On Replacing PID Controller with Deep Learning Controller for DC Motor System

Kangbeom Cheon, Jaehoon Kim, Moussa Hamadache, and Dongik Lee
School of Electronics Engineering, Kyungpook National University, Daegu, Korea
Email: ckb0120@naver.com, destiny-be@hanmail.net, assuom25@yahoo.fr, dilee@knu.ac.kr

***Abstract*—Many techniques are implemented in the industry to control the operation of different actuators on field. Within these actuators, the DC motor is a popular tool. The output of the DC motor, the speed can be controlled to drive several industrial parts. There are different type of controllers for such application including linear and nonlinear controllers, adaptive controllers, and artificial neural network controllers. This paper addresses the use of deep learning algorithm to design the controller; to explore the feasibility of applying deep learning into control problems. The proposed deep learning controller is designed by learning PID controller which is most commonly used in industry. The input/output of the PID controller are used as the learning data set for the deep learning network. ADBN (Deep Belief Network) algorithm is used to design the deep learning controller. The simulation is performed using Matlab/Simulink and the detailed results of a comparison study between the proposed deep learning controller and a PIDcontroller was conducted to demonstrate the performance and effectiveness of the proposed algorithm.**

***Index Terms*—deep learning, deep learning controller, conventional neural networks, DBN, PID controller**

I. Introduction

The machine learning algorithms can lead to significant advances in automatic control. The biggest single advance occurred nearly four decades ago with the introduction of the Expectation-Maximization (EM) algorithm for training Hidden Markov Models (HMMs) [1]. With the EM algorithm, it became possible to develop control systems for real world tasks using the richness of Gaussian mixture models (GMM) [2] to represent the relationship between HMM states and the reference input.

GMMs have a number of advantages that make them suitable for modeling the probability distributions over vectors of input features that are associated with each state of an HMM [3]. Despite all their advantages, GMMs have a serious short coming – they are statistically inefficient for modeling data that lie on or near a non-linear manifold in the data space [3].

Artificial neural networks trained by back-propagating error derivatives have the potential to learn much better models of data that lie on or near a nonlinear manifold [3]. Over the last few years, advances in both machine learning algorithms and computer hardware have led to more efficient methods for training deep neural networks (DNNs) that contain many layers of non-linear hidden units and a very large output layer known as the deep learning algorithms.

Recently deep learning has been attracting a significant attention from the wide range of applications. Compare to the conventional neural networks, the key features of deep learning are to have more hidden layers and neurons, and to improve learning performance. Using these features, large and complex problems that could not be solved with conventional neural networks can be resolved by deep learning algorithms. Consequently, deep learning has been applied to various applications including pattern recognition and classification problems; for example, speech recognition [3], handwritten digit recognition [4], human action recognition [5], and so on. However, to the best knowledge of the authors, no result has been published in the automatic control field. Thus, this paper focuses on presenting the utilizing possibility of deep learning in control areas. This study was designed to mimic the PID controller using a DBN algorithm. The simulation is performed using Matlab/Simulink and the detailed results of a comparison study between the proposed deep learning controller and a PID controller was conducted to demonstrate the performance and effectiveness of the proposed algorithm.

This paper is organized as follows. The deep learning is described in section 2. In section 3, the design of deep learning controller is explained. The comparison details between the proposed deep learning controller and a PID controller are presented with the simulation results are shown in section 4. Finally, a conclusion and future works follows in section 5.

II. Deep Learning

Deep learning has many layers of hidden units and it also allows many more parameters to be used before over-fitting occurs. The generative pre-training creates many layers of feature detectors that become progressively more complex [6]. A subsequent phase of discriminative fine-tuning, using the standard back-propagation algorithm, then slightly adjusts the features in every layer to make them more useful for discrimination [6]. Thus, for deep learning, a deep architecture is used.

Manuscript received September 11, 2014; revised December 21, 2014.

doi: 10.12720/joace.3.6.452-456

Deep learning is extended algorithm from conventional neural networks, where the number of hidden layers and the number of neurons are more than those of conventional neural networks. In control system, conventional neural networks are well documented and used as a tool for controller design [7], system identification [8], auto-tuning [9], and compensator [10]. In contrary, the deep learning is not used yet, although it is more effective algorithm than conventional neural network, especially in big data. Further, the deep learning algorithm uses a deep architecture. There are several types of deep architectures, among them; the well-known architecture is a DBN.

A. Deep Belief Network Framework (DBN)

A DBN algorithm has two procedures, the pre-training procedure and the fine-tuning procedure. In the first, the observation vector *v* will be pre-trained inan RBM(Restricted Boltzmann Machine) layer to generate an intermediate data vector *v'* and most importantly to calculate the initial weights of the second procedure, the fine-tuning.

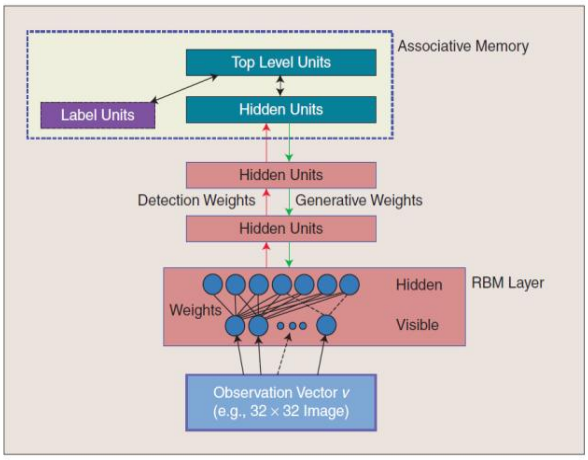

Figure 1. Deep belief network framework [4].

During these procedures, the RBM which is basically composed of three layers is the core difference of the DBN algorithm compare to the conventional neural network. Further, since the RBM is an unsupervised learning, so it has no target data. Moreover, the RBM is responsible for generating the set of weight's initial value that makes the learning better [11].The framework of the DBN algorithm is shown in Fig. 1. This figure indicates that the DBN algorithm has three steps:

Step 1: the input data (the observation vector *v*) goes into the visible layer of RBM, and then by considering the first weights value the data will be transferred to the hidden layer.

Step 2: the first hidden layer becomes second visible layer and transfers the data to second hidden layer by considering the second weights value. In the same way, the second hidden layer becomes third visible layer and transfers the data to third hidden layer. The output of the third hidden layer becomes the initial conditions of the training procedure. Both step 1 and step 2 forms the pre-training procedure that will generate the initial values of weights.

Step 3: This step represents the fine-tuning procedure, in which the learning process will be performed by changing the weights so that the input data follows the target data, similarly to the MLP (Multi-Layer Perceptron) algorithm. The explained DBN is used to develop the deep learning controller.

III. Design of Deep Learning Controller

A. Deep Learing Controller

To design the deep learning controller, a PID controller was first performed and checked, see Fig. 2(a); and then by using the performed PID's input/output information as the input/target data of the learning algorithm, respectively, the deep learning controller was tuned to be capable of replacing the original PID controller, see Fig. 2(b), so finally the DC motor will be controlled just by the deep learning controller, see Fig. 2(c).

The considered deep learning algorithm is based on the Deep Neural Network toolbox developed by Tanaka [12], in which was used as a pattern recognition tool, but in this paper it is used as a tool to design the controller, where, it is modified in which it's hidden layer consists of two layers, each layer contains 50 neurons. Thus, the iteration number was 300, the learning rate was 0.01, and the dropout rate was 0.5.

There are different setting methods for an RBM including: BBPRBM (Beta-Bernoulli Process RBM) [13], GBRBM (Gaussian-Bernoulli RBM), and BBRBM (Bernoulli-Bernoulli RBM). The BBPRBM which gives the better performances was considered where the sigmoid function was used as the activation function.

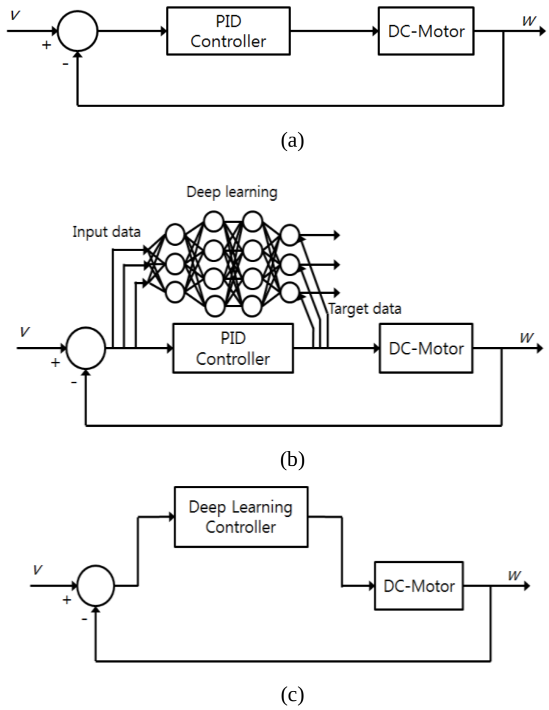

Figure 2. Design of deep learning controller.

B. DC Motor Discription

The dynamic equations of the considered DC motor are as follows:

$$\frac{dw}{dt}=\frac{1}{J}\left(K_t i-bw\right) \tag{1}$$

$$\frac{di}{dt}=\frac{1}{L}\left(-Ri+V-K_e w\right) \tag{2}$$

where, J is the moment of inertia of the rotor, K_t is motor torque constant, i is the armature current, b is motor viscous friction constant, L is electric inductance, R is electric resistance, and K_e is electromotive force constant.

The simulation of the DC motor was performed in Matlab/Simulink as shown in Fig. 3. The parameter values of the considered DC motor can be found in Table I.

TABLE I. DC MOTOR PARAMETERS SETTING

Parameter	Value	Parameter	Value
J	0.01	L	0.5
b	0.1	R	1
K_t	0.01	K_e	0.01

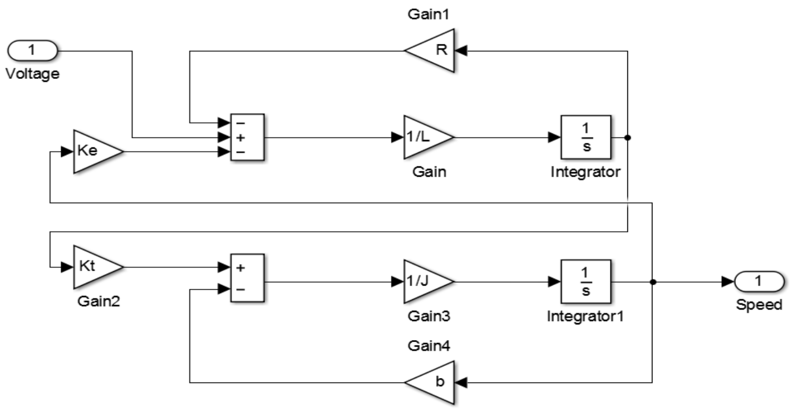

Figure 3. Simulink block of DC motor system.

The total feedback control of DC motor based on deep learning controller in MATLAB environment is given in Fig. 4, where, the input and output of the system are voltage V and angular speed$w$, respectively.

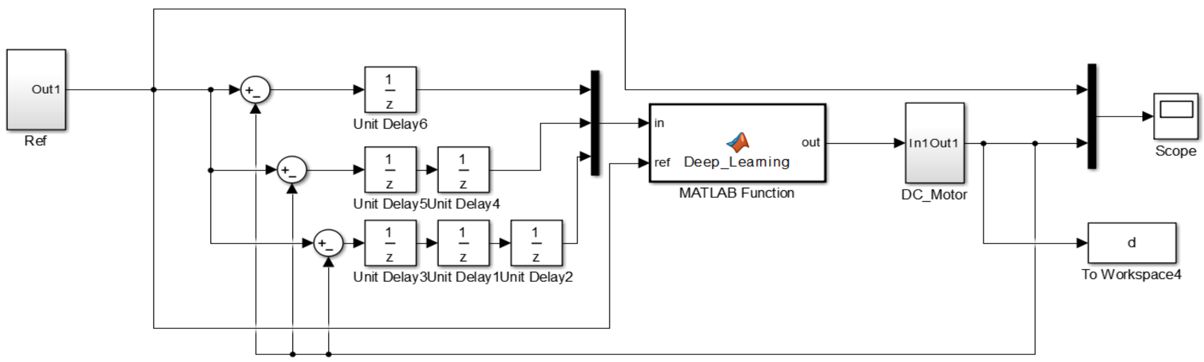

Figure 4. Feedback control of DC motor based on deep learning controller in Matlab/Simulink.

IV. SIMULATION RESULT

The simulation was conducted in two scenarios to check the performances of the proposed controller: Scenario. 1: the DC motor was excited with a simple step input; Scenario. 2: the DC motor was excited with a more complex input, the cascade step input.

A. Scenario. 1: A Simple Step Input

PID controller is the most commonly used in industry and has been universally accepted in industrial control, because of its robustness and functional simplicity. Thus, to design the deep learning controller, a PID controller was computed and executed first, till the results were satisfied, as shown in Fig. 5. And then, the DBN algorithm learned the PID controller to design the proposed deep learning controller. The performances of the proposed deep learning controller were almost as good as the PID controller as shown in Fig. 6.

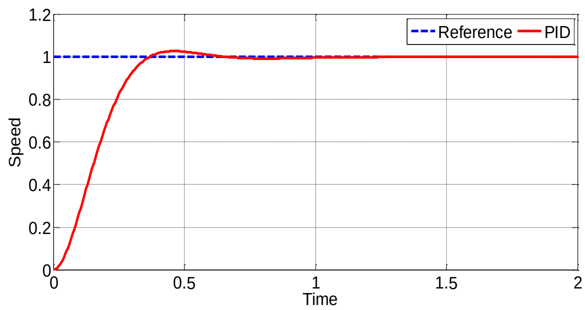

Figure 5. System response using the PID controller in the scenario. 1.

The simulation was performed in Matlab/Simulink environment and the detailed results of a comparison study between the proposed deep learning controller and a PID controller was conducted using the residuals and the RMSE variations. The simulation was performed with a sampling time t_s= 0.005s and a run time t_r= 40s.

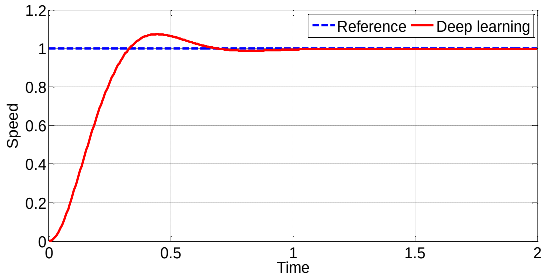

Figure 6. System response using the deep learning controller in the scenario. 1.

Fig. 7 shows that the residuals were big in the transient part, but after just 0.7s the residuals were nulled for both the PID and the proposed deep learning controllers. Further, the RMSE variations for both controllers were very small (~ 10^{-2}). Thus, as shown in Fig. 7, the residual and RMSE results demonstrate the effectiveness of the proposed deep learning controller to be used as a tool to control the DC motor output, the speed.

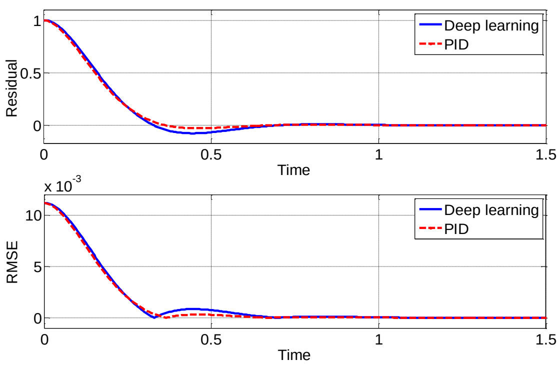

Figure 7. The residual and RMSE variations of deep learning controller and PID controller.

B. Scenario. 2: Acomplex Step Input

A cascade step input was considered to verify the proposed controller in a more complex environment; the results are shown in Fig. 8 and Fig. 9.

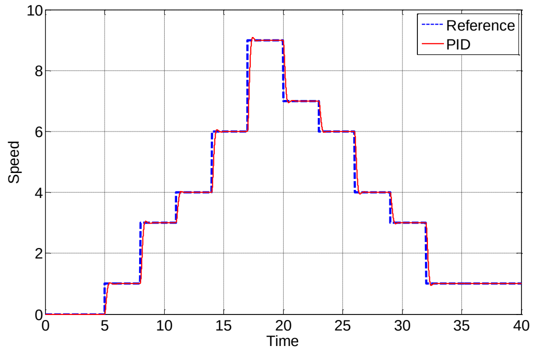

Figure 8. System response using the PID controller in the scenario. 2.

Fig. 8 and Fig. 9 show that the performances of the proposed deep learning controller were almost as good as the PID controller similarly to the results of scenario 1. Further, the summary of the comparison between the PID controller and the deep learning controller are given in Table II.

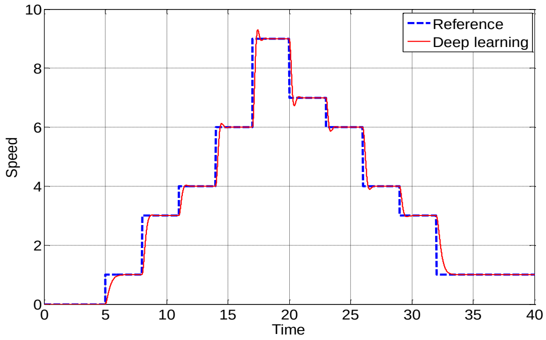

Figure 9. System response using the deep learning controller in the scenario. 2.

TABLE II. RMSE RESULT

Method / Scenario	PID controller	Deep learning controller
Scenario 1	0.0539	0.0554
Scenario 2	0.3097	0.3659

V. CONCLUSION

In this paper, a deep learning controller based on DBN algorithm was designed to explore the ability of applying the deep learning algorithm to the control problems. A comparison study between the PID controller and the proposed deep learning controller was performed to verify the feasibility of the use of deep learning in control theory. The simulation results demonstrate the effectiveness of the proposed deep learning controller to be used as a control tool.

ACKNOWLEDGMENT

This research was supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the C-ITRC (Convergence Information Technology Research Center) support program (NIPA-2014-H0401-14-1004) supervised by the NIPA (National IT Industry Promotion Agency).

REFERENCES

[1] X. Zhu, C. Guan, J. Wu, Y. Cheng, and Y. Wang, "Expectation-Maximization Method for EEG-Based Continuous Cursor Control," *EURASIP Journal on Applied Signal Processing* vol. 1, p. 26., 2007.

[2] M. Toussaint, A. Storkey, and S. Harmeling, "Expectation-Maximization methods for solving (PO) MDPs and optimal control problems," *Inference and Learning in Dynamic Models*, 2010.

[3] G Hinton, L Deng, D Yu, GE Dahl, “Deep Neural Networks for Acoustic Modeling in Speech Recognition,” *IEEE Signal Processing Magazine*, Nov 2012.

[4] I Arel, DC Rose, TP Karnowski, “Deep Machine Learning—A New Frontier in Artificial Intelligence Research,” *IEEE Computational Intelligence Magazine*, Nov 2010.

[5] S Ji, W Xu, M Yang, K Yu, "3D Convolutional Neural Networks for Human Action Recognition," *IEEE Transon Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, Jan 2013.

[6] A Mohamed, G E Dahl, and G Hinton, "Acoustic Modeling using Deep Belief Networks," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 20 no. 1, pp. 14-22, 2012.

[7] M Lee, HS Choi, "A Robust Neural Controller for Underwater Robot Manipulators", *IEEE Transactions on Neural Networks*, Nov, 2000.

[8] V Prasad, BW Bequette, "Nonlinear System Identification and Model Reduction Using Artificial Neural Networks," *Computers & Chemical Engineering*, December, 2003.

[9] J Chen, TC Huang, "Applying Neural Networks to On-line Updated PID Controllers for Nonlinear Process Control," *Journal of Process Control*, 2004.

[10] B Xu, SR Pandian, N Sakagami, F Petry, "Neuro-fuzzy control of underwater vehicle-manipulator systems," *Journal of the Franklin Institute*, vol. 349, issue. 3, pp. 1125-1138, April 2012.

[11] G Hinton, S Osindero, YW Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, vol. 18, issue. 7, July, 2006.

[12] M Tanaka and M Okutomi, "A Novel Inference of a Restricted Boltzmann Machine," *International Conference on Pattern Recognition (ICPR2014)*, August, 2014.

[13] R Mittelman, H Lee, B Kuipers, "Weakly Supervised Learning of Mid-Level Features with Beta-Bernoulli Process Restricted Boltzmann Machines," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June, 2013.

K. Cheon received his B.S. degree in Electronic Engineering from Kyungpook National University, Korea, in 2013. He is currently a M.S. candidate in Control Engineering from Kyungpook National Universtity. His research interest focuses on the design of intelligentcontroller, artificial intelligence, neural networks, and deep learning.

J. Kim received his B.S. and M.S.in Electronics Engineering from Kyungpook National University, Korea, in 2011 and 2014, respectively. Recently he is a PhD candidate in DECS laboratory, School of Electronics Engineering, Kyungpook National University. His research interests are filtering, signal processing, fault detection and diagnosis, and non-linear control.

M. Hamadache received his State Engineering degree in automation of industrial processes from the University of M'hamed Bougara Boumerdes (UMBB), Algeria, in 2008. Since September 2009, he has been a PhD candidate in DECS Lab, School of Electronics Engineering, Kyungpook National University, Daegu, South Korea. His main research area is model-based and data-driven fault detection and diagnosis approaches in rotating machinery and smart actuators.

D. Lee received his B.S. and M.S. in Control Engineering from Kyungpook National University, Korea, in 1987 and 1990, respectively. In 2002, he received a PhD from Sheffield University, England. He is now an Assistant Professor of the School of Electronics Engineering, Kyungpook National University. His research interest focuses on the design of real-time networked control for various safety-critical applications, including submarines, autonomous underwater vehicles and intelligent automobiles.

