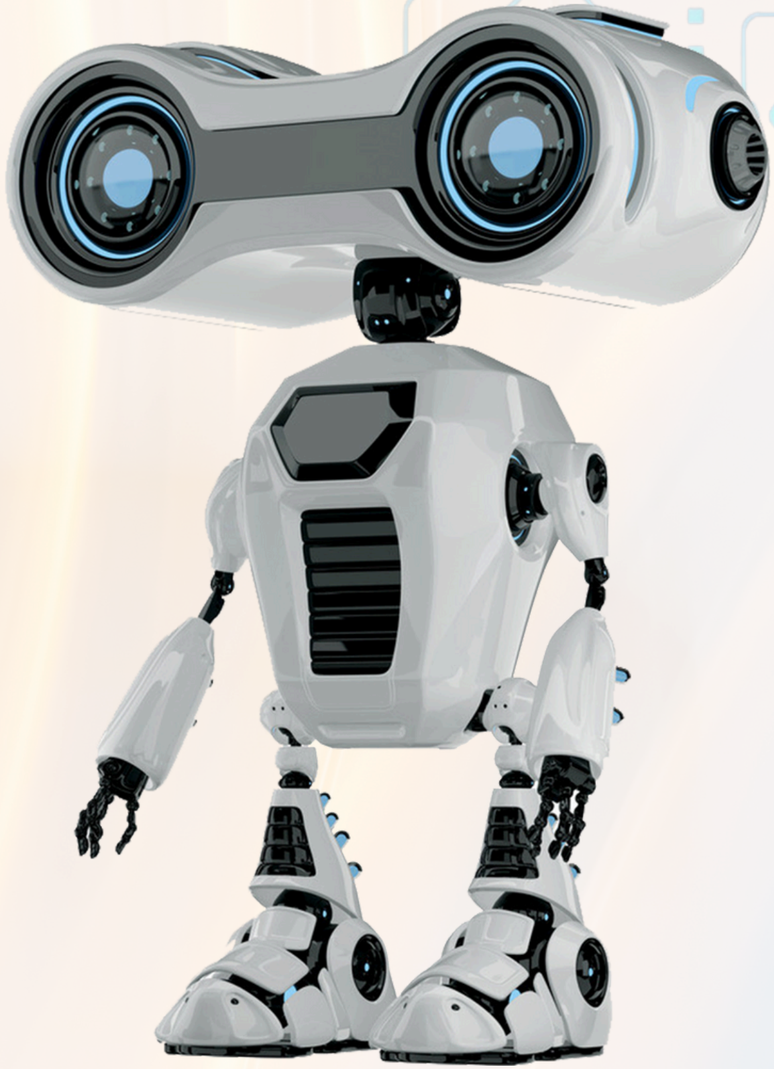
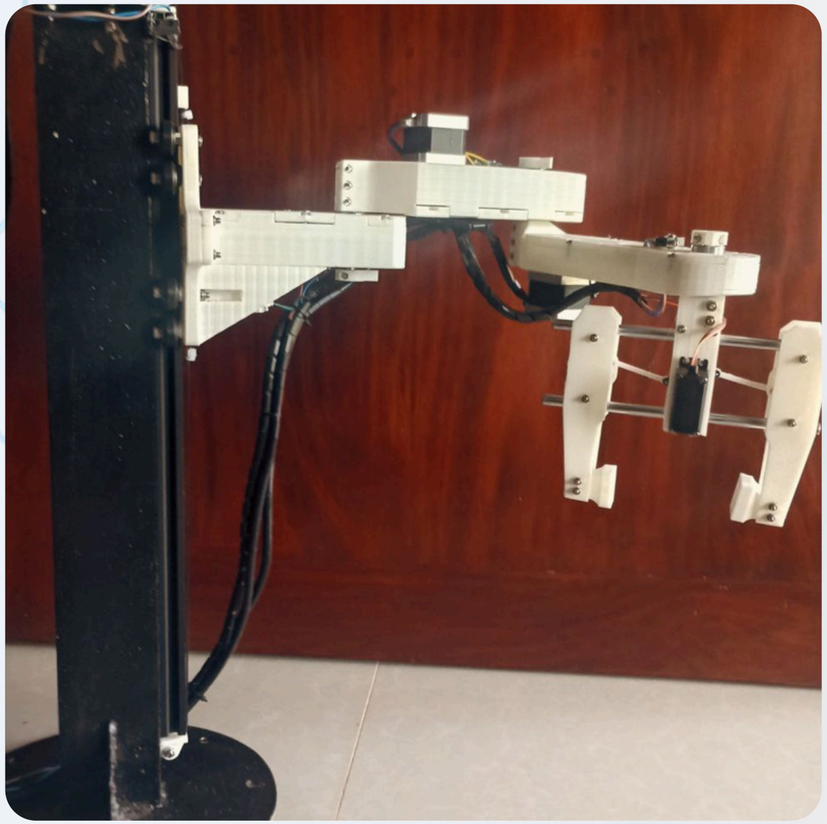
Kinematic Analysis of a Robot Arm

Hirusha Maduwantha - 200370K

Robotics Mini Project



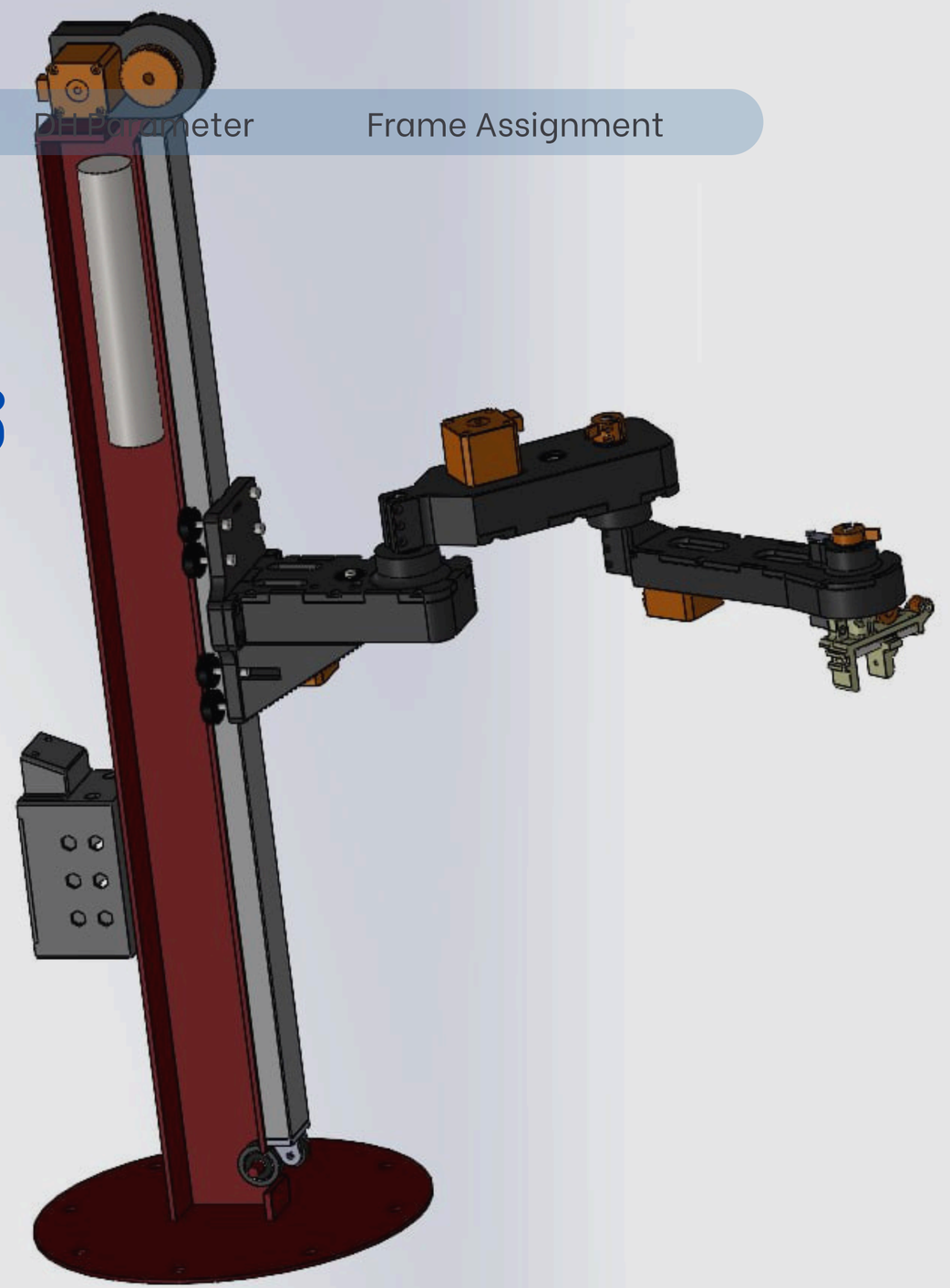


Introduction

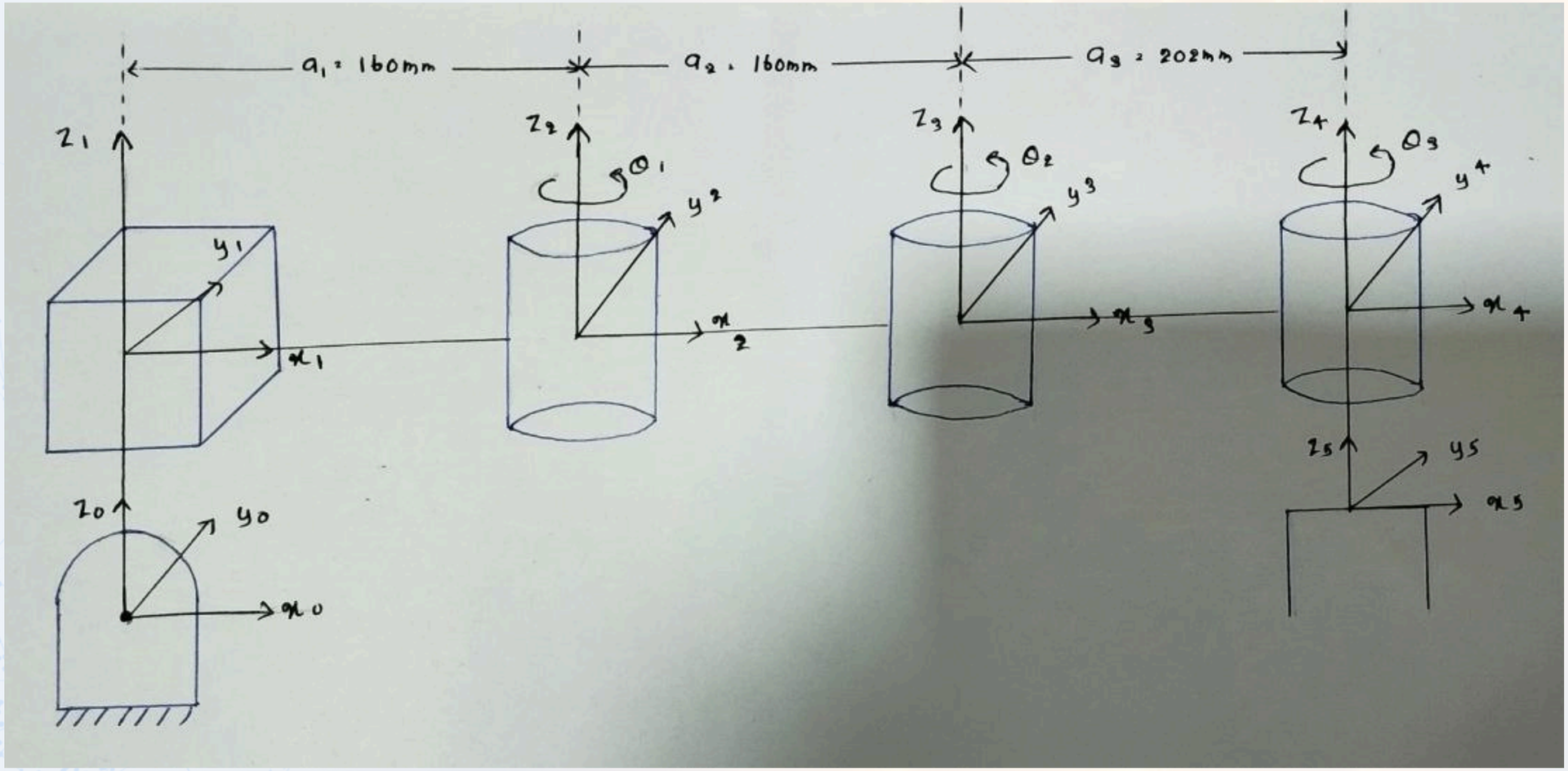
- The project involves the kinematic analysis of a SCARA (PRRR) robot arm with 4 DoF.
- Tasks include forward kinematics, inverse kinematics, Jacobian analysis, and a pick-and-place demonstration.
- Designed and built a functional SCARA robot with a gripper, powered by stepper motors, capable of holding objects up to 0.8 kg

SCARA Robot Details

- Type: PRRR SCARA Robot with 4 DoF.
- Dimensions:
  - a1=160 mm
  - a2=220 mm
  - a3=202 mm
  - d4=60 mm
- Components:
  - Arduino Uno, CNC Shield, DRV8825 Drivers
  - NEMA 17 stepper motors,3D-printed parts.
- 12V, 10A power supply.
- Functional gripper to handle round and square objects.



SCARA Robot-Coordinate frame assignments



Denavit-Hartenberg (DH) Parameters

Joint (iii)	ai (mm)	αi	di(mm)	θi
1 (Prismatic)	0	0	d1(variable)	0
2 (Revolute)	160	0	0	θ1(variable)
3 (Revolute)	160	0	0	θ2(variable)
4 (Revolute)	202	0	0	θ3(variable)
5	0	0	60	0

Forward Kinematics

- Transformation matrix for each joint:

$$\begin{bmatrix} C_\theta & -S_\theta C_\alpha & S_\theta S_\alpha & aC_\theta \\ S_\theta & C_\theta C_\alpha & -C_\theta S_\alpha & aS_\theta \\ 0 & S_\alpha & C_\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- End-effector transformation matrix:

$$T_4{}^0 = T_1{}^0 \times T_2{}^1 \times T_3{}^2 \times T_4{}^3$$

- Position of the end-effector (x,y,z) calculated from the matrix.

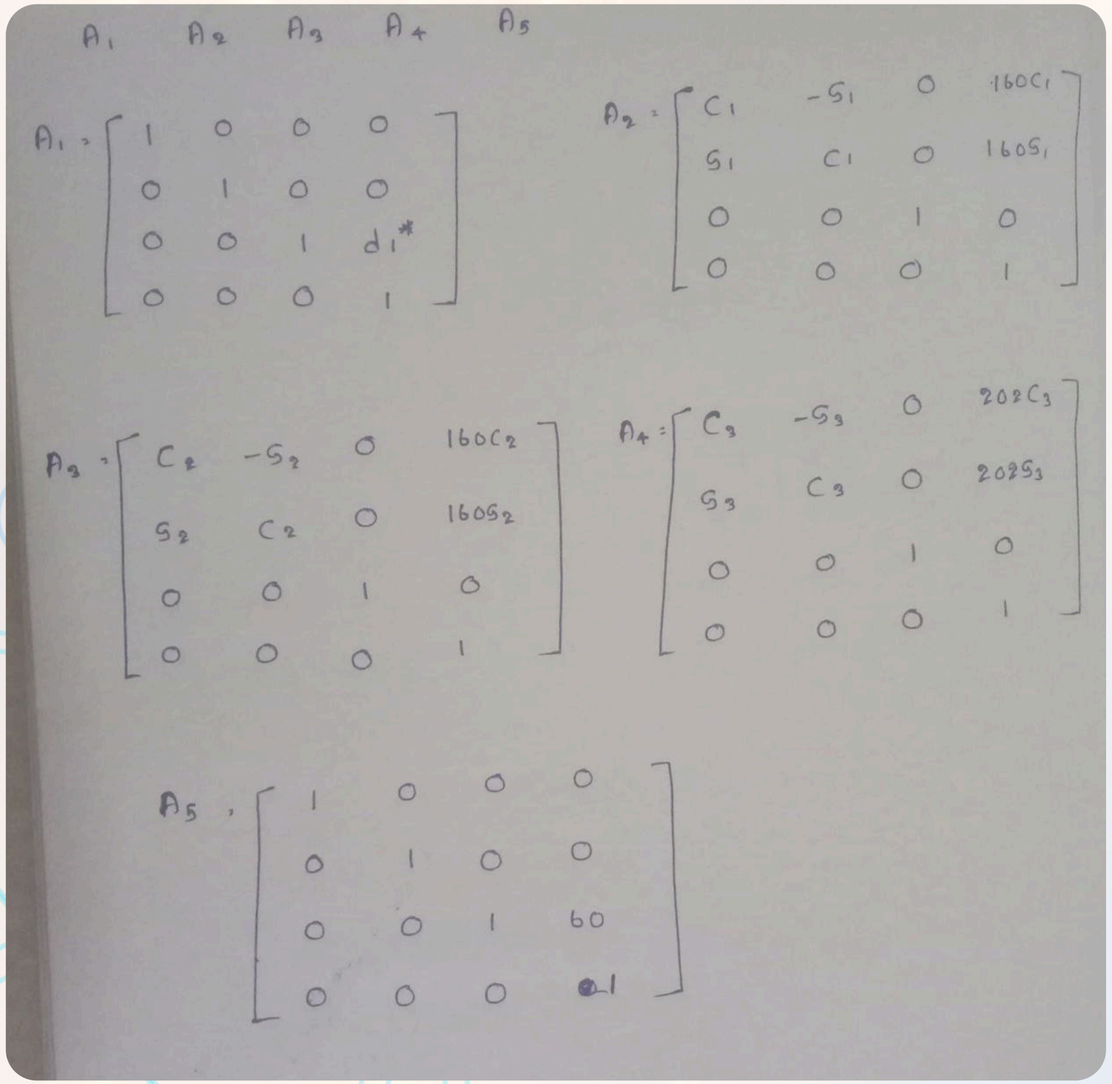
Forward Kinematics

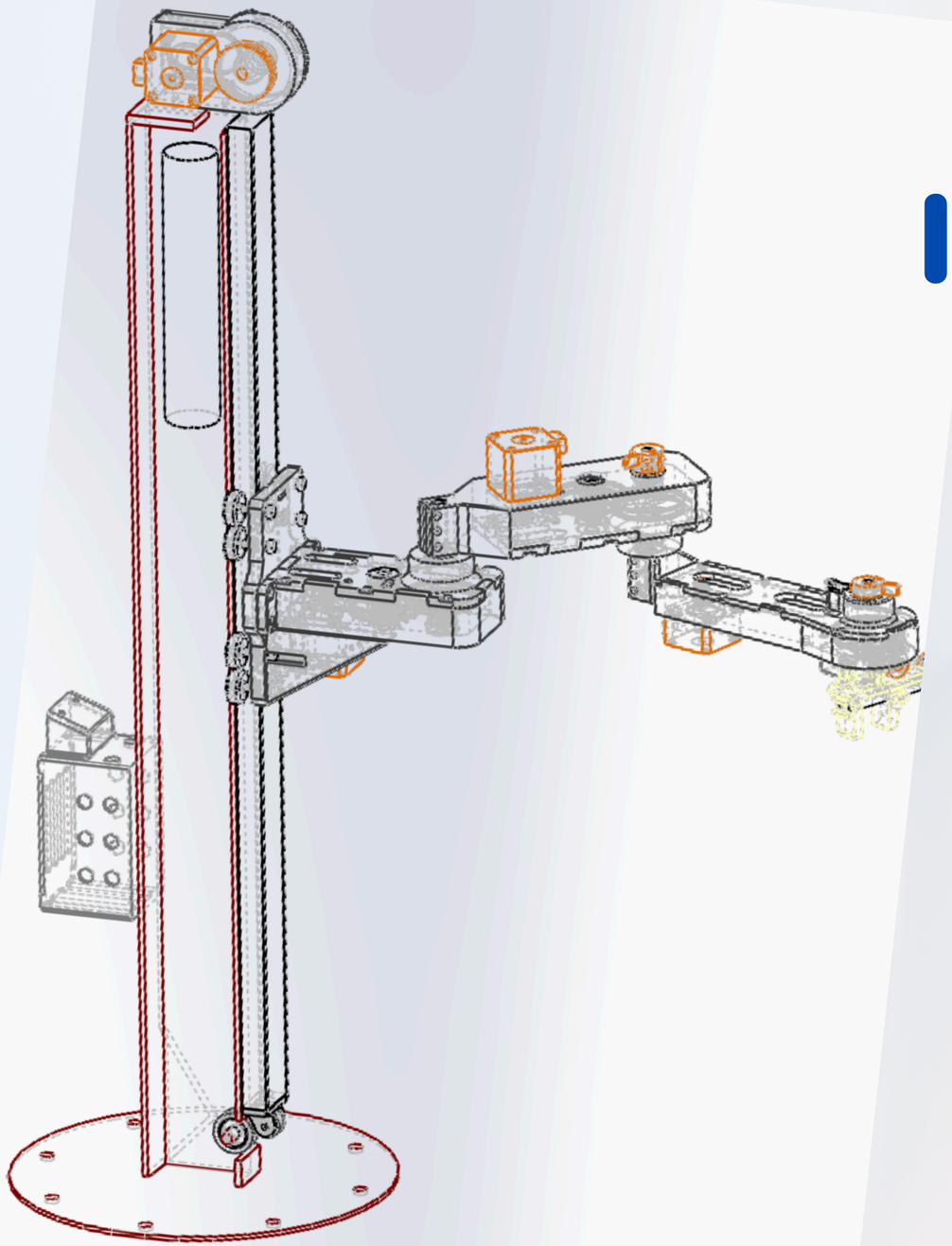
For the end-effector position:

- x=a1sin(90)+a2cos(θ1)+a3cos(θ1+θ2)
- x = 160 + 160 cos(θ1)+202cos(θ1+θ2)

- y=a1cos(90)+a2sin(θ1)+a3sin(θ1+θ2)
- y= 0 +160sin(θ1)+202sin(θ1+θ2)

- z = z0 + d1*





Inverse Kinematics

Solve for θ1 and θ2:

1. Projection on XY plane:

$$r = \sqrt{x^2 + y^2} \quad \text{(distance from origin in XY-plane)}.$$

Use the law of cosines to solve for θ2:

$$\cos\theta_2 = \frac{r^2 - a_2^2 - a_3^2}{2a_3a_2}.$$

$$\theta_2 = \arccos(\cos\theta_2).$$

Solve for θ1 using:

$$\theta_1 = \arctan 2(y, x) - \arctan 2\left(\frac{a_3 \sin\theta_2}{a_2 + a_3\cos\theta_2}\right).$$

Manipulator Jacobian

The Jacobian matrix is:

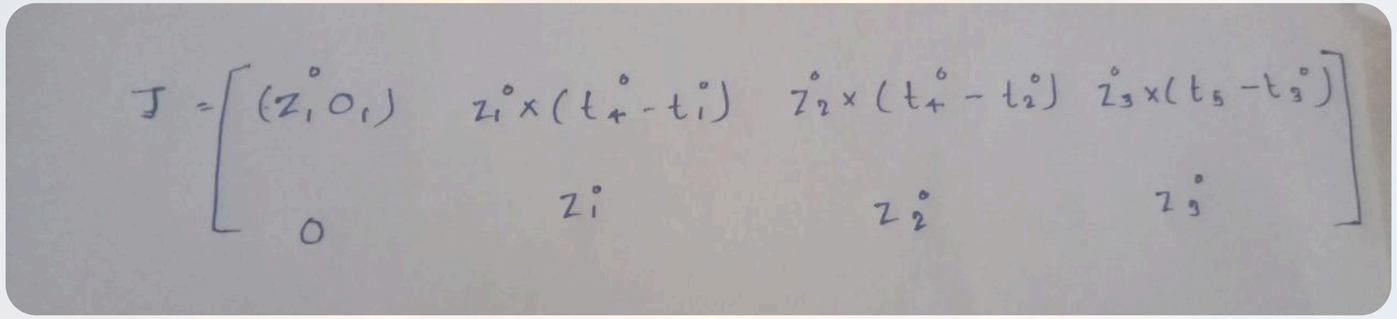
$$J = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix},$$

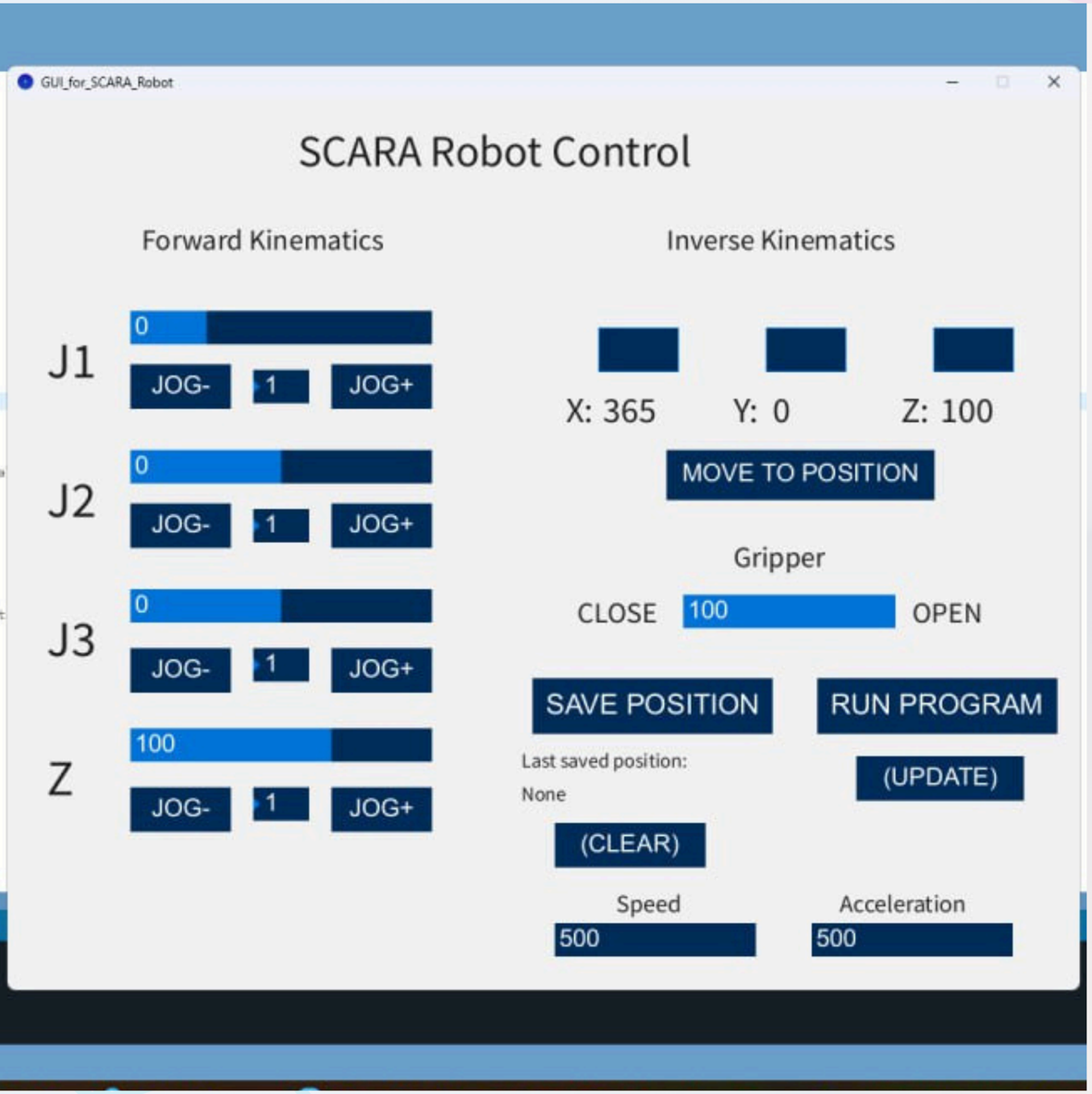
Where:

$$J_v = \begin{bmatrix} \mathbf{z}_0 & \mathbf{z}_1 \times (\mathbf{o}_n - \mathbf{o}_1) & \mathbf{z}_2 \times (\mathbf{o}_n - \mathbf{o}_2) & \mathbf{z}_3 \times (\mathbf{o}_n - \mathbf{o}_3) \end{bmatrix},$$

and:

$$J_\omega = \begin{bmatrix} \mathbf{0} & \mathbf{z}_1 & \mathbf{z}_2 & \mathbf{z}_3 \end{bmatrix}.$$





Graphical User Interface (GUI)

Provides an interactive interface for controlling the SCARA robot.

Key features include:

- Save positions and run predefined programs.
- Update, clear, and adjust speed and acceleration parameters.

Homing

- The robot resets to its base position before executing the task.

Arduino Uno

SCARA_Robot.ino

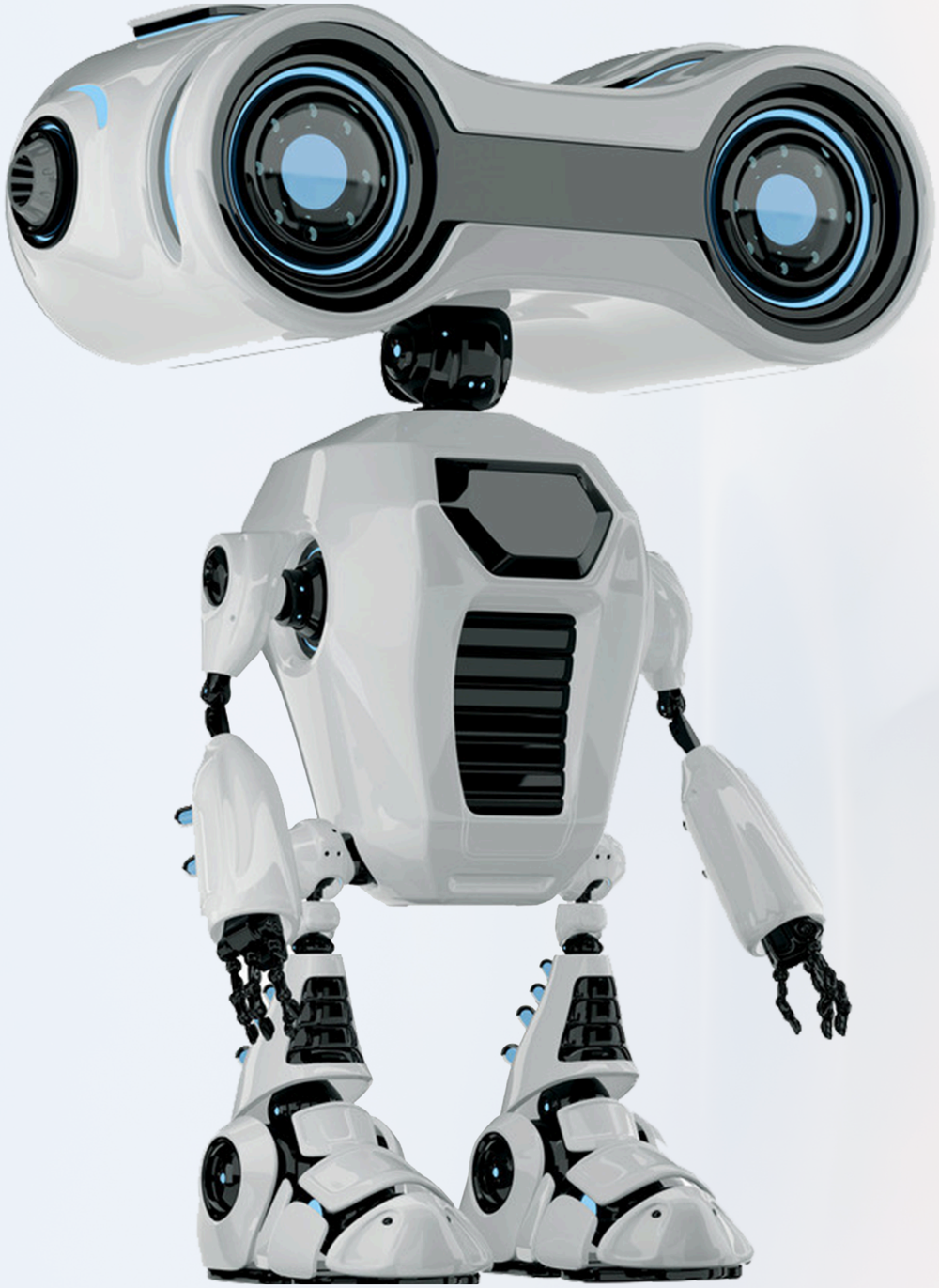
```
242
243  void homing() {
244    // Homing Stepper3
245    while (digitalRead(limitSwitch3) != 1) {
246      stepper3.setSpeed(-1100);
247      stepper3.runSpeed();
248      stepper3.setCurrentPosition(-1580); // When limit switch pressed set position to 0 steps
249    }
250    delay(20);
251
252    stepper3.moveTo(0);
253    while (stepper3.currentPosition() != 0) {
254      stepper3.run();
255    }
256
257    // Homing Stepper2
258    while (digitalRead(limitSwitch2) != 1) {
259      stepper2.setSpeed(1300);
260      stepper2.runSpeed();
261      stepper2.setCurrentPosition(4350); // When limit switch pressed set position to -5440 steps
262    }
263    delay(20);
264
265    stepper2.moveTo(0);
266    while (stepper2.currentPosition() != 0) {
267      stepper2.run();
268    }
269
```

Output

Ln 13, Col 32 Arduino Uno on COM41 [not connected]

Pick and place task

- Pick a box from a given coordinate.
- Place it in a designated square.
- Record joint angles and end-effector positions



Questions?

Thank You!

+94 70 3321835

maduwanthalhh.20@uom.lk

github/maduwanthasl