Automating a Heat Shrink Tubing Process

Shahrokh Yousefi-Darani
University of North Texas
shahrokhyousefidarani@my.unt.edu

Nihal Joshua
University of North Texas
nihaljoshua@my.unt.edu

Huseyin Bostanci
University of North Texas
huseyin.bostanci@unt.edu

Enrique Barbieri
University of North Texas
enrique.barbieri@unt.edu

Abstract

Heat shrink tubing is used to insulate wire conductors, protect wires, and to create cable entry seals in wire harnessing processes. Performing this sensitive process manually is time-consuming, the results are strongly dependent on the operator's expertise, and the process presents safety concerns. Alternatively, automating the process minimizes the operators' direct interaction, decreases the production cost over the long term, and can improve quantitative and qualitative production indicators dramatically.

This paper describes the automation of a heat shrink tubing prototype machine that benefits the wire harnessing industry. As is the case with any product development project, the prototype is a result of identifying user needs, interpreting them to meet tangible engineering parameters, and performing a conceptual design based on cost and manufacturing constraints. The prototype consists of an instrumented heat chamber on a linear positioning system fitted with one or more heat guns. The chamber design allows directing hot air onto the wire harness uniformly through radially-distributed channels. Heat exposure time as a major factor in the tubing shrinking process can be adjusted by controlling the linear speed of the heat chamber. The linear positioning system is designed to move the heat chamber along the wire harness as the proper shrinkage temperature level is reached. A control unit manages the actuator position continuously by measuring the actual chamber's position, speed, and temperature. A model-based design approach is followed to design the control system, and MATLAB/Simulink is used as the simulation environment. A programmable logic controller is selected as the controller implementation platform. The prototype machine will undergo testing to meet required industrial standards.

Introduction

This article describes the integration of a practical solution for an engineering problem of direct industrial relevance. Although we target the wire harnessing industry as an application-specific example, the procedure can be expanded for a variety of other control and automation applications in industrial environments.

The process of interest is wire harnessing, that is, the assembling of conducting wires bound together to transmit signals or electric power. The wires are typically bound by straps, cable lacing, sleeves, electrical tape, or a weave of extruded string. Wire harnesses provide several benefits over loose wires. Many aircraft, automobiles, and spacecraft contain large bundles of wires that would stretch over several kilometers if fully extended. By putting all of these wires into a harness, they can be better secured against the adverse effects of vibrations, abrasions, and moisture. Moreover, space utilization is optimized, and the risk of an electrical short is reduced. Since the worker has only one harness to install, installation time is also decreased and the process can be easily standardized [1].

The wire harnessing process consists of several steps to produce quality wire harnesses and involves engineering design and development efforts, labor work, and machinery operation. Wire harnesses are usually designed according to geometric and electrical requirements so the process sequences and specifications are determined to suit a particular application.

A heat shrink tube is ordinarily made of nylon or polyolefin, which has the capability of shrinking about its diameter axis when heated. This feature allows the tube to be used in a wide range of applications, from near microscopically-thin-wall tubing to rigid, heavy-wall tubing. Among all existing applications, heat shrink tubes can be used to insulate wire conductors in wire harnessing processes. Heat shrink tubes can also be used to repair the insulation on wires or to bundle them together, to protect wires, and to create cable entry seals. Besides serving as an electrical insulator, the heat shrink tube provides environmental protection against dust, solvents, and other foreign materials and is mechanically held in place by its tight fit [2].

In order to shrink the tube in a wire harness process, the tube is first fitted on the wire bundle before making the connection and then shrunk to wrap tightly around the joint by heating with a hot air gun or other heating source. Uncontrolled heating can cause uneven shrinkage, physical damage, and insulation failure. If overheated, the tubing can melt, scorch, or catch fire like any other plastic [3]. Thus, the heating process should be controlled precisely to result in the desired tube profile. Typically, skilled operators accomplish the job manually, which is time-consuming and prone to safety concerns. Hence, automating the process minimizes the operators' direct interaction, reduces the production cost over the long term, and can improve quantitative and qualitative production indices dramatically.

Although the heat shrink tubing process is done manually in many industries, some manufacturers provide automated solutions for this application. For some products, the work pieces are loaded into the machine manually and passed through a heating tunnel automatically. This feature makes this solution ideal for large production requirements because the throughput is determined by the feeding rate at which the operator loads the assemblies. These kinds of machines can address the precise shrinkage specifications due to automatic movement of work pieces, but they do not easily process varying lengths of wire harnesses as the heating tunnel dimensions are limited.

Based on the discussion presented above, a gap exists between industry's needs concerning the heat shrink tubing process and existing solutions in the market. Similar to any new product development project, identifying the user needs, interpreting them to the tangible engineering parameters, and performing a conceptual design based on cost and manufacturing constraints are the step-by-step phases that need to be carried out. These phases are described in the remainder of the paper.

Conceptual System Design

Conceptually, the team settled on the design and fabrication of a heat chamber fitted with two heat guns, dimensioned to allow the controlled passing of up to a 3-inch diameter wire bundle wrapped by heat-shrink tubing, and connected to a motor-driven linear motion system that would accommodate varying wire harness lengths from 1 to 15 feet. The chamber's linear speed and inside temperature are monitored and controlled to expose the wire harness and tubing to the appropriate temperature for a pre-determined time, resulting in the desired tubing shrinkage.

A small-scale prototype shown in Figure 1 was developed first to evaluate the chamber design, shown in Figure 2, with short segments of wire bundles. This prototype consists of an instrumented heat chamber on a linear positioning system and fitted with two symmetrically located heat guns. The chamber design allows for the directing of hot air from the heat guns onto the wire harness through channels. These air channels are radially distributed to provide uniform heating and also angled in the direction of movement to pre-heat the unshrunk portion of the harness.

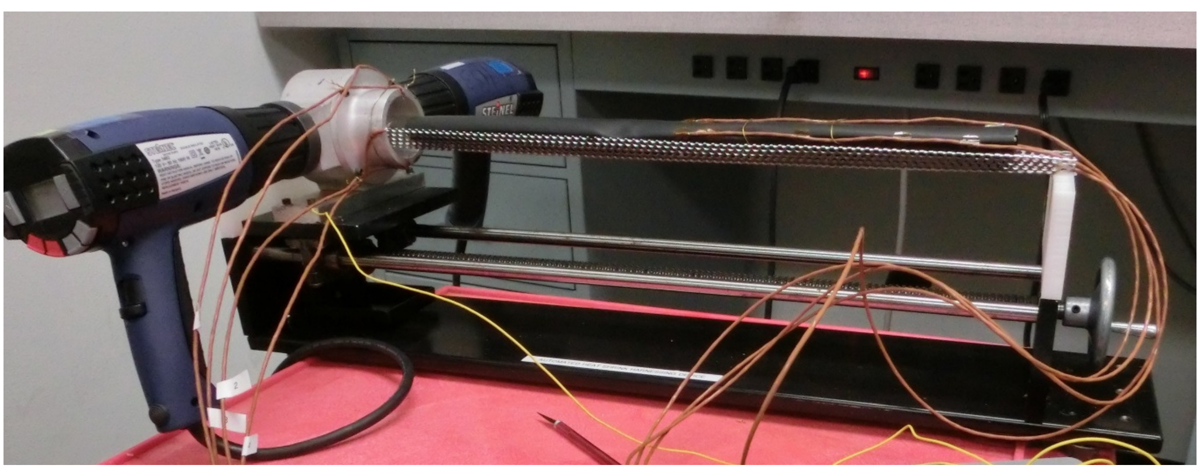

Figure 1. Prototype heat shrink tubing machine

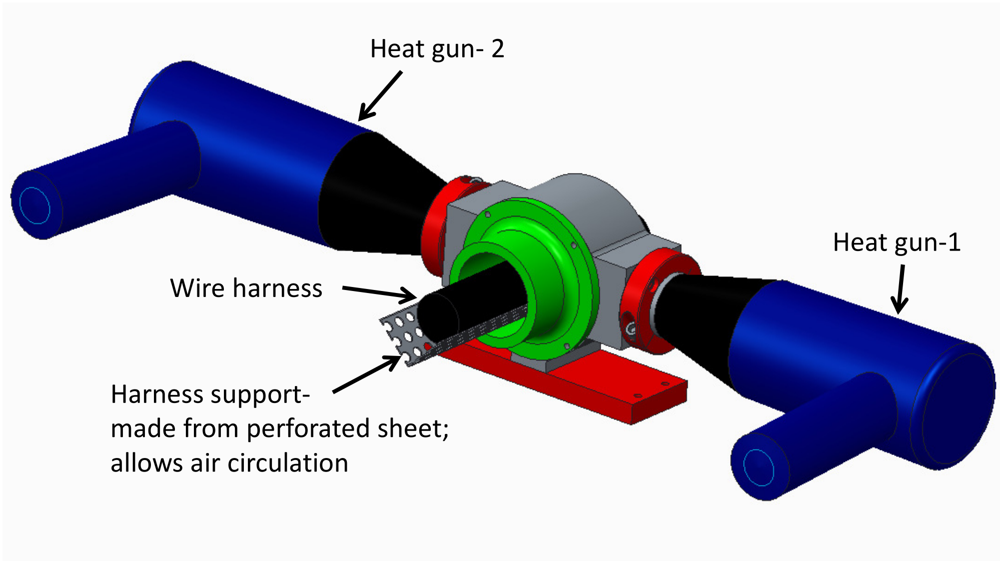

Figure 2. Prototype heat chamber design

A linear positioning system is designed to move the heat chamber along the wire harness as the recommended shrinkage temperature level (175-200^{o} C) is reached. Due to ambient conditions and potential variations in the heat guns operation, a series of experiments were conducted on a tubing segment and provided data that relates the hot air temperature/flow rate at the chamber exit and the linear speed of the chamber for a proper shrinkage process. A look-up table approach was selected relating the air temperature inside the heat chamber and desired speed values (T_d, ω_d). As the temperature fluctuates around its desired set-point, the speed set-point is adjusted accordingly to compensate. This is technically an open-loop solution; however, a closed-loop strategy requires monitoring of the actual tube shrinking—perhaps using an image processing system, which would prove to be unnecessarily complex and cost ineffective.

The other requisite is a controller unit to manage the actuator position and speed to meet the expected heat exposure performance; nowadays, building control units with the help of embedded controller devices is common. To develop such a control unit, alongside the selection and configuration of an appropriate hardware, the control algorithm should be developed and implemented in a software environment. Then the software is deployed into the hardware platform. Based on the controlled process, plant environment characteristics, and control philosophy, different platforms could be used for different applications. A programmable logic controller (PLC) is selected for this application (versus other kinds of embedded controllers such as ASIC, FPGA, DSP). The main reason for this selection is that a heat shrink tubing machine will work in a harsh industrial environment, and in such cases, PLCs are the best choices. The other reason is that since a PLC is an integrated control system, it will provide isolation, signal conditioning, and current/voltage amplification needed for interfacing with sensor and actuator layers. Eventually, PLCs will communicate easily with other devices like HMI panels and personal computers via predefined standard protocols.

System Modeling and Control Structure

Modeling methods are divided into two major groups: data-driven and analytical models. Data-driven modeling uses techniques like system identification, whereas analytical modeling creates a block diagram model that realizes differential/algebraic equations governing system dynamics. A type of analytical modeling is physical modeling, where a model is created by connecting blocks that represent the physical elements that the actual plant consists of. This project benefits from this kind of modeling. In other words, the heat shrink tubing machine or plant is broken down to its physical building blocks and each block will be modeled separately. Next, all modeled building blocks connect together to model the whole system.

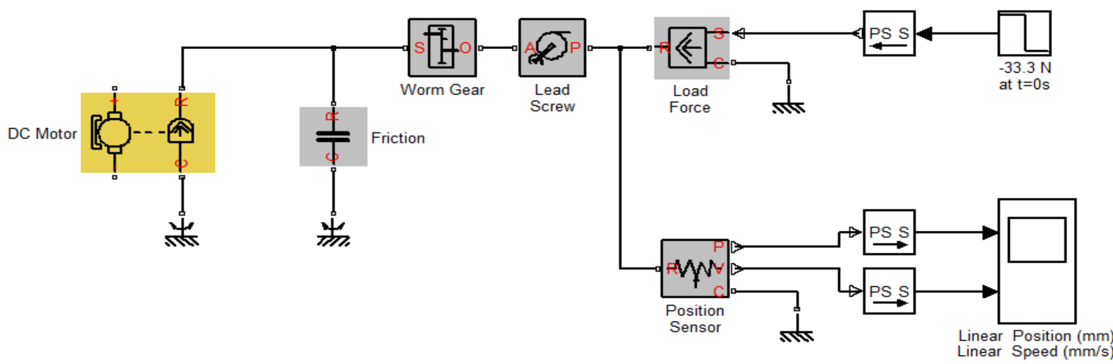

Figure 3. Linear positioning system MATLAB/Simulink model

MATLAB/Simulink from Mathworks has been selected as the simulation environment. It is a multi-domain package that enables this software to be a perfect choice for developing control systems and testing system-level performance. The Simscape library in MATLAB/Simulink provides building blocks from mechanical, electrical, thermal, and other physical domains that make it possible to model a complete physical system without dealing with mathematical equations directly. The designer needs to set up some attributes for each building block. The Simscape model automatically generates the differential equations that represent the system's behavior. These equations are integrated with the rest of the system model and are solved directly. Simscape elements connect together with physically modeled connections and that is why each parameter and variable has its own physical unit, with all unit conversions handled automatically [4].

The heat chamber in the prototype machine is driven manually by turning a screw rod with an attached handle that results in the linear movement (Figure 1). In the full-scale system, a servo motor will be employed to drive the heat chamber automatically. The resulting system model is shown in Figure 3 where the DC motor block represents the equivalent electric circuit of the selected motor and includes the electrical and mechanical characteristics of the motor. The friction block next to the DC motor shows rotational friction between rotating parts that come into physical contact with each other. The friction value is calculated as a function of relative velocity. The worm gear and lead screw blocks represent the mechanisms needed for converting the rotational movement to linear displacement. The load force block model is an ideal force source that is controlled based on the input signal. The word "ideal" means it is powerful enough to maintain a constant force regardless of the velocity at the source terminals. The total 7.49 pound (3.4 kg) load weight of the chamber (including clamps, shell, and base) and two heat guns will give us a 33.32 N load force for simulation purposes. Eventually, the position sensor block simulates a translational motion sensor, and its outputs are linear speed and position.

The automatic heat shrink tubing machine will work while interacting with no other manufacturing facility except for the human operators. In sequence, the operator starts up the machine, places the wire harnesses into it, and supervises its operation mode. The operation modes may be changed by event signals triggered by devices such as push buttons, sensors, and internal signals, which are defined in the control logic and rely on internal variables. Such a control scheme is modeled with a discrete event system (DES). DES is a system that has discrete state space and an event-driven dynamics, i.e., the state can only change as a result of instantaneous events occurring asynchronously over time [5]. In this context, state-chart has been traditionally used to describe these kinds of systems, although there are also other methods such as Petri-Net models. MATLAB/Simulink possesses the capability to develop and simulate a controller in a state-chart diagram. Figure 4 illustrates the corresponding idea for supervisory control purposes.

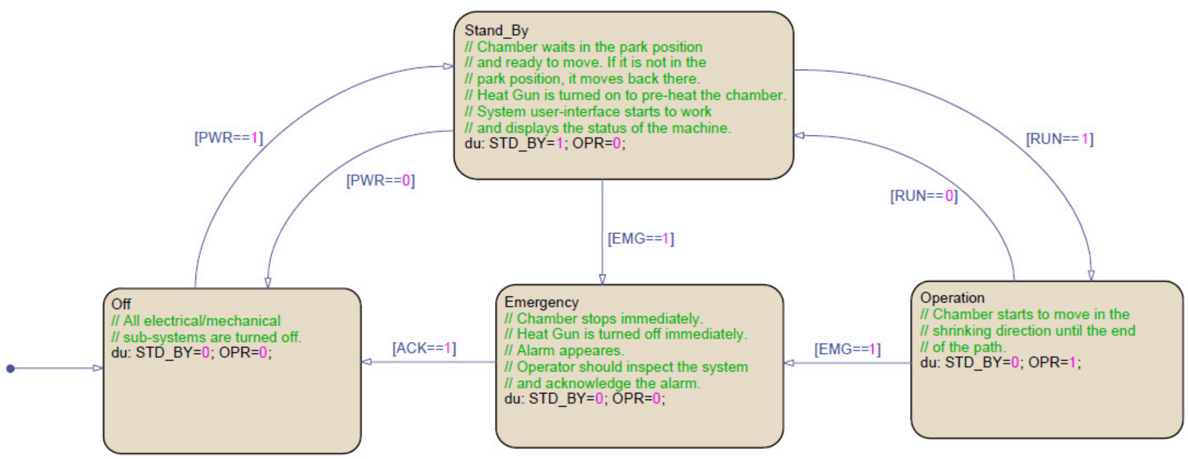

Figure 4. State-chart diagram to manage the operation mode

Among all existing single-input/single-output controllers, one of the most common is the error-driven proportional, integral, derivative or PID control. Many complex control systems may use controller units whose main control building blocks are PID control modules. In a typical PID controller, the derivative (D) term demands more care than using proportional (P) or integral (I) control due to possible noise amplification. In many applications, the I-term or, more generally, the PI-term performs satisfactorily in rendering the system able to track constant set-points or equivalently forcing the error signal to zero. Although more complex to design, in this project we will exploit the benefits of a cascade PI structure consisting of an outer loop controller (master), which controls the primary physical parameter, here the angular speed of the motor, and an inner loop controller (slave), which reads the output of the outer loop controller as set point, usually controlling a more rapidly changing parameter, here the motor's current. It can be shown that the working frequency of the PI controller in cascade style is increased and the time constant of the whole system is reduced [6].

The PI controller parameters K_P and K_I are tuned to shape the closed-loop system characteristics, including response speed, settling time, and overshoot, to guarantee stability and acceptable steady state error. The most common method for tuning a PI controller is based on trial and error using, for example, the SISO tool of MATLAB. There are also analytical methods, such as the Root-Locus, and other frequency domain techniques and practical methods, such as Ziegler-Nichols. These methods all provide a first approximation and the result usually needs further manual adjustment by the designer [7]. In the current project, the tuner utility in the PID controller building block is used to design the controller parameters. The inner loop must be tuned first while the outer loop is not applied. Then, the inner loop is set in the tracking mode when the outer loop is tuned. Figure 5 shows the schematic block diagram of the controller designed in MATLAB/Simulink.

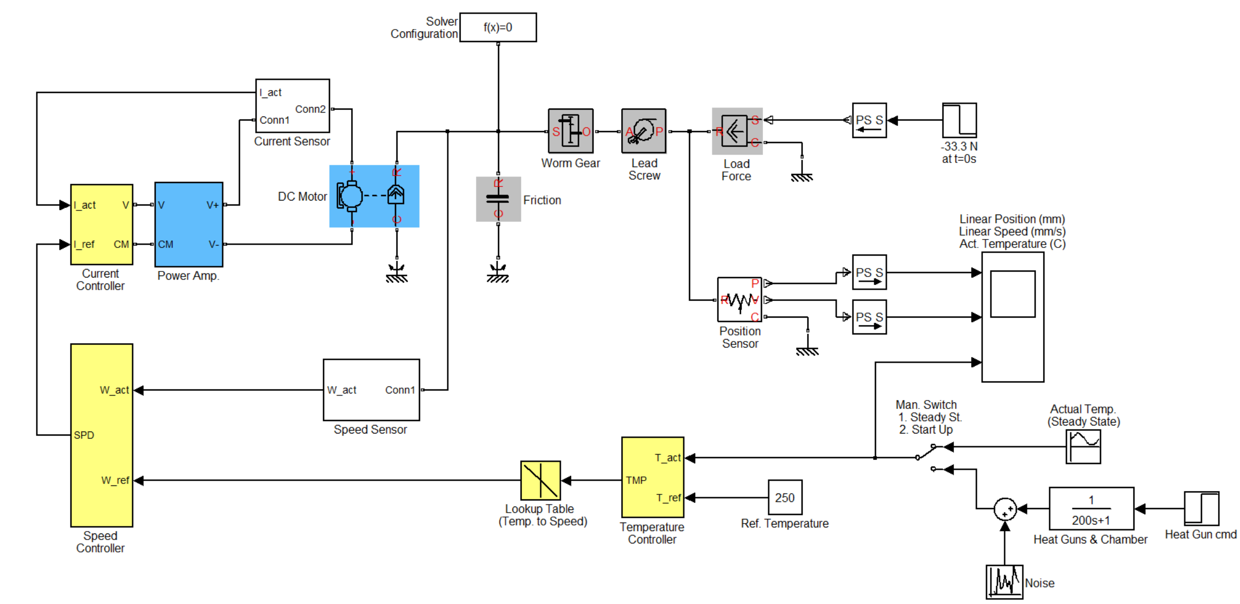

Figure 5. System model along with the designed controller block diagram

Controller Implementation

PLCs stand to be a good choice for implementing the controller in the current application. They meet the computation needs, either arithmetically or logically. Their effectively shielded packaging lets them work well in industrial environments having electromagnetic noises and dust. Because power consumption is not as much a concern in a typical industrial controller design as it is a concern in portable device design, this factor does not play a determinant role in hardware selection process. Other benefits of PLCs are ease of upgrade to higher performance versions, availability of technical support by third-parties, possibility of performing minor logic modifications by trained technicians, and existing standard communication protocols for HMI systems. Another justification in using PLCs rather than another kind of embedded controller is that a PLC is designed to work in a real- time manner. The inputs are read at one time and saved. The logic then is processed sequentially and, at the end, the outputs will be updated. This allows precise timing of execution and things such as endless loops will be minimized. This is an important concept in industrial automation systems, where an undesired delay could result in a costly consequence. Always, the cycle time that it takes to execute the logic is measured, and if it exceeds a predefined value, the developer knows that there is a real problem and the PLC needs to execute the timeout sequence. Although most of the mentioned features are feasible in many other kinds of embedded controllers, because PLCs come as pre-configured structures, any allocated time and cost could be spent on control algorithm instead of implementation techniques.

After selecting the appropriate hardware platform, software implementation is carried out according to the existing programming standards. As far as PLCs are concerned, a variety of programming languages are based on the IEC-61131-3 standard, with each one fit for a specific application. For example, Structure Text is known as a high level PLC programming language that is used for complicated algorithms while graphical languages like Ladder Diagram or Function Block are more suitable for simple logics. The latter group is not as flexible as the Structure Text, but it is easier to trace and debug, and that is why mostly engineers tend to use this kind of language. In the existing project, Structure Text programming language is selected to implement the part of the controller software that corresponds to the system level logic (supervisory control), while Ladder Diagram is used for the chamber position controller. The Structure Text code for system level logic is created by automatic code generation from the controller developed in MATLAB/Simulink. This will decrease the possible errors and development time dramatically. Using the Simulink PLC Coder utility, control system designers can spend more time finetuning the algorithm through rapid prototyping and experimentation, and less time on coding effort. The generated code will be imported into the relevant integrated development environment (IDE). As a result, the application code will be compiled and deployed to the PLC.

Controller Test

Before integrating the implemented controller with the real heat shrink tubing machine, a round of tests should be performed to check the implemented controller performance. These tests help to detect any possible problem and fix it in the right time and before driving the real instruments. Due to computational and graphical capabilities of MATLAB/Simulink, it makes sense to keep this software package in the controller test process, even after the preliminary design and simulation phase. In the former test section, both controller and system under control (heat shrink tubing machine) were modeled and examined in the MATLAB/Simulink environment, but now what is subject to test here is the real control unit. So, in the next test step, the real controller (PLC) is connected to the simulated machine model in MATLAB/Simulink, and controller performance is examined. Generally, this approach is addressed as hardware-in-the-loop (HIL) test; however, in a typical HIL test, the simulated object under control should be run on a hardware platform and an operating system with a real-time kernel. It is clear that MATLAB/Simulink cannot present a real-time behavior while running on an ordinary operating system like Microsoft Windows, but this test still could be helpful.

In the experimental setup, the first step is to provide a solution for data exchange between the controller and the heat shrink tubing machine model in MATLAB/Simulink. One solution for feeding the needed data from MATLAB to the controller unit and vice versa is the use of special I/O modules that are installed on personal computers and supported by MATLAB. Such systems may be suitable for usual laboratory tests but are rarely used in industrial applications because not only they increase the test cost considerably, but they also create many integration problems [8]. Another solution could be construction of an application programming interface in MATLAB, which listens to the traffic on the PLC network and, if necessary, returns data. In comparison with the former solution, the main advantage of this approach is that MATLAB does not have to be integrated with the peripheral cards, and its main disadvantages are that the building up of such an interface is time-consuming and the result is not standard [8].

Eventually, a common solution would be to use the OPC (OLE for process control) standard, a solid and efficient method to establish a communication between MATLAB/Simulink and PLC. OPC technology makes it possible for software and hardware from different brands to integrate and presents an easy and effective solution for communication between PC-based applications such as MATLAB/Simulink on one side, and process devices such as PLCs on the other side [9]. Figure 6 demonstrates the data communication architecture schematic in the developed test system. In MATLAB, the OPC Toolbox provides blocks in the Simulink environment for interacting with a typical OPC server.

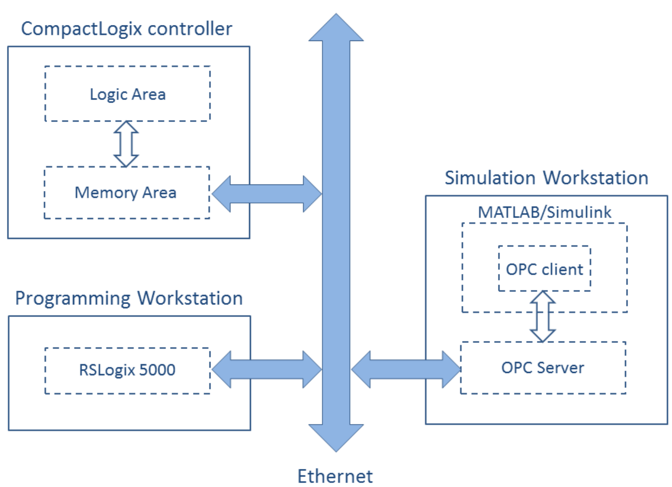

Figure 6. Data communication architecture in the experimental setup

Figure 7 pictures the experimental setup. It consists of a process simulation workstation, network switch, PLC, and PLC programming workstation. Process simulation workstation has two features: MATLAB/Simulink that simulates the process and the OPC server, which is installed in the same computer. Communication of the OPC server with the PLC is achieved via this computer's network interface. Although in the final system the controller signal transfer is done via its I/O modules, in the test system based on OPC server, signal transfer is achieved temporarily by PLC's memory area.

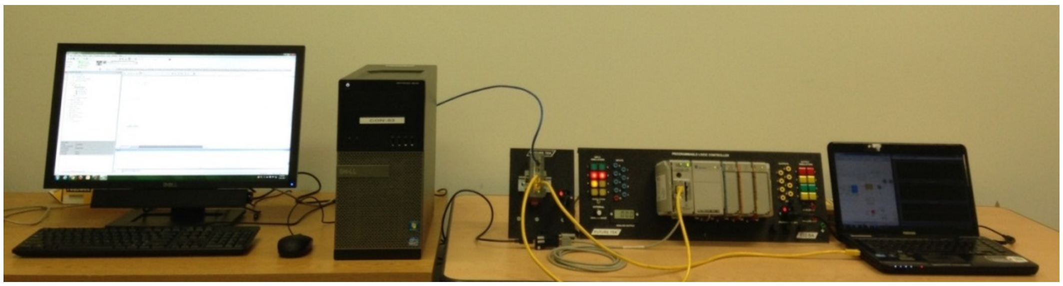

Figure 7. Experimental setup

Three process signals are measured and sent to the PLC via OPC Write block: chamber temperature (to determine the motor speed set point), motor speed (as the master PID loop variable), and motor current (as the slave PID loop variable). The process model receives the control signal (armature driving voltage) from the PLC side via OPC Read block. Figure 8 compares motor actual speed for the modeled and the implemented controller (amplitudes are not in scale due to sensor calibration factor). In each case, the controller is connected to the modeled process.

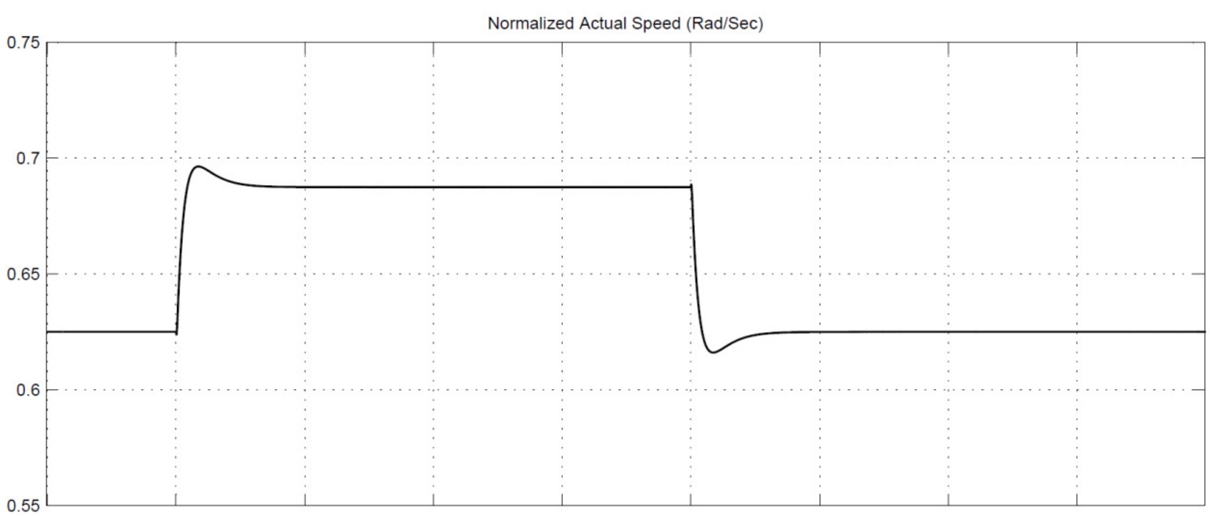

Figure 8a. Motor speed, controller model applied to the process model

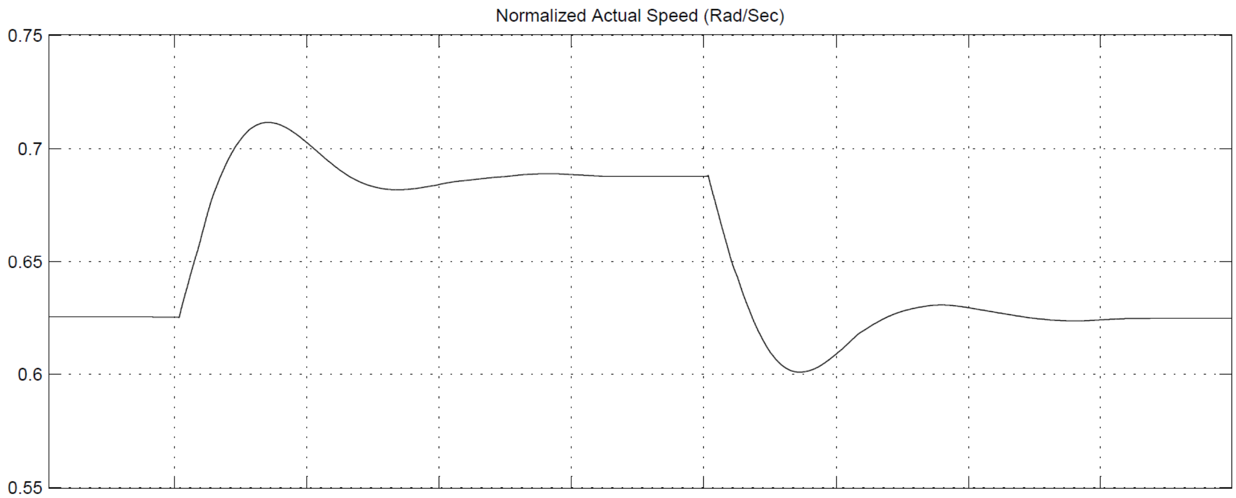

Figure 8b. Motor speed, controller implementation applied to the process model

Figure 9 also exhibits motor current for both simulated scenarios. These figures show that the real controller behaves sufficiently similar to the modeled controller.

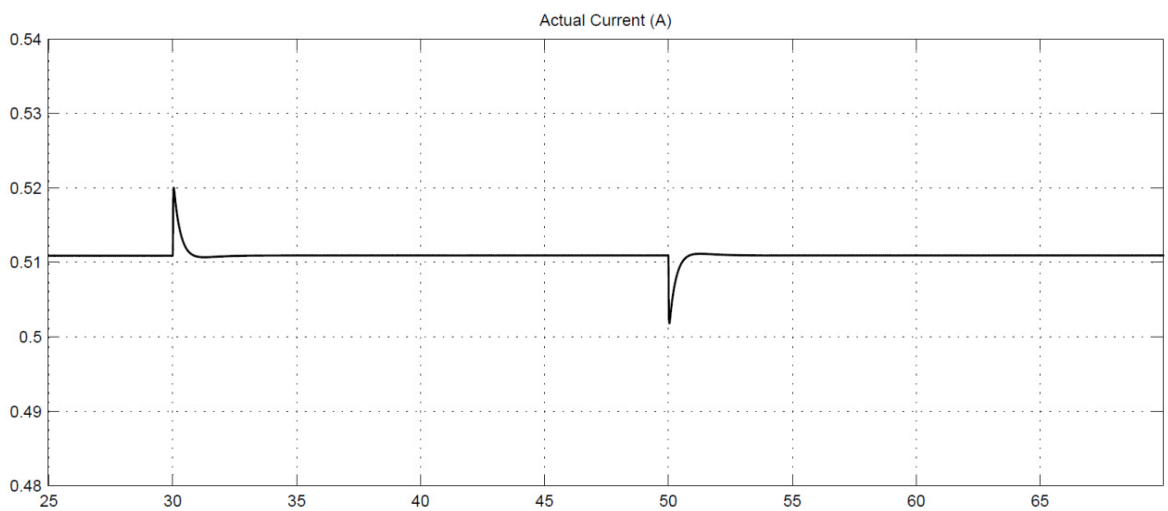

Figure 9a. Motor current, controller model applied to the process model

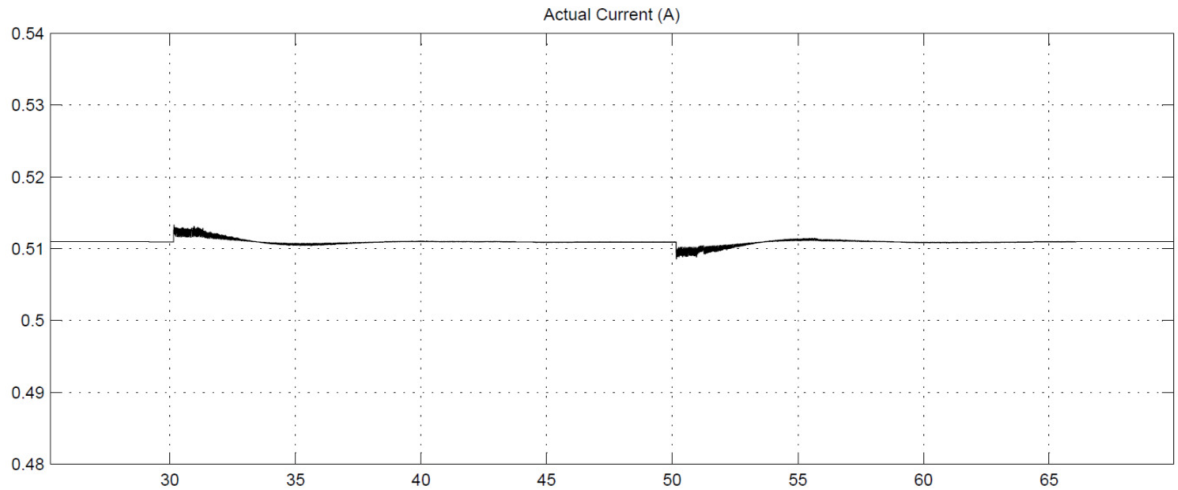

Figure 9b. Motor current, controller implementation applied to the process model

Conclusion

This paper described a problem in the wire harnessing industry and proposed a practical solution based on the current technology in the field of industrial control and automation. A characteristic that distinguishes this project from many other similar works is the development of a PLC-based control system according to the model-based design guidelines. Although model-based control design is a known method, its standards and procedures have not been utilized enough in the PLC-based control projects. The model-based approach used is an elegant way to generate the PLC code automatically and to save time and reduce the error contingency. In this approach, even after implementation of the controller, it is possible to switch back to the simulation phase, modifying the controller parameters, and after observing the simulated outputs, transferring the needed changes into the real controller. The next phase is the integration of the linear positioning hardware with the implemented controller and examination of the system performance.

References

[1] Cable Harness. (2013, November 23). *Wikipedia*. Retrieved from http://en.wikipedia.org/wiki/Cable_harness
[2] Heat-Shrink Tubing. (2013, November 27). *Wikipedia*. Retrieved from http://en.wikipedia.org/wiki/Heat-shrink_tubing
[3] Learning Center. (2002-2014). *CableOrganizer.com*. Retrieved from http://www.cableorganizer.com/learning-center
[4] Simscape. (1994-2014). *MathWorks.*Retrieved from http://www.mathworks.com/products/simscape/
[5] Raimundo, M., & Affonso Guedes, L. (2010, January 1). *Control and Plant Modeling for Manufacturing Systems Using Basic Statecharts*. doi: 10.5772/7192.

[6] PID Controller. (2014, February 7).Wikipedia. Retrieved from http://en.wikipedia.org/wiki/PID_controller
[7] AbdElhamid, A. S. (2012). Cascade Control System of Direct Current Motor. *World Applied Sciences Journal*, *18*(12), 1680-1688.
[8] Persin, S., Tovornik, B., & Muskinja, N. (2003). OPC-Driven Data Exchange between MATLAB and PLC-Controlled System. *International Journal of Engineering Education*, *19*(4), 586–592.
[9] Lieping, Z., Aiqun, Z., & Yunsheng, Z. (2007). On Remote Real-time Communication between MATLAB and PLC Based on OPC Technology. *Proceedings of the 26th Chinese Control Conference*, 545-548.

Biographies

SHAHROKH YOUSEFI-DARANI is a graduate research assistant in the Department of Engineering Technology at the University of North Texas. His research area is mostly focused on industrial control and automation. He defended his thesis in April 2014 and will receive the MS degree in Electrical Systems in August. In May 2014, he accepted an electrical engineering position at Patti Engineering in Austin, Texas.

NIHAL JOSHUA is a graduate research assistant in the Department of Engineering Technology at the University of North Texas, pursuing the MS degree with a major in Mechanical Systems. He expects to graduate in December 2014.

HUSEYIN BOSTANCI joined the University of North Texas in 2012 and is an assistant professor of Mechanical Engineering Technology. He received the MS in Nuclear Engineering from the University of New Mexico in 2001, was a lead R&D engineer at RINI Technologies in Oviedo, FL, from 2001-12, and received the PhD in Mechanical Engineering from the University of Central Florida in 2010. His research interests include thermal management of high power devices, energy efficient building technologies, and R&D on spray cooling. He is a member of ASME, ASEE, and ASHRAE.

ENRIQUE BARBIERI joined the University of North Texas in 2012 and is professor and chair of the Department of Engineering Technology. He received the PhD in Electrical Engineering from Ohio State University in 1988. He was on the faculty of the School of Engineering at Tulane University, New Orleans, LA, (1988-2002) as assistant professor of Electrical Engineering and associate professor of Electrical Engineering & Computer Science. He served as chair of Electrical Engineering & Computer Science during (1996-98) and in 2002, joined the College of Technology at the University of Houston as professor and chair of the Department of Engineering Technology (2002-09). He served as the college associate dean for research and graduate studies (2009-10), as a member of the Executive Council of the Texas Manufacturing Assistance Center (2006-11), chair of the council (2007-09), and director of the Center for Technology Literacy (2006-10). He is a senior member of IEEE—Control Systems Society and a member of ASEE.