A Multidisciplinary Framework for using emerging Computing Systems in Engineering Education

Joseph Callenes, Tan Run En, Aria Pegah
California Polytechnic State University, San Luis Obispo, CA
{jcallene, erunen, apegah}@calpoly.edu

***Abstract*—This Innovative Practice Full Paper presents a multidisciplinary framework for using emerging computing systems in engineering education. The computing landscape is in the midst of significant changes. Hardware and software systems are becoming highly data intensive, specialized, parallel, and heterogeneous. With these developments, new applications across a wide range of disciplines are being enabled, from agriculture, mechanical and control systems, and many other physical sciences. For this reason, new approaches for teaching, via hands-on experiences, with these systems is needed. In this paper, we propose a novel framework that can be used to support learning objectives for these systems, at the intersection of multiple disciplines (e.g. computer science, data science, mechanical/aerospace/industrial/computer engineering). Specifically, we describe a flexible computing platform, that supports the exploration of custom hardware and software, interfaces, communications, and integrated system design. The RISC-V based system-on-chip architecture and supporting software and programming tools provide a unique opportunity for students to quickly learn advanced computing concepts critical to future computing systems. An implementation of the framework is described in the context of an important class of control applications, system identification and motion control. As these systems scale, analyzing non-linear system dynamics requires non-traditional approaches, e.g. using modern learning-based models and tools. The proposed approach allows for students to explore the impact of computing abstractions, data-intensive system design, and the potential for cross layer optimizations, while also learning key concepts related to parallel programming models and architectures. Based on this case study, we also provide an analysis and discussion of potential curricular activities that could leverage the framework. A key obstacle for such engineering education research, is the lack of familiarity that students in different disciplines have for the other fields of study, making a complete understanding of overall system design principles difficult. This paper presents a framework for multidisciplinary synergy that alleviates this problem and elevates computing systems to a potentially critical position in engineering education.**

***Index Terms*—computing education, multidisciplinary engineering, cross-layer design**

I. Introduction

Computer systems are increasingly weaved into the fabric of everyday life until they are indistinguishable from it. Past advances in semiconductor technology have provided economical, abundant, and reliable computing resources, which have enabled countless breakthroughs in science, medicine, and agriculture, and improved the lives of so many. Due to physical limitations, the computing landscape is in the midst of significant changes. Computing systems are becoming becoming highly data intensive, specialized, parallel, and heterogeneous. Hardware examples include energy-efficient parallel compute fabrics, specialized SoC architectures, and mixed-signal designs [1]. At the same time, new applications across a variety of disciplines are pushing the speed, size, and energy of these systems to physical barriers. Autonomous, robotic, and control systems scale in computational intelligence, numbers of sensors/actuators, and size [2].

Engineers across multiple disciplines will need to design and build systems that balance constraints (e.g. performance, energy, safety, security) across application and hardware layers. Computing and engineering education needs techniques and tools that provides students across multiple disciplines the opportunity to investigate cross-layer design, understand trade-offs between subsystems, and efficiently design/utilize abstractions. In this paper we describe a multidisciplinary framework that can support learning objectives for designing and utilizing these systems. Specifically, a flexible computing platform is described, that can be leveraged for a variety of custom computing models, interfaces, communications, and and integrated designs. A RISC-V-based system-on-chip (SoC) architecture and programming tools provide students an opportunity to quickly explore computing design concepts that are critical for future computing systems.

The contributions of the paper are:

- We identify a need for hands-on teaching techniques and tools that utilize emerging computing fabrics, which are highly data intensive, specialized, parallel, and heterogeneous.
- We propose a framework that supports learning objectives at the intersection of multiple disciplines for these systems, including the ability to investigate cross-layer designs, understand trade-offs between different subsystems, and efficiently design/utilize abstractions.
- We describe an implementation of the framework in the context motion control and system identification and an analysis of results.

The rest of the paper is organized as follows. Section II describes related work, including computer architecture and control learning tools and labs. We introduce our multidiscplinary framework in Section III. In Section IV we describe a case study of using the approach in the context of

motion control and system identification applications. Section V concludes.

II. Related Work

Much recent work has investigated platforms and tools, including remote techniques for teaching students about modern computer organizations, architectures, and compilation techniques [3]. Architectural simulators are commonly used for students to explore and obtain a deeper understanding of the underlying hardware operation [4]–[6]. Some research has also begun to consider emerging computing trends, high performance computing systems [7]. cloud-based systems [8], and analog-based hardware [9]. This paper also proposes an learning architecture and associated set of tools, however the architecture incorporates more recent architectural innovations and is designed with flexibility, cross-layer design, and accessibility for students from multiple disciplines.

There also exists much recent work on platforms and tools that emphasis the teaching of programming and computational thinking [10]. Prior work has also investigated the use of robotics as central application for introducing a variety of computing concepts and at varying levels of education [11], [12]. These approaches emphasis skills such as abstraction, decomposition, algorithms, data representation, etc. Much recent work also exists platforms and labs designed for control oriented disciplines [2], [13]–[15]. The framework described in this paper also supports teaching similar core computational skills, however it also integrates hardware/system/application aspects in order to 1) enable cross-layer design skills and 2) bridge the gap between disciplines allowing students to obtain a deeper understanding of overall system design.

III. Learning Architecture

A key obstacle for such engineering education research, is the lack of familiarity that students in different disciplines have for the other fields of study, making a complete understanding of overall system design principles difficult. This paper presents a framework for multidisciplinary synergy that alleviates this problem and elevates computing systems to a potentially critical position in engineering education.

Figure 1 shows an overview of the approach for the framework. Pre-built hardware modules and interfaces are included with the hardware architecture in order for students approaching from an application perspective to have the ability to quickly and efficiently start investigating the impact of design decisions across boundaries. Similarly, kernels and trained models representative of important application classes, e.g. Kalman filter, object recognition, etc., are included in order for students approaching from a hardware perspective to quickly start investigating system design. There exists a growing ecosystem of open-source hardware tools (e.g. RISC-V), in addition to corresponding software tools [16]. These tools are leveraged for the general purpose portion of the framework. RISCV is supported by a growing shared software ecosystem and has both proprietary and open-source core implementations. It is also appropriate for all levels of

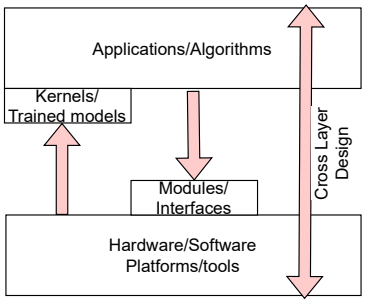

Fig. 1. Overview of System Approach. Pre-built hardware modules, kernels,and trained modules, allow for students to explore full system integration. The approach also allows for investigation of system design across abstraction layers.

computing systems, from microcontrollers to supercomputers (32, 64, and 128 bit variants, and a 16-bit compressed format).

Figure 2 shows an overview of our proposed learning architecture. On the left, is the high level system, representative of modern control and embedded systems. These systems increasingly are incorporating more sensors and actuators with a variety of interfaces and protocols. Additionally, these systems are increasingly networked, i.e. organized into distributed control systems, in order to scale and meet the performance requirements of systems that pushing to physical barriers. At the core of these system we propose a learning architecture, on the right, which includes general purpose and specialized prebuilt acceleration logic. This framework allows students to explore tradeoffs with energy, performance, safety, and security. A flexible model/inference accelerator is shown on the right and provides an efficient fabric for mapping emerging data data-intensive models, e.g. deep-learning models. Additional support modules, supporting compression, event triggering, model control are also provided with the accelerator. A control hub module for providing low-level hardware support for distributed control primitives. Finally, these modules also a provide a template for future student projects that seek to add new extensions or accelerators to the system.

IV. Case Study

We present results of using the framework in the context of undergraduate research activities focused on learning the properties and dynamics of a steel beam using an interferometer and custom designed system on chip. Piezo actuators attached to the steel beam are used to create a moment upon an applied voltage. This project merges the disciplines of mechanical engineering with electrical and computer engineering. Various closed-loop control algorithms of the beam using a real-time processor were used, in order to expose mechanical engineering students to the increasing computational and modeling capabilities of emerging systems. The project also exposed electrical and computer engineering students to the application of using FPGA and SoCs for real-world system-oriented problems.

The system for this case study focused on a flexible beam model as seen in Figure 3. This is well studied physical

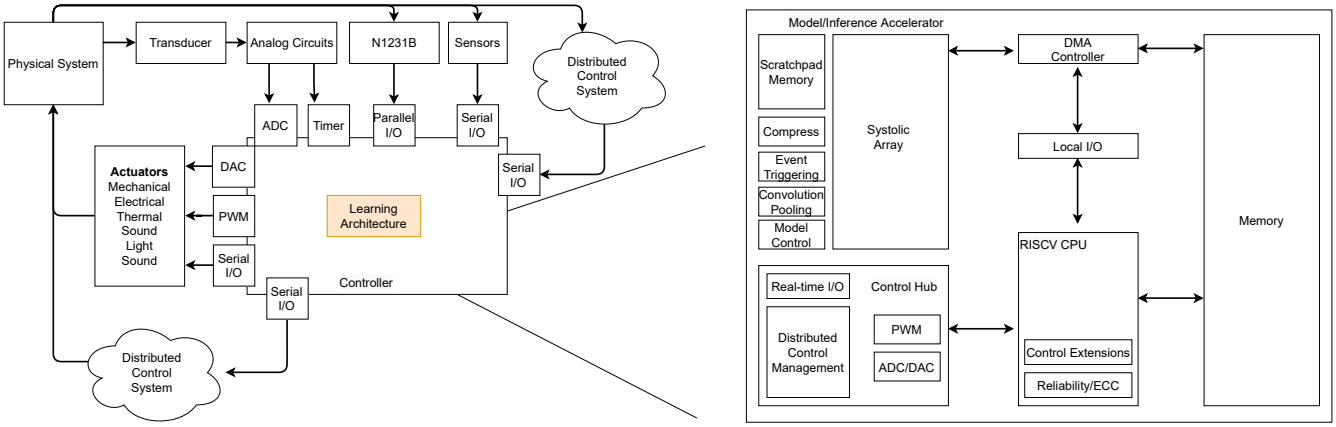

Fig. 2. Learning Architecture Overview. Distributed control system with multiple interfaces (right) and learning hardware architecture with flexible accelerator design (right).

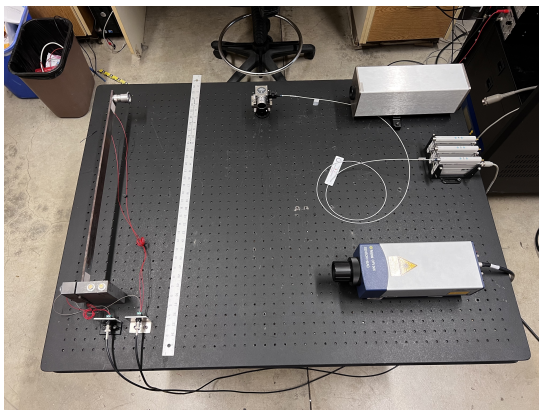

Fig. 3. Flexible beam model. Used for studying systems focused on system identification and motion control applications.

structure that can be used to emulate a variety of other physical problems. The applications that we focused on for this study include system identification and motion control. System identification is the process of determining physical parameters governing the dynamics of the system. System identification can be used for building more accurate models and also improving the accuracy of related control systems. Motion control is an important class of application related to moving machines/structures in controlled manner.

A. Sensors and Actuators

For this case study we used a laser interferometer for measuring the position of the beam. The laser interferometer system was mounted on an optics table alongside the metallic beam under test, it produces a laser with a specified wavelength which is reflected off a retroreflector and back to the laser subsystem. The retroreflector is a mirror magnetically attached to the beam. The measured distance can then be extracted via the phase-shift between the transmitted and received electromagnetic wave.

A circuit board with PCI traces connects via a PCI interface to a host computer and is used to interface with the laser interferometer. The connection between control board and the interferometer are two coaxial cables, which sends the reference and sensor waveform to the board. The board computes and translates the distance into digital form.

In addition to the interferometer measurement system, we also included a set of Piezo actuators. The piezo patches are capacitor-like ceramic plates glued to The metal beam. Upon an applied voltage across the patches, a torque is created on the metal beam, causing it to bend. The direction depends on the applied voltage's polarity. This disturbance caused by the piezo patches can then by measured by the phase shift of the laser light within the retroreflector and is converted to an analog signal via E1708A dynamic receiver. The analog signal, along with a reference signal is then sent to the laser measurement board where their relative phase are compared and the position is extracted and converted into a 32-bit digital signal.

For this case study the group considered different options for the computing subsystem, interfaces, and for integrating the sensors and actuators. By integrating the sensors and actuators a motion control system can be implemented.

B. Controller and SoC Boards

The learning architecture was deployed to a FPGA, and used as the central focus for the controller subsystem. Figure 4 shows the FPGA board, with connection to the interformeter board (32-bit parallel interface). The FPGA, a Digilent Genesys Zu SoC board, combines programmable logic FPGA with dedicated I/O, a general purpose and real-time ARM core, and a GPU on the same chip. As part of this case study, electrical and computer engineering students explored tradeoffs related to a varying architectural, operating system, and interface support. As part of the cross-layer system design the group also integrated an external controller device [17], which provides a simple interface for mechanical engineering students to directly export the matlab/simulink models to run

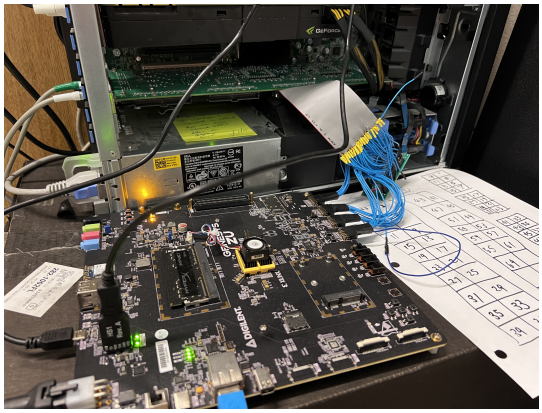

Fig. 4. FPGA and interferometer sensor board. Used for implementing the flexible learning architecture.

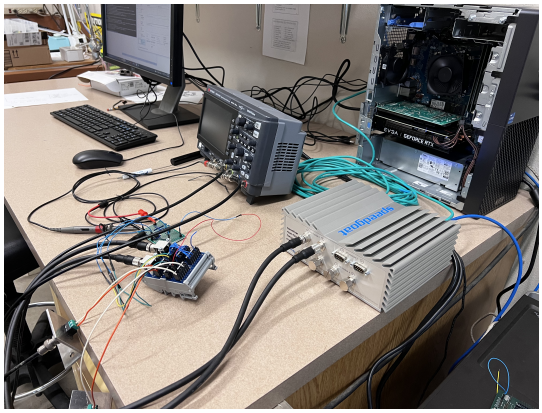

Fig. 5. Actuator amplifiers and Speedgoat controller.

on the physical system [18]. The external device is preloaded with a real-time operating system and can be controlled by a computer with Matlab and Simulink installed. This tool-flow allowed for the Mechanical Engineering students to quickly prototype and experiment with closed-loop control system and custom mathematical models. Both the external controller and learning architecture on the FPGA are commanded via a desktop computer running a regular Windows 10 operating system. Figure 5 shows a picture of the lab setup for the external controller connected to host PC, learning architecture, and the piezo actuators.

C. Communication Interfaces

As discussed in Section III, as systems scale, they increasingly incorporate a variety of devices, that may also use a variety of interfaces, including networking interfaces. As part of this case study, the students also explored tradeoffs related to the communication interfaces, especially in the context of latency and bandwidth. At the interface between the controller and the Piezo Patches a Low Pass Filter followed by a Piezo Amplifier was used. The motion on the beam causes the retroreflector to move and thus, can be measured by the interferometer subsystem. For this case study, we also investigated the use of a common serial interface, via an ethernet controller, between the learning architecture on the FPGA and external modules. A simple packet, following the UDP protocol, was used to send one sample from the interferometer in the payload. Figure 6 shows an overview of the physical setup, networked system, and various communication interfaces.

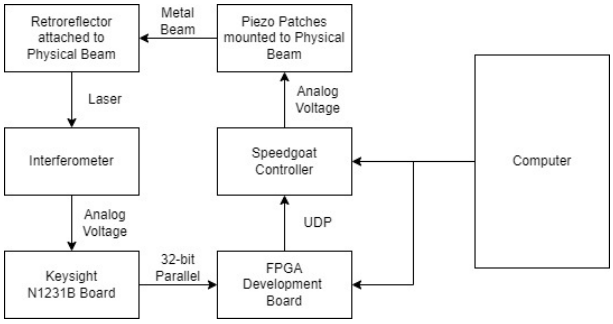

Fig. 6. High Level Block Diagram of Physical Setup.

There are two methods of data collection from the laser board itself: the PCI interface at the base of the board and four proprietary parallel interfaces on the top of the board. The interferometer allows data samples to be collected via the PCI interface. Attached to the PCI interface are a number of sample registers that can store either 16 or 32-bit position and velocity values. Data can then be pulled from these sample registers via a software command sent via the PCI bus. Although these sample registers can be updated at a theoretical rate of 4.75 ns, the data age of the sample registers relative to any given PCI sample request is 197 ns.

The interferometer provides a Windows and Linux driver and C API for setup, control, and data collection from the interferometer board. In Windows, the maximum theoretical usable sample collection time from measurement to data availability to the driver is 2.5 us.

In addition to the PCI interface, data samples can also be collected via a parallel hardware interface. All four of these interfaces are physically identical but are not all used in tandem. There are two data access modes associated with the interferometer hardware interfaces: direct mode and addressable mode.

Direct mode involves direct access to the position measurement for each individual axis. The key advantage of utilizing direct access mode is that all three axes can be sample simultaneously, with the obvious drawback being that a large number of parallel interface pins need to be dealt with. Addressable mode allows the user to sample each of the three axes from a single parallel interface header via a hardware multiplexer, but due to this a minimum of three cycles must pass for all axes to be sampled.

As part of the case study, the students pushed the limits of the software interface and also explored the use of the parallel hardware interface for sample collection. The parallel interface was connected to a learning architecture implemented the on FPGA.

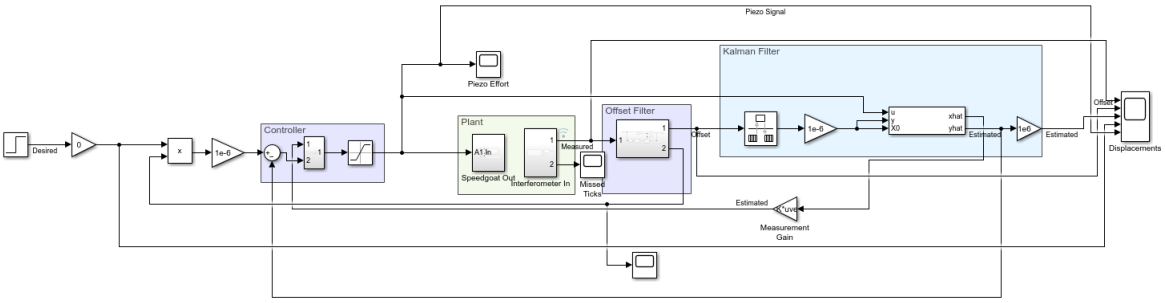

Fig. 7. Simulink model for exploring LQR control, exported to physical system.

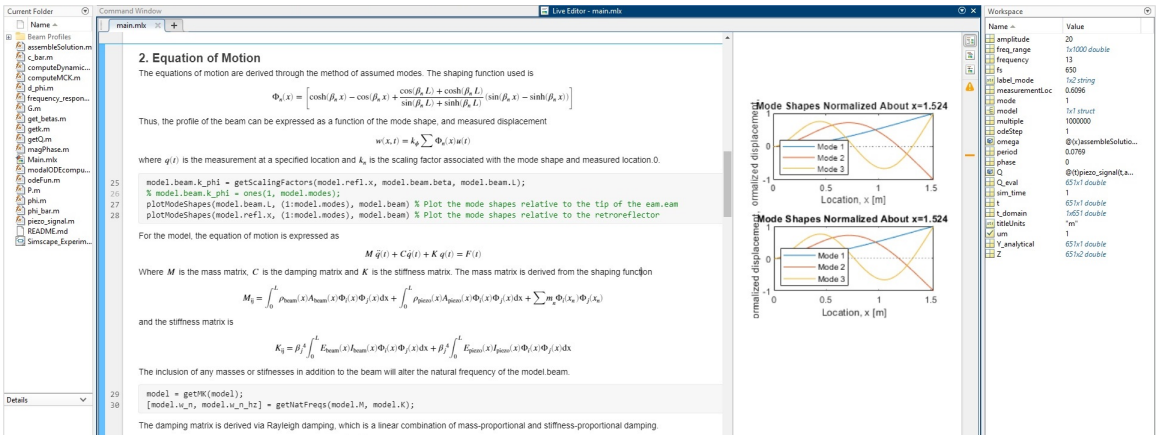

Fig. 8. Example Matlab Live notebook. Allows students to run experiments, code, and document results in one interactive location.

D. Application/Algorithm Tools

As described in Section III, high-performance control systems increasingly require as little overhead as possible when executing their algorithms and models. This performance constraints motivates the need to explore new cross-layer designs, and leverage low-level implementations, including bare-metal C implementations. Programming and ultimately optimizing these low level systems can be challenging for students approaching from control perspective, such as mechanical engineering students. As part of the case study we developed low level supporting modules and programming interfaces that exposed key functionality control for the students exploring the high-level control. To aid students in exploring the cross-layer design we also developed supporting modules for simulink, which allows for a easy learning curve and is also a toolset which mechanical engineering students are commonly already familiar with. Providing an interface for simulink allow for these students to quickly develop and study the control algorithms.

Simulink is a tool included within MATLAB that provides a graphics user interface (GUI) for modeling, simulating, and analyzing physical systems [18]. Systems are created within the application by placing, configuring, and connecting blocks that represent different portions of the physical system or algorithm being modeled. The model is then "built", by generating C code on the back-end, compiling that code, and then running the compiled application on the hardware target, whether that be the development PC or a remote target such as an embedded system. Figure 7 and 13 show examples of simulink models that were developed and investigate as part of this case study. In addition to the graphical tools used for designing the systems control, we also developed matlab live notebooks, which allow for code, diagrams, and documentation to be embedded in one interactive location. This tool provide an effective approach for students investigating new cross layer designs, as well as students new to the system, being able to quickly get experiments running and understand the different subsystems. Figure 8 shows an example of an interactive notebook used in the case study.

As described previously, students quickly ran into latency constraints when attempting to sample data from the measurement subsystem and run the control algorithms via simulink running on the host machine with a full operating system. From a control perspective, a minimum sampling rate of 500-1kHz from the laser interferometer subsystem was required in order to build a practical system identification and motion control application.

As part of the design space exploration, students also learned about different performance and timing tradeoffs when utilizing different flexible general purpose hardware and soft-

ware. For example, the students used the framework to develop a controller around the host machine (i.e. the development machine). Simulink was used to generate and export a C application. Simulink also has support for linking a C library and making function calls from that library from the Simulink model. The learning framework also includes a C library for accessing and configuring the interferometer, e.g. setting up and shutting down the board, pulling a sample from the position registers and transforming that sample into human readable units. These pre-generated modules can be compiled for Linux and Windows systems.

Introducing this library within the Simulink system model allowed the students to better understand timing constraints in the context of the control system. Students learned how tight timing restrictions of the physical system necessitated the use mechanisms to guarantee more deterministic and low latency timing, such as found with the Simulink Desktop Real-Time (SLDRT) software used for the model execution. The conventional approach for running the Simulink models on top of the full system, i.e. with OS kernel, resulted in prohibitive timing and dropped states. Students are attempted to use the SLDRT runtime for executing their models, which gives the tasks higher priority to compute and I/O resources. However, Windows system calls and precompiled binaries, both of which are being used by the learning architecture libraries, cannot be used in this execution mode. Therefore, most of the models needed to be run at the application level and the only specific I/O devices could be controlled at the kernel level. Overall, students were able to explore these constraints when using high level tools, such as simulink. Since the custom interferometer library cannot be run within the Simulink real-time kernel, it must be run atop the Windows kernel, meaning it must contend with other process for compute and I/O resources. As a result, performance suffers significantly, with sample rates on the host machine only able to achieve sample rates in the 10s of Hz, significantly below the rates what students learned would be required from application perspective.

E. Framework Results

This section describes example experimental results that were collected an analyzed in the context of the case study. Frequency Response. Any mechanical part has a certain natural frequency. For example, a ruler fixed onto the edge of a table will vibrate back and forth at a certain frequency when disturbed. A vehicle will bounce up and down at a certain frequency on uneven road. This experiment's purpose is to identify the frequency response of the metallic beam under test.

The first test was accomplished without the external controller, using only the FPGA and learning architecture as the central computing system for conducting the experiment. Attached to the FPGA is a digital to analog converter, and the analog output is sent to the piezo amplifier followed by the piezo patches of the beam. At the same time, the amplitude of the vibration is measured and recorded by the learning architecture implemented on the FPGA.

Using a USB-UART as a connection between the FPGA and a desktop computer, a sampling rate of 480Hz was achieved. A pseudocode of the instructions given to the learning architecture is as follow:

1) Send out a sinusoidal 3.3V signal at a specific frequency for 5 seconds, allowing the beam to reach steady-state.
2) Sample the interferometer data for 1 second.
3) Compute the amplitude by using the min and max of the sampled data.
4) Repeat with increasing frequency.

The resulting bode plot from the sine sweep is shown in Figure 9, with clear peaks at 11.5Hz (First Mode), 78.7Hz (Second Mode), 231.1Hz (Third Mode):

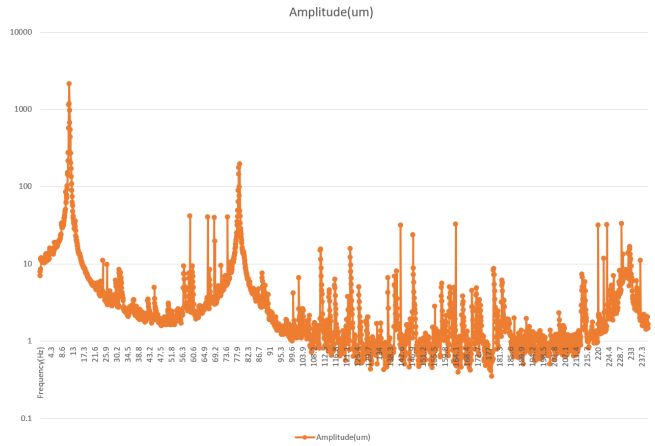

Fig. 9. Frequency Response Plot from FPGA

Students were then able to verify that these experimental results were comparable with the theoretical results, where the first, second and third modes are 13.8Hz, 86.5Hz and 244.2Hz. The 20.0%, 9.9% and 5.7% error were likely caused by the mass of the retroreflector mounted at the tip of the beam which is not accounted for in the theoretical model. Figure 10 shows the theoretical response as built from pre-built libraries included with Matlab.

Motion Control. The control for these applications can implemented using a variety of hardware and software combinations (e.g. Kalman filter, PID controller, NN-based, ensemble learning, analog, digital, general purpose, SoC, specialized acceleration logic, etc.). The design decisions for the control system will also exhibit various tradeoffs as described in Section III.

One of the configurations that the students explored is shown in Figure 11. This approach was developed using an external controller device and the USB-UAR interface, at a sampling rate of 480Hz, alongside an NIDAQ PCIe Card capable of generating analog voltage from a computer.

The desired outcome is that the beam actuate in a square wave, which cannot be accomplished by simply sending a square wave signal to the piezo patches, due to the intrinsic vibrations found in the metal beam. Figure 12 shows this

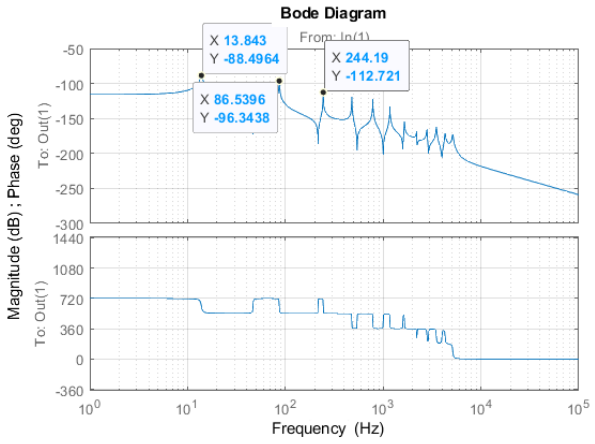

Fig. 10. Theoretical Frequency Response from Matlab

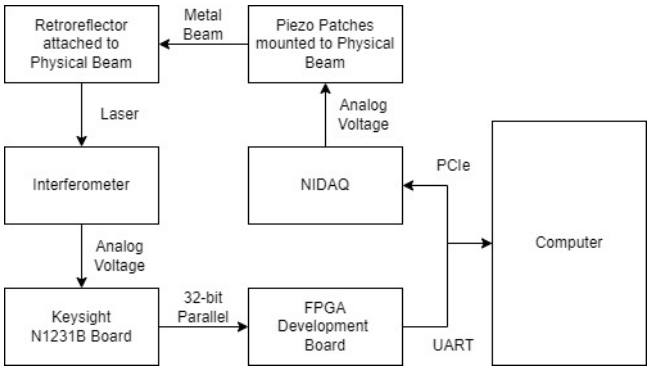

Fig. 11. Block Diagram of example System Architecture where closed-loop control is conducted with the aid of an external controller device.

response using the open-loop system, where the square wave voltage commands are sent to the patches directly.

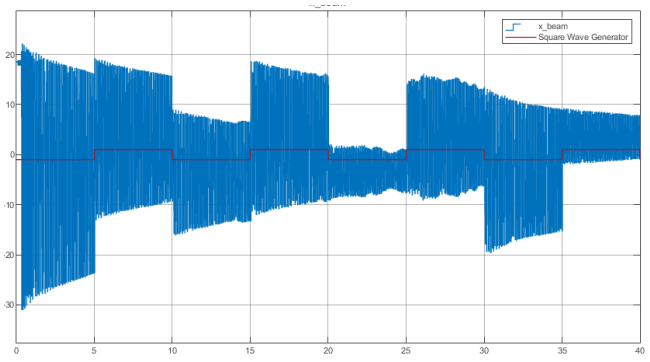

Fig. 12. Open Loop Response, where a square wave voltage is sent to the piezo patches directly, Y-axis = Displacement (um), X-axis = Time(s)

The students then utilized a simple PID controller, to dampen the response of the controlled square wave, shown in Figure 13. Once deployed, the PID parameters are continuously adjusted until a more accurate result is achieved. Figure 14 shows an example where the closed-loop response is achieved with lower average error.

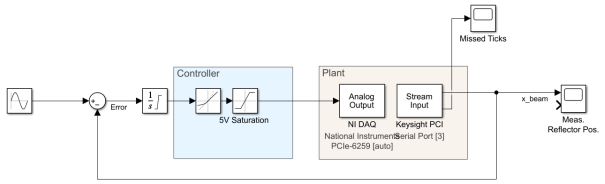

Fig. 13. Simulink Block Diagram for Closed Loop Control

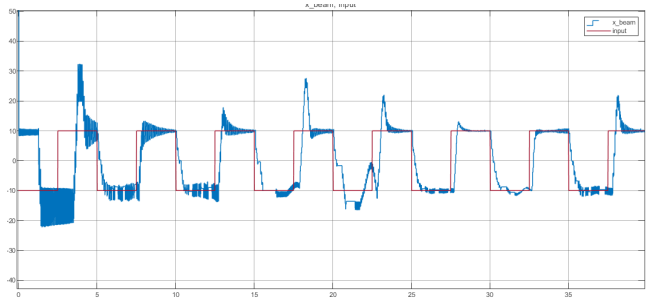

Fig. 14. An example of a recorded Closed Loop Response

Students observed that the imperfect time-domain response is affected by several factors. One of the primary causes was the limitations of the magnitude and non-deterministic nature of the latency exhibited by the host computer. From these tests students were also able to learn about the tradeoffs with timing guarantees and dynamic performance optimizations included in complex modern processor designs and modern operating systems.

V. Conclusion

This paper presents a multidisciplinary framework for using emerging computing systems in engineering education. Hardware and software systems are becoming highly data intensive, specialized, parallel, and heterogeneous. With these developments, new applications across a wide range of disciplines are being enabled, from agriculture, mechanical and control systems, and many other physical sciences. For this reason, new approaches for teaching, via hands-on experiences, with these systems is needed. In this paper, we propose a novel framework that can be used to support learning objectives for these systems, at the intersection of multiple disciplines (e.g. computer science, data science, mechanical/aerospace/industrial/computer engineering). Specifically, we describe a flexible computing platform, that supports the exploration of custom hardware and software, interfaces, communications, and integrated system design. The RISC-V based system-on-chip architecture and supporting software and programming tools provide a unique opportunity for students to quickly learn advanced computing concepts critical to future computing systems. The flexible learning architecture and associated tools can be used for different curriculum found across engineering disciplines that utilize project-based and active learning techniques. An implementation of the framework

is described in the context of an important class of control applications, system identification and motion control. As these systems scale, analyzing non-linear system dynamics requires non-traditional approaches, e.g. using modern learning-based models and tools. The proposed approach allows for students to explore the impact of computing abstractions, data-intensive system design, and the potential for cross layer optimizations, while also learning key concepts related to parallel programming models and architectures.

REFERENCES

[1] R. Bryant, M. D. Hill, T. Kazior, D. Lee, J. Liu, K. Nahrstedt, V. Narayanan, J. M. Rabaey, H. T. Siegelmann, N. R. Shanbhag, N. Verma, and H. P. Wong, "Nanotechnology-inspired information processing systems of the future," *CoRR*, vol. abs/2005.02434, 2020. [Online]. Available: https://arxiv.org/abs/2005.02434

[2] M. Čech, J. Königsmarková, M. Goubej, T. Oomen, and A. Visioli, "Essential challenges in motion control education," *IFAC-PapersOnLine*, vol. 52, no. 9, pp. 200–205, 2019, 12th IFAC Symposium on Advances in Control Education ACE 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896319305336

[3] S. Sohoni, K. Kearney, and R. Damron, "A platform for computer engineering education," 06 2014, pp. 24.87.1–24.87.14.

[4] P. Schaumont and I. Verbauwhede, "The exponential impact of creativity in computer engineering education," in *2013 IEEE International Conference on Microelectronic Systems Education (MSE)*, 2013, pp. 17–20.

[5] R. Figueiredoy, J. Fortesy, R. Eigenmanny, N. Kapadiay, S. Adabalay, V. Tayloryy, J. Miguel-alonsozz, L. Vidalz, and J.-j. Chenz, "Network-computer for computer architecture education: a progress report," 09 2001.

[6] Y. Li, J. Niu, S. Matutu, and Q. Qi, "Teaching practice reforms towards software-hardware collaboration in computer system ability training-taking fpga design course as an example," in *2021 IEEE Frontiers in Education Conference (FIE)*, 2021, pp. 1–5.

[7] N. M. Mwasaga and M. Joy, "Contextualizing micro high performance computing artifacts in higher education," in *2021 IEEE Frontiers in Education Conference (FIE)*, 2021, pp. 1–9.

[8] M. Soltys, "Cloudifying the curriculum with AWS," *CoRR*, vol. abs/2002.04020, 2020. [Online]. Available: https://arxiv.org/abs/2002.04020

[9] A. L. Fitch, H. H. C. Iu, and D. D. C. Lu, "An analog computer for electronic engineering education," *IEEE Transactions on Education*, vol. 54, no. 4, pp. 550–557, 2011.

[10] M. Boudreaux, S. Wang, S. Andrei, O. Urbina, and D. A. Sisk, "Integrating programming and engineering concepts using raspberry pi and scratch," in *2021 IEEE Frontiers in Education Conference (FIE)*, 2021, pp. 1–8.

[11] K. Tengler, O. Kastner-Hauler, and B. Sabitzer, "A robotics-based learning environment supporting computational thinking skills — design and development," in *2021 IEEE Frontiers in Education Conference (FIE)*, 2021, pp. 1–6.

[12] D. Copp, J. Isaacs, and J. Hespanha, "Programming, robotics, and control for high school students," *Advances in Engineering Education*, 06 2020.

[13] N. Houshangi, "Establishing a multidisciplinary control system laboratory," in *2007 Annual Conference & Exposition*, no. 10.18260/1-2–2456. Honolulu, Hawaii: ASEE Conferences, June 2007, https://strategy.asee.org/2456.

[14] H. Gurocak, A. Ater Kranov, and R. Williams, "Industrial motion control as a context for mechatronics education," 06 2012, pp. 25.772.1–25.772.9.

[15] A. Squires, J. Ph.D., D. Bodner, M. Okutsu, D. Ingold, P. Dominick, R. Reilly, W. Watson, and D. Gelosh, "Investigating an innovative approach for developing systems engineering curriculum: The systems engineering experience accelerator," 06 2011, pp. 22.970.1–22.970.22.

[16] T. McGrew, E. Schonauer, and P. Jamieson, "Framework and tools for undergraduates designing risc-v processors on an fpga in computer architecture education," in *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2019, pp. 778–781.

[17] "Speedgoat: Baseline Real-time target machine," https://www.speedgoat.com/products-services/speedgoat-real-time-target-machines/baseline, 2022, [Online; accessed 26-April-2022].

[18] S. Documentation, "Simulation and model-based design," 2020. [Online]. Available: https://www.mathworks.com/products/simulink.html