Application-Centered Methodology for Teaching Programmable Logic Controllers

Condoor, S.S.
Saint Louis University, St. Louis, Missouri

Abstract

The paper discusses the methodology used for teaching programmable logic controllers – a part of the Mechatronics curriculum. The goal of the curriculum is to nurture skills that can help in implementing a Mechatronics project from the functional specifications. To this end, we developed a teaching paradigm involving several modules each with six identical steps. In this teaching paradigm, students see:

1. tangible results after every module,
2. the applicability of mechatronics concepts in real-world situations,
3. the importance of looking for alternative approaches for solving a given task, and
4. understanding of the product/process design

The paper describes the methodology, presents the applications used (and the relevant hardware and software concepts), shows the gradual increase in complexity, and presents the advantages of the methodology based on student reviews.

Introduction

Mechatronics refers to the synergistic integration of mechanical engineering (structural design and kinematics), electronic control, and system concepts in the design of industrial products and processes [1]. It bridges the existing gaps between mechanical and electrical engineering. It is becoming increasing important with the new smart products which require close integration of both mechanical and electrical engineering concepts. We noticed two opportunities which can be addressed by introducing mechatronics curriculum. The opportunities are:

1. ME students are unsure of handling electrical engineering portion of any project. The deficiencies range from their inability to:
   a. translate wiring diagrams into physical hardware,
   b. pick the right actuator or a control system, and
   c. design and implement a mechatronics system in the capstone design course.

   These observations are consistent with other members of mechatronics teaching community [2]
2. A current competency gap that exists between the teaching and the practice of engineering is industrial controls [3]. As a result, the entry-level engineers require significant amount of training and time before they become productive.

Recognizing that a hands-on exposure to industrial control will provide tremendous benefits to the students, mechatronics curriculum at Saint Louis University was developed geared primarily to real-world applications with an emphasis on system design and implementation aspects.

Curriculum Information

The curriculum involves two courses: Mechatronics, a required course for the ME students, and Advanced Mechatronics, a technical elective. For both courses, students meet for two lectures and one lab each week. The lectures are held in the Mechatronics Projects Laboratory. As a result, students are surrounded by mechatronics equipment. The hardware provokes intellectual curiosity before and after the lectures and and also, handy for illustration. Students typically work in groups of three or four on the project assignments.

The lab equipment includes two dedicated conveyors (built in-house), two MicroLogix controller-based motor control stations, two Mitsubishi robots with Flexmation tables (includes several sensors and actuators, and a small conveyor), variable frequency drives, servo controllers, and several SLC-500s with different modules. The software includes Rockwell software RSView, RSEmulate, RSLogix software for PLCs and Roboware for robot control.

The curriculum trains the students in the design and implementation of mechatronics systems. The course material includes sensors and transducers, actuating devices, digital logic, programmable logic controllers, control strategies, robotics and interfacing. The teaching pedagogy uses kinesthetic/tactile style teaching that uses concrete activities to convey abstract concepts. To encourage kinesthetic learning, the curriculum is delivered in modules. Each module has six identical activities as shown in Fig. 1.

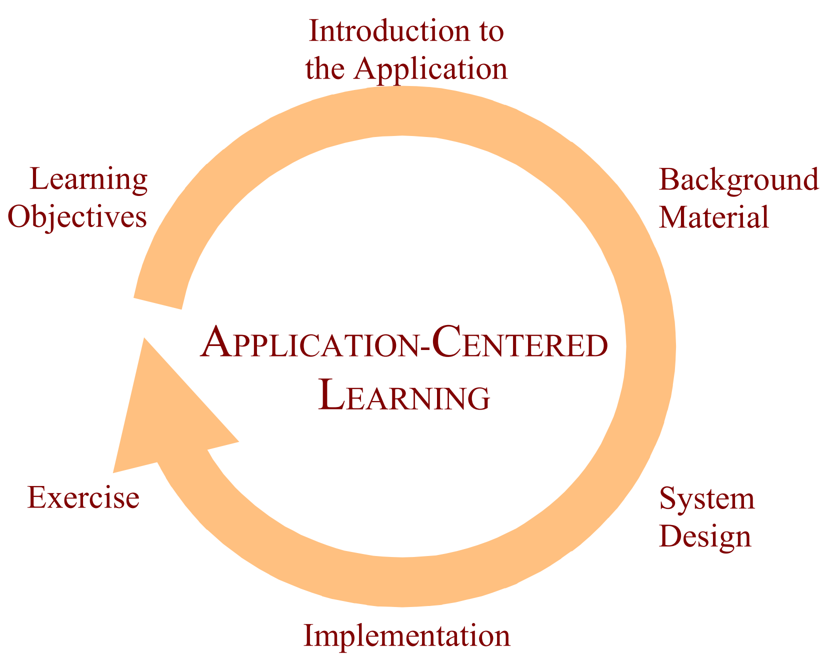

Fig. 1. Application Centered Learning

The steps are:

1. Learning objectives – Students are exposed to the learning objectives of the module.
2. Introduction to an application – A concrete industrial task is presented in terms of functional specifications. As the course progresses, the functional specifications become more open-ended and requires the student to think deeply about the overall system. It promotes diversity of solutions as no two groups design for the same exact set of functional specifications.
3. Background material – The background material is similar to pieces of a jig-saw puzzle. The material covers concepts in both hardware and software that are useful for the completion of the project.
4. System Design – Similar to assembling the puzzle, students synthesize the solution by developing pseudo code and timing diagrams, specifying PLC I/O configuration, creating wiring and ladder diagrams. During the design process, students are forced to look at alternative methods for solving the problem and then, select the appropriate method. Thereby, they learn the best design practices. Also, students are required to develop testing scheme.
5. Implementation – Students learn implement (perform hardware connections, create ladder diagram and download the program) and test the solution.
6. Exercise – Students are required to solve another design task involving similar concepts. The exercises are designed to promote learning of new concepts in addition to the repetitive use of the known concepts (commands and hardware).

The modules used for teaching PLCs are presented in the following table. As discussed earlier, each module introduces and application in terms of functional specifications (Refer second column). Inputs and outputs are minimum required for the application. It is necessary to add more inputs and outputs to perform the given application. The functional specifications become more open-ended as the course progresses. For instance, students must think of fail-safes and necessary sensors after the fourth module. The hardware and software concepts learned in each module are shown in the third and fourth columns.

<table>
<tr><th rowspan="2">Module No</th><th rowspan="2">Application Example</th><th colspan="2">Concepts</th></tr>
<tr><th>Hardware</th><th>Software</th></tr>
<tr><td>1</td><td>Conveyor control
Inputs: Start and stop buttons
Outputs: Conveyor drive motor
Functional Specification:
1. Start button should start the conveyor.
2. Conveyor drive motor should be ON until stop is pressed.</td><td>1. Basic electric wiring.
2. Application of relays to isolate the control circuit from the power circuit.</td><td>1. Bit Instructions:
Examine Open
Examine Closed
Output Energize
Output Latch
Output Unlatch
One-shot rise
2. Communication between PC and PLC
a. Creating and documenting programs.
b. Download and upload programs.
c. Monitor system values.</td></tr>
<tr><td>2</td><td>Conveyor control with a warning light
Inputs: Start, stop and emergency stop buttons
Outputs: Conveyor drive motor, warning light
Functional Specification:
1. Start button should start the warning light.
2. Warning light should flash at about one Hz frequency for ten seconds.
3. After that, the conveyor motor should start.
4. Stop button should stop the conveyor motor. If stop pressed during the initial ten seconds, the warning light should shutoff.
5. Emergency stop button should switch off the system.</td><td>1. Wiring for emergency stop</td><td>1. Timer Instructions:
a. Timer on delay
2. Implicit reset of timers.
3. Timing diagram for timer enable, timing and done bits.
4. Use of system level clock bits to flash at different frequencies.</td></tr>
</table>

Module No	Application Example	Concepts	
		Hardware	Software
3	**Conveyor control with warning and status lights** Inputs: Start, stop and reset buttons Outputs: Conveyor drive motor, warning light, status light Functional Specification: 1. Functional specifications of Module 2. 2. Status light should be on when the conveyor is operating. 3. Status light should switch off ten seconds after the conveyor belt motor switches off. 4. Status light should flash after ten minutes of cumulative motor operation. This simulates the requirement for a periodic maintenance activity for the motor. 5. The reset button should reset the flashing status light and start the cumulative timing again.		1. Timer Instructions: a. Timer off delay b. Retentive timers c. Timer reset
4	**Blockage-sensitive conveyor** Inputs: Start and stop buttons, inlet and outlet sensors Outputs: Warning light, conveyor drive motor Functional Specification: 1. Functional specifications of Module 2. 2. The inlet and outlet sensors monitor the number of packages coming onto and off of the conveyor. If the number of packages on the conveyor exceeds 20 at any time, then assume blockage on the conveyor. Stop the conveyor. Flash the warning light at 0.5 Hz.	1. Wiring for sourcing and sinking inputs and outputs.	1. Counter Instructions a. Count up b. Count down c. Reset counters

Module No	Application Example	Concepts	
		Hardware	Software
5	**Blockage-sensitive conveyor with product workstations** Inputs: Start and stop buttons, inlet and outlet sensors Outputs: Warning light, conveyor, fill station, cap station, label and reject station motors Functional Specifications: 1. Functional specifications of Module 4. 2. The bottle must be rejected at the reject station which is at the end of the conveyor. *The workstations are simulated by colored lights.		1. Bit shift 2. Master control relay
6	**Blockage-sensitive conveyor with product workstations and quality check** Inputs: Same as module 5 Outputs: Same as module 5 Functional Specifications: 1. Functional specifications of Module 5. 2. If a bad bottle enters the system, additional value (filling, labeling and capping) should not be added to it.	This module uses the previous modules and simulates a complete industrial system.	
7	**Stepper motor control** Functional Specifications: 1. Run the stepper motor at desired speed.	1. Difference between unipolar and bipolar stepper motors. 2. Interfacing PLC with a stepper motor driver and a stepper motor.	1. Pulse Train Output 2. Changing the direction 3. High speed counter

Module No	Application Example	Concepts	
		Hardware	Software
8	**Washing machine** Functional Specifications: 1. Control the sequence of operation a washing machine cycle (soak, wash, rinse and dry cycles) [4]		1. Sequencer
9	**DC motor speed control** Functional Specifications: 1. Control the speed of the DC motor.	1. Solid state relays	1. Pulse Width Modulation
10	**Temperature control** Inputs: Thermocouple Outputs: Heater Functional specifications: 1. Control a closed-chamber temperature at the desired level.	1. Analog sensors	1. Scaling the input signal 2. PID Control

Discussion and Results

As discussed, this teaching pedagogy uses kinesthetic/tactile style to nurture mechatronics concepts. Students see tangible results after every module and the applicability of mechatronics concepts in real-world situations. In the process, they also gain abstract and less tangible skills such as the importance of looking for alternative approaches for solving a given task, and understanding of the product/process design.

The response from the students was very positive. In their written evaluation of the course, they found the lab to be very useful. They commented on the applicability of the course in their senior design projects. They were able to successfully implement mechatronics concepts in their capstone design projects. Eighty percent of the students from mechatronics course continued to advanced mechatronics course. Even in the exit interviews, students expressed the opinion that mechatronics curriculum made significant difference in their confidence level.

References

1. Ashley, S. [1997, May] Getting a Hold on Mechatronics, *Mechanical Engineering*, ASME Press.
2. Harrison, O. [2002], Hands-on = Minds-on: Bringing Mechatronics to Life Without Laboratory Time, *Proceeding of 2002 ASEE SE Section Annual Conference*.
3. Southall, S. [2001] An Industrial Controls Course Sequence for Manufacturing Engineering Technology, *Proceeding of 2001 ASEE SE Section Annual Conference*.

4. Webb, J.W. and Reis, R.A. [1999] *Programmable Logic Controllers: Principles and Applications*, Prentice Hall, NY.

Biographical Information

DR. SRIDHAR CONDOOR
Dr. Condoor is an associate professor of Aerospace & Mechanical Engineering at Parks College of Engineering & Aviation, Saint Louis University. His research interests include Design Theory & Methodology, Mechatronics and Consumer Product Design. He is an author of two books and several technical papers. He is a member of ASME and ASEE.