An Intelligent Machine Condition Monitoring Model for Servo Systems

Hayri Mutlu, Mustafa Caner Akuner, and Gazi Akgun

Abstract — **The installation of industrial servo systems and the determination of control parameters are limited to the skills and knowledge of the commissioner. In addition, commissioned systems are often not re-optimized if environmental influences or loads change. The goal of this research is to create an artificial neural network (ANN) model for servo systems that will keep the servo system's proportional, integral, and derivative (PID) parameters working optimally. For this process, a machine condition monitoring algorithm developed with the ANN technique, which uses the data such as actual current, torque, power, position to be obtained from the servo system on an industrial controller, for the control and rearrangement of the parameters.**

Index Terms — **Servo System, Artificial Neural Network, PLC, ProfiNET**

I. Introduction

SERVO MOTORS are preferred in the industry because of their fast and precise positioning features [1]. Servo motors are controlled by servo drives that contain many adjustable parameters that will significantly affect their response [2]. The whole formed together with the servo motor and the driver used for its control is described as a servo system. Servo systems are time-varying and non-linear control structures [3] [4].

Servo systems must be installed in accordance with the mechanics and load characteristics they drive. PID controllers are widely used in industrial settings. The basic concept of PID controller is to regulate error, calculated with desired and actual state set to zero. To compensate the system response, the error proportional gain (Kp), as well as derivative (Kd) and integral gains (Ki), are used for calculate the control signal (u_c) as seen in Eq.1.

$$u_c = K_p e + K_d \frac{de}{dt} + K_i \int_0^t e(t)dt \qquad (1)$$

One method for calculate control gains is to manually determine the parameters by an expert, and the other is to use auto-tune features in the servo drives if they are supported in the industrial applications. The commissioning of industrial servo systems and the determination of their parameters are limited to the skills and knowledge of the technical personnel. Research for the auto-tune feature has also shown that it does not fully design gain parameters for different servo equipment [2]. It simply assigns relatively safe and rigid parameter values based on the inertia imposed on the motor, which can sacrifice some of the controller performance. Therefore, the gain parameters still need to be manually fine-tuned through an expert to ensure the response meets the needs of the application. In general, there is no specific method of adjusting the gain. It depends on the experience of the specialist and the adjustment procedure is also time consuming [2]. Furthermore, when environmental influences or loads change, commissioned systems are frequently not re-optimized.

It is aimed to develop an adaptive machine condition monitoring algorithm that will enable these systems to work most efficiently under all conditions. Reduce the need for expert personnel in the installation step of the systems, and provide more accurate PID parameters value assignment by detecting the errors.

The adaptive PID controller developed with artificial neural networks instead of traditional PID tuning methods and algorithms produces better results in dynamic response performance in nonlinear systems in the industry, according to simulation and experiments [5], [6]. As a result, the machine condition monitoring algorithm is built using artificial neural networks, which is a more innovative model for evaluating and recalculating the current and speed PID gain parameters in the servo driver.

Current, torque and vibration data are generally preferred for different types of motors to feed artificial neural networks [7] [8]. However, data obtained exclusively from servo drives is

HAYRI MUTLU, is with Department of Mechatronics Engineering University of Marmara University, Istanbul, Turkey, (e-mail: hayri.mutlu@gmail.com).
https://orcid.org/0000-0003-0584-3827

MUSTAFA CANER AKUNER, is with Department of Mechatronics Engineering University of Marmara University, Istanbul, Turkey, (e-mail: caner.akuner@gmail.com).
https://orcid.org/0000-0001-8397-3454

GAZI AKGUN, is with Department of Mechatronics Engineering University of Marmara University, Istanbul, Turkey, (e-mail: gazi.akgun@marmara.edu.tr).
https://orcid.org/0000-0002-8154-5883

Manuscript received Nov 04, 2021; accepted Dec 17, 2021.
DOI: 10.17694/bajece.1018947

preferred in this study so that the relevant algorithm could be applied to existing devices without the need for a new sensor or equipment. In order to obtain the training data to be used in the training of the ANN, an experimental setup is designed, which will enable it to be obtained via the servo driver and apply different load characteristics. After the obtained data are processed, the training process is completed with the Matlab Neural Net fitting tool and using the Levenberg-Marquardt method [9] [10].

Programmable Logical Controller (PLC) is often used as a real-time controllers of automation systems and applications due to its flexibility, reliability, and relatively low-cost advantage in controlling complex systems [11],[12]. They are primarily used in the control of both machinery and industrial processes as an essential component of industrial automation systems. Rather than running the developed algorithm on a computer or a different controller, it is hoped that it will be integrated into PLCs and applied to existing systems without requiring new hardware. PLCs are not developed to run artificial intelligence software, but it is claimed that PLCs run ANN algorithms and meet the performance expectation [10] [13]. As a result, the developed ANN algorithm has been converted into a program that will be executed by the PLC.

PLC should communicate with the servo driver for acquisition of data to feed ANN algorithm. Also, the cross data to be used to feed the ANN must be acquired with simultaneous and deterministic time intervals. ProfiNet IRT real-time communication protocol is preferred as the protocol to meet this requirement. When compared with similar protocols in motion control applications in the studies examined, it has been observed that it stands out compared to other protocols, especially thanks to its low latency and deterministic structure [14] [15] [16]. In the consideration of all these information, a PLC that supports ProfiNet IRT protocol, and a servo system are recommended as hardware for the intelligent machine condition monitoring algorithm to be implemented for the problem that is aimed to be solved.

Ziegler Nichols Z-N method, which is widely used in the industry, is preferred to adjust the PID controllers to test and compare the performance of the developed system [17].

II. METHOD

The servo system is first commissioned and made operational in this study, after which experiments are carried out with various load characteristics and PID parameters to obtain data. As shown in Fig.1, the obtained data are used in ANN training. The performance of the ANN is tested by experiments obtained with different loads that are not used in the training process. The ANN model is redesigned as a Simulink program. PLC codes are generated from the Simulink program. ANN algorithm has been added to this program on top of data acquisition and parameter writing functions from the servo system. Then, the tests are repeated with the load characteristics not used in the training in the full experimental setup, and the results are compared with the classical methods.

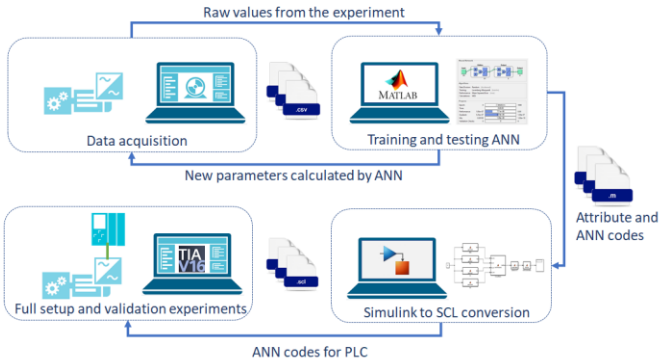

Raw values from the experiment
Data acquisition
.csv
MATLAB
Training and testing ANN
New parameters calculated by ANN
.m
Attribute and ANN codes
TIA V16
Full setup and validation experiments
.scl
Simulink to SCL conversion
ANN codes for PLC

Fig.1. Flow Chart

A. *Servo Systems*

Servo systems can be defined as the sum of a synchronous motor with a feedback and an electronic power and control units capable of driving this motor in closed-loop servo mode.

Motor drivers consist of two main structures, a control unit that enables the motors to be controlled with different control techniques, and a power unit that will provide the electrical energy that will enable the motor to move.

We can define it as control units for structures that control the outputs of the power unit with v/f, vector, and servo control characteristics, which are mostly preferred in the industry. These control units may have input terminals and/or communication interfaces that allow them to receive motor movement orders directly from their own control unit as well as from an upper control unit.

As seen in the Fig.2, in the servo control type, it consists of 3 control loops, which enable to control the position, speed and current of the motor. The motion of the motor is controlled by cascade PI control scheme in the servo driver we have chosen for this control process.

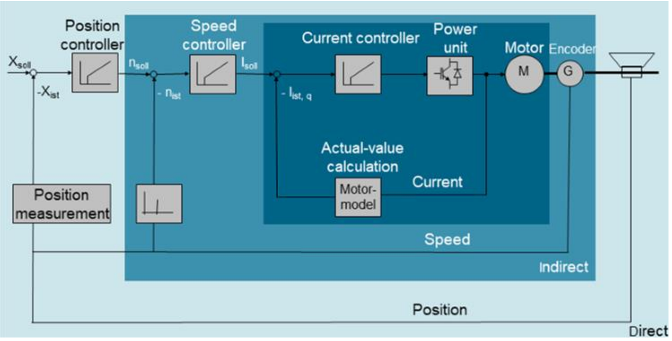


Fig.2. Servo drive control diagram

B. *Artificial Neural Networks*

Algorithms that imitate the work of the nerves in the brain are called artificial neural networks. These networks consist of artificial neurons connected to each other by synapse-like structures. The connection between each neuron carries information from that neuron to the other neuron. The neuron that receives the information performs the necessary operation and transfers this information to the next neuron [18]. It is developed by Frank ROSENBLATT and an artificial nerve called “perceptron” connects the inputs from the external environment or other cells to the cell with weights. Then It calculates the sum of the weighted inputs and the bias parameter, which is the criterion of how easily the cell can be stimulated. The output of the artificial nerve is calculated by activation function with v_k, sum of the weighted input. [19].

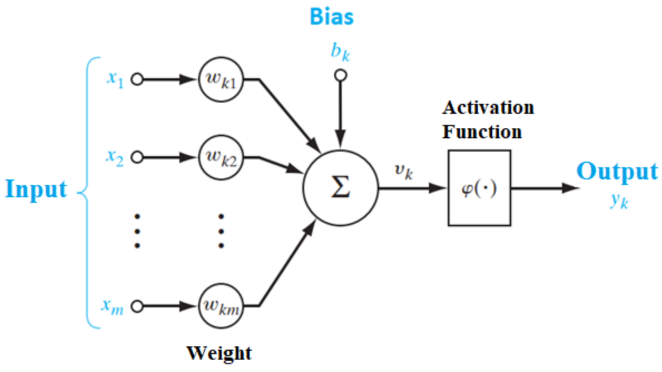


Fig.3. Perceptron model

1) *Training Artificial Neural Networks*

The process of determining the weight values and bias coefficients in the structure of each neural network that makes up the ANN is called training the ANN. Calculating the local minimum point in the cost function of minimized error between actual value and input data set is required for this process. There are several different techniques used in calculating. Some of the techniques we can use include Levenberg-Marquardt (LM), Bayesian Regularization, and Scaled Conjugate Gradient. The LM algorithm, which is another type of Newton's algorithm, is used between ANN and feedforward back propagation algorithms. The LM algorithm is an advantageous one in that it converges quickly, contains few parameters, and operates only with first-order partial derivatives [20]. The LM algorithm is defined in the Eq.2 below.

$$\Delta w(k) = -(J_k^T J_k + \mu_k \mathrm{I})^{-1} J_k^T e_k$$
$$W(k+1) = W(k) + \Delta w(k) \qquad (2)$$

Where W is the weight vector, I is the unit matrix, μ is the combination coefficient. As stated in Eq. 4, the error matrix and Eq. 3 represent the detailed Jacobian matrix and consist of first order derivatives of the error matrix according to the weights [20].

$$J = (PxM)xN \qquad (3)$$

Where P is the number of training samples, M is the number of outputs, and N is the number of weights (the number of hidden layer neurons).

$$e = (PxM)xI \qquad (4)$$

Where P is the number of training samples, M is the number of outputs, and I is the unit matrix and the e is error vector

2) *Creation of the Data Set*

A data set is the labeled and ready-to-use data obtained from the real world or simulations to train ANN. In addition, the values that show some of the specially defined features of the data set, which are mostly used as vectors, are called the features of the data [21].

As we mentioned before, the experimental data is exported from Starter, which is the Siemens S120 servo drive commissioning software, is labeled and transferred to MATLAB. In addition, the normalization of the experimental data is performed. After this process, a MATLAB function is used again to generate the features of the experimental data. In this part root mean square (RMS), integrated absolute value (IAV), mean absolute value (MAV) functions of the data, which are seen in equations 5 to 8, respectively, are used.

$$\mathrm{x_{rms}} = \sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n}} \qquad (5)$$

$$\mathrm{x_{wl}} = \sum_{i=1}^{n} |\mathrm{x_i} - \mathrm{x_{i-1}}| \qquad (6)$$

$$x_{iav} = \sum_{i=1}^{n} |x_i| \qquad (7)$$

$$x_{mav} = \frac{\sum_{i=1}^{n} |x_i|}{n} \qquad (8)$$

In Fig.4, the data obtained from the experiment performed after adjusting the PID values in the no-load condition of the servo system are shown graphically.

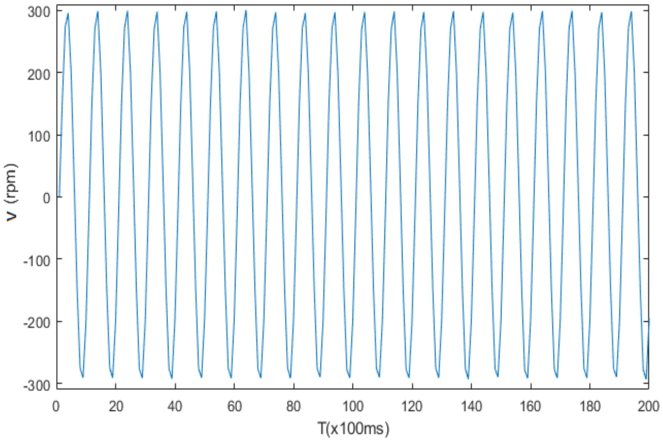


Fig.4. Example of raw speed data read from the drive

The raw data obtained are normalized before being used to train the ANN, and then the RMS values are calculated by shifting 32 samples in a sampling window consisting of 256 values. Windowing functions are functions that multiply signal segments. With these functions, it is ensured that the center parts of the signal window parts are highlighted. The graph of this feature data used for training of ANN is also shown in Fig.5.

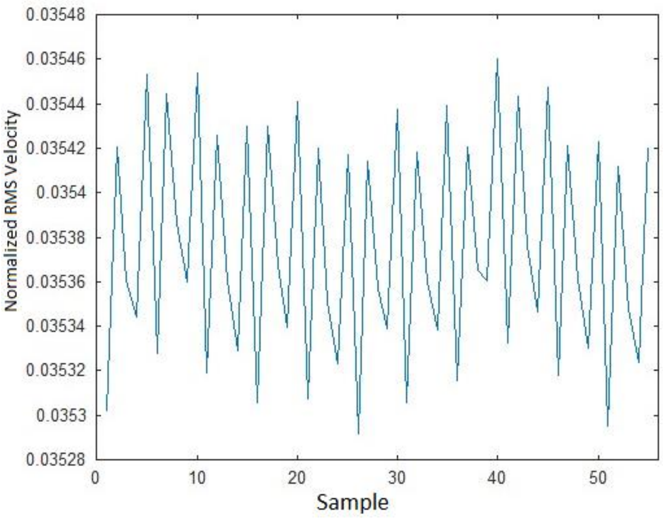


Fig.5. Example of RMS attribute calculated velocity data

3) *Creating ANNs with MATLAB*

In the training process of ANN, Neural Net Fitting tool integrated into MATLAB is used to solve the data fitting problem by using a two-layer feed-forward network. This tool is an application component that allows us to separate the training data, validation data and test data, train according to the selected algorithm, validate the training, define the network architecture, and train the network. In addition, this tool produces many outputs that allows us to evaluate training performance, such as histogram of errors and regression analysis. Finally, it can generate it as a MATLAB function in the completed ANN algorithm. The ability to produce an integrated solution in ANN training has enabled us to prefer this tool in ANN training. With the regression, performance analysis and experiments using histograms, which can be seen in Fig.6 , the training validation and test data rates and the number of hidden layers has been optimized.

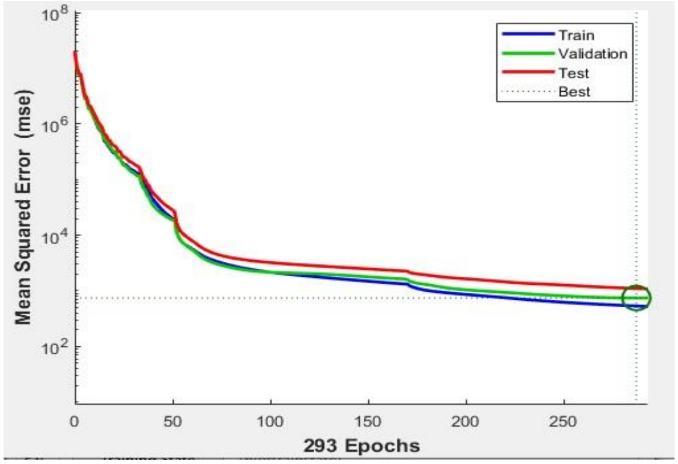


Fig.6. Training performance graph

Furthermore, the generated ANN is tested and verified using a variety of experiments that are not included in the training data.

4) *Generating Simulink Code*

To run the obtained ANN model on the PLC, the algorithms used in MATLAB/Simulink must be converted to a coding language supported by PLCs for feature calculation, creation of the data array that will feed the ANN, and singularization of the result. For this process, the PLC code generator function available in Simulink is used. Then, MATLAB functions are transferred to the Simulink and the block structure as seen in Fig.7 is established.

In this structure, *X* refers to the data taken from the real world. In the second part, the feature calculation is made and the following *ConcatMatrix* is combined with the ANN feed data array with the calculated features. In the third operation, ANN is run and recalculated PID values are found, and these data are sent to *OutConsantrator* block to make a single data group to be uploaded to the Servo system.

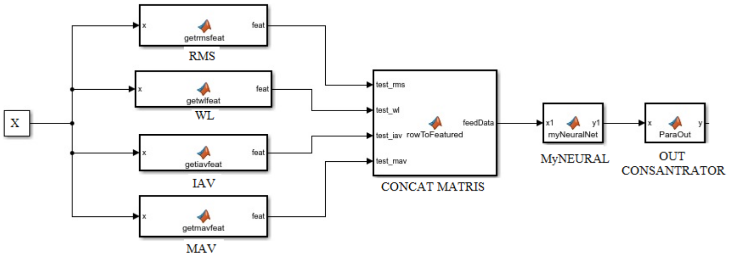


Fig.7. Simulink program

Then, each function is converted one by one into the SCL language specified in PLC programming standards.

C. *Industrial Control Systems*

PLCs, one of the control systems used in industrial environments, and ProfiNet, one of the communication protocols developed to meet industry needs, are preferred as stated in the introduction of this study.

1) *PLC*

Programmable logic controllers are devices designed to operate in a wide temperature band, in industrial environments with vibration and electromagnetic noise, and have a wide range of uses from machine automation to process automation due to their stable working structure. The two main parameters that determine the cycle rate of these devices are the program memory unit size and one command processing speed. These parameters are designed to meet the needs of the specific area where PLCs will be used in industry. There are compact models with input and output terminal units, as well as fully modular PLCs.

In this study, Siemens S7-1500 series product is preferred. This device is a modular PLC, designed for the control of mid-to-high-end machines with 300KB programming memory and 40 ns bit-instruction speed. The MATLAB Simulink PLC code generator tool supports this device. It also supports the ProfiNet IRT feature without requiring any additional hardware.

These controllers are used in industry generally with a single core CPU. PLCs with this design cannot perform multitasking at the single time. Predefined interrupts are used in operating systems when this type of multitasking is desired. In cases where it is desired to perform an operation with simultaneous time intervals, PLCs' that designed with cyclic interrupts can be chosen.

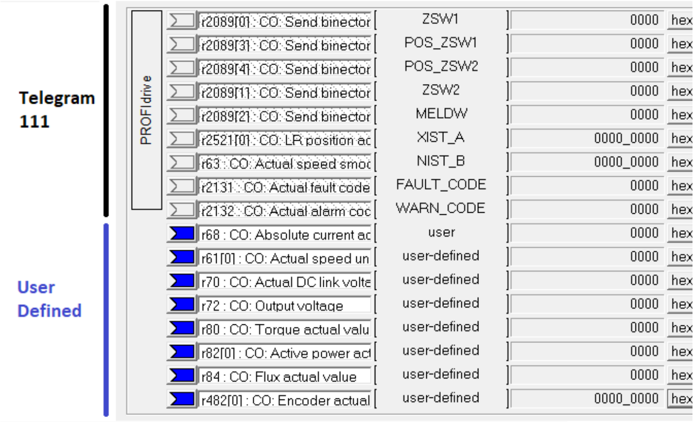
2) ProfiNet

ProfiNet is a real-time communication protocol that uses a standard Ethernet interface and communicates at Level 2 from the OSI layers. ProfiNet can also be defined as a set of protocols consisting of different sub-protocols customized according to application needs. ProfiEnergy, ProfiSafe, Profinet IO can be given as an example.

ProfiNet IRT, on the other hand, is defined as an advanced type of real-time communication protocol that guarantees communication data to be sent and received within a predetermined synchronization time. This customized protocol is preferred especially in motion control operations because of its speed and prevention of data loss. The most important thing to remember about using this protocol is that it requires ProfiNet IRT support on both the controller and the devices to be controlled, as well as all network elements that connect these two types.

In this study, the IRT synchronization time is determined as "*1 ms"* and the configuration is made as specified in the network settings of the PLC, to ensure that the number of devices we communicate with is small and to provide the best possible speed of communication.

The ProfiNet Siemens standard telegram 111, which will be used in communication with the servo driver, is used to send motion orders to the motor and to upload the calculated PID parameters to the driver. However, to transfer the actual measurement values required to feed the ANN to the PLC, a user-defined 8-word data has been added to the tail of the transmit section of this telegram. Fig.8 shows the telegram configuration made in the Starter software.



Telegram 111	PROFIdrive	r2089[0] : CO: Send binector	ZSW1	0000	hex
		r2089[3] : CO: Send binector	POS_ZSW1	0000	hex
		r2089[4] : CO: Send binector	POS_ZSW2	0000	hex
		r2089[1] : CO: Send binector	ZSW2	0000	hex
		r2089[2] : CO: Send binector	MELDW	0000	hex
		r2521[0] : CO: LR position ac	XIST_A	0000_0000	hex
		r63 : CO: Actual speed smoc	NIST_B	0000_0000	hex
		r2131 : CO: Actual fault code	FAULT_CODE	0000	hex
		r2132 : CO: Actual alarm coc	WARN_CODE	0000	hex
User Defined		r68 : CO: Absolute current ac	user	0000	hex
		r61[0] : CO: Actual speed un	user-defined	0000	hex
		r70 : CO: Actual DC link volte	user-defined	0000	hex
		r72 : CO: Output voltage	user-defined	0000	hex
		r80 : CO: Torque actual valu	user-defined	0000	hex
		r82[0] : CO: Active power act	user-defined	0000	hex
		r84 : CO: Flux actual value	user-defined	0000	hex
		r482[0] : CO: Encoder actual	user-defined	0000_0000	hex

Fig.8. ProfiNet telegram configuration

III. EXPERIMENTAL SETUP

An experimental setup has been designed in which different load characteristics can be applied to train the proposed ANN. For this purpose two screws are added to a composite wheel that we attached to the motor shaft, and various weights are placed on these screws, and experiments are conducted. The experiments are started with the condition where the motor is completely unloaded, respectively. The motor tune process has been completed under these conditions. After this process, data record for 3.3 seconds with an input signal as sinusoidal velocity-position function depending on the time that we had previously determined with the same physical conditions. In this data, 8 values to be used to train the ANN are cross sampled with 100 millisecond time intervals. After this step, the related experiment is repeated by changing the speed and current PID parameters, respectively. Then the experiments are continued with a balanced load placement. The same test procedure is applied and repeated for the new physical conditions. Next step, an unbalanced load is created by attaching a single weight, and the relevant samples are taken by repeating the same test procedures. Finally, to create a variable momentum, the load is placed on the screw with a sliding motion, and the experimental procedures are repeated, and the data to be train ANN is collected and recorded. In addition, to better feed the ANN, the tuned values obtained in the experiments are used in cross conditions, and training data are obtained. After obtaining the relevant data, the ANN model is trained.

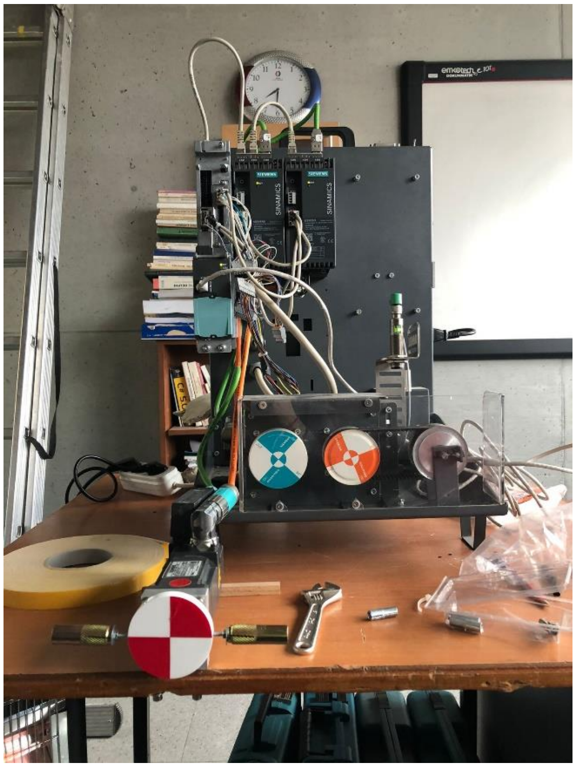


Fig.9. Experimental setup for ANN training

As seen in the flow chart in Fig.10, during the training of the ANN, the experiments are repeated by changing the test weight to create an independent data for the tests, both by attaching two weights for the balanced load and by attaching a single weight for the unbalanced load. The values obtained from the ANN are compared with these values, and the performance of the ANN is decided, and the most suitable ANN is decided by testing the experiments in a real experimental setup with reduced weight for the 3 most suitable ANN designs. The traction current of the motor and the tracking error in the speed data are used and the most successful ANN model parameters are embedded to the industrial controller.

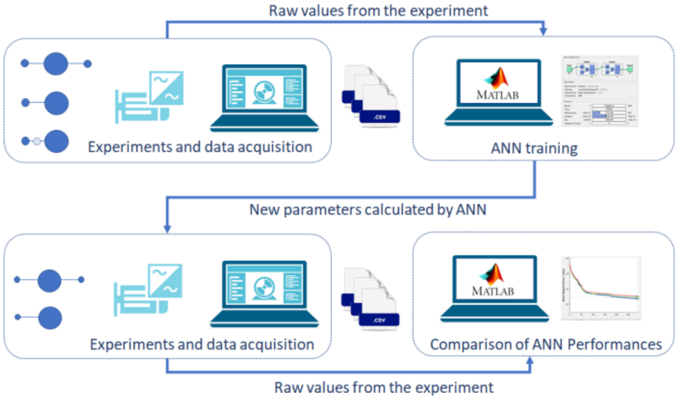

Raw values from the experiment
Experiments and data acquisition
.csv
MATLAB
ANN training
New parameters calculated by ANN
Experiments and data acquisition
.csv
MATLAB
Comparison of ANN Performances
Raw values from the experiment

Fig.10. Flow chart for ANN training

Finally, in our experimental setup, the servo system is arranged to communicate with ProfiNet IRT via PLC to test the proposed system. The PLC program is rearranged to send the motion command to the servo system. In addition, the actual measurement values that ANN will use are read and the results produced by ANN are write to the relevant PID parameters.

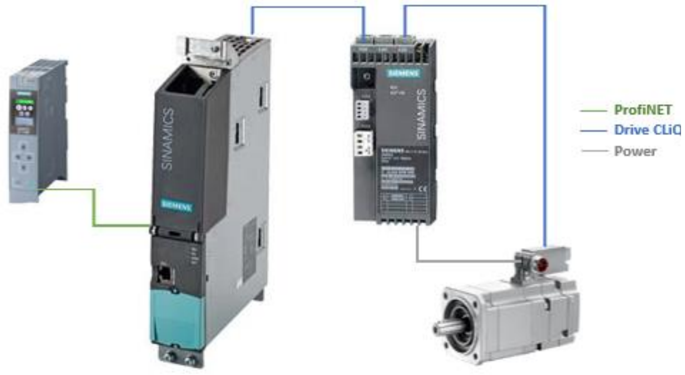


Fig.11. Components of experimental setup

Samples are taken in full experimental setup with a lighter and balanced loaded weight that is not used in ANN trainings in the newly arranged experimental set. The results of the experiments are compared in the discussion, which is the next section of this study.

IV. DISCUSSIONS

Experiments are repeated using Z-N tuning techniques, which are commonly used in the industry, to compare the results of the previous experiments with the results of the most recent experimental setup. First and foremost, it should be noted that the Z-N technique is an experimental technique that begins with observational data and is then constrained by the knowledge and abilities of the person who commissioned the system for optimization. As a result, following the Z-N process, the current and speed values are monitored online, the driver is re-adjusted using manual optimization techniques, and the experiment is repeated. As shown in Figure 12, three different results are obtained, with which we can compare the data. First, we look at the feedback data obtained with the speed control set value and the mean squared error (MSE) data generated in the three systems. As seen in Figure 12 and Table 1, the follow-up errors in the values obtained with the ANN technique as 2010 which is much less than Z-N techniques. This shows that with these values calculated by ANN and their continuous control and regulation, better results can be obtained completely independent of the capabilities of the person who commissioned them.

TABLE I
SPEED MSE COMPARISON TABLE

	Z-N	Optimized	ANN
MSE Speed	171978	23549	2010

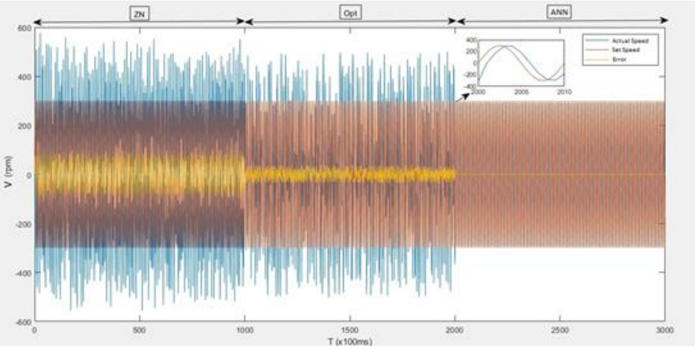


Fig.12. Velocity-Time comparison trend

Another control data is the current of the motor. In the motor driven by using the parameters obtained with the Z-N technique, the motor current was measured as 3.405 amperes on average. By using manual optimization techniques, the motor current has been reduced to an average of 2.952 amps. In the experiment performed using the parameters obtained from the ANN technique, the motor current was measured as 0.048 ampere on average. As seen in Figure 13, current is much less and stable in the same motion characteristic. We can say that the motors controlled by the values calculated with the ANN work more efficiently by looking at the results of the relevant experiments.

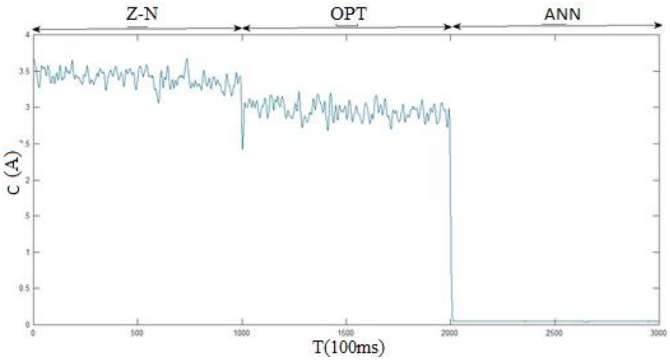


Fig.13. Current-Time comparison trend

V. CONCLUSION

Given these facts, we can conclude that the ANN technique can be used to calculate the PID coefficients of servo systems and that these ANN algorithms can be successfully implemented within certain limits in the control objects known as PLCs. Furthermore, one of the most important features of this system is that it collects data from the system at sampling intervals of every 3.3 seconds and updates the relevant parameters with the new value calculated by the ANN algorithm based on the new situation it observes It demonstrates that the system has an adaptive structure, and that the proposed intelligent method can be used to determine the optimal control parameters, especially in systems where power and torque demand can change over time.

REFERENCES

[1] «Jaen-Cuellar, A. Y., de J. Romero-Troncoso, R., Morales-Velazquez, L., & Osornio-Rios, R. A. (2013). PID-controller tuning optimization with genetic algorithms in servo systems. International Journal of Advanced Robotic Systems, 10(9), 324.».

[2] «Chen, C. W., Chang, L. K., Liao, Y. T., Chung, C. H., Su, W. C., Chen, K. S., & Tsai, M. C. (2020, November). Tuning of Servo Drive Controller Based on Boosted Tree Model and Particle Swarm Optimization. In 2020 23rd International Conference on Electrical».

[3] «Lin, Q. S., Yao, Y. F., & Wang, J. X. (2010, November). Simulation and application of neural network PID auto-tuning controller in servo-system. In 2010 2nd International Workshop on Database Technology and Applications (pp. 1-4). IEEE.».

[4] Firoozian, R. (2014). Servo motors and industrial control theory. Springer..

[5] «Jingjing, X., & Jiaoyu, L. (2013, May). Neural network PID controller auto-tuning design and application. In 2013 25th Chinese Control and Decision Conference (CCDC) (pp. 1370-1375). IEEE.».

[6] «Aftab, M. S., & Shafiq, M. (2015, February). Adaptive PID controller based on Lyapunov function neural network for time delay temperature control. In 2015 IEEE 8th GCC Conference & Exhibition (pp. 1-6). IEEE.».

[7] «Kumar, S., Mukherjee, D., Guchhait, P. K., Banerjee, R., Srivastava, A. K., Vishwakarma, D. N., & Saket, R. K. (2019). A comprehensive review of condition based prognostic maintenance (CBPM) for induction motor. Ieee Access, 7, 90690-90704.».

[8] «Samhouri, M., Al-Ghandoor, A., Ali, S. A., Hinti, I., & Massad, W. (2009). An intelligent machine condition monitoring system using time-based analysis: neuro-fuzzy versus neural network. Jordan Journal of Mechanical and Industrial Engineering, 3(4), 294-».

[9] Watson, G. A. (Ed.). (2006). Numerical analysis: proceedings of the Biennial Conference held at Dundee, June 28-July 1, 1977 (Vol. 630). Springer..

[10] «Duymazlar, O., Engin, M., & Engin, D. (2020, June). Embedded Artificial Neural Network on PLCs to Predict Nonlinear System Responses. In 2020 9th Mediterranean Conference on Embedded Computing (MECO) (pp. 1-4). IEEE.».

[11] «Jung, I. S., Mulman, B. M., Thapa, D., Koo, L. J., Bae, J. H., Hong, S. H., ... & Wang, G. N. (2008, October). PLC control logic error monitoring and prediction using Neural Network. In 2008 Fourth International Conference on Natural Computation (Vol. 2,».

[12] «Canedo, A., Ludwig, H., & Al Faruque, M. A. (2014). High communication throughput and low scan cycle time with multi/many-core programmable logic controllers. IEEE Embedded Systems Letters, 6(2), 21-24.».

[13] «Li, J., & Gómez-Espinosa, A. (2018, November). Improving PID control based on neural network. In 2018 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE) (pp. 186-191). IEEE.».

[14] «Dias, A. L., Sestito, G. S., & Brandao, D. (2017). Performance analysis of profibus dp and profinet in a motion control application. Journal of Control, Automation and Electrical Systems, 28(1), 86-93.».

[15] «Fontanelli, D., Macii, D., Rinaldi, S., Ferrari, P., & Flammini, A. (2013, May). Performance analysis of a clock state estimator for PROFINET IO IRT synchronization. In 2013 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)».

[16] «Wu, X., Xie, L., & Lim, F. (2014, October). Network delay analysis of EtherCAT and PROFINET IRT protocols. In IECON 2014-40th Annual Conference of the IEEE Industrial Electronics Society (pp. 2597-2603). IEEE.».

[17] «Liu, L., Shan, L., Yan, J., Liu, C., & Dai, Y. (2018, June). An improved BFO algorithm for optimising the PID parameters of servo system. In 2018 Chinese Control And Decision Conference (CCDC) (pp. 3831-3836). IEEE.».

[18] «Alan, M. (2020) Biosignal classification and disease prediction with deep learning. Master Thesis, Marmara Universities institute for graduate studies in pure and applied sciences, Istanbul, Turkey.».

[19] Haykin, S. S. (2009). Neural networks and learning machines/Simon Haykin..

[20] «OKKAN, U., SERBEŞ, Z. A., & GEDİK, N. (2018). MATLAB ile Levenberg-Marquardt algoritması tabanlı YSA uygulaması: Aylık yağış-akış modellemesi. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 9(1), 351-362.».

[21] «Akgun, G. (2015) Data driven predictive control of exoskeleton for hand rehabilitation. Master Thesis, Marmara Universities institute for graduate studies in pure and applied sciences, Istanbul, Turkey.».

BIOGRAPHIES

HAYRI MUTLU is born in 1983, in Haskova, Bulgaria. He received his BSc. In Mechatronics Education from Marmara University in 2007. Currently, he is continuing his MSc. study at Marmara University. He instructs training about PLC, HMI, and Industrial Networking at Siemens Turkey.

MUSTAFA CANER AKUNER is born in 1968, in Samsun, Turkey. He received his BSc. in Electrical Education from Marmara University in 1986. His M.S. and Ph.D. degrees from Marmara University 1993 and 1999, respectively. He is currently head of Mechatronics Engineering department in Marmara University.

GAZI AKGUN is born in 1982, in Denizli, Turkey. He received his BSc. in Electrical Education from Dicle University in 2004. His M.S. and Ph.D. degrees from Marmara University 2015 and 2019, respectively. He is currently assistant professor at Mechatronics Engineering department in Marmara University.