Black Sea Journal of Engineering and Science
doi: 10.34248/bsengineering.1238543

Open Access Journal
e-ISSN: 2619 – 8991

Research Article
Volume 6 - Issue 2: 87-97 / April 2023

A HYBRID ALGORITHM FOR ADAPTIVE NEURO-CONTROLLERS

Mustafa DEMİRTAŞ[1]*

[1]*Bursa Uludağ University, Faculty of Engineering, Department of Electrical and Electronics Engineering, 16059, Bursa, Türkiye*

Abstract: In this study, a novel hybrid algorithm consisting of the least mean square and backpropagation neural network is proposed to auto-adjust adaptive proportional integral derivative (PID) controller gains for improving the transient response of linear systems. The hybrid approach comprises the scheme of the two algorithms running in parallel and updates PID gains simultaneously. All algorithms are implemented on the same linear system and present a general framework for different scenarios such as initial PID gains, learning rates, and target functions. The results show that the presented hybrid algorithm has better accuracy, precision, F1-score, adaptability, and robustness than origin algorithms, and significantly improves the controllability in most of the system scenarios. It also exhibits better performance in periodic incremental and decremental targets compared to origin algorithms. Different hybridization levels are also simulated and are highlighted as significant features of their performance. This work can be expanded to the combination of other well-known algorithms, paving the way to significant improvements in control system applications.

Keywords: Parallel algorithms, Neuro-controllers, Hybrid intelligent systems, Artificial neural networks, Automatic control

***Corresponding author:** Bursa Uludağ University, Faculty of Engineering, Department of Electrical and Electronics Engineering, 16059, Bursa, Türkiye
E mail: mustafademirtas@uludag.edu.tr (M. DEMİRTAŞ)
Mustafa DEMİRTAŞ https://orcid.org/0000-0001-6832-4341

Received: January 18, 2023
Accepted: March 11, 2023
Published: April 01, 2023
Cite as: Demirtaş M. 2023. A hybrid algorithm for adaptive neuro-controllers. BSJ Eng Sci, 6(2): 87-97.

1. Introduction

Proportional integral derivative (PID) control which depends on a common feedback form is one of the earlier control types (Ang et al., 2005). The first usage of the PID controller was pneumatic devices and solid-state electronics where the traces are found in the 1940s, before arriving at today's implementation of computer structures (Verma and Padhy, 2020). It is very popular in control system theory and applications because of its simple control structure, algorithm, good robustness, and stability. However, the PID controller has disadvantages in that it is not suitable for non-linear and long-time-delay systems since the arrangement and investigation of suitable P, I, and D parameters and their combination is a cumbersome process. In the last few decades, by evolving computer technology and control theories, it was possible to incorporate the innovations of estimation algorithms and neural networks into the area of control problems (Guo et al., 2009; Hernández-Alvarado et al., 2016; Adar, 2021).

Adaptive algorithms, which can globally stabilize systems having spike noises, bounded external disturbances, and time-varying parameters with no limitation on signals in the closed-loop system, still exist in many commercial control systems (Huo and Xiong, 2019). Among them, the adaptive PID controllers are the most popular (Bolton, 2015). An adaptive PID controller adapts to the process conditions on-line by making necessary changes in the values of K_p and K_i (Conker and Baltacioglu, 2020). This type of controller benefits from several advantages, such as the ability to ensure system stability, not excessively relying on models, achieving a given system target asymptotically, and improving itself in response to changes in system dynamics.

Backpropagation neural network (BPN) is one of the most well-known and common methods which is used to minimize the error of possible objective functions (Oztekin and Ozgan, 2012; Chen and Gu, 2020). Although this approach seems suitable for control system identification problems, however, it has some drawbacks in real-time applications because of long training time and slow performance issues (Orozco-Tupacyupanqui et al., 2016). Another well-known approach in literature for minimizing the error is the LMS algorithm. This algorithm is generally used in detection and estimation theory (Antony Dhas and Chandrasekaran, 2019; Zayyani and Javaheri, 2021). It is also possible to use this algorithm for minimizing the error of the objective function (Akhyar and Omatu, 1993; Guo et al., 2009; Hernández-Alvarado et al., 2016). One of the main drawbacks of LMS is that it is sensitive to the input function, and this leads very difficultly to determine the learning rate that guarantees stability (Haykin, 2005). Further normalization of input power solves this problem (Haykin and Widrow, 2003). The adaptive neural network PID controller and adaptive least mean square PID controller, which are new kinds of controller have been offered and developed in order to get rid of the above-mentioned problems (Guo et al., 2009; Bai and Zhang, 2018).

The integration of more than one algorithm has also demonstrated promising results in many adaptive control problems, and further research is required in this direction. Different types and combinations of hybrid algorithms have been developed in many papers (Qiao et al., 2017; Mahmoodabadi et al., 2018; Moayedi et al., 2019; Pandey et al., 2021; Carvalho et al., 2021; Tamer et al., 2021; Alkrwy et al., 2021, El-Nagar et al., 2022) to find gains of different controller types to enhance the system transient response as well as to ensure the robustness and stability of the system. Text detection and character recognition are successfully achieved by weighted naïve Bayes classifier and deep neural network (DNN)-based adaptive galactic swarm optimization (GSO) (Pandey et al., 2021). A hybrid neuro particle-based optimization (PSO) of the artificial neural network (ANN) is investigated for slope stability calculation (Moayedi et al., 2019). A control optimization system based on hybrid intelligent technology is proposed to obtain the minimum energy consumption in the wastewater treatment process (Qiao et al., 2017). Compared to the traditional PID and data-driven adaptive optimal controller (DDAOC) methods, the simulation results of the proposed method show better performance. Another hybrid controller based on the robust decoupled sliding mode and adaptive feedback linearization is being studied (Mahmoodabadi et al., 2018). A control algorithm based on the weighting sum of the feedback linearization (FBL) and decoupled sliding mode control (DSMC) methods is proposed, as the main idea of this is to enhance efficiency and robustness against uncertainties. The results show that the dynamic responses obtained from the proposed hybrid controller are much faster than those obtained from the FBL, DSMC, and other approaches considered in the literature. In (Alkrwy et al., 2021), a new method for adjustment of the PID parameters to improve the tracking performance of DC motors also provides optimal stability by creating a hybrid PID - Crow search algorithm (CSA) predictive model for tuning parameters of the PID controller of DC motors. The presented results proved that the proposed hybrid system provides the best set of transient responses (rise time, stability time, and minimization in settling time and eliminating steady-state error) specifications compared to four different CSA releases based on various performance response indicators. Moreover, it is compared with other tuning methods such as PSO-based console, and Ziegler Nichols tuning method. To handle a nonlinear system, a new hybrid deep learning neural network controller (HDLNNC) is proposed based on a self-organizing map of the Kohonen procedure and Hebbian learning (El-Nagar et al., 2022). It can be concluded that the robustness of the proposed hybrid controller has better performance and faster recovery ability from parameter variations and disturbance signals as compared to multilayer feed-forward neural network controllers. In (Carvalho et al., 2021), the development of a Fuzzy-PID hybrid controller to control a quadrotor Unmanned Aerial Vehicles' (UAV) height stability is discussed. The performances of traditional PID and the proposed hybrid controller are also given. From the results, both PID and Fuzzy-PID controllers could perform the attitude control of the UAV. However, the hybrid control strategy obtained some advantages, such as self-adjustment through system variations. The PSO algorithm is implemented to optimize and tune PI controller parameters of DC bus voltage control of the shunt active power filter (Tamer et al., 2021). The reference current is obtained using an adaptive linear neuron. These networks are trained online using the LMS algorithm. The proposed hybrid control algorithm presents higher efficiency in terms of harmonic current mitigation, power factor correction, and DC-link voltage regulation. The comparison of hybrid algorithms compared to their origin algorithms is summarized in Table 1.

The aim of this paper is the presentation of a novel hybrid scheme for the adaptive PID controller by combining two traditional algorithms which are BPN and LMS, to take advantage of both. These algorithms have not been previously combined in the literature not only for optimization but also for control problems. Generally, the proposed hybrid algorithm improves the transient response and has a much higher probability of convergence than that of BPN and it is faster than LMS

Table 1. Performance of hybrid algorithms compared to their origin algorithms

Hybridized Methods	Performance comparisons
DNN-GSO (Pandey et al. 2021)	Improved precision, accuracy, and F1-score
ANN-PSO (Moayedi et al. 2019)	Improved R2 and lower RMSE
PID-DDAOC (Qiao et al. 2017)	Energy consumption decreased by 13.6%
DMSC-FBL (Mahmoodabadi et al. 2018)	Improved robustness and effectiveness
CSA-PID (Alkrwy et al. 2021)	Improved the steady-state error, step response stability, overshoot, rising time, and settling time
HDLNNC (El-Nagar et al. 2022)	Fast learning, stable controller, reduced system uncertainties
Fuzzy-PID (Carvalho et al. 2021)	Reduced overshoots and undershoots
PSO-LMS-PI (Tamer et al. 2021)	Higher efficiency, improved power factor

It differs in a certain way from the aforementioned studies by using simultaneous weight vector updates to assign new PID gains. The performance metrics of the proposed method are found superior to the origin algorithms. The results also verify that this methodology provides a better response in periodic incremental and

decremental systems. However, it is difficult to guarantee its effectiveness in the first step response of some scenarios, because the hybrid algorithm essentially depends on the success of origin algorithms.
To summarize, the main contributions of this paper are:

- A hybrid framework is proposed, consisting of BPN and LMS algorithms, for improving the transient response of a linear system.
- The performance of each algorithm and comparison with the hybrid algorithm is identified.
- Different scenarios are tested to figure out the sensitivity of the hybrid algorithm.
- A weighting coefficient is assigned to adjust the hybridization level, and the performances of different hybridizations are compared.

2. Material and Methods

2.1. System Structure and PID Controller

The PID controller which combines the parallel connection of P, I, and D types of the controller is an essential element of a feedback control system. Even though different combinations of these control types exist such as P, I, D, PI, PD, and PID controllers, pure I and D controller are not recommended. The PID controllers are given relatively better results compared to the other type of controllers for a linear system (Jaleel and Thanvy, 2013).
The structure which is shown in Figure 1 is a closed loop of the discrete-time system. In this system, r(n) is input, e(n) is error signal, u(n) is control input, y(n) is output, and the plant is a controllable process. In our system, all variables are in the discrete-time domain.
The equation (1) of the digital incremental PID controller is given as follows (Guo et al., 2009):

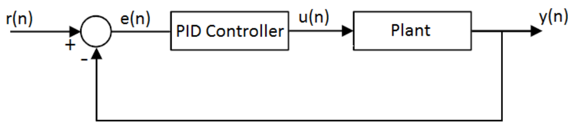

Figure 1. Structure of the discrete-time system.

$$u(n) = u(n-1) + K_p\big(e(n) - e(n-1)\big) + K_i\, e(n) + K_d\big(e(n) - 2e(n-1) + e(n-2)\big) \quad (1)$$

where K_p is proportional, K_i is integral, and K_d is the derivative gains of the PID controller. These parameters should be auto-adjusted. Three parameters describe the whole system adjustment, in this work, it will be mentioned how those should be adjusted in three different approaches.
The implementation is applied to the linear system using MATLAB. The linear model of the system was given in equation 2:

$$y(n+1) = 0.998y(n) + 0.232u(n) \quad (2)$$

Where the system is a single input single output of a temperature control process for a water bath. The details of the plant and results of experiments on a physical system can be found in (Akhyar and Omatu, 1993). The experiments were carried out when the volume of a water bath was 7 liters, the power of the electric heater was 600 W, the references were 40 °C, and the sampling time was 30 seconds. The control input was limited to between 0 and 5 volts. They used a pure BPN-based adaptive PID controller and reported successful results for both simulation and implementation.
This work adds a parallel running LMS algorithm to the previous version of the work (Akhyar and Omatu, 1993), providing efficient improvements in weight updates. To investigate the sensitivity of the hybrid algorithm, the simulations are repeated for different initial PID gains, learning rates, targets, and hybridization levels. It is important to note that the initial settings of PID are determined to make the system stable. The PID gains are not limited, because it is difficult and time-consuming, sometimes impossible to converge, for the proposed learning algorithms.

2.2. Back Propagation Neural Network Algorithm

A Back-Propagation neural network (BPN) is a multilayer feed-forward neural network algorithm. It is called a multilayer network since, in addition to the input layer and output layer, the model contains a hidden layer with a definitive amount of processing elements heavily connected to both layers. This hidden layer provides adaptivity between input and output in a nonlinear fashion. The feed-forward structure explains the learning scheme which feeds the signal from the input to the output layer through the hidden layer and provides a possible output signal. The difference between the output and the expected signal is fed backwardly through the network in order to arrange the weight parameters using the gradient descent algorithm (Dogo et al., 2018). In this paper, the neural net consists of an input layer of 3 inputs and 3 outputs at the output layer where the PID parameters are obtained.
The algorithm of BPN is shown as follows: w_{ij} is the hidden layer's weight value, x_j (j=1,2,.... m) is the input of the network. Equations 3 and 4 are described as the hidden layer's input and output relations.

$$net_h(n) = \sum_{j=1}^{m} w_{ji} x_j \quad (3)$$

$$Outo_h(n) = f\big(net_h(n)\big) \qquad h = 1,2,\ldots.m \quad (4)$$

The activation function of the hidden layer and the output layer is the sigmoid function. The equations of the neural network were given in equations 5, 6, and 7 (Guo et al., 2009):

$$f(x) = \frac{1}{1+e^{-x}} \quad (5)$$

$$net_o(n) = \sum_{k=1}^{m} w_{ik} Outo_h \qquad h = 1,2, \dots . m \tag{6}$$

$$Outo_o(n) = g\big(net_o(n)\big) \tag{7}$$

One of the BPN approaches for adapting the system parameters in different control structures has been described in (Akhyar and Omatu, 1993). In this approach, the plant is controlled and adopted using a BPN algorithm as it has been shown in Figure 2. Inputs of the neural network are time-delay added PID controller output, time-delay added plant output, and reference input. Three inputs of the PID controller are BPN outputs which are K_p, K_i, and K_d parameters, the other is a system error. In order to calculate the system error, reference input minus output is required.

Normalized input data is used in the BPN algorithm. The numbers of reference nodes and neurons are 300 and 10, respectively. The learning parameter (μ) and momentum parameter are set as a variable. A sigmoidal function is used as an activation function for both the hidden layer and the output layer.

2.3. Least Mean Square Algorithm

The least mean square (LMS) algorithm is an adaptive algorithm based on the steepest descent method. It is a search algorithm which uses for estimating the gradient vector of the data. LMS algorithm has an iterative procedure for updating the weight vector which leads to determining the minimum mean square error. It is widely used in various applications of adaptive filtering, detection, and estimation (Bai and Zhang, 2018; Spelta and Martins, 2020; Karchi and Kulkarni, 2021).

LMS algorithm was given in equations 8, 9, and 10:

$$w(n+1) = w(n) + \Delta w(n) \tag{8}$$

$$\Delta w(n) = \mu e(n) x(n) \tag{9}$$

$$e(n) = d(n) - y(n) \tag{10}$$

where x_k is an input vector, w_k is weight vector (k=0,1,…,N-1), e(n) is error signal, y(n) and d(n) are output and desired signal, respectively (Hernández-Alvarado et al., 2016, Huo and Xiong, 2019). Equations 8, 9, and 10 describe the process of the LMS algorithm. The μ is the learning rate and Δw(n) is the updating difference of the weight vector which is produced in every cycle of the algorithm. The structure of PID based on the LMS controller was given in Figure 3. Every new weight vector multiplies with the PID parameters K_p, K_i, and K_d.

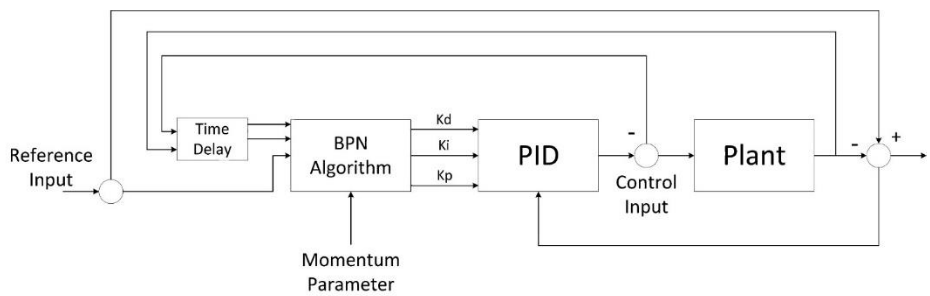

Figure 2. System schematic of BPN-based PID.

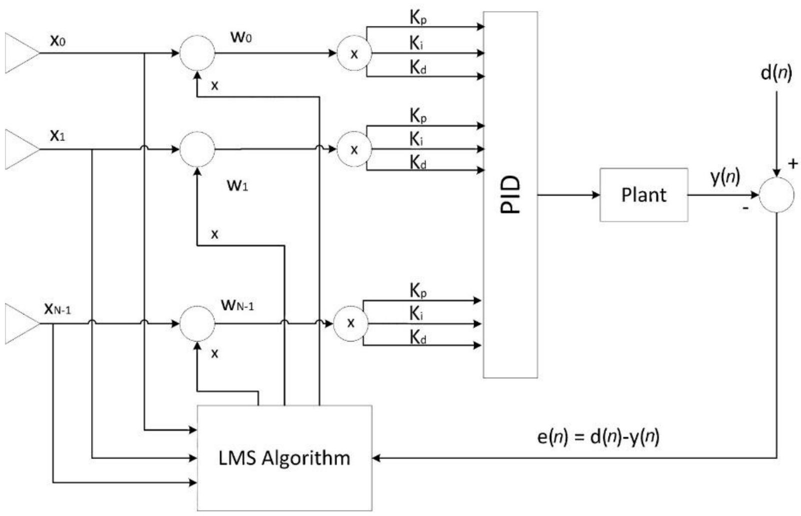

Figure 3. Structure of adaptive LMS-based PID controller.

2.4. Hybrid Algorithm

A hybrid algorithm is based on the parallel connection of the LMS and BPN algorithms. Each algorithm runs simultaneously and calculates its weight update value, then each result is multiplied with PID parameters K_p, K_i, and K_d at the same time. The structure of the hybrid adaptive PID (HAPID) algorithm was given in Figure. 4.

The hybrid algorithm continues to run until the absolute value error is reduced to 1E-4. The resulting closed-loop system is necessarily stable for any randomly chosen initial weights in the range of [0, 1] and arbitrary PID gains within a specified range. However, with different system scenarios, the closed-loop system again must be stable, but the closed-loop performance is expected to vary within this entire stability region. In the hybrid update scheme, while the LMS weight update value is the same for each PID controller gain, BPN produces different update values for each PID controller gain.

The updating algorithm (equations 11-16) of the hybrid scheme is given by:

$$K_p(n) = P, K_i(n) = i, K_d(n) = d \quad for \quad n = 0 \tag{11}$$

$$LMS_{update} = wc * w(n+1) \tag{12}$$

$$BPN_{update,i} = wc * Outo_o(i), \quad i = 1 \; to \; 3 \tag{13}$$

$$K_p(n+1) = K_p(n) * LMS_{update} * BPN_{update,1} \tag{14}$$

$$K_i(n+1) = K_i(n) * LMS_{update} * BPN_{update,2} \tag{15}$$

$$K_d(n+1) = K_d(n) * LMS_{update} * BPN_{update,3} \tag{16}$$

where P, I, and D are initial values of PID gains, K_p, K_i, and K_d are adaptively updated PID gains in every cycle, wc is the weighting coefficient used to determine the hybridization level of HAPID, LMS_{update} and BPN_{update} are update vectors obtained because of LMS and BPN runs. The others parameters described in previous sections.

3. Results and Discussion

The response of all algorithms was systematically investigated using different parameters. Firstly, the learning rate of all algorithms was changed. In this case, the number of neurons, inputs, and momentum parameters that affect the BPN-based PID algorithm was kept constant. Thus, the learning rate effect was analyzed for each scenario. Moreover, the effect of the different initial system parameters for K_p, K_i, and K_d was observed for the learning rate at 0.9. Furthermore, different inputs are also implemented in Scenario 4. In scenarios 1 through 4, the full effect (100%) of the LMS and BPN algorithm is used. Other scenarios (between Scenarios 5 and 12) are created to test the first four scenarios with different hybridization levels. Two different levels of hybridization which are "50% BPN+50% LMS" and "25% BPN+75% LMS" are implemented. The variables of all scenarios are given in Table 2.

The algorithm needs to have a good initial PID parameter set to start with in order to exhibit satisfactory performance. Thus, the PID parameters used in Scenario 1 are initially determined by the Ziegler–Nichols tuning method (Patel, 2020). In other words, the current values of the K_p, K_i, and K_d gains are increased or decreased utilizing the proposed algorithms. Initial system parameters are then changed to show the effectiveness of the algorithm within stability limits.

All algorithms have improved responses when the learning rate increases. In particular, the improvement of the LMS-based PID and the HAPID algorithm responses is more compared to BPN-based PID. Figure 5 shows the overall system and step response of all proposed algorithms for Scenario 1, respectively. The LMS-based PID and HAPID algorithm responses have an under-damped response. The BPN-based PID algorithm has a deadbeat response. The rise of the LMS-based PID and HAPID algorithms is almost the same and is less compared to the BPN-based PID algorithm.

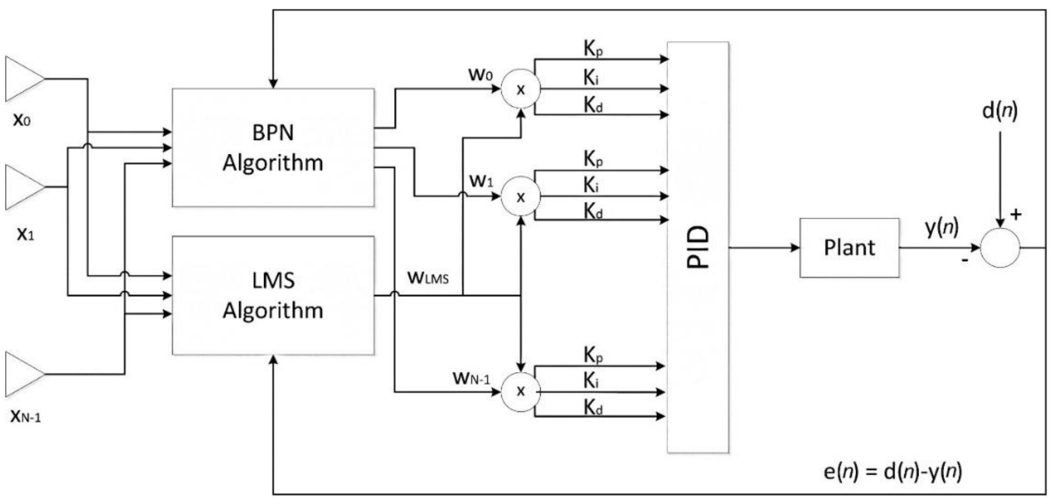

Figure 4. Structure of the HAPID algorithm.

Table 2. The system parameters for performance analysis

		Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
K_p		1.4	1.4	1.4	1.4	1.4	1.4
K_i		0.05	0.05	0.5	0.5	0.05	0.05
K_d		0.01	0.01	0.01	0.01	0.01	0.01
μ		0.5	0.9	0.9	0.9	0.5	0.5
Inputs		1,2,3	1,2,3	1,2,3	2,4,3	1,2,3	1,2,3
Hybrid	LMS	%100	%100	%100	%100	%50	%75
Level	BPN	%100	%100	%100	%100	%50	%25
		Scenario 7	Scenario 8	Scenario 9	Scenario 10	Scenario 11	Scenario 12
K_p		1.4	1.4	1.4	1.4	1.4	1.4
K_i		0.05	0.05	0.5	0.5	0.5	0.5
K_d		0.01	0.01	0.01	0.01	0.01	0.01
μ		0.9	0.9	0.9	0.9	0.9	0.9
Inputs		1,2,3	1,2,3	1,2,3	1,2,3	2,4,3	2,4,3
Hybrid	LMS	%50	%75	%50	%75	%50	%75
Level	BPN	%50	%25	%50	%25	%50	%25

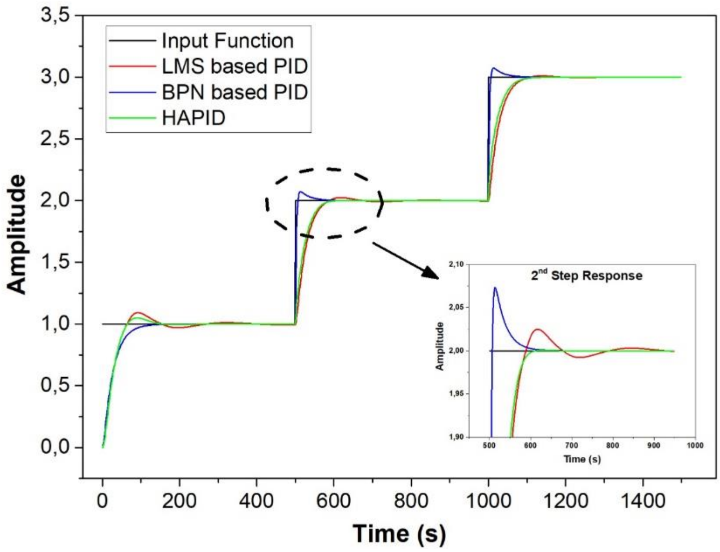

Figure 5. Overall system response of scenario 1 (inset: 2nd-step response).

The steady-state time of HAPID and BPN-based PID is almost equal, however, HAPID is faster than BPN-based PID. Considering the 2nd and 3rd-step responses, the BPN-based PID has almost the same overshoot and steady-state time. The LMS-based PID algorithm exhibits under-damped responses with lower overshoot and longer steady-state time than the BPN-based PID. The HAPID has a deadbeat response and less steady-state time compared to other algorithms.

Overall system response and 2nd-step response for increased learning rate (Scenario 2) are shown in Figure 6. When the learning rate is increased, the step response of HAPID turns from an under-damped response to a dead-beat response and a faster algorithm compared to others. The overshoot of LMS-based PID reduces but steady-state time is longer compared to others. The response of the BPN-based PID algorithm does not affect by the change in the learning rate. Figure 7 shows the overall system, 1st, and 3rd-step responses of Scenario 3. In Scenario 3, we change the initial integral parameter of the PID controller. Step and magnitudes of inputs are the same as in Scenarios 1 and 2. An interesting result was observed. The BPN based PID controller has a good 1st-step response compared to others, however, when the step number increases the LMS-based PID and HAPID responses suppress the oscillations and get better compared to the BPN-based PID. Particularly, the HAPID algorithm has the best response to the 3rd-step.

Similarly, Figure 8 shows the overall system, 1st, and 3rd-step response, respectively. Initial parameters, learning rate, and the number of inputs in Scenario 4 are the same as in Scenario 3. Initial input magnitudes are increased to two and decreased step input is added, thus we can observe the change in initial response and both increased and decreased step inputs. The HAPID improves the 1st-step response for increased initial input. Overall results for Scenario 4 show that the HAPID has the best response for all step responses, especially for a decreased step input. All algorithms have under-damped response, but the HAPID exhibits less overshoot and steady-state time.

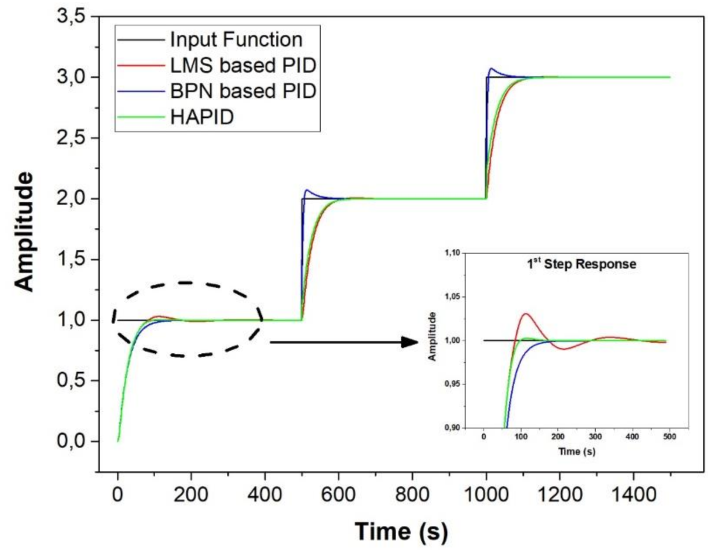

Figure 6. Overall system response of scenario 2 (inset:1st-step response).

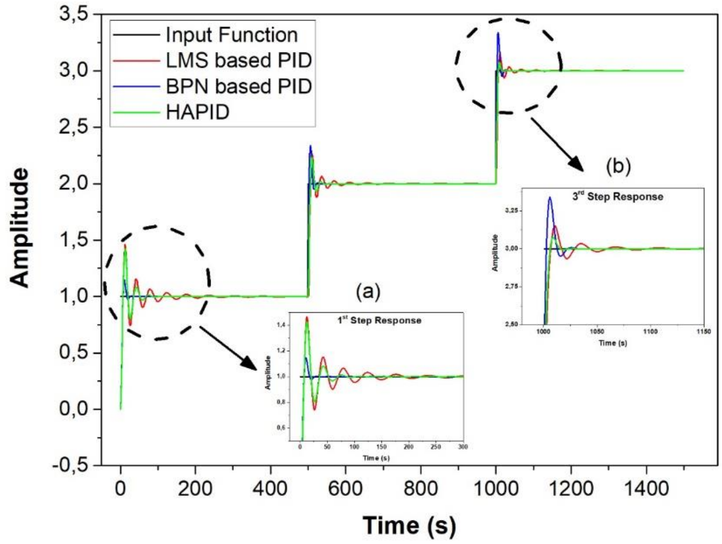

Figure 7. Overall system response of scenario 3 (inset (a): 1st-step response, inset (b):3rd-step response).

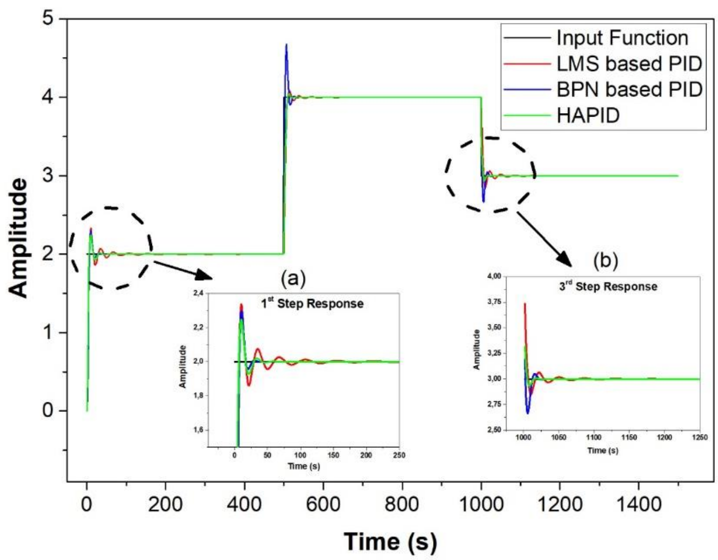

Figure 8. Overall system response of scenario 4 (inset (a):1st-step response, inset (b):3rd-step response).

Apart from the individual performance of algorithms, change in the hybridization level of algorithms with different scenarios are shown in Figures 9-12. Results obtained with Scenarios 1, 5, and 6 are compared in Figure 9. The HAPID, which has a full effect on all step inputs, gives better results in all aspects i.e. in terms of undershoot, overshoot, and settling time. Although the HAPID with full effect at the 1^{st}-step input makes a little overshoot, it has improved the response to the dead-beat after the 2^{nd}-step input. Other hybrid levels tended to improve their response at each step input, respectively. Scenario 5 exhibits a better result than Scenario 6. This result supports that the LMS algorithm affects more than the BPN algorithm for given system parameters.

Figure 10 shows the performance comparison of Scenarios 2, 7, and 9. It is seen that the proposed HAPID with full effect does not yield better performance in all step inputs compared to the different hybridization levels. In the first two-step inputs, Scenario 2 showed the best response, while in the 3^{rd}-step, Scenario 7 showed a better performance than the others. Scenario 2 does not show any improvement depending on the number of step inputs, while the other two scenarios exhibit a noticeable improvement. It is proof that performance can be improved with different hybridization according to the input signal.

Convergence characteristics for Scenarios 3, 9, and 10 are shown in Figure 11. It is noted in simulation results that Scenario 3 performs better as compared to others, and only Scenario 3 can reach the zero steady-state error in the 1^{st}-step response. As the effect of the algorithms decreases, fluctuations in their response increase. When the effect of the LMS and BPN algorithms is halved, the controller's response slows down, and the amount of overshoot and undershoot increases. Moreover, controller performance worsens when the effect of the BPN is increased and the LMS is decreased. The system with the same parameter has been simulated in the case of increased step amplitude and decremental input. The corresponding responses of Scenarios 4, 11, and 12 are given in Figure 12. In line with the results in Figure 11, Scenario 4 gives a very good response to suppress fluctuations.

The performance comparison of HAPID is discussed below. However, hybridization based on LMS and BPN papers is found rare. Therefore, five different efficient and well-known optimization algorithms are implemented to test the effectiveness of this proposed algorithm. The results obtained by the proposed approach and existing LMS, BPN, GSO, CSA, and PSO-based adaptive algorithms are shown in Table 3. The accuracy, precision, and F1-score of the proposed HAPID are found superior to the other five conventional optimization algorithms.

The trade-off between computational complexity and accuracy is an important consideration for the LMS, BPN, and HAPID algorithms. The LMS algorithm has a lower computational complexity than the BPN algorithm. While it may be less powerful than the BPN algorithm, it is computationally efficient and suitable for applications with a large number of data samples. In contrast, the BPN algorithm is more computationally intensive than the LMS algorithm, but it is more powerful and versatile. The BPN algorithm can be used for various applications, which require higher accuracy and more complex decision-making. The HAPID algorithm, on the other hand, comes with a computational burden, as it adds more steps and parameters that increase the overall complexity of the algorithm.

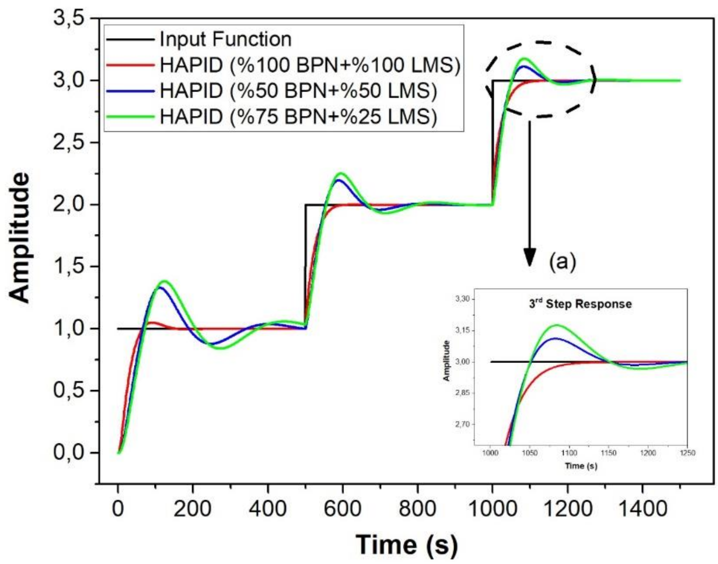

Figure 9. Performance comparison between different hybridization levels for scenario 1, scenario 5, and scenario 6 (inset (a):3^{rd}-step response).

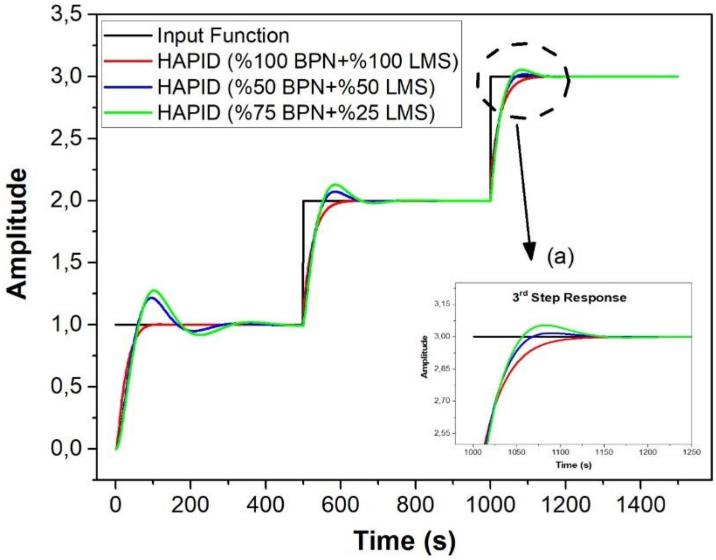

Figure 10. Performance comparison between different hybridization levels for scenario 2, scenario 7, and scenario 8 (inset (a):3rd-step response).

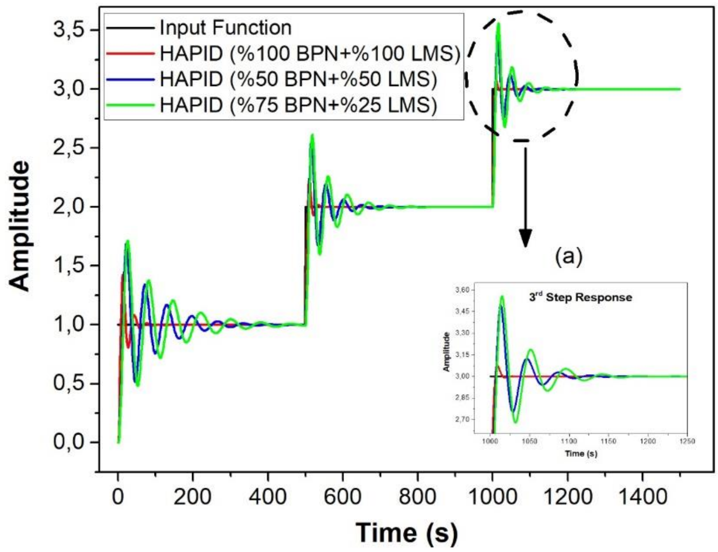

Figure 11. Performance comparison between different hybridization levels for scenario 3, scenario 9, and scenario 10 (inset (a):3rd-step response).

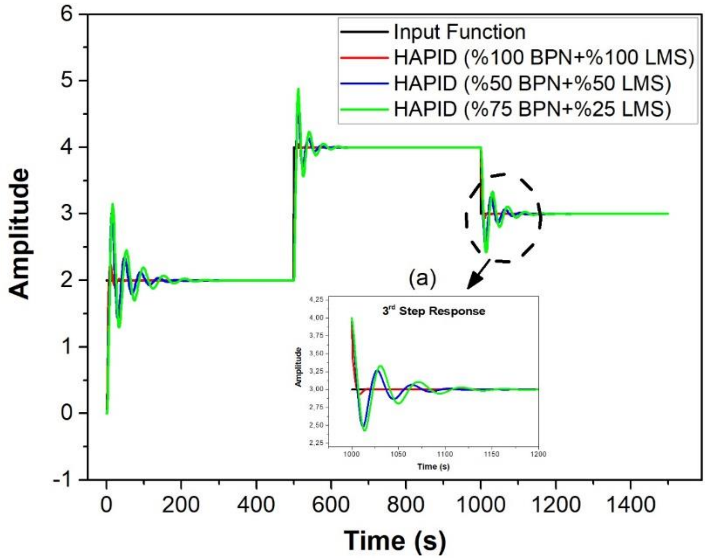

Figure 12. Performance comparison between different hybridization levels for scenario 4, scenario 11, and scenario 12 (inset (a):3rd-step response).

Table 3. Performance comparison of different optimization algorithms with HAPID

	Accuracy (%)	Precision (%)	F1-score (%)
HAPID	97.2	93.6	94.0
LMS	85.2	86.1	86.7
BPN	90.3	91.3	86.9
GSO	95.1	92.2	89.2
CSA	94.0	92.4	91.7
PSO	88.3	87.1	84.4

The trade-off between computational complexity and accuracy for LMS, BPN, and HAPID depends on the specific application in question. For applications with a small number of data samples, the LMS algorithm may be more suitable due to its lower computational complexity. In contrast, for applications requiring high accuracy and complex decision-making, the BPN or HAPID algorithm may be necessary, even though they come with higher computational complexity. To summarize, balancing the trade-off between computational complexity and accuracy is critical when designing and evaluating LMS, BPN, and HAPID. A thorough evaluation of the algorithm's performance on benchmark datasets, including assessments of its accuracy, precision, speed, and scalability, can help determine whether the added complexity is justified in practice.

4. Conclusion

In this study, a novel hybrid approach for adaptively adjusting the K_p, K_i, and K_d parameters of the PID controller using parallel hybridization of LMS and BPN has been proposed for a linear system. Moreover, the effect of the learning rate, initial input, and decreased step has been investigated for overall system responses. Compared to other well-known algorithms, the accuracy, precision, and F1-score are efficiently improved by the proposed approach. The hybrid approach has shown better control performance for a high learning rate with less time than origin algorithms. The response of the hybrid approach is also better for a decreased step input. More comparisons between the different hybridization levels are also figured out. Overall, the HAPID is more adaptable in improving the steady-state error, the controller step response stability, overshoot, rising time, and settling time. Better performance can be obtained by defining a weighting coefficient to decide the contribution of algorithms for each step input. It can also experiment with running algorithms sequentially, such as first BPN then LMS, or by changing this order. One another idea for increasing the performance is the embedding of deep learning in such parallel hybridization. Other implementations and modifications of the BPN/LMS method are under investigation.

Author Contributions

The percentage of the author contributions is present below. The author reviewed and approved final version of the manuscript.

	M.D.
C	100
D	100
S	100
DCP	100
DAI	100
L	100
W	100
CR	100
SR	100
PM	100
FA	100

C=Concept, D= design, S= supervision, DCP= data collection and/or processing, DAI= data analysis and/or interpretation, L= literature search, W= writing, CR= critical review, SR= submission and revision, PM= project management, FA= funding acquisition.

Conflict of Interest

The author declared that there is no conflict of interest.

References

Adar NG. 2021. Real time control application of the robotic arm using neural network based inverse kinematics solution. Sakarya Univ J Sci, 25(3): 849-857.

Akhyar S, Omatu S. 1993. Self-tuning PID control by neural networks. IJCNN '93-Nagoya: Proceedings of 1993 International Joint Conference on Neural Networks, October 25-29, 1993, New York, US, pp: 2749-2752.

Alkrwy A, Hussein AA, Atyia TH, Khamees M. 2021. Adaptive tuning of PID controller using crow search algorithm for DC motor. Mater Sci Eng, 1076: 012001.

Ang K. H., Chong G., Li Y. 2005. PID control system analysis, design, and technology, IEEE Trans. Control Syst. Technol., vol. 13, no. 4, pp: 559-576.

Antony Dhas MM, Chandrasekara S. 2019. Particle swarm intelligence based univariate parameter tuning of recursive least square algorithm for optimal heart sound signal filtering. Gazi Univ J Sci, 32(3): 928-943.

Bai C, Zhang Z. 2018. A least mean square based active disturbance rejection control for an inertially stabilized platform. Optik, 174: 609-622.

Bolton W. 2015. Instrumentation and Control Systems. Newness-Elsevier, New York, US, pp: 99-121.

Carvalho G, Guedes I, Pinto M, Zachi A, Almeida L, Andrade F, Melo AG. 2021. Hybrid PID-Fuzzy controller for autonomous UAV stabilization. 14th IEEE International Conference on

Industry Applications, August 15-18, 2021, São Paulo, Brazil, pp: 1296-1302.

Chen C, Gu GX. 2020. Generative deep neural networks for inverse materials design using backpropagation and active learning. Adv Sci, 7(5): 1-10.

Conker C, Baltacioglu MK. 2020. Fuzzy self-adaptive PID control technique for driving HHO dry cell systems. Int J Hydrogen Ener, 45(49): 26059-26069.

Dogo EM, Afolabi OJ, Nwulu NI, Twala B, Aigbavboa CO. 2018. A comparative analysis of gradient descent-based optimization algorithms on convolutional neural networks. International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS), December 21-22, 2018, Belagavi, India, pp: 92-99.

El-Nagar AM, Zaki AM, Soliman FAS, El-Bardini M. 2022. Hybrid deep learning controller for nonlinear systems based on adaptive learning rates. Int J Control, DOI: 10.1080/00207179.2022.2067080.

Guo B, Liu H, Luo Z, Wang F. 2009. Adaptive PID controller based on BP neural network. First IITA International Joint Conference on Artificial Intelligence, April 25-26, 2009, Hainan Island, China, pp: 148.

Haykin S, Widrow B. 2003. Least-mean-square adaptive filters. John Wiley & Sons, New York, US, pp: 175-241.

Haykin S. 2005. Adaptive filter theory. Pearson Education, Lahor, India, pp: 365-438.

Hernández-Alvarado R, García-Valdovinos LG, Salgado-Jiménez T, Gómez-Espinosa A, Fonseca-Navarro F. 2016. Neural network-based self-tuning PID control for underwater vehicles. Sensors, 16: 1429.

Hou Z, Xiong S. 2019. On model-free adaptive control and its stability analysis. IEEE Transact Auto Control, 64(11): 4555-4569.

Jaleel JA, Thanvy N. 2013. A comparative study between PI, PD, PID and lead-lag controllers for power system stabilizer. International Conference on Circuits, Power and Computing Technologies, March 20-21i 2013, New York, US, pp: 456-460.

Karchi N, Kulkarni DB. 2021. Development and analysis of adaptive PID controller using LMS algorithm for distribution generation-inverter. Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT), September 15-17, 2021, Tamil Nadu, India, pp: 1-7.

Mahmoodabadi MJ, Soleimani T, Sahnehsaraei MA. 2018. A hybrid optimal controller based on the robust decoupled sliding mode and adaptive feedback linearization. J Info Tech Control, 47(2): 295-309.

Moayedi H, Bui DT, Gör M, Pradhan B, Jaafari A. 2019. The feasibility of three prediction techniques of the artificial neural network, adaptive neuro-fuzzy inference system, and hybrid particle swarm optimization for assessing the safety factor of cohesive slopes. ISPRS Int J Geo-Inf, 8(9): 391.

Orozco-Tupacyupanqui W, Nakano-Miyatake M, Perez-Meana H. 2016. A new step-size searching algorithm based on fuzzy logic and neural networks for LMS adaptive beamforming systems. Turk J Electr Eng Comput Sci, 24(5): 4322-4338.

Öztekin E, Ozgan K. 2012. Analysis of thick plates on elastic foundation by back-propagation artificial neural network using one parameter foundation model. Int J Eng Appl Sci, 4(1): 67-76.

Pandey D, Pandey BK, Wairya S. 2021. Hybrid deep neural network with adaptive galactic swarm optimization for text extraction from scene images. Soft Comput, 25: 1563-1580.

Patel VV. 2020. Ziegler-Nichols tuning method. Resonance, 25: 1385-1397.

Qiao J, Han G, Han H, Yang C, Li W. 2017. A hybrid intelligent optimal control system design for wastewater treatment process. J Info Tech Control, 46(3): 382-394.

Spelta MJM, Martins WA. 2020. Normalized LMS algorithm and data-selective strategies for adaptive graph signal estimation. Signal Proces, 167: 107326.

Tamer A, Zellouma L, Benchouia MT, Krama A. 2021. Adaptive linear neuron control of three-phase shunt active power filter with anti-windup PI controller optimized by particle swarm optimization. Comput Elect Eng, 96: 107471.

Verma B, Padhy PK. 2020. Robust fine tuning of optimal PID controller with guaranteed robustness. IEEE Transact Indust Electr, 67(6): 4911-4920.

Zayyani H, Javaheri A. 2021. A robust generalized proportionate diffusion LMS algorithm for distributed estimation. IEEE Transact Circuits Syst II, 68(4): 1552-1556.