

# PLC and SCADA Based Real-Time Monitoring and Control of Networked Processes

**Dilşad Engin**

Control and Automation Technology Department  
Ege University, Ege Higher Vocational School  
İzmir, Turkey  
dilsad.engin@ege.edu.tr

**Serkan Pınar**

Control and Automation Technology Department  
Ege University, Ege Higher Vocational School  
İzmir, Turkey

**Mustafa Engin**

Electronics Technology Department  
Ege University, Ege Higher Vocational School  
İzmir, Turkey  
mustafa.engin@ege.edu.tr

**Mücahid Candan**

Mechatronics Department  
Ege University, Ege Higher Vocational School  
İzmir, Turkey

**Abstract**—In this study, we present PID control and real-time monitoring of a networked system via supervisory control and data acquisition (SCADA) software and programmable logic controller (PLC). Monitoring and data acquisition of the controlled variables in real time need arose in today's networked systems. SCADA systems are the modern instrument for real-time control and monitoring of geographically dispersed facilities and factory automation where data acquisition and open and closed-loop control through a central control room are critical to the operation of the system. Whenever complex processes demand system-level intelligence, SCADA systems best suit for this purpose. In our study, real-time monitoring and control of level and temperature processes, and an AC motor via a motor drive with Modbus RTU protocol that was adapted to communicate with a SCADA server where the AC motor drives a conveyor belt for bottle filling plant is implemented. The remote real-time monitoring and control of the system was operated through three PLCs and a SCADA server. The results have shown that by means of remote control of the devices in the field and parameter screens, the operating values were changed according to the requirements of the system, enabling them to work more efficiently, possible changes in any parameter value, motor states and fault conditions were monitored by remote reading of the drive registers, enabling the necessary intervention in a timely manner. The ability to transfer data between PLCs on the network via industrial Ethernet provided optimum data flow and data synchronization in process control, while allowing to program more than one PLC at the same time. Since the software we used in the project supports online changes via networked PLCs, it saves time and service costs by eliminating problems that may occur in more than one PLC from a remote location.

**Keywords**—SCADA system; real time monitoring; networked systems; Modbus RTU and TCP communication; PLC; PID control

## I. INTRODUCTION

The challenging industry requirements, and the increasing expectations of the customers compel the production and manufacturing companies to faster, smarter, and more reliable technologies. Networked-based control systems for industrial tasks reduce error risks by continuously monitoring the system and are incorporating the process and machinery with the control software running on the on-site and remote controllers. Consequently, less downtime, flexibility in production and better quality of products can be achieved. Efficient energy consumption and resource management in production is another asset. Therefore, to achieve efficient products, various distributed network control architectures comprising industrial communication protocols are merged by the supervisory control systems [1]. Many large-scale industrial processes use Supervisory Control and Data Acquisition (SCADA) systems to deliver the increasing production volume, to monitor the equipment located at numerous sites that require management and supervision of the processes.

SCADA system is a server, industrial network and control instruments-based process control and scheduling automation system. SCADA system can supervise the operation of field devices, whereas it has functions to acquire data, control equipment, detect states of the events, adjust parameters, inquire alarm signals, and display historical data. Additionally, they provide automation capabilities to improve operational reliability, increase Quality of Service (QoS), and additional performance issues [2].

SCADA systems are broadly used for real-time control and monitoring of geographically dispersed facilities, where data acquisition and open-loop control through a central control room are critical to the operation of the system. Some of SCADA application areas are energy management in electrical power grids and electricity generating plants [3], water distribution and wastewater collection systems [4], oil and gas transportation [5], tunnel and railway automation.

We have utilized SCADA system for real-time control and monitoring of level and temperature processes. Also, a PLC-based control and real-time monitoring of an asynchronous AC motor via a motor

drive with Modbus RTU protocol was adapted to communicate with a SCADA server where the AC motor drives a conveyor belt for bottle filling plant.

In factory automation, distributed control systems are widespread where data are collected from numerous sensors and automatic control of many actuators are performed. Real-time monitoring, data acquisition, alarm monitoring of the controlled variables is vital in many processes. If data monitoring is sufficient for the processes, many real-time monitoring software equipped with hardware can be utilized. In case of manual control whenever necessary, and entering set points, proportional, integral, and derivative (PID) parameters, setting analog/digital alarms, tracing analog signals, etc. are required, then SCADA systems are more suitable for these purposes.

SCADA control center accomplishes alarm monitoring and control tasks for the field over long-distance communication networks, where data is also processed. The field controllers, mostly PLCs, perform the automatic or operator-driven supervisory commands based on data received from remote stations. PLCs perform local operations such as proportional valve or speed control or ON/OFF control of solenoids, collect data from sensors, and monitor local alarm conditions. The communication network in SCADA systems can be stated as:

- Communication between local controllers and field instruments, and operating equipment
- Short-distance communication between local controllers, and between controller and HMI
- Long-distance communication between local controllers and host computers, and between host and client computers

The control modes of actuators used in factory automation is ON/OFF or proportional control. Modern industry mostly requires PID control of processes in production and manufacturing systems. Consequently, many researchers work on improving the stability [6] and robustness [7], obtaining the optimal tuning parameters of the PID controller [8],[9] and developing self-tuning algorithms [10]. However, besides the advantages, most autotuning software cannot determine an accurate process model when the process data is noisy or oscillatory, and if the process actually performs in a nonlinear manner [11]. In this work, our aim was basically on real-time monitoring and control of plants we have chosen a manual tuning of the PID parameters for the processes providing less complexity.

This paper is organized as follows: in Section II, we described the system components as PLC, HMI, SCADA system, industrial communication protocols used, the ethernet gateway to connect the AC drive, a

serial device, to the SCADA server using Modbus TCP, and technical specification of the hardware system. In Section III, we presented the control of the networked system. The PID control for the temperature and level are implemented on two PLCs whereas one AC motor drive is controlled with another PLC for carrying bottles on a conveyor belt that would be filled. The flowcharts as well as part of the control programs are presented in this section for control and communication purposes. In Section IV, results are discussed.

## II. SYSTEM ARCHITECTURE

PID control of level and temperature system with two PLCs and a SCADA server is implemented in this study. The speed of a pump was controlled for regulating the input flow rate to the level tank. In our system, since the bottles on a conveyor belt controlled by an AC motor would be filled, motor speed and direction control were performed with a third PLC. While AC motor drive parameters (speed, direction, acceleration and deceleration) were sent using Modbus RTU protocol through PLC program, monitoring of data and parameter entry through SCADA screen was performed using Modbus TCP/IP protocol via the Ethernet Gateway.

One controller performed heating the fluid and keeping the level of the tank to ensure maximum heat transfer to the fluid whereas the heated fluid was used in a level tank. An ultrasonic level transmitter was selected for level measurement. To bring the level to the desired set point, a pump motor located on the input line adjusted the flow rate of the liquid to the tank with the 0-10V control signal from the controller. TM241CEC24R PLC with PID control feature was selected as the control device equipped with analog input and output modules that convert the analog data from the ultrasonic transmitter to digital and the data sent to the control valve into analog. The schematic drawing of the installed system is shown in Fig. 1.

### A. Programmable Logic Controllers

In factory automation, process control of dynamic variables is performed using digital PID controllers or PLCs. The identification of PID parameters are performed via auto-tuning and/or self-tuning algorithms embedded in PLCs [12]. Even there are no existing algorithms in the PLCs, it is performed by developing and embedding these algorithms [8], [13], [14].

PLCs are industry-standard, reliable control devices for automatic control of a wide variety of industrial systems in sequential control, process and motion control. Innovations in microprocessor technology and improvements in software have embedded more attributes and competencies to the PLC, and as a result, it has begun to outperform other control devices [15].

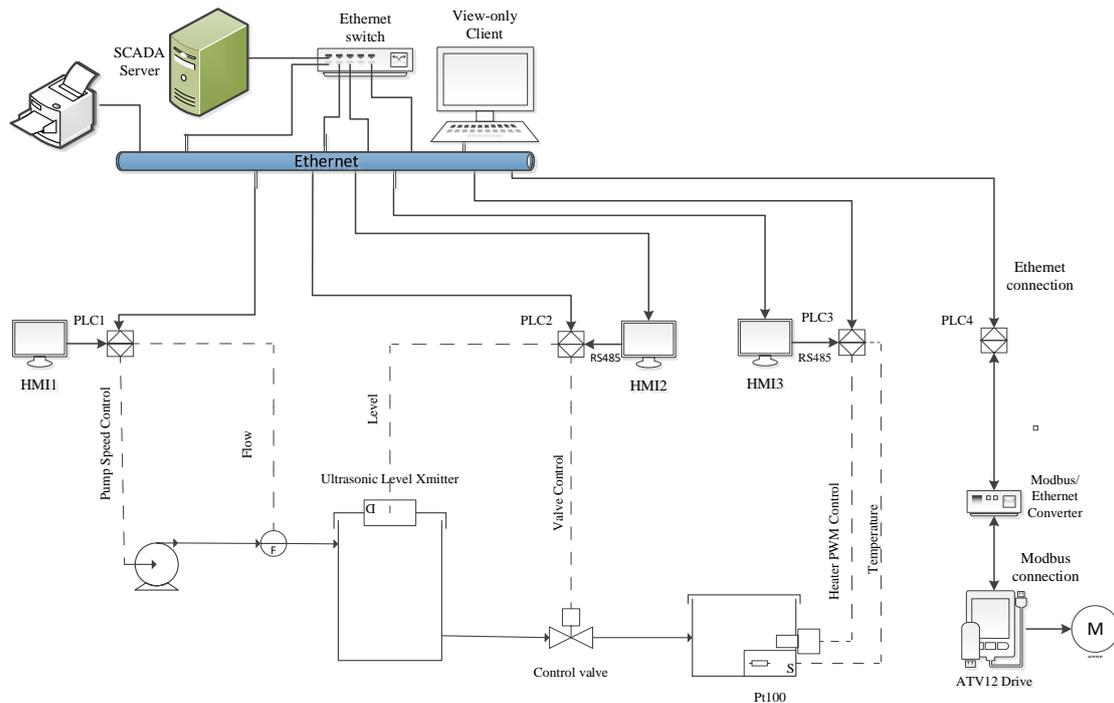


Fig. 1 PID control of networked temperature and level process controllers

Although the cost of microprocessors is lower, one of the reasons why PLCs are preferred is that they minimize the time spent on electronic design. Other advantages provided by PLCs are as follows [16][15][17]:

- control of complex systems,
- quick and easy software changes can be re-applied to control systems, thus flexible control
- more complicated controls with the aid of its calculation capabilities,
- troubleshooting instruments bring the advantages such as easier programming and reduced downtime,
- embedded I/O modules and communications ports (RS232/485, Ethernet, CAN, etc.)
- PLC's modularity facilitates extension using a wide range of digital and analog I/O modules as well as temperature, safety, and communication modules for various applications.
- resistant to working in harsh environments such as polluted air, humidity, vibration, temperature variations, electromagnetic interference confronted in the factory.

### B. Supervisory Control and Data Acquisition Systems

Today's Supervisory Control and Data Acquisition systems combine programmable logic controllers, human machine interface (HMI) workstations and network communication systems in one integrated system [18]. Many programming languages and software packages are required to be used, from the configuration of each component in this integrated system to programming, from the design of the graphic

displays to the configuration settings of the industrial communication system.

Because there are many aspects of SCADA application software, it is important to use a structured organized approach to the design and development of this software; therefore, it is very important to establish standards.

### C. Industrial Communication Protocols

In industrial automation systems, communication protocols allow access to various devices as HMI and PLC, PLC and SCADA server, and others. The industrial Ethernet protocol for data communication used in this work are Modbus TCP for communication between PLC and SCADA server containing industrial devices.

Communication between PLC and HMI uses serial line over Modbus RTU protocol. HMI also has USB port for serial communication and an ethernet port for Modbus TCP communication. AC motor drive used in this system can receive and transfer data using Modbus RTU protocol where it can communicate with the PLC. However, this protocol is not suitable for data exchange with the SCADA server. For this purpose, we used EGX150 Ethernet gateway to connect the Modbus serial device to Ethernet TCP/IP infrastructures (Fig. 2).

#### 1) Modbus TCP Communication and Control of AC Drive

Altivar 12 (ATV12) speed control device, also known as AC motor drive or inverter, is used to control the speed, direction, acceleration, and deceleration of the conveyor motor. With the PLC program, parameter settings were made using the internal HMI to be controlled directly with the Modbus RTU protocol, and

the control of the conveyor belt with the desired parameters was achieved through the program written with the commands used for Modbus communication with the PLC.

The EGX150 Ethernet gateway which is used to connect Modbus serial devices to Ethernet TCP/IP infrastructures or to control remote installations, and the SCADA server computer using the Modbus TCP/IP protocol, adjusting the variable speed drive parameters communicating with the Modbus RTU protocol and setting the asynchronous motor via this drive. A conveyor belt that it set in motion was controlled.

Ethernet gateways provide fast and reliable Ethernet connectivity in the most demanding applications. A gateway provides fast and efficient transmission of data to devices such as meters, monitors, motor drives that communicate serial data via RS232C or RS485, with full Ethernet connectivity. It provides data transfer over Ethernet connection of all devices using serial communication protocols to other control devices and SCADA server computers in the network using one of the Ethernet TCP / IP protocols.

Therefore, in addition to providing connection to devices using serial communication protocols via PLCs, variable speed motor drive parameters can be changed and monitored, for example by directly accessing the SCADA screen. The Modbus TCP/IP feature of the gateway were used within the scope of the study. Modbus TCP/IP protocol offers master/slave communication between devices and TCP/IP communicates over an Ethernet connection. Modbus TCP/IP exchanges data between the Ethernet Gateway Link150 and other compatible Modbus TCP/IP devices over TCP port 502.

(RSTP), and Device Profile for Services Web Services (DPWS).

## 2) TM4ES4 Ethernet Module

TM4ES4 Ethernet module is an industrial "Ethernet switch" that not only provides an Ethernet port for PLCs that do not have an embedded Ethernet interface, but also enables network elements to communicate with each other. This module, shown in Fig. 2, has 4 Ethernet ports with RJ45 connectors. An industrial network was established using two of these Ethernet switches in the system.

## D. Technical Specifications of Hardware System

In the system (Fig. 3), ultrasonic level transmitter and Pt-100 temperature sensor were used for measuring process variables. Level control was provided by speed control of a variable speed pump motor on the input line, while temperature control was conducted by PWM control of an electric heater for proportional control purpose. With a 3-phase 380 V, 50 Hz, 4-pole, low voltage squirrel cage AC motor, the open loop control of the conveyor belt system was performed with the ATV12 AC motor drive via the data sent through the PLC program. ATV12H075M2 is a variable speed drive for asynchronous motors up to 0.75 kW with a 1-phase supply voltage and 220-240V 3-phase output.

TM4ES4 industrial "Ethernet switch" and CAT5e cables were used in the network installation. EGX150 Ethernet gateway with 2 Ethernet ports and RS-232C and RS485 serial ports was used for data transfer using Modbus TCP/IP protocol between SCADA server and ATV12 motor drive.

Schneider Electric Modicon TM241CEC24R PLC

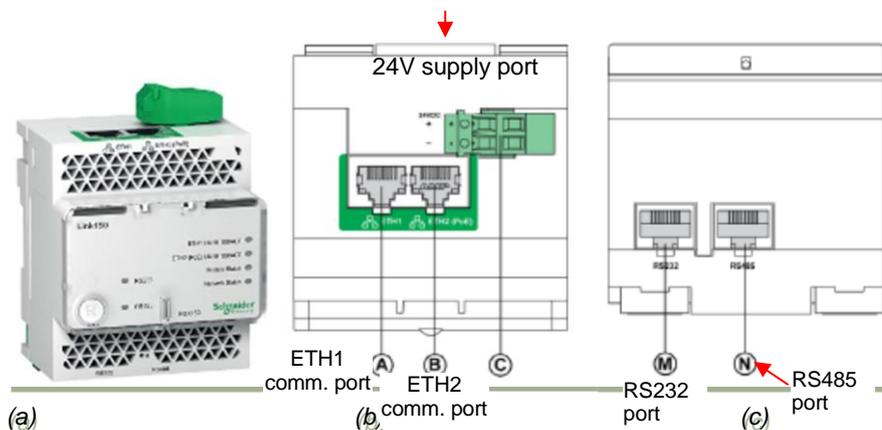


Fig. 2 (a) EGX150 Ethernet gateway, (b) top view, (c) bottom view

Other features are, Hypertext Transfer Protocol (HTTP) that provides web server functionality over TCP port 80; Hypertext Transfer Protocol Secure (HTTPS) which adds a layer of security to data transferred over a Secure Sockets Layer (SSL) or Transport Layer Security (TLS) protocol connection providing encrypted communication and secure connection between remote user and Link150; Transfer File Transfer Protocol (FTP) that is used to transfer firmware updates to the Ethernet Gateway Link150 over the TCP 21 port; Address Resolution Protocol (ARP), Rapid Spanning Tree Protocol

was used for each process and motion control, and Schneider Electric HMI S5T touch screen operator panels were used for temperature and level control processes.

## III. CONTROL OF THE NETWORKED SYSTEM

### A. PID Control of Level and Temperature Processes

A feedback is commonly used in proportional-integral-differential (PID) controller in industrial control systems, where the error value  $e(t)$  between the

measured variable  $y(t)$  and set point  $v(t)$  is calculated and reduced to a minimum value by adjusting the process control inputs. The resulting controlled variable is used to activate the actuator that will have a direct effect on the process variable  $u(t)$ .

The PID control algorithm in the feedback control loop, whose block diagram is shown in Fig. 4, consists of three parameters: proportional (P), integral (I) and derivative (D). P is based on the current error, I depends on the accumulation of past errors, and D is the prediction of future errors based on the current rate of change [19]. As used in this project, the weighted sum of these three movements in the level control process changes the speed of a pump motor to bring the level to the desired value, while in the temperature

control process, it adjusts the duty cycle of the PWM control signal sent to the heater.

M241 PLC's "FB\_PID" function block is a standard PID function block with manual tuning, hold function, bumpless transfer from manual mode operation to automatic mode, damping time ( $T_d$ ) to filter the overshoot due to derivative action, and anti-reset wind-up to avoid the saturation or wind-up in integral action. The PID output  $u(t)$  is as shown in (1).

$$u(t) = K_p \left( e(t) + \frac{1}{T_n} \int e(t) dt + \frac{T_v}{T_d} \frac{de(t)}{dt} \right) \quad (1)$$

$$e(t) = v(t) - y(t)$$

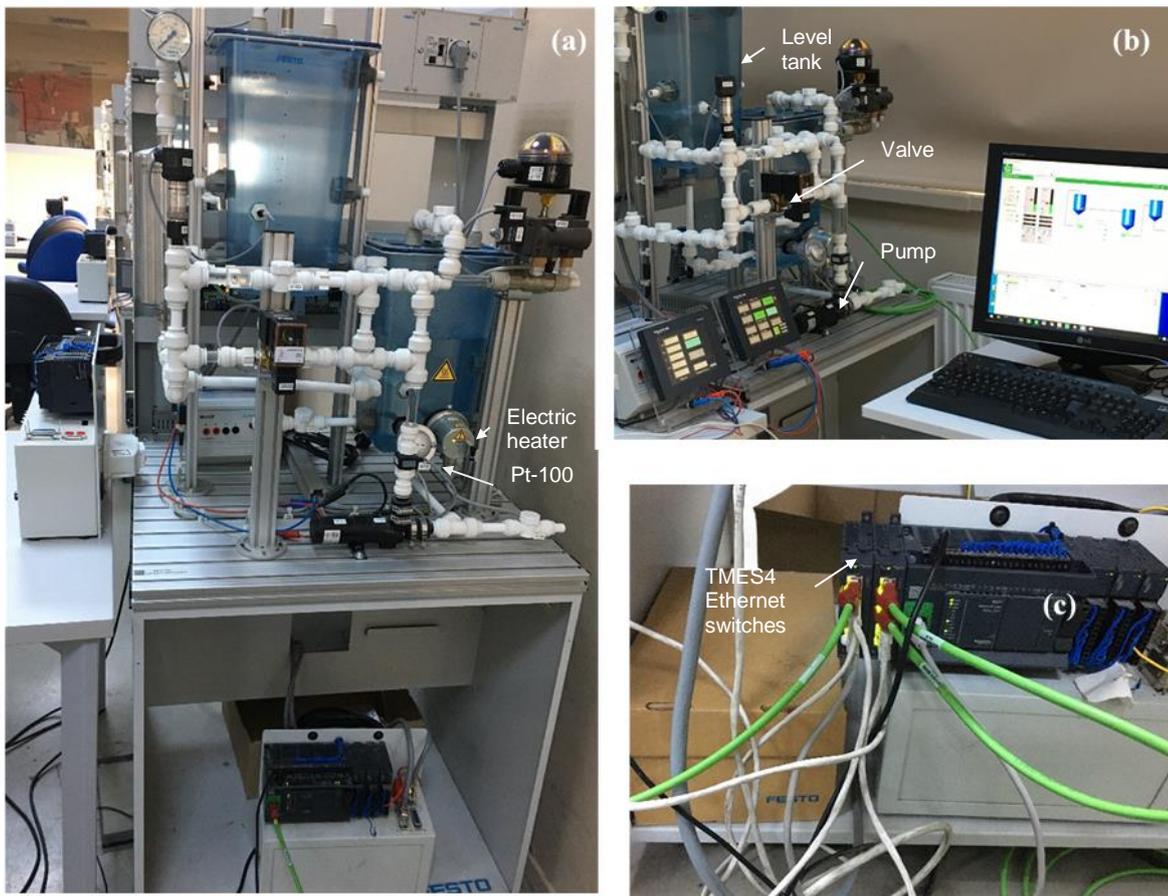


Fig. 3 (a) Heating process, (b) level process, (c) PLCs, motor drive and SCADA server connections through TM4ES4 Ethernet switch with CAT5e cables

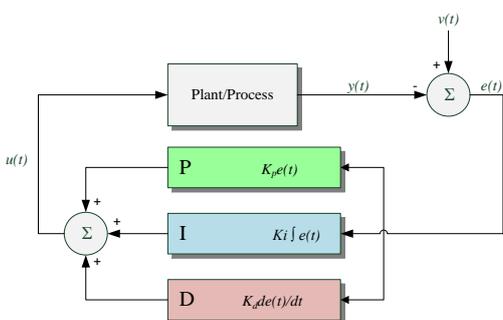


Fig. 4 Block diagram of a PID controller in a feedback loop

$K_p$  refers to the proportional gain that controls the amplitude of the output signal sent to the actuator, but eventually causes a steady-state error. An integral time,  $T_n$ , is set to generate an additional control signal to the controller to correct this error. Adjusting the derivative time,  $T_v$ , which depends on the current rate of change of the error signal, adds an additional control signal to reduce overshoots, improving dynamic performance. This derivative action is used when necessary to decrease the response time. Derivative mode of control is not added in the controller as the level control in a tank has good speed of response [20], but steady state accuracy is required. Consequently, the level controller is a PI controller. For

the temperature controller, a damped response with a short settling time is required with zero steady state error. Therefore, PID controller is selected for temperature control process.

PI and PID control parameters were manually typed in HMI touch panel in automatic control mode (Fig.5), and PID faceplate, which is a small workspace (Fig.6), in manual control from SCADA workstation screen. The controller settings are presented in Table 1.

*A. Control of Level and Temperature Processes and Data Monitoring with SCADA Software*

The temperature and level values on SCADA screen in runtime is shown in Fig. 6. Since the system operates in automatic mode in PLC, operation and parameter entry from the SCADA screen are not allowed. The measured temperature and level values are transferred from the PLC to the SCADA server computer and displayed on the left screens.

TABLE I. PARAMETER SETTING FOR THE PROCESS CONTROLLERS

Parameters	LIC101	TIC104
Controlled variable	Liquid level	Temperature
Controlled variable range	0–9 l (volumetric)	25–100 °C
Manipulated variable	Opening of valve or varying pump speed	Duty cycle
Manipulated variable range	0–100%	0–100%
Control action	Reverse	Reverse
Kp	45	650
Tn (ms)	1000	2500
Tv (ms)	0	1000

When the system operation is changed to manual mode from SCADA server by setting Auto/Man button to “1”, these parameters are then entered from the graphical screen. Consequently, the heater and pump motor (or proportional valve if selected) operate according to the commands from the SCADA screen. The control programs provide manual control from the SCADA server, as well as automatic control of the system via PLC, together with the control program in CitectSCADA 2016 SCADA software (Fig. 7 for temperature and Fig. 8 for level control, respectively).

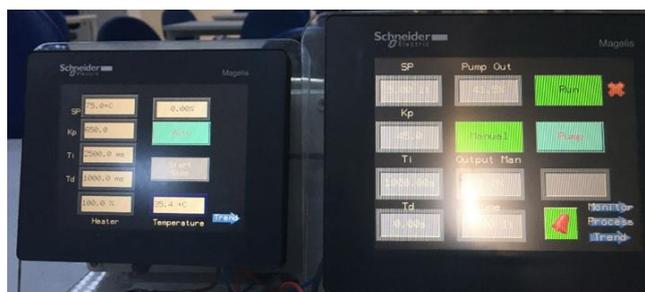


Fig. 5 PI and PID control parameters entry via HMI touch screen in automatic mode of control

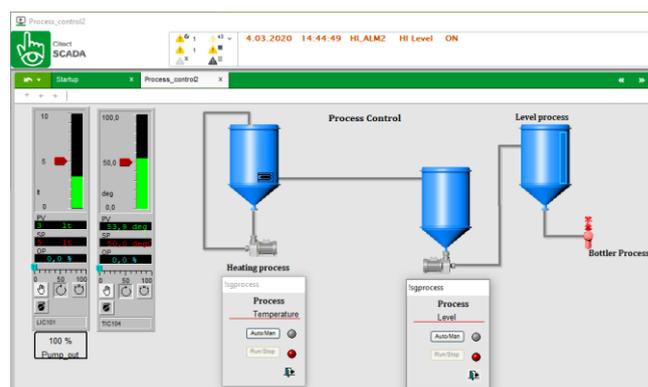


Fig. 5 SCADA graphic display for level (LIC101) and temperature (TIC104) PID parameter entry in runtime

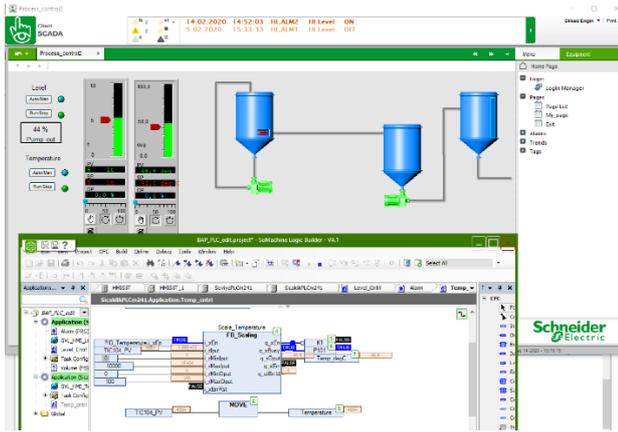


Fig. 7 SCADA screen (top) and PLC program (bottom) for temperature control in runtime

In SCADA systems, fault detection is done promptly by operators who monitor the alarms, and the technical team can intervene in the fault as soon as possible, preventing the systems from being out of service for a long time, thus increasing production efficiency. For this purpose, two alarm tags, as seen on the active alarms page (Fig. 9), were added and the data to generate an alarm received from a level switch was transferred to the SCADA computer to prevent the heater tank from running empty, thus preventing the heater from failing. To prevent overflow in level tank because of an error that may occur in the PID control and to create an alarm, data is gathered from another level switch.

One of the features provided by SCADA systems is to plot trend curves and to monitor instantaneous and historical data graphically. The trend tags added using this feature and the change graphs of analog signals are observed from the Process Analyst page in runtime. The trend for liquid level calculated as the volume of liquid in liters in the tank, and the trend for temperature value acquired from PLCs was plotted through the SCADA program as shown in Fig. 10.

**B. AC Motor Control and Modbus RTU Communication**

The conveyor belt controlled by the asynchronous motor was driven by a variable speed motor drive ATV12H075M2, which is considered as a part of the bottle filling facility in the system. This drive, which can communicate via Modbus RTU protocol, is connected to the RS485 port of the PLC for serial communication. However, this device, which is not directly connected to the network, can only be accessed via PLC. Address assignment, serial data transmission rate, length of data to be transmitted, parity bit selection was made for Modbus RTU communication on ATV12.



Fig. 8 Active alarms page in runtime

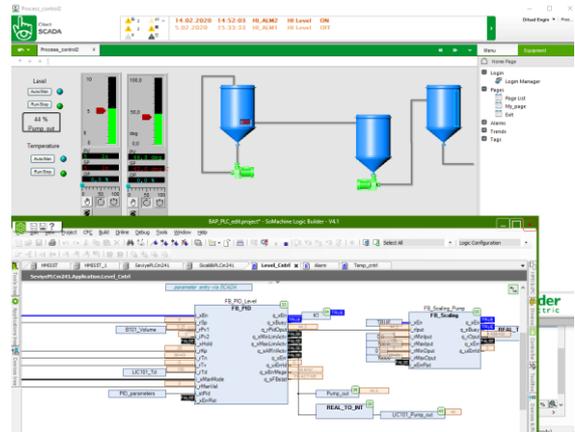


Fig. 6 SCADA screen (top) and PLC program (bottom) for level control in runtime

The same settings were made on the PLC side that will make Modbus RTU communication.

The address set on the drive is written as "1.1" in Modbus RTU format, and the addressing of the device to be communicated is performed by using ADDM command for the PLC controlling the AC motor drive to format and decode the Modbus RTU address. Before launching communication, address of the remote device is read, checked whether the connection is established, and if the connection is established, it is checked whether the remote device is busy or not. When the conditions are met, data transfer started via Modbus commands. The master sends data read request to the slave, if the slave is not busy, it responds to this request and sends data packet to the master. When necessary, the master sends a write request through the data packet to the slave with the Modbus write command.

While using Modbus read and write commands, parameters such as data type, length of data to be read/written, starting word address where data will be read or written are entered together with address data. With the written program, the ETA register for the run/stop and error states of the variable speed motor drive ATV12 was read, and the written speed values were read from the LFR register by writing to the LFR register for speed control. The ACC and DEC registers for the acceleration and deceleration of the motor were also accessed by Modbus commands in the PLC

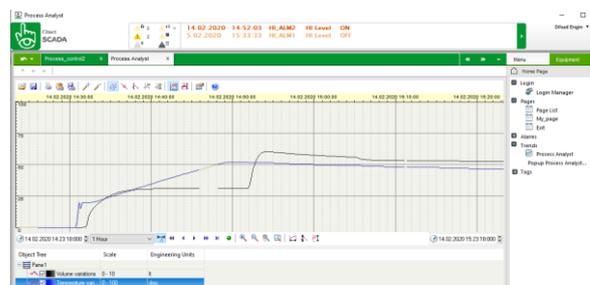


Fig. 9 Process Analyst page in runtime

program. Optionally, a multi-register read/write command is used to write to and read from the SP2-SP8 registers, which are used to write speed values.

The flow chart of the program for data communication is given in Fig. 11.

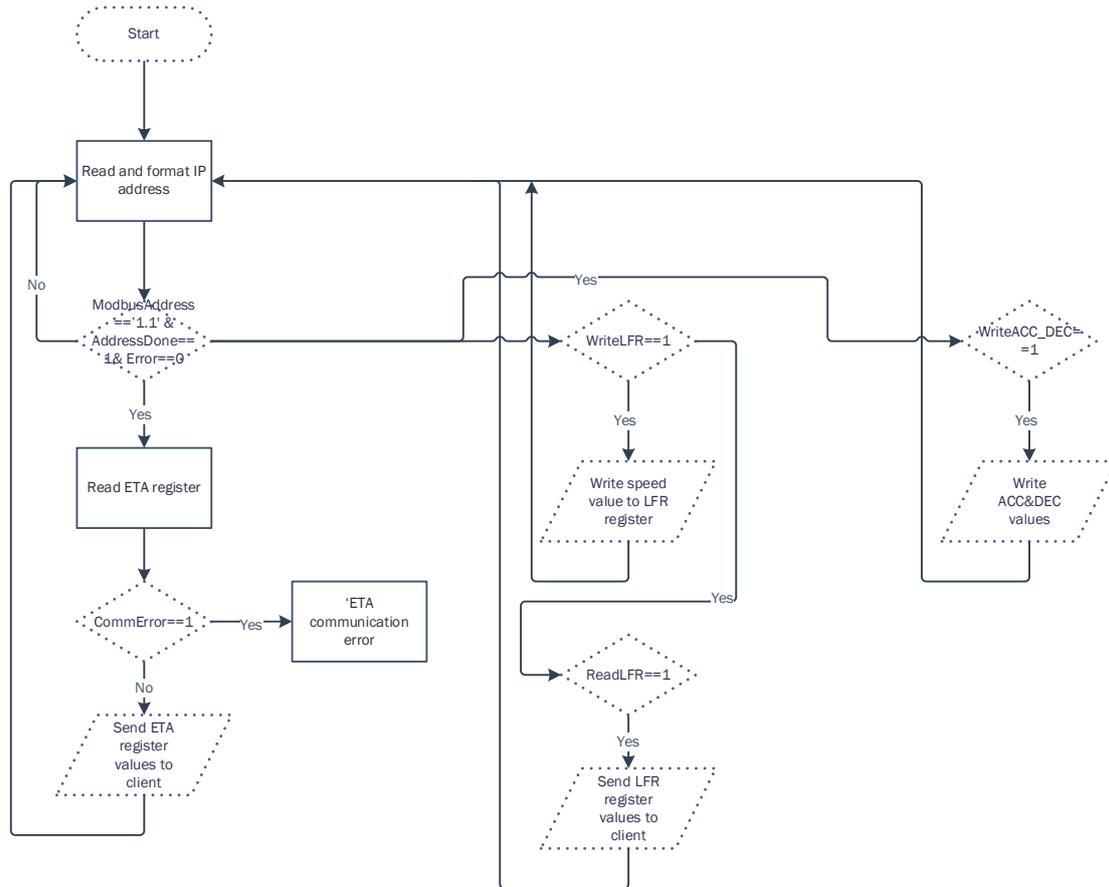


Fig. 10 Flowchart for Modbus RTU communication between ATV12 and M241 PLC

### C. AC Motor Control and Modbus TCP/IP Communication

EGX150 Ethernet gateway is used to perform remote speed and direction control and monitoring of motor drive via SCADA server in the network. This gateway is a communication device that provides connectivity between Ethernet (Modbus TCP/IP) and Modbus serial line devices and allows Modbus TCP/IP clients to access data from slave serial devices. It also allows serial master devices to access data from distributed slave devices over an Ethernet network.

As seen in Fig. 12, EGX150 is used to communicate with the SCADA server computer, which

communicates with the devices on the network with the Modbus TCP/IP protocol, with the ATV12H075M2 variable speed motor drive, which can communicate with the Modbus RTU protocol over the serial line.

Modbus read/write commands are used to communicate ATV12 and SCADA server connected to the network through the gateway. To write the address in TCP "string" format, "3" has been added to the right of the 192.168.1.103 address of the PLC using CONCAT commands. ADDM command is used for Modbus TCP address formatting and address decoding of PLC controlling the temperature process.

During communication, it was checked whether the

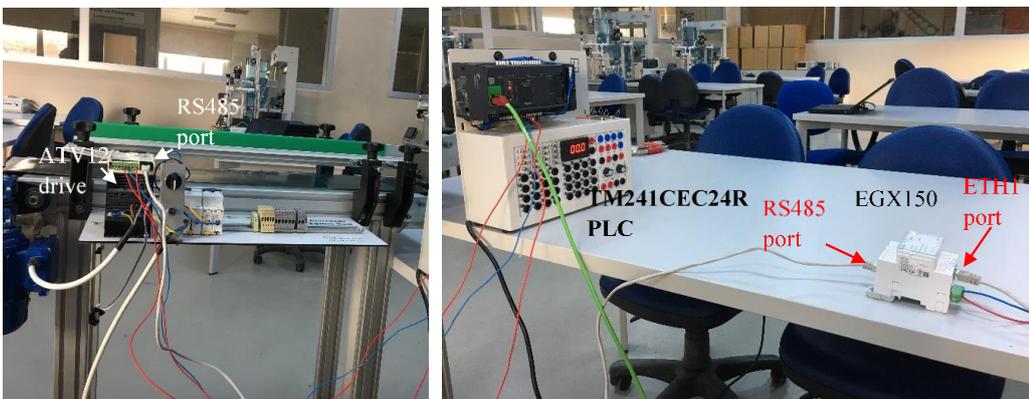


Fig. 11 Connection to ATV12 RS485 serial line on the left, PLC and EGX150 Ethernet gateway with motor drive control

IP address of the remote device is read, and the connection is established, and if the connection is established, whether the remote device is busy. When the conditions are met, data transfer was started via Modbus commands. The client sends read request to the server, if the server is not busy, it responds to this request and sends the data packet to the client.

With another program written to communicate with the motor drive via Modbus TCP/IP protocol, the ETA register was read for run/stop and error situations, as in the program written for serial communication, and the speed values written to the LFR register for speed control were also read from the LFR register. The ACC and DEC registers for the acceleration and deceleration of the motor were also read and written with the Modbus commands. Optionally, a multi-register write/read command is used to write to and read from the SP2 – SP8 registers, which are used to write speed values. The flow chart of the program for this data communication is like the one in Fig. 11.

For writing and reading the parameters, a visual screen shown in Fig. 13 on the left was created by using Visualization of the SoMachine software package, and the system control was performed

through this screen by the program given in Fig. 13 on the right.

#### D. AC Motor Control and Data Monitoring with SCADA Software

The graphic screen created for controlling the variable speed motor drive from the SCADA computer and monitoring the data, which is connected to the network with the EGX150 Ethernet gateway, is shown in Fig. 14. The speed value sent to the drive is multiplied by 0.1 Hz, and the acceleration and deceleration values are multiplied by 0.1s.

While control is provided with run/stop and direction control buttons in SCADA, parameters are entered from numerical screens defined for motor frequency, acceleration, and deceleration values. Motor direction data can also be viewed with SCADA screen animations. The screen change command to switch to the temperature and level process control page is provided by clicking the "Process Control" text on the visual screen.

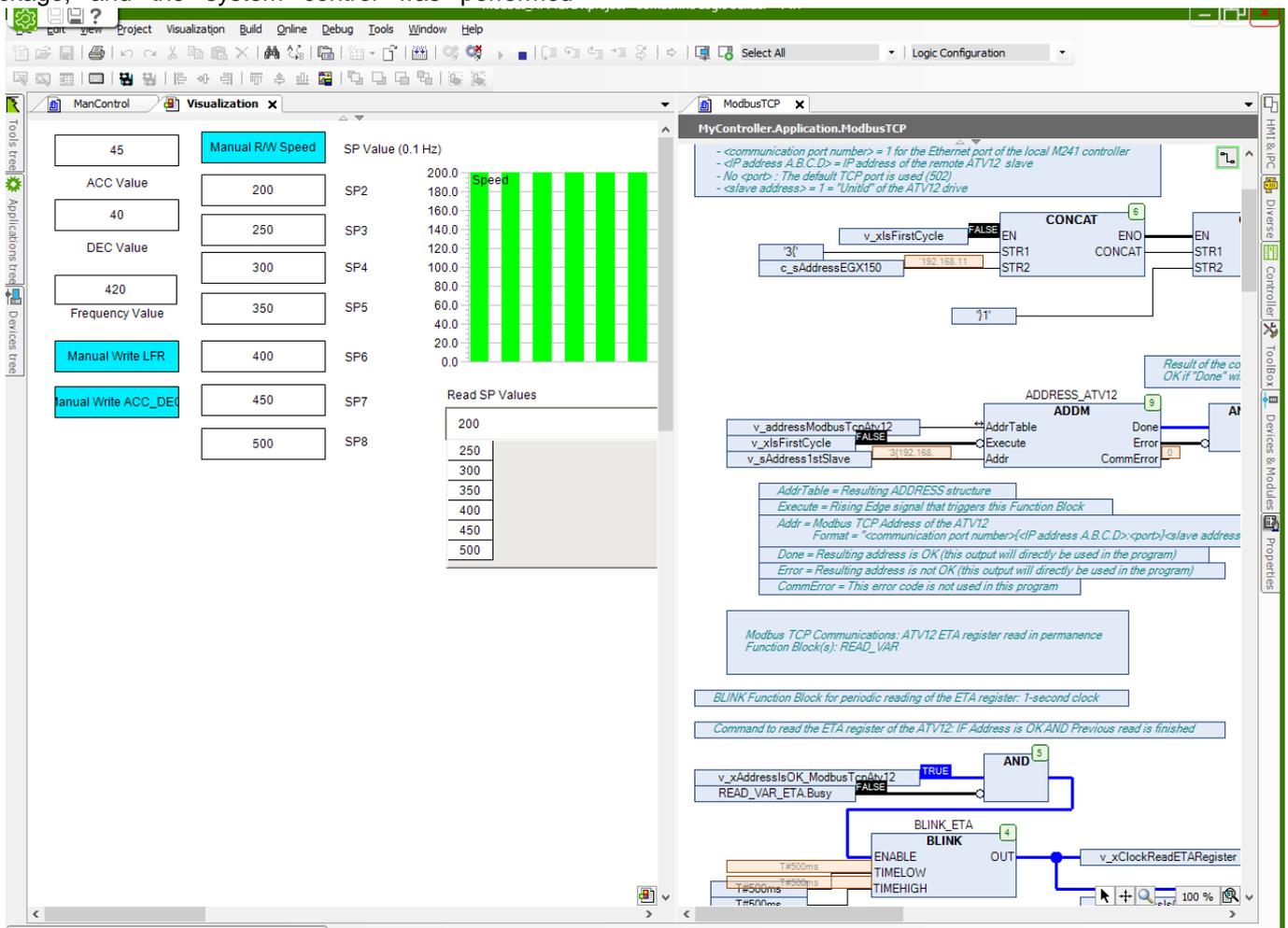


Fig. 12 AC motor control visualization screen on the left and part of the program including Modbus TCP/IP communication addressing on the right

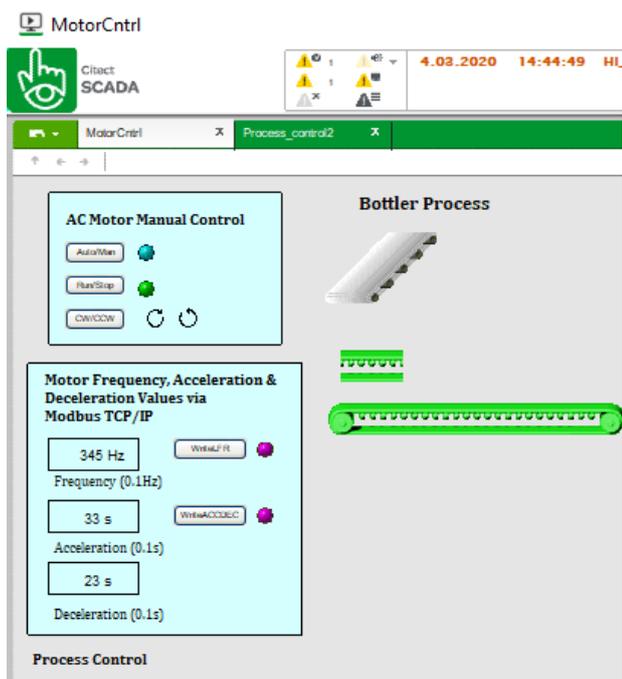


Fig. 13 SCADA screen for AC motor control

Using CitectSCADA 2016 software, settings have been made to provide supervised control of the system, to add I/O devices to be controlled, monitor data, generate analog and digital alarms, plot trend curves and store changes. Since communication with PLC will be provided by adding variables, their addresses were written same as the addresses in the PLC program. The variables used in HMIs were transferred to the SoMachine software integrated Vije Designer program.

#### IV. RESULTS AND CONCLUSION

Today, the digitization in industry significantly improves productivity and efficiency of autonomous production processes where personnel, machines, instruments, intelligent processes, and manufacturing mechanisms communicate and cooperate with each other directly. This is also the vision of Industry 4.0. Leading companies worldwide have accelerated their work on this issue. In this paper, we aimed to develop software that can be used in automation systems and to automatically control machines and processes. In this context, this work is an industrial system used in factory automation in a laboratory environment, providing remote monitoring of data and control of outputs by using Ethernet-based Modbus TCP/IP and Modbus RTU protocols by implementing a network structure that enables PLCs and motor drives in Control and Automation and Mechatronics Laboratories to communicate in distributed control architecture. The software for the network system was developed, and communication between the SCADA server and three PLCs and a variable speed motor drive was ensured within the network structure.

Instead of local control of the machines with PLCs and remote-control hardware and software, the integrated operation of the machines can be monitored from a single center. The use of systems consisting of

PLC and control computers, which allows for the detection and remote intervention of problems from this control center, because of monitoring and reporting the production stages, make automation process more efficient. An industrial network was established for data transfer at device level and supervisor computer level in the control of output elements via PLCs. For this purpose, Ethernet based industrial communication protocol Modbus TCP/IP, and Modbus RTU protocol for serial communication are used.

A variable speed asynchronous motor drive capable of serial communication via the Modbus RTU protocol with the Ethernet gateway was connected to the network and communicated with the SCADA server computer via Ethernet-based Modbus TCP/IP protocol. In this way, the remote motor drive parameters were read and the parameters that needed to be changed were written to the drive through a remote computer. In this way, by means of remote control of the devices in the field and parameter screens, the operating values were changed according to the requirements of the system, enabling them to work more efficiently. Possible changes in any parameter value, motor start/stop states and fault conditions were monitored by remote reading of the drive registers, making it possible to make the necessary intervention in a timely manner.

The ability to transfer data between PLCs on the network via industrial Ethernet provides optimum data flow and data synchronization in process control, while allowing to program more than one PLC program at the same time. Since the SoMachine PLC programming software used in the project supports online changes via networked PLCs, it saves time and service costs by eliminating problems that may occur in more than one PLC from a remote location. Therefore, PLCs on the network were programmed through a single computer, configuration settings were changed when necessary, and the changes in the program were reloaded to all PLCs or selected PLCs at the same time.

We have accomplished our goal in remote real-time monitoring of a networked system through a SCADA software and controlled the system via three PLCs.

#### ACKNOWLEDGMENT

The authors thank to "Scientific Research Projects Coordination Unit" of Ege University for funding the project 18-EMYO-001.

#### REFERENCES

- [1] J. Velagic, A. Kaknjo, N. Osmic, and T. Džananovic, "Networked based control and supervision of induction motor using OPC server and PLC," in *Proceedings Elmar - International Symposium Electronics in Marine*, 2011, pp. 251–255.
- [2] "SCADA Systems," *Motorola, Inc. White Paper*, 2007.
- [3] F. M. Enescu, N. Bizon, and I. C. Hoarca, "Energy Management of the Grid-Connected PV Array," in *Power Systems*, N. Mahdavi Tabatabaei, E.

Kabalci, and N. Bizon, Eds. Cham: Springer International Publishing, 2020, pp. 255–288.

[4] D. Babunski, E. Zaev, A. Tuneski, and D. Bozovic, "Optimization methods for water supply SCADA system," in *2018 7th Mediterranean Conference on Embedded Computing, MECO 2018 - Including ECYPS 2018, Proceedings*, 2018, pp. 1–4.

[5] X. Wang, G. Li, and X. Wei, "PLC-based SCADA system for oil storage and application," in *2011 International Conference on Electric Information and Control Engineering, ICEICE 2011 - Proceedings*, 2011, pp. 1539–1541.

[6] Q. H. Seer and J. Nandong, "Stabilization and PID tuning algorithms for second-order unstable processes with time-delays," *ISA Trans.*, vol. 67, pp. 233–245, 2017.

[7] J.-C. Jeng, W.-L. Tseng, and M.-S. Chiu, "A one-step tuning method for PID controllers with robustness specification using plant step-response data," *Chem. Eng. Res. Des.*, vol. 92, no. 3, pp. 545–558, 2014.

[8] C. Yu Jin, K. H. Ryu, S. W. Sung, J. Lee, and I. B. Lee, "PID auto-tuning using new model reduction method and explicit PID tuning rule for a fractional order plus time delay model," *J. Process Control*, vol. 24, no. 1, pp. 113–128, Jan. 2014.

[9] J. Pongfai, X. Su, H. Zhang, and W. Assawinchaichote, "A novel optimal PID controller autotuning design based on the SLP algorithm," *Expert Syst.*, vol. n/a, no. n/a, p. e12489, Nov. 2019.

[10] A. Maghareh, S. J. Dyke, and C. E. Silva, "A Self-tuning Robust Control System for nonlinear real-time hybrid simulation," *Earthq. Eng. Struct. Dyn.*, vol. n/a, no. n/a, Feb. 2020.

[11] C. Engineering, "Pros and cons of autotuning control Part 2," *Control Engineering Magazines and Newsletters*. CFE Media, 02-Aug-2018.

[12] E. B. Priyanka, C. Maheswari, and B. Meenakshipriya, "Parameter monitoring and control during petrol transportation using PLC based PID controller," *J. Appl. Res. Technol.*, vol. 14, no. 2, pp. 125–131, Apr. 2016.

[13] B. B. Alagoz, A. Ates, and C. Yeroglu, "Auto-tuning of PID controller according to fractional-order reference model approximation for DC rotor control," *Mechatronics*, vol. 23, no. 7, pp. 789–797, Oct. 2013.

[14] Z. Dandan, Z. Zhiyun, Y. Meng, H. Yue, and G. Chen, "Application of relay feedback auto-tuning algorithm in a liquid level PID control system," in *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, 2015, vol. 2015-March, pp. 4313–4315.

[15] E. R. Alphonsus and M. O. Abdullah, "A review on the applications of programmable logic controllers (PLCs)," *Renewable and Sustainable*

*Energy Reviews*, vol. 60. Elsevier, pp. 1185–1205, 01-Jul-2016.

[16] H. Jack, *Automated Manufacturing Systems with PLCs*, 6th ed. 2009.

[17] S. Da'na, A. Sagahyoon, A. Elrayes, A. R. Al-Ali, and R. Al-Aydi, "Development of a monitoring and control platform for PLC-based applications," *Comput. Stand. Interfaces*, vol. 30, no. 3, pp. 157–166, Mar. 2008.

[18] S. G. McCrady, "Introduction," in *Designing SCADA Application Software*, S. G. McCrady, Ed. Oxford: Elsevier, 2013, pp. 1–10.

[19] M. Araki, "Control systems, robotics and automation - PID control," *Encyclopedia of Life Support Systems*, vol. II. UNESCO, pp. 1–23, 2002.

[20] M. A. Johnson, "PID Control Technology," in *PID Control: New Identification and Design Methods*, J. Crowe, K. K. Tan, T. H. Lee, R. Ferdous, M. R. Katebi, H.-P. Huang, J.-C. Jeng, K. S. Tang, G. R. Chen, K. F. Man, S. Kwong, A. Sánchez, Q.-G. Wang, Y. Zhang, Y. Zhang, P. Martin, M. J. Grimbale, D. R. Greenwood, M. A. Johnson, and M. H. Moradi, Eds. London: Springer London, 2005, pp. 1–46.