$$T_{des} = \begin{bmatrix} 0.2629 & -0.9176 & -0.2980 & -0.0627 \\ -0.9044 & -0.1269 & -0.4073 & -0.7436 \\ 0.3359 & 0.3766 & -0.8633 & 0.6973 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Output:**

- Final Transformation ( $T_f$ ):

$$T_f = \begin{bmatrix} 0.2629 & -0.9176 & -0.2980 & -0.0627 \\ -0.9045 & -0.1269 & -0.4073 & -0.7436 \\ 0.3359 & 0.3766 & -0.8633 & 0.6972 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Status:** Passed

**Test Case 4****Title:** Redundancy Resolution Test 4**Input:**- Initial Transformation ( $T_i$ ):

$$T_i = \begin{bmatrix} -0.7230 & 0.6897 & 0.0391 & -0.5291 \\ 0.6749 & 0.7173 & -0.1732 & -0.4759 \\ -0.1475 & -0.0989 & -0.9841 & 0.1505 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

- Desired Final Transformation ( $T_{des}$ ):

$$T_{des} = \begin{bmatrix} -0.1873 & -0.2514 & -0.9496 & -0.3506 \\ -0.1718 & 0.9602 & -0.2203 & -0.5019 \\ 0.9672 & 0.1219 & -0.2230 & -0.0136 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Output:**- Final Transformation ( $T_f$ ):

$$T_f = \begin{bmatrix} -0.1873 & -0.2514 & -0.9496 & -0.3506 \\ -0.1718 & 0.9602 & -0.2203 & -0.5019 \\ 0.9672 & 0.1219 & -0.2230 & -0.0136 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Status:** Passed**Test Case 5****Title:** Redundancy Resolution Test 5**Input:**- Initial Transformation ( $T_i$ ):

$$T_i = \begin{bmatrix} -0.8535 & -0.3474 & 0.3884 & 0.5000 \\ -0.5068 & 0.3799 & -0.7738 & -0.6928 \\ 0.1213 & -0.8573 & -0.5003 & 0.5399 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

- Desired Final Transformation ( $T_{des}$ ):

$$T_{des} = \begin{bmatrix} 0.8139 & -0.5113 & -0.2759 & -0.1654 \\ 0.4204 & 0.1904 & 0.8871 & 0.2352 \\ -0.4011 & -0.8380 & 0.3699 & 0.8596 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Output:**- Final Transformation ( $T_f$ ):

$$T_f = \begin{bmatrix} 0.8139 & -0.5113 & -0.2759 & -0.1654 \\ 0.4204 & 0.1904 & 0.8871 & 0.2352 \\ -0.4011 & -0.8380 & 0.3699 & 0.8596 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Status:** Passed

## Damped Least-Square Inverse Kinematics Test Report

This report summarizes the results of running the `test_DLS_inverse_kinematics.m` script. The script tests the Damped Least Squares (DLS) inverse kinematics implementation for a KUKA iiwa robot. All test cases passed successfully.

### Test Case 1

**Title:** Random Initial and Final Configurations for Test 1

**Input:**

- **Initial Transformation ( $T_i$ ):**

$$T_i = \begin{bmatrix} 0.0342 & -0.5031 & -0.8636 & -0.2375 \\ 0.7982 & -0.5062 & 0.3266 & 0.7620 \\ -0.6015 & -0.7004 & 0.3842 & 0.4509 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

- **Desired Final Transformation ( $T_{\text{des}}$ ):**

$$T_{\text{des}} = \begin{bmatrix} -0.3874 & -0.8674 & -0.3122 & -0.2902 \\ -0.5699 & -0.0409 & 0.8207 & -0.2361 \\ -0.7246 & 0.4959 & -0.4785 & 0.9085 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Output:**

- **Final Transformation from Inverse Kinematics ( $T_f$ ):**

$$T_f = \begin{bmatrix} -0.3874 & -0.8674 & -0.3123 & -0.2902 \\ -0.5699 & -0.0409 & 0.8207 & -0.2361 \\ -0.7246 & 0.4959 & -0.4785 & 0.9085 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Status:** Passed

## Test Case 2

**Title:** Random Initial and Final Configurations for Test 2

**Input:**

- **Initial Transformation ( $T_i$ ):**

$$T_i = \begin{bmatrix} 0.7661 & 0.0486 & -0.6409 & 0.6090 \\ -0.6093 & -0.2622 & -0.7483 & -0.5273 \\ -0.2044 & 0.9638 & -0.1712 & 0.4619 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

- **Desired Final Transformation ( $T_{\text{des}}$ ):**

$$T_{\text{des}} = \begin{bmatrix} -0.4356 & -0.8090 & -0.3946 & -0.2674 \\ -0.3914 & -0.2245 & 0.8924 & -0.2798 \\ -0.8106 & 0.5432 & -0.2189 & -0.1997 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Output:**

- **Final Transformation from Inverse Kinematics ( $T_f$ ):**

$$T_f = \begin{bmatrix} -0.4356 & -0.8090 & -0.3946 & -0.2674 \\ -0.3914 & -0.2246 & 0.8924 & -0.2798 \\ -0.8106 & 0.5432 & -0.2189 & -0.1997 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Status:** Passed

## Test Case 3

**Title:** Random Initial and Final Configurations for Test 3

**Input:**

- **Initial Transformation ( $T_i$ ):**

$$T_i = \begin{bmatrix} -0.1002 & -0.7179 & 0.6889 & -0.4548 \\ 0.6160 & 0.4990 & 0.6095 & 0.5286 \\ -0.7813 & 0.4855 & 0.3922 & 0.8021 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

- **Desired Final Transformation ( $T_{\text{des}}$ ):**

$$T_{\text{des}} = \begin{bmatrix} -0.7542 & 0.1407 & -0.6414 & 0.2591 \\ -0.2366 & -0.9694 & 0.0656 & 0.7431 \\ -0.6126 & 0.2013 & 0.7644 & 0.5350 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Output:**

- **Final Transformation from Inverse Kinematics ( $T_f$ ):**

$$T_f = \begin{bmatrix} -0.7542 & 0.1407 & -0.6414 & 0.2591 \\ -0.2366 & -0.9694 & 0.0656 & 0.7431 \\ -0.6126 & 0.2013 & 0.7644 & 0.5350 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Status:** Passed

#### Test Case 4

**Title:** Random Initial and Final Configurations for Test 4

**Input:**

- **Initial Transformation ( $T_i$ ):**

$$T_i = \begin{bmatrix} -0.5763 & -0.8163 & 0.0398 & -0.2526 \\ 0.4922 & -0.3078 & 0.8143 & -0.1436 \\ -0.6524 & 0.4888 & 0.5791 & 1.1383 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

- **Desired Final Transformation ( $T_{\text{des}}$ ):**

$$T_{\text{des}} = \begin{bmatrix} 0.4596 & -0.4903 & 0.7405 & -0.0789 \\ 0.3696 & 0.8638 & 0.3425 & 0.1835 \\ -0.8076 & 0.1163 & 0.5782 & 0.8410 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Output:**

- **Final Transformation from Inverse Kinematics ( $T_f$ ):**

$$T_f = \begin{bmatrix} 0.4596 & -0.4903 & 0.7405 & -0.0789 \\ 0.3696 & 0.8638 & 0.3425 & 0.1835 \\ -0.8076 & 0.1163 & 0.5782 & 0.8410 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Status:** Passed

#### Test Case 5

**Title:** Random Initial and Final Configurations for Test 5

**Input:**

- **Initial Transformation ( $T_i$ ):**

$$T_i = \begin{bmatrix} 0.5397 & 0.8268 & 0.1587 & -0.5720 \\ -0.8314 & 0.4939 & 0.2547 & 0.2029 \\ 0.1322 & -0.2694 & 0.9539 & 0.9353 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

- **Desired Final Transformation ( $T_{\text{des}}$ ):**

$$T_{\text{des}} = \begin{bmatrix} 0.9736 & 0.2091 & -0.0919 & -0.2955 \\ -0.2247 & 0.9493 & -0.2199 & 0.4106 \\ 0.0412 & 0.2348 & 0.9712 & 0.7209 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Output:**

- **Final Transformation from Inverse Kinematics ( $T_f$ ):**

$$T_f = \begin{bmatrix} 0.9736 & 0.2091 & -0.0919 & -0.2955 \\ -0.2247 & 0.9493 & -0.2199 & 0.4106 \\ 0.0412 & 0.2348 & 0.9712 & 0.7209 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Status:** Passed

## Inverse Kinematics Comparison Test Report

This report summarizes the results of the inverse kinematics comparison tests performed using the `test_IK_comparison.m` script. The script compares the performance of different inverse kinematics methods for a KUKA iiwa robot. All test cases passed successfully.

---

### Test Case 1: Inverse Jacobian

**Title:** Inverse Jacobian

**Input:**

- **Initial Transformation ( $T_i$ ):**

$T_i$  = Randomly generated transformation matrix

- **Desired Final Transformation ( $T_{\text{des}}$ ):**

$$T_{\text{des}} = \begin{bmatrix} -0.5332 & 0.4307 & -0.7282 & -0.2595 \\ -0.8367 & -0.3955 & 0.3788 & 0.5770 \\ -0.1248 & 0.8112 & 0.5712 & 0.8642 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Output:**

- **Final Transformation ( $T_f$ ):**



$$T_f = \begin{bmatrix} -0.5332 & 0.4307 & -0.7282 & -0.2595 \\ -0.8367 & -0.3955 & 0.3788 & 0.5770 \\ -0.1248 & 0.8112 & 0.5712 & 0.8642 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

- **Number of Iterations:**  $ik\_iter = 11$

**Status:** Passed

---

## Test Case 2: Transpose Jacobian

**Title:** Transpose Jacobian

**Input:**

- **Initial Transformation ( $T_i$ ):**

$T_i$  = Randomly generated transformation matrix

- **Desired Final Transformation ( $T_{des}$ ):**

$$T_{des} = \begin{bmatrix} -0.5332 & 0.4307 & -0.7282 & -0.2595 \\ -0.8367 & -0.3955 & 0.3788 & 0.5770 \\ -0.1248 & 0.8112 & 0.5712 & 0.8642 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Output:**

- **Final Transformation ( $T_f$ ):**

$$T_f = \begin{bmatrix} -0.5331 & 0.4306 & -0.7282 & -0.2588 \\ -0.8368 & -0.3954 & 0.3788 & 0.5754 \\ -0.1248 & 0.8113 & 0.5711 & 0.8632 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

- **Number of Iterations:**  $tranpose\_iter = 705$

**Status:** Passed

---

## Test Case 3: Redundancy Resolution

**Title:** Redundancy Resolution

**Input:**

- **Initial Transformation ( $T_i$ ):**

$T_i$  = Randomly generated transformation matrix

- **Desired Final Transformation ( $T_{\text{des}}$ ):**

$$T_{\text{des}} = \begin{bmatrix} -0.5332 & 0.4307 & -0.7282 & -0.2595 \\ -0.8367 & -0.3955 & 0.3788 & 0.5770 \\ -0.1248 & 0.8112 & 0.5712 & 0.8642 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

**Output:**

- **Final Transformation ( $T_f$ ):**

$$T_f = \begin{bmatrix} -0.5332 & 0.4307 & -0.7282 & -0.2595 \\ -0.8367 & -0.3955 & 0.3788 & 0.5770 \\ -0.1248 & 0.8112 & 0.5712 & 0.8642 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

- **Number of Iterations:**  $red\_iter = 11$

**Status:** Passed

---

## Test Case 4: Damped Least Squares (DLS)

**Title:** Damped Least Squares

**Input:**

- **Initial Transformation ( $T_i$ ):**

$T_i$  = Randomly generated transformation matrix

- **Desired Final Transformation ( $T_{\text{des}}$ ):**

$$T_{\text{des}} = \begin{bmatrix} -0.5332 & 0.4307 & -0.7282 & -0.2595 \\ -0.8367 & -0.3955 & 0.3788 & 0.5770 \\ -0.1248 & 0.8112 & 0.5712 & 0.8642 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

- **Damping Factor:**  $\lambda = 0.01$

**Output:**

- **Final Transformation ( $T_f$ ):**

$$T_f = \begin{bmatrix} -0.5332 & 0.4307 & -0.7282 & -0.2595 \\ -0.8367 & -0.3955 & 0.3788 & 0.5770 \\ -0.1248 & 0.8112 & 0.5712 & 0.8642 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

- **Number of Iterations:**  $dls\_iter = 10$

**Status:** Passed

---

## Summary

All test cases passed successfully, confirming the correctness of the inverse kinematics methods. The number of iterations required for each method is summarized below:

Method	Iterations
Inverse Jacobian	11
Transpose Jacobian	705
Redundancy Resolution	11
Damped Least Squares	10