Rupesh Kanna Kaviyasree Narayanan
Akash Patel

DH Parameters and Optimization

Abstract

In this project, we will be using joints, links and MATLAB, AppDesigner, and Robotics Toolbox to learn and implement the design, assembly, and optimization of a two-link robotic arm. The goal is to create a model based on the DH parameters, implement forward kinematics, and optimize the model to reduce error of the DH parameters. We built a user-friendly graphical interface (GUI) that enables joint control and end-effector movement visualization. Optimization of the DH parameters was done using fminsearch to minimize the difference between the simulated and real-world measurements.

Introduction

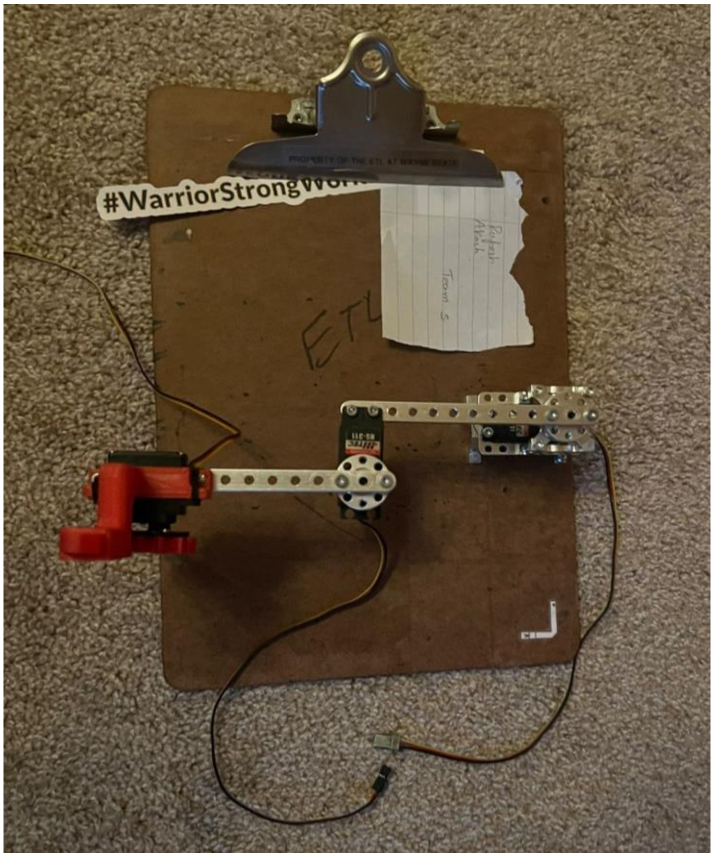
Robot arms are vital tools in various fields such as manufacturing, healthcare, and research due to their precision and performing repetitive tasks. This project involves building a two-link robot arm, determining its DH parameters, and developing a forward kinematics model. The DH parameters allow us to describe the geometry of the robot using link lengths, twist angles, and joint angles, which is crucial for simulating the robot's movement.

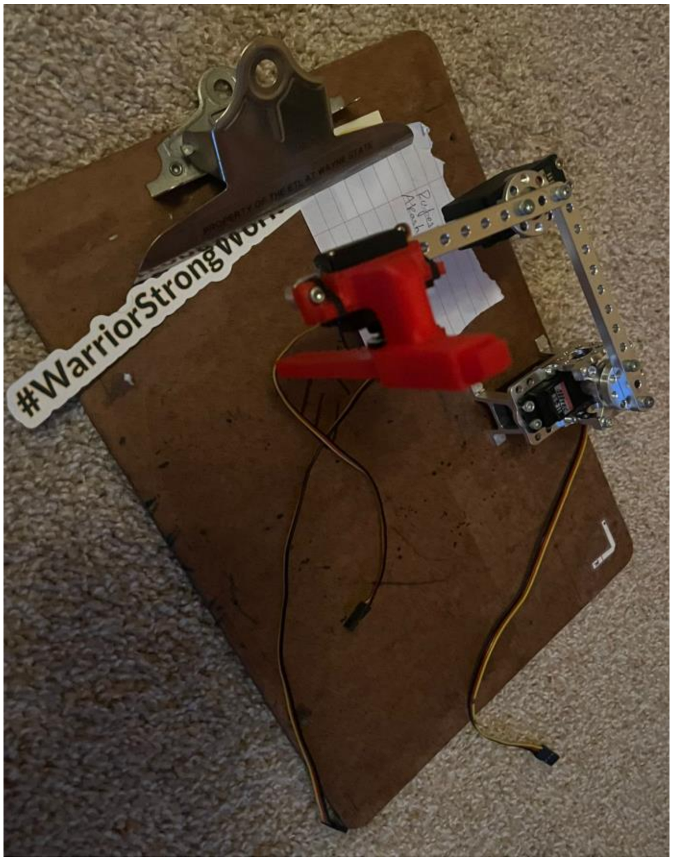
Forward kinematics helps predict the position and orientation of the end-effector based on the joint angles. This mathematical model is key to controlling the arm and ensuring it behaves as expected in real-world applications. The goal is to compare the performance of the physical robot to the model and optimize the DH parameters to improve accuracy. The development of GUI allows for user interaction with the robot's virtual model.

Coordinate System Setup

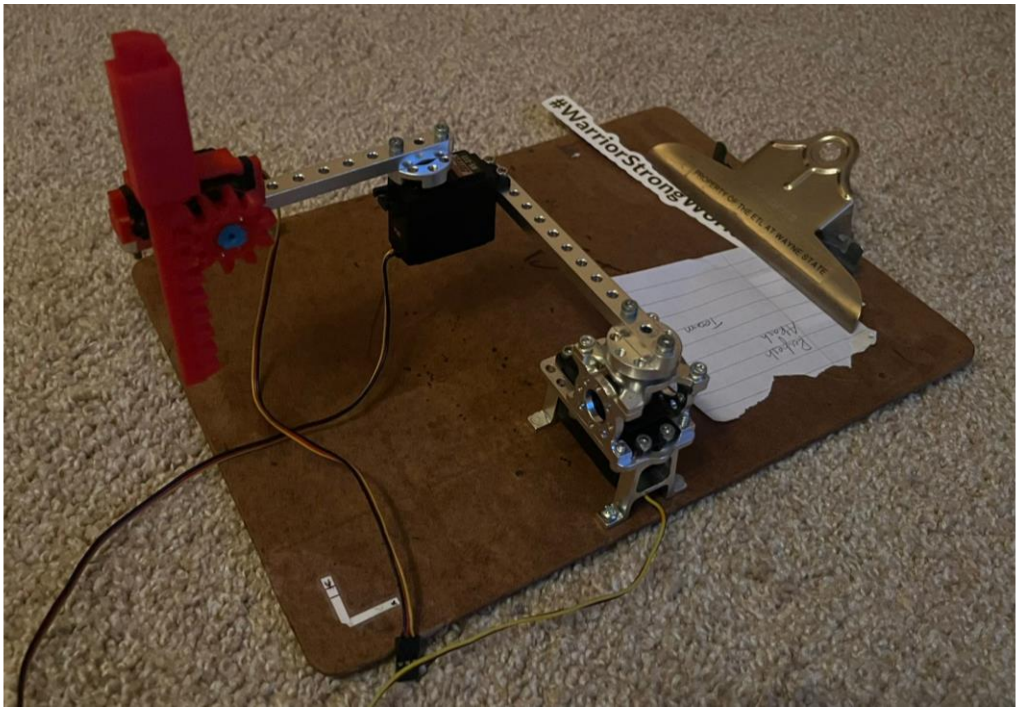
Coordinate systems were placed at both the end-effector and the left bottom corner of the clipboard. We used toothpicks to represent the axes at each joint in the system. The goal was to simplify measurements and ensure that the graphical model in MATLAB would match the physical robot's setup.

Rupesh Kanna Kaviyasree Narayanan
Akash Patel





Rupesh Kanna Kaviyasree Narayanan
Akash Patel



DH Parameter Measurement

The DH parameters for the robot were measured manually based on the physical robot dimensions. These parameters define the position and orientation of each link and were used to build the forward kinematics model in the MATLAB Robotics Toolbox. The DH parameters include:

a: the distance between consecutive joint axes.
d: the distance along the z-axis between consecutive link frames.
Theta: the rotation of a link around the z-axis.

The GUI

The robot's graphical model was built using Peter Corke's Robotics Toolbox in MATLAB. The SerialLink class was used to define the robot's links based on the DH parameters measured earlier. Forward kinematics was computed using the fkine function, which provides the transformation matrix describing the end-effector's position and orientation.

Example code for calculating the end-effector's position:

```
  q = test(1:2)';  % Joint angles;

  T =fkine(robot1,q);

 % plot(robot1,q)

  pos = T.t
```

Rupesh Kanna Kaviyasree Narayanan
Akash Patel

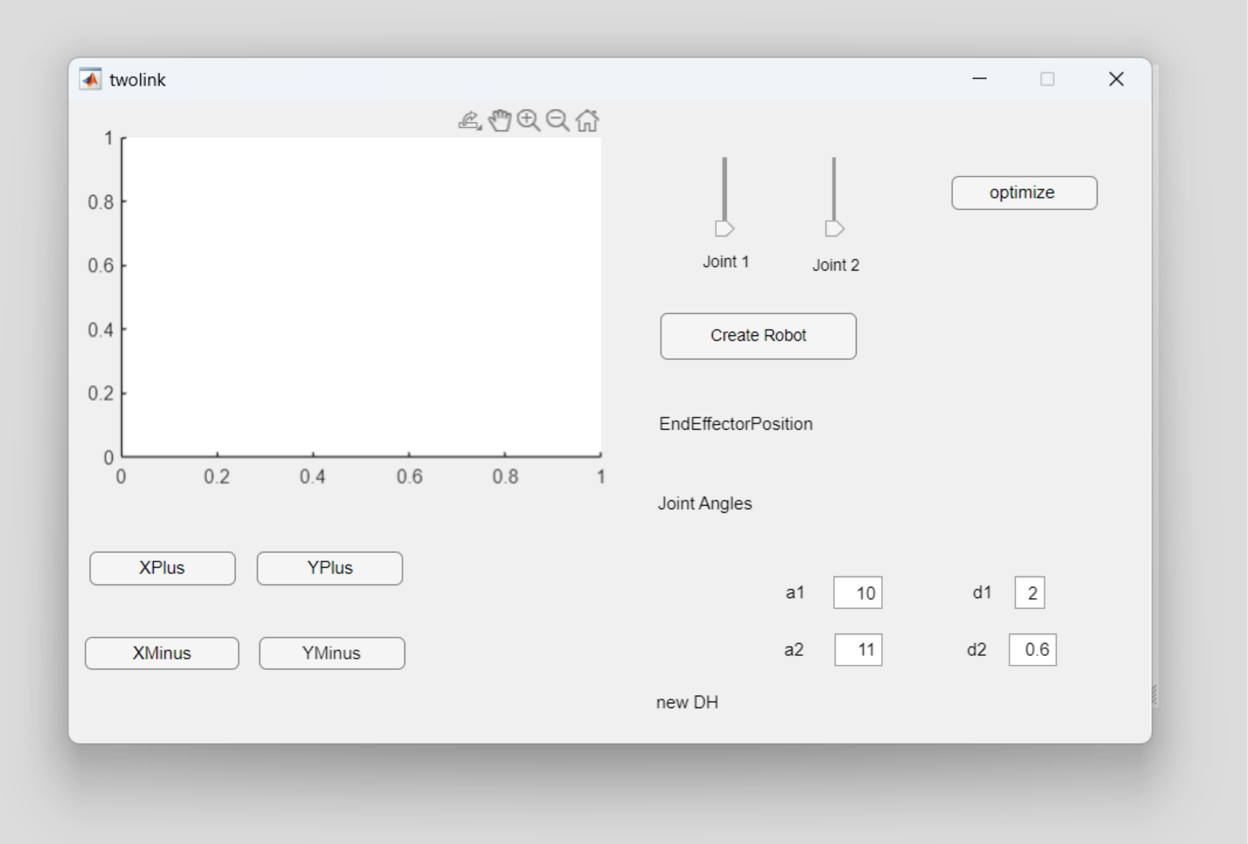
AppDesigner was used to develop a graphical user interface (GUI) that allowed users to control the robot's joints interactively. The GUI included sliders for each joint angle, buttons for incrementing the end-effector’s position, and real-time display of the end-effector's coordinates.

Here’s an example of the code used to define the two links in the robot:

```
L(1) = Link ('d',para(1), 'a', para(3), 'alpha', 0,'offset',deg2rad(18.435));

L(2) = Link ('d', para(2), 'a',para(4), 'alpha', 0,'offset',deg2rad(19));

robot1 = SerialLink(L);
```

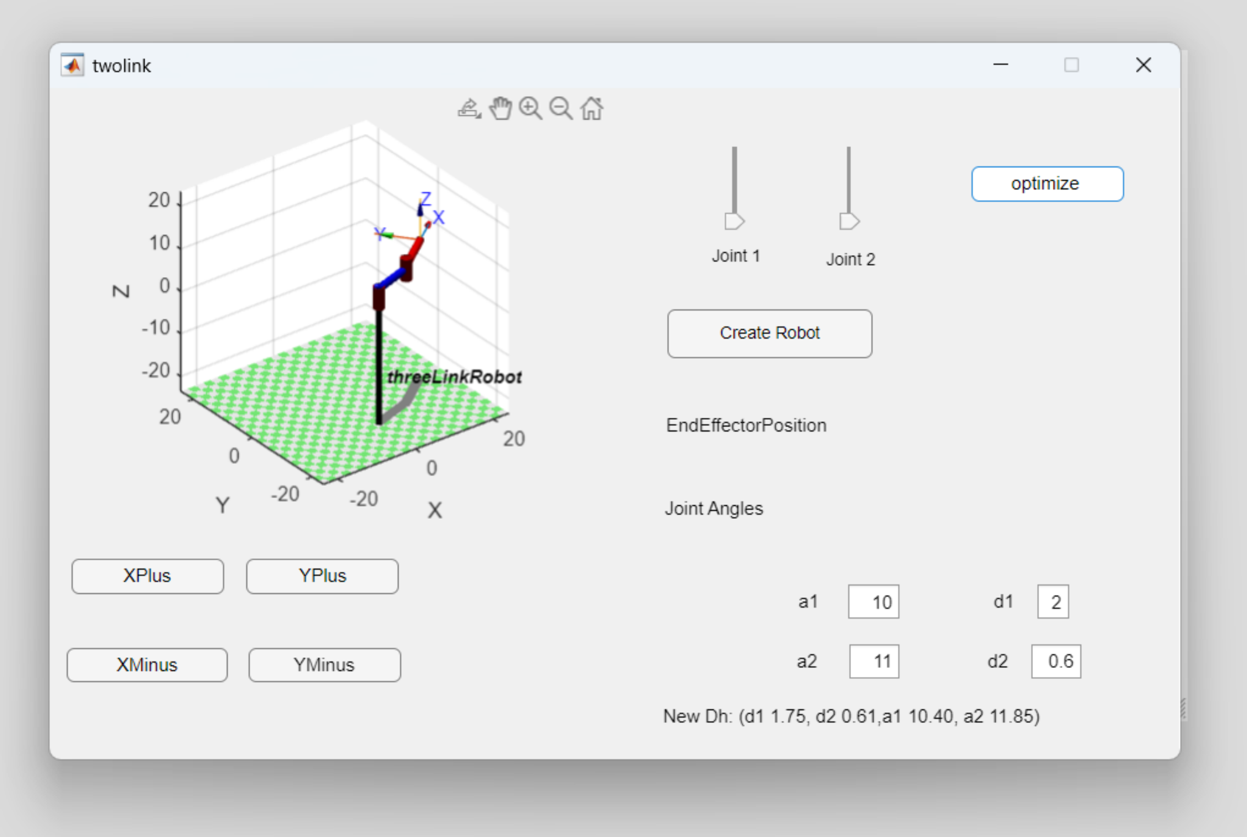


a1 10

d1 2

a2 11

d2 0.6



a1 10 d1 2

a2 11 d2 0.6

New Dh: (d1 1.75, d2 0.61,a1 10.40, a2 11.85)

Optimization

We used MATLAB's fminsearch function to minimize this error. The optimization process adjusts the DH parameters to minimize the difference between the measured and simulated end-effector positions by iterative methods.

The error is calculated as:

Z_add = para(1)+para(2);

Real = sqrt((test(3)-0).^2 + (test(4)-0).^2);

Model = sqrt((pos(1)-0).^2 + (pos(2)-0).^2);

each_err = sqrt(sum((Real-Model).^2));

err=err + each_err

Here's how the optimization was done:

thea1 =19.29;

thea2=16.7;

para = [a1 a2 d1 d2];

NEwfunction(para)

Discussion and Conclusions

The project successfully demonstrated how forward kinematics can be used to model a robotic arm, and how optimization techniques can improve the accuracy of this model. This project highlights the practical applications of robot arms in automation, where accurate positioning and movement control are critical. Similar techniques could be used in fields like manufacturing, where precise control of robotic manipulators is required, or in healthcare, where robotic arms assist in surgeries.

Appendix

The Code

```
classdef threeLinkGui_final < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        figure1                   matlab.ui.Figure
        JointAnglesLabel          matlab.ui.control.Label
        newDHLabel                matlab.ui.control.Label
        optimizeButton            matlab.ui.control.StateButton
        d2EditField               matlab.ui.control.NumericEditField
        d2EditFieldLabel          matlab.ui.control.Label
        d1EditField               matlab.ui.control.NumericEditField
        d1EditFieldLabel          matlab.ui.control.Label
        a2EditField               matlab.ui.control.NumericEditField
        a2EditFieldLabel          matlab.ui.control.Label
        a1EditField               matlab.ui.control.NumericEditField
        a1EditFieldLabel          matlab.ui.control.Label
        EndEffectorPositionLabel  matlab.ui.control.Label
        YMinusButton              matlab.ui.control.StateButton
        YPlusButton               matlab.ui.control.StateButton
        XMinusButton              matlab.ui.control.StateButton
        XPlusButton               matlab.ui.control.StateButton
        text3                     matlab.ui.control.Label
        text2                     matlab.ui.control.Label
        Create                    matlab.ui.control.Button
        slider2                   matlab.ui.control.Slider
        slider1                   matlab.ui.control.Slider
        axes1                     matlab.ui.control.UIAxes
    end


    properties (Access = private)
        robot =[]; % This is a Serial Link Created robot
        jointangles= [0 0]; % these are the joint angles of the robot
        endEffectorPos = [0 0];% these are the positons of the robot
        increment = .5

    end

    methods (Access = public)

           function updateRobotPlot(app)


           app.robot.plot(app.jointangles);
            T = app.robot.fkine(app.jointangles);
             pos = T.t';
             app.endEffectorPos=pos;
            updateEndEffectorPositionLabel(app)
           end

        function updateEndEffectorPositionLabel(app)
           app.robot.plot(app.jointangles);
            app.EndEffectorPositionLabel.Text = sprintf('End-Effector Position:
(%.2f, %.2f,)', ...
                app.endEffectorPos(1), app.endEffectorPos(2));
              app.JointAnglesLabel.Text = sprintf('Joint Angles: (%.2f, %.2f,)', ...
                rad2deg(app.jointangles(1)), rad2deg(app.jointangles(2)));
        end



        function err = func3(app,test)
        err= 0;
        for i=1:5
            q = test(1:2)';  % Joint angles;
            %Forward kinematics
            T =fkine(app.robot,q);
            pos = T.t;       % model  coordinates
            d = test(3:4);    % Measured coordinates
            % distance of real measurements
            Real = sqrt((test(3)-0).^2 + (test(4)-0).^2);

            %distance of model measurements
            Model = sqrt((pos(1)-0).^2 + (pos(2)-0).^2);
            % find the error of real measurements and model measurements
            each_err = sqrt(sum((Real-Model).^2));
            err=err + each_err;

           % app.jointangles = q';
           %  updateEndEffectorPositionLabel(app)
        end
        %app.robot.plot(q)
        err = err/5
        end
    end


    % Callbacks that handle component events
```

Rupesh Kanna Kaviyasree Narayanan
Akash Patel

```
    methods (Access = private)

        % Callback function: Create, a1EditField, a2EditField, d1EditField,
        %
        % ...and 1 other component
        function Create_Callback(app, event)
            %create a simple two link robot.
L(1) = Link ('d',app.d1EditField.Value, 'a',app.a1EditField.Value, 'alpha', 
0,'offset',deg2rad(18.435));
 L(2) = Link ('d', app.d2EditField.Value, 'a',app.a2EditField.Value, 'alpha', 
0,'offset',deg2rad(19));
            app.robot = SerialLink(L, 'name', 'threeLinkRobot');
            app.robot.base = [0 -11.5 6.5]; % base offset y =-11.5 and z= 6.5 cm
            app.robot.plot (app.jointangles); %plotting initial joint angles
        end

        % Value changed function: slider1
        function slider1_Callback(app, event)
            %get the values of the sliders from each joint
            app.jointangles(1) = app.slider1.Value;
            app.jointangles(2) = app.slider2.Value;

           app.updateRobotPlot();

        end

        % Value changed function: slider2
        function slider2_Callback(app, event)
            app.jointangles(1) = app.slider1.Value;
            app.jointangles(2) = app.slider2.Value;

           app.updateRobotPlot();

        end

        % Value changed function: XMinusButton, XPlusButton, YMinusButton,
        % ...and 1 other component
        function Button(app, event)

                if app.XPlusButton.Value
                    app.endEffectorPos(1) = app.endEffectorPos(1) + app.increment;
                    app.XPlusButton.Value=false; % returns the button to zero state

            elseif app.XMinusButton.Value
                    app.endEffectorPos(1) = app.endEffectorPos(1) - app.increment;
                    app.XMinusButton.Value=false; % returns the button to zero state


            elseif app.YPlusButton.Value
                    app.endEffectorPos(2) = app.endEffectorPos(2) + app.increment;
                    app.YPlusButton.Value=false; % returns the button to zero state

            elseif app.YMinusButton.Value
                    app.endEffectorPos(2) = app.endEffectorPos(2) - app.increment;
                    app.YMinusButton.Value=false; % returns the button to zero state
                end

            % U take endEffectorPos and put into a 4x4 matrix
            U = [-0 -1 0 app.endEffectorPos(1);1 -0 0 app.endEffectorPos(2);0 1 1 0;0
0 0 1];
           %  inverse kinematics
            G = app.robot.ikunc(U); % takes endEffectorPos to and find jointangles

            app.jointangles = G; % places G into   app.jointangles
             updateEndEffectorPositionLabel(app) % send to funtion




        end

        % Value changed function: optimizeButton
        function optimizeButtonValueChanged(app, event)
          if app.optimizeButton.Value

       C = [app.d1EditField.Value app.d2EditField.Value app.a1EditField.Value
app.a2EditField.Value]; % puts old DH parameters in array

    %  fid = fopen('dataRad.txt', 'r'); % open files
    fid = fopen('dataRad.txt', 'r'); % open files
      s = fgets(fid); % get Data
      test = sscanf(s, '%f %f %f %f\n'); % puts data into a matrix

            % Call fminsearch with function handle
            output = fminsearch(@(test) app.func3(test), C);
            % prints out outpus in to gui
            app.newDHLabel.Text = sprintf('New Dh: (d1 %.2f, d2 %.2f,a1 %.2f, a2
%.2f)', output(1),output(2), output(3), output(4) );
             app.optimizeButton.Value=false; % returns the button to zero state
          end

        end
    end

    % Component initialization
    methods (Access = private)

        % Create UIFigure and components
        function createComponents(app)

            % Create figure1 and hide until all components are created
            app.figure1 = uifigure('Visible', 'off');
            app.figure1.Position = [582 397 741 439];
            app.figure1.Name = 'twolink';
            app.figure1.Resize = 'off';
            app.figure1.HandleVisibility = 'callback';
            app.figure1.Tag = 'figure1';

            % Create axes1
            app.axes1 = uiaxes(app.figure1);
            app.axes1.FontSize = 13;
            app.axes1.NextPlot = 'replace';
            app.axes1.Tag = 'axes1';
            app.axes1.Position = [13 176 363 248];

            % Create slider1
            app.slider1 = uislider(app.figure1);
            app.slider1.Limits = [0 3.14159265358979];
            app.slider1.MajorTicks = [];
            app.slider1.Orientation = 'vertical';
            app.slider1.ValueChangedFcn = createCallbackFcn(app, @slider1_Callback,
true);
            app.slider1.MinorTicks = [];
            app.slider1.Tag = 'slider1';
            app.slider1.FontSize = 11;
            app.slider1.Position = [448 353 3 50];

            % Create slider2
            app.slider2 = uislider(app.figure1);
            app.slider2.Limits = [0 3.14159265358979];
            app.slider2.MajorTicks = [];
            app.slider2.Orientation = 'vertical';
            app.slider2.ValueChangedFcn = createCallbackFcn(app, @slider2_Callback,
true);
            app.slider2.MinorTicks = [];
            app.slider2.Tag = 'slider2';
            app.slider2.FontSize = 11;
            app.slider2.Position = [523 353 3 50];

            % Create Create
            app.Create = uibutton(app.figure1, 'push');
            app.Create.ButtonPushedFcn = createCallbackFcn(app, @Create_Callback,
true);
            app.Create.Tag = 'Create';
            app.Create.FontSize = 11;
            app.Create.Position = [406 264 134 32];
            app.Create.Text = 'Create Robot';

            % Create text2
            app.text2 = uilabel(app.figure1);
            app.text2.Tag = 'text2';
            app.text2.HorizontalAlignment = 'center';
            app.text2.VerticalAlignment = 'top';
            app.text2.WordWrap = 'on';
            app.text2.FontSize = 11;
            app.text2.Position = [422 313 58 24];
            app.text2.Text = 'Joint 1';

            % Create text3
            app.text3 = uilabel(app.figure1);
            app.text3.Tag = 'text3';
            app.text3.HorizontalAlignment = 'center';
            app.text3.VerticalAlignment = 'top';
            app.text3.WordWrap = 'on';
            app.text3.FontSize = 11;
            app.text3.Position = [497 316 58 19];
            app.text3.Text = 'Joint 2';

            % Create XPlusButton
            app.XPlusButton = uibutton(app.figure1, 'state');
            app.XPlusButton.ValueChangedFcn = createCallbackFcn(app, @Button, true);
            app.XPlusButton.Text = 'XPlus';
            app.XPlusButton.Position = [15 110 100 23];

            % Create XMinusButton
            app.XMinusButton = uibutton(app.figure1, 'state');
            app.XMinusButton.ValueChangedFcn = createCallbackFcn(app, @Button, true);
            app.XMinusButton.Text = 'XMinus';
            app.XMinusButton.Position = [12 52 106 23];

            % Create YPlusButton
            app.YPlusButton = uibutton(app.figure1, 'state');
            app.YPlusButton.ValueChangedFcn = createCallbackFcn(app, @Button, true);
            app.YPlusButton.Text = 'YPlus';
            app.YPlusButton.Position = [130 110 100 23];

            % Create YMinusButton
            app.YMinusButton = uibutton(app.figure1, 'state');
            app.YMinusButton.ValueChangedFcn = createCallbackFcn(app, @Button, true);
            app.YMinusButton.Text = 'YMinus';
            app.YMinusButton.Position = [131 52 100 23];

            % Create EndEffectorPositionLabel
            app.EndEffectorPositionLabel = uilabel(app.figure1);
            app.EndEffectorPositionLabel.Position = [405 209 303 23];
            app.EndEffectorPositionLabel.Text = 'EndEffectorPosition';

            % Create a1EditFieldLabel
            app.a1EditFieldLabel = uilabel(app.figure1);
            app.a1EditFieldLabel.HorizontalAlignment = 'right';
            app.a1EditFieldLabel.Position = [480 94 25 22];
            app.a1EditFieldLabel.Text = 'a1';

            % Create a1EditField
            app.a1EditField = uieditfield(app.figure1, 'numeric');
            app.a1EditField.ValueChangedFcn = createCallbackFcn(app,
@Create_Callback, true);
            app.a1EditField.Position = [524 94 34 22];
            app.a1EditField.Value = 10;

            % Create a2EditFieldLabel
            app.a2EditFieldLabel = uilabel(app.figure1);
            app.a2EditFieldLabel.HorizontalAlignment = 'right';
            app.a2EditFieldLabel.Position = [479 55 25 22];
            app.a2EditFieldLabel.Text = 'a2';

            % Create a2EditField
            app.a2EditField = uieditfield(app.figure1, 'numeric');
            app.a2EditField.ValueChangedFcn = createCallbackFcn(app, 
@Create_Callback, true);
            app.a2EditField.Position = [525 55 33 22];
            app.a2EditField.Value = 11;

            % Create d1EditFieldLabel
            app.d1EditFieldLabel = uilabel(app.figure1);
            app.d1EditFieldLabel.HorizontalAlignment = 'right';
            app.d1EditFieldLabel.Position = [608 94 25 22];
            app.d1EditFieldLabel.Text = 'd1';

            % Create d1EditField
            app.d1EditField = uieditfield(app.figure1, 'numeric');
            app.d1EditField.ValueChangedFcn = createCallbackFcn(app, 
@Create_Callback, true);
            app.d1EditField.Position = [648 94 21 22];
            app.d1EditField.Value = 2;

            % Create d2EditFieldLabel
            app.d2EditFieldLabel = uilabel(app.figure1);
            app.d2EditFieldLabel.HorizontalAlignment = 'right';
            app.d2EditFieldLabel.Position = [604 55 25 22];
            app.d2EditFieldLabel.Text = 'd2';

            % Create d2EditField
            app.d2EditField = uieditfield(app.figure1, 'numeric');
            app.d2EditField.ValueChangedFcn = createCallbackFcn(app, 
@Create_Callback, true);
            app.d2EditField.Position = [644 55 33 22];
            app.d2EditField.Value = 0.6;

            % Create optimizeButton
            app.optimizeButton = uibutton(app.figure1, 'state');
            app.optimizeButton.ValueChangedFcn = createCallbackFcn(app, 
@optimizeButtonValueChanged, true);
            app.optimizeButton.Text = 'optimize ';
            app.optimizeButton.Position = [605 367 100 23];

            % Create newDHLabel
            app.newDHLabel = uilabel(app.figure1);
            app.newDHLabel.Position = [403 19 295 22];
            app.newDHLabel.Text = 'new DH';

            % Create JointAnglesLabel
            app.JointAnglesLabel = uilabel(app.figure1);
            app.JointAnglesLabel.Position = [404 155 273 22];
            app.JointAnglesLabel.Text = 'Joint Angles';

            % Show the figure after all components are created
            app.figure1.Visible = 'on';
        end
    end

    % App creation and deletion
    methods (Access = public)
```

Rupesh Kanna Kaviyasree Narayanan
Akash Patel

```
        % Construct app
        function app = threeLinkGui_final

            runningApp = getRunningApp(app);

            % Check for running singleton app
            if isempty(runningApp)

                % Create UIFigure and components
                createComponents(app)

                % Register the app with App Designer
                registerApp(app, app.figure1)
            else

                % Focus the running singleton app
                figure(runningApp.figure1)

                app = runningApp;
            end

            if nargout == 0
                clear app
            end
        end

        % Code that executes before app deletion
        function delete(app)

            % Delete UIFigure when app is deleted
            delete(app.figure1)
        end
    end
end
```