6-DOF SCARA Robot Simulation

Installation & Operating Instructions

Based on Ben-Gurion University Research

July 2, 2025

1 System Requirements

- Python 3.7+
- OS: Windows 10+, macOS 10.14+, Ubuntu 18.04+
- RAM: 4GB minimum
- Graphics: OpenGL 2.1 compatible

2 Installation

2.1 Install Python

Windows: Download from `https://python.org` (check "Add to PATH")
macOS: `brew install python`
Linux: `sudo apt install python3 python3-pip`

2.2 Install Dependencies

```
# Create virtual environment (recommended)
python -m venv scara_env

# Activate environment
# Windows: scara_env\Scripts\activate
# macOS/Linux: source scara_env/bin/activate

# Install required packages
pip install numpy matplotlib pillow
```

2.3 Setup Files

```
mkdir scara_simulation
cd scara_simulation
# Save the Python code as: scara_simulator.py
```

3 Operation

3.1 Run Simulation

```
python scara_simulator.py
```

Expected output:

```
SCARA Robot Simulator initialized
Base height: 800.0 mm
Planning trajectory...
Creating robot animation...
Animation saved as 'scara_robot_simulation.gif'
```

3.2 Simulation Features

- 6-DOF robot with joint limits
- Trajectory planning with trapezoidal velocity profile
- Real-time 3D visualization
- Workspace boundary display
- Animated GIF export

4 Customization

4.1 Modify Trajectory

Edit in `run_simulation()` method:

```
start_position = np.array([-150.0, 150.0, 1050.0])  # [x,y,z] mm
end_position = np.array([0.0, 400.0, 1200.0])       # [x,y,z] mm
```

4.2 Adjust Motion Parameters

Edit in `plan_trajectory()` method:

```
motion_time = 30.0              # Total time (seconds)
acceleration_time = 10.0        # Acceleration phase
deceleration_start_time = 20.0  # Deceleration start
```

5 Robot Configuration

Joint	Type	Range	Description
θ_1	Rotational	± 180	Base rotation
l_1	Prismatic	0-500 mm	Vertical extension
l_2	Prismatic	0-500 mm	Horizontal extension
θ_2	Rotational	± 180	Wrist pitch
θ_3	Rotational	± 135	Wrist roll
θ_4	Rotational	± 180	End effector

6 Forward Kinematics

End effector position equations:

$$x = -150 \sin\theta_1 - 150 \cos\theta_3 \sin\theta_1 - l_2 \sin\theta_1 - 150 \cos\theta_1 \cos\theta_2 \sin\theta_3 \tag{1}$$

$$y = 150 \cos\theta_1 + 150 \cos\theta_1 \cos\theta_3 + l_2 \cos\theta_1 - 150 \cos\theta_2 \sin\theta_1 \sin\theta_3 \tag{2}$$

$$z = l_1 + 150 \sin\theta_2 \sin\theta_3 + 800 \tag{3}$$

7 Troubleshooting

7.1 Common Issues

Import Error:

```
pip install numpy matplotlib pillow
```

No Animation Window:

- Ensure GUI environment available
- For SSH: use `ssh -X`

Slow Performance:

```
# Reduce animation frames
frame_indices = range(0, len(trajectory_data['joint_configs']), 4)

# Increase time step
time_step = 0.08  # Instead of 0.04
```

GIF Export Fails:

```
# Disable GIF export
self.create_robot_animation(trajectory, save_animation=False)
```

8 Verification

Test installation:

```
python -c "import numpy, matplotlib; print('Dependencies OK')"
python scara_simulator.py
```

Successful run should display 3D animation window and generate GIF file.