Journal of Science and Technology (JST)
Volume 2, Issue 6, Nov - December 2017, PP 13-18
www.jst.org.in *ISSN:2456-5660*

Comparison of Various Controller Techniques for the Implementation of Hybrid Applications

Rashmi Singh[1], Shraddha Ramani[2]
[1](PG Scholar M.tech in Power Electronics, RITEE College/ CSVTU, India)
[2](Department of Electronics & Telecommunication, RITEE College/ CSVTU, India)

Abstract : *This paper provides a literature review and extensions of controllers. The designs and how the particular controller work is explained. The analog to digital converter is the key components in modern electronics system. The converter used to convert ac to dc with low losses like Buck, Boost, Bridgeless Buck-Boost used with the controllers to control output of this converters, Proportional integrator (PI), Proportional differentiator (PD), Proportional integrator and differentiator (PID) and Neural Network (NN) are used and will explained with their designs, equations and working. In this paper we address the advancement of Neural Network over Proportional integrator and differentiator.*

Keywords – *Buck-Boost Converter, PD Controller, PI Controller, PID Controller, Neural Network.*

I. INTRODUCTION

The Controllers are used to continue calculate the error value as a difference between desired value and measured value and applies a correction based on particular controller. Proportional control provides stability against small disturbances. The controllers play an important role in almost all application field, which makes it important to review the technology trends every few years. Understanding the performance of controller is the key for designers to predict the evolution of future systems, especially in multiple output SMPS.

II. CONTROLLER ARCHITECTURE

A. PD CONTROLLER: Proportional-Derivative control is useful for fast response controllers that do not need a steady-state error of 0. Proportional controllers are fast. Derivative controllers are fast. The two together is very fast. PD Controller uses a same principle to create a difference between the measured and reference value. The response of a PD controller can be characterized by two numbers: the *damping ratio* and the *natural frequency*. If the damping ratio is less than one, then the system will gradually approach the target. If the damping ratio is greater than one, the system will shoot past the target before returning. The natural frequency describes how quickly the system approaches the target. Proportional-Derivative or PD control combines proportional control and derivative control in parallel.

1. Proportional Action

Proportional action provides an instantaneous response to the control error. This is useful for improving the response of a stable system but cannot control an unstable system by itself. Additionally, the gain is the same for all frequencies leaving the system with a nonzero steady-state error.

2. Derivative Action

Derivative action acts on the derivative or rate of change of the control error. This provides a fast response, as opposed to the integral action, but cannot accommodate constant errors (i.e. the derivative of a constant, nonzero error is 0). Derivatives have a phase of +90 degrees leading to an anticipatory or predictive response. However, derivative control will produce large control signals in response to high frequency control errors such as set point changes.

DISADVANTAGE OF PD CONTROLLER

[1] PD controller are not suited for lightly damped and initially unstable systems.
[2] The set point to a PD controller (in most cases) will require insertion of a lag.

B. PI CONTROLLER: The Proportional-Integral (PI) Controller is a proportional controller and an integrator.

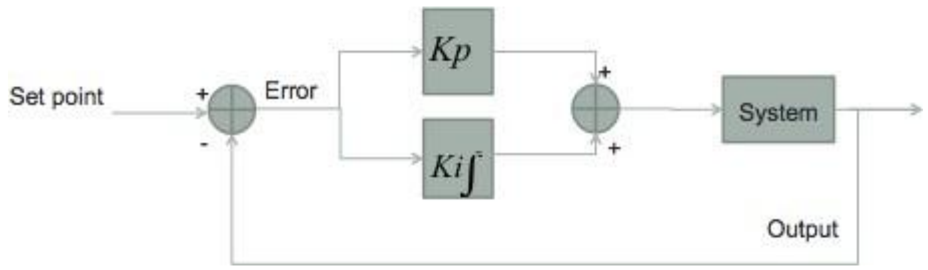

Figure 1 Block diagram of PI Controller

A PI controller is feedback control loop that calculates an error signal by taking the difference between the output of the system and the set point.The set point is the level at which we'd like to have our system running.

The MATLAB command to create a PI controller is
PI comp = @(z) tf([1 z],[1 0]);

where

z is the zeros in a PI controller.

C. PID CONTROLLER

The PID controller is the combination of three actions i.e. proportional, derivative, and integral. The proportional, integral, and derivative terms are summed to calculate the output of the PID controller.

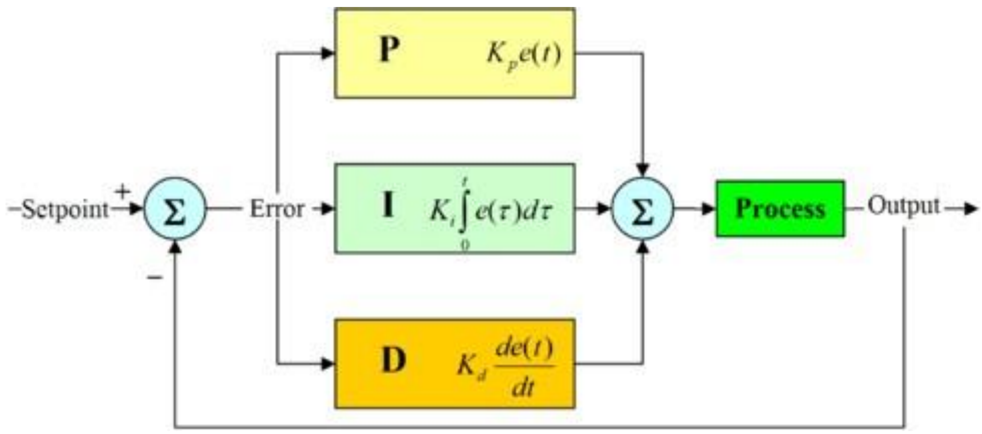

Figure 2 Block diagram of PID Controller

Where,

K_p= is the proportional gain, a tuning parameter,

K_i=is the integral gain, a tuning parameter,

K_d= is the derivative gain, a tuning parameter,

PROPORTIONAL TERM

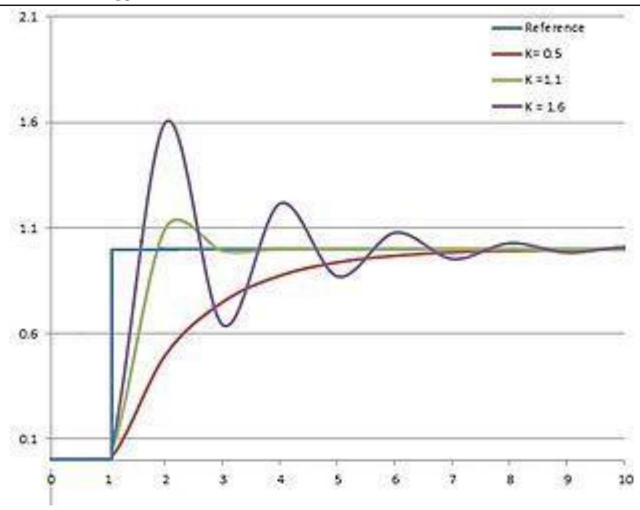

Figure 3 Plot of PV vs Time, for three values of Kp (Ki and Kd held constant)

The output produced by the proportional term is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant K_p is called the proportional gain constant. *Tuning* a control loop is the adjustment of its control parameters (proportional band/gain, integral gain/reset, derivative gain/rate) to the optimum values for the desired control response. Stability (no unbounded oscillation) is a basic requirement, but beyond that, different systems have different behavior, different applications have different requirements, and requirements may conflict with one another. PID tuning is a difficult problem, even though there are only three parameters and in principle is simple to describe, because it must satisfy complex criteria within the limitations of PID control.

LIMITATIONS OF PID CONTROLLER

While PID controllers are applicable to many control problems, and often perform satisfactorily without any improvements or only coarse tuning, they can perform poorly in some applications, and do not in general provide optimal control. The fundamental difficulty with PID control is that it is a feedback control system, with constant parameters, and no direct knowledge of the process, and thus overall performance is reactive and a compromise. While PID control is the best controller in an observer without a model of the process,[2] better performance can be obtained by overtly modeling the actor of the process without resorting to an observer.

PID controllers, when used alone, can give poor performance when the PID loop gains must be reduced so that the control system does not overshoot, oscillate or hunt about the control set point value. They also have difficulties in the presence of non-linearity's, may trade-off regulation versus response time, do not react to changing process behavior (say, the process changes after it has warmed up), and have lag in responding to large disturbances.

The most significant improvement is to incorporate feed-forward control with knowledge about the system, and using the PID only to control error. Alternatively, PIDs can be modified in more minor ways, such as by changing the parameters (either gain scheduling in different use cases or adaptively modifying them based on performance), improving measurement (higher sampling rate, precision, and accuracy, and low-pass filtering if necessary), or cascading multiple PID controllers.

III. ANALYSIS OF NEURAL NETWORK

The Neural Network has been motivated right from its inception by the recognition that the human brain computes in an entirely different way from the conventional digital computer. The brain is highly complex, non-linear, parallel computer. It has the capability to organize its structural constituent, known as neurons, so as to perform certain computations (pattern recognition, motor control).

A. NETWORK ARCHITECTURES

The manner in which the neurons of a neural network are structured is intimately linked with the learning algorithm used to train the network. There are three fundamentally different classes of network architectures:

1. SINGLE-LAYER FEEDFORWARD NETWORKS

In the layered neural network the neurons are organized in the form of layers. In the simplest form of a layered network, we have an input layer of source nodes that projects onto an output layer of neural, but not vice versa. In other words this network is strictly a feed forward or acyclic type.

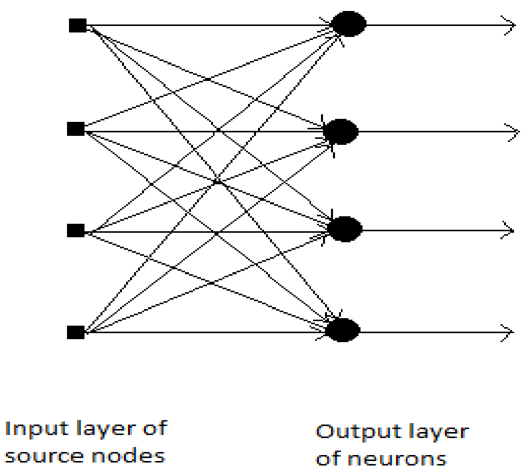

Figure 4Fedforward network with a single layer of neurons.

The figure shows the case of four nodes in both the input and output layers. We do not count the input layer of source nodes because no computation is performed there.

2. MULTILAYER FEEDFORWARD NETWORK

The second class of feed forward neural network distinguishes itself by the presence of one or more hidden layer, whose computation nodes are correspondingly called hidden neurons or hidden units. The function of hidden neurons is to intervene between the external input and the network output in some useful manner. By adding one or more hidden layer, the network is enable to extract higher-order statistics In a rather loose scene the network acquires a global perspective despite its local connectivity due to the extra set of synaptic connection and the extra dimension neural interaction. The ability of hidden neurons to extract higher-order statistics is particularly valuable when the size of the input layer is large.

The source in the input layer of the network supply respective elements of the activation pattern (input vetor), which constitute the input signals applied to the neurons (computation nodes) in the second layer (i.e. the first hidden layer).The output signals of the second layers are used as inputs 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 signal of the preceding layer only. The set of output signal of the neurons in the output (final) layer of the network constitutes the overall response of the network to the activation pattern supplied the source nodes in the input (first) layer. The architectural graph in figure 5 illustrates the layout of the multilayer feed forward neural network for the case of a single hidden layer. For brevity the network in figure 5 is referred to as a 8-4-2 network because it has 8 source nodes, four hidden neurons, and two output neurons.

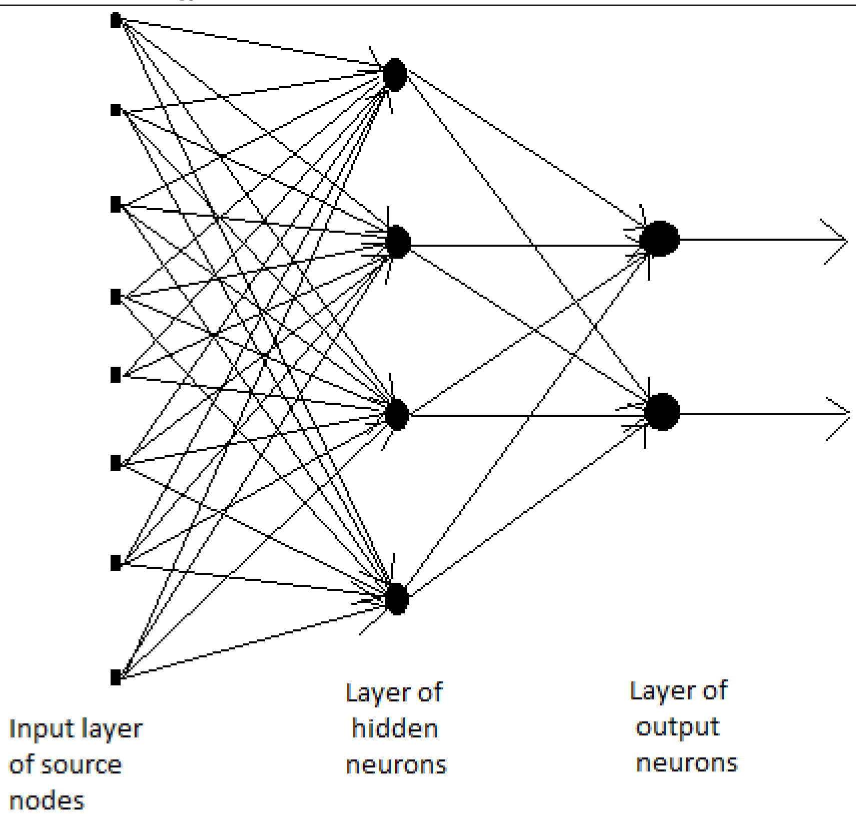

Figure 5 Fully connected feedforward network with one hidden layer and one output layer.

IV. EQUATIONS OF NEURAL NETWORK

There are mainly two categories of neural network architectures Feedforward and Feedback Network. The Feedforward consist of several layers connected as shown in fig 5. The last layer is called output or visible layer, and the other layers are called the hidden layer. In most applications one usually uses one or two hidden layers. The number of neurons per hidden layer depends on the problem considered. The usual way to specify it by trial and error, and of course the more difficult the problem, the large the required size of the hidden layer.

Let $y_i^{(l)}$ denote the output of the i^{th} neuron of layer l ($y_i^{(0)}$ denote the i^{th} input to the network). The function of the network is given by

$$y_i^{(l)} = f\left[\sum_{j=1}^{N_{l-1}} w_{ij}^{(l,l-1)} y_j^{(l-1)} + \theta_i^{(l)}\right],$$

l =1. . . . . . . L, i=1 ....N_l,

where $w_{ij}^{(l,l-1)}$ denotes the weight from neuron j of layer l-1 to neuron I of layer l, $\theta_i^{(l)}$ is the threshold of neuron I of layer l.

V. CONCLUSION

We have analyzed the performance of PD, PI, PID, and Neural Network showing their controlling action. A general network structure for networks is also presented, in which the network is consist of a number of layers connected to each other in a feedforward manner. In general, we have observed that training feedforward network is somewhat faster and less susceptible to local minima. Here we have conclude that the neural network is more advance than the proportional integral derivative controller hence it work on the principle of counting or iterations by which it minimizes error and provide fast processing.

Acknowledgements

I would like to thank Miss Shraddha Ramani for her guidance and support and I also would like to thank Head of Department Mrs.Zoonubiya Ali for preparing this paper.

REFERENCES

[1] Araki, M. *"PID Control"* (PDF).

[2] Bennett, Stuart (1993*). A history of control engineering, 1930-1955.* IET. p. p. 48. ISBN 978-0-86341-299-8.

[3] Veselý, V., Rosinová, D.: *Robust PSD Controller Design*, Editors: Fikar, M., Kvasnica, M., In Proceedings of the 18th International Conference on Process control, Tatranská Lomnica, Slovakia, 565–570, 2011.

[4] Bennett, Stuart (November 1984). *"Nicholas Minorsky and the automatic steering of ships*" IEEE Control Systems Magazine. **4** (4): 10–15. doi:10.1109/MCS.1984.1104827. ISSN 0272-1708. Archived from the original (PDF)on June 8, 2011.

[5] Minorsky, Nicolas (1922). "*Directional stability of automatically steered bodies"*. J. Amer. Soc. Naval Eng. **34** (2): 280–309. doi:10.1111/j.1559-3584.1922.tb04958.x.

[6] Bennett 1993, p. 67

[7] Bennett, Stuart (1996). *"A brief history of automatic contro*l" (PDF). IEEE Control Systems Magazine. IEEE. **16** (3): 17–25. doi:10.1109/37.506394.

[8] Bennett, Stuart (June 1986). *A history of control engineering*, 1800-1930. IET. pp. 142–148. ISBN 978-0-86341-047-5.

[9] *"A Brief Building Automation History"*. Retrieved 2011-04-04.

[10] Neuhaus, Rudolf. *"Diode Laser Locking and Linewidth Narrowing"*. Retrieved June 8, 2015.

[11] *"Position control system"* . Hacettepe University Department of Electrical and Electronics Engineering.

[12] Kebriaei, Reza; Frischkorn, Jan; Reese, Stefanie; Husmann, Tobias; Meier, Horst; Moll, Heiko; Theisen, Werner. *"Numerical modelling of powder metallurgical coatings on ring-shaped parts integrated with ring rolling"*. Material Processing Technology. **213** (1): 2015–2032.

[13] *"Introduction: PID Controller Design"*. University of Michigan.

.