Path Synthesis of crank-rocker mechanisms using neural network

Shashank Sharma

December 15, 2017

1 Problem Statement

Mechanism synthesis problem can mostly be classified into path, motion and function synthesis problems. Each of these problems have been too complex to be solved for a generalized n-bar one d.o.f. mechanism. Thus, most of the literature deals with synthesis problem for the simplest case in family of all one d.o.f mechanisms, the 4-bar mechanism. Different types of possible joints (revolute or prismatic) and coupler point motion (closed or open) introduces further complexities into the system. Many analytic and approximate approaches have been proposed in an attempt to solve these problems. In this report, the focus is on Path synthesis problem for closed loop 4-bar mechanisms. The problem focus has been visualized in Fig 1

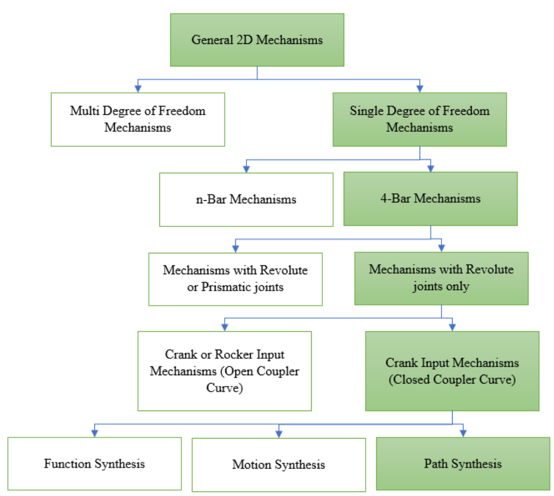

Figure 1: Problem focus area

2 Mechanism Description

2.1 Parameters

A four bar mechanism (at any instant) can be described using ten parameters. Those parameters are

- Actuating Fixed Pivot (x_0, y_0)
- Link Lengths (a, b, c, d)
- Fixed link angle (β)
- Coupler point (f, g)
- Crank angle (ϕ)

Figure 2 is a visual description of these parameters.

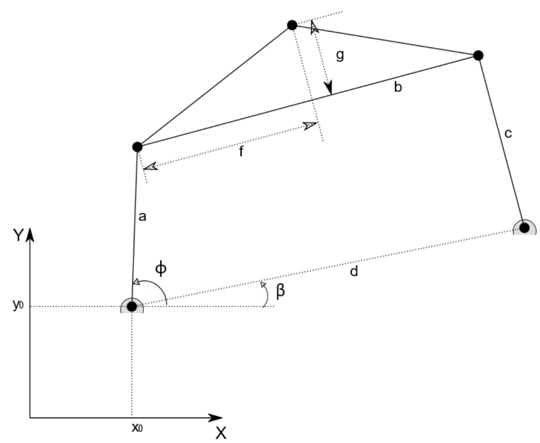

Figure 2: Problem focus area

2.2 Shape Invariance

The objective of our problem is of path shape matching. Thus, it is fruitful to delve into how each of the mechanism parameters affects the shape of coupler motion.

Changing $(x0, y0, \beta)$ affects only the position but not the shape of coupler curve. As a result, these parameters can be fixed to specific values and a family of four-bar path can be generated which is representative of all possible paths. These parameters have been fixed to (0, 0, 0) in our implementation.

Similarly, scaling the mechanism does not change the coupler curve shape as long as relative ratio of length remain conserved. Thus (d) can be fixed to (1) i.e. the length of fixed link is scaled to unit length.

Also, changing () changes the start point but not the shape for a closed loop mechanism, thus can be fixed to (0).

Each of these parameters thus do not have any affect to the shape of Path of coupler motion. Ignoring them and creating paths using just the parameters (a, b, c, f, g) still represents to all possible normalized paths of a four bar mechanism.

2.3 Optimal Transmission Angle

Another mechanism property of immense importance is its Transmission Angle. The optimum angle is 90^o. However, Since the angle will be constantly changing during the motion cycle of the mechanism, there will be a position at which the transmission angle will deviate most from 90^0. Minimizing this maximum transmission angle gives two additional constraints.

$$b = c = \sqrt{\frac{a^2 + d^2}{2}} \tag{1}$$

It must be noted that this equation implicitly satisfies design constraint of crank- coupler mechanisms when $a/d < 1$. Thus, reducing the mechanism parameter space to a mere 3 variables namely (a, f, g) gives us a family of optimum closed loop four bar mechanisms.

3 Data Generation

3.1 Path Generation

To create a Neural Network model which does path synthesis, a training dataset is needed. In the previous section, it has been discussed that three design parameters (a, f, g) are sufficient to describe all the four bar mechanisms of interest. In our implementation, the domain has been taken as $a \in (0, 1), f \in (2, 2), g \in (2, 2)$. The domain constraint on a is in accordance to family of closed loop mechanisms. Constraints on f, g are in place because mechanism having extremely large coupler compared to other links is usually not of engineering interest.

Once a set of a, f, g are selected, location of coupler location can be calculated using loop closure and coupler point equation. In our implementation, a set of 40 sample points representing this path is calculated. This output can be summarized as

$$Path = \{x_i, y_i\} \quad k \in [1, 40] \tag{2}$$

A sample mechanism and its path generated for parameters $(a, f, g) = (0.5594, 1.9817, 1.0667)$ is shown in Fig 3

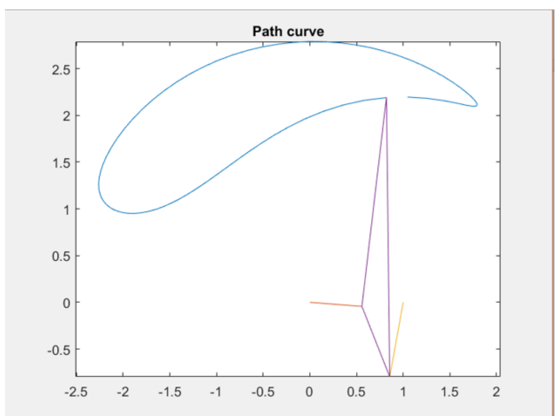

Figure 3: Path generated for a sample (a, f, g)

3.2 Path Normalization

The path calculated in previous section has to be normalized to remove its scale, orientation, location and start point information. This is done as we are interested in shape fitting. Once a mechanism describing the same shape is found, it is easy to transform to get the actual mechanism.

First, the path is rotated to align its principal axes with x axis. This removes the rotational information.

Then, its bounding box is calculated and the lower left corner is translated to (0,0) in global frame. This removes the location information from curve.

Scaling is subsequently done to make the width of bounding box unit length. This operation removes the relative size information.

Start point is chosen to be the one nearest to the bottom left corner of bounding box. This standardizes the representation of two closed curves with different start points.

The mathematical formulation of the procedure is as follows

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}_{normalized} = \begin{bmatrix} \frac{\cos\alpha}{w} & \frac{\sin\alpha}{w} & -\frac{P_x}{w} \\ -\frac{\sin\alpha}{w} & \frac{\cos\alpha}{w} & -\frac{P_y}{w} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}_{original} \tag{3}$$

where α is the direction of major principal axis with respect to x-axis, w is the width of the bounding box and (P_x, P_y) is lower left corner of the bounding box. This gives an array of normalized path points.

After these four operations, the path we get is normalized and isolates the shape of coupler motion.

The normalized path for sample mechanism with parameters $(a, f, g) = (0.5594, 1.9817, 1.0667)$ is shown in Fig 4

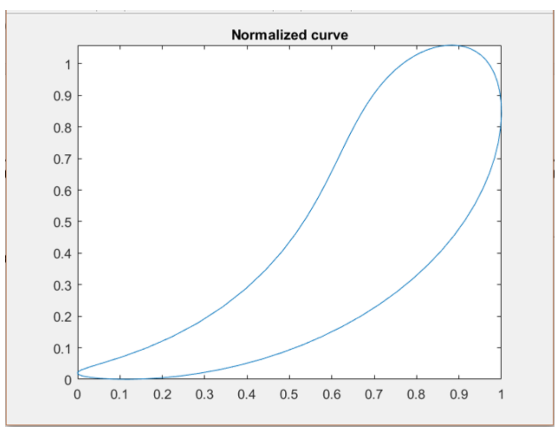

Figure 4: Normalized Path for sample mechanism

3.3 Wavelet Descriptors

A wavelet decomposition is applied to normalized path to get its descriptors. Three step 1D Discrete wavelet Transform is carried out on x and y independently. Dauberchies 3 wavelets are used during the wavelet decomposition.

The decomposition process can be written as represented below.

$$F = A + \sum_{m=k}^{M} D_m \tag{4}$$

Here, F is the original signal, A is the approximate signal and D_m is the m^{th} detailed signal. This decomposition is carried out for each x and y coordinate independently.

It is observed that Detail signals carry extremely little information in case of Four bar motion. Finally, a set of 9 descriptors each for x and y coordinate are calculated and combined into a vector. Thus, path is described by wavelet descriptors as follows

$$Path = \{d_i\} \quad k \in [1, 18] \tag{5}$$

The x and y coordinate wavelet descriptors for normalized path of sample mechanism with parameters $(a, f, g) = (0.5594, 1.9817, 1.0667)$ is shown in Fig 5 and Fig 6

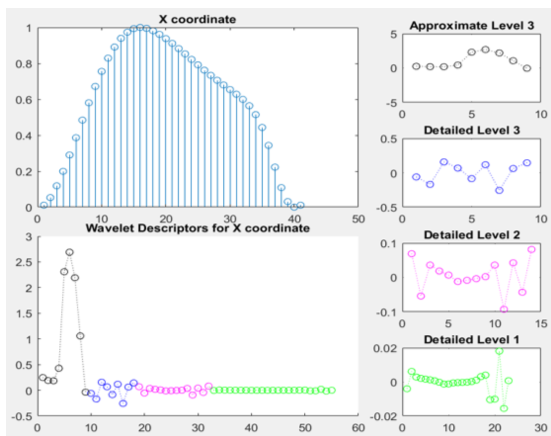

Figure 5: X coordinate Wavelet Descriptors of Normalized Path

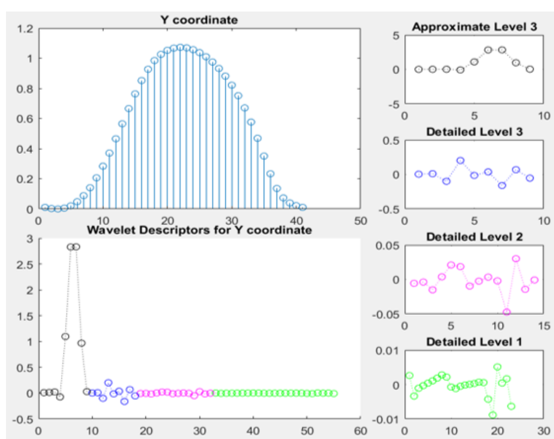

Figure 6: Y coordinate Wavelet Descriptors of Normalized Path

4 Training the Neural Network

The task of training neural network can now be carried out. The input and output to the NN is as follows

$$Input = \{d_i\} \quad k \in [1, 18] \tag{6}$$

$$Output = \{a, f, g\} \tag{7}$$

Data for 10,000 randomly select optimal mechanisms was generated and used in training phase. Data is split in 70:15:15 ratio for Training, Validation and Testing respectively.

Mean Squared error measure was the metric used by neural network to minimize difference between outputs and targets. A multilayer feed-forward NN with linear transfer function layer at input and output has been used. Log-sigmoid or Tan-sigmoid function in hidden layers is used. A variety of NN architecture with different amount of hidden layers were tried to gain insight in which architecture was better. The permutations tried were

- Case 1- 1 tansig hidden layer with 22 nodes
- Case 2- 1 tansig hidden layer with 44 nodes
- Case 3- 2 tansig hidden layer with 22 nodes
- Case 4- 2 logsig hidden layer with 22 nodes

Results for each case is discussed in subsequent sections

Over-fitting to the test data was prevented by stopping training if error for evaluation set increases for more than 6 epochs.

4.1 Case 1

In this case, linear transfer function layer at input and output is present. One hidden layer of tansig function with 22 nodes is present in the middle. For this NN, the best fit mean squared error for test data is .1 in magnitude. The architecture and training error convergence is displayed in Fig 7 and Fig 8.

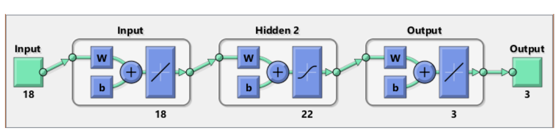

Figure 7: Neural Network Architecture

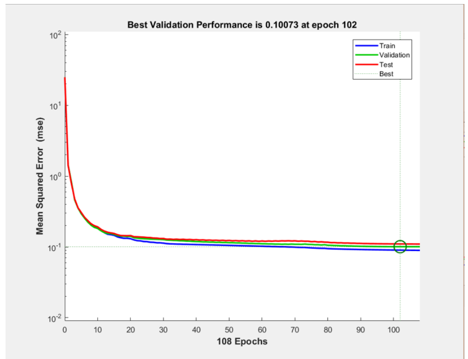

Figure 8: Training error measure

4.2 Case 2

The architecture for this NN is same except that hidden tansig layer has 44 nodes i.e. double than previous. In this instance, the best fit mean squared error for test data is .0795 in magnitude which is somewhat better than the previous attempt. The architecture and training error convergence is displayed in Fig 9 and Fig 10.

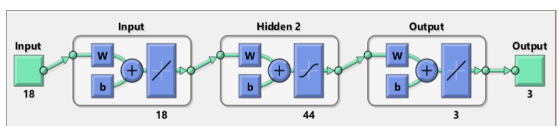

Figure 9: Neural Network Architecture

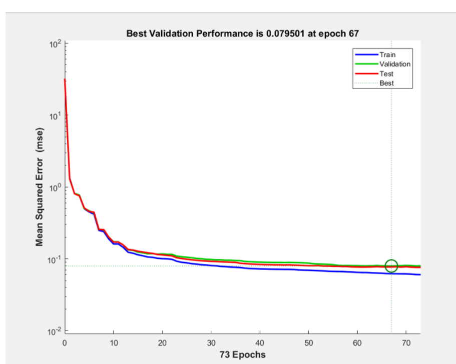

Figure 10: Training error measure

4.3 Case 3

The architecture for this NN has two hidden tansig layer with 22 nodes i.e. double hidden layers than Case 1. In this instance, the best fit mean squared error for test data is .0289 in magnitude which is much better than the previous attempt. The architecture and training error convergence is displayed in Fig 11 and Fig 12.

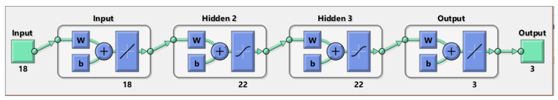

Figure 11: Neural Network Architecture

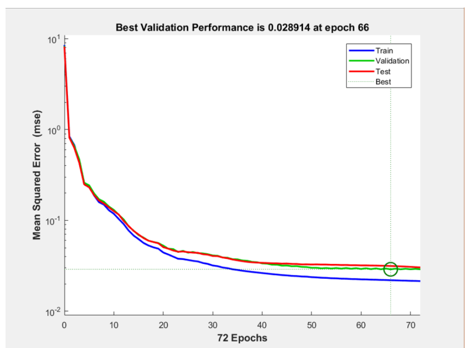

Figure 12: Training error measure

4.4 Case 4

The architecture for this NN is same as Case 3 except that hidden layer use logsin instead of tansin. In this instance, the best fit mean squared error for test data is .0437 in magnitude which is worse than the case using tansin in hidden layers. The architecture and training error convergence is displayed in Fig 13 and Fig 14.

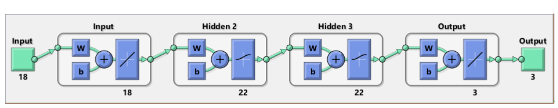

Figure 13: Neural Network Architecture

4.5 Observations

From above test cases we can conclude

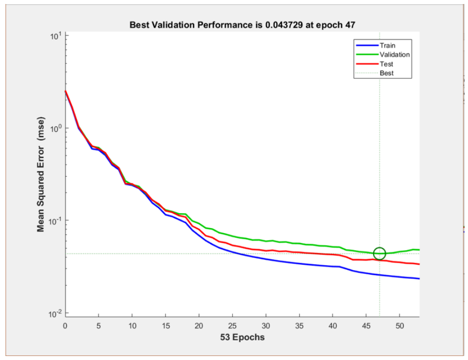

Figure 14: Training error measure

	Actual parameters	Predicted by NN
Sample 1	(0.3052, -0.5843, 0.2318)	(0.3899, -0.3313, 0.1086)
Sample 2	(0.1917, 0.3534, -1.8288)	(0.1149, 0.3147, -1.5754)
Sample 3	(0.4726, 0.3516, 1.4836)	(0.6998, 0.0708, 1.5706)
Sample 4	(0.8354, 1.6383, -1.9941)	(0.7866, 1.6048, -2.0914)

Table 1: Test design parameters (a, f, g)

- The more nodes NN has, the better it approximates data.
- For Path synthesis problem, increasing layers seems to make more difference than increasing nodes in same layer.
- Tansig achieves better approximation than logsig.

5 Testing the Neural Network

The trained best trained NN i.e. Case 3 is now tested with real data. A path for known mechanism is inputted to the NN. The mechanism predicted by NN is compared to actual mechanism for some sample cases in Fig 15, 16, 17, 18. The parameter values are given in Table 1. The Blue curve is the targeted path while the red path is the one approximated by NN.

6 Conclusion

From the results, its apparent that there is still exceptionally large scope of improvement in the results predicted by trained Neural network. It can also be noted that detail wavelet descriptors for four bar motion have extremely low magnitude. Also, the NN is unable to give alternate configurations for the same mechanism if only one configuration was in training set. Thus, other methods for one-many mapping function need to be looked into.

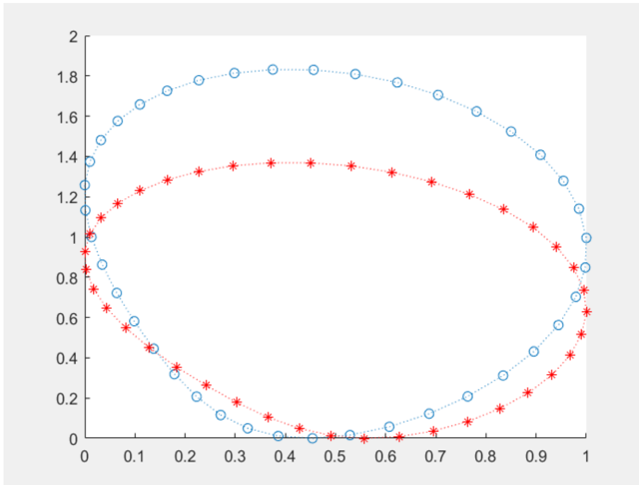

Figure 15: Test sample 1

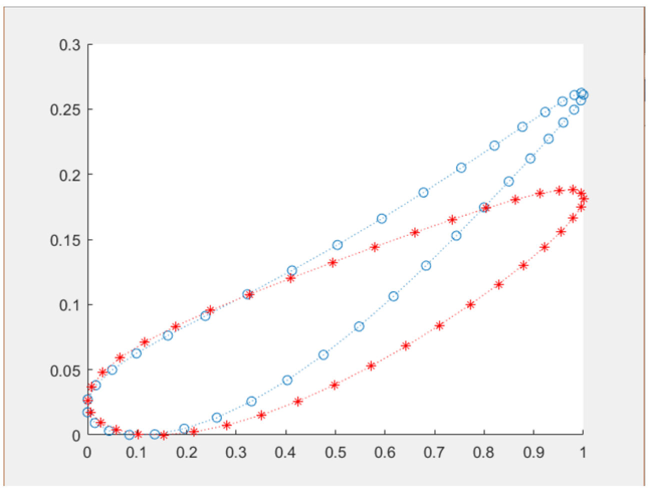

Figure 16: Test sample 2

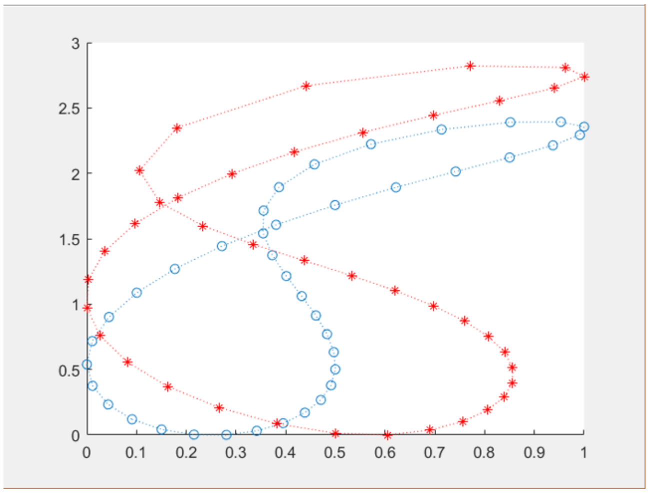

Figure 17: Test sample 3

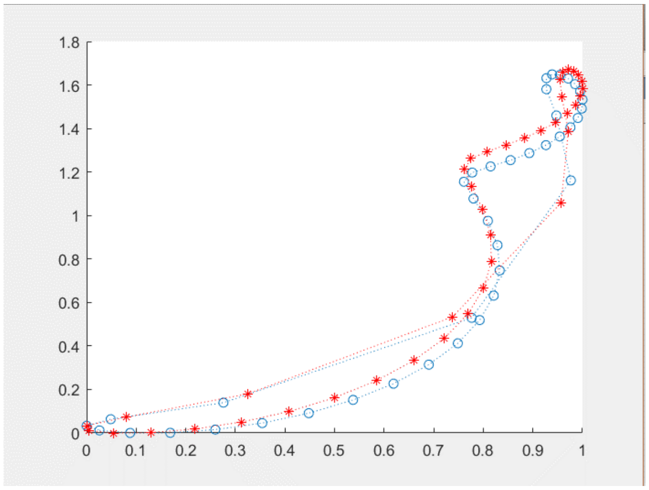

Figure 18: Test sample 4