Proceedings of IMECE2006
2006 ASME International Mechanical Engineering Congress and Exposition
November 5 – 10, 2006, Chicago, Illinois, USA

IMECE2006-13761

MECHATRONIC SYSTEM INTEGRATION FOR SENIOR STUDENTS

Dana Kulić and Elizabeth Croft
Department of Mechanical Engineering
University of British Columbia
6250 Applied Science Lane
Vancouver, BC, Canada
ecroft@mech.ubc.ca

ABSTRACT

This paper describes the design and implementation of a senior level course in mechatronic system integration for students completing a mechatronics engineering option in mechanical engineering. The course is designed to give students theoretical and practical experience with a large-scale mechatronic system, and a variety of control, sensing and actuating architectures. The lecture component of the course introduces students to large-scale project integration and interface design, as well as system architecture design. Students learn about alternative control hardware platforms commonly used in industry, such as motion control hardware, field programmable gate arrays and programmable logic controllers. The selection and system integration of various industrial sensors, including vision, are presented. Students also learn about networked control and discrete event control approaches for large-scale industrial systems.

The course contains a significant practical laboratory component. In a series of laboratory sessions, students develop and implement subsystems of a part sorting machine, culminating in the integration and demonstration of an automated, autonomous, sensor driven electro-mechanical system for sorting randomly delivered parts.

The course offers students a theoretical background as well as significant practical experience with large scale mechatronics systems, as would be encountered in industry. This paper describes the lecture and laboratory content, and the experiences from the first offering of the course.

INTRODUCTION

As defined by Centikunt [1], mechatronics "consists of the synergistic integration of three distinct traditional engineering fields for the system level design process. These fields are: (1) Mechanical Engineering, where the word '*mecha*' is taken from, (2) Electrical or electronics engineering, where the part of the word '*tronics*' is taken from, and (3) computer science." The field of mechatronics education has been expanding rapidly over the last twenty years. Over that time, mechatronics education has evolved and expanded across the world [2, 3]. The offered programs have evolved from multi-disciplinary course offerings, through specialized course development to specialized mechatronics curricula development, both in undergraduate and graduate programs [3]. At the University of British Columbia, a mechatronics (electro-mech) program has been offered since 1994 [4]. The program was initially designed as a multi-disciplinary program, where students based in mechanical engineering took additional courses in electrical engineering and computer science. The program has recently been re-designed to allow for expansion, and to include additional new specialized mechatronics courses.

A problem with a purely multi-disciplinary approach, where students take existing courses from the mechatronics sub-disciplines, is that students do not learn about mechatronic systems, i.e., how to integrate the various mechanical, electrical and software components comprising a mechatronic system, and do not learn about system wide design trade-offs and interface design. To address this issue, a series of new courses was developed for the mechatronic program: a third year course on system modeling (MECH 366), and three fourth year courses covering: sensors and actuators (MECH 420), digital control (MECH 467), and mechatronic system instrumentation (MECH 421). This paper describes the mechatronic system instrumentation and integration course (MECH 421). The goal of this course is to give students theoretical and practical experience with large-scale mechatronic system integration, using a variety of control, sensing and actuating architectures that are commonly used in industry. One can note that, without the "groundwork courses" already taken by the students in digital logic design, real time programming, electronic power circuits and electro mechanics as well as the courses listed above, as part of fully integrated mechatronics program [4], the ability to successfully teach such a course would be severely limited.

Related Work

A large number of project based mechatronics courses have been described in the literature [5-11]. Mechatronics courses tend to be project based, with a significant laboratory component, so that students can learn and apply the material through hands-on experience. However, the majority of course development described in the literature is focused on introductory courses, usually offered to second year students, or offered in the senior year as a first course elective [5, 7, 10, 11]. Few papers focus on advanced, senior level mechatronics courses for undergraduate students.

Durfee [8] describes a graduate level mechatronics course, which is also open to senior undergraduate students. The course is project based, requiring students to select a product, device or system that they will design and build during the semester. The project control must be based on an embedded processor. Formal lectures focus on CPU architecture, assembly language programming, digital and analog electronics, sensor and actuator interfacing and real-time programming, but students spend the majority of the course work in the lab, working on the semester long project.

Gardner et al. [9] describe a joint graduate-undergraduate course in mechatronics, designed as a second course in mechatronics. The course lectures focus on microcontroller architecture and programming, and also cover sensors, motion control, serial data communication, discrete digital control and packaging, prototyping and fabrication. Students complete a series of discrete labs covering microprocessor based digital and analogue I/O and motion control. In addition, students work on a class-wide project to build a robotic rover. Students are divided into groups so that each group works on a different subsystem of the robot.

For many of the project based courses, the project domain is research based, for example, in robotics [12, 13]. Few projects are motivated based on industrial or commercial systems. Hannaford and Kuhn [14] describe a novel mechatronics course focused on bridging the gap between engineering education and industrial products. The course is designed around laboratory segments where commercial electronics products such as a portable compact disk player are reverse engineered and analyzed by the students. The course is divided into modules, based on the components of the electronic products. Each cohort of students improves upon the design achieved by the previous year's class. The students are also responsible for delivering text modules, which are then used as instructional material in subsequent years. The course strives to appeal to students across disciplines (mechanical, electrical and software engineering), which are then grouped into multi-disciplinary teams for the laboratory projects.

A large majority of projects in mechatronics courses are single-microprocessor based [15]; few contain other control architectures commonly used in industry, such as Field Programmable Gate Arrays (FPGA), or Programmable Logic Controllers (PLC). In addition, the authors were unable to find published examples of mechatronics course labs or projects that systematically cover networked communication and control, a common feature of industrial mechatronics systems. Of course most electrical engineering departments offer courses on FPGAs, PLCs and networked control, however, these courses are not normally accessible to mechanical engineering or mechatronics students at the undergraduate level. The goal of the mechatronic system integration course at the University of British Columbia was to give students practical experience with various control architectures, modeled on large-scale industrial mechatronics systems.

COURSE DESIGN

MECH 421 was designed as part of the new mechatronics curriculum at the University of British Columbia. It is taken in the final year of a four year Bachelor of Applied Science program. The course is comprised of eight units, covering various mechatronics components of large-scale systems.

The course contains a major lab/project component. As part of the lab, students design and implement a part-sorting system. In each lab, the students work on developing and commissioning a component of the part-sorting system. The labs progressively build in complexity, as students use completed subsystems from earlier sessions to complete later laboratory sessions. At the end of the semester, students integrate and demonstrate the entire system.

To enable students to participate in hands-on large-scale mechatronic system integration in the short time frame available during a single semester, the mechanical components of the system are provided, allowing students to work on sensor selection and placement, control and process algorithm development and integration. In addition, the National Instruments LabVIEW programming environment is used for the majority of the software, to allow students to easily interface to the hardware, and gain experience with a large variety of mechatronic instrumentation, without spending a lot of time on low-level driver and firmware development (covered to some extent in other courses).

A schematic diagram of the part sorting system hardware is shown in Figure 1. Figure 2 shows a picture of the part sorting system.

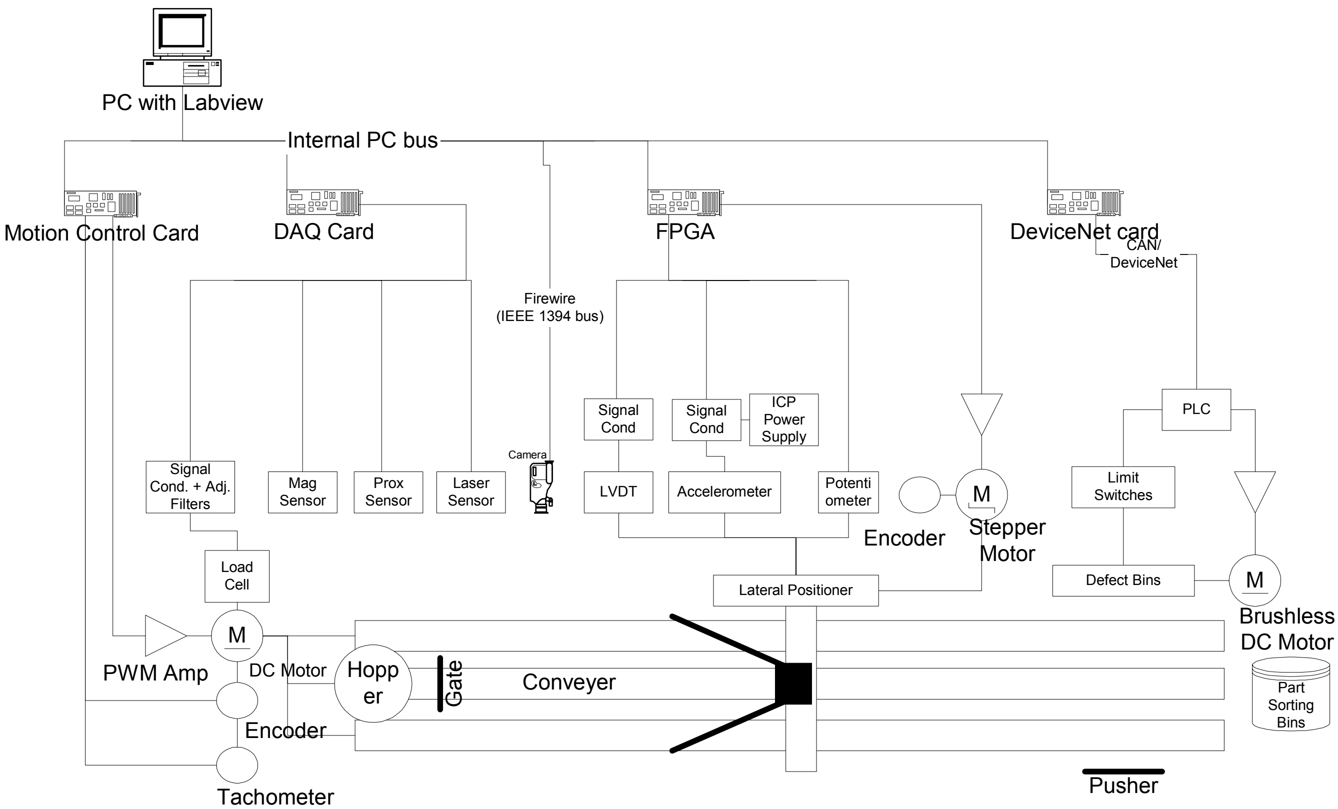

Figure 1. Part Sorting System Overview.

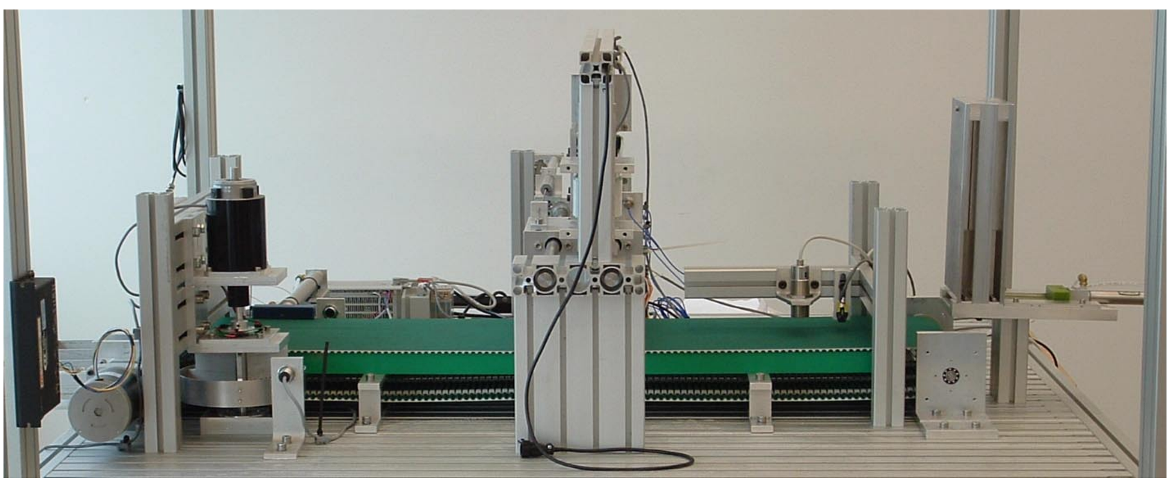

Figure 2. Part Sorting System.

The part sorting system consists of three parallel conveyer tracks actuated by a direct current (DC) motor. The DC motor is controlled through the NI PCI-7342 motion control card, mounted in a personal computer (PC) running LabVIEW. Parts to be sorted are stacked in the hopper (with varying orientations) and released via a gate to the center conveyor, actuated with a pneumatic ram. A magnetic, proximity and laser sensor are available to detect the presence and type of part on the conveyer. These sensors can be read from the NI PCI-6221 data acquisition card mounted inside the PC.A color CCD camera mounted above the conveyers is also available for part classification. The camera is connected to the PC via the Firewire (IEEE-1394) bus.

A lateral positioning system moves the part from the center conveyor to the left or right conveyor. The lateral positioner is actuated by a stepper motor. The stepper motor is controlled through an NI PCI-7831R FPGA card, mounted inside the PC. A reject pusher, actuated by a pneumatic ram, is available to remove defective parts from the conveyer. Defective parts are pushed into one of 3 defect bins, which are rotated into place via a brushless DC motor. The brushless DC motor, the reject pusher and the hopper gate are controlled through an Allen Bradley Micrologix 1000 PLC. The PLC is connected to the PC through a DeviceNET network.

Sample parts (right angle brackets) to be sorted by the system are shown in Figure 3. There are three types of parts: a thick aluminum part, a thin aluminum part, and a thin Delrin™ acetal polymer part. The parts are 57mm square, have 5mm holes, and are either 6.3 or 9.5 mm think. These three types of parts must be separated onto their respective conveyers by the lateral positioner. In addition, parts may be defective. The middle and right side parts in Figure 3 are correct, while the part on the left hand side is defective due to incorrect hole placement. Defective parts may have holes missing, extra holes, or holes incorrectly placed. The system must recognize the defective parts, and remove them from the conveyer using the reject pusher. Rejected parts are also sorted by material and thickness, by rotating the correct bin of the reject tray into place before activating the reject pusher.

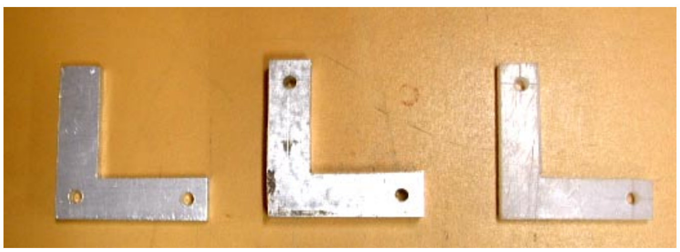

Figure 3. Sample Parts to be Sorted.

The students' objective for the design of the part sorting system is to maximize part throughput while minimizing the classification error.

There is a tight coupling between each lecture unit and laboratory components. Students learn the relevant material in the lecture, and immediately apply the knowledge in the associated laboratory/project component. In addition, the course is taught in parallel with MECH 420 (Sensors and Actuators), and there is some coupling between the material and laboratory exercises of MECH 420 and MECH 421. For example, in the MECH 420 lab, students will analyze the behavior and characteristics of the various position sensors; the same sensors are then used by the students to implement the part sorting system. Table 1 summarizes the 8 units covered in the course and the laboratory activities supporting these units.

Course Organization

This course was offered for the first time in the fall semester of 2005/06. The hardware for the part sorting system was designed and developed prior to the start of the course, and a sample implementation of the entire system developed by two masters level students, as a project for their M. Eng. degree.

The course consisted of 2 hours of lectures a week, and 2 hours of formal laboratory time, which alternated on a two week schedule with the accompanying MECH 420 labs. There was no required textbook for the course; course notes summarizing the lecture material and lab manuals were distributed through the course website [16].

Students also had access to the lab for an additional 6 hours a week as required. This time was especially important in the later parts of the course, during the more complicated labs and the final integration, when students were rarely able to complete the entire lab within the formally scheduled time. A teaching assistant was present during the scheduled lab sessions, and during the additional time to offer guidance with hardware issues and debugging. In addition, students had open access to a computer lab containing all the software used in the course. This lab was used to allow the students to complete their pre-lab work, and develop algorithm software in LabVIEW or RSLogix prior to coming to the lab.

Table 1. Course Units and Associated Laboratory Activities.

Unit #	Unit	Laboratory Activity
1	Introduction	Part Detection and Sorting
2	Motion Control	Conveyer Belt Control
3	Machine Vision	Vision Based Part Sorting
4	FPGA	Stepper Motor Control
5	Networked Communications	DeviceNET Communications
6	PLCs	Reject Tray Control
7	Discrete Systems	System Integration
8	Fault Finding	System Integration

Unit 1: Introduction to Mechatronic Systems

Lecture Content

In the introductory unit, students are introduced to interface design, the importance of interface design for larger systems, and the characteristics of a well designed interface. Practical examples using the LabVIEW programming environment are presented. Alternative system architecture configurations are introduced, and their advantages and disadvantages overviewed.

Laboratory Content

In the first laboratory session, students develop the part detection and sorting subsystem using proximity and distance sensors. The objective is to gain experience with sensor selection and integration. Students are given a choice of sensors that can be connected to the data acquisition card. Students are already familiar with the characteristics and response behavior of the available sensors from a previous laboratory exercise in the companion MECH 420 course.

Students must also select where the sensors will be placed along a mounting bar above the central conveyer belt immediately in front of the hopper gate.

For this laboratory exercise, students must detect when a part has been released onto the conveyer and classify the correct part type (thin aluminum, thick aluminum or Delrin), for any planar orientation of the part. Defective part classification is deferred to a later session. Students must develop an algorithm for detecting and classifying the parts using their selected sensors, and implement the algorithm in LabVIEW. They test and debug their algorithm on the part sorting hardware, both with a stationary conveyer, and with the conveyer moving at various speeds.

Unit 2: Motion Control

Lecture Content

This unit includes a brief review of proportional-integral-derivative (PID) control, focusing on implementation issues such as integrator windup and dealing with high frequency noise. Single axis trajectory planning is introduced, and various position, velocity and acceleration profiles are outlined. Students also complete a small assignment, developing software for single axis trajectory generation. [1]

Laboratory Content

In this laboratory, students develop the trajectory generation and control module for the DC motor running the conveyer system. The objective is to gain experience operating and controlling a simple DC motor control system. Students are already familiar with DC motor performance characteristics from a preceding MECH 420 laboratory exercise.

Students develop LabVIEW software for position and velocity control of the DC motor using motion control hardware. Students also develop a plan for tuning the PID gains of the controller. Students then debug their software and tune the PID controllers using the part sorting system hardware. Students also observe various trajectory generation algorithms and characterize their performance.

Unit 3: Machine Vision

Lecture Content

In this unit, the basic algorithms for industrial machine vision applications are introduced. The image acquisition hardware and processes are described, for charged coupled device (CCD) and complementary metal oxide semiconductor (CMOS) cameras. Common image processing algorithms are overviewed, including histogram equalization and smoothing, Gaussian, gradient and Laplacian filtering. The theoretical foundations of filtering theory are briefly introduced. Image analysis techniques such as segmentation, binary morphology and blob analysis are described. Students are also provided with implementation and output examples using the LabVIEW IMAQ toolbox.

Laboratory Content

In the vision lab, students use the Firewire camera to determine whether parts on the conveyer are defective. The objective is to gain experience using and developing simple vision algorithms for industrial automation. Students must develop an algorithm for detecting and classifying parts using machine vision. Students must detect the exact location of the part, detect the location of the holes within the part, and analyze the number and relative placement of the holes to determine if the part is defective. Students use the LabVIEW IMAQ toolbox to implement their algorithm.

Students then debug and test their algorithm using the sorting system hardware. Students must first demonstrate that their algorithm works correctly when the parts are stationary. Students then use the conveyer control software developed in the previous unit to move the conveyer at various velocities, and test that their vision system operates correctly when the parts are moving. Students also analyze the performance of their algorithm in terms of execution time and robustness to motion blur.

Unit 4: Field Programmable Gate Arrays

Lecture Content

In this unit, Field Programmable Gate Array (FPGA) hardware is introduced. A brief review of finite state machines is provided, as well as basic logic hardware such as latches and flip-flops. FPGA configurations and systems from several manufacturers are overviewed. The motivation for using FPGAs is discussed. The software development process when using FPGA hardware is described. Guidelines for programming an FPGA are provided, such as taking advantage of parallel execution, handling shared resources, integer math and timing functions. Code examples are also reviewed using the NI FPGA programming palette.

Laboratory Content

In the FPGA laboratory, students develop the lateral positioner subsystem for moving the classified parts onto the correct conveyer track. The objective of the laboratory is for students to gain experience using an FPGA controller (to operate the subsystem) and a stepper motor for actuation.

The lateral positioner is a linear slide driven by a timing belt and a stepper motor. The slide is positioned in one of the three locations depending on which type of part is released from the hopper. Students have an LVDT, rotary potentiometer, accelerometer, and DC tachometer available for use in the control of the lateral positioner. Figure 4 shows a close up image of the lateral positioner. Students gain familiarity with these sensors and stepper motor behavior in a preceding MECH 420 laboratory.

[1] A further course on computer control and trajectory generation for mechatronic systems is given simultaneously in MECH 467 following the introduction to control given in Mech 366.

Figure 4. Lateral Positioner.

For this laboratory, students must select the sensors to be used and develop an algorithm for controlling the lateral positioner. They must implement their algorithm using the NI FPGA software and hardware. The must develop a test plan for verifying proper access to all the hardware used in their implementation. After verifying that all the hardware is connected and communicating correctly with the PC, students debug and test their algorithm. They next test their system by using the conveyer subsystem developed in Lab 2 to advance the parts down the conveyer, and manually trigger the lateral positioner to move the parts to the correct track.

Unit 5: Networked Communications

Lecture Content

In this unit, distributed and networked mechatronic systems are introduced. Parallel and serial data transmission and modes are described. Various network topologies and network access control protocols are described and compared. Students are also assigned additional readings to compare the performance of three sample control networks: the Ethernet, ControlNet and DeviceNet [17].

In the second part of the unit, the controller area network (CAN) and DeviceNet protocols are presented in more detail. The DeviceNet application layer is described, and the devices and communication types outlined. An example DeviceNet setup is presented, and sample code for configuring DeviceNet in LabVIEW is provided.

Laboratory Content

In the network lab, students develop the interface for communicating between the LabVIEW PC and the PLC, using a simple DeviceNET network. The PLC controls the reject removal system. The network layout is shown in Figure 5. The objective of the laboratory is gain experience with an industrial communication network, operating with different controllers. This lab is combined with the PLC lab described in the next section.

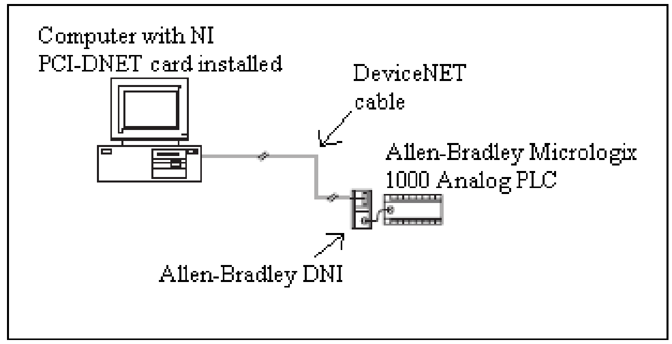

Figure 5. DeviceNET Network.

In the communication part of the lab, students configure the DeviceNET Interface (DNI) adapter that allows the PLC to communicate through the DeviceNET network. They next configure the DeviceNET card running from the PC, and develop and test a LabVIEW interface program to send and receive data from the PLC. As part of this process, they must design and implement the message structure and format, and implement a handshaking procedure. During the lab, students debug their configuration and interface, and demonstrate that they can receive and send messages between the PC and the PLC, using the DeviceNET interface.

Unit 6: Programmable Logic Controllers

Lecture Content

In this unit, students are introduced to PLCs. The hardware components of a typical PLC are described, and the reasons for using PLCs outlined. Students are introduced to ladder logic programming constructs and structure, as well as multi-bit data handling methods.

Laboratory Content

In the PLC lab, students develop the reject removal subsystem, to remove defective parts from the conveyer. They develop a PLC program to control the brushless DC motor actuating the reject tray, and the reject pusher. The objective of this laboratory is to allow students to gain experience with PLC programming and interfacing.

There is a separate bin for each type of defective part. The three bins are on a rotating tray controlled by the brushless DC motor. The reject tray must be rotated so that the correct bin is facing the reject pusher when the part passes the tray. There are 3 infrared limit switches that are used to position the tray correctly. There is also a through-beam sensor immediately in front of the pneumatic ram to detect when a part is on the conveyer. An overhead view of the reject removal subsystem is shown in Figure 6.

Figure 6. Reject Removal Subsystem.

Students must develop and implement an algorithm for correctly positioning the reject tray and activating the reject pusher. Students implement their program using the Allen Bradley RSLogix 500 programming environment, which provides a ladder-logic programming interface. Students download and debug their program on the PLC. Once the reject subsystem is working, students integrate the subsystem with the communication interface. Students must demonstrate that the reject removal subsystem can be triggered and operates correctly via commands from the PC.

Unit 7: Discrete Systems

Lecture Content

Unit 7 introduces students to the modeling and control of discrete event systems. Two methods are covered: Ramadge-Wonham automata theory and Petri nets. The graphical modeling of discrete event systems using these two methods is overviewed, and the construction of supervisory controllers for discrete event systems is developed. The analysis of the properties of discrete event system is introduced. Students are also asked to complete additional readings on the topic from the literature [18-20].

There is no laboratory content for this unit.

Unit 8: Fault Finding

Lecture Content

In this unit, formal methods for debugging and faultfinding are introduced. Various techniques for automated fault detection are introduced, such as replication checks, timing checks, watchdog timers, parity and error coding, and diagnostics. Fault analysis methods such as Fault Tree Analysis (FTA) and Failure Modes and Effects Analysis (FMEA) are overviewed.

Laboratory Content

In the final two weeks of the course, students integrate all the previously developed subsystems of the part sorting system, incorporating part feeding, part detection, part classification, part positioning and defective part rejection. The objective of the final lab exercise is to allow students to gain experience integrating mechatronic subsystems and performing system level debugging and optimization.

Students are asked to design a top-level algorithm for sequencing and interfacing the various subsystems. This also may involve modifying some of the subsystem interfaces, allowing students to gain appreciation for good interface design. Students must also develop a test plan for bringing the subsystems on-line one at a time, and verifying that the subsystems are working correctly within the overall system. Students also develop all test code required for implementing their test plan.

In the lab, students execute their test plan to integrate and debug the complete system. Once the system is working, students work to optimize the performance of the system, by increasing the sorting speeds and analyzing and eliminating sorting errors.

Student Evaluation

The emphasis of the entire MECH 421 course was on the practical application of the material in the project/lab. To underscore this emphasis, 60% of the final grade was assigned to the project, and 40% to the final exam.

The project grade was divided between the pre-labs (20%), in lab demonstrations (10%), lab reports (20%), final system performance (30%), and the final report (20%). The high percentage assigned to the prelab reflects the fact that students do a significant amount of preparatory work in the pre-lab, to ensure that they maximize their use of the hardware while in the lab. For each lab, students are expected to develop their algorithms and write and successfully compile the required code prior to coming to the lab, so that they are ready to immediately start debugging in the lab. At the end of each lab, students must demonstrate to the TA that the subsystem is operating correctly to receive the in lab demonstration marks.

The final system performance was graded based on system speed and classification accuracy. Each student team was given 3 trial runs, and the average across the three runs was taken as the final score.

EXPERIENCES FROM THE FIRST OFFERING OF THE COURSE

Despite the challenging complexity level of the course project, all the student groups were able to successfully implement and integrate all the subsystems, and demonstrate the correct operation of their final systems. The grading metric was used was a weighted function of the time taken to complete a part sorting run and the number of classification errors. Students were extremely enthusiastic about the lab project, and spent considerable time optimizing the performance of their systems. The student teams developed innovative strategies for overcoming some of the hardware limitations of the system to further improve sorting performance. Stills extracted from videos of the final student demonstrations are shown in Figure 7 and Figure 8.

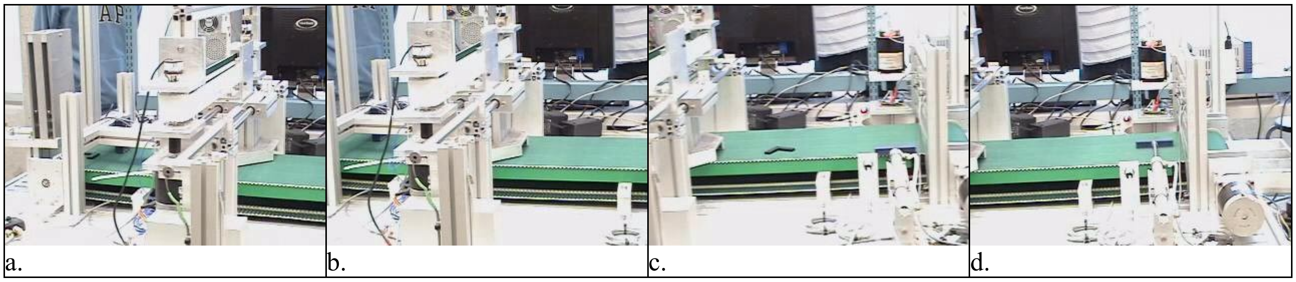

Figure 7 - Video Frames from the Final System Demonstration

Figure 8 - Students Configuring the System prior to the Final System Demonstration

In frame (a) of Figure 7, a part is seen leaving the parts hopper. In frame b, the lateral positioner is activated to move the part to the correct conveyor. In frame c, the part is advancing along the conveyer, and in frame d, the part has been identified as a reject and is being removed by the reject ram.

When evaluating the course, all the students assessed the course as highly relevant to their professional needs, and found the course material interesting. In particular, a number of students commented positively on the usefulness and relevance of the labs.

These results indicate that students recognize and value the importance of gaining experience with larger scale systems and mechatronic system integration.

CONCLUSIONS

A novel course for senior mechatronics students was developed as part of the mechatronics curriculum at the University of British Columbia. The aim of the course is to introduce students to a variety of sensors, actuators and control architectures commonly encountered in industry, but not usually covered in a traditional curriculum, and to give students some experience with building, integrating and debugging a large scale mechatronic system. To achieve these goals, the course contained a significant project, modeled on an exemplar industrial part sorting system. The course was structured as a series of units, each built around a subsystem of the part sorting system. In each unit, students heard lecture content outlining the theoretical background relevant to each subsystem, and immediately applied the knowledge to implement and debug the associated subsystem. Students also gained experience with system-wide integration and debugging by implementing the complete system by the end of the course.

The positive response of the students during the first offering of the course confirms the value of teaching industrially relevant integration skills on a fairly complex platform to mechatronics students.

ACKNOWLEDGMENTS

The authors would like to acknowledge Mark Leusink, David Lonnenberg and Gord Wright for their hard work in building and running the lab hardware, and providing valuable assistance to the students. The assistance of Professors Boris Stoeber and Xiaodong Lu in the later curriculum development is also greatly appreciated.

REFERENCES

[1] S. Centikunt, *Mechatronics*: John Wiley and Sons, 2007.

[2] M. Acar, "Mechatronics Challenge for the Higher Education World," *IEEE Transactions on Components, Packaging, and Manufacturing Technology - Part C*, vol. 20, no. 1, pp. 14 - 20, 1997.

[3] M. Grimheden and M. Hanson, "Mechatronics - the evolution of an academic discipline in engineering education," *Mechatronics*, vol. 15, pp. 179 - 192, 2005.

[4] Y. Altintas and E. A. Croft, "Electro-mechanical design engineering: a progress report and future directions for mechatronics education," *International Journal of*

Mechanical Engineering Education, vol. 30, no. 4, pp. 325 - 339, 2002.

[5] D. G. Alciatore and M. B. Histand, "Mechatronics at Colorado State University," *Mechatronics*, vol. 5, no. 7, pp. 799 - 810, 1995.

[6] J. E. Carryer, "The Design of Laboratory Experiments and Projects for Mechatronics Courses," *Mechatronics*, vol. 5, no. 7, pp. 787 - 797, 1995.

[7] J. E. Carryer, "Undergraduate Mechatronics at Stanford University," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 585 - 591, 1999.

[8] W. K. Durfee, "Designing smart machines: teaching mechatronics to mechanical engineers through a project-based, creative design course," *Mechatronics*, vol. 5, no. 7, pp. 775 - 785, 1995.

[9] J. F. Gardner, J. S. Lamancusa, and H. J. Sommer, "Mechatronics II: Advanced Mechatronics for Mechanical Engineering Students," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 472 - 476, 1999.

[10] J. B. Hargrove, "Curriculum, equipment and student project outcomes for mechatronics education in the core mechanical engineering program at Kettering University," *Mechatronics*, vol. 12, pp. 343 - 356, 2002.

[11] A. Ranaweera, B. Bamieh, and V. Parmenter, "Sensors, actuators and computer interfacing laboratory course at the University of California at Santa Barbara," *Mechatronics*, vol. 15, pp. 639 - 650, 2005.

[12] N. J. Brown and O. T. Brown, "Mechatronics "a graduate perspective"," *Mechatronics*, vol. 12, pp. 159 - 167, 2002.

[13] M. A. Minor and S. Meek, "Integrated and structured project environment in mechatronics education," in *ASEE Anual Conference and Exposition*, pp. 11819 - 11834, 2002.

[14] B. Hannaford and K. Kuhn, "A "hands-on" course in consumer electronics design," *Mechatronics*, vol. 5, no. 7, pp. 753 - 762, 1995.

[15] V. Giurgiutiu, J. Lyons, D. Rocheleau, and W. Liu, "Mechatronics/mictrocontroller education for mechanical engineering students at the University of South Carolina," *Mechatronics*, vol. 15, pp. 1025 - 1036, 2005.

[16] D. Kulić, Online: www.mech.ubc.ca/~mech421

[17] F. L. Lian, J. R. Moyne, and D. M. Tilbury, "Performance Evaluation of Control Networks: Ethernet, ControlNet and DeviceNet," *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 66 - 83, 2001.

[18] T. Murata, "Petri Nets: Properties, Analysis and Applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541 - 580, 1989.

[19] J. L. Peterson, "Petri Nets," *Computing Surveys of the ACM*, vol. 9, no. 3, pp. 223 - 252, 1977.

[20] P. J. G. Ramadge and W. M. Wonham, "The Control of Discrete Event Systems," *Proceedings of the IEEE*, vol. 77, no. 1, pp. 81 - 98, 1989.

