Dicle University
Journal of Engineering
https://dergipark.org.tr/tr/pub/dumf
duje.dicle.edu.tr

Research Article

An application for the PID-based optimizer loop: Estimation of the annual production regression models of Malatya’s apricot

Furkan Nur DENİZ

Inonu University, Electrical-Electronics Engineering Department, furkan.deniz@inonu.edu.tr, Orcid No: 0000-0002-2524-7152

ARTICLE INFO

Article history:

Received 19 July 2022
Received in revised form 17 August 2022
Accepted 28 August 2022
Available online 30 September 2022

Keywords:

PID-based optimizer, nonlinear optimization, regression model, apricot, estimation

ABSTRACT

In this study, a data analysis application for the PID-based optimizer loop, which was previously proposed in a former study, is carried out. In this application, quadratic and cubic polynomial regression models were obtained for the estimation of annual apricot production by using the yearly total apricot production data of Malatya between 1991 and 2020. In addition, an average of these regression model estimations was calculated to increase estimation reliability. Annual apricot production amount was estimated by using the regression models obtained with the PID-based optimizer system between 2021-2025. The results were compared with the results obtained with the Matlab curve fitting toolbox.

Doi: 10.24012/dumf.1145295

Introduction

The use of Proportional-Integral-Derivation (PID) laws in control systems has played an important role after the idea of feedback became a great solution for control systems [1]. P, I, D actions in feedback can consider present, past and future control errors, respectively [1]. PID controller performs the minimization of the error signal in feedback control systems.

The use of PID laws in various nonlinear optimization problems have revealed a different and interesting application field besides their conventional use for the minimization of error signals in control systems. For example, it has been presented that the problems related to reaching the global minimum can be solved by providing better convergence speed by using PID actions in the conventional gradient descent algorithm [2]. In another study, a three term backpropagation algorithm was developed by adding a proportional action to the standard backpropagation algorithm, and it was shown to be successful in terms of convergence rate and local minimums for artificial neural networks training [3]. As a PID control system-like approach, stability analyses have been performed for the proposed three term backpropagation algorithm and stability conditions have also been presented [4]. Later, a problem was defined in terms of discrete-time PID actions for the training of neural networks and used for estimation error [5]. Also, apart from feedback control systems, PID actions were used in adaptive parameter settings for the neural networks [6]. There are some studies in the literature in which the PID plays a role in various optimization methods such as stochastic optimization [7]–[9].

In addition to these studies, an effective algorithm has been presented for solving nonlinear, unconstrained and multi-parameter optimization problems using a PID-based optimizer system [10]. The PID-based optimizer system is based on convergence of the error to zero using a slope-sensitive objective function in a closed-loop feedback system [10]. Thus, it has been shown that the consideration of the slope direction of the objective function increases the convergence speed and accuracy of the minimization process [10].

In the current study, second-order (quadratic) and third-order (cubic) polynomial regression models were obtained by using the PID-based optimizer system for the estimation of annual apricot production in Malatya. Average of the estimates of polynomial models was calculated to improve estimation consistency and reliability of these models. Annual apricot production estimates for the years 2021-2025 were calculated by using these polynomial regression models. The obtained results were compared with the 2nd and 3rd order regression models obtained using the Matlab curve fitting toolbox [11]. Previously, estimates of annual apricot production for Turkey were presented using various time series analyses in some previous studies [12], [13]. In this study, an application of the PID-based optimizer system [10] was demonstrated for the polynomial modelling of the annual apricot production data from Malatya region.

The primary aim of this study is to show employment of the PID-based optimizer system in data modelling application. It is observed that the results obtained for the 2nd order polynomial regression models are satisfactory when compared to the results obtained with the Matlab curve fitting toolbox.

However, the modelling results indicated that the performance of the PID-based optimizer in this modelling application depends on initial parameters of the PID system. Another point to be reminded here is that more real data are needed for more consistent time-dependent of the apricot production.

Brief Introduction of the PID-Based Optimizer System

In this section, the use of PID (Proportional-Integral-Derivative) controller as an optimizer in a closed-loop system is presented.

The PID-based optimizer system [10] works to solve unconstrained, multi-parameters and nonlinear optimization problem for $\forall u_1, u_2, u_3, .., u_p \in R$ and $F(u_1, u_2, u_3, .., u_p) \geq 0$ given as

$$\min F(u_1, u_2, u_3, .., u_p) \quad (1)$$

Figure 1 shows a block diagram of PID-based optimizer system. The PID controller promises to reduce the error signal in the closed-loop control system where r is the reference signal and q is the output signal [10].

The closed loop control system makes an effort to bring the error signal closer to zero by considering the error signal $e = r - q$. The PID controller is given by

$$u(t) = k_p e(t) + k_i \int_0^t e(t)dt + k_d \frac{de(t)}{dt} \quad (2)$$

where $u(t)$ is controller signal, and k_p, k_i and k_d are proportional, integral and derivative gains, respectively [14]. In feedback control systems, the error signal $e(t)$ goes to zero when the controller gains are adjusted to optimal values with various tuning methods. The PID-based optimizer configuration is slightly different from the closed-loop control system and is designed as shown in Figure 1 [10].

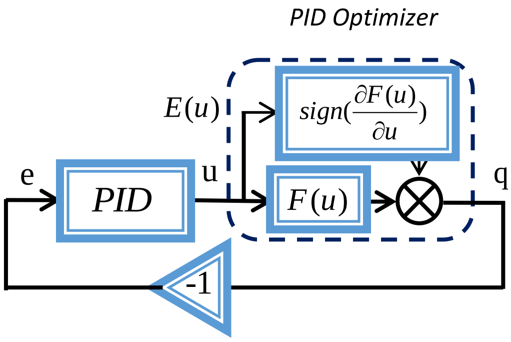

Figure 1. A block diagram of the proposed PID optimizer system [10].

In this configuration, the plant, which is controlled by the PID controller in a closed loop control system, is replaced by the function $E(u)$ which was called a slope sentient function model [10]. The function $E(u)$ is expressed as

$$E(u) = F(u) sign(\frac{\partial F(u)}{\partial u}) \quad (3)$$

where $F(u)$ is an objective function and it is used for optimization process [10]. The value of the sign function is 1 when the slope of the function is positive, and the value of the sign function is -1 when the slope of the function is negative. The value of the sign function is 0 when the objective function arrives at the global minimum points. In this way, the PID-based optimizer loop in Figure1 can achieve the convergence of the objective function's value to zero such that $\lim_{t\to\infty} F(u(t)) \to 0$.

To prove the limitation and convergence of the PID based optimizer system, the boundedness of PID optimizer dynamics and the strong convergence boundaries were explained in detail in [10]. To briefly mention here, the objective function is primarily considered as the Lyapunov energy function. Thus, it has been shown that if the value of the objective function decreases, the value of the derivative of the energy function is also negative. As a result, it has been proven in [10] that the PID optimizer-based system limits the value of the objective function.

In [10], it was proved by using the discrete final value theorem that the error function goes to zero ($\lim_{n\to\infty} e(n) \to 0$) for the discrete time PID based optimizer loop.

The PID controller to be used in a discrete-time optimizer loop can be restated as follows:

$$u(n) = k_p e(n) + k_i \sum_{i=0}^{n} e(n) T_s + k_d \frac{e(n) - e(n-1)}{T_s} \quad (4)$$

In Equation (4), T_s is unit time increment, and the discrete time parameter n is equal to t / T_s. The PID optimizer given in Figure 1 is generalized as in Figure 2 for multi-parameter optimization problems [10].

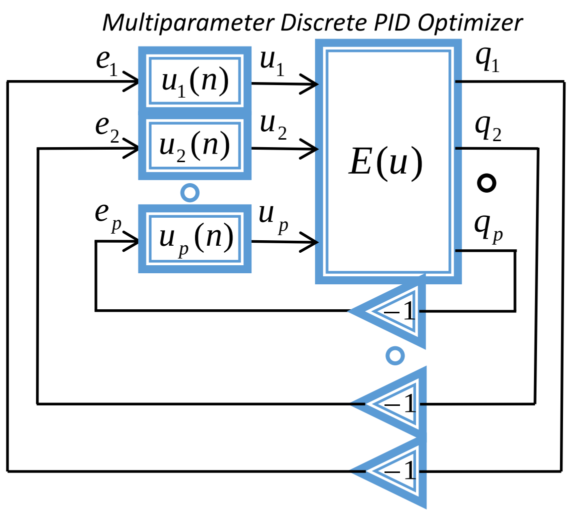

Figure 2. Multiparameter discrete PID optimizer system [10].

A flowchart in Figure 3 is demonstrated for the calculation steps which were explained in [10].

The flowchart can be summarized as follows: Firstly, initial values, PID parameters and maximum iteration numbers are determined. PID parameters are determined by trial-and-error method. Then, the slope sentient function models are calculated, and the outputs of all loops are obtained. The error functions are calculated by multiplying -1 with the outputs over the feedback. Then, the outputs of the discrete time PID optimizer are calculated by using (4) for all loops. If the stopping conditions are satisfied, the algorithm ends, otherwise it returns to the calculation of slope sensitive function models.

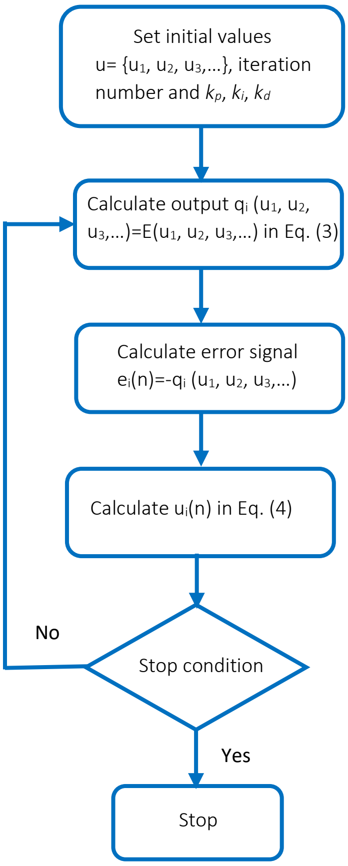

Figure 3. Flowchart of the PID-based optimizer system.

Estimation of the Annual Production Regression Models of Malatya's Apricot

In this section, polynomial models for estimation of Malatya region apricot production are obtained by using PID-based optimizer as nonlinear multi-parameter optimization.

To that end, annual apricot production data given in Table 1 are used. These data are taken from [15] for the years 1991-2004, from [16] for the years 2005-2009 and from [17] for the years 2010-2020. Using these real data given in Table 1, second order and third order polynomial regression models are obtained. Also, the robustness and reliability of the estimated data is improved by average of these polynomial models' data. In addition, the results, which are obtained with the PID-optimizer system, are compared with the results obtained by using the Matlab curve fitting toolbox [11].

Table 1. Apricot production between 1991-2020.

Year	**Production ($x10^3$ tons)**	**Year**	**Production ($x10^3$ tons)**
1991	**154**	2006	**243**
1992	**161**	2007	**268**
1993	**94**	2008	**363**
1994	**263**	2009	**340**
1995	**132**	2010	**221**
1996	**84**	2011	**410**
1997	**144**	2012	**510**
1998	**297**	2013	**412**
1999	**166**	2014	**39**
2000	**331**	2015	**336**
2001	**268**	2016	**381**
2002	**122**	2017	**673**
2003	**217**	2018	**401**
2004	**350**	2019	**392**
2005	**500**	2020	**352**

Second-order Polynomial Regression Model

Second-order (quadratic) polynomial regression model is given by

$$y = ax^2 + bx + c \quad (5)$$

To calculate the coefficients of the second-order regression model, one can solve the following minimization problem [10] by using the years and production data in Table 1.

$$\min_{\{a,b,c\}} F(a,b,c) = \frac{1}{2}\sum_{i=1}^{n}(ax_i^2 + bx_i + c - y_i)^2 \quad (6)$$

where x_i is the year vector [1991,1992, …., 2020] and y_i is the production vector [154, 161, …, 352] listed in Table 1. In addition, the initial values of polynomial coefficients [a_0, b_0, c_0] are set to be [0,0,0]. According to initial configuration of PID optimizer, algorithm can converge to minimum in 100000 iterations and the value of objective function does not change significantly. Therefore, maximum iteration number is set to 100000. Since the values of the year vectors are large values to solve this numerical optimization, the years data are set to appropriate values by applying in the form of $x = x_i - \min(x_i)$. The PID parameters $[k_p, k_i, k_d]$ are set to be [9*e*-17, 6*e*-9,9*e*-17].

Calculating the slope sentient function $E(u)$ and the error signal for each iteration, the PID-based optimizer system works to reach the optimal coefficients of the quadratic regression model.

The following 2nd order polynomial model is obtained with the PID optimizer system considering $x = x_i - \min(x_i)$.

$$y = -0.1807x^2 + 15.5039x + 114.2005 \quad (7)$$

Then, the 2nd order polynomial model in Equation (8) is found with the Matlab curve fitting toolbox.

$$y = -0.1376x^2 + 14.16x + 121.4 \quad (8)$$

In Figure 4, 2nd order polynomial model curves, which are obtained by the PID optimizer system and the Matlab curve fitting toolbox, and the real data are presented comparatively. Also, this figure shows the estimated data for the years 2021-2025. The estimated data for 2nd order polynomial models are given in Table 2. To evaluate data modelling performance, the performance criteria RMSE and MAE are calculated and listed in Table 3. One can say that the PID optimizer-based system performs as well as Matlab curve fitting toolbox when Tables 2 and 3 are analysed.

Figure 5 indicates the residuals of 2nd order polynomial models obtained by the PID optimizer system and Matlab curve fitting toolbox. It can be concluded that the deviations of the PID optimizer system and the Matlab curve fitting toolbox are almost the same. Here, it can be stated that the PID optimizer system will be more successful by choosing more optimal values for the PID parameters $[k_p, k_i, k_d]$.

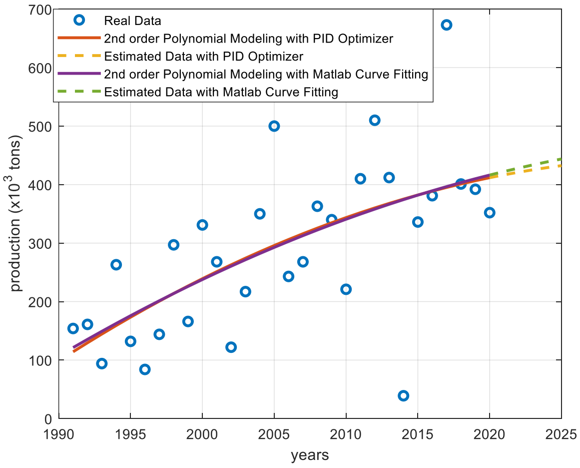

Figure 4. Comparisons of real data with 2nd order polynomial models obtained with PID Optimizer system and Matlab curve fitting toolbox.

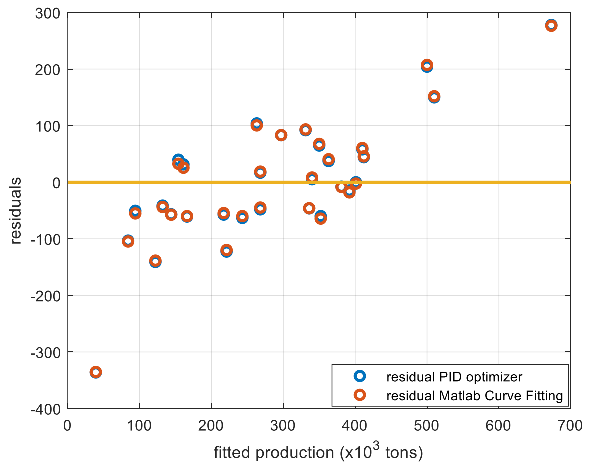

Figure 5. Plots of residuals for 2nd order polynomial models obtained with PID optimizer system and Matlab curve fitting toolbox.

Third-order Polynomial Regression Model

Third-order (cubic) polynomial regression model is given by

$$y = ax^3 + bx^2 + cx + d \quad (9)$$

To estimate the coefficients of the third-order regression model, the minimization problem can be expressed as:

$$\min_{\{a,b,c,d\}} F(a,b,c,d) = \frac{1}{2}\sum_{i=1}^{n}(ax_i^3 + bx_i^2 + cx_i + d - y_i)^2 \quad (10)$$

To solve this optimization problem by using the real data given in Table 1, the initial values [a_0, b_0, c_0, d_0] are set to be [0,0,0,0] and the maximum number of iterations is taken as 100000. The PID parameters $[k_p, k_i, k_d]$ are set to be [1*e*-9, 9*e*-9,20*e*-10].

As in the 2nd order polynomial model estimation, a bias value normalization in the form of $x = x_i - \min(x_i)$ is considered, since the year values are large for the optimization process. Using PID-based optimizer system, third order polynomial model is obtained as

$$y = -0.0052\,x^3 + 0.0061\,x^2 + 13.8659\,x + 93.5877 \quad (11)$$

Also, using Matlab curve fitting toolbox, the third order polynomial model is found as

$$y = -0.0103x^3 + 0.3117x^2 + 9.036x + 132.7 \quad (12)$$

Comparisons of the third order polynomial models given in Equations (11) and (12) and the estimated data for the years 2021-2025 are demonstrated in Figure 6. In addition, estimated apricot productions are given in Table 2. Considering evaluation of the regression models according to performance criteria in Table 3, the third order polynomial model obtained with the PID optimizer system is slightly different from Matlab curve fitting results.

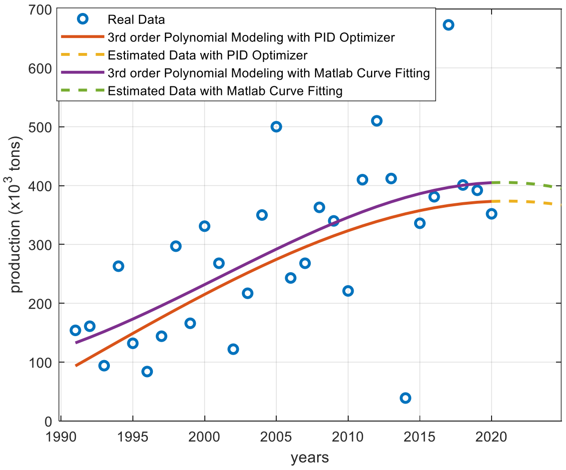

Figure 6. Comparisons of real data with 3rd order polynomial models obtained with PID Optimizer system and Matlab curve fitting toolbox.

The third order polynomial model obtained with Matlab curve fitting has lower RMSE and MAE values. Here, it is clear that the PID parameters $[k_p, k_i, k_d]$ used for the PID optimizer system need better settings. Therefore, the hyper-parameter optimization of the PID-based optimizer was suggested as a future study [10].

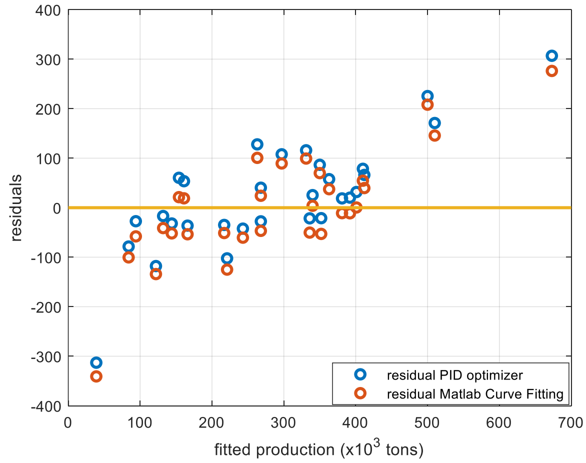

Figure 7. Plots of residuals for 3rd order polynomial models obtained with PID optimizer system and Matlab curve fitting toolbox.

Figure 7 shows the graphs of the residuals of 3rd order polynomial models obtained with the PID optimizer system and Matlab curve fitting toolbox. One can state that the deviations of the PID optimizer system for negative residual values are smaller than the deviations of the Matlab curve fitting toolbox.

Average of the Second-order and the Third-order polynomial models

In this subsection, we used ensemble averaging of the second and the third order polynomial models to improve consistency in the apricot production estimations. Taking average of several estimations from different models can decrease estimation errors when they are randomly distributed. The averaging estimates of models was also discussed for data modelling in [18].

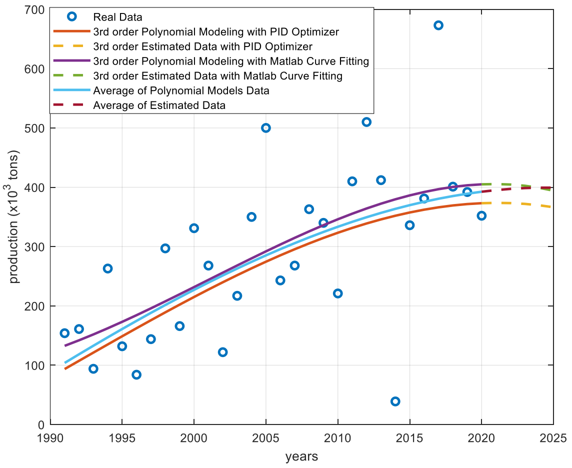

Figure 8. Comparisons of real data with 3rd order polynomial models obtained with PID optimizer system and Matlab curve fitting toolbox and average of polynomial models.

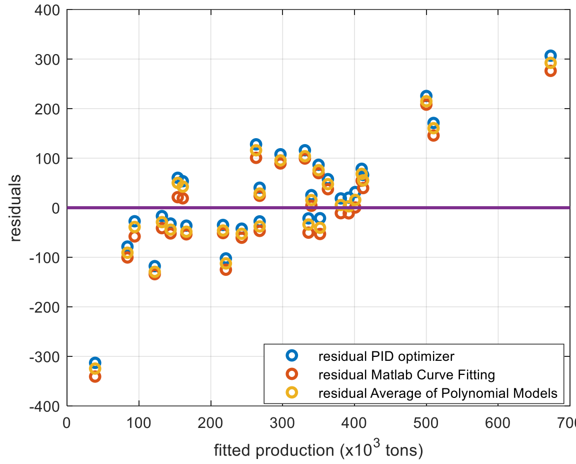

Figure 9. Plots of residuals for 3rd order polynomial models obtained with PID optimizer system and Matlab curve fitting and average of polynomial models.

Figure 8 shows results of the 3rd order models obtained with PID optimizer and Matlab curve fitting and the average of the second and the third order polynomial models. The average estimated data for the years 2021-2025 are demonstrated with dashed lines in the figure and listed in Table 2. RMSE and MAE values are given in Table 3. Considering the RMSE and the MAE values, it is observed that the average of the polynomial models is closer to the 3rd order polynomial model obtained with Matlab curve fitting and this indicates performance improvements of ensemble average models. Figure 8 clearly shows that averaging models from PID optimizers can be used to improve 3rd order estimation performance. The residuals of the 2nd and 3rd order polynomial models and average of these models are illustrated in Figure 9.

Table 2. Estimated Data of the Polynomial Models

Year	**Production ($x10^3$ tons)**				
	2nd order curve fitting	2nd order PIDO	3rd order curve fitting	3rd order PIDO	Average of estimate PIDO
2021	422	**417**	405	**374**	**395**
2022	428	**421**	405	**373**	**397**
2023	434	**425**	403	**372**	**399**
2024	439	**429**	399	**370**	**399**
2025	444	**432**	394	**366**	**399**

Table 3. Evaluation of the Polynomial Models

	2nd order curve fitting	2nd order PIDO	3rd order curve fitting	3rd order PIDO	Average of estimate PIDO
RMSE	110.5	**110.54**	110.37	**113.15**	**111.18**
MAE	80.84	**80.59**	79.27	**82.19**	**80.64**

Conclusions

In this study, the PID-based optimizer system was used to estimate the annual production of Malatya's apricot. Second order and third order polynomial regression models were obtained with the PID-based optimizer system and compared with the polynomial models which are obtained with the Matlab curve fitting toolbox. Also, the average of the polynomial models obtained with PID-based optimizer are calculated. As a result, the second order polynomial regression model is as successful as the Matlab curve fitting toolbox's model. However, the third-order polynomial regression model needs to be improved. To improve the performance of the third order polynomial regression model, the PID parameters should be optimally chosen by considering each loop of the PID-based optimizer system. This issue can be addressed in a future study.

Ethics Committee Approval

There is no need to obtain permission from the ethics committee for the article prepared.

Conflict of Interest Statement

There is no conflict of interest with any person / institution in the article prepared.

References

[1] K. J. Åström and T. Hägglund, "The future of PID control," *Control Eng. Pract.*, vol. 9, no. 11, pp. 1163–1175, Nov. 2001, doi: 10.1016/S0967-0661(01)00062-4.

[2] R. Vitfhal, P. Sunthar, and C. H. D. Rao, "The Generalized Proportional-Integral-Derivative (PID) Gradient Descent Back Propagation Algorithm," *Neural Networks*, vol. 8, no. 4, pp. 563–569, 1995.

[3] Y. H. Zweiri, J. F. Whidborne, and L. D. Seneviratne, "A three-term backpropagation algorithm," *Neurocomputing*, vol. 50, pp. 305–318, 2003.

[4] Y. H. Zweiri, L. D. Seneviratne, and K. Althoefer, "Stability analysis of a three-term backpropagation algorithm," *Neural Networks*, vol. 18, pp. 1341–1347, 2005, doi: 10.1016/j.neunet.2005.04.007.

[5] X. Jing and L. Cheng, "An Optimal PID Control Algorithm for Training Feedforward Neural Networks," *IEEE Trans. Ind. Electron.*, vol. 60, no. 6, pp. 2273–2283, 2013.

[6] G. Zeng, X. Xie, M. Chen, and J. Weng, "Adaptive population extremal optimization-based PID neural network for multivariable nonlinear control systems," *Swarm Evol. Comput.*, vol. 44, pp. 320–334, 2019, doi: 10.1016/j.swevo.2018.04.008.

[7] J. Li, Y. Yuan, T. Ruan, J. Chen, and X. Luo, "A proportional-integral-derivative-incorporated stochastic gradient descent-based latent factor analysis model," *Neurocomputing*, vol. 427, pp. 29–39, 2021, doi: 10.1016/j.neucom.2020.11.029.

[8] H. Wang, Y. Luo, W. An, and Q. Sun, "PID Controller-Based Stochastic Optimization Acceleration for Deep Neural Networks," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 31, no. 12, pp. 5079–5091, 2020.

[9] D. Wang, M. Ji, Y. Wang, H. Wang, and L. Fang, "Spi-Optimizer: An Integral-Separated PI Controller For Stochastic Optimization," in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, no. 61722209, pp. 2129–2133.

[10] B. B. Alagoz, F. N. Deniz, and M. Koseoglu, "An efficient PID-based optimizer loop and its application in De Jong's functions minimization and quadratic regression problems," *Syst. Control Lett.*, vol. 159, p. 105090, 2022, doi: 10.1016/j.sysconle.2021.105090.

[11] "MATLAB Release 2020b, The MathWorks, Inc., Natick, Massachusetts, United States." 2020.

[12] B. K. Topuz, M. Bozoğlu, U. Başer, and N. A. Eroğlu, "Forecasting of Apricot Production of Turkey by Using Box-Jenkins Method," *Turkish J. Forecast.*, vol. 02, no. 2, pp. 20–26, 2018.

[13] A. S. Uzundumlu, T. Karabacak, and A. Ali, "Apricot production forecast of the leading countries in the period of 2018-2025," *Emirates J. Food Agric.*, vol. 33, no. 8, pp. 682–690, 2021, doi: 10.9755/ejfa.2021.v33.i8.2744.

[14] K. Ogata, *Modern Control Engineering*, Prentice H. 2002.

[15] K. Çatı and S. Yıldız, "Türkiye'de Kuru Kayısı Üretim Ve Pazarlama Problemleri Ve Çözüm Önerileri," *İktisadi ve İdari Bilim. Derg.*, vol. 21, no. 1, 2007.

[16] Fırat Kalkınma Ajansı, "Kayısı araştırma raporu," 2010.

[17] Tarımsal Ekonomi Ve Politika Geliştirme Enstitüsü TEPGE, "Ürün Raporu-Kayısı," 2021.

[18] D. Ari and B. B. Alagoz, "Modeling Daily Financial Market Data by Using Tree-Based Genetic Programming," in *2021 International Conference on Information Technology (ICIT)*, 2021, pp. 382–386.