

Application of Neural Networks for MPP of Solar PV Panel

R. Kiranmayi

Associate Professor, Department of EEE, JNTUHCEH, Hyderabad, India

ABSTRACT: This paper presents the construction of a model for PV panel and for computing and tracking the maximum power point using neural networks concept. This model takes into account the series and shunt resistances of the panel. The equivalent circuit and the basic equations of a PV cell panel in Standard Test Conditions (STC), as well as parameters extraction from the data sheet values are shown. Based on these equations, a PV panel model for KL Solar 60W panel, which is able to predict the panel behavior in different temperature and irradiance conditions is built. Photovoltaic module consists of many solar cells which are electrically connected and placed between glass and tedlar plate. The model of a photovoltaic module consisting of 36 solar cells in series is presented in this study.

KEYWORDS: Modeling, Maximum Power Point Tracking (MPPT), Photovoltaic (PV), Maximum Power Point (MPP), Artificial Neural Networks(ANN).

I. INTRODUCTION

Solar arrays are used in wide variety of applications in many facets of life like day to day applications, industrial applications, terrestrial applications and so on. For best utilization, the photovoltaic cells must be operated at the maximum power point. However the maximum power point varies with illumination, temperature, radiation and other ageing factors.

Insolation level, cell temperature and array voltage set the limits at the maximum power available from PV array. The limit imposed by insolation levels on the available PV power can be relieved by the use of solar path tracker, while the limit imposed by the cell temperature on the PV system may be relieved by improved thermal conductivity between PV cells and the environment. However, the insolation levels and the cell temperature determine only the best obtainable matching. The array voltage determines the real matching. The mismatch can be improved by the use of a peak power point tracker converter to track the maximum power point in response range of the PV panel.

II. SOLAR CELL MODEL DEVELOPMENT

The theoretical models of the solar cell are derived from solid state physics theory. The starting point for the derivation of basic solar cell equation is pn junction. From the dc model of solar cell

$$I_L = I_{ph} - I_0 \{ \exp[K_0(V + I_L R_{sc})] - 1 \} - V/R_{sh}$$

This equation can be solved to obtain V as

$$V = R_{sh} [I_{ph} - I_L - I_0 \{ \exp[e(V + I_L R_{sc})/AKT] - 1 \}]$$

where

$$K_0 = e/AKT$$

The V-I and the P-V characteristics for the above shown model are obtained using the following equations in MATLAB.

$$\begin{aligned} i_1 &= I_{ph}/I_{sc} \\ i_0 &= I_0/I_{sc} \\ r_s &= [\log((i_1 + i_0 - 1)/i_0)] \\ r_p &= 1/[i_1 + i_0 - (i_0 * \exp(x))] \end{aligned}$$

where,

$$x = K_0 * V_{oc}$$

By substituting the values of R_{sh} and R_{sc} and other constants from datasheet values, the values of voltage and current are obtained.

III. ARTIFICIAL NEURAL NETWORKS FOR MPPT

Neural Network is a machine that is designed to model the way to which the brain performs a particular task. The network is implemented using electronic components or is simulated in software using digital computer. Neural networks perform usual computations through process of learning. Neural network is a massively parallel distributed processor made up of simple processing units, which has a natural property of storing experimental knowledge and making it available for use. The primary significance of the neural network is the ability of the network to learn from its environments and to improve its performance through learning. It learns about its environment through an interactive process of adjustments applied to its synaptic weights and biases. The network becomes more knowledgeable about its environment after each iteration of learning process.

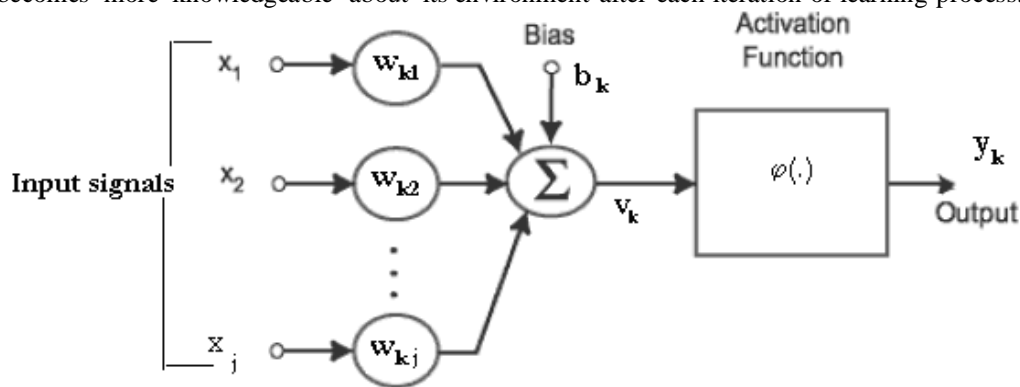


Fig.1 Model of a neuron

A neuron is a information processing unit that is basic unit to the operation of neural network. The block diagram given above shows the model of a neuron. An artificial neuron consists of various inputs, much like the biological neuron. A signal x_j at the input of synapse j connected to neuron k is multiplied by the synaptic weight w_{kj} . The first subscript refers to the neuron in question and the second subscript refers to the input end of the synapse to which the weight refers. The weight w_{kj} is positive if the associated synapse is excitatory; it is negative if the synapse is inhibitory. When inputs are fed to the neuron, the summation unit will initially find the Net-value. When inputs are fed to the neuron, the summation unit will initially find the Net-value. The Net value is the product of input value and corresponding connection weight. i.e, input value $x(i)$ of each input to the neuron is multiplied with the associated connection weight $w(i)$. In simplest case, these products are summed and fed to the transfer function. Also, a neuron has a bias value, which affects the net value. A bias of a neuron is set to a random value, when the network is initialized. The connection weights and bias of all neurons in the network will be changed (other than neurons in the input layer) during training phase. The model of a neuron also includes an externally applied threshold θ_k that has the effect of lowering the net input of the activation function. An activation function is for limiting the amplitude of the output of a neuron. The activation is a linear or non- linear function. A particular activation is chosen to satisfy some specification of the problem that the neuron is attempting to solve. The activation function is denoted by $\phi(.)$. The most commonly used activation functions are Simple thresholding (hard limiting), Squashing function (sigmoid), Gaussian function and Linear function.

The procedure used to perform the learning process is known as learning algorithm. Its main function is to modify the synaptic weights of the network in an orderly fashion to attain desired design objective. Learning methods in neural networks are classified into three types as Supervised learning, Unsupervised learning and Reinforced learning.

A multilayer feed forward network consists of a set of source nodes that constitute the input layer, one or more hidden layers of computation nodes and an output layer. The input signal propagates through the network in a forward direction, on a layer-by-layer basis. These neural networks are commonly referred to as multilayer perceptrons. The function of the hidden neurons is to intervene between the external input and the network output. By adding one or more hidden layers, the network is enabled to extract higher-order statistics, for the network

acquires a global perspective despite its local connectivity by virtue of the extra set of synaptic connections and the extra dimensions of neural interactions. The ability of hidden neurons to extract higher-order statistics is particularly valuable when the size of the input layer is large. The source nodes in the input layer of the network supply respective elements of the activation pattern (input vector), which constitute the input signals applied to the neurons (computational nodes) in the second layer (i.e. the first hidden layer). The output signals of the second layer are used as input to the third layer, and so on for the rest of the network. Typically, the neurons in each layer of the network have as their inputs the output signals of the preceding layer only. The set of output signals of the neurons in the output (final) layer of the network which constitutes the overall response of the network to the activation pattern supplied by the source nodes in the input first layer.

The error back – propagation process consists of two passes through the different layers of the network: a forward pass and a backward pass. In the forward pass, an activity pattern (input vector) is applied to the source nodes of the network and its effect propagates through the network, layer by layer. Finally, a set of outputs is produced as the actual response of the network. During the forward pass the synaptic weights of the network are all fixed. During the backward pass, on the other hand, the synaptic weights are all adjusted in accordance with the error-correction rule. Specifically, the actual response of the network is subtracted from a desired response to produce an error signal. This error signal is then propagated backward through the network against the directions of the synaptic connections. The synaptic weights are adjusted so as to make the actual response of the network move closer to the desired response. The learning process performed with the algorithm is called the back-propagation learning.

The solar module considered is the KL solar company's 60W PV module. The solar module is able to output a maximum power of 60W. The specifications of the solar module as supplied by the manufacturer's data sheet are as shown in Table 2.

Table.1 Specifications of 60 W PV panel

At Temperature = 25°C		
Open circuit voltage	Voc	21.0 V
Short circuit current	Isc	3.74A
Voltage at max.power	Vm	17.1V
Current at max power	Im	3.5A
Maximum power	Pm	60.0W

For implementation of neural networks to the given PV panel, a three layered feed forward network is created using newff command. The network is created with two neurons in the input layer, 50 neurons in the hidden layer and one neuron in the output layer. The two inputs to the network are the insolation and the temperature. Totally 23 sets of input data samples are given to the network for training. The targets of the network are specified to be the values of voltages that are obtained from modeling for a particular input set among training sets. The activation functions used at the input and output layers are tansig and purelin functions respectively. The tansig activation function calculates its output according to

$$\text{tansig}(n) = 2/(1+\exp(-2*n))-1$$

The purelin activation function calculates its output according to

$$\text{purelin}(n) = n$$

The basic structure of the neural network is shown below in Fig 2

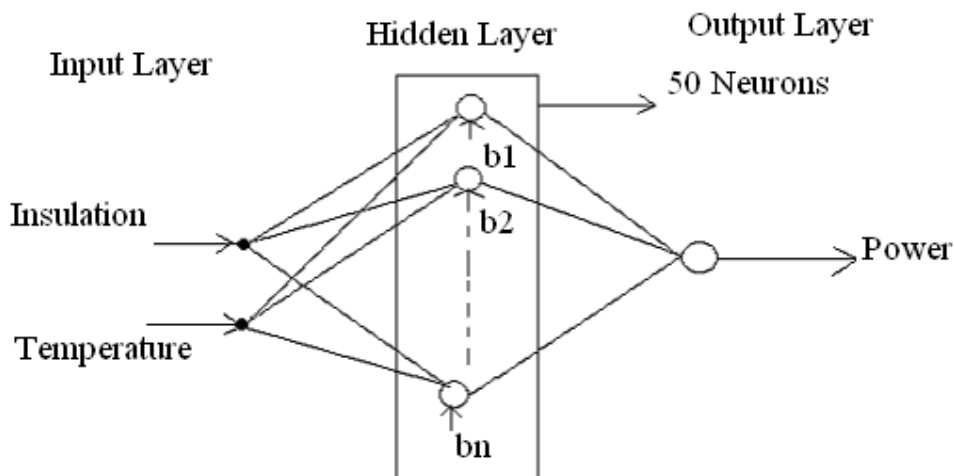


Fig.2 Structure of control network

The error criterion considered for training is Mean Square Error. The learning function is taken to be `trainlm`. “Trainlm” is a network training function that updates weight and bias values according to Levenberg-Marquardt (LM) optimization, one of the most popularly used algorithms which is a type of numerical optimization technique that has many advantages like it requires less number of data for training the network, achieves accurate results and produces accurate results even if the system is not completely controllable and observable. The error back propagation learning is implemented for updating the weights of the network in order to minimize the mean square error. This algorithm consists of two passes, forward pass and backward pass. In the forward pass, 23 pairs of insolation and temperature are given to the network. As these are given all at a time, this type of training is called Batch training. These inputs propagate through the network layer-by-layer and the output voltage is generated. These inputs are used for computing the gradient and updating the network weights and biases. Now the output voltage is compared with the corresponding target value and then the error which is the difference between these two values is propagated through the network in the backward pass. During this process, the weights of the network are updated recursively.

Training occurs according to the `trainlm`'s training parameters. The performance goal is specified as 0.15. The maximum number of epochs are set to 30. This is done on trial basis. When the epoch number is mentioned to be low than this number, training is stopped without reaching the performance. Maximum validation failures is kept as 5. Mu decrease factor is mentioned as 0.01. The time interval for showing the Epoch's progress is set to 50. This shows the training curve for every 50 epochs. The other parameters are set to their default values. The trained network thus obtained is tested using 9 sets of data. A neural network simulink block is generated using `gensim(net)` command. After the network is generated, 9 sets of insolation and temperature are given to the network for validating the network. Each data set is given to the network at different instances of time with an interval of 1 sec. The network is simulated. The curves for voltage, current and power are obtained.

The algorithm for the implementation has the following steps:

- Step-1: Construct the network and initialize the synaptic weights with random values.
- Step-2: Apply the input sets to the network
- Step-3: Set the parameters of the network and calculate the corresponding output values by training the network.
- Step-4: Compare the actual outputs with the desired outputs and determine a measure of the error.
- Step-5: Determine the amount by which each weight is to be changed and make corrections to each weight..
- Step-6: Repeat Step-3 to Step-4 until the error for the sets in the training set is reduced to an acceptable value.
- Step-7: Validate the so formed network using testing sets

IV. RESULTS

The trained network is validated with 9 sets of insolation and temperature conditions. The corresponding voltages and powers are tabulated.

Table.2 Voltage and power values from ANN

Insolation (W/m²)	Temperature (°C)	Target Voltage obtained from Modeling(V)	Power obtained from Modeling(W)	Voltage obtained from ANN based MPPT Method(V)	Power obtained from ANN based MPPT Method(W)
960	24	16.96	56.84	16.9698	56.8380
175	5	13.1	7.892	12.04	7.6394
1000	25	17.1	59.9159	17.0862	59.8885
700	42	18.52	44.82	19.2	44.8192
965	24.125	16.96	57.21	16.98	57.2030
930	12	16.6	48.64	16.5	48.6104
995	24.87	17.14	59.41	17.0674	59.4013
820	40	18.46	53.14	18.05	52.9013
980	24.5	17.08	58.31	17.02	58.3007

V. CONCLUSION

The operational requirement of any electrical device is that it has to cater maximum available power to the utility in which case it can be said that the device is operating with the maximum efficiency, because the power output of the device will be maximum. The solar photovoltaic arrays come in various power outputs to meet the load requirement. However, maximization of power from a solar photovoltaic array is of special interest as the efficiency of the solar array is very low. Methodology adopted and implementation of ANN for deriving MPP of considered 60 W solar panel is given. The controller action is implemented in MATLAB Simulink environment. The obtained maximum power and voltage values after training show that the power and voltage values obtained are very close to target values and hence ANN is one of the powerful tools in obtaining maximum power point of solar PV panels.

VI. ACKNOWLEDGEMENTS

I acknowledge the almighty for his blessings and my family for their constant support and also my guides and students for their contribution towards this work.



REFERENCES

- [1] Rauschenbach, H.S., 1980. Solar Cell Array Design Handbook. The principles and technology of photovoltaic energy conversion. New York; VanNostrand.
- [2] Chomsuwan, K., P. Prisuwan and V. Monyakul, 2002. Photovoltaic grid connected inverter using two-switch buck-converter. In: Conf. Rec. 29th Photovoltaic Specialists Conf., pp: 1527-1530. J.R. Mitchell, "Comments on bode compensator design," IEEE Trans. Automat. Contr., vol. AC-22, pp.869-870, Oct.1977.
- [3] Green, M.A., 1982. Solar Cells: Operating principles technology and system applications. Englewood cliffs, NJ; Prentice Hall.
- [4] R. Kiranmayi, K. Vijay Kumar Reddy, M. Vijaya Kumar, "Modeling and a MPPT methods for Solar Cells", Medwell Journal of Engineering and Applied Sciences 3 (1): 137-142, 2008.

BIOGRAPHY



Mrs R.Kiranmayi, Graduated in Electrical and Electronics Engineering in the year 1993 from Jawaharlal Nehru Technological University, Anantapur and Post Graduated in Electrical Power Systems in the year 1995 from the same university. She is presently working as Associate Professor in Electrical Engineering on deputation at JNTUH college of Engineering, Hyderabad.