CHAPTER 4

COMPUTER PROGRAM FOR ALLOCATING REACTIVE POWER

4.1 Computer Program for Allocating Reactive Power

The software used in this study is MATLAB. It can support the helpful functions, (neural network toolbox). The MATLAB program for allocating reactive power is composed of two parts: training and learning phase and recall phase. In the training and learning phase, the neural nets is trained to return a specific output when give a specific input. This is done by continuous training on a set of training data. In the recall phase, the neural nets return output based on the input.

The thesis considers two cases which are five-bus sample system and 21-bus system in central I of Electricite Du Lao (EDL) in Lao PDR system or EDL21-bus system.

4.2 Preparation of Historical Data of Load and Corresponding Factors

The inputs for this neural network as follows, the structure of artificial neural network back propagation five-bus sample system is shown in Figure. 4.1. For five - bus sample system, input layer has 3 nodes which are reactive power from each load bus. Output layer has 6 nodes which are reactive power from each generator to load bus. Hidden layer has 20 nodes. The inputs and outputs are described in Table 4.1.

Table 4.1 Description of input and outputs of ANN for five-bus sample system

Input and output	Neurons	Description
I_1- I_3	3	Reactive Loads
O_1-O_6	6	Allocating reactive power between generator and load

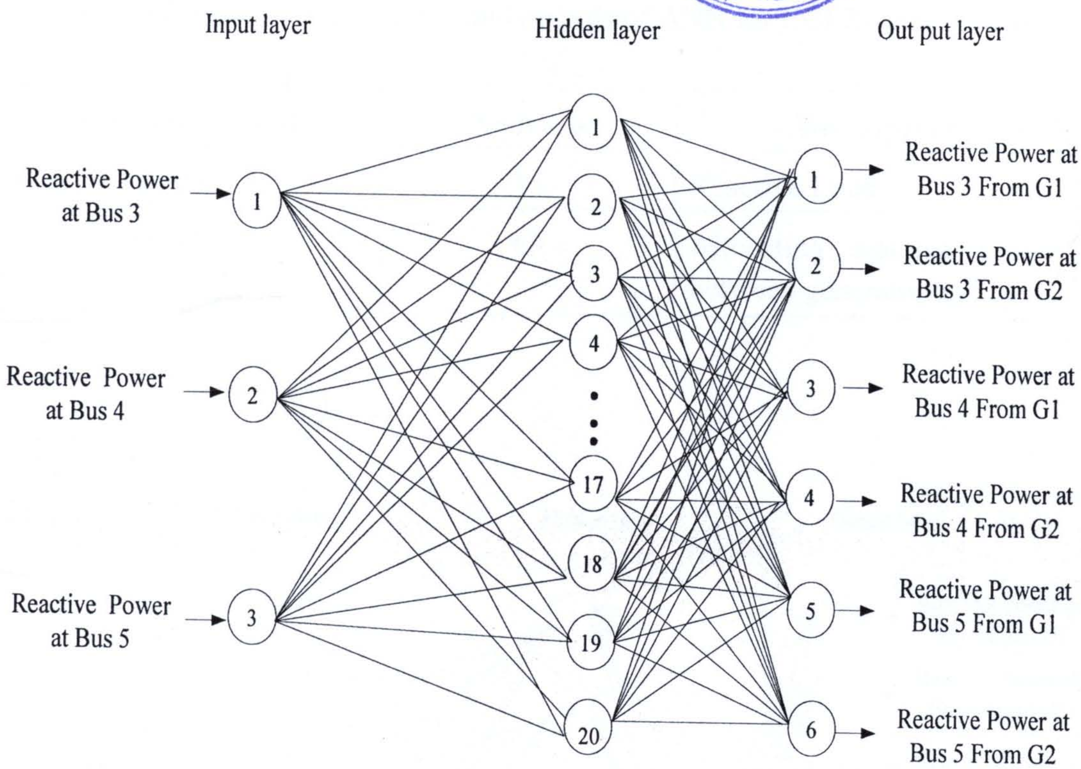

Figure 4.1 Structure of artificial neural network for allocating reactive power of five-bus sample system

Furthermore, the inputs of the learning process in neural network is using 2-day daily load curve and performing load flow analysis for 48 hours of load demand for 3 inputs.

In the error-backpropagation learning process in the neural network, the target outputs have to be predetermined as the reference values of computed from Y-bus matrix of the training and learning processes. The outputs is using 2-day daily load curve and performing load flow analysis for 48 hours of load demand for 6 outputs.

The structure of artificial neural network back propagation 21-bus in central I of Electricite Du Lao (EDL) in Lao PDR system or EDL21-bus system is shown in Figure 4.2. For 21-bus system, input layer has 15 nodes which are reactive power from each load bus. Output layer has 60 nodes which are reactive power from each generator to load bus. Hidden layer has 60 nodes. The inputs and outputs are described in Table 4.2.

Table 4.2 Description of input and outputs of ANN for EDL21-bus system

Input and output	Neurons	Description
I_1- I_{15}	15	Reactive Loads
O_1-O_{60}	60	Allocating reactive power between generator and load

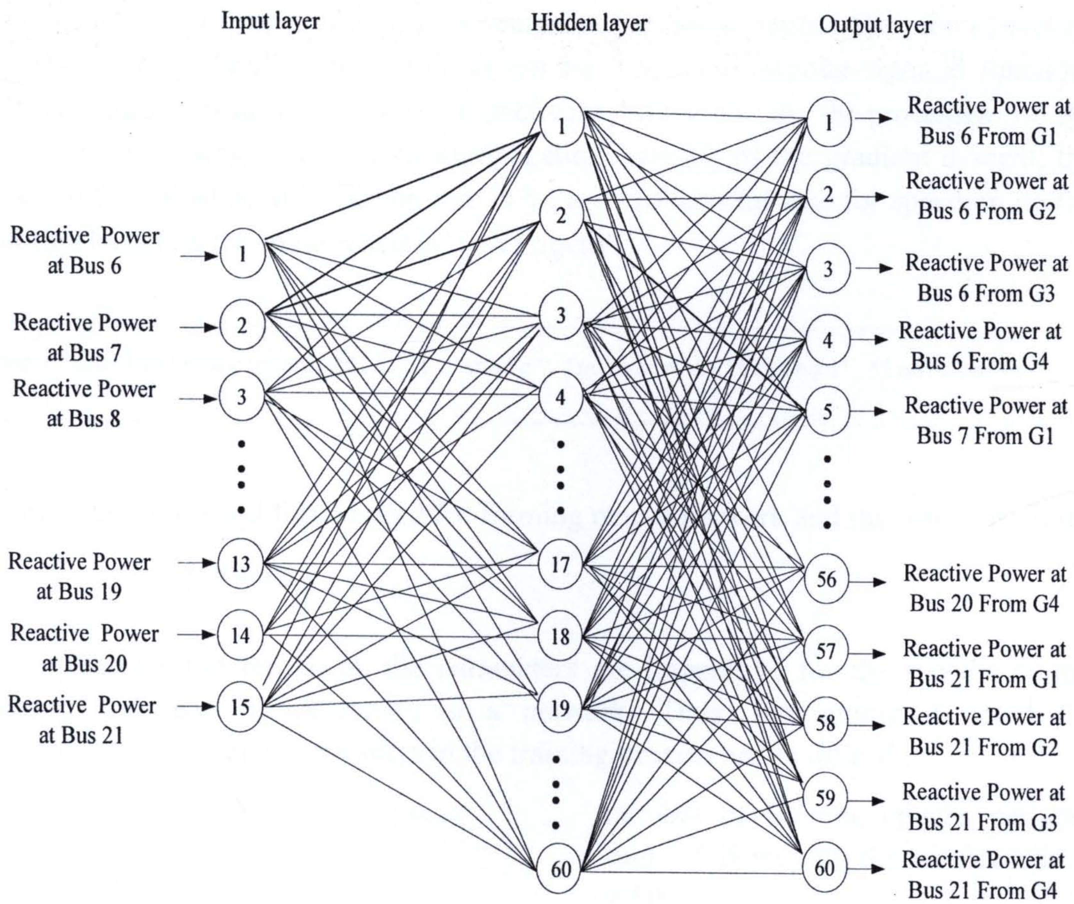

Figure 4.2 Structure of artificial neural network for allocating reactive power of EDL 21-bus system

The inputs of the learning process in neural network is using 2- day daily load curve and performing load flow analysis for 48 hours of load demand for 15 inputs and The outputs is using 2-day daily load curve and performing load flow analysis for 48 hours of load demand for 60 outputs.

4.3 Neural Network Establishment and Functions of Training and Learning

In MATLAB, the commands used for neural network establishment, and the functions of training and learning processes given in such commands are available, as shown in the Figure 4.3.

The command 'newff' is applied in forming one-hidden-layer network with 20 neurons in the hidden layer and 6 neuron in the output layer. Also, the activating functions for the hidden and output layers are 'tansig' (bipolar-sigmoid function) and 'purelin' (linear function), respectively. Moreover , in the processes of the feedforward training and error-backpropagation learning of the gradient descent, the function for adjusting the training rate is 'traingd' and the one for speeding up the learning based on the momentum is 'learngdm'.

```
Net= newff(minmax(pn),[20,6],{'tansig','purelin'},'traingd','learngdm);
```

Figure 4.3 Command format used for forming neural network and the functions of training and learning

Besides the functions, the parameters are imperative for the success of the training and learning processes in a network. Thus, the commands used for determining applicable parameters in the training process are as follows:

(1) `net.trainparam.show`:	number of training epochs per one time of displaying the result, default = 50
(2) `net.trainparam.lr`:	training rate, default = 0.05
(3) `net.trainparam.epochs`:	maximum number of training epochs, default = 300
(4) `net.trainparam.goal`:	performance goal required from training, default = 1e-5

The application of these commands is illustrated in the Figure 4.4. Some in the figure can be ignored if the parameters are assumed to be equal to their own defaults.

```
net.trainparam.show=2000;
net.trainparam.lr=0.05;
net.trainparam.epochs=3000000;
net.trainparam.goal=1e-5;
```

Figure 4.4 Commands used for setting parameters of neural network

4.4 Neural Network Training and Solution Determination

The command `net=train(net,p,t)` is applied in the training process. It activates the set up network. Subsequently, the training process in the network will be performed based on the function in this command and the predetermined parameters used in such a function. The amount of parameters is depended on the number of assumed functions.

In order to investigate the effectiveness of the training process, the solution determination of the performance test for the trained network is used. As shown in the Figure 4.5, to compute "`an`" from the command `sim (net,pn)`, the normalized input (`pn`) is solved in (`net`).

```
an=sim(net,pn);
```

Figure 4.5 Testing commands for solution determination of neural network