International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 11, Issue 8 [August. 2022] PP: 141-155

Evaluation of Actual Configuration Angles of A Four Bar Linkage For Precision Applications: Simulink Approach.

1. ENGR. D.C UGWUEGBU
DEPARTMENT OF MECHANICAL ENGINEERING, FEDERAL POLYTECHNIC NEKEDE, OWERRI.
2. OGUNKUNLE DELE KEHINDE
DEPARTMENT OF MECHANICAL ENGINEERING, FEDERAL POLYTECHNIC NEKEDE, OWERRI.
&
3. MR. EWURUM TENNISON. I
DEPARTMENT OF MECHANICAL ENGINEERING, FEDERAL POLYTECHNIC NEKEDE, OWERRI.

ABSTRACT

The study, evaluation of actual configuration angles of a four bar linkage for precision applications through simulink approach, was successfully carried out. Block models; 3 binary links, 4 revolute joints, 2 pivot mount, 2 rigid transforms, 1 world frame, 1 solver (ODE 3) configuration, 1 mechanism configuration and block ports, base with follower frame were used to build the simulink four bar linkage model with specified block parameters. Simulation of the four bar linkage model was allowed to run for 20 seconds. Position angles specified before model simulation were +30, +30, - 23 and 0 degrees for the corresponding revolute joints. Simulation results showed that the actual position angles of the four bar links were evaluated to be – 92.4196, -223.375, 0 and + 101.66 degrees, corresponding to **Revolute Joint, Revolute Joint 1, Revolute Joint2 and Revolute Joint3** respectively. However, the specified and actual velocities of links were found to be the same values being +15 rad/s, +25rad/s, 0 and +12rad/s respectively after the simulink model simulation, at high priorities. The regression coefficient of 1 indicated that there is a close and positive relationship between position angles specified and actual position angles of revolute joints. Actual position angles and specified position angles data were trained using levenberg marquardt algorithm at 70% training data, 15% test data and 15% validation data respectively. Results revealed that the best performance level was at 0 and at epoch 0 and this suggested lowest values of specified position angles. The researchers made the following recommendations: Four bar linkage must be configured at their actual position angles to improve mechanism task precision, Revolute joints must be adequately lubricated to encourage free rotation motion, etc.

Keywords ---- MATLAB, algorithm, training data, simulink, actual angle, four bar linkage model, specified angles.

Date of Submission: 18-08-2022 Date of Acceptance: 02-09-2022

1. INTRODUCTION

Background of the Study

Determination of actual configuration angles of a mechanical four-bar-linkage has been a problem in the design and synthesis of machine mechanisms. Four bar linkage is the foundation of machine design and creation and capable of producing rotational, translational and oscillation motions through the movement of crank. A linkage is called a mechanism if two or more links are movable with respect to a fixed link (Gupta, 2012).

Plecnik and McCarthy (as cited in Ugwuegbu and Ewurum, 2022) stated that, four-bar-linkage synthesis is to determine link dimensions of the linkage that achieves prescribed task positions. Traditionally, linkage synthesis is divided into three types, motion generation, function generation, and path generation. Each of the synthesis types has been extensively researched and has many applications in various engineering fields, but, in many situations, configuration angle of the links receives little attention of the researchers. Four-bar- linkages are usually designed to take an input and produce a different output, altering motion, velocity, acceleration, and applying mechanical advantage.

The analysis of four-bar-linkage mechanisms, involving aspects related to geometrical configuration angles, lead to the geometrical conditions for the lengths of a four-bar linkage, and establishes the roles played by each link. Hence, the study aims to evaluate actual configuration angles of a four bar linkage for precision applications through simulink approach.

Statement of the problem

Assembling a four-bar-linkage at configuration angles less than or greater than the actual angles, can cause premature failure of linkages, machine noise and the links may be unable to perform the required task. The design and synthesis of four-bar-linkage has been extensively researched and has many applications in various engineering fields, but, in many situations, configuration angle of the links receives little attention of the researchers.

.McCarthy and Soh (2010) stated that four-bar-linkage is a series of four rigid links connected with joints (revolute joints) to form a closed chain. Each link has two joints, and the joints have various degrees of freedom to allow relative motion. Four-bar-linkage play a key role in industrial machines, robotics, conveyors, diesel engines, automatic machines and therefore, demands accurate configuration angles if the machines must perform satisfactorily. A four-bar linkage mechanism, formed by four links connected in a closed chain with revolute joints, each obtained by grounding a different link, provides a great number of configuration angle problems that will affect machine precision. It is on this note that the paper aims at evaluating actual configuration angles of a four bar linkage for precision applications through simulink approach.

Objective of the Study

The aim of this study is to evaluate the actual configuration angles of a four-bar-linkage for precision applications through simulink approach. Specifically, the study would:

1. Use specified configuration angle of links to determine actual angle of links.
2. Create a simulink model involving degree of freedom.
3. Create a simulink model involving configuration angles.

Significance of the Study

The findings would enable machine / mechanism manufacturers to design and build four-bar-linkage with no configuration angle error.

Scope of the Study

This research would focus on evaluating the actual configuration angles of a four bar linkage through simulink approach. So, all efforts would be directed towards the general objective. The results were gotten from simulink in matlab and mechanics explorer, conducted in Federal Polytechnic Nekede, within South East of Nigeria. Results may be subject to variations with other software.

2. Literature Review

Gupta (2012) studied theory of machines and simple mechanisms and he concluded that linkages are the foundation of machine design and creation and capable of producing rotational, translational and oscillation motions through the movement of crank. A linkage is called a mechanism if two or more links are movable with respect to a fixed link.

McCarthy and Soh (2010) examined geometric design of linkages and discovered that four-bar-linkage is a series of four rigid links connected with joints (revolt joints) to form a closed chain. Each link has two joints, and the joints have various degrees of freedom to allow relative motion. They also added that Four-bar-linkage play a key role in industrial machines, robotics, conveyors, diesel engines, and therefore, demands accurate configuration angles if the machines must perform satisfactorily. Plecnik and McCarthy (as cited in Ugwuegbu and Ewurum, 2022) studied numerical synthesis of six bar linkage for mechanical computation and they opined that, four-bar-linkage synthesis is to determine link dimensions of the linkage that achieves prescribed task positions. Traditionally, linkage synthesis is divided into three types, motion generation, function generation, and path generation. Jain and Gupta (2011) experimented four bar linkage and they concluded that it is made up of four links, three movable links, four joints, one fixed link and one constraint.`

3. Methodology

Evaluation of actual configuration angles of a four-bar-linkage was carried out using simulink in matlab. Simulink in the matlab command window contains block models that were used to represent all the elements of four-bar-linkage. The block models gotten from simscap-mechanics, second generation includes: 3 binary links, 4 revolute joints, 2 pivot mount, 2 rigid transforms, 1 world frame, 1 solver configuration and 1 mechanism configuration. Block ports (base and follower frame) were connected as shown in **fig 1.0** below. The block parameters as specified below were also entered in the individual block model. ODE3 solver was chosen due to its tight tolerance in fixed step solving. Simulation was allowed to run for 20seconds.

4. Results and Presentations

Block Parameters

1. **Binary link A parameters:** length = 10cm, width = 2cm, thickness = 0.8cm, peg radius = 0.4cm, mass density = 2700kg/m3, link color = [0.25 0.4 0.7], peg color = [1 0.6 0.25].
2. **Binary link A1 parameters:** length = 35cm, width = 2cm, thickness = 0.8cm, peg radius = 0.4cm, mass density = 2700kg/m3, link color = [0.25 0.4 0.7], peg color = [1 0.6 0.25].
3. **Binary link B parameters:** length = 25cm, width = 2cm, thickness = 0.8cm, peg hole radius = 0.4cm, mass density = 2700kg/m3, link color = [0.25 0.4 0.7].
4. **Revolute Joint position specified:** +30 degrees and velocity = 15 rad/s
5. **Revolute Joint1 position specified:** +30 degrees and velocity = 25 rad/s
6. **Revolute Joint3 position specified:** 0 degrees and velocity = 12 rad/s
7. **Revolute Joint2 position specified:** -23 degrees and velocity = 0 rad/s
8. **Rigid transform 1 and 2 :** method = standard axis, axis = +Y , offset = 15cm.
9. **Pivot Mount:** hexagonal outer radius = 0.4cm, hexagon thickness = 0.8cm, peg radius = 0.4cm, mass density = 2700kg/m3, link color = [0.25 0.4 0.7], peg color =[[1 0.6 0.25].
10. **Mechanism Configuration:** linearization delta = 0.001, gravity = -9.8066m/s2.
11. **Solver Configuration:** ODE3, tolerance = 1e09, sample time = 0.001, nonlinear iterations = 3, iteration mode = 2

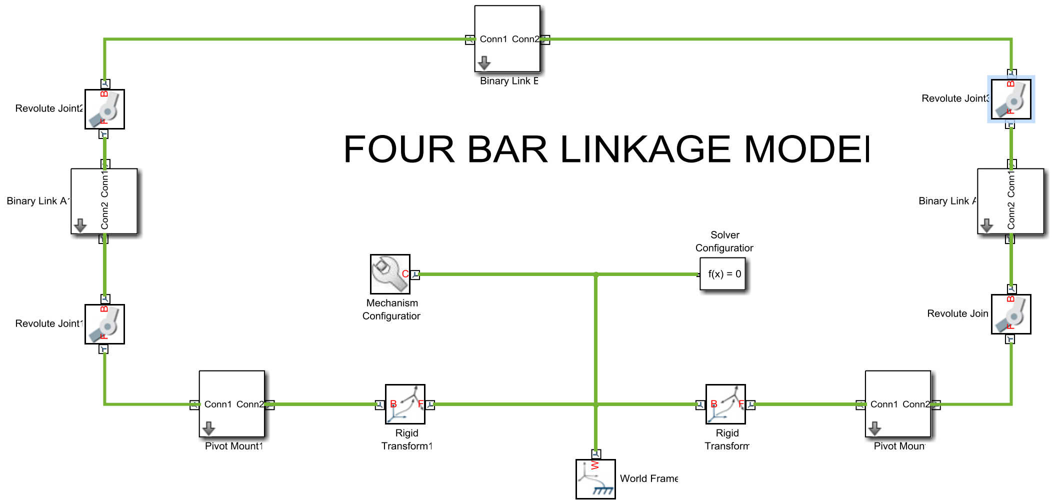


Fig 1.0: Simulink model of four-bar-linkage

Table1.0: simulation result for position angles and velocity of links.

S/N	JOINTS	Position angle specified(degrees)	Position actual angle	Velocity specified(rad/s)	Actual velocity	Priorities
1	Revolute joint	+30	-92.4196	+15	+15	High
2	Revolute joint 1	+30	-223.375	+25	+25	High
3	Revolute joint 2	-23	0	0	0	Low
4	Revolute joint 3	0	+101.66	+12	+12	High

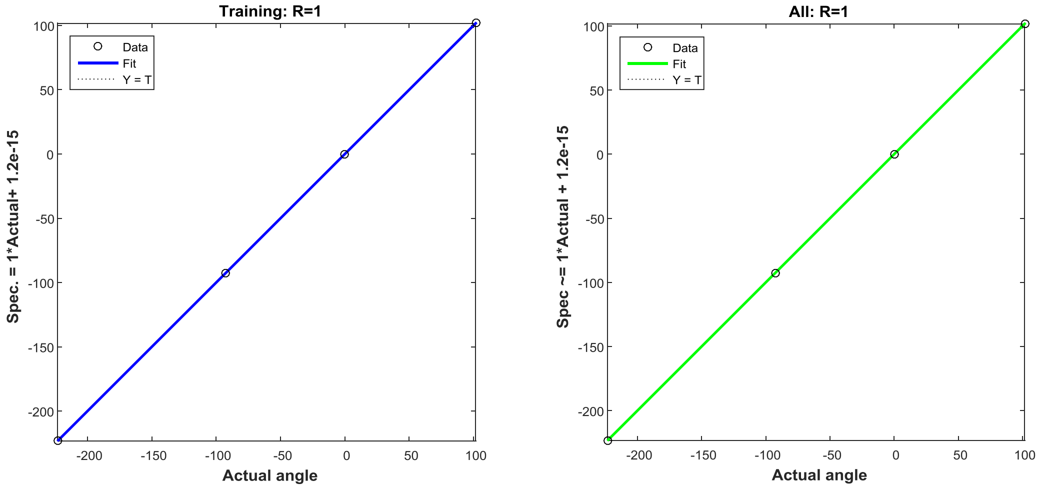


Fig 1.0.1: Regression graph of specified angle against actual angle of links.

1.8 MATLAB codes for the graph above

```
%CREATEFIGURE1(X1, YMATRIX1, X2)
%  X1:  vecfunction createfigure1(X1, YMatrix1, X2)tor of x data
%  YMATRIX1:  matrix of y data
%  X2:  vector of x data

% Create figure
figure1 = figure('Tag','PLOTREGRESSION1','NumberTitle','off',...
    'Name','Regression (plotregression)');

% Create subplot
subplot1 = subplot(1,2,1,'Parent',figure1);
%% Uncomment the following line to preserve the X-limits of the axes
% xlim(subplot1,[-223.375 101.66]);
%% Uncomment the following line to preserve the Y-limits of the axes
% ylim(subplot1,[-223.375 101.66]);
box(subplot1,'on');
hold(subplot1,'on');

% Create multiple lines using matrix input to plot
plot1 = plot(X1,YMatrix1,'Parent',subplot1);
set(plot1(1),'DisplayName','Y = T','LineStyle',':','Color',[0 0 0]);
set(plot1(2),'DisplayName','Fit','LineWidth',2,'Color',[0 0 1]);

% Create plot
plot(X2,X2,'Parent',subplot1,'DisplayName','Data','Marker','o',...
    'LineStyle','none',...
    'Color',[0 0 0]);

% Create xlabel
xlabel({'Actual angle'},'FontWeight','bold','FontSize',12);

% Create ylabel
ylabel({'Spec. = 1*Actual+ 1.2e-15'},'FontWeight','bold','FontSize',12);
```

```
% Create title
title('Training: R=1','FontWeight','bold','FontSize',12);

% Create legend
legend1 = legend(subplot1,'show');
set(legend1,'Location','northwest');

% Create subplot
subplot2 = subplot(1,2,2,'Parent',figure1);
%% Uncomment the following line to preserve the X-limits of the axes
% xlim(subplot2,[-223.375 101.66]);
%% Uncomment the following line to preserve the Y-limits of the axes
% ylim(subplot2,[-223.375 101.66]);
box(subplot2,'on');
hold(subplot2,'on');

% Create multiple lines using matrix input to plot
plot2 = plot(X1,YMatrix1,'Parent',subplot2);
set(plot2(1),'DisplayName','Y = T','LineStyle',':','Color',[0 0 0]);
set(plot2(2),'DisplayName','Fit','LineWidth',2,'Color',[0 1 0]);

% Create plot
plot(X2,X2,'Parent',subplot2,'DisplayName','Data','Marker','o',...
    'LineStyle','none',...
    'Color',[0 0 0]);

% Create xlabel
xlabel({'Actual angle'},'FontWeight','bold','FontSize',12);

% Create ylabel
ylabel({'Spec ~= 1*Actual + 1.2e-15'},'FontWeight','bold','FontSize',12);

% Create title
title('All: R=1','FontWeight','bold','FontSize',12);

% Create legend
legend2 = legend(subplot2,'show');
set(legend2,'Location','northwest');

% uicontrol currently does not support code generation, enter 'doc
uicontrol' for correct input syntax
% In order to generate code for uicontrol, you may use GUIDE. Enter 'doc
guide' for more information

% uicontrol(...);
```

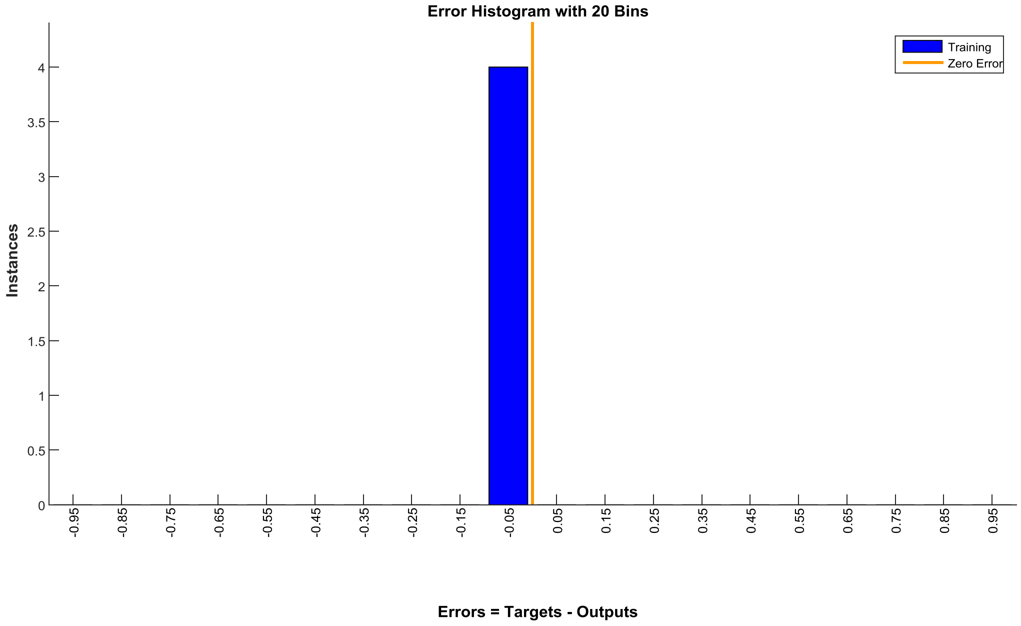


Fig 1.0.2: Error Histogram Graph of specified and actual angles.

1.9 MATLAB codes for the graph above

```
function createfigure1(xvector1, XData1, YData1)
%CREATEFIGURE1(XVECTOR1, XDATA1, YDATA1)
%  XVECTOR1:  bar xvector
%  XDATA1:  line xdata
%  YDATA1:  line ydata
% Create figure
figure1 = figure('Tag','TRAINING_PLOTERRHIST','NumberTitle','off',...
    'Name','Neural Network Training Error Histogram (ploterrhist), Epoch 0,
Performance goal met.');

% Create axes
axes1 = axes('Parent',figure1,...
    'XTick',[-0.95 -0.85 -0.75 -0.65 -0.55 -0.45 -0.35 -0.25 -0.15 -
0.0499999999999999 0.05 0.15 0.25 0.35 0.45 0.55 0.65 0.75 0.85 0.95],...
    'Position',[0.13 0.23 0.775 0.695],...
    'XTickLabel','');
%% Uncomment the following line to preserve the X-limits of the axes
% xlim(axes1,[-1 1]);
%% Uncomment the following line to preserve the Y-limits of the axes
% ylim(axes1,[0 4.4]);
hold(axes1,'on');

% Create multiple lines using matrix input to bar
bar1 = bar(xvector1,xvector1,'Visible','off','Parent',axes1);
set(bar1(1),'DisplayName','Training','FaceColor',[0 0 1],'Visible','on');

% Create line
line(XData1,YData1,'Parent',axes1,'DisplayName','Zero
Error','LineWidth',2,...
    'Color',[1 0.6 0]);

% Create text
text('Parent',axes1,'Units','normalized','HorizontalAlignment','center',...
    'FontWeight','bold',...
    'FontSize',12,...
    'String','Errors = Targets - Outputs',...
    'Position',[0.5 -0.22 0]);

% Create text
text('Parent',axes1,'HorizontalAlignment','right','Rotation',90,...
    'String','-0.95',...
    'Position',[-0.95 -0.05 0]);

% Create text
text('Parent',axes1,'HorizontalAlignment','right','Rotation',90,...
    'String','-0.85',...
    'Position',[-0.85 -0.05 0]);

% Create text
text('Parent',axes1,'HorizontalAlignment','right','Rotation',90,...
    'String','-0.75',...
    'Position',[-0.75 -0.05 0]);

% Create text
text('Parent',axes1,'HorizontalAlignment','right','Rotation',90,...
    'String','-0.65',...
    'Position',[-0.65 -0.05 0]);

% Create text
text('Parent',axes1,'HorizontalAlignment','right','Rotation',90,...
    'String','-0.55',...
    'Position',[-0.55 -0.05 0]);

% Create text
text('Parent',axes1,'HorizontalAlignment','right','Rotation',90,...
    'String','-0.45',...
    'Position',[-0.45 -0.05 0]);

% Create text
text('Parent',axes1,'HorizontalAlignment','right','Rotation',90,...
    'String','-0.35',...
    'Position',[-0.35 -0.05 0]);

% Create text
text('Parent',axes1,'HorizontalAlignment','right','Rotation',90,...
    'String','-0.25',...
    'Position',[-0.25 -0.05 0]);

% Create text
text('Parent',axes1,'HorizontalAlignment','right','Rotation',90,...
    'String','-0.15',...
    'Position',[-0.15 -0.05 0]);

% Create text
text('Parent',axes1,'HorizontalAlignment','right','Rotation',90,...
    'String','-0.05',...
    'Position',[-0.0499999999999999 -0.05 0]);

% Create text
text('Parent',axes1,'HorizontalAlignment','right','Rotation',90,...
    'String','0.05',...
    'Position',[0.05 -0.05 0]);

% Create text
```

```
text('Parent',axes1,'HorizontalAlignment','right','Rotation',90,...
    'String','0.15',...
    'Position',[0.15 -0.05 0]);

% Create text
text('Parent',axes1,'HorizontalAlignment','right','Rotation',90,...
    'String','0.25',...
    'Position',[0.25 -0.05 0]);

% Create text
text('Parent',axes1,'HorizontalAlignment','right','Rotation',90,...
    'String','0.35',...
    'Position',[0.35 -0.05 0]);

% Create text
text('Parent',axes1,'HorizontalAlignment','right','Rotation',90,...
    'String','0.45',...
    'Position',[0.45 -0.05 0]);

% Create text
text('Parent',axes1,'HorizontalAlignment','right','Rotation',90,...
    'String','0.55',...
    'Position',[0.55 -0.05 0]);

% Create text
text('Parent',axes1,'HorizontalAlignment','right','Rotation',90,...
    'String','0.65',...
    'Position',[0.65 -0.05 0]);

% Create text
text('Parent',axes1,'HorizontalAlignment','right','Rotation',90,...
    'String','0.75',...
    'Position',[0.75 -0.05 0]);

% Create text
text('Parent',axes1,'HorizontalAlignment','right','Rotation',90,...
    'String','0.85',...
    'Position',[0.85 -0.05 0]);

% Create text
text('Parent',axes1,'HorizontalAlignment','right','Rotation',90,...
    'String','0.95',...
    'Position',[0.95 -0.05 0]);

% Create xlabel
xlabel({''});

% Create ylabel
ylabel('Instances','FontWeight','bold','FontSize',12);

% Create title
title('Error Histogram with 20 Bins','FontWeight','bold','FontSize',12);

% Create legend
legend(axes1,'show');

% uicontrol currently does not support code generation, enter 'doc
uicontrol' for correct input syntax
% In order to generate code for uicontrol, you may use GUIDE. Enter 'doc
guide' for more information
```

```
% uicontrol(...);
```

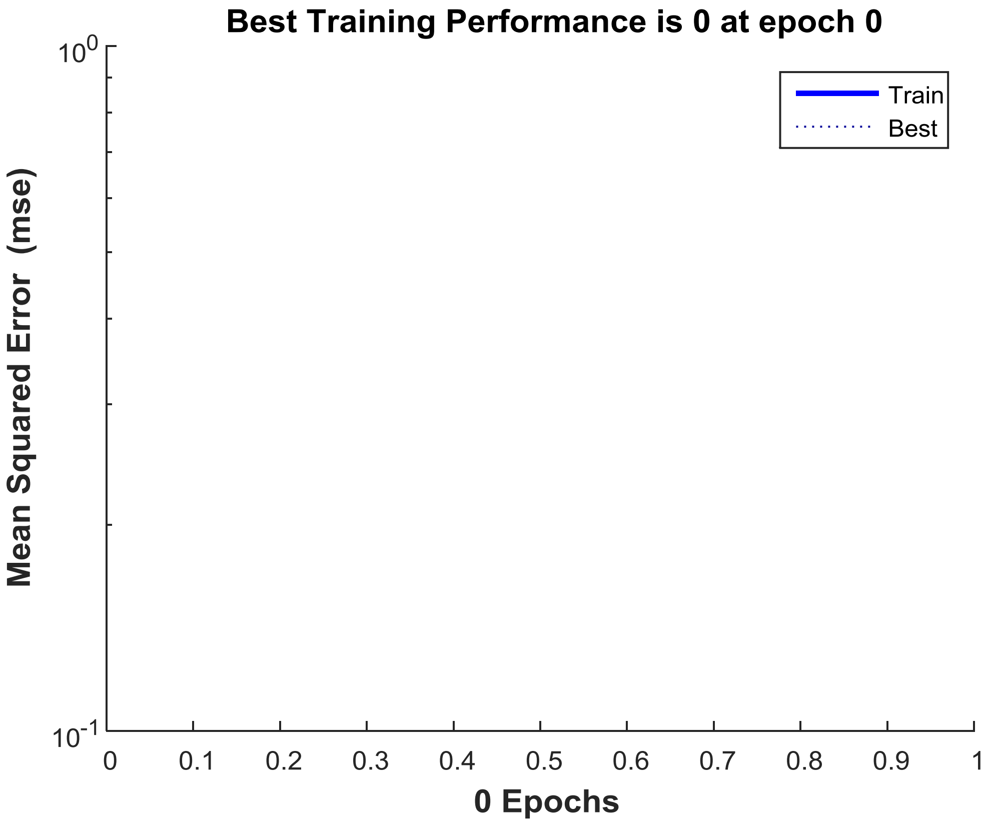


Fig1.0.3: training performance graph of actual and specified angles.

2.0 MATLAB codes for the graph above

```
function createfigure1(X1, YMatrix1, X2, Y1, X3, Y2)
%CREATEFIGURE1(X1, YMATRIX1, X2, Y1, X3, Y2)
%  X1:  vector of x data
%  YMATRIX1:  matrix of y data
%  X2:  vector of x data
%  Y1:  vector of y data
%  X3:  vector of x data
%  Y2:  vector of y data
% Create figure
figure1 = figure('Tag','TRAINING_PLOTPERFORM','NumberTitle','off',...
    'Name','Neural Network Training Performance (plotperform), Epoch 0,
Performance goal met.');

% Create axes
axes1 = axes('Parent',figure1,'YScale','log');
%% Uncomment the following line to preserve the X-limits of the axes
% xlim(axes1,[0 1]);
%% Uncomment the following line to preserve the Y-limits of the axes
% ylim(axes1,[0 1]);
hold(axes1,'on');

% Create multiple lines using matrix input to semilogy
semilogy1 = semilogy(X1,YMatrix1,'LineWidth',2,'Parent',axes1);
set(semilogy1(1),'DisplayName','Train','Color',[0 0 1]);
set(semilogy1(2),'Color',[0 0.8 0]);
set(semilogy1(3),'Color',[1 0 0]);
set(semilogy1(4),'MarkerSize',16,'Marker','o','LineWidth',1.5,...
    'LineStyle','none',...
    'Color',[0 0 0.6]);

% Create semilogy
semilogy(X2,Y1,'DisplayName','Best','LineStyle',':','Color',[0 0 0.6]);

% Create semilogy
semilogy(X3,Y2,'LineStyle',':','Color',[0 0 0]);

% Create xlabel
xlabel('0 Epochs','FontWeight','bold','FontSize',12);

% Create ylabel
ylabel('Mean Squared Error  (mse)','FontWeight','bold','FontSize',12);

% Create title
title('Best Training Performance is 0 at epoch 0','FontWeight','bold',...
    'FontSize',12);

% Create legend
legend(axes1,'show');

% uicontrol currently does not support code generation, enter 'doc
uicontrol' for correct input syntax
% In order to generate code for uicontrol, you may use GUIDE. Enter 'doc
guide' for more information

% uicontrol(...);
```

2.1 MATLAB SCRIPT AND FUNCTION FOR THE TRAINED DATA ARE SHOWN BELOW.

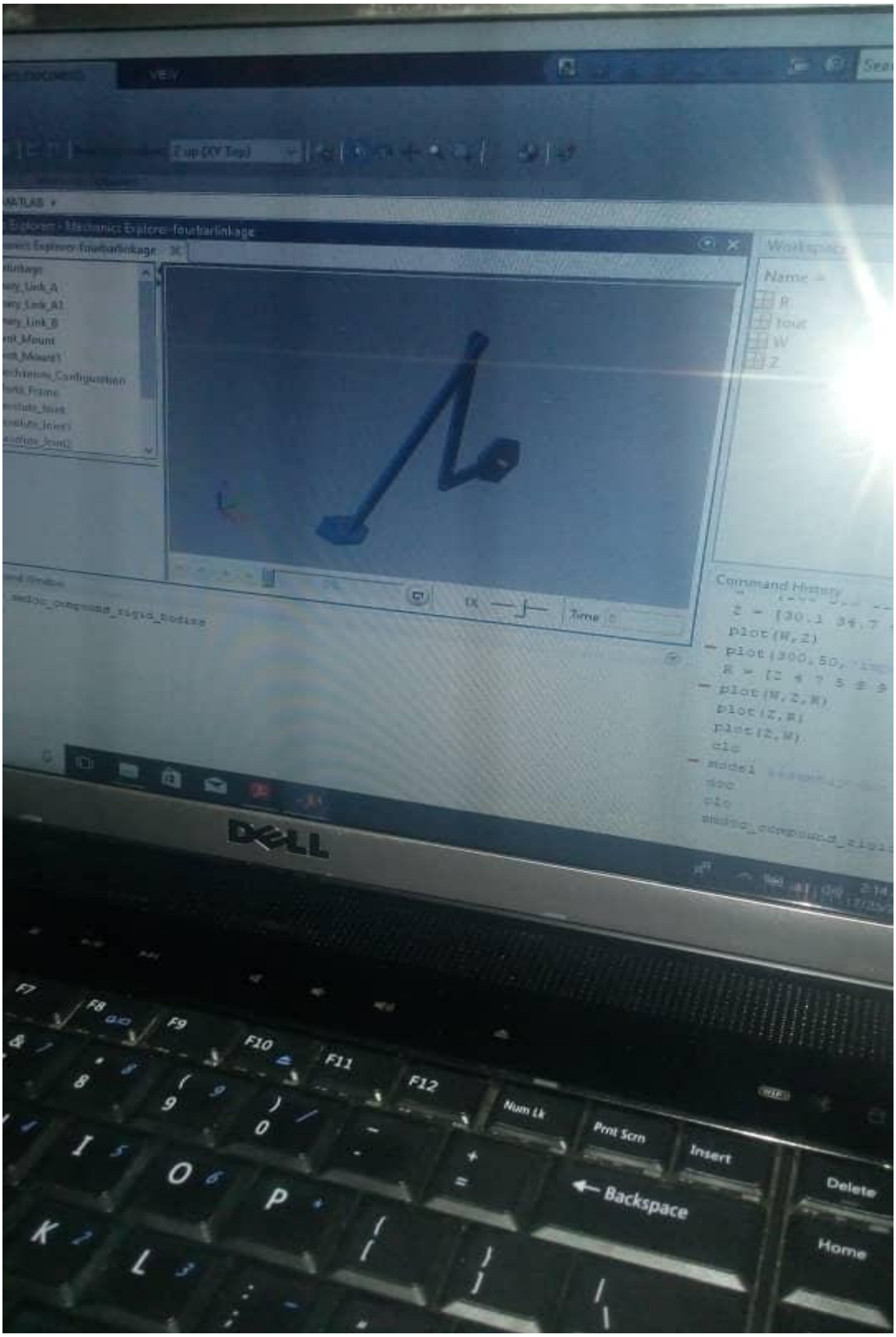
```
% Solve an Input-Output Fitting problem with a Neural Network
% Script generated by Neural Fitting app
% This script assumes these variables are defined:
%
%   data - input data.
%   data - target data.

x = data;
t = data;

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainlm';  % Levenberg-Marquardt backpropagation.

% Create a Fitting Network
hiddenLayerSize = 10;
net = fitnet(hiddenLayerSize,trainFcn);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Train the Network
[net,tr] = train(net,x,t);
```

```
% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotfit(net,x,t)
```

2.1.0 MATLAB FUNCTION

```
 function [Y,Xf,Af] = myNeuralNetworkFunction(X,~,~)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
%
% [Y] = myNeuralNetworkFunction(X,~,~) takes these arguments:
%
%   X = 0xTS cell, 0 inputs over TS timsteps
%
% and returns:
%   Y = 0xTS cell of 0 outputs over TS timesteps.
%
% where Q is number of samples (or series) and TS is the number of
timesteps.

%#ok<*RPMT0>

% ===== NEURAL NETWORK CONSTANTS =====


% ===== SIMULATION ========

% Format Input Arguments
isCellX = iscell(X);
if ~isCellX, X = {X}; end;

% Dimensions
TS = size(X,2); % timesteps

% Allocate Outputs
Y = cell(0,TS);

% Time loop
for ts=1:TS

end

% Final Delay States
Xf = cell(0,0);
Af = cell(0,0);

% Format Output Arguments
if ~isCellX, Y = cell2mat(Y); end
```

```
end

% ===== MODULE FUNCTIONS ========
```



```
plot(W,Z)
plot(300,50,
R = [2 4 7 5 8 9
plot(W,Z,R)
plot(Z,R)
plot(Z,W)
clc
model
doc
clc
smdoc_compound_rigi
```

Fig 1.1: Simulation of four-bar-linkage model in mechanics explorer.

SIMULINK MODELING OF DEGREES OF FREEDOM OF FOUR-BAR-LINKAGE

The equation expressing the degrees of freedom of a four-bar-linkage is adopted from Khurmi, 2012 and it is as follows:

$$\boldsymbol{n} = 3(\boldsymbol{l} - 1) - 2\boldsymbol{j} \quad \text{…. (Khurmi, 2012)}$$

Where n = degrees of freedom, l = number of links and j = number of binary joints.

The simulink model is shown in fig 1.2 below.

2.2.1 MATLAB code to compute degrees of freedom.

```
clc
% MATLAB programme to compute degrees of freedom,DOF of fourbarlinkage
% l = number of links
% j = number of binary joints
enter_number of links,l = input('enter number of links');
disp('number of links:')
number of binary joints,j = ('enter number of binary joints');
disp('number of binary joints:')
% n = 3*(l - 1) - 2*j;
n = 3*( 'number of links' - 1)- 2*'number of binary joints';
disp('degrees of freedom :')
disp(n)
end
```

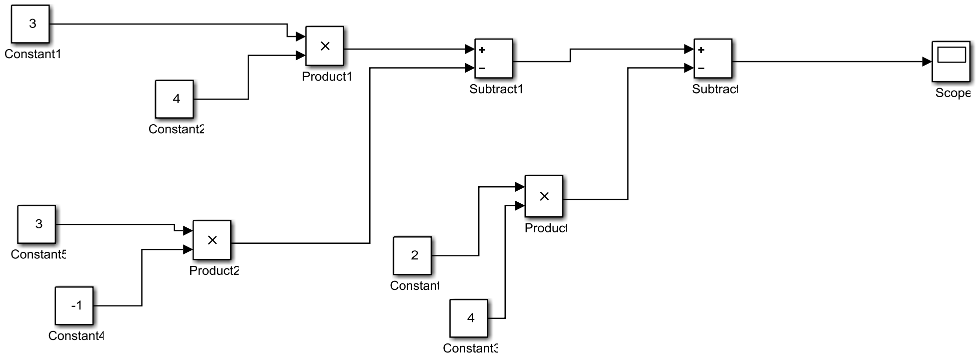


Fig 1.2: simulink model of degrees of freedom.

SIMULINK MODEL OF CONFIGURATION ANGLES OF FOUR-BAR-LINKAGE

The simulink model of configuration angles difference of four-bar-linkage with ramp signal is shown in fig 1.3 below. Freudenstein’s equation model was adopted for the simulink model.

$$\cos(\boldsymbol{\theta} - \emptyset) = \boldsymbol{k_1} \cos \emptyset - \boldsymbol{k_2} \cos \boldsymbol{\theta} + \boldsymbol{k_3} \quad \text{…(Khurmi,2012)}$$

$$where\ k_1, k_2\ and\ k_3\ are\ ratio\ of\ link\ lengths.$$

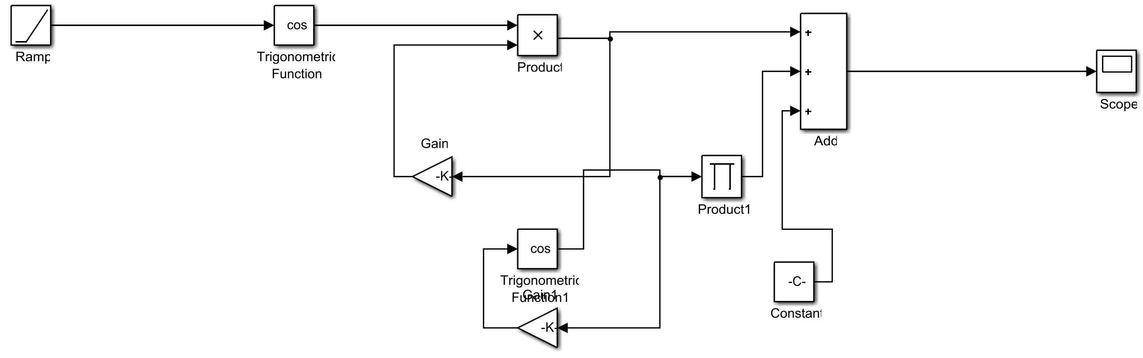


Fig 1.3: simulink model of configuration angles of four-bar-linkage.

The simulink model of configuration angles difference of four-bar-linkage with sine wave signal is shown in **fig 1.4** below. Freudenstein’s equation model was adopted for the simulink model.

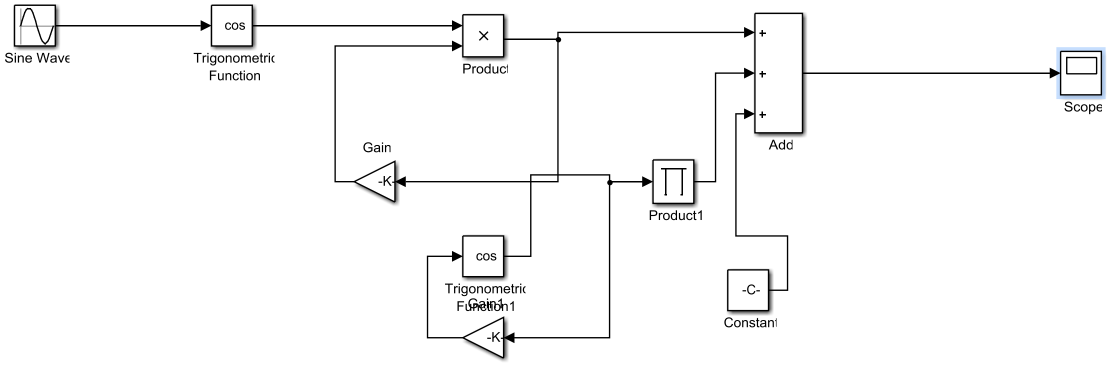


Fig 1.4: simulink model of configuration angles of four-bar-linkage with wave signal.

Discussion of Findings

The outcomes of the study, evaluation of actual configuration angles of a four bar linkage for precision applications through simulink approach were discussed here. According to **table 1.0,** the mechanism links position angles specified were +30, +30, - 23 and 0 degrees for the corresponding revolute joints. The block parameters as specified were also entered in the individual block model. ODE3 solver was used to run the simulation problem for 20seconds. Simulation results showed that the actual position angles of the four bar links were evaluated to be – 92.4196, -223.375, 0 and + 101.66 degrees, corresponding to **Revolute Joint, Revolute Joint 1, Revolute Joint2 and Revolute Joint3** respectively. However, the specified and actual velocities of links were found to be the same values being +15 rad/s, +25rad/s, 0 and +12rad/s respectively after the simulink model simulation, at high priorities. **Graph of Fig 1.0.1,** with the regression coefficient of 1 indicated that there is a close and positive relationship between position angles specified and actual position angles of revolute joints.

Training of actual position angles and position angles specified data using levenberg marquardt algorithm at 70% training data, 15% test data and 15% validation data respectively revealed that the best performance level was at 0 and at epoch 0 and this revealed lowest values of position angles specified. In addition, Simulink blocks were used to create models for degrees of freedom and configuration angles of a four bar linkage. Furthermore, matlab scripts/codes for graphs and models were also presented.

Conclusion

Obviously, results from the study revealed that the specified/designed position angles of a four bar linkage varies from actual/operational position angles with closed and positive relationship. The study also proved that the specified and actual velocities of a four bar linkage remained the same.

Recommendations

The following recommendations were suggested based on the study:
1) Four bar linkage must be configured at their actual position angles to improve mechanism task precision.
2) Revolute joints must be adequately lubricated to encourage free rotation motion.
3) This study can also be done in future using other training algorithms and advanced program for generalization.

REFERENCES

[1]. Arup, J. (2016). Adaptive design on crank and slotted lever mechanism. Journal of Engineering Technology 4(1).

[2]. Khurmi, R.S & Gupta, J.K. (2014). Theory of machines. New Dehi: Khanna Publishers.

[3]. Rajput, R.K. (2012). Strength of materials. New Dehi: Khanna Publishers.

[4]. Sagar, S. & Thakare, P. S.(2012). Computer Aided Modeling and Position Analysis of Crank and Slotted Lever Mechanism. Retrieved from: www.researchgate.net.

[5]. Shelare,S., Thakare, P.S. & Handa, C.C. (2012). Computer aided modeling and position analysis of crank and slotted lever mechanism. International Journal of Mechanical Engineering and Robotics Research 2(1).

[6]. Shahbaz, M.S., Sujata, P. K. & Dahake, S.A.(2017). Computer aided modeling and analysis of crank and slotted lever quick return mechanism. International Journal of Engineering Science and Research Technology6(2).

[7]. Ugwuegbu, D.C & Ewurum, T.I. (2022). Estimate of Crank Length for Best Performance of a Quick Return Motion Mechanism. International Journal of Advances in Engineering and Management 4(5), pp. 321-328.

[8]. Ugwuegbu, D.C & Ewurum, T.I. (2022). Computer Aided Design and Computer Aided Manufacturing(CAD/CAM). Owerri: Ingenious Publishers.