of

Modeling and Control of a Buck DC-DC Converter Based on Artificial Neural Network

Hussain K. Sayed

Abstract - The dc-dc converters are highly efficient tools to supply power to different systems, they have a nonlinear behavior and variations at their main parameters could affect their stability. This paper proposes the Artificial Neural network (ANN) methodology to control the output voltage of a dc- dc buck converters. The mathematical model and simulation of the nonlinear dc-dc buck converter is represented in this work .The conventional PID controller of a dc-dc converter is illustrated as an example of comparison. The proposed technique is evaluated on a dc-dc buck converter under different operating conditions (no-load and full load condition) by using MATLAB Simulink software. The results proved the fast response and robustness of the proposed neural controller with no significant additional cost were compared with traditional PID controller technique.

Index Terms- Power electronics, Buck converter, PID control, neural network, modeling and analysis, Continuous conduction mode (CCM), discontinuous conduction mode (DCM), NARMA –L2

1 Introduction

Several new DC-DC converter topologies are being developed, having high efficiency and simple control scheme, to meet an increasing demand. These include modeling and analysis, improving the steady state and dynamic performance etc. Therefore the development of power electronics have become the dc-dc converters in efficient tools used to supply power and as efficient regulator to feed different electronics systems. The converters have several advantages compared with conventional linear regulator based in voltage or current divisions, which are inefficient due to its output is limited to voltage lower than the input, and that they present low power density [1]. Instead the switching regulators are efficient turning energy, due to the low losses in their switching states and they can work in high frequency, which improvement the dynamic behavior of the converter [1]. In addition the converters permit to obtain output upper than its input. The converter has a nonlinear behavior and its stability affected by its parameters, due to the uncertainly present in these, so that difficult procedures have to be made approaching its design parameters to obtain a mathematic model that represent the converter .General PI and PID controllers are widely used for dc-dc converter control applications. But it does not give satisfactory results when control parameters, loading conditions and dc-dc buck converter itself art changed. The artificial neural networks (ANN) can be designed without the exact model of the system. This approach of ANN design guarantees the stable operation even if there is a change in the system parameters. For ANN, it is sufficient to understand the general behavior of the system. Such as ANN was designed and analyzed for dc-to-dc converters. control scheme; in system modeling, sometimes it is required to identify some parameters, and this can be achieved by using artificial neural networks (ANN) [2]. Artificial neural systems can be defined as cellular systems which have the capability of acquiring, storing, and utilizing experiential knowledge [3]. Followed by the study and design of the ANN controllers proposed in this document, which are the optimal control that permits to obtain an optimal model of the plant, and finally results of the different strategies were compared with one of them was chosen for practical implementation. Thus the main reason of using neural networks (ANN) is due to their ability to approach any function, linear or not [4].All the results obtained are showed in digital simulations and they show different characteristics and responses of each controller.

2 DC - DC Buck Converter Principles

DC-DC Converters are power electronic systems that convert one level of electrical voltage into another level by a switching action. The dc-dc Buck converter can have two distinct modes of operation: Continuous conduction mode (CCM) and discontinuous conduction mode (DCM). In practice, a converter may operate in both modes, which have significantly different characteristics. Therefore, a converter and its control should be designed based on both modes of operation. However, for this work we only consider the dc-dc converters operated in CCM.

ISSN 2229-5518

Circuit Operation of figure 1: When the switch (d) is ON for a time duration (d*T)
Where: d = switching status (0 for OFF statue or 1 for ON statue).
T = duration of operation of switching device (TON or TOFF).
The switch conducts the inductor current and the diode becomes reverse biased. This results in a positive voltage VL = Vg – Vo across the inductor.
Where: VL is inductor voltage.
Vg is dc input voltage.
Vo is output voltage.

This voltage causes a linear increase in the inductor current (i_L). When the switch is turned OFF, $i_L$because of the inductive energy storage, continues to flow. This current now flows through the diode, and VL = -Vo for a time duration (1-d)*T until the switch is turned on again. This converter gives an output voltage v0 smaller than the input voltage vg.

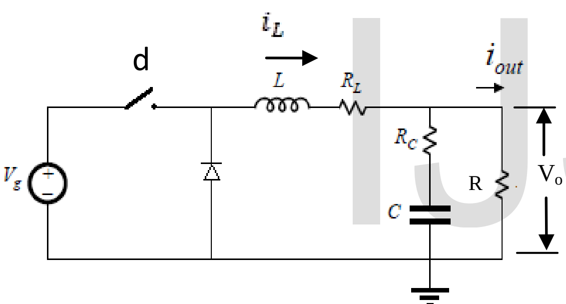


Figure (1) dc-dc Buck converter circuit

3 Mathematical Modeling and Simulation of a DC-DC Buck Converter

For the model consist of RL and Rc. These topologies are nonisolated, that is, the input and output voltages share a common ground. There are, however, isolated derivations of these non-isolated topologies. The power supply topology refers to how the switches, output inductor, and output capacitor are connected. Each topology has unique properties. These properties include the steady-state voltage conversion ratios, the nature of the input and output currents, and the character of the output voltage ripple. Another important property is the frequency response of the duty-cycle-to output- voltage transfer function. The most common and probably the simplest power stage topology is the buck power stage, sometimes called a step-down power stage. [5]

By using Kirchhoff's Voltage Law (KVL) and Kirchhoff's Current Law (KCL), System equations are obtained as shown below and these laws can be applied on the other dc – dc converter (boost, buck-boost and cuk) converters.

$$\frac{di_L}{dt} = \frac{1}{L}\left(V_g .d - i_L R_L - v_o\right) \quad \ldots\ldots\ (1)$$

$$\frac{dv_C}{dt} = \frac{1}{C}\left(i_L - i_{out}\right) \quad \ldots\ldots\ldots\ldots\ldots\ (2)$$

$$v_o = v_c + R_c\ (i_L - i_{out}) \quad \ldots\ldots\ldots\ldots\ (3)$$

Where: $i_{out} = \frac{V_o}{R}$

The open loop dc-dc buck converter with simulation is shown in figure (2). The output performance and output current of the system under no-load and full load condition can be shown in figure (3) and figure (4). Parameters used in the simulation studies are given below:
Vg = 12 volt , L = 1 µH , RL = 80 mΩ , C = 376 µF , Rc = 5 mΩ , d = 1 (duty cycle) , R = 28 Ω (load). [5]

4 PID Controller

A proportional – integral - derivative controller (PID controller) is a generic control loop feedback mechanism widely used in industrial control systems. A PID
Controller attempts to correct the error between a measured process variable and a desired set point by calculating and then outputting a corrective action that can adjust the process accordingly.
The PID controller calculation (algorithm) involves three separate parameters; Proportional, Integral and Derivative values. The Proportional value determines the reaction to the current error, the Integral determines the reaction based on the sum of recent errors and the Derivative determines the reaction to the rate at which the error has been changing. The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve or the power supply of a heating element. By "tuning" the three constants in the PID controller algorithm the PID can provide control action designed for specific process requirements. The response of the controller can be described in terms of the responsiveness of the controller to an error, the degree to which the controller overshoots the setpoint and the degree of system oscillation. Note that the use of the PID algorithm for control does not guarantee optimal control of the system. First estimation is the equivalent of the proportional action of a PID controller, but PID controllers

do not have the ability to learn and must be set up correctly. Selecting the correct gains for effective control is known as tuning the controller. [5]

Traditionally, these parameters are determined by a trial and error approach.

Manual tuning of PID controller is very tedious, time consuming and laborious to implement, especially where the performance of the controller mainly depends on the experiences of design engineers [6].

The modeling of a dc-dc buck converter with PID controller is shown in figure (5). The output performance and output current of the system under no-load and full load condition under PID controller can be shown in figure (6) and figure (7). The simulation model for a dc-dc buck converter for load change with PID controller is shown in the figure (5). By setting the proportional gain Kp to 8, Ki to 280, and Kd to 0.001. These parameters are determined by a trial and error approach.

5 NEURAL CONTROLLER

The model of an artificial neuron that closely matches a biological neuron is given by an op-amp summer like configuration shown in figure (8).

Where x1, x2, x3… are input signals, each of the input signal flows through a gain called synaptic weight. The weight can be positive (excitory) or negative (Inhibitory) corresponding, respectively, to acceleration or inhibition [7].

The summing nodes accumulate all the input weighted signals and then pass to the output through the transfer function which is usually nonlinear. The transfer function can be step or threshold type, signum type, or linear threshold type. The transfer function can also be nonlinear continuously varying type, such as sigmoid, inverse-tan, hyperbolic, or Gaussian type. The sigmoidal transfer function is most commonly used, and it is given by

$$Y = \frac{1}{1+e^{-\alpha x}} \quad \text{............ (4)}$$

Where α is the coefficient or gain which adjusts the slope of the function. With high gain, this function approaches a step function. The sigmoidal function is nonlinear, monotonic, differentiable, and has the largest incremental gain at zero signal, and these properties are of particular interest.

In general, neural networks can be classified as feedforward and feedback types depending on the interconnection of the neurons. At present, the majority of the problems use feedforward architecture, and it is of direct relevance to power electronics and motion control applications.

Figure (9) shows the structure of a feedforward multiplayer network with two input and two output signals. The topology is based on Perceptron which was proposed by Rosenblatt in 1958. The circles represent neurons and the dots in the connections represent the weights.

The network has three layers, defined as input layer (a), hidden layer (b), and output layer (c). The hidden layer functions as a connection between the input and the output layers. The input and output layers have neurons equal to the respective number of signals. The input layer neurons do not have transfer functions, but there are scale factors, as shown, to normalize the input signals. The number of hidden layers and the number of neurons in each hidden layer depend on the network design considerations. The input layer transmits the signals to the hidden layer, and the hidden layer, in turn, transmits the signals to the output layer, as shown. The network can be fully connected or partially connected.

5-1 Back Propagation Training

Back-Propagation training algorithm is most commonly used in a feedforward neural networks as mentioned before.

For this reason, a feedforward network is often defined as “back-prop” network.

In the beginning, the network is assigned random positive and negative weights. For a given input signal pattern, step by step calculations are made in the forward direction to derive the output pattern. A cost functional given by the squared difference between the net output and the desired net output for the set of input patterns is generated and this is minimized by gradient descent method altering the weights one at time starting from the output layer. The equations for the output of a single processing unit are given as:

$$Net_j^p = \sum_{i=1}^{N} W_{ij} X_i \quad \text{.......... (5)}$$

$$Y_j^p = fj\left(Net_j^p\right) \quad \text{............... (6)}$$

Where j is the processing unit under consideration, p is the input pattern number X_i is the output of the i^{th} neuron connected to the j^{th} neuron, W_{ij} is the connection weight between the i^{th} and j^{th} neurons. Net_j^p is the output of the summing node, i.e., the j^{th} neuron activation signal, N is the number of the neurons feeding the j^{th} neuron, f j is the nonlinear differentiable transfer function (usually sigmoid), and Y_j^p is the output of the corresponding neuron. For the input pattern p, the squared output error for all the output layer neurons of the network is given as

$$E_p = \frac{1}{2}(d^p - y^p)^2 = \frac{1}{2}\sum_{j=1}^{5}\left(d_j^p - y_j^p\right)^{2!} \quad \text{... (7)}$$

Where d_j^p is the desired output of the j^{th} neuron in the output layer y_j^p, is the corresponding actual output, S is the dimension of the output vector $\mathbf{y}^{\mathbf{p}}$ is the actual net output vector, and $\mathbf{d}^{\mathbf{p}}$ is the corresponding desired output vector.
The total squared error E for the set of P patterns is then given by

$$E = \frac{1}{2}\sum_{p=1}^{p} E_p = \frac{1}{2}\sum_{p=1}^{p} \sum_{j=1}^{s}\left(d_j^p - y_j^p\right)^{2!} \quad \text{... (8)}$$

The weights are changed to reduce the cost functional E in a minimum value by gradient descent method, as mentioned. The weight update equation is then given as:

$$W_{ij}(t+1) = W_{ij}(t)^{\eta}\left[\frac{\delta E_p}{\delta W_{ij}(t)}\right] \quad \text{..... (9)}$$

Where η is the learning rate, Wi j (t +1) is the new weight and Wi j (t) is the old weight. The weights are updated for all the P training patterns. Sufficient learning is achieved when the total error E summed over the patterns falls below a prescribed threshold value. The iterative process propagates the error back-propagation [7, 8, 9].

5-2 NARMA – L2 NEURAL CONTROLLER

In this work, the NARMA –L2 architecture is applied with the aid of the Neural Network Toolbox of MATLAB software. The identification can be summarized by the flowing steps:
a- The first step in using feedback linearization (or NARMA-L2 control) is to identify the system to be controlled.
Neural network is trained to represent the forward dynamics of the system. One standard model that has been used to represent general discrete-time nonlinear systems is the NARMA-L2 model [10]:
y(k +d) = N[y(k),y(k – 1),…, y(k –n+ 1),
u(k),u(k– 1),…,u(k–n+ 1)] ……(10)
where u(k)is the system input, and y(k)is the system output and k ,d, n are integral number and N is the function of the output system after identification.
b- The next step is to make the output system follows some reference trajectory
by developing a nonlinear controller of the form:
y(k +d) = yr(k +d) …… (11)
u(k) = G[y(k),y(k – 1),…, y(k –n+ 1), yr(k +d),u(k– 1),…,u(k–m+ 1)] ….. (12)
The problem with using this controller is:
Training neural network to minimize mean square error needs to use dynamic back propagation which quite slows [11].
One solution is to use approximate models to represent the system. The controller used in this section is based on the NARMA-L2 approximate model:
y^(k+d)=f[y(k),y(k–1),…,y(k–n+1),u(k–1),…,u(k–m+1)]+ g[y(k),y(k –1),…,y(k–n+1),u(k– 1) ,…,u(k–m+ 1)]u(k)
….. (13)
Where the next controller input is not contained inside the nonlinearity. The advantage of this form is that controlled input make the system output follows the reference equation (3). The resulting controller is:

$$u(k) = \frac{y_r(k+1) - f[y(k),y(k-1),.y(k-n+1)..u(k-n+1)]}{g[y(k),..y(k-n+1),u(k-n+1)]} \quad \text{...... (14)}$$

Using this equation directly can cause realization problems, because must determine the control input based on the output at the same time, i.e:
y(k+d)=f[y(k),y(k–1),..,y(k n+1),u(k),u(k–1),…,u(k–n+1)]+ g[y(k),…,y(k–n+1),u(k),…,u(k–n+1)] u(k+1) …… (15)
Figure (11) is referred to block diagram of the proposed dc-dc buck converter with NARMA-L2 controller. The output performance and inductance current of the system under no-load and full load condition under NARMA-L2 controller can be shown in figure (12) and figure (13).

6 Simulation Result

The response for output voltage and output current of the system with PID controller are shown in figure (6) and figure (7), we note that the overshoot of the terminal voltage is (12.5%), the rise time is (1.5*10-5 sec) and the settling time is (0.2 sec).
Figure (11) illustrates simulation system of a dc-dc buck converter with ANN. The controlling steps and output response is discussed in the following. The graphs shown in figure (12) and figure (13) show the performance results of the ANN and the response for the terminal voltage and output current, we note that the overshoot is (0 volt), the rise time and the settling time is (9.5*10-5 sec). Figure (14) and figure (15) shows the comparative of the output voltage (Vo) and the output current (iload) between PID & ANN.

7 Conclusion

In a DC-DC Converter application, it is desired to obtain a constant output voltage Vo(t)=Vo , in spite of disturbances in Vg(t) and ioad(t) , and in spite of disturbance in the converter circuit element values. Controlled DC-DC voltage static converter using the neural network has been investigated. Simulation results are satisfactory in term of stability. It can be shown that higher dynamic and adaptable regulation can be obtained. It is noted that the adjustment of the neuronal network is more obvious than that PID correctors. The proposed technique shows that the ANN is more effective than the PID. The improved damping performance by the neurocontrollers allows the system to be operated closer to its stability limit during steady state, and still remain stable after severe disturbances. The neural network also has a fast processing

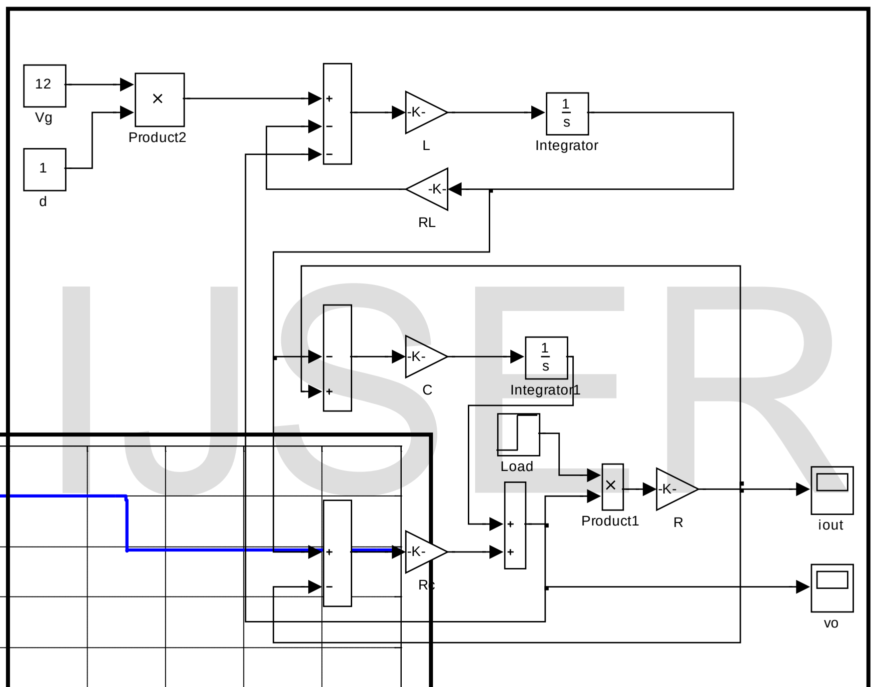
ISSN 2229-5518
speed than PID controller which gives faster rise time, no overshoot and zero steady state error.

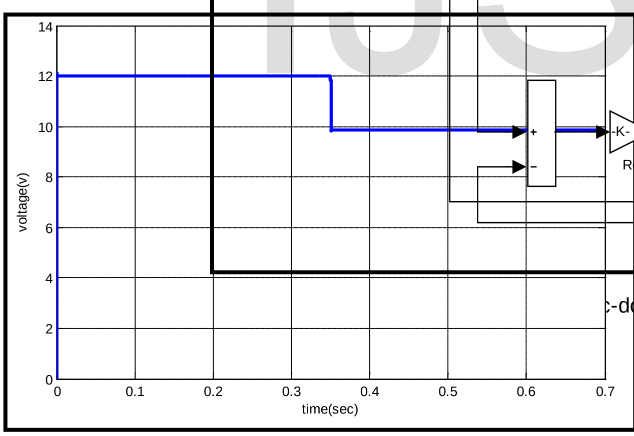
8 References

[1] Muhammad, R., "Power Electronics handbook", Academic Press Series in Engineering, 2001, pp. 211.
[2] Anas N. Al-Rabadi and Othman M.K. Alsmadi, "Model Reduction-Based Control of the Buck Converter Using Linear Matrix Inequality and Neural Networks", Proceedings of the International MultiConference of Engineers and Computer Scientists , Vol II IMECS 2009, March 18 - 20, 2009, Hong Kong.
[3] J. M. Zurada, "Artificial Neural Systems", West Publishing Company, New York, 1992.
[4] J-N. Marie-Francoise and H. Gualous, A. Berthon, " dc to dc Converter with Neural Network Control for ON-Board Electrical Energy Management", UTBM-L2ES, University Franche-Comte rue Thierry MIEG, F90010 Belfort, France.
[5] Muhamad Farhan Bin Umar Baki, "Modeling and Control of dc to dc Converter (Buck) " ,Faculty of Electrical & Electronic Engineering University Malaysia Pahang (UMP), MAY- 2008.
[6] Ziegler, J.G., and Nichols, "Optimum settlings for automatic controllers", Trans. On ASME., vol. 64, pp.759-768 ,N.B. 1942.
[7] B. K. Bose, "Expert System, Fuzzy logic, and neural network Applications in power Electronics and motion control", Proceedings of the IEEE, Vol. 82, NO. 8, 1303-1321. World Academy of Science, Engineering and Technology 46 2008 899 August 1994.
[8] A. G. Aissaoui, M. Abid, H. Abid, And A. Tahour," A Neural Controller for Synchronous Machine" ,World Academy of Science, Engineering and Technology 46,pp 893- 900,2008.
[9] M. Chow, R. N. Sharpe and J. C. Hung, "On the application and design of artificial neural networks for motor fault detection – Part I", IEEE Transaction on Industry Electronics, Vol. 40, NO. 2, 181- 188,April 1993.
[10] Becerra ,V. M., "Process Control Simulation using Simulink", Sept. 2000, Updated for Matlab 6.5 – June,2003.
[11] Mehpotra ,K. and Mohan ,C. K., " Elements of Artificial Neural Networks", proceeding Japan text book pp.41 ,2001.

ISSN 2229-5518

Figure (3) Output voltage (vo)





c-dc buck converter simulation

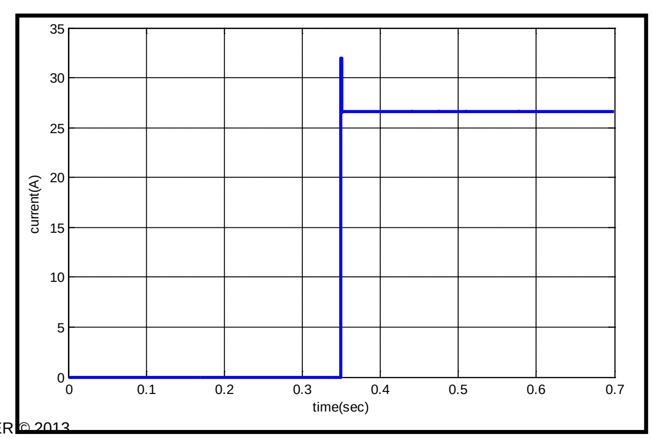


Figure (3) Output voltage (v_o)

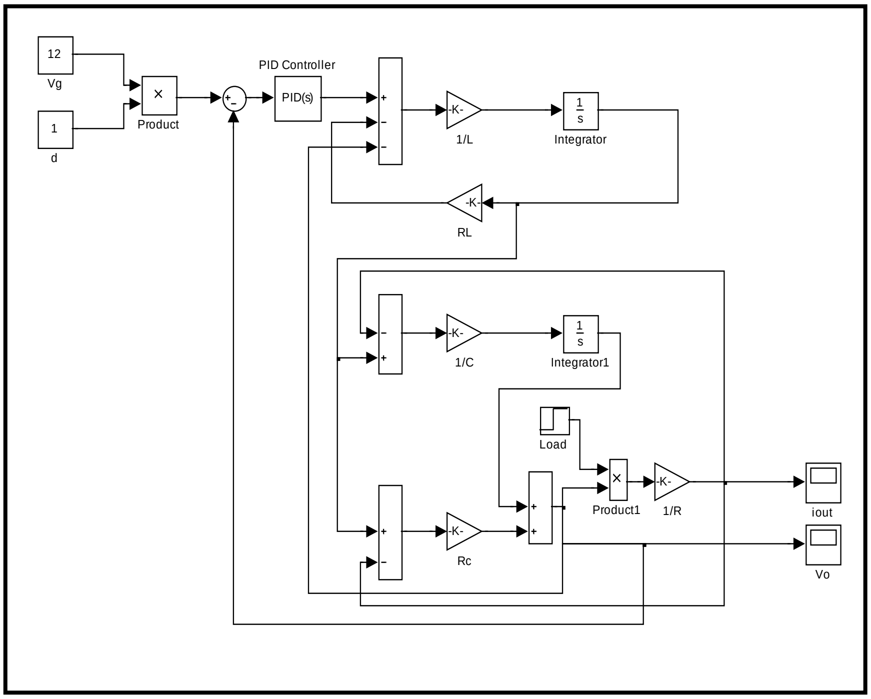
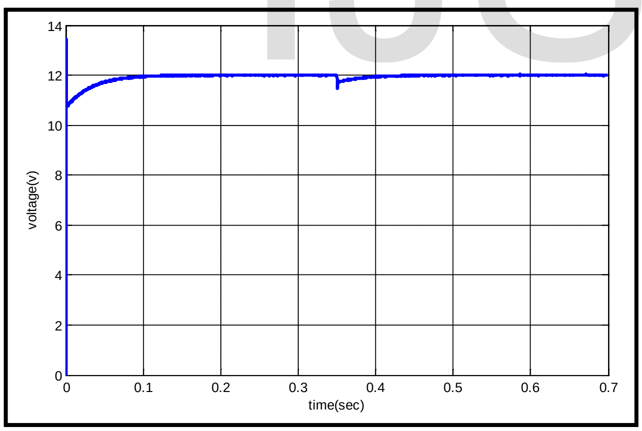


Figure (5) PID controller of a dc-dc buck converter simulation



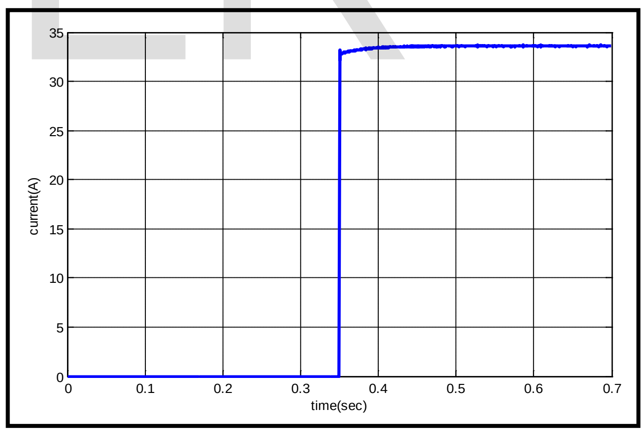
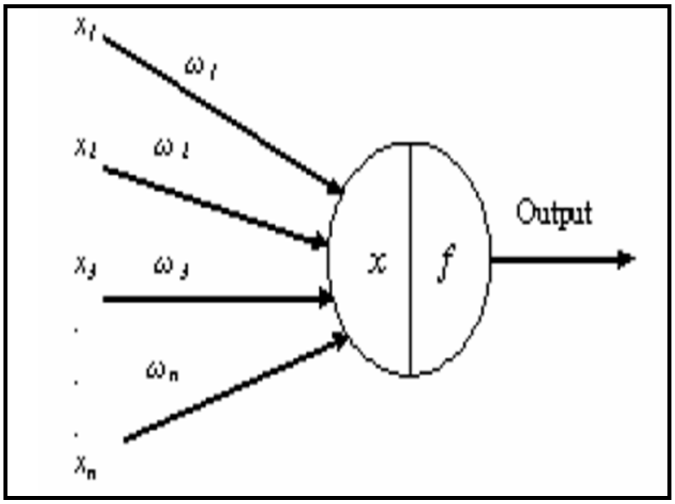


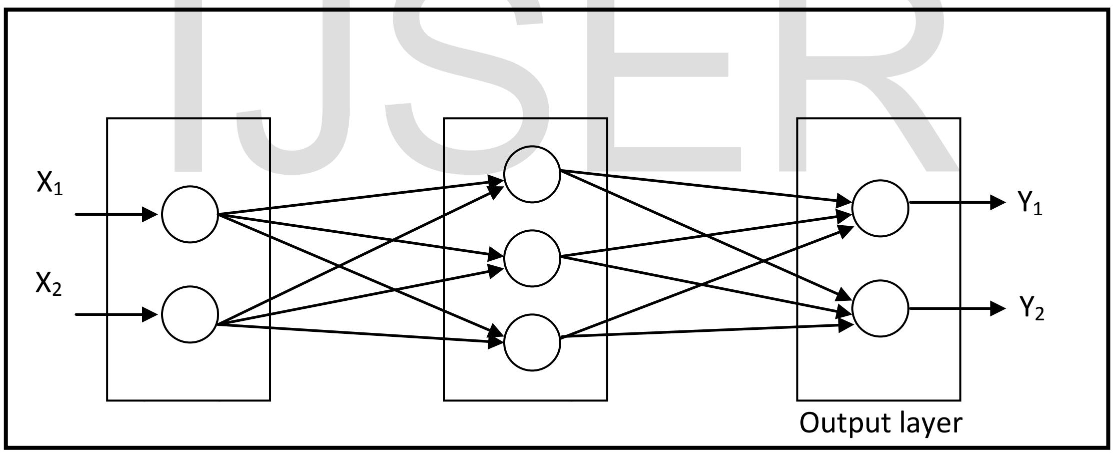
Figure (7) Load current (iout) with PID controller

Figure (6) Output voltage (vo) with PID controller

ISSN 2229-5518



Figure(8) Structure of an artificial neuron



Figure(9) Structure of a feedforward multiplayer network

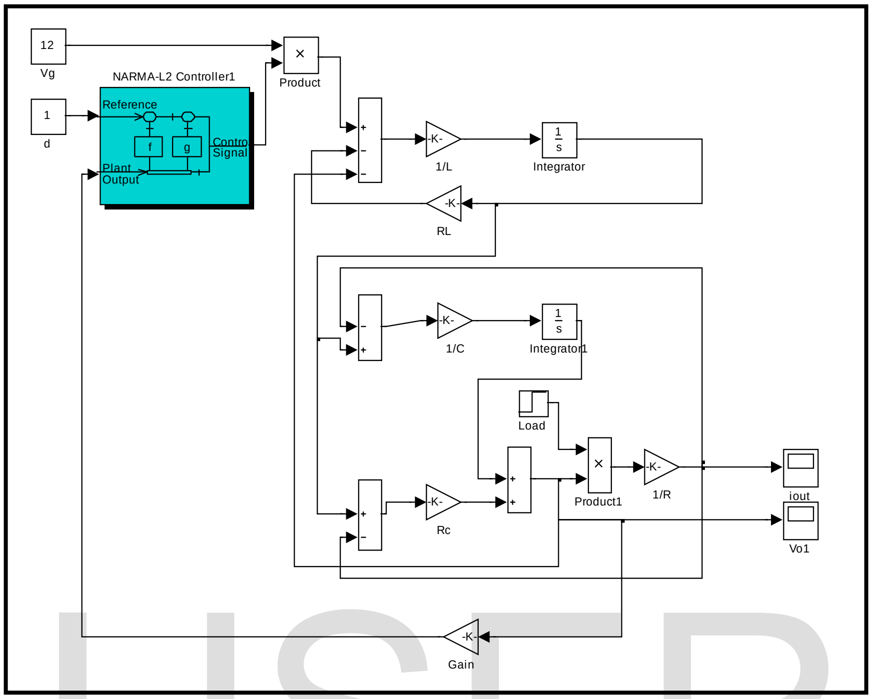


Figure (11) ANN controller of a dc-dc buck converter simulation

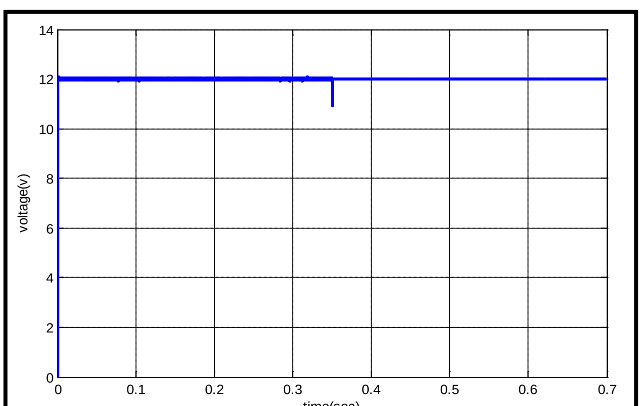


Figure (12) Output voltage (vo) with ANN controller

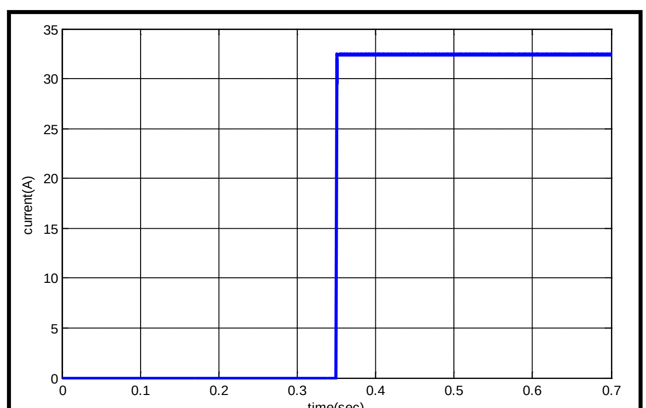


Figure (13) Load current (iout) with ANN controller

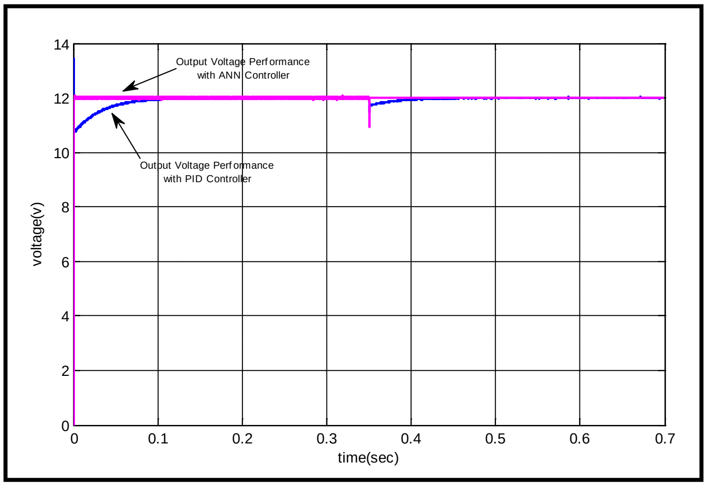


Figure (14) Comparative of the Output Voltage (Vo) between

PID & ANN Controller

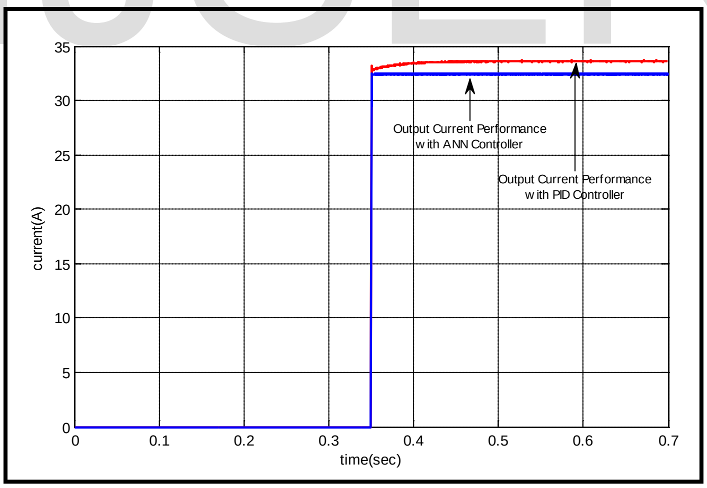


Figure (15) Comparative of the Output Current (iload) between

PID & ANN Controller