TECHNISCHE FAKULTÄT

FRIEDRICH-ALEXANDER

UNIVERSITÄT

ERLANGEN-NÜRNBERG



**Robot Mechanisms and User Interfaces WS24/25**

**MDP Calculation Report**

submitted to

Chair of Autonomous Systems and Mechatronics

Faculty of Electrical Engineering

Friedrich–Alexander University

by

MDP Group 4

Arun Ravindranath

Lucie Bonenberger

Ajinkya Sathe

Aravind Palakkal

Surya Manusani

Date of Submission: December 19, 2024

*Professor*:

Prof. Dr.-Ing. Philipp Beckerle

*Supervisor*:

Rodrigo J. Velasco-Guillen, M.Sc.

**Contents**

[1. The robot and the interface 1](#_Toc1)

[2. Kinematics analysis 2](#_Toc2)

[2.1. For the robot 2](#_Toc3)

[2.1.1. Frame assignments 2](#_Toc4)

[2.1.2. DH-parameters (symbolic) 2](#_Toc5)

[2.1.3. differential kinematics (symbolic) 2](#_Toc6)

[2.1.4. workspace analysis 6](#_Toc7)

[2.1.5. singularity analysis (symbolic) 6](#_Toc8)

[2.1.6. manipulability analysis. 7](#_Toc9)

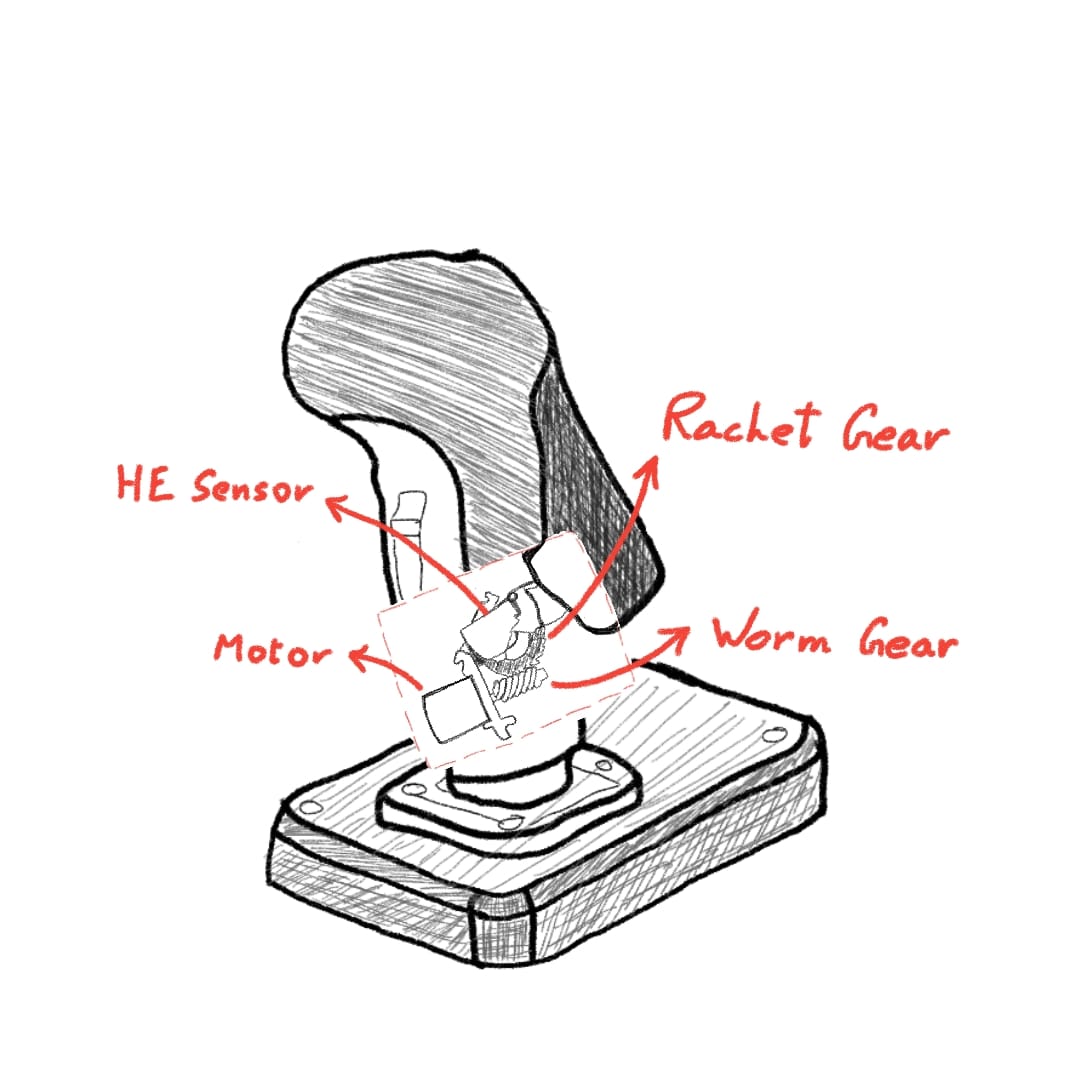
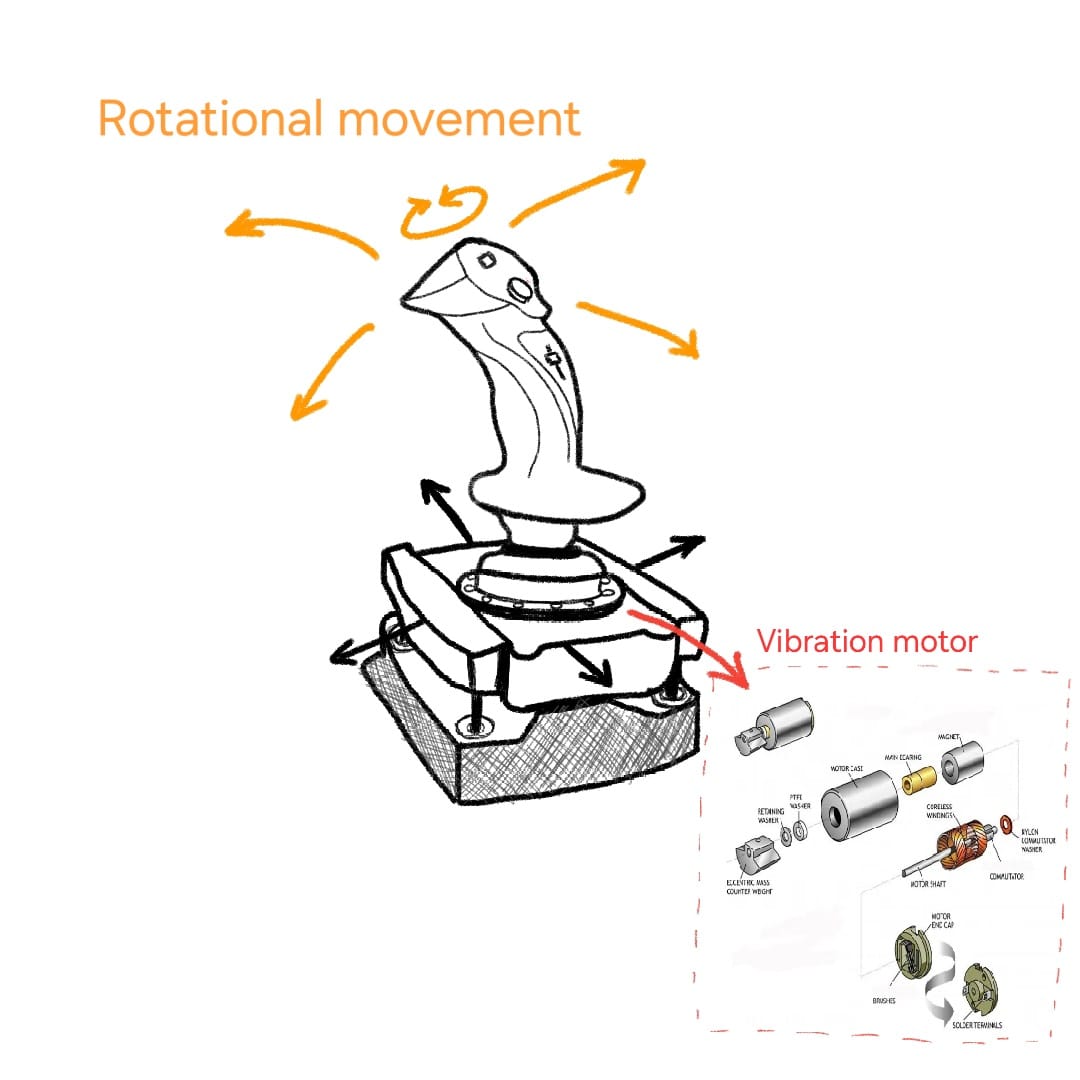
[2.2. For the Interface 7](#_Toc10)

[2.2.1. Generic angle measurement (symbolic), i.e., mapping resistances to angles 7](#_Toc11)

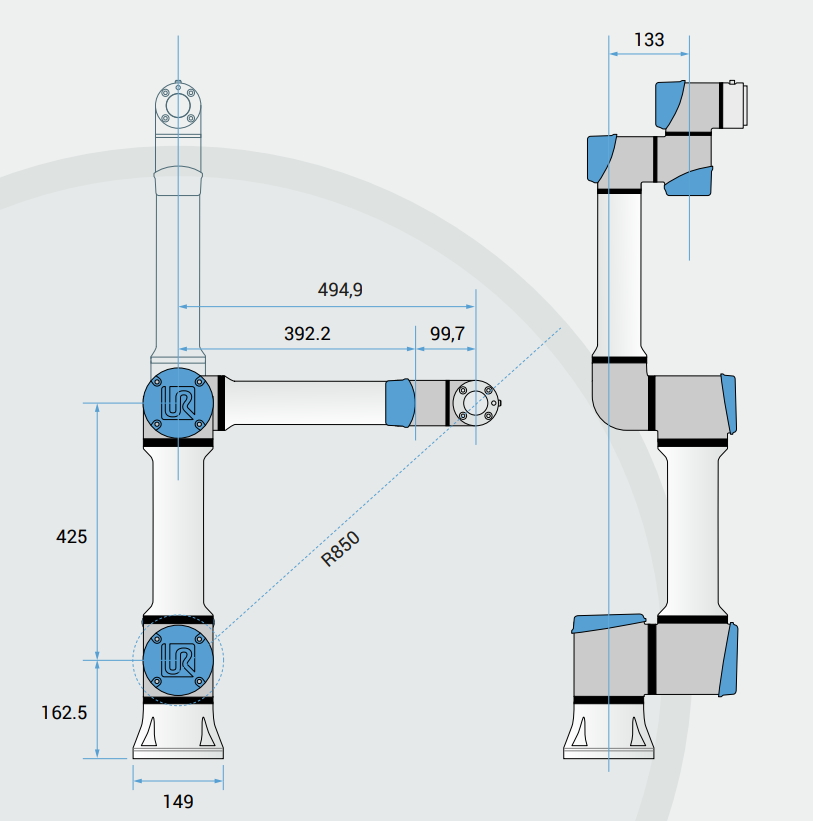
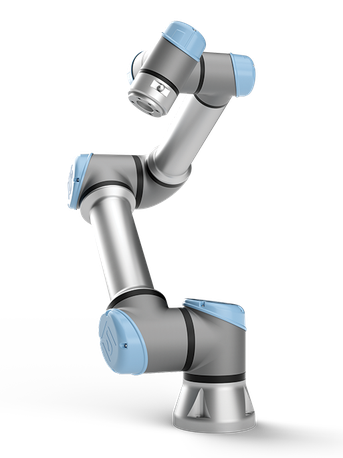
[2.2.2. Open-loop cartesian velocity control of the end-effector, i.e., mapping from rotation angles of the interface to cartesian velocity of the end-effector, and mapping cartesian velocities to joint velocities 7](#_Toc12)

[2.2.3. Mapping between gripping force and haptic feedback 7](#_Toc13)

# The robot and the interface



The interface is an innovative design concept utilizing a dual-stick configuration aimed at enhancing motion control in robotics. The system is designed to separate movement control from end-effector operation, offering a more intuitive user experience. A control slider allows users to switch between modes seamlessly, while resistance triggers, utilizing a worm gear assembly, ensure precise actuation. The joystick, mounted on a suspended platform, provides five degrees of freedom (DOF) detected via potentiometers, with an additional thumb-stick dedicated to translational Z-axis movements. The interface connects to a computer via USB-C, serving both power and data transmission needs.



The robot is a UR5e universal robot, which has a payload of 5kg. For the object, a maximum weight of 2.5 kg and a maximum effective diameter of 100 mm is considered.

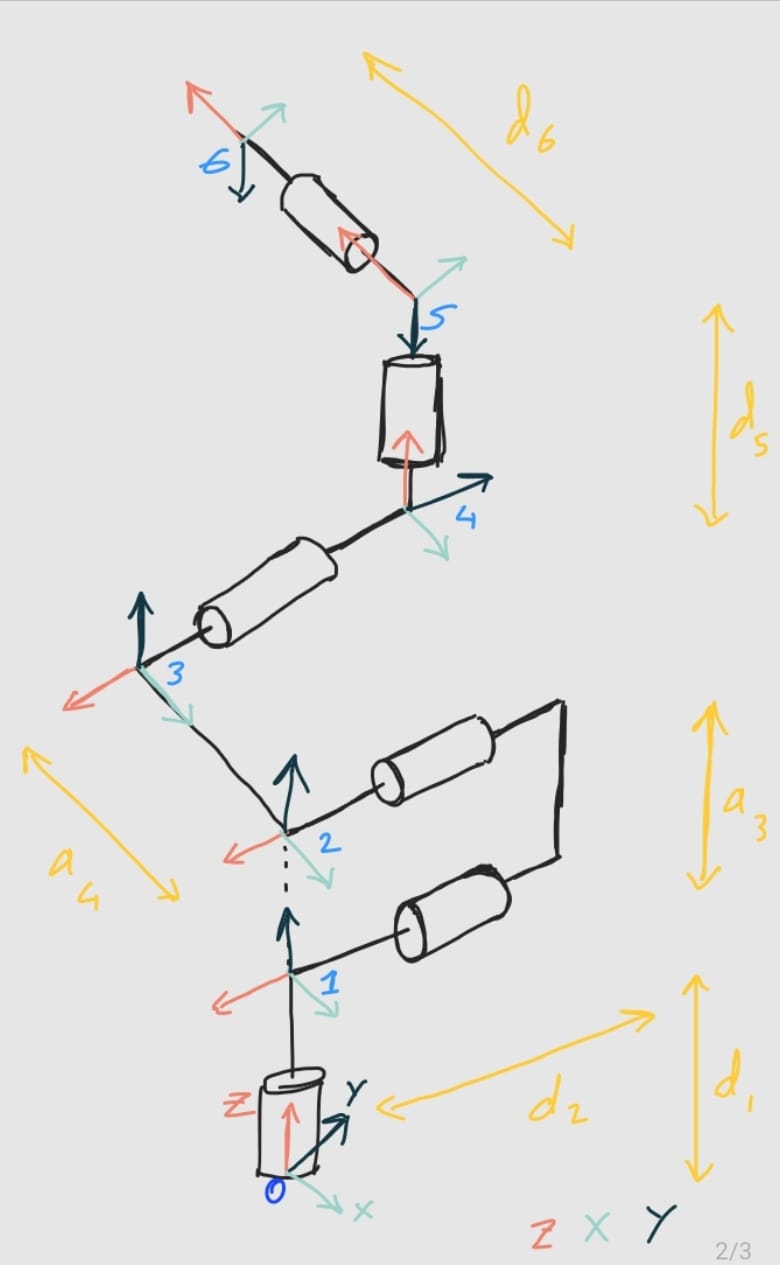
# Kinematics analysis

Translational movement

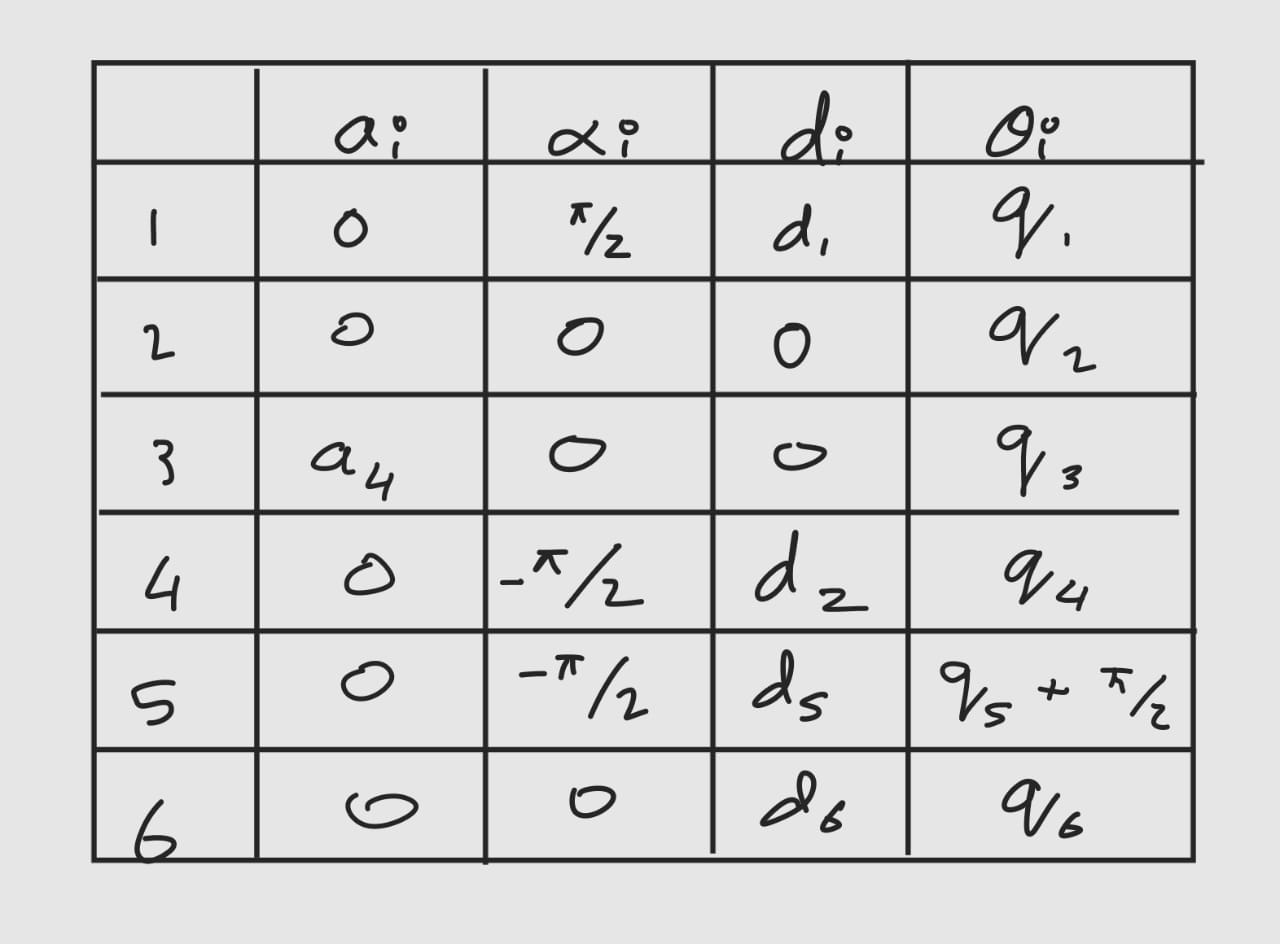
The kinematic analysis of the robot and the interface was done using MATLAB:

## For the robot

## Frame assignments

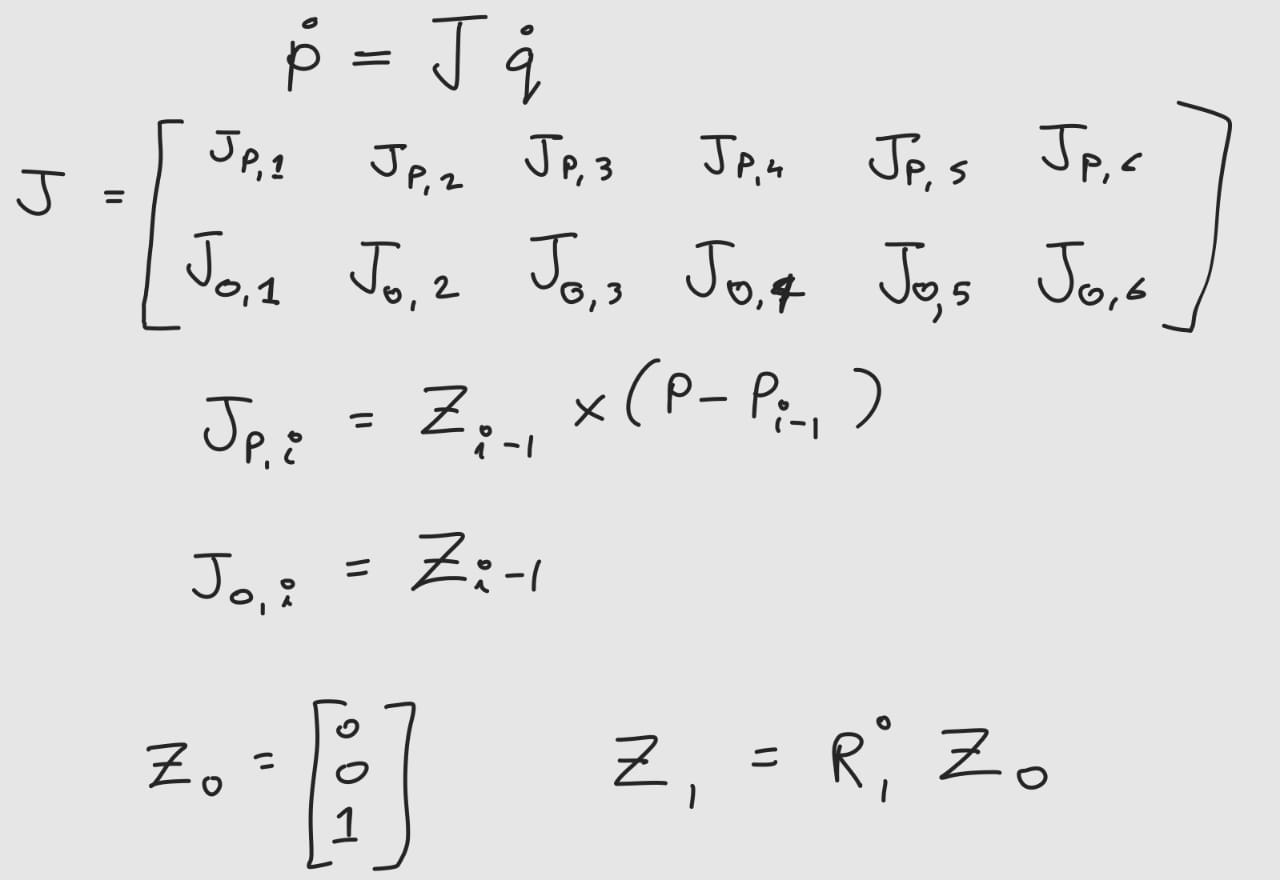


## DH-parameters (symbolic)



## differential kinematics (symbolic)

We use differential kinematics, as the interface is joystick based. This makes forward an inverse kinematics difficult to calculate.



P-dot represents the end-effector velocities (both translational and rotational), Q-dot represents the joint-angles of the robot arm, and J represents the Jacobian matrix.

The following is the MATLAB code to calculate the Jacobian matrix:

syms theta1 theta2 theta3 theta4 theta5 theta6 real

syms d1 d2 d3 d4 d5 d6 real

syms a1 a2 a3 a4 a5 a6 real

syms alpha1 alpha2 alpha3 alpha4 alpha5 alpha6 real

syms q1 q2 q3 q4 q5 q6 real

q = [theta1; theta2; theta3; theta4; theta5; theta6];

d1 = d1; d2 = 0; d3 = 0;

d4 = 133; d5 = 99.7; d6 = 133;

a1 = 0; a2 = 0; a3 = 425;

a4 = 0; a5 = 0; a6 = 0;

alpha1 = (pi/2); alpha2 = 0; alpha3 = 0;

alpha4 = -(pi/2); alpha5 = -(pi/2); alpha6 = 0;

theta1 = (pi/3); theta2 = (pi/6); theta3 = (pi/3);

theta4 = (pi/6); theta5 = (pi/4)+(pi/2); theta6 = (pi/6);

DH = [theta1 alpha1 a1 d1;

theta2 alpha2 a2 d2;

theta3 alpha3 a3 d3;

theta4 alpha4 a4 d4;

theta5 alpha5 a5 d5;

theta6 alpha6 a6 d6];

% Initialize

T = eye(4);

T\_all = sym(zeros(4, 4, 6));

% Forward Kinematics

for i = 1:6

theta = DH(i,1);

d = DH(i,2);

a = DH(i,3);

alpha = DH(i,4);

T\_i = [cos(theta) -sin(theta)\*cos(alpha) sin(theta)\*sin(alpha) a\*cos(theta);

sin(theta) cos(theta)\*cos(alpha) -cos(theta)\*sin(alpha) a\*sin(theta);

0 sin(alpha) cos(alpha) d;

0 0 0 1];

T = T \* T\_i;

T\_all(:,:,i) = T; % Save each intermediate transformation matrix

end

pe = T(1:3, 4); % End-effector pos

R = T(1:3, 1:3); % End-effector rot

J = sym(zeros(6, 6));

for i = 1:6

R\_prev = T\_all(1:3, 1:3, i);

p\_prev = T\_all(1:3, 4, i);

if i == 1

z\_prev = [0; 0; 1]; % Base frm z-axis

p\_prev = [0; 0; 0]; % Base frm org

else

z\_prev = T\_all(1:3, 3, i-1); % Prev z-axis

end

Jv = cross(z\_prev, pe - p\_prev);

Jw = z\_prev;

J(1:3, i) = Jv; % Lin vel

J(4:6, i) = Jw; % Ang vel

end

J = simplify(J);

## workspace analysis

The workspace is analyzed by generating multiple configurations of the robot by varying joint angles across their range. The end-effector positions for each configuration are computed, and the reachable points are plotted. Below is the MATLAB code for workspace analysis:

%% Workspace Analysis

n\_samples = 50; % Number of samples per joint

q\_ranges = linspace(-pi, pi, n\_samples);

workspace\_points = []; % Store reachable points

for q1 = q\_ranges

for q2 = q\_ranges

for q3 = q\_ranges

DH(:,1) = [q1; q2; q3; pi/4; pi/6; pi/3];

T = eye(4);

for i = 1:6

theta = DH(i,1);

alpha = DH(i,2);

a = DH(i,3);

d = DH(i,4);

T\_i = [cos(theta), -sin(theta)\*cos(alpha), sin(theta)\*sin(alpha), a\*cos(theta);

sin(theta), cos(theta)\*cos(alpha), -cos(theta)\*sin(alpha), a\*sin(theta);

0, sin(alpha), cos(alpha), d;

0, 0, 0, 1];

T = T \* T\_i;

end

workspace\_points = [workspace\_points; T(1:3,4)'];

end

end

end

% Plot workspace

figure;

scatter3(workspace\_points(:,1), workspace\_points(:,2), workspace\_points(:,3), 1, 'b.');

title('Workspace of the 6-DOF Robot');

xlabel('X (m)');

ylabel('Y (m)');

zlabel('Z (m)');

grid on;

## singularity analysis (symbolic)

disp('Jacobian matrix:');

disp(J);

if det(J) ~= 0

J\_inv = inv(J); % Compute inverse of the Jacobian

disp('Inverse of the Jacobian matrix:');

disp(J\_inv);

else

disp('Jacobian matrix is singular and not invertible.');

end

## manipulability analysis.

%% Manipulability Analysis

disp('Performing manipulability analysis...');

manipulability = sqrt(det(J \* J'));

disp(['Manipulability measure: ', num2str(manipulability)]);

## For the Interface

## Open-loop cartesian velocity control of the end-effector

We use the Jacobian matrix (J) to map joystick inputs (Cartesian velocity) to joint velocities

## Mapping between gripping force and haptic feedback

**Calculation Aspect**

**The robot’s 6DOF motion typically includes:**

* **3 translational DOFs**: X, Y, Z (movement in Cartesian space).
* **3 rotational DOFs**: Roll, Pitch, Yaw (orientation changes).

**Example Mapping:**

* **Joystick 1**:
  + X-axis: Move the robot along the X-axis.
  + Y-axis: Move the robot along the Y-axis.
* **Joystick 2**:
  + X-axis: Control yaw.
  + Y-axis: Control pitch.
* **Trigger or Slider Potentiometers**: Control Z-axis movement or roll.

The potentiometer's output voltage can directly or indirectly control these motions, depending on whether the joystick is configured for position control, velocity control, or force/torque control.

**Software Implementation**

1. **Gripper Sensor Data Processing**
   * Collect data from:
     1. Force/tactile sensors for Fgrip.
     2. Strain sensors for gripper deformation.
   * Preprocess data (filtering, smoothing).
   * Transmit processed data to the joystick controller.
2. **Feedback Control Loop**

Implement in your controller's firmware:

1. Read joystick position Jpos​ and gripper sensor data Fgrip​.
2. Calculate feedback intensity feedback​.
3. Adjust joystick feedback actuators (motors or vibrators) based on feedback.