Closed-loop Speed Control for a Three-Phase Alternating-Current Motor using a Modbus Network

Rodríguez, Martín[1]; **Baes, Mario**[1]; **Jara, Ricardo**[1]; **López, Yadira**[2]; **Orozco, Santos**[3]

[1] *Universidad Tecnológica del Norte de Aguascalientes, Departamento de Ingeniería Mecatrónica, Rincón de Romos, México*
[2] *Universidad de Guadalajara, Departamento de Ciencias Exactas y Tecnología, Lagos de Moreno, México*
[3] *Universidad Autónoma de la Ciudad de México, Colegio de Ciencia y Tecnología, Ciudad de México, México*

Abstract: This paper presents a closed-loop controller for speed regulation of a three-phase induction motor by means of a Modbus network, which is compound only by industrial-purpose hardware, namely two programmable logic controllers, a human-machine interface and a variable-frequency drive. This represents a step beyond the state of the art because the results in similar works reported in the literature were obtained using personal computers, academic-purpose hardware or a combination of both academic-industrial technologies. The speed regulation was achieved through a proportional-integral controller executed in a programmable logic controller, whose feedback was supplied by another similar device completely dedicated to read and decode an industrial optical encoder; furthermore, the control signals were commanded to the alternating current motor using a variable-frequency drive. The proposed control network was tested through diverse regulation experiments, under different conditions, in order to assess its effectiveness and robustness. The obtained results were successful in terms of accuracy of the speed set-point tracking in comparison to the most similar works of the state of the art, namely zero steady-state error was achieved, 12 seconds earlier for 75 % larger references and 22 % less RMSE than other works, which are the best results reported so far. Such successful achievements encourage the later implementation of more advanced control techniques on three-phase-motor based industrial machinery using only industrial-purpose hardware.

Keywords: Three-phase motor control, PLC based control, Modbus networks, AC motor speed regulation

Control de Velocidad en Malla Cerrada para un Motor de Corriente Alterna Trifásico utilizando una Red Modbus

Resumen: Este artículo presenta un controlador en malla cerrada para regular la velocidad de un motor de inducción trifásico por medio de una red Modbus que únicamente utiliza hardware de propósito industrial, propiamente dos controladores lógicos programables, una interfaz humano-máquina y un variador de frecuencia. Esto representa una ventaja respecto al estado del arte, ya que los resultados de trabajos similares reportados en la literatura fueron obtenidos utilizando computadoras personales o hardware de propósito académico además del hardware industrial. La regulación de velocidad fue lograda a través de un controlador proporcional-integral ejecutado en un controlador lógico programable, cuya realimentación es suministrada por otro dispositivo similar que está completamente dedicado a la lectura de un codificador óptico industrial; asimismo, las señales de control se enviaron al motor de corriente alterna usando un variador de frecuencia. La red de control propuesta fue probada por medio de diversos experimentos de regulación, bajo diferentes condiciones, para evaluar su efectividad y robustez. Los resultados obtenidos fueron satisfactorios en términos de la precisión del seguimiento de la consigna de velocidad en comparación a los trabajos más parecidos del estado del arte, propiamente se eliminó el error en el estado estacionario 12 segundos antes para referencias 75 % más grandes y un RMSE 22 % menor que otros trabajos, los cuales son los mejores resultados reportados hasta el momento. Dichos logros motivan a la futura implementación de técnicas de control más avanzadas en máquinas accionadas por motores trifásicos utilizando únicamente hardware de propósito industrial.

Palabras claves: Control de motores trifásicos, Control basado en PLC, Redes Modbus, Regulación de velocidad de motores de CA

1. Introduction

Several machines that compound diverse manufacturing processes are actuated by three-phase alternating current (AC) induction motors because of its simplicity, robustness and cost, therefore, controlling its operation is a highly relevant task in industry (Dukare and Jagtap, 2017). Another relevant technology for industrial processes is the programmable logic controller (PLC), whose reliability, operation simplicity, electrical and mechanical robustness and high speed turn it into the preferred device in industry for the execution of automation and control algorithms (Dukare and Jagtap, 2017). Additionally, the possibility of developing PLC net-

*santos.orozco@uacm.edu.mx
Recibido: 23/11/2023
Aceptado: 11/03/2024
Publicado en línea: 31/08/2024
10.33333/rp.vol54n1.03

works that might include other industrial-purpose hardware, such as human-machine interfaces (HMI), increases the aforesaid preference of PLC and stand out their versatility. For such a reason, several manufacturing companies in Mexico are reluctant to the use of computers, embedded systems or laboratory-purpose hardware and software to automate or control their machinery, even though these devices might execute cutting-edge automation and control algorithms that could be beneficial to optimize or improve industrial processes.

Considering the aforementioned problem, an appropriate alternative for the industry could be the implementation of automation and control algorithms using just industrial-purpose hardware. In this context, and recalling the idea of the wide usage of three-phase AC motors, several researchers have reported PLC-based control systems for such electrical machines (Verma and Potdar, 2017) in the literature. One of the most remarkable results was presented by Saad and Arrofiq (2012), in which the authors programmed a fuzzy controller for an AC motor within the PLC, but they used a laboratory-type sensor, namely a DC motor adapted to work as a generator. Similarly, Howimanporn et al. (2016) reported an optimized PID controller for a conveyor belt actuated by an AC motor, nonetheless, the authors also used laboratory-purpose hardware to send the control signals to the motor. Some examples of the usage of laboratory-purpose software can be found in the works of Al-Manfi (2019) and Awdaa et al. (2020); in both cases the closed-loop control for the motor is implemented by means of high-cost software executed on a personal computer (PC) in addition to the PLC.

On the other hand, some works that only use industrial hardware have been reported as well, for instance, the speed control of a three-phase motor for a centrifugal machine of a sugar manufacturing process was reported by Pramudijanto et al. (2015). The authors implemented a predictive PID controller on a PLC, however, they do not specify the technology used for feedback, the execution of the control algorithm is not explained and their results are not clearly presented in plots, but only a 10.79 root median squared error (RMSE) is presented. In addition, Mahesh and Ramachandra (2017) presented a PLC-based P controller using just industrial-purpose hardware, obtaining successful results with an average of 2 % steady-state error. Nevertheless, the authors do not show plots of the behavior of the speed and they do not present theoretical foundations. Another PLC based closed-loop controller without laboratory-purpose hardware is reported in the work of Demir et al. (2019), achieving a zero steady-state error in 40 seconds for their largest step reference from 0 to 1000 [RPM]. Despite these successful results, it is worth mentioning that the authors did not program the control algorithm but they only implemented and configured the proportional-integral-derivative (PID) controller functional block of the PLC programming environment. This represents a limitation if it is pretended to implement more advanced control techniques.

Besides, some researchers have tried to exploit the versatility of the PLC networks to control the speed of AC motors. For example, the use of open-platform communications (OPC) servers using Ethernet and other field buses to connect the PLC with variable-frequency drives (VFD) were presented by Velagic et al. (2011) and Al-Manfi (2019), in order to implement PID controllers. However, the use of a non-industrial-purpose PC within industrial environments is still a drawback for such approaches since the robustness of the system might be compromised. Recently, a very interesting fully industrial setup for induction motors control was reported by Vadi et al. (2022), these authors use a PLC Profibus network to communicate the PLC with the VDF, but the closed-loop control action is still missing.

Considering the information found in the literature, the problem of implementing closed-loop control systems using just industrial-purpose hardware is not fully overcome. In this context, the contribution of this paper is twofold: in one hand, up to the best authors' knowledge, this is the first time that a PI controller is analyzed, clearly explained and fully programmed within a PLC using an industrial encoder as feedback. This represents a step towards the implementation of more complex control systems on industrial devices. On the other hand, the control system uses a Modbus network in order to communicate the feedback-analyzing PLC with the control PLC, the VFD and also an HMI that is useful to command the target speed and also to monitor the measured velocity. In order to assess the correct operation and the robustness of the full setup, the following experiments were put into effect: firstly, the controller was commanded with large target velocities to prove its stability in the whole error space; later on, the importance of the integral action is evaluated and highlighted through the disturbance rejection capability assessment. The successful results demonstrated that the proposed control architecture is suitable to effectively control three-phase AC motors by means of only industrial hardware, eliminating the necessity of high-cost additional software or laboratory-purpose hardware that is more likely to fail in industrial environments. Furthermore, the developed control algorithm yielded better results than those reported in the literature in terms of time of response and quantitative performance indices. Finally, the implementation of a Modbus network stands out the versatility of the industrial hardware and opens the possibility of involving more devices and adding functionalities depending on the application requirements.

The organization of this paper is as follows: firstly, the used methods are described in Section 2, namely the industrial hardware network is described in addition to the mathematical model of the plant as well as the control system design and its implementation. Then, the Section 3 addresses the experiments carried out and the obtained results are discussed and contrasted against the state of the art. Finally, the Section 4 concludes the paper.

2. Methodology

Below, the full system setup used for this research is described, as well as the development and implementation of the proposed control technique using the industrial hardware network.

2.1 System Description

The implemented experimental system is displayed in Figure 1. It is composed of two control and power panels, respectively, as well as an independent station dedicated for the speed data acquisition from the three-phase AC motor. The first panel is integrated by a Micrologix 1500 PLC (hereinafter PLC1) and a HMI Panel View Plus 600 touch screen. Furthermore, the 1761-NET-ENI and 1761-NET-AIC modules have been adapted to the PLC1 to communicate it with other devices through the Ethernet and RS845 com-

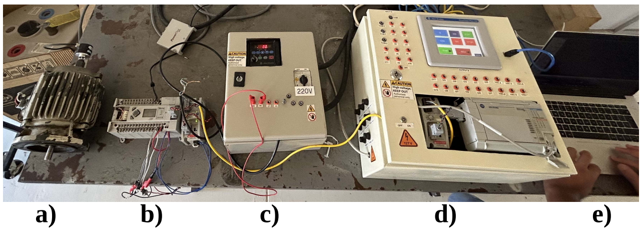

Figure 1. Experimental setup. a) AC motor. b) Speed-acquisition PLC. c) VDF. d) Panel with HMI and PLC. e) PC for PLC programming.

munication standards, respectively. The second panel contains a PowerFlex 40 VFD that feeds the AC motor with three-phase 220 [VAC]. Such a motor demands a 0.5 [HP] power, and its nominal speed is 1,800 [RPM], moreover, it is equipped with a 600-pulses-per-revolution quadrature encoder attached to the shaft.

Firstly, it was tried to implement the whole control system in the PLC1, however, the encoder counting and the control system algorithms execution interfered with each other affecting the correct control operation. Thus, it was decided to implement a Micrologix 1400 PLC (hereinafter PLC2), to which a 1761-NET-AIC module was also incorporated. This PLC was chosen since six of its digital inputs are capable to operate as high-speed counters (HSC), namely up to 100 [kHz], which turns this PLC into the most competitive model within the Micrologix PLC family in this regard, however, the same problem of interfering algorithms arose while executing the whole control system in the PLC2. In order to solve such a problem, it was decided to implement a dedicated station to capture and process the encoder signal for its later transmission and usage.

The reference data for the implemented speed control system, in [RPM], is entered by the user through a graphical interface embedded into the HMI screen. In the same interface, the speed data of the motor shaft is displayed also in [RPM] after having been conditioned by the Micrologix 1400 PLC (hereinafter PLC2). Likewise, the magnitude of the error and the control signal sent to the VFD, both in [Hz], are displayed in two other labels. Some buttons to perform the functions of start, stop and reverse rotation are included as well to manipulate the AC motor.

The communication between the devices in charge of both the acquisition of the feedback signal, as well as sending the control signal, is carried out through the Modbus protocol, supported by the 1761-NET-AIC module and by the VFD itself, through an RS485 input provided. Meanwhile, for the interaction between the HMI screen and the Micrologix 1500 PLC, the Ethernet IP protocol is used, associated with the 1761-NET-ENI module.

Based on its interaction with all the elements of the implemented control system, the corresponding control algorithm is executed in the PLC1. In this case, a discrete Proportional Integral (PI) controller. It is worth mentioning that the code developed, built on the IEC-61131-3 programming standard (ladder language), does not use any specialized function for the application of the mentioned PI controller, but the algorithm was completely developed from scratch using a numerical integration.

2.2 Plant Modelling

In order to obtain a transfer function that relates the three-phase motor speed with the VDF frequency, the well-known Ziegler-Nichols reaction curve method was implemented.

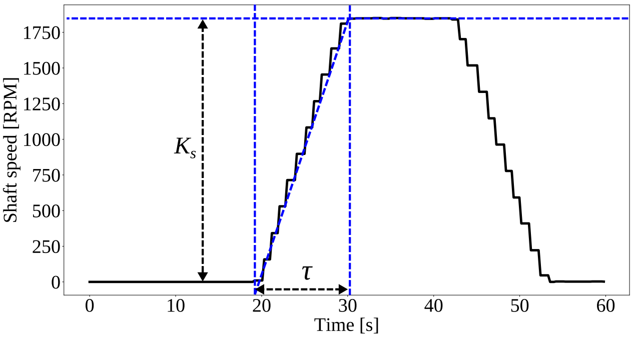

Figure 2. Reaction curve and Ziegler-Nichols parameters.

According to the parameters obtained from the Ziegler-Nichols reaction curve method, as shown in Figure 2, the transfer function representing the relation between the shaft speed of the three-phase AC motor with respect to the frequency input is given by

$$G(s) = \frac{K_s}{\tau s + 1} e^{-Ls}, \tag{1}$$

in this case, $K_s = 1{,}800$ and $\tau = 11$ and $L = 0$. It is worth mentioning that this modelling approach, besides its wide use in industry, is well supported in the literature for its use in electrical motors (Ustun and Demirtas, 2009; Bhatti et al., 2016; Avdeev and Vyngra, 2020; Nicola and Nicola, 2021).

In order to adequately synchronize the Modbus network, a sampling period of 250 [ms] was implemented. Thus, a discretization of the system (1) was put into effect based on the aforementioned period and the zero-order hold method and resulting in the discrete-time transfer function

$$G(z) = \frac{a}{z - b}, \tag{2}$$

with $a = 40.45$ and $b = 0.9775$ according to the discretization process. Now this transfer function is helpful to design a discrete-time closed-loop control system.

2.3 Control System Design

The control target in this work is to lead the shaft speed of the AC motor $\omega(k)$ to a desired value $\omega^d(k)$, with $k \geq 0$ denoting the sample number. In this context, a feedback controller is required to reduce to zero the error between the desired speed and the measured speed, namely, $e(k) = \omega^d(k) - \omega(k)$. Since (2) is a first-order system without delay, a Proportional-Integral (PI) controller is suitable to regulate the shaft speed (Landau and Zito, 2010), therefore, considering the error in the $\mathscr{Z}$ domain, $E(z) = \Omega^d(z) - \Omega(z)$, the transfer function of the mentioned controller is (Moudgalya, 2007)

$$\frac{U(z)}{E(z)} = K\left[1 + \frac{T_i}{T_s}\frac{z-1}{z}\right], \tag{3}$$

where $K > 0$ is the proportional gain to be defined, $T_i > 0$ is the integral time and T_s is the sampling period. Even though the control parameters can be determined using the Ziegler-Nichols approach, the system delay is zero and hence, such a methodology is not applicable. Besides, the gain tuning procedure was conducted by analyzing the pole of the closed-loop transfer function with only the proportional controller, which is given by

$$\frac{\Omega(z)}{\Omega^d(z)} = \frac{Ka}{z+(Ka-b)}. \tag{4}$$

Considering the stability condition, $-1 < z < 1$ (Ogata, 2015), the proportional gain can be computed with

$$\frac{b-1}{a} < K < \frac{1+b}{a}\ . \tag{5}$$

Once the stability is ensured with the proportional controller, the integral action was added yielding the following closed-loop transfer function (Landau and Zito, 2010)

$$\frac{\Omega(z)}{\Omega^d(z)} = \frac{Ka\left[\frac{T_s}{T_i}z+1-\frac{T_s}{T_i}\right]}{z^2+\left(K\frac{T_s}{T_i}a-b\right)z+Ka\left(1-\frac{T_s}{T_i}\right)}, \tag{6}$$

where T_i is the integration time. In this work $K = 0.01$ was selected, which satisfies the stability criterion for the P controller. In addition, the integration time was determined considering a period of 25 [ms] to allow a correct pulse count from the encoder, therefore, the ratio T_s/Ti must be 0.025 and consequently, the integration time was computed as 20 [s]. Finally, the stability of the closed-loop system with PI controller (6) is guaranteed since both poles $z_{1,2} = 0.4728 \pm 0.7526j$ are inside the unit circle of the complex plane.

2.4 Controller Implementation

The controller (3) in terms of equations of differences can be expressed as (Bolton, 2021)

$$u(k) = Ke(k) + K\frac{T_i}{T_s}e(k) + u(k-1); \tag{7}$$

this last expression was programmed in the PLC1, using the ladder language within the RsLogix programming environment. In order to compute the control signal $u(k)$, the error was firstly computed by means of both the target speed $\omega^d(k)$ and the shaft speed $\omega(k)$. The first one can be directly commanded through the HMI touch panel with a range from 0 to 1,800 [RPM] and later on is sent to the PLC1 using the Modbus network. The latter is obtained from the encoder attached to the shaft of the AC motor; the generated pulses are counted at a 20 [kHz] frequency with the PLC2 and later transferred to the PLC1 also through the Modbus network; the use of the PLC2 is justified since only this device was capable to accurately read the encoder pulses while spinning at the nominal shaft speed. Next, the PLC1 computes the control signal $u(k)$, as it was mentioned before, and transforms it into frequency units with a range from 0 to 600, which is also sent via the Modbus network to the VFD. Finally, such a signal is translated to the range from 0.0 to 60.0 [Hz] and then applied to the AC motor. The block diagram of the full control system is depicted in Figure 3, where all the signals and the devices processing them are illustrated. In addition, the flow diagram of the control system algorithm executed by the PLC1 is shown in Figure 4, where the interaction between the PLC and other devices can be observed meanwhile it executes the proposed control system.

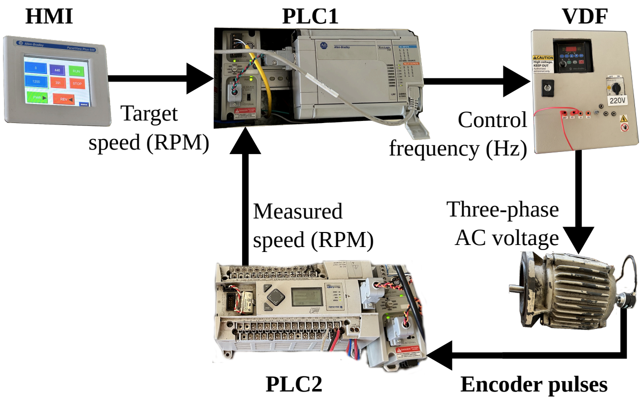

Figure 3. Block diagram of the control system network.

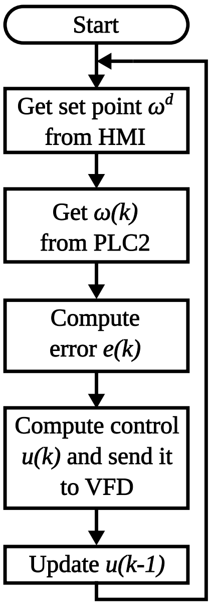
Start
Get set point ω^d from HMI
Get $\omega(k)$ from PLC2
Compute error $e(k)$
Compute control $u(k)$ and send it to VFD
Update $u(k-1)$

Figure 4. Block diagram of the control system algorithm.

3. Experiments and Results Discussion

In order to evaluate the correct operation and robustness of the proposed Modbus control network, some experiments were performed. The details of their conduction and the results are detailed below. A video of the experiments can be found at `https://youtu.be/zDgAH1HhDGA`.

3.1 PI Controller using Large Step References

Once the PI controller algorithm was programmed within the PLC, the first test to evaluate the performance of the implemented system consisted in widely changing the set-point speed. In this way, starting from a total inactivity state of the motor (0 [RPM]), the maximum admissible speed value was commanded, namely 1,800 [RPM]. After the motor speed converges to the commanded value, the set-point was shifted to 50 [RPM]. This process was repeated by commanding diverse large steps as the assigned reference values, which are displayed in Table 1. Likewise, it is possible to appreciate the graphic behavior of such changes in Figure 5, and how the response is established in the assigned value for each case analyzed, nullifying in all cases the steady-state error and yielding an overshoot of less than 2 %.

Table 1. Large Step References in [RPM]

Stages	Desired [RPM]
1	1,800
2	50
3	1,100
4	40
5	1,200
6	20

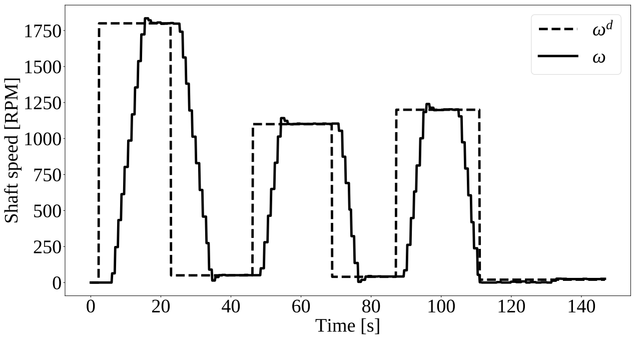

Figure 5. Time evolution of the controlled shaft speed with large set-point changes.

3.2 Disturbances Rejection Assessment

After validating the proper behavior of the PI controller while facing sudden changes in the speed reference value, some other experiments were carried out to visualize and compare the responses obtained using both the PI controller and a proportional (P) controller, after applying a disturbance torque to the AC motor shaft. The main target of this experiment is to evaluate whether the proposed controller would recover the set-point speed of the motor shaft, despite the presence of disturbances, such as the application of load on the shaft for instance.

Fort the case of the P controller, as shown in Figure 6, it can be seen how the speed drops from 120 [RPM] to almost zero after applying a disturbance. The controller is unable to set an output that can effectively cope with the effect of the load. On the contrary, it seems that the control signal is similar with or without the presence of disturbances as can be seen in Figure 7. Therefore, this control action alone is insufficient to keep the shaft speed at the desired value in presence of disturbances.

Regarding the action of the PI controller, in Figure 8 it can be seen how eventually, after the shaft speed drops due to the action of the imposed load, the controller computes and supplies a larger amount of energy leading to maintain the magnitude of the speed at the entered target value. Likewise, in Figure 9 it is observed how after having established a considerable energy to mitigate the load effect, the speed is again regulated to its given set point.

In order to provide a quantitative result to show which performance is better among both controllers (Balestrino et al., 2006), the Integral of the Absolute Error (IAE) and the Integral of the Absolute value multiplied by Time of the Error (ITAE) were computed as follows

$$IAE = \frac{1}{2}\sum_{k=0}^{N}|e(k)+e(k+1)|, \tag{8}$$

$$ITAE = \frac{1}{2}\sum_{k=0}^{N}k|e(k)+e(k+1)|, \tag{9}$$

where N is the number of samples of the experiment. The resulting values are listed in Table 2, where it can be observed that the PI controller significantly reduces both indices.

Table 2. Performance indices for the closed-loop controllers

Index	P	PI
IAE	10.9×10^3	7.5×10^3
ITAE	18.1×10^3	8.4×10^3

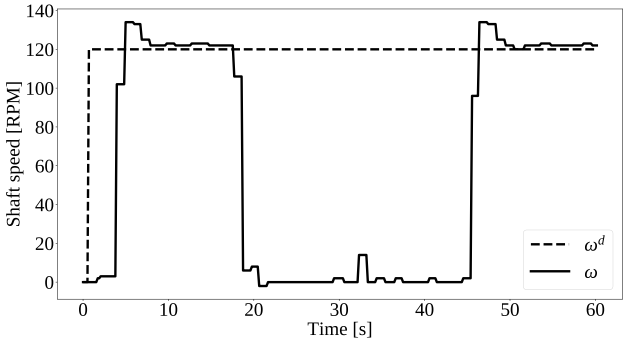

Figure 6. Time evolution of the controlled shaft speed with P control and application of disturbances

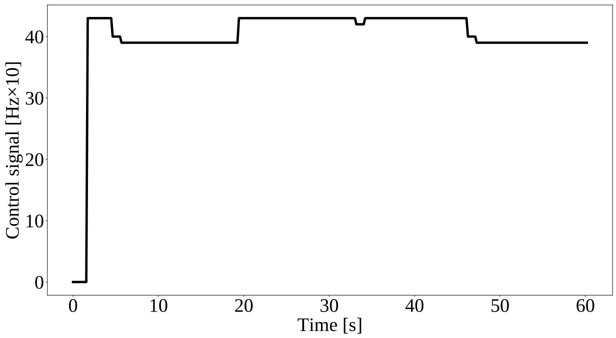

Figure 7. Time evolution of the P control signal with application of disturbances

3.3 Results Discussion

Figure 5 that the proposed PI controller (7) executed by the Modbus network devices is effective for speed-regulation despite how large the set points from each other are. No steady-state error and a 2 % overshoot can be also appreciated which is acceptable for this industrial machine. The importance of including an integral action to improve the robustness of a control system is stood out

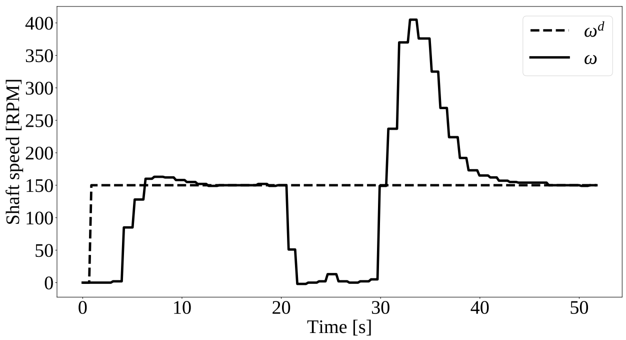

Figure 8. Time evolution of the controlled shaft speed with PI control and application of disturbances

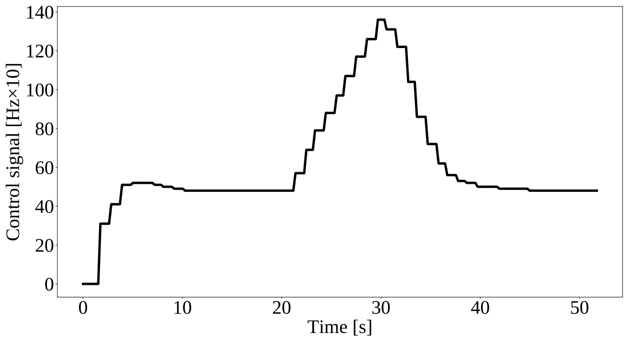

Figure 9. Time evolution of the PI control signal with application of disturbances

from the disturbance rejection assessment. Figure 8 shows how the speed is recovered after a disturbance using the PI controller (7) in contrast to Figure 6, where it can be observed that the shaft remained steady during the disturbance application when using the control P (4). Likewise, Figure 9 shows that the PI control signal increased while the disturbance was being applied, in contrast to the P control signal in Figure 7, where no significant change can be appreciated, resulting in insufficient energy to mitigate the disturbance action. The computed indices (8) and (9) shown in Table 2 are consistent with the graphical results, showing that the PI controller reduced the error caused by the applied disturbance.

Moreover, the obtained results in this work were contrasted against those obtained by Pramudijanto et al. (2015) and Demir et al. (2019), since these are the only similar works presenting their results. For the case of the predictive PID controller reported by Pramudijanto et al. (2015), the RMSE was computed using

$$RMSE = \sqrt{\frac{\sum_{k=0}^{N} e(k)^2}{N}}, \tag{10}$$

resulting on a value of 10.79 for similar-to-step references. In this work, the RMSE was computing with (10) as well resulting on a value of 8.41 for larger true-step references, which is 22 % less than the RMSE reported by Pramudijanto et al. (2015). Now considering Demir et al. (2019), it is worth remembering that the authors used the PID functional block available in their programming environment, meanwhile in this work, the full control algorithm was developed from scratch. Figure 10 shows the speed regulation plots obtained by Demir et al. (2019), where it can be observed that the AC motor speed reached the steady state at the desired value after 40 seconds from 0 to 1000 [RPM]. In contrast in this work, the steady state at the desired value was achieved in 18 seconds from 0 to 1750 [RPM]. Hence, the results presented in this work overcome those obtained by Demir et al. (2019) in terms of faster response for larger errors, namely, the proposed controller reduces 12 seconds the set point steady state accomplishment for a 75 % larger error.

4. Conclusions

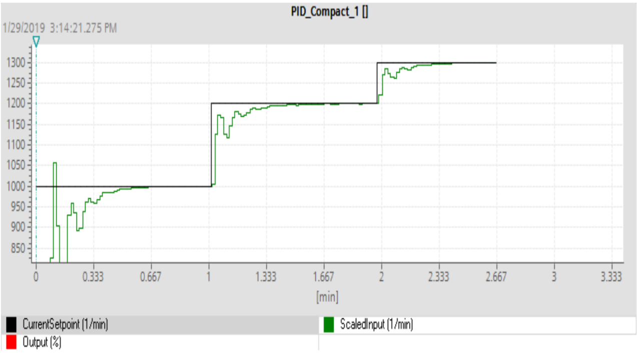

Figure 10. Results obtained by Demir et al. (2019)

In this work, the implementation of a closed-loop controller to regulate the shaft speed of a three-phase AC motor was presented. Such an implementation was performed using only industrial-purpose hardware, which is an advantage over the state-of-the-art approaches so far, in addition to the completely novel use of a Modbus network to perform a digital control system for an /11/202. The control system technique used was a PI controller, whose performance and robustness were evaluated by way of the following experiments: 1) commanding large set-points to the PI controller to verify its effectiveness throughout the whole error signal span; 2) comparing whether or not using the integral action in order to highlight its necessity for disturbance rejection.

From the first experiment, the results are considered successful since the controller is capable to lead the shaft speed to a desired value despite how far from the current condition is; besides, the steady state error is practically nullified.

For the case of the disturbance rejection assessment, it can be seen that the integral action works correctly, since the control signal increased to mitigate the effects of the disturbance and it reduced the IAE and ITAE performance indices as well. This leads to the conclusion that the integral action would be helpful to maintain the commanded speed despite the presence of a constant load on the shaft, for instance, using the AC motor to actuate a conveyor belt or to move another type of machine.

Finally, the obtained results from speed regulation can be considered superior with respect to the state of the art since the RMSE value at steady-state error was reduced 22 % with respect to the results presented by Pramudijanto et al. (2015) and the steady state at the desired set point was accomplished 12 seconds earlier than the controller presented by Demir et al. (2019) for a 75 % larger error.

From the successful results of these experiments, a correct operation and performance of the Modbus network can be concluded as well. All the set of these promising results encourages the use of only industrial-purpose networked hardware to implement more complex control systems rather than traditional high-cost academic-purpose hardware and software, which are less interesting for manufacturing industry in Mexico.

ACKNOWLEDGEMENTS

The authors would like to thank the Universidad Autónoma de la Ciudad de México for supporting the project UACM CCYT-2023-IMP-02. This contribution is part of the results of such a project. In addition, the National Researcher Candidate with CVU No. 319148 thanks the National Humanities, Science and Technology Council (CONAHCYT) for the support given.

REFERENCES

Al-Manfi, A. A. (2019). Implementation of remote self-tuning fuzzy PID controller for induction motor through ethernet. *J. of Electrical Systems*, 15(1): 1059–1065. http://journal.esrgroups.org/jes/papers/15_1_2.pdf.

Avdeev, B. A. and Vyngra, A. V. (2020). Simulation of variable frequency controlled AC induction motor operating on nonlinear load. In *Proc. of IEEE Conf. of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*. pp: 2346–2349. https://doi.org/10.1109/EIConRus49466.2020.9038963.

Awdaa, M., Obed, A., and Saleh, A. (2020). Three-phase induction motor SVPWM-FOC control based on PLC Matlab translation approach. In *IOP Conf. Series: Materials Science and Engineering*. pp: 1–8. https://doi.org/10.1088/1757-899X/745/1/012028.

Balestrino, A., Landi, A., Medaglia, M., and Satler, M. (2006). Performance indices and tuning in process control. In *Proc. of IEEE 14th Mediterranean Conf. on Control and Automation (MED)*. pp: 1–6. https://doi.org/10.1109/MED.2006.328852.

Bhatti, S. A., Malik, S. A., and Daraz, A. (2016). Comparison of PI and IP controller by using Ziegler-Nichols tuning method for speed control of DC motor. In *Proc. of IEEE Int. Conf. on Intelligent Systems Engineering (ICISE)*. pp: 330–334. https://doi.org/10.1109/INTELSE.2016.7475144.

Bolton, W. (2021). *Instrumentation and Control systems*. Newnes.

Demir, M. C., Cengiz, T., and Eminoğlu, İ. (2019). Experimental setup for AC motor speed control by using PLC. *J. of Scientific Perspectives*, 3(3): 177–188. https://doi.org/10.26900/jsp.3.018.

Dukare, P. D. and Jagtap, P. (2017). Induction motor control by stator voltage control method using PLC: A review. *Int. Research J. of Engineering and Technology*, 4(4): 92–95. https://www.irjet.net/archives/V4/i4/IRJET–V4I422.pdf.

Howimanporn, S., Thanok, S., Chookaew, S., and Sootkaneung, W. (2016). Speed control technique for conveyor using PSO based PID with programmable logic controller. In *Proc. of IEEE/SICE Int. Symp. on System Integration (SII)*. pp: 670–675. https://doi.org/10.1109/SII.2016.7844076.

Landau, I. D. and Zito, G. (2010). *Digital control systems: design, identification and implementation*. Springer.

Mahesh, K. and Ramachandra, B. (2017). Speed control of three phase induction motor using PLC under open and closed loop condition. *Int. J. Engineering Research Applications*, 7(1): 34–39. https://doi.org/10.9790/9622–0701043439.

Moudgalya, K. M. (2007). *Digital control*. John Wiley & Sons.

Nicola, M. and Nicola, C.-I. (2021). Tuning of PI speed controller for PMSM control system using computational intelligence. In *Proc. of IEEE 21st Int. Sym. on Power Electronics (EE)*. pp: 1–6. https://doi.org/10.1109/Ee53374.2021.9628297 .

Ogata, K. (2015). *Discrete-time control systems*. Pearson.

Pramudijanto, J., Ashfahani, A., Fatoni, A., and Nugroho, S. A. (2015). PLC-based PID-predictive controller design for 3-phase induction motor on centrifugal machine for sugar manufacturing process. In *Proc. of IEEE Int. Con. on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA)*. pp: 49–52. https://doi.org/10.1109/ICAMIMIA.2015.7508001.

Saad, N. and Arrofiq, M. (2012). A PLC-based modified-fuzzy controller for PWM-driven induction motor drive with constant V/Hz ratio control. *Robotics and Computer-Integrated Manufacturing*, 28(2): 95–112. https://doi.org/10.1016/j.rcim.2011.07.001.

Ustun, S. V. and Demirtas, M. (2009). Modeling and control of V/f controlled induction motor using genetic-ANFIS algorithm. *Energy conversion and management*, 50(3): 786–791. https://doi.org/10.1016/j.enconman.2008.09.032.

Vadi, S., Bayindir, R., Toplar, Y., and Colak, I. (2022). Induction motor control system with a programmable logic controller (PLC) and profibus communication for industrial plants—An experimental setup. *ISA Transactions*, 122: 459–471. https://doi.org/10.1016/j.isatra.2021.04.019.

Velagic, J., Kaknjo, A., Osmic, N., and Dzananovic, T. (2011). Networked based control and supervision of induction motor using OPC server and PLC. In *Proc. of IEEE Int. Symp. on Electronics in Marine (ELMAR)*. pp: 251–255. https://ieeexplore.ieee.org/abstract/document/6044283.

Verma, P. and Potdar, R. M. (2017). A review on AC drive controller using PLC. *Int. J. of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering*, 5(9):pp: 53–55. https://doi.org/10.17148/IJIREEICE.2017.5913.

BIOGRAPHIES

Martín Rodríguez. In 2010 he obtained the Bachelor degree in Mechatronics Engineering from the Polytechnic University of Aguascalientes. Since 2014, he is associate professor at the Mechatronics Engineering Department of the Technological University of Northern Aguascalientes. His research interests are robotics, industrial automation, computer vision and lean manufacturing systems. He has participated as coach in several contests and he has been invited to deliver conferences and tutorials related to robotics and automation.

Mario Baes. In 2023, he obtained the Technical Degree in Mechatronics with speciality in Automation from the Technological University of Northern Aguascalientes in Mexico, working with industrial communication protocols for diverse applications. He is currently enrolled as undergraduate student in the Mechatronics Engineering program at the same institution, where he still collaborates in several research projects related to industrial automation.

Ricardo Jara. In 2011 he obtained the Bachelor degree in Mechatronics Engineering from the Polytechnic University of Aguascalientes. He got the Master of Science degree in Electrical Engineering from the Research and Advanced Studies Center of the National Polytechnic Institute in Guadalajara in 2016. Since 2017, he is associate professor at the Technological University of Northern Aguascalientes, collaborating in the Mechatronics Engineering department. His research aims to develop projects regarding automatic control.

Yadira López. She obtained the Bachelor degree in Electronics Engineering in 2009 from the Aguascalientes Institute of Technology. In 2012, she was awarded with the Master of Science degree in Electrical Engineering from the same institution. In 2022 she got the doctoral degree from the University of Guadalajara, and since 2015 she is full professor at the Technological University of Northern Aguascalientes, where she also leads the Academic Group of Mechatronic Systems. Her research interests are the interferometry and the optical instrumentation.

Santos Orozco. In 2020, he obtained the Doctor of Science degree in Automatic Control from the Research and Advanced Studies Center of the National Polytechnic Institute in Mexico City, developing robust controllers for humanoid robots and drones, and collaborating with important robotics laboratories in Germany and France. From 2020 to 2022 he worked as a researcher at the PRISMA Lab of the University of Naples in Italy, developing control systems for aerial manipulators. Currently he is full professor at the Science and Technology College of the Autonomous University of Mexico City. His research interests are the automatic control theory and its applications.