# 3-Link Robot Arm: Design Explained

## The Concept

This 3-link robot arm design represents a classic mechanical system combining stability, articulation, and precise object manipulation. The robot features a sturdy base, three identical links connected by rotational joints, and a specialized end effector for gripping objects.

## Key Components

### Base

- **Design**: Square base (200×200×50mm) for maximum stability
- **Evolution**: Initially designed as a circular platform but changed to square for better balance
- **Implementation**: Centered at origin with dark gray coloring for visual grounding

### Links

- **Design**: Three identical links (40×40×180mm) forming the arm's reach
- **Evolution**: Originally varied in length, standardized to simplify kinematics
- **Implementation**: Red coloring for high visibility during operation

### Joints

- **Design**: Four rotational joints with Y-axis rotation (25mm radius)
- **Evolution**: Increased from 20mm to 25mm radius after stress testing

- **Implementation**: Yellow cylinders marking each articulation point

## End Effector

- **Design**: Specialized gripper with central channel and sliding mechanisms
- **Evolution**: Most complex component, evolved from simple gripper to multi-feature design
- **Implementation**: Orange coloring with ergonomic cutout for improved handling

## Code Implementation Highlights

The robot is built programmatically using FreeCAD's Python API with these key approaches:

1. **Parametric Design**
   : All dimensions defined as variables for easy modification

   python

   ```
   BASE_WIDTH = 200
   BASE_LENGTH = 200
   BASE_HEIGHT = 50

   LINK_WIDTH = 40
   LINK_DEPTH = 40
   ```

```
LINK1_HEIGHT = 180
```

2. **Cumulative Positioning**
   : Each component positioned based on previous elements

   python

   ```
   joint2_pos = Base.Vector(0, 0, BASE_HEIGHT + JOINT_HEIGHT/2 +
   LINK1_HEIGHT + JOINT_HEIGHT/2 + LINK2_HEIGHT)
   ```

3. **Complex Geometry**
   : End effector created using boolean operations

   python

   ```
   ee_with_channel = ee_base.cut(channel)
   ee_shape = ee_temp.cut(cutout_cyl)
   ```

4. **Component Organization**
   : Logical grouping for easier manipulation

   python

   ```
   robot_group = doc.addObject("App::DocumentObjectGroup", "Robot")
   robot_group.addObject(base_obj)
   robot_group.addObject(joint0)
   ```

# Design Iterations

## Base Evolution

1. Circular → Square design for stability
2. Height increased from 30mm to 50mm after balance testing

## Link System Refinement

1. Variable lengths → Uniform 180mm for manufacturing simplicity
2. Cross-section optimized to 40×40mm for strength-to-weight ratio

## Joint Improvements

1. Z-axis → Y-axis rotation for better movement patterns
2. Radius increased for improved structural integrity

## End Effector Development

1. Simple block → Channel design → Rail system → Dual grippers
2. Added ergonomic cutout for improved object handling

## Technical Challenges Solved

1. **Weight Distribution**: Base dimensions increased to improve stability
2. **Positioning Precision**: Implemented cumulative height calculations
3. **Complex Geometry**
   : Used boolean operations with error handling

```
python

try:
ee_shape = ee_temp.cut(cutout_cyl)
except:
ee_shape = ee_base  # Fallback option
```

# Future Directions

The design provides a foundation for enhancements including:

- Dynamic joint control for posing
- Inverse kinematics for precise positioning
- Material properties for physical simulation
- Animation capabilities for movement demonstration

This 3-link robot arm successfully balances form and function through thoughtful design choices and efficient implementation, creating a versatile platform suitable for educational purposes and practical applications.